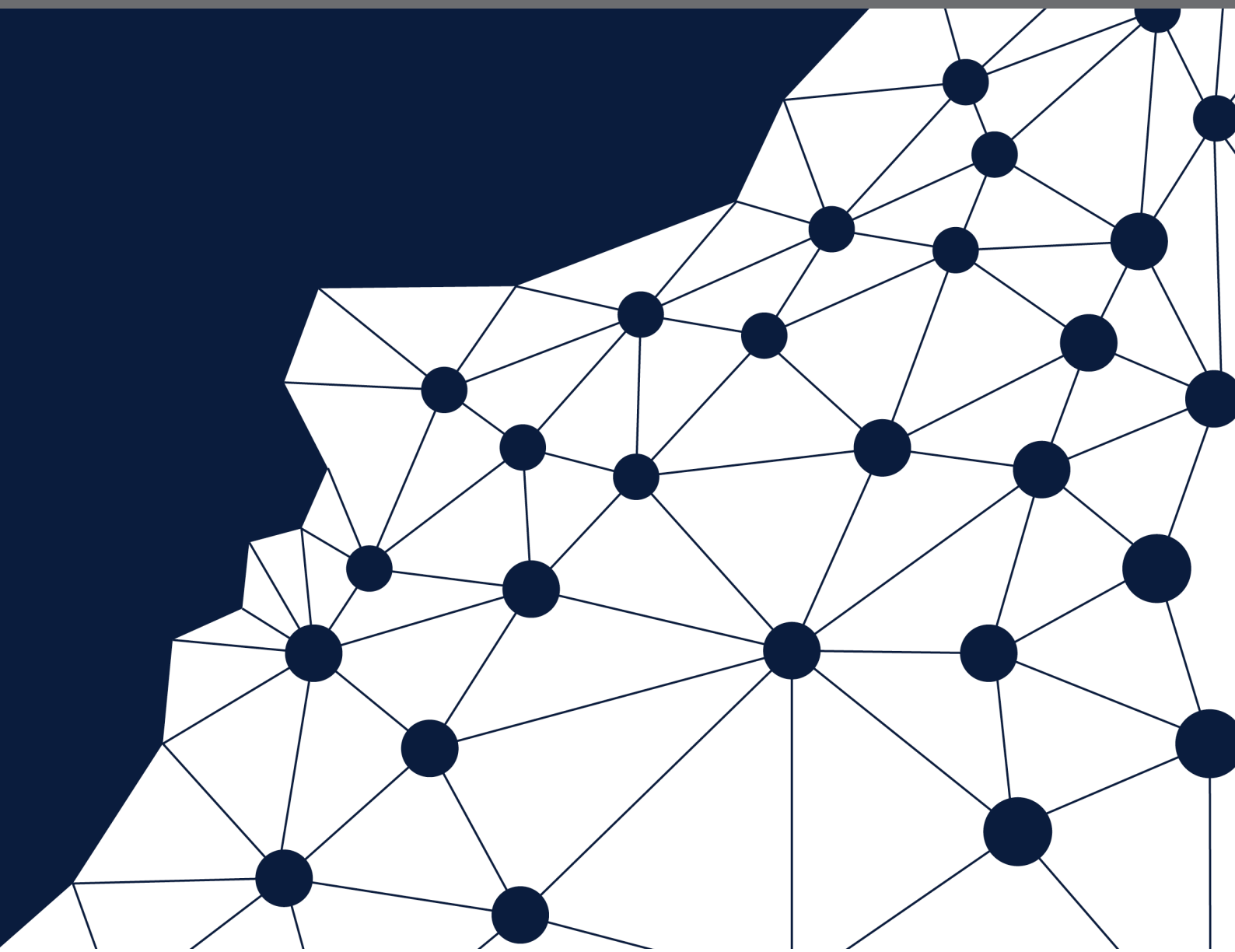
NEURAL COMPUTATION IN EMBODIED CLOSED-LOOP SYSTEMS FOR THE GENERATION OF COMPLEX BEHAVIOR: FROM BIOLOGY TO TECHNOLOGY

EDITED BY: Poramate Manoonpong and Christian Tetzlaff

PUBLISHED IN: Frontiers in Neurorobotics



frontiers Research Topics

Frontiers Copyright Statement

© Copyright 2007-2018 Frontiers Media SA. All rights reserved.

All content included on this site, such as text, graphics, logos, button icons, images, video/audio clips, downloads, data compilations and software, is the property of or is licensed to Frontiers Media SA ("Frontiers") or its licensees and/or subcontractors. The copyright in the text of individual articles is the property of their respective authors, subject to a license granted to Frontiers.

The compilation of articles constituting this e-book, wherever published, as well as the compilation of all other content on this site, is the exclusive property of Frontiers. For the conditions for downloading and copying of e-books from Frontiers' website, please see the Terms for Website Use. If purchasing Frontiers e-books from other websites or sources, the conditions of the website concerned apply.

Images and graphics not forming part of user-contributed materials may not be downloaded or copied without permission.

Individual articles may be downloaded and reproduced in accordance with the principles of the CC-BY licence subject to any copyright or other notices. They may not be re-sold as an e-book.

As author or other contributor you grant a CC-BY licence to others to reproduce your articles, including any graphics and third-party materials supplied by you, in accordance with the Conditions for Website Use and subject to any copyright notices which you include in connection with your articles and materials.

All copyright, and all rights therein, are protected by national and international copyright laws.

The above represents a summary only. For the full conditions see the Conditions for Authors and the Conditions for Website Use.

ISSN 1664-8714
ISBN 978-2-88945-605-5
DOI 10.3389/978-2-88945-605-5

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: researchtopics@frontiersin.org

NEURAL COMPUTATION IN EMBODIED CLOSED-LOOP SYSTEMS FOR THE GENERATION OF COMPLEX BEHAVIOR: FROM BIOLOGY TO TECHNOLOGY

Topic Editors:
Poramate Manoonpong, University of Southern Denmark, Denmark; Vidyasirimedhi Institute of Science and Technology, Thailand; Nanjing University of Aeronautics and Astronautics, China
Christian Tetzlaff, Georg-August-Universität Göttingen, Germany

Brain and Body by Poramate Manoonpong under CC-BY

How can neural and morphological computations be effectively combined and realized in embodied closed-loop systems (e.g., robots) such that they can become more like living creatures in their level of performance? Understanding this will lead to new technologies and a variety of applications.

To tackle this research question, here, we bring together experts from different fields (including Biology, Computational Neuroscience, Robotics, and Artificial Intelligence) to share their recent findings and ideas and to update our research community. This eBook collects 17 cutting edge research articles, covering neural and morphological computations as well as the transfer of results to real world applications, like prosthesis and orthosis control and neuromorphic hardware implementation.

Citation: Manoonpong, P., Tetzlaff, C., eds. (2018). Neural Computation in Embodied Closed-Loop Systems for the Generation of Complex Behavior: From Biology to Technology. Lausanne: Frontiers Media. doi: 10.3389/978-2-88945-605-5

Table of Contents

05 ***Editorial: Neural Computation in Embodied Closed-Loop Systems for the Generation of Complex Behavior: From Biology to Technology***
Poramate Manoonpong and Christian Tetzlaff

SECTION 1
EMBODIED CLOSED-LOOP SYSTEMS

10 ***Adaptive Control Strategies for Interlimb Coordination in Legged Robots: A Review***
Shinya Aoi, Poramate Manoonpong, Yuichi Ambe, Fumitoshi Matsuno and Florentin Wörgötter

31 ***A Minimal Model Describing Hexapedal Interlimb Coordination: The Tegotae-Based Approach***
Dai Owaki, Masashi Goda, Sakiko Miyazawa and Akio Ishiguro

44 ***Self-Organized Behavior Generation for Musculoskeletal Robots***
Ralf Der and Georg Martius

61 ***A Neural Dynamic Architecture for Reaching and Grasping Integrates Perception and Movement Generation and Enables On-Line Updating***
Guido Knips, Stephan K. U. Zibner, Hendrik Reimann and Gregor Schöner

75 ***Neurodynamics in the Sensorimotor Loop: Representing Behavior Relevant External Situations***
Frank Pasemann

SUBSECTION 1.1
SENSORY AREAS

87 ***An Adaptive Neural Mechanism for Acoustic Motion Perception With Varying Sparsity***
Danish Shaikh and Poramate Manoonpong

104 ***Real-Time Biologically Inspired Action Recognition From Key Poses Using a Neuromorphic Architecture***
Georg Layher, Tobias Brosch and Heiko Neumann

SUBSECTION 1.2
MOTOR AREAS

125 ***Fast Dynamical Coupling Enhances Frequency Adaptation of Oscillators for Robotic Locomotion Control***
Timo Nachstedt, Christian Tetzlaff and Poramate Manoonpong

139 ***Development and Training of a Neural Controller for Hind Leg Walking in a Dog Robot***
Alexander Hunt, Nicholas Szczecinski and Roger Quinn

SUBSECTION 1.3
HIGHER INTEGRATIVE AREAS

155 ***A Neurocomputational Model of Goal-Directed Navigation in Insect-Inspired Artificial Agents***
Dennis Goldschmidt, Poramate Manoonpong and Sakyasingha Dasgupta

172 ***Closed-loop Robots Driven by Short-Term Synaptic Plasticity: Emergent Explorative vs. Limit-Cycle Locomotion***
Laura Martin, Bulcsú Sándor and Claudius Gros

183 ***ReaCog, a Minimal Cognitive Controller Based on Recruitment of Reactive Systems***
Malte Schilling and Holk Cruse

206 ***Motor-Skill Learning in an Insect Inspired Neuro-Computational Control System***
Eleonora Arena, Paolo Arena, Roland Strauss and Luca Patané

SUBSECTION 1.4
BODY

223 ***Morphological Properties of Mass–Spring Networks for Optimal Locomotion Learning***
Gabriel Urbain, Jonas Degrave, Benonie Carette, Joni Dambre and Francis Wyffels

SECTION 2
TECHNOLOGY TRANSFER

236 ***Modular Neural Mechanisms for Gait Phase Tracking, Prediction, and Selection in Personalizable Knee-Ankle-Foot-Orthoses***
Jan-Matthias Braun, Florentin Wörgötter and Poramate Manoonpong

253 ***Biomechanical Reconstruction Using the Tacit Learning System: Intuitive Control of Prosthetic Hand Rotation***
Shintaro Oyama, Shingo Shimoda, Fady S. K. Alnajjar, Katsuyuki Iwatsuki, Minoru Hoshiyama, Hirotaka Tanaka and Hitoshi Hirata

260 ***Obstacle Avoidance and Target Acquisition for Robot Navigation Using a Mixed Signal Analog/Digital Neuromorphic Processing System***
Moritz B. Milde, Hermann Blum, Alexander Dietmüller, Dora Sumislawska, Jörg Conradt, Giacomo Indiveri and Yulia Sandamirskaya

frontiers in Neurorobotics

EDITORIAL
published: 30 August 2018
doi: 10.3389/fnbot.2018.00053

Editorial: Neural Computation in Embodied Closed-Loop Systems for the Generation of Complex Behavior: From Biology to Technology

Poramate Manoonpong[1,2,3]* *and Christian Tetzlaff*[4]

[1] *Embodied AI and Neurorobotics Lab, Centre for BioRobotics, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark,* [2] *Bio-inspired Robotics and Neural Engineering Lab, School of Information Science and Technology, Vidyasirimedhi Institute of Science and Technology, Rayong, Thailand,* [3] *Institute of Bio-Inspired Structure and Surface Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China,* [4] *Bernstein Center for Computational Neuroscience, Third Institute of Physics, Georg-August-Universität Göttingen, Göttingen, Germany*

Keywords: embodiment, neural computation, sensorimotor coordination, orthosis and prosthesis, motor planning/control, learning and memory, autonomous robot, neuromorphic computing

Editorial on the Research Topic

Neural Computation in Embodied Closed-Loop Systems for the Generation of Complex Behavior: From Biology to Technology

1. INTRODUCTION

The brain of biological organisms is a highly complex and very efficient computing unit. It can deal with a multitude of tasks from low-level sensorimotor coordination to high-level cognition. Specifically, it can process high-dimensional sensory information and, dependent on this, generate coordinated motor commands in real time, resulting in actions (like, locomotion and manipulation). Simultaneously, it can also perform cognitive functions (such as navigation, goal-oriented behavior, reasoning and decision making, interaction, communication). This amazing performance is achieved by using the full capacity of its neural dynamics, learning, memory, and adaptation as well as by interacting with the environment through its body (i.e., sensory-motor system). Thus, actions and cognition require dynamical brain-body-environment interactions and thereby cannot be disembodied. A traditional view of embodiment has also emphasized that complex behavior emerges from continuous and dynamical interactions between computational and physical means with the environment (Wilson, 2002; see also the embodiment scheme in Pfeifer et al., 2007). While this radical scientific concept has been promoted since the last three decades (Brooks, 1991; Chiel and Beer, 1997; Calvo and Gomila, 2008; Pfeifer et al., 2014), the detailed interaction of the (neural) computation within and across different brain areas, as the sensory, motor, and higher integrative areas, with the environment to generalize complex and adaptive behaviors have not been fully addressed.

According to this, this Research Topic called researchers from different fields (including Biology, Computational Neuroscience, Robotics, and Artificial Intelligence) to share their recent developments and results and to update our research community with remaining open questions. The topic has in total 17 articles which cover neural and morphological computations as well as the transfer of results to real world applications, like prosthesis and orthosis control and

OPEN ACCESS

Edited and reviewed by:
Florian Röhrbein, Technische Universität München, Germany

***Correspondence:**
Poramate Manoonpong
poma@mmmi.sdu.dk

Received: *14 June 2018*
Accepted: *09 August 2018*
Published: *30 August 2018*

Citation:
Manoonpong P and Tetzlaff C (2018) Editorial: Neural Computation in Embodied Closed-Loop Systems for the Generation of Complex Behavior: From Biology to Technology. Front. Neurorobot. 12:53. doi: 10.3389/fnbot.2018.00053

neuromorphic hardware implementation. Eight articles focus on the three main areas (sensory, motor, and integrative areas) of the controller or brain (**Figure 1**). Among these, two focus on neural computation mechanisms in sensory areas for action recognition of a human agent

(Layher et al.) and for acoustic motion perception (Shaikh and Manoonpong), twointegrative areas for motor-skill learning (Arena et al.), navigation learning (Goldschmidt et al.), motor planning and internal representations (Schilling and Cruse), and self-organized complex locomotion patterns (Martin et al.). In addition, two articles present neural closed-loop architectures that link between sensory and motor areas for reaching and grasping (Knips et al.) and for, e.g., obstacle avoidance behavior (Pasemann). Three articles consider a tight interaction between the body and the sensory and motor areas for sensorimotor coordination of legged robots (Aoi et al.; Owaki et al.) and a robot arm (Der and Martius). One article provides an insight on the computation of morphological body for optimal locomotion learning (Urbain et al.). Regarding to technology transfer, two articles show the transfer of the principles of the nervous system for orthosis (Braun et al.) and prosthesis control (Oyama et al.) and one shows the transfer to neuromorphic hardware-based control (Milde et al.). Based on these contributions, we organize subsections into two main categories: Embodied closed-loop systems and their technology transfer.

2. OVERVIEW

2.1. Embodied Closed-Loop Systems

An embodied closed-loop system or a brain-body-environment system (Chiel and Beer, 1997) is generally formed by three main ingredients: Nervous system (or controller), body, and the environment. The nervous system has in general three subareas: Sensory, motor, and higher integrative areas. Environmental information is perceived through sensors and processed in the sensory areas. The sensory areas can be also influenced by forward internal models (Kawato, 1999) embedded in the higher integrative areas for sensory prediction and noise cancellation (von Holst and Mittelstaedt, 1950; Blakemore et al., 1999) as well as state estimations (Frens and Donchin). The outputs of the sensory areas are transmitted to motor and higher integrative areas. The motor areas translate the sensory information into motor commands to control the body. They also send a copy of motor commands (efference copy) to the forward models. The

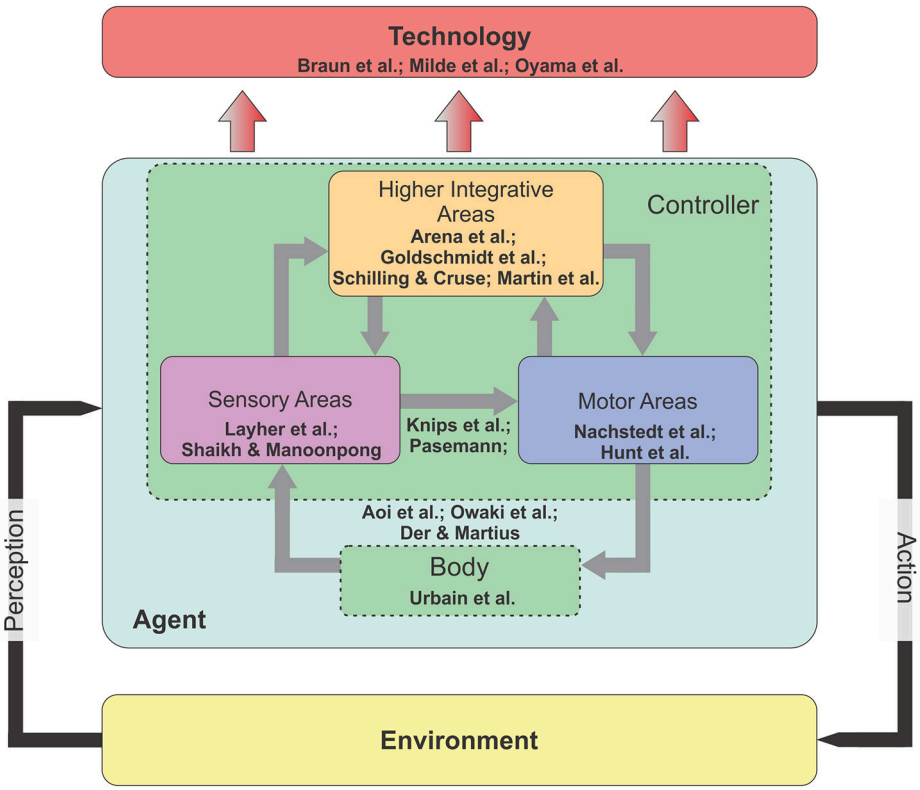


FIGURE 1 | The diagram of an embodied closed-loop system. The system concerns an agent that is situated in the environment. It can perceive the environmental information through its sensors and perform its actions using its motors. In principle, the agent consists of two main components: Nervous system (or controller) and body. There are three main areas inside the nervous system: Sensory, motor, and higher integrative areas. In the embodied perspective, while all external and internal stimuli are processed in the nervous system, this computational processing can be offloaded to the body (i.e., morphological computation) for a successful interaction with the environment (see text for more details).

higher integrative areas have multiple complex functions with multiple-time scales plasticity (i.e., long-term and short-term synaptic plasticity) for motor-skill learning, navigation learning, motor planning, and internal representations. Their outputs project to both sensory and motor areas. Through these internal (inside the controller) and external (between the body and the environment) interactions in the closed-loop system, adaptive behaviors emerge.

Regarding interactions in embodied closed-loop systems for adaptive behavior generation, in the Research Topic, Aoi et al. review various adaptive locomotor behaviors that emerge from the interactions between the body dynamics, the nervous system, and the environment in animals and legged robots. They identify key factors and mechanisms for adaptations to different speeds, environmental situations, body properties, and tasks. The factors and mechanisms include CPGs, sensory feedbacks, forward model, learning model, and muscle stiffness. Owaki et al. present a novel and minimal Tegotae-based approach, that exploits the interactions between foot contact sensor signals, body dynamics, and the environment for adaptive interlimb coordination of a hexapod robot. The approach can generate various insects' gait patterns that allows the robot to adapt to different locomotion speeds, changes in the weight distribution, and leg amputation. Der and Martius report self-organized behavior of an anthropomorphic musculoskeletal robot arm. The behavior emerges from the interaction between a self-learning neural controller (nervous system), the elastically musculoskeletal arm (body), and an object (environment) through proprioceptive sensory feedback. Through the agent-environment coupling, the robot can perform handshaking, pendulum swinging, bottle shaking, rotating a wheel, wiping a table, and hand-eye coordination.

As a part of embodied closed-loop systems, dynamical system-based architectures that link between sensory and motor areas are introduced in the Research Topic. Knips et al. present a neural dynamic architecture for reaching and grasping objects. It integrates several modules, having functions for scene representation, concurrent object classification and pose estimation, behavioral organization, and movement generation, into a large dynamical system of an anthropomorphic robot arm equipped with a Kinect sensor. In addition to the perception, integration, and movement generation, the architecture can also allow for the online adaptation of the performed movement of the robot for successful completion of the grasp. Pasemann proposes the exploitation of the discrete-time neurodynamics of networks in a sensorimotor loop for generating adaptive behavior, like adaptive obstacle avoidance of mobile robots. The behavior generation is a result from a projection of attractor transients or meta-transients, embedded in neurodynamics, to the motor space.

2.1.1. Sensory Areas

In a closed-loop scenario, agents have to extract and process relevant information from the environment, they are situated in. For this, the initial step is to perceive at least parts of the environment via the sensory system. Thereby, the sensory modality can be multifaceted and requires various types of sensors in the system, as for visual, touch, or sound perception. As the next step, the sensory system has to preprocess the perceived environmental information to support the computation done in subsequent areas. Based on experimental insights from the lizard auditory system (Fletcher and Thwaites, 1979; Christensen-Dalsgaard and Manley, 2005), Shaikh and Manoonpong developed a model of the auditory system that can learn to perceive a sound, and to process the information to localize its source and to estimate the speed and direction of the motion of the source. Different to previous approaches, the model can track the source, although it is occluded for a certain duration. By using the model for the auditory system of a wheeled robot, the robot was always able to perceive, to locate, and, by its sensor-motor interaction, to face the sound source.

The extraction of more abstract but complex concepts from the environmental stimulus stream requires, in general, a larger and more sophisticated sensory system. Layher et al. trained a deep neural network to recognize human poses from a stream of images. The pose recognition is based only on features of human body motions and shapes not requiring feedback from higher integrative areas. By implementing the network on a neuromorphic hardware, the recognition process becomes real time with about 1,000 frames per second. Remarkably, the system already shows indications of generalization of poses.

2.1.2. Motor Areas

Central pattern generators (CPGs) have been identified as one of key mechanisms in the motor areas particulary for locomotion control. The principle of biological CPGs has been widely used for robot locomotion control (Ijspeert, 2008). Although CPGs do not need any external input or feedback to produce basic rhythmic activity, they still require sensory feedback to adapt and tune their produced activity, e.g., their frequency or phase. Reactive and adaptive mechanisms have been introduced for this purpose (Buchli et al., 2006). A reactive mechanism can entrain the CPG signal where the frequency of the signal matches to sensory feedback. However, if the feedback disappears, the CPG signal will return to its intrinsic frequency. By contrast, an adaptive mechanism modifies the intrinsic frequency of the CPG permanently. Here, Nachstedt et al. propose a novel frequency adaptation mechanism through fast dynamical coupling (AFDC) of a CPG model. It is an extension of the standard frequency adaptation mechanism (Righetti et al., 2009) and based on dynamically adapting the coupling strength of sensory feedback to a CPG model. Using this AFDC technique, they achieve fast and precise adaptation for a wide range of initial intrinsic and target frequencies without the need for parameter fine tuning.

Hunt et al. report a CPG-based motor control circuit with sensory feedback and an automatic process for neural parameter setting. It is based on the known connectivity of mammalian locomotor systems. The process, faster and more reliable than manual tuning, can tune neural parameters to generate adaptive locomotion in the rear legs of a dog-like robot driven by artificial muscles. Using the CPG-based control approach, they show that the dog-like robot can adapt its stepping continuously and maintains rhythmic walking.

2.1.3. Higher Integrative Areas

Different to sensory and motor areas, higher integrative areas are associated with cognitive processes as learning, planning, navigation, or generalization. For instance, Goldschmidt et al. developed a system, which learned in a reward-based manner to represent the path an agent has walked. By this, the agent is able to robustly localize itself within the environment and to find back to the home position. Thereby, the resulting behaviors are quite similar to behaviors of insects as desert ants.

Interestingly, the neuronal network underlying cognitive processes can be quite small by using the computational resources of other areas or by utilizing the temporal dynamics of the system (Buonomano and Maass, 2009). Martin et al. use the ongoing dynamics of short-term synaptic plasticity (Tsodyks and Markram, 1997) in a three-neuron system to switch between different, complex motor patterns. Here, Schilling and Cruse show that already a small neuronal network is sufficient for successful planning within an environment and generalization to other environments, if the system recruits and orders the resources of the downstream motor areas. In other words, the small network reorders diverse reactive behaviors, each stored in a different part of the motor area, to adapt according to new environments.

There is a clear indication that higher integrative areas are not mandatory for cognitive processes (Cruse and Wehner, 2011). Arena et al. show in a theoretical model that learning within the Drosophila mushroom body, which is in general associated with the sensory processing of olfactory inputs, adapts motor commands or primitives in the motor area. Thus, by changes in the sensory area, the sensory-motor relations are updated yielding new behaviors. This was demonstrated on a six-legged robot, which can learn by this mechanism to climb up stairs. The authors also address the role of Neural Reuse as one of the possible keys for the emergence of complex behaviors in simple brains.

2.1.4. Body

Apart from neural computation in the nervous system, morphological computation also contributes to the generation of complex behavior. Morphological computation considers that certain processes can be performed by the body instead of the nervous system (Pfeifer and Bongard, 2006). In other words, it captures conceptually the observation that biological systems utilize their flexible and compliant morphology to conduct computations required for a successful interaction with their complex environments. There are numerous illustrative applications of morphological computation and embodiment for efficient locomotion in biological systems (Dickinson et al., 2000) and artificial systems (McGeer, 1990; Jayaram and Full, 2016; Manoonpong et al., 2016). Here, Urbain et al. present an analysis of the trade-offs between morphology, efficiency of locomotion, and the ability of a mechanical body. This is done by using a detailed dynamical model of a Mass-Spring-Damper (MSD) network. They also analyze the computational capacity of a MSD body to generate motor control signals and integrate the signals as feedback to a locomotion controller.

2.2. Technology Transfer

Analyzing the neural computation in closed-loop systems, on the one hand, yields insights of the underlying neural dynamics and principles and, on the other hand, provides new solutions for technological control problems. Braun et al. developed a neural controller which tracks and predicts the gait of a patient to control the gait-supporting knee-ankle-foot orthosis. This controller is independent of the actual environmental situation, as walking on a flat terrain or climbing stairs, and requires a minimal feedback from the patient.

Based on adaptive principles in neural circuits, Oyama et al. developed an adaptive controller for a hand prosthesis. A standard controller requires the user to interfere to avoid errors given in different environmental conditions. By contrast, the adaptive controller self-adapts according to the new environmental state or different hand poses.

Milde et al. transferred the neural controller and the whole neural sensory processing onto neuromorphic hardware. By implementing this hardware architecture, they developed an autonomous, neuromorphic robotic agent, which is able to avoid obstacles and to acquire targets. Due to the neural nature of the controller, the agent behaves robustly according to unexpected changes in the environment.

3. CONCLUSION

The Research Topic presents an embodied closed-loop approach that considers the interaction of (neural) computation across sensory, motor, and higher integrative areas with the agent's body and the environment. The studies in this Topic cover the broad spectrum of this approach and show that, indeed, complex behaviors emerge from the interplay between different parts of an agent. Thereby, the majority of these studies focus on the interplay between a subset of the available parts. The results from these studies confirm that the embodied approach can be a powerful method to develop autonomous robotic agents performing complex behaviors and it can even be a key to understand high-level cognition. Given these and further studies, it is now possible to address the interaction between all parts of an agent's controller (brain), body, and the environment.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

We thank all authors contributing with their work to this Research Topic.

We acknowledge funding by the H2020-FETPROACT project Plan4Act (no. 732266). PM acknowledges support from the

Human Frontier Science Program under grant agreement no. RGP0002/2017, from Vidyasirimedhi Institute of Science and Technology (VISTEC)-research funding (Thailand) on Bio-inspired Robotics, and from the Thousand Talents program of China. CT acknowledges support provided by the SFB 1286 from the German Research Foundation (DFG).

REFERENCES

Blakemore, S.-J., Wolpert, D. M., and Frith, C. D. (1999). The cerebellum contributes to somatosensory cortical activity during self-produced tactile stimulation. *NeuroImage* 10, 448–459. doi: 10.1006/nimg.1999.0478

Brooks, R. (1991). New approaches to robotics. *Science* 253, 1227–1232. doi: 10.1126/science.253.5025.1227

Buchli, J., Righetti, L., and Ijspeert, A. J. (2006). Engineering entrainment and adaptation in limit cycle systems. *Biol. Cybern.* 95:645. doi: 10.1007/s00422-006-0128-y

Buonomano, D. V., and Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nat. Rev. Neurosci.* 10, 113–125. doi: 10.1038/nrn2558

Calvo, P., and Gomila, T. (2008). *Handbook of Cognitive Science: An Embodied Approach*. San Diego, CA: Elsevier.

Chiel, H. J., and Beer, R. D. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends Neurosci.* 20, 553–557. doi: 10.1016/S0166-2236(97)01149-1

Christensen-Dalsgaard, J., and Manley, G. A. (2005). Directionality of the lizard ear. *J. Exp. Biol.* 208, 1209–1217. doi: 10.1242/jeb.01511

Cruse, H., and Wehner, R. (2011). No need for a cognitive map: decentralized memory for insect navigation. *PLoS Comput. Biol.* 7:e1002009. doi: 10.1371/journal.pcbi.1002009

Dickinson, M. H., Farley, C. T., Full, R. J., Koehl, M., Kram, R., and Lehman, S. (2000). How animals move: an integrative view. *Science* 288, 100–106. doi: 10.1126/science.288.5463.100

Fletcher, N. H., and Thwaites, S. (1979). Physical models for the analysis of acoustical systems in biology. *Q. Rev. Biophys.* 12, 25–65. doi: 10.1017/S0033583500002596

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Jayaram, K., and Full, R. J. (2016). Cockroaches traverse crevices, crawl rapidly in confined spaces, and inspire a soft, legged robot. *Proc. Natl. Acad. Sci. U.S.A.* 113, E950–E957. doi: 10.1073/pnas.1514591113

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Curr. Opin. Neurobiol.* 9, 718–727. doi: 10.1016/S0959-4388(99)00028-8

Manoonpong, P., Petersen, D., Kovalev, A., Wörgötter, F., Gorb, S. N., Spinner, M., et al. (2016). Enhanced locomotion efficiency of a bio-inspired walking robot using contact surfaces with frictional anisotropy. *Sci. Rep.* 6:39455. doi: 10.1038/srep39455

McGeer, T. (1990). "Passive walking with knees," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on* (IEEE) (Cincinnati, OH), 1640–1645.

Pfeifer, R., and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. Cambridge: MIT Press.

Pfeifer, R., Iida, F., and Lungarella, M. (2014). Cognition from the bottom up: on biological inspiration, body morphology, and soft materials. *Trends Cogn. Sci.* 18, 404–413. doi: 10.1016/j.tics.2014.04.004

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Righetti, L., Buchli, J., and Ijspeert, A. (2009). Adaptive frequency oscillators and applications. *Open Cybern. Syst. J.* 3, 64–69. doi: 10.2174/1874110X00903010064

Tsodyks, M., and Markram, H. (1997). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proc. Natl. Acad. Sci. U.S.A.* 94, 719–723. doi: 10.1073/pnas.94.2.719

von Holst, E., and Mittelstaedt, H. (1950). Das reafferenzprinzip. *Naturwissenschaften* 37, 464–476. doi: 10.1007/BF00622503

Wilson, M. (2002). Six views of embodied cognition. *Psychon. Bull. Rev.* 9, 625–636. doi: 10.3758/BF03196322

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Manoonpong and Tetzlaff. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

REVIEW
published: 23 August 2017
doi: 10.3389/fnbot.2017.00039

Adaptive Control Strategies for Interlimb Coordination in Legged Robots: A Review

Shinya Aoi[1*†], *Poramate Manoonpong*[2*†], *Yuichi Ambe*[3], *Fumitoshi Matsuno*[4] *and Florentin Wörgötter*[5]

[1] Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University, Kyoto, Japan, [2] Embodied AI & Neurorobotics Lab, Centre for Biorobotics, Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark, [3] Department of Applied Information Sciences, Graduate School of Information Sciences, Tohoku University, Aoba-ku, Japan, [4] Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University, Kyoto, Japan, [5] Bernstein Center for Computational Neuroscience, Third Institute of Physics, Georg-August-Universität Göttingen, Göttingen, Germany

Walking animals produce adaptive interlimb coordination during locomotion in accordance with their situation. Interlimb coordination is generated through the dynamic interactions of the neural system, the musculoskeletal system, and the environment, although the underlying mechanisms remain unclear. Recently, investigations of the adaptation mechanisms of living beings have attracted attention, and bio-inspired control systems based on neurophysiological findings regarding sensorimotor interactions are being developed for legged robots. In this review, we introduce adaptive interlimb coordination for legged robots induced by various factors (locomotion speed, environmental situation, body properties, and task). In addition, we show characteristic properties of adaptive interlimb coordination, such as gait hysteresis and different time-scale adaptations. We also discuss the underlying mechanisms and control strategies to achieve adaptive interlimb coordination and the design principle for the control system of legged robots.

Keywords: legged robot, interlimb coordination, adaptation, sensorimotor interaction, central pattern generator

OPEN ACCESS

Edited by:
Tom Ziemke,
University of Skövde and Linköping University, Sweden

Reviewed by:
Auke Ijspeert,
École Polytechnique Fédérale de Lausanne, Switzerland
Holk Cruse,
Bielefeld University, Germany

****Correspondence:***
Shinya Aoi
shinya_aoi@kuaero.kyoto-u.ac.jp
Poramate Manoonpong
poma@mmmi.sdu.dk

† These authors have contributed equally to this work.

Received: *26 November 2016*
Accepted: *31 July 2017*
Published: *23 August 2017*

Citation:
Aoi S, Manoonpong P, Ambe Y, Matsuno F and Wörgötter F (2017) Adaptive Control Strategies for Interlimb Coordination in Legged Robots: A Review. Front. Neurorobot. 11:39. doi: 10.3389/fnbot.2017.00039

1. INTRODUCTION

Animals produce adaptive motor behaviors by skillfully manipulating their complicated and redundant musculoskeletal systems. Locomotion is an important behavior required in daily life. Gait selection in accordance with the situation, such as speed and environment, is a prominent adaptive motor function. Humans walk bipedally and use walking and running gaits. Quadruped animals use four legs and produce walking, trotting, and galloping gaits. Hexapod insects use six legs and create metachronal (wave), tetrapod, and tripod gaits as well as intermediate stepping patterns forming a continuum. These gaits, including the transitions and the intermediate stepping patterns for hexapods, are generated through the intralimb and interlimb coordination of leg movements. Intralimb coordination is the relationship between segments or joints within one leg, whereas interlimb coordination is the relationship between legs. For example, in the adaptive control of intralimb coordination, peak timings of ankle plantar flexion, knee extension, and hip extension are out of phase during the human walking gait, but they are shifted and almost in phase during the human running gait (Diedrich et al., 1998). In the adaptive control of interlimb

coordination, the footfall sequence between legs changes (Muybridge, 1957), and the sequence is mainly explained by the relative phases between leg movements, because the leg movements are periodic with almost the same period for each leg (note that different frequencies between the legs have been sometimes observed in insects due to the flexibility in the stepping patterns, Pearson and Franklin, 1984). In the quadrupedal walking gait, although the left and right legs move in anti-phase, the ipsilateral front and hind legs do not. In contrast, in the quadrupedal trotting gait, the ipsilateral front and hind legs as well as the left and right legs move in anti-phase; that is, the diagonal legs move in phase (Hildebrand, 1965). Measured data analyses performed to clarify the gait mechanisms have suggested that gaits are selected based on metabolic and biomechanical factors (Margaria, 1938; Hoyt and Taylor, 1981; Farley and Taylor, 1991). However, reports on the roles of these factors in determining the gaits (Hreljac, 1993; Minetti et al., 1994; Raynor et al., 2002; Wickler et al., 2003) are conflicting, and so the underlying mechanism remains unclear.

To elucidate adaptive motor functions in animals, neurophysiological and biomechanical studies have been independently conducted. Neurophysiological studies mainly investigate the configurations and activities of the neural system, whereas biomechanical studies generally examine the functional roles of the musculoskeletal system. However, locomotion is generated through dynamic interactions among the neural system, the musculoskeletal system, and the environment. It is thus difficult to fully analyze the locomotion mechanism from a single perspective. In addition, gaits are viewed as self-organized patterns in such complex dynamical systems (Schöner et al., 1990; Diedrich et al., 1998; Griffin et al., 2004; Schilling et al., 2013a). The stability structure of gaits has been identified from the response of perturbations, especially by phase oscillators and phase response curves (Couzin-Fuchs et al., 2015; Funato et al., 2016) based on the phase reduction theory (Kuramoto, 1984). However, it is difficult to understand how the stability structure is generated due to the complex nature of interactions between the dynamic factors in locomotion. To fully elucidate the locomotion mechanism, integrated studies of neural and musculoskeletal systems are required to find the processes that create adaptive locomotor behavior.

Recently, to reveal the locomotion mechanism, legged robots have attracted attention. A robot's mechanical system with actuators, such as electric motors and pneumatic and hydraulic actuators, has been used to investigate the dynamic role of the musculoskeletal system in locomotion. The control system of the robot has been developed based on neurophysiological findings and employs various sensors, such as a touch sensor, load cell, acceleration sensor, gyro sensor, laser range scanner, and vision system. This approach allows us to emulate and investigate gait generation through dynamic interactions between the neural system, the musculoskeletal system, and the environment. In particular, central pattern generators (CPGs), which are located in the spinal cord of vertebrates and in the thoracic ganglia of invertebrates, are an important factor for elucidating the locomotion mechanism (Grillner, 1975; Orlovsky et al., 1999; MacKay-Lyons, 2002) and have aided the development of locomotion control systems of legged robots. A CPG is a group of interconnected neurons that can be activated to generate a motor pattern without the requirement of sensory feedback. The evidence that supports this hypothesis was originally shown by Brown (1911). In addition to the open-loop control function, CPGs receive sensory feedbacks to modulate motor commands. This closed-loop structure of sensory feedbacks is crucial to achieve adaptive behavior depending on the situation. Various CPG models have been proposed by using neural or oscillator networks and implemented in control legged robots [see review by Ijspeert (2008)]. For example, Taga and Shimizu (1991) and Taga (1995) conducted a pioneering study of a CPG model for human bipedal locomotion. They employed an articulated multi-link system for the body mechanical model and neural oscillators developed by Matsuoka (1985) for the CPG model. This CPG model received sensory signals of local and global information for locomotion. They demonstrated that adaptive locomotion is established through the interaction between body dynamics, oscillator dynamics, and environment; they called this "global entrainment." Although complex and robust locomotion behavior can be achieved by purely reflexive control mechanisms (Cruse et al., 1998; Manoonpong et al., 2007; Lewinger and Quinn, 2011; Schilling et al., 2013a,b) and classical machine learning control (Bongard et al., 2006; Cully et al., 2015) instead of using CPG models, the CPG concept and modeling have had a large influence on the studies of legged robots.

In this review, we focus on the adaptive control of interlimb coordination in locomotion. We introduce adaptive interlimb coordination for animals and legged robots induced by various factors (locomotion speed, environmental situation, body properties, and task). In addition, we show characteristic properties of adaptive interlimb coordination in animals and robots, such as gait hysteresis and different time-scale adaptations. Finally, we discuss the underlying mechanisms and control strategies to achieve adaptive interlimb coordination and the design principle for the control system of legged robots.

2. ADAPTIVE INTERLIMB COORDINATION IN ANIMALS AND ROBOTS

2.1. Speed-Dependent Adaptation

The most general adaptive interlimb coordination appears when varying the locomotion speed in legged animals. This has been observed even in spinal cats on treadmills (Forssberg and Grillner, 1973; Orlovsky et al., 1999), in which the phase relationship between the legs changes and the gait varies among walking, trotting, and galloping. In reported studies, the spinal cords of cats were transected from the brain, but they still received sensory feedback through the contact between their feet and the belt. The sensory signals changed with the belt speed change, which induced their gait transitions. This result highlights the important contribution of sensorimotor interaction to adaptive interlimb coordination. Quadruped robots have achieved adaptive interlimb coordination that depends on locomotion speed by modeling spinal CPGs with local sensory feedback (Maufroy et al., 2010; Aoi et al., 2011,

2013b; Owaki et al., 2013; Fukuoka et al., 2015; Owaki and Ishiguro, 2017). This can be seen in the following examples. **Figure 1** shows a quadruped robot, the control system, and the experimental results of the walk–trot transition in Aoi et al. (2013b) (this robot showed hysteresis in the gait transition, as discussed in Section 3.1). **Figure 2**, which is from the work by Fukuoka et al. (2015), presents quadruped gaits transitioned from a walk at slow speeds to a trot at medium speeds, and a transverse gallop at high speeds. **Figure 3**, which is from the work by Owaki and Ishiguro (2017), also shows spontaneous gait transitions from a lateral-sequence (L-S) walk to a trot and even to a gallop of a quadruped robot with respect to the locomotion speed without neural coupling. These robotics studies used simple neural oscillators or phase oscillators for the CPG model and produced leg motions from the oscillator phases. More specifically, one oscillator created one leg motion and the phase relationship between the oscillators determined the gait. Each oscillator phase was regulated through local sensory information of the leg, such as foot contact and leg loading, occurring only within one leg.

As an important control architecture in these robotics studies, the phase relationship between the oscillators was not predefined and the oscillators were only weakly coupled or decoupled. That is, the gait was not determined by the oscillator dynamics using strong coupling (Schöner et al., 1990; Canavier et al., 1997; Ito et al., 1998; Golubitsky et al., 1999), but by the interaction between whole-body dynamics and oscillator dynamics through local sensory feedback. The interlimb coordination was generated only in a self-organizing manner among the neural dynamics, the body dynamics, and the environment.

Similar adaptive interlimb coordination in accordance with gait speed also appears in hexapod insects, such as stick insects (Wilson, 1966; Graham, 1972; Cruse, 1990; Grabowska et al., 2012), cockroaches (Hughes, 1952; Delcomyn, 1971; Pearson, 1976; Bender et al., 2011), and flies (Strauß and Heisenberg, 1990; Wosnitza et al., 2013; Berendes et al., 2016). In particular, stick insects and flies smoothly change their interlimb coordination in accordance with gait speed (Wilson, 1966; Graham, 1972; Wosnitza et al., 2013). More specifically, the relative phases between the legs continuously change in a linear fashion for gait speed. This is similar to some mammals, including sheep, but is different from other mammals, including dogs. In mammals such as dogs, the gait transitions have relative leg phases that change suddenly in a sigmoid fashion (Alexander and Jayes, 1983). Although it is suggested that cockroaches achieve interlimb coordination mainly by the CPG itself (Fuchs et al., 2011), the CPG by itself does not produce a coordinated motor pattern for stick insect walking, because sensory feedback is important (Bässler and Wegner, 1983; Büschges et al., 1995; Büschges et al., 2008). Cruse and his colleagues proposed an artificial neural network, named Walknet, which controls leg movements based on six different rules to regulate interlimb coordination by sensory information (note that the controller of the individual leg operates without CPG). The rules were empirically derived from the behavioral experiments of stick insects [see reviews by Cruse et al. (1998), Dürr et al. (2004) and Schilling et al. (2013a)]. Three of the rules were designed by disturbing leg movements on a slippery surface. The rules changed the cycle duration of a leg based on sensory information of the neighboring legs. As a result of sensorimotor interaction, the insect models controlled by Walknet produced a continuum of locomotion patterns, such as tripod, tetrapod, and wave gaits, and intermediate stepping patterns, as observed in stick insects. In addition, the models were used for various situations, such as walking on uneven surfaces (Kindermann, 2002), leg amputation (Schilling et al., 2007), negotiating curves (Schilling et al., 2013b), and climbing over large gaps (Bläsing, 2006), and the locomotor behavior was comparable to that of stick insects. Tóth and Daun-Gruhn (2016) developed neural network models based on Hodgkin Huxley dynamics and integrated them with musculoskeletal models to explain the interlimb coordination mechanism of insects. Although their models did not produce intermediate stepping patterns as observed in stick insects (Wilson, 1966; Graham, 1972) and flies (Wosnitza et al., 2013), their results suggest that the connection between the levator-depressor neuromuscular systems of the different legs is necessary to replicate the primary features of tripod and tetrapod gaits. Ambe et al. (2013, 2015) used simple phase oscillators with local sensory feedback of foot contact information for a hexapod robot, in a manner similar to the quadruped robots mentioned above. They produced a continuum of locomotion patterns, such as metachronal and tripod gaits and intermediate stepping patterns, through embodied sensorimotor interaction, without predefining the patterns in accordance with the locomotion speed. In addition, one important aspect shown was positive velocity feedback during the stance of stick insects (Bässler, 1976). The positive velocity feedback has been tested on a robot (Schmitz et al., 2008).

Similarly, myriapods, such as centipedes, change their interlimb coordination depending on gait speed. Myriapods have a long and flexible body axis and produce body undulations when the gait speed increases (Manton, 1965). In addition to the amplitude increase of the undulations, the phase relationship between ipsilateral leg movements changes in synchronization with the body segment movements of the undulations. In Aoi et al. (2007, 2013a), a multilegged robot with six body segments and twelve legs, which use torsional springs for body axis flexibility, was developed. The robot showed body undulations through a supercritical Hopf bifurcation of straight walking by increasing the locomotion speed, and so showed dependence of body undulations on speed, as was similar to the dependence shown by centipedes.

2.2. Environment-Dependent Adaptation

The advantage of using legs in mobile motion for animals and machines is to gain high traversability even in complex environments by manipulating the foot contact positions. However, the traversability of legged robots is still far from reaching the level of animals. During locomotion, the leg motion consists of the stance phase, in which the foot is in contact with the ground, and the swing phase, in which the foot is lifted off the ground. In the stance phase, the leg supports the body against gravity and produces propulsive and decelerating forces to move the body through the interaction between the

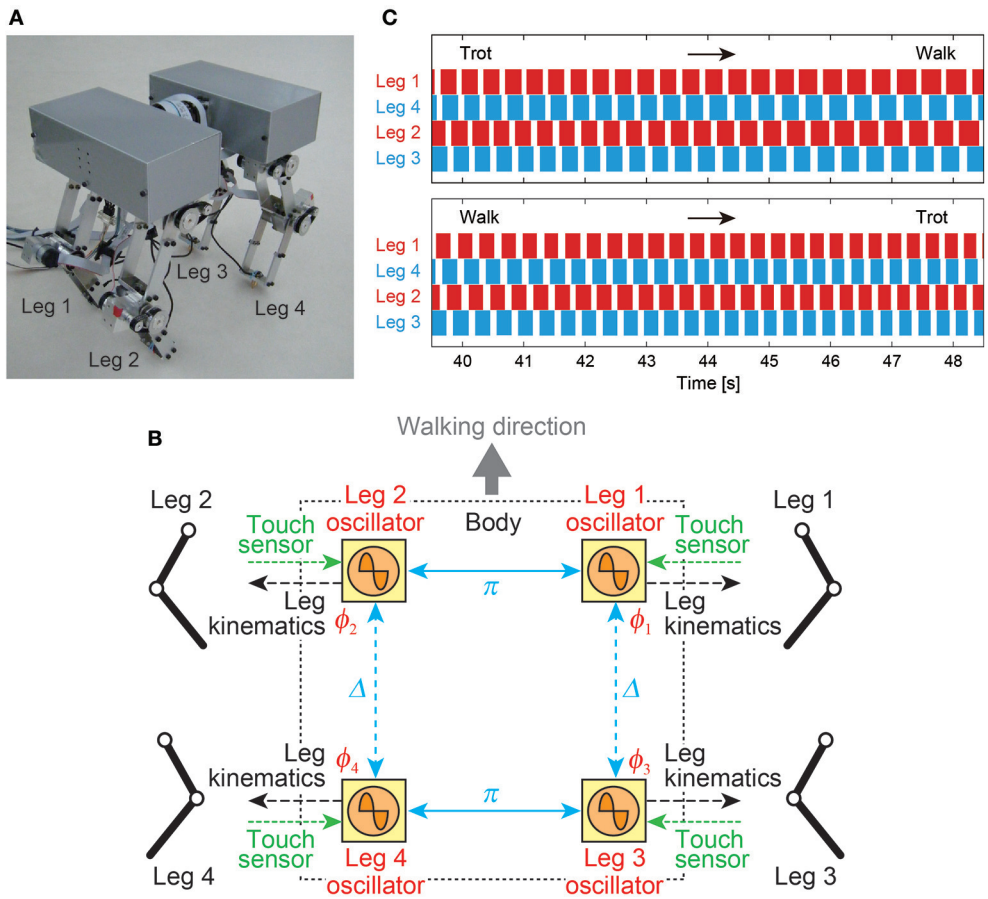


FIGURE 1 | Walk–trot transition of a quadruped robot induced by changing the locomotion speed. **(A)** Quadruped robot. **(B)** CPG-based phase oscillator network with local sensory feedback of foot contact information. Left and right leg oscillators are anti-phase and front and hind leg oscillators (Δ: phase difference) are only weakly coupled. **(C)** Footfall sequences during the trot-to-walk and walk-to-trot transitions (white: swing phase, colored: stance phase). These figures were modified from Aoi et al. (2013b).

foot and the ground. Geometric properties of the ground vary. These properties include being flat terrain, sloped terrain, or irregular and rough terrain. The physical properties of the ground also change. These properties include hard and slippery ground like stone, soft ground like loose soil, and flowable and penetrable ground like sand. The interaction between the foot and the ground is crucial to create locomotion, and real-time adaptation of motor behavior is required according to the ground situation. Animals actually show adaptive interlimb coordination depending on the environmental situation. To control legged robots, it is crucial to clarify and apply the dynamical principles of animals.

Manoonpong and his colleagues developed a series of modular neural CPG-based locomotion control for legged robots (Manoonpong et al., 2008, 2013; Steingrube et al., 2010; Goldschmidt et al., 2014; Xiong et al., 2014, 2015; Dasgupta et al., 2015; Grinke et al., 2015). They showed that using this control approach leads to adaptive interlimb coordination that allows the robots to deal with complex environments, such as walking over difficult terrain (Steingrube et al., 2010; Manoonpong et al., 2013; Goldschmidt et al., 2014; Xiong et al., 2014, 2015; Dasgupta et al., 2015) and avoiding obstacles in an unknown cluttered area (Manoonpong et al., 2008; Grinke et al., 2015), as observed in insects. For example, they implemented modular neural control with an adaptive chaotic CPG-based network and sensory feedback on a hexapod robot (**Figures 4A,B**; Steingrube et al., 2010). Due to the intrinsically chaotic dynamics of the CPG similar to that observed in certain biological CPGs (Rabinovich and Abarbanel, 1998), the dynamics were exploited to generate various walking patterns depending on the environmental condition. In their setup, the robot showed a tetrapod gait for standard walking, a wave gait for up-slope walking, a mixture gait between wave and tetrapod gaits for down-slope walking, and a tripod gait for fast walking to perform fast phototaxis (**Figure 4C**). However, this implementation of discrete gaits does not necessarily correspond to the situation found in insects. In addition to these multiple gaits, the chaotic dynamics especially contributed to self-untrapping of a leg from a hole in the ground (**Figure 4B**) and thereby enhanced foothold searching behavior. In Dasgupta et al. (2015), Goldschmidt et al. (2014),

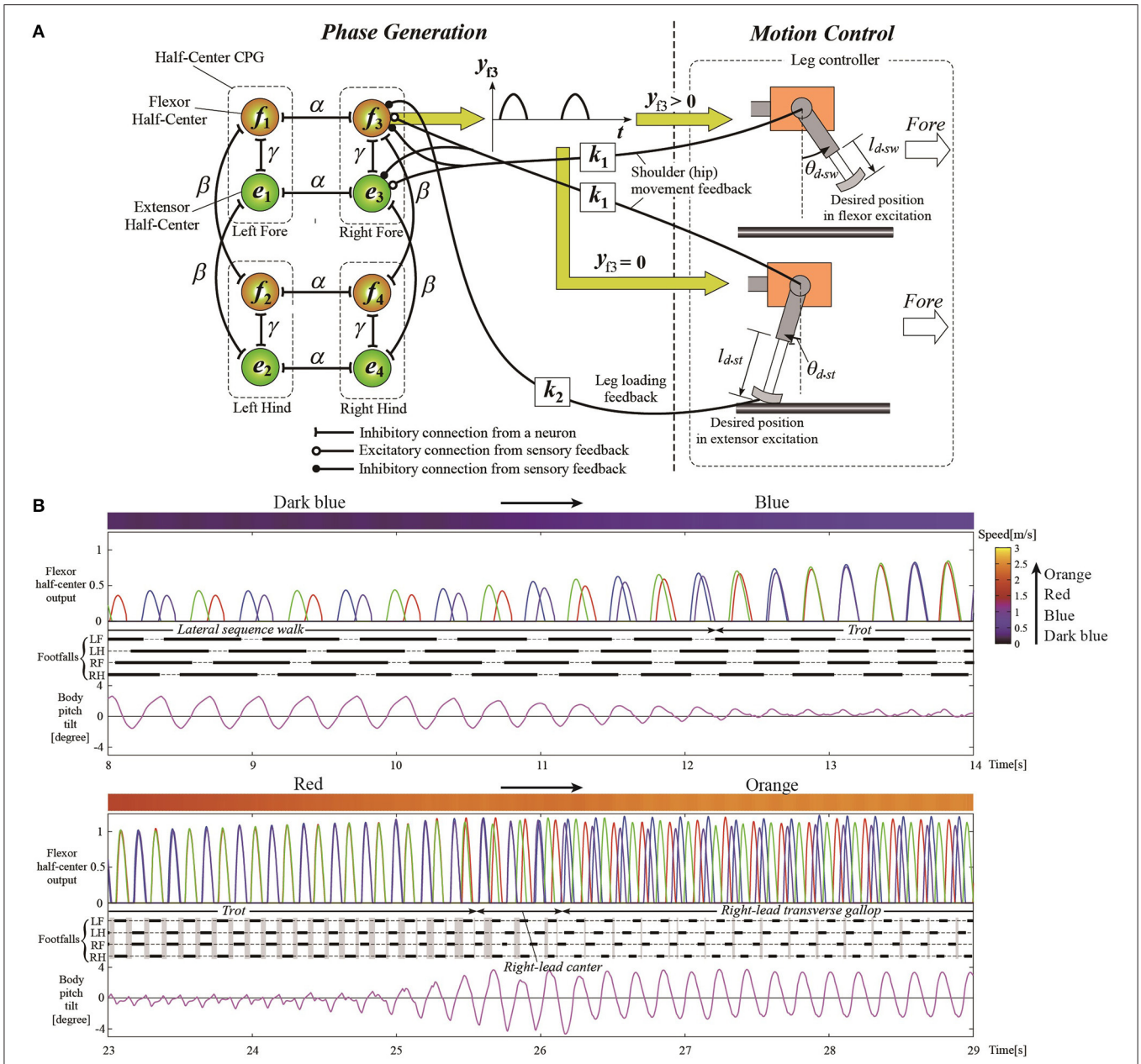


FIGURE 2 | Simulation results of walk, trot, and gallop gaits with different speeds for a quadruped model. **(A)** The quadruped model and its hard-wired neural oscillator network with leg loading feedback for interlimb coordination. **(B)** The CPG outputs for controlling the legs, different emerged gaits from the interlimb coordination, and body movement around the pitch axis. The walking speeds are shown as the top meter bars (dark blue to blue and red to orange) where speed level is denoted as the color meter on the top right gradient bar. The convex curves indicate the flexor half-center outputs for left foreleg (LF, blue), left hindleg (LH, red), right foreleg (RF, green), and right hindleg (RH, purple), which lead to the swing phase. The thick lines indicate the stance phase and the thin dashed lines refer to the swing phase. These figures were modified from Fukuoka et al. (2015) with permission.

and Manoonpong et al. (2013) integrating forward models into the modular neural control enabled the robot to effectively predict its walking state in order to extend or elevate its legs during the swing and stance phases while walking on complex terrains. With this setup, the robot walked on uneven terrain by using a tetrapod gait and climbed over high obstacles as well as up a flight of stairs by using a wave gait. Moreover, it successfully crossed a large gap by using a caterpillar gait, where each left and right pair of legs moved simultaneously. In this situation, however, stick insects show more complex behavior than caterpillar coordination, which is adopted only rarely, if at all (Blaesing and Cruse, 2004). In Xiong et al. (2014), modular neural control was extended by introducing muscle models based on virtual agonist-antagonist mechanisms

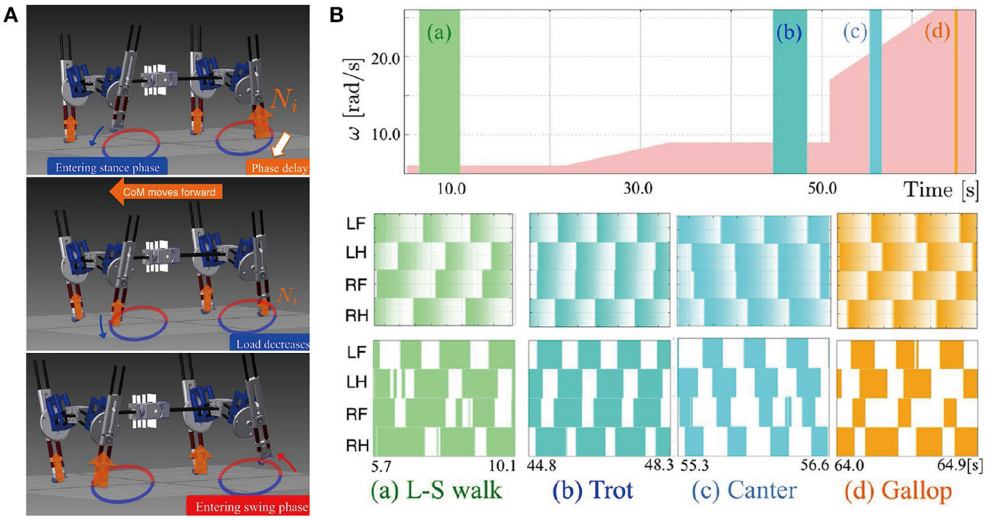


FIGURE 3 | Walk-trot-gallop transitions of a quadruped robot achieved by changing its locomotion speed. **(A)** The quadruped robot with interlimb coordination generated by non-wired simple phase oscillators (CPGs) with continuous phase modulation. The oscillator phases are modulated with respect to the magnitude of local load sensing N_i. **(B)** Walking speed and gait diagrams of different locomotion modes (walk, trot, canter, and gallop). The pink area shows the change of the treadmill speed with respect to the value of ω. The colored areas in the gait diagrams mean the stance phase, during which the sensor value N_i becomes greater than a threshold value. These figures were modified from Owaki and Ishiguro (2017) with permission.

(VAAM), and neuromechanical control was produced to achieve leg compliance. Combining neuromechanical control with sensorimotor learning results in energy-efficient walking using different gaits with corresponding leg compliances (Xiong et al., 2015). The robot efficiently walked on different surfaces including sponge, gravel, fine gravel, and grass. For adaptation to the avoidance of obstacles in a cluttered environment, an adaptive neural sensory processing network with synaptic plasticity was introduced to the modular neural control (Grinke et al., 2015). The adaptive processing network could drive different turning behaviors with short-term robot memory. As a consequence, the robot walked around and adapted its turning behavior to avoid obstacles in different situations and to avoid sharp corners or deadlocks (**Figure 4D**). In addition to the modular neural control approach, Schneider et al. (2012) developed bio-inspired control, which combines Walknet (mentioned above) with higher level control and planning (**Figures 5A,B**), for adaptive interlimb coordination of the hexapod robot HECTOR. By using this control technique, versatile behaviors (e.g., gap crossing, obstacle crossing, and global planning to avoid or attack obstacles) can be generated to deal with complex environments (**Figure 5C**). Furthermore, Schilling and Cruse (2017) expanded Walknet to invent new behaviors and test them by internal simulation before using them in reality. Arena et al. (2017) proposed multilayered CPG-based locomotion control with insect inspired motor-skill learning. It can adaptively coordinate the limbs of a Drosophila-like hexapod robot for stable walking and obstacle climbing.

When horses walk up an incline (Wickler et al., 2003) or when they carry weights (Farley and Taylor, 1991), the trot-to-gallop transition speed is reduced. Hexapod insects, such as stick insects, cockroaches, and beetles, change their gait depending on the slope of the ground (Spirito and Mushrush, 1979; Pelletier and Caissie, 2001; Grabowska et al., 2012). Furthermore, while cockroaches use the tripod gait during normal walking, the gait changes to metachronal when they are tethered on a supported ball to decrease loading (Spirito and Mushrush, 1979); uphill slope and loading induce similar effects on their gaits (Tang and Macmillan, 1986). Fujiki et al. (2013a) extended the control system of a quadruped robot (**Figure 1B**) for a hexapod walker and showed that the gait changed between tripod and metachronal gaits through the sensorimotor interaction depending on the loading and slope angle, as observed in insects.

Fukuoka et al. (2003), Fukuoka and Kimura (2009), and Kimura et al. (2007a,b) used the neural oscillators developed by Matsuoka (1985) to control quadruped robots (Tekken series). They incorporated models of various reflexes, such as the flexor reflex, extensor reflex, and vestibulospinal reflex, based on sensory information. In addition, they modeled the tonic labyrinthine response to adjust the rolling motion to synchronize with the pitching motion. The robots produced robust locomotion over irregular terrain, such as steps and slopes, while inducing the gait transition between walking and trotting.

When the ground is flowable like sand, the leg penetrates deeply into the ground during locomotion. Consequently, the interaction with the ground to produce lift, drag, and thrust forces becomes complicated [see review by Aguilar et al. (2016)]. Li et al. (2009, 2013) used a tripod gait for a hexapod robot and produced locomotor performance similar to that in hard ground by adjusting the leg shape and leg motion with a force model of the robot moving in granular media.

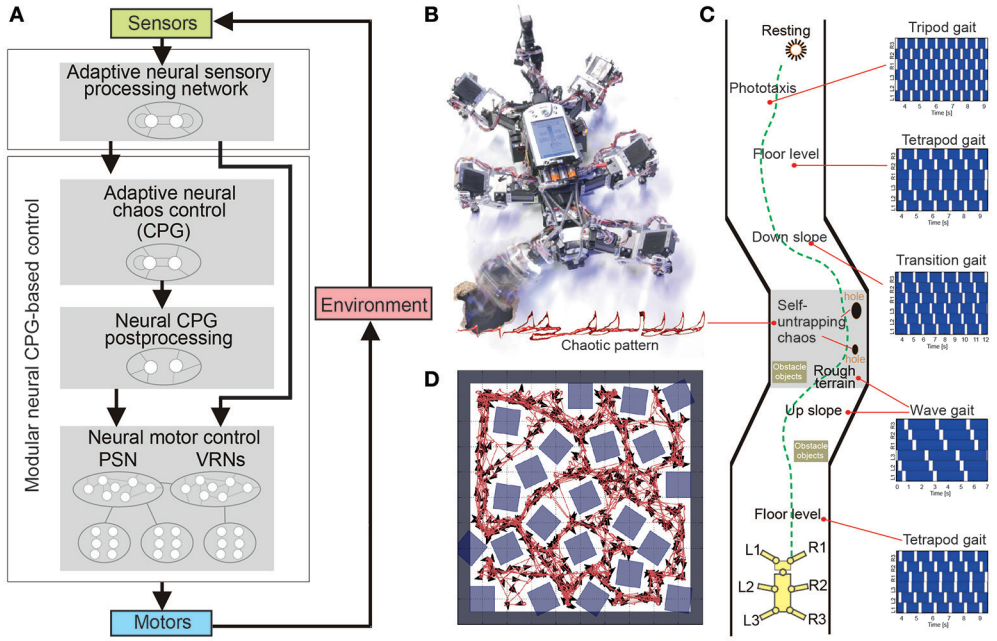


FIGURE 4 | Environment-dependent adaptation of a hexapod robot under modular single CPG-based locomotion control. **(A)** Modular neural locomotion control consisting of adaptive chaos control (acting as a CPG), CPG post-processing, and neural motor control. Locomotion control can generate multiple gaits (wave, transition, tetrapod, and tripod gaits) and a chaotic pattern for locomotion on different terrains. A combination of neural control with an adaptive neural sensory processing network can generate adaptive obstacle avoidance for avoiding obstacles in a complex environment. **(B)** Hexapod robot AMOS with a chaotic movement pattern for self-untrapping (i.e., freeing itself when its leg is trapped in a hole). The red light trace recorded from an LED installed at the leg shows the chaos motion of the leg. **(C)** Walking experiment on different terrains (floor, slopes, rough terrain, and holes in the ground) where different gaits were used. The white areas in the gait diagrams indicate the swing phase and the dark blue areas refer to the stance phase. **(D)** Adaptive obstacle avoidance in a simulated complex environment with obstacles, sharp corners, and narrow passages. These figures were modified from Grinke et al. (2015) and Steingrube et al. (2010).

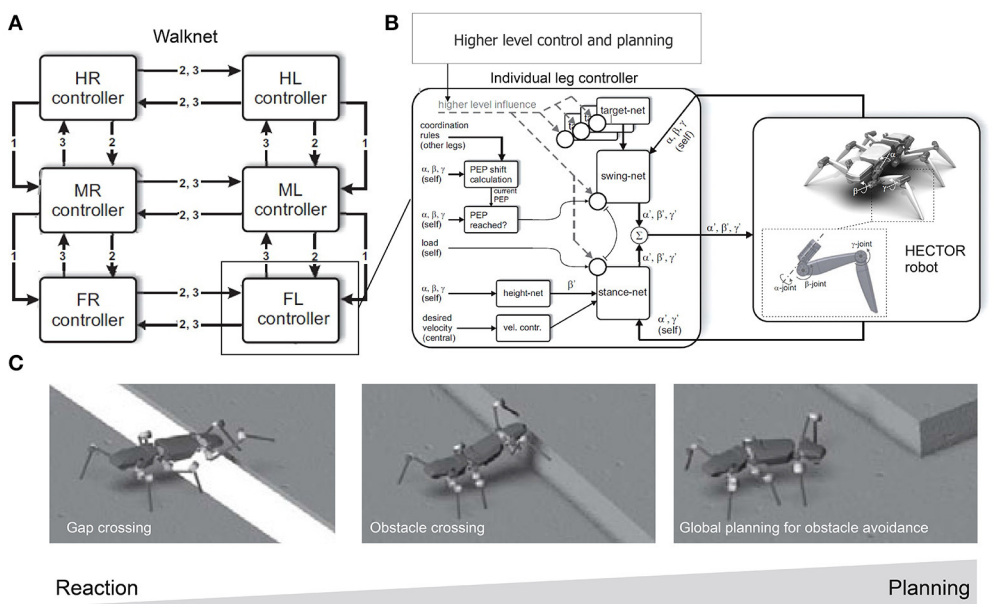


FIGURE 5 | Walknet with higher level control and planning for different locomotion behavior generation of the hexapod robot HECTOR for different environments. **(A)** Bio-inspired control Walknet with interlimb coordination rules [rules 1, 2, and 3; see Schilling et al. (2013a) and Schneider et al. (2012) for details]. **(B)** Setup of an individual leg controller with higher-level control and planning. Its outputs drive the leg joints of HECTOR. **(C)** Different desired locomotion behaviors that can be generated by the control approach to deal with complex environments. These figures were modified from Schneider et al. (2012) with permission.

Along with the adaptation to slopes, rough terrain, cluttered areas, and flowable areas, interlimb adaptation dealing with an asymmetric environmental condition has been investigated. For the asymmetric condition, split-belt treadmills have been used in studies of humans (Dietz et al., 1994; Reisman et al., 2005; Morton and Bastian, 2006), cats (Yanagihara and Kondo, 1996; Frigon et al., 2013), crayfishes (Müller and Cruse, 1991a,b), and stick insects (Bässler and Wegner, 1983; Foth and Graham, 1983). The treadmills have two parallel belts with independently controlled speeds and thus are capable of artificially creating left–right symmetric and asymmetric environments for walking (tied configuration: same speed between the belts, split-belt configuration: different speeds between the belts). Although the details are discussed later in Section 3.2, adaptive interlimb coordination has been observed in accordance with the belt speed condition. Such an adaptation appeared even in spinal cats (Forssberg et al., 1980; Frigon et al., 2013). Otoda et al. (2009) developed a sensory-driven controller without a CPG model for a two-dimensional biped robot and Fujiki et al. (2013b) used simple phase oscillators for the CPG model of a biped robot with local sensory feedback of the foot contact information, as was similarly done with the abovementioned quadruped and hexapod robots that achieved adaptive interlimb coordination (Aoi et al., 2011, 2013b; Ambe et al., 2013, 2015; Fujiki et al., 2013a). The biped robots achieved adaptive interlimb coordination on split-belt treadmills.

2.3. Body-Dependent Adaptation

Animals show adaptive motor behavior also due to changes in their body properties. As mentioned above, they change walking patterns when carrying weights or reducing their loads (Tang and Macmillan, 1986; Farley and Taylor, 1991). For fast locomotion, such as the galloping gait of cursorial quadrupeds and the undulatory walk of centipedes, the appearance of trunk and body-segment movements suggests that body flexibility is crucial for adaptive locomotion (Alexander, 1988). In Aoi et al. (2011), a quadruped walker was controlled by simple phase oscillators with local sensory foot contact information (**Figure 1B**) and the change in trunk flexibility induced the walk–trot transition, where walking and trotting gaits appeared for a hard trunk and a soft trunk, respectively. In Aoi et al. (2007, 2013a, 2016), a centipede-like multilegged robot showed the gait transition from straight walking to body undulatory walking through a Hopf bifurcation by changing the body axis flexibility.

One of the advantages to using many legs for mobile motion, as in insects and myriapods, is the avoidance of losing mobility completely by leg damage due to injury and predation. Through adaptive control of interlimb coordination, even complete leg loss does not prevent walking (Grabowska et al., 2012). To clarify how interlimb coordination changes with leg loss, amputations of single legs of stick insects have been performed in order to investigate changes of the relative phases between the legs depending on which leg is amputated (Graham, 1977). Dasgupta et al. (2015) used neural CPG-based control with distributed adaptive forward models for the hexapod robot, as mentioned above, and demonstrated that the robot successfully kept walking straight with a slightly modified tetrapod gait through adaptation despite the damaged right middle leg. Ren et al. (2015) extended the chaotic CPG controller introduced above (Steingrube et al., 2010) to a controller of multiple chaotic CPGs depending on the number of legs (**Figure 6A**). They demonstrated that the six-legged robot (AMOSII) could continue walking by changing the interlimb coordination in accordance with the disabled leg(s) (**Figures 6B,C**). Walknet, which identifies the behavior of stick insects, as introduced above, was able to coordinate the movements of the remaining legs so that a six-legged walker could continue walking when some legs were amputated (Kindermann, 2002; Schilling et al., 2007). Besides these bio-inspired control approaches, Cully et al. (2015) proposed an alternative machine learning based approach consisting of two main parts: an automatically generated, pre-computed, behavior-performance map, and a trial-and-error learning algorithm (**Figure 7**). The behavior-performance map contains a number of interlimb coordination parameters that can generate approximately 13,000 different gaits. The trial-and-error learning algorithm is used to search for successful robot locomotion behaviors from the map with respect to robot body condition. They showed that this approach allows a hexapod robot to walk and rapidly find a walking behavior that can compensate for damage. Although all these approaches predefined interlimb connections, another approach based on the concept of emergent locomotion (i.e., walking patterns appearing as a result of stabilization in a self-organizing manner, Schilling et al., 2013a) from tight interaction between neural systems, musculoskeletal systems, and the environment has been explored for body-dependent adaptation. For example, Barikhan et al. (2014) proposed multiple decoupled neural CPGs with local sensory feedback (**Figure 6D**). This approach exploited the interaction between neural and body dynamics through foot contact feedback to achieve self-organized locomotion and to allow a hexapod robot to quickly adapt its locomotion to deal with morphological changes [e.g., leg damage (**Figure 6E**) or asymmetric leg lengths between the front and hind legs]. Tsuchiya et al. (2002) used simple phase oscillators with local sensory foot contact information to control a ten-legged robot to establish adaptive interlimb coordination, as mentioned above for quadruped and hexapod robots. The leg loss induced the change in interlimb coordination, and the change reduced the degradation of locomotion performance, such as gait speed.

2.4. Task-Dependent Adaptation

Animals often encounter a situation in which they have to change locomotor behavior. For example, when an obstacle appears in a walking path, they step over the obstacle, or turn to the right or the left to avoid collision with the obstacle (this is also related to environment adaptation). Such a task is mainly generated by modulating the leg movements, and thus adaptive control of intralimb coordination is important. However, also important is adaptive control of interlimb coordination. To step over an obstacle, the leading limb first clears the obstacle and then the trailing limb follows it. The foot of the leading limb must be raised higher than usual to avoid collision with the obstacle, and this motion delays foot contact. Especially for bipedal and

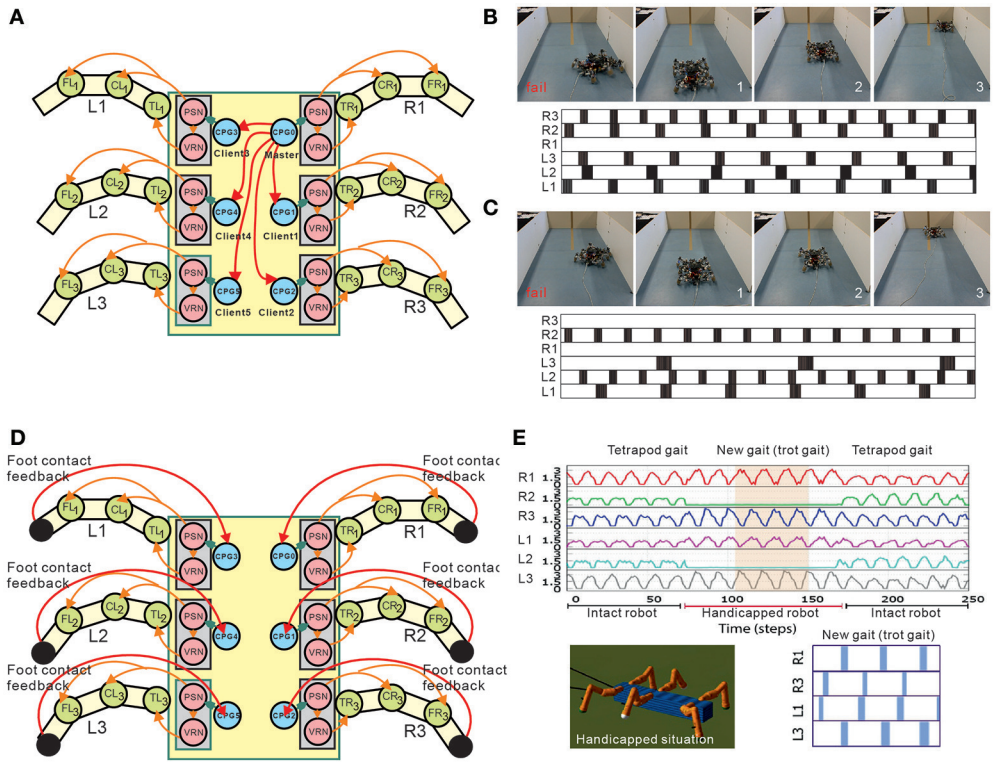


FIGURE 6 | Adaptation to deal with leg malfunction of a hexapod robot under modular multiple CPG-based locomotion control. **(A)** Implementation of multiple chaotic CPGs on a hexapod robot. In the setup, the connections between CPGs are predefined. **(B,C)** Two example scenarios of real robot experiments with disabled legs. In **(B)**, the robot walked with the R1 leg disabled and in **(C)** it walked with the R1 and R3 legs disabled. For each subfigure, the upper panel shows one snapshot of a fail situation (before learning) and three snapshots of a success situation (after learning). The lower panel shows the gait (i.e., suitable leg frequencies) after learning. A black area means that the leg touched the ground, and a white area indicates that the leg was in the air. In this setup, the robot learned to find a proper combination of oscillation frequencies of different legs for malfunction compensation. **(D)** Implementation of multiple CPGs with foot contact feedback on a hexapod robot. This setup does not have predefined coordination between the CPGs. Intralimb coordination emerges from the interactions between the body dynamics and the environment through foot contact feedback of each leg. **(E)** Example of the robot experiment dealing with a temporary handicapped situation. The ground reaction forces during movement of the robot with middle legs temporarily disabled show that the robot quickly adapted to a new gait (i.e., trot gait) and was able to continue walking properly. These figures were modified from Barikhan et al. (2014) and Ren et al. (2015).

quadrupedal animals, the foot of the trailing limb must be raised after foot contact of the leading limb; otherwise, the obstacle avoidance task will fail because the contralateral limb does not support the body at the onset of raising the trailing limb (Aoi et al., 2013c).

Turning behavior to change walking direction is used for various tasks, such as target pursuit (Szczecinski et al., in press) and obstacle avoidance (**Figures 4D**, **5C**). Knops et al. (2013) controlled a mechanical model of a stick insect's middle legs by using a neural network model based on Hodgkin Huxley dynamics and produced turning behavior with two different strategies observed in stick insects walking on a slippery surface: switching the inner middle leg from forward to sideward, or from forward to backward stepping. In Aoi et al. (2016), the turning maneuverability of a centipede-like multilegged robot was enhanced via straight walk instability induced by the Hopf bifurcation by changing the body axis flexibility. Although arthropods with sprawling legs have a low center of mass and thus cannot effectively lean, mammals with erect legs have a high center of mass and can use body leaning to help turning. The relative phase between legs in human turning shifts from anti-phase due to the left–right asymmetry of the turning movement (Courtine and Schieppati, 2003). In Aoi and Tsuchiya (2007), simple phase oscillators with local sensory feedback about foot contact information were used for turn walking of a biped robot, as was used for walking on a split-belt treadmill. The relative phase between legs shifted depending on the turning radius to compensate for the left–right asymmetry induced by body leaning; this shift allowed the robot to achieve high turning performance.

The transition from quadrupedal gait to upright and bipedal gait is a challenging task for legged robots, because it requires drastic changes in locomotor movements (Asa et al., 2009; Aoi et al., 2012; Kobayashi et al., 2015). In particular, because the robot has to raise its trunk so that the arms leave the ground, an adequate relationship between the supporting limb locations and the center of mass location is important. That is, adequate interlimb and trunk coordination is crucial;

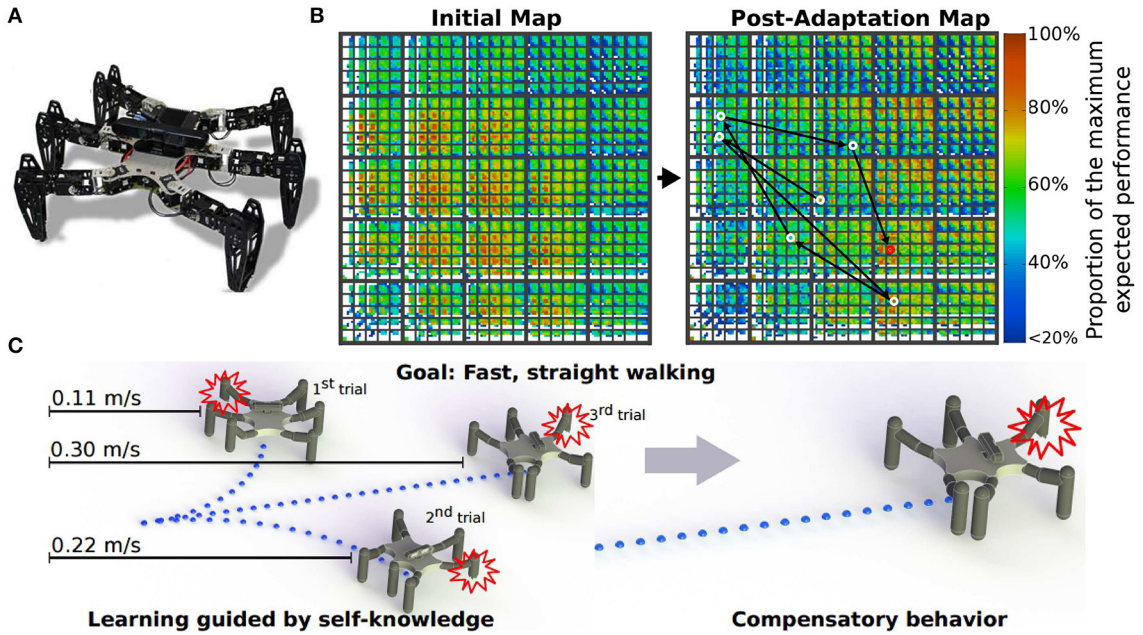


FIGURE 7 | Machine learning approach for locomotion and damage recovery of a hexapod robot. **(A)** Robot setup. **(B)** Example of a behavior-performance map and its post-adaptation for damage recovery. Each colored pixel represents the highest-performing behavior discovered during map creation at that point in six-dimensional behavior space. **(C)** Robot adaptation using the trial-and-error algorithm to recover from leg damage. These figures were modified from Cully et al. (2015) with permission.

otherwise, the robot easily falls over. In Aoi et al. (2012), simple phase oscillators with sensory regulation by ground contact information of the arms and legs were used for a biped robot (**Figures 8A,B**). The controller was extended based on the concept of kinematic synergy (Freitas et al., 2006; Ivanenko et al., 2007; Latash, 2008; Funato et al., 2010) to change the robot movements for gait transition and allowed the robot to successfully change the gait from quadrupedal to upright and bipedal (**Figure 8C**).

Legged robots are useful for search and rescue missions. In this case, the ground is not only irregular but also fragile, like an area with scattered debris and collapsed buildings, on which surfaces may collapse when put under external forces, such as the pressure from a robot's leg. It is important to check the ground condition in such situations by using haptic information of the legs to secure stable walking. In Ambe and Matsuno (2016), a control mechanism with haptic sensory feedback for terrain determination was proposed. With the control mechanism, a quadruped robot can sense whether the foothold is stable through its force sensor when it puts its leg on the ground. In addition, this mechanism produces adequate interlimb coordination so that the robot never stumbles, even if the foothold collapses in the probe motion. As a result, the robot can effectively walk on unstable terrain and avoid stumbling and causing a large collapse of the surrounding area (**Figure 9**). Other methods have also been proposed to estimate fragile and slippery footholds based on haptic feedbacks and image information (Tokuda et al., 2003; Hoepflinger et al., 2010, 2013).

3. CHARACTERISTIC PROPERTIES OF ADAPTIVE INTERLIMB COORDINATION

3.1. Hysteresis in Gait Transition

As discussed in Section 2.1, animals change their walking patterns depending on their locomotion speed. In general, locomotion speed has a large sudden change at gait transition in overground walking. However, using treadmills, which can control gait speed, we can investigate the speed-dependent gait transition mechanism by smoothly and continuously changing the belt speed of the treadmills. It has been reported in humans and some quadruped animals that the gait changes at different speeds depending on whether the speed is increasing or decreasing, and that a speed range exists in which different gaits are used. In other words, gait transitions may exhibit hysteresis (Diedrich et al., 1998; Heglund and Taylor, 1998; Raynor et al., 2002; Griffin et al., 2004). **Figure 10A** shows the relative phase between the right front and hind legs of a dog walking on a treadmill for walk-to-trot and trot-to-walk transitions induced by changing the belt speed (Aoi et al., 2013b). This figure shows hysteresis in the walk–trot transition. Such a phenomenon is difficult to explain by triggering the gait transition based on metabolic and biomechanical factors. The dynamical system approach might provide useful insights into such a gait transition mechanism (Diedrich et al., 1998).

Quadruped robots controlled by simple phase oscillators with local sensory foot contact information, as introduced in Section 2.1, showed hysteresis in the walk–trot transition induced

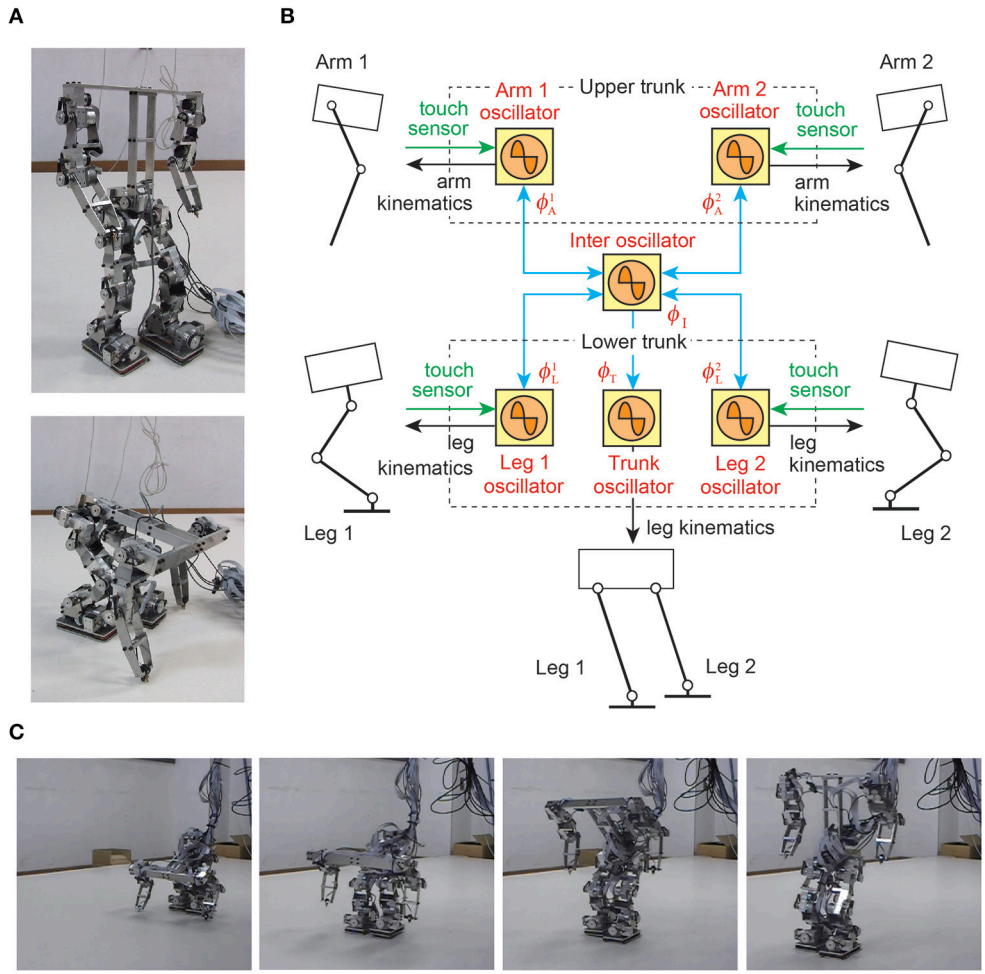


FIGURE 8 | Gait transition of a biped robot from quadrupedal to upright and bipedal gait. **(A)** Biped robot. **(B)** Phase oscillator network with local sensory foot contact information for the biped robot. **(C)** Gait transition experiment. These figures were modified from Aoi et al. (2012).

by changing the locomotion speed (**Figure 10B**; Aoi et al., 2011, 2013b). Because walking and trotting gaits are mainly distinguished by the relative phases of the ipsilateral legs, a stability analysis using the return maps of the relative phases clarified the stability structure of the gaits. **Figure 10C** shows the return maps obtained at three different speeds. While only one stable relative phase exists in the left and right figures, two stable and one unstable relative phases exist in the middle figure. The stable and unstable relative phases explain that hysteresis is generated through two saddle-node bifurcations induced by changing the locomotion speed (**Figure 10D**). From this result, a potential function is derived, as shown in **Figure 10E**. It suggests that gait transition is explained by switching the stability of self-organized patterns in the complex dynamical system.

Gait transition hysteresis also appears in other legged robots controlled by CPG models with sensory feedback, e.g., in the walk–run transition of a biped model (Taga and Shimizu, 1991) and the metachronal–tripod gait transition of hexapod robots (Kimura et al., 1993; Fujiki et al., 2013a; note that insects do not clearly show abrupt transitions, but a continuum of locomotion patterns).

3.2. Two Different Time-Scale Adaptations

When the environment suddenly changes, locomotor behavior is rapidly modulated to adapt to the environmental variation and successively shows gradual regulation for gaining a new locomotor pattern. This behavior suggests that motor learning occurs. This has been observed in interlimb coordination during locomotion. In particular, the split-belt treadmill walking mentioned above is a good example.

The regulation of motor behavior in split-belt treadmill walking appears in various locomotor factors. However, the factors related to interlimb coordination, such as the relative phase between the legs, step length, and center of pressure profile, and those related to intralimb coordination, such as the duty factor and stride length, show different trends (**Figure 11**). A sudden environmental variation rapidly changes the factors; this is called "early adaptation". Although the intralimb coordination factors do not show further change,

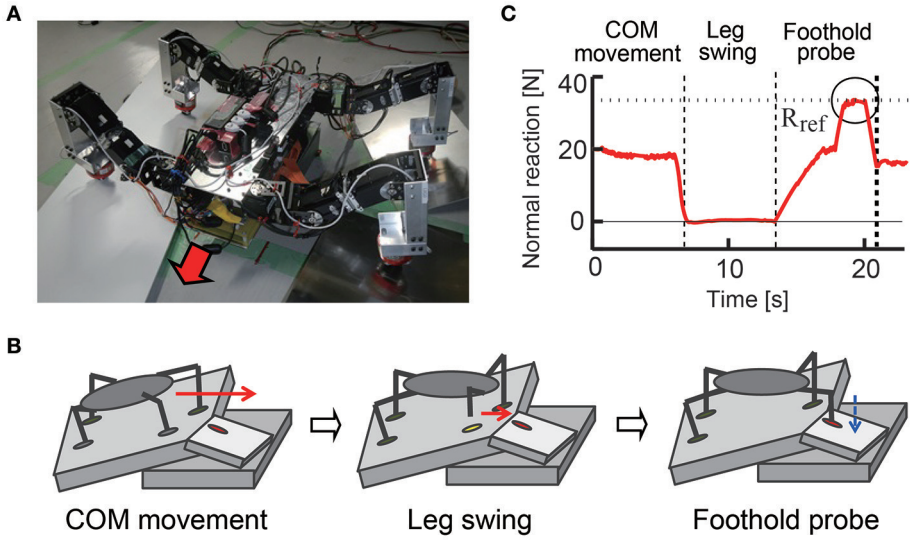


FIGURE 9 | Walking on fragile irregular terrain. **(A)** Quadruped robot with load sensors on feet. **(B)** Process to find footfold condition for fore right leg. The robot moves the center of mass by standing on all four legs, then swings a leg, and probes the foothold by applying force gradually. The robot repeats this process for each of the four legs. **(C)** Time response of normal reaction force of the leg in experiments. The robot applies force over a reference value R_{ref} in the grope phase to ensure that the ground is solid enough for walking, but the robot never applies force over R_{ref} in the other phases. These figures were modified from Ambe and Matsuno (2016).

the interlimb coordination factors tend to gradually return to their original state after early adaptation; this is called "late adaptation". This means that interlimb coordination has two types of adaptations with different time scales. Furthermore, when the environment is returned its original state, the interlimb coordination factors move in the opposite direction to the early adaptation, which shows the after-effects.

Rapid changes in the locomotor factors have been observed during split-belt treadmill walking of spinal cats (Forssberg et al., 1980; Frigon et al., 2013). These rapid changes suggest that early adaptation is induced by sensorimotor interaction in the spinal cord. On the other hand, humans with cerebellar damage do not show late adaptation or after-effects during split-belt treadmill walking, and it appears that the cerebellum contributes to late adaptation and the after-effects (Morton and Bastian, 2006; although split-belt experiments have been performed for arthropods (Bässler and Wegner, 1983; Foth and Graham, 1983; Müller and Cruse, 1991a,b), the results showed that they do not necessarily need learning, which may underestimate their adaptation ability). Otoda et al. (2009) modeled the stepping reflex to modulate the touchdown angle of the swing leg and introduced the adjustment of proportional control gain at the hip joint of the stance leg as the cerebellar function producing split-belt treadmill walking of a two-dimensional biped robot, although they did not use a CPG model with adaptation. In contrast, Fujiki et al. (2015) incorporated a cerebellar learning model into the spinal CPG model (**Figure 12B**). The CPG model was composed of simple phase oscillators with sensory reflex by local foot contact information and was used in Fujiki et al. (2013b) as mentioned above. The learning model modulated the foot contact timing of each leg through the evaluation of prediction error by using the local sensory foot contact information of each leg. Biped robot experiments on a split-belt treadmill (**Figure 12A**) showed adaptive intralimb and interlimb coordination (**Figures 12C,D**). In particular, despite the lack of direct interlimb coordination control, early and late adaptations and after-effects were observed in interlimb coordination, and showed strong similarities to those observed in humans.

Rapid modulation by the sensory reflex model and gradual modulation by the learning model changed the pitching moment depending on the belt speed condition through the body dynamics of the robot (**Figure 13**). The pitching moment change induced spatiotemporal modification of the robot movements and altered various locomotor factors. The sensory reflex model secured the ability to continue walking against the environmental change, and the cerebellar learning model modulated the robot movements under those conditions to make walking smoother and more efficient through optimization (minimization of prediction error of foot contact timing). For simple human behaviors, such as arm reaching movements, learning models that aim to minimize jerk or torque-change have been proposed (Flash and Hogan, 1985; Uno et al., 1989). However, for human locomotion, it remains unclear what factors are predicted and how to facilitate the learning. This is partly because locomotion is a whole-body movement through limb movement and posture controls, and is governed by complicated dynamics including foot contact and lift off, which change the physical constraints. Robot experiments with neurophysiologically inspired control models are useful for examining potential control models through the comparison of results obtained from human measured data and clarification of dynamical mechanisms.

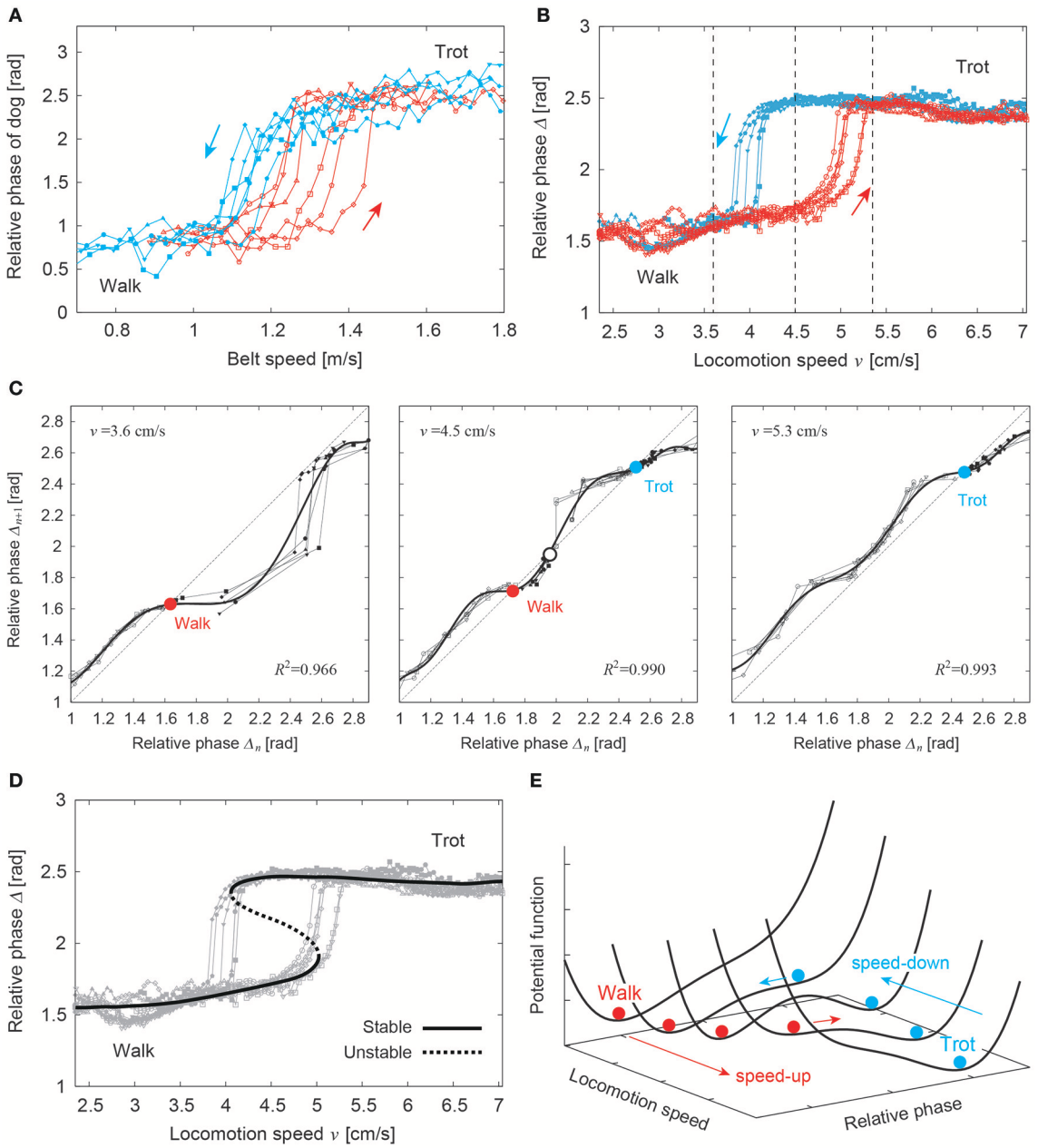


FIGURE 10 | Hysteresis in the walk–trot transition. **(A)** Relative phases of ipsilateral legs of a dog for walk-to-trot and trot-to-walk transitions induced by changing the belt speed. **(B)** Relative phases of a quadruped robot for walk-to-trot and trot-to-walk transitions induced by changing the locomotion speed. The three speeds indicated by vertical dotted lines are used in **(C)**. **(C)** Stability analyses using return maps for the relative phases at three speeds. The bold lines are approximated polynomial functions of the return maps. **(D)** Estimated stable and unstable relative phases from the stability analyses showing two saddle-node bifurcations. **(E)** Possible potential function that shows hysteresis. These figures were modified from Aoi et al. (2013b).

4. KEY FACTORS AND MECHANISMS FOR ADAPTIVE INTERLIMB COORDINATION

In the previous sections, we presented adaptive interlimb coordination of animals and legged robots to deal with different locomotion speeds, environmental situations, body properties, and tasks. Here, we discuss key factors and mechanisms underlying the adaptive control of interlimb coordination.

One of the key mechanisms is the CPGs, which are located in the spinal cord of vertebrates and in the thoracic ganglia of invertebrates. Except for the anti-phase activity of antagonistic excitatory motoneurones, no feature of the pilocarpine-induced rhythm appears to correspond to any motor output observed in stick insects (Büschges et al., 1995). However, neurophysiological studies have revealed that CPGs are important for locomotion (Grillner, 1975; Orlovsky et al.,

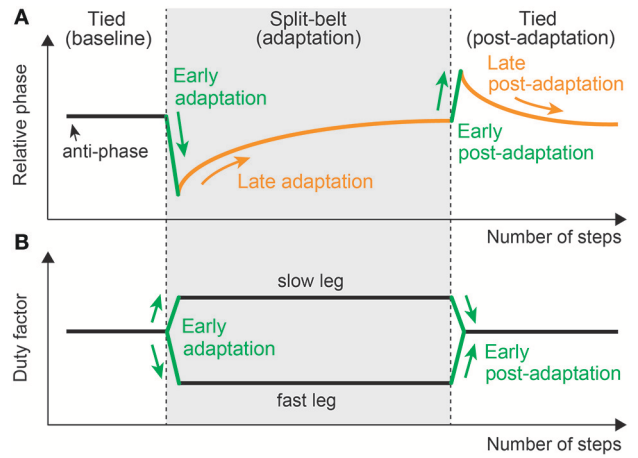


FIGURE 11 | Changes in locomotor factors during human split-belt treadmill walking, where the belt speed condition changes from the tied to the split-belt configuration (adaptation period) and returns to the tied configuration (post-adaptation period). **(A)** Relative phase between legs as one of the interlimb coordination factors. This shows both early and late adaptations when the environmental change occurs. When the environment is returned, after-effects appear. **(B)** Duty factor of the legs as one of the intralimb coordination factors. This shows only early adaptation when the environment changes. These figures were modified from Fujiki et al. (2015).

1999; MacKay-Lyons, 2002). A CPG is a group of interconnected neurons that can be activated to generate a motor pattern without the requirement of sensory feedback. As described in Ijspeert (2008), various CPG models with different levels of complexity have been proposed, from detailed biophysical models using Hodgkin-Huxley neurons (Traven et al., 1993; Cataldo et al., 2006; Holmes et al., 2006; Bungay and Campbell, 2009) and connectionist models using leaky-integrator neurons or integrate-and-fire neurons (Buchanan, 1992; Arena, 2000) to abstract models using coupled oscillators (Ijspeert et al., 2007; Chung and Slotine, 2010; Yu et al., 2014). Although some robot studies have shown that complex insect behavior, such as continuous gait transition, walking over irregular ground including a large gap, and curve walking with an irregular step pattern, can be replicated without CPG models (Cruse et al., 1998; Lewinger and Quinn, 2011; Schilling et al., 2013a,b), these CPG models have improved locomotion control of legged robots, such as the control of speed (Ijspeert, 2008) and robustness against sensory noise as well as sensory failure (Di Canio et al., 2016). In particular, key issues for controlling legged robots are design of feedforward and feedback controllers and integration of these controllers. The CPG models give us useful ideas for the design and integration so that the integrated controller works in a biologically plausible fashion (comparison between the controllers with and without CPG models would be useful to find the contribution of the CPG models).

Most research has employed abstract CPG models with hardwired connections to motor units for generating different basic locomotor behaviors, such as walking and swimming. Switching between different gaits or locomotion modes can be done by using simple external input signals (Kirchner et al., 2002; Ijspeert et al., 2007; Manoonpong et al., 2008). Though CPGs acting as open-loop control are the key for production of basic rhythmic locomotion, sensory feedback is a very important factor needed for adaptations to different speeds, environments, bodies, and tasks, as described in previous sections for adaptive interlimb coordination. Combining CPGs with sensory feedback results in closed-loop control with adaptability. For robotic implementation, different sensory feedback affecting CPG activities includes proprioceptive feedback (e.g., joint/leg movement and force) and exteroceptive feedback (e.g., foot contact and vision). Such feedback can modulate the frequency, phase, and magnitude of CPG activities [see review by Buschmann et al. (2015)].

Frequency modulation (also known as entrainment, Buchli et al., 2006) uses feedback information to adapt the frequency of the CPG so that the frequencies of the feedback and the CPG are synchronized (Nachstedt et al., 2017). Usually, joint angle feedback is used for this process in robotics studies (Endo et al., 2004; Buchli and Ijspeert, 2008; Di Canio et al., 2016) and frequency modulation has been mainly employed for adaptations of locomotion speed (Harischandra et al., 2011; Di Canio et al., 2016) and body change (Ren et al., 2015). In contrast, phase modulation typically uses foot contact and foot loading feedbacks to adjust the phase of CPGs to regulate the swing and stance phase durations, depending on the situation. In particular, the phase resetting mechanism, which has often been used for phase modulation in legged robots, was developed from the phase shift and rhythm resetting behaviors by the tactile sensor feedback in cats (Conway et al., 1987; Duysens, 1997; Schomburg et al., 1998; Rybak et al., 2006; Frigon et al., 2010) and stick insects (Büschges, 1995; Bässler and Büschges, 1998). The functional role of phase resetting has been investigated by the integration with musculoskeletal models and muscle synergy hypothesis (Aoi et al., 2010, 2013c; Aoi and Funato, 2016), and the control strategy was implemented in legged robots and helped to improved the robustness of their walking (Tsuchiya et al., 2002; Aoi and Tsuchiya, 2005, 2007; Nomura et al., 2009; Aoi et al., 2011, 2012, 2013b; Ambe et al., 2013, 2015; Fujiki et al., 2013a,b, 2015). Phase modulation has also been widely used for different adaptations including locomotion speed (Tsuchiya et al., 2002; Aoi et al., 2011, 2013b; Ambe et al., 2013, 2015; Fujiki et al., 2013a; Owaki et al., 2013; Fukuoka et al., 2015; Owaki and Ishiguro, 2017), environmental condition (Aoi and Tsuchiya, 2005; Aoi et al., 2010; Fujiki et al., 2013a,b, 2015), body properties (Tsuchiya et al., 2002; Aoi et al., 2011; Fujiki et al., 2013a; Owaki et al., 2013; Barikhan et al., 2014), and task (Aoi and Tsuchiya, 2007; Aoi et al., 2012, 2013c). Magnitude modulation uses different types of feedback, such as force and vision, to regulate the magnitude of the CPG. This regulation is indirectly achieved through premotor neuron networks (Buschmann et al., 2015). Goldschmidt et al. (2014) and Grinke et al. (2015) employed this strategy by using visual feedback for environment-dependent adaptation, such as hexapod robots climbing over an obstacle or turning away from it.

One can also achieve adaptive interlimb coordination by integrating these CPG modulation techniques with other

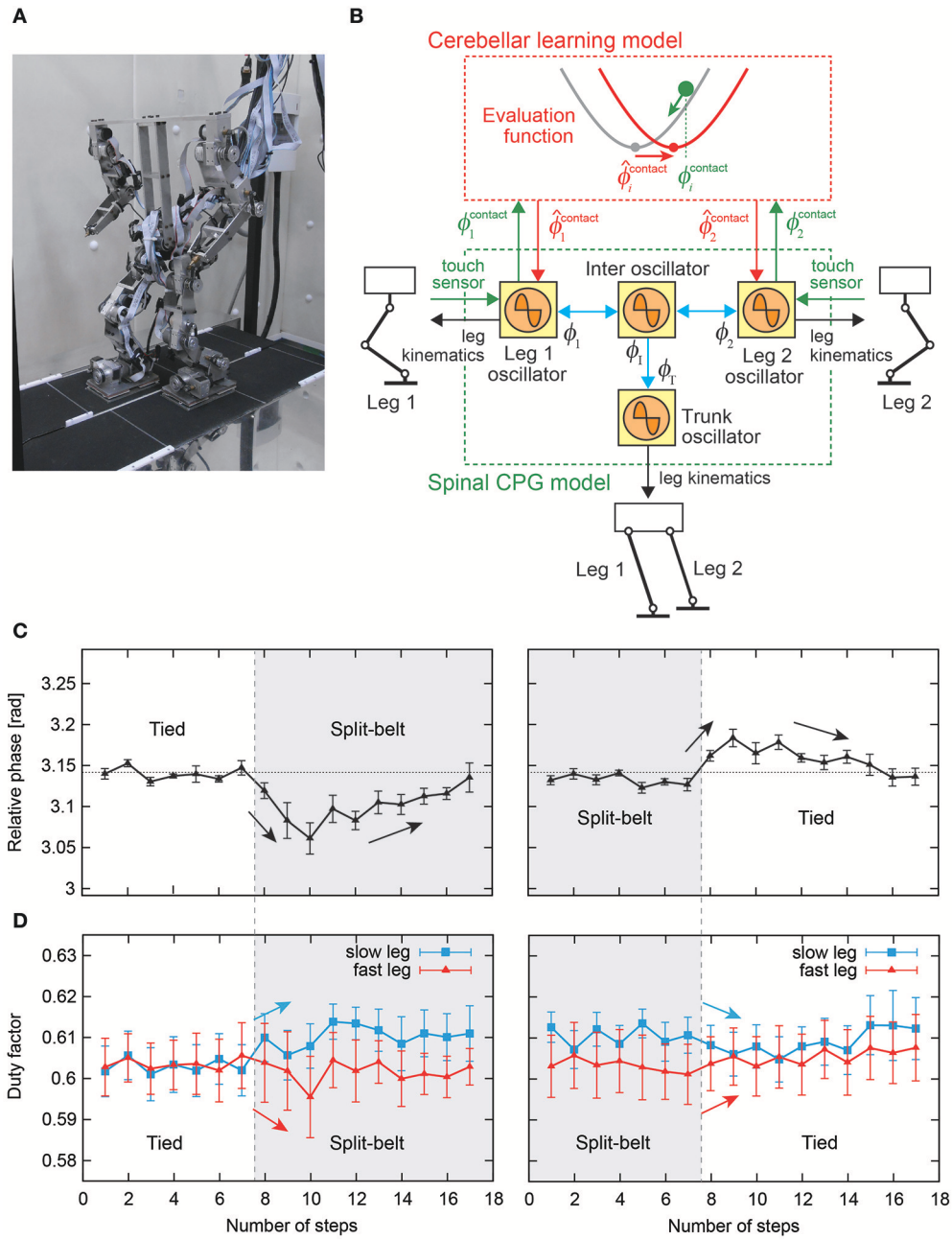


FIGURE 12 | Split-belt treadmill walking experiment of a biped robot. **(A)** Biped robot and split-belt treadmill. **(B)** Spinal CPG and cerebellar learning models using simple phase oscillators with sensory reflex by local foot contact information and modulation of foot contact timing through the evaluation of prediction error. **(C)** Relative phase between legs of the biped robot. **(D)** Duty factor of legs of the biped robot. These figures were modified from Fujiki et al. (2015).

bio-inspired approaches, such as adaptive muscle stiffness control (Xiong et al., 2015). Manoonpong et al. (2013) showed that bio-inspired forward models that translate motor commands or efference copies into expected sensory feedback are important components for environment-dependent adaptation, i.e., walking on different terrains. By using a split-belt treadmill, Fujiki et al. (2015) showed that cerebellar learning models to regulate motor commands while minimizing the prediction error are also important for environment-dependent adaptation. **Table 1** roughly categorizes the key mechanisms that have been used for different adaptations.

In addition to these bio-inspired key factors (CPGs, sensory feedbacks, forward model, learning model, and muscle stiffness), which are usually applied to independent control of individual legs or joints, most of the studies explicitly design complete

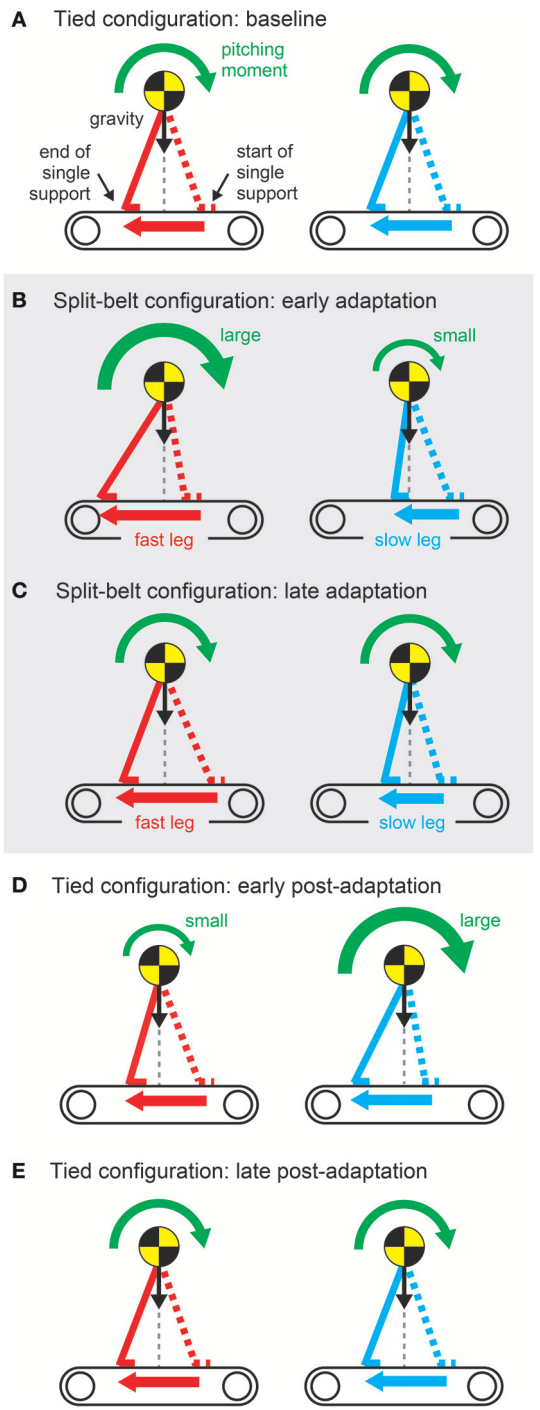


FIGURE 13 | Pitching moment change due to belt-speed change **(A–E)** through sensory reflex and learning regarding foot contact timing. These figures were modified from Fujiki et al. (2015).

interlimb connections to obtain the desired locomotor behaviors. This results in limitations of adaptive and flexible interlimb coordination (e.g., Kirchner et al., 2002; Ijspeert et al., 2007; Harischandra et al., 2011; Manoonpong et al., 2013; Ren et al., 2015). To overcome these limitations, a proposed alternative paradigm achieves interlimb coordination by local sensing, body-environment interactions, and weakly-coupled or decoupled CPGs (Tsuchiya et al., 2002; Aoi et al., 2011, 2013b; Shim and Husbands, 2012; Ambe et al., 2013, 2015; Fujiki et al., 2013a; Owaki et al., 2013; Barikhan et al., 2014; Owaki and Ishiguro, 2017), rather than by predefined interlimb connections. Although the proposed paradigm leads to high flexibility and adaptability in interlimb coordination, it sometimes encounters unstable locomotion. Phase resetting, which modulates the CPG phase based on the sensory reflex, as mentioned above, is one of the solutions to obtain flexible and adaptive interlimb coordination while keeping stability in locomotion (Tsuchiya et al., 2002; Aoi and Tsuchiya, 2007; Aoi et al., 2011, 2012, 2013b,c; Ambe et al., 2013, 2015; Fujiki et al., 2013a,b, 2015). However, this uses only phase modulation and has limitations. Thus, one future research study in this direction is to find a method that can autonomously form the plastic connections for stable but still flexible and adaptive locomotion. Furthermore, the interactions of CPGs, sensory feedback, body dynamics, forward model, learning model, and muscle stiffness for highly adaptive, robust, and energy-efficient locomotion remain to be explored.

TABLE 1 | Key mechanisms used for different adaptations.

Adaptation	Key mechanism
Speed-dependent	CF, CP, PF
Environment-dependent	CF, CP, PF, CP+LM, CPG+FM, CPG+FM+MS
Body-dependent	CF, CP, PF, ML
Task-dependent	CP, CM, PF

CPG, Central pattern generator; CF, CPG frequency modulation; CP, CPG phase modulation; CM, CPG magnitude modulation; PF, Pure feedback; FM, Forward model; LM, Learning model; MS, Muscle stiffness adaptation; ML, Other machine learning approaches.

5. CONCLUSION

Although walking animals create adaptive locomotor behavior by skillfully manipulating their complicated and redundant musculoskeletal systems, the underlying mechanisms are still unclear. Designing the control architecture for legged robots to autonomously achieve such adaptability is still a challenge. Although some legged robots produced adaptive locomotor behaviors by purely engineering approaches without inspiration from biological systems, neurophysiological findings such as CPG organizations and sensorimotor interactions are useful for designing the control system of legged robots. Robot experiments with CPG models and sensory feedbacks are insightful from a dynamic viewpoint for understanding gait generation and adaptation in a self-organizing manner among neural dynamics, body dynamics, and environment. In this review, we showed adaptive interlimb coordination in the locomotion of animals and legged robots induced by various factors, such as locomotion speed, environmental situation, body properties, and tasks. We also showed characteristic properties

of adaptive interlimb coordination, such as gait hysteresis and different time-scale adaptations. Legged robots are becoming a valuable tool for understanding the locomotion mechanism including interlimb coordination. In the future, together with the improvement of robotics systems, such as actuators and sensors, it will be important to enhance biological plausibility and feasibility by the integration with sophisticated models of neural and musculoskeletal systems, such as the Hodgkin-Huxley model and the muscle-tendon unit model, and to extract dynamical features by integration with simple models, such as the template model (Full and Koditschek, 1999; Holmes et al., 2006). Furthermore, it will also be important to further improve and develop analytical methods, such as phase reduction theory (Kuramoto, 1984) and synergy analysis (Ivanenko et al., 2004; Latash, 2008), to clarify essential factors from multiple and redundant data.

AUTHOR CONTRIBUTIONS

SA and PM contributed to the conception and design of the paper. SA, PM, and YA reviewed the relevant literature and wrote the paper. FM and FW revised the paper critically for important intellectual content. All authors approved the paper for publication.

ACKNOWLEDGMENTS

This paper is supported in part by Grant-in-Aid for Young Scientists (A) 17H04914 from the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) of Japan, by the Inamori Foundation, Japan, by the Kyoto Technoscience Center, Japan, and by the Centre for BioRobotics (CBR) at the University of Southern Denmark (SDU, Denmark).

REFERENCES

Aguilar, J., Zhang, T., Qian, F., Kingsbury, M., McInroe, B., Mazouchova, N., et al. (2016). A review on locomotion robophysics: the study of movement at the intersection of robotics, soft matter and dynamical systems. *Rep. Prog. Phys.* 79:110001. doi: 10.1088/0034-4885/79/11/110001

Alexander, R. M. C. N. (1988). Why mammals gallop. *Am. Zool.* 28, 237–245. doi: 10.1093/icb/28.1.237

Alexander, R. M. C. N., and Jayes, A. S. (1983). A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *J. Zool. Lond.* 201, 135–152. doi: 10.1111/j.1469-7998.1983.tb04266.x

Ambe, Y., Aoi, S., Nachstedt, T., Manoonpong, P., Wörgötter, F., and Matsuno, F. (2015). "Embodied sensorimotor interaction for hexapod locomotion," in *Proceedings of International Symposium on Swarm Behavior Bio-Inspired Robotics* (Kyoto), 340–343.

Ambe, Y., and Matsuno, F. (2016). "Leg-grope walk": strategy for walking on fragile irregular slopes as a quadruped robot by force distribution. *Robomech. J.* 3:7. doi: 10.1186/s40648-016-0046-2

Ambe, Y., Nachstedt, T., Manoonpong, P., Wörgötter, F., Aoi, S., and Matsuno, F. (2013). "Stability analysis of a hexapod robot driven by distributed nonlinear oscillators with a phase modulation mechanism," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots System* (Tokyo), 5087–5092.

Aoi, S., Egi, Y., Sugimoto, R., Yamashita, T., Fujiki, S., and Tsuchiya, K. (2012). Functional roles of phase resetting in the gait transition of a biped robot from quadrupedal to bipedal locomotion. *IEEE Trans. Robot.* 28, 1244–1259. doi: 10.1109/TRO.2012.2205489

Aoi, S., Egi, Y., and Tsuchiya, K. (2013a). Instability-based mechanism for body undulations in centipede locomotion. *Phys. Rev. E* 87:012717. doi: 10.1103/PhysRevE.87.012717

Aoi, S., and Funato, T. (2016). Neuromusculoskeletal models based on the muscle synergy hypothesis for the investigation of adaptive motor control in locomotion via sensory-motor coordination. *Neurosci. Res.* 104, 88–95. doi: 10.1016/j.neures.2015.11.005

Aoi, S., Katayama, D., Fujiki, S., Tomita, N., Funato, T., Yamashita, T., et al. (2013b). A stability-based mechanism for hysteresis in the walk–trot transition in quadruped locomotion. *J. R. Soc. Interface* 10:20120908. doi: 10.1098/rsif.2012.0908

Aoi, S., Kondo, T., Hayashi, N., Yanagihara, D., Aoki, S., Yamaura, H., et al. (2013c). Contributions of phase resetting and interlimb coordination to the adaptive control of hindlimb obstacle avoidance during locomotion in rats: a simulation study. *Biol. Cybern.* 107, 201–216. doi: 10.1007/s00422-013-0546-6

Aoi, S., Ogihara, N., Funato, T., Sugimoto, Y., and Tsuchiya, K. (2010). Evaluating functional roles of phase resetting in generation of adaptive human bipedal walking with a physiologically based model of the spinal pattern generator. *Biol. Cybern.* 102, 373–387. doi: 10.1007/s00422-010-0373-y

Aoi, S., Sasaki, H., and Tsuchiya, K. (2007). A multilegged modular robot that meanders: investigation of turning maneuvers using its inherent dynamic characteristics *SIAM J. Appl. Dyn. Syst.* 6, 348–377. doi: 10.1137/060664756

Aoi, S., Tanaka, T., Fujiki, S., Funato, T., Senda, K., and Tsuchiya, K. (2016). Advantage of straight walk instability in turning maneuver of multilegged locomotion: a robotics approach. *Sci. Rep.* 6:30199. doi: 10.1038/srep30199

Aoi, S., and Tsuchiya, K. (2005). Locomotion control of a biped robot using nonlinear oscillators. *Auton. Robots* 19, 219–232. doi: 10.1007/s10514-005-4051-1

Aoi, S., and Tsuchiya, K. (2007). Adaptive behavior in turning of an oscillator-driven biped robot. *Auton. Robots* 23, 37–57. doi: 10.1007/s10514-007-9029-8

Aoi, S., Yamashita, T., and Tsuchiya, K. (2011). Hysteresis in the gait transition of a quadruped investigated using simple body mechanical and oscillator network models. *Phys. Rev. E* 83:061909. doi: 10.1103/PhysRevE.83.061909

Arena, E., Arena, P., Strauss, R., and Patané, L. (2017). Motor-skill learning in an insect inspired neuro-computational control system. *Front. Neurorobot.* 11:12. doi: 10.3389/fnbot.2017.00012

Arena, P. (2000). The Central Pattern Generator: a paradigm for artificial locomotion. *Soft Comput.* 4, 251–266. doi: 10.1007/s005000000051

Asa, K., Ishimura, K., and Wada, M. (2009). Behavior transition between biped and quadrupedwalking by using bifurcation. *Robot. Auton. Syst.* 57, 155–160. doi: 10.1016/j.robot.2008.04.005

Barikhan, S. S., Wörgötter, F., and Manoonpong, P. (2014). "*Multiple decoupled CPGs with local sensory feedback for adaptive locomotion behaviors of bio-inspired walking robots*," in *From Animals to Animats 13, Lecture Notes in Computer Science,* eds A. P. del Pobil, E. Chinellato, E. Martinez-Martin, J. Hallam, E. Cervera, and A. Morales (Cham: Springer), 65–75.

Bässler, U. (1976). Reversal of a reflex to a single motoneuron in the stick insect Carausius morosus. *Biol. Cybern.* 24, 47–49. doi: 10.1007/BF00365594

Bässler, U., and Büschges, A. (1998). Pattern generation for stick insect walking movements–multisensory control of a locomotor program. *Brain Res. Rev.* 27, 65–88. doi: 10.1016/S0165-0173(98)00006-X

Bässler, U., and Wegner, U. (1983). Motor output of the denervated thoracic ventral nerve cord in the stick insect Carausius morosus. *J. Exp. Biol.* 105, 127–145.

Bender, J. A., Simpson, E. M., Tietz, B. R., Daltorio, K. A., Quinn, R. D., and Ritzmann, R. E. (2011). Kinematic and behavioral evidence for a distinction between trotting and ambling gaits in the cockroach Blaberus discoidalis. *J. Exp. Biol.* 214, 2057–2064. doi: 10.1242/jeb.056481

Berendes, V., Zill, S. N. Büschges, A., and Bockemühl, T. (2016). Speed-dependent interplay between local pattern-generating activity and sensory signals during walking in Drosophila. *J. Exp. Biol.* 219, 3781–3793. doi: 10.1242/jeb.146720

Blaesing, B., and Cruse, H. (2004). Stick insect locomotion in a complex environment: climbing over large gaps. *J. Exp. Biol.* 207, 1273–1286. doi: 10.1242/jeb.00888

Bläsing, B. (2006). Crossing large gaps: a simulation study of stick insect behavior. *Adapt. Behav.* 14, 265–285. doi: 10.1177/105971230601400307

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science* 314, 1118–1121. doi: 10.1126/science.1133687

Brown, T. G. (1911). The intrinsic factors in the act of progression in the mammal. *Proc. R. Soc. Lond. B* 84, 308–319. doi: 10.1098/rspb.1911.0077

Buchanan, J. T. (1992). Neural network simulations of coupled locomotor oscillators in the lamprey spinal cord. *Biol. Cybern.* 66, 367–374. doi: 10.1007/BF00203673

Buchli, J., and Ijspeert, A. J. (2008). Self-organized adaptive legged locomotion in a compliant quadruped robot. *Auton. Robots* 25, 331–347. doi: 10.1007/s10514-008-9099-2

Buchli, J., Righetti, L., and Ijspeert, A. J. (2006). Engineering entrainment and adaptation in limit cycle systems–From biological inspiration to applications in robotics. *Biol. Cybern.* 95, 645–664. doi: 10.1007/s00422-006-0128-y

Bungay, S. D., and Campbell, S. A. (2009). Modelling a respiratory central pattern generator neuron in Lymnaea stagnalis. *Can. Appl. Math. Quart.* 17, 283–291. Available online at: http://www.math.ualberta.ca/ami/CAMQ/table_of_content/vol_17/17_2a.htm

Büschges, A. (1995). Role of local nonspiking interneurons in the generation of rhythmic motor activity in the stick insect. *J. Neurobiol.* 27, 488–512. doi: 10.1002/neu.480270405

Büschges, A., Akay, T., Gabriel, J. P., and Schmidt, J. (2008). Organizing network action for locomotion: insights from studying insect walking. *Brain Res. Rev.* 57, 162–171. doi: 10.1016/j.brainresrev.2007.06.028

Büschges, A., Schmitz, J., and Bässler, U. (1995). Rhythmic patterns in the thoracic nerve cord of the stick insect induced by pilocarpine. *J. Exp. Biol.* 198, 435–456.

Buschmann, T., Ewald, A., von Twickel, A., and Büschges, A. (2015). Controlling legs for locomotion–insights from robotics and neurobiology. *Bioinspir. Biomim.* 10:041001. doi: 10.1088/1748-3190/10/4/041001

Canavier, C. C., Butera, R. J., Dror, R. O., Baxter, D. A., Clark, J. W., and Byrne, J. H. (1997). Phase response characteristics of model neurons determine which patterns are expressed in a ring circuit model of gait generation. *Biol. Cybern.* 77, 367–380. doi: 10.1007/s004220050397

Cataldo, E., Byrne, J. H., and Baxter, D. A. (2006). "Computational model of a central pattern generator," in *Computational Methods in Systems Biology, Lecture Notes in Computer Science,* ed C. Priami (Berlin: Springer), 242–256.

Chung, S. J., and Slotine, J. J. (2010). "On synchronization of coupled Hopf-Kuramoto oscillators with phase delays," in *Proceedings of IEEE International Conference on Decision Control* (Atlanta, GA), 3181–3187.

Conway, B. A., Hultborn, H., and Kiehn, O. (1987). Proprioceptive input resets central locomotor rhythm in the spinal cat. *Exp. Brain Res.* 68, 643–656. doi: 10.1007/BF00249807

Courtine, G., and Schieppati, M. (2003). Human walking along a curved path. II. Gait features and EMG patterns. *Eur. J. Neurosci.* 18, 191–205. doi: 10.1046/j.1460-9568.2003.02737.x

Couzin-Fuchs, E., Kiemel, T., Gal, O., Ayali, A., and Holmes, P. (2015). Intersegmental coupling and recovery from perturbations in freely running cockroaches. *J. Exp. Biol.* 218, 285–297. doi: 10.1242/jeb.112805

Cruse, H. (1990). What mechanisms coordinate leg movement in walking arthropods? *Trends Neurosci.* 13, 15–21. doi: 10.1016/0166-2236(90)90057-H

Cruse, H., Kindermann, T., Schumm, M., Dean, J., and Schmitz, J. (1998). Walknet—a biologically inspired network to control six-legged walking. *Neural Netw.* 11, 1435–1447. doi: 10.1016/S0893-6080(98)00067-7

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature* 521, 503–507. doi: 10.1038/nature14422

Dasgupta, S., Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2015). Distributed recurrent neural forward models with synaptic adaptation and CPG-based control for complex behaviors of walking robots. *Front. Neurorobot.* 9:10. doi: 10.3389/fnbot.2015.00010

Delcomyn, F. (1971). The locomotion of the cockroach *Periplaneta americana*. *J. Exp. Biol.* 54, 443–452.

Di Canio, G., Stoyanov, S., Larsen, J.C., Hallam, J., Kovalev, A., Kleinteich, T., et al. (2016). A robot leg with compliant tarsus and its neural control for efficient and adaptive locomotion on complex terrains. *Artif. Life Robot.* 21, 274–281. doi: 10.1007/s10015-016-0296-3

Di Canio, G., Stoyanov, S., Balmori, I. T., Larsen, J. C., and Manoonpong, P. (2016). "Adaptive combinatorial neural control for robust locomotion of a biped robot," in *From Animals to Animats 14, Lecture Notes in Computer Science*, eds E. Tuci, A. Giagkos, M. Wilson, and J. Hallam (Cham: Springer), 317–328.

Diedrich, F. J., and Warren, W. H., Jr. (1998). Why change gaits? Dynamics of the walk-run transition. *J. Exp. Psychol. Hum. Percept. Perform.* 21, 183–202. doi: 10.1037/0096-1523.21.1.183

Dietz, Z., Zijlstra, W., and Duysens, J. (1994). Human neuronal interlimb coordination during split-belt locomotion. *Exp. Brain Res.* 101, 513–520. doi: 10.1007/BF00227344

Dürr, V., Schmitz, J., and Cruse, H. (2004). Behaviour-based modelling of hexapod locomotion: linking biology and technical application. *Arth. Struct. Develop.* 33, 237–250. doi: 10.1016/j.asd.2004.05.004

Duysens, J. (1997). Fluctuations in sensitivity to rhythm resetting effects during the cat's step cycle. *Brain Res.* 133, 190–195. doi: 10.1016/0006-8993(77)90063-4

Endo, G., Morimoto, J. Nakanishi, J., and Cheng, G. (2004). "An empirical exploration of a neural oscillator for biped locomotion control," in *Proceedings of IEEE International Conference on Robotics and Automation* (New Orleans, LA), 3036–3042.

Farley, C. T., and Taylor, C. R. (1991). A mechanical trigger for the trot-gallop transition in horses. *Science* 253, 306–308. doi: 10.1126/science.1857965

Flash, T., and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *J. Neurosci.* 5, 1688–1703.

Forssberg, H., and Grillner, S. (1973). The locomotion of the acute spinal cat injected with clonidine i.v. *Brain Res.* 50, 184–186. doi: 10.1016/0006-8993(73)90606-9

Forssberg, H., Grillner, S., Halbertsma, J., and Rossignol, S. (1980). The locomotion of the low spinal cat. II. Interlimb coordination. *Acta. Physiol. Scand.* 108, 283–295. doi: 10.1111/j.1748-1716.1980.tb06534.x

Foth, E., and Graham, D. (1983). Influence of loading parallel to the body axis on the walking coordination of an insect. II. Contralateral effects. *Biol. Cybern.* 48, 149–157. doi: 10.1007/BF00318082

Freitas, S. M. S. F., Duarte, M., and Latash, M. L. (2006). Two kinematic synergies in voluntary whole-body movements during standing. *J. Neurophysiol.* 95, 636–645. doi: 10.1152/jn.00482.2005

Frigon, A., Hurteau, M.-F., Thibaudier, Y., Leblond, H., Telonio, A., and D'Angelo, G. (2013). Split-belt walking alters the relationship between locomotor phases and cycle duration across speeds in intact and chronic spinalized adult cats. *J. Neurosci.* 33, 8559–8566. doi: 10.1523/JNEUROSCI.3931-12.2013

Frigon, A., Sirois, J., and Gossard, J.-P. (2010). Effects of ankle and hip muscle afferent inputs on rhythm generation during fictive locomotion. *J. Neurophysiol.* 103, 1591–1605. doi: 10.1152/jn.01028.2009

Fuchs, E., Holmes, P., Kiemel, T., and Ayali, A. (2011). Intersegmental coordination of cockroach locomotion: adaptive control of centrally coupled pattern generator circuits. *Front. Neural Circuits* 4:125. doi: 10.3389/fncir.2010.00125

Fujiki, S., Aoi, S., Funato, T., Tomita, N., Senda, K., and Tsuchiya, K. (2013a). Hysteresis in the metachronal-tripod gait transition of insects: a modeling study. *Phys. Rev. E* 88:012717. doi: 10.1103/PhysRevE.88.012717

Fujiki, S., Aoi, S., Funato, T., Tomita, N., Senda, K., and Tsuchiya, K. (2015). Adaptation mechanism of interlimb coordination in human split-belt treadmill walking through learning of foot contact timing: a robotics study. *J. R. Soc. Interface* 12:20150542. doi: 10.1098/rsif.2015.0542

Fujiki, S., Aoi, S., Yamashita, T., Funato, T., Tomita, N., Senda, K., et al. (2013b). Adaptive splitbelt treadmill walking of a biped robot using nonlinear oscillators with phase resetting. *Auton. Robots* 35, 15–26. doi: 10.1007/s10514-013-9331-6

Fukuoka, Y., Habu, Y., and Fukui, T. (2015). A simple rule for quadrupedal gait generation determined by leg loading feedback: a modeling study. *Sci. Rep.* 5:8169. doi: 10.1038/srep08169

Fukuoka, Y., and Kimura, H. (2009). Dynamic locomotion of a biomorphic quadruped Tekken robot using various gaits: walk, trot, free-gait and bound. *Appl. Bionics Biomech.* 6, 63–71. doi: 10.1155/2009/743713

Fukuoka, Y., Kimura, H., and Cohen, A. (2003). Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *Int. J. Robot. Res.* 22, 187–202. doi: 10.1177/0278364903022003004

Full, R. J., and Koditschek, D.E. (1999). Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *J. Exp. Biol.* 202, 3325–3332.

Funato, T., Aoi, S., Oshima, H., and Tsuchiya, K. (2010). Variant and invariant patterns embedded in human locomotion through whole body kinematic coordination. *Exp. Brain Res.* 205, 497–511. doi: 10.1007/s00221-010-2385-1

Funato, T. Yamamoto, Y., Aoi, S., Imai, T., Aoyagi, T., Tomita, N., et al. (2016). Evaluation of the phase-dependent rhythm control of human walking using phase response curves. *PLoS Comput. Biol.* 12:e1004950. doi: 10.1371/journal.pcbi.1004950

Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2014). Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots. *Front. Neurorobot.* 8:3. doi: 10.3389/fnbot.2014.00003

Golubitsky, M., Stewart, I., Buono, P. L., and Collins, J. J. (1999). Symmetry in locomotor central pattern generators and animal gaits. *Nature* 401, 693–695. doi: 10.1038/44416

Grabowska, M., Godlewska, E., Schmidt, J., and Daun-Gruhn, S. (2012). Quadrupedal gaits in hexapod animals - inter-leg coordination in free-walking adult stick insects. *J. Exp. Biol.* 215, 4255–4266. doi: 10.1242/jeb.073643

Graham, D. (1972). A behavioural analysis of the temporal organisation of walking movements in the 1st instar and adult stick insect (Carausius morosus). *J. Comp. Physiol.* 81, 23–52. doi: 10.1007/BF00693548

Graham, D. (1977). The effect of amputation and leg restraint on the free walking coordination of the stick insect Carausius morosus. *J. Comp. Physiol. A* 116, 91–116. doi: 10.1007/BF00605519

Griffin, T. M., Kram, R., Wickler, S. J., and Hoyt, D. F. (2004). Biomechanical and energetic determinants of the walk-trot transition in horses. *J. Exp. Biol.* 207, 4215–4223. doi: 10.1242/jeb.01277

Grillner, S. (1975). Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiol. Rev.* 55, 247–304.

Grinke, E., Tetzlaff, C., Wörgötter, F., and Manoonpong, P. (2015). Synaptic plasticity in a recurrent neural network for versatile and adaptive behaviors of a walking robot. *Front. Neurorobot.* 9:11. doi: 10.3389/fnbot.2015.00011

Harischandra, N., Knüsel, J., Kozlov, A., Bicanski, A., Cabelguen, J.-M., Ijspeert, A., et al. (2011). Sensory feedback plays a significant role in generating walking gait and in gait transition in salamanders: a simulation study. *Front. Neurorobot.* 5:3. doi: 10.3389/fnbot.2011.00003

Heglund, N. C., and Taylor, C. R. (1998). Speed, stride frequency and energy cost per stride: how do they change with body size and gait? *J. Exp. Biol.* 138, 301–318.

Hildebrand, M. (1965). Symmetrical gaits of horses. *Science* 150, 701–708.

Hoepflinger, M. A., Hutter, M., Gehring, C., Bloesch, M., and Siegwart, R. (2013). "Unsupervised identification and prediction of foothold robustness," in *Proceeding of IEEE International Conference on Robotica Automation* (Karlsruhe), 3293–3298.

Hoepflinger, M. A., Remy, C. D., Hutter, M., Spinello, L., and Siegwart, R. (2010). "Haptic terrain classification for legged robots," in *Proceedings of IEEE International Conference on Robotics Automation* (Anchorage, AK), 2828–2833.

Holmes, P., Full, R. J., Koditschek, D., and Guckenheimer, J. (2006). The dynamics of legged locomotion: Models, analyses, and challenges *SIAM Rev.* 48, 207–304. doi: 10.1137/S0036144504445133

Hoyt, D. F., and Taylor, C. R. (1981). Gait and the energetics of locomotion in horses. *Nature* 292, 239–240. doi: 10.1038/292239a0

Hreljac, A. (1993). Preferred and energetically optimal gait transition speeds in human locomotion. *Med. Sci. Sports Exerc.* 25, 1158–1162. doi: 10.1249/00005768-199310000-00012

Hughes, G. M. (1952). The co-ordination of insect movements. I. The walking movements of insects. *J. Exp. Biol.* 29, 267–285.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J. M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *Science* 315, 1416–1420. doi: 10.1126/science.1138353

Ito, S., Yuasa, H., Luo, Z., Ito, M., and Yanagihara, D. (1998). A mathematical model of adaptive behavior in quadruped locomotion. *Biol. Cybern.* 78, 337–347. doi: 10.1007/s004220050438

Ivanenko, Y. P., Cappellini, G., Dominici, N., Poppele, R. E., and Lacquaniti, F. (2007). Modular control of limb movements during human locomotion. *J. Neurosci.* 27, 11149–11161. doi: 10.1523/JNEUROSCI.2644-07.2007

Ivanenko, Y. P., Poppele, R. E., and Lacquaniti, F. (2004). Five basic muscle activation patterns account for muscle activity during human locomotion. *J. Physiol.* 556, 267–282. doi: 10.1113/jphysiol.2003.057174

Kimura, H., Fukuoka, Y., and Cohen, A. H. (2007a). Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *Int. J. Robot. Res.* 26, 475–490. doi: 10.1177/0278364907078089

Kimura, H., Fukuoka, Y., and Cohen, A. H. (2007b). Biologically inspired adaptive walking of a quadruped robot. *Phil. Trans. R. Soc. A* 365, 153–170. doi: 10.1098/rsta.2006.1919

Kimura, S., Yano, M., and Shimizu, H. (1993). A self-organizing model of walking patterns of insects. *Biol. Cybern.* 69, 183–193. doi: 10.1007/BF00198958

Kindermann, T. (2002). Behavior and adaptability of a six-legged walking system with highly distributed control. *Adapt. Behav.* 9, 16–41. doi: 10.1177/105971230200900103

Kirchner, F., Spenneberg, D., and Linnemann, R. (2002). "A biologically inspired approach toward robust real-world locomotion in legged robots," in *Neurotechnology for Biomimetic Robots,* eds J. Ayers, J. Davis, and A. Rudolph, (Cambridge, MA: MIT Press), 419–447.

Knops, S., Tóth, T. I., Guschlbauer, C., Gruhn, M., and Daun-Gruhn, S. (2013). A neuromechanical model for the neuronal basis of curve walking in the stick insect. *J. Neurophysiol.* 109, 679–691. doi: 10.1152/jn.00648.2012

Kobayashi, T., Aoyama, T., Sekiyama, K. and Fukuda, T. (2015). Selection algorithm for locomotion based on the evaluation of falling risk. *IEEE Trans. Robot.* 31, 750–765. doi: 10.1109/TRO.2015.2426451

Kuramoto, Y. (1984). *Chemical Oscillations, Waves, and Turbulences*. Berlin: Springer-Verlag.

Latash, M. L. (2008). *Synergy*. New York, NY: Oxford University Press.

Lewinger, W. A., and Quinn, R. D. (2011). Neurobiologicallyï£¡based control system for an adaptively walking hexapod. *Indust. Robot. Int. J.* 38, 258–263. doi: 10.1108/01439911111122752

Li, C., Umbanhowar, P. B., Komsuoglu, H., Koditschek, D. E., and Goldman, D. I. (2009). Sensitive dependence of the motion of a legged robot on granular media. *Proc. Natl. Acad. Sci. U.S.A.* 106, 3029–3034. doi: 10.1073/pnas.0809095106

Li, C., Zhang, T., and Goldman, D. I. (2013). A terradynamics of legged locomotion on granular media. *Science* 339, 1408–1411. doi: 10.1126/science.1229163

MacKay-Lyons, M. (2002). Central pattern generation of locomotion: a review of the evidence. *Phys. Ther.* 82, 69–83. doi: 10.1093/ptj/82.1.69

Manoonpong, P., Geng, T., Kulvicius, T., Porr, B., and Wörgötter, F. (2007). Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput. Biol.* 3:e134. doi: 10.1371/journal.pcbi.0030134

Manoonpong, P., Parlitz, U., and Wörgötter, F. (2013). Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines. *Front. Neural Circuits* 7:12. doi: 10.3389/fncir.2013.00012

Manoonpong, P., Pasemann, F., and Wörgötter, F. (2008). Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines. *Robot. Auton. Syst.* 56, 265–288. doi: 10.1016/j.robot.2007.07.004

Manton, S. M. (1965). The evolution of arthropodan locomotory mechanisms. Part 8. Functional requirements and body design in Chilopoda, together with a comparative account of their skeleto-muscular systems and an Appendix on A comparison between burrowing forces of annelids and chilopods and its bearing upon the evolution of the arthropodan haemocoel. *J. Linn. Soc. Zool.* 46, 251–484. doi: 10.1111/j.1096-3642.1965.tb00500.x

Margaria, R. (1938). Sulla fisiologia e specialmente sul consumo energetico della marcia e della corsa a varie velocita ed inclinazioni del terreno. *Atti Acc. Naz. Lincei.* 7, 299–368.

Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol. Cybern.* 52, 367–376. doi: 10.1007/BF00449593

Maufroy, C., Kimura, H., and Takase, K. (2010). Integration of posture and rhythmic motion controls in quadrupedal dynamic walking using phase modulations based on leg loading/unloading. *Auton. Robots* 28, 331–353. doi: 10.1007/s10514-009-9172-5

Minetti, A. E., Ardigo, L. P., and Saibene, F. (1994). Mechanical determinants of the minimum energy cost of gradient running in humans. *J. Exp. Biol.* 195, 211–225.

Morton, S. M., and Bastian, A. J. (2006). Cerebellar contributions to locomotor adaptations during splitbelt treadmill walking. *J. Neurosci.* 26, 9107–9116. doi: 10.1523/JNEUROSCI.2622-06.2006

Müller, U., and Cruse, H. (1991a). The contralateral coordination of walking legs in the crayfish *Astacus leptodactylus*. I. Experimental results. *Biol. Cybern.* 64, 429–436. doi: 10.1007/BF00224710

Müller, U., and Cruse H. (1991b). The contralateral coordination of walking in the crayfish *Astacus leptodactylus*. II. Model calculations. *Biol. Cybern*. 64, 437–446. doi: 10.1007/BF00224711

Muybridge, E. (1957). *Animals in Motion*. New York, NY: Dover Publications.

Nachstedt, T., Tetzlaff, C., and Manoonpong, P. (2017). Fast dynamical coupling enhances frequency adaptation of oscillators for robotic locomotion control. *Front. Neurorobot.* 11:14. doi: 10.3389/fnbot.2017.00014

Nomura, T., Kawa, K., Suzuki, Y., Nakanishi, M., and Yamasaki, T. (2009). Dynamic stability and phase resetting during biped gait. *Chaos* 19:026103. doi: 10.1063/1.3138725

Orlovsky, G. N., Deliagina, T., and Grillner, S. (1999). *Neuronal Control of Locomotion: from Mollusc to Man*. New York, NY: Oxford University Press.

Otoda, Y., Kimura, H., and Takase, K. (2009). Construction of a gait adaptation model in human split-belt treadmill walking using a two-dimensional biped robot. *Adv. Robot.* 23, 535–561. doi: 10.1163/156855309X420057

Owaki, D., Kano, T., Nagasawa, K., Tero, A., and Ishiguro, A. (2013). Simple robot suggests physical interlimb communication is essential for quadruped walking. *J. R. Soc. Interface* 10:20120669. doi: 10.1098/rsif.2012.0669

Owaki, D., and Ishiguro, A. (2017). A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping. *Sci. Rep.* 7:277. doi: 10.1038/s41598-017-00348-9

Pearson, K. (1976). The control of walking. *Sci. Am.* 235, 72–86. doi: 10.1038/scientificamerican1276-72

Pearson, K. G., and Franklin, R. (1984). Characteristics of leg movement and patterns of coordination in locusts walking on rough terrain. *Int. J. Robot. Res.* 3, 101–112. doi: 10.1177/027836498400300209

Pelletier, Y., and Caissie, R. (2001). Behavioural and physical reactions of the Colorado potato beetle, *Leptinotarsa decemlineata* (Say) (Coleoptera: Chrysomelidae) walking on a slanted surface. *Biol. Cybern.* 84, 269–277. doi: 10.1007/s004220000214

Reisman, D. S., Block, H. J., and Bastian, A. J. (2005). Interlimb coordination during locomotion: What can be adapted and stored? *J. Neurophysiol.* 94, 2403–2415. doi: 10.1152/jn.00089.2005

Rabinovich, M. I., and Abarbanel, H. D. I. (1998). The role of chaos in neural systems. *Neuroscience* 87, 5–14. doi: 10.1016/S0306-4522(98)00091-8

Raynor, A. J., Yi, C. J., Abernethy, B., and Jong, Q. J. (2002). Are transitions in human gait determined by mechanical, kinetic or energetic factors? *Hum. Mov. Sci.* 21, 785–805. doi: 10.1016/S0167-9457(02)00180-X

Ren, G., Chen, W., Dasgupta, S., Kolodziejski, C. Wörgötter, F., and Manoonpong, P. (2015). Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation. *Inform. Sci.* 294, 666–682. doi: 10.1016/j.ins.2014.05.001

Rybak, I. A., Shevtsova, N. A., Lafreniere-Roula, M., and McCrea, D. A. (2006). Modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *J. Physiol.* 577, 617–639. doi: 10.1113/jphysiol.2006.118703

Schilling, M., and Cruse, H. (2017). ReaCog, a minimal cognitive controller based on recruitment of reactive systems. *Front. Neurorobot.* 11:3. doi: 10.3389/fnbot.2017.00003

Schilling, M., Cruse, H., and Arena, P. (2007). Hexapod Walking: an expansion to Walknet dealing with leg amputations and force oscillations. *Biol. Cybern.* 96, 323–340. doi: 10.1007/s00422-006-0117-1

Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H. (2013a). Walknet, a bio-inspired controller for hexapod walking. *Biol. Cybern.* 107, 397–419. doi: 10.1007/s00422-013-0563-5

Schilling, M., Paskarbeit, J., Hoinville, T., Hüffmeier, A., Schneider, A., Schmitz, J., et al. (2013b). A hexapod walker using a heterarchical architecture for action selection. *Front. Comput. Neurosci.* 7:126. doi: 10.3389/fncom.2013.00126

Schmitz, J., Schneider, A., Schilling, M., and Cruse, H. (2008). No need for a body model: positive velocity feedback for the control of an 18-DOF robot walker. *Appl. Bionics Biomech.* 5, 135–147. doi: 10.1155/2008/737239

Schneider, A., Paskarbeit, J., Schaeffersmann, M., and Schmitz, J. (2012). "Hector, a new hexapod robot platform with increased mobility-control approach, design and communication," in *Advances in Autonomous Mini Robots,* eds U. Rückert, S. Joaquin, and W. Felix (Berlin: Springer), 249–264.

Schomburg, E. D., Petersen, N., Barajon, I., and Hultborn, H. (1998). Flexor reflex afferents reset the step cycle during fictive locomotion in the cat. *Exp. Brain Res.* 122, 339–350. doi: 10.1007/s002210050522

Schöner, G., Jiang, W. Y., and Kelso, J. A. S. (1990). A synergetic theory of quadrupedal gaits and gait transitions. *J. Theor. Biol.* 142, 359–391. doi: 10.1016/S0022-5193(05)80558-2

Shim, Y., and Husbands, P. (2012). Chaotic exploration and learning of locomotion behaviors. *Neural Comput.* 24, 2185–222. doi: 10.1162/NECO_a_00313

Spirito, C. P., and Mushrush, D. L. (1979). Interlimb coordination during slow walking in the cockroach I. effects of substrate alterations. *J. Exp. Biol.* 78, 233–243.

Steingrube, S., Timme, M., Wörgötter, F., and Manoonpong, P. (2010). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nat. Phys.* 6, 224–230. doi: 10.1038/nphys1508

Strauß, R., and Heisenberg, M. (1990). Coordination of legs during straight walking and turning in *Drosophila melanogaster*. *J. Comp. Physiol. A* 167, 403–412. doi: 10.1007/BF00192575

Szczecinski, N. S., Getsy, A. P., Martin, J. P., Ritzmann, R. E., and Quinn, R. D. (in press). Mantisbot is a robotic model of visually guided motion in the praying mantis. *Arthropod. Struct. Dev.* doi: 10.1016/j.asd.2017.03.001

Taga, G. (1995). A model of the neuro-musculo-skeletal system for human locomotion I. Emergence of basic gait. *Biol. Cybern.* 73, 97–111. doi: 10.1007/BF00204048

Taga, G. Yamaguchi, Y., and Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biol. Cybern.* 65, 147–159. doi: 10.1007/BF00198086

Tang, T. P., and Macmillan, D. L. (1986). The effects of sensory manipulation upon interlimb coordination during fast walking in the cockroach. *J. Exp. Biol.* 125, 107–117.

Tokuda, K., Toda, T., Koji, Y., Konyo, M., Tadokoro, S., and Alain, P. (2003). "Estimation of fragile ground by foot pressure sensor of legged robot," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (Kobe), 447–453.

Tóth, T. I., and Daun-Gruhn, S. (2016). A three-leg model producing tetrapod and tripod coordination patterns of ipsilateral legs in the stick insect. *J. Neurophysiol.* 115, 887–906. doi: 10.1152/jn.00693.2015

Traven, H. G., Brodin, L., Lansner, A., Ekeberg, O., Wallen, P., and Grillner, S. (1993). Computer simulations of NMDA and non-NMDA receptor-mediated synaptic drive: sensory and supraspinal modulation of neurons and small networks. *J. Neurophysiol.* 70, 695–709.

Tsuchiya, K., Aoi, S., and Tsujita, K. (2002). "Locomotion control of a multi-legged locomotion robot using oscillators," in *Proceedings of IEEE International Conference on Systems, Man Cybernetics* (Hammamet).

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optical trajectory in human multi-joint arm movement - minimum torque-change model. *Biol. Cybern.* 61, 89–101. doi: 10.1007/BF00204593

Wickler, S. J., Hoyt, D. F., Cogger, E. A., and Myers, G. (2003). The energetics of the trot-gallop transition. *J. Exp. Biol.* 206, 1557–1564. doi: 10.1242/jeb.00276

Wilson, D. M. (1966). Insect walking. *Annu. Rev. Entomol.* 11, 103–122. doi: 10.1146/annurev.en.11.010166.000535

Wosnitza, A., Bockemühl, T., Dübbert, M., Scholz, H., and Büschges, A. (2013). Inter-leg coordination in the control of walking speed in *Drosophila*. *J. Exp. Biol.* 216, 480–491. doi: 10.1242/jeb.078139

Xiong, X., Wörgötter, F., and Manoonpong, P. (2014). Neuromechanical control for hexapedal robot walking on challenging surfaces and surface classification. *Robot. Auton. Syst.* 62, 1777–1789. doi: 10.1016/j.robot.2014.07.008

Xiong, X., Wörgötter, F., and Manoonpong, P. (2015). Adaptive and energy efficient walking in a hexapod robot under neuromechanical control and sensorimotor learning. *IEEE Trans. Cybern.* 46, 2521–2534. doi: 10.1109/TCYB.2015.2479237

Yanagihara, D., and Kondo, I. (1996). Nitric oxide plays a key role in adaptive control of locomotion in cat. *Proc. Nat. Acad. Sci. U.S.A.* 93, 13292–13297. doi: 10.1073/pnas.93.23.13292

Yu, J., Tan, M., Chen, J., and Zhang, J. (2014). A survey on CPG-inspired control models and system implementation. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 441–456. doi: 10.1109/TNNLS.2013.2280596

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Aoi, Manoonpong, Ambe, Matsuno and Wörgötter. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 09 June 2017
doi: 10.3389/fnbot.2017.00029

A Minimal Model Describing Hexapedal Interlimb Coordination: The Tegotae-Based Approach

Dai Owaki[1]*, *Masashi Goda*[1], *Sakiko Miyazawa*[1] and *Akio Ishiguro*[1,2]

[1] Research Institute of Electrical Communication, Tohoku University, Sendai, Japan, [2] Japan Science and Technology Agency, CREST, Saitama, Japan

Insects exhibit adaptive and versatile locomotion despite their minimal neural computing. Such locomotor patterns are generated via coordination between leg movements, i.e., an interlimb coordination, which is largely controlled in a distributed manner by neural circuits located in thoracic ganglia. However, the mechanism responsible for the interlimb coordination still remains elusive. Understanding this mechanism will help us to elucidate the fundamental control principle of animals' agile locomotion and to realize robots with legs that are truly adaptive and could not be developed solely by conventional control theories. This study aims at providing a "minimal" model of the interlimb coordination mechanism underlying hexapedal locomotion, in the hope that a single control principle could satisfactorily reproduce various aspects of insect locomotion. To this end, we introduce a novel concept we named "Tegotae," a Japanese concept describing the extent to which a perceived reaction matches an expectation. By using the Tegotae-based approach, we show that a surprisingly systematic design of local sensory feedback mechanisms essential for the interlimb coordination can be realized. We also use a hexapod robot we developed to show that our mathematical model of the interlimb coordination mechanism satisfactorily reproduces various insects' gait patterns.

Keywords: hexapedal locomotion, interlimb coordination, local sensory feedback, central pattern generator (CPG), Tegotae

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark Odense, Denmark

Reviewed by:
Jean-Baptiste Mouret, Institut National de Recherche en Informatique et en Automatique, France
Sakyasingha Dasgupta, IBM Research–Tokyo, Japan

***Correspondence:**
Dai Owaki
owaki@riec.tohoku.ac.jp

Received: *11 November 2016*
Accepted: *22 May 2017*
Published: *09 June 2017*

Citation:
Owaki D, Goda M, Miyazawa S and Ishiguro A (2017) A Minimal Model Describing Hexapedal Interlimb Coordination: The Tegotae-Based Approach. Front. Neurorobot. 11:29. doi: 10.3389/fnbot.2017.00029

1. INTRODUCTION

Insects exhibit tremendously versatile gait patterns owing to their locomotion speed and physical/environmental conditions (Hughes, 1957; Graham, 1972, 1977; Cruse, 1976; Foth and Graham, 1983a,b; Dean, 1991; Zollikofer, 1994a,b,c; Noah et al., 2004; Goldman et al., 2006; Sponberg and Full, 2008; Grabowska et al., 2012; Wosnitza et al., 2013). These locomotor patterns are generated via their interlimb coordination mechanism. Biological findings suggest that interlimb coordination in hexapedal locomotion is controlled largely in a decentralized manner by neural networks located in thoracic ganglia (Pearson and Iles, 1969, 1973; Bässler and Wegner, 1983; Dean, 1989; Brekowitz and Laurent, 1996). Thus, clarifying this interlimb coordination mechanism is expected to allow us to obtain the key to understanding the control principle underlying animals' agile locomotion and for realizing truly adaptive legged robots that could not be realized solely by conventional control methods.

Aiming to elucidate the mechanism responsible for the interlimb coordination in hexapedal locomotion, various studies have been conducted to date by focusing on specific insects, e.g.,

stick insects (Graham, 1972, 1977; Cruse, 1976; Foth and Graham, 1983a,b; Dean, 1991; Grabowska et al., 2012) and cockroaches (Hughes, 1957; Pearson and Iles, 1969; Noah et al., 2004; Goldman et al., 2006; Sponberg and Full, 2008) and/or by focusing on control paradigms, e.g., *central pattern generators* (CPGs) (Pearson and Iles, 1973; Bässler and Wegner, 1983; Bässler, 1986, 1993; Ryckebusch and Laurent, 1993; Büschges et al., 1995, 2004; Bässler and Büschges, 1998; Büschges, 2005; Borgmann et al., 2009; Daun-Gruhn and Büschges, 2011; Marder and Bucher, 2011) and *chains of reflexes* (Cruse, 1983, 1990; Cruse et al., 1998; Dürr et al., 2004; Schilling et al., 2013). The knowledge obtained from these past studies deepened biological understanding of the interlimb coordination mechanism greatly; however, the diversity of these approaches may have confused roboticists who want to build adaptive insect-like hexapod robots via bio-inspired approaches (Kimura et al., 1993; Beer et al., 1997; Altendorfer et al., 2001; Ritzmann et al., 2004; Ambe et al., 2013; Manoonpong et al., 2013).

In order to address this problem, in this study, we attempt to capture the control principle essential to understanding the interlimb coordination in a concise form that could help bridge the gap between biologists and roboticists, in the hope that a single control principle could adequately reproduce various aspects of insect locomotion. Since reduction is required for understanding the essence, we build a "minimal model" of the interlimb coordination mechanism on the basis of a mathematically tractable highly abstract model. To this end, we employ a unique approach in this study. We introduce a novel concept we named "Tegotae," a Japanese concept describing the extent to which a perceived reaction matches an expectation. We then introduce a Tegotae function, which is a function that quantitatively measures Tegotae, whereby we can design a decentralized interlimb coordination mechanism in a systematic manner. We validated the Tegotae-based interlimb coordination model by using a physical hexapod robot that we developed. We confirmed that the model adequately reproduced various aspects of insect locomotion patterns. We expect that our minimal model, systematically derived from the concept of Tegotae, will provide substantial insight into the essence of the interlimb coordination mechanism to roboticists as well as biologists.

The following section presents the materials and methods used in this study. First, we describe a basic building block for the interlimb coordination mechanism. Second, we explain the Tegotae concept and the design scheme of local sensory feedback using the Tegotae-based approach. Third, we explain the developed robotic platform in detail. Section 3 presents the experimental results to validate our Tegotae-based control for the interlimb coordination mechanism. Finally, in Section 4, we discuss our results and future work.

2. MATERIALS AND METHODS

2.1. Basic Building Block of Interlimb Coordination Mechanism Employed

To capture the control principle essential for the interlimb coordination mechanism, which works largely in a decentralized manner in insects' thoracic ganglia, it is important to determine a basic building block to be used for the distributed control system. From a control perspective, past studies have intensively argued mainly from the viewpoint of two distinct control paradigms: chains of reflexes (Cruse, 1983, 1990; Cruse et al., 1998; Dürr et al., 2004; Schilling et al., 2013) and CPGs (Pearson and Iles, 1973; Bässler and Wegner, 1983; Bässler, 1986, 1993; Ryckebusch and Laurent, 1993; Büschges et al., 1995, 2004; Bässler and Büschges, 1998; Büschges, 2005; Borgmann et al., 2009; Daun-Gruhn and Büschges, 2011; Marder and Bucher, 2011). In the chain-of-reflex approach, a control system is modeled by using many chained discontinuous reflexive events, in which locomotion can be generated purely from the interaction between sensory feedback signals and the body. However, the discontinuity in this approach may impede mathematical tractability (Daun-Gruhn and Büschges, 2011). In contrast, in the CPG approach, a control system is modeled by using directly coupled oscillators to generate feedforward motor commands, based on a continuous dynamical system, i.e., a set of differential equations, for the interlimb coordination. Considering the mathematical tractability stemming from a continuous model, we employ the CPG approach as a control paradigm. The CPG approach offers various ways to model a basic building block at different levels of abstraction (Ijspeert, 2008), ranging from detailed models using a single cell (Hodgkin and Huxley, 1952; Hellgren et al., 1992) to abstract oscillator models (Fitz-Hugh, 1969; Van der Pol, 1972; Kuramoto, 1984). Here we use a *phase oscillator* (Kuramoto, 1984) for each leg to build a minimal model of the interlimb coordination mechanism on the basis of a highly abstract model.

The time evolution of the oscillator phase is described by a differential equation as follows:

$$\dot{\phi}_i = \omega + f_i, \tag{1}$$

where ω is the intrinsic angular velocity; ϕ_i is the phase of the oscillator implemented into the ith leg; and f_i is a local sensory feedback term, which plays an essential role in the interlimb coordination. This equation is one of the abstract oscillator models, i.e., the *Kuramoto model* (Kuramoto, 1984) (a case without coupling between oscillators and with local sensory feedback f_i), which describes a one-dimensional, reduction model of oscillatory behaviors. Using the trigonometric functions ($\sin\phi_i, \cos\phi_i$, etc.) of oscillator phases enables us to generate a periodic motor command to control the legs of a robot. As an example of implementation, we describe the target angles $\tilde{\theta}_{yaw,i}$ and $\tilde{\theta}_{roll,i}$ for the proportional and derivative (PD) control of the motors (as explained in Section 2.4 and **Figure 6** in detail) through the following equations:

$$\tilde{\theta}_{yaw,i} = -A\cos\phi_i, \tag{2}$$

$$\tilde{\theta}_{roll,i} = \begin{cases} B\sin\phi_i, \text{when } 0 \le \phi_i < \pi, \\ B'\sin\phi_i, \text{when } \pi \le \phi_i < 2\pi, \end{cases} \tag{3}$$

where A, B, and B' are user-defined parameters, describing amplitudes in the yaw and roll direction for leg motion (see

Section 2.4 and **Table 1**). Thus, the ith leg is actively controlled according to ϕ_i such that the ith leg is in the swing phase when $0 \leq \phi_i < \pi$, i.e., $\sin\phi_i > 0$, and in the stance phase when $\pi \leq \phi_i < 2\pi$, i.e., $\sin\phi_i < 0$, as shown in **Figure 1**. Below, we explain how we design local sensory feedback f_i by introducing the concept of "Tegotae" in a systematic manner.

2.2. Tegotae and Tegotae Function

Here we explain the core concept Tegotae in detail. Tegotae is a novel concept describing the extent to which a perceived reaction matches an expectation (intention) of a controller. For ease of understanding, let us explain it metaphorically. Imagine you want to lean against a wall nearby. Note that what you want to do, i.e., leaning against the wall, is regarded as the intention of the controller, i.e., your nervous system. When you lean against the wall, if you feel that the reaction force from the wall is sufficient for supporting your body, we say "good" Tegotae is obtained. If the reaction force you receive is insufficient (imagine the wall were a curtain/screen for example), "bad" Tegotae is obtained. Notice that Tegotae stems not only from the reaction received from the environment, but also from the consistency between the perceived reaction and the intention/expectation of the controller, i.e., what the controller wants to do.

Now the question is how to quantify Tegotae. Of course, there are various ways to accomplish this. As the initial step of the investigation, we quantify Tegotae in the simplest mathematical form, i.e., a function based on the type of separation of variables as follows:

$$T_i(\phi_i, N) = C(\phi_i)S(N). \tag{4}$$

Hereafter, we refer to the function T_i as the "Tegotae function"—a function that quantitatively measures Tegotae. ϕ_i is a control variable (in this case the phase of the oscillator), and N is the sensory information obtained from multiple sensors embedded in the body. Note that, the Tegotae function T_i is expressed as the product of two functions $C(\phi_i)$ and $S(N)$: the former is a function expressing the intention of the controller, and the latter denotes the reaction obtained from the environment. Here, we design T_i such that it becomes more positive when enhanced Tegotae is detected. Next, we explain how we can design the sensory feedback term f_i by using T_i.

2.3. Tegotae-Based Control

Given that the Tegotae function is defined, the local sensory feedback term f_i is designed in such a way that the control system modulates ϕ_i in order to increase the amount of Tegotae received. Thus, because a continuous system is used, f_i is expressed simply as the partial derivative of the Tegotae function T_i with respect to the control variable ϕ_i, as follows:

$$f_i = \frac{\partial T_i(\phi_i, N)}{\partial \phi_i}. \tag{5}$$

Note that we can systematically design decentralized controllers by only designing the Tegotae functions required.

Now, the question is how to define $T_i(\phi_i, N)$ to satisfactorily reproduce the hexapedal interlimb coordination observed in insect locomotion. In this study, we define $T_i(\phi_i, N)$ as follows:

$$T_i(\phi_i, N) = \sigma_1 T_{i,1}(\phi_i, N) + \sigma_2 T_{i,2}(\phi_i, N), \tag{6}$$

$$T_{i,1}(\phi_i, N) = (-\sin\phi_i)N_i^V, \tag{7}$$

$$T_{i,2}(\phi_i, N) = \sin\phi_i \left(\frac{1}{n_L} \sum_{j \in L(i)}^{n_L} k_j N_j^V \right). \tag{8}$$

As Equation (6) indicates, $T_i(\phi_i, N)$ consists of two Tegotae functions, $T_{i,1}(\phi_i, N)$ and $T_{i,2}(\phi_i, N)$, both of which are linearly coupled via the positive constants σ_1 and σ_2. The suffix i denotes the leg number ($i : 1, 2, \ldots, 6$). Sensory information N consists of vertical ground reaction forces (GRFs) acting on each leg $N = [N_1^V, N_2^V, \ldots, N_6^V]^T$. $L(i)$ denotes a set consisting of the legs neighboring the ith leg, and n_L is the number of elements in $L(i)$ and k_j ($k_a, k_p, k_c \geq 0$) denotes the weight for each GRF N_j^V, as shown in **Figure 2**. Further, we present a detailed explanation of the approach we followed when designing these two Tegotae functions.

$T_{i,1}$ quantifies Tegotae on the basis of the information that is only locally available at the corresponding leg; when the local controller intends to be in the stance leg ($-\sin\phi_i > 0$), and

TABLE 1 | Parameters for each experiment.

Common parameters	Gain for T_1	σ_1	0.2
	Weight for anterior N_j^V	k_a	0.7
	Weight for posterior N_j^V	k_p	0.1
	Weight for contralateral N_j^V	k_c	0.2
	Leg amplitude of anterior-posterior swing motion	A	15°
	Leg amplitude of up-down swing motion	B	20°
	Leg amplitude of stance motion	B'	5°
Section 3.1	Intrinsic angular velocity	ω	2.0
	Gain for T_2	σ_2	1.2
Section 3.2	Intrinsic angular velocity	ω	2.0 → 4.0 (40.0–42.0 s)
	Gain for T_2	σ_2	1.2
Section 3.3	Intrinsic angular velocity	ω	2.0 → 4.0 (40.0–42.0 s)
	Gain for T_2	σ_2	1.2
	Load		500 g
Section 3.4	Intrinsic angular velocity	ω	2.0
	Gain for T_2	σ_2	1.2
Section 3.5	Intrinsic angular velocity	ω	2.0
	Gain for T_2	σ_2	0.0

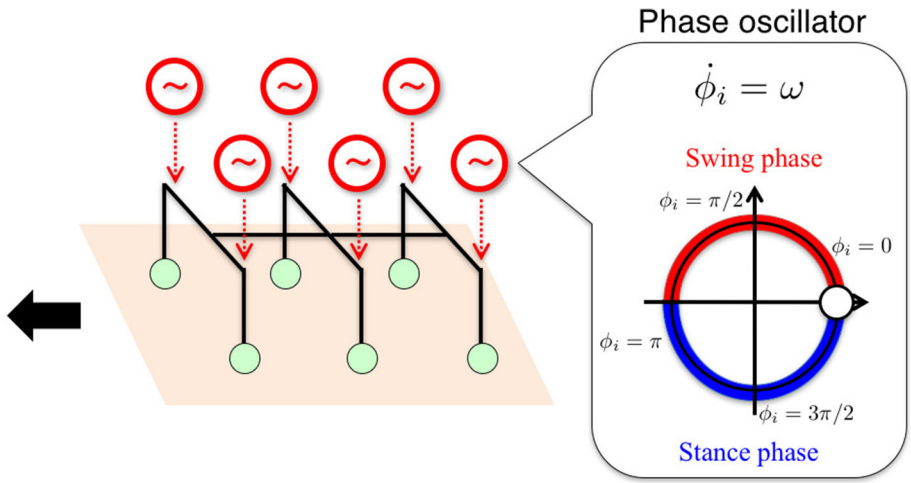


FIGURE 1 | Schematic of the basic building block for the control system. We used a *phase oscillator* (Kuramoto, 1984) with local sensory feedback for each leg for hexapedal interlimb coordination. The *i*th leg is actively controlled (see **Figure 5**) according to ϕ_i such that the *i*th leg is in the swing phase when $0 \leq \phi_i < \pi$ and in the stance phase when $\pi \leq \phi_i < 2\pi$.

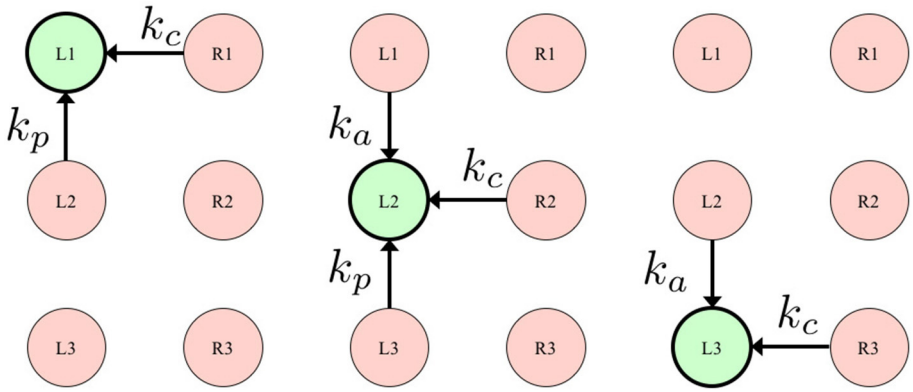


FIGURE 2 | Definition of *L*(*i*), describing a set consisting of the legs neighboring the *i*th leg. The left, center, and right figures show the set of left fore (L1), left middle (L2), and left hind (L3) legs, respectively. k_j ($k_a, k_p, k_c \geq 0$) denotes the weight for each GRF N_j.

results in receiving a ground reaction force ($N_i^V > 0$) (**Figure 3**, top), $T_{i,1}$ evaluates this situation as "good" Tegotae, and returns a positive value.

On the other hand, $T_{i,2}$ quantifies Tegotae on the basis of the relationship between the movements of the corresponding leg and its neighboring legs; when the local controller intends to be in the swing phase ($\sin\phi_i > 0$) and its neighboring legs offer good support to the body at that time ($\frac{1}{n_L}\sum_{j\in L(i)}^{n_L} k_j N_j^V > 0$) (**Figure 3**, bottom), $T_{i,2}$ evaluates that the corresponding leg adequately establishes a relationship with its neighboring legs and returns a positive value.

By substituting Equations (6–8) into Equations (1) and (5), we obtain our interlimb coordination mechanism as follows:

$$\dot{\phi}_i = \omega - \sigma_1 N_i^V \cos\phi_i + \sigma_2 \left(\frac{1}{n_L} \sum_{j\in L(i)}^{n_L} k_j N_j^V \right) \cos\phi_i. \quad (9)$$

Introduction of the Tegotae-based approach enables us to easily design a minimal model for hexapedal interlimb coordination in a systematic manner.

2.4. Robotic Platform for the Validation of Proposed Control Scheme

Figure 4 shows the structure of our hexapod robot. The robot consists of six leg segments (**Figure 5**) and a body segment. The robot is 0.40 m long, 0.30 m wide, 0.20 m high, and weighs 2.4 kg. The leg and body consist of carbon fiber rods and acrylonitrile butadiene styrene (ABS) resin printed using a 3-D printer. For each leg, we used two servo motors (Futaba Corporation, Japan: RS405CB), which generate leg motion during the swing and stance phases according to the corresponding oscillator phase (**Figure 5B**). As shown in **Figure 6**, we describe the target angles $\tilde{\theta}_{yaw,i}$ and $\tilde{\theta}_{roll,i}$ for proportional and derivative (PD) control of

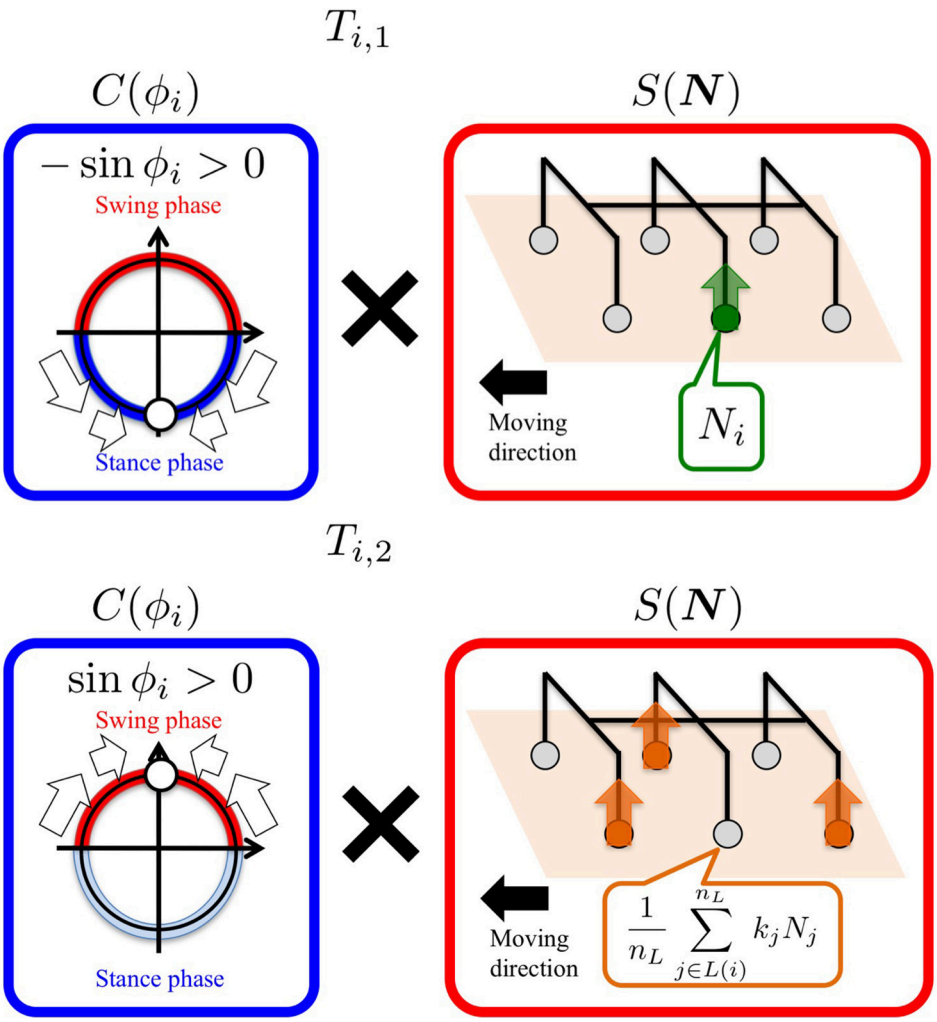


FIGURE 3 | Definition of the "Tegotae" functions, which are expressed as the product of $C(\phi_i)$ and $S(N)$. We design T_i such that it becomes more positive when enhanced Tegotae is detected. The upper and lower figures show the $T_{i,1}$ and $T_{i,2}$ functions for the left middle leg (L2), respectively.

the motors through the following equations:

$$\tilde{\theta}_{yaw,i} = -A\cos\phi_i, \tag{10}$$

$$\tilde{\theta}_{roll,i} = \begin{cases} B\sin\phi_i, \text{when } 0 \le \phi_i < \pi, \\ B'\sin\phi_i, \text{when } \pi \le \phi_i < 2\pi. \end{cases} \tag{11}$$

Based on this control scheme, we can generate periodic leg motion as shown in **Figure 5B**. From the viewpoint of neurophysiological findings for a locomotor CPG system in animals (Lafreniere-Roula and McCrea, 2005; Rybak et al., 2016), Equation (9) corresponds to the rhythm generator (RG) and Equations (10) and (11) correspond to a pattern formation (PF) network in the two-level CPG concept. For the robot, we choose parameter values A, B, B' for the geometric path of the foot by tuning them through trial and error as shown in **Table 1**. We employ passive springs (MISUMI Corporation: WM8-20, 2.9 N/mm) in each leg for shock absorption. Furthermore, we use three-axis force sensors (OptoForce Ltd., Hungary: OMD-20-SE-40N) in the feet of the robot to detect ground reaction forces (GRFs), as shown in **Figure 5A**.

The body contains a main control board. We calculate the oscillator phase in each leg by using microcontrollers (mbed NXP LPC1768) on the main control board. We manipulated each servo motor installed in the legs using proportional-derivative (PD) control as explained above.

3. EXPERIMENTAL RESULTS

To verify the proposed control scheme in the real world, we conducted five experiments: (i) steady walking, (ii) gait transition according to locomotion speed, (iii) adaptability to change in weight distribution, (iv) adaptability to leg amputation, and (v) effect of local sensory feedback. The control parameters that were used in experiments with the hexapod robot (Sections 3.1–3.5) are listed in **Table 1**. We conducted over 10 trials for each

experiment: each trial was conducted on a treadmill for a period of 50 s using randomly selected initial phases.

3.1. Steady Walking

Figure 7 shows the results of measurements conducted when our robot was engaged in steady walking. Here, we set the parameter $\omega = 2.0$ rad/s. **Figure 7** shows the gait diagram (upper graph) and time evolution of the oscillator phases of the legs (lower graph, $\sin\phi_i$) for the period 0.0–20.0 s. In the gait diagram, the colored regions represent the stance phase, which is distinguished by using the threshold data value (1.5 N: less than 10% of the maximum force detected) from the force sensor.

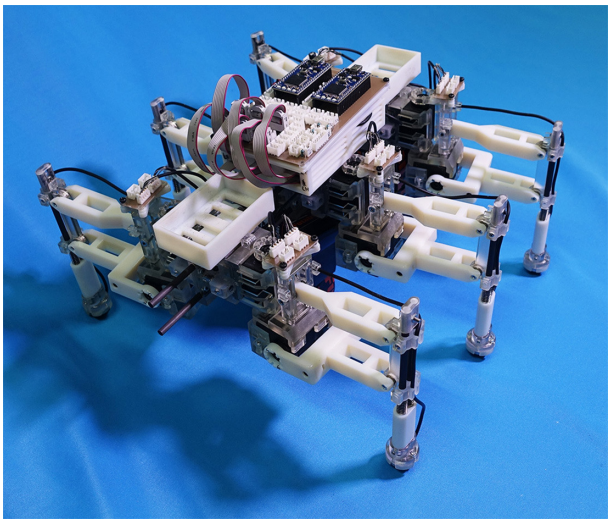


FIGURE 4 | Hexapod robot developed for the study. The robot is 0.40 m long, 0.30 m wide, 0.20 m high, and weighs 2.4 kg.

Hereafter, we use the gait diagrams and movies (i.e., Movies S1–S3) recorded by a video camera as a qualitative evaluation index and the average duty factors (the ratio of the stance phase to one period) as a quantitative evaluation index. For the quantitative analysis, the duty factors obtained by the gait diagrams reflect the direction of the robot motion (i.e., straightness) because the asymmetric duty factors in the left and right legs indicate turning in the locomotion. Moreover, the duty factors indirectly represent the foot point velocity during the locomotion because the leg trajectory of our robot is determined in response to oscillator phases (**Figure 6**). Thus, the data of the duty factors from the gait diagrams indirectly include physical information about the speed and the direction of the locomotion (see SM for more details). The gait pattern rapidly converges from the initial phase relationship to a tetrapod gait—the ipsilateral feet touch the ground in the order of hind, middle, and fore legs—within approximately two periods. Furthermore, we tested the effect of the variation in the initial oscillator phases on the gait patterns. The results confirmed that the initial patterns converged to the same gait patterns from any initial phase relationship (in 10 out of 10 trials: 100%).

3.2. Gait Transitions According to Locomotion Speed

We tested the ability of the proposed control scheme to change the gait patterns according to the locomotion speed by linearly changing the parameter ω from 2.0 to 4.0 rad/s during the time period 40.0 to 42.0 s. **Figure 8A** shows the gait diagram (upper graph) and the time evolution of oscillator phases of legs (lower graph, $\sin\phi_i$), during the time period 30.0–50.0 s in this experiment. After ω was chenged, the gait pattern spontaneously changed from that of a tetrapod to that of a tripod—the (L1, R2, L3) and (R1, L2, R3) feet alternately touch the ground in the anti-phase (Movie S1). **Figure 8B** shows the profile of vertical and

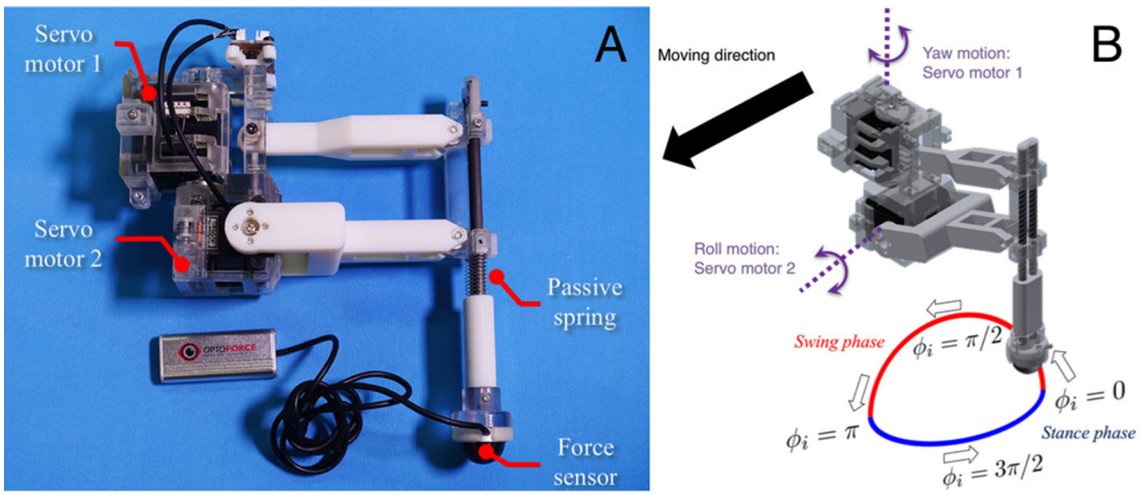
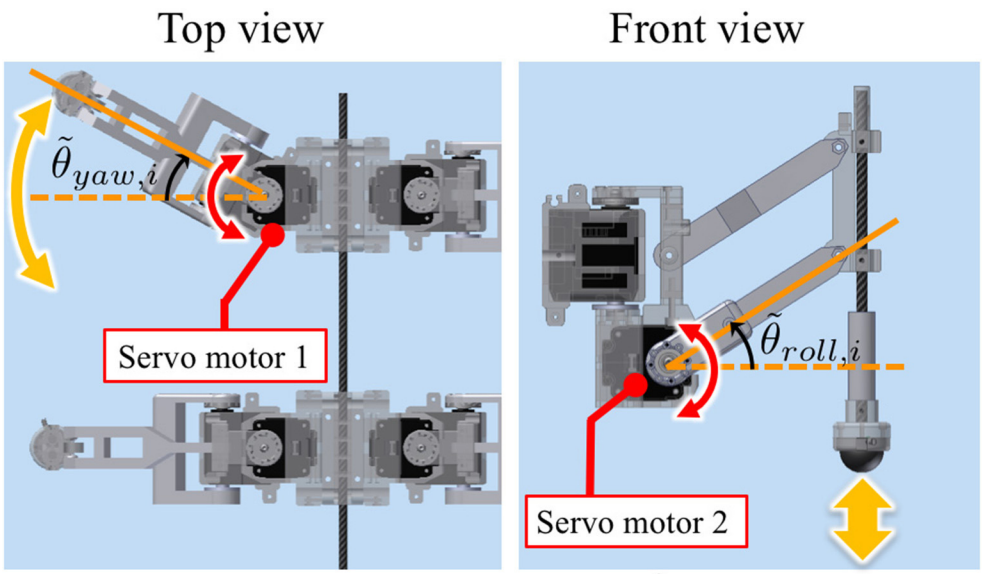


FIGURE 5 | Detailed structure of the leg segment of the robot. **(A)** The leg consists of carbon fiber rods and ABS resin printed using a 3-D printer. The feet contain three-axis force sensors to detect GRFs. **(B)** Each leg is equipped with two servo motors, which generate leg motion during the swing and stance phases according to the corresponding oscillator phase.



$$\tilde{\theta}_{yaw,i} = -A\cos\phi_i \qquad \tilde{\theta}_{roll,i} = \begin{cases} B\sin\phi_i (0 \le \phi_i < \pi) \\ B'\sin\phi_i (\pi \le \phi_i < 2\pi) \end{cases}$$

FIGURE 6 | Leg trajectory for a single leg, where $\tilde{\theta}_{yaw,i}$ and $\tilde{\theta}_{roll,i}$ denote the target angles for proportional and derivative (PD) control of the motors in the yaw and roll directions, respectively. Based on this control scheme, we can generate periodic leg motion as shown in **Figure 5B**.

horizontal GRFs (N_i^V and N_i^H) in the same experiment. In this figure, the upper, middle, and lower graphs show the GRF profile of the front (L1), middle (L2), and hind (L3) legs, respectively. Furthermore, we confirmed this result for the gait transition in all 10 trials (10/10: 100%). The results indicate that leg coordination is appropriately modified according to the locomotion speed via Tegotae-based control.

3.3. Adaptability to Change in Weight Distribution

Here, we show the adaptability of our robot to changes in weight distribution by applying a load (500 g) to the hind portion of the body (upper photograph in **Figure 9**). The lower graphs in **Figure 9** show the experimental result. Here, we changed the parameter ω from 2.0 to 4.0 rad/s during the period 40.0 to 42.0 s as in the previous gait transition experiments (Section 3.2). After changing ω, the gait pattern did not change to that of a tripod; instead, a tetrapod gait was maintained (Movie S2). We obtained the same results in 10 out of 10 trials (100%). **Figure 10** compares the average duty factor of the front, middle, and hind legs without and with the load for 10 trials (ω = 4.0 rad/s). The duty factor, which is the ratio of the stance phase to one period, was calculated by using the gait patterns during six periods for each trial. This result indicates that the duty factor of the loaded hind legs is larger than that of legs that do not bear any load. This result demonstrates the adaptability of our proposed control scheme to changes in the weight distribution without requiring prior data about these changes.

3.4. Adaptability to Leg Amputation

Figure 11 shows the experimental results of the leg amputation test after both of the middle legs were amputated. In spite of the amputation, the robot was able to continue walking. Furthermore, the gait patterns converged to a *trot* or an *L-S walk* gait observed in quadrupeds—*i. e.* the (L1, R3) and (R1, L3) feet alternately touch the ground in nearly anti-phase, or more precisely, focusing on the timing of touch down, the feet touch the ground in the order from L1, R3, R1, L3 (Movie S3). **Figure 12** compares the average duty factor of the front, middle, and hind legs for 10 trials of the leg amputation experiment. The duty factor of each leg was modulated according to the remaining number of legs, which mainly resulted in increasing the duty factor of the hind legs. Furthermore, we confirmed that the initial patterns converged to the same gait patterns from any initial phase relationship (in 10 out of 10 trials: 100%). These results also indicate that the proposed control scheme can achieve interlimb coordination according to the physical properties of the robot's body in a self-organizing manner, without any predefined gait patterns.

3.5. Effect of Local Sensory Feedback Concerning Neighboring Legs

The usefulness of our proposed local sensory feedback was verified based on the Tegotae approach by conducting experiments with the following conditions: we set the parameters $\omega = 2.0$, $\sigma_1 = 0.2$, $\sigma_2 = 0$, which is a model similar to our previous model for quadrupeds (Owaki et al., 2012; Owaki and Ishiguro, 2017) or Barikhan's model for hexapod models

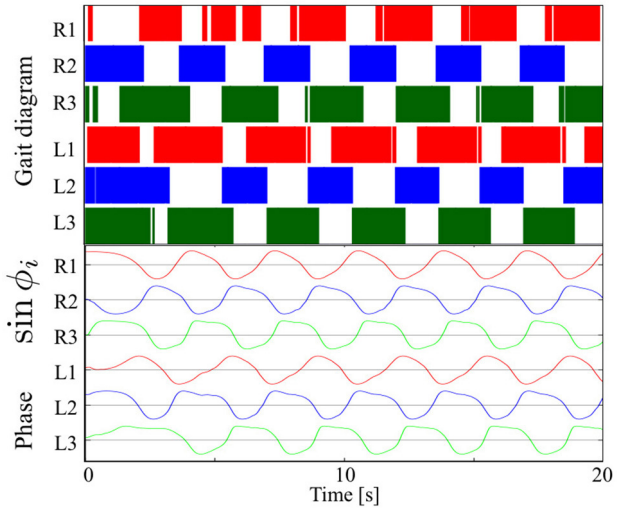


FIGURE 7 | Upper graph: Gait diagram. **Lower graph:** Corresponding phase $\sin\phi_i$. The gait pattern rapidly converges from the initial phase relationship to a tetrapod gait, in which the ipsilateral feet touch the ground in the order hind, middle, and fore legs, within approximately two periods. The results confirmed that the initial patterns converged to the same gait patterns from any initial phase relationship (in 10 out of 10 trials: 100%).

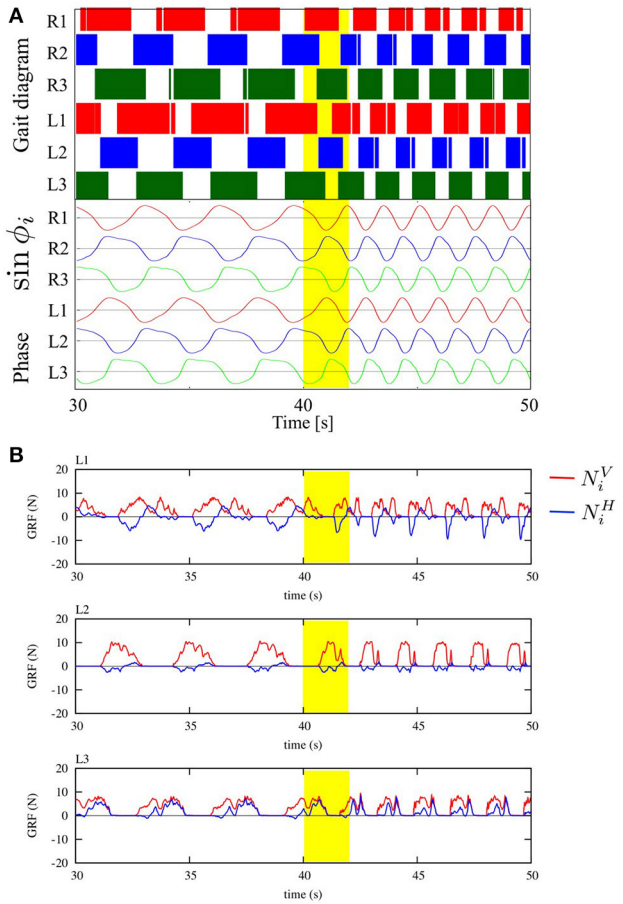


FIGURE 8 | (A) Upper graph: Gait diagram. Lower graph: Corresponding phase $\sin\phi_i$. We found spontaneous transition from the gait of a tetrapod to that of a tripod, in which the (L1, R2, L3) and (R1, L2, R3) feet alternately touch the ground in anti-phase, by changing only parameter ω from 2.0 to 4.0 rad/s in the period from 40.0 to 42.0 s (yellow highlight in the graph, Movie S1). We confirmed the same result for the gait transition in all 10 trials (10/10: 100%). **(B)** The profile of vertical and horizontal GRFs (N_i^V and N_i^H). The upper, middle, and lower graphs show the GRF profile of the front (L1), middle (L2), and hind (L3) legs, respectively.

(Barikhan et al., 2014). We conducted 10 trials in this experiment using randomly selected initial phases. **Figure 13** shows the experimental results obtained using these parameters. The gait patterns mostly did not converge to insect-like gaits, e.g., tetrapod/tripod gaits, but converged to other patterns under many initial conditions (in 7 out of 10 trials: 70%) in this model. In these gaits, the left legs touched the ground in the order L3, L2, and L1 (hind to fore), whereas the right legs touched in the order R1, R2, and R3 (fore to hind). This result indicates that the model with only the second term in Equation (9) (similar to Barikhan's model) sometimes reproduced a gait pattern similar to that of insects, but its robustness against the initial conditions was insufficient.

4. DISCUSSION

The purpose of this study was to provide a minimal model for the interlimb coordination in hexapedal locomotion based on a novel concept named Tegotae. Using the Tegotae-based approach has enabled us to show how we can design the local sensory feedback for a decentralized interlimb coordination mechanism in a systematic manner. Moreover, we have demonstrated that our hexapod robot, which was developed for the validation of the proposed control scheme, satisfactorily reproduced various aspects of insect locomotion, i.e., steady walking, gait transition according to locomotion speed, and adaptability to changes in weight distribution and to leg amputation. As shown in **Figure 8B**, the role arrangement of the fore, middle, and hind legs can be achieved via the interlimb coordination mechanism: (i) the fore legs mainly generate breaking forces (N_i^H was mainly negative), (ii) middle legs mainly support the body (N_i^V was larger than those for the other legs), and (iii) hind legs mainly generate propulsion forces (N_i^H was mainly positive). Such adaptive behaviors are commonly observed for various species of insects, as shown in **Table 2**. This suggests that our Tegotae-based interlimb coordination model captures the essential mechanism for hexapedal interlimb coordination. As a control experiment, if we set the parameters $\sigma_1 = \sigma_2 = 0$, i.e., a condition without local sensory feedback, we can easily imagine that interlimb coordination did not occur, but the phase relationship between leg movement maintains the initial condition. Thus, in order to determine the usefulness of the proposed local sensory feedback, we verified the effect of the second and third terms of Equation (9) in Section 3.

In the previous study on quadruped locomotion (Owaki et al., 2012; Owaki and Ishiguro, 2017), we have proposed a simple interlimb coordination rule that well reproduced various

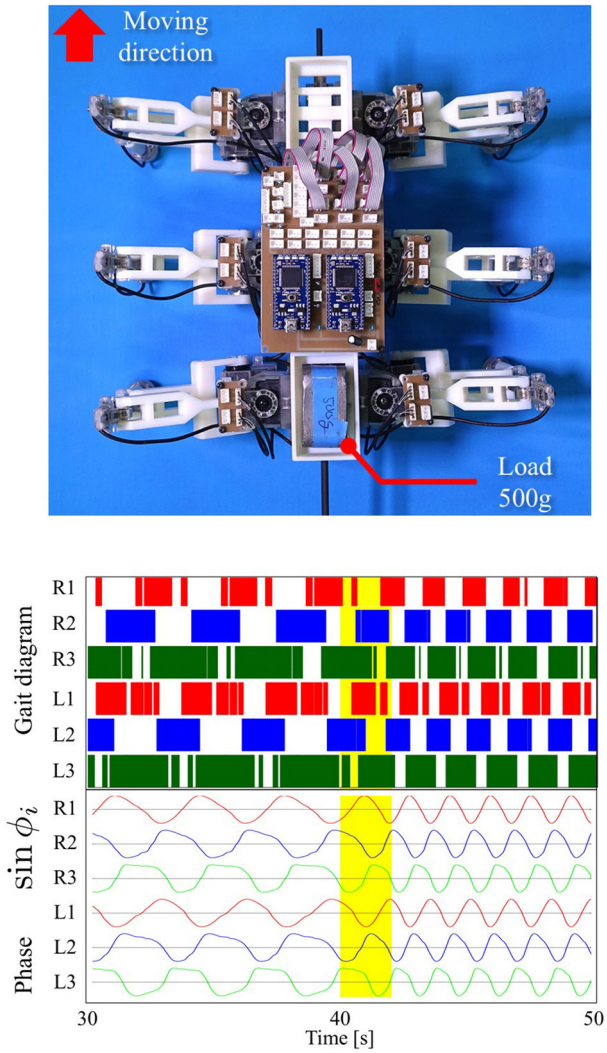


FIGURE 9 | Top: Location of a load (500 g) applied to our robot. **Bottom:** Gait diagram and corresponding phase $\sin\phi_i$. After changing ω, the gait pattern did not change to that of a tripod; instead, a tetrapod gait was maintained (Movie S2). We found the same results in 10 out of 10 trials (100%).

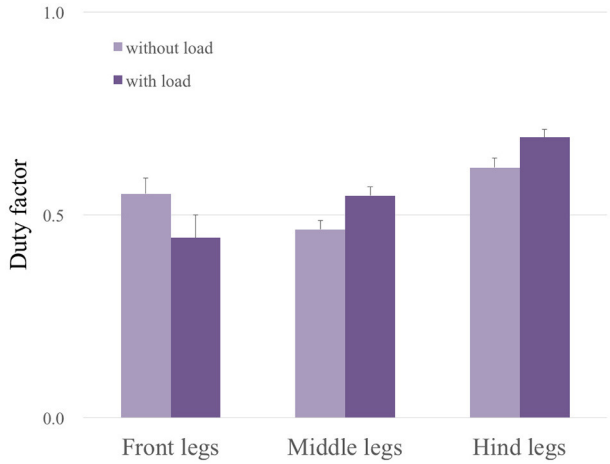


FIGURE 10 | Average duty factor of each leg without and with a load through 10 trials ($\omega = 4.0$ rad/s). This result indicates that the duty factors of the loaded hind legs and middle legs are larger than those of legs without a load, whereas the duty factor of the front legs becomes smaller.

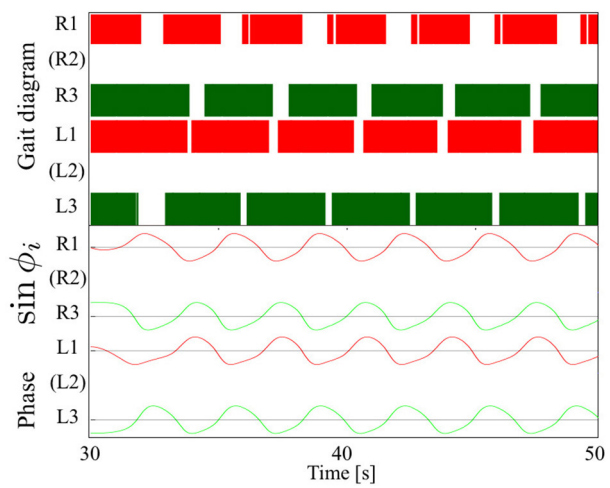


FIGURE 11 | Upper graph: Gait diagram. **Lower graph:** Corresponding phase $\sin\phi_i$. The gait patterns converged to a *trot* or an *L-S walk* gait (in 10 out of 10 trials: 100%) observed in quadrupeds, in which case the (L1, R3) and (R1, L3) feet alternately touch the ground in anti-phase (Movie S3).

quadruped gait patterns and well explained the underlying mechanism. The second term in Equation (9) corresponds to the quadruped interlimb coordination rule. Inspired by our model, Barikhan et al. (2014) also implemented an almost identical mechanism for a hexapedal interlimb coordination model and verified its usefulness by reproducing some insect-like locomotion in simulations. However, although our experiments about the effect of the third term in Equation (9) in Section 3.5 indicate that the model with only the second term in Equation (9) sometimes reproduces a gait pattern similar to that of insects, but its robustness against the initial conditions was insufficient. This is because the local load information on quadrupeds is totally reflected by physical information throughout the whole body (Owaki et al., 2012; Owaki and Ishiguro, 2017), whereas that on hexapods does not sufficiently include physical information for interlimb coordination. Thus, we concluded that the third term in Equation (9), which used sensory information about load distribution in neighboring legs, is essential for the reproduction of insect-like gait patterns and gait transitions. Moreover, we have already reported the local sensory feedback mechanism in Equation (9), but we did not previously confirmed the gait transition from tetrapod to tripod and the adaptability to change in the weight distribution and leg amputation (Goda et al., 2016). Here, we newly introduce anterior-posterior asymmetry in the parameter k_a and k_p, which mainly resulted in the stable gait transition according to locomotion speed, i.e., from tetrapod to tripod as well as the adaptability to change according to

FIGURE 12 | Average duty factor of front, middle, and hind legs in the leg amputation experiment in 10 trials. The duty factor of the hind legs mainly increased in the case of two-leg amputation experiments.

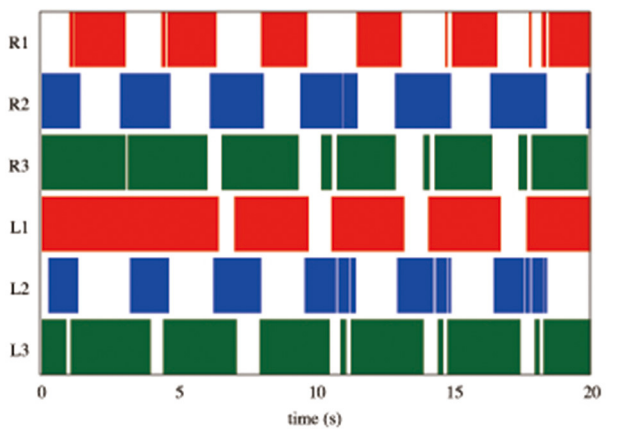


FIGURE 13 | Experimental results: $\omega = 2.0, \sigma_1 = 0.2, \sigma_2 = 0$. The gait patters mostly did not converge to insect-like gaits, e.g., tetrapod/tripod gaits, but converge to other patterns under many initial conditions (in 7 out of 10 trials: 70%). In these gaits, the left legs touched the ground in the order L3, L2, and L1 (hind to fore), whereas the right legs touched in the order R1, R2, and R3 (fore to hind).

the weight distribution and as a results of leg amputation. Our main contribution is the versatility of reproduced behaviors concerning insects' locomotion: Barikhan's model (Barikhan et al., 2014) differs from ours in that it did not reproduce the gait transition from tetrapod to tripod and did not exhibit adaptability against changes in the weight distribution and robustness against initial conditions. Furthermore, our approach is unique; we have discussed the common underlying mechanism of interlimb coordination in the locomotion of both vertebrates and arthropods by using legged robots.

The proposed interlimb coordination model shows adaptability to changes in the weight distribution of the robot's body, where the gait pattern did not change to a tripod gait but maintained a tetrapod gait after changing ω and the average duty factor of the loaded hind legs automatically became larger than those of the unloaded fore legs. These results were reproduced in a self-organizing manner by using Tegotae-based control, without any need to provide prior data about these changes. We additionally obtained biological evidence for the adaptability to changes in the weight distribution by conducting experiments using two crickets (*Gryllus bimaculatus*). These experiments are described in the Supplementary Material in detail. Our results using the robot clearly show good agreement with our biological evidence of the influence of the load on the leg coordination in crickets: with a load, (1) they exhibit a tetrapod gait and (2) increase the duty factor of the middle and hind legs. Furthermore, another experiment using fruit flies confirmed the same effect of a vertical load (Mendes et al., 2014), which suggests that such adaptability is observed for various species of insects. This fact strongly supports that the essentiality of using vertical GRFs N_i^V for sensory information $S(N)$ when designing a Tegotae function for hexapedal interlimb coordination.

Furthermore, our model exhibited adaptability to the physical conditions resulting from a two-leg amputation. If we use a predefined neural connection for a tripod gait—where the (L1, R2, L3) and (R1, L2, R3) legs are in-phase—, we cannot reproduce a trot or an L-S walk pattern—where the (L1, R3) and (R1, L3) feet alternately touch the ground in nearly anti-phase—when the two legs are amputated (**Figure 11**). Owing to the Tegotae-based interlimb coordination mechanism using both local (N_i) and neighboring (N_j) load information (Equation 9), gait patterns were self-organized in response to load distribution stemming from the remaining number of legs, which is one of the advantages of our approach. Some biological studies have suggested that insects generally exhibit the L-S walk when their two middle legs are amputated. Hughes (1957) have shown that two-middle-leg amputee cockroaches exhibited a gait—the touch-down order was (L3, L1, R3, R1), i.e., the L-S walk in quadrupeds. Graham (1977) and Grabowska et al. (2012) have shown that two-middle-leg amputee stick insects exhibited the same gait as cockroaches (Hughes, 1957) because the contralateral touch down timing became same such that gaits could be symmetric about the body axis to ensure its stability. Here, we did not actually conduct various leg-amputation tests; we can expect adaptability to some extent against some conditions, e.g., amputating a front/hind leg, owing to the potential of our model, as we have shown. However, because our model did not include any directional or posture controls and learning algorithms as in Ren et al. (2015) and Cully et al. (2015) (here, we mainly focus on real-time adaptability), its direction of motion would vary according to the physical properties: a front-left-leg amputated robot will turn left when moving forward. According to the patterns of leg amputation, insects exhibit modulation of their spatial footfall patterns, i.e., they change the landing location of a stance leg to maintain their posture stability (Hughes, 1957; Graham, 1977; Cruse, 1983; Grabowska et al., 2012); thus, we intend to apply an additional Tegotae-based controller for the modulation of spatial footfall patterns, resulting in the adaptation to a large number of leg amputations.

TABLE 2 | Observed adaptive behavior various species of insects have in common.

Velocity change	Our robot Tetrapod → tripod	Stick insect (Graham, 1972) Tetrapod → tripod	Fruit fly (Mendes et al., 2013) Tetrapod → tripod
Load on their body	Our robot Tetrapod	Cricket (results in the SM) Tetrapod	Fruit fly (Mendes et al., 2014) Tetrapod
Amputating two middle legs	Our robot Trot/L-S walk	Cockroach (Hughes, 1957) L-S walk	Stick insect (Graham, 1977; Grabowska et al., 2012) Wave/L-S walk

In insect locomotion, it is well known that two types of sensory signals play an essential role in leg coordination: (1) sensory signals about the position and velocity of joints during movement (Büschges, 2005; Pearson et al., 2006) and (2) force signals from the leg segments (Pearson, 1972; Bässler, 1977; Cruse, 1985a,b; Duysens et al., 2000; Zill et al., 2004). Such sensory signals modulate not only the timing (phase) but also the magnitude of neural output stems from the nervous system, e.g., CPGs (Grillner, 2003; Büschges, 2005). In our Tegotae-based approach, as a first step for the investigation, we use only vertical GRFs N_i^V detected by force sensors installed in the legs to modulate the phase of oscillators. The obtained control principle, where both local and neighboring leg load information is essential for the interlimb coordination, agrees with biological evidence (Pearson, 1972; Bässler, 1977; Cruse, 1985a,b; Duysens et al., 2000; Zill et al., 2004). To reproduce increased adaptability to different surfaces and typed of movement, e.g., uneven terrain, uphill/downhill, similar to insects, other types of sensory signals, e.g., horizontal GRFs, would requires us to design additional Tegotae functions. Furthermore, modulation of the magnitude of motor output from neural systems will also lead to a change in landing location of a stance leg for negotiating various leg amputation patterns as discussed in the above paragraph. These topics seem to be of general interest and will also be studied in further investigations.

In the past two decades, various hexapod robots were developed with the aim of reproducing the adaptive functions of insects and to understand their control mechanisms (Kimura et al., 1993; Beer et al., 1997; Altendorfer et al., 2001; Ritzmann et al., 2004; Steingrube et al., 2010; Ambe et al., 2013; Manoonpong et al., 2013; Dasgupta et al., 2015; Ramdya et al., 2017). Ours was the first study of its kind to demonstrate various aspects of insect locomotion with a minimal control principle without any interlimb neural communication between oscillators. To the best of our knowledge, no studies have been reported in which adaptability was reproduced in a completely self-organized manner by only using local and neighboring load information. In the CPG approach as a control paradigm in this study, local sensory feedback f_i is described simply as a partial differential of the Tegotae function T_i with respect to the control variable ϕ_i. This aspect of our model also suggests a new design scheme of local sensory feedback in the chain-of-reflex approach based on the discontinuous basic process, which should also be discussed as a next step. Our minimal model, which is systematically derived from the concept of Tegotae, is expected to provide substantial insight into the essence of the hexapedal interlimb coordination mechanism to roboticists as well as biologists.

AUTHOR CONTRIBUTIONS

AI and DO conceived the research and managed the data collection. MG and SM designed the robot and conducted the experiments. MG, SM, and DO conducted the analyses. All authors wrote the manuscript together.

FUNDING

We acknowledge the support provided by the Japan Science and Technology Agency (CREST).

ACKNOWLEDGMENTS

We are grateful to R. Kobayashi (Hiroshima University), H. Aonuma (Hokkaido University), and T. Kano (Tohoku University) for their helpful comments.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00029/full#supplementary-material

REFERENCES

Altendorfer, R., Moore, N., Komsuoglu, H., Buehler, M., Brown, H. B. Jr., McMordie, D., et al. (2001). Rhex: a biologically inspired hexapod runner. *Auton. Robots* 11, 207–213. doi: 10.1023/A:1012426720699

Ambe, Y., Nachstedt, T., Manoonpong, P., Worgotter, F., Aoi, S., and Matsuno, F. (2013). "Stability analysis of a hexapod robot driven by distributed nonlinear oscillators with a phase modulation mechanism," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 5087–5092.

Barikhan, S. S., Wörgötte, F. F., and Manoonpong, P. (2014). "Multiple decoupled cpgs with local sensory feedback for adaptive locomotion behaviors of bio-inspired walking robots," in *SAB2014* (Castellon), 65–75.

Bässler, U. (1977). Sense organs in the femur of the stick insect and their relevance to the control of position of the

femur-tiibia-joint. *J. Comp. Physiol.* 121, 99–113. doi: 10.1007/BF00614183

Bässler, U. (1986). On the definition of central pattern generator and its sensory control. *Biol. Cybern.* 54, 65–69. doi: 10.1007/BF00337116

Bässler, U. (1993). The walking- (and searching-) pattern generator of stick insects, a modular system composed of reflex chains and endogenous oscillators. *Biol. Cybern.* 69, 305–317. doi: 10.1007/BF00203127

Bässler, U., and Büschges, A. (1998). Pattern generation for stick insect walking movements -multisensory control of a locomotor program. *Brain Res. Rev.* 27, 65–88. doi: 10.1016/S0165-0173(98)00006-X

Bässler, U., and Wegner, U. (1983). Motor output of the denervated thoracic ventral nerve cord in the stick insect carausius morsus. *J. Exp. Biol.* 105, 127–145.

Beer, R. D., Quinn, R. D., Chiel, H. J., and Ritzmann, R. E. (1997). Biologically inspired approaches to robotics: what can we learn from insects? *Commun. ACM* 40, 30–38. doi: 10.1145/245108.245118

Borgmann, A., Hooper, S. L., and Buschges, A. (2009). Sensory feedback induced by front-leg stepping entrains the activity of central pattern generators in caudal segments of the stick insect walking system. *J. Neurosci.* 29, 2972–2983. doi: 10.1523/JNEUROSCI.3155-08.2009

Brekowitz, A., and Laurent, G. (1996). Central generation of grooming motor patterns and interlimb coordination in locusts. *J. Neurosci.* 16, 8079–8091.

Büschges, A. (2005). Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion. *J. Neurophysiol.* 93, 1127–1135. doi: 10.1152/jn.00615.2004

Büschges, A., Ludwar, B., Bucher, D., Schmidt, J., and DiCaprio, R. A. (2004). Synaptic drive contributing to rhythmic activation of motoneurons in the deafferented stick insect walking system. *Eur. J. Neurosci.* 19, 1856–1862. doi: 10.1111/j.1460-9568.2004.03312.x

Büschges, A., Schmitz, J., and Bässler, U. (1995). Rhythmic pattern in the thoracic nerve cord of the stick insect induced by pilocarpine. *J. Exp. Biol.* 198, 435–456.

Cruse, H. (1976). The function of the legs in the free walking stick insect, carausis morosus. *J. Comp. Physiol.* 112, 235–262. doi: 10.1007/BF00606541

Cruse, H. (1983). The influence of load and leg amputation upon coordination in walking crustaceans: a model calculation. *Biol. Cybern.* 49, 119–125. doi: 10.1007/BF00320392

Cruse, H. (1985a). Coactivation influences between neighboring legs in walking insects. *J. Exp. Biol.* 114, 513–519.

Cruse, H. (1985b). Which parameters control the leg movement of a walking insect? II. the start of the swing phase. *J. Exp. Biol.* 116, 357–362.

Cruse, H. (1990). What mechanisms coordinate leg movement in walking arthropods? *Trends Neurosci.* 13, 15–21. doi: 10.1016/0166-2236(90)90057-H

Cruse, H., Kindermann, T., Schumm, M., Dean, J., and Schmitz, J. (1998). Walknet - a bioinspired network to control six-legged walking. *Neural Netw.* 11, 1435–1447. doi: 10.1016/S0893-6080(98)00067-7

Cully, A., Clune, J., Tarapore, D., and Mouret, J. (2015). Robots that can adapt like animals. *Nature* 521, 503–507. doi: 10.1038/nature14422

Dasgupta, S., Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2015). Distributed recurrent neural forward models with synaptic adaptation and cpg-based control for complex behaviors of walking robots. *Front. Neurorobot.* 9:10. doi: 10.3389/fnbot.2015.00010

Daun-Gruhn, S., and Büschges, A. (2011). From neuron to behavior: dynamic equation-based prediction of biological processes in motor control. *Biol. Cybern.* 105, 71–88. doi: 10.1007/s00422-011-0446-6

Dean, J. (1989). Leg coordination in the stick insect carausis morosus: effect of cutting thoracic connectives. *J. Exp. Biol.* 145, 103–131.

Dean, J. (1991). Effect of load on leg movement and step coordination of the stick insect carausius morosus. *J. Exp. Biol.* 159, 449–471.

Dürr, V., Schmitz, J., and Cruse, H. (2004). Behavior-based modeling of hexapod locomotion: linking biology and technical application. *Arthropod Struct. Dev.* 33, 237–250. doi: 10.1016/j.asd.2004.05.004

Duysens, J., Clarac, F., and Cruse, H. (2000). Load-regulating mechanisms in gait and posture: comparative aspects. *Physiol. Rev.* 80, 83–133.

Fitz-Hugh, R. (1969). *Mathematical Models of Excitation and Propagation in Nerve*. New York, NY: McGraw-Hill Book Co.

Foth, E., and Graham, D. (1983a). Influence of loading parallel to the body axis on the walking coordination of an insect I. Inspilateral effects. *Biol. Cybern.* 47, 17–23.

Foth, E., and Graham, D. (1983b). Influence of loading parallel to the body axis on the walking coordination of an insect II. Contralateral changes. *Biol. Cybern.* 48, 149–157. doi: 10.1007/BF00318082

Goda, M., Miyazawa, S., Itayama, S., Owaki, D., Kano, T., and Ishiguro, A. (2016). "Understanding interlimb coordination mechanism of hexapod locomotion via "tegotae"-based control," in *Living Machines 2016* (Edinburgh), 441–448.

Goldman, D. I., Chen, T. S., Dudek, D. M., and Full, R. J. (2006). Dynamics of rapid vertical climbing in cockroaches reveals a template. *J. Exp. Biol.* 209, 2990–3000. doi: 10.1242/jeb.02322

Grabowska, M., Godlewska, E., Schmidt, J., and Daun-Gruhn, S. (2012). Quadrupedal gaits in hexapod animals - inter-leg coordination in free-walking adult stick insects. *J. Exp. Biol.* 215, 4255–4266. doi: 10.1242/jeb.073643

Graham, D. (1972). A behavioural analysis of the temporal organization of walking movements in the 1st instar and adult stick insect (*carausius morosus*). *J. Comp. Pysiol.* 81, 23–52. doi: 10.1007/BF00693548

Graham, D. (1977). The effect of amputation and leg restraint on the free walking coordination of the stick insect carausisus morosus. *J. Comp. Physiol.* 116, 91–116. doi: 10.1007/BF00605519

Grillner, S. (2003). The motor infrastructure: from ion channels to neuronal networks. *Nat. Rev. Neurosci.* 4, 573–586. doi: 10.1038/nrn1137

Hellgren, J., Grillner, S., and Lansner, A. (1992). Computer simulation of the segmental neural network generating locomotion in lamprey by using populations of network interneurons. *Biol. Cybern.* 68, 1–13. doi: 10.1007/BF00203132

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Hughes, G. M. (1957). The co-ordination of insect movements II. The effect of limb amputation and the cutting of commissures in the cockroach (*blatta orientalis*). *J. Exp. Biol.* 34, 306–333.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Kimura, S., Yano, M., and Shimizu, H. (1993). A self-organizing model of walking patterns of insects. *Biol. Cybern.* 69, 183–193. doi: 10.1007/BF00198958

Kuramoto, Y. (1984). *Chemical Oscillations, Waves, and Turbulence*. Berlin; Heidelberg; New York, NY: Springer-Verlag.

Lafreniere-Roula, F. M., and McCrea, D. A. (2005). Deletions of rhythmic motoneuron activity during fictive locomotion and scratch provide clues to the organization of the mammalian central pattern generator. *J. Neurophysiol.* 94, 1120–1132. doi: 10.1152/jn.00216.2005

Manoonpong, P., Parlitz, U., and Wörgötter, F. (2013). Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines. *Front. Neural Circuits* 7:12. doi: 10.3389/fncir.2013.00012

Marder, E., and Bucher, D. (2011). Central pattern generators and the control of rhythmic movements. *Curr. Biol.* 11, 986–996. doi: 10.3389/fncir.2013.00012

Mendes, C. S., Bartos, I., Akay, T., Márka, S., and Mann, R. S. (2013). Quantification of gait parameters in freely walking wild type and sensory deprived *Drosophila melanogaster*. *eLife*. 2:e00231. doi: 10.7554/eLife.00231

Mendes, C. S., Rajendren, S. V., Bartos, I., Márka, S., and Mann, R. S. (2014). Kinematic responses to changes in walking orientation and gravitational load in *Drosophila melanogaster*. *PLoS ONE* 9:e109204. doi: 10.1371/journal.pone.0109204

Noah, J. A., Quimby, L., Frazier, S. F., and Zill, S. N. (2004). Walking on a 'peg leg': extensor muscle activities and sensory feedback after distal leg denervation in cockroaches. *J. Comp. Physiol.* 190, 217–231. doi: 10.1007/s00359-003-0488-x

Owaki, D., and Ishiguro, A. (2017). A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping. *Sci. Rep.* 7:277. doi: 10.1038/s41598-017-00348-9

Owaki, D., Kano, T., Nagasawa, T., Tero, A., and Ishiguro, A. (2012). Simple robot suggests physical interlimb communication is essential for quadruped walking. *J. R. Soc. Interface*. 10:20120669. doi: 10.1098/rsif.2012.0669

Pearson, K., Ekeberg, O., and Buschges, A. (2006). Assessing sensory function in locomotor systems using neuro-mechanical simulations. *Trends Neurosci.* 29, 625–631. doi: 10.1016/j.tins.2006.08.007

Pearson, K. G. (1972). Central programming and reflex control of walking in the cockroach. *J. Exp. Biol.* 56, 173–193.

Pearson, K. G., and Iles, I. F. (1969). Discharge patterns of coxal levator and depressor motoneurons of the cockroach, *Perillaneta americana*. *J. Exp. Biol.* 52, 139–165.

Pearson, K. G., and Iles, J. F. (1973). Nervous mechanisms underlying intersegmental co-ordination of leg movements during walking in the cockroach. *J. Exp. Biol.* 58, 725–744.

Ramdya, P., Thandiackal, R., Cherney, R., Asselborn, T., Benton, R., Ijspeert, A. J., et al. (2017). Climbing favours the tripod gait over alternative faster insect gaits. *Nat Commun.* 8:14494. doi: 10.1038/ncomms14494

Ren, G., Chen, W., Dasgupta, S., Kolodziejski, C. K., Wörgötter, F., and Manoonpong, P. (2015). Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation. *Inform. Sci.* 294, 666–682. doi: 10.1016/j.ins.2014.05.001

Ritzmann, R. E., Quinn, R. D., and Fischer, M. S. (2004). Convergent evolution and locomotion through complex terrain by insects, vertebrates and robots. *Arthropod. Struct. Dev.* 33, 361–379. doi: 10.1016/j.asd.2004.05.001

Rybak, I. A., Shevtsova, N. A., Lafreniere-Roula, M., and McCrea, D. A. (2016). Modeling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *J. Physiol.* 557, 617–639. doi: 10.1113/jphysiol.2006.118703

Ryckebusch, S., and Laurent, G. (1993). Rhythmic patterns evoked in locust leg motor neurons by the muscarinic agonist pilocarpine. *J. Neurophysiol.* 69, 1583–1595.

Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H. (2013). Walknet, a bio-inspired controller for hexapod walking. *Biol. Cybern.* 107, 397–419. doi: 10.1007/s00422-013-0563-5

Sponberg, S., and Full, R. J. (2008). Neuromechanical response of musculo-skeletal structures in cockroaches during rapid running on rough terrain. *J. Exp. Biol.* 211, 433–446. doi: 10.1242/jeb.012385

Steingrube, S., Timme, M., Wörgötter, F., and Manoonpong, P. (2010). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nat. Phys.* 6, 224–230. doi: 10.1038/nphys1508

Van der Pol, B. (1972). On relaxation oscillations. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* 2, 978–992. doi: 10.1080/14786442608564127

Wosnitza, A., Bockemuhl, T., Dubbert, M., Scholz, H., and Buschges, A. (2013). Inter-leg coordination in the control of walking speed in drosophila. *J. Exp. Biol.* 216(Pt 3), 480–491. doi: 10.1242/jeb.078139

Zill, S., Schmitz, J., and Buschges, A. (2004). Load sensing and control of posture and locomotion. *Arthropod. Struct. Dev.* 33, 273–286. doi: 10.1016/j.asd.2004.05.005

Zollikofer, C. P. E. (1994a). Stepping pattern in ants I. Influence of speed and curvature. *J. Exp. Biol.* 192, 95–106.

Zollikofer, C. P. E. (1994b). Stepping pattern in ants II. Influence of body morphology. *J. Exp. Biol.* 192, 107–108.

Zollikofer, C. P. E. (1994c). Stepping pattern in ants III. Influence of load. *J. Exp. Biol.* 192, 119–127.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Owaki, Goda, Miyazawa and Ishiguro. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 16 March 2017
doi: 10.3389/fnbot.2017.00008

Self-Organized Behavior Generation for Musculoskeletal Robots

Ralf Der[1] *and Georg Martius*[2,3]*

[1] Institute for Computer Science, University of Leipzig, Leipzig, Germany, [2] IST Austria, Klosterneuburg, Austria, [3] Autonomous Learning Group, Max Planck Institute for Intelligent Systems, Tübingen, Germany

With the accelerated development of robot technologies, control becomes one of the central themes of research. In traditional approaches, the controller, by its internal functionality, finds appropriate actions on the basis of specific objectives for the task at hand. While very successful in many applications, self-organized control schemes seem to be favored in large complex systems with unknown dynamics or which are difficult to model. Reasons are the expected scalability, robustness, and resilience of self-organizing systems. The paper presents a self-learning neurocontroller based on extrinsic differential plasticity introduced recently, applying it to an anthropomorphic musculoskeletal robot arm with attached objects of unknown physical dynamics. The central finding of the paper is the following effect: by the mere feedback through the internal dynamics of the object, the robot is learning to relate each of the objects with a very specific sensorimotor pattern. Specifically, an attached pendulum pilots the arm into a circular motion, a half-filled bottle produces axis oriented shaking behavior, a wheel is getting rotated, and wiping patterns emerge automatically in a table-plus-brush setting. By these object-specific dynamical patterns, the robot may be said to recognize the object's identity, or in other words, it discovers dynamical affordances of objects. Furthermore, when including hand coordinates obtained from a camera, a dedicated hand-eye coordination self-organizes spontaneously. These phenomena are discussed from a specific dynamical system perspective. Central is the dedicated working regime at the border to instability with its potentially infinite reservoir of (limit cycle) attractors "waiting" to be excited. Besides converging toward one of these attractors, variate behavior is also arising from a self-induced attractor morphing driven by the learning rule. We claim that experimental investigations with this anthropomorphic, self-learning robot not only generate interesting and potentially useful behaviors, but may also help to better understand what subjective human muscle feelings are, how they can be rooted in sensorimotor patterns, and how these concepts may feed back on robotics.

Keywords: self-organization, robot control, musculoskeletal, tendon-driven, learning, anthropomimetic, self-exploration

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark Odense, Denmark

Reviewed by:
Subramanian Ramamoorthy, University of Edinburgh, UK
Hazem Toutounji, ZI Mannheim, Germany

***Correspondence:**
Georg Martius
georg.martius@tuebingen.mpg.de

Received: *19 August 2016*
Accepted: *07 February 2017*
Published: *16 March 2017*

Citation:
Der R and Martius G (2017) Self-Organized Behavior Generation for Musculoskeletal Robots. Front. Neurorobot. 11:8. doi: 10.3389/fnbot.2017.00008

1. INTRODUCTION

Control is a ubiquitous theme of life and technology. When reaching for a cup of coffee or walking through the mountains, our nervous system controls all movements with great ease, despite the great uncertainty involved in controlling the muscles, the complexity of the task and many other factors. That this simplicity is an illusion is seen as soon as trying to program a robot for

doing a task. While the complexity of programming stands as a challenge for decades, in recent times considerable progress has been achieved by new materials (Kim et al., 2013), powerful actuators (Raibert et al., 2008), the improved theory of control (Siciliano et al., 2009), but in particular by the tremendous increase in computational power that allows modeling and physically realistic simulations of very complex systems to improve planning and control (Mordatch et al., 2012; Erez et al., 2013; Posa et al., 2014) and even allows to simulate large controlled muscular body systems (Yamane and Nakamura, 2011), or find new perspectives for artificial evolution (Bongard, 2015) by exploiting super computer power. Also there are a variety of new control paradigms around, best demonstrated by the amazing locomotion abilities of the Boston dynamics robots, like BigDog, PETMAN and others. These are ingeniously engineered systems for realizing a specific set of tasks with their highly specialized bodies. The DARPA challenge also presents numerous examples of progress but also reveals a realm of failures of these systems even under remote control. Alternatively, the so-called embodied AI recognizes that the body can be very helpful in reducing both design efforts and computational load on the controller. The exploitation of the specific properties of the body, sometimes called *morphological computation* (Paul, 2004; Pfeifer and Gómez, 2009; Hauser et al., 2012) is an active field of research with many impressive results, see Pfeifer and Bongard (2006) and Pfeifer and Scheier (1999), opening new perspectives for both robot control and our understanding of human sensorimotor intelligence (Pfeifer et al., 2012).

The embodied approach seems to be favored in systems with strong physical effects, like soft robotic systems or elastically actuated robots, where the engineering approaches may run into severe difficulties. Though there are a number of interesting results, for instance in employing neural learning to obtain goal-directed behavior, e.g., Manoonpong et al. (2007), Shim and Husbands (2012), Toutounji and Pasemann (2014), and Tetzlaff et al. (2014) using fast synaptic plasticity as in this work, or using simplified spring-models (Park and Kim, 2015), a systematic embodied approach for controlling such systems is not available so far. This is not a surprise, given the aim of exploiting the physical dynamics which is strongly embodiment specific. In this paper we will not aim at a general solution to physics based deliberate control but will investigate the possible role of self-organization (SO) and its general phenomenology in robotics. We will devote this paper to systems with extended embodiment, consisting of a Myorobotics arm connected to a physical subsystem with an internal dynamics of its own. The arm is a muscle-tendon driven (MTD) mechanical system with strong embodiment effects. The controller is a one-layer feedforward neural network which may drive systems into self-organization by a specific learning rule—differential extrinsic plasticity (DEP)—as introduced recently in Der and Martius (2015). It was applied to a number of systems in simulation producing a great variety of behavior. In a slightly modified form, it will face here a new challenge with MTD systems with their strong embodiment effects.

To introduce this paper's topics and claims, imagine that you get an object, a half-filled bottle for that matter, attached to the tip of your forearm such that you can neither know orientation nor identity of the object. When sitting in the dark you probably will start doing something, trying to find out about the object's properties. The idea is, while moving the bottle around, you feel the reaction from the water when hitting the walls of the bottle. Intrigued by this signal and driven by curiosity, you may vary the direction of the shaking motion to end up with shaking parallel to the bottle axis, as the strongest and most coherent force response is coming from there. Without vision or any other external information on the attached object, motor signals are based on the sensor values, i.e., the muscle tensions, modulated by the force responses of the subsystem's internal dynamics. Humans will describe this as feeling the muscles (or the embodiment in general) and generating actions out of this feeling. Generally, behavior is a direct result of the agent-environment coupling, here the dynamical contact between the agent, the arm with its "brain," and the attached object.

Similarly, with DEP learning, the self-excited motion patterns of the arm are guided, or piloted, by the object's internal dynamics. Specifically, an attached pendulum drives the arm into a circular motion, a half-filled bottle produces axis oriented shaking behavior, a wheel is getting rotated, and wiping patterns emerge automatically in a table-plus-brush setting. This is of interest for the self-organized acquisition of behavioral primitives but there is more: as the emerging patterns are object specific, we may say that the robot was able of identifying the object's identity by just the feedback through the (unknown) internal dynamics of the object. Identifying means that our self-learning system responds with a specific sensorimotor pattern for each object attached to the arm. So, this is a cognitive act closely related to the self-organized discovery of Gibson's object affordances, in particular for dynamical interactions, see below. The observation that DEP learning elicits just these subtle effects unknown so far is the central result of this paper.

Acquired with an anthropomorphic robot (arm), these findings may also provide answers to more general questions in human related cognitive science. Specifically, while the phenomenon of feeling the embodiment (and acting out of this feeling) is easy to grasp from the subjective human perspective, understanding it from the objective scientific perspective becomes very demanding. We claim that our experimental investigation with the self-learning anthropomorphic robot may help to better understand what the subjective human feelings are and how they relate to artificial beings so that this knowledge eventually will help building machines that are in behavior closer to humans.

The paper is organized as follows: In the next section we introduce the DEP learning rule for the controller and give a first discussion of properties, in particular of balancing at the edge of instability which is loosely related to the edge of chaos concept. We present in Section 3 the the experiments with the robot, **Figure 3** for an overview of the experimental settings and **Table 1** for a list of videos documenting the various experiments. Throughout the paper, we present different

TABLE 1 | Experiments.

Title	Description	Section	Video
Handshake	Human robot interaction by manually imposing a periodic movement	3.4	Video 1 (Supplementary Material)
Arm with pendulum	Suspending a weight from the tip of the arm: self-excitation of a circular pendulum mode	3.5.1	Video 2 (Supplementary Material)
Pendulum responses	Motors are stopped. Recording spring forces of a swinging suspended weight	3.5.1	Video 3 (Supplementary Material)
Shaking horizontal	Horizontally attached bottle, half filled: Response stronger, shaking horizontally, following the axis of the bottle	3.5.2	Video 4 (Supplementary Material)
Shaking vertical	Vertical attachment, half filled: shaking direction mainly along the (now vertical) axis	3.5.2	Video 5 (Supplementary Material)
How to rotate a wheel	Arm attached frontally to a revolvable bar/wheel.	3.6	Video 6 (Supplementary Material)
Rotating wheel II	Parallel wheel—arm arrangement	3.6	Video 7 (Supplementary Material)
Wiping table	Arm with brush starts to wipe a table	3.7	Video 8 (Supplementary Material)
Wiping table modes	Different wiping patterns from reloaded controllers	3.7	Video 9 (Supplementary Material)
Sensor disruptions	With visual input for hand. Camera is turned during behavior. Fast reorganization	3.8	Video 10 (Supplementary Material)
Hand-eye coordination	Coordination develops, such that arm follows a dummy hand	3.8	Video 11 (Supplementary Material)

The videos can be watched at http://playfulmachines.com/MyoArm-1.

methods for the theoretical analysis based on dynamical system theory. Specifically, we introduce in Section 3.5.1 the eigenvalue spectrum of the linearized dynamical operator, in Section 3.5.2 parametric plots for visualizing the "purity" of a behavior, in Section 3.6 local Lyapunov exponents, and in Section 3.7 Hilbert transforms for analyzing more quantitatively the emerging sensorimotor patterns. Central to the paper is the piloting effect introduced in Section 3.3 which explains how the robot may develop a feeling for the internal dynamics of an object, see also Section 3.6 for its relation to the concept of object affordances. This is followed by Section 4 discussing the findings. Some mathematical details are provided in Section 5 (Supplementary Material).

2. ROBOT BEHAVIOR AS A SELF-EXCITED PHYSICAL MODE

The controller we propose is a function that receives at time t a vector of sensor values $x_t \in \mathbb{R}^n$ and sends a vector of motor values $y_t \in \mathbb{R}^m$. In the applications, we use a neurocontroller realized by a one-layer feed-forward network as

$$y_i = g\left(\kappa_i z_i\right) \tag{1}$$

for neuron i, where

$$z_i = \sum_{j=1}^{n} C_{ij} x_j \tag{2}$$

is the postsynaptic potential and C_{ij} is the synaptic connection strength to input j. We use tanh-neurons, i.e., the activation function $g(z) = \tanh(z)$ to get motor commands between +1 and -1. This is also the reason why we did not include a bias term in Equation (1).

An important ingredient for the intended self-excitation of behavioral modes is a controlled destabilization of the system. With a fixed C, this destabilization is controlled by the gain factors κ_i in Equation (1) which regulate the feedback strength for each motor channel i individually. In the experiments we used the definition[1] $\kappa_i = \kappa/\|C_i\|$ where κ regulates the overall feedback strength and $\|C_i\|$ is the norm of the synaptic vector of neuron i. The setup is displayed in **Figure 1**.

2.1. Learning Dynamics

As we aim at self-organization of behavior, we have to define the control signals in a self-consistent way on the basis of the history of sensor signals alone. Let us introduce $x'_t = x_{t+\theta}$, the vector of the sensor values received in the next time step, where θ is a time lag with $\theta = 1$ in the derivations given below (time is measured in discrete update-steps, here 1/100 s).

The self-organized definition of the controller outputs is realized in the following way. Let us postulate the existence of a forward model given by the (possibly state dependent) matrix A so that

$$x'_t = A_t y_t + \xi_t \tag{3}$$

where ξ is the modeling error. This describes the physical dynamics over one time step. Introducing M which is the inverse or pseudoinverse of A we require y to be a function of the future sensor values x',

$$y_t \stackrel{!}{=} M_t x'_t \tag{4}$$

Together with the destabilization, Equation (4) displays the essential idea of our approach to make the system active while keeping motor signals compliant with the world dynamics. In a

[1]This needs a regularization, i.e., in the experiments we use $\kappa_i = \kappa/(||C_i|| + \lambda)$ with $\lambda > 0$ is very small.

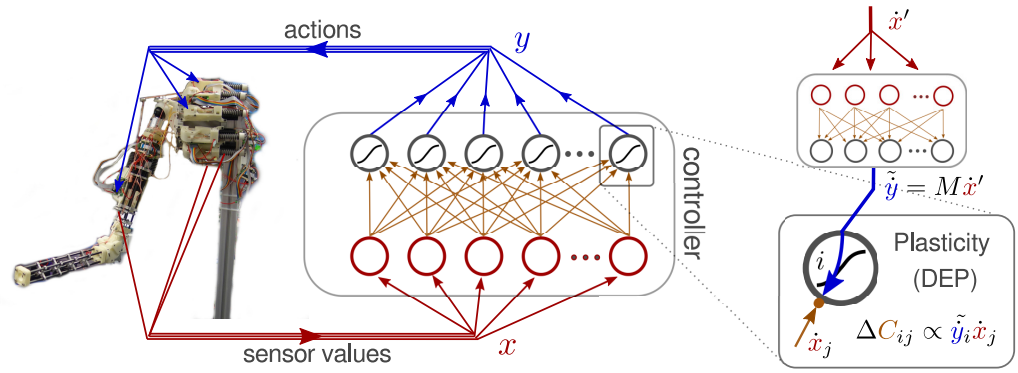


FIGURE 1 | Neural controller network connected to the Myorobotic arm. The inset on the right illustrates the synaptic plasticity rule, called differential extrinsic plasticity (DEP) (Der and Martius, 2015). It is driven by a modified differential Hebbian law, multiplying the time derivatives of the incoming sensor values $\dot{x}$ with the virtual motor values $\tilde{\dot{y}}$, which are generated by the inverse model (Equation 4) from the next input's derivative $\dot{x}'$. In the case of the arm the inverse model is essentially a one-to-one mapping of sensor to motor values.

sense, Equation (4) means that the world's responses, represented by x', signals the controller what to do. But of course the world (i.e., the future sensor values x'_t) is also controlled by the controller through the actions y (Equation 3). The interplay of these effects is the ultimate reason for the self-excitation of modes by self-amplification of system responses.

However, we cannot use Equation (4) directly for generating the control signal y as it contains the future. So, we must find a model for relating the future sensor signals x'_t to their past, i.e., $x_t, x_{t-1}, \ldots$. In other words, we need a time series predictor for the sensor dynamics. Following the derivation in Section 5.1 (Supplementary Material) we obtain eventually the update rule

$$\tau \Delta C_t = M_t \dot{x}'_t \hat{x}_t^\top - C_t \tag{5}$$

or in coordinate representation (omitting the time index)

$$\tau \Delta C_{ij} = \sum_k M_{ik} \dot{x}'_k \hat{x}_j^\top - C_{ij} \tag{6}$$

where $\hat{x} = \dot{x}\|\dot{x}\|^{-2}$, see also **Figure 1**. The matrix M defines the sensor to motor mapping which is one-to-one for normal sensors and negated one-to-one for the delay sensors in the experiments of this paper, see Section 5.2 in Supplementary Material, so the sum in Equation (6) reduces to 2 terms. In general M can be more complicated and can be learned in a prior step.

In accordance with earlier work (Der and Martius, 2015), we call this update rule differential extrinsic plasticity (DEP), though there is a difference with $\dot{x}$ replaced with $\hat{x}$ as the second factor in the update. Equation (5) becomes stationary if

$$C_{ij} = \sum_k M_{ik} \langle \dot{x}'_k \hat{x}_j^\top \rangle \tag{7}$$

where $\langle \ldots \rangle$ is the moving time average. Equation (7) is an important consequence of the update rule, showing that learning converges toward behaviors with a fixed point in correlation space, here a fixed pattern of velocity correlations in sensor space, corresponding to specific attractors in state space. In principle such a fixed correlation pattern corresponds to any behavior like crawling, walking, running, hopping or the like of any amplitude and frequency. If the controller were sufficiently expressive and the sensor to motor mapping appropriate, any (cyclic) mode could potentially be realized by this correlation learning. With the matrix M used in this paper, the spectrum of (stable) behaviors is of course restricted but the variety of the observed motion patterns, see below, is still interesting. To enhance self-organization into periodic patterns, we introduce additional sensors which are copies of the primary sensors but are delayed by a fixed time-delay d, see Section 5.3 in Supplementary Material for technical details.

For the analysis in terms of dynamic systems theory to be given below, we will need the dynamic operator

$$L = MC \tag{8}$$

which describes the mapping from state x to x' for the linearized dynamics (Jacobian of linearized system), see Section 5.1 in Supplementary Material for details. The above learning rule differs from the DEP rule introduced in Der and Martius (2015) by the normalization factor $\|\dot{x}\|^{-2}$ introduced with Equation (6) above. In the experiments this leads to a more continuous activity in the behaviors avoiding potential pauses of inactivity. In relation to our earlier work on predictive information maximization (PiMax) (Martius et al., 2013) there are several differences: the DEP rule uses derivatives of the sensors values for learning where PiMax uses the raw ones, PiMax requires to perform a matrix inversion of the noise-correlation matrix which is not needed here, and finally the resulting behaviors obtained from PiMax get high-dimensional (in terms of attractor dimension, see Martius and Olbrich, 2015 for details) whereas the DEP rule yields low-dimensional behaviors as we will see in the analysis below.

2.2. Properties

The irreducible conjunction of state and parameter dynamics creates a meta-system—formed by controller, body, and environment—with a rich variety of all kinds of attractors. These

can be deliberately switched by manipulative disturbances, creating an attractor meta-dynamics (Gros et al., 2014). This explains why we observe so many different behaviors in the experiments.

2.2.1. Meta-Parameters

Furthermore, there are three parameters in this approach κ, τ and d, which act as meta-parameters for changing the "character" of the SO process. κ determines roughly the amplitude of behavior. In the experiments, the appropriate value for κ is easily found: when increasing κ gradually, a critical value $\kappa_c \approx 1$ is eventually reached. Using $\kappa > \kappa_c$ the amplitude a of an emerging motion pattern is roughly $a \propto \kappa - \kappa_c$ for small a. For larger κ the non-linearities come stronger into play such that the amplitude is never above 1. The time lag of the delay sensors d determines the preferred frequency. The parameter τ determines the time scale for taking previous sensor values into account. This has effects on how quickly the controller parameters are wandering around if not yet in a stationary behavior. It is advisable to have it similar or larger to the period of the expected behavior.

2.2.2. Least Biasing

The implementation of the controller is explicitly given by Equation (1) together with the update rule Equation (5) which obviously has no system specific components. In the experiments we start always with the least biased initial condition, putting the controller matrix $C = 0$ so that all actuators are in their central position. A basic requirement for a "genuine" approach to SO is its independence of specific properties of the controlled system. Obviously, this is realized here in an ideal manner by both the structure of the approach and because there is no specific goal, no target signal, no platform specific information and no biasing.

2.2.3. Theoretical Analysis

It would be interesting and helpful if the wide spectrum of self-organizing behavior could be given a quantitative analysis. In goal oriented learning this can be done by some performance criterion, assessing the difference between actual and intended behavior. However, this seems not appropriate in a true self-organization scenario like that of the present paper. Still one may ask for a profound theoretical analysis of what these systems actually are doing. This paper contributes to that task by presenting several such measures which are partly a bit unorthodox but were quite successful for analyzing behavior generated by the DEP learning rule. Central is the use of dynamical systems theory in several aspects. Specifically, we investigate below the eigenvalue spectrum of the linearized dynamical operator $L = MC$ as introduced in Equation (8), using it for assessing the nature, and the stability of periodic motions, the prevalent modes in this paper. We use local Lyapunov exponents as a more quantitative concept of dynamical system theory, arguing that they may be a first guess for the claimed realization of an edge of chaos system, see Section 3.6 below. Also, parametric plots have proven a viable tool for visualizing the nature of behavior and last but not least, Hilbert transforms of the sensor signals were used for analyzing the phase relations between sensor and motor signals, thereby quantifying the closure of the sensorimotor loop, see Section 3.7.

The nature of the dynamical system generated by the learning rule may also be quantified by a number of methods from complexity theory, information theory (Bialek et al., 2001) and more evolved tools from non-linear dynamics (Kantz and Schreiber, 2004). Akin to this paper are methods for analyzing emergent behavior (Lungarella and Sporns, 2006; Ay et al., 2008; Wang et al., 2012; Schmidt et al., 2013) using information theory. A new quantification based on excess entropy (predictive information) and attractor dimension was recently proposed in Martius and Olbrich (2015) and applied to similar self-organizing behavior as found in this paper. However, there long traces of repetitive behavior where recorded in simulations to estimate entropies. Unfortunately it is impossible to perform this analysis for the fast online learning of the synaptic dynamics, given the time scale of a few seconds or minutes for the behavior generation.

There is some pioneering work in using dynamical systems theory for analyzing behavior generation by fast synaptic plasticity. In Sándor et al. (2015) and Gros (2015), the interesting concept of an attractor metadynamics was introduced which is close to the scenario of this paper. However, their analysis, while pointing in the right direction, is restricted so far to rather simple physical systems in simulation, so that we did not apply it in this paper. Related ideas may also be found in Toutounji and Pasemann (2014, 2016).

2.2.4. Edge of Chaos—The Working Regime for Self-Organization

An essential feature of our approach is the possibility to chose, by the parameter κ, the working regime at the boundary between stable and unstable dynamics. This working regime may be associated with the somewhat vague "edge of chaos" concept (Langton, 1990; Mitchell et al., 1993; Kauffman, 1995; Bertschinger and Natschläger, 2004; Natschläger et al., 2005). As is known from dynamical system theory, this region is not well defined but is otherwise of eminent interest for understanding both life and creativity in natural and artificial beings. Unfortunately, with systems of the physical complexity considered here, a strict mathematical analysis of this region, e.g., by global Lyapunov coefficients, is out of reach of this paper. Nevertheless, in a sense, one can observe in the videos the edge of chaos hypothesis, i.e., to live somewhere between order and fully developed chaos. In fact, on the one hand the systems react very sensitively on weak perturbations, in particular one may observe that the further development of behavior is determined by the initial kick the system experiences or by the interaction with attached objects with an internal dynamics. This extremely sensitive reaction to perturbations is a signature of chaos. On the other hand, see the pendulum video or the bottle shaking experiments, the system also has a high degree of organization as demonstrated by the emergence of long-lived regular orbits. This is the order aspect of the scenario.

Developing quantitative measures for the edge of chaos regime may get the robotic community interested in this very rich, intellectually appealing, and potentially highly useful branch of

dynamical system theory based robotics. But this is a topic of future research.

2.2.5. Platforms for Embodied AI

Finally, let us discuss on which platforms our controller is likely to create useful behavior. First of all, the system has to provide sensory feedback about acting physical forces to make embodiment effects perceivable by the controller. This is, for instance, not the case if all perturbations are perfectly compensated by a low-level PID controller. Secondly, there should be sensors reporting a similar quantity as used to control the actuators, e.g., position sensor for position control or force sensors for force control. Additional sensors are typically integrated into the loop if they show a definite response (correlation) to the motor patterns. Thirdly, the behaviors of interest should be oscillatory. Since we only need the main sensor-to-motor wiring information about the particular robot (which can also be learned) and do not require any other specific information, we expect our system to work with a wide variety of machines including soft robots, but this remains for future research.

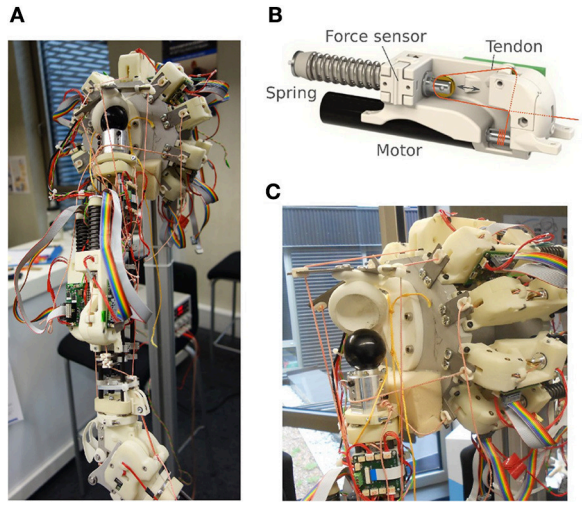


FIGURE 2 | Myorobotic arm (A), a single muscle element **(B)**, and a dislocated shoulder **(C)**. The dislocation happens wickedly as soon as the tendons are getting slack.

3. EXPERIMENTS

The above defined controller was used in the experiments with a tendon driven arm-shoulder system from the Myorobotics toolkit (Marques et al., 2013), see **Figure 2**. The system has 11 artificial muscles, 8 in the shoulder and 2 in the elbow and one affecting both. However, two of the shoulder muscles where disconnected. The muscles are composed of a motor winding up a tendon connected to a spring, see **Figure 2B**. The length of a tendon l is given by the motor encoders and the spring compression by f which is in the interval $[-\alpha, 1-\alpha]$ where α defines pretension (here $\alpha = 0.1$). The length of the tendons is normalized to $l \in [-1, 1]$. We define the sensor values as

$$x_i = l_i + \beta f_i \tag{9}$$

where β regulates the integration of the spring-compression. In the experiments, β was simply set to 1 without further tuning. It is expected that this choice is not critical. After the initialization, where the arm is put in a defined initial position, all tendons are tightened to their pretension, and all l_i are set to zero, the system is put into a position control mode where the controller output y_i defines a target tendon length for each tendon. In the experiments we used the following parameter settings: $\kappa = 0.5$, $\tau = 1\,s$ (Equations 1, 5), delay sensor lag: 0.5 s (Section 5.3 in Supplementary Material), a time distance between x and x' of $0.08\,s$, $r = 10^{-3}$ (Equation 22), and an update frequency of the control loop of 100 Hz.

3.1. Peculiarities of Muscle-tendon Driven Systems

There are a number of features which make the muscle-tendon driven (MTD) systems different from classical robots with joints under rigorous motor control, i.e., the motor positions directly translate into joint angles and into poses. Naively one could think that control is very easy, realized by just pulling the right strings (tendons) for getting a desired arm pose. However, life is much more difficult due to a number of annoying effects. The most obvious effect is seen when tendons are getting slack so that contact with the physical state of the arm is lost altogether. This has to be avoided by keeping a permanent tension on the tendons, which poses another problem: The tension can only be achieved by tightening each tendon up against all the others, each individual tension being reported by the spring length. This means that (i) there are infinitely many combinations of tension forces for a single arm pose and (ii) that the action of a single motor will be reflected in a change of spring length of all other muscles. In other words, actuating a single muscle is reflected by a pattern of sensory stimulation—a whole-body answer.

Furthermore, the combination of friction effects and muscle-pose ambiguity leads to a hysteresis effect. After driving the arm by a sequence of motor commands from pose A to pose B one ends up in a different pose and muscle configuration than A after moving back by reversing the motor commands. In general, this makes the translation of a kinematic trajectory for the arm into motor programs difficult, even more so if there are loads and high velocities involved. Also, the classical approach of learning a model by motor babbling becomes problematic because actions cannot be chosen independently.

We conducted several experiments listed in **Table 1** which demonstrate the essential features of the control scheme. All experiments are done with the same controller with the same initialization ($C = 0$) so that it is only the physical situation that differs between the experiments.

We strongly recommend consulting the videos for better understanding which can be found at http://playfulmachines.com/MyoArm-1.

3.2. Self-Regulated Working Regime

Before presenting the experiments in more detail, let us take a look at the sensorimotor coupling that is created by our controller. One of the crucial features is the self-regulation into a working regime where the tendons are kept under tension even in very rapid motions with notable loads. This is very important as it guarantees the signals from the controller to be executed in a definite way. As a result, in all experiments we never had to face a shoulder dislocation, see **Figure 2C**, which may happen promptly if tendons are getting loose. This is of some importance as this sensible working regime emerges without any additional tuning or calibrating (Wittmeier et al., 2012) the system. For that, the specific sensor configuration (Equation 9) seems to be important, but we did not study it systematically yet and expect other configurations to work as well. A more rigorous analysis in terms of the local Lyapunov exponents will be give in Section 3.6 below.

3.3. The Piloting Effect. Feeling the Embodiment

In the Introduction, we presented a thought experiment illustrating the main features of this work. We did not yet carry out this experiment with humans, but the scenario of getting piloted by the subsystem toward activities of strongest response is just what we observe with the learning arm for a series of very different objects, ranging from the pendulum to the wheel to the wiping a table setting. In any of those situations we could not only observe the piloting effect but also support it by quantitative analysis. Let us remember that any motion of the arm impacts on the inner dynamics which reacts back on the arm via the force response of the internal dynamics, like the water hitting the wall of the bottle. These force responses modulate the sensor values (measuring the length of the tendons) and may become self-amplifying under the learning rule as substantiated by the following arguments (which still need more theoretical support). Point one is that these signals, though tiny, generically may be systematic, building correlations over space and time. Examples are the slow swaying motion of the pendulum or the inertia motions of the water. As the DEP rule enhances correlations by the learning process, any systematic signal persisting over the time scale of learning contributes to the correlation pattern with an enhanced strength. In the experiments, the time scale set by τ was one second, about the same as the internal dynamics of the subsystems. This seems to be the main cause of the piloting effect. Furthermore, the learning system was seen to be the host without preferences of a wide spectrum of attractors giving rise to a kind of attractor morphing. Meaning the learning rule changes the dynamics such that the attractors continuously change, all modulated by the systematic force responses from the subsystem. In other words, the learning system has no resistance to being piloted into a resonance with the subsystem. The piloting by the subsystem is the leading mechanism in the experiments described in the following.

3.4. Manipulability

The dominance of the physical responses makes the system manipulable as any externally applied forces—like a physical robot human interaction—leave their footprint in the sensor values via the changing spring tension. For instance, the arm can always be stopped by simply holding it. The reason is not that the motors are too weak. Instead, $\dot{x} = 0$ is a fixed point of the dynamics of the meta-system to which it relaxes if the mechanical degrees of freedom are frozen manually[2].

Moreover, the system can be entrained by manual interaction into specific behaviors. We demonstrate this in the handshake experiment, see **Figure 3A** and Video 1 in Supplementary Material, where the user is trying to move the arm in a periodic pattern. Besides the possibility to train a robot in this way, the most interesting point is the subjective feeling that comes about when interacting with the robot. In the beginning of such an interplay, the robot seems to have a will of its own as it resists the motions the user is trying to impose. But after a short time the robot follows the human more and more and eventually is able (and "willing") to uphold the imposed motion by itself, see **Figure 4**. Otherwise, depending also on the human partner, the meta-system of robot and human may "negotiate" a joint motion pattern which might be left if the human quits the loop. This can be understood by realizing that any periodic patterns creates a fixed correlation pattern in Equation (7). If the imposed patterns match one of the stable ones, the robot is controlling this pattern by itself. In fact, in the experiments, one can well observe that a "compliant" human is intrigued to follow the system as much as its own intentions, ending up in an orchestrated human-machine dynamical pattern.

Training of a robot by directly imposing motions is not new. The common approaches generate a kinematic trajectory which is afterwards translated into the motor commands by well known engineering methods. This method may run into some difficulties due to the peculiarities of our MTD system discussed in Section 3.1. With DEP learning, imposing the patterns is a process of creative interaction with the system, see also the training of wiping patterns in Section 3.7.

3.5. Emerging Modes

As already mentioned above, DEP learning as formulated in Equation (1) drives systems toward attractors in state space corresponding to fixed velocity correlation patterns in sensor space. The selection of a specific attractor may be realized by the self-amplification of a dynamical seed, generically provided by an initial perturbation from e.g., gravitational forces or by tipping the arm.

3.5.1. Self-Excited Pendulum Modes

In a first experiment, we suspend a weight (the bottle) from the tip of the arm, see **Figure 3B**. With the pivot point (arm) at rest the pendulum may realize ellipsoidal or circular motion patterns with fixed frequency. In general, when considering a pendulum with moving pivot it can perform chaotic motions under certain trajectories of the pivot point. With the pendulum attached to the MyoArm, the motions of the weight exert small inertia forces on the arm which change the spring tensions and

[2]This effect involves the normalization factors and fades away once the regularization comes into play. After that, the system tries to move to the global attractor $\dot{x} = x = 0$.

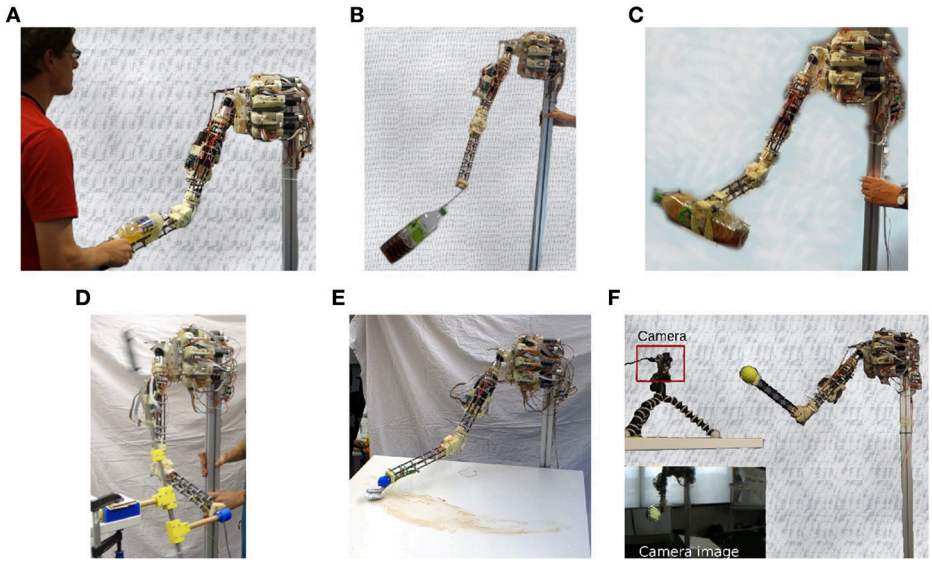


FIGURE 3 | Experimental setups. Handshaking **(A)**, pendulum swinging **(B)**, bottle shaking **(C)**, rotating a wheel **(D)**, wiping a table **(E)**, and hand-eye coordination **(F)**. All experiments are performed with the same controller.

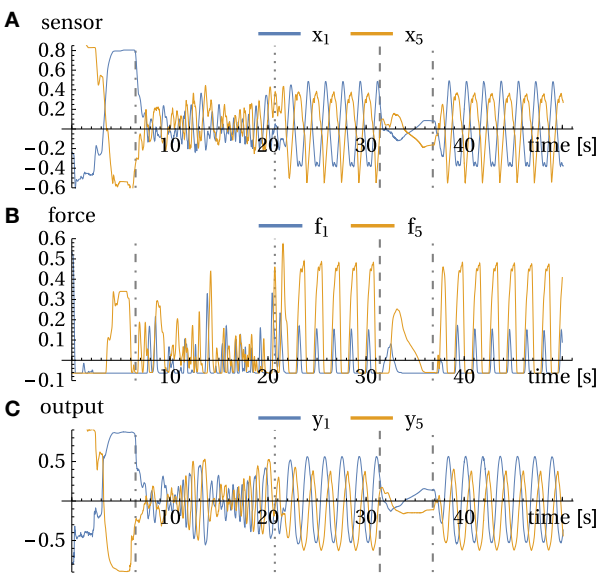


FIGURE 4 | Handshake experiment. (A) sensor values x, **(B)** forces f, and **(C)** motor values y for channels 1 and 5. Events: 6 s: operator is grasping the arm and starts the handshake; 21 s: freezing of parameters and release at 31 s. 35.5 s: bringing arm into resting position, it stays there until 37 s where it got perturbed. See also corresponding Video 1 in Supplementary Material.

thereby leave a footprint in the sensor values. To illustrate this point, **Figure 5** displays the sensor reading for the swinging pendulum with the motors being stopped. While being tiny, these reactions are systematic, leading to the self-excitation of resonant modes according to the piloting effect described in Section 3.3 above.

In Video 2 (Supplementary Material) it can be seen[3] directly how latent velocity correlations are being amplified to end up in stable circular motion patterns of the pendulum. The experiment starts in a situation where the motor activities have settled to rest, interrupted by occasional bursts leaving irregular footprints in the sensor values. As to the piloting effect, we have to verify that, starting with this irregular behavior, the compound system is driven into a resonance with the pendulum and that this resonance behavior is dominated by the (tiny) force responses of the pendulum. This may be supported by analyzing the time lag between measured force and driving signal (motor commands). As shown by **Figure 6A**, the incipiently rather irregular phase relation is followed by a constant phase from time $t > 40$ on. This convergence to a stable mode is also seen by the time evolution of the controller matrix C, see **Figure 6C**.

Let us consider here, as a further bit of analysis, the eigenvalue spectrum of the dynamical operator $L = MC$, which has proven very useful in this work. Actually, if the system would obey the linearized dynamics, any cyclic behavior should be reflected by the existence of a pair of complex eigenvalues. There might be more of such pairs if there are different frequencies involved. Though questionable due to nonlinearities and deficiencies of the linear operator, this analysis may yield reliable results as seen in the pendulum case: **Figure 6B** clearly displays just such a pair of eigenvalues with absolute value (not shown) a little above one. All other eigenvalues have a absolute value significantly smaller than one which makes the corresponding modes short lived[4]. The latter point was investigated in terms of the local Lyapunov

[3]Note that later in the experiment, the string of the pendulum was shortened such that a different sensorimotor coordination emerges.

[4]This is true in particular for the other complex eigenvalue with roughly half the value, apparently belonging to a subharmonics but this still needs some more analysis.

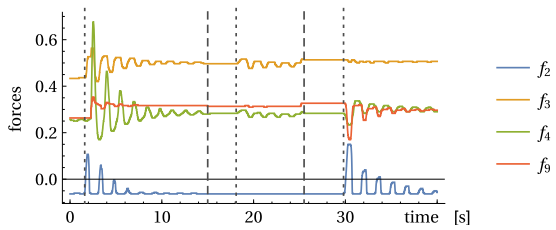


FIGURE 5 | Force measurement with static arm. Displayed are the force measurements with swinging bottle but without active arm movements for muscles 2, 3, 4, and 9 (for clarity). Dotted lines indicate when the bottle was manually set into motion and at dashed lines it was stopped, see Video 3 in Supplementary Material.

exponents, see Section 3.6 below, for remarks on that method. Apart from identifying the oscillatory modes, this eigenvalue analysis also confirms the substantial dimensionality reduction which is also known as a signature of self-organization.

3.5.2. Bottle Shaking Modes

In a next series of experiments we attached a bottle filled with some liquid to the tip of the arm in either horizontal or vertical orientation, see **Figure 3C**. These experiments are meant to support our hypothesis on the piloting effect, i.e., that, under the DEP learning rule, the emerging motion patterns are defined eventually by force responses of the subsystem. With the bottle, the force response is solely generated by the internal motions of the water, i.e., when the water is hitting either the walls or top and bottom of the bottle. Similar to the pendulum, starting with spontaneous movements, the arm soon reaches an oscillatory mode with strong force answers. In the experiment, the emerging shaking motions are indeed more or less aligned with the axis orientation of the bottle, see Videos 4, 5 in Supplementary Material, in correspondence to the piloting effect.

We also performed a more quantitative analysis by using parametric plots to characterize the state dynamics. Oriented at the arm's geometry, we identified two pairs of motor values (y_1, y_3) and (y_6, y_9) which are expected to be discriminating the direction of the arm movement, i.e., to have different phase relations for the horizontal and vertical arm movements, respectively. When plotting the time course of (y_1, y_3) and (y_6, y_9) in the plane, fixed phase relations translate into typical ellipsoidal figures. In **Figures 7C–F** we compare the phase relation for the horizontal and vertical setup (violet and orange line, respectively) for two behavioral modes (see **Figures 7A,B** for the time course and intervals) and indeed find that they are different and often orthogonal to each other. The emerging motion pattern is determined by the axis direction of the bottle, with the reactive forces of the water as the only information for that direction. Metaphorically, the robot can "read" the information about the nature of the environment by just getting into dynamical contact with the latter in a completely self-organized way.

In **Figures 7G,H** we present the time evolution of the matrix elements C_{3j} representing the connection to the motor unit 3. As starting from the zero-initialization, one can see how first correlations build up due to the dynamics of the C matrix (Equation 5). The following behavior is highly transient until convergence is (roughly) reached where the dynamics gets more stationary. Any perturbation or change in conditions leads to an adjustment of the controller, always aiming for a mode where high velocity correlations appear.

3.6. Rotating a Wheel

A further example for the piloting mechanism (Section 3.3) and the discovery of dynamic object affordances (as discussed below) is the robot arm connected to a wheel, see **Figure 3D**. In Der and Martius (2015), the emergence of rotational modes was demonstrated for a humanoid robot with revolution joints and in simulation. With the MyoArm, we have a much more challenging situation. In the experiments, the tip of the arm is attached to the crank of a wheel, implemented as a revolvable bar with weights for giving it the necessary moment of inertia. In Video 6 (Supplementary Material), initially the connection between the arm and the wheel was rather loose so that for small movements there is no definite response from the rotation of the wheel. After improving this connection, an initial push by the experimenter was sufficient to excite a rotation mode that persists over time and is stable under mild perturbations. It is as if the controller "understood" how to rotate the wheel, although it is just the result of force exchange in combination with correlation learning, i.e., by the mechanism described in Section 3.3. When positioning the wheel in parallel to the arm, the modes were emerging even more readily as seen in Video 7 (Supplementary Material). Furthermore, the system may be changed in frequency by changing just the time-delay d as shown earlier (Martius et al., 2016).

For an analysis, we may use here the method of local Lyapunov exponents, given by the eigenvalues of the dynamical operator $L = MC$ transforming sensor states x to x' under the linearized dynamics. **Figure 8A** displays the results. The point of interest are the two largest exponents which are slightly above zero. They represent the rotational mode. Being above zero means that they are actually instable which was to be expected given the slight destabilization of the system controlled by the parameter κ. However, the system dynamics is kept from exploding by the nonlinearities so that the rotation modes are stable but all other modes have to die out, i.e., their Lyapunov exponents have to be below zero. It is also illustrative to consider the absolute change of the controller matrix as displayed in **Figure 8B** (top). At the beginning of a new mode the changes are large and then settle to a background level. When, for instance, the rotation is externally changed (second 40 and 71) then again a high rate of change is observed. The coupling of the sensors to motors also changes qualitatively between the modes as illustrated at the example of motor 6 in **Figure 8B** (bottom).

The constitutive role of the body-environment coupling is also seen if a torque is applied to the axis of the wheel. Through this external force we may give the robot a hint of what to do. When in the fluctuating phase, the torque immediately starts the rotation which is then taken over by the controller. Otherwise, we can also "advise" the robot to rotate the wheel in the opposite direction. This can be considered as a kinesthetic training procedure, helping the robot in finding and realizing its task through direct mechanical influences.

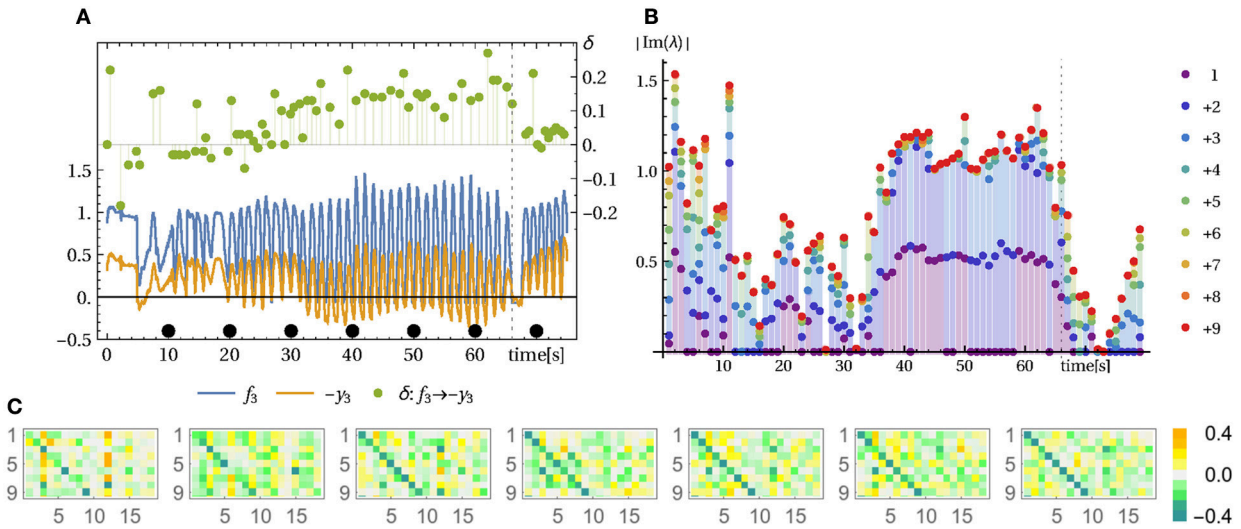


FIGURE 6 | Pendulum modes. See Video 2 in Supplementary Material. **(A)** Force sensors and control signal of muscle 3 and their time lag. The measured force (spring compression) and the control signal y (desired tendon length) follow a similar trajectory with inverted sign (note $-y$). The time lag δ (right axis in seconds) between force and motor value (same result for other muscles) indicates that initially the control and the environmental influences are not in sync whereas in the swinging mode (from 33 s on) a stable phase/time-lag relation is observed. **(B)** Displayed are the absolute imaginary parts of the eigenvalues of the linearized system dynamics (Jacobian L, below Equation 5) (averaged over 1 s) and cumulatively plotted $(1, 1+2, 1+2+3, \ldots)$. During the pronounced oscillation between 35 and 68 s there is one pair of dominant complex eigenvalues. **(C)** Corresponding controller parameter C at the seconds $10, 20, \ldots, 70$ (from left to right) as indicated by the black dots in **(A)**. At second 66 the string of the bottle was shortened causing the mode to break down immediately, see Video 2 in Supplementary Material.

Finally, these results can also be of interest for elucidating the spontaneous discovery of object affordances. Following Gibson (1977) theory of affordances, object affordances are defined as a relation between an agent and its environment through its motor and sensing capabilities (e.g., graspable, movable, or eatable and so on). In this sense, in the same way as a chair affords sitting or a knob affords twisting, the wheel in our experiment affords rotating it, the bottle affords shaking and pouring and so on. This is of immediate interest for embodied AI as affordances are prerequisites for planning complex actions. Because our controller generates dynamic and typically oscillatory movements it can only discover dynamic afforcances, such as shaking, turning etc. but will not find static ones such as sitting on a chair or leaning against a wall.

3.7. Wiping

In the case of the wheel setup, above, the embodiment strongly constrains the possible motion patterns. In the next setup the agent-environment coupling imposes a much milder restriction on the behavior: the robot is equipped with a brush and a table is placed in its work-space, see **Figure 3E**. The table height is about 5 cm above the initialized resting position. Video 8 in Supplementary Material demonstrates how, by the combination of the restricting table surface and the manual force, the robot is guided into the two-dimensional wiping mode. Actually, even without this guidance the system typically learns a wiping behavior, because movements perpendicular to the table are strongly damped such that the directions along the table plane may create the highest velocity correlation and thus dominate the generated motion patterns. Later in this video, the robot is forced by hand into a different behavior.

The analysis of the dynamics during this experiment revealed that the wiping patterns where not stationary as it appeared in the video, but are actually slowly drifting. We devised a method to quantify such high-dimensional oscillatory behavior. It considers the phase difference between the different degrees of freedom. For each oscillatory signal we can associate a phase variable that continuously runs from $-\pi$ to π using the Hilbert transform. Now we can compute the phase difference between the signals from different sensors, for instance. Post-processing is applied to avoid unnecessary 2π phase jumps and to smoothen the signal for better visibility.

In a stable oscillation, the phase difference should stay constant over time. In **Figure 9A**, these phase differences are presented for the wiping experiment. One can see that already before manual interaction, the meta-system is in a transient behavior, with changing phase relations slowly over time. We interpret this as a wandering through the metastable cyclic attractors induced by the learning dynamics. We may also call this a self-induced attractor morphing. During interaction (second 11 onward) the changes are initially stronger, fading out later. After releasing the arm (second 22), behavior persists for a few seconds and then is again drifting away. The corresponding controller matrices also show a significantly different structure in the course of the experiment. With the phase analysis using Hilbert transform we can thus analyze pseudo-stationary high-dimensional motion patterns and we believe this methods is also helpful to analyze other systems where attractor morphing occurs.

So, what appeared as stationary actually was a transient behavior. As explained above, there is a potentially infinite reservoir of attractors in C-space, with the learning dynamics

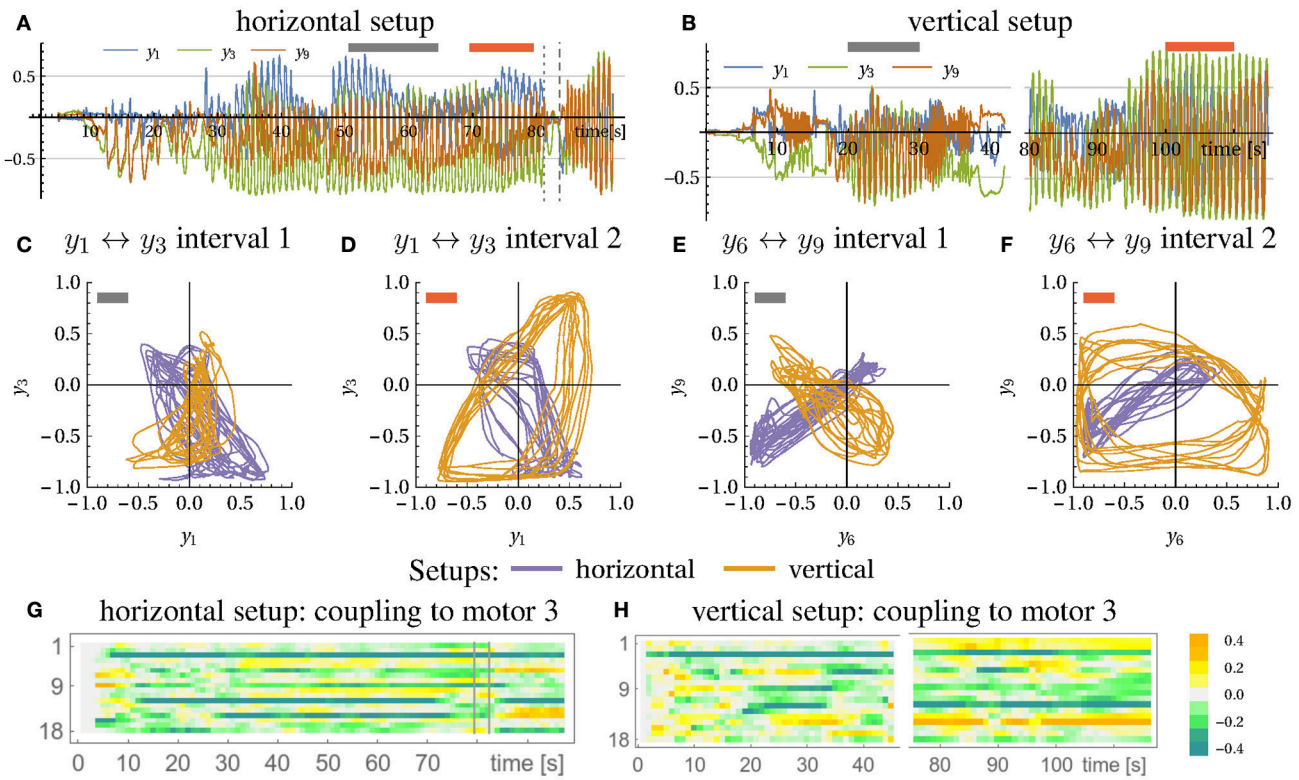


FIGURE 7 | Horizontal and vertical bottle shaking experiment. Depicted are the time traces of the motor values for the horizontal setup **(A)**, see Video 4 in Supplementary Material, and the vertical setup **(B)**, see Video 5 in Supplementary Material. At the marked regions (gray and red bar) both setups are compared in **(C–F)** with respect to their motor relation (motor 1 vs. 3 and 6 vs. 9). It is visible that the motions in both setups are mostly orthogonal to each other. **(G,H)** shows the evolution of the coupling of the 18 sensors to muscle 3 over time (corresponding to row 3 in *C*). In both cases the system starts at $C = 0$. In the horizontal case the arm was stopped and released at times indicated by vertical lines.

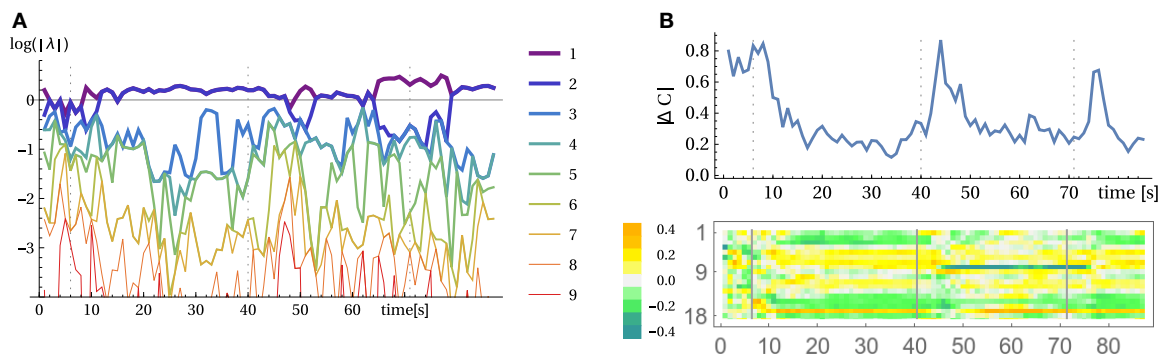


FIGURE 8 | Analyzing the wheel rotation for the parallel setup. (A) Local Lyapunov exponents of the linearized dynamics, i.e., logarithm of absolutes of the eigenvalues of *L* (see below Equation 5). **(B)** Temporal evolution of the controller matrix *C*. Absolute changes of *C* over one second (top) and changes in the coupling of all sensors to motor 6 (row 6 of C over time). The arm started to rotate at second 7 and at second 40 and 71 the rotation was manually inverted (vertical lines) see Video 7.

slowly and continuously morphing these attractors. Being more or less a speculation so far, this opens a view into a fascinating species of dynamical systems generated by the learning rule in specific agent-environment couplings. Moreover, this also should substantially improve our understanding of the edge of chaos hypothesis as an overarching concept.

Otherwise, by simply storing the weights (C) of the controller, these patterns can be collected into a repertoire. Video 9 in Supplementary Material shows the recall of and switching between such wiping modes, see **Figure 9B**. For the transition into a different mode the controller was changed abruptly, nevertheless a smooth transition into the new behavior occurs,

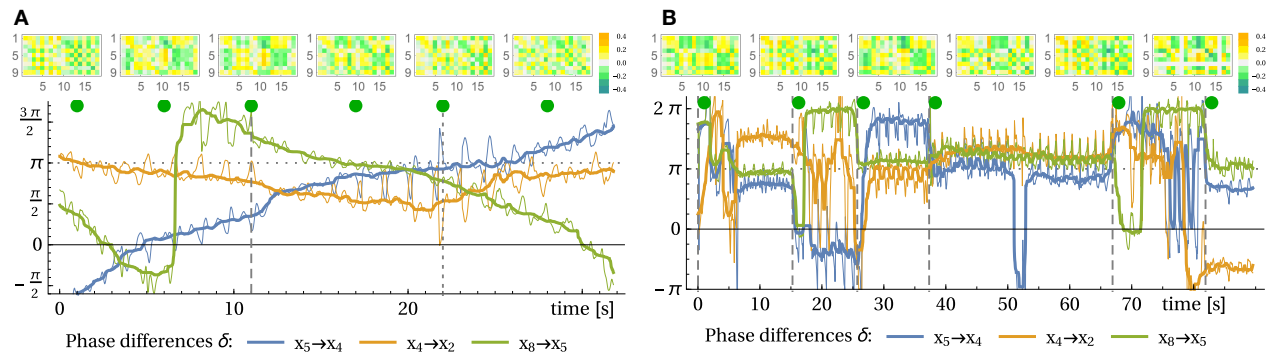


FIGURE 9 | Learning to wipe a table with a brush and recall of different wiping patterns. (A) Shown are the phase differences between a selection of sensor values (bottom) and the controller matrices (top) at different points in time indicated by green dots. The thick lines show the sliding median of 2 s windows for better visibility. Note that jumps of 2π are equivalent to no phase change. See corresponding Video 8 in Supplementary Material (Time 0 in the plot is at the first cut in the video). From second 11 (dashed line) to 22 (dotted line) the arm was trained to perform a different movement, which persists for a few seconds until the system drifts away. **(B)** Recall of previously stored behaviors. At vertical dashed lines, a static controller was loaded. Phase differences between a selection of sensor values (bottom) and the controller matrices (top) (times, see green dots). See corresponding Video 9 in Supplementary Material. Observe the transients between the behaviors, which are sometimes long, e.g., 15 s for controller 4.

suggesting that most static controllers have a large basin of attraction.

3.8. Hand-Eye Coordination

In the previous experiments, the sensorimotor loop was closed in proprioceptive space alone, muscle lengths and tensions generating muscle feelings with the ensuing piloting effect, see Section 3.3. This section investigates the integration of additional sensors given by a camera reporting the spatial coordinates of a green colored object connected to the tip of the arm, called the *fist* in the following. The camera was positioned to observe the arm from the front, see **Figure 3F**, but other positions would also work. The $x - y$ coordinates of the object are obtained from the green pixels' center of gravity, whereas the z coordinate is given by the size of the pixel cluster. These coordinates are scaled between -1 and +1 as all the other sensors. To better compete with the 9 proprioceptive sensors, the corresponding synaptic weights were multiplied by a factor of 3 (before normalization). No other measures were taken, in particular, all entries for the vision channels in the model matrix M were put to zero in accordance with the least biasing commitment described in Section 2.2. In the experiments, we observed that the robot engaged into all kinds of trajectories similar to those of the purely proprioceptive case, i.e., as if the camera were not present. However, a simple inspection of the C matrix reveals a strong involvement of the vision channels in the generation of the modes, see the red-framed rows in **Figures 10C,D**. The constitutive role of the camera can also be seen by the following experiment.

3.8.1. Adaptation to Sensor Transformations—Rotating the Camera

In this setting we rotate the camera about its optical axis while the system is running and DEP learning is on, with a time scale of a few seconds. Initially the camera is rotated about its axis to -90 degrees, see **Figure 10E**. When a relatively stable motion occurs (limit cycle), the camera is slowly rotated to a normal orientation (0 degrees). During that process, the motion pattern of the arm changes until, after stopping the camera rotation, a new attractor behavior is reached. Together with **Figure 10** this shows that the emerging patterns are generated with the camera closely integrated[5]. Eventually, upon rotating the camera further to +90 degrees, the motion of the arm even stops until, after about 15 s, a new consistent behavior emerges, see Video 10 in Supplementary Material and **Figure 10**. The experiment shows that DEP learning generates motion patterns with the camera tightly integrated, i.e., proprioceptive and vision channels are strongly mixed. We remark that readaptation and reorganization of behavior takes place on a time scale of a few seconds.

3.8.2. Hand-Eye Coordination. Emerging Central Pattern Generator

As discussed above, DEP learning potentially integrates all sensor channels, converging toward a fixed point in correlation space which corresponds to a periodic motion pattern in state space. This is seen from the parametric plots in **Figure 11C**, first row displaying a proprioceptive vs. one of the vision channels. Despite the strong perturbations in the complex physical setting, a distinct phase relation between vision and proprioception is seen. This is another corroboration of the integrative strength of DEP.

In a next experiment, we investigate the acquired sensorimotor mappings in more detail. During learning the camera delivers a periodic trajectory in a 3D space, correlated tightly with proprioception. What if we substitute the camera coordinates by those of a fake, or virtual, trajectory. In the experiment, we wait until the system, with the camera included, settled into a stable motion pattern. Then, we freeze the controller matrix C and cover the fist with a white cap making

[5]During a periodic motion pattern the controller matrix C stays relatively constant, i.e., a fixed point in correlation space is reached.

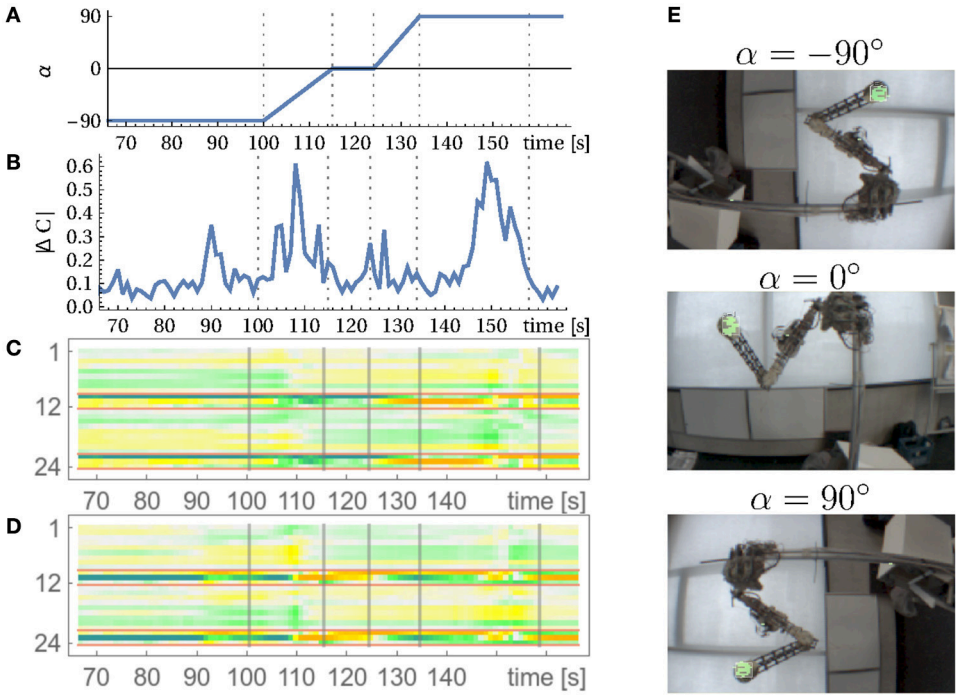


FIGURE 10 | Adaptation to disruptive changes in the vision sensors. During the experiment the camera was rotated about its optical axis. **(A)** Camera angle in degrees and corresponding camera images in **(E)**. **(B)** Change of the controller matrix *C* over time (averaged over 1 s). **(C,D)** Evolution of the coupling of the 24 sensors to one muscle (3 and 8) over time (row 3 and 8 of *C* over time). Red-framed sensors are the vision sensors (and their time-delayed version). Vertical lines indicate times of camera rotation and the point of reentering a stable motion at 160 s. See corresponding Video 10 in Supplementary Material and the text for details.

it invisible to the camera's green object detector so that the vision sensors are frozen. Now we use a dummy fist (green ball attached to a stick) to generate camera coordinates by hand, see **Figures 11D,E** for a normal and a dummy fist camera view, respectively.

As demonstrated in Video 11 (Supplementary Material), moving the dummy generates defined movements of the arm, although the arm would typically not follow the dummy if it is arbitrarily moved. However, if the dummy is moved along a similar path as the original movement, the arm is following the dummy, it can be even driven into trajectories with various velocities, and can be stopped deliberately, see Video 11 in Supplementary Material. In **Figure 11A** the time trace of one of the vision sensors and a proprioceptive sensor for the course of the experiment visualizes this behavior. By comparing the parametric plots in **Figures 11B,C**, first and second row we confirm the similarity between the original and the virtual camera trajectory. On the other hand, **Figures 11B,C**, third row shows that a different relation between the sensors occur if the dummy trajectory is in the opposite direction.

Another interesting point is that behaviors can not only be replayed and combined, as demonstrated in the wiping case, but also be driven by virtual trajectories with (moderately) varying shapes and velocities. This can be operationalized for deliberate control. For instance, a central pattern generator could be used to generate the virtual trajectory, giving the opportunity to systematically vary frequency and shape of the emerging behaviors. Furthermore, the emergence of hand-eye coordination and the possibility to deliberately control the arm using virtual trajectories could be of some interest for the development in infants during Piaget's first phase.

3.9. Perspectives for Goal Oriented Behavior

Though this paper is devoted to robotic self-organization, let us have a remark on generating user chosen behaviors. The basic idea is the following: the classical control setting is a two level hierarchy where the goal driven controller is applied directly to the low level PID controller realizing the action execution. Here, we advocate for the inclusion of a third, intermediary level, meaning that the higher-level controller is realizing its goals by manipulating the above mentioned meta-system with its wealth of latent behaviors waiting to be excited. How this could be effectively done is still to be investigated. However, the potential success of this extended hierarchy of control is suggested by the experiments. In fact, if we are able to influence the meta-system by hand, why not by just superimposing additional motor signals on the self-regulated meta-system. The use of the approach is encouraged by the mentioned ability of the meta-system to uphold a resilient working regime even under extreme external perturbations, preventing, for instance, shoulder dislocations.

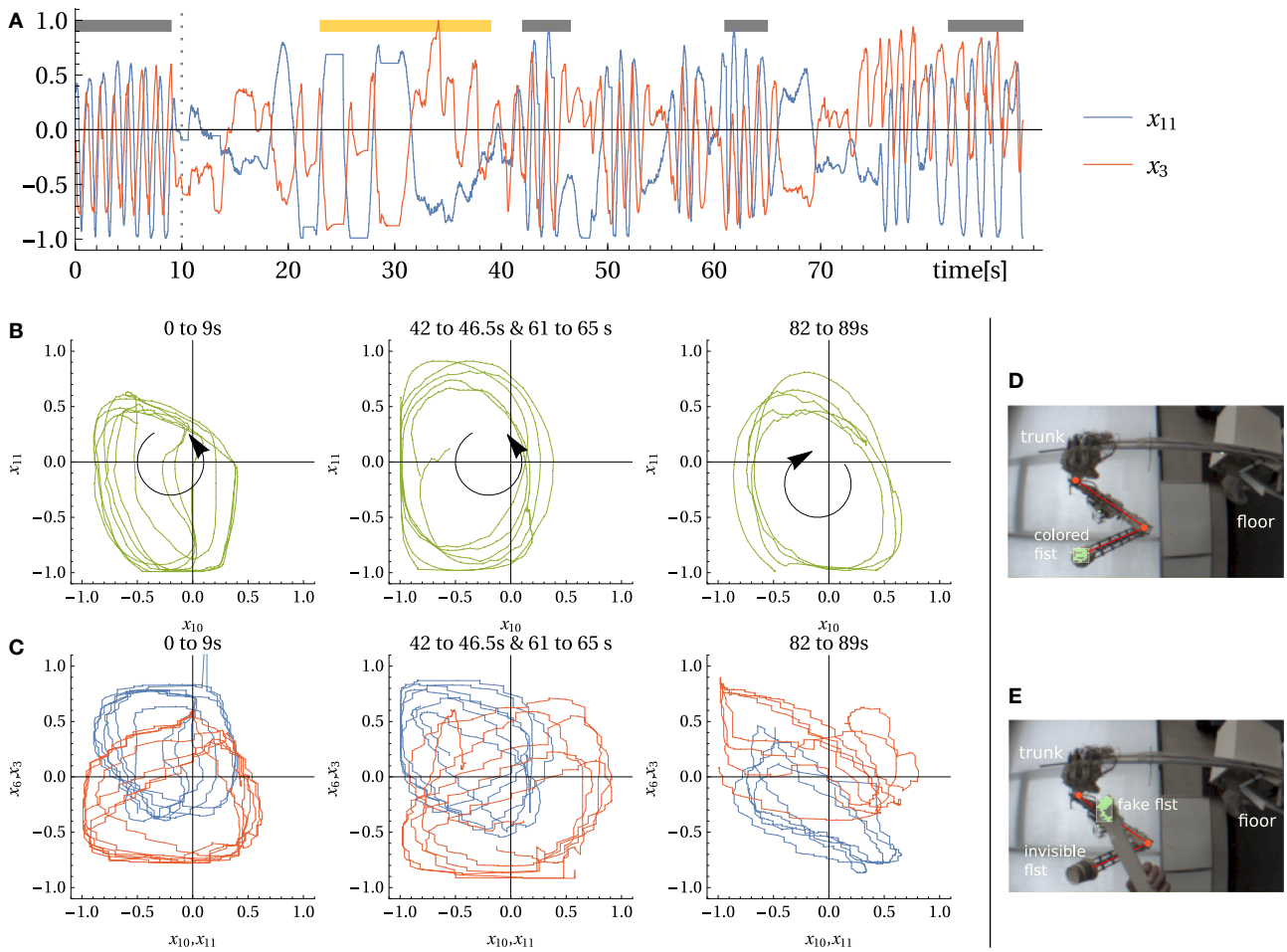


FIGURE 11 | Experiment with camera input. Hand-eye coordination and tracking. (A) a proprioceptive sensor x_3 and a vision sensor x_{11} (up-down direction) over time. The vertical line indicates when the fist was covered with a cap (see **E**). Black bars indicate time intervals used in **(B,C)**. The yellow bar indicates the cut out part of the corresponding Video 11 in Supplementary Material. **(B)** Trajectory in vision sensor space for different parts. Left: *original* movement (with normal camera sight **(D)**, middle: two *similar driven* behaviors, right: *inverted direction* movement. **(C)** The same trajectory relating vision to proprioception sensors $x_{11} \rightarrow x_3$ and $x_{10} \rightarrow x_6$. **(D,E)** camera picture for normal and dummy-fist case.

4. DISCUSSION

This paper is seen as a further step toward a general theory and practical realization of self-organization (SO) for embodied AI. There are many facets to such a general idea worth to be investigated. In many cases, SO is considered as either self-exploration for scrutinizing the gross properties of the system (to be deliberately controlled afterwards), or (wishfully) used for the acquisition of behavior primitives. While this is often ticked-off as superfluous, to be replaced by well known methods like motor babbling, SO definitely has its realm if systems become larger. This has been demonstrated by a number of successful examples (Der and Martius, 2012, 2013, 2015; Der, 2016) attributing SO a much wider range of applicability. We claim that the results of this paper are a further step as they extend that range to composed systems consisting of the actual robot connected to a subsystem with an unknown internal dynamics. In the paper we ask how a robot may establish dynamical contact with a subsystem, eventually recognizing its identity, if there is no information or model of the subsystem's inner dynamics. Humans seem to have no problems there as they develop a feeling, by their muscle tensions, for the reactions of the subsystem. However, it is not clear what this subjective feeling is and how it is used for controlling the interacting system.

As a first insight offered by our DEP controlled robot, we note that the artificial system does not need any curiosity or other higher level concepts for producing the observed human like behaviors. Oriented at the similarity between our anthropomorphic robot and human behavior, we may question the ontological status of these higher level concepts also in humans. Furthermore, we could reveal a very subtle but dominating effect: by the mere feedback through the internal

dynamics of the object, the robot is learning to answer with a very specific sensorimotor pattern to each of the objects. So, the robot discovers the identity of the attached object without knowing anything of its dynamical properties which may be very complex like the water in the bottle. This may be a further example how the robot can both model and substantiate concepts from cognitive science, here Gibson's object affordances. Furthermore, as we could uncover by the analytical tools developed in this work, the emergence of the combined mode and the eventual identification of the attached object—by establishing dynamical contact—is explained by a subtle mechanism which we call piloting.

Unfortunately, due to the high complexity of the system and the subtlety of the effect, a rigorous mathematical analysis is not possible so far. Nevertheless, using some concepts of dynamical system theory, we could establish tentative findings. By keeping the system at the border to instability we find a potentially infinite reservoir of (limit cycle) attractors "waiting" to be excited. Besides converging toward one of these attractors, the rich reservoir of further phenomena could possibly be related to concepts like attractor meta-dynamics (Gros, 2015; Sándor et al., 2015), the so called meta-transients (Negrello and Pasemann, 2008) and the mentioned self-induced attractor morphing. Altogether, these concepts may serve as a characteristic for self-organized behavior in the sensorimotor loop, possibly endowing even the edge of chaos concept with a new realm. There again, we emphasize that the outstanding sound mathematical analysis of these concepts can more reliably reveal their enormous potential for constructing and building such self-learning machines with their creative properties.

It is also important to note that "reading" the object's properties through the mere feedback from its internal dynamics is a direct consequence of those dynamical system properties. Considering the similarity with human behavior again, we may ask if humans also work in this dynamical regime at the border of instability and what the possible consequences are. It must be left to future work to reveal the thereby expected cross fertilization between robotics and cognitive science. Furthermore, the spontaneous identification of dynamical object affordances may be also of some interest for both robotics and embodied AI.

In short, we claim that experimental investigation with anthropomorphic, self-learning robots not only generates interesting behaviors in complex robotic systems. It may also help to better understand what subjective human feelings of physical interactions are, how they can be rooted in sensorimotor patterns, and how these concepts may feed back onto robotics. Hopefully, this knowledge may eventually help building machines that are as close to humans as possible.

Last but not least, let us briefly compare our results with the literature on SO in robotics. While this paper focuses on the SO of behavior for robots of a given morphology, much of the literature is devoted to SO for self-assembling and self-repairing (Murata and Kurokawa, 2012), and eventually self-replicating (Griffith et al., 2005) systems. Very influential for the topic is the paper Pfeifer et al. (2007) presenting the whole spectrum of bioinspired robotics. The central idea is that control is outsourced to the morphological and material properties, see also Hauser et al. (2012), Pfeifer and Gómez (2009), Paul (2004), Pfeifer and Bongard (2006), Pfeifer and Scheier (1999), and Pfeifer et al. (2012). This is in line with our work, as our controller is developing everything from the interplay with the physics of the system. However, to our knowledge previous work does not reach robots of such complexity as demonstrated here. Related to our work is the multiple attractor concept (Tani and Ito, 2003; Gros, 2015; Sándor et al., 2015), which was not yet applied to real robots. Another body of literature exists on SO in swarms (Bonabeau et al., 1997, 1999; Rubenstein et al., 2014; Blum and Groß, 2015) to get swarm intelligence (Engelbrecht, 2006; Nouyan et al., 2008), but there is no relation to our work which is devoted to the development of individual robots.

AUTHOR CONTRIBUTIONS

RD and GM conceived and conducted the experiments. GM analyzed the data. RD and GM wrote the paper.

ACKNOWLEDGMENTS

We thank Alois Knoll for inviting us to work with the Myorobotic arm-shoulder system at the TUM. Special thanks go also to Rafael Hostettler for helping us with the robot and control framework. GM received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no. [291734].

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00008/full#supplementary-material

REFERENCES

Ay, N., Bertschinger, N., Der, R., Guttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *Eur. Phys. J. B* 63, 329–339. doi: 10.1140/epjb/e2008-00175-0

Bertschinger, N., and Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Comput.* 16, 1413–1436. doi: 10.1162/089976604323057443

Bialek, W., Nemenman, I., and Tishby, N. (2001). Predictability, complexity and learning. *Neural Comput.* 13:2409. doi: 10.1162/089976601753195969

Blum, C., and Groß, R. (2015). "Swarm intelligence in optimization and robotics," in *Springer Handbook of Computational Intelligence* (Berlin; Heidelberg: Springer), 1291–1309.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Number 1. New York, NY: Oxford University Press.

Bonabeau, E., Theraulaz, G., Deneubourg, J.-L., Aron, S., and Camazine, S. (1997). Self-organization in social insects. *Trends Ecol. Evol.* 12, 188–193. doi: 10.1016/S0169-5347(97)01048-3

Bongard, J. (2015). Using robots to investigate the evolution of adaptive behavior. *Curr. Opin. Behav. Sci.* 6, 168–173. doi: 10.1016/j.cobeha.2015.11.008

Der, R. (2016). In search for the neural mechanisms of individual development: behavior-driven differential hebbian learning. *Front. Robot. AI* 2:37. doi: 10.3389/frobt.2015.00037

Der, R., and Martius, G. (2012). *The Playful Machine - Theoretical Foundation and Practical Realization of Self-Organizing Robots.* Berlin; Heidelberg: Springer.

Der, R., and Martius, G. (2013). "Behavior as broken symmetry in embodied self-organizing robots," in *Advances in Artificial Life, ECAL 2013*, eds P. Liò, O. Miglino, G. Nicosia, S. Nolfi, and M. Pavone (Cambridge, MA: MIT Press), 601–608. doi: 10.7551/978-0-262-31709-2-ch086

Der, R., and Martius, G. (2015). Novel plasticity rule can explain the development of sensorimotor intelligence. *Proc. Natl. Acad. Sci. U.S.A.* 112, E6224–E6232. doi: 10.1073/pnas.1508400112

Engelbrecht, A. P. (2006). *Fundamentals of Computational Swarm Intelligence*. Hoboken, NJ: John Wiley & Sons.

Erez, T., Lowrey, K., Tassa, Y., Kumar, V., Kolev, S., and Todorov, E. (2013). "An integrated system for real-time model predictive control of humanoid robots," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on* (New York, NY: IEEE), 292–299.

Gibson, J. J. (1977). *The Theory of Affordances, in Perceiving, Acting, and Knowing. Towards an Ecological Psychology.* Hoboken, NJ: John Wiley & Sons.

Griffith, S., Goldwater, D., and Jacobson, J. M. (2005). Robotics: self-replication from random parts. *Nature* 437, 636–636. doi: 10.1038/437636a

Gros, C. (2015). *Complex and Adaptive Dynamical Systems: A Primer*. Berlin; Heidelberg: Springer.

Gros, C., Linkerhand, M., and Walther, V. (2014). "Attractor metadynamics in adapting neural networks," in *Artificial Neural Networks and Machine Learning–ICANN 2014* (Berlin; Heidelberg: Springer), 65–72.

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., and Maass, W. (2012). The role of feedback in morphological computation with compliant bodies. *Biol. Cybern.* 106, 595–613. doi: 10.1007/s00422-012-0516-4

Kantz, H., and Schreiber, T. (2004). *Nonlinear Time Series Analysis*, Vol. 7. Cambridge, UK: Cambridge University Press.

Kauffman, S. (1995). *At Home in the Universe: The Search for the Laws of Self-organization and Complexity*. New York, NY: Oxford University Press.

Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi: 10.1016/j.tibtech.2013.03.002

Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* 42, 12–37. doi: 10.1016/0167-2789(90)90064-V

Lungarella, M., and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Comput. Biol.* 2:e144. doi: 10.1371/journal.pcbi.0020144

Manoonpong, P., Geng, T., Kulvicius, T., Porr, B., and Wrgtter, F. (2007). Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput. Biol.* 3:e134. doi: 10.1371/journal.pcbi.0030134

Marques, H. G., Imtiaz, F., Iida, F., and Pfeifer, R. (2013). Self-organization of reflexive behavior from spontaneous motor activity. *Biol. Cybern.* 107, 25–37. doi: 10.1007/s00422-012-0521-7

Martius, G., Der, R., and Ay, N. (2013). Information driven self-organization of complex robotic behaviors. *PLoS ONE* 8:e63400. doi: 10.1371/journal.pone.0063400

Martius, G., Hostettler, R., Knoll, A., and Der, R. (2016). "Compliant control for soft robots: emergent behavior of a tendon driven anthropomorphic arm," in 2016 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (New York, NY), 767–773.

Martius, G., and Olbrich, E. (2015). Quantifying emergent behavior of autonomous robots. *Entropy* 17,7266. doi: 10.3390/e17107266

Mitchell, M., Hraber, P. T., and Crutchfield, J. P. (1993). Revisiting the edge of chaos: evolving cellular automata to perform computations. *Comp. Syst.* 7, 89–130.

Mordatch, I., Todorov, E., and Popovic, Z. (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* 31:43. doi: 10.1145/2185520.2185539

Murata, S., and Kurokawa, H. (2012). *Self-Organizing Robots*, vol. 77. Berlin; Heidelberg: Springer.

Natschläger, T., Bertschinger, N., and Legenstein, R. (2005). "At the edge of chaos: real-time computations and self-organized criticality in recurrent neural networks," in *Advances in Neural Information Processing Systems 17*, eds L. K. Saul, Y. Weiss, and L. Bottou (MIT Press), 145–152. Available online at: https://papers.nips.cc/paper/2671-at-the-edge-of-chaos-real-time-computations-and-self-organized-criticality-in-recurrent-neural-networks

Negrello, M., and Pasemann, F. (2008). Attractor landscapes and active tracking: the neurodynamics of embodied action. *Adapt. Behav.* 16, 196–216. doi: 10.1177/1059712308090200

Nouyan, S., Campo, A., and Dorigo, M. (2008). Path formation in a robot swarm. *Swarm Intell.* 2, 1–23. doi: 10.1007/s11721-007-0009-6

Park, H.-W., and Kim, S. (2015). Quadrupedal galloping control for a wide range of speed via vertical impulse scaling. *Bioinspir. Biomimet.* 10:025003. doi: 10.1088/1748-3190/10/2/025003

Paul, C. (2004). "Morphology and computation," in *Proceedings of International Conference on Simulation of Adaptive Behavior* (Cambridge, MA: MIT Press), 33–38.

Pfeifer, R., and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. Cambridge, MA: MIT Press.

Pfeifer, R., and Gómez, G. (2009). "Morphological computation - connecting brain, body, and environment," in *Creating Brain-Like Intelligence*, eds B. Sendhoff, E. Körner, O. Sporns, H. Ritter, and K. Doya (Berlin; Heidelberg: Springer), 66–83.

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Pfeifer, R., Lungarella, M., and Iida, F. (2012). The challenges ahead for bio-inspired 'soft' robotics. *Commun. ACM* 55, 76–87. doi: 10.1145/2366316.2366335

Pfeifer, R., and Scheier, C. (1999). *Understanding Intelligence*. Boston, MA: MIT Press.

Posa, M., Cantu, C., and Tedrake, R. (2014). A direct method for trajectory optimization of rigid bodies through contact. *Int. J. Robot. Res.* 33, 69–81. doi: 10.1177/0278364913506757

Raibert, M., Blankespoor, K., Nelson, G., and Playter, R. (2008). Bigdog, the rough-terrain quadruped robot. *IFAC Proc. Vol.* 41, 10822–10825. doi: 10.3182/20080706-5-kr-1001.01833

Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295

Sándor, B., Jahn, T., Martin, L., and Gros, C. (2015). The sensorimotor loop as a dynamical system: how regular motion primitives may emerge from self-organized limit cycles. *Front. Robot. AI* 2:31. doi: 10.3389/frobt.2015.00031

Schmidt, N. M., Hoffmann, M., Nakajima, K., and Pfeifer, R. (2013). Bootstrapping perception using information theory: case study in a quadruped robot running on different grounds. *Adv. Comp. Syst.* 16:1250078. doi: 10.1142/S0219525912500786

Shim, Y., and Husbands, P. (2012). Chaotic exploration and learning of locomotion behaviors. *Neural Comput.* 24, 2185–2222. doi: 10.1162/NECO_a_00313

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. Berlin; Heidelberg: Springer Science & Business Media.

Tani, J., and Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* 33, 481–488. doi: 10.1109/TSMCA.2003.809171

Tetzlaff, C., Dasgupta, S., Kulvicius, T., and Wörgötter, F. (2014). The use of hebbian cell assemblies for nonlinear computation. *Sci. Rep.* 5:12866. doi: 10.1038/srep12866

Toutounji, H., and Pasemann, F. (2014). Behavior control in the sensorimotor loop with short-term synaptic dynamics induced by self-regulating neurons. *Front. Neurorobot.* 8:19. doi: 10.3389/fnbot.2014.00019

Toutounji, H., and Pasemann, F. (2016). "Autonomous learning needs a second environmental feedback loop," in *Computational Intelligence*, eds K. Madani, A. Dourado, A. Rosa, J. Filipe, and J. Kacprzyk (Cham: Springer), 455–472.

Wang, X. R., Miller, J. M., Lizier, J. T., Prokopenko, M., and Rossi, L. F. (2012). Quantifying and tracing information cascades in swarms. *PLoS ONE* 7:e40084. doi: 10.1371/journal.pone.0040084

Wittmeier, S., Gaschler, A., Jäntsch, M., Dalamagkidis, K., and Knoll, A. (2012). "Calibration of a physics-based model of an anthropomimetic robot using evolution strategies," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (New York, NY: IEEE), 445–450. doi: 10.1109/IROS.2012.6385591

Yamane, K., and Nakamura, Y. (2011). "Robot kinematics and dynamics for modeling the human body," in *Robotics Research: The 13th International Symposium ISRR*, eds M. Kaneko and Y. Nakamura (Berlin; Heidelberg: Springer), 49–60. doi: 10.1007/978-3-642-14743-2_5

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Der and Martius. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 02 March 2017
doi: 10.3389/fnbot.2017.00009

A Neural Dynamic Architecture for Reaching and Grasping Integrates Perception and Movement Generation and Enables On-Line Updating

*Guido Knips[1], Stephan K. U. Zibner[1], Hendrik Reimann[2] and Gregor Schöner[1]**

[1] Institute for Neural Computation, Ruhr-University Bochum, Bochum, Germany, [2] Department of Kinesiology, Temple University, Philadelphia, PA, USA

Reaching for objects and grasping them is a fundamental skill for any autonomous robot that interacts with its environment. Although this skill seems trivial to adults, who effortlessly pick up even objects they have never seen before, it is hard for other animals, for human infants, and for most autonomous robots. Any time during movement preparation and execution, human reaching movement are updated if the visual scene changes (with a delay of about 100 ms). The capability for online updating highlights how tightly perception, movement planning, and movement generation are integrated in humans. Here, we report on an effort to reproduce this tight integration in a neural dynamic process model of reaching and grasping that covers the complete path from visual perception to movement generation within a unified modeling framework, Dynamic Field Theory. All requisite processes are realized as time-continuous dynamical systems that model the evolution in time of neural population activation. Population level neural processes bring about the attentional selection of objects, the estimation of object shape and pose, and the mapping of pose parameters to suitable movement parameters. Once a target object has been selected, its pose parameters couple into the neural dynamics of movement generation so that changes of pose are propagated through the architecture to update the performed movement online. Implementing the neural architecture on an anthropomorphic robot arm equipped with a Kinect sensor, we evaluate the model by grasping wooden objects. Their size, shape, and pose are estimated from a neural model of scene perception that is based on feature fields. The sequential organization of a reach and grasp act emerges from a sequence of dynamic instabilities within a neural dynamics of behavioral organization, that effectively switches the neural controllers from one phase of the action to the next. Trajectory formation itself is driven by a dynamical systems version of the potential field approach. We highlight the emergent capacity for online updating by showing that a shift or rotation of the object during the reaching phase leads to the online adaptation of the movement plan and successful completion of the grasp.

Keywords: neural dynamics, dynamic field theory, autonomous reaching, autonomous grasping, online updating

OPEN ACCESS

Edited by:
Christian Tetzlaff,
Max Planck Institute for Dynamics and Self Organization (MPG), Germany

Reviewed by:
Florentin Wörgötter,
University of Göttingen, Germany
Thomas Wennekers,
Plymouth University, UK

***Correspondence:**
Gregor Schöner
gregor.schoener@ini.rub.de

Received: *02 November 2017*
Accepted: *14 February 2017*
Published: *02 March 2017*

Citation:
Knips G, Zibner SKU, Reimann H and Schöner G (2017) A Neural Dynamic Architecture for Reaching and Grasping Integrates Perception and Movement Generation and Enables On-Line Updating. Front. Neurorobot. 11:9. doi: 10.3389/fnbot.2017.00009

1. INTRODUCTION

Object-oriented reaching and grasping in natural settings, a key element of human-robot cooperation, continues to be a challenge for autonomous robots (Herzog et al., 2012). Humans grasp and handle objects fluently, of course, although these are among the harder movement tasks, learned in infancy (Thelen et al., 1996), but with continued development for close to 10 years of life (Schneiberg et al., 2002). Humans easily reach and grasp objects that they see for the first time or that are partially occluded. They may grasp an object after closing their eyes. Anytime during movement preparation or execution, humans may update the motor plan when the object shifts or rotates (Desmurget and Grafton, 2000). This performance entails, in humans, a close coupling among perceptual processes including gaze control, shift of attention, segmentation, recognition, and pose estimation of the object, as well as between perception and motor processes including initiating, coordinating, and terminating reach and grasp movements.

Robotic approaches to grasping (reviewed in Carbone, 2013) have traditionally made strong demands on what perception delivers, often based on object models. Except for visual servoing, those approaches are most appropriate for static situations with well-known objects. In contrast, recent work has employed simpler perceptual processes, that deliver fast estimates of pose and grasp parameters and enable grasping objects that move with a conveyer belt (Cowley et al., 2013). Another recent line of work learns to extract grasp parameters that are linked to probabilistic models that enable generalization beyond the trained poses, and lead to most impressive real time grasping performance (Huang et al., 2013). Related work learns grasp primitives from demonstration (Herzog et al., 2014), from exhaustive simulation (Curtis and Xiao, 2008), from examples of object categories (Madry et al., 2012), or based on tactile feedback (Platt et al., 2006). Explicit modeling of the uncertainty of grasp parameters provides a potential solution (Li et al., 2016).

This paper is based on two hypotheses. First, we think that we may learn from how humans generate reaching and grasping movements. For instance, as a major theme that we address here, we believe that reaching and grasping is possible in humans with much simpler, lower-level perceptual representations than traditionally assumed in autonomous robotics. The perceptual processes engage attention and enable continuous online coupling to the sensory surface. Another example is at the level of control: The nature of actuation through muscles that act as relatively soft, tunable strings makes it possible to grasp without a precise estimate of grasp points. It is enough to set the equilibrium length of muscles in the hand to a posture inside the object and the muscles will then generate grip forces through their peripheral reflex loops (Santello et al., 2016). In this paper, we address the first, but not yet the second idea.

The other hypothesis is, in a sense, the converse. Many of the neural processes underlying human movement that is directed at objects have not yet been comprehensively understood in neuroscience (Andersen and Cui, 2009; Lisman, 2015). This means that neurally based process models do not stand ready to be imported into robotics. But this also means that how the component processes work together in the nervous system needs to be better understood. Integrated models demonstrate reaching and grasping in neurally grounded ways that may make a contribution to understanding neural function.

Our research agenda is thus to build an integrated model of reaching and grasping based on neural process accounts inspired by the human mind. We do this based on the theoretical framework of Dynamic Field Theory (DFT, see Schöner, 2008 for an introduction, Schöner et al., 2015 for a systematic tutorial), a neurally grounded set of concepts that address visual representations, coordinate transforms, attentive selection, working memory, and behavioral organization. To build and implement a complete model of reaching for and grasping novel objects, we propose a neurally inspired computational architecture.

All processes are modeled as neural dynamics, so that the entire architecture is essentially one big dynamical system. The theoretical framework of Dynamic Field Theory (DFT) provides the means to represent information, to perform detection and selection decisions, to model attention, track time varying input, and to store information in working memory. Instabilities of the neural dynamics create the discrete events from time-continuous processes at which processes are initiated and terminated (Sandamirskaya et al., 2013). The neural dynamics interfaces with attractor dynamics that generate movements and control the robotic arm and hand (Reimann et al., 2011). The model builds on earlier work on scene representation (Zibner et al., 2011a), and on the simultaneous recognition of objects and estimation of their pose (Faubel and Schöner, 2009). We show how neural dynamics enable integrating and organizing all component processes, from the perception to the initiation and termination of robotic movements (Richter et al., 2012).

The approach is tested on a robotic agent called CAREN consisting of a Kuka LWR4 with seven degrees of freedom, with an attached Schunk Dextrous Hand (SDH) featuring additional seven degrees of freedom and tactile sensors. The arm is mounted on a Schunk PR 90 rotary module with one degree of freedom. We are using a Kinect camera to perceive the scene (see **Figure 1**).

This work is innovative in two different ways. On the one hand, this work is part of a research program in which robotic demonstrations are used to evaluate theoretical models of human cognition and behavior (Adams et al., 2000). Neural dynamics is a theoretical perspective within this program in which process models are formulated that may be linked to real sensory and motor systems (Erlhagen and Bicho, 2006). Previously, neural dynamics has been used to demonstrate reaching (Strauss and Heinke, 2012; Fard et al., 2015; Strauss et al., 2015). We expand on this work by including the autonomous sequential organization of the behavior and addressing grasping as well. Ours is one of the first demonstrations that cover the complete path from sensing to acting in a difficult task, that includes attention, recognition, estimation, executive control, movement planning, and control. In this demonstration, we integrate four separate neural dynamics models of component processes for scene representation (Zibner et al., 2011a), object classification with concurrent pose estimation (Faubel and Schöner, 2009),

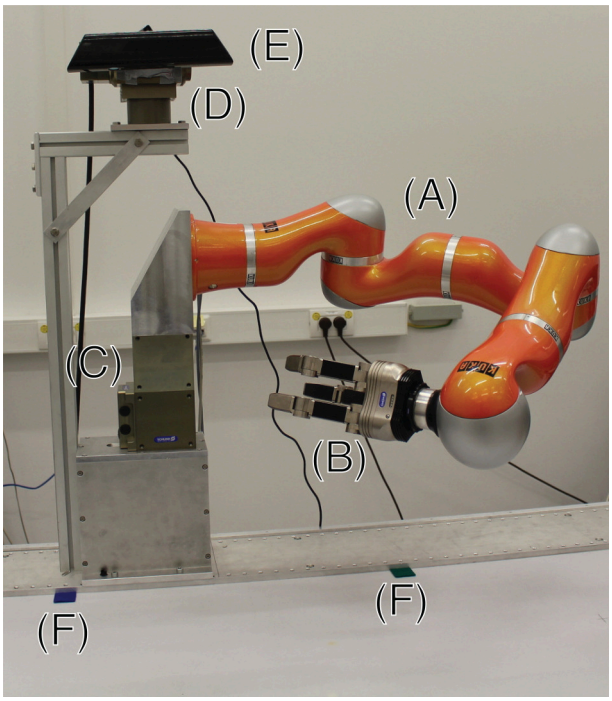


FIGURE 1 | This figure shows CAREN, the robotic platform used in this work. It consists of a Kuka LWR4 (A), a Schunk SDH (B), and two Schunk rotary modules, one used as trunk (C), the other as pan-tilt head (D). A Kinect (E) is attached to the head of the robot. Two markers (green and blue) are placed on the table surface (F). They denote the *x*-axis of the table's coordinate system.

behavioral organization (Richter et al., 2012), and movement generation (Reimann et al., 2011).

On the other hand, in direct comparison to approaches to grasping that are unconstrained by analogies with human cognition, the strength of the present work is the capacity to accomodate online updating to changing sensory information, while at the same time addressing the sequential organization of behavior and perception. For instance, work like Huang et al. (2013) has powerful online updating of the grasping action itself, but has a highly simplified perceptual system and limited behavioral flexibility. We think of online updating as a characteristic and attractive property of the neural organization of reaching and grasping and this is why we focus on demonstrating it here.

2. METHODS

We begin by providing a survey over the component processes involved in autonomous grasping and the over-all flow of activation in the neural dynamics architecture (**Figure 2**). Perception (on the left) consists of scene representation and object recognition. Scene representation entails the processes of visual exploration, which sequentially attends to subregions of the scene that may contain objects and commits an estimate of local height at each attended location to working memory. Visual exploration is a precondition of the query behavior, which processes a cue that defines a target object, brings matching locations into the attentional foreground and thus enables the process of object recognition to take over. Object recognition entails two interacting processes, shape classification, and pose estimation. Shape classification determines the type of grasp that will be used for the current target object, while pose estimation specifies parameters of the reach and the grasp such as hand orientation. Once both processes have converged, a sequence of actions executes the grasp (illustrated on the right). Initially, two behaviors are activated: "Open hand" does what the name suggests and "approach" drives the hand to a point close to the target object while orienting the hand based on a pose estimate. After both behaviors are completed, the "grasp" behavior moves the fingers. Up to that point, online updating of the classification and pose estimation processes is possible, after this point, online updating is suppressed. After detecting contact of the hand on the object's surface through tactile feedback, the "lift" behavior is activated, which raises the arm with the grasped object upwards from the table surface.

Although this description suggests that the individual behaviors and processes are separate modules, in reality they are all just subsets of one large system of differential and integro-differential equations, the neural dynamics, whose solutions evolve continuously in time. These equations are coupled internally according to the architecture and to online sensory inputs. Online updating is thus a pervasive property of the architecture and neural dynamics approaches, in general. We now take a closer look at the elementary building blocks of the architecture to illustrate how neural dynamics and, specifically, DFT, are organize the interaction of the behaviors and processes.

2.1. Dynamic Neural Fields

Dynamic neural fields are the building blocks of Dynamic Field Theory (DFT). Continuous neural activation patterns, $u(x,t)$, defined over a feature dimension, x, evolve in time according to an integro-differential equation that has been proposed as a simplified model of cortical neural dynamics (Amari, 1977):

$$\tau \dot{u}(x,t) = -u(x,t) + h + s(x,t) + \int w(x-x')\sigma(u(x',t))dx'.$$

Here, τ determines the time scale on which activation evolves. The $-u$-term endows this neural dynamics with the fundamental stability mechanism that creates different kinds of attractor solutions under different conditions. The attractor at the resting level, $h < 0$, is stable in the absence of external input, $s(x,t)$. When such input from other neural fields or from sensory surfaces remains small, the attractor is shifted to $h + s(x,t)$. When inputs become sufficiently strong so that this solution reaches a threshold given by the sigmoidal nonlinearity, $\sigma(\cdot) = 1/(1+\exp(-\beta\cdot))$, this attractor becomes unstable. The system switches to a new attractor state, a localized peak of activation that is sustained by local excitatory and global inhibitory interaction characterized by the interaction kernel, $w(\Delta x)$. The instability at which a switch to such a self-stabilized peak solution occurs is the *detection instability*, used to implement detection decisions

in DFT. Localized peaks become unstable at the *reverse detection instability* at lower levels of input. Multi-modal inputs may lead to the formation of a self-stabilized peak at a single location in the field. This is how *selection decisions* are realized in DFT. Under appropriate conditions (for resting level and interaction strength), self-stabilized peaks may remain stable once the inducing localized input, $s(x, t)$, is removed. Dynamic fields may be analogously defined over multi-dimensional spaces. Such sustained peaks of activation are the model of working memory in DFT. See Schöner et al. (2015) for a systematic exposition of the mathematical and conceptual structure of DFT. The stability regimes described here depend, of course, on parameter values. Typical values of the main parameters of the neural field dynamics used throughout the architecture are: $\tau = 100\,ms$, $\beta = 100$, h between -15 and -5, global inhibition between 0.01 and 0.5, excitatory interaction 1, width of exitatory interaction kernel between 3 and 5.

2.2. Neural Dynamics of Behavioral and Process Organization

Zero-dimensional neural activation fields are essentially discrete activation nodes described by a differential equation analogous to Equation 1:

$$\tau \dot{u}(t) = -u(t) + h + s(t) + w\sigma(u(t))$$

This dynamics may have an "off" attractor at negative levels of activation, and an "on" attractor at positive levels of activation. The "off" attractor may disappear in a detection instability at sufficiently high levels of input, s. The "on" attractor may disappear in a reverse detection instability at sufficiently low levels of input, s. Both attractors may co-exist bistably for intermediate levels of input. Such nodes are used in DFT to represent the activation and deactivation of categories, processes, or behaviors. For the organization of processes and behaviors, pairs of such activation nodes form an executive control unit (ECU, see Richter et al., 2012). When the *intention* node of an ECU is "on," it provides spatially homogenous excitatory input (a "boost") to parts of the architecture that is responsible for executing an associated process or behavior. The *Condition of Satisfaction* (CoS) node is activated when sensory or internal inputs are detected that indicate the completion of a process or behavior. CoS nodes inhibit the intention node, turn "off" the associated process or behavior. A third node may be joined to an ECU to represent a working memory of CoS activation, which maintains a record of the past completion of a processing step. Typical values of the parameters of the neural dynamic nodes used throughout the architecture are: $\tau = 100\,ms$, $\beta = 100$, h between -1 and -2, global inhibition 0.01.

2.3. Visual Processing Pathway

The autonomous neural dynamics of visual processing controls exploratory attentional processes that build a working memory representation of the scene, which can be queried to activate a particular target object. A second block of processes determines object identity through classification and estimates object pose to determined grasp parameters.

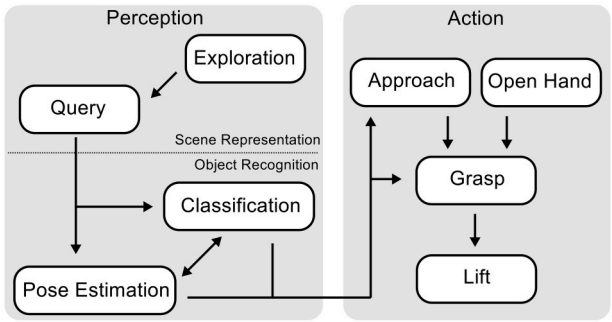


FIGURE 2 | Schematic overview over the behaviors that make up the reaching and grasping architecture and how they interact.

2.3.1. Scene Representation

The architecture contains an expanded version of a neural dynamic system for scene representation (Zibner et al., 2011a), in which neural dynamic nodes implement a form of process organization (Richter et al., 2012) to enable the autonomous visual exploration of the scene which can transition into a query mode that focusses attention on a target object in the scene. **Figure 3** expands this part of the complete architecture. As a cue to locations on a table surface, at which objects may be placed, we use color and visual depth estimates obtained from a Kinect sensor that views the scene in the work space of the robot arm. The idea is that color saturation on the homogeneous table surface guides attention to candidate locations. The height over the table surface estimated at these locations is then used to decide if an object is present (Petsch and Burschka, 2010).

Specifically, we use the Point Cloud Library (Rusu and Cousins, 2011), to find the largest surface in the RGB-D data, which is then identified as the table surface. Height and color maps are extracted in world coordinates. The distribution of saturation in the color map is passed through a sigmoid function and provides input to a neural field defined over the table surface (the space field in the green box of **Figure 3**). Only regions on the table at which saturation reaches a threshold level drive the neural field through a detection instability and induce a self-stabilized local peak of activation. This effectively suppresses outliers and filters out the noise that is typical of RGB-D data. The field is operated in a dynamic regime in which multiple self-stabilized peaks may coexist. It functions as a salience map for color (Itti et al., 1998).

The color salience space field provides input to a second neural field, the *attention field*, also defined over the table surface. This field is operated in the dynamic regime in which a single localized peak is stable at any time, implementing a selection decision. A self-stabilized peak in this field implements, therefore, selective attention and provides the attentional focus for the rest of the architecture. Height estimates from the subregion on the table, at which activation in the attention field is above threshold, are input into a one-dimensional neural field, the *height field*. The selected spatial region and the neural activation pattern representing height estimates are crossed to provide input into a three-dimensional field, the *space-height field* (on the top right

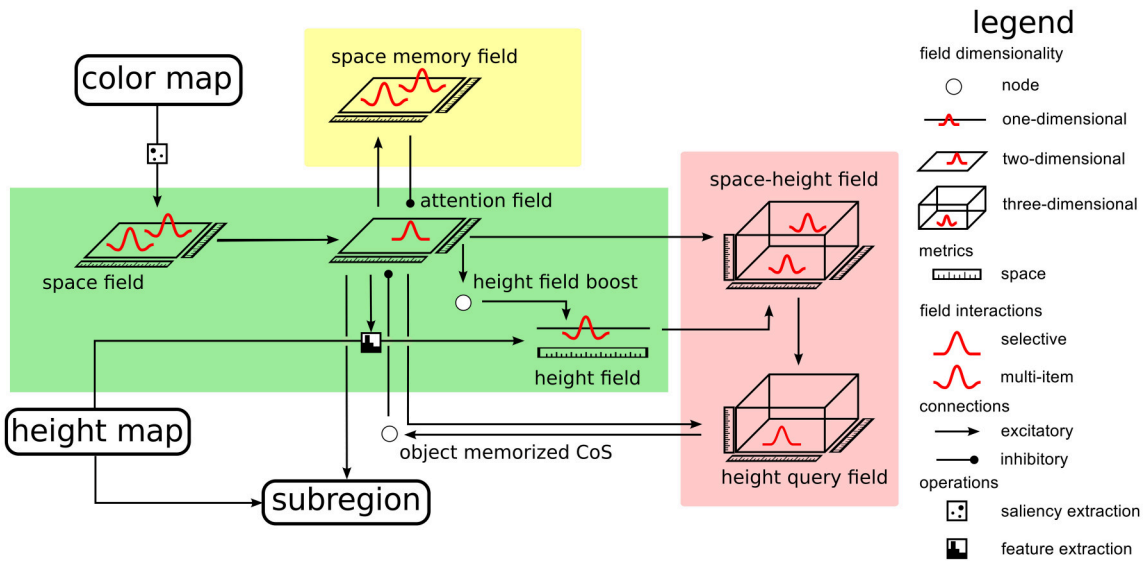


FIGURE 3 | The portion of the architecture responsible for scene representation.

in the red box of **Figure 3**). For details of how the combination of two lower-dimensional inputs can be used to drive a higher-dimensional field, please refer to Zibner et al. (2011a) or Chapter 9 of Schöner et al. (2015). The space-height field is operated in multi-peak working memory mode, so that it represents the location on the table and height of a potential object as a self-sustained peak, even after the attentional and height inputs are removed. It provides input to a second, three-dimensional field, the *height query field*, that is operated in single-peak mode and thus selects location and the associated height. Input from the attention field controls the location at which input from the space-height field may induce a peak. The height query field thus serves to retrieve a stored object location and height from scene memory.

To guide visual exploration, a multi-peak field over the table surface, the *space memory field*, keeps track of all locations that have come into the attentional focus of the system. A sustained peak of activation is induced each time selective attention is focussed at a location. The space memory field in turns inhibits the attention field and thus biases the process of attentional selection away from locations that have previously been the focus of attention. Autonomous exploration is now organized by a Condition of Satisfaction connection from the height query field into the attention field. Every time a peak has been successfully selected in the height query field, this signals that a memory has been created that matches the currently selected location and currently estimated height. This is the CoS of memory formation and inhibits the attention field, deleting the self-stabilized peak there in reverse detection instability. As a result, the peak in the height query field is no longer supported by selective attention and also decays, releasing the attention field from inhibition. The attention field is ready to select the next location for spatial attention. Inhibitory input from the space memory field now tends to inhibit return to the same location or other recently attended locations, biasing the selection process to new locations with salient color input. This process of visual exploration is continuously ongoing, confirming past memories in the space-height field, updating such memories or creating new such memories as needed.

Autonomous visual exploration can be interrupted at any time by a query for a target object, that triggers the estimation of grasp parameters. The target object can be specified by a spatial cue or by cues of characteristic object features, such as color (for a more detailed description of the querying behavior, see Zibner et al., 2011b). There is a set of neural nodes that activate and deactivate parts of the architecture by boosting or deboosting the resting levels of the associated fields. Not all of those nodes are plotted in the survey over the architecture for simplicity (see a description in the first part of the Results Section for the functional role of these nodes).

2.3.2. Shape Classification and Pose Estimation

Estimation of grasp parameters is based on a recurrent architecture for object recognition (Faubel and Schöner, 2009). In the original work, a weighted sum of object templates, one for each known object, is compared to the current input image. Applying cascaded transformation operations of shift, rotation, and scaling) to the current input and matching the transformed input to each of the memorized templates (by cross-correlation, "C") yields a competitive weight of each template. Dynamic neural nodes compete with each other, leading to the selection of the template in a classification decisions. In a concurrent process, all templates are weighted with the current activation level of their dynamic neural node and summed. This inverse cascade of image transformations is applied and a match to the input image in each possible pose provides input into neural activation fields defined over the pose parameters for shift, rotation, and scaling. These fields are operated in a single-peak mode so

that an emerging self-stabilized peak represents a selection decision among poses. The concurrent upward classification, and downward pose estimation processes converge in closed loop, activating an object identity representation in the set of neural nodes, and a pose estimate in the set of neural fields.

For the present purpose, we replace learned object templates with simple geometric shapes (square, circle, oblong rectangles). The subregion on the table that the attentional focus defines provides the visual input to the shape classification and pose estimation system. The two-layer decision architecture of the original model was further simplified into single layer decision fields connected each to a single inhibitory node that slows down the decision process, allowing multiple candidate peaks to form before a decision emerges. **Figure 4** gives an overview of the resulting architecture. The different stages of pose estimation are highlighted by the background color: translation (red), rotation (yellow), and scaling (green). The set of neural nodes that makes shape classification is highlighted in blue.

As the shape classification and poste estimation process converges, it delivers a shape candidate whose location is specification more precisely within the table surface than the attentional systems does. The scaling and rotation estimates together with features of the shape category are used to determined the grasp parameters, represented in the *grasp decision field* Oblong objects with a low height are grasped from above, while cylindrical objects and cuboids with a square base with sufficient height are grasped from the side. The latter objects need different approach movements prior to grasping, since cylinders, unlike cuboids, can be grasped sideways equally well from any direction.

Note, that the estimation process is continuously coupled to visual input through the attentional channel. As a result, changes in the scene are fed into the pose fields enabling online updating of the grasp parameters. In the current version of the model, online updating occurs only with respect to two dimensions of the task, translating, and rotating the gripper.

2.4. Reaching and Grasping

This section explains how data from the scene representation and the shape classification/pose estimation systems are used to

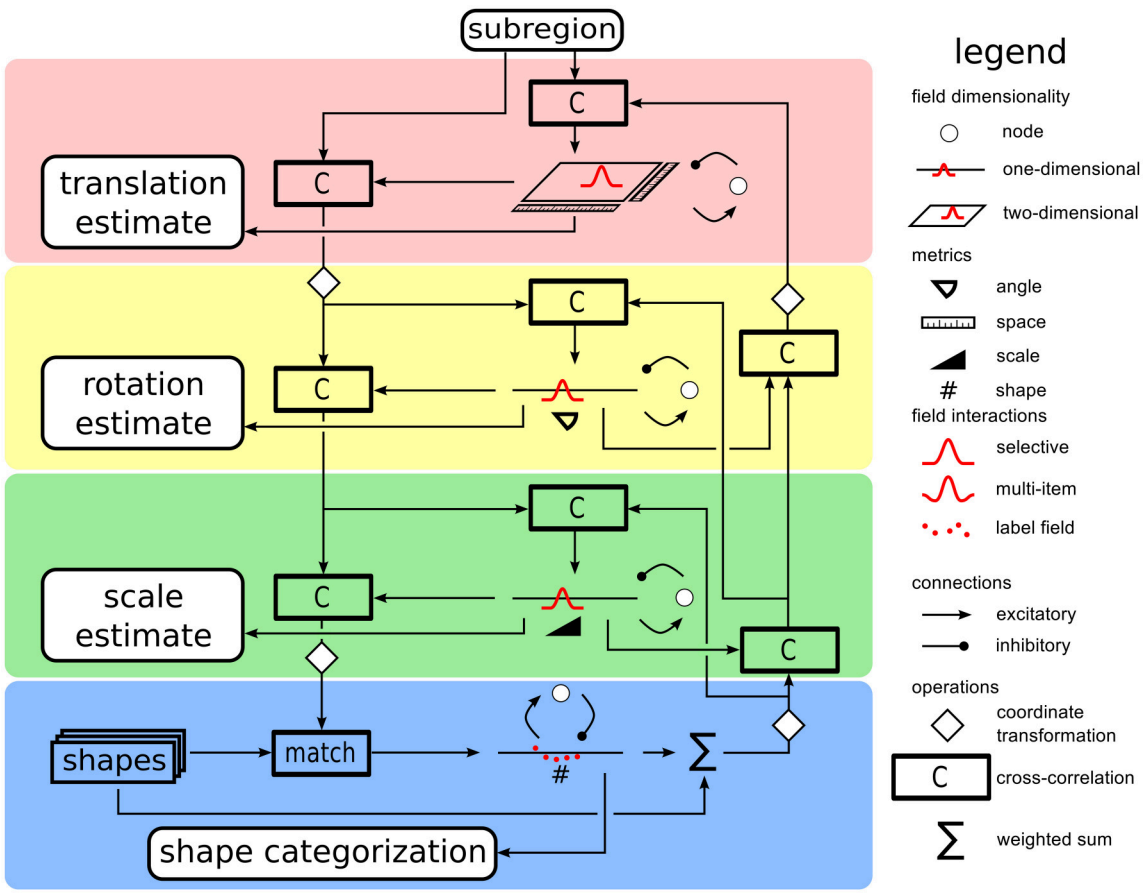


FIGURE 4 | A sketch of the shape/pose estimation system used to classifify the attended part of the visual scene into a shape category and to concurrently estimate its pose. Along the downward pathway on the left, the input image is transformed based on the current estimates of translation, rotation, and scaling before being compared to the stored shape templates at the bottom. Along the upward path on the right, the current weighted sum of shape templates is inversely transformed by scaling and rotation operations. Cross-correlations with the input image yield updates to pose estimates. The pose fields in the center column feed into the representation of grasp parameters.

generate movement and to grasp an object. The overall scheme is as follows. Depending on the object pose parameters (position, height, rotation, and shape) and the current arm configuration, a desired wrist position and orientation for the hand are computed. These desired values are then set as attractors in a dynamical system that generates movement for the arm. The movement unfolds autonomously in three phases organized by a neural dynamics of the type reviewed earlier (Section 2.2). First, the hand is opened, brought close to the object, and oriented in a way that enables grasping the object. Second, the hand is moved through the remaining distance to the object, and is closed. The third phase begins when the object has been grasped as signaled by the tactile sensors on the fingers. The hand is then moved upward in space, lifting the object. This sequence of actions is generated by a neural dynamics of behavioral organization that is illustrated in **Figure 5**.

2.4.1. Generating Motor Commands

Motor commands are generated from desired values for the wrist position and hand orientation using the attractor dynamic approach (Reimann et al., 2011). To move the wrist, movement speed, and direction are controlled separately. The rate of change of movement direction depends on the angle between the current movement velocity, $\vec{v}$, and the vector, $\vec{k}$, from the wrist position to the target position,

$$\phi = \arccos\left(\frac{\left(\vec{v}, \vec{k}\right)}{|\vec{v}||\vec{k}|}\right). \tag{1}$$

Reducing this angle to zero corresponds to changing the movement direction into the direction in which the target lies. This constraint is imposed by the dynamics of that angle, given by

$$\dot{\phi} = -\alpha_{\text{dir}}\phi, \tag{2}$$

which is linear, simplifying Reimann et al. (2011). Here, α_{dir} is a rate factor.

To translate this constraint into a motion command for the robotic arm, consider the direction, $\vec{v}_{\perp}$, in which the movement vector, $\vec{v}$, is changed. It is perpendicular to $\vec{v}$ and lies in the plane spanned by $\vec{v}$ and $\vec{k}$. Computed as:

$$\vec{v}_{\perp} = |\vec{v}|\frac{(\vec{k} \times \vec{v}) \times v}{|(\vec{k} \times \vec{v}) \times v|}. \tag{3}$$

and normalized to have the same length as $\vec{v}$.

Combining the two equations we determine the direction in which the wrist's velocity vector in cartesian space should change so as to bring the hand closer to the target location:

$$\vec{f}_{\text{dir}} = \vec{v}_{\perp}(\dot{\phi} - \dot{\phi}_{\text{dev}}). \tag{4}$$

Here, $\dot{\phi}_{\text{dev}}$ is the rate at which the direction from the hand to the target changes due to the movement, $\vec{v}$, of the hand in space. The direction of change lies in the appropriate plane and is proportional to the rate of change of the direction to the target corrected for the rate of change of that direction that is induced by the movement of the wrist in space.

To control movement speed, its rate of change, $\dot{v}$, is proportional to the difference between the current speed, $v = |\vec{v}|$, and a desired speed v_{des}:

$$\vec{f}_{\text{vel}} = \frac{\vec{v}}{v}(-\alpha_{\text{vel}}(v - v_{\text{des}})) \tag{5}$$

where α_{vel} is a rate constant. As a contribution to the rate of change of the 3D velocity vector, this contribution lies in the direction of the current velocity.

A third contribution to the dynamics of the hand velocity vector slows down the hand when it is close to the target object in order to reduce any impact in case of misestimation and collision. A local safe control law is proportional to the distance between hand position, $\vec{g}$, and target position, $\vec{p}$:

$$v_{\text{local}} = -\beta_{\text{pos}}(\vec{g} - \vec{p}), \tag{6}$$

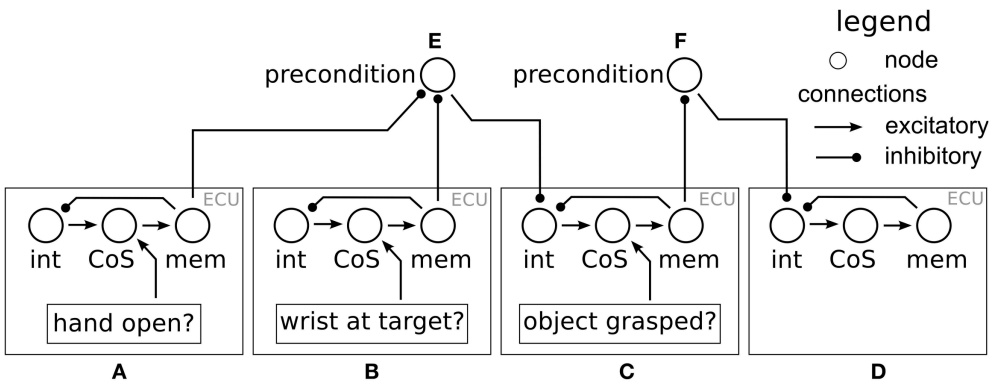


FIGURE 5 | The neural dynamics of behavioral organization used for movement generation. There are four ECUs: open hand **(A)**, approach target **(B)**, grasp target **(C)**, and lift **(D)**. The precondition node **(E)** ensures that the grasp behavior is only activated once **(A,B)** have met their CoS. The precondition node **(F)** ensures that the grasp behavior has met its CoS before the object is lifted.

and is expanded in vector form as

$$\vec{f}_{\text{pos}} = -\alpha_{\text{pos}}(\vec{v} - \min\{|v_{\text{local}}|, v_{\text{des}}\}\frac{v_{\text{local}}}{|v_{\text{local}}|}), \quad (7)$$

where α_{pos} and β_{pos} are two rate factors. The introduction of v_{des} is a change over the approach of Reimann et al. (2011) intended as a safety measure to delimit movement speeds of the arm.

The rates of change of the hand's velocity vector in Cartesian space are transformed into joint space with the help of the pseudo-inverse, J_p^+, of the Jacobian matrix of the wrist position. The three contributions are then summed after each contribution is weighted with a sigmoidal factor that reflects the distance of the hand to the target. The result is the planned angular acceleration, $\vec{F}$, of the robotic arm in joint space:

$$\begin{aligned}\vec{F} &= \sigma(|\vec{k}| - d_{\text{thr}})(J_p^+ \cdot \vec{f_{\text{dir}}} + J_p^+ \cdot \vec{f_{\text{vel}}}) \\ &\quad + (1 - \sigma(|\vec{k}| - d_{\text{thr}}))J_p^+ \cdot \vec{v_{\text{vel}}}. \end{aligned} \quad (8)$$

This control strategy for the hand's position largely follows Reimann et al. (2011). The control law of the hand's orientation is formulated for the three Euler angles of the hand used as target angles for the three most distal joints of the arm. The desired rotation matrix R can thus be split into three subsequent rotations around three fixed axes. For each of these three most distal joints, θ_i, the angular acceleration $\ddot{\theta}_i$ is proportional to the deviation between the current joint angle, θ_i, and the desired joint angle, $\theta_{i,\text{des}}$, corrected for by the current angular velocity, v_{θ_i}, induced by the movement of the hand in space according to Equation 9:

$$\ddot{\theta}_i = -\alpha_{\text{rot}}(v_{\theta_i} - \beta_{\text{rot}}(\theta_i - \theta_{i,\text{des}})). \quad (9)$$

Here, α_{rot} and β_{rot} are rate constants of the dynamics.

Finally, the opening and closing of the hand is controlled through a linear first order dynamical system:

$$\dot{\vec{\theta}} = -\alpha_{\text{hand}}(\vec{\theta} - (w_{\text{grasp}}\vec{\theta}_{\text{closed}} + w_{\text{approach}}\vec{\theta}_{\text{open}})). \quad (10)$$

This dynamical system has attractors either at a joint angle configuration, $\vec{\theta}_{\text{open}}$, corresponding to an open hand or at a joint angle configuration, $\vec{\theta}_{\text{closed}}$, corresponding to a closed hand. These joint configurations depend on the shape template of the object to be grasped.

2.4.2. Target Positions and Orientations

Desired positions, g, for the wrist are defined for the approach, grasp and lift behaviors, as well as for different grasp types. All approach points for the different object types are updated online. The target point, $\vec{g}_{\text{approach}}$, for the approach behavior depends on the grasp type. For vertical objects, it lies in a horizontal plane at two thirds of the object's height at a certain distance from the object that depends on the object's shape. For cylindrical objects, the vector, $\vec{k}$, from the current wrist position to the object position is projected onto the table plane to obtain the direction from which to grasp. For objects with a square base, one of the four sides is selected. This entails computing the inner product of $\vec{k}$ with each of four vectors that are orthogonal to each side. Using four competing neural nodes, the vector that best matches is selected. For objects that are grasped from above, the approach point is at a fixed distance above the object. A weighted sum

$$\vec{g}_{\text{approach}} = \frac{1}{n}\sum_i w_i\vec{g}_i, \quad (11)$$

over the n different object types is used to calculate the instantaneous approach point. The values for w_i are the output values of the *grasp decision field*.

For the target point of the grasping behavior, we use a point on the object vector, $\vec{k}$, at a certain distance, d_i, from the object

$$\vec{g}_{\text{grasp}} = \frac{1}{n}\sum_i w_i d_i \frac{-\vec{k}}{|\vec{k}|}. \quad (12)$$

To lift the object, a position, $\vec{g}_{\text{lift}}$ is set to a point 50 cm above the table surface located directly above the current position.

The current target position for the movement generation system is then set to the weighted sum over all these different target points

$$\vec{g} = w_{\text{approach}}\,\vec{g}_{\text{approach}} + w_{\text{grasp}}\,\vec{g}_{\text{grasp}} + w_{\text{lift}}\,\vec{g}_{\text{lift}}, \quad (13)$$

in which the weight factors are the activation states, w_i, of the corresponding behavior.

The orientation of the hand at grasp is chosen so that the opening of the hand points toward the object and the fingers are aligned with the object's surfaces. For tall, narrow objects that are grasped from the side, the palm is chosen to be oriented perpendicular to the table surface. For flat objects that are grasped from above, the palm is oriented parallel to the table. Again a sum is used to obtain the desired orientation of the hand from these contributions, weighted with the activation level of the associated shape class.

3. RESULTS

A first goal of our experimental work is to illustrate how the neural dynamic architecture generates the time courses of visual exploration, shape classification and pose estimation, and movement generation. In each case, we aim to show how transitions between different phases of behavior emerge autonomously from the space time continuous dynamical systems. Although we inspect the three components of scene representation, shape classification, and movement generation, one by one, these componets are tightly coupled in the overall neural architecture and evolve in parallel. The second goal is to demonstrate and assess the properties of the neural architecture in achieving reaching and grasping actions. We report three sets of experiments that probe online updating with respect to three dimensions of the task (grasping, translating, rotating). In the following sections, we first give detailed account of the general flow of neural activation through the dynamic fields and nodes. Then we report the results of the three experiments set up to probe specific characteristics of the system.

3.1. Time Course of Scene Representation

As long as there is no active cue, the neural architecture of scene representation (**Figure 3**) performs visual exploration which can be described as follows. The distribution of color over the table surface is captured by the space field that forms one peak at each location with salient color. These peaks provide localized input to the attention field, which generates a single peak and inhibits all alternative locations. This peak masks input from the height map to the height field so that only height measurements within this window contribute. A neural node that detects a peak in the

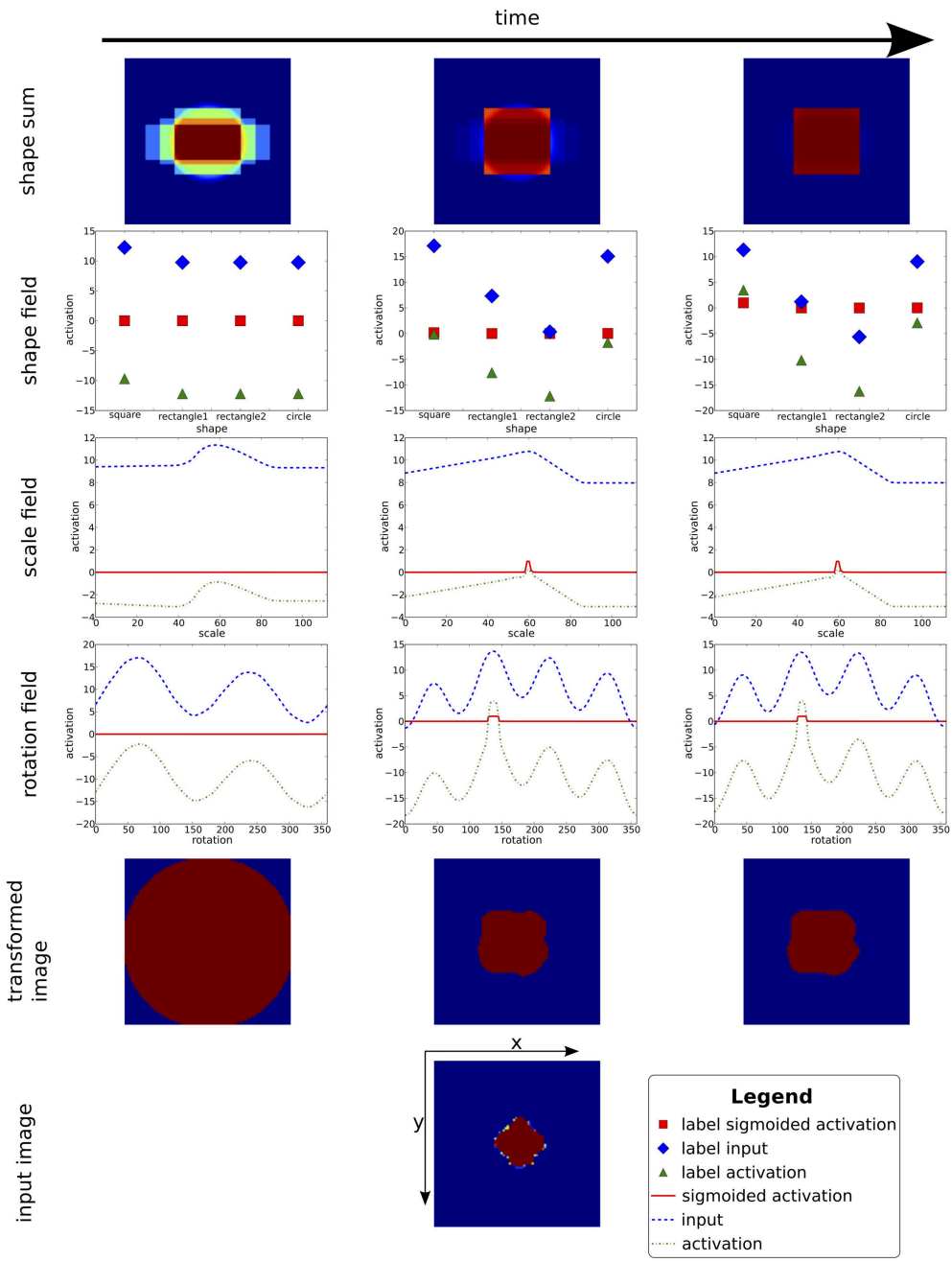


FIGURE 6 | This figure shows the time course of the convergence process of the pose estimation and shape classification processes. In each column, the current weighted sum of shapes is shown on top. Below, inputs and activation levels of the fields representing the shape weights, as well as scale and rotation estimates are displayed (the translation estimate is not shown since the attentional blob sufficiently centers the input image, trivializing this estimation). At the bottom, the raw and transformed input image is shown. At the beginning (time passes from left to right) the transformed input is blurred out and the estimation fields only contain sub-threshold activity. While the process converges, the estimation fields select pose candidates. With the fixed pose, the shape field converges onto a classification of the base shape. Note that this is a recurrent process, that is, pose estimates and shape classification converge in parallel and support each other.

attention field provides a boost to the height field, which together with significant input from the masked height input may induce a peak in this field.

The attention and height fields now contain separate representations of spatial position and height. Spatial input projects as a cylinder localized in space, elongated along height into the three-dimensional space-height field. Height input projects as a slice localized along color, extended along space. Where these inputs intersect, a localized peak arises that binds height to location. This peak induces localized input into the three-dimensional space-height query field, which receives at the same time a cylinder of input localized in space, extended along height from the peak in the attention field. These inputs overlap and create a matching peak in the height query field. A CoS node detects this peak and inhibits the attention field, triggering a cascade of reverse detections in the attention, height, and height query fields, followed by de-activation of the CoS node itself, and a release from inhibition of the attention field. Parallel to this cascade of instabilities, the looking memory field has stabilized a sustained peak at the currently attended location which projects inhibitorily back onto that same location in the attention field. Upon the release from inhibition from the CoS node, the attention field selects a new salient location for activation, that is not typically the same as the previously examined location.

This form of visual exploration runs continuously and completely autonomously, in an ongoing sequence of shifts of attention. This ongoing sequence is interrupted when a cue is given from the outside, for example, by a human operator. The cue resets the attention field through a short burst of inhibition and acts as a mask to the input path from the color map, amplifying the specified color. When the attention field recovers from inhibition, it now selects a location matching the cued color. This attentional peak induces activation from working memory of the height value associated with that location, which can now be handed on to the reach and grasp module.

3.2. Time Course of Shape Classification and Pose Estimation

With the activation of the cueing behavior, a peak in the attention field defines a window of attention, that channels input to the shape classification and pose estimation portion of the architecture (**Figure 4**). The CoS node of the cueing behavior provides a boost to the resting level boost of all estimation fields, which gets the estimation process started. The classification nodes are all equal and at resting level.

At the beginning of the process (see left column of **Figure 6**), the sum of shape templates in the top-down path is a homogeneous mixture of every known shape. Since all shapes are stored in a centered fashion, even this sum provide a cue to translation estimates. Over time, the pose estimation fields build up peaks, which compete within the fields for selection. As these estimates sharpen, the cross-correlations at every stage of pose transformation produce increasingly precise input to the pose fields. The match between the transformed input image and the stored shapes improves at the same time (middle column of **Figure 6**). The pose estimates converge somewhat earlier than the neural nodes that make shape selection, which operate on a slightly slower time scale (right column of **Figure 6**). At this point both the top-down pathway as well as the bottom-up pathway are fully converged onto candidate estimates, but are still reactive to changes in the input (e.g., caused by rotating or shifting the target object). Both bottom-up and top-down pathways participate in this bootstrap process.

3.3. Time Course of Movement Generation

Figure 7 illustrates the time line of the neural dynamics of behavioral organization of movement generation. Initially, none of the movement intention nodes is active, since no object has been recognized yet. When all fields of the pose estimation system have stabilized a peak, movement generation is initiated. The *approach* behavior and the *open hand* behavior become active at the same time and unfold in parallel. The *open hand* behavior terminates once the hand is open, while the *approach* behavior continues until the wrist of the arm has reached a certain target point and the hand is oriented correctly. The successful completion of either behavior is signaled through the respective CoS node. Once both CoS nodes become activated, the *grasp* behavior is activated. The arm moves the remaining distance to the object while the hand is closing. Pressure sensors in the fingers signal to the CoS node of the *grasp* behavior which is activated once a grasp is detected. The *grasp* intention node is deactivated by its CoS, and the *lift* behavior is activated. The series of snapshots of the robot arm during the reaching toward and grasping of an object is shown in **Figure 8**. This instance of reaching and grasping contains online updating as the object is moved and rotated by the experimenter after the movement has been initiated. We examine online updating next.

3.4. Three Experiments to Probe Online Updating

The task is to successfully reach for and grasp an object that is positioned on the table in front of the robot and then lift it up without losing grip, even if the object's pose is changed after the beginning of a trial. To assess performance, we count a grasp and lift as successful, if the object is lifted without losing grip. Failures include tipping over the object, closing the fingers

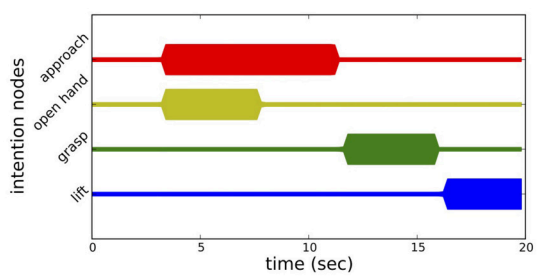


FIGURE 7 | Time course of the elementary processing units of movement generation. Each line represents the activation level of the intention node of a behavior through line thickness.

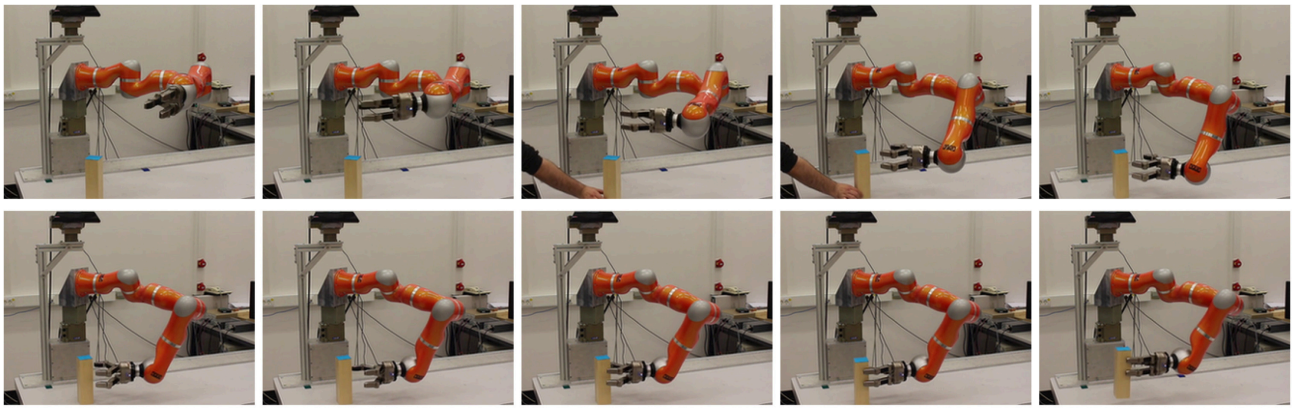


FIGURE 8 | This figure shows snapshots of a reaching and grasping trial. The third and fourth snapshot show a human intervening in the scene by moving and rotating the target object. Shortly after this intervention, the grasp approach adapts to the new pose leading to a successful grasp in the new pose followed by lifting up of the object.

without grasping, and not lifting the object. In addition, we also count trials as failed if the experimenters have to intervene with a safety stop due to singular arm configurations or any form of collision. In some cases, the grasp was executed successfully even without precise estimates (e.g., orientation estimate is off, base shape is not detected correctly). We count such trials as errors in classification.

For the experiments, we used a set of three simple wooden objects. The objects relate to the different grasps that the architecture is capable of executing: One cylinder and two cuboids, one with square base shape, the other with an oblong base shape (see **Figure 9** and **Table 1**). The object recognition system uses three geometric shapes that loosely fit the base shapes of the objects, that is, scale and aspect-ratio of the templates are close to those of the objects.

For practical reasons, the trunk degree of freedom of the robot was kept constant at 0° or -45° during all trials. This is a small number of trials to singular arm configurations, which our approach did not explicitly avoid. This limitation should be overcome in future implementations and illustrates how the trunk degree of freedom helps to cover a large workspace.

3.4.1. Grasping without Online Updating

In a first experiment, we placed a single object from the object pool onto the table in front of the robot. We picked five different positions, P_1–P_5 (see **Figure 10**) and multiple orientations for the square cuboid ($0^\circ, 30^\circ, 60^\circ$) and the oblong cuboid ($0^\circ, 45^\circ, 90^\circ, 135^\circ$). For the cylinder, we repeated each trial three times, for a total of 50 trials in experiment 1.

The performance of plain grasps without online updating is shown in **Table 2**. To minimize singular arm configurations, the sideways grasps were executed with the trunk joint at -45°, while the top grasps were executed with a trunk joint angle of 0°. Of the 50 trials, 46 were successful (92% success rate). Individual trials failed due to a singularity in the arm configuration (twice) or failed recovery from a lost peak in the estimation architecture (twice). **Table 2** also contains the classification rate in all successful trials. In three trials, the base shape of the object was not detected correctly, but nonetheless the object was grasped successfully.

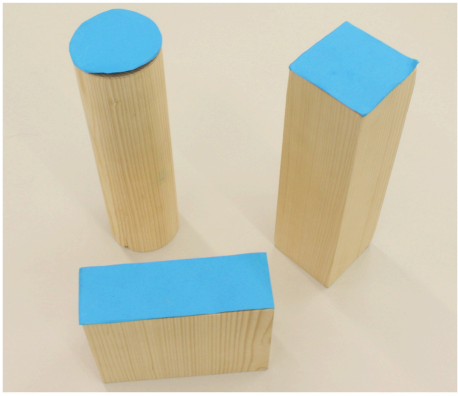


FIGURE 9 | The three simple wooden objects used in the experiments are shown. Each object is colored blue on its top surface. Blue was used as query cue to indicate the target object.

TABLE 1 | Object sizes.

Object	Size axis 1	Size axis 2	Height
Cylinder (cm)	7	7	25
Square (cm)	7	7	25
Rectangle (cm)	15	5	12

3.4.2. Online Updating of Position

The second experiment investigates the tracking capabilities for position changes. For this experiment, we placed the cylindrical object in one of the five starting positions P_1–P_5. For positions P_1–P_4, we moved the object by hand toward position P_5 once the arms started moving, covering a distance of 10 cm in roughly one

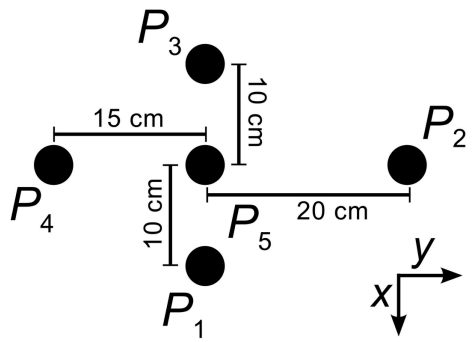


FIGURE 10 | This is the object placement layout for all experiments. Point P_5 is located at a distance of 45 cm along the *x*-axis of the world coordinate frame originating in the center of the robot's trunk. The shifts of points P_1 to P_4 are illustrated.

TABLE 2 | Results of first experiment.

	Cylinder	Square cuboid	Oblong cuboid	Total
No. of successful trials	14	13	19	46
Success rate (%)	93.33	86.67	95	92
Successful classifications	12	12	19	43
Classification rate (%)	85.71	92.31	100	93.48

to two seconds. When the object started in position P_5, we instead moved the object in the direction of one of the four other starting positions by 10 cm. These eight conditions are tested three times for a total of 24 trials, which the robot performed at a fixed trunk joint angle of 0°.

Of the 24 trials, 21 were successful (87.5% success rate). In seven trials the cylinder was erroneously recognized as a square cuboid from the start or after the hand of the experimenter had touched the object to move it to the new position (see **Table 3** for a listing of successful trials per condition). Three trials were counted as failed due to a safety stop of the experiment. In two of these cases, the arm configuration reached a singularity, in the third case the fingers almost collided with the object due to an erroneous position estimate. **Table 3** also lists the rate of correct classification in successful trials.

3.4.3. Online Updating of Orientation

For the third experiment, we picked the two cuboids, which require a distinct hand orientation for grasping. Each object was placed in one of the starting positions P_1–P_5. Once the arm started moving, we turned the object in place around 45° within about one second. We repeated this three times for each object and starting position, altering the starting orientation and turning direction, ending up with 30 trials. Grasps of the square cuboid were performed with a trunk joint angle of 45°, while the top grasp object was grasped with the trunk being at 0° (see first experiment).

Out of 30 trials, 25 were successful (83.34% success rate, see **Table 4**). We repeated two trials for the square cuboid due to an erroneous estimate of base shape (circle instead of square). Of the two failed trials for the square object, one was a safety stop near a singular arm configuration, while the other failed due to an error in behavioral organization (the fingers did not open). The three failed trials of the longish cuboid comprise two wrongly estimated orientations (and safety stops before collision) and one approach was aborted by the behavioral organization caused by a reverse detection in an estimation field.

TABLE 3 | Results of second experiment.

	$P_1 \rightarrow P_5$	$P_2 \rightarrow P_5$	$P_3 \rightarrow P_5$	$P_4 \rightarrow P_5$	Total
No. of successful trials	3	3	3	2	11
Success rate (%)	100	100	100	66.67	91.67
Classification rate (%)	66.67	0	66.67	50	45.45
	$P_5 \rightarrow P_1$	$P_5 \rightarrow P_2$	$P_5 \rightarrow P_3$	$P_5 \rightarrow P_4$	Total
No. of successful trials	3	1	3	3	10
Success rate (%)	100	33.33	100	100	83.33
Classification rate (%)	100	100	100	66.67	90

TABLE 4 | Results of third experiment.

Square	P_1	P_2	P_3	P_4	P_5	Total
No. of successful trials	2	3	3	3	2	13
Success rate (%)	66.67	100	100	100	66.67	86.67
Oblong	P_1	P_2	P_3	P_4	P_5	Total
No. of successful trials	3	1	2	3	3	12
Success rate (%)	100	33.33	66.67	100	66.67	80

4. CONCLUSION

The neural dynamics architecture presented in this paper integrates modules that have previously been developed for scene representation, concurrent object classification and pose estimation, behavioral organization, and movement generation into one big dynamical systems. Sequences of perceptual events induce reach and grasp actions, as the architecture goes through controlled instabilities. As a result, the system is open to time-varying sensor information at all times. We demonstrated on-line updating of reaching and grasping movements to shifts and rotations of the object. The architecture also responds flexibly at the level of organization. When a target object is removed, the perceptual and motor actions are abandoned and the system returns to scene exploration. When the concurrent object classification and pose estimation fail to converge, for instance, because the object is too different from a learned template, then the perceptual process terminates and the system similarly returns to scene exploration.

The stability of all relevant states in the neural dynamics is critical for both integration and online updating. Attractor states are robust to the changes in the dynamics of a component that occur as the component is coupled into the larger architecture.

Instabilities, at which attractor states disappear, are controlled through the mechanism of a condition of satisfaction.

Although we have evaluated the implementation of the architecture quantitatively, the current model is a demonstration of principle, that has not yet fully exploited all features of the approach. We did not use the size estimates obtained from the object classification system, for instance, and have made use of only a small number of shape templates, which in addition resemble the target objects and do not show generalization to objects of different shape. The avoidance of obstacles was not a focus of this work. We believe that the human-like organization and the smooth temporal structure of behavior in the neural dynamics architecture will prove most useful when cognitive robots cooperate with humans. On-line updating is critical there, as human users will not always wait for their turn.

Finally, we did not yet address the issue of learning to grasp. There are two obvious parts of the architecture that could benefit from learning. One is the set of geometric shapes used during classification and pose estimation, the other is the grasp type associated with each geometric body. Naturally, the set of geometric shapes should arise from exposure to a large amount of graspable objects. Any learning process has to address the challenge of making the decision if the shape of an object can be sufficiently matched by an existing shape from the set or if the object shape should be added to the set of templates. This may also include a pruning process to remove shapes if they become obsolete by adding better-fitting shapes to the set. The links between grasp types and geometric shapes will also have to be established by a learning process. To decide if a grasp type is suitable for a geometric body (considering the base shape and the height), one may use a reinforcement learning approach by trying different grasp types for the same object and using the CoS activation (or its absence) of the grasping and lifting behaviors as positive or negative reinforcement signals. The links may also be established by learning from demonstration (see, for example, Herzog et al., 2012). If both learning of geometric shapes and links to grasp types are done concurrently, one might run into a chicken-egg problem of not being able to learn one without a mature state of the other. A developmental process of first restricting possible shapes and executable grasps to small and primitive sets and bootstrapping the architecture with increasing complexity over time is a possible procedure to overcome this dilemma.

AUTHOR CONTRIBUTIONS

All authors designed the study. GK, SZ, and HR defined the model. GK implemented the model and performed the stimulations and experiments. GS wrote the paper, using input from the other authors.

FUNDING

We gratefully acknowledge funding through the EU project "NeuralDynamics" (GS, coordinator).

REFERENCES

Adams, B., Breazeal, C., Brooks, R. A., and Scassellati, B. (2000). Humanoid robots: a new kind of tool. *IEEE Intell. Syst. Appl.* 15, 25–31. doi: 10.1109/5254.867909

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybernet.* 27, 77–87.

Andersen, R. A., and Cui, H. (2009). Intention, action planning, and decision making in parietal-frontal circuits. *Neuron* 63, 568–583. doi: 10.1016/j.neuron.2009.08.028

Carbone, G., (ed.). (2013). *Grasping in Robotics, Vol. 10*. London, UK: Springer.

Cowley, A., Cohen, B., Marshall, W., Taylor, C. J., and Likhachev, M. (2013). "Perception and motion planning for pick-and-place of dynamic objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo), 816–823.

Curtis, N., and Xiao, J. (2008). "Efficient and effective grasping of novel objects through learning and adapting a knowledge base," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Nice), 2252–2257.

Desmurget, M., and Grafton, S. (2000). Forward modeling allows feedback control for fast reaching movements. *Trends Cogn. Sci.* 4, 423–431. doi: 10.1016/S1364-6613(00)01537-0

Erlhagen, W., and Bicho, E. (2006). The dynamic neural field approach to cognitive robotics. *J. Neural Eng.* 3, R36–R54. doi: 10.1088/1741-2560/3/3/R02

Fard, F. S., Hollensen, P., Heinke, D., and Trappenberg, T. P. (2015). Modeling human target reaching with an adaptive observer implemented with dynamic neural fields. *Neural Netw.* 72, 13–30. doi: 10.1016/j.neunet.2015.10.003

Faubel, C., and Schöner, G. (2009). "A neuro-dynamic architecture for one shot learning of objects that uses both bottom-up recognition and top-down prediction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (St. Louis, MO), 3162–3169.

Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Asfour, T., and Schaal, S. (2012). "Template-based learning of grasp selection," in *IEEE International Conference on Robotics and Automation (ICRA)* (St. Paul, MN: IEEE), 2379–2384.

Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Bohg, J., Asfour, T., et al. (2014). Learning of grasp selection based on shape-templates. *Auton. Robots* 36, 51–65. doi: 10.1007/s10514-013-9366-8

Huang, B., El-Khoury, S., Li, M., Bryson, J. J., and Billard, A. (2013). "Learning a real time grasping strategy," in *IEEE International Conference on Robotics and Automation (ICRA)* (Karlsruhe), 593–600.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 1254–1259.

Li, M., Hang, K., Kragic, D., and Billard, A. (2016). Dexterous grasping under shape uncertainty. *Robot. Auton. Syst.* 75, 352–364. doi: 10.1016/j.robot.2015.09.008

Lisman, J. (2015). The challenge of understanding the brain: where we stand in 2015. *Neuron* 86, 864–882. doi: 10.1016/j.neuron.2015.03.032

Madry, M., Song, D., and Kragic, D. (2012). "From object categories to grasp transfer using probabilistic reasoning," in *IEEE International Conference on Robotics and Automation (ICRA)* (St. Paul, MN), 1716–1723.

Petsch, S., and Burschka, D. (2010). "Estimation of spatio-temporal object properties for manipulation tasks from observation of humans," in *IEEE International Conference on Robotics and Automation (ICRA)* (Anchorage, AK), 192–198.

Platt, R., Grupen, R. A., and Fagg, A. H. (2006). "Learning grasp context distinctions that generalize," in *6th IEEE-RAS International Conference on Humanoid Robots* (Genova: IEEE), 504–511.

Reimann, H., Iossifidis, I., and Schöner, G. (2011). "Autonomous movement generation for manipulators with multiple simultaneous constraints using the attractor dynamics approach," in *IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai), 5470–5477.

Richter, M., Sandamirskaya, Y., and Schöner, G. (2012). "A robotic architecture for action selection and behavioral organization inspired by human cognition,"

in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vilamoura: IEEE), 2457–2464.

Rusu, R. B., and Cousins, S. (2011). "3D is here: point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 1–4.

Sandamirskaya, Y., Zibner, S. K. U., Schneegans, S., and Schöner, G. (2013). Using dynamic field theory to extend the embodiment stance toward higher cognition. *New Ideas Psychol.* 31, 322–339. doi: 10.1016/j.newideapsych.2013.01.002

Santello, M., Bianchi, M., Gabiccini, M., Ricciardi, E., Salvietti, G., Prattichizzo, D., et al. (2016). Hand synergies: integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Phys. Life Rev.* 17, 54–60. doi: 10.1016/j.plrev.2016.06.007

Schneiberg, S., Sveistrup, H., McFadyen, B., McKinley, P., and Levin, M. F. (2002). The development of coordination for reach-to-grasp movements in children. *Exp. Brain Res.* 146, 142–154. doi: 10.1007/s00221-002-1156-z

Schöner, G. (2008). "Dynamical systems approaches to cognition," in *Cambridge Handbook of Computational Cognitive Modeling*, ed R. Sun (Cambridge: Cambridge University Press), 101–126.

Schöner, G., Spencer, J. P., and the DFT Research Team (2015). *Dynamic Thinking: A Primer on Dynamic Field Theory*. New York, NY: Oxford University Press.

Strauss, S., and Heinke, D. (2012). A Robotics-based approach to modeling of choice reaching experiments on visual attention. *Front. Psychol.* 3:105. doi: 10.3389/fpsyg.2012.00105

Strauss, S., Woodgate, P. J., Sami, S. A., and Heinke, D. (2015). Choice reaching with a LEGO arm robot (CoRLEGO): the motor system guides visual attention to movement-relevant information. *Neural Netw.* 72, 3–12. doi: 10.1016/j.neunet.2015.10.005

Thelen, E., Corbetta, D., and Spencer, J. P. (1996). The development of reaching during the first year: the role of movement speed. *J. Exp. Psychol. Hum. Percept. Perform.* 22, 1059–1076. doi: 10.1037/0096-1523.22.5.1059

Zibner, S. K. U., Faubel, C., Iossifidis, I., and Schöner, G. (2011a). Dynamic neural fields as building blocks for a cortex-inspired architecture of robotic scene representation. *IEEE Trans. Auton. Ment. Dev.* 3, 74–91. doi: 10.1109/TAMD.2011.2109714

Zibner, S. K. U., Faubel, C., and Schöner, G. (2011b). "Making a robotic scene representation accessible to feature and label queries," in *Proceedings of the First Joint IEEE International Conference on Development and Learning and on Epigentic Robotics, ICDL-EPIROB* (Frankfurt).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Knips, Zibner, Reimann and Schöner. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

HYPOTHESIS AND THEORY
published: 03 February 2017
doi: 10.3389/fnbot.2017.00005

Neurodynamics in the Sensorimotor Loop: Representing Behavior Relevant External Situations

Frank Pasemann *

Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany

In the context of the dynamical system approach to cognition and supposing that brains or brain-like systems controlling the behavior of autonomous systems are permanently driven by their sensor signals, the paper approaches the question of neurodynamics in the sensorimotor loop in a purely formal way. This is carefully done by addressing the problem in three steps, using the time-discrete dynamics of standard neural networks and a fiber space representation for better clearness. Furthermore, concepts like meta-transients, parametric stability and dynamical forms are introduced, where meta-transients describe the effect of realistic sensor inputs, parametric stability refers to a class of sensor inputs all generating the "same type" of dynamic behavior, and a dynamical form comprises the corresponding class of parametrized dynamical systems. It is argued that dynamical forms are the essential internal representatives of behavior relevant external situations. Consequently, it is suggested that dynamical forms are the basis for a memory of these situations. Finally, based on the observation that not all brain process have a direct effect on the motor activity, a natural splitting of neurodynamics into vertical (internal) and horizontal (effective) parts is introduced.

Keywords: neurodynamics, behavior control, sensorimotor loop, mathematical concepts, neural representations

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark, Denmark

Reviewed by:
Ralf Der, Leipzig University, Germany
Jörn Fischer, Mannheim University of Applied Sciences, Germany

***Correspondence:**
Frank Pasemann
frank.pasemann@uni-osnabrueck.de

Received: *14 November 2016*
Accepted: *17 January 2017*
Published: *03 February 2017*

Citation:
Pasemann F (2017) Neurodynamics in the Sensorimotor Loop: Representing Behavior Relevant External Situations. Front. Neurorobot. 11:5. doi: 10.3389/fnbot.2017.00005

1. INTRODUCTION

From a neurocybernetics perspective the dynamical systems approach to embodied cognition can be traced back to the work of Ashby (Ashby, 1960) and von Foerster (Von Foerster, 1960). The assumption is that a living organism, in order to survive, must be able to develop internally some stable "entities" (von Foerster) which refer to or classify objects and situations in the physical world. These "entities" are the result of cognitive and sensorimotor processes developing through continuous interactions of an individual with its specific environment. On the other hand, cognitive and sensorimotor processes, relevant for the behavior of the individual, depend on the formation of these stable structures; i.e., they are complementary in the sense that one defines or implies the other. The assumption was, that an organism must be able to relate *discrete* internal structures to relevant aspects of its own interaction with its environment.

Although, the underlying processes are continuous these internal "entities" have to be discrete because the referenced objects or situations are discrete features of the environment. They also have to be "stable" in a certain time domain. On the other hand, due to changing sensorimotor or cognitive processes, they have to get "unstable" in the sense that different references have to be built up; i.e., new "stability domains" have to be visited or formed.

To pursue the dynamical systems approach to embodied cognition in this spirit, this paper will consider an individual as an autonomous system called an *animat*. An animat Dean (1998) and Guillot and Meyer (2001) is a simulated or physical robot equipped with sensors and actuators, and a neural network for behavior control. The neural controllers then have to operate in the so called *sensorimotor* loop, getting inputs from sensory signals and generating motor signals, which in turn will lead to new sensor inputs. The essential role of these closed loop processes for living or live-like systems has been discussed over several decades now from various points of view (Bishop, 1960; Beer, 1995; Di Paolo, 2003; Philipona et al., 2004; Hülse et al., 2007; Zahedi et al., 2010; Sándor et al., 2015). Here we use a purely formal approach and carefully analyze the dynamical description by making successive approximations to these processes.

Neurocontrollers, mimicking their biological counterparts, are considered as recurrent neural networks which in general allow for dynamical properties. That is, for fixed synaptic weights, bias terms and inputs such a network can be described as a dynamical system. Then, assuming that a neurocontroller is driven by slow sensor inputs, it will be properly described as a parametrized family of dynamical systems, where sensor inputs (and proprioceptive signals as well) are considered in a first approximation as parameters of such a family. Furthermore, for every parameter value the corresponding dynamical system may have a manifold of different attractors. The postulated internal "entities" then will be identified with the basins of attraction of *parametrically stable* neurodynamical systems. The interaction with the environment then may change the references to situations in the external world by changing parameter values given, for instance, by the sensor signals. This process of changing references will be described by so-called *bifurcations*.

For theoretical reasons, parameters are assumed to change so slowly that the system can approach its asymptotic states. This is often not the case for realistic sensor inputs. So, in a second step we will introduce sequences of neural states called *meta-transients* as for instance in Negrello and Pasemann (2008) Negrello (2011), and Toutounji and Pipa (2014).

In general these meta-transients can not be given an interpretation as trajectories of a dynamical system, mainly because the inducing sequence of sensor signals is not a trajectory of a dynamical system on sensor space. Instead, because of the closed loop, it is superposition of movements in the environment and the result of motor actions. The case where one has access to controlling parameters has often been discussed in geometric control theory (Gardner, 1983; Sussmann, 1983; Respondek, 1996; Kloeden et al., 2013). There then one can generalize the concept of attractors and the like. Although we do not find this approach applicable for the dynamics in the sensorimotor loop we will work with a comparable view.

Finally, these meta-transients have to be mapped to motor neurons, inducing then actions of the animats body; i.e., its behavior. Due to this projection not all elements of the neural system will be involved directly in the generation of motor signals. This leads naturally to a fiber structure over the motor space allowing to introduce the concepts of vertical or *internal* neurodynamics, having no direct effect on behavior, and a horizontal or *effective* neurodynamics, the projection of which generates the movements of the animat.

To clarify concepts, the paper will address the discrete-time neurodynamics of networks composed of standard sigmoidal neurons of additive type. Using this simplifying setup, it is assumed that the aspects described in the following are transferable also to neural systems employing more biologically plausible or other types of neurons. The basic concern here is to specify the role of, for example, attractors, basins of attraction, transients, bifurcations and stability properties in the context of systems acting in a sensorimotor loop.

Approaching the description of neurodynamics in the sensorimotor loop in three steps, we will first define the type of neurodynamics studied in this paper (Section 2), exemplifying it by some well known results. Assuming that sensor inputs are slow when compared to the activity dynamics of the neural system, we argue in Section 3 that neural systems in the sensorimotor loop are effectively described by parametrized families of dynamical systems, were parameters correspond to the sensor inputs. Other parameters, not considered here, are, for instance, signals coming from proprioceptors and the synaptic weights of the network, the change of which usually is associated with learning. Referring to the more realistic situations, meta-transients are introduction in Section 4. Finally, Section 5 discusses the generation of motor signals resulting from a projection of attractor transients or meta-transients, respectively, to the motor space; this then allows to differentiate between so called effective and internal neurodynamics. Finally the sensorimotor loop is closed through the environment by a formal mapping from motor space M to sensor space S. The paper concludes with a discussion of the possible role the introduced concepts can play for understanding neural representations of behavior relevant situations in the external world and, correspondingly, for a notion of memory which is not based on specific attractors like, for instance, fixed point attractors in Hopfield networks.

2. NEURODYNAMICS

Besides the body of an animat, three different parts of it will be discerned: The "brain" considered as a recurrent neural network N with n neurons. Its sensor neurons will prescribe the sensor space S, and the output neurons will define the motor space M. Sensor space S and motor space M are the interfaces of the "brain" N to the physical world. Assuming strictly the point of view of an animat, the world for an animat is what happens on its sensor surface. We describe these parts more concrete as follows.

A state $a(t) \in A \subset \mathbb{R}^n$ of the neural system N at time t is characterized by the activation of all its n neurons. Correspondingly, the state space A is called the *activation space* or phase space of N. It is a manifold of dimension $dim(A) = n$. Neural states may be represented in an equivalent way by the outputs $o(t)$ of the n neurons, and we call the corresponding state space of the network N its *output space* A^*.

The *sensor space* S consists of all possible sensor inputs, i.e., a sensor state $s(t) \in S$ at time t consists of all sensor values at time t. The sensor space is assumed to be a bounded manifold

S of dimension $\dim S = m$, where m denotes the number of distinct sensor elements. S may be subdivided into modality spaces corresponding, for example, to visual, acoustic, or haptic inputs.

A motor state $m(t) \in M$ at time t is given by the activation of the motor neurons at time t driving the various actuators of the animat. Thus, the *motor space* of the animat is an open bounded manifold M of dimension $\dim M = k$, the number of all its motor neurons. The motor space M may be segmented into different domains responsible e.g., for head movement, eye movement, driving wheels, arms, and so forth. Special domains may be related to corresponding domains in sensor space: Fixed infrared sensors may be, for instance, related only to the wheels domain; but with pan-tilt-camera vision is related to wheels, and pan-tilt-motors, et cetera.

2.1. Discrete-Time Neurodynamics

For a general introduction into the theory of dynamical system see for example (Abraham and Shaw, 1992; Hirsch et al., 2012; Strogatz, 2014). Here, in a first approximation we will understand the neural system N as a discrete-time dynamical system (Kloeden et al., 2013); i.e., on its activation space A there exists a differentiable map $\phi : \mathbb{Z} \times A \to A$, called the *flow*, with the following properties:

(i) $\phi(0, a_0) = a_0$, for all $a_0 \in A$.
(ii) $\phi(s + t, a_0) = \phi(s, \phi(t, a_0))$, for all $s, t \in \mathbb{Z}$ and $a_0 \in A$,

were $\mathbb{Z}$ denotes the set of nonnegative integers. In the following we consider a neural network N with activation space $A \subset \mathbb{R}^n$, writing it as $N(A)$, which is composed of n standard additive neurons with sigmoid transfer function $\tau := \tanh$. The flow of this system is then generated by a diffeomorphism $f : A \to A$ given in component form by

$$a_i(t+1) := \theta_i + \sum_{j=1}^{n} w_{ij}\, \tau(a_j(t)), \quad i = 1, \ldots, n, \tag{1}$$

where θ_i represents a constant bias term of neuron i, w_{ij} the synaptic strength or weight from neuron j to neuron i, and τ denotes the transfer function. Thus, the output of neuron i is given by $o_i := \tau(a_i)$, and for the output space we have $A^* \subset (-1, 1)^n$.

The neural system $N(A)$, considered as a dynamical system, will be denoted by (A, f). In this section terms like the bias terms θ_i and synaptic weights w_{ij} are assumed to be constant. This means that we consider an isolated system; i.e., there is no neural plasticity involved, and sensor inputs are not considered.

Furthermore, we endow the vector space A with an Euclidean metric d_τ induced by the transfer function τ; i.e.,

$$d_\tau(a, a') := d(\tau(a), \tau(a')) = \sqrt{\sum_{i=1}^{n} (\tau(a_i) - \tau(a'_i))^2}\,.$$

Due to the saturation domains of the sigmoid τ the distance of activity states corresponding to very high (positive or negative) activations is very small.

The flow on the state space A is then defined by

$$\phi(t, a_0) := a(t) = f^t(a_0) = \underbrace{f \circ f \circ \ldots \circ f \circ f}_{t\ times}(a_0)\,,$$

where $a_0 \in A$ is called the *initial state*. The flow ϕ satisfies the group property; i.e., with initial condition $\phi(0, a_0) = a_0$ one has

$$\phi(n, \phi(m, a_=)) = f^n \circ f^m(a_0) = f^{n+m}(a_0) = \phi(n + m, a_0)\,.$$

Example 1: The dynamics of 2-neuron networks have been analyzed extensively, in the continuous-time case as well as the discrete-time case, because already these simple systems, under certain conditions, can show all possible dynamical features: They can exhibit fixed point attractors as well as periodic, quasiperiodic and chaotic attractors, and even show co-existing attractors for one and the same condition (Wilson and Cowan, 1972; Marcus and Westervelt, 1989; Wang, 1991; Beer, 1995). Here we recall some of the results, which can be found for example in Pasemann (2002), to demonstrate basic properties of recurrent neural networks for this most simple case. So, let (A, f) denote the two-dimensional system given by two neurons (compare **Figure 1**) satisfying the equations

$$\begin{aligned} a_1(t+1) &:= \theta_1 + w_{11}\, \tau(a_1(t)) + w_{12}\, \tau(a_2(t))\,, \\ a_2(t+1) &:= \theta_2 + w_{21}\, \tau(a_1(t)) + w_{22}\, \tau(a_2(t))\,. \end{aligned} \tag{2}$$

As a bounded dissipative dynamical systems, the time development of neural states can be characterized by attractors and transients. We first recall some basic definitions.

A time-sequence of states

$$O(a_0) := \{a_0, a(1), \ldots, a(t), \ldots\}\,, \quad a_0 \in A\,, \tag{3}$$

is called an *orbit* or a *trajectory* of the system starting from $a_0 \in A$. An orbit $O(a_0)$ is called *periodic* of period $p \geq 1$ if $a(p) = a_0$, and p is the smallest integer such that this equation holds. For $p = 1$ the orbit is called a stationary state or a *fixed point* of the system. A *p-periodic point* is a state on a p-periodic orbit $O(a_0) = \{a_0, a(1), \ldots, a(p)\}$. It corresponds to a fixed point of the p-th iterate f^p of the map f:

$$f^p(a) := \underbrace{f \circ f \circ \cdots \circ f}_{p-times}(a) = a\,, \quad a \in A\,.$$

Let $U \subset A$ denote a subset which is *invariant* under the action of f; i.e. $f(U) = U$. A closed and bounded set $\Gamma \subset U$ is called

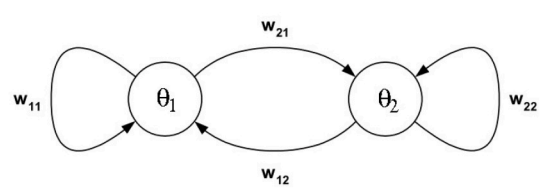


FIGURE 1 | A 2-neuron network.

an *attractor* of the dynamical system (A, f), if $f(\Gamma) = \Gamma$ and there exists an $\varepsilon > 0$ such that

$$d(a_0, \Gamma) \leq \varepsilon, \; a_0 \in U, \;\; \text{implies that} \;\; d(a(t), \Gamma) \to 0 \;\; \text{as} \;\; t \to \infty\,.$$

There are different types of attractors: Fixed points, periodic orbits (a finite set of periodic points) as in **Figure 2**, *quasiperiodic orbits* represented by a dense set of points on a closed line, and so called *chaotic attractors* which are characterized, for instance, as a fractal set in A (compare also **Figure 3** and Abraham and Shaw, 1992; Hirsch et al., 2012; Strogatz, 2014). If Γ is the only attractor of a system (A, f), then it is called a *global* attractor.

The *basin of attraction* $B(\Gamma)$ of an attractor Γ is the set of all initial conditions $a_0 \in A$ such that $d(a(t), \Gamma) \to 0$ as $t \to \infty$. Thus, the basin of attraction of Γ is considered as the set of all orbits attracted by Γ. A *transient* $O(\Gamma)$ of a system (A, f) is an orbit in the basin of an attractor Γ.

A dynamical system (A, f) can have more than one attractor. Then we say that the system has several *co-existing* attractors. For instance, in **Figure 4** four co-existing period-2 attractors and their basins with regular boundaries are shown. **Figure 5** displays several co-existing attractors separated by fractal basin boundaries.

Often one uses the metaphor "landscape" to describe a dynamical system (A, f) qualitatively. This refers exactly to what we defined as the flow of the dynamical system (A, f). One can think about water running downhill into a sink when referring to transients approaching an attractor. Basin boundaries then correspond to water partings. An *attractor-landscape*, denoted by $[A]$, then visualizes the different types of attractors present in the system together with their basins of attraction and basin boundaries as shown in the figures above.

Two different dynamical systems can have similar landscapes in the sense that there is the same number and type of attractors involved; but attractors, as well as the corresponding basin boundaries, may be deformed with respect to each other. If one can map the attractor-landscape of one system onto the attractor-landscape of the other system such that orbits are mapped one-to-one onto each other by preserving the time direction, then the qualitative behavior of such systems is comparable. This situation is formalized by the following

Definition 1. *Two discrete-time dynamical systems* (A, f) *and* (B, g) *are said to be* topologically conjugate, *if there exists a homeomorphism* $\psi : A \to B$, *such that* $f \circ \psi = \psi \circ g$, *i.e., such*

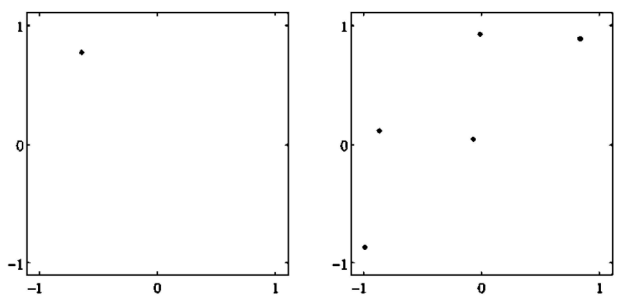


FIGURE 2 | Examples of attractors in (o_1, o_2)-output space for a two neuron system (2). (Left) A fixed point attractor. **(Right)** A period-5 attractor. (Parameters are given in **Table A1** in the Appendix referring to networks sys1 and sys2.)

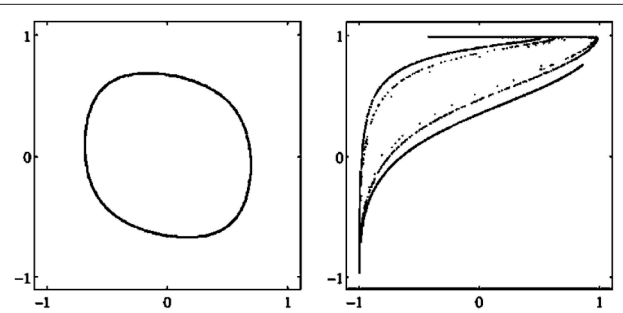


FIGURE 3 | Examples of attractors in (o_1, o_2)-output space for a two neuron system (2). (Left) A quasiperiodic attractor. **(Right)** A chaotic attractor. (Parameters are given in **Table A1** in the Appendix referring to networks sys3 and sys4.)

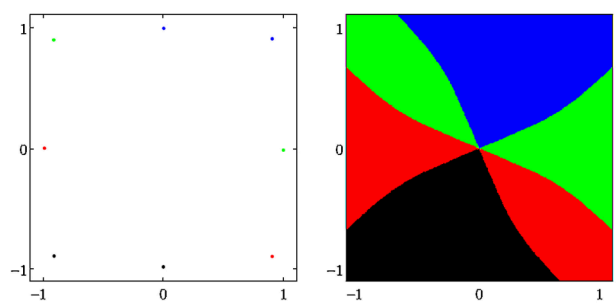


FIGURE 4 | (Left) Four co-existing period-2 attractors in (o_1, o_2)-output space. **(Right)** Their basins of attraction (for parameters see **Table A1**, network sys5).

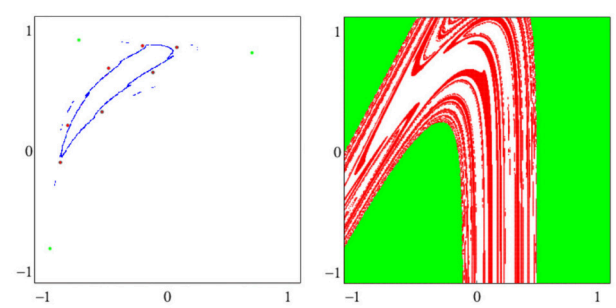


FIGURE 5 | (Left) A period-3 attractor (green) and a period-7 attractor (red) in (o_1, o_2)-output space, co-existing with two chaotic attractors, one cyclic with period 14. **(Right)** The corresponding basins of attraction; the two basins of the chaotic attractors are white. The system is given as sys6 in **Table A1**.

that the following diagram commutes:

$$\begin{array}{ccc} A & \xrightarrow{f} & A \\ \psi \downarrow & & \downarrow \psi \\ B & \xrightarrow[g]{} & B \end{array}$$

3. PARAMETRIZED FAMILIES OF DYNAMICAL SYSTEMS

In the last section the bias terms θ_i and synaptic weights w_{ij}, $i, j = 1, \ldots, n$, were held constant, and one can consider them as parameters of the neural system (A, f). For different bias terms or synaptic weight one gets different dynamical systems. Thus, we introduce a *parameter space* $Q \subset \mathbb{R}^q$ for a neural system (A, f) as a q-dimensional Euclidean manifold (Q, h) with metric h. A parameter vector $\rho = (\theta, w) \in Q$ is given by the bias vector θ and the weight matrix w of the network $N(A)$. Thus, one has $\dim Q = q = n \cdot (n + 1)$.

As a next step we argue that the sensor inputs to the neural system $N(A)$ can be assumed to act as parameters of the neurodynamics. Because brain-like systems will always act in a sensorimotor loop, the sensor signals $s(t) \in S$ will always drive the neurodynamical system $N(A)$. Assuming in a first approximation that the sensor signals $s(t)$ change so slowly that the orbits of the neural system are always able to converge to an attractor, then they can be considered as varying parameters. For that reason we will subsume the sensor signals $s(t)$ as part of the bias terms $\theta(t) := \hat{\theta} + s(t)$, with $\hat{\theta} =$ constant.

A neural system then has to be described as a *parametrized family of discrete-time dynamical systems* denoted by $(A, f; Q)$, with $A \subset \mathbb{R}^n$ the activation space, $Q \subset \mathbb{R}^q$ the parameter space, and a differentiable map $f : Q \times A \to A$. For a specific parameter vector $\rho \in Q$, we write $f_\rho : A \to A$ for the corresponding dynamical system, and denote the *q-parameter family* of neurodynamical systems also by (A, f_ρ), $\rho \in Q$. The only varying parameters considered in the following are the bias terms θ_i, $i = 1, \ldots, n$. As stated above, other parameters of the animats brain, like synaptic weights w_{ij} are constant.

We may now look at the "brain" as a fiber structure over parameter space Q (compare **Figure 6**): To every $\rho \in Q$ there is attached the activation space A together with the flow ψ_ρ corresponding to $\rho \in Q$; i.e., there is a whole attractor-landscape, denoted by $[A]_\rho$, attached to every parameter $\rho \in Q$.

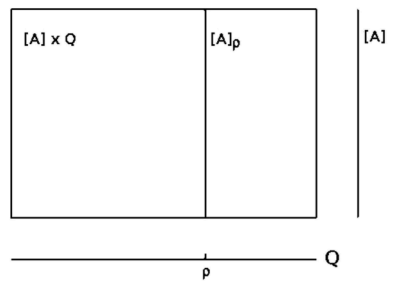


FIGURE 6 | The fiber structure of a neural system: there is an attractor-landscape $[A]_\rho$ attached to every parameter ρ.

3.1. Parametric Stability

Now, given two different parameter vectors ρ and ρ' in Q, one may ask if the corresponding attractor-landscapes are similar or not in the sense that there exist a homeomorphism carrying oriented orbits onto oriented orbits, especially attractors onto attractors. Using definition 1 we introduce the following

Definition 2. *Given a neurodynamical system* $(A, f; Q)$. *Two different parameters* ρ, $\rho' \in Q$ *are said to be* homologous *if the corresponding dynamic systems* (A, f_ρ) *and* $(A, f_{\rho'})$ *are topologically conjugate; i.e., if the following diagram commutes:*

$$\begin{array}{ccc} A & \xrightarrow{f_\rho} & A \\ \psi \downarrow & & \downarrow \psi \\ A & \xrightarrow[f_{\rho'}]{} & A \end{array}$$

If two parameter vectors ρ, $\rho' \in Q$ are homologous, then the corresponding neurodynamics have qualitative the same behavior; i.e., attractors and basin boundaries may be deformed. In **Figures 7**, **8**, for example, attractors and output signals of an oscillatory 2n-network with two different bias terms are displayed. The two attractor-landscapes $[A]_\rho$ and $[A]_{\rho'}$ corresponding to homologous parameters θ, θ' are qualitatively, i.e., topologically, the same.

This leads us to an essential concept, that of *parametric stability*, which we define in correspondence to the concept of structural stability in the general theory of dynamical systems (Thom, 1989).

Definition 3. *Given a neurodynamical system* $(A, f; Q)$ *and a parameter vector* $\rho_0 \in Q$. *Then the system* (A, f_{ρ_0}) *is called* parametrically stable, *if there exists an* $\epsilon > 0$ *such that for every* $\rho \in Q$ *satisfying* $||\rho - \rho_0|| < \epsilon$ *the systems* (A, f_ρ) *are topologically conjugate to* (A, f_{ρ_0}).

Definition 4. *Given a neurodynamical system* $(A, f; Q)$. *The* domain of parametric stability *corresponding to a parameter vector* $\rho_0 \in Q$, *denoted by* $\mathcal{P}(\rho_0) \subset Q$, *is the maximally connected parameter set in* Q *containing all* $\rho \in Q$ *which are homologous to* $\rho_0 \in Q$.

Thus, all systems (A, f_ρ) with $\rho \in \mathcal{P}(\rho_0)$ are topologically conjugate to (A, f_{ρ_0}).

Parametrically stable systems are essential for modeling experimental situations: If the experimental inaccuracy is smaller than a domains of parametric stability, then the model remains valid in spite of experimental perturbations. More general, parametric stability is an essential concept, because interesting real (i.e., physical, biological, etc.) phenomena are of course

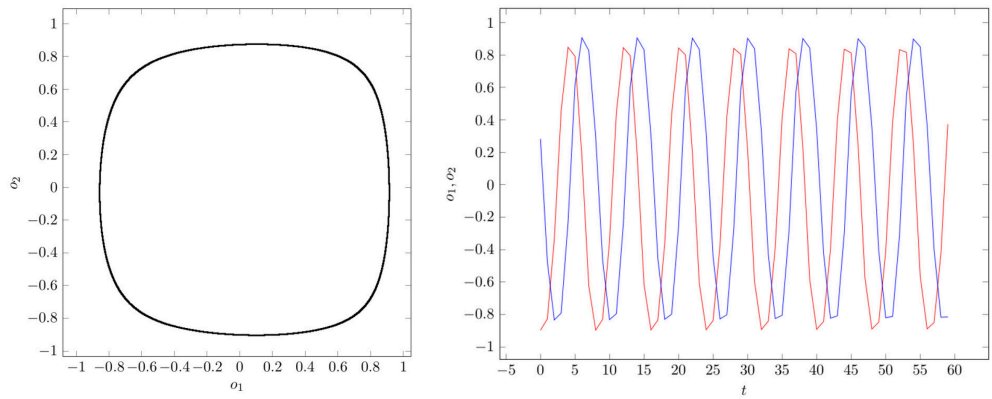


FIGURE 7 | (Left) Attractor in (o_1, o_2)-output space. **(Right)** Output signals of the 2-neuron oscillator $Osci_1$ with parameters given in **Table A1**.

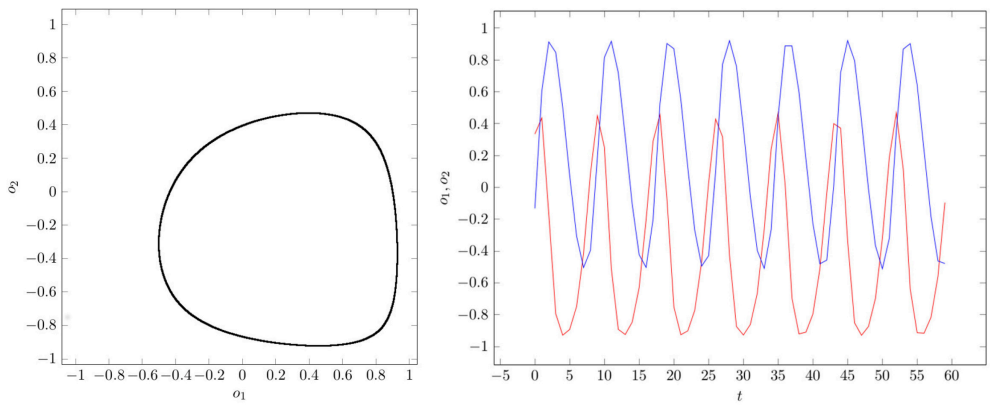


FIGURE 8 | (Left) Attractor in (o_1, o_2)-output space. **(Right)** Output signals of the 2-neuron oscillator $Osci_2$ with parameters given in **Table A1**.

those which are stable under small perturbations of their defining conditions. For instance, a convergent neural network may stay convergent under a small perturbation of their parameters.

3.2. Bifurcations

As a second step to describe the dynamics of neural systems we have assumed that the dynamics depends on control parameters, that is, on variables that vary much more slowly than the states of the system. Suppose these parameters change along a smooth path $\rho(t) \in Q$. If all $\rho(t)$ for $t \in [t_1, t_2]$ are homologous, the corresponding neurodynamical systems will show qualitatively the same behavior, although the attractors and their basins in activation space A will move and deform. To such a situation we refer to as a *morphing attractor-landscape* with its *morphing attractors* (Negrello and Pasemann, 2008; Negrello, 2011; Toutounji and Pipa, 2014).

But the path $\rho(t)$ may reach a point ρ_c in parameter space Q where the behavior of a system changes qualitatively, i.e., the type and/or numbers of attractors will change, when the path crosses ρ_c. Such points $\rho_c \in Q$ are called critical parameters or *bifurcation points*. Thus, bifurcation points are associated with the appearance of topologically non-conjugate systems. The values of $\rho_c \in Q$ are called the *bifurcation values*. The appearance of bifurcations in a system are often studied with the help of *bifurcation diagrams*. These are demonstrations of attractor sequences resulting from the variation of only one control parameter (compare **Figure 10**).

The (closed) subspace $\mathcal{K} \subset Q$ of all bifurcation points is called the *bifurcation set* of the system $(A, f; Q)$. Bifurcation sets are sets in Q (i.e., curves, surfaces, hyperspaces) which separate different domains of parametric stability.

Example 2: As the most simple example we will discuss a single neuron with self-connection w as a 2-parameter family of dynamical systems $(A, f; Q)$ given by

$$a(t+1) = \theta + w \cdot \tau(a(t)), \quad t \in \mathbb{Z}, \tag{4}$$

(compare also Pasemann, 1993a for a single neuron with logistic function $\sigma(x) = (1 + e^{-x})^{-1}$ as transfer function). Stability analysis tells us that for $|w| < 1$ there exist only global fixed points. Otherwise one will find bi-stable systems for $w > 1$, and a domain with global period-2 attractors for $w < -1$. Typical bifurcation diagrams are shown in **Figure 10**.

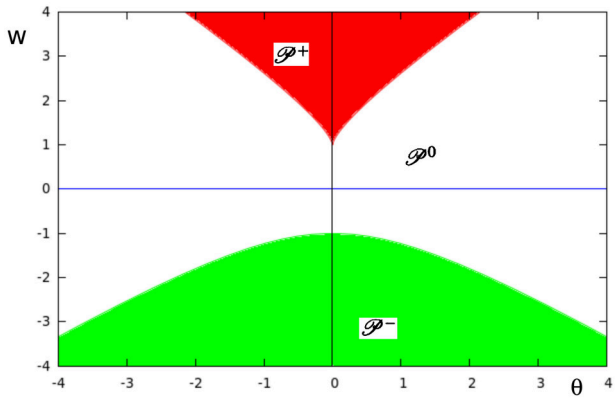


FIGURE 9 | The parameter space $Q \subset R^2$ of a single neuron with self-connection w with its three domains of parametric stability: $\mathcal{P}^0$ (white) relates to global fixed point attractors, $\mathcal{P}^+$ (red) to bi-stable systems, and $\mathcal{P}^-$ (green) to period-2 oscillations. These domains are separated by bifurcation sets $\mathcal{K}^+$ and $\mathcal{K}^-$, respectively.

In **Figure 9** the three different domains of parametric stability in $Q \subset R^2$ are shown: Here $\mathcal{P}^0$ (white) denotes the parameter domain for systems having a global fixed point attractor, $\mathcal{P}^+$ (red) refers to bi-stable systems, and $\mathcal{P}^-$ (green) to oscillatory systems. They are separated by bifurcation sets $\mathcal{K}^+$ and $\mathcal{K}^-$ in Q. Thus, a single neuron with self-connection comes in three dynamical forms (compare definition 5).

At $\mathcal{K}^+$, that is for $w \geq 1$, there are saddle-node bifurcations, and at $\mathcal{K}^-$, that is for $w \leq -1$, there are period-doubling bifurcations. This can be clearly seen in the bifurcation diagrams of **Figure 10**. They show that a single neuron with positive self-coupling can act as a hysteresis element (short term memory), whereas a neuron with negative self-connection can serve as a switchable oscillator (compare also Pasemann, 1993a).

What should be taken from this simple example is, that in situations where there are parameter domains for which there are coexisting attractors, it depends on the direction from which a path $\rho(t)$ in parameter space Q hits a bifurcation set $\mathcal{K} \subset Q$ (compare **Figure 10**). This leads to phenomena, called *generalized hysteresis* effects, demonstrating that the development of the system depends crucially on the history of the system. And therefore the behavior of these *path-dependent* systems will not be explicitly deducible from the knowledge of their actual state. This is one reason for the "complexity" of neural systems, and a source of their fascinating faculties.

Having clarified the decisive role of domains of parametric stability $\mathcal{P} \subset Q$ for the behavior of parametrized family of dynamical systems, it is natural to associate to a non-critical parameter vector $\rho^* \in Q$ a set of dynamical systems (A, f_ρ) which are parametrically stable with respect to $\rho^* \in Q$. With reference to the designation of Thom Thom (1989), we give the following

Definition 5. *Given a system $(A, f; Q)$, and let $\rho_0 \in Q$ denote a non-critical parameter vector. A* dynamical form *of $(A, f; Q)$ is a connected set $\mathcal{F}_{\rho_0} \subset Diff(A)$ of dynamical systems (A, f_ρ) which are topologically conjugate to (A, f_{ρ_0}).*

Assuming that changing parameter values correspond to changing sensor signals, one can deduce that if a sequence of signals stays in a certain domain of parametric stability $\mathcal{P}$, the dynamics of the neural system stays qualitatively the same. And therefore we can assume that the resulting behavior of the controlled system, the animat, will not change dramatically.

4. META-TRANSIENTS

In the next step we will have to ease the restrictions on the parameters by assuming that the sensor signals can change so fast that the activations $a(t)$ of the neurodynamical system (A, f_ρ) can not approach an attractor $\Gamma \subset A$ asymptotically.

In the following the considered parameters will be the sensor inputs $s(t)$ of an animat, and all other parameters are fixed. Due to properties of the environment, or due to the behavior of the animat, its sensor inputs may change so fast that they can not be considered as parameters in the strict mathematical sense.

Such a situation is often described in terms of the dynamics of non-autonomous systems. But it is different from the situations covered by control theory (Gardner, 1983; Sussmann, 1983; Respondek, 1996) or by skew-product systems (Kloeden et al., 2013) in so far as a sequence of such sensor inputs is neither the trajectory of a dynamical system in parameter space, nor is it a well defined sequence leading to a preexisting goal. Here the sensor inputs depend on the dynamics of the physical environment (exo-motion) as well as on the movements/actions of the animat itself (ego-motion). We will come to that later again.

Assuming that parameters change almost as fast as the internal states, the resulting sequence of states is no longer that of a transient to one and the same attractor. Suppose the neural system at time t is in a state $a(t)$ on a definite transient $O(\Gamma_{\rho(t)})$ to an attractor $\Gamma_{\rho(t)}$ of the neural system $(A, f_{\rho(t)})$. If the parameter vector a short time later satisfies $\rho(t+k) \neq \rho(t)$ the corresponding state $a(t+k)$ will be an element of a different transient $O(\Gamma_{\rho(t+k)})$ to a different attractor $\Gamma_{\rho(t+k)} \subset A$.

So, let $\sigma_\theta := \{s(t), s(t+1), s(t+2), \ldots\}$ denote such a sequence of sensor inputs represented by a sequence of parameter vectors $\theta(t)$ in Q. This will induce a sequence of states $\alpha(\sigma_\theta) := \{a(t), a(t+1), a(t+2), \ldots\}$ on A with

$$a(t+1) = f_{\rho(t)}(a(t)), \quad \text{that is,} \quad a_i(t+1) = \theta_i(t) + \sum_{j=1}^{n} w_{ij}\, a_j(t). \tag{5}$$

Such a sequence $\alpha(\sigma_\theta)$ in A will be called a *meta-transient* (Negrello and Pasemann, 2008). Thus, a meta-transient is not a transient of a dynamical system, but it is a sequence of states $a(t) \in A$ following the morphing attractors of a sequence of the parametrized dynamical systems $(A, f_{\rho(t)})$. The projection of such a meta-transient on A back to the parameter space Q then gives the sequence of "driving" parameter values σ_θ.

If we define a map $\Phi : Q \times A \to A$ associated with the given parametrized family of dynamical systems by

$$\Phi(\rho, a) = f_\rho(a), \quad a \in A,$$

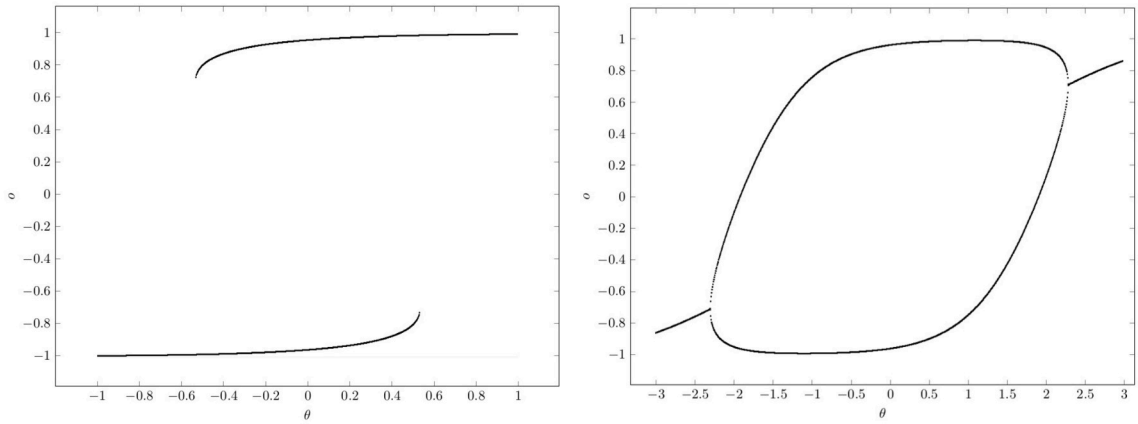


FIGURE 10 | Bifurcation diagrams for output space A^* of a single neuron with self-connection w; the bias term θ is varied back and forth over the interval. (Left) Demonstrating bi-stability and hysteresis for $w = 2.0$. **(Right)** Switching on and off a period-2 oscillator for $w = -2.0$.

then the elements of a meta-transient $\alpha(\sigma_\theta)$ are generated by this map according to

$$\alpha(\sigma_\theta) = \cdots \circ f_{\rho(t+2)} \circ f_{\rho(t+1)} \circ f_{\rho(t)}(a(t)) \,.$$

For example, if the input to a neuron with excitatory self-connection is slow when compared with the internal dynamics one will observe a clear hysteresis signal as in **Figure 10**. If the input signal changes much faster, then there will not be "jumps" at the boundaries of the hysteresis domains but a kind of "squashed" hysteresis loop will appear, as was observed for instance in Manoonpong et al. (2010) for the dynamics resulting from audio input signals.

Furthermore, if all the parameter values, corresponding to the sequence σ_θ of sensor inputs, lie in one and the same domain of parametric stability $\mathcal{P}$, the behavior of the animat's body will not change dramatically, and one may describe it as "the same." But if a sequence of parameter values crosses a bifurcation set $\mathcal{K}$ in parameter space Q the system may behave in a very different way.

5. PROJECTIONS TO MOTOR SPACE *M*

All the dynamics discussed so far has the goal to generate appropriate body movements. Therefore, the only interesting thing here is the effect of the activities of the neural system which activate the motor neurons. Thus, we have to project the meta-transients $\alpha(\sigma_\theta)$ on phase space A to the motor space M with $\dim(M) = k < n$. This projection, denoted by $\Pi : A \to M$, is assumed here to correspond to the application of a one-layer feedforward network (compare **Figure 11**). The activations of the k motor neurons then are spanning the output layer, and we define

$$\Pi(a)_j := \sum_{i=1}^{n} w_{ji}\tau(a_i) \,, \quad a \in A, \quad j = 1, \ldots, k, \tag{6}$$

where w_{ji}, $i = 1, \ldots, n$, $j = 1, \ldots, k$ denote the weights from the n internal neurons to the k motor neurons. The activation of the jth motor neuron $m_j \in M$ having a bias value θ_j^M is then given by

$$m_j = \theta_j^M + \Pi(a)_j \,, \quad a \in A, \quad j = 1, \ldots, k. \tag{7}$$

Such a motor neuron in general will not be connected to all of the brains neurons. Therefore, there will be many internal states $a \in A$ which will project to identical motor activations $m \in M$. This will give the second fiber structure of the sensorimotor loop, where the fiber $F_m \subset A$ over $m \in M$ is given by

$$F_m := \{a \in A \mid \Pi(a) = m\} \,, \quad m \in M. \tag{8}$$

Then, what is observable is the behavior of the animat generated by a sequence of motor states

$$\mu(\sigma_\theta) := \{m(t), m(t+1), m(t+2), \ldots\} \tag{9}$$

which corresponds to a given meta-transient $\alpha(\sigma_\theta)$ on A; that is, with $\rho(t) \in Q$, $a(t) \in A$, and bias terms of motor neurons $\theta^M \in \mathbb{R}^k$ one has

$$m(t) = \theta^M + \Pi \circ \Phi(\rho(t), a(t)) \,. \tag{10}$$

From the projection argument it is clear that not the whole state space A is of direct relevance for the behavior of the animat. It is obvious that the activity of neurons not connected to the motor neurons do not have a direct effect on the behavior of the animat. Therefore an attractor in A, if it is a fixed point, a periodic orbit or even a chaotic attractor, may be projected to only one and the same motor state $m \in M$; attractors, their transients or meta-transients may then have little or no effect on motor activities at all.

To reflect this property we introduce a splitting of every state $a \in A$ into a so called *horizontal* and a *vertical* part; i.e.,

$$a = a^v + a^h \,, \quad \text{with} \quad \Pi(a^v) := 0 \,. \tag{11}$$

And due to this splitting we have a direct decomposition of the space of brain states A into horizontal and vertical parts; i.e.,

$$A = A^v \oplus A^h\,, \tag{12}$$

where A^v is given as $A^v = \ker \Pi$.

Let there be $l \geq k$ internal neurons being directly connected with neurons in the motor layer; they serve as an l-dimensional input space $B \subset A^h$ of the feedforward network (compare **Figure 11**). Furthermore, due to the geometry of feedforward networks (Pasemann, 1993b), in general there is a $(l-k)$-dimensional linear subspace $C_m \subset B$ on which the activation of the motor neurons is constant.

The dynamics directly relevant for behavior then will actually live in the horizontal state space $A^h \subset A$. Correspondingly, what will lead to an effective behavior is a sequence of horizontal states given by a *horizontal meta-transient* on A

$$\alpha^h(\sigma_\theta) := \{a^h(t), a^h(t+1), a^h(t+2), a^h(t+3), \ldots\}\,. \tag{13}$$

Going back to section 3 let us consider again a discrete-time dynamical systems $f_\rho : A \to A$ with *fixed* parameter vector $\rho \in Q$. Then, post hock, we can introduce a well-defined splitting of the dynamical system f_ρ into vertical and horizontal parts by

$$f_\rho(a) = f^v_\rho(a^v) + f^h_\rho(a^h)\,, \quad f^v_\rho(a^h) := 0\,, \quad f^h_\rho(a^v) := 0\,. \tag{14}$$

It is obvious that only the *horizontal dynamics* $f^h_\rho : A^h \to A^h$ contributes to the observable behavior of an animat, whereas the *vertical dynamics* $f^v_\rho : A^v \to A^v$ will describe brain processes which may be associated to a dynamical kind of memory, to association, planning, dreaming, contemplation, and the like; that is, to the cognitive faculties of the brain.

Furthermore, suppose that two dynamical systems f_ρ and $f_{\rho'}$ with $\rho, \rho' \in \mathcal{P}, \mathcal{P} \subset Q$ a domain of parametric stability (compare section 3.1), are topologically conjugate. Then it is reasonable that their horizontal components f^h_ρ and $f^h_{\rho'}$ will generate motor states in M which lead to variants of a specific behavior. The next example gives a demonstration of this situation.

Example 3: In evolutionary robotics one often used the motor dynamics of a system as a fitness criterion to reduce the "ineffective" higher dimensional neurodynamics of evolved controllers to analyzable, minimalistic solutions for which the discussed effects could be studied (Wischmann and Pasemann, 2006; von Twickel et al., 2011; Pasemann et al., 2012). Here only a simple example of a neurocontroller may be given by the following recurrent neural network (**Figure 11**). It provides an obstacle avoiding behavior of a Khepera-like Robot (Toutounji and Pasemann, 2016). It uses five distance sensors (sensor layer) and two motor neurons (motor layer).

The hidden layer (the "brain") has eight neurons, but only two of them project to the two-dimensional motor space M. Though the brain dynamics runs in an 8-dimensional state space A only a 2-dimensional subspace $B \subset A$ determines the motor activity directly. What is going on in the 6-dimensional vertical state space has no immediate effect on the behavior of the robot. Indirectly, of course, the dynamics on A can

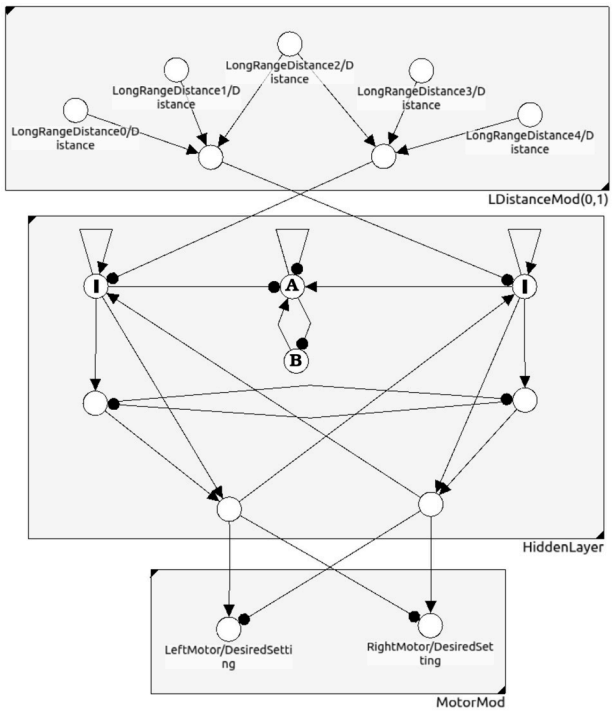


FIGURE 11 | A neural network for obstacle avoidance behavior of a two-wheeled robot. One discerns between the sensor space, the so called "brain" (hidden layer), and the motor space.

influence the behavior of the robot; for instance, the over-critical excitatory self-connections of the input neurons I (**Figure 11**) control the turning angle of the robot at walls. The submodule in $N(A)$ composed of the two neurons A and B has an interesting dynamics not influencing the motor behavior. They display a "chaotic" meta-transient while the the robot is turning, ending up in a period-2 attractor after a complete right turn, and in a period-4 attractor after a complete left turn. This internal (vertical) dynamics does not contribute to the behavior of the robot, but can be used as a kind of memory for subsequent decisions. The over all performance of this controller is comparable to that of the 2-neuron network called the MRC (minimal recurrent controller) in Hülse et al. (2004).

5.1. Closing the Loop

Every activity of the motor neurons will change the sensor input to the system (compare **Figure 12**). In this sense we have a closed loop, and one may call it the *ego-motion-loop*. But the essential point is, that this loop has to go through the environment of the system; i.e., how the motor activity is reflected by the sensor input depends, first, on the appearance and properties of the environment, and second, on processes in the environment itself, called *exo-motion*. This may lead to a discrimination of sensor input variations into those which are due to changes of the motor signals, and those which are due to changes in the environment only (Philipona et al., 2003).

That this inextricable fusion of two influences can not be described as a control theoretical type of closed loop with

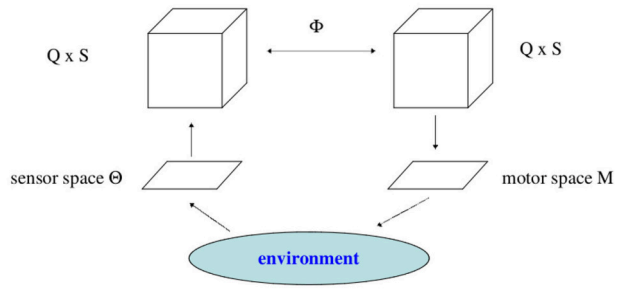


FIGURE 12 | The sensorimotor loop through the environment.

an additional noise term is clear by two facts: First, what is happening in the physical environment of an animat in general will not be a well defined process, and, second, the motor outputs, as we have seen, are not necessarily a direct reflex of the sensor inputs. Planning, focusing, ignoring performed by the vertical brain activation dynamics are modulating the reaction to sensor inputs. Thus, even formally it is difficult to describe the neural dynamics in the sensorimotor loop in terms of a control-theoretical model.

6. DISCUSSION

The description of biological brains as dynamical systems is often assumed to be an appropriate approach to describe cognition and the behavior of animals (Port and Van Gelder, 1995; Thelen and Smith, 1996). Based on the observation that the typical activity of an animat is a reaction to its environment, we used the sensorimotor loop to carefully approach the dynamics hypothesis in three steps. Relying on experiences in the field of evolutionary robotics (Nolfi and Floreano, 2000) we used discrete-time neurodynamics to, first, describe the (isolated) brains as dynamical systems. Having realized that (living) brains are always driven by sensor inputs, we made clear that the description of brains as parametrized families of dynamical systems is more appropriate. This allowed to introduce the concept of parametric stability which helped to formalize the general observation that a certain behavior is robust against "noise," and can be classified as "the same," although the initializing sensor inputs vary over a larger domain.

In a third step, assuming that sensor inputs may change so fast that they can not be assumed to serve as parameters in the mathematical sense (compare for instance Manoonpong et al., 2005), we were compelled to introduce the concept of meta-transients to describe the brains activity in a sensorimotor loop. These meta-transients in general will *not* be describable as orbits of a dynamical system. Finally, we used the fact that not all of the brains activity is directly reflected in the motor performance to discern between the brains effective (horizontal) and internal (vertical) activations.

In a more general sense the horizontal part is associated more with the sensorimotor pathways, whereas the vertical part is assigned to the higher centers of the brain, associated with cognitive faculties of a system. Of course horizontal and vertical processes are not decoupled and depend on each other; they are processes on one and the same highly recurrent network. As usual, higher centers are assumed to check the adequacy of the activities along the sensorimotor pathways; they are modulating the sensorimotor flow of signals. On the other hand, the vertical processes are permanently restricted by the "horizontal" flow of signals; otherwise, that is, without sensor inputs, they will run freely into perhaps noxious states of brain and body.

Following a purely formal approach to neurodynamics, we introduced in Section 3.2 the concept of parametric stability and the associated concept of a dynamical form. We think that these concepts may help to discuss questions concerning the representation of objects or, in this context better, behavior relevant situations in the external world.

From the dynamical point of view certain patterns of sensor inputs will be associated with the existence of certain attractors in activation space A; or otherwise stated, with the existence of a certain attractor-landscape. Because one has to assume that the brains dynamics is always driven by sensor inputs (including proprioception) it is more plausible to refer to a basin of attraction as a candidate for representing an external situation. Taking our argument for meta-transients serious it becomes obvious, that a dynamical form, associated with a certain type of behavior, is a reasonable representative for behavior relevant situations in the external world. Thus, taking parametric stability as essential for the reproducible identification of "the same" situations gives a reasonable conceptual basis for treating brain dynamics induced by an ever changing complex environment.

If one approves this interpretation then it will also allow for a less restrictive dynamical view on memory. Neural memories usually are represented by asymptotically stable fixed points, like in Hopfield's associative-memory model, or are conceived as periodic, quasiperiodic, or even chaotic attractors of neural networks. In fact, the correspondence between attractors and memories is one of the fundamental aspects of neural networks. But, as we have seen, situated in a sensorimotor loop and driven by sensor inputs, the best we can expect is that attractors of a neural network serve as kinds of symbols, while the system always runs on transients to these attractors (or on meta-transients). So in a first step memory should be associated with the basins of certain attractors. Taken that the natural situation is such that neural systems in the sensorimotor loop run on meta-transients, we have to assume that the union of all basins of attraction, belonging to the possibly morphing attractors of a dynamic form, should be identified with the memory of certain behavior relevant external situations. We will call this kind of memory model a *blurred* memory. The relation between learning and blurred memory will be the subject of further research.

7. AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and approved it for publication.

ACKNOWLEDGMENTS

The author would like to thank the Institute of Cognitive Science, University of Osnabrück, for its hospitality and support.

REFERENCES

Abraham, R., and Shaw, C. D. (1992). *Dynamics: The Geometry of Behavior*, Vol. 2. Redwood City, CA: Addison-Wesley.

Ashby, W. (1960). *Design for a Brain: The Origin of Adaptive Behaviour*. London: Chapman & Hall.

Beer, R. D. (1995). A dynamical systems perspective on agent-environment interaction. *Artif. Intell.* 72, 173–215. doi: 10.1016/0004-3702(94)00005-L

Bishop, G. (1960). "Feedback through the environment as an analog of brain functioning," in *Self-Organizing Systems*, eds M. C. Yovitts and S. Cameron (New York, NY: Pergamon), 122–152.

Dean, J. (1998). Animats and what they can tell us. *Trends Cogn. Sci.* 2, 60–67. doi: 10.1016/S1364-6613(98)01120-6

Di Paolo, E. A. (2003). "Organismically-inspired robotics: homeostatic adaptation and teleology beyond the closed sensorimotor loop," in *Dynamical Systems Approach to Embodiment and Sociality*, eds K. Murase and T. Asakura (Adelaide: Advanced Knowledge International), 19–42.

Gardner, R. B. (1983). *Differential Geometric Methods Interfacing Control Theory*. Boston, MA: Birkhauser.

Guillot, A., and Meyer, J.-A. (2001). The animat contribution to cognitive systems research. *Cogn. Sys. Res.* 2, 157–165. doi: 10.1016/S1389-0417(01)00019-5

Hirsch, M. W., Smale, S., and Devaney, R. L. (2012). *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Oxford: Academic Press.

Hülse, M., Wischmann, S., Manoonpong, P., von Twickel, A., and Pasemann, F. (2007). "Dynamical systems in the sensorimotor loop: On the interrelation between internal and external mechanisms of evolved robot behavior," in *50 Years of Artificial Intelligence*, eds M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer (Berlin: Springer), 186–195.

Hülse, M., Wischmann, S., and Pasemann, F. (2004). Structure and function of evolved neuro-controllers for autonomous robots. *Connect. Sci.* 16, 249–266. doi: 10.1080/09540090412331314795

Kloeden, P., Pötzsche, C., and Rasmussen, M. (2013). "Discrete-time nonautonomous dynamical systems," in *Stability and Bifurcation Theory for Non-Autonomous Differential Equations*, eds R. Johnson and M. P. Pera (Berlin: Springer), 35–102.

Manoonpong, P., Pasemann, F., Fischer, J., and Roth, H. (2005). Neural processing of auditory signals and modular neural control for sound tropism of walking machines. *Int. J. Adv. Rob. Sys.* 2, 223–234. doi: 10.5772/5786

Manoonpong, P., Pasemann, F., Kolodziejski, C., and Wörgötter, F. (2010). "Designing simple nonlinear filters using hysteresis of single recurrent neurons for acoustic signal recognition in robots," in *International Conference on Artificial Neural Networks* (Berlin: Springer), 74–383.

Marcus, C., and Westervelt, R. (1989). Dynamics of iterated-map neural networks. *Phys. Rev. A* 40:501. doi: 10.1103/physreva.40.501

Negrello, M. (2011). *Invariants of Behavior: Constancy and Variability in Neural Systems*. New York, NY: Springer Science and Business Media.

Negrello, M., and Pasemann, F. (2008). Attractor landscapes and active tracking: the neurodynamics of embodied action. *Adaptive Behav.* 16, 196–216. doi: 10.1177/1059712308090200

Nolfi, S., and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-organizing Machines*. Cambridge: MIT Press.

Pasemann, F. (1993a). Dynamics of a single model neuron. *Int. J. Bifurcat. Chaos* 2, 271–278.

Pasemann, F. (1993b). "Geometry of feedforward networks," in *Classical and Quantum Systems - Foundations and Symmetries*, eds H. D. Doebner, W. Scherer, and F. Schroeck (Singapore: World Scientific), 784–790.

Pasemann, F. (2002). Complex dynamics and the structure of small neural networks. *Network* 13, 195–216. doi: 10.1080/net.13.2.195.216

Pasemann, F., Rempis, C. W., and von Twickel, A. (2012). "Evolving humanoid behaviors for language games," in *Language Grounding in Robots*, eds L. Steels and M. Hild (New York, NY: Springer), 67–86.

Philipona, D., O'regan, J., Nadal, J.-P., and Coenen, O. (2004). Perception of the structure of the physical world using unknown sensors and effectors. *Adv. Neural Inf. Process. Syst.* 16, 945–952.

Philipona, D., O'Regan, J. K., and Nadal, J.-P. (2003). Is there something out there? inferring space from sensorimotor dependencies. *Neural computat.* 15, 2029–2049. doi: 10.1162/089976603322297278

Port, R. F., and Van Gelder, T. (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. Cambridge: MIT Press.

Respondek, W. (1996). "Geometric methods in nonlinear control theory," in *Neural Adaptive Control Technology*, eds R. Zbikowski and K. Hunt (London: World Scientific), 115–152.

Sándor, B., Jahn, T., Martin, L., and Gros, C. (2015). The sensorimotor loop as a dynamical system: how regular motion primitives may emerge from self-organized limit cycles. *arXiv preprint arXiv:1511.04338*.

Strogatz, S. H. (2014). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Boulder, CO: Westview Press.

Sussmann, H. J. (1983). Lie brackets, real analyticity and geometric control. *Differ. Geomet. Control Theory* 27, 1–116.

Thelen, E., and Smith, L. B. (1996). *A Dynamic Systems Approach to the Development of Cognition and Action*. Cambridge: MIT Press.

Thom, R. (1989). *Structural Stability and Morphogenesis*. Boston, MA: Addison Wesley Publishing Company.

Toutounji, H., and Pasemann, F. (2016). "Autonomous learning needs a second environmental feedback loop," in *Computational Intelligence*, eds J. J. Merelo, A. Rosa, J. M. Cadenas, A. D. Correia, K. Madani, A. Ruano, and J. Filipe (Berlin: Springer), 455–472.

Toutounji, H., and Pipa, G. (2014). Spatiotemporal computations of an excitable and plastic brain: neuronal plasticity leads to noise-robust and noise-constructive computations. *PLoS Comput. Biol.* 10:e1003512. doi: 10.1371/journal.pcbi.1003512

Von Foerster, H. (1960). "On self-organizing systems and their environments," in *Self-Organizing Systems*, eds M. C. Yovitts and S. Cameron (New York, NY: Pergamon), 31–50.

von Twickel, A., Büschges, A., and Pasemann, F. (2011). Deriving neural network controllers from neuro-biological data: implementation of a single-leg stick insect controller. *Biol. Cybern.* 104, 95–119. doi: 10.1007/s00422-011-0422-1

Wang, X. (1991). Period-doublings to chaos in a simple neural network: An analytical proof. *Complex Sys.* 5, 425–44.

Wilson, H. R., and Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* 12, 1.

Wischmann, S., and Pasemann, F. (2006). "The emergence of communication by evolving dynamical systems," in *International Conference on Simulation of Adaptive Behavior* (Berlin: Springer), 777–788.

Zahedi, K., Ay, N., and Der, R. (2010). Higher coordination with less controla result of information maximization in the sensorimotor loop. *Adapt. Behav.* 18, 338–355. doi: 10.1177/1059712310375314

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Pasemann. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

APPENDIX

The following is a list of parameters corresponding to the designated 2-neuron networks used for demonstrations in this paper.

TABLE A1 | Parameters for 2-neuron neural networks (Equation 2) discussed in the paper.

System	θ_1	θ_2	w_{11}	w_{12}	w_{21}	w_{22}
sys1	−2.0	0.0	0.0	1.6	−1.6	0.0
sys2	−1.6	0.0	−3.0	1.6	−1.6	0.0
sys3	0.0	0.0	1.1	0.6	−0.6	0.9
sys4	−3.0	0.0	−4.8.	2.3	−2.3	0.0
sys5	0.0	0.0	−1.5	1.5	1.5	1.5
sys6	−2.51	0.0	−2.75	1.5	−1.5	0.0
$Osci_1$	0.0	0.1	1.0	−1.0	1.0	1.0
$Osci_2$	0.0	0.5	1.0	−1.0	1.0	1.0

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 09 March 2017
doi: 10.3389/fnbot.2017.00011

An Adaptive Neural Mechanism for Acoustic Motion Perception with Varying Sparsity

*Danish Shaikh * and Poramate Manoonpong*

Embodied AI and Neurorobotics Laboratory, Centre for BioRobotics, Maersk Mc-Kinney Moeller Institute, University of Southern Denmark, Odense, Denmark

Biological motion-sensitive neural circuits are quite adept in perceiving the relative motion of a relevant stimulus. Motion perception is a fundamental ability in neural sensory processing and crucial in target tracking tasks. Tracking a stimulus entails the ability to perceive its motion, i.e., extracting information about its direction and velocity. Here we focus on auditory motion perception of sound stimuli, which is poorly understood as compared to its visual counterpart. In earlier work we have developed a bio-inspired neural learning mechanism for acoustic motion perception. The mechanism extracts directional information via a model of the peripheral auditory system of lizards. The mechanism uses only this directional information obtained via specific motor behaviour to learn the angular velocity of unoccluded sound stimuli in motion. In nature however the stimulus being tracked may be occluded by artefacts in the environment, such as an escaping prey momentarily disappearing behind a cover of trees. This article extends the earlier work by presenting a comparative investigation of auditory motion perception for unoccluded and occluded tonal sound stimuli with a frequency of 2.2 kHz in both simulation and practice. Three instances of each stimulus are employed, differing in their movement velocities–0.5°/time step, 1.0°/time step and 1.5°/time step. To validate the approach in practice, we implement the proposed neural mechanism on a wheeled mobile robot and evaluate its performance in auditory tracking.

Keywords: acoustic motion perception, binaural acoustic tracking, sound localisation, correlation-based learning, lizard peripheral auditory system

OPEN ACCESS

Edited by:
Shuai Li,
Hong Kong Polytechnic University,
Hong Kong

Reviewed by:
Xiaosu Hu,
University of Michigan, USA
Noman Naseer,
Air University, Pakistan
Muhammad Umer Khan,
Air University, Pakistan

****Correspondence:***
Danish Shaikh
danish@mmmi.sdu.dk

Received: *03 December 2016*
Accepted: *20 February 2017*
Published: *09 March 2017*

Citation:
Shaikh D and Manoonpong P (2017) An Adaptive Neural Mechanism for Acoustic Motion Perception with Varying Sparsity. Front. Neurorobot. 11:11. doi: 10.3389/fnbot.2017.00011

1. INTRODUCTION

Historically motion perception has been extensively studied in the context of visual tracking. This comes as no surprise as it is the dominant sense for humans and most animal species. In humans it plays an important role in visuomotor coordination tasks such as catching a ball (Oudejans et al., 1996). In the animal kingdom, motion perception is a crucial element that is relevant to sustenance and survival. It is particularly important in conditions where the target being tracked is sporadically occluded (Morgan and Turnbull, 1978) such as a predator tracking a moving prey that occasionally disappears from view behind trees.

A simple correlation-based neural circuit for motion detection in vision that selectively responds to direction and velocity given monocular visual input has been proposed decades ago by Reichardt (1969). Such low-level motion detectors however have not been reported for audition.

Auditory motion perception has therefore been suggested by Carlile and Leung (2016) to exist as a higher level system, similar to binocular, attention-modulated third-order visual motion detectors. The authors furthermore suggest that such third-order systems likely respond to snapshots of location information extracted from binaural cues. However, visual tracking experiments in the context of smooth eye pursuit that utilised periodic occlusion of the target indicate that target velocity may be a significant spatial information source (Barnes and Asselman, 1992; Churchland et al., 2003; Orban de Xivry et al., 2008).

Given a means to estimate a moving target's relative location and information regarding the time during which subsequent estimates are determined, the target's velocity can be derived. Here we demonstrate that the target velocity for continuous unoccluded as well as occluded acoustic targets could be learned based on the determination of these two pieces of information. We frame the problem of acoustic motion perception as an active acoustic tracking task. Active acoustic tracking entails movement of the acoustic organs to track an object, which is a natural auditory tracking behaviour. The dynamics of auditory tracking in cats with disconnected optical nerves, which disabled visual processing, have been behaviourally investigated (Beitel, 1999). The recorded head motion of these animals while tracking a series of click sounds emitted by a rotating loudspeaker suggested sound localisation being performed in a series of steps. The animals first displayed a rapid saccade-like head-orienting response to localise the target within the frontal sound field. This was followed by successive head movement cycles where the head would overshoot and pause, ensuring that the target's location remained close to the median plane.

1.1. Auditory Localisation Cues for Spatial Motion Perception

There are three types of cues available for auditory localisation– the difference in arrival times of a sound (interaural time difference or ITD), the difference in sound level (interaural level difference or ILD) and spectral information (direction-dependent energy minimisation over the entire frequency spectrum due to filtering by the outer ear). Several animals such as frogs, crickets and lizards utilise only ITD cues for sound localisation. For these animals spectral and ILD cues are unavailable due to lack of pinnae and the diffraction of sound around the head respectively. Using difference cues for localisation requires two ears with a frequency-dependent displacement between them. Generating ILD cues requires a sufficiently large head between the ears. The dimensions of the head should however be at least greater than the half-wavelength of the sound signal to successfully generate ILD cues. This creates an acoustic shadow inside which the relative sound amplitude is reduced. ITD cues can however be generated without the need of such obstructions, but do depend on the displacement between the ears and the angle of incidence of the sound with respect to the median plane. Here we restrict ourselves to acoustic tracking of a moving sound signal using only ITD cues extracted from microphones.

A sound signal moving in a given direction with a constant velocity with respect to the microphones generates dynamically varying ITD cues. The instantaneous values of these cues are dependent on the relative instantaneous position of the sound signal, while the rate with which they vary is dependent on the relative movement speed of the sound signal. Actively tracking a moving sound signal therefore requires transforming these relative position- and velocity-dependent cues into a desired behaviour, for example robotic orientation or phonotaxis. One must first determine the instantaneous spatial location of the sound signal to within the desired threshold of the instantaneous tracking error. This localisation must then be successively repeated sufficiently quickly to minimise the tracking error.

1.2. Relevance of Acoustic Motion Perception

There are several applications where actively tracking an acoustic target can be of interest. In robot phonotaxis applications, the robot could localise acoustic signals and navigate toward them (Reeve and Webb, 2003; Oh et al., 2008). In audio-visual teleconferencing systems, dynamically-steered microphone systems that automatically orient toward a speaker as they move about in a room could maximise the power of the incoming audio signal or orient a video camera toward the current speaker (Wang and Chu, 1997; Brandstein and Ward, 2001). Social robots that respond to sound and/or speech input from the human are another example. The verbal human-robot interaction element in social robots is deemed to be more natural and richer if the robot's acoustomotor response orients and maintains its gaze as well as auditory focus on the subject of interest (Nakadai et al., 2000; Okuno et al., 2003) in motion. For example, a human walks around in a room while addressing the robot via either directed or undirected speech commands.

Conventional acoustic tracking techniques (Liang et al., 2008a; Tsuji and Suyama, 2009; Kwak, 2011; Ju et al., 2012, 2013; Nishie and Akagi, 2013) are passive in that they require no movement of the listener. All of these techniques extract ITD cues for localisation by utilising multi-microphone arrays with at least four microphones. Typical arrays comprise an order of magnitude more microphones arranged in various geometric configurations such as linear, square, circular or in distributed arrays. ITD-based sound localisation and tracking techniques also tend to utilise computationally intensive algorithms such as particle filtering to compute the relative sound signal location from raw ITD data (Ward et al., 2003; Lehmann, 2004; Valin et al., 2007; Liang et al., 2008b; Ning et al., 2015). More conventional approaches are based on the generalised cross-correlation technique (Knapp and Carter, 1976) or the more recent steered response power technique (DiBiase, 2000; DiBiase et al., 2001; Zotkin and Duraiswami, 2004; Dmochowski et al., 2007; Cai et al., 2010; Wan and Wu, 2010; Marti et al., 2013; Zhao et al., 2013; Lima et al., 2015). Employing a larger number of microphones can improve localisation accuracy but at the expense of greater computational complexity and costly hardware for synchronisation and processing of multi-channel acoustic signals.

1.3. Contribution of the Present Work

We have previously reported a system for acoustic motion perception (Shaikh and Manoonpong, 2016) employing two microphones that implements a neural learning mechanism. The learning utilises a mathematical model that mimics the functionality of the auditory processing performed by the lizard peripheral auditory system (Wever, 1978). The system provides sound direction information and has been characterised via bio-faithful mathematical modelling (Zhang, 2009). The parameters of the model have been determined from biophysical data recorded from live lizards (Christensen-Dalsgaard and Manley, 2005). The model has also been implemented on a number of robotic platforms as reviewed in Shaikh et al. (2016). The neural learning mechanism has been adapted from the Input Correlation (ICO) learning approach (Porr and Wörgötter, 2006), which itself has been derived from a class of differential Hebbian learning rules (Kosko, 1986). The neural mechanism is considered to be a first step toward the development of a biologically-plausible neural learning mechanism for acoustic motion perception. The mechanism has been validated in simulation for tracking a continuous unoccluded acoustic signal moving with a constant and unknown angular velocity along a semi-circular trajectory. It has also been shown to learn various target angular velocities in separate simulated trials.

Here we implement the neural learning mechanism and compare its tracking performance for three different types of sound signals–continuous unoccluded, periodically occluded and randomly occluded. We first implement the neural mechanism in simulation that allows a robotic agent to learn to track a virtually-moving continuous unoccluded sound signal for a set of three different and unknown target angular velocities. As earlier the virtual sound signal is a pure tone moving along a semi-circular trajectory. To validate the tracking performance in practice, the learned synaptic weights representing a given target angular velocity are then used directly on a wheeled mobile robot that also implements the neural mechanism.

Next we implement another instance of the neural mechanism in simulation to learn to track a periodically occluded acoustic signal, moving with a constant but unknown angular velocity along a semi-circular trajectory. The occluded acoustic signal is implemented as an intermittent signal, i.e., it has a continuous unoccluded sound for a constant interval followed by complete silence for a constant interval. The silence implies that the signal is occluded and therefore inaudible. The acoustic tracking performance is evaluated in simulation for a constant "duty cycle" of sound emission. In this manner, the acoustic tracking performance is again evaluated for a set of three different target angular velocities identical to those used earlier. An instance of the simulation results is validated in practice via robotic trials with the wheeled mobile robot.

Finally, we implement a third instance of the neural mechanism in simulation to learn to track an occluded acoustic signal as described earlier, however with a randomly varying duty cycle. The signal moves as before with a constant but unknown angular velocity along a semi-circular trajectory. We evaluate the acoustic tracking performance for a set of three different target angular velocities identical to those used earlier. The main contribution of this work lies in systematically investigating the comparative performance of a neural closed-loop learning mechanism in learning the angular velocity of an acoustic stimulus with varying sparsity.

This article is organised in the following manner. Section 2 provides background information about the lizard peripheral auditory system and its equivalent model as well as about ICO learning. Section 3 presents the adaptive neural acoustic tracking architecture, the experimental setup and the robot model. Section 4 shows the experimental results in both simulation and practice. Section 5 summarises the work and discusses future directions.

2. BACKGROUND

2.1. The Lizard Peripheral Auditory System

The remarkable sensitivity of the peripheral auditory system (Christensen-Dalsgaard and Manley, 2005; Christensen-Dalsgaard et al., 2011) of lizards such as the bronze grass skink or *Mabuya macularia*, and the tokay gecko or *Gekko gecko* as depicted in **Figure 1A** is quite well understood. This "directionality" enables the animal to extract the relative position of a relevant sound signal. The lizard ear achieves a directionality higher than that of any known vertebrate (Christensen-Dalsgaard and Manley, 2005). This is due to an internal acoustical connection formed by efficient sound transmission through internal pathways in the head as depicted in **Figure 1B**, between the animal's two eardrums.

In spite of the peripheral auditory system's relatively small dimensions (the eardrums for most lizard species are separated by 10–20 mm), the range of sound wavelengths over which it exhibits strong directionality (Christensen-Dalsgaard et al., 2011) is relatively wide (340–85 mm, corresponding to 1–4 kHz). Within this range of frequencies the sound pressure difference between the eardrums is negligible due to acoustic diffraction around the animal's head, thus generating almost negligible (1–2 dB) ILD cues. The system thus relies on μs-scale interaural phase differences between incoming sound waves at the two ears due to the physical separation. These phase differences, corresponding to ITDs, are used extract information about sound direction relative to the animal. The system essentially converts these relatively tiny phase differences into relatively larger (up to 40 dB) interaural vibrational amplitude differences (Christensen-Dalsgaard and Manley, 2005). These amplitude differences encode sound direction information. Each eardrum's vibrations are the result of the superposition of two acoustic components generated due to sound interference in the internal pathways–an external sound pressure acting on the eardrum's periphery and an equivalent internal sound pressure acting on its interior. This leads to the ipsilateral (toward the sound signal) amplification of eardrum vibrations and contralateral (away from the sound signal) cancellation of eardrum vibrations. In other words, the ear closer to the relevant sound signal vibrates more strongly as compared to the ear further away from it. The relative phase difference between the incoming sound waves at the two eardrums determines the relative strengths of their vibrations.

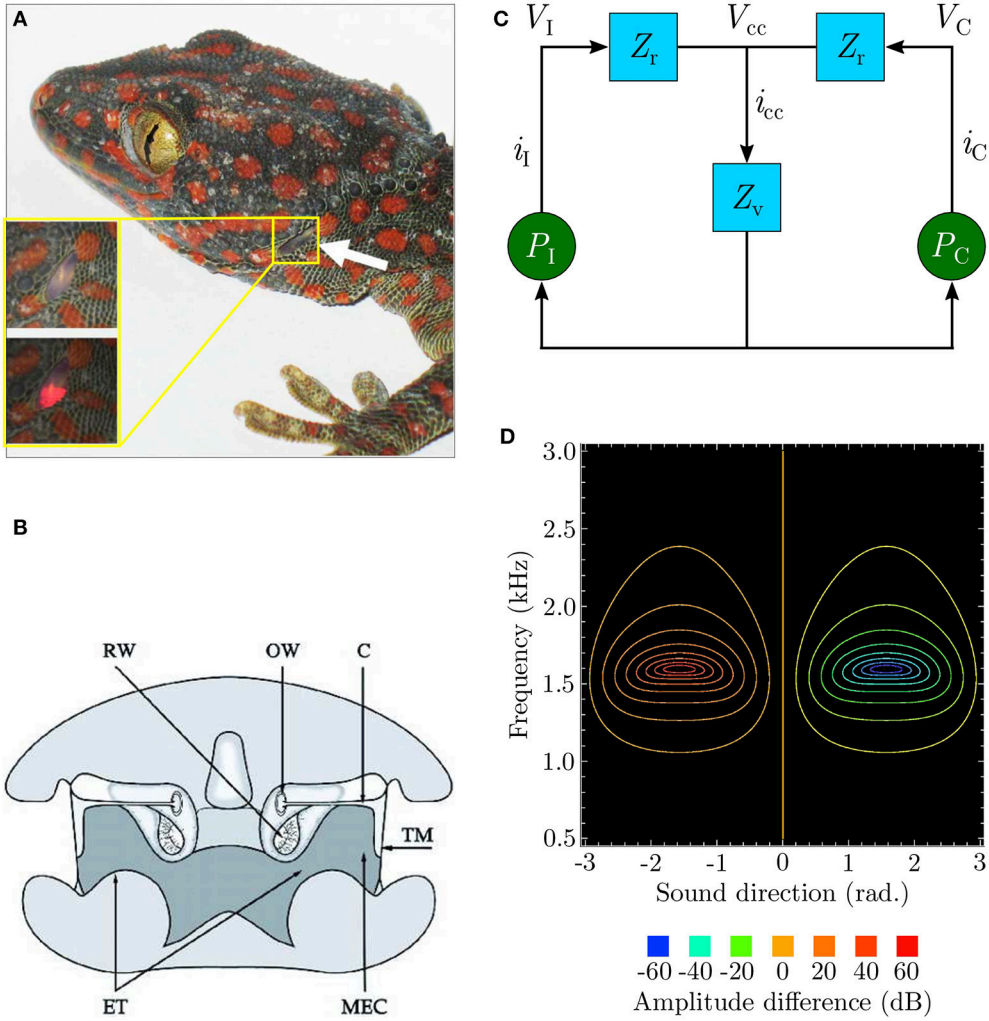


FIGURE 1 | (A) An eardrum visible on the side of the gecko head (redrawn from Christensen-Dalsgaard et al., 2011). **(B)** Early cross-sectional diagram of the lizard (*Sceloporus*) auditory system (taken from Christensen-Dalsgaard and Manley, 2005). **(C)** Ideal lumped-parameter circuit model (based on Fletcher and Thwaites, 1979; Fletcher, 1992 and redrawn from Zhang, 2009). Voltages V_I and V_C respectively represent sound pressures P_I and P_C at the ipsilateral and contralateral eardrums. Currents i_I and i_C, respectively represent the vibrations of the ipsilateral and contralateral eardrums due to the sound pressures acting upon them. Impedances Z_r model the combined acoustic filtering due to the mass of the eardrums and stiffness of the Eustachian tube through the central cavity connecting the tympani to each other. Impedance Z_v models the acoustic filtering effects of the central cavity itself. Voltage V_{cc} represents the resultant sound pressure in the central cavity due to the interaction of the internal sound pressures experienced from either side. This causes current i_{cc} to flow, representing the movement of sound waves inside the central cavity as the pressure inside it varies. **(D)** Contour plot (redrawn from Zhang, 2009) modelling binaural subtraction of the ipsilateral and contralateral responses as defined by Equation (2).

An equivalent electrical circuit model of the peripheral auditory system as depicted in **Figure 1C** (Fletcher and Thwaites, 1979; Fletcher, 1992) allows the directionality to be visualised as shown in **Figure 1D** as a difference signal computed by subtracting the vibrational amplitudes of the eardrums. Labelling the vibrational amplitudes of the ipsilateral and contralateral eardrums respectively as i_I and i_C, the difference signal can be formulated as

$$\left|\frac{i_I}{i_C}\right| = \left|\frac{G_I \cdot V_I + G_C \cdot V_C}{G_C \cdot V_I + G_I \cdot V_C}\right| , \quad (1)$$

where frequency-dependent gains G_I and G_C respectively model the effect of sound pressure on the motion of the ipsilateral and contralateral eardrum. These gains are essentially analogue filters in signal processing terminology with their coefficients determined experimentally from eardrum vibration measurements for individual lizards via laser vibrometry (Christensen-Dalsgaard and Manley, 2005). Expressing i_I and i_C in decibels,

$$i_{ratio} = 20\left(\log|i_I| - \log|i_C|\right)\ \text{dB}\,. \quad (2)$$

The model responds well for sound frequencies within the range 1–2.2 kHz, with a peak response at approximately 1.6 kHz. i_{ratio} is positive for $|i_{\text{I}}| > |i_{\text{C}}|$ and negative for $|i_{\text{C}}| > |i_{\text{I}}|$. The model's symmetry implies that the model's response $|i_{\text{ratio}}|$ is identical on either side of the centre point $\theta = 0°$ as well as locally symmetrical within the sound direction range $[-90°, +90°]$ (considered henceforth as the range of interest of sound direction). The difference signal expressed as Equation (2) provides information about sound direction in that its sign indicates whether the sound is coming from the ipsilateral side (positive sign) or from the contralateral side (negative sign), while its magnitude corresponds to the relative angular displacement of the sound signal with respect to the median.

2.2. Input Correlation (ICO) Learning

Since the proposed neural mechanism is derived from the ICO learning algorithm (Porr and Wörgötter, 2006), this section gives a brief introduction to the algorithm. The algorithm, depicted as a neural mechanism in **Figure 2**, is online unsupervised learning. Its synaptic weight update is driven by cross-correlation of two types of input signals–one or multiple "predictive" signal(s) which are stimuli occurring earlier in time and a "reflex" signal which is a stimulus occurring later in time, that arrives after a finite delay and drives an unwanted response or reflex. The learning goal of ICO learning is to predict the occurrence of the reflex signal by utilising the predictive signal. This allows an agent to react earlier, before the reflex signal occurs. The agent essentially learns to execute an anticipatory action to avoid the reflex.

The output O_{ICO} of the ICO learning mechanism is a linear combination of the reflex input x_0 and the N predictive input(s) x_{k} where $\text{k} = 1, \ldots, N$ and $N \in \mathbb{N}$. O_{ICO} is formulated as

$$O_{\text{ICO}} = \rho_0 x_0(t) + \sum_{\text{k}=1}^{N} \rho_{\text{k}}(t) x_{\text{k}}(t) \,. \tag{3}$$

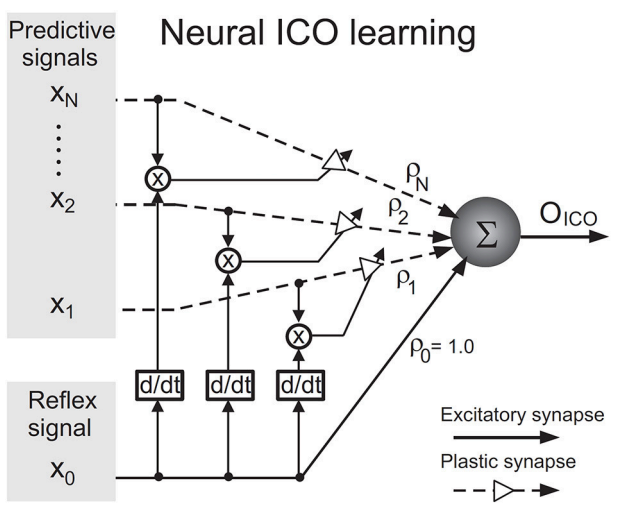


FIGURE 2 | Neural circuit for input correlation learning (taken from Manoonpong et al., 2013).

The synaptic weight ρ_0 of the reflex input is assigned a constant positive value such as 1.0, representing a reflex signal whose strength does not change over time. During learning, the synaptic weight(s) ρ_{k} of the predictive signal(s) $x_{\text{k}}(t)$ are updated through differential Hebbian learning (Kosko, 1986; Klopf, 1988) using the cross-correlation between the predictive and reflex inputs. The synaptic weight update rule is given by

$$\frac{d\rho_{\text{k}}(t)}{dt} = \mu x_{\text{k}}(t) \frac{dx_0(t)}{dt} \,, \text{k} = 1, \ldots, N \,. \tag{4}$$

The learning rate μ, usually set to a value less than 1.0, determines how fast the neural mechanism can learn to avoid the reflex signal from occurring. The synaptic weights ρ_{k} tend to stabilise when the reflex signal is nullified, which implies that the reflex signal has been successfully avoided. ICO learning is characterised by its fast learning speed and stability of synaptic weight updates and has been successfully applied to real robots to generate adaptive behaviour (Manoonpong et al., 2007; Porr and Wörgötter, 2007; Manoonpong and Wörgötter, 2009).

3. MATERIALS AND METHODS

We define the task of acoustic tracking as follows–a robotic agent must learn to track a moving acoustic signal. The robot learns the target's angular velocity by matching it with its (the robot's) own angular turning velocity. The correct angular turning velocity should allow the agent to rotate along a fixed axis sufficiently quickly so as to align itself toward the instantaneous position of the acoustic signal. The signal is moved in the horizontal plane along a pre-defined semi-circular arc-shaped trajectory with an unknown velocity in an unknown but fixed direction. To solve this task we employ an adaptive neural architecture (Shaikh and Manoonpong, 2016) that combines the auditory preprocessing of the lizard peripheral auditory model with a neural ICO-based learning mechanism.

3.1. The Neural Architecture

The neural mechanism is embedded within the task environment as a closed-loop circuit as depicted in **Figure 3**. The goal of the learning algorithm is to learn the temporal relationship between the perceived position of the target sound signal *before* turning and *after* turning. The synaptic weights of the neural mechanism encode this temporal relationship and they can then be used to calculate the correct angular turning velocity. A given set of learned synaptic weights can however only represent a given angular velocity. This is because the temporal relationship between the perceived position of the target sound signal *before* turning and *after* turning depends on the angular turning velocity. Therefore, the synaptic weights must be re-learned to obtain a new angular turning velocity.

The output of the neural mechanism is the angular velocity ω, defined as the angular deviation per time step, required to turn the robot quickly enough to orient toward the target sound signal in one time step. The rotational movements of the robotic agent translate ω into corresponding ITD cues. The peripheral auditory model (PAM), based on these cues,

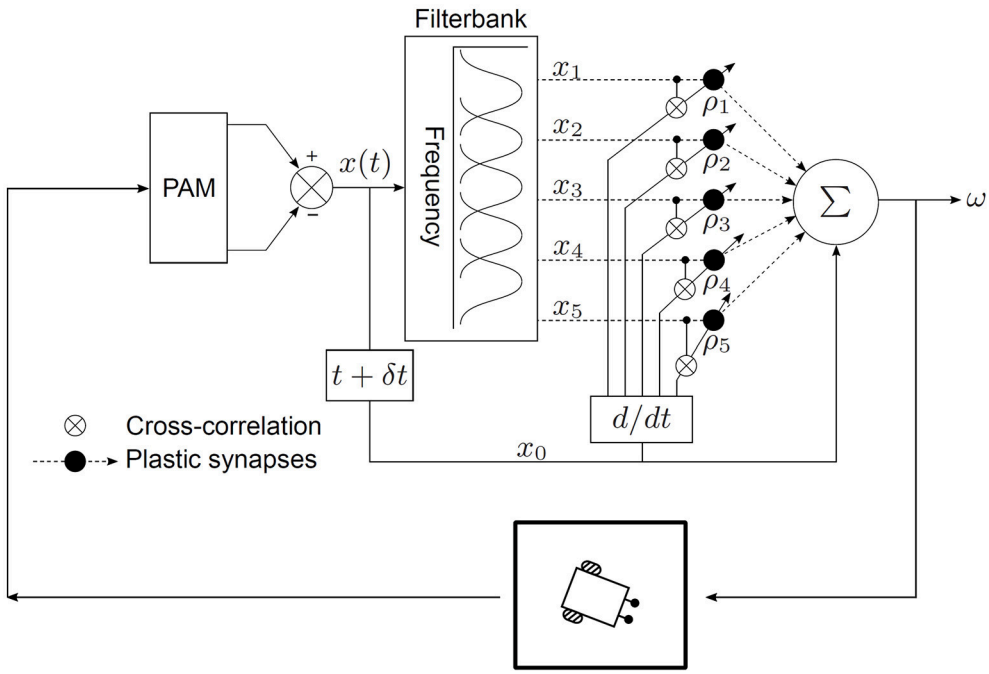


FIGURE 3 | Neural mechanism for acoustic tracking as a closed-loop system.

computes a difference signal $x(t)$ which encodes information regarding sound direction. Practically, $x(t)$ is the difference between the modelled vibrational amplitudes of the left and right eardrums in response to sound input, i.e., it is essentially i_{ratio} as defined by Equation (2). A filter bank decomposes $x(t)$ into sound frequency-dependent components $x_{\text{k}}(t)$, where $\text{k} = 1, \ldots, N$, to extract frequency information. Each of these components encodes the extracted sound direction information within a specific frequency band. Practically, these components are the difference between the modelled vibrational amplitudes of the left and right eardrums in response to sound input. This step is necessary since the peripheral auditory model provides ambiguous information regarding the sound direction in the absence of sound frequency information. The ambiguity is a result of the difference signal $x(t)$ having identical values for multiple positions of the sound signal if the sound frequency is unknown (see **Figure 1D**). The filter bank comprises five bandpass filters. The centre frequencies of these filters lie at 1.2, 1.4, 1.6, 1.8, and 2.0 kHz within the relevant response range. Each filter has a 3 dB cut-off frequency of 200 Hz. This results in $N = 5$ filtered difference signals at the output of the filter bank. The magnitude responses of the individual filters in the filter bank represent the receptive fields of individual auditory neurons. These spectro-temporal receptive fields (Aertsen et al., 1980) are essentially the range of sound frequencies over which the neurons are optimally stimulated. The filtered difference signals $x_{\text{k}}(t)$ are then used as inputs that are correlated with the derivative of the unfiltered difference signal $x_0(t)$. The input signals $x_{\text{k}}(t)$ represent the earlier-occurring predictive stimuli used to estimate the instantaneous sound direction before turning, while the unfiltered difference signal $x_0(t)$ represents the later-occurring "reflex" stimuli or the retrospective signal generated after turning.

In traditional ICO learning the synaptic weights are stabilised once the reflex signal is nullified, thereby creating a behavioural response that prevents future occurrences of the reflex signal. In our case, as soon as the target sound signal moves to a new position along its trajectory, a new and finite retrospective signal x_0 corresponding to the new position is generated. This signal is then nullified after turning if the correct synaptic weights have been learned, and then the target sound signal moves to a new position along its trajectory. Our approach can therefore be considered as one successful step of ICO learning being successively repeated for each new position of the target sound signal as it moves along its trajectory. This implies that the synaptic weights can grow uncontrollably if the learning is allowed to continue indefinitely. A stopping criterion for the learning was therefore introduced to avoid this condition–the learning stops when the tracking error θ_e becomes less than 0.5°. θ_e is defined as the difference between the orientation of the robot and the angular position of the sound signal in *one* time step. In other words, the learning stops when the robot is able to orient itself toward a position that is within 0.5° from the position of the sound signal within *one* time step.

3.2. The Experimental Setup

The experimental setup in simulation comprises a virtual loudspeaker array as depicted in **Figure 4** which generates relevant pure tone sounds at a 2.2 kHz frequency. This frequency is chosen because sufficient directional information from the peripheral auditory model is available at this frequency. The

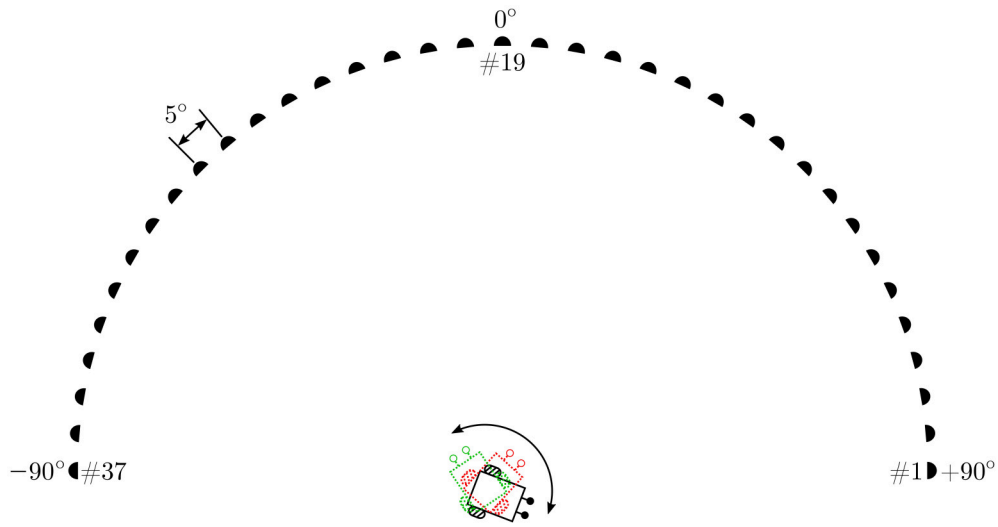


FIGURE 4 | The simulation setup.

array consists of 37 loudspeakers numbered #1–#37 from right to left, arranged in a semi-circle in the azimuth plane. The angular displacement between consecutive loudspeakers is 5°. The loudspeakers are turned on sequentially, starting from the loudspeaker at one of the ends of the array, to simulate the motion of a continuously moving sound signal (albeit in discrete steps). To maintain the continuity of the sound the next loudspeaker plays immediately after the previous loudspeaker has stopped. A given tone can therefore be moved with a given angular velocity across the array along a semi-circular trajectory from either the left or the right side. The angular velocity of the sound signal is defined as the angular displacement in radians per time step. A given loudspeaker, when turned on, plays a tone for 10 time steps before turning off and at the same instant the next consecutive loudspeaker turns on. This process is repeated until the sound reaches the last loudspeaker in the array. The movement of sound from loudspeaker #1 to loudspeaker #37 is defined as one complete learning iteration. Since one iteration may be insufficient to learn the correct angular velocity of the target sound signal, the learning is repeated over multiple iterations until the stopping criterion is met. After the completion of one learning iteration, the sound signal starts again from loudspeaker #1 in the next learning iteration. The direction of movement of sound is chosen to be from the right side (+90°) to the left side (−90°) of the array.

The robot that should track the moving target sound signal is positioned at the mid-point of the diameter of the semi-circle and is only allowed to rotate in the azimuth plane along a fixed axis. The robot must turn with a sufficiently large angular turning velocity to orient toward the instantaneous position of sound signal before the sound signal moves to the next position along its trajectory. The angular velocity of the robot is defined as the angular rotation in radians per time step. The goal of the learning algorithm is to learn the correct angular velocity that allows the robot to turn and orient toward the current loudspeaker in *one* time step, starting from the time step at which that loudspeaker started playing the tone.

The learning at every time step occurs as follows. The robotic agent is initially oriented toward a random direction toward the right side of the array. Loudspeaker #1 emits a tone and the robot uses the sound direction information extracted by the peripheral auditory model to turn toward the currently playing loudspeaker with an angular velocity ω (computed using the initial values of the synaptic weights) given by

$$\omega = \rho_0 x_0 + \sum_{k=1}^{N} \rho_k x_k, \text{ where } N = 5. \tag{5}$$

After the turn is complete, the robot once again extracts sound direction information via the peripheral auditory model and computes the retrospective signal $x_0(t + \delta t)$. The strength of $x_0(t + \delta t)$ depends on the relative position of the sound signal with respect to the orientation of the robotic agent after it has performed a motor action in the task environment. Therefore, this retrospective signal acts as the feedback information that is used to update the synaptic weights.

The synaptic weights ρ_k are then updated according to the learning rule

$$\frac{d\rho_k(t)}{dt} = \mu x_k(t) \frac{dx_0(t)}{dt}, \text{ where } k = 1, \ldots, N. \tag{6}$$

After 10 time steps loudspeaker #1 is deselected and the next loudspeaker in the array (loudspeaker #2) is selected. This learning procedure is repeated for all loudspeakers in succession.

We use three different angular velocities for the sound signals–0.5°/time step, 1.0°/time step and 1.5°/time step. These values were chosen primarily because the loudspeaker array in the experimental setup in practice is restricted to sound signal displacements that are multiples of 5°. The neural parameters

for all trials are set to the following values–the learning rate $\mu = 0.0001$ and synaptic weight for the retrospective signal x_0, $\rho_0 = 0.00001$. All plastic synaptic weights ρ_k are initially set to zero and updated according to Equation (6).

We first implement a new instance of the neural learning mechanism in simulation. The mechanism allows a robotic agent to learn the synaptic weights required to track a continuous unoccluded sound signal in simulation. The initial orientation of the robotic agent is randomly chosen to be 116° to emphasise that the learning is independent of any specific initial orientation. The continuous unoccluded sound can be viewed as a sound with 100% sound emission duty cycle, i.e., there are no breaks in the sound emission. We evaluate the acoustic tracking performance in simulation for a set of three different target angular velocities–0.5°/time step, 1.0°/ time step and 1.5°/time step. We then verify the simulation results in practice for a target angular velocity of 1.5°/time step by recreating the experimental setup in the form of robotic trials. We employ a wheeled mobile robot, as described in Section 3.3, to track a continuous unoccluded pure tone sound signal that is moved along a semi-circular virtual loudspeaker array as depicted in **Figure 5**. The array has an identical configuration as the one used in the simulation setup and is located in a sound-dampening chamber to minimise acoustic reflections. The synaptic weights used on the robot are those learned offline in simulation.

We then use another identical instance of the neural mechanism in the same simulation setup as before to learn to track a virtual pure tone sound signal that is periodically occluded, i.e., it is structured as a constant sound for a constant interval followed by complete silence for a constant interval. This sound emission duty cycle is set to 60%. The target sound signal again moves with a constant but unknown angular velocity along a semi-circular trajectory as described earlier. The initial orientation of the robotic agent in simulation is randomly chosen to be 97° to emphasise that the learning is independent of any specific initial orientation. In this manner, the acoustic tracking performance is evaluated in simulation for a set of three different target angular velocities–0.5°/time step, 1.0°/time step and 1.5°/time step. The simulation results are validated for a target angular velocity of 1.5°/time step in practice via robotic trials with the mobile robot as described earlier. The synaptic weights used on the robot are again those learned offline in simulation.

Finally, we use a third instance of the neural mechanism in simulation to learn to track a virtual pure tone sound signal that is occluded as described earlier but with a randomly varying duty cycle of sound emission. During learning, for every loudspeaker the sound emission duty cycle is chosen from a uniform random distribution between 10 and 90%. As before, the target sound signal moves with a constant but unknown angular velocity along a semi-circular trajectory as described earlier. The initial orientation of the robotic agent in simulation is randomly chosen to be 97°, once again to emphasise that the learning is independent of any specific initial orientation. We evaluate the acoustic tracking performance in simulation for a set of three different target angular velocities–0.5°/time step, 1.0°/time step and 1.5°/time step. We once again validate the simulation results in practice for a target angular velocity of 1.5°/time step on the mobile robot as described earlier. The sound emission duty cycles for each loudspeaker in the robotic trial are again randomly chosen from a uniform random distribution between 10 and 90%. This implies that the sequence of duty cycles is not identical to that used in the simulated trials. As earlier, the synaptic weights used on the robot are those learned offline in simulation.

3.3. The Robot Model

Figures 6A,B respectively depict the mobile robot used in the robotic trials and its kinematics. The basic platform is assembled

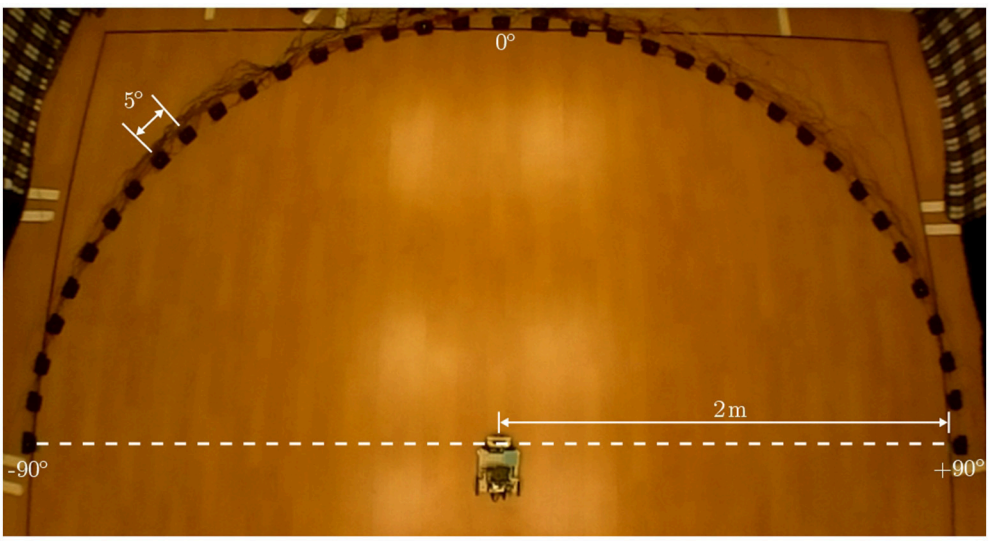


FIGURE 5 | The experimental arena.

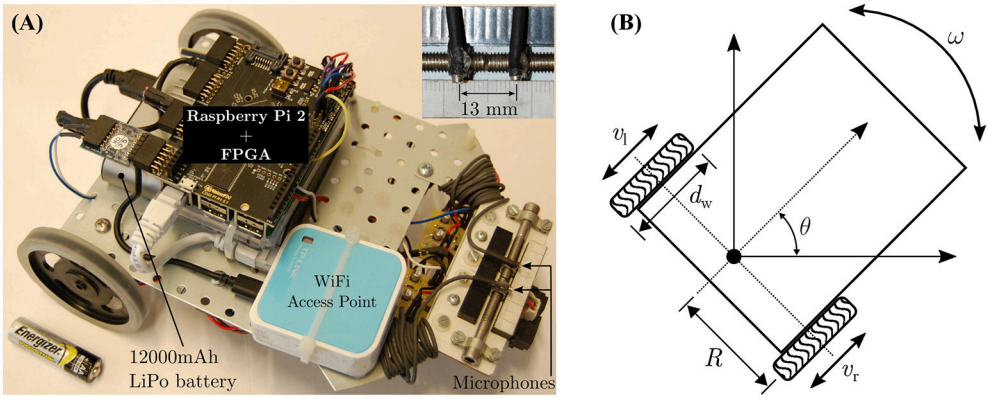


FIGURE 6 | (A) The mobile robot and **(B)** its kinematics.

with components from the Robotics Starter Kit from Digilent Inc.–the chassis, the DC motors (6 V), the corresponding H-bridge motor drivers, the rear wheels and a front omnidirectional ball caster wheel. The peripheral auditory model and the neural mechanism is implemented on a Raspberry Pi 2 (Model B+ from the Raspberry Pi Foundation) controller, which is paired with a FPGA board (model LOGI Pi from ValentFX). A dual channel analogue-to-digital (ADC) driver is implemented on the FPGA IC (Integrated Circuit) using the VHDL (VHSIC Hardware Description Language) programming language (VHSIC stands for Very-High-Speed Integrated Circuits). The VHDL design for the ADC driver is synthesised or compiled via a proprietary software tool (Xilinx Integrated Synthesis Environment or ISE from Xilinx Inc.) into a hardware-level binary "bitstream" containing all the necessary information to properly configure and program the logic into the FPGA chip. The driver reads in raw audio data from a dual channel 12-bit simultaneous ADC that digitises the signals from two omnidirectional microphones (model FG-23329-P07 from Knowles Electronics LLC) mounted 13 mm apart at the front of the robot (see inset in **Figure 6**). Since the peripheral auditory model's parameters have been derived from laser vibrometry measurements from a lizard with 13 mm separation between its eardrums, the microphone separation must match that value. Any other separation would create a mismatch between the ITD cues to which the peripheral auditory model is tuned and the actual ITD cues. A WiFi access point (model TL-WR802N from TP-LINK Technologies Co. Ltd.) allows wireless access to the robot controller for programming purposes. A 12,000 mAh lithium polymer power bank (model Xtorm AL450 from A-solar bv) serves as the power source for the robot.

The robot's kinematics are used to convert the learned angular rotation in degrees per second into the rotational speed in revolutions per minute (rpm) for the robot's wheels. One time step in simulation corresponds to 0.2 s, such that a learned angular turning velocity of θ degrees per time step implies that the robot should turn by θ degrees in a time period $t_\theta = 0.2$ s. This value is chosen because in the experimental setup the software controlling the loudspeaker array can only switch between consecutive loudspeakers at least every 2 s. Since the robot can only rotate along a fixed axis, the wheels travel along a semi-circular arc of length $\widehat{\mathrm{L}}$ when the robot performs a turn. Therefore, an angular displacement of θ degrees corresponds to the arc length $\widehat{\mathrm{L}}$ in millimetres as given by

$$\widehat{\mathrm{L}} = 2\pi R \frac{\theta}{360^\circ}, \tag{7}$$

where R is the radius in millimetres of the arc along which the wheels travel, and is essentially the distance between the centre of rotation of the robot and the centre of either wheel. Assuming v_l and v_r as the rotational velocities of the left and right wheels respectively, to rotate through an arc length $\widehat{\mathrm{L}}$ the two wheels must turn with identical angular velocities $|v_\mathrm{l}| = |v_\mathrm{r}| = v$ but in opposite directions (to perform a leftward rotation, v_l is considered as having a negative value and v_r is considered as having a positive value). The angular velocity $\omega = \frac{\widehat{\mathrm{L}}}{t_\theta}$ mm/s. The wheel rotational velocity v in rpm is given by

$$\begin{aligned} v &= \frac{\widehat{\mathrm{L}}}{t_\theta} \cdot \frac{1}{\pi d_\mathrm{w}} \cdot 60\,\mathrm{s} \\ &= 2\pi R \frac{\theta}{360^\circ} \cdot \frac{1}{t_\theta} \cdot \frac{1}{\pi d_\mathrm{w}} \cdot 60\,\mathrm{s}, \end{aligned} \tag{8}$$

where d_w is the diameter of the wheel. For the robot, R is measured to be 80 mm and d_w is measured to be 70 mm. Substituting for R, d_w and t_θ into Equation (8), the mathematical conversion between the robot's angular velocity in degrees per second into the corresponding wheel velocity v in rpm can be formulated as

$$v = 2\pi \cdot 80\,\mathrm{mm} \cdot \frac{\theta}{360^\circ} \cdot \frac{1}{0.2\,\mathrm{s}} \cdot \frac{1}{\pi 70} \cdot 60\,\mathrm{s} \approx 3.81 \cdot \theta. \tag{9}$$

Using Equation (9), the wheel velocities required by the robot corresponding to the three angular velocities 0.5°/time step, 1.0°/time step and 1.5°/time step are calculated to be approximately 19 rpm, 38 rpm and 57 rpm respectively. These rpm values represent the no-load wheel velocities, i.e., when

the DC motor shafts experience zero load. In practice these "ideal" rpm values will be adversely affected by the weight of the robot, the friction between the wheels and the ground and the instantaneous battery capacity. To approach real-life motion constraints during tracking, the effects of these physical quantities are deliberately not modelled. The speed commands for the wheels are therefore manually matched to the corresponding wheel velocities under load. This is done by making the robot perform an on-the-spot turn on the ground in the experimental arena, and determining via trial and error the speed command (which is the duty cycle for the signals controlling the motor drivers) for which the wheels complete the necessary revolutions in 1 min. This ensures that the effects of the aforementioned quantities are taken into account while the robot is tracking the sound signal during the robotic trials. Furthermore, there may be a mismatch between the characteristics of the individual DC motors of the robot. This may result in a mismatch between the angular velocities of the motor shafts even though both motors receive identical speed commands. To compensate for any potential mismatch, the robot is once again made to perform on-the-spot turns on the arena floor and the speed commands were fine-tuned via trial and error to generate turns of 0.5°, 1.0°, and 1.5° in 0.2 s.

Video footage of the robotic trials was recorded from an overhead camera (Raspberry Pi camera module from the Raspberry Pi Foundation). The footage was analysed with a video analysis software tool (Tracker version 4.95 from Open Source Physics (Open Source Physics, 2016) to determine the amount by which the robot turned for each loudspeaker. The robot's rotation angles were extracted by manually tracking a green LED (Light Emitting Diode) on the robot. The tracking was done for relevant video frames in which the robot was completely still after completing a turn, to determine its deviation from the reference. **Figure 7** depicts a screenshot of Tracker software.

4. RESULTS AND DISCUSSION

4.1. Simulation Trials

Figure 8 depicts the evolution of the tracking error θ_e during learning for a target angular velocity of 1.5°/time step as an example. Corresponding data for target angular velocities of 1.0°/time step and 0.5°/time step is illustrated respectively in Figures 8-1, 8-2 (see files "image1.pdf" and "image2.pdf" respectively in the Supplementary Materials). The insets show θ_e for a single iteration as an example. θ_e reduces exponentially over time for all three types of acoustic stimuli–continuous unoccluded sound (see **Figure 8A**), periodically occluded sound with a 60% sound emission duty cycle (see **Figure 8B**) and randomly occluded sound with a random sound emission duty cycle (see **Figure 8C**).

The spikes in θ_e as visible in the insets are a result of a mismatch between the last angular position toward which the robotic agent was pointing and the new angular position of the target sound signal as it moves along its trajectory. This mismatch generates finite ITD cues from which the robotic agent extracts sound direction information using the peripheral auditory model. The robotic agent then turns toward the sound signal with the last learned angular turning velocity, thereby

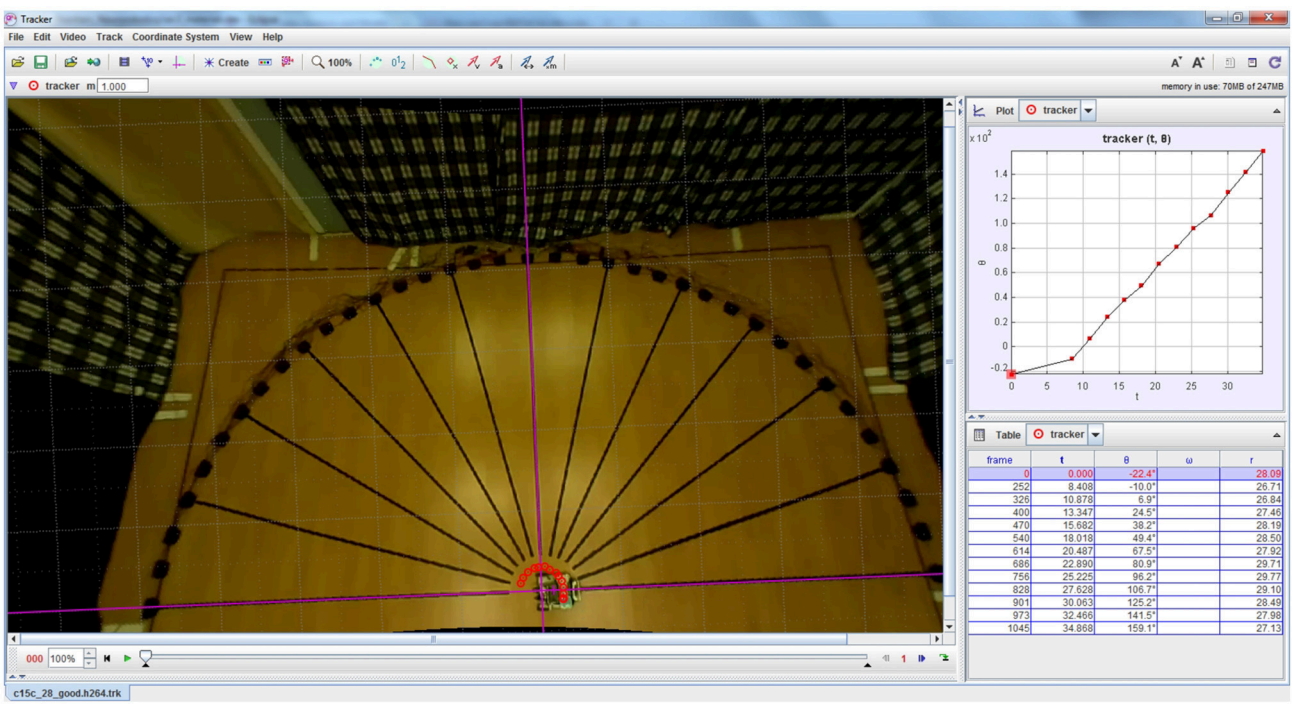


FIGURE 7 | Example screenshot of Tracker software used for extracting the robot's turning angles from overhead video footage of the robotic trials. The red circles indicate the location of the LED on the robot that is used for computing its angular rotation.

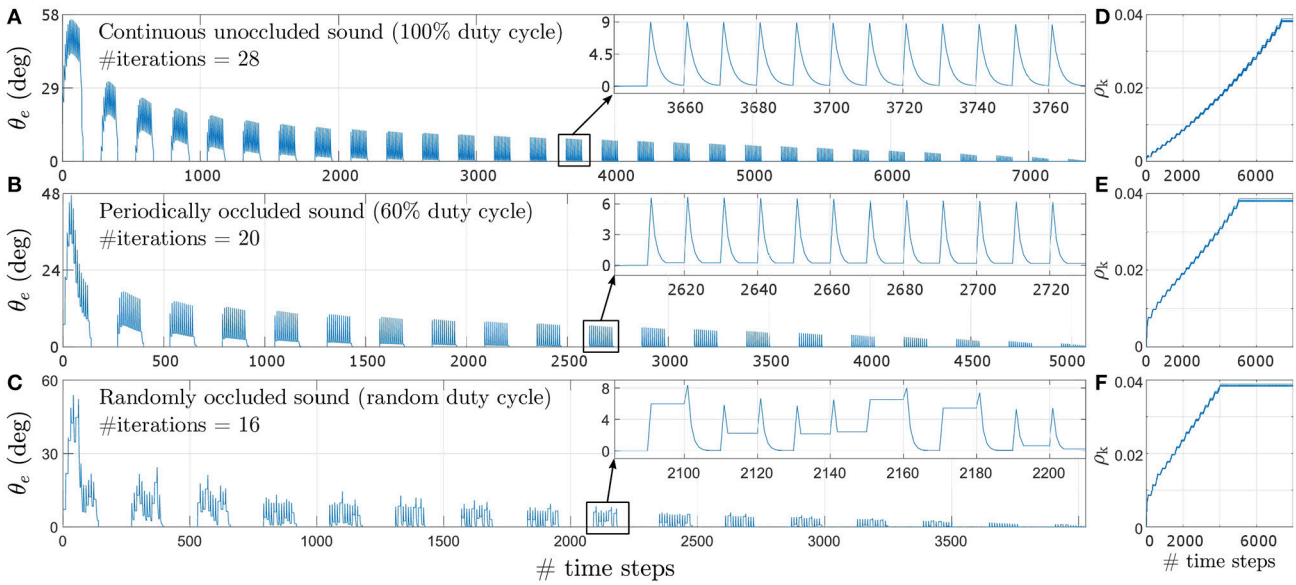


FIGURE 8 | Tracking error θ_e for a target angular velocity of 1.5°/time step for varying duty cycles of sound emission. (A) Continuous unoccluded sound. **(B)** Periodically occluded sound with 60% duty cycle. **(C)** Randomly occluded sound with random duty cycle. The insets show θ_e for a single iteration as an example. **(D)** Synaptic weights for continuous unoccluded sound. **(E)** Synaptic weights for periodically occluded sound with 60% duty cycle. **(F)** Synaptic weights for randomly occluded sound with random duty cycle.

reducing the maximum tracking error. This process repeats for each time step, exponentially reducing the overall tracking error, until the stop criterion is met.

The number of iterations required for the synaptic weights to converge toward their final values is relatively lower for sparse or occluded sound signals as compared to unoccluded sound signals. For an occluded signal, the number of time steps for which the sound is emitted per loudspeaker decreases. Consequently, the number of weight updates per loudspeaker also decreases. For example, for a 60% duty cycle, the weights are updated is six of the ten time steps per loudspeaker. When the loudspeaker stops playing there is no sound input and the peripheral auditory model's outputs are balanced, resulting in the difference signal $x(t)$ being nullified. This implies that both the retrospective signal $x_0(t)$ and the predictive signal $x_k(t)$ become zero. The weight increment given by the update rule in Equation (6) is therefore also zero when there is no sound present. From the perspective of the robotic agent's behaviour, this implies that the robotic agent does not move in the absence of sound. This is because there is no directional information available and the robotic agent "assumes" that it is already oriented toward the target sound position. Thus for occluded sound signals the robotic agent takes relatively fewer turns for each loudspeaker as compared to the number of turns taken for each loudspeaker for the unoccluded sound signal. This means that when the sound moves to a new position along its trajectory, the tracking error is relatively larger for occluded sound signals as compared to the unoccluded sound signal. This implies a relatively greater mismatch between the actual angular position of the loudspeaker and the orientation of the robotic agent and therefore relatively larger values for both the predictive and retrospective signals. Consequently, the synaptic weight update is also relatively larger for occluded sound signals as compared to unoccluded sound signal from the very first iteration as evident from **Figures 8D,E**. These large changes at the very beginning of the learning bring the synaptic weights relatively closer to their optimal values earlier in the learning process, and thus fewer subsequent iterations are required to bring the weights to their optimal values. Therefore, for a given target angular velocity the total number of iterations required for the synaptic weights to converge decreases for occluded sound signals as compared to the unoccluded sound signal.

The change in tracking error θ_e for a pure tone sound signal that is randomly occluded with a sound emission duty cycle between 10 and 90% for each loudspeaker is depicted in **Figure 8C** for the target angular velocity of 1.5°/time step. The insets show θ_e for a single iteration as an example. The uneven spikes visible in the insets indicate that the tracking error θ_e is different for different angular positions of the target sound signal. For each new angular position of the target sound signal, the learning algorithm increments the synaptic weights corresponding to the randomly selected sound emission duty cycle currently in effect for that angular position. Therefore, the synaptic weight increments are also random as evident in **Figure 8F**. As discussed earlier, a relatively smaller sound emission duty cycle results in relatively fewer weight updates. This implies that when the current duty cycle is relatively small, the robotic agent makes relatively fewer turns and thus the tracking error may only decrease to a finite non-zero value. When the target sound signal subsequently moves to the new

consecutive angular position along its trajectory, the tracking error increases again due to mismatch between the last angular position estimated by the robotic agent and the new angular position of the target sound signal. The amount of mismatch depends on the last learned angular turning velocity of the robotic agent. This in turn depends on the sound emission duty cycle for the last target angular position and that for the current target angular position. If the new sound emission duty cycle is relatively larger than the last one then there are relatively more weight updates. The tracking error may either reduce to zero or to another finite but non-zero value for that particular target angular position.

As an example, the relationship between the predictive signal $x_5(t)$ and the derivative of the retrospective signal $\frac{dx_0(t)}{dt}$ and the corresponding weight updates can be seen in **Figure 9** over one iteration of the learning. In the figure the sound signal is moving with an angular velocity of 1.5°/time step. Similar relationships corresponding to target angular velocities of 1.0°/time step and 0.5°/time step are illustrated respectively in Figures 9-1, 9-2 (see files "image3.pdf" and "image4.pdf" respectively in the Supplementary Materials). The respective weight updates (normalised for comparison) for all three types of acoustic stimuli–continuous unoccluded sound (see **Figure 9D**), periodically occluded sound with a 60 % sound emission duty cycle (see **Figure 9E**) and randomly occluded sound with a random sound emission duty cycle (see **Figure 9F**)–reflect the dependence of the size of the weight increments on the sound emission duty cycle as discussed earlier. The small initial spikes seen in the weight updates are a result of the $\frac{dx_0(t)}{dt}$ term in Equation (6) being initially positive and then becoming negative in the subsequent time step. The retrospective signal $x_0(t)$ is first positive due to the sound signal moving further away from the robotic agent. In the subsequent time step the robot reacts by turning toward the sound signal, thereby reducing $x_0(t)$. This results in the derivative $\frac{dx_0(t)}{dt}$ being negative. This leads to a negative weight increment which decreases the weight in the subsequent time steps after the spike. The term $\frac{dx_0(t)}{dt}$ becomes negative because the robot always turns toward the sound signal, which reduces $x_0(t)$.

A more thorough investigation of the effect of decreasing sound emission duty cycle on the number of iterations required to learn the target angular velocity within the given error bounds is depicted in **Figure 10**. The number of iterations required for the synaptic weights to converge decreases with decreasing sound emission duty cycle, i.e., with increasing sparsity of sound stimulus as described earlier.

The number of iterations required for the synaptic weights to converge also decreases for increasing angular velocity of the sound signal. This can be seen in Figures 8-1, 8-2 (see files "image1.pdf" and "image2.pdf" respectively in the Supplementary Materials). For increasing target angular velocity, the mismatch between the angular position toward which the robot was oriented after its last turn and the current position of the sound signal is relatively greater. This results in relatively

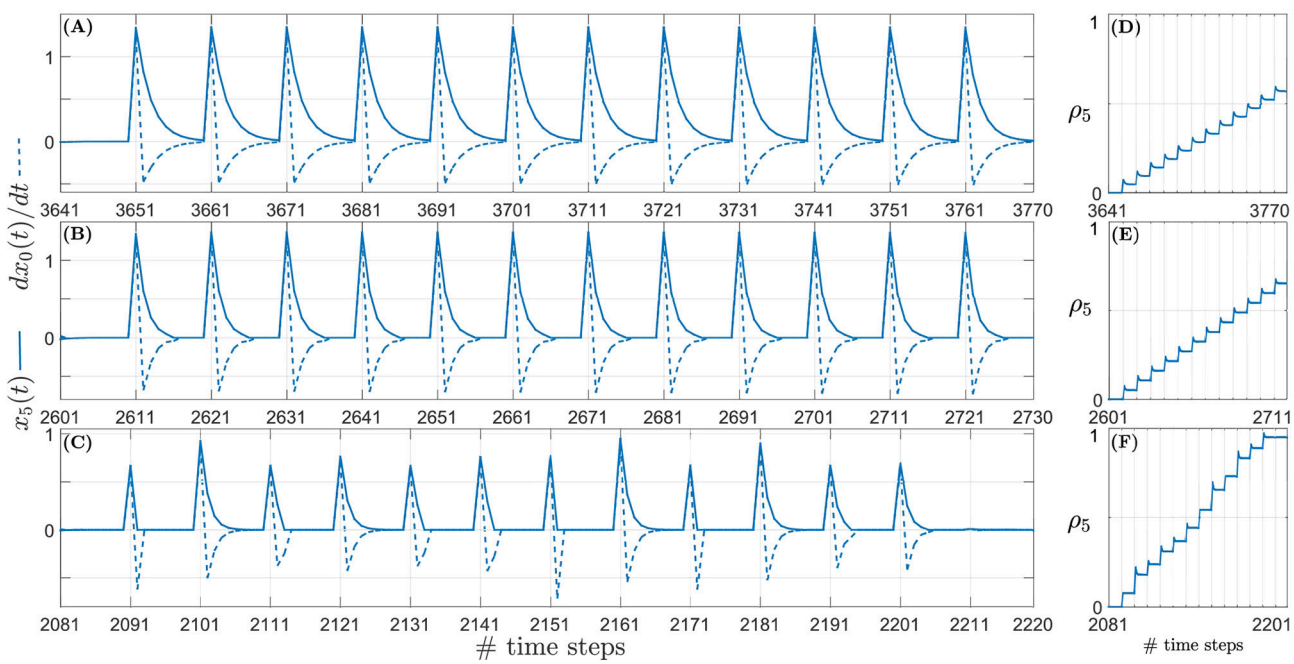


FIGURE 9 | Example snapshots of the synaptic weight updates (right column) corresponding to the correlation between the predictive signal $x_5(t)$ (solid line, left column) and the derivative of the retrospective signal $\frac{dx_0(t)}{dt}$ (dashed line, left column) for a sound signal moving with an angular velocity of 1.5°/time step. (A) Continuous unoccluded sound. **(B)** Periodically occluded sound with 60% duty cycle. **(C)** Randomly occluded sound with random duty cycle. **(D)** Synaptic weights for continuous unoccluded sound. **(E)** Synaptic weights for periodically occluded sound with 60% duty cycle. **(F)** Synaptic weights for randomly occluded sound with random duty cycle.

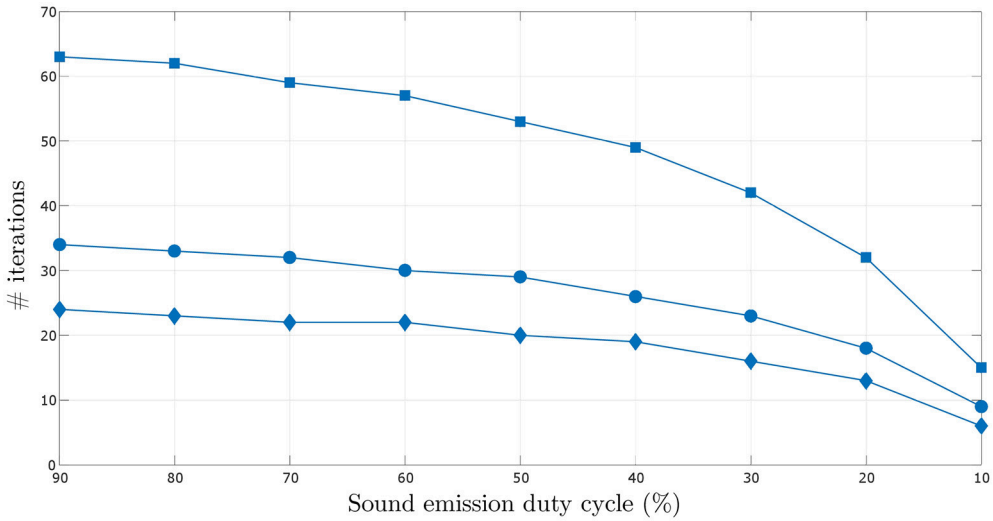


FIGURE 10 | Comparison of the number of iterations required for the synaptic weights to converge over varying sound emission duty cycles in the range 10–90%, for the three target angular velocities–0.5°/time step (square markers), 1.0°/time step (circular markers) and 1.5°/time step (diamond markers).

larger predictive signals $x_k(t)$, and therefore a relatively larger correlation term $x_k(t)\frac{dx_0(t)}{dt}$ per time step in Equation (6). This consequently leads to relatively faster weight updates, reducing the total number of time steps and thus iterations taken to learn the correct angular velocity.

4.2. Real Robot Implementation

Individual robotic trials are conducted for continuous unoccluded as well as occluded sound signals. In all trials, the sound signal is moved virtually in the experimental arena as depicted in **Figure 5** with an angular velocity of 1.5°/0.2 s. We present video footage of the trials in which the robot's tracking behaviour after learning can be seen. Supplementary Videos #1, #2 and #3 (see files titled "video1.mp4", "video2.mp4" and "video3.mp4" respectively in the Supplementary Materials) respectively show the tracking behaviour for the continuous unoccluded sound signal with a duty cycle of 100%, the periodically occluded signal with a duty cycle of 60% and the randomly occluded signal with a random duty cycle between 10 and 90% for each loudspeaker. As evident from the video footage, in all trials the robot is able to successfully perceive the acoustic motion of the sound signal and orient toward the currently playing loudspeaker as indicated by a green LED mounted on the top of the loudspeaker.

Figure 11 depicts the tracking performance during the robotic trials for all three sound signals in terms of the tracking error θ_e. In the robotic trials, the robot's performance is relatively good. Small errors in tracking are observed during the trials as evident from the recorded video footage. Even after undertaking compensatory actions as described in Section 3.3, errors in tracking are unavoidable under real-life conditions due to ambient noise introduced in the sound signals. The robot manages to compensate for any positive or negative tracking errors (that are introduced by respectively turning either too fast or too slow) for a given loudspeaker by respectively making relatively smaller or larger turns for the next loudspeaker. This is because the difference signal $x_(t)$ generated by the peripheral auditory model also provides some information regarding the sound direction (see Section 2.1) that the neural mechanism uses to automatically compensate for tracking errors, even though the synaptic weights are fixed.

The tracking errors are relatively greater for the randomly occluded sound signal as compared to those for the unoccluded and periodically occluded signals. There is a consistent offset from the reference that implies that the robot's turns consistently lag behind the currently playing loudspeaker. This is in agreement with the consistent offset between the alignment of the robotic agent and the angular location of the sound signal (i.e., a consistently non-zero tracking error) observed in simulation as evident in the inset in (**Figure 8C**). This is because the synaptic weights learned for a randomly occluded sound do not correspond to any single sound emission duty cycle. Instead, the algorithm learns the "best possible" values for the synaptic weights that fit all the sound emission duty cycles. This implies that in the robotic trials there will always be an offset between the angular location of the loudspeaker and the orientation of the robot as well.

In our experiments the loudspeaker sequence was never broken during a trial. Breaking the sequence, for example by not playing a given loudspeaker, the sound signal will jump forward along the trajectory by an amount that is twice the nominal displacement. For example, if the angular velocity of the sound signal is 0.5°/time step (or 5°/10 time steps) skipping a single loudspeaker would displace the sound source by 10° in 20 time steps. Assuming that the number of time steps between consecutive loudspeakers is unchanged, the angular speed will

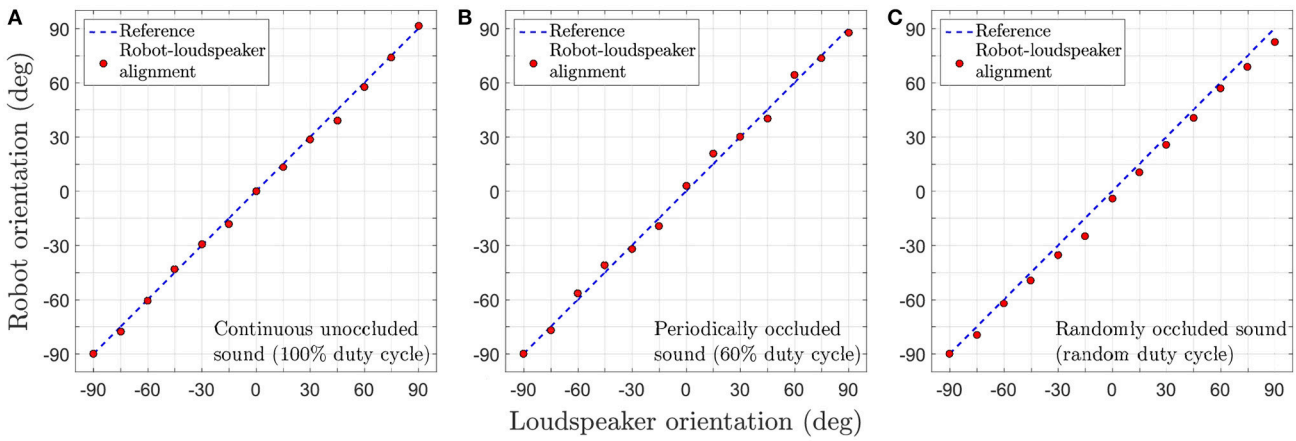


FIGURE 11 | Tracking performance during the robotic trials for a target angular velocity of 1.5°/time step. (A) Continuous unoccluded sound. **(B)** Periodically occluded sound with 60% duty cycle. **(C)** Randomly occluded sound with random duty cycle.

however remain unchanged. This forward jump in the sound signal will cause the synaptic weights to be updated by a relatively larger amount than usual. This would accelerate the learning during any such forward jumps in the sound signal, resulting in relatively faster convergence toward the optimal synaptic weights.

Furthermore, the loudspeaker sequence cannot be random because that would imply a sound signal moving with a randomly varying angular velocity in a randomly varying direction. For example, if the loudspeaker sequence is #1 → #2 → #7 → #4 . . ., then the angular velocity of sound signal will vary as 0.5°/time step → 2.5°/time step → 1.5°/time step → . . ., and the direction of motion will vary as *left*→*left*→*right*→ . . .. This would cause the synaptic weights to increase when the sound signal moves from left to right and decrease when it moves from right to left. The size of weight update, which corresponds to the angular speed, would vary randomly as well. As a consequence of these effects, the synaptic weights will not converge. This implies that the neural mechanism cannot learn a target angular velocity that is not constant.

We have employed a semi-circular trajectory for the sound signal in all experiments to simplify the problem of motion perception such that there is a 1:1 relationship between the agent's learned angular turning velocity and the target's angular velocity. The problem of motion perception is essentially the same in the case of a target moving along linear trajectory with a constant velocity. This is because the temporal relationship between the perceived position of a target sound signal *before* turning and *after* turning depends only on the signal's velocity and not on the shape of the trajectory. Therefore, a robotic agent using the proposed neural mechanism can still learn an angular turning velocity that corresponds to the target linear velocity. In the case of more complicated target trajectories comprising both linear and angular components, the neural mechanism may only learn the average velocity over the entire trajectory.

The neural mechanism is furthermore not limited to a specific sound frequency as its functionality is independent of the sound frequency. For a different sound frequency the peripheral auditory model generates a different difference signal that still encodes sound direction. The neural mechanism essentially uses the direction information in terms of the sign of difference signal to drive the synaptic weight updates in the right direction. However, the size of weight updates is dependent on the absolute magnitude of the difference signal. Therefore, for a different sound frequency but keeping all other neural parameters unchanged, the number of iterations taken for the synaptic weights to converge will be different.

5. CONCLUSIONS AND FUTURE DIRECTIONS

We present an adaptive neural learning mechanism, derived from ICO learning, that employs a synaptic weight update rule adapted from differential Hebbian learning. The neural mechanism was able to successfully learn the constant and unknown angular velocity of a continuous unoccluded pure tone virtual sound signal moving along a semi-circular trajectory in simulation. We also investigated the performance of the neural mechanism in the presence of sparsity in acoustic stimulus. We used three different types of acoustic stimuli each having a sound frequency of 2.2 kHz–continuous unoccluded sound, periodically occluded sound with a 60% sound emission duty cycle and randomly occluded sound with random sound emission duty cycle chosen from a uniform distribution within the range 10–90%.

We first implemented an instance of the neural mechanism in simulation. The neural mechanism was able to learn the angular velocity of the continuous unoccluded sound signal in simulation for three different target angular velocities–1.5°/time step, 1.0°/time step and 0.5°/time step. We validated the acoustic tracking performance of the neural mechanism after learning via robotic trials in tracking a virtually-moving continuous unoccluded sound signal with angular velocity of 1.5°/time step.

We then investigated whether a second instance of the neural mechanism could learn the angular velocity of an

target sound signal that was periodically occluded with a 60% sound emission duty cycle. The sound signal moved with a constant and unknown angular velocity along a semi-circular trajectory as before. The neural mechanism was able to learn the angular velocity of the periodically occluded sound signal in simulation for three different target angular velocities–1.5°/time step, 1.0°/time step and 0.5°/time step. We validated the acoustic tracking performance of the neural mechanism after learning via robotic trials in tracking a virtually-moving but periodically occluded sound signal with angular velocity of 1.5°/time step.

Finally we investigated whether a third instance of the neural mechanism could learn the angular velocity of a target sound signal that was randomly occluded with a randomly varying duty cycle uniformly distributed within the range 10–90%. Once again, the sound signal moved with a constant and unknown angular velocity along a semi-circular trajectory. The neural mechanism was able to learn the angular velocity of the randomly occluded sound signal in simulation for three different target angular velocities–1.5°/time step, 1.0°/time step and 0.5°/time step. We validated the acoustic tracking performance of the neural mechanism after learning via robotic trials in tracking a virtually-moving but randomly occluded sound signal with angular velocity of 1.5°/time step.

In all robotic trials the robot was relatively successful in tracking the sound signal in spite of the absence of compensation for possible detrimental effects such as ambient noise, mismatch between the robot's motor characteristics, ground friction and depletion rate of the battery.

The neural mechanism implements a purely reactive closed-loop system; the robot only turns *after* the target sound signal has moved to a new location along its trajectory and it always follows the sound signal. There is an unavoidable positive and finite delay between the target sound signal moving to its new location and the robot completing its turn. In the simulation this time delay is of one time step and in practice with the real robot it is the sum of the time step and the non-deterministic processing time in the sensorimotor loop. Predatory animals that utilise tracking behaviour, to catch prey for example, tend to be able to predict its future position. Such prediction is clearly more advantageous for the predator to minimise the neural sensorimotor delays (Nijhawan and Wu, 2009; Franklin and Wolpert, 2011) and to maximise its chances of success. Behavioural evidence for predictive tracking mechanisms has been reported in salamanders (Borghuis and Leonardo, 2015) and dragonflies (Dickinson, 2015; Mischiati et al., 2015a,b) that use vision for prey capture. In the auditory domain, the barn owl, *Tyto alba* is well known for auditory prey capture (Konishi, 1973). Both behavioural (Langemann et al., 2016) and neurophysiological (Witten et al., 2006; Weston and Fischer, 2015) evidence has been reported for auditory motion representation in the barn owl. Lizards such as the Mediterranean house geckos, *Hemidactylus tursicus*, are known to prey on crickets and have been observed to orient and navigate toward loudspeakers playing male cricket calls (Sakaluk and Belwood, 1984). However, to the best of our knowledge there is no study or evidence reported in the literature of predictive mechanisms involved in lizard acoustic prey capture. The presented neural mechanism may be used to predict the future position of the sound signal by allowing the learning to continue such that the synaptic weights increase beyond those that correspond to the actual angular velocity of the target sound signal. After successful learning the robot would then turn quickly enough to orient toward a future position of the sound signal. Such a mechanism could be considered as an internal forward model (Wolpert et al., 1995) for acoustic motion perception.

AUTHOR CONTRIBUTIONS

DS: Scientific problem formulation, implementation, experimentation, and manuscript preparation. PM: Support in scientific discussion and manuscript preparation.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00011/full#supplementary-material

REFERENCES

Aertsen, A., Johannesma, P., and Hermes, D. (1980). Spectro-temporal receptive fields of auditory neurons in the grassfrog. *Biol. Cybern.* 38, 235–248. doi: 10.1007/BF00337016

Barnes, G., and Asselman, P. (1992). Pursuit of intermittently illuminated moving targets in the human. *J. Physiol.* 445, 617–637. doi: 10.1113/jphysiol.1992.sp018943

Beitel, R. (1999). Acoustic pursuit of invisible moving targets by cats. *J. Acoust. Soc. Am.* 105, 3449–3453. doi: 10.1121/1.424671

Borghuis, B., and Leonardo, A. (2015). The role of motion extrapolation in amphibian prey capture. *J. Neurosci.* 35, 15430–15441. doi: 10.1523/JNEUROSCI.3189-15.2015

Brandstein, M., and Ward, D. (eds.). (2001). *Microphone Arrays: Signal Processing Techniques and Applications, 1st Edn.* Digital Signal Processing. Berlin; Heidelberg: Springer.

Cai, W., Wang, S., and Wu, Z. (2010). Accelerated steered response power method for sound source localization using orthogonal linear array. *Appl. Acoust.* 71, 134–139. doi: 10.1016/j.apacoust.2009.07.015

Carlile, S., and Leung, J. (2016). The perception of auditory motion. *Trends Hear.* 20:2331216516644254. doi: 10.1177/2331216516644254

Christensen-Dalsgaard, J., and Manley, G. (2005). Directionality of the Lizard Ear. *J. Exp. Biol.* 208, 1209–1217. doi: 10.1242/jeb.01511

Christensen-Dalsgaard, J., Tang, Y., and Carr, C. (2011). Binaural processing by the gecko auditory periphery. *J. Neurophysiol.* 105, 1992–2004. doi: 10.1152/jn.00004.2011

Churchland, M., Chou, I.-H., and Lisberger, S. (2003). Evidence for object permanence in the smooth-pursuit eye movements of monkeys. *J. Neurophysiol.* 90, 2205–2218. doi: 10.1152/jn.01056.2002

DiBiase, J. (2000). *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments using Microphone Arrays*. Ph.D. thesis, Brown University.

DiBiase, J., Silverman, H., and Brandstein, M. (2001). *Robust Localization in Reverberant Rooms*. Berlin; Heidelberg: Springer.

Dickinson, M. (2015). Motor control: how dragonflies catch their prey. *Curr. Biol.* 25, R232–R234. doi: 10.1016/j.cub.2015.01.046

Dmochowski, J., Benesty, J., and Affes, S. (2007). A generalized steered response power method for computationally viable source localization. *IEEE Trans. Audio Speech Lang. Process.* 15, 2510–2526. doi: 10.1109/TASL.2007.906694

Fletcher, N. (1992). *Acoustic Systems in Biology*. New York, NY: Oxford University Press.

Fletcher, N., and Thwaites, S. (1979). Physical models for the analysis of acoustical systems in biology. *Q. Rev. Biophys.* 12, 25–65. doi: 10.1017/S0033583500002596

Franklin, D., and Wolpert, D. (2011). Computational mechanisms of sensorimotor control. *Neuron* 72, 425–442. doi: 10.1016/j.neuron.2011.10.006

Ju, T., Shao, H., and Peng, Q. (2013). "Tracking the moving sound target based on distributed microphone pairs," in *Proceedins of 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (Piscataway, NJ), 330–334.

Ju, T., Shao, H., Peng, Q., and Zhang, M. (2012). "Tracking the moving sound target based on double arrays," in *Computational Problem-Solving (ICCP), 2012 International Conference on* (Piscataway, NJ), 315–319.

Klopf, A. (1988). A neuronal model of classical conditioning. *Psychobiology* 16, 85–125.

Knapp, C., and Carter, G. (1976). The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Process.* 24, 320–327. doi: 10.1109/TASSP.1976.1162830

Konishi, M. (1973). How the owl tracks its prey: experiments with trained barn owls reveal how their acute sense of hearing enables them to catch prey in the dark. *Amer. Sci.* 61, 414–424.

Kosko, B. (1986). Differential Hebbian learning. *AIP Conf. Proc.* 151, 277–282. doi: 10.1063/1.36225

Kwak, K. (2011). "Sound source tracking of moving speaker using multi-channel microphones in robot environments," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on* (Piscataway, NJ), 3017–3020.

Langemann, U., Krumm, B., Liebner, K., Beutelmann, R., and Klump, G. (2016). "Moving objects in the Barn owl's auditory world," in Physiology, *Psychoacoustics and Cognition in Normal and Impaired Hearing,* eds P. van Dijk, D. Başkent, E. Gaudrain, E. de Kleine, A. Wagner, and C. Lanting (Cham: Springer International Publishing), 219–227. doi: 10.1007/978-3-319-25474-6_23

Lehmann, E. (2004). *Particle Filtering Methods for Acoustic Source Localisation and Tracking*. Ph.D. thesis, The Australian National University.

Liang, Z., Ma, X., and Dai, X. (2008a). Robust tracking of moving sound source using multiple model kalman filter. *Appl. Acoust.* 69, 1350–1355. doi: 10.1016/j.apacoust.2007.11.010

Liang, Z., Ma, X., and Dai, X. (2008b). Robust tracking of moving sound source using scaled unscented particle filter. *Appl. Acoust.* 69, 673–680. doi: 10.1016/j.apacoust.2007.04.001

Lima, M., Martins, W., Nunes, L., Biscainho, L., Ferreira, T., Costa, M., et al. (2015). A volumetric SRP with refinement step for sound source localization. *IEEE Signal Process. Lett.* 22, 1098–1102. doi: 10.1109/LSP.2014.2385864

Manoonpong, P., Geng, T., Kulvicius, T., Porr, B., and Wörgötter, F. (2007). Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput. Biol.* 3:e134. doi: 10.1371/journal.pcbi.0030134

Manoonpong, P., Kolodziejski, C., Wörgötter, F., and Morimoto, J. (2013). Combining correlation-based and reward-based learning in neural control for policy improvement. *Adv. Comp. Syst.* 16:1350015. doi: 10.1142/s021952591350015x

Manoonpong, P., and Wörgötter, F. (2009). "Adaptive sensor-driven neural control for learning in walking machines," in *Neural Information Processing: 16th International Conference, ICONIP 2009, Bangkok, December 1-5, 2009, Proceedings, Part II* (Berlin; Heidelberg: Springer), 47–55.

Marti, A., Cobos, M., Lopez, J., and Escolano, J. (2013). A steered response power iterative method for high-accuracy acoustic source localization. *J. Acoust. Soc. Am.* 134, 2627–2630. doi: 10.1121/1.4820885

Mischiati, M., Lin, H.-T., Herold, P., Imler, E., Olberg, R., and Leonardo, A. (2015a). Internal models direct dragonfly interception steering. *Nature* 517, 333–338. doi: 10.1038/nature14045

Mischiati, M., Lin, H.-T., Herold, P., Imler, E., Olberg, R., and Leonardo, A. (2015b). Neuroscience: dragonflies predict and plan their hunts. *Nature* 517, 279–280. doi: 10.1038/nature14078

Morgan, M., and Turnbull, D. (1978). Smooth eye tracking and the perception of motion in the absence of real movement. *Vis. Res.* 18, 1053–1059. doi: 10.1016/0042-6989(78)90034-2

Nakadai, K., Lourens, T., Okuno, H., and Kitano, H. (2000). "Active audition for humanoid," in *In Proceedings of 17th National Conference on Artificial Intelligence (AAAI-2000) (2000), AAAI* (Menlo Park, CA: AAAI), 832–839.

Nijhawan, R., and Wu, S. (2009). Compensating time delays with neural predictions: are predictions sensory or motor? *Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci.* 367, 1063–1078. doi: 10.1098/rsta.2008.0270

Ning, F., Gao, D., Niu, J., and Wei, J. (2015). "Combining compressive sensing with particle filter for tracking moving wideband sound sources," in *Signal Processing, Communications and Computing (ICSPCC), 2015 IEEE International Conference on* (Piscataway, NJ), 1–6.

Nishie, S., and Akagi, M. (2013). "Acoustic sound source tracking for a moving object using precise doppler-shift measurement," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European* (Piscataway, NJ), 1–5.

Oh, Y., Yoon, J., Park, J., Kim, M., and Kim, H. (2008). A name recognition based call-and-come service for home robots. *IEEE Trans. Consum. Electr.* 54, 247–253. doi: 10.1109/TCE.2008.4560082

Okuno, H., Nakadai, K., Hidai, K.-I., Mizoguchi, H., and Kitano, H. (2003). Human-robot non-verbal interaction empowered by real-time auditory and visual multiple-talker tracking. *Adv. Robot.* 17, 115–130. doi: 10.1163/156855303321165088

Open Source Physics (2016). *Tracker: Video Analysis and Modeling Tool*. Available online at: http://physlets.org/tracker/

Orban de Xivry, J.-J., Missal, M., and Lefèvre, P. (2008). A dynamic representation of target motion drives predictive smooth pursuit during target blanking. *J. Vis.* 8:6. doi: 10.1167/8.15.6

Oudejans, R., Michaels, C., Bakker, F., and Dolné (1996). The relevance of action in perceiving affordances: perception of catchableness of fly balls. *J. Exp. Psychol. Hum. Percept. Perform.* 22, 879–891. doi: 10.1037/0096-1523.22.4.879

Porr, B., and Wörgötter, F. (2007). Fast heterosynaptic learning in a robot food retrieval task inspired by the limbic system. *Biosystems* 89, 294–299. doi: 10.1016/j.biosystems.2006.04.026

Porr, B., and Wörgötter, F. (2006). Strongly improved stability and faster convergence of temporal sequence learning by utilising input correlations only. *Neural Comput.* 18, 1380–1412. doi: 10.1162/neco.2006.18.6.1380

Reeve, R., and Webb, B. (2003). New neural circuits for robot phonotaxis. *Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci.* 361, 2245–2266. doi: 10.1098/rsta.2003.1274

Reichardt, W. (eds.). (1969). "Movement perception in insects," in *Processing of Optical Data by Organisms and by Machines* (New York, NY: Academic Press), 465–493.

Sakaluk, S., and Belwood, J. (1984). Gecko phonotaxis to cricket calling song: a case of satellite predation. *Anim. Behav.* 32, 659–662. doi: 10.1016/S0003-3472(84)80141-4

Shaikh, D., Hallam, J., and Christensen-Dalsgaard, J. (2016). From "ear" to there: a review of biorobotic models of auditory processing in lizards. *Biol. Cybern.* 110, 303–317. doi: 10.1007/s00422-016-0701-y

Shaikh, D., and Manoonpong, P. (2016). "An adaptive neural mechanism with a lizard ear model for binaural acoustic tracking," in *From Animals to Animats 14: 14th International Conference on Simulation of Adaptive Behavior, SAB 2016,* eds E. Tuci, A. Giagkos, M. Wilson, and J. Hallam (Cham: Springer International Publishing), 79–90.

Tsuji, D., and Suyama, K. (2009). "A moving sound source tracking based on two successive algorithms," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on* (Piscataway, NJ), 2577–2580. doi: 10.1109/iscas.2009.5118328

Valin, J.-M., Michaud, F., and Rouat, J. (2007). Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robot. Auton. Syst.* 55, 216–228. doi: 10.1016/j.robot.2006.08.004

Wan, X., and Wu, Z. (2010). Improved steered response power method for sound source localization based on principal eigenvector. *Appl. Acoust.* 71, 1126–1131. doi: 10.1016/j.apacoust.2010.06.011

Wang, H., and Chu, P. (1997). "Voice source localization for automatic camera pointing system in videoconferencing," in *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Vol.1* (Washington, DC: IEEE Computer Society), 187–190.

Ward, D. B., Lehmann, E. A., and Williamson, R. C. (2003). Particle filtering algorithms for tracking an acoustic source in a reverberant environment. *IEEE Trans. Speech Audio Process.* 11, 826–836. doi: 10.1109/TSA.2003.818112

Weston, C., and Fischer, B. (2015). Optimal prediction of moving sound source direction in the owl. *PLoS Comput. Biol.* 11:e1004360. doi: 10.1371/journal.pcbi.1004360

Wever, E. (1978). *The Reptile Ear: Its Structure and Function*. Princeton, NJ: Princeton University Press.

Witten, I. B., Bergan, J. F., and Knudsen, E. I. (2006). Dynamic shifts in the owl's auditory space map predict moving sound location. *Nat. Neurosci.* 9, 1439–1445. doi: 10.1038/nn1781

Wolpert, D., Ghahramani, Z., and Jordan, M. (1995). An internal model for sensorimotor integration. *Science* 269, 1880–1882. doi: 10.1126/science.7569931

Zhang, L. (2009). *Modelling Directional Hearing in Lizards*. Ph.D. thesis, Maersk Mc-Kinney Moller Institute, Faculty of Engineering, University of Southern Denmark.

Zhao, X., Tang, J., Zhou, L., and Wu, Z. (2013). Accelerated steered response power method for sound source localization via clustering search. *Sci. China Phys. Mech. Astron.* 56, 1329–1338. doi: 10.1007/s11433-013-5108-3

Zotkin, D., and Duraiswami, R. (2004). Accelerated speech source localization via a hierarchical search of steered response power. *IEEE Trans. Speech Audio Proc.* 12, 499–508. doi: 10.1109/TSA.2004.832990

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Shaikh and Manoonpong. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 22 March 2017
doi: 10.3389/fnbot.2017.00013

Real-Time Biologically Inspired Action Recognition from Key Poses Using a Neuromorphic Architecture

Georg Layher, Tobias Brosch and Heiko Neumann*

Institute of Neural Information Processing, Ulm University, Ulm, Germany

Intelligent agents, such as robots, have to serve a multitude of autonomous functions. Examples are, e.g., collision avoidance, navigation and route planning, active sensing of its environment, or the interaction and non-verbal communication with people in the extended reach space. Here, we focus on the recognition of the action of a human agent based on a biologically inspired visual architecture of analyzing articulated movements. The proposed processing architecture builds upon coarsely segregated streams of sensory processing along different pathways which separately process form and motion information (Layher et al., 2014). Action recognition is performed in an event-based scheme by identifying representations of characteristic pose configurations (key poses) in an image sequence. In line with perceptual studies, key poses are selected unsupervised utilizing a feature-driven criterion which combines extrema in the motion energy with the horizontal and the vertical extendedness of a body shape. Per class representations of key pose frames are learned using a deep convolutional neural network consisting of 15 convolutional layers. The network is trained using the *energy-efficient deep neuromorphic networks* (*Eedn*) framework (Esser et al., 2016), which realizes the mapping of the trained synaptic weights onto the *IBM Neurosynaptic System* platform (Merolla et al., 2014). After the mapping, the trained network achieves real-time capabilities for processing input streams and classify input images at about 1,000 frames per second while the computational stages only consume about 70 mW of energy (without spike transduction). Particularly regarding mobile robotic systems, a low energy profile might be crucial in a variety of application scenarios. Cross-validation results are reported for two different datasets and compared to state-of-the-art action recognition approaches. The results demonstrate, that (I) the presented approach is on par with other key pose based methods described in the literature, which select key pose frames by optimizing classification accuracy, (II) compared to the training on the full set of frames, representations trained on key pose frames result in a higher confidence in class assignments, and (III) key pose representations show promising generalization capabilities in a cross-dataset evaluation.

Keywords: action recognition, key pose selection, deep learning, neuromorphic architecture, IBM neurosynaptic system

OPEN ACCESS

Edited by:
Poramate Manoonpong,
University of Southern Denmark
Odense, Denmark

Reviewed by:
Johannes Bill,
Heidelberg University, Germany
Quansheng Ren,
Peking University, China

****Correspondence:***
Georg Layher
georg.layher@uni-ulm.de

Received: *25 November 2016*
Accepted: *21 February 2017*
Published: *22 March 2017*

Citation:
Layher G, Brosch T and Neumann H (2017) Real-Time Biologically Inspired Action Recognition from Key Poses Using a Neuromorphic Architecture. Front. Neurorobot. 11:13. doi: 10.3389/fnbot.2017.00013

1. INTRODUCTION

Analyzing and understanding the actions of humans is one of the major challenges for future technical systems aiming at visual sensory behavior analysis. Acquiring knowledge about what a person is doing is of importance and sometimes even crucial in a variety of scenarios. In the context of automated surveillance systems, action analysis is an essential ability, allowing to identify potential threads emanating from an individual or a group of persons. In Human-Computer-Interaction (HCI), action analysis helps in understanding the objectives and intentions of a user and increases the potential of a system to adapt to the specific context of an interaction and appropriately support, guide or protect the user. Moreover, recognizing actions in the surrounding area is an integral part of interpreting the own situative context and environment, and thus is in particular crucial for mobile robotic systems which may find themselves embedded in a variety of different situations.

In the presented work, as the first main contribution, a *feature-driven* key pose selection method is proposed, which is driven by combining two features in the biological motion input, namely extrema in the temporal motion energy signal and the relative extent of a subject's pose. Such temporally defined features (from the motion stream) help to automatically select key pose representations. The use of these dynamic features has been motivated by psychophysical investigations (Thirkettle et al., 2009) which demonstrate that humans select specific poses in a continuous sequence of video input based on such criteria. We first show how such key poses define events within articulated motion sequences and how these can be reliably and automatically detected. The proposed processing architecture builds upon coarsely segregated streams of sensory processing along different pathways which separately process form and motion information (Giese and Poggio, 2003). An interaction between the two processing streams enables an automatic selection of characteristic poses during learning (Layher et al., 2014). To use such recognition functionality in an autonomous neurobiologically inspired recognition system various constraints need to be satisfied. Such neurobiological systems need to implement the underlying processes along the processing and recognition cascade which defines the parts of their cognitive functionality.

As the second key contribution, we employ here an *energy efficient deep convolutional neural network* (Eedn; Esser et al., 2016) to realize the key pose learning and classification, which achieves a computationally efficient solution using a sparse and energy efficient implementation based on neuromorphic hardware. This allows us to establish a cascaded hierarchy of representations with an increasing complexity for key pose form and motion patterns. After their establishment, key pose representations allow an assignment of a given input image to a specific action category. We use an offline training scheme that utilizes a deep convolutional neural network with 15 convolutional layers. The trained network runs on IBM's TrueNorth chip (Merolla et al., 2014; Akopyan et al., 2015). This solution renders it possible to approach faster than real-time capabilities for processing input streams and classify articulated still images at about 1,000 frames per second while the computational stages consume only about 70 mW of energy. We present cross-validation results on an action recognition dataset consisting of 14 actions and 22 subjects and about 29,000 key pose frames, which show a recall rate for the presented approach of about 88%, as well as a comparison to state-of-the-art action recognition approaches on a second dataset. To show the generalization capabilities of the proposed key pose based approach, we additionally present the results of a cross-dataset evaluation, where the training and the testing of the network was performed on two completely separate datasets with overlapping classes.

2. RELATED WORK

The proposed key pose based action recognition approach is motivated and inspired by recent evidences about the learning mechanisms and representations involved in the processing of articulated motion sequences, as well as hardware and software developments from various fields of visual sciences. For instance, empirical studies indicate, that special kinds of events within a motion sequence facilitate the recognition of an action. Additional evidences from psychophysics, as well as neurophysiology suggest that both, form and motion information contribute to the representation of an action. Modeling efforts propose functional mechanisms for the processing of biological motion and show how such processing principles can be transfered to technical domains. Deep convolutional networks make it possible to learn hierarchical object representations, which show an impressive recognition performance and enable the implementation of fast and energy efficient classification architectures, particularly in combination with neuromorphic hardware platforms. In the following sections, we will briefly introduce related work and results from different scientific fields, all contributing to a better understanding of action representation and the development of efficient action recognition approaches.

2.1. Articulated and Biological Motion

Starting with the pioneering work of Johansson (1973), perceptual sciences gained more and more insights about how biological motion might be represented in the human brain and what the characteristic properties of an articulated motion sequence are. In psychophysical experiments, humans show a remarkable performance in recognizing biological motions, even when the presented motion is reduced to a set of points moving coherently with body joints (*point light stimuli*; PLS). In a detection task, subjects were capable of recognizing a walking motion within about 200 ms (Johansson, 1976). These stimuli, however, are not free of – at least configurational – form information and the discussion about the contributions of form and motion in biological motion representation is still ongoing (Garcia and Grossman, 2008). Some studies indicate a stronger importance of motion cues (Mather and Murdoch, 1994), others emphasize the role of configurational

form information (Lange and Lappe, 2006). Even less is known about the specific nature and characteristic of the visual cues which facilitate the recognition of a biological motion sequence. In Casile and Giese (2005), a statistical analysis as well as the results of psychophysical experiments indicate that local opponent motion in horizontal direction is one of the critical features for the recognition of PLS. Thurman and Grossman (2008) conclude, that there are specific moments in an action performance which are "more perceptually salient" compared to others. Their results emphasize the importance of dynamic cues in moments when the distance between opposing limbs is the lowest (corresponding to local opponent motion; maxima in the motion energy). On the contrary, more recent findings by Thirkettle et al. (2009) indicate, that moments of a large horizontal body extension (co-occurring with minima in the motion energy) facilitate the recognition of a biological motion in a PLS.

In neurophysiology, functional imaging studies (Grossman et al., 2000), as well as single-cell recordings (Oram and Perrett, 1994) indicate the existence of specialized mechanisms for the processing of biological motion in the *superior temporal sulcus* (STS). STS has been suggested to be a point of convergence of the separate dorsal "where" and the ventral "what" pathways (Boussaoud et al., 1990; Felleman and Van Essen, 1991), containing cells which integrate form and motion information of biological objects (Oram and Perrett, 1996) and selectively respond to, e.g., object manipulation, face, limb and whole body motion (Puce and Perrett, 2003). Besides the evidence that both form and motion information contribute to the registration of biological motion, action specific cells in STS are reported to respond to static images of articulated bodies which in parallel evoke activities in the *medio temporal* (MT) and *medial superior temporal* (MST) areas of the dorsal stream (*implied motion*), although there is no motion present in the input signal (Kourtzi and Kanwisher, 2000; Jellema and Perrett, 2003). In line with the psychophysical studies, these results indicate that poses with a specific feature characteristic (here, articulation) facilitate the recognition of a human motion sequence.

Complementary modeling efforts in the field of computational neuroscience suggest potential mechanisms which might explain the underlying neural processing and learning principles. In Giese and Poggio (2003) a model for the recognition of biological movements is proposed, which processes visual input along two separate form and motion pathways and temporally integrates the responses of prototypical motion and form patterns (snapshots) cells via asymmetric connections in both pathways. Layher et al. (2014) extended this model by incorporating an interaction between the two pathways, realizing the automatic and unsupervised learning of key poses by modulating the learning of the form prototypes using a motion energy based signal derived in the motion pathway. In addition, a feedback mechanism is proposed in this extended model architecture which (I) realizes sequence selectivity by temporal association learning and (II) gives a potential explanation for the activities in MT/MST observed for static images of articulated poses in neurophysiological studies.

2.2. Action Recognition in Image Sequences

In computer vision, the term vision-based action recognition summarizes approaches to assign an action label to each frame or a collection of frames of an image sequence. Over the last decades, numerous vision-based action recognition approaches have been developed and different taxonomies have been proposed to classify them by different aspects of their processing principles. In Poppe (2010), action recognition methods are separated by the nature of the image representation they rely on, as well as the kind of the employed classification scheme. Image representations are divided into *global representations*, which use a holistic representation of the body in the *region of interest* (ROI; most often the bounding box around a body silhouette in the image space), and *local representations*, which describe image and motion characteristics in a spatial or spatio-temporal local neighborhood. Prominent examples for the use of whole body representations are *motion history images* (MHI) (Bobick and Davis, 2001), or the application of *histograms of oriented gradients* (HOG) (Dalal and Triggs, 2005; Thurau and Hlavác, 2008). Local representations are, e.g., employed in Dollar et al. (2005), where motion and form based descriptors are derived in the local neighborhood (cuboids) of spatio-temporal interest points. Classification approaches are separated into *direct classification*, which disregard temporal relationships (e.g., using histograms of prototype descriptors, Dollar et al., 2005) and *temporal state-space models*, which explicitly model temporal transitions between observations (e.g., by employing *Hidden Markov models* (HMMs) Yamato et al., 1992, or *dynamic time warping* (DTW) Chaaraoui et al., 2013). For further taxonomies and an exhaustive overview of computer vision action recognition approaches we refer to the excellent reviews in Gavrila (1999); Aggarwal and Ryoo (2011); Weinland et al. (2011).

The proposed approach uses motion and form based feature properties to extract key pose frames. The identified key pose frames are used to learn class specific key pose representations using a deep convolutional neural network (DCNN). Classification is either performed framewise or by temporal integration through majority voting. Thus, following the taxonomy of Poppe (2010), the approach can be classified as using *global representations* together with a *direct classification* scheme. Key pose frames are considered as temporal events within an action sequence. This kind of action representation and classification is inherently invariant against variations in (recording and execution) speed. We do not argue that modeling temporal relationships between such events is not necessary in general. The very simple temporal integration scheme was chosen to focus on an analysis of the importance of key poses in the context of action representation and recognition. Because of the relevance to the presented approach, we will briefly compare specifically key pose base action recognition approaches in the following.

2.3. Key Pose Based Action Recognition

Key pose based action recognition approaches differ in their understanding of the concept of key poses. Some take a phenomenological perspective and define key poses as events which possess a specific feature characteristic giving rise to their peculiarity. There is no a priori knowledge available about whether, when and how often such *feature-driven* events occur within an observed action sequence, neither during the establishment of the key pose representations during training, nor while trying to recognize an action sequence. Others regard key pose selection as the result of a statistical analysis, favoring poses which are easy to separate among different classes or maximally capture the characteristics of an action sequence. The majority of approaches rely on such statistical properties and either consider the *intra-* or the *inter-class* distribution of image-based pose descriptors to identify key poses in action sequences.

Intra-Class Based Approaches

Approaches which evaluate *intra-class* properties of the feature distributions regard key poses as the most representative poses of an action and measures of centrality are exploited on agglomerations in pose feature spaces to identify the poses which are most common to an action sequence. In Chaaraoui et al. (2013), a contour based descriptor following (Dedeoğlu et al., 2006) is used. Key poses are selected by repetitive *k*-means clustering of the pose descriptors and evaluating the resulting clusters using a compactness metric. A sequence of nearest neighbor key poses is derived for each test sequence and dynamic time warping (DTW) is applied to account for different temporal scales. The class of the closest matching temporal sequence of key poses from the training set is used as the final recognition result. Based on histograms of oriented gradients (HOG) and histograms of weighted optical flow (HOWOF) descriptors, Cao et al. (2012) adapt a local linear embedding (LLE) strategy to establish a manifold model which reduces descriptor dimensionality, while preserving the local relationship between the descriptors. Key poses are identified by interpreting the data points (i.e., descriptors/poses) on the manifold as an adjacent graph and applying a *PageRank* (Brin and Page, 1998) based procedure to determine the vertices of the graph with the highest centrality, or relevance.

In all, key pose selection based on an *intra-class* analysis of the feature distribution has the advantage of capturing the characteristics of one action in isolation, independent of other classes in a dataset. Thus, key poses are not dataset specific and – in principle – can also be shared among different actions. However, most *intra-class* distribution based approaches build upon measures of centrality (i.e., as a part of cluster algorithms) and thus key poses are dominated by frequent poses of an action. Because they are part of transitions between others, frequent poses tend to occur in different classes and thus do not help in separating them. Infrequent poses, on the other hand, are not captured very well, but are intuitively more likely to be discriminative. The authors' are not aware of an *intra-class* distribution based method which tries to identify key poses based on their infrequency or abnormality (e.g., by evaluating cluster sizes and distances).

Inter-Class Based Approaches

Approaches based on *inter-class* distribution, on the other hand, consider highly discriminative poses as key poses to separate different action appearances. Discriminability is here defined as resulting in either the best classification performance or in maximum dissimilarities between the extracted pose descriptors of different classes. To maximize the classification performance, Weinland and Boyer (2008) propose a method of identifying a vocabulary of highly discriminative pose exemplars. In each iteration of the forward selection of key poses, one exemplar at a time is added to the set of key poses by independently evaluating the classification performance of the currently selected set of poses in union with one of the remaining exemplars in the training set. The pose exemplar, which increases classification performance the most is then added to the final key pose set. The procedure is repeated until a predefined number of key poses is reached. Classification is performed based on a distance metric obtained by either silhouette-to-silhouette or silhouette-to-edge matching. Liu et al. (2013) combine the output of the early stages of an HMAX inspired processing architecture (Riesenhuber and Poggio, 1999) with a center-surround feature map obtained by subtracting several layers of a Gaussian pyramid and a wavelet laplacian pyramid feature map into framewise pose descriptors. The linearized feature descriptors are projected into a low-dimensional subspace derived by principal component analysis (PCA). Key poses are selected by employing an adaptive boosting technique (AdaBoost; Freund and Schapire, 1995) to select the most discriminative feature descriptors (i.e., poses). A test action sequence is matched to the thus reduced number of exemplars per action by applying an adapted local naive Bayes nearest neighbor classification scheme (LNBNN; McCann and Lowe, 2012). Each descriptor of a test sequence is assigned to its *k* nearest neighbors and a classwise voting is updated by the distance of a descriptor to the respective neighbor weighted by the relative number of classes per descriptor. In Baysal et al. (2010), noise reduced edges of an image are chained into a contour segmented network (CSN) by using orientation and closeness properties and transformed into a 2-adjacent segment descriptor (k-AS; Ferrari et al., 2008). The most characteristic descriptors are determined by identifying *k* candidate key poses per class using the *k*-medoids clustering algorithm and selecting the most distinctive ones among the set of all classes using a similarity measure on the 2-AS descriptors. Classification is performed by assigning each frame to the class of the key pose with the highest similarity and sequence-wide majority voting. Cheema et al. (2011) follow the same key pose extraction scheme, but instead of selecting only the most distinctive ones, key pose candidates are weighted by the number of false and correct assignments to an action class. A weighted voting scheme is then used to classify a given test sequence. Thus, although key poses with large weights have an increased influence on the final class assignment, all key poses take part in the classification process. Zhao and Elgammal (2008) use an information theoretic approach to select key frames within action

sequences. They propose to describe the local neighborhood of spatiotemporal interest points using an intensity gradient based descriptor (Dollar et al., 2005). The extracted descriptors are then clustered, resulting in a codebook of prototypical descriptors (visual words). The pose prototypes are used to estimate the discriminatory power of a frame by calculating a measure based on the conditional entropy given the visual words detected in a frame. The frames with the highest discriminatory power are marked as key frames. Chi-square distances of histogram based spatiotemporal representations are used to compare key frames from the test and training datasets and majority voting is used to assign an action class to a test sequence.

For a given pose descriptor and/or classification architecture, *inter-class* based key pose selection methods in principle minimize the recognition error, either for the recognition of the key poses (e.g., Baysal et al., 2010; Liu et al., 2013) or for the action classification (e.g., Weinland and Boyer, 2008). But, on the other hand, key poses obtained by *inter-class* analysis inherently do not cover the most characteristic poses of an action, but the ones which are the most distinctive within a specific set of actions. Applying this class of algorithms on two different sets of actions sharing one common action might result in a different selection of key poses for the same action. Thus, once extracted, key pose representations do not necessarily generalize over different datasets/domains and, in addition, sharing of key poses between different classes is not intended.

Feature-Driven Approaches

Feature-driven key pose selection methods do not rely on the distribution of features or descriptors at all and define a key pose as a pose which co-occurs with a specific characteristic of an image or feature. Commonly employed features, such as extrema in a motion energy based signal, are often correlated with pose properties such as the degree of articulation or the extendedness. Compared to statistical methods, this is a more pose centered perspective, since parameters of the pose itself are used to select a key pose instead of parameters describing the relationship or differences between poses.

Lv and Nevatia (2007) select key poses in sequences of 3D-joint positions by automatically locating extrema of the motion energy within temporal windows. Motion energy in their approach is determined by calculating the sum over the L^2 norm of the motion vectors of the joints between two temporally adjacent timesteps. 3D motion capturing data is used to render 2D projections of the key poses from different view angles. Single frames of an action sequence are matched to the silhouettes of the resulting 2D key pose representations using an extension of the Pyramid Match Kernel algorithm (PMK; Grauman and Darrell, 2005). Transitions between key poses are modeled using action graph models. Given an action sequence, the most likely action model is determined using the Viterbi Algorithm. In Gong et al. (2010), a key pose selection mechanism for 3D human action representations is proposed. Per action sequence, feature vectors (three angles for twelve joints) are projected onto the subspace spanned by the first three eigenvectors obtained by PCA. Several instances of an action are synchronized to derive the mean performance (in terms of execution) of an action. Motion energy is then defined by calculating the Euclidean distance between two adjacent poses in the mean performance. The local extrema of the motion energy are used to select the key poses, which after their reconstruction in the original space are used as the vocabulary in a bag of words approach. During recognition, each pose within a sequence is assigned to the key pose with the minimum Euclidean distance resulting in a histogram of key pose occurrences per sequence. These histograms serve as input to a support vector machine (SVM) classifier. In Ogale et al. (2007), candidate key poses are extracted by localizing the extrema of the mean motion magnitude in the estimated optical flow. Redundant poses are sorted out pairwise by considering the ratio between the intersection and the union of two registered silhouettes. The final set of unique key poses is used to construct a probabilistic context-free grammar (PCFG). This method uses an *inter-class* metric to reject preselected key pose candidates and thus is not purely feature-driven.

Feature-driven key pose selection methods are independent of the number of different actions within a dataset. Thus, retraining is not necessary if, e.g., a new action is added to a dataset and the sharing of key poses among different actions is in principle possible. Naturally, there is no guarantee, that the selected poses maximize the separability of pose or action classes.

3. MODEL/METHODS

To realize an energy efficient implementation for key pose based action recognition, the proposed model uses a neuromorphic deep convolutional neural network (DCNN) to selectively learn representations of key poses which are assigned to different action classes. In the preprocessing phase, optical flow is calculated on the input sequences and key pose frames are selected in an unsupervised manner. Form and motion information is calculated for each key pose frame. The concatenated form and motion information is then used as the input to the DCNN. In the following, detailed information about the image preprocessing, the key pose selection automatism and the structure and functionality of the DCNN are presented. All simulations were carried out using a neuromorphic computing paradigm and mapped to the IBM TrueNorth hardware platform (Merolla et al., 2014).

3.1. Key Pose Selection and Image Preprocessing

During preprocessing, two elementary processing steps are performed. First, the key pose selection is performed by automatically analyzing simple motion and form parameters. Second, the final input to the network is calculated by combining the form and motion representations I^{form} and I^{motion} obtained by simple image-based operations.

Key Pose Selection

The key pose selection process operates upon two parameters, namely (I) local temporal extrema in the motion energy and (II) the extendedness of a subject at a given timestep. Optical flow is calculated using a differential method, as suggested in the *Lucas-Kanade* optical flow estimation algorithm (Lucas and

Kanade, 1981). Given an image sequence $I(\mathbf{x}, t)$, the optical flow $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ at timestep t and position $\mathbf{x} = (x, y)$ is estimated in a local neighborhood $N(\mathbf{x})$ by minimizing

$$\sum_{\mathbf{y} \in N(\mathbf{x})} W(\mathbf{x} - \mathbf{y})^2 [I_x(\mathbf{y}, t) u(\mathbf{x}, t) + I_y(\mathbf{x}, t) v(\mathbf{x}, t) + I_t(\mathbf{y}, t)]^2, \quad (1)$$

where $W(\mathbf{x} - \mathbf{y})$ increases the influence of the optical flow constraints within the center of the local neighborhood (for details see Barron et al., 1994). The spatiotemporal derivatives I_x, I_y and I_t are estimated by convolution of the image sequences with the forth-order central difference $[-1, 8, 0, -8, 1]/12$ and it's transpose in the spatial and the first-order backward difference $[-1, 1]$ in the temporal domain. A separable 2D kernel with 1D coefficients of $[1, 4, 6, 4, 1]/16$ is used to realize the weighted integration of the derivatives within a 5×5 spatial neighborhood ($N(\mathbf{x})$)[1]. The use of the *Lucas-Kanade* algorithm is not a hard prerequisite for the proposed approach. Other types of optical flow estimators might be applied as well (e.g., (Brosch and Neumann, 2016), which is capable to be executed on neuromorphic hardware). The overall motion energy E^{flo} is then calculated by integrating the speed of all estimated flow vectors within the vector field.

$$E^{\text{flo}}(t) = \sum_{\mathbf{x} \in I(\mathbf{x},t)} \|\mathbf{u}(\mathbf{x}, t)\|_2 = \sum_{\mathbf{x} \in I(\mathbf{x},t)} \sqrt{u(\mathbf{x}, t)^2 + v(\mathbf{x}, t)^2}, \quad (2)$$

Motion energy is smoothed by convolving the estimated motion energy with a Gaussian kernel, $\tilde{E}^{\text{flo}}(t) = (E^{\text{flo}} * G_\sigma)(t)$. In the performed simulations, $\sigma = 2$ and $\sigma = 4$ were used dependent on the dataset[2]. Potential key pose frames are then marked by identifying the local extrema of the motion energy signal.

$$\mathcal{K}^{\text{flo}} = \{I(t), t \in [1, ..., T] | \text{t is a local extremum of } \tilde{E}^{\text{flo}}(t)\}, \quad (3)$$

The relative horizontal and vertical extent of a given pose at time t is then used to reject local extrema with an extent smaller than a predefined percentual threshold λ, as defined by:

$$\mathcal{K} = \mathcal{K}^{\text{flo}} \cap \mathcal{K}^{\text{ext}}. \quad (4)$$

with

$$\begin{aligned} \mathcal{K}^{\text{ext}} = \{ & I(t), t \in [1, ..., T] \mid (\text{Ext}^{\text{ver}}(t) > (1 + \lambda)\overline{\text{Ext}}^{\text{ver}}) \\ & \vee (\text{Ext}^{\text{ver}}(t) < (1 - \lambda)\overline{\text{Ext}}^{\text{ver}}) \\ & \vee (\text{Ext}^{\text{hor}}(t) > (1 + \lambda)\overline{\text{Ext}}^{\text{hor}}) \\ & \vee (\text{Ext}^{\text{hor}}(t) < (1 - \lambda)\overline{\text{Ext}}^{\text{hor}})\} \end{aligned} \quad (5)$$

In the performed simulations, values of $\lambda = 0.1$ and $\lambda = 0.05$ were used for the two different datasets. The percentual thresholds were determined manually with the aim to compensate for differences in the temporal resolution of the datasets. The horizontal and vertical extent Ext^{hor} and Ext^{ver} are derived framewise by estimating the width and the height of the bounding box enclosing the body shape. The extent of a neutral pose is used as the reference extent $\overline{\text{Ext}}^{\text{hor}}$ and $\overline{\text{Ext}}^{\text{ver}}$, which are derived from the width and height of the bounding box in the first frame of a sequence. Silhouette representations, and thus the bounding boxes of the bodies, are available for both datasets used in the simulations. In constrained recording scenarios, silhouettes can be extracted by background subtraction or using the optical flow fields calculated for the selection of the key pose frames. **Figure 1A** shows the motion energy signal $\tilde{E}^{\text{flo}}$ together with the extent Ext^{hor} and Ext^{ver} and their reference values. A strong correlation between the motion energy and the extent of the pose can be seen. In **Figure 1B**, examples for the horizontal and the vertical extent are displayed for a neutral and a extended posture. While the motion energy allows an identification of temporal anchor points in a motion sequence, the extent helps in selecting the most characteristic ones.

Form and Motion Representations

For each selected key pose frame $I^{\text{key}} \in \mathcal{K}$, a form representation is derived by estimating the spatial derivatives I_x^{key} and I_y^{key} and combining them into one contour representation I^{con} by concatenating the orientation selective maps (see **Figure 2**, second row). The final form representation is then obtained by applying a logarithmic transformation emphasizing low range values and normalizing the response amplitudes, using the transformation:

$$I_{\log}^{\text{con}} = \log(1 + 5|I^{\text{con}}|) \quad (6)$$

$$I^{\text{form}} = \frac{I_{\log}^{\text{con}}}{\max(I_{\log}^{\text{con}})} \quad (7)$$

Likewise, for each key pose frame I^{key}, optical flow is separated into vertical and horizontal components and concatenated (see **Figure 2**, first row). The resulting motion representation I^{flo} is log-transformed and normalized. As for the contrast mapping, the transformation is given through:

$$I_{\log}^{\text{flo}} = \log(1 + 5|I^{\text{flo}}|)) \quad (8)$$

$$I^{\text{motion}} = \frac{I_{\log}^{\text{flo}}}{\max(I_{\log}^{\text{flo}})} \quad (9)$$

The form representations I^{form} and the motion representations I^{motion} are combined to an overall input representation I^{input} (**Figure 2**, last column). I^{input} is then used as an input for the training of the DCNN described in the following section.

3.2. Learning of Class Specific Key Pose Representations

A neuromorphic deep convolutional neural network was used to establish classwise representations of the preselected and suppress wrapping key pose frames using a supervised learning scheme. The network was implemented using the *energy-efficient deep neuromorphic networks* (*Eedn*) framework (Esser

[1]In the presented simulations, the MATLAB® implementation of the *Lucas-Kanade* flow estimation algorithm was used.

[2]The values of σ were chosen manually to take different temporal resolutions into account.

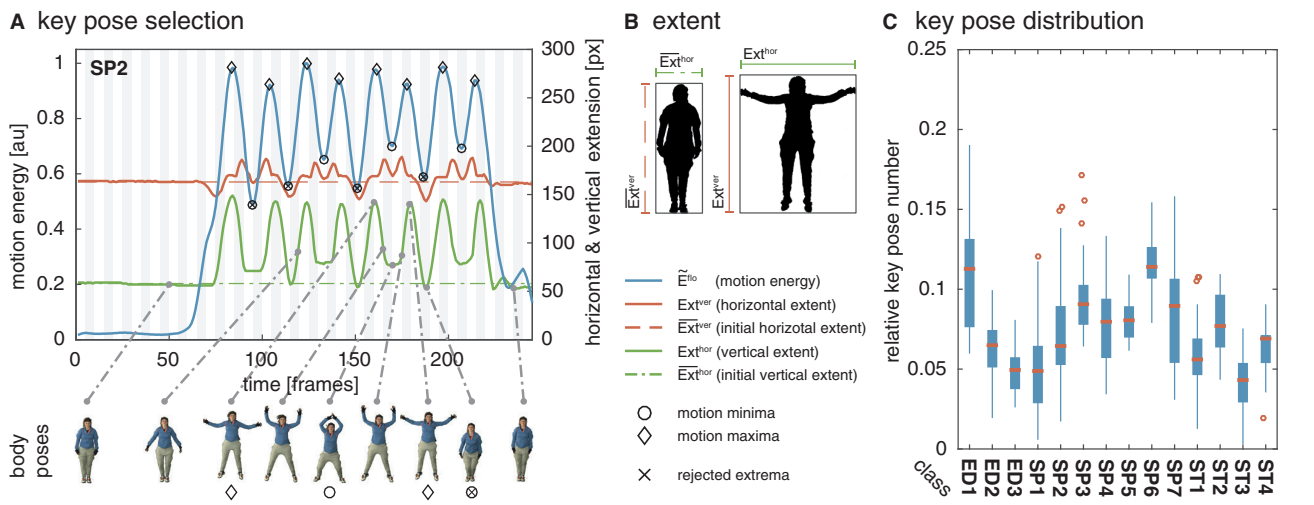


FIGURE 1 | Key pose selection. For each action sequence, key pose frames are selected by identifying minima and maxima in the motion energy which co-occur with a sufficiently increased extent of the body pose. In **(A)**, the smoothed motion energy $\tilde{E}^{\text{flo}}$ (blue) together with the horizontal (red) and vertical (green) extent of the body pose Ext^{hor} and Ext^{ver} and their reference values $\overline{\text{Ext}}^{\text{ver}}$ and $\overline{\text{Ext}}^{\text{hor}}$ (dashed, dash-dotted) are displayed for one of the actions used as part of the simulations (*SP2*/jumping jack). At the bottom, body poses for several frames are shown. Local minima in the motion energy are marked by a circle, local maxima by a diamond. Extrema which are rejected as a key pose frame because of an insufficient extent are additionally marked with ×. **(B)** Shows an example for the horizontal and vertical extent of a neutral and a highly articulated body pose. The first frame of each action sequence is defined to be the neutral pose. **(C)** Shows the relative number of identified key poses per action sequence for the uulmMAD dataset used for the simulations (see Section 4.1). Written informed consent for the publication of the exemplary images was obtained from the displayed subject.

et al., 2016), which adapts and extends the training and network functions of the *MatConvNet* toolbox (Vedaldi and Lenc, 2015). In the following for readers' convenience, we will briefly recapitulate and summarize key aspects of the framework and its extensions presented in Esser et al. (2016). In the framework, the weights established through learning match the representation scheme and processing principles used in neuromorphic computing paradigms. The structure of the DCNN follows one of the network parameter sets presented by Esser et al. (2016), which show a close to state-of-the-art classification performance on a variety of image datasets and allow the trained network to be run on a single IBM TrueNorth chip (Merolla et al., 2014).

A deep convolutional neural network is typically organized in a feedforward cascade of layers composed of artificial neurons (LeCun et al., 2010), which process the output of the proceeding layer (afferent synaptic connections) and propagate the result to the subsequent one (efferent synaptic connections). Following the definition in Esser et al. (2016), an artificial cell j in a DCNN calculates a weighted sum over the input to that cell, as defined by:

$$s_j = \sum_{xy} \sum_{f} in_{xyf} w_{xyfj}, \tag{10}$$

where in_{xyf} are the signals in the input field of cell j at locations (xy) in the spatial and (f) in the feature domain and, w_{xyfj} the respective synaptic weights. In the following, we will use the linear index i to denote locations in the (xyf) space-feature cube. Normalizing the weighted sum over a set of input samples (batch normalization) allows to accelerate the training of the network by standardizing s_j as defined through:

$$\tilde{s}_j = \frac{s_j - \mu_j}{\sigma_j + \epsilon} + b_j, \tag{11}$$

with $\tilde{s}_j$ the standardized weighted sum, μ_j the mean and σ_j the standard deviation of s calculated over the number of training examples within a batch (Ioffe and Szegedy, 2015). b_j is a bias term, allowing to shift the activation function $\phi(\bullet)$, and ϵ guarantees numerical stability. The output activation of the artificial neuron is calculated by applying an activation function on the standardized filter response:

$$r_j = \phi(\tilde{s}_j). \tag{12}$$

Weight adaptation is performed through gradient descent by applying error backpropagation with momentum (Rumelhart et al., 1986). In the forward phase, an input pattern is propagated through the network until the activations of the cells in the output layer are obtained. In the backward phase, the target values of an input pattern are used to calculate the cross entropy C given the current and the desired response of the output layer cell activations, as defined by:

$$C = -\sum_{j=1}^{M} \nu_j \ln(r_j) = -\sum_{j=1}^{M} \nu_j \ln(\phi(\tilde{s}_j)), \tag{13}$$

with M denoting the number of cells in the output layer. Here, ν_j is the one-hot encoded target value (or teaching signal) of a cell

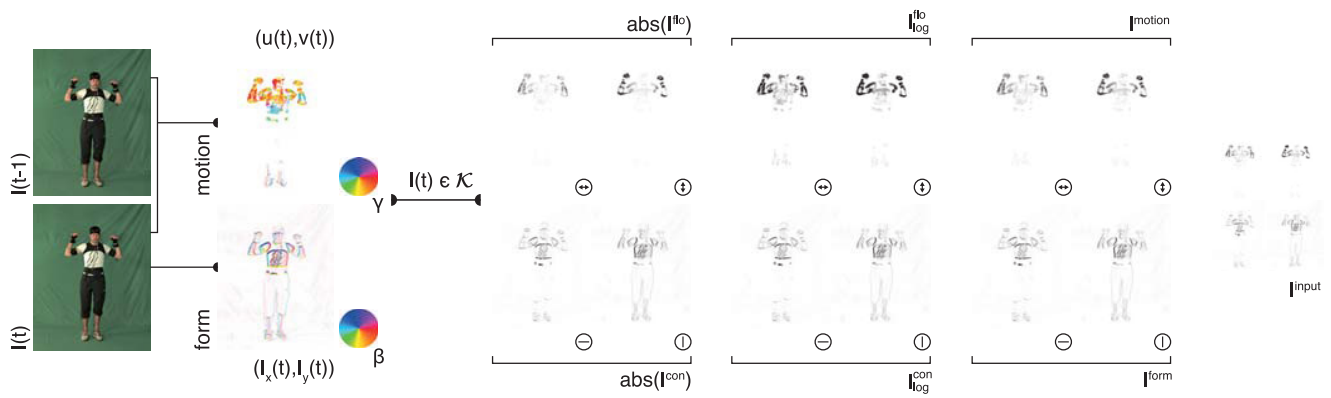


FIGURE 2 | Input generation. Each frame of an action sequence is transformed into a combined motion and form representation. In the top row, the estimated optical flow is displayed in the second column (direction γ color-encoded) for two consecutive frames (first column). The optical flow field is then separated into horizontal and vertical components (third column) and their absolute value is log transformed (forth column) and normalized (fifth column). Form representations (bottom row) are derived framewise by estimating the horizontal and vertical derivatives I_x and I_y (second column, gradient orientation with polarity β color-encoded). The resulting contrast images are then log-transformed and normalized. The form and motion representations are combined into a single feature map I^{input} which is then fed into the convolutional neural network. Image sizes are increased for a better visibility. Written informed consent for the publication of exemplary images was obtained from the shown subject.

j with activation r_j. A *softmax* function is employed as activation function in the output layer, as defined through:

$$\phi(\tilde{s}_j) = \frac{e^{\tilde{s}_j}}{\sum_{k=1}^{M} e^{\tilde{s}_k}}. \tag{14}$$

The cross entropy error $E(t) = C$ is then propagated backwards through the network and the synaptic weight adaptation is calculated for all cells in the output and hidden layers by applying the chain rule. The strength of weight adaptation Δw_{ij} is given through:

$$\Delta w_{ij}(t) = -\eta \frac{\partial E(t)}{\partial w_{ij}} + \alpha \Delta w_{ij}(t-1) = -\eta \delta_j in_i + \alpha \Delta w_{ij}(t-1), \tag{15}$$

$$\text{with } \delta_j = \begin{cases} (r_j - v_j) & \text{if } j \text{ is a neuron in the output layer} \\ \phi'(\tilde{s}_j) \sum_k \delta_k w_{jk} & \text{if } j \text{ is a neuron in a hidden layer,} \end{cases} \tag{16}$$

which includes a momentum term for smoothing instantaneous weight changes. Here, k is the index of cells in the layer succeeding cell j, t describes the current training step, or iteration, and η denotes the learning rate. The momentum factor $0 \leq \alpha \leq 1$ helps the network to handle local minima and flat plateaus on the error surface. After the backward pass, weights are finally adapted by:

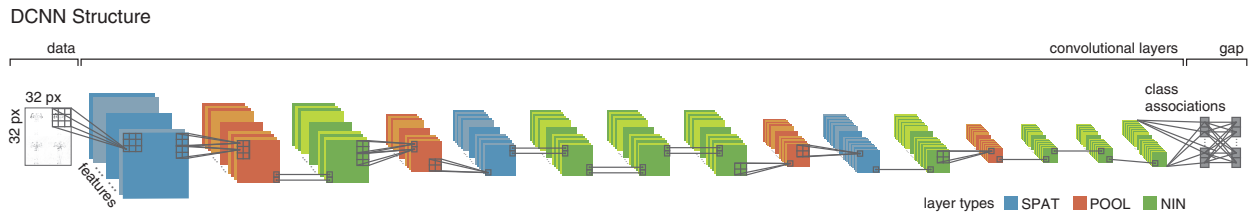
$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t). \tag{17}$$

To ensure the compatibility to neuromorphic processing principles, a binary activation function $\phi(\tilde{s}_j)$ is applied in the hidden layers (for details see Section 3.3).

Within a convolutional layer, weights w_{ij} of a cell j are shared over multiple input fields, which are arranged as a regular grid in the source layer. The calculation of the weighted sum during the forward, as well as the integration of the error derivative during the backward pass can be formulated as a convolution with the input from the source, or the error signal from the succeeding layer. The weights w_{ij} act as the filter (or convolution) kernel, $\tilde{s}_j$ as the filter response and r_j as the output of an artificial cell. The *size* and *stride* of a filter allow to adjust the size and the overlap of the input fields to a filter in the source layer. A small stride results in an increased overlap and thus a large number of output values. The number of *features* defines how many different filters are employed in a layer. The weight matrices of the cells within a layer can be separated into *groups* of filters, which define the set of input features from the source layer covered by a filter[3].

It is a common practice to construct deep neural networks by employing convolutional layers for feature extraction in the lower layers and connect them with (one or more) fully connected layers (equivalent to *Multilayer Perceptrons*/MLPs) on top for classification purposes. In contrast, the proposed network follows the strategy of *global average pooling* (gap) proposed in Lin et al. (2013) and applied in Esser et al. (2016). In the final convolutional layer of the network, one feature map is generated for each category of the classification problem. Instead of a full connectivity, the average value of each class-associated feature map is propagated to the output (softmax) layer. Due to their association to classes, the feature maps can directly be interpreted as confidence maps. Following the softmax layer, the cross-entropy error is calculated using one-hot encoded target values v_j and propagated back through the network (according to Equation 16). Networks using parameter-free *global average pooling* layers in combination with softmax are less prone to overfitting (compared to MLPs) and increase the robustness to spatial translations (for details see Lin et al., 2013).

[3] In **Figure 3**, the weight matrices in the convolutional layer 5 have a dimensionality of $3 \times 3 \times 8$, since they receive input from 256 feature maps in layer 4 which are separated into 32 groups of filters, each receiving input from 8 feature maps.



layer	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[prediction layer]	[softmax layer]
size	3 x 3	4 x 4	1 x 1	2 x 2	3 x 3	1 x 1	1 x 1	1 x 1	2 x 2	3 x 3	1 x 1	2 x 2	1 x 1	1 x 1	1 x 1		
stride	1	2	1	2	1	1	1	1	2	1	1	2	1	1	1		
features	12	252	256	256	512	512	512	512	512	1,024	1,024	1,024	1,024	1,024	2,016		
groups	1	2	2	8	32	4	4	4	16	64	8	32	8	8	8		
# cells	12,288	64,512	65,536	16,384	32,768	32,768	32,768	32,768	8,192	16,384	16,384	4,096	4,096	4,096	8,064	# classes	

FIGURE 3 | Deep convolutional neural network structure. The implemented DCNN follows the structure proposed in Esser et al. (2016) and employs three different convolutional layer types (layers 1–15). *Spatial layers* (SPAT; colored in blue) perform a linear filtering operation by convolution. *Pooling layers* (POOL; colored in red) decrease the spatial dimensionality while increasing the invariance and diminishing the chance of overfitting. *Network-in-network* layers (NIN; colored in green) perform a parametric cross channel integration (Lin et al., 2013). The proposed network consists of a data (or input) layer, 15 convolutional layers and a prediction and softmax layer on top. Each of the cells in the last convolutional layer (layer 15) is associated with one class of the classification problem. In the prediction layer, the average class-associated activations are derived (*global average pooling*/gap) and fed into the softmax layer (i.e., one per class), where the cross-entropy error is calculated and propagated backwards through the network. The parameters used for the convolutional layers of the network are given in the central rows of the table. In the last row, the number of artificial cells per layer is listed. The cell count in the *prediction* and *softmax* layer depends on the number of categories of the classification task (i.e., the number of actions in the dataset).

The employed network consists of 15 convolutional layers, which implement three different types of convolutional operations. *Spatial* layers (SPAT) perform a standard convolution operation, *pooling* layers (POOL) reduce the spatial dimensions by applying a convolution with a large stride (Springenberg et al., 2014), *network-in-network* layers (NIN) are realized by convolutional layers with a size of $1x1$ and a stride of 1 and act as cross channel integration layers (Lin et al., 2013). The network structure is summarized in **Figure 3**. Each of the cells in the last convolutional layer (layer 15) is assigned to one class. During learning, activities in this layer are averaged per feature map and fed into the *softmax* layer. For recognition, the average output of the cell populations associated to the individual classes are used as prediction values and serve as the final output r_c^{class} of the network (*prediction* layer in **Figure 3**).

3.3. Neuromorphic Implementation

Processing actual spikes in hardware, the execution of a DCNN on a neuromorphic platform poses several constraints on the activity and weight representation schemes. Since the processing architecture of the TrueNorth neuromorphic platform is based on event-based representations, the gradual activations need to be mapped onto a spike-based mechanism. To be in conformity with these processing principles, Esser et al. (2016) employ a binary activation function, as defined by:

$$\phi(\tilde{s}_j) = \begin{cases} 1 \text{ if } \tilde{s}_j \geq 0 \\ 0 \text{ otherwise,} \end{cases} \tag{18}$$

and ternary synaptic weights ($w_{xyf} \in \{-1, 0, 1\}$). For the backpropagation of the error signal, the derivative of the binary activation is approximated linearly in the range of $[0, 1]$, as given through:

$$\frac{\partial \phi(\tilde{s}_j)}{\partial \tilde{s}_j} \approx \max(0, 1 - |\tilde{s}_j|). \tag{19}$$

During training, a copy of the model weights is held in a *shadow network*, which allows gradual weight adaptation. Weight updates are performed on values in the shadow network using high precision values. For the forward and backward pass, the hidden weights w_{ij}^h in the shadow network are clipped to $[-1, 1]$ and mapped to the ternary values using rounding and hysteresis, following:

$$w_{ij}(t) = \begin{cases} -1 & \text{if } w_{ij}^h(t) \leq -0.5 - h \\ 0 & \text{if } w_{ij}^h(t) \geq -0.5 + h \wedge w_{ij}^h(t) \leq 0.5 - h \\ 1 & \text{if } w_{ij}^h(t) \geq 0.5 + h \\ w_{ij}(t-1) & \text{otherwise} \end{cases} \tag{20}$$

(for details refer to Esser et al., 2016). The hidden weights w_{ij}^h allow the synaptic connection strengths to switch between the ternary values based on small changes in the error gradients obtained during backpropagation, while the hysteresis factor h prevents them from oscillating. The parameters for the training of the network were chosen according to Esser et al. (2016), using a momentum factor of $\alpha = 0.9$ and a learning rate of $\eta = 20$ (reduced by a factor of 0.1 after 2/3 and 5/6 of the total training iterations). The hysteresis factor h was set to 0.1. The mapping of the training network on the TrueNorth platform was performed by the *Eedn* framework. Training was carried out on

Nvidia GPUs, testing was performed on the IBM TrueNorth NS1e board.

The IBM TrueNorth chip consists of 4,096 interconnected neurosynaptic cores with 1 million spiking neurons and 256 million configurable synaptic connections. For the execution of the network on the TrueNorth chip, the trained network parameters are mapped to hardware using an abstraction of a TrueNorth program called *Corelet* (Amir et al., 2013). The platform independent *Corelets* translate the network parameters into a TrueNorth specific configuration, which can be used to program the parameters of the neurons and synaptic connection strengths on the chip. For details on *Corelets* and the mapping of the DCNN on neuromorphic hardware platforms refer to Amir et al. (2013); Esser et al. (2016).

3.4. Temporal Integration of Framewise Class Predictions

After the training of the DCNN, classification is either performed framewise by directly selecting the class corresponding to the cell population in layer 15 with the maximum average activation, or by integrating the individual framewise classification results using majority voting in temporal windows or over the full sequence.

For framewise classification, a key pose frame is identified in an input image sequence $I(\mathbf{x}, t)$ and preprocessed as described in Section 3.1. The resulting input map I^{input} is fed into the DCNN and the class label c associated to the cell population in layer 15 with the maximum average output r_c^{class} defines the class prediction for I^{input}. The value of r_c^{class} can directly be interpreted as the confidence in the prediction.

In sliding window based classification, the predicted class labels for key pose frames are collected within temporal windows of size n [frames], which are shifted over the input sequence $I(\mathbf{x}, t)$. The class with the most frequent occurrence of key pose frames determines the class predicted for the window (majority voting). At the moment, we do not use the confidence r_c^{class} of the predictions as weights for the voting. Note that it is not guaranteed, that key pose frames occur in all temporal windows. Windows which do not contain key poses are not used for evaluation.

Full sequence classification follows the same principle as sliding window based classification, but collects all key pose frames within a sequence. Thus, the amount of temporal information integrated in the voting process might differ substantially from sequence to sequence.

4. DATASETS

The proposed action recognition approach was evaluated using two different action datasets. Due to the higher number of subjects and actions, we focused our analysis on the *uulm multiperspective action dataset* (uulmMAD). In addition, we analyzed the performance on the widely used *Weizmann* dataset to allow a comparison to other approaches and to perform a cross-dataset evaluation of overlapping classes. In the following, we will briefly describe the main characteristics of the two datasets.

4.1. uulmMAD

The *uulm multiperspective action dataset*[4] (uumlMAD; Glodek et al., 2014) consists of data from 31 subjects performing actions from the areas of everyday life (ED), sport/fitness (SP) and stretching (ST). Eight of the actions are repeated three times, six actions are performed four times with varying speed. Altogether, each action is performed either 93 or 124 times. Actions were recorded in front of a greenscreen using three synchronized cameras and the body posture was captured in parallel by an inertial motion capturing system worn by the subjects. To decrease the likelihood of similar visual appearances, the motion capture suit was covered by additional clothes whenever possible. **Figure 4** shows the 14 actions together with a characteristic picture, an abbreviation and a short description for each action.

[4] Available via https://www.uni-ulm.de/imagedb.

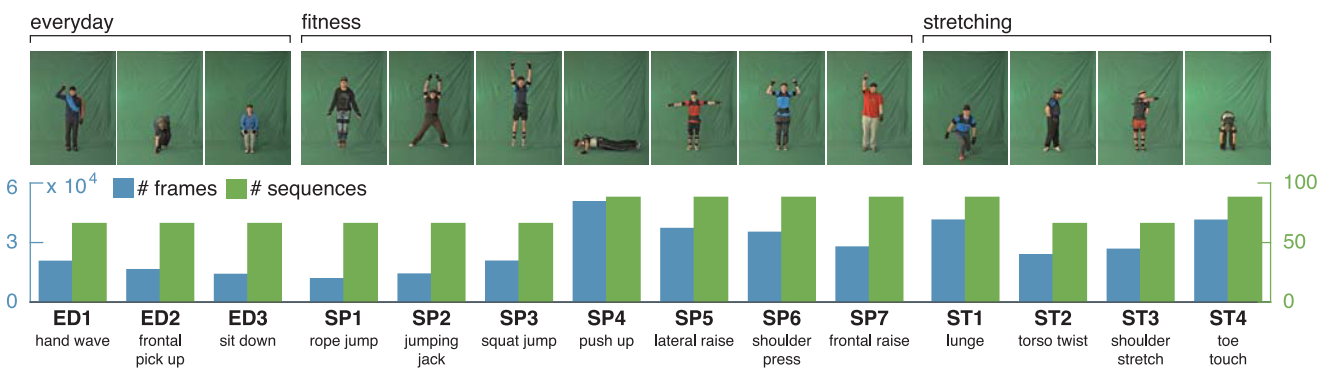


FIGURE 4 | uulmMAD – uulm multiperspective action dataset. The uulmMAD dataset contains 14 actions in the area of everyday activities, fitness/sports and stretching performed by 31 subjects. Per subject, eight of the actions are repeated three times, six actions are performed four times with varying speed. Actions were recorded by three synchronized cameras (frontal, diagonal and lateral) with a frame rate of 30 Hz and an inertial motion capturing system with a sample rate of 120 Hz. Silhouettes were extracted using chromakeying. At the time we carried out the simulations, silhouettes were available for 22 subjects. In the first row exemplary pictures are shown for all actions. The number of videos (green) and total sum of frames (blue) which were available for the evaluation are displayed in the second row. At the bottom, an abbreviation for each action is defined and a short description is given. Written informed consent for the publication of exemplary images was obtained from the displayed subjects.

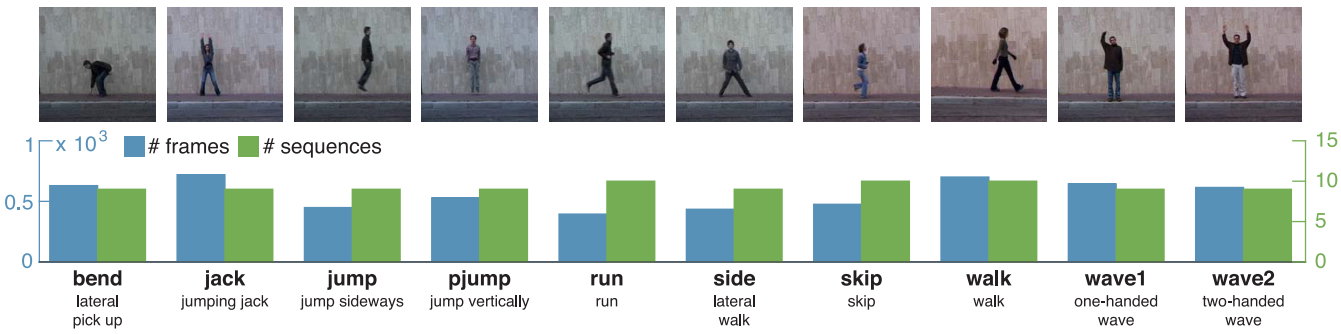


FIGURE 5 | Weizmann action dataset. The Weizmann dataset (Gorelick et al., 2007) is one of the most commonly used action recognition datasets and consist of ten actions, recorded for nine subjects. Actions are performed once (occasionally twice) per subject in front of a static background. Silhouette representations are provided for all sequences. Representative images are displayed alongside with the number of frames and sequences, a label and a short description per class.

At the time we carried out the simulations, silhouette representations were available for all sequences of 22 subjects. Since the silhouettes are used to calculate an estimate of the horizontal and vertical extent of a pose, only the frontal recordings of this subset of subjects were used within the evaluation. Some action pairs (e.g., ED2 and ST4) in the dataset are deliberately intended to appear visually similar and thus be difficult to separate. In total, the sequences used for evaluation contain 381, 194 frames, of which 28, 902 are selected by the key pose selection procedure.

4.2. Weizmann Dataset

To allow a comparison with different action recognition approaches, simulations were additionally carried out using a well established action dataset. The *Weizmann dataset*[5] (see **Figure 5**; Gorelick et al., 2007) consists of ten actions performed by nine subjects. Actions are mostly performed once per subject, although some actions are occasionally performed twice. Actions are captured in 25 *Hz* from a frontoparallel perspective in front of a uniform background.

Silhouettes are available for all subjects and sequences. In total, the sequences contain 5, 594 frames, 1, 873 of which are identified as key pose frames by using the procedure described in Section 3.1.

5. RESULTS

Several simulations were carried out to evaluate the performance of the proposed key pose based action recognition approach. The simulations were intended to address questions related to (I) the overall performance of the approach on different datasets using a framewise, as well as windowed and full sequence majority voting recognition schemes, (II) a comparison to other action recognition methods, (III) a juxtaposition of key pose based and full sequence learning, and (IV) cross-dataset evaluation. Since action recognition datasets—in particular, in case of framewise recognition—are often highly imbalanced, we provide different types of performance measures, as well as classwise performance values for the most essential results. Since the nomenclature and definition of performance measures vary largely in the pattern recognition and machine learning community we will briefly define and describe the reported measures to allow a better comparability. For a comprehensive discussion on performance measures, we refer to Sokolova and Lapalme (2009) and the contributions of D. Powers, e.g. (Powers, 2013).

[5]Available via http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html.

TABLE 1 | Performance measures.

Measure	Abbreviation	Definition
Recall	$\mathrm{Rec_M}$	$\frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{tp}_i}{\mathrm{tp}_i+\mathrm{fn}_i}$
Informedness	$\mathrm{Inf_M}$	$\sum_{i=1}^{N}\frac{\mathrm{tp}_i+\mathrm{fp}_i}{\mathrm{tp}_i+\mathrm{fn}_i+\mathrm{tn}_i+\mathrm{fp}_i}\times(\frac{\mathrm{tp}_i}{\mathrm{tp}_i+\mathrm{fn}_i}+\frac{\mathrm{tn}_i}{\mathrm{fp}_i+\mathrm{tn}_i}-1)$
Markedness	$\mathrm{Mark_M}$	$\sum_{i=1}^{N}\frac{\mathrm{tp}_i+\mathrm{fn}_i}{\mathrm{tp}_i+\mathrm{fn}_i+\mathrm{tn}_i+\mathrm{fp}_i}\times(\frac{\mathrm{tp}_i}{\mathrm{tp}_i+\mathrm{fp}_i}+\frac{\mathrm{tn}_i}{\mathrm{fn}_i+\mathrm{tn}_i}-1)$
Matthews Correlation	$\mathrm{MCC_M}$	$\pm\sqrt{\mathrm{Mark_M}\times\mathrm{Inf_M}}$

In a multiclass classification problem with N classes tp_i (true positives) are commonly defined as the number of correct acceptances (hits) for a class C_i ($i \in [1, ..., N]$), fn_i as the number of false rejections (misses), tn_i as the number of correct rejections of samples of different classes $\mathrm{C}_{j\neq i}$ and fn_i (false negatives) as the number of false acceptances (false alarms). Together, these four counts constitute the confusion matrix and allow to derive a variety of measures describing the performance of a trained classification system. The ones used for the evaluation of the presented results are listed alongside with an abbreviation and their definition in **Table 1**.

All multiclass performance measures are calculated using macro averaging (M), since using micro averaging, classes with a large number of examples would dominate the averaging. $\mathrm{Rec_M}$, often referred to as (average) recognition rate or somewhat misleading as (classification) accuracy, might be the performance measurement most frequently used in the action recognition literature and describes the average percentage of correctly identified positive examples per class. $\mathrm{Inf_M}$ reflects how informed the decision of a classifier is in comparison to chance, whereas $\mathrm{Mark_M}$ follows the inverse concept by describing how likely the prediction variable is marked by the true variable

(Powers, 2013). Note, that when calculating the average per class values of $\mathrm{Inf_M}$ and $\mathrm{Mark_M}$ are weighted by the $\mathrm{Bias}_i = \frac{\mathrm{tp}_i+\mathrm{fp}_i}{\mathrm{tp}_i+\mathrm{fn}_i+\mathrm{tn}_i+\mathrm{fp}_i}$ and the $\mathrm{Prevalence}_i = \frac{\mathrm{tp}_i+\mathrm{fn}_i}{\mathrm{tp}_i+\mathrm{fn}_i+\mathrm{tn}_i+\mathrm{fp}_i}$, respectively. The Matthews Correlation Coefficient $\mathrm{MCC_M}$ can be derived by calculating the geometric mean of $\mathrm{Inf_M}$ and $\mathrm{Mark_M}$ and expresses the correlation between predicted classes and true values.

Leave-one-subject-out cross-validation (LOSO) was performed in all test scenarios and the resulting average performance measures are reported together with the corresponding standard deviations. In the following, rates are either reported in a range of [0, 100] or [0, 1] (due to limited space).

5.1. Classification Performance

The equivalent network structure (see Section 3.2) was used to train the network on the two datasets described in Section 4. In case of the uulmMAD dataset, 28,902 key pose frames (per class average 2,064.43, std 1,097.16) were selected and used as the training input. 576 cells in the last convolutional layer (layer 15) of the CNN were assigned to each of the 14 classes in the dataset. The network was trained in 150,000 iterations. Testing was performed using the preselected key pose frames of the test subject as input. The average population activation of the cells assigned to each class was used to infer the final classification decision (for an exemplary activation pattern see **Figure 8**). **Figure 6** summarizes classification results obtained for different temporal integration schemes of single frame classification results. A framewise classification scheme allows to recognize an action in an instant when the key pose frame is presented to the network. This kind of immediate decision might be crucial for systems which rely on decisions in real time. Not only the processing speed, but also the time necessary to sample and construct the action descriptors is relevant in this context. **Figure 6A** summarizes the framewise

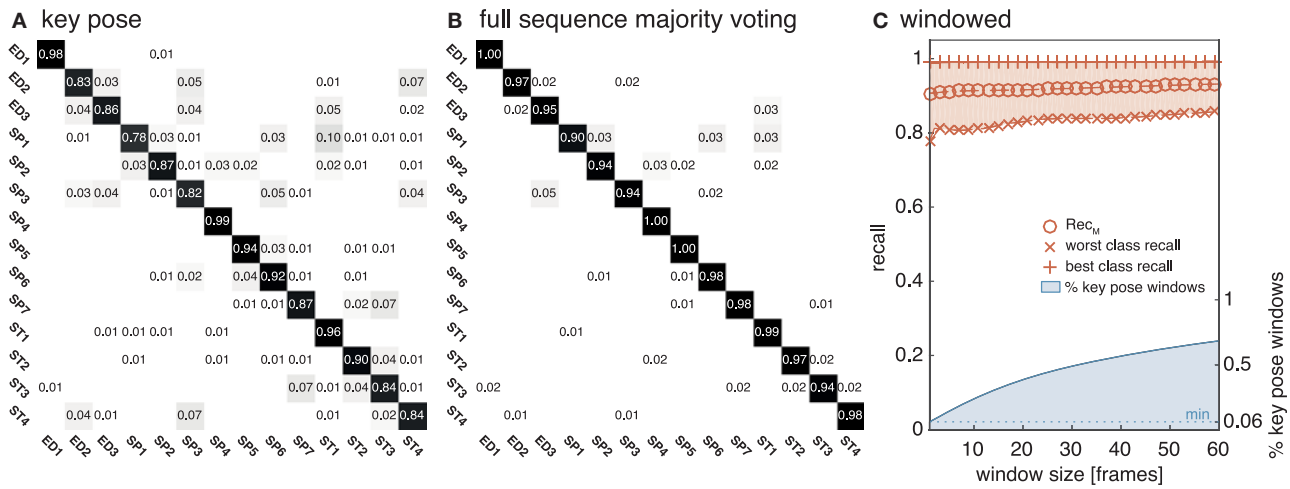


FIGURE 6 | uulmMAD classification performance. The proposed network was trained on the key pose frames extracted from the uulmMAD action recognition dataset. **(A)** Shows the per class classification rates obtained by single key pose frame classification. This allows the recognition of an action in the instant a key pose frame emerges in the input sequence. Average classwise recall (on the diagonal) ranges from 0.78 to 0.98. Some of the notable confusions between classes can be explained by a large visual similarity (e.g., between ED2 and ST4). In **(B)** sequence level majority voting was applied. The final decision is made after evaluating all key pose frames within an action sequence and determining the class with the most frequent occurrence of key poses. The resulting per class values of $\mathrm{Rec_M}$ range from 0.94 to 1.00. A sliding window based classification scheme was evaluated in **(C)**. The best and worst per class average recall values together with the average value of $\mathrm{Rec_M}$ are displayed for temporal window sizes from 1 to 60 frames. In addition, the percentage of windows containing one or more key pose frames (and thus allow a classification of the action) is shown (blue line).

TABLE 2 | uulmMAD classification performance.

			$\mathrm{Rec_M}$	$\mathrm{Inf_M}$	$\mathrm{Mark_M}$	$\mathrm{MCC_M}$
Single		All frames	67.56 ± 6.06	0.703 ± 0.062	0.762 ± 0.041	0.732 ± 0.051
		Key poses	88.65 ± 5.66	0.915 ± 0.045	0.915 ± 0.043	0.915 ± 0.044
Majority	Windowed*	5 [4]	89.64 ± 4.97	0.919 ± 0.041	0.920 ± 0.040	0.920 ± 0.040
		10 [9]	89.98 ± 4.69	0.922 ± 0.039	0.922 ± 0.039	0.922 ± 0.039
		20 [19]	90.47 ± 4.48	0.924 ± 0.038	0.924 ± 0.037	0.924 ± 0.037
		Full sequence	96.73 ± 2.84	0.981 ± 0.014	0.970 ± 0.033	0.975 ± 0.021

Size [overlap]

classification rates per class (average Rec_M of 0.887, std 0.057). Some of the confusions between classes might be explained by similar visual appearances of the key poses (e.g., ED2 and ST4). Accumulating the classified key poses over a sequence by majority voting increases the classification performance (average Rec_M of 0.967, std 0.028, compare **Figure 6B**), but requires to analyze all frames of a sequence and is thus not well suited for real time applications. As a compromise between classification speed and performance, a sliding window based approach was evaluated. In **Figure 6C**, the best and worst average per class recall is displayed together with the Rec_M for window sizes of $n = [1, ..., 60]$, each with an overlap of $n - 1$. In addition, the relative number of windows which contain at least one key pose (and thus allow a classification) is shown. **Table 2** summarizes the classification performance for different single frame and temporal integration schemes. Single frame performance is, in addition, reported for the evaluation of not only the key pose but the full set of frames. As can be seen, the classification performance decreases significantly but the average recall of Rec_M of 67.56 (std 6.06) indicates, that the learned key pose representations are still rich enough to classify a majority of the frames correctly. Note, that the relative number of correct classifications clearly exceeds the percentage of key pose frames in the dataset (per class average of 7.46 %, std 2.19 %, compare **Figure 1C**).

The model was additionally trained using the Weizmann dataset (Gorelick et al., 2007, see Section 4.2). 1,873 frames (per class average 187.30, std 59.51) were selected as key pose frames utilizing the combined criterion developed in Section 3.1. Except for the number of output features encoding each class (806), the same network and learning parameters were applied. As for the uulmMAD dataset, **Figure 7** gives an overview over the classification performance, by showing confusion matrices for single key pose frame evaluation (**Figure 7A**), full sequence majority voting (**Figure 7B**), as well as best and worst class recall for different sized windows of temporal integration (**Figure 7C**). In comparison to the results reported for the uulmMAD dataset, the gap between the best and worst class recall is considerably increased. This might be explained by a different overall number of available training examples in

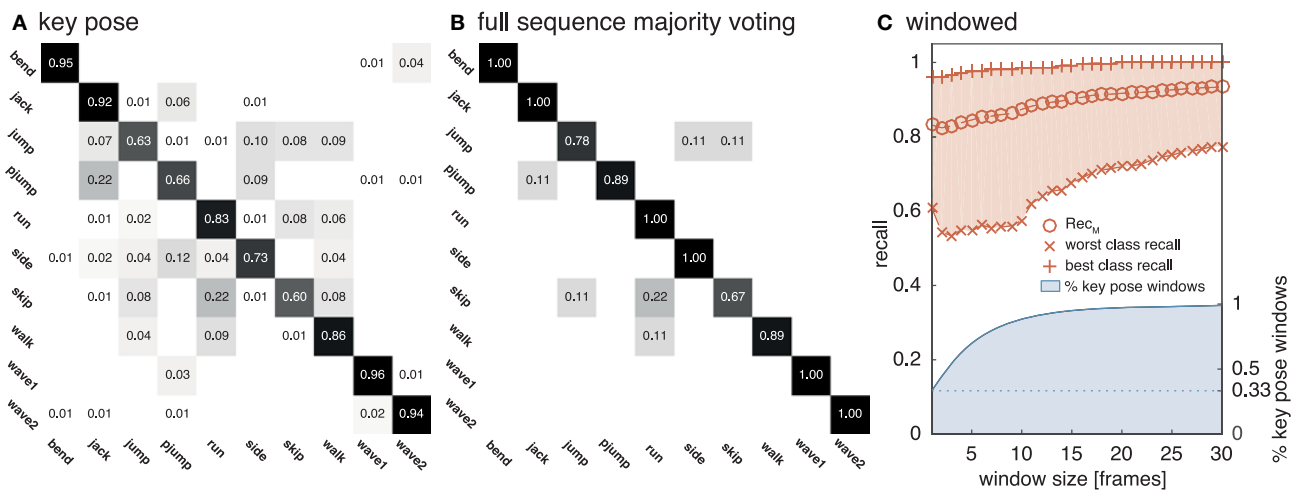


FIGURE 7 | Weizmann classification performance. The network was evaluated on the Weizmann dataset to allow a comparison to other approaches. As in **Figure 6**, **(A)** shows the classifications rates for a classification of single key pose frames per class. **(B)** Displays classwise recognition results for a full sequence evaluation using majority voting. Similar visual appearances might explain the increased rate of confusions for some of the classes (e.g., *run* and *skip*). In **(C)** the average best and worst per class recall values and Recl_M are reported for temporal window sizes between 1 and 30 frames together with the relative number of windows which contain at least one frame classified as key pose.

TABLE 3 | Weizmann classification performance.

			Rec_M	Inf_M	Mark_M	MCC_M
Single		All frames	77.15 ± 6.46	0.810 ± 0.056	0.794 ± 0.068	0.801 ± 0.061
		Key poses	82.15 ± 5.81	0.844 ± 0.061	0.827 ± 0.070	0.835 ± 0.065
Majority	Windowed*	5 [4]	83.50 ± 5.12	0.877 ± 0.043	0.860 ± 0.053	0.868 ± 0.047
		10 [9]	86.40 ± 5.58	0.920 ± 0.027	0.878 ± 0.067	0.899 ± 0.044
		20 [19]	90.35 ± 7.34	0.966 ± 0.023	0.898 ± 0.093	0.930 ± 0.057
		Full sequence	92.22 ± 8.33	0.980 ± 0.023	0.879 ± 0.128	0.927 ± 0.079

Size [overlap]

the datasets (the per class average of training examples in the uulmMAD dataset exceeds the Weizmann dataset by a factor of 11.02), higher visual similarities between the classes (the most prominent confusions are observed for *skip*, *jump* and *pjump*), the lack of a sufficient number of descriptive key poses, or a combination hereof. A direct relationship of the classwise performance and the per class number of key pose frames available for training cannot be observed. Even though the least number of key pose frames was extracted for the class *bend*, the second best recall value was achieved. As for the uulmMAD dataset, performance measures are reported for different single frame and temporal integration schemes in **Table 3**. Again, the trained key pose representations achieve a considerable performance even when tested per frame on all frames of the action sequences ($\text{Rec}_\text{M} = 77.15$, std 6.46). **Table 4** compares the reported classification results on the Weizmann dataset to state-of the art single frame based (second block) and sequence level approaches (third block). In particular, other key pose based action recognition approaches are listed (first block). The direct comparison of different classification architectures, even when evaluated on the same dataset, is often difficult, since different evaluation strategies may have been applied. Thus, whenever possible, the number of considered classes (sometimes the class *skip* is excluded) and the evaluation strategy is listed together with classification performance and speed. Evaluation strategies are either leave-one-subject-out (LOSO), leave-one-action-out (LOAO) or leave-one-out (LOO, not specifying what is left out) cross-validation.

On a sequence level, the classification performance of the proposed approach is on par with almost all other key pose based methods. Only Liu et al. (2013) achieved a noteworthy higher performance (recall of 100). It is important to stress that the compared methods substantially differ in their key pose selection procedures and thus in the underlying conceptual definition of key poses. For example, Weinland and Boyer (2008) and Liu et al. (2013) select key poses that maximize the classification performance in a validation subset of the dataset, whereas (Baysal et al., 2010; Cheema et al., 2011) select and weight candidate pose descriptors dependent on their distinctiveness with respect to the other classes contained in the dataset. In Chaaraoui et al. (2013), key poses are selected independently per class using clustering in combination with a compactness metric. All the above mentioned approaches, except the last one, rely on *inter-class* distributions of pose descriptors to identify key poses, implicitly stating that representativeness is equivalent to distinctiveness (among a known set of classes). If the task at hand is to separate an a priori defined set of actions, this seems to be the superior way of defining key poses for the establishment of temporally sparse representations of actions. On the other hand such poses always describe differences based on comparisons and do not necessarily capture characteristic poses of an action.

The presented approach follows a different principle. Certain properties of image or skeleton based pose features are assumed

TABLE 4 | Weizmann comparison to other approaches.

Catagory		# Actions	Evaluation	fps	Temporal range		
					Sub-sequence		Full sequence
					Recall	# Frames	Recall
Key pose	Weinland and Boyer, 2008	10	LOSO	–	–	–	93.6
	Baysal et al., 2010	9	LOO	–	–	–	92.6
	Cheema et al., 2011	9	LOO	–	–	–	91.6
	Chaaraoui et al., 2013	9	LOSO	124	–	–	92.8
	Liu et al., 2013	10	LOSO	–	–	–	**100**
Single frame	Niebles and Fei-Fei, 2007	9	LOSO	–	55	1	72.8
	Fathi and Mori, 2008	10	LOO	0.25–5	**99.9**	1	**100**
	Schindler and van Gool, 2008	9	LOSO	–	93.5	1	**100**
	Hoai et al., 2011	10	LOSO	–	87.7	1	–
Full sequence	Jhuang et al., 2007	9	–	0.83	–	–	98.8
	Klaser et al., 2008	10	LOSO	–	–	–	84.3
	Grundmann et al., 2008	9	LOAO	–	–	–	94.6
	Ikizler and Duygulu, 2009	9	LOO	–	–	–	**100**
	Bregonzio et al., 2009	10	LOSO	–	–	–	96.7
	Sun and Liu, 2012	10	LOO	–	–	–	97.8
	Beaudry et al., 2016	10	LOO	51.63	–	–	**100**
	Presented approach	10	LOSO	**1,000**	82.2	1	92.2

Bold values indicate maximum recall/fps values per column.

to co-occur with characteristic body configurations and thus are used to identify key pose frames. The feature characteristic indicating a key pose and the representations/descriptors used for the recognition of a pose do not necessarily have a close relationship. In doing so, we accept the fact that the selected poses are not guaranteed to be very distinctive and some even may occur in more than one action in exactly the same way. Key poses are assumed to be the most representative poses of a particular action, not in comparison, but in general. Nevertheless, the presented results demonstrate that a *feature-driven*, pose centered key pose selection mechanism is capable of achieving the same level of performance, without loosing generality.

Most key pose based approaches in the literature try to assign single frames of an image sequence to key pose frames with a high similarity, temporally integrate the result (e.g., by using histograms or majority voting) and perform a classification of the action on a sequence level. The result of single frame action recognition based on the extracted key poses (directly linking key poses to actions) is rarely reported. Single frame based approaches (see **Table 4**, second block), however, try to perform action classification using information solely extracted within one frame (two frames if optical flow is part of the descriptor) and achieve impressive results. In direct comparison, the single frame performance of the presented approach (Rec_M of 82.15 for key pose evaluation and 77.15 for the classification of all single frames, compare **Table 3**) cannot compete with the other methods, which, on the contrary, utilize all frames during learning to maximize classification performance in the test training dataset. The presented approach, however, achieves a single frame performance of $\text{Rec}_\text{M} = 77.15$ when evaluated over all frames, although in case of the Weizmann dataset only a per class average of 33.84 % (std 8.63 %) of all frames is used for training.

In the third block of **Table 4**, selected approaches performing action recognition on a sequence level using a variety of different representations and classification architectures are listed. Note that in an overall comparison, (I) due to the transfer on neuromorphic hardware, the presented approach achieves the highest processing speed[6] while consuming a minimal amount of energy, and (II) due to fact, that we aim at executing the model on a single TrueNorth chip we only use input maps with a resolution of 32 × 32 (using 4,064 of the 4,096 cores available on one chip). This is no limitation of the employed *Eedn* framework, which allows to realize models which run on systems with more than one chip (Esser et al., 2016; Sawada et al., 2016). An increased input resolution, as well as the use of more than two flow direction and contour orientation maps might help in separating classes with a high visual similarity (e.g., *skip*, *jump*, and *run*).

5.2. Comparison to Full Sequence Learning

To address the question whether and how the proposed classification architecture might benefit from using all frames (as opposed to only key pose frames) during training, we performed exactly the same training and testing procedure twice on the uulmMAD dataset. First, only key pose frames were presented during training, while second, all frames were provided during the training phase. Likewise, testing was performed just on the preselected key pose frames, as well as the full set of frames. **Table 5** compares the average recall under the different training (rows) and testing conditions (columns) for single frame evaluation and sequence level majority voting.

In both cases, training and testing on key pose frames achieves the highest performance. However, the observed differences between the two training conditions could not shown to be significant, neither when testing on key poses nor on the full set of frames. Nevertheless, having a closer look at the activation patterns of the network reveals some insights on the effectiveness of the two variants of trained representations. **Figure 8** shows the average activation pattern of the 14 cell populations in layer 15 assigned to the individual classes of a network trained on key pose frames and tested on all frames of the action SP2 (jumping jack). The displayed activation levels clearly show how

TABLE 5 | uulmMAD key pose versus all frame learning.

Train \ Test	Framewise		Majority voting	
	Key poses	All frames	Key poses	All frames
Key poses	**88.65 ± 5.66**	67.56 ± 6.06	**96.73 ± 2.84**	93.29 ± 7.05
All frames	85.84 ± 7.15	72.84 ± 8.25	95.27 ± 3.93	94.70 ± 7.03

Bold values indicate the maximum average recall for framewise and full sequence majority voting classification schemes.

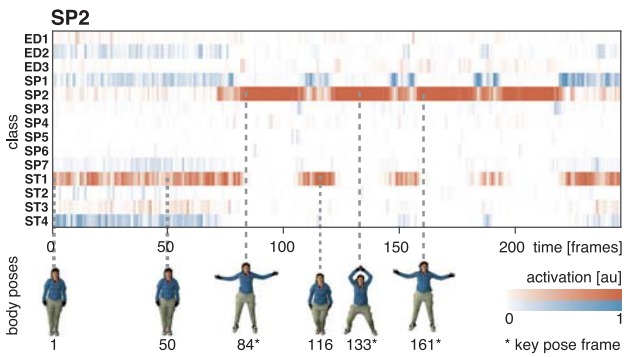


FIGURE 8 | Activation of cell populations. The activations of the cell populations in the last convolutional layer of the DCNN assigned to the 14 classes of the uulmMAD dataset are displayed for a network trained only on key pose frames and tested on all frames of the action SP2 (jumping jack). The activation level of the cell population with the maximum activation (red) and the remaining populations (blue) is encoded by color intensity. Corresponding poses are displayed for selected frames (bottom row). Key pose frames are marked by asterisks. The activation pattern shows how the cell population assigned to the class SP2 selectively responds to frames in the temporal neighborhood of the corresponding key pose frames. At the beginning and the end of the sequence, as well as in between the occurrence of key pose frames, different cell populations achieve the maximum average activation and thus result in misclassifications. Written informed consent for the publication of exemplary images was obtained from the shown subject.

[6]Image preprocessing and key pose selection is not integrated in the estimated processing time. Optical flow estimation can be performed on a second TrueNorth chip (Brosch and Neumann, 2016).

the trained representations of the corresponding class selectively respond within the temporal neighborhood of the key pose frames. Frames sampled from periods without the presence of key pose frames (at the beginning and the end of the sequence, as well as in between key pose frames) result mostly in a large activation of other cell populations and thus in misclassifications. This is in line with the results shown in **Table 5**, which indicate that classification performance increases under both training conditions when testing is only performed on key pose frames. At this point we can conclude that, compared to a training on the full set of frames, key pose based learning of class specific representations at least performs at an equal level. Whether there is any benefit of training exclusively on key pose frames next to an increased learning speed, remains, however, an open question. **Figure 9** summarizes the per class activation levels of the cell populations which resulted in a correct classification. For almost all classes (except ED3), the activation level is significantly increased when training was performed on key pose frames only. This might become a very important property in situations where it is not an option to accept any false negatives. Applying a threshold on the activation levels would allow to eliminate false negatives, while key pose based training would decrease the number of positive examples rejected by the fixed threshold. Thus, thresholding might further increase the performance for the key pose based training reported so far. Taken together, key pose based learning achieves a slightly increased classification performance with an increased selectivity of the cell populations and thus a higher confidence of the classification decisions.

5.3. Cross-Dataset Evaluation

Learning to classify input samples and the associated representations is conducted with the aim to robustly predict future outputs and, thus, generalize for new input data. Here, we evaluate such network capability by evaluating the classification of the trained network using input data across different datasets. More precisely, cross-dataset evaluation was performed to evaluate how the learned representations generalize over different datasets. The preselected key pose frames of the uulmMAD and the Weizmann dataset were used for both training and testing constellations. Performance is reported for two classes, one being *one-handed wave* (ED1 and *wave1*), which is available in both datasets. The second class was formed by combining the visually similar classes SP2/SP6 and *jack*/*wave2* during evaluation into one joint class *raising two hands*. Training was performed on the full set of classes in both cases. Thus, for *one-handed wave* a random guess classifier would achieve a recall of either 7.14 (uulmMAD) or 10.00 (Weizmann). In case of the combined class *raising two hands*, the recall chance level increases to 14.29 (uulmMAD) and 20.00 (Weizmann), respectively. **Table 6** shows the result for *one-handed wave* for the two testing (row) and training (column) setups alongside with exemplary pictures of the classes from both datasets. When training was performed on the Weizmann dataset, the recall performance for examples from the uulmMAD dataset is still considerable (loss of 24.07). Training on the uulmMAD and testing on the Weizmann dataset results in an increased performance loss, but still achieves a recall of 53.03.

In case of the combined class *raising two hands*, the performance loss is below 30 for both training and testing configurations. **Table 7** shows the achieved performance in detail for each of the four classes in isolation and their combination. Note that when trained on the uulmMAD dataset, *jumping jack* is recognized almost without any loss of performance. Vice versa, SP2 is often confused with *wave2* when training was performed on the Weizmann dataset. This may be explained by the large visual similarities between the classes.

The proposed approach shows promising generalization capabilities, which might partially be explained by the class-independent, *feature-driven* selection of the key pose frames.

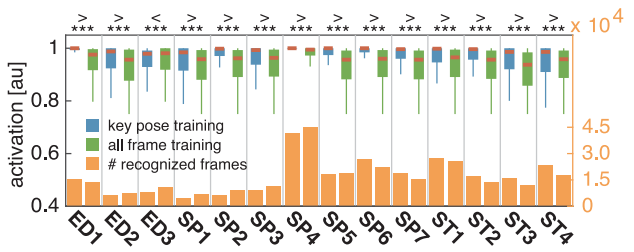


FIGURE 9 | Comparison of activation levels. The activation levels of the cell populations which resulted in correct classifications are displayed per class for key pose based (blue) and all frame (green) training alongside with the total number of correct classifications under both conditions (yellow). In case of key pose based training, activation levels are significantly increased, reflecting a higher confidence of the classification decision. Increased confidences are useful in situations where thresholding is applied on the activation level, e.g., to reduce the number of false negatives.

6. CONCLUSION AND DISCUSSION

The presented work consists of two main contributions. First, a *feature-driven* key pose selection mechanism is proposed, which builds upon evidences about human action perception. The selection mechanism does not utilize any information about the inter- or intra-class distribution of the key poses (or key pose descriptors) to optimize the classification accuracy. It is demonstrated, that the classification accuracy is on par with state-of-the-art key pose based action recognition approaches, while only motion and form related feature characteristics are used to select a key pose frame. Second, we propose a biologically inspired architecture combining form and motion information to learn hierarchical representations of key pose frames. We expect such hierarchical feature representations to make the recognition more robust against clutter and partial occlusions, in comparison to holistic shape representations of the full body configurations used in previous approaches. Form and motion pattern representations are established employing a neuromorphic deep convolutional neural network. The trained network is mapped onto the *IBM Neurosynaptic System* platform, which enables a computationally and energy efficient execution.

6.1. Relation to Other Work

The presented results demonstrate, that classifying actions using a minimal amount of temporal information is in principle

TABLE 6 | Cross-dataset evaluation one-handed wave.

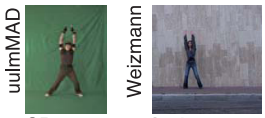
Test \ Train	uulmMAD	Weizmann	Loss
uulmMAD	**100**	75.93	24.07
Weizmann	53.03	**100**	46.97

ED1 **wave1**

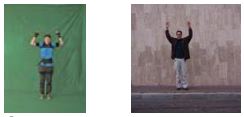
Bold values indicate the maximum recall values per column. Written informed consent for the publication of exemplary images was obtained from the shown subjects (uulmMAD).

TABLE 7 | Cross-dataset evaluation raising two hands.

Test \ Train		uulmMAD			Weizmann			Loss
		SP2	**SP6**	**Comb**	***jack***	***wave2***	**Comb**	**Comb**
uulmMAD	SP2	**96.97**	0.00	97.92	24.65	51.04	70.30	27.62
	SP6	1.14	**97.73**		0.00	64.90		
Weizmann	*jack*	95.96	0.00	79.80	**100**	0.00	100	20.20
	wave2	31.82	31.82		0.00	**100**		



SP2 **jack**



SP6 **wave2**

Bold values indicate the maximum recall values per column. Written informed consent for the publication of exemplary images was obtained from the shown subjects (uulmMAD).

possible. This is in line with results from other action recognition approaches. For example, Schindler and van Gool (2008) reported that actions can be successfully recognized using snippets of three or even less frames. In their work, the length of the temporal window used for the classification of an action sequence was systematically varied. The most important result was that a reliable action recognition can be achieved by only using individual snippets, i.e. up to three consecutive frames in temporal order. The question whether there are special "key snippets" of frames, which are particularly useful for the recognition of an action and how they might be defined, however, remains open.

Inspired by evidences from perceptual studies (Thurman and Grossman, 2008; Thirkettle et al., 2009), key poses are potential candidates for representing such special events in articulated motion sequences. Unlike the majority of other approaches reported in the literature (e.g., Baysal et al., 2010; Liu et al., 2013), the proposed key pose selection mechanism identifies key pose frames without optimizing the *inter-class* distinctiveness or classification performance of the selected key poses. The *feature-driven* selection criterion proposed in this work combines form and motion information and allows the identification of key poses without any knowledge about other classes. It extends a previous proposal utilizing local temporal extrema in the motion energy as a function of time (Layher et al., 2014) by additionally taking a measure of extendedness of the silhouette shape into account. Given that these features are entirely data-driven, this has two major implications. On the one hand, the selected poses are independent of any other class and thus are more likely to generalize over different sets of actions. This property is appreciated and valuable in many applications since it does not require any prior knowledge about the distribution of classes/poses in other datasets. On the other hand, there is no guarantee, however, that a learned key pose representation is not part of more than one action and thus results in ambiguous representations. This may lead to drawbacks and deteriorations of the model performance in terms of classification rates for rather ambiguous sequences with similar pose articulations. We argue that, although the proposed key pose selection criterion might not result in the best classification performance on all action recognition datasets in isolation, it selects key pose frames which capture the nature of an action in general (independent of a specific dataset). In addition, the reported results demonstrate, that there is no substantial loss in performance when comparing the proposed *feature-driven* key pose selection mechanism to performance optimizing key pose approaches in literature. In contrast to other action recognition approaches building upon convolutional neural networks, the proposed model does not aim at establishing representations which capture the temporal relationship between successive frames. This can be accomplished by e.g., directly feeding spatiotemporal input to the network and applying 3D convolutions (e.g., Baccouche et al., 2011; Ji et al., 2013) or by applying a *multiple spatio-temporal scales neural network* (MSTNN; Jung et al., 2015). Instead, in this work, the employed DCNN exclusively aims at identifying class specific key pose frames as events in an image (and optical flow) stream.

The investigation reported in this work adds an important piece to the debate of how representations for action sequence analysis might be organized. Some previous approaches have utilized motion and form information for the classification of action categories. For example, Giese and Poggio (2003) proposed that biological motion sequences representing articulated movements of persons is subdivided into two parallel streams in primate visual cortex. In particular, the authors argue that motion patterns are represented in a hierarchy and these are paralleled by regular temporal sampling of static frames from the same input sequence. This model architecture has been extended

in Layher et al. (2014) suggesting that instead of representing sequences of static frames only key poses need to be selected. As a candidate criterion, the motion energy is calculated over time and local energy minima depict reversal points of bodily articulation. Such reversals, in turn, most likely coincide with extremal articulations and thus can be utilized to select a key pose in such articulation sequences. While these models focus on cortical architecture of visual dorsal and ventral streams, other computer vision approaches also consider combinations of motion and form information for action recognition. While the proposal of Jhuang et al. (2007) builds on a hierarchy of cascaded form and motion representations, the approach of Schindler and van Gool (2008) also utilized two parallel streams of motion and form processing. Both streams generate feature vectors of equal length which are subsequently concatenated including a weighting of the relative strength of their contribution. An evaluation of the relative weights showed that a fusion with 70 % motion against a 30 % form feature concatenation yielded the best performance on the Weizmann dataset. On the contrary, Schindler et al. (2008) demonstrated that emotion categories can be classified using static images only which are processed by a multi-scale bank of filters with subsequent pooling operation and dimension reduction. Our findings add new insights to the investigation of utilizing form/shape and motion information in biological/articulated motion analysis for action recognition. Our findings highlight that key poses defined by events of temporal extrema in motion energy and dynamic object silhouette features reliably reflect a high information content regarding the whole action sequence. In other words, key poses can be detected by an entirely *feature-driven* approach (without utilizing any a priori model of actions in the sequence) and that the associated temporal events contain a high proportion of the information about the main components of the action sequence.

We successfully trained a DCNN of 15 convolutional layers on the key pose frames used as input, which were assigned to different action classes. The network was trained using the *energy-efficient deep neuromorphic networks* (*Eedn*) framework (Esser et al., 2016) and executed on a TrueNorth NS1e board (Merolla et al., 2014). The results show that action recognition can be performed on mobile robotic platforms under real-time constraints while consuming a minimal amount of energy. The reduced energy consumption and the high performance in classification rate (compare **Table 4**) makes such a model architecture a valuable candidate for applications in mobile or remote control scenarios in which autonomy in energy supply and external control are constraints of core importance. The automatic selection of key pose information for the classification mechanism is a key step to make use of the demonstrated parameters.

Although some classes contained examples with highly similar visual appearances, the network shows an impressive single frame recognition performance when tested on key frames. Even when tested on the full set of frames, recognition performance is still significantly above chance level. Using a simple temporal integration scheme, we show that the results are on par with competing key pose based action recognition approaches (**Table 4**). Cross-dataset evaluation of classes with the same/a similar visual appearance in both datasets shows how the learned representations generalize over the different datasets (training was performed on the full set of classes).

6.2. Shortcomings and Possible Further Improvements

Currently, the optical flow estimation and the key pose selection are performed prior to the training and the classification of input sequences. To realize a complete neuromorphic implementation of the presented approach, optical flow can be estimated as well on neuromorphic hardware following the principles described in Brosch and Neumann (2016). A neuromorphic implementation of localizing the local extrema in the motion energy and the extendedness of a person's silhouette could be realized on top of the flow estimation process. In addition, dynamic vision sensors (e.g., iniLabs DVS128) are an option to directly feed a network similar to the proposed one with spike-based sensory streams. First attempts to realize an action recognition system using such sparse asynchronous data streams have already shown promising results (Tschechne et al., 2014).

The presented approach does not make use of any temporal relationship between the identified events (key poses) in an action sequence. Thus, the reversed, or scrambled presentation of images (and optical flow) of a sequence would result in an assignment to an action class, although, the visual appearance of the sequence is totally different. A modeling or learning of the temporal relationships between the key pose frames, e.g., their temporal order, would help in reducing ambiguities and thus increase sequence-wide or windowed classification rates. In case of the proposed approach, this could be achieved by employing, e.g., *long short-term memory* cells (LSTM; Hochreiter and Schmidhuber, 1997), which are candidates to realize the learning of temporal relationships without loosing the invariance against changes in speed. The simple majority voting based integration scheme was chosen, because of hardware limitations and to focus on an analysis of the importance of key poses in the context of action representation and recognition.

We also did not apply a weighted majority voting scheme using the confidences of the frame-wise predictions or apply thresholding on the predictions. Both strategies might further increase the classification performance but again would weaken the focus on the analysis of key pose base representations of action sequences.

The proposed architecture of a deep convolutional neural network (DCNN) as depicted in **Figure 3** builds increasingly more complex feature representations through learning from initial simple features. It would be interesting to investigate the feature selectivities of the feature representations that have been established by the learning. Such a study would potentially shed light about the structure of the feature compositions (and their hierarchical organization) which lead to the selectivity of the key poses in relation to the action sequences to be classified. Some approaches analyzing the low-, intermediate-, and higher-level feature representations have recently been proposed in the literature (Zeiler and Fergus, 2014; Güçlü and van Gerven, 2015; Mahendran and Vedaldi, 2016). Such approaches have so far investigated CNNs for static inputs only. For that reason, some principles might also be useful for the analysis of key pose representations. In addition, the consideration of short-term

spatio-temporal feature representations will help to extend the scope of the overall study of visualizing internal representations after learning. We expect necessary major efforts to carefully develop an extended set of tools which is beyond the scope of the modeling investigation presented here.

Overall, the presented results show, that the learned key pose representations allow the classification of actions using a minimal amount of temporal information. By implementing the proposed DCNN on the TrueNorth chip, we show that real-time action recognition relying on the proposed principles is possible while consuming a minimal amount of energy, as reported for the runtime environments of the *IBM Neurosynaptic System* (Esser et al., 2016).

AUTHOR CONTRIBUTIONS

Conceived and designed the approach: GL, TB, HN; Implemented the architecture: GL, TB; Performed the simulations: GL; Analyzed the data: GL; Wrote the paper: GL, HN.

FUNDING

This research has been supported by the Transregional Collaborative Research Centre SFB/TRR 62 "A Companion Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG). HN has been supported by the collaborative project "SenseEmotion" funded by the German Federal Ministry of Education and Research (BMBF).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support via a field test agreement between Ulm University and IBM Research Almaden, and particularly the TrueNorth team at IBM for their support. In addition, the support of NVIDIA Corporation with the donation of a Tesla K40 GPU used for this research is also gratefully acknowledged. The authors like to express their gratefulness for the reviewers' efforts and suggestions that helped to improve the manuscript.

REFERENCES

Aggarwal, J., and Ryoo, M. (2011). Human activity analysis: a review. *ACM Comput. Surv.* 43, 16:1–16:43. doi: 10.1145/1922649.1922653

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., et al. (2015). Truenorth: design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Trans. Comput. Aided Design Integr. Circ. Syst.* 34, 1537–1557. doi: 10.1109/TCAD.2015.2474396

Amir, A., Datta, P., Risk, W. P., Cassidy, A. S., Kusnitz, J. A., Esser, S. K., et al. (2013). "Cognitive computing programming paradigm: a corelet language for composing networks of neurosynaptic cores," in *International Joint Conference on Neural Networks (IJCNN), 2013* (IEEE), 1–10. Available online at: http://ieeexplore.ieee.org/abstract/document/6707078/?reload=true

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2011). "Sequential Deep Learning for Human Action Recognition," in *2nd International Workshop on Human Behavior Understanding (HBU),* eds A. A. Salahand and B. Lepri (Berlin; Heidelberg: Springer), 29–39.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *Int. J. Comput. Vis.* 12, 43–77. doi: 10.1007/BF01420984

Baysal, S., Kurt, M. C., and Duygulu, P. (2010). "Recognizing human actions using key poses," in *20th International Conference on Pattern Recognition (ICPR), 2010* (IstanbulL), 1727–1730. Available online at: http://ieeexplore.ieee.org/abstract/document/5597477/

Beaudry, C., Péteri, R., and Mascarilla, L. (2016). An efficient and sparse approach for large scale human action recognition in videos. *Mach. Vis. Appl.* 27, 529–543. doi: 10.1007/s00138-016-0760-z

Bobick, A. F., and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Trans. Patt. Anal. Mach. Intel.* 23, 257–267. doi: 10.1109/34.910878

Boussaoud, D., Ungerleider, L. G., and Desimone, R. (1990). Pathways for motion analysis: cortical connections of the medial superior temporal and fundus of the superior temporal visual areas in the macaque. *J. Comp. Neurol.* 296, 462–495. doi: 10.1002/cne.902960311

Bregonzio, M., Gong, S., and Xiang, T. (2009). "Recognising action as clouds of space-time interest points," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009* (Miami, FL), 1948–1955. Available online at: http://ieeexplore.ieee.org/abstract/document/5206779/

Brin, S., and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw.* 30, 107–117. doi: 10.1016/s0169-7552(98)00110-x

Brosch, T., and Neumann, H. (2016). "Event-based optical flow on neuromorphic hardware," in *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), 2015* (New York, NY: ICST), 551–558. doi: 10.4108/eai.3-12-2015.2262447

Cao, X., Ning, B., Yan, P., and Li, X. (2012). Selecting key poses on manifold for pairwise action recognition. *IEEE Trans. Indust. Inform.* 8, 168–177. doi: 10.1109/TII.2011.2172452

Casile, A., and Giese, M. A. (2005). Critical features for the recognition of biological motion. *J. Vis.* 5:6. doi: 10.1167/5.4.6

Chaaraoui, A. A., Climent-Pérez, P., and Flórez-Revuelta, F. (2013). Silhouette-based human action recognition using sequences of key poses. *Patt. Recogn. Lett.* 34, 1799–1807. doi: 10.1016/j.patrec.2013.01.021

Cheema, S., Eweiwi, A., Thurau, C., and Bauckhage, C. (2011). "Action recognition by learning discriminative key poses," in *IEEE Workshops of the International Conference on Computer Vision (ICCV Workshops), 2011* (Barcelona), 1302–1309. Available online at: http://ieeexplore.ieee.org/abstract/document/6130402/

Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005*, Vol. 1 (San Diego, CA: IEEE), 886–893. Available online at: http://ieeexplore.ieee.org/abstract/document/1467360/

Dedeoğlu, Y., Töreyin, B. U., Güdükbay, U., and Çetin, A. E. (2006). "Silhouette-based method for object classification and human action recognition in video," in *European Conference on Computer Vision (ECCV), Workshop on HCI Computer Vision in Human-Computer Interaction, 2006*, eds T. S. Huang, N. Sebe, M. S. Lew, V. Pavlović, M. Kölsch, A. Galata, and B. Kisačanin (Berlin; Heidelberg: Springer), 64–77.

Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). "Behavior recognition via sparse spatio-temporal features," in *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS), 2005* (Beijing: IEEE), 65–72. Available online at: http://ieeexplore.ieee.org/abstract/document/1570899/

Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. Natl. Acad. Sci. U.S.A.* 113, 11441–11446. doi: 10.1073/pnas.1604850113

Fathi, A., and Mori, G. (2008). "Action recognition by learning mid-level motion features," in *IEEE Conference on Computer Vision and Pattern Recognition*

(CVPR), 2008 (IEEE), 1–8. Available online at: http://ieeexplore.ieee.org/abstract/document/4587735/

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47. doi: 10.1093/cercor/1.1.1

Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Trans. Patt. Anal. Mach. Intel.* 30, 36–51. doi: 10.1109/TPAMI.2007.1144

Freund, Y., and Schapire, R. E. (1995). "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory*, ed P. Vitányi (Berlin; Heidelberg: Springer), 23–37. doi: 10.1007/3-540-59119-2_166

Garcia, J. O., and Grossman, E. D. (2008). Necessary but not sufficient: motion perception is required for perceiving biological motion. *Vis. Res.* 48, 1144–1149. doi: 10.1016/j.visres.2008.01.027

Gavrila, D. M. (1999). The visual analysis of human movement: a survey. *Comput. Vis. Image Unders.* 73, 82–98. doi: 10.1006/cviu.1998.0716

Giese, M. A., and Poggio, T. (2003). Neural mechanisms for the recognition of biological movements. *Nat. Rev. Neurosci.* 4, 179–192. doi: 10.1038/nrn1057

Glodek, M., Layher, G., Heilemann, F., Gawrilowicz, F., Palm, G., Schwenker, F., et al. (2014). "uulmMAD–a human action recognition dataset for ground-truth evaluation and investigation of view invariances," in *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction: Third IAPR TC3 Workshop on Multimodal Pattern Recognition of Social Signals in Human-Computer Interaction (MPRSS), 2014* (Cham: Springer), 77–91. doi: 10.1007/978-3-319-14899-1_8

Gong, W., Bagdanov, A. D., Roca, F. X., and Gonzàlez, J. (2010). "Automatic key pose selection for 3D human action recognition," in *6th International Conference on Articulated Motion and Deformable Objects (AMDO), 2010*, eds F. J. Perales and R. B. Fisher (Berlin; Heidelberg: Springer), 290–299.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *Trans. Patt. Anal. Mach. Intel.* 29, 2247–2253. doi: 10.1109/TPAMI.2007.70711

Grauman, K., and Darrell, T. (2005). "The pyramid match kernel: discriminative classification with sets of image features," in *10th IEEE International Conference on Computer Vision (ICCV), 2005*, Vol. 2 (Beijing), 1458–1465. Available online at: http://ieeexplore.ieee.org/abstract/document/1544890/

Grossman, E., Donnelly, M., Price, R., Pickens, D., Morgan, V., Neighbor, G., et al. (2000). Brain areas involved in perception of biological motion. *J. Cogn. Neurosci.* 12, 711–720. doi: 10.1162/089892900562417

Grundmann, M., Meier, F., and Essa, I. (2008). "3D shape context and distance transform for action recognition," in *19th International Conference on Pattern Recognition (ICPR), 2008* (Tampa, FL), 1–4. Available online at: http://ieeexplore.ieee.org/abstract/document/4761435/

Güçlü, U., and van Gerven, M. A. J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Hoai, M., Lan, Z.-z., and De la Torre, F. (2011). "Joint segmentation and classification of human actions in video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011* (Colorado Springs, CO), 3265–3272. Available online at: http://ieeexplore.ieee.org/abstract/document/5995470/

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Ikizler, N., and Duygulu, P. (2009). Histogram of oriented rectangles: a new pose descriptor for human action recognition. *Image Vis. Comput.* 27, 1515–1526. doi: 10.1016/j.imavis.2009.02.002

Ioffe, S., and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Jellema, T., and Perrett, D. I. (2003). Cells in monkey STS responsive to articulated body motions and consequent static posture: a case of implied motion? *Neuropsychologia* 41, 1728–1737. doi: 10.1016/S0028-3932(03)00175-1

Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007). "A biologically inspired system for action recognition," in *IEEE International Conference on Computer Vision (ICCV), 2007* (Rio de Janeiro: IEEE), 1–8. Available online at: http://ieeexplore.ieee.org/abstract/document/4408988/

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Trans. Patt. Anal. Mach. Intel.* 35, 221–231. doi: 10.1109/TPAMI.2012.59

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Percept. Psychophys.* 14, 201–211. doi: 10.3758/BF03212378

Johansson, G. (1976). Spatio-temporal differentiation and integration in visual motion perception. *Psychol. Res.* 38, 379–393. doi: 10.1007/BF00309043

Jung, M., Hwang, J., and Tani, J. (2015). Self-organization of spatio-temporal hierarchy via learning of dynamic visual image patterns on action sequences. *PLoS ONE* 10:e0131214. doi: 10.1371/journal.pone.0131214

Kläser, A., Marszałek, M., and Schmid, C. (2008). "A spatio-temporal descriptor based on 3D-gradients," in *19th British Machine Vision Conference (BMVC), 2008*, eds M. Everingham and C. Needham (Leeds: British Machine Vision Association), 99.1-99.10. Available online at: http://www.bmva.org/bmvc/2008/papers/275.html

Kourtzi, Z., and Kanwisher, N. (2000). Activation in human MT/MST by static images with implied motion. *J. Cogn. Neurosci.* 12, 48–55. doi: 10.1162/08989290051137594

Lange, J., and Lappe, M. (2006). A model of biological motion perception from configural form cues. *J. Neurosci.* 26, 2894–2906. doi: 10.1523/JNEUROSCI.4915-05.2006

Layher, G., Giese, M. A., and Neumann, H. (2014). Learning representations of animated motion sequences - a neural model. *Top. Cogn. Sci.* 6, 170–182. doi: 10.1111/tops.12075

LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). "Convolutional networks and applications in vision," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2010* (Paris), 253–256. Available online at: http://ieeexplore.ieee.org/abstract/document/5537907/

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *CoRR*, abs/1312.4400.

Liu, L., Shao, L., Zhen, X., and Li, X. (2013). Learning discriminative key poses for action recognition. *IEEE Trans. Cybernet.* 43, 1860–1870. doi: 10.1109/TSMCB.2012.2231959

Lucas, B. D., and Kanade, T. (1981). "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), 1981* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 674–679.

Lv, F., and Nevatia, R. (2007). "Single view human action recognition using key pose matching and viterbi path searching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007* (IEEE), 1–8. Available online at: http://ieeexplore.ieee.org/abstract/document/4270156/

Mahendran, A. and Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. *Int. J. Comput. Vis.* 120, 233–255. doi: 10.1007/s11263-016-0911-8

Mather, G., and Murdoch, L. (1994). Gender discrimination in biological motion displays based on dynamic cues. *Proc. R. Soc. Lond. B. Biol. Sci.* 258, 273–279. doi: 10.1098/rspb.1994.0173

McCann, S., and Lowe, D. G. (2012). "Local naive bayes nearest neighbor for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, 3650–3656. Available online at: http://ieeexplore.ieee.org/abstract/document/6248111/

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Niebles, J. C., and Fei-Fei, L. (2007). "A hierarchical model of shape and appearance for human action classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007* (Minneapolis, MN: IEEE), 1–8. Available online at: http://ieeexplore.ieee.org/abstract/document/4270157/

Ogale, A. S., Karapurkar, A., and Aloimonos, Y. (2007). "View-invariant modeling and recognition of human actions using grammars," in *Dynamical Vision*, eds R. Vidal, A. Heyden, and Y. Ma (Berlin; Heidelberg: Springer), 115–126.

Oram, M., and Perrett, D. (1994). Responses of anterior superior temporal polysensory (STPa) neurons to "Biological Motion" stimuli. *J. Cogn. Neurosci.* 6, 99–116. doi: 10.1162/jocn.1994.6.2.99

Oram, M. W., and Perrett, D. I. (1996). Integration of form and motion in the anterior superior temporal polysensory area (STPa) of the macaque monkey. *J. Neurophysiol.* 76, 109–129.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image Vis. Comput.* 28, 976–990. doi: 10.1016/j.imavis.2009.11.014

Powers, D. M. W. (2013). "A computationally and cognitively plausible model of supervised and unsupervised learning," in *6th International Conference on Advances in Brain Inspired Cognitive Systems (BICS), 2013*, eds D. Liu, C. Alippi, D. Zhao, and A. Hussain (Berlin; Heidelberg: Springer), 145–156. doi: 10.1007/978-3-642-38786-9_17

Puce, A., and Perrett, D. (2003). Electrophysiology and brain imaging of biological motion. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 358, 435–445. doi: 10.1098/rstb.2002.1221

Riesenhuber, M., and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat. Neurosci.* 2, 1019–1025. doi: 10.1038/14819

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating error. *Nature* 323, 533–536. doi: 10.1038/323533a0

Sawada, J., Akopyan, F., Cassidy, A. S., Taba, B., Debole, M. V., Datta, P., et al. (2016). "Truenorth ecosystem for brain-inspired computing: scalable systems, software, and applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2016* (Piscataway, NJ: IEEE Press), 12:1–12:12.

Schindler, K., Gool, L. V., and de Gelder, B. (2008). Recognizing emotions expressed by body pose: a biologically inspired neural model. *Neural Netw.* 21, 1238–1246. doi: 10.1016/j.neunet.2008.05.003

Schindler, K., and van Gool, L. (2008). "Action snippets: how many frames does human action recognition require?" in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008* (Anchorage), 1–8. Available online at: http://ieeexplore.ieee.org/abstract/document/4587730/

Sokolova, M., and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Inform. Process. Manage.* 45, 427–437. doi: 10.1016/j.ipm.2009.03.002

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2014). Striving for simplicity: the all convolutional net. *CoRR*, abs/1412.6806.

Sun, Q., and Liu, H. (2012). "Action disambiguation analysis using normalized google-like distance correlogram," in *11th Asian Conference on Computer Vision (ACCV), 2012*, eds K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu (Berlin; Heidelberg: Springer), 425–437. doi: 10.1007/978-3-642-37431-9_33

Thirkettle, M., Benton, C. P., and Scott-Samuel, N. E. (2009). Contributions of form, motion and task to biological motion perception. *J. Vis.* 9:28. doi: 10.1167/9.3.28

Thurau, C., and Hlavác, V. (2008). "Pose primitive based human action recognition in videos or still images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008* (IEEE), 1–8. Available online at: http://ieeexplore.ieee.org/abstract/document/4587721/

Thurman, S. M., and Grossman, E. D. (2008). Temporal "Bubbles" reveal key features for point-light biological motion perception. *J. Vis.* 8:28. doi: 10.1167/8.3.28

Tschechne, S., Sailer, R., and Neumann, H. (2014). "Bio-inspired optic flow from event-based neuromorphic sensor input," in *6th International Workshop on Artificial Neural Networks in Pattern Recognition (IAPR), 2014*, eds N. El Gayar, F. Schwenker, and C. Schwenker (Cham: Springer) 171–182. doi: 10.1007/978-3-319-11656-3_16

Vedaldi, A., and Lenc, K. (2015). "Matconvnet: convolutional neural networks for matlab," in *23rd ACM International Conference on Multimedia (ACMMM), 2015* (New York, NY: ACM), 689–692. doi: 10.1145/2733373.2807412

Weinland, D., and Boyer, E. (2008). "Action recognition using exemplar-based embedding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008* (Anchorage: IEEE), 1–7. Available online at: http://ieeexplore.ieee.org/abstract/document/4587731/

Weinland, D., Ronfard, R., and Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Comput. Vis. Image Underst.* 115, 224–241. doi: 10.1016/j.cviu.2010.10.002

Yamato, J., Ohya, J., and Ishii, K. (1992). "Recognizing human action in time-sequential images using hidden markov model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1992* (IEEE), 379–385. Available online at: http://ieeexplore.ieee.org/abstract/document/223161/

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *13th European Conference on Computer Vision (ECCV),* eds D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Cham: Springer International Publishing), 818–833. Available online at: link.springer.com/chapter/10.1007/978-3-319-10590-1_53

Zhao, Z., and Elgammal, A. (2008). "Information theoretic key frame selection for action recognition," in *Proceedings of the British Machine Vision Conference (BMVC), 2008*, eds M. Everingham and C. Everingham (BMVA Press), 109.1–109.10. Available online at: http://www.bmva.org/bmvc/2008/papers/143.html

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Layher, Brosch and Neumann. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 21 March 2017
doi: 10.3389/fnbot.2017.00014

Fast Dynamical Coupling Enhances Frequency Adaptation of Oscillators for Robotic Locomotion Control

Timo Nachstedt[1,2]*, *Christian Tetzlaff*[2,3] *and Poramate Manoonpong*[4]

[1] Third Institute of Physics, Universität Göttingen, Göttingen, Germany, [2] Bernstein Center for Computational Neuroscience, Göttingen, Germany, [3] Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany, [4] Embodied AI and Neurorobotics Lab, Centre for BioRobotics, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark

Rhythmic neural signals serve as basis of many brain processes, in particular of locomotion control and generation of rhythmic movements. It has been found that specific neural circuits, named central pattern generators (CPGs), are able to autonomously produce such rhythmic activities. In order to tune, shape and coordinate the produced rhythmic activity, CPGs require sensory feedback, i.e., external signals. Nonlinear oscillators are a standard model of CPGs and are used in various robotic applications. A special class of nonlinear oscillators are adaptive frequency oscillators (AFOs). AFOs are able to adapt their frequency toward the frequency of an external periodic signal and to keep this learned frequency once the external signal vanishes. AFOs have been successfully used, for instance, for resonant tuning of robotic locomotion control. However, the choice of parameters for a standard AFO is characterized by a trade-off between the speed of the adaptation and its precision and, additionally, is strongly dependent on the range of frequencies the AFO is confronted with. As a result, AFOs are typically tuned such that they require a comparably long time for their adaptation. To overcome the problem, here, we improve the standard AFO by introducing a novel adaptation mechanism based on dynamical coupling strengths. The dynamical adaptation mechanism enhances both the speed and precision of the frequency adaptation. In contrast to standard AFOs, in this system, the interplay of dynamics on short and long time scales enables fast as well as precise adaptation of the oscillator for a wide range of frequencies. Amongst others, a very natural implementation of this mechanism is in terms of neural networks. The proposed system enables robotic applications which require fast retuning of locomotion control in order to react to environmental changes or conditions.

Keywords: adaptive frequency oscillator, central pattern generator, neural networks, resonance tuning, locomotion control

OPEN ACCESS

Edited by:
Shuai Li,
Hong Kong Polytechnic University,
Hong Kong

Reviewed by:
Paolo Arena,
University of Catania, Italy
Patrick Henaff,
University of Lorraine, France

***Correspondence:**
Timo Nachstedt
timo.nachstedt@
phys.uni-goettingen.de

Received: *18 October 2016*
Accepted: *24 February 2017*
Published: *21 March 2017*

Citation:
Nachstedt T, Tetzlaff C and Manoonpong P (2017) Fast Dynamical Coupling Enhances Frequency Adaptation of Oscillators for Robotic Locomotion Control. Front. Neurorobot. 11:14. doi: 10.3389/fnbot.2017.00014

1. INTRODUCTION

Rhythmic processes are of central importance for many aspects of biological life (Winfree, 1967; Barkai and Leibler, 2000; Goldbeter et al., 2012). Examples include the cardiac rhythm, various circadian rhythms and, in particular, all forms of biological locomotion like walking, flying or swimming. The latter are controlled by specific neural circuits, so called central pattern generators

(CPGs) (Hooper, 2001; Marder and Bucher, 2001). Theoretical models of CPGs range from detailed biophysical models (Hellgren and Grillner, 1992) to pure mathematical oscillators (Matsuoka, 1985). In general, CPGs can be described as nonlinear oscillators which have been used in numerous applications for different variants of robotic control problems (Nakamura et al., 2007; Ijspeert, 2008; Pinto et al., 2012; Nassour et al., 2014; Santos et al., 2017). For instance, compared to purely reflexive control schemes (Foth and Bässler, 1985; Cruse et al., 1995), oscillator-controlled robots enable more stable and robust locomotion (Kimura et al., 2001; Righetti and Ijspeert, 2008).

CPGs do not require any external input or feedback to produce basic rhythmic activity. However, they still require feedback signals to adapt and tune their produced activity, for instance its frequency. For the theoretical concept of nonlinear oscillators, a universal mechanism to adapt the intrinsic frequency of an oscillator according to the frequency of an external periodic signal, which is coupled to the oscillator, was formulated by Righetti et al. (2006). This frequency adaptation schema is applicable to many different types of oscillators. In contrast to the well-known phenomenon of entrainment, which is a purely reactive mechanism with only transient effect on the oscillatory system (Buchli et al., 2006), the frequency adaptation schema modifies the intrinsic frequency of the system permanently. Oscillators with this schema are commonly called adaptive frequency oscillators (AFOs). Several applications of AFOs have been proposed including adaptive control of compliant robots (Righetti et al., 2009), pendulum swing-up problems (Spong, 1995; Furuta, 2003), understanding, simulation and support of human locomotion (Ronsse et al., 2011a; Tropea et al., 2015; Santos et al., 2017), mimicking of fish swimming (Wang et al., 2013), frequency analysis of an input signal (Buchli et al., 2008), and construction of limit cycles of arbitrary shape (Righetti et al., 2009). However, all of these applications suffer from significantly long adaptation times.

For a given oscillatory system, the dynamics of a standard AFO is determined by only two parameters: the strength of the coupling of the external signal to the oscillator and the learning rate of the parameter determining the intrinsic frequency of the system. Here, we show that, when choosing these two parameters, one has to make a compromise between speed and precision of the resulting adaption dynamics. Furthermore, we demonstrate that the optimal parameters for a certain balance of speed and precision strongly depend on the initial intrinsic frequency of the oscillator and on the target frequency, i.e., the frequency of the external signal. As a result, situation-specific fine-tuning of the parameters is necessary.

In contrast, we propose an extension of the standard frequency adaptation mechanism which provides both fast as well as precise adaptation for a wide range of initial intrinsic and target frequencies without the need for parameter fine tuning. In the following, we call this mechanism "Adaptation through Fast Dynamical Coupling" (AFDC). It is based on dynamically adapting the coupling strength of the external signal. If the difference between the current intrinsic frequency and the target frequency is high, the coupling strength is increased in order to accelerate the adaptation. If the difference between the current intrinsic frequency and the target frequency becomes small, the coupling strength is reduced to increase the precision of the adaptation. This process is autonomous and can be integrated into the dynamical equations of the system. Neither the current intrinsic nor the target frequency need to be explicitly available as the mechanism solely relies on signal correlations. We compare the adaptation processes obtained by regular AFOs with those obtained with the new AFDC mechanism by means of quantitative measures of speed and precision of the adaptation. We find that the AFDC mechanism clearly outperforms standard AFOs within the tested frequency interval covering two orders of magnitudes.

2. RESULTS

2.1. Standard Adaptive Frequency Oscillator

In very general terms, an oscillator is an autonomous dynamical system with at least one limit cycle attractor (Buchli et al., 2006). Naturally, every two-dimensional oscillatory system (x, y) can be expressed as a system of two equations $\dot{x}(t) = g_x(x(t), y(t), \theta)$ and $\dot{y}(t) = g_y(x(t), y(t), \theta)$ where the functions g_x and g_y define the dynamics of the system. We require that these two functions do not only depend on the state variables x and y but also explicitly on a variable θ which determines the intrinsic oscillation frequency f of the system. The function $f(\theta)$ may be of an arbitrary shape and in many cases is not explicitly known. We only assume it to be monotonic. The system can be transformed into an adaptive frequency oscillator (AFO) by coupling it to an external signal $F(t)$:

$$\begin{aligned} \dot{x}(t) &= g_x(x(t), y(t), \theta(t)) + \epsilon F(t) \\ \dot{y}(t) &= g_y(x(t), y(t), \theta(t)). \end{aligned} \tag{1}$$

Here, ϵ denotes the coupling strength. Furthermore, additional dynamics of the θ-variable are introduced (Righetti et al., 2006):

$$\dot{\theta}(t) = \pm\eta F(t) \frac{y(t)}{\sqrt{x(t)^2 + y(t)^2}}. \tag{2}$$

with a learning rate η. The sign on the right-hand side depends on the direction of oscillation of the actual oscillatory system in the phase space. Note that in the original publication (Righetti et al., 2006), always $\eta = \epsilon$ is chosen as it emerges naturally when deriving the adaptation rule from analyzing the effect of the periodic external signal F on the phase velocity of the oscillator (Righetti et al., 2006). Apart from this, however, there is no a priori reason why this choice should provide optimal adaptation results. It has been shown that, using this rule, a wide range of oscillators can adapt their intrinsic frequencies to the frequency of basically any external periodic signal $F(t)$. In this contribution, we consider the Hopf oscillator (**Figure 1A**), which possesses a harmonic limit cycle, and the Van der Pol oscillator (Van der Pol, 1920) (**Figure 1B**), which, depending on the choice of parameters, exhibits highly non-harmonic oscillations.

For analyzing a given adaptation process, we start with an oscillator with an initial frequency variable θ_0 corresponding to

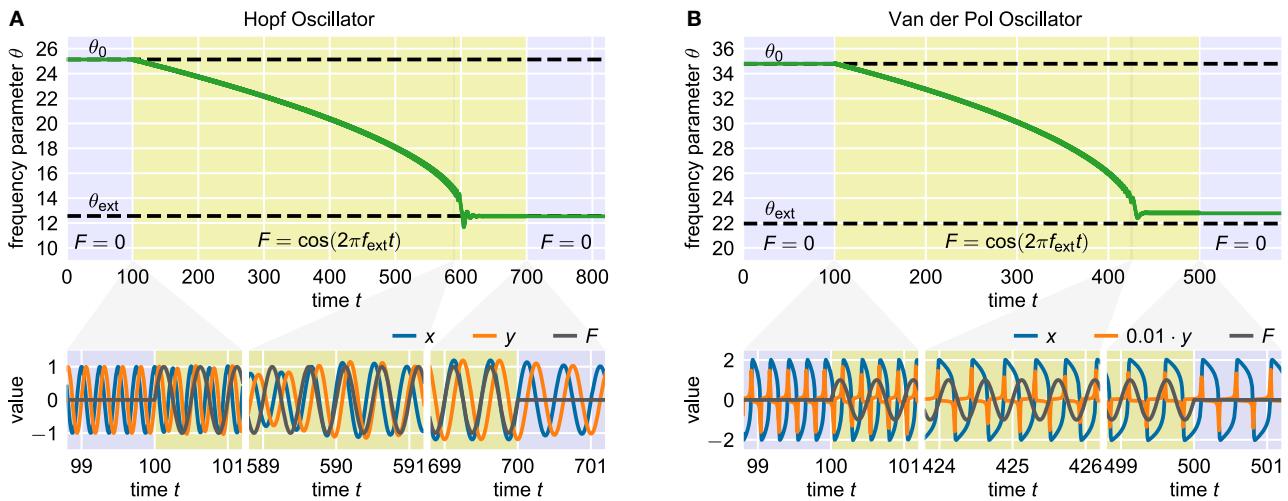


FIGURE 1 | Adaptation of two standard adaptive frequency oscillators. The upper panels show the time course of the frequency determining parameter θ. The time during which the external signal is applied to the system is indicated by the yellow shaded area. The dashed horizontal lines indicate the values θ_0 and θ_{ext} corresponding to the initial intrinsic frequency f_0 and the target frequency f_{ext} of the external signal, respectively. The panels below show the oscillating state variables x and y and the external signal F at different short time windows during the adaptation process. In both cases, the initial intrinsic frequency of the oscillator is $f_0 = 4.0$ and the external signal is a sine wave with unit amplitude and frequency $f_{\text{ext}} = 2.0$. **(A)** Adaptive frequency Hopf oscillator with $\mu = 1.0$ and $\epsilon = \eta = 1.0$ (see Methods). The initial value of the parameter θ is given by $\theta_0 = 2\pi f_0 \approx 25.1$. Accordingly, the value corresponding to the frequency of the external signal is $\theta_{\text{ext}} = 2\pi f_{\text{ext}} \approx 12.6$. The external signal is applied for $100 \leq t < 700$. **(B)** Adaptive frequency Van der Pol oscillator with $\mu = 100.0$ and $\epsilon = \eta = 0.7$ (see Methods). The values of the parameter θ corresponding to f_0 and f_{ext} are $\theta_0 \approx 34.8$ and $\theta_{\text{ext}} \approx 22.0$ (see Methods). The external signal is applied for $100 \leq t < 500$.

an initial intrinsic frequency $f_0 = f(\theta_0)$. Here, the function $f(\theta)$ is not explicitly known but can be numerically approximated. We denote the target frequency, i.e., the frequency of the external signal, by f_{ext}. Furthermore, we define the target value θ_{ext} as the value of θ such that $f_{\text{ext}} = f(\theta_{\text{ext}})$ for the given oscillator. The frequency variable θ is not modified by the adaptation rule (Equation 2) as long as the external signal F is zero ($t < 100$ in **Figure 1A**). After the onset of the external signal, θ is slowly adapted toward the target value θ_{ext} ($100 < t < 700$ in **Figure 1A**). The adaptation rate increases as θ gets closer to θ_{ext}. The final adaptation phase is typically characterized by a small θ-overshoot before it converges toward a quasi-constant state with only small periodic fluctuations ($600 < t < 700$ in **Figure 1A**). Now, when removing the external signal, i.e., setting $F = 0$, the oscillator maintains oscillations at the adapted frequency ($t > 700$ in **Figure 1A**). Note that it is not guaranteed that the finally reached value of θ corresponds exactly to θ_{ext}. In contrast, in some cases significant deviations can be observed (**Figure 1B**). As it turns out, reducing this deviation is only possible when accepting longer adaptation times.

2.1.1. Speed vs. Precision Trade-Off

In many applications, for instance in robotic systems, it is usually desired to have systems that are able to adapt to new situations or circumstances quickly. In contrast, AFOs with the usual choice of parameters require many periods of oscillations to complete a given adaptation process. The convergence time of the adaptation process, i.e., the time between the onset of the external signal and the quasi-convergence of the frequency parameter θ of the oscillator, can be adjusted by manipulating the coupling strength ϵ in Equation (1) or the learning rate η in Equation (2) (**Figure 2**). However, increasing ϵ or η does not only increase the speed of the frequency adaptation but also increases the general influence of the external signal on the oscillatory system. As a result, the dynamics of the parameter θ, once it has converged to a quasi-stable state, is affected as well (**Figure 3**). On the one hand, high learning rates η lead to increased fluctuations of the parameter θ in the finally reached state. On the other hand, higher values of ϵ result in a higher offset of the finally reached mean value $\bar{\theta}$ from the value θ_{ext}. Therefore, shorter convergence times in the standard AFO systems go hand in hand with a loss of precision. Naturally, this trade-off complicates real-world applications of the mechanism.

2.1.2. Quantitative Adaptation Quality Measures

In order to quantitatively capture the trade-off between speed and precision, we introduce three measures characterizing the quality of a given adaptation process (**Figure 3**). As already discussed, in many applications fast adaptation is desired. This is captured by the convergence time Δ which measures the time interval between the onset of the external signal and the last deviation of the intrinsic frequency f of the system (determined by θ) of more than 5% (10% for the Van der Pol oscillator) from the finally reached mean value $\bar{f}$. The precision of the adaptation, in turn, is reflected by two measures. First, the intrinsic frequency to which the system converges should be as close as possible to the frequency of the external signal. This is measured by the frequency offset δ which is given by the offset of the finally reached mean value of the intrinsic frequency from the frequency of the external signal. Second, the fluctuations of the intrinsic

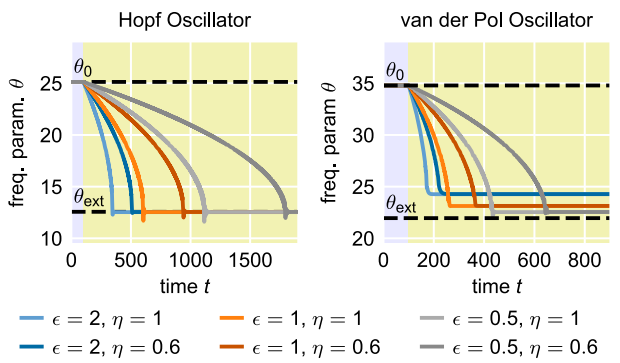


FIGURE 2 | Influence of the coupling strength ϵ and the learning rate η on the speed of adaptation of standard adaptive frequency oscillators. The yellow shaded area indicates the time during which the external signal is applied. In all cases, the initial intrinsic frequency of the oscillator is $f_0 = 4.0$ and the frequency of the external unit sine-wave signal is $f_{\text{ext}} = 2.0$. For the adaptive Hopf oscillator, we choose $\mu = 1.0$. For the adaptive van-der-Pol oscillator, we choose $\mu = 100.0$ (see Methods).

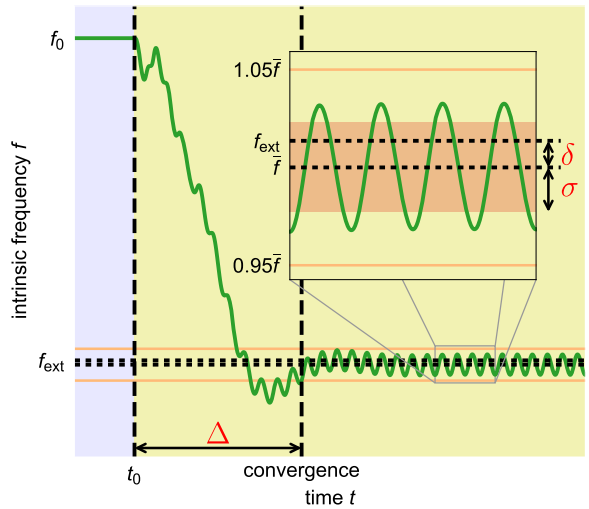


FIGURE 3 | Quantitative measures to capture the quality of an adaptation process. Shown is the time course of the intrinsic frequency of an adaptive frequency oscillator during the adaptation to an external periodic signal with high coupling constant ϵ and high learning rate η. The yellow shaded area indicates the time during which the external signal is applied. The inset shows a close up of the data in the indicated area. We introduce three measures to quantify the quality of a given frequency adaption process. The convergence time Δ is the time interval between the onset of the external signal at time t_0 and the last deviation of the intrinsic frequency of the oscillator of more than 5% (orange horizontal lines) from the finally reached average intrinsic frequency $\bar{f}$. The frequency offset δ measures the difference between the final average intrinsic frequency $\bar{f}$ and the target frequency of the external signal f_{ext}. In order to also capture the periodic fluctuations of the intrinsic frequency from the average value $\bar{f}$, we additionally introduce the final frequency fluctuation σ given by the standard deviation of the oscillations of the intrinsic frequency f in the finally reached state (area shaded in light red in the inset). The shown time course of the intrinsic frequency is taken from an adaptive frequency Hopf oscillator with $\mu = 1.0$, $\epsilon = 5.0$, $\eta = 5.0$, and $f_0 = 2.0$ adapting to an external unit sine-wave signal with frequency $f_{\text{ext}} = 1.0$.

frequency around its mean value should be low as otherwise the value of the intrinsic frequency when switching off the external signal depends on the exact point of time of this event. The magnitude of these fluctuations is measured by σ which equals the standard deviation of the intrinsic frequency f in the converged state.

To allow interpretation of these measures independently from the chosen internal and external frequencies, we introduce relative measures scaled by the frequency f_{ext} or the cycle duration f_{ext}^{-1} of the external signal, respectively: $\tilde{\Delta} = \Delta / f_{\text{ext}}^{-1}$, $\tilde{\delta} = \delta / f_{\text{ext}}$ and $\tilde{\sigma} = \sigma / f_{\text{ext}}$. In addition, we define a quality index Q combining these three relative measures into a single scalar value:

$$Q = \max\left(1 - \frac{\tilde{\Delta}}{\tilde{\Delta}_{\max}} - \frac{|\tilde{\delta}|}{\tilde{\delta}_{\max}} - \frac{\tilde{\sigma}}{\tilde{\sigma}_{\max}}, 0\right). \tag{3}$$

Here, $\tilde{\Delta}_{\max}$, $\tilde{\delta}_{\max}$, and $\tilde{\sigma}_{\max}$ are the maximum values of the respective measures which we allow for a reasonably good adaptation process. Accordingly, a Q value close to 1 corresponds to a very fast as well as very precise adaptation process. A value of $Q = 0$, in contrast, indicates that $\tilde{\Delta} > \tilde{\Delta}_{\max}$, $\tilde{\delta} > \tilde{\delta}_{\max}$, $\tilde{\sigma} > \tilde{\sigma}_{\max}$ or the weighted sum (Equation 3) of the individual measures is larger than 1. In the following, if not stated otherwise, we use $\tilde{\Delta}_{\max} = 100$, $\tilde{\delta}_{\max} = 0.05$ and $\tilde{\sigma}_{\max} = 0.05$.

2.1.3. Finding Optimal Parameters

For an easy application of an adaptive oscillator in a given setup, no fine tuning of the system parameters for the specific application context should be necessary. It is therefore necessary to find a system which is able to adapt its intrinsic frequency to a wide range of external frequencies without the need for any parameter adaptation other than the one of the frequency determining parameter θ. It turns out, however, that already for the comparable simple case of the harmonic Hopf oscillator, the range of frequencies for which a given set of parameters allows fast as well as precise adaptations is very limited (**Figure 4**). Higher values of ϵ and η increase the intervals of initial intrinsic frequencies f_0 and external frequencies f_{ext} for which fast adaptation is achieved (left column in **Figure 4**). In contrast, small frequency offsets $\tilde{\delta}$ are achieved only for small values of the coupling strength ϵ (second column in **Figure 4**) and small values of the learning rate η enable small fluctuations as measured by σ (third column in **Figure 4**). The compilation of these observations is reflected by only small intervals of initial intrinsic and external frequencies for which the quality index Q attains non-zero values (right column in **Figure 4**).

Trying to find parameters that allow fast and precise adaptation for a range of initial intrinsic and external target frequencies spanning two orders of magnitudes reveals that actually no ϵ-η-combination allows for an average adaptation quality index $\langle Q \rangle$ higher than approximately 0.12 (**Figure 5**). We conclude that a standard AFO with a fixed set of parameters is not capable to provide fast as well as precise adaptation over a wide range of frequencies.

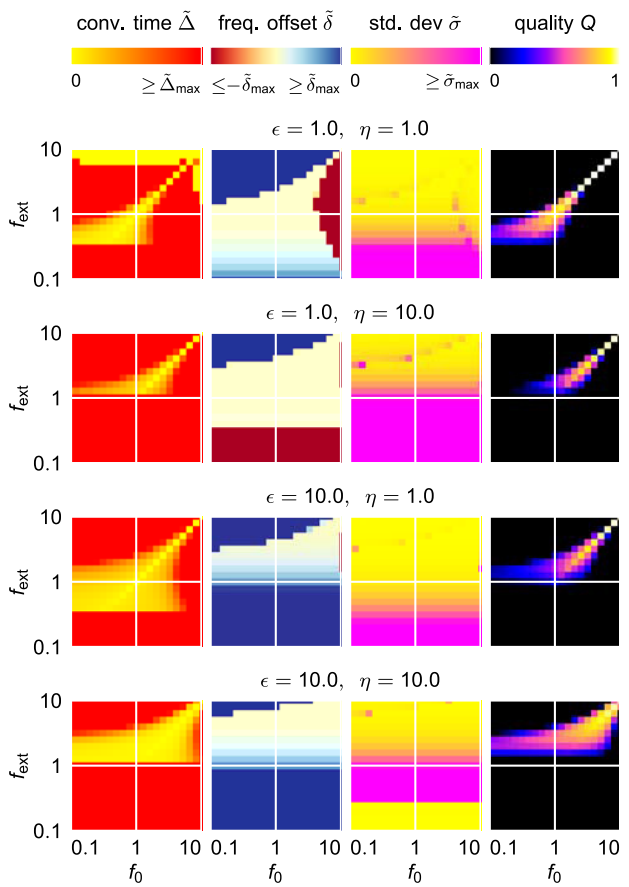


FIGURE 4 | Adaptation quality measures of the adaptive frequency Hopf oscillator in the (f_0, f_{ext}) frequency space for different values of the coupling strength ϵ and the learning rate η. For every given (ϵ, η)-parameter pair, from left to right, the relative convergence time $\tilde{\Delta}$, the relative final frequency offset $\tilde{\delta}$, the final relative frequency fluctuation $\tilde{\sigma}$, and the combined quality measure Q are shown in the plane spanned by the initial intrinsic frequency f_0 and the frequency of the external unit sine-wave signal f_{ext}. As the convergence time is defined as the time difference between the onset of the external signal and the last point of time of more than 5% deviation of the intrinsic frequency from the final average, for high values of $\tilde{\sigma}$, the convergence time cannot be reasonably determined, i.e., takes very high values. For the same reason, even on the diagonal $f_0 = f_{ext}$, high convergence times are measured for low values of f_{ext}.

2.2. Fast Dynamical Coupling Mechanism

As discussed, no fixed value pair for the coupling strength ϵ and the learning rate η suffices for fast and precise adaptation over a wider range of initial intrinsic and external target frequencies. In order to obtain a system without the requirement for application-specific fine-tuning, the down- or up-scaling of coupling strength and learning rate has to be accomplished in a self-organized manner. Here, we propose such a system. Instead of coupling the external signal $F(t)$ directly to the oscillator, we now use a filtered signal $P(t)$:

$$\begin{aligned} \dot{x}(t) &= f_x(x(t), y(t), \theta(t)) + P(t) \\ \dot{y}(t) &= f_y(x(t), y(t), \theta(t)). \end{aligned} \tag{4}$$

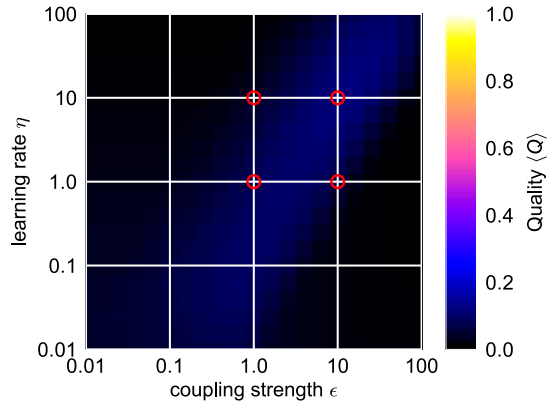


FIGURE 5 | Average combined quality measure $\langle Q \rangle$ for different parameter values of the frequency adaptive Hopf oscillator. For every parameter pair of coupling strength ϵ and learning rate η, the average adaption quality measure $\langle Q \rangle$ over the logarithmically sampled space of initial intrinsic frequencies f_0 and frequencies of the external signal f_{ext} is shown ($0.1 < f_0, f_{ext} < 10$). The red circles indicate the four cases shown in **Figure 4**. In each case, the external signal is a sine-wave with unit amplitude.

Accordingly, also the adaptation of θ is based on $P(t)$:

$$\dot{\theta}(t) = \pm \eta P(t) \frac{y(t)}{\sqrt{x(t)^2 + y(t)^2}}. \tag{5}$$

$P(t)$ is given by a weighted difference of the external signal $F(t)$ and the oscillator variable $x(t)$:

$$P(t) = \epsilon(t)F(t) - \beta(t)x(t) \tag{6}$$

with the adaptive coupling strengths $\epsilon(t)$ and $\beta(t)$. Following the discussion of the quality measures introduced before, for an optimal adaptation process, the dynamics of $\epsilon(t)$ and $\beta(t)$ has to fulfill two requirements: as long as the difference between the intrinsic frequency f and the target frequency f_{ext} of the external signal is high, $P(t)$ should basically be an amplified version of $F(t)$ in order to accelerate the adaptation process. In contrast, when f is already close to f_{ext}, $P(t)$ is supposed to attain values close to zero such as to reduce the influence of the external signal to a minimum. Both of these requirements can be fulfilled by adapting $\beta(t)$ and $\epsilon(t)$ according to a combination of correlation-based growth and a passive decay toward a low resting value. For $\beta(t)$, we propose the following dynamics:

$$\tau \dot{\beta}(t) = \beta_0 - \beta(t) + \kappa P(t)x(t) \tag{7}$$

with time constant τ and correlation learning rate κ. The value of β scales the subtraction of the system variable x from the external signal $F(t)$ in Equation (6). The product of P and x (averaged over time) is large if the difference between the intrinsic frequency f and the external target frequency f_{ext} is low. At this stage, the influence of the external signal on the oscillator should be reduced, i.e., the amplitude of P should be decreased, as done by increasing β. The proposed dynamics for $\epsilon(t)$ are very similar:

$$\tau \dot{\epsilon}(t) = \epsilon_0 - \epsilon(t) + \kappa F(t)P(t). \tag{8}$$

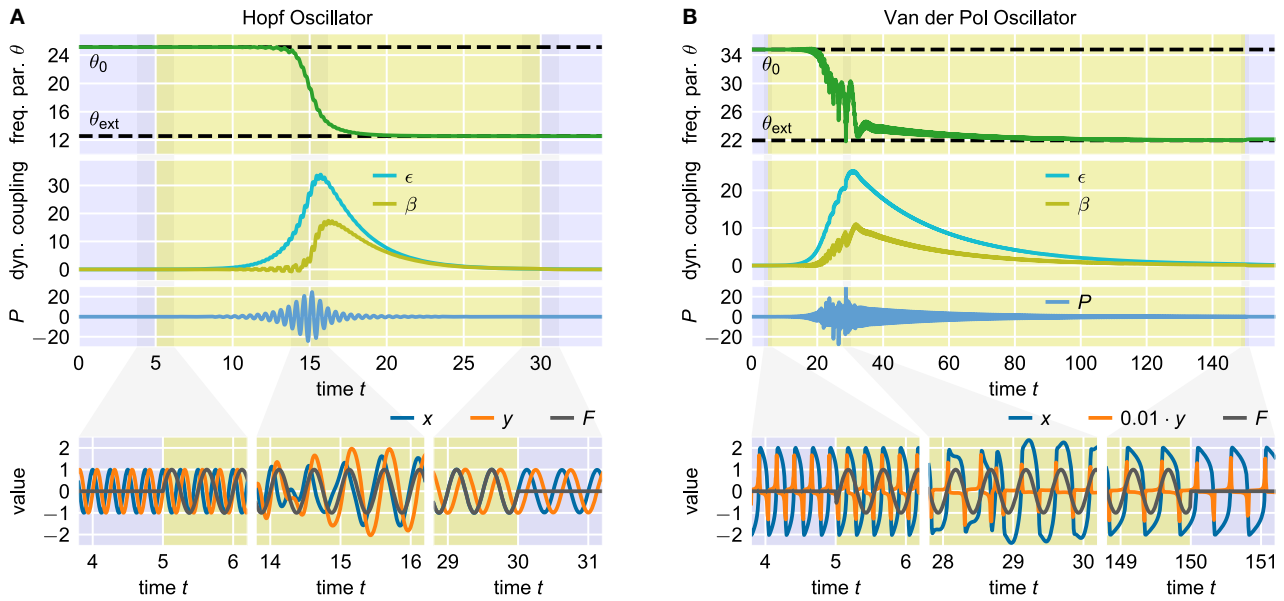


FIGURE 6 | Adaptation of two oscillators with the AFDC mechanism. The upper-most panels show the time course of the frequency determining parameter θ. The time during which the external signal is applied to the system is indicated by the yellow shaded area. The dashed horizontal lines indicate the initial value θ_0 and the value θ_{ext} corresponding to the exact frequency of the external signal. The second panels from the top show the time course of the adaptive coupling strengths β and ϵ. The third panels from the top show the time course of the filtered external signal P. The panels on the bottom show the oscillating state variables x and y and the external signal F at different short time windows during the adaptation process. In both cases, the initial intrinsic frequency of the oscillator is $f_0 = 4.0$ and the external signal is a sine wave with unit amplitude and frequency $f_{\text{ext}} = 2.0$. **(A)** Hopf oscillator with AFDC mechanism with $\mu = 1.0$, $\eta = 0.5$, $\kappa = 5.0$, $\tau = 2.0$, $\beta_0 = 0.0$ and $\epsilon_0 = 0.01$. The initial value of the parameter θ is given by $\theta_0 = 2\pi f_0 \approx 25.1$, the value corresponding to the frequency of the external signal is $\theta_{\text{ext}} = 2\pi f_{\text{ext}} \approx 12.6$. The external signal is applied for $5 \leq t < 30$. **(B)** Van der Pol oscillator with AFDC mechanism with $\mu = 100.0$, $\eta = 2.0$, $\kappa = 5.0$, $\tau = 15.0$, $\beta_0 = 0.0$ and $\epsilon_0 = 0.01$. The values of the parameter θ corresponding to f_0 and f_{ext} are determined to be $\theta_0 \approx 34.8$ and $\theta_{\text{ext}} \approx 22.0$ (see Methods). The external signal is applied for $5 \leq t < 150$.

The value of ϵ scales the influence of the external signal F on the filtered signal P (Equation 6). If the averaged product of F and P is large, this implies that the subtraction of x in Equation (6) cannot cancel the addition of F, i.e., the internal frequency of the oscillator is different from the target frequency of the external signal. Thus, an increase of ϵ is desired to increase the influence of the signal on the system and to herewith increase the adaptation speed. However, as for $\beta(t) \approx 0$, the last term of Equation (8) can be approximated by $\kappa F(t)^2 \epsilon(t)$, without adaptation of $\beta(t)$, the value of $\epsilon(t)$ would not return to ϵ_0 as long as the external signal is present and therefore would not allow precise adaptation. Only the interplay of the dynamics of $\epsilon(t)$, which detects the onset of an external signal with a frequency different from the intrinsic frequency of the oscillator, and of $\beta(t)$, which detects when the adaptation is nearly completed, allows fast as well as precise adaptation. In the following, we call the described mechanism "Adaptation through Fast Dynamical Coupling" (AFDC).

The process of frequency adaptation supported by the AFDC mechanism can be separated into several stages (**Figure 6**) as qualitatively described in the following: Before the onset of an external signal ($F = 0$), the average product of P and F is zero and the adaptive coupling constants β and ϵ converge toward their resting values $\beta_0/(1 + \kappa \bar{x^2})$ and ϵ_0. Here, $\bar{x^2}$ is the mean over time of the squared signal x^2. As soon as the external signal is applied, the average product of P and F gets positive (Equation 6). As a result of this, ϵ starts to increase (Equation 8). A higher value of ϵ, in turn, increases the average product of P and F. This establishes a positive feedback loop that leads to a fast increase of the amplitude of P. The high amplitude of P results in a large influence of the external signal on the oscillator (Equation 4) as well as in a fast adaptation of the frequency determining variable θ toward the frequency of F (Equation 5). As a consequence of both of these effects, the oscillator follows the external frequency implying a positive correlation between P and x. This correlation leads to an increase of β (Equation 7). Higher values of β decrease the amplitude of P (Equation 6) and, as such, also the average product between P and x. This is a negative feedback loop. Note that a decrease of the amplitude of P also reduces the average product of P and F and therefore breaks the positive feedback loop between ϵ and the average product of P and F (Equation 8) yielding a decline of both β and ϵ to their respective resting values. At this point, switching off the external signal does not significantly change the system dynamics as the influence of the external signal has already been reduced to a minimum.

In summary, the described interplay of the dynamics of the two adaptive coupling constants β and ϵ scales up the magnitude of the external signal as long as adaptation of θ is needed and reduces it once the value corresponding to the frequency of the external signal is reached.

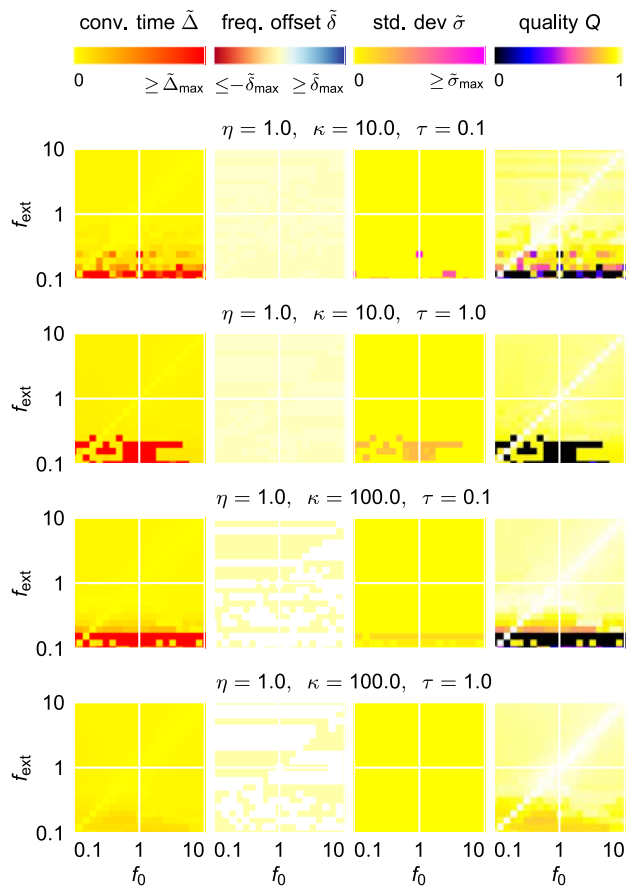


FIGURE 7 | Adaptation quality measures for the Hopf oscillator with AFDC mechanism in the f_0-f_{ext}-frequency space for different parameter values. For each of the given parameter tuples (η, κ, τ), from left to right, the relative convergence time $\tilde{\Delta}$, the relative final frequency offset $\tilde{\delta}$, the final relative frequency fluctuation $\tilde{\sigma}$, and the combined quality measure Q for the given parameter values of η, κ, and τ are shown in the plane spanned by the initial intrinsic frequency f_0 and the frequency of the external unit sine-wave signal f_{ext}.

2.2.1. Adaptation Quality in Frequency Space

The dynamics of the AFDC mechanism is mainly dominated by three free parameters: The time scale τ of the adaptive coupling strengths, the correlation learning rate κ and the learning rate η of the frequency determining variable θ. While, in general, an oscillator equipped with an AFDC mechanism shows more tolerance with respect to large frequency ranges, certain parameter combinations allow a faster or more precise adaptation over a larger frequency range (**Figure 7**). Some combinations (for instance $\eta = 1.0, \kappa = 100.0, \tau = 1.0$) result in a very good performance, as indicated by high values of Q, for the complete range of initial intrinsic frequencies f_0 and frequencies f_{ext} of the external signal analyzed here.

This is also reflected by the frequency space averaged quality $\langle Q \rangle$ (**Figure 8**). For a sufficiently high κ ($\kappa \gtrsim 3$), parameters ϵ and η can be found with an average quality value close to the theoretical maximum of 1 corresponding to very fast adaptation without significant frequency offset or frequency oscillations in the finally reached state (**Figure 8**). A comparison of the performance of the best found configuration of the regular adaptive Hopf oscillator with the performance of the best found configuration of the Hopf oscillator with AFDC mechanism shows that the AFDC mechanism outperforms the regular AFO mechanism in terms of all quality measures (**Figure 9A**). The same holds true for the comparison of the regular adaptive Van der Pol oscillator with the respective AFDC implementation (**Figure 9B**). In contrast to a regular AFO, the AFDC mechanism manages to provide fast and precise frequency adaptation over a wide frequency range with a fixed set of parameters.

Note that the values of the additional parameters ϵ_0 and β_0 do not significantly influence the dynamics of the mechanism as long as they are chosen reasonably low.

2.2.2. Neural Implementation

The AFDC mechanism relies on dynamically adapting the coupling strengths ϵ and β. In terms of signal flow, ϵ can be interpreted as a feedforward coupling from the external signal to the filtered signal P. The value of β, in turn, determines the strength of feedback coupling from the oscillator back to P. A standard way to implement this kind of signal flow between different entities is in terms of artificial neural networks. Neural networks are composed of multiple comparably simple computational units, the neurons, which project signals to each other via so-called synapses. Every synapse is characterized by a scalar value, the synaptic weight, which determines the efficacy of the synaptic signal transmission.

There exist neuron models on many different levels of abstraction, ranging from simple binary units to complex biophysical plausible spiking models. Here, we restrict ourselves to a very basic model of point-like neurons described by time-discrete dynamics. It has been shown that already a fully connected network of only two of these very simple neurons suffices to autonomously produce oscillatory signals (Pasemann et al., 2003). In every time step, each neuron sums up the incoming outputs from other neurons as well as from itself weighted by the respective synaptic weights. This sum is transformed into the new neural output by means of a sigmoidal transfer function. The weight matrix of this two neuron network is given by a scaled rotational matrix for a rotation angle φ. The value of φ monotonically controls the frequency of the obtained oscillatory signal of this so-called SO(2)-oscillator.

As already shown earlier (Nachstedt et al., 2012), a neural SO(2)-oscillator with neurons H_0 and H_1 can be extended by an AFDC mechanism by introducing an additional neuron H_2 into the system (**Figure 10A**). Now, this neural implementation can be understood as a special implementation of the general AFDC mechanism. The additional neuron H_2 is used to calculate the filtered external signal P by receiving synapses from both the external input F and the output of neuron H_0. The latter takes the role of the variable x of the general oscillators discussed above. The synaptic weight w_{2F} of the synapse from the external signal F to the additional neuron H_2 implements the dynamics of the ϵ coupling. The weight w_{20} of the synapse from the oscillator neuron H_0 to the neuron H_2 takes the role of β. Adapting the synaptic weights according to Equations (7) and (8)

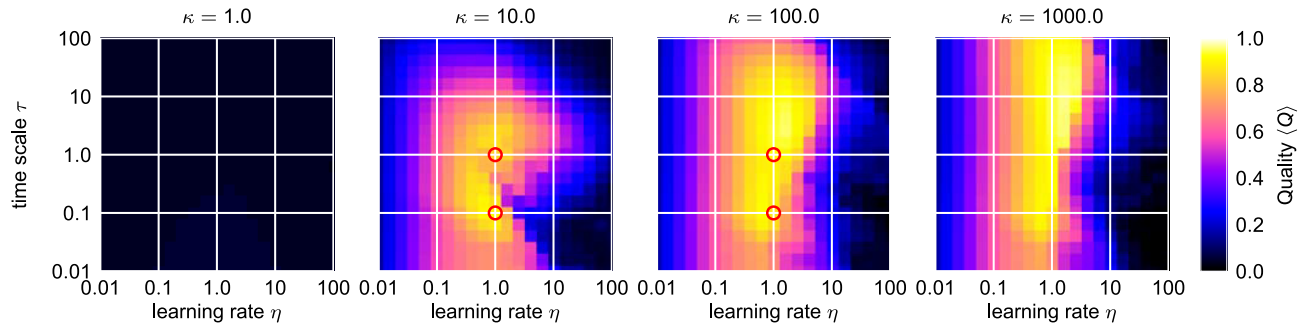


FIGURE 8 | Average combined quality measure ⟨Q⟩ for different values of κ in the ϵ-η-parameter space of the Hopf oscillator with AFDC mechanism. For every parameter triple of coupling strength ϵ, frequency learning rate η and correlation learning rate κ, the average adaption quality measure ⟨Q⟩ over the logarithmically sampled space of initial intrinsic frequencies f_0 and frequencies of the external signal f_{ext} is shown ($0.1 < f_0, f_{ext} < 10$). In each case, the external signal is a sine-wave with unit amplitude. The red circles indicate the four cases shown in **Figure 7**. Comparing these results to the ones obtained for the standard AFO in **Figure 5**, the AFDC mechanism provides significant higher quality values indicating versatility with respect to different initial intrinsic and external target frequencies.

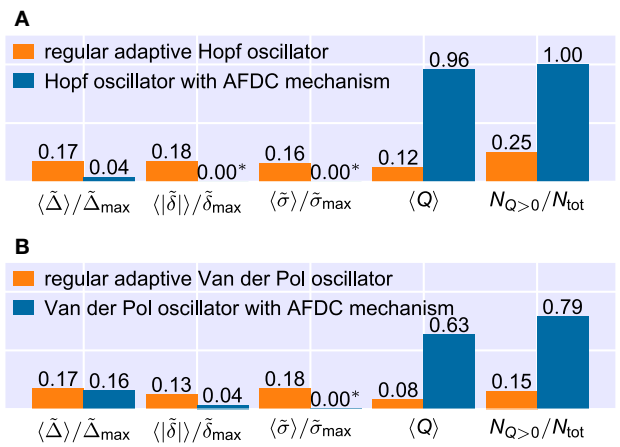


FIGURE 9 | Comparison of the frequency space averaged adaptation quality measures for the best found configurations of the regular adaptive oscillators and the respective oscillators with AFDC mechanism. Note that the averages of the relative convergence time $\tilde{\Delta}$, the final frequency offset $|\tilde{\delta}|$ and the relative final frequency fluctuation $\tilde{\sigma}$ include only values from (f_0, f_{ext})-frequency pairs in which the combined quality measure Q has a nonzero value. The ratio of the number $N_{Q>0}$ of (f_0, f_{ext})-pairs for which the quality Q has a nonzero value and the total number N_{tot} of frequency pairs is shown on the very right. All numbers are rounded. See methods for the used parameter values. **(A)** For the Hopf oscillator, all parameters are identical to the ones used in **Figures 7**, **8**. **(B)** For the Van der Pol oscillator, we adapt the maximal allowed values of the quality measures. We use $\tilde{\Delta}_{max} = 200$, $\tilde{\delta}_{max} = 0.10$ and $\tilde{\sigma}_{max} = 0.05$. In addition, we calculate $\tilde{\delta}$ and $\tilde{\sigma}$ directly from the frequency determining variable θ and consider the last deviation of θ of more than 10% from the finally reached mean value $\bar{\theta}$ to determine the adaptation time Δ. (*) Values shown as 0.00 are too small to be resolved in the figure. For the Hopf oscillator with AFDC mechanism, we find $\langle|\tilde{\delta}|\rangle/\tilde{\delta}_{max} \approx 5.3 \cdot 10^{-7}$ and $\langle\tilde{\sigma}\rangle/\tilde{\sigma}_{max} \approx 8.5 \cdot 10^{-8}$. For the Van der Pol oscillator with AFDC mechanism, the average normalized final frequency fluctuation is $\langle\tilde{\sigma}\rangle/\tilde{\sigma}_{max} \approx 2.5 \cdot 10^{-3}$.

effectively introduces synaptic plasticity into the system (Abbott and Nelson, 2000). In contrast to earlier publications (Nachstedt et al., 2012), here, the weight w_{02} of the synapse feeding the filtered signal P into the oscillator is simply kept constant.

The adaptation of the intrinsic oscillation frequency by modifying the parameter φ and hereby the synaptic weights of the neural SO(2)-oscillator is a long-lasting change. The discussed plasticity of the synaptic weights w_{20} and w_{2F}, in contrast, has a transient character. The combination of these two different kinds of dynamics results in a fast and precise adaptive neural oscillator (**Figure 10B**) (Nachstedt et al., 2012). This shows that the AFDC mechanism can be easily integrated into existing neural control schemes, for instance, in robotic applications. In addition, the successful implementation of the AFDC mechanism in a time-discrete system shows that the concept can be generalized to this class of dynamical systems.

2.2.3. Closed-Loop Locomotion Control

In addition to the open-loop scenarios studied so far, the AFDC mechanism also allows to apply adaptive oscillators in closed-loop scenarios where fast adaptation toward a specific frequency is required. A classical problem of robotic locomotion control is the task to find the optimal frequency to drive the legs of a walking machine. For animals, it has been found that the frequency during locomotion is tightly related to the resonant frequency of the free swinging leg (Holt et al., 1990). This way, animals are able to maintain energy efficiency during locomotion (Ahlborn and Blake, 2002). Furthermore, it has been proposed that animals actively modify the resonant frequency of their legs in order to optimize for different walking speeds (Ahlborn and Blake, 2002).

Given that CPGs control locomotion, adaptation of CPGs toward a system's resonant frequency to optimize locomotion has been repeatedly investigated and modeled (Verdaasdonk et al., 2006, 2009). A simplistic model of this control problem is given by a mathematical pendulum which is driven by a torque signal according to the output of an oscillator (Nachstedt et al., 2012). The most energy-efficient control is achieved if the pendulum is driven at its resonant frequency determined by its physical length l and its mass m as well as the current amplitude of its oscillation. Here, a neural SO(2)-oscillator with AFDC mechanism is used to control the torque applied to the pendulum. The control loop

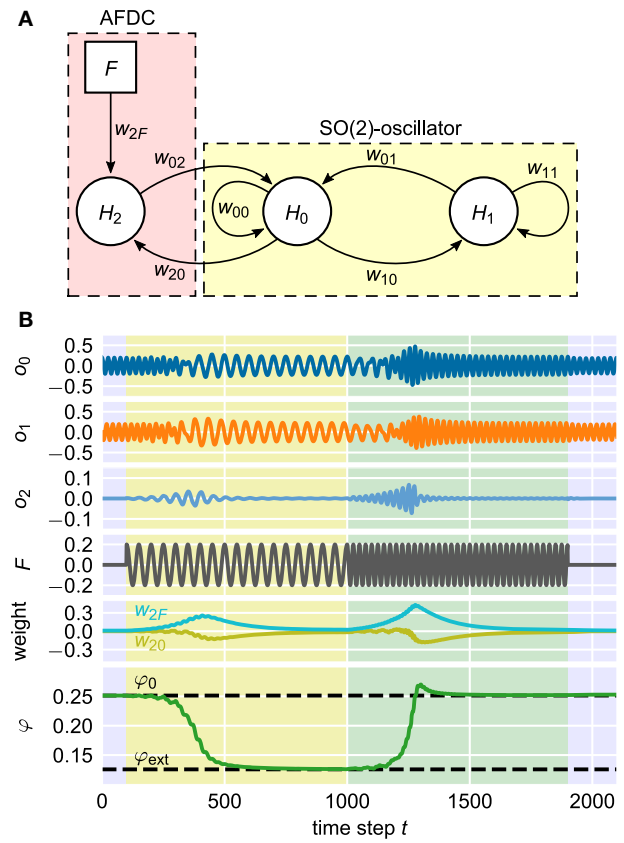


FIGURE 10 | Neural implementation of the AFDC mechanism. (A) The neurons H_0 and H_1 are fully connected by the synapses w_{00}, w_{01}, w_{10}, and w_{11} and form a neural SO(2)-oscillator. The neuron H_2 calculates the signal P which is the weighted difference between the external signal F and the activity value of H_0. Accordingly, the weight w_{2F} corresponds to the coupling strength ϵ and the weight w_{20} represents the variable β of the AFDC mechanism. The weight w_{02} can either be fixed at a positive value or adapted with similar dynamics as w_{20} and w_{2F}. **(B)** Example adaptation of the neural oscillator. It is initialized with an intrinsic frequency of $f_0 = 0.04$ corresponding to a value of $\varphi_0 = 0.25$ of the internal frequency determining variable. At time step $t = 100$, an external signal with a frequency of $f_{ext} = 0.02$ is applied until time step $t = 1,000$ (yellow shaded area). For $1,000 < t < 1,900$, the frequency of the external signal is changed to $f_{ext} = 0.04$ (green shaded area). For $t \geq 1,900$, there is no external signal. Shown from top to bottom are the activities o_i of the neurons H_i ($i \in \{1, 2, 3\}$), the external signal F, the synaptic weights w_{20} and w_{2F} and the frequency determining variable φ of the SO(2)-oscillator.

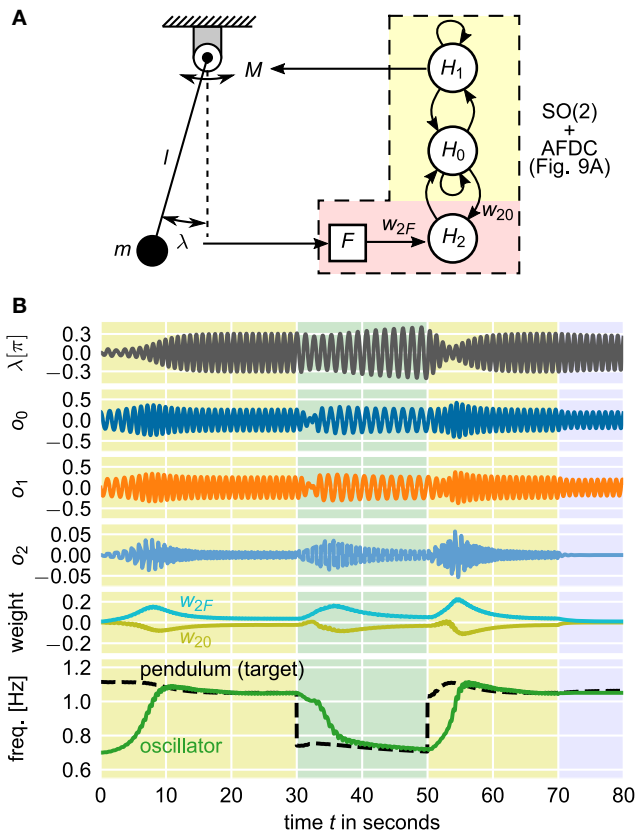


FIGURE 11 | Closed-loop pendulum control using a neural SO(2)-oscillator with AFDC mechanism. Energy-efficient control is realized if the pendulum is driven at its resonant frequency. **(A)** The output o_1 of neuron H_1 controls the torque M driving the pendulum with length l and mass m. The current angular displacement λ is converted into the external signal F which is fed back to the adaptive oscillator. The neural network is updated with a frequency of 25 Hz. **(B)** Simulation of the system with varying pendulum length l. The initial length of the pendulum is $l_0 = 0.2$ m. At $t = 30$ s, the length is changed to $l_1 = 0.4$ m. At $t = 50$ s, the original length l_0 is restored. At $t = 70$ s, the feedback connection from the pendulum to the oscillator is cut to demonstrate that the oscillator has indeed learned the correct frequency to drive the pendulum. Shown are the current angular displacement λ of the pendulum, the outputs o_0, o_1, and o_2 of the three neurons, the synaptic weights w_{2F} and w_{20} of the plastic synapses of the AFDC mechanism, and the intrinsic frequency of the oscillator and the resonant frequency of the undamped and undriven pendulum (target frequency for the oscillator). The resonant frequency of the pendulum does not only depend on the current physical properties of the pendulum but also on the current amplitude of its oscillations.

is closed by feeding the current position of the pendulum as external signal back to the oscillator (**Figure 11A**).

In this closed-loop system, the current frequency of the pendulum is completely determined by the current frequency of the driving neural oscillator. The observed oscillation frequencies of the pendulum and the neural oscillator are therefore always identical. Still, it is possible to adapt the intrinsic frequency of the neural oscillator to the target frequency given by the resonant frequency of the pendulum. The information about the difference between the intrinsic frequency of the oscillator and the resonant frequency of the pendulum is encoded in the phase relation between the internal oscillation and the feedback signal received as external signal by the oscillator. In particular, driving the pendulum at its resonant frequency is characterized by a phase shift of $\pi/2$ between the applied torque and the current angular position of the pendulum. In the neural SO(2)-oscillator, the same phase shift is found between the outputs of the neurons H_0 and H_1. Therefore, when using the output of H_1 to control the torque applied to the pendulum, at resonant frequency, the current angular position of the pendulum is exactly in phase with the output of neuron H_0. This corresponds to the converged state of the AFDC mechanism. If, in turn, the oscillation frequency of the neural SO(2)-oscillator is different from the resonant frequency of the pendulum, the output of H_0 and the angular

position of the pendulum are not in phase. The respective phase difference encodes the information about the difference between the intrinsic frequency of the neural oscillator and the resonant frequency of the pendulum and allows the adaptation of the former into the direction of the latter.

In our simulation, we first let the neural SO(2)-oscillator with AFDC mechanism adapt its intrinsic frequency toward the pendulum's resonant frequency ($0\,\mathrm{s} < t < 30\,\mathrm{s}$ in **Figure 11B**). We then simulate a change of the physical properties of the driven system by abruptly changing the length l of the pendulum. Accordingly, the neural oscillator readapts its intrinsic frequency to the new resonant frequency of the pendulum ($30\,\mathrm{s} < t < 50\,\mathrm{s}$ in **Figure 11B**). Afterwards, we change the length l back to its original value. Finally, we cut the feedback connection from the pendulum to the oscillator ($t > 70\,\mathrm{s}$ in **Figure 11B**) demonstrating that the oscillator has actually learned the proper frequency to drive the pendulum.

3. DISCUSSION

Transferring key concepts of biological solutions for complex control problems to robotic applications has been proven to be a promising approach regarding the adaptivity, robustness, versatility and agility found in biological organisms (Pfeifer et al., 2007). One especially successful concept is the one of using oscillators, i.e., CPGs, to control complex locomotion. As such, the study of nonlinear oscillators, their entrainment and adaptation properties and possible applications in robotics has gained a lot of interest. The here presented AFDC mechanism overcomes the demonstrated trade-off between speed and precision inherent to regular AFOs as introduced by Righetti et al. (2006). As a result, the AFDC mechanism allows fast and precise adaptation to external signals for a wide range of frequencies with a fixed set of parameters.

Since the discovery of the AFO mechanism, various different mechanisms to improve or extend the adaptation capabilities have been proposed. Subtracting the output of an oscillator from the external signal, as also done in the AFDC mechanism, was used to decompose a signal into its Fourier components (Ronsse et al., 2011b) with the help of an array of AFOs. In order to more reliably find the basic frequency of the external signal, it was proposed to combine a single adaptive frequency oscillator with a Fourier decomposition (Petric et al., 2011). The detailed interaction between multiple AFOs has been studied in the context of networks of self-adaptive dynamical systems (Rodriguez and Hongler, 2014). As an alternative to adapting the system parameters in order to modify the frequency of an oscillator, switching between different oscillation frequencies of an oscillator operated in the chaotic regime by dynamically stabilizing different periods was demonstrated (Steingrube et al., 2010).

The main novelty of the here presented mechanism is the introduced dynamics of the adaptive coupling strengths between the external signal and the filtered signal as well as between the output of the oscillator and the filtered signal. This dynamics temporally increases the influence of the signal on the oscillator as long as it is necessary to achieve fast adaptation and decreases it once precision is needed toward the end of the adaptation process. Adaptive coupling strengths have been proposed earlier as a method to increase the synchronization in a network of phase oscillators with fixed intrinsic frequencies (Ren and Zhao, 2007). The interaction of the transient dynamics of the adaptive coupling strengths on the one hand and the permanent change of intrinsic frequency on the other hand resembles the interplay of long-term (Wood et al., 2011) and short-term (Zucker and Regehr, 2002) plasticity in biological organisms. The interplay of long-term and short-term plasticity in biological system has already been shown to be highly relevant for biological motor control, in particular for fast network reconfiguration (Nadim and Manor, 2000).

The AFDC mechanism increases the complexity of the oscillatory system by the addition of two dynamical equations. Their interplay is required to first scale up the influence of the external signal and later on reduce it again. In particular, this interplay is enabled by the weighted difference P of the external signal F and the oscillator variable x. The correlation of P and F determines the growth of the adaptive coupling constant ϵ which, in turn, increases the correlation of P and F. To counterbalance this self-enhancing dynamics, a second dynamic variable, i.e., β, is required. To make β increase, F and x have to be correlated which is the case once the oscillator has attained the externally applied frequency. This delay of the onsets of the growth processes of ϵ and β is crucial for the AFDC mechanism and cannot be realized by a single variable.

In this contribution, we focused on the Hopf oscillator and the Van der Pol oscillator for the detailed analyses of the regular AFO and the AFDC mechanism. It remains an interesting question for future research in how far the results obtained for these oscillators regarding the frequency-independent choice of parameters as well as regarding the quality measures of the adaptation process generalize to other types of nonlinear oscillators (Rayleigh, 1877; Duffing, 1918; Fitzhugh, 1961).

As the dynamics of the coupling strengths in the AFDC mechanism is solely correlation based, we showed that it is easy to implement the mechanism in neural networks. We demonstrated this by discussing the already earlier published neural time-discrete SO(2)-oscillator with AFDC mechanism (Nachstedt et al., 2012). This special realization of the AFDC mechanism has already been successfully applied in different robotic applications including self-organized control of a snake-like robot (Nachstedt et al., 2013), adaptive control of a robot leg with compliant tarsus (Canio et al., 2016b), and bipedal locomotion with robustness against global loss of sensory feedback (Canio et al., 2016a). In contrast to the neural implementation, the new general formulation of the AFDC mechanism makes it possible to apply the mechanism to all kinds of existing applications of regular AFOs (Buchli et al., 2005). Additionally, it allows the usage of adaptive oscillators in completely new scenarios where, up to now, regular AFOs could not provide sufficiently fast as well as precise adaptation.

4. MATERIALS AND METHODS

4.1. Hopf Oscillator

The regular Hopf oscillator with the state variables x and y is given by the following system of dynamical equations:

$$\begin{aligned}\dot{x}(t) &= \left(\mu - r(t)^2\right) x(t) - \theta y(t)\\ \dot{y}(t) &= \left(\mu - r(t)^2\right) y(t) + \theta x(t)\end{aligned} \tag{9}$$

with $r(t) = \sqrt{x(t)^2 + y(t)^2}$. The variable $\mu > 0$ determines the amplitude of the oscillations. Without an external signal ($F(t) = 0$), this system possesses an asymptotically stable and harmonic limit cycle with an angular frequency of exactly θ.

4.1.1. Adaptive Frequency Hopf Oscillator

The Hopf oscillator can be turned into an adaptive frequency oscillator by coupling an external signal F to the system and introducing the dynamics described by Equation (2) to the parameter θ. The complete system is given by

$$\begin{aligned}\dot{x}(t) &= \left(\mu - r(t)^2\right) x(t) - \theta(t) y(t) + \epsilon F(t)\\ \dot{y}(t) &= \left(\mu - r(t)^2\right) y(t) + \theta(t) x(t)\\ \dot{\theta}(t) &= -\eta F(t) \frac{y(t)}{\sqrt{x(t)^2 + y(t)^2}}.\end{aligned} \tag{10}$$

4.1.2. Hopf Oscillator with Fast Dynamical Coupling

The Hopf oscillator equipped with the AFDC mechanism is given by the following system of differential equations:

$$\begin{aligned}\dot{x}(t) &= \left(\mu - r(t)^2\right) x(t) - \theta(t) y(t) + P(t)\\ \dot{y}(t) &= \left(\mu - r(t)^2\right) y(t) + \theta(t) x(t)\\ \tau \dot{\beta}(t) &= \beta_0 - \beta(t) + \kappa P(t) x(t)\\ \tau \dot{\epsilon}(t) &= \epsilon_0 - \epsilon(t) + \kappa F(t) P(t)\\ \dot{\theta}(t) &= -\eta P(t) \frac{y(t)}{\sqrt{x(t)^2 + y(t)^2}}\end{aligned} \tag{11}$$

with $P(t) = \epsilon(t) F(t) - \beta(t) x(t)$.

4.2. Van der Pol Oscillator

The Van der Pol oscillator with the state variables x and y is defined as follows:

$$\begin{aligned}\dot{x}(t) &= y(t)\\ \dot{y}(t) &= \mu \left(1 - x(t)^2\right) y(t) - \theta^2 x(t).\end{aligned} \tag{12}$$

The parameter $\mu > 0$ determines the "degree of nonlinearity" of the system. For $\mu = 0$, the system is harmonic. The intrinsic frequency f depends in a nonlinear and non-trivial way on the parameter θ. We use a Fourier transform in conjunction with a sequence of nested intervals to determine the values of θ corresponding to a given frequency f.

4.2.1. Adaptive Frequency Van der Pol Oscillator

The adaptive frequency formulation of the Van der Pol Oscillator coupled to a time-dependent external signal $F(t)$ requires a positive sign in Equation (2):

$$\begin{aligned}\dot{x}(t) &= y(t) + \epsilon F(t)\\ \dot{y}(t) &= \mu \left(1 - x(t)^2\right) y(t) - \theta(t)^2 x\\ \dot{\theta}(t) &= +\eta F(t) \frac{y(t)}{\sqrt{x(t)^2 + y(t)^2}}.\end{aligned} \tag{13}$$

4.2.2. Van der Pol Oscillator with Fast Dynamical Coupling

Applying the AFDC mechanism to the Van der Pol oscillator is described by the following system of differential equations:

$$\begin{aligned}\dot{x}(t) &= y(t) + P(t)\\ \dot{y}(t) &= \mu \left(1 - x(t)^2\right) y(t) - \theta(t)^2 x\\ \tau \dot{\beta}(t) &= \beta_0 - \beta(t) + \kappa P(t) x(t)\\ \tau \dot{\epsilon}(t) &= \epsilon_0 - \epsilon(t) + \kappa F(t) P(t)\\ \dot{\theta}(t) &= +\eta P(t) \frac{y(t)}{\sqrt{x(t)^2 + y(t)^2}}\end{aligned} \tag{14}$$

with $P(t) = \epsilon(t) F(t) - \beta(t) x(t)$.

4.3. Neural SO(2)-Oscillator

We use standard additive time-discrete neurons H_i, $i \in \{0, \ldots, N-1\}$, where N is the number of neurons in the network. The activation a_i of neuron H_i at time $t+1$ is given by the sum of incoming presynaptic neural firing rates o_j weighted by the synaptic weights w_{ij} at time t:

$$a_i(t+1) = \sum_{j=0}^{N-1} w_{ij}(t) o_j(t), \quad i = 0, \ldots, N-1. \tag{15}$$

The activation a_i of neuron H_i is transformed into its firing rate o_i by a sigmoidal transfer function:

$$o_i(t) = \tanh\left(a_i(t)\right). \tag{16}$$

The pure SO(2)-network consists of $N = 2$ fully connected neurons H_0 and H_1. The synaptic weight matrix is chosen according to

$$\begin{pmatrix} w_{00}(t) & w_{01}(t)\\ w_{10}(t) & w_{11}(t)\end{pmatrix} = \alpha \cdot \begin{pmatrix} \cos\varphi(t) & \sin\varphi(t)\\ -\sin\varphi(t) & \cos\varphi(t)\end{pmatrix} \tag{17}$$

with $0 < \varphi(t) < \pi$ the frequency determining parameter. The factor α determines the amplitude as well as the nonlinearity of the oscillations. We use $\alpha = 1.01$ to obtain very harmonic oscillations and an approximately linear relationship between φ and the intrinsic frequency of the oscillator.

4.3.1. SO(2)-Oscillator with Fast Dynamical Coupling

In order to equip the neural SO(2)-oscillator with the AFDC mechanism, an additional neuron $H_2$is introduced. The external signal $F(t)$ is fed into the neuron H_2 via a synapse w_{2F}. The neuron H_2 calculates the filtered version of the external signal and receives signals via the synapses w_{20} $(= \beta)$ and w_{2F} $(= \epsilon)$ governed by the following plasticity rules:

$$\begin{aligned} w_{20}(t+1) &= w_{20}(t) + (\beta_0 - w_{20}(t) - \kappa o_2(t) o_0(t))/\tau \\ w_{2F}(t+1) &= w_{2F}(t) + (\epsilon_0 - w_{2F}(t) - \kappa F(t) o_2(t))/\tau. \end{aligned} \tag{18}$$

In accordance with our earlier publication (Nachstedt et al., 2012), we simplify the frequency adaptation rule of the AFDC mechanism and reformulate it in terms of the signals arriving at neuron H_0:

$$\varphi(t+1) = \phi(t) + \eta w_{02}(t) o_2(t) w_{01}(t) o_1(t). \tag{19}$$

For the example adaptation process (**Figure 10B**), we use $\alpha = 1.01, \eta = 1, \kappa = 100, \tau = 100, \beta_0 = 0$ and $\epsilon_0 = 0.01$.

4.4. Mathematical Pendulum

The angular displacement λ of a mathematical pendulum with length l and mass m is described by the following differential equation:

$$\ddot{\lambda} = -\frac{g}{l}\sin\lambda - \frac{D}{ml^2}\dot{\lambda} + \frac{M}{ml^2} \tag{20}$$

with the gravitational acceleration g, the external torque M evoked on the system and the damping constant D. The resonant frequency f_{res} of the undamped $(D = 0)$ and undriven $(M = 0)$ mathematical pendulum is given by Ochs (2011):

$$f_{\text{res}} = \frac{\omega_0}{4K(k)} \tag{21}$$

with

$$k = \frac{\dot{\lambda}^2 + 4\omega_0^2(\sin\frac{\lambda}{2})^2}{4\omega_0^2} \tag{22}$$

and $\omega_0 = \sqrt{\frac{g}{l}}$. $K(k)$ is the complete elliptic integral of the first kind. In Equation (22), the current values of the angular displacement λ and the angular velocity $\dot{\lambda}$ are used to obtain the current total energy of the system. For our simulations, we use $g = 9.81\,\text{mm}^{-2}$ and $D = 0.005\,\text{kg}\,\text{m}^2\,\text{s}^{-1}$.

4.5. Numerical Integration

The integrations of the different differential systems are carried out using the odeint method of the scipy python package (Jones et al., 2001). This methods relies on the LSODA algorithm (Brown and Hindmarsh, 1989) from the FORTRAN library odepack (Hindmarsh, 1983). The LSODA algorithm utilizes an adaptive step size.

4.6. Frequency and Parameter Scans

For the frequency scans performed for the adaptive Hopf oscillator (**Figures 4**, **5**) and the adaptive Van der Pol oscillator as well as for the respective oscillators with AFDC mechanism (**Figures 7**, **8**), we sample the frequency space in the range $0.1 \leq f_0, f_{\text{ext}} \leq 10.0$. We consider 21 sample values uniformly spaced on a logarithmic axis of f_0 and f_{ext} and investigate the behavior of the oscillators for all possible 21^2 (f_0, f_{ext})-pairs. For every frequency pair, in the case of the regular adaptive oscillators, we sample 21 parameter values again uniformly spaced on a logarithmic axes of each ϵ and η in the range $0.01 \leq \epsilon, \eta \leq 100$. Therefore, we investigate a total of 21^4 $(f_0, f_{\text{ext}}, \epsilon, \eta)$-configurations for each regular adaptive oscillator. In the case of the oscillators with AFDC mechanism, the parameters are investigated in the ranges $0.01 \leq \eta, \tau \leq 100$ and $1 \leq \kappa \leq 1,000$ again with 21 samples in every parameter dimension yielding a total of 21^5 sampled $(f_0, f_{\text{ext}}, \eta, \kappa, \tau)$-configurations each.

The best sampled parameter values of the frequency space averaged combined quality measure $\langle Q \rangle$ are $\epsilon \approx 15.85$ and $\eta \approx 15.85$ for the regular adaptive Hopf oscillator and $\eta \approx 1.58$, $\kappa \approx 398.11$ and $\tau \approx 3.98$ for the Hopf oscillator with AFDC mechanism (**Figure 9**). For the regular adaptive Van der Pol oscillator, we find $\epsilon \approx 0.0158$ and $\eta = 1.0$ to perform best while $\eta \approx 0.158$, $\kappa = 100$ and $\tau \approx 1.585$ yield the best result for the Van der Pol oscillator with AFDC mechanism.

AUTHOR CONTRIBUTIONS

The AFDC mechanism was developed by TN and PM. TN and CT planned the presented analyses and experiments. Implementation and analysis of the data was done by TN. TN, CT, and PM wrote and reviewed the manuscript.

FUNDING

The research leading to these results has received funding from the Federal Ministry of Education and Research (BMBF) Germany to the Göttingen Bernstein Center for Computational Neuroscience under grant numbers 01GQ1005A [TN, PM] and 01GQ1005B [CT] and from the International Max Planck Research School for Physics of Biological and Complex Systems (IMPRS-PBCS) by stipends of the country of Lower Saxony with funds from the initiative Niedersächsisches Vorab and of the University of Göttingen [TN].

ACKNOWLEDGMENTS

We thank Florentin Wörgötter for fruitful discussions. We acknowledge support by the Open Access Publication Funds of the Göttingen University.

REFERENCES

Abbott, L. F., and Nelson, S. B. (2000). Synaptic plasticity: taming the beast. *Nat. Neurosci.* 3(Suppl.), 1178–1183. doi: 10.1038/81453

Ahlborn, B. K., and Blake, R. W. (2002). Walking and running at resonance. *Zoology (Jena)* 105, 165–174. doi: 10.1078/0944-2006-00057

Barkai, N., and Leibler, S. (2000). Circadian clocks limited by noise. *Nature* 403, 267–268. doi: 10.1038/35002258

Brown, P. N., and Hindmarsh, A. C. (1989). Reduced storage matrix methods in stiff ODE systems. *Appl. Math. Comput.* 31, 40–91. doi: 10.1016/0096-3003(89)90110-0

Buchli, J., Righetti, L., and Ijspeert, A. J. (2005). "A dynamical systems approach to learning: a frequency-adaptive hopper robot," in *Proceedings of the 8th European Conference on Advances in Artificial Life*, eds M. S. Capcarrère, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis (Berlin; Heidelberg: Springer), 210–220.

Buchli, J., Righetti, L., and Ijspeert, A. J. (2006). Engineering entrainment and adaptation in limit cycle systems: from biological inspiration to applications in robotics. *Biol. Cybern.* 95, 645–664. doi: 10.1007/s00422-006-0128-y

Buchli, J., Righetti, L., and Ijspeert, A. J. (2008). Frequency analysis with coupled nonlinear oscillators. *Physica D* 237, 1705–1718. doi: 10.1016/j.physd.2008.01.014

Canio, G. D., Stoyanov, S., Balmori, I. T., Larsen, J. C., and Manoonpong, P. (2016a). "Adaptive combinatorial neural control for robust locomotion of a biped robot," in *From Animals to Animats 14: Proceedings of the 11th International Conference on Simulation of Adaptive Behavior* (Cham: Springer International Publishing), 317–328.

Canio, G. D., Stoyanov, S., Larsen, J. C., Hallam, J., Kovalev, A., Kleinteich, T., et al. (2016b). A robot leg with compliant tarsus and its neural control for efficient and adaptive locomotion on complex terrains. *Art. Life Robot.* 21, 274–281. doi: 10.1007/s10015-016-0296-3

Cruse, H., Bartling, C., Dreifert, M., Schmitz, J., Brunn, D. E., Dean, J., et al. (1995). Walking: a complex behavior controlled by simple networks. *Adapt. Behav.* 3, 385–418. doi: 10.1177/105971239500300403

Duffing, G. (1918). *Erzwungene Schwingungen bei Veränderlicher Eigenfrequenz, 1st Edn*. Braunschweig: F. Vieweg u. Sohn.

Fitzhugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.* 1, 445–466. doi: 10.1016/S0006-3495(61)86902-6

Foth, E., and Bässler, U. (1985). Leg movements of stick insects walking with five legs on a treadwheel and with one leg on a motor-driven belt. I. general results and 1:1-coordination. *Biol. Cybern.* 51, 313–318. doi: 10.1007/BF00336918

Furuta, K. (2003). "Control of pendulum: From super mechano-system to human adaptive mechatronics," in *Proceedings of the IEEE Conference on Decision and Control* (Piscataway, NJ: IEEE), 1498–1507.

Goldbeter, A., Gérard, C., Gonze, D., Leloup, J. C., and Dupont, G. (2012). Systems biology of cellular rhythms. *FEBS Lett.* 586, 2955–2965. doi: 10.1016/j.febslet.2012.07.041

Hellgren, J., and Grillner, S. (1992). Computer simulation of the segmental neural network generating locomotion in lamprey by using populations of network interneurons. *Biol. Cybern.* 68, 1–13. doi: 10.1007/BF00203132

Hindmarsh, A. C. (1983). "ODEPACK, a systematized collection of ode solvers," in *IMACS Transactions on Scientific Computation: Scientific Computing*, eds R. S. Stepleman, C. Carver, R. Peskin, W. F. Ames, and R. Vienevetsky (Amsterdam: North-Holland), 55–63.

Holt, K. G., Hamill, J., and Andres, R. O. (1990). The force-driven harmonic oscillator as a model for human locomotion. *Hum. Mov. Sci.* 9, 55–68. doi: 10.1016/0167-9457(90)90035-C

Hooper, S. L. (2001). "Central pattern generators," in *Encyclopedia of Life Sciences* (New York, NY: Wiley), 1–12. Available online at: http://onlinelibrary.wiley.com/doi/10.1038/npg.els.0000032/abstract

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural. Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Jones, E., Oliphant, T., Peterson, P., and Al, E. (2001). *Scipy: Open Source Scientific Tools for Python*.

Kimura, H., Fukuoka, Y., and Konaga, K. (2001). Adaptive dynamic walking of a quadruped robot using a neural system model. *Adv. Robot.* 15, 859–878. doi: 10.1163/156855301317198179

Marder, E., and Bucher, D. (2001). Central pattern generators and the control of rhythmic movements. *Curr. Biol.* 11, R986–R996. doi: 10.1016/S0960-9822(01)00581-4

Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol. Cybern.* 52, 367–376. doi: 10.1007/BF00449593

Nachstedt, T., Wörgötter, F., and Manoonpong, P. (2012). "Adaptive neural oscillator with synaptic plasticity enabling fast resonance tuning," in *International Conference on Artificial Neural Networks, ICANN2012* (Berlin; Heidelberg: Springer), 451–458. doi: 10.1007/978-3-642-33269-2_57

Nachstedt, T., Wörgötter, F., Manoonpong, P., Ariizumi, R., Ambe, Y., and Matsuno, F. (2013). "Adaptive neural oscillators with synaptic plasticity for locomotion control of a snake-like robot with screw-drive mechanism," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation* (Piscataway, NJ: IEEE), 3389–3395.

Nadim, F., and Manor, Y. (2000). The role of short-term synaptic dynamics in motor control. *Curr. Opin. Neurobiol.* 10, 683–690. doi: 10.1016/S0959-4388(00)00159-8

Nakamura, Y., Mori, T., Sato, M. A., and Ishii, S. (2007). Reinforcement learning for a biped robot based on a CPG-actor-critic method. *Neural Netw.* 20, 723–735. doi: 10.1016/j.neunet.2007.01.002

Nassour, J., Hénaff, P., Benouezdou, F., and Cheng, G. (2014). Multi-layered multi-pattern CPG for adaptive locomotion of humanoid robots. *Biol. Cybern.* 108, 291–303. doi: 10.1007/s00422-014-0592-8

Ochs, K. (2011). A comprehensive analytical solution of the nonlinear pendulum. *Eur. J. Phys.* 32, 479–490. doi: 10.1088/0143-0807/32/2/019

Pasemann, F., Hild, M., and Zahedi, K. (2003). "SO(2)-networks as neural oscillators," in *Computational Methods in Neural Modeling*, ed J. Mira (Berlin; Heidelberg: Springer), 1042–1042.

Petric, T., Gams, A., Ijspeert, A. J., and Zlajpah, L. (2011). On-line frequency adaptation and movement imitation for rhythmic robotic tasks. *Int. J. Rob. Res.* 30, 1775–1788. doi: 10.1177/0278364911421511

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Pinto, C. M. A., Rocha, D., and Santos, C. P. (2012). Hexapod robots: new CPG model for generation of trajectories. *SIAM J. Numer. Anal.* 7, 15–26. doi: 10.1063/1.3636775

Rayleigh, J. W. S. (1877). *The Theory of Sound, 1st Edn*. London: Macmillan and Co.

Ren, Q., and Zhao, J. (2007). Adaptive coupling and enhanced synchronization in coupled phase oscillators. *Phys. Rev. E* 76, 1–6. doi: 10.1103/PhysRevE.76.016207

Righetti, L., Buchli, J., and Ijspeert, A. J. (2006). Dynamic hebbian learning in adaptive frequency oscillators. *Physica D* 216, 269–281. doi: 10.1016/j.physd.2006.02.009

Righetti, L., Buchli, J., and Ijspeert, A. J. (2009). Adaptive frequency oscillators and applications. *Open Cybern. Syst. J.* 3, 64–69. doi: 10.2174/1874110X00903010064

Righetti, L., and Ijspeert, A. J. (2008). "Pattern generators with sensory feedback for the control of quadruped locomotion," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Piscataway, NJ: IEEE), 819–824.

Rodriguez, J., and Hongler, M. O. (2014). Networks of self-adaptive dynamical systems. *IMA. J. Appl. Math.* 79, 201–240. doi: 10.1093/imamat/hxs057

Ronsse, R., Lenzi, T., Vitiello, N., Koopman, B., Van Asseldonk, E., De Rossi, S. M. M., et al. (2011a). Oscillator-based assistance of cyclical movements: model-based and model-free approaches. *Med. Biol. Eng. Comput.* 49, 1173–1185. doi: 10.1007/s11517-011-0816-1

Ronsse, R., van den Kieboom, J., and Ijspeert, A. J. (2011b). "Automatic resonance tuning and feedforward learning of biped walking using adaptive oscillators," in *Proceeding of the Conference in Multibody Dynamics 2011, ECCOMAS Thematic Conference* (Louvain: Université catholique de Louvain), 4–7.

Santos, C. P., Alves, N., and Moreno, J. C. (2017). Biped locomotion control through a biomimetic CPG-based controller. *J. Intell. Robot. Syst.* 85, 47–70. doi: 10.1007/s10846-016-0407-3

Spong, M. W. (1995). Swing up control problem for the acrobot. *IEEE Control Syst. Mag. N. Y.* 15, 49–55. doi: 10.1109/37.341864

Steingrube, S., Timme, M., Wörgötter, F., and Manoonpong, P. (2010). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nat. Phys.* 6, 224–230. doi: 10.1038/nphys1508

Tropea, P., Vitiello, N., Martelli, D., Aprigliano, F., Micera, S., and Monaco, V. (2015). Detecting slipping-like perturbations by using adaptive oscillators. *Ann. Biomed. Eng.* 43, 416–426. doi: 10.1007/s10439-014-1175-5

Van der Pol, B. (1920). A theory of the amplitude of free and forced triode vibrations. *Radio Rev.* 11, 701–710.

Verdaasdonk, B. W., Koopman, H. F., and Helm, F. C. (2006). Energy efficient and robust rhythmic limb movement by central pattern generators. *Neural Netw.* 19, 388–400. doi: 10.1016/j.neunet.2005.09.003

Verdaasdonk, B. W., Koopman, H. F., and van der Helm, F. C. (2009). Energy efficient walking with central pattern generators: from passive dynamic walking to biologically inspired control. *Biol. Cybern.* 101, 49–61. doi: 10.1007/s00422-009-0316-7

Wang, T., Hu, Y., and Liang, J. (2013). Learning to swim: a dynamical systems approach to mimicking fish swimming with CPG. *Robotica* 31, 361–369. doi: 10.1017/S0263574712000343

Winfree, A. T. (1967). Biological rhythms and the behavior of populations of coupled oscillators. *J. Theor. Biol.* 16, 15–42. doi: 10.1016/0022-5193(67)90051-3

Wood, R., Baxter, P., and Belpaeme, T. (2011). A review of long-term memory in natural and synthetic systems. *Adapt. Behav.* 20, 81–103. doi: 10.1177/1059712311421219

Zucker, R. S., and Regehr, W. G. (2002). Short-term synaptic plasticity. *Annu. Rev. Physiol.* 64, 355–405. doi: 10.1146/annurev.physiol.64.092501.114547

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Nachstedt, Tetzlaff and Manoonpong. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 04 April 2017
doi: 10.3389/fnbot.2017.00018

Development and Training of a Neural Controller for Hind Leg Walking in a Dog Robot

Alexander Hunt[1]*, *Nicholas Szczecinski*[2] *and Roger Quinn*[2]

[1] Department of Mechanical and Materials Engineering, Portland State University, Portland, OR, USA, [2] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, USA

Animals dynamically adapt to varying terrain and small perturbations with remarkable ease. These adaptations arise from complex interactions between the environment and biomechanical and neural components of the animal's body and nervous system. Research into mammalian locomotion has resulted in several neural and neuro-mechanical models, some of which have been tested in simulation, but few "synthetic nervous systems" have been implemented in physical hardware models of animal systems. One reason is that the implementation into a physical system is not straightforward. For example, it is difficult to make robotic actuators and sensors that model those in the animal. Therefore, even if the sensorimotor circuits were known in great detail, those parameters would not be applicable and new parameter values must be found for the network in the robotic model of the animal. This manuscript demonstrates an automatic method for setting parameter values in a synthetic nervous system composed of non-spiking leaky integrator neuron models. This method works by first using a model of the system to determine required motor neuron activations to produce stable walking. Parameters in the neural system are then tuned systematically such that it produces similar activations to the desired pattern determined using expected sensory feedback. We demonstrate that the developed method successfully produces adaptive locomotion in the rear legs of a dog-like robot actuated by artificial muscles. Furthermore, the results support the validity of current models of mammalian locomotion. This research will serve as a basis for testing more complex locomotion controllers and for testing specific sensory pathways and biomechanical designs. Additionally, the developed method can be used to automatically adapt the neural controller for different mechanical designs such that it could be used to control different robotic systems.

Keywords: central pattern generator, dog, artificial muscle, locomotion, walking

OPEN ACCESS

Edited by:
Poramate Manoonpong,
University of Southern Denmark
Odense, Denmark

Reviewed by:
Daniel P. Ferris,
University of Michigan, USA
Horacio Rostro Gonzalez,
Universidad de Guanajuato, Mexico

***Correspondence:**
Alexander Hunt
ajh26@pdx.edu

Received: *25 November 2016*
Accepted: *15 March 2017*
Published: *04 April 2017*

Citation:
Hunt A, Szczecinski N and Quinn R (2017) Development and Training of a Neural Controller for Hind Leg Walking in a Dog Robot. Front. Neurorobot. 11:18. doi: 10.3389/fnbot.2017.00018

1. INTRODUCTION

Controlling complex robots using traditional control methods with on-line optimization and "single brain" control becomes increasingly difficult and computationally intensive as more degrees of freedom and more points of contact are added. This is in stark contrast with the animal kingdom, in which high redundancy is the norm, and complex interactions with the environment are often accomplished with ease. For example, having more feet on the ground makes an individual animal's

control and balance easier, rather than harder. Big or small, it takes little mental effort on the part of the animal to change from fast speeds to slow speeds, change gaits, start turning, step over an object, respond to ground slip, or move from concrete to loose dirt.

Unfortunately, animals are immensely complex, and the majority of our current robots barely resemble any animals in the world today. Instead of muscles for actuation, our most agile robots use electric motors (Seok et al., 2015) or hydraulics (Raibert et al., 2008; Boaventura et al., 2013). For determining body states and sensing the world, modern robots rely on a few strategically placed sensors instead of an animal's wide net of somatic sensory neurons spread across its whole body. For control, instead of a highly distributed and hierarchical network of neurons, a single algorithm is often used to calculate the exact position of each joint needed to maintain stability and provide locomotion.

All this is beginning to change however, as details on how biomechanics and neural systems provide advantages to moving around in the world are being uncovered. The compliant nature of muscles can automatically reject perturbations and significantly reduce the burden on the control system (Loeb et al., 1999; Jindrich and Full, 2002). To take advantage of this, actuators which add compliance and greater control of force are being developed (Pratt and Williamson, 1995; Thorson and Caldwell, 2011; Rollinson et al., 2013; Schilling et al., 2013b). A compliant actuator combined with the tri-segmented shape of the legs (Fischer and Blickhan, 2006) produces a mechanical system which is robust to perturbations capable of performing dynamic walking with open-loop control (Spröwitz et al., 2014).

Neural control of locomotion is a complex interaction of rhythm generation, sensory processing, feed-forward muscle activation, and sensory feedback systems. Central pattern generators (CPGs) are sub-circuits located in the spinal cord which are responsible for repetitive behaviors such as walking and breathing. CPGs are capable of oscillating and providing a patterned output either with or without external input. CPGs coordinate complex muscle activations to help the animal achieve proper timing to accomplish a given task. They have been found to be involved in a large variety of movement behaviors including the leech heartbeat (Arbas and Calabrese, 1987), human breathing and gasping (Tryba et al., 2006), lobster digestion (Meyrand et al., 1994), turtle scratching (Mortin and Stein, 1989), and locomotion in stick insects (Bässler and Büschges, 1998), lamprey (Cohen et al., 1992), cats (Brown, 1914), and mice (Hägglund et al., 2013).

Modeling of these circuits show that CPGs coordinate multiple segments into predictable patterns during locomotion through entrainment of the CPG to the mechanical systems they control (Iwasaki and Zheng, 2006; Markin et al., 2010). For example, a set of CPGs that are coupled similarly to that of a lamprey have been shown to produce a traveling wave along the body that provides forward locomotion (Ekeberg, 1993). It was shown that this wave can be easily modified by sensory feedback to allow the model to adapt to its surroundings and produce more robust waves for both water and land (Ekeberg and Grillner, 1999; Ijspeert et al., 1999; Bicanski et al., 2013). Similar evidence has shown that sensory feedback can be used to coordinate multiple CPGs in leech swimming and stick insect, cricket, and cockroache walking without direct coupling of the CPGs (Bässler and Büschges, 1998; Ekeberg et al., 2004; Akay and Büschges, 2006; Chen et al., 2011; Szczecinski et al., 2014).

Less is known about the organization of CPGs in mammals than in insects and other invertebrates. Early theories hypothesized the existence of a single CPG per leg, driving transitions between stance and swing (Brown, 1914). However, more recent models utilize multiple oscillating circuits at multiple hierarchical levels (McCrea and Rybak, 2008) supported by recent neurological data (Zhong et al., 2012). Mammalian CPG systems may look more similar to those in insects than previously hypothesized (Büschges and Borgmann, 2013). Models of CPGs coordinated through sensory feedback pathways have been shown to successfully replicate many behaviors in mammalian systems and produce coordinate motion for multiple joints (Ekeberg and Pearson, 2005; Amrollah and Henaff, 2010; Markin et al., 2010; Hunt et al., 2014, 2015a; Li et al., 2016). However, these models have not been tested on a robot, and it is difficult to determine whether they are true in real world physics or possibly exploiting the simplified physics of a simulation.

These advances in understanding of the neuro-mechanical control of locomotion have led to an increase in bio-inspired robots (see Ijspeert, 2008, 2014; Iada and Ijspeert, 2016 for recent reviews) with simultaneous goals of building more advanced and adaptable robots in addition to developing a better understanding of the theories produced from the experimental work. Modern biologically inspired walking robots fall into one of two categories: abstracted biologically-inspired or biology-first. Several abstracted biologically-inspired approaches have effectively demonstrated many principles of animal locomotion. Hopf oscillator-driven robots such as Amphibot and Salamandra Robotica II provide valuable insights into how changing sensory feedback can be used to adapt CPGs and produce rhythmic movement entrained to the mechanics of the robot and its surrounding (Crespi et al., 2005, 2013). AMOS and HECTOR are two robots which are built around machine learning of specific tasks. AMOS is controlled by a large recurrent neural network trained by reservoir computing methods to estimate the leg's state and anticipate future sensory information (Dasgupta et al., 2015). HECTOR uses many feedforward artificial neural networks to map between different states, such as mapping joint angles to the height of a leg (Schilling et al., 2013a). Both these robots are also able to effectively integrate sensory information to produce adaptive, rhythmic output. Additionally, several robots have been controlled with dynamic spiking neural networks (Rostro-Gonzalez et al., 2015; Espinal et al., 2016). All these robots produce adaptive locomotion over diverse terrain, but their controllers abstract many principles of animals' nervous systems, limiting their applications as neurobiological research tools.

Other robots use a biology-first approach to controller design. Biology-first approaches begin with known connectivity from the animal, and set parameter values in the control networks to match data from the animal. RoboLobster, Bill-Ant and

LegConNet control walking with finite state controllers based on previous state-based models of locomotion (Ayers, 2004; Lewinger and Quinn, 2010; Rutter et al., 2011). Locomotion direction is changed by modifying local reflexes that cause transitions between the finite states of leg motion. OCTAVIO uses an artificial neural network assembled from modular subnetworks, much like the work we present in this paper (von Twickel et al., 2011). The biped built by Klein and Lewis and Redbot both demonstrated how a spiking neural network can be used to produce locomotion in a biped robot (Klein and Lewis, 2012). These robots have controllers that mimic the logic and structure of the animal's nervous systems, and as such, serve as tools for investigating neurobiological hypotheses, however, all these controllers were developed by hand tuning parameter values, and are limited by the engineer's ability to calibrate the system.

To improve the applicability and performance of these robots, methods are being developed for setting parameter values in these networks. A major component of these methods focus on breaking the problem into several more easily solved subproblems. These subproblems are solved individually, and often in a specific order to build up the complexity of the network. Redbot uses a staged genetic algorithm process to set stepping frequency, gait, and finally joint angle profiles (Russell et al., 2007). This controller, however, does not use sensory feedback, an important component for adaptive locomotion. The controller for MantisBot, and is formulated around steady state activity of the neural system, however, walking has not yet been demonstrated with this robot (Szczecinski et al., 2015). In previous work, we developed a training process which utilizes many of the same tools as MantisBot (Szczecinski et al., 2017) and sets parameter values in a locomotory network for forward locomotion of a rat simulation (Hunt et al., 2015b). In the work presented in this paper, we demonstrate the broader applicability of this process by applying the same procedure to a dog-like robot to generate adaptive, forward walking.

The key contributions of this paper are (1) the testing of a synthetic nervous system for dynamic walking on a hardware model of a dog's rear legs actuated by artificial muscles, and (2) the validation of an automatic, repeatable method for setting parameter values in a synthetic neural system composed of a CPG locomotion network without requiring a mechanical simulation. Additionally, this work demonstrates the validity of using synthetic neural controllers for controlling dynamic robotic locomotion and acts as a launching point for developing more complex controllers for adaptive locomotion.

2. METHODS

2.1. Robot Architecture

Puppy (**Figure 1**) is a four legged robot with 12 planar joint degrees of freedom (three per leg), first introduced in Aschenbeck et al. (2006). It is 57.5 cm tall, 60 cm long, 23 cm wide, and weighs 6.8 kg (15 lbs). Each joint has an antagonistic pair of 10 mm Festo MXAM-10-AA (Festo Inc.) actuators, also known as "fluidic artificial muscles," that are energized by compressed air. Motion is constrained to the sagittal plane by two plastic sheets (see **Figure 1**). A 2.3 kg (5 lb) counterweight was hung through a pulley on a linear slider and attached to the center of the robot, partially supporting the robot's weight for the trials presented in this manuscript. The robot's hind legs walked on the treadmill. The front legs were suspended above the belt to prevent interference.

Each actuator has separate input and exhaust valves controlled by a single board real-time, reconfigurable input output module, sbRIO-9602 (National Instruments), with an embedded field programmable gate array (FPGA). The sbRIO was connected via a 10/100 Ethernet port to a host computer running Windows 7 on an Intel i7-2770K. Each actuator is connected in parallel to a Freescale MPX5700 GP gauge pressure sensor. Joint angles are collected from a Vishay Spectrol 140-0-0-103 potentiometer placed at each joint. Analog data from the joints and pressure sensors is converted to digital data for the sbRIO with a custom board developed by Osmisys, Inc. Velocity data, calculated by differentiating length data, was filtered by a 2nd order lowpass Butterworth filter with a normalized cutoff frequency of 0.01 Hz, applied after differentiation.

The overall control architecture is illustrated in **Figure 2**. The neural control system is simulated using Animatlab (Cofer et al., 2010). The neural controller outputs motor neuron activations for each of the muscles and receives muscle afferent feedback values via virtual serial ports with Labview. Labview uses the motor neuron values to calculate desired muscle force output and then calculates the pressure required to produce that force. Desired muscle force is calculated by adapting the Hill muscle model (Hill, 1970) (**Figure 3**) parameter values to the artificial muscle where tension, T, is developed in the muscle according to:

$$\frac{dT}{dt} = \frac{k_{se}}{b}\left(k_{pe}x + b\dot{x} - \left(1 + \frac{k_{pe}}{k_{se}}\right) \cdot T + A\right), \tag{1}$$

where x is the muscle length, k_{se} and k_{pe} are the series and parallel stiffness, and b is the viscous damping constant. A is the activation level of the muscle,

$$A = A_m * A_l. \tag{2}$$

A_m is the sigmoid adapter equation,

$$A_m = \frac{F_{max}}{1 + \exp(\mathrm{C}(V_o - V)) + \mathrm{B}}. \tag{3}$$

F_{max} is the maximum muscle force, C is the maximum slope of the sigmoid, V is the membrane voltage of the motor neuron, and V_o and B describe the voltage and force offsets of the sigmoid. A_l is the length-tension relationship,

$$A_l = 1 - \frac{(l - l_{rest})^2}{l_{width}^2}, \tag{4}$$

where l_{rest} is the length at which the muscle can provide the most force and l_{width} is the length from l_{rest} at which the muscle can provide no force.

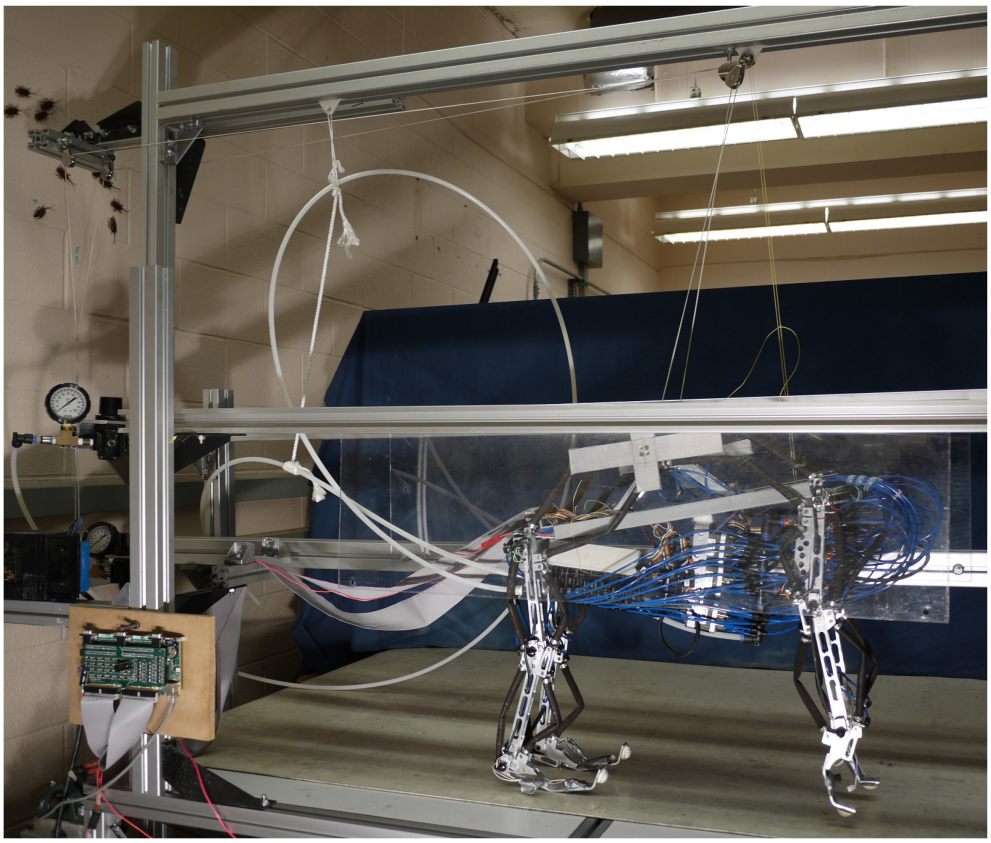


FIGURE 1 | The robot is constrained to motion in the sagital plane and a counterweight pulley system is used to reduce the effective weight of the robot and encourage a center position on the belt.

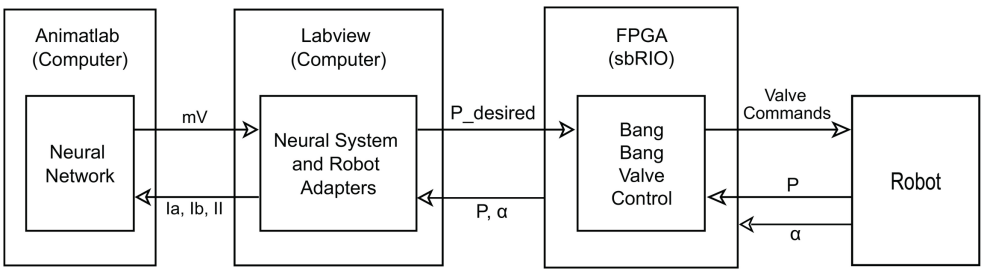


FIGURE 2 | Diagram of control layout. The neural system is simulated on the computer in Animatlab (Cofer et al., 2010). The neural controller uses muscle afferent feedback (Ia, Ib, and II) and internal neural dynamics to output motor neuron activations in mV for each of the muscles via virtual serial ports with Labview. Labview uses the motor neuron values and current readings of joint angles to calculate desired pressure values and passes these to the FPGA. It also uses the current pressure and joint angles to calculate the muscle afferent feedback and passes this to Animatlab. The FPGA uses the current pressure and desired pressure to perform bang-bang valve control on the actuators. It also passes the current pressure and angle readings from the robot to Labview.

The series spring element, k_{se}, simulates the tendon and is very stiff ($10^7 N/m$). k_{pe} is calculated such that all stretching under the maximum expected load is absorbed by the parallel and series elements,

$$k_{pe} = \frac{k_{se} \cdot F_{max}}{k_{se}(l_{max} - l_{min}) - F_{max}}. \quad (5)$$

To develop the length-tension relationship, the maximum output force was set to 509 N (based on extrapolation of the actuator fit curve found in Hunt, 2015 at 90 psi). Length parameter values were unique for each muscle and set such that l_{rest} was equal to the length with no pressure and no load, and the l_{width} was set such that $A_l = 0$ when the muscle was at its shortest length with no load under 90 psi. The peak velocity of the muscles (v_{max}) was calculated from empirical testing and used

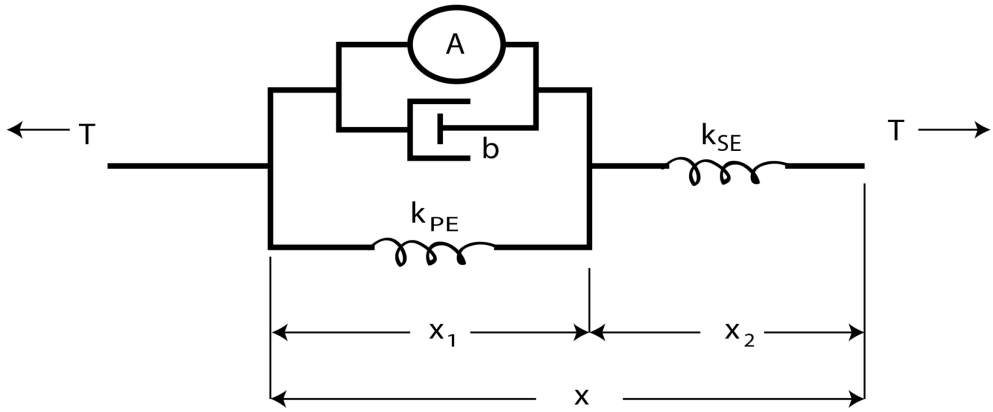


FIGURE 3 | The Hill muscle model (Hill, 1970) is modeled as an active contractile element in parallel with a damper and spring element. These components are attached in series with a stiffer spring element approximating the tendon.

to set b such that $b = F_{max}/v_{max}$. The values $V_o = -50\,mV$, $C = 121.46$, and $B = -1.17$ are found by solving Equation (3) for the conditions: $A_m(-100\,mV) = 0$; $A_m(-10\,mV) = 0.99 * F_{max}$; and $A_m(-50\,mV) = 0.5 * F_{max}$.

The commanded pressure values are calculated from the empirical model of the actuators derived in Hunt (2015). In this model, the commanded tension from Animatlab and current geometry of the robot are used to calculate the commanded pressure for each of the artificial muscles with the equation

$$P = 254\,\text{kPA} + 1.23\,\frac{\text{kPA}}{\text{mN}} \cdot T + 15.6\,\text{kPA} \cdot S + 192\,\text{kPA} \cdot \tan\left(2.03\left(\frac{k}{-0.33\,\frac{1}{mN} \cdot F + \max(k)} - 0.46\right)\right), \quad (6)$$

where S is the state of the artificial muscle in which 1 indicates the muscle is shortening and -1 indicates lengthening. For stability, this value was changed from the binary values calculated originally to continuous linearly scaled values based on the maximum velocity of the muscle. This commanded pressure is sent to the FPGA. Because of limited bandwidth, the valve controller on the FPGA opens the inlet or exhaust valve until the actual pressure reading is within ±15 kPa of the commanded pressure, and then closes the valve.

The sbRIO collects joint angle data and muscle pressure data and passes this information to the Labview computer program for processing. Labview converts the joint angle data to muscle lengths such that

$$l_m = \mathrm{a}_m + \mathrm{b}_m \cos\left(\alpha_m + \theta_m\right). \quad (7)$$

a_m, b_m, and θ_m are unique constants based on the specific geometry of the robot and α_m is the joint angle. Muscle force is then calculated from pressure and length using a lookup table built on Equation (6). Types Ia, Ib, and II muscle afferents are calculated for the neural control system. Though this feedback is simplified, it captures the main function of each type,

$$\mathrm{Ia} = k_a\dot{x} \qquad \mathrm{Ib} = k_b T \qquad \mathrm{II} = k_c x. \quad (8)$$

where k_a, k_b, and k_c are gain parameters whose values are set such that the injected current is 20 nA when the muscle is at its maximum velocity, tension, and length, respectively.

2.2. Neural Network Architecture

The neurons in the control network have leaky integrator dynamics. The leaky integrator model captures the most basic behavior of neurons and allows for more complex dynamics to be added without increasing the complexity of the rest of the network. It is capable of modeling individual non-spiking interneurons, the firing rate of a population of neurons, or a single spiking neuron after a spiking threshold is included. This work is not concerned with the specifics of how action potentials are generated and has neglected Hodgkin-Huxley sodium and potassium currents. In this work, each neuron is used to model the average firing rate of a population of spiking neurons. The dynamical equations that describe their behavior can be found in Szczecinski et al. (2017).

2.2.1. Joint Control

The connectivity of the Zhong locomotor model (Zhong et al., 2012) was chosen as the basis for the neural control system for low level control. Since our focus is on understanding how sensory feedback affects the timing and activation of motor neurons, the presented model neglects the highest level CPG, and is simplified to a single network for each joint with a pattern formation layer and lower level afferent feedback networks (**Figure 4**).

Intra-joint sensory feedback controls each joint. Positive force feedback (Prochazka et al., 1997) provides self exciting Ib feedback to each muscle. As tension within a muscle increases, the motor neuron is excited further to apply even more tension. Though this leads to a destabilizing influence in most control systems, the length-tension properties of the muscles and geometric alignment of the musculoskeletal system prevent unstable behavior. This influence helps the animal compensate for unexpected increased loads during walking. Cross inhibitory velocity feedback through Ia pathways limits muscle speed

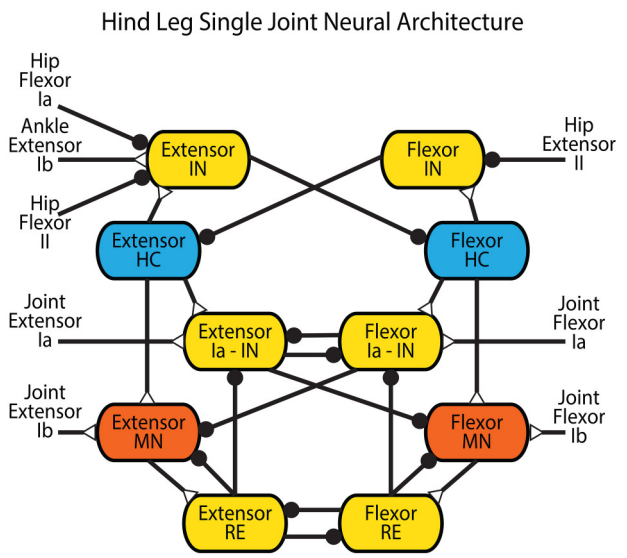


FIGURE 4 | Network architecture for a single joint in the hind leg adapted from Zhong et al. (2012). Blue neurons are CPG half-centers (HC) with additional sodium currents. Red neurons are motor neurons (MN) used to provide activation to the muscles in the rat simulation or the actuators in the robot. Yellow neurons are interneurons (IN) and Renshaw cells (RE). Feedback from the entire leg is applied directly to the CPG of each joint in the form of hip flexor Ia and II, hip extensor II, and ankle extensor Ib feedback. Extensor and flexor Ia and Ib feedback from each joint feed directly back onto the joint control through Ia interneurons or directly onto the motor neuron. Synapses that terminate in a close circle indicate an inhibiting synapse while those that terminate in an open triangle indicate an excitatory synapse.

(McCrea et al., 1980; Lundberg, 1981; Pratt and Jordan, 1987; Jankowska, 1992; Geertsen et al., 2011). When a muscle is stretched quickly, it inhibits the antagonist via the Flexor or Extensor Ia - IN interneuron.

2.2.2. Leg Control

Intra-leg sensory feedback connections are derived from proposed coordination mechanisms in mammalian literature. Stance-to-swing transition is the most studied phenomenon, and is caused both by reduced firing in Ib Golgi tendon afferents and increased firing from hip flexor stretching (Pearson, 2008). This integration of signals is shown in **Figure 4** as inhibitory connections from the "Hip Flexor Ia" and "Hip Flexor II" afferent feedbacks and an excitatory connection from the "Ankle Extensor Ib" afferent feedback onto the "Extensor Interneuron" for each joint. Stance is initiated by reduced firing of the hip flexor type II afferent or increased firing of hip extensor type II afferent (McVea et al., 2005; Akay et al., 2014). This indicates that the hip is forward, causing contraction of the hip and ankle extensors. This is realized as an inhibitory connection from the "Hip Extensor II" afferent feedback onto the "Flexor Interneuron" for each joint.

2.2.3. Inter-Leg Control

Commissural interneurons encourage an alternating gait between the legs. These connections mimic those that have been found in mice (Talpalar et al., 2013) and cats (Jankowska, 2008), and further described with neural modeling (Rybak et al., 2013). In these models, the interneuronal connections are between high level leg CPGs, which are not included in our model. Because we have a CPG for each joint, our commissural interneurons are made to act on the most proximal joint, which drives the protraction and retraction of the leg. The hip joint CPGs are connected with inhibitory and excitatory commissural internerons (CINi and CINe), and the rest of the CPGs remain unconnected. These pathways are set such that the CINi pathways provide three times as much inhibition as the CINe provides excitation, similar to related models (Rybak et al., 2013) and more than an order of magnitude weaker than other synapses within the model. These connections are illustrated in **Figure 5**. Parameter values were used from our previous work Hunt et al. (2015a).

2.3. Calculating MN Activations

The motor neurons are the interface between the neural and mechanical systems. The motion of the robot and the dynamics of the actuators dictate the motor neuron activations during locomotion, which the neural system must be tuned to produce. This section describes how we calculate the motor neuron activations.

2.3.1. Joint Torques and Kinematic Motions

To determine kinematic and dynamic motions for the robot, models of the hind and fore legs during stance and swing were developed in Simulink-SimMechanics (Mathworks, Inc.). A cubic spline was fit to predetermined angles and duty cycles for touchdown, midstance, liftoff, and midswing based on walking whippets, a species of dog with similar limb proportions and body mass to Puppy (Fischer and Lilje, 2011). The data for the walking kinematics was averaged from 7 dogs with an average stepping period of 0.54 s, and a speed of 1.01 m/s, or 1.97 body lengths/s.

Swing torques were calculated by adding friction to the joints and doing a forward dynamic analysis using the equations of motion. The calculation of stance torques was done by building a closed chain system with a fore and hind leg on the ground at one time. A proportional-derivative (PD) controller at each joint was used to produce a kinematic trajectory similar to that collected from whippets (Fischer and Lilje, 2011). The PD controller torques are the torques required to produce whippet-like locomotion with Puppy. The stance data and swing data were concatenated assuming a 50% duty cycle and smoothed non-linearly to remove discontinuities at the edges (**Figure 6**).

2.3.2. Calculating Muscle Tension and MN Activation

Muscle tensions during locomotion were calculated using the joint torques in the previous section and the active lengths of the muscles during locomotion. A unique solution was obtained by assuming only one muscle per joint is activated at a time (Hooper et al., 2009). The active muscle must produce the previously calculated torques as well as overcome torques created by the passive forces produced in each muscle.

Passive forces were calculated using Equation (1) with $A = 0$. Muscle length (x) and muscle velocity ($\dot{x}$) were calculated using

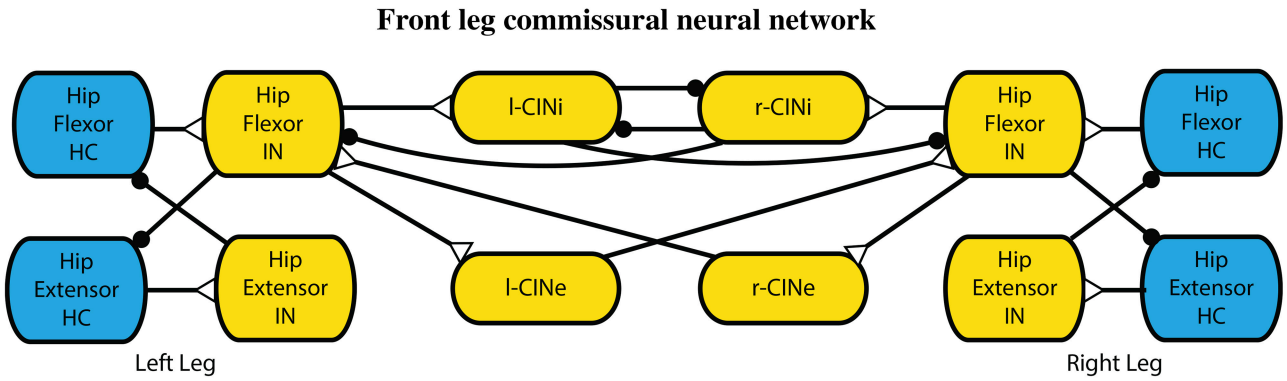


FIGURE 5 | Interleg commissural interneronal network for coordinating legs into an alternating gait. While one leg is in hip extension, it provides limited inhibition to the other leg extensors causing it to stay in the flexion state longer. Synapses that terminate in a close circle indicate an inhibiting synapse while those that terminate in an open triangle indicate an excitatory synapse.

a forward kinematic model of the Festo attachment points and joint kinematics. The derivative was discretized and T was solved for at the next time step based on the previous tension,

$$T_{i+1} = T_i + \Delta t \cdot \frac{k_{se}}{c}\left(k_{pe}x_i + c\dot{x}_i - \left(1 + \frac{k_{pe}}{k_{se}}\right) \cdot T_i\right). \quad (9)$$

Starting with $T = 0$ and repeating this process for several step cycles produces a periodic steady-state tension profile that resists the ground-force and dynamic torques. The active muscle, then, must overcome this passive muscle force, the ground force, and dynamic forces. The active muscle force is calculated by using a bisection root-finder to balance the static and dynamic forces acting on each joint for each time step. The motor neuron activation is calculated by solving Equations (2) and (9) with a bisection root-finder.

2.4. Training CPG Network Output

Training the CPG network output is performed with the same four step process as is presented in (Hunt et al., 2015b) for the simulation of a walking rat. This process is similar to the staged evolution technique used to evolve parameters for Redbot locomotion and other systems (Inada and Ishii, 2004; Russell et al., 2007). A review of the process is below.

Each leg network (which includes three joints) consists of 82 neurons with 12 parameters each, and 134 synapse connections with 4 parameters each. The large number of parameters is a result of the complexity of the biologically-based model that we use to control each joint (see **Figure 4**) (Zhong et al., 2012). Many parameter values were set using basic heuristics such as resting voltage (−60 mV), time constant (5 ms), and relative size (1). Even after these simplifications, approximately 90 parameters per leg, mostly synapse strengths, still needed to be set. Because of the large number of possible local solutions, the design and training of the CPG network was done over the course of four iterations in which progressively more complete networks were tuned. First, parameter values within the CPG were tuned to generate appropriate rhythm and response properties. Second, synapses from sensory neurons to the CPG were tuned to generate the intended CPG activity during walking. Third, synapses from the CPG to the MNs were tuned to obtain the proper MN activation. Finally, afferent feedback from the muscles to the MNs was tuned to further refine MN activation. This entire tuning process was performed *without* a physics-based simulation and then the results were tested on the Puppy robot.

2.4.1. CPG Design

The first step is designing a CPG for the pattern formation layer of a single joint which is capable of producing the desired phase transitions in response to sensory feedback. The system is composed of two mutually inhibitory neurons called half-centers (HCs), each with persistent sodium channels. It has the same basic set of equations as has been used in other recent modeling work (Daun-Gruhn et al., 2009). These channels provide nonlinear positive reinforcement to membrane voltage fluctuations, which make sustained oscillation possible. Mutual inhibition is implemented via non-spiking interneurons (INs). Each HC excites an IN, which inhibits the other HC, as shown in **Figure 4**. Though this CPG is composed of only 4 non-spiking neurons, it exhibits many of the same shapes, behaviors, and responses to perturbations that exist in the average spiking frequency of reciprocally inhibited spiking neurons with postinhibitory rebound (Perkel and Mulloney, 1974; Pinsker, 1977; Ayers and Selverston, 1979). It also has the same network architecture as the pattern formation neural pools used in the Zhong locomotor model, and the oscillatory dynamics are also governed by a slowly activating and deactivating persistent sodium current.

Our previous work described a bifurcation parameter, δ, which controls the CPG's endogenous rhythm and sensitivity to inputs (Szczecinski et al., 2017). The CPG oscillates endogenously if $\delta > 0$. When δ is near to 0, it easily entrains with incoming sensory signals. As δ increases, it less easily entrains

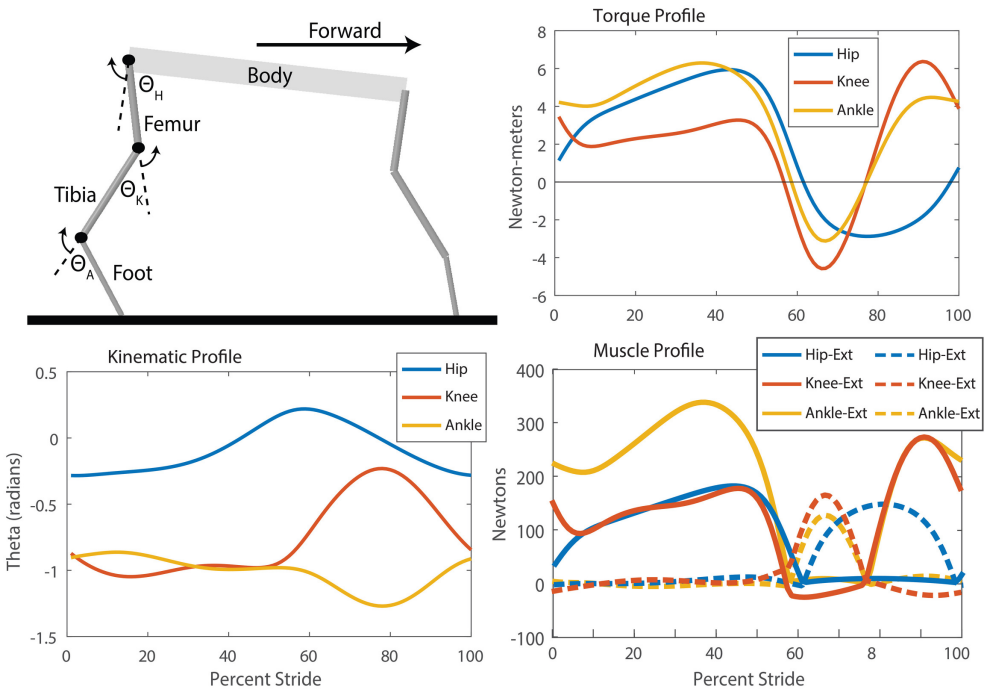


FIGURE 6 | SimMechanics model, kinematic, torque, and muscle tension profiles for the puppy robot which were used to produce training data for the neural system. Extension refers to a positive angle movement and flexion refers to a negative angle movement. Though the knee and ankle joints have a period of little movement during stance, analysis of the torque profile and calculation of the muscle forces show the high torques and tensions required to maintain those angles. Distinct periods of alternating extension and flexion muscle activation are observable for each muscle through a step cycle. Additionally, the hip flexor remains active for longer than the knee and ankle, indicating that the step cycle has more modes than "stance" and "swing."

with sensory signals. Each joint of Puppy is controlled by a CPG in which $\delta = 0.1$. In addition, the slope of m_∞, h_∞, and G_{Na} were adjusted until the CPG's bursts peaked approximately 20% above the high equilibrium point, and the endogenous period was twice that of the robot's intended stepping period.

2.4.2. CPG Entrainment

The second step in choosing parameter values for the network to produce the intended MN activations is to tune the synapses from sensory neurons to the CPG, such that the CPG both entrains to the sensory information and produces the MN activations calculated in the previous section. In our network, sensory feedback synapses onto the CPG according to rules discovered in vertebrates, described in Section 2.2 (e.g., hip flexor stretch encourages a transition from stance to swing Pearson, 2008, etc.). The synaptic conductance and threshold of these pathways determine how they affect the CPG's phase (Szczecinski et al., 2017), meaning that they must be carefully calculated for Puppy to walk properly.

Two steps are required to tune the synapses from sensory neurons to the CPGs. First, the intended walking kinematics are used to find the type Ia, Ib, and II afferents during normal walking motion. These are the signals that entrain the CPG into the proper phase for walking. Second, a neural simulation is assembled in which the calculated muscle afferents are input to the CPG. A fitness function, $f_1(V_{thresh}, G_{syn})$, is calculated that describes how well the CPG entrained to the sensory information,

$$f_1(V_{thresh}, G_{syn}) = (P-P_o)^2+(Se-Se_o)^2+(Sf-Sf_o)^2+\sum(G_{syn}), \tag{10}$$

where P is the oscillation period, S_e is the timing of the extensor MN's rising edge, S_f is the timing of the flexor MN's rising edge, and G_{syn} is a vector of conductance values for the synapses under consideration. V_{thresh} is a vector of the conductance threshold for the same synapses. Terms with the subscript "o" are the intended values. Note that synaptic conductances are penalized, preventing synapse conductances from becoming too large.

G_{syn} and V_{thresh} were found to minimize f_1 with a two-step optimization process. First, a genetic algorithm (GA) was used as a global search of the parameter space. The GA was initialized with a population of 1,500 possible parameter value combinations. At the end of every generation, the worst 50% of solutions were eliminated, and the others were randomly selected for mating with a performance-based weighting. Mating was performed with single-crossover, and the mutation rate was 0.1%. Once the GA completed five generations, the best solution was used as the starting point for a Nelder-Mead simplex minimizer. Thus, the parameter space was first globally sampled, and then serially refined to find a promising solution.

2.4.3. CPG Output

In the third step, the CPG output synapse strength was trained to produce activations of the motor neurons with a peak magnitude that matches each desired motor neuron activation and a minimum of no activation at some point in the cycle. Similar to entraining the CPGs, we used the GA from the previous section with a population of 300 parameter value combinations for 5 generations and refined the best solution with a Nelder-Mead routine. The fitness function is

$$f_2(x) = (\max(E) - \max(E_o))^2 + (\max(F) - \max(F_o))^2 + \min(E) + \min(F), \quad (11)$$

where E and F are the a single cycle of extensor and flexor motor neuron patterns and E_o and F_o are the desired patterns.

2.4.4. Afferent Influence of MN Activation

In the last step, afferent feedback was trained to help shape the MN output and provide additional force if necessary to overcome changes in foot placement (excitatory Ib feedback), or reduced force if the leg is moving too quickly (inhibitory Ia feedback). All neurons and pathways involved in these networks were designed to be completely continuous over all possible ranges. The fitness function for the final training is

$$f_3(x) = (\max(E) - \max(E_o))^2 + (\max(F) - \max(F_o))^2 + \min(E) + \min(F) + (E - E_o)^2 + (F - F_o)^2. \quad (12)$$

3. RESULTS

3.1. Offline Training Results

The final results of the training can be seen in **Figure 7**. A clear relationship between the training data and the network output is observed. A step cycle with the desired period is produced based on expected sensory feedback. All muscles are active at the correct point of the step cycle, with extensors active during stance and flexors active during swing. The transitions between the stance and swing phases are close to the desired transition point of the step cycle based on expected sensory feedback. Additionally, five of the six activation curves follow within 10% of the magnitudes for the inverse dynamics calculated activation values.

For the hip, extensor output at the beginning of stance and flexor output at the beginning of swing are both a little high, but final output is within 5% of the training curve. The transition from stance to swing in the hip occurs 10% earlier than the training data anticipates; however, this is a phenomenon observed in kinematic data for dogs (Fischer and Lilje, 2011) and other mammals (Fischer et al., 2002). Additionally, knee extensor output is initially within a few percent of the desired angle, and it maintains much higher output during stance than the training data. The knee flexor output peaks at a higher magnitude than the training data, however, this is not for long. The transition timings from stance to swing and swing back to stance are directly in time with the expected feedback and training data. For the ankle, both trained ankle output for extensor and flexor activity follow the training data shape and are within a few percent of the desired output. Here, like the knee, the transitions from stance to swing and back to stance are directly timed with the expected sensory feedback and training data.

3.2. Robot Results

The trained network output MN activity based on expected sensory feedback is nearly as expected and results in robot walking. With the applied trained network and the commissural inter-leg network, the hind legs perform sustained, alternating stepping at a period of 0.83 s. The walking robot had approximately a 50–50% stance to swing duty cycle. Data presented in **Figures 8–11** is for a stepping speed of 1 m/s or 1.67 body lengths/s. A screen capture of a step sequence is shown in **Figure 8** (See Supplementary Material Video 1).

The average MN activations, muscle tensions, and joint kinematics for 38 right and left steps can be seen in **Figure 9**. Average extensor MN activations have peaks that are within 10% of intended magnitude, while flexor activity peaks are lower. Relative timing between the joints is as expected, with hip, knee, and ankle flexors transitioning to swing at about the same time, and knee and ankle extensors activating mid-swing before the hip extensors at the beginning of stance. When comparing averaged activity, overall activity is more spread out than desired activations, however, activity during single steps show sharp transitions and distinct off periods as can be seen in **Figure 10**.

Sensory signals produce adaptive motions by changing step timing. The transition from swing to stance occurs with increasing Hip Extensor II feedback (**Figure 10**, column one, solid arrow). The transition from stance to swing occurs with increasing Hip Flexor II feedback and a drop in Ankle Extensor Ib feedback (**Figure 10**, column one, dashed arrow). These sensory changes cause the CPGs to rapidly change phase between extension and flexion. The CPG change produces a corresponding rapid change in MN voltage and change in motion. These transitions vary in timing depending on the voltage values and rate of change for sensory feedback neurons (**Figure 10**, column 2).

Afferent feedback also provides shaping of MN activation activity. During walking, the contribution to MN output from the CPGs drop over time due to the decreased level in activity of the CPG neurons. However, the desired MN activation at the end of swing and stance increases over time for the hip muscles (**Figure 7**). The synthetic neural controller achieves this with local hip extensor and flexor Ib excitatory feedback pathways as is seen in row three of **Figure 10**. This activation is even more pronounced in the robot than was calculated with inverse dynamics or predicted by the offline training and neural simulation.

Comparisons between the right and left leg show activations and joint angles with similar shapes and peak amplitudes within a few percent of each other, except with a small phase delay (**Figure 11**).

4. DISCUSSION

The robotic system demonstrated here shows the sufficiency of the known neural system for timing joints and producing the necessary kinematic motions. Our work reaffirms the work

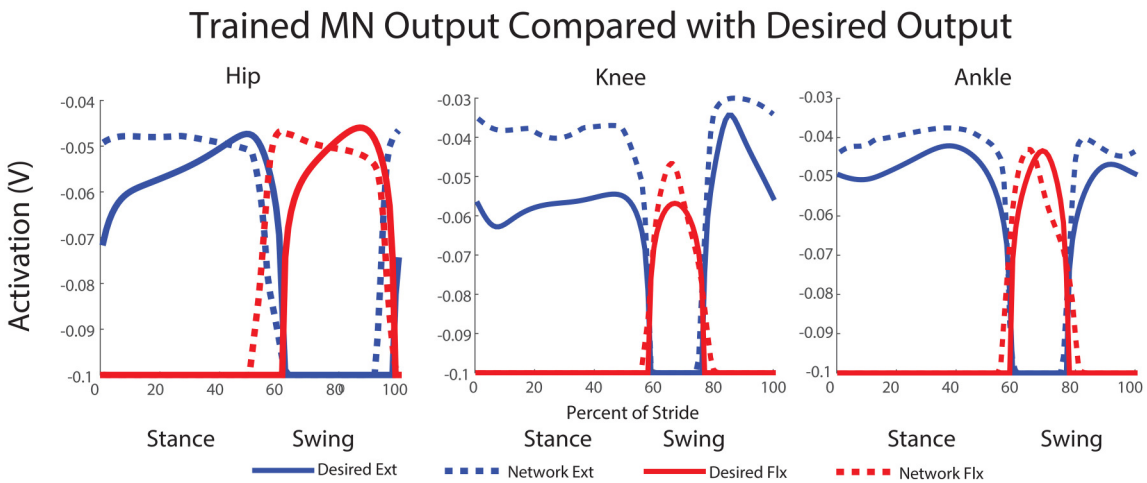


FIGURE 7 | Trained network output MN activity compared with desired motor neuron activations. This output is simulated using expected feedback and is not the actual MN output of the walking robot. The transitions between stance and swing phases are close to the desired transition point of the step cycle and most activation curves follow within 10% of the magnitudes for the activation values calculated with inverse dynamics.

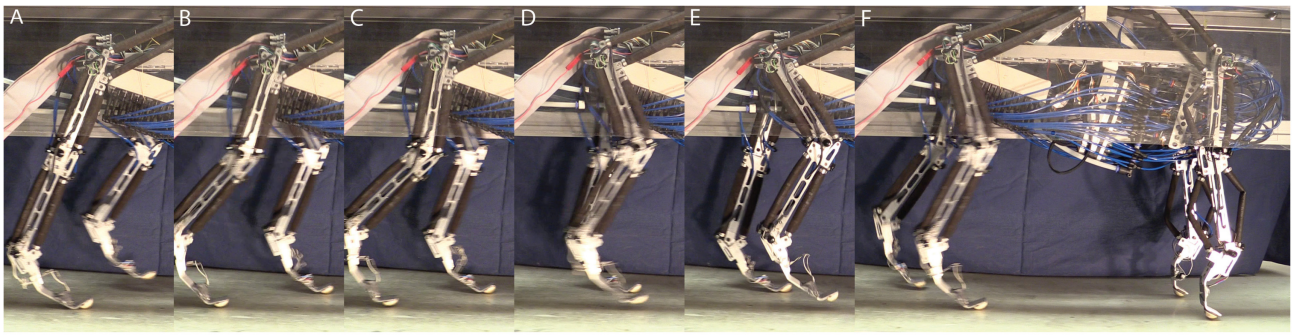


FIGURE 8 | Series of screen shots demonstrating hind leg walking in Puppy. (A) Left leg is in swing and the right leg is in stance. **(B)** Double support phase. **(C)** As the left leg takes up more weight, and the right leg moves further back, it begins to enter swing. **(D)** While the right leg swings, the left leg bears the weight of the robot. **(E)** When the forward swing position is reached, the right leg begins extension. **(F)** When the right leg touches down and begins to bear load the left leg enters swing and the process is repeated.

by Klein and Lewis (2012) that dynamic neural systems are affective tools for controlling dynamic walking systems. Our work expands upon this by implementing a more detailed model of intra-leg sensory pathways and demonstrates that the proposed mechanisms are effective for regulating stance and swing timing, as well as muscle force production for forward walking by adapting each step individually. Additionally, our work demonstrates a network controller that can produce locomotion at faster speeds and with less external support than this previous work.

Our work also demonstrates the larger applicability of the parameter value setting method first presented in Hunt et al. (2015b). This method was first developed for setting locomotion parameter values in a simulation model of a rat actuated by a Hill muscle model. Compared with the dog robot, the rat simulation has a different kinematic configuration, different stepping frequency, different actuators, and different torque demands. Despite all these differences, the same method is effective for setting parameter values in the rat simulation and the dog robot.

The method for setting parameter values in the stepping network presented here significantly reduces time to application in two ways. First, by having an autonomous method for setting parameter values, the computer is able to remove the guesswork involved and evaluate possible parameter values at a much faster speed than a human. Second, by eliminating the need for physics-based simulations or hardware, the method is able to iterate through possible parameter value choices several orders of magnitude faster than with a simulation or hardware in the loop. This methods works by evaluating the network with expected sensory feedback, assuming locomotion speed, kinematics, and forces are occurring as designed.

Despite differences in sensory signals that occur when the robot actually walks vs. those that were predicted, the simulated

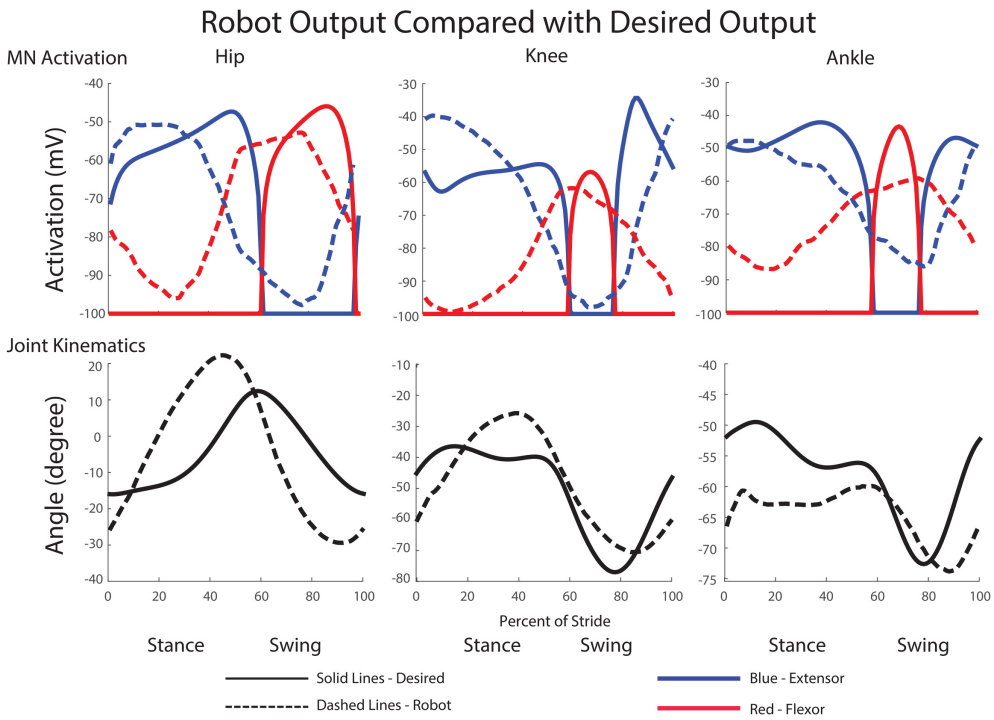


FIGURE 9 | Comparison of desired motor neuron activations and joint kinematics with those produced during walking motions in the robot. Depicted data is the average of 38 steps at approximately 1 m/s. Motorneuron activation magnitudes and timing match closely with the desired magnitudes. Additionally, joint excursion is close to the desired maximum flexion and extension values.

neural system maintains effective control of locomotion. We believe this is the case because of the robust design of the locomotory circuit combined with the stable design of the legs. The central pattern generator ensures that stepping remains continuous despite deviations in sensory signals. Additionally, the sensory feedback pathways are able to adapt the locomotion steps and maintain stability while there are variations in stepping behavior. This confirms the effectiveness of the neural organization and different sensory signals and pathways implemented in our neural model for rhythm generation (Zhong et al., 2012), joint coordination (McVea et al., 2005; Akay et al., 2006; Pearson, 2008; Akay et al., 2014), leg coordination (Jankowska, 2008; Rybak et al., 2013; Talpalar et al., 2013), and motor neuron activity regulation (Jankowska, 1992; Prochazka et al., 1997; Zhong et al., 2012). The Ib and Ia feedback pathways that modulate motor neuron output add significant control to the robot. Positive Ib feedback adds additional MN activation when load is encountered on a muscle, enabling it to pull harder to overcome obstacles. In terms of walking, this means pushing harder on the ground if the stance leg is in a position where the muscles have low mechanical advantage. Negative Ia feedback reduces MN activation when the joint is moving too quickly, slowing down stance or swing.

This work also demonstrates a method for determining the required motor neuron activations from desired kinematics and a model of the robot. Though these torques were within 20% of peak torques recorded in the greyhound (another dog of similar limb proportions and body mass to Puppy) (Colborne et al., 2006), the method required the implementation of a PD controller, which can be very sensitive to parameter values. Recent advances in the fields of biology and biomechanics have led to more sophisticated methods for calculating joint torques using both kinematic and dynamic (force) data from the animal itself, leading to interesting implementations of biorobotic systems (Andrada et al., 2013; Karakasiliotis et al., 2016). As this data becomes available for dogs, we can use it to refine the required joint torque output of the robot similar to what we did in the simulation of rat locomotion (Hunt et al., 2015b). However, when this data is not available, e.g., it either has not be collected yet for a particular animal or when a robot has a unique morphology, our work demonstrates the effectiveness of this approach non-the-less.

4.1. Possible Causes of Error

Though there are some observable differences between the animal data and robot motion and control, the presented controller is a starting point for developing further improvements. For example, the Hip Extensor motorneuron activity has significant additional activation early in stance phase, which is a result of training the CPG output synapse to match the highest desired MN activity. This could be improved through the inclusion of additional pathways and different training methods. The training of the output strength could be based on the highest point of initial MN activity or additional feedback pathways may be required to limit the knee extensor activity during stance.

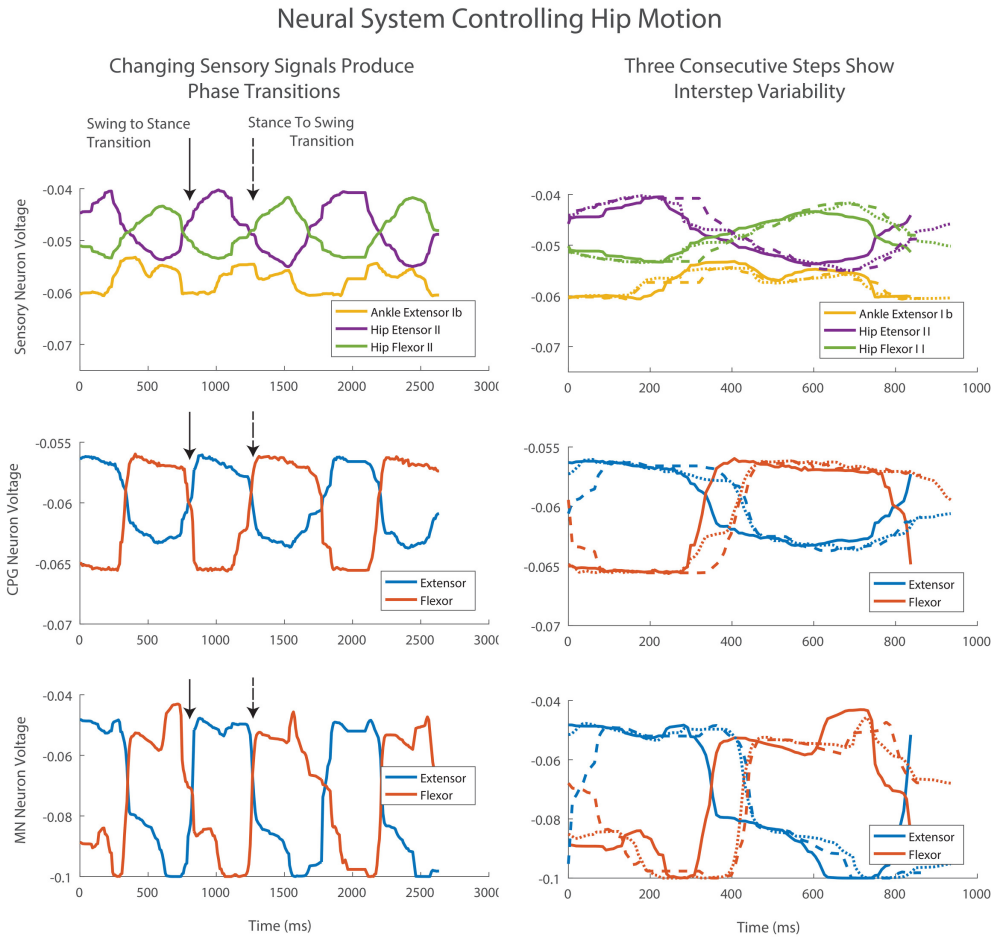


FIGURE 10 | Three consecutive steps of the left hind leg. Column one shows the steps in sequence. The transition from swing to stance occurs with increasing Hip Extensor II feedback (solid arrow). The transition from stance to swing occurs with increasing Hip Flexor II feedback and a drop in Ankle Extensor Ib feedback (dashed arrow). Afferent feedback provides the desired increased MN activation at the end of swing and stance despite a drop in CPG neural activations. This activation is even more pronounced than predicted by the offline training and neural simulation. Column two shows the same three steps beginning at foot touchdown. Differences in sensory signals provide adaptation and changes in the CPG level transition timing as well as MN activity levels.

All joint peak angles are accurate within 5–15°. The largest errors occur with the hip. Errors in hip peaks are possibly due to the delays in communication between Animatlab and Labview and the robot. The hip is the only joint to provide feedback on position, and this delay would impact the sensory signal which causes transitions in the neural system to lag real time of the robot. The response of the robot would then be additionally delayed by the returning communication. There is no such delay built into the training of the neural system. In the future, we could simulate such a delay in our training method, or improve the bandwith between the robot and the neural controller.

Observations of individual step data reveal larger variations occurring on a step by step basis with sharper transitions and higher peak heights in MN activity than is noticeable in the average data. This indicates that the neural system is adapting the stride and adjusting its control continuously. This also shows the adverse effects of working with data that is averaged from multiple steps. Though averaged data shows important information, it does not depict the whole picture where individual variety and adaptation play an important role in locomotion.

Another product of using the averaged data is potentially incomplete training of the sensory feedback in both setting thresholds, and setting strengths of local Ia and Ib feedback parameter values. Though sensory feedback could be modulated by thresholds in the animal, the thresholds were not trained in our work because we used a single feedback signal without noise. While training, the reliance on this expected input caused the system to become overly dependent on exact threshold points, and small changes in feedback strength produced significant effects on behavior. Additionally, there is not enough available data on how intra-joint Ia and Ib pathways affect walking to properly train and set these weights off-line. Intra-joint feedback is likely instrumental in changing force production on a step-by-step basis, and training these pathways using average data may never be sufficient for adaptive, animal-like walking.

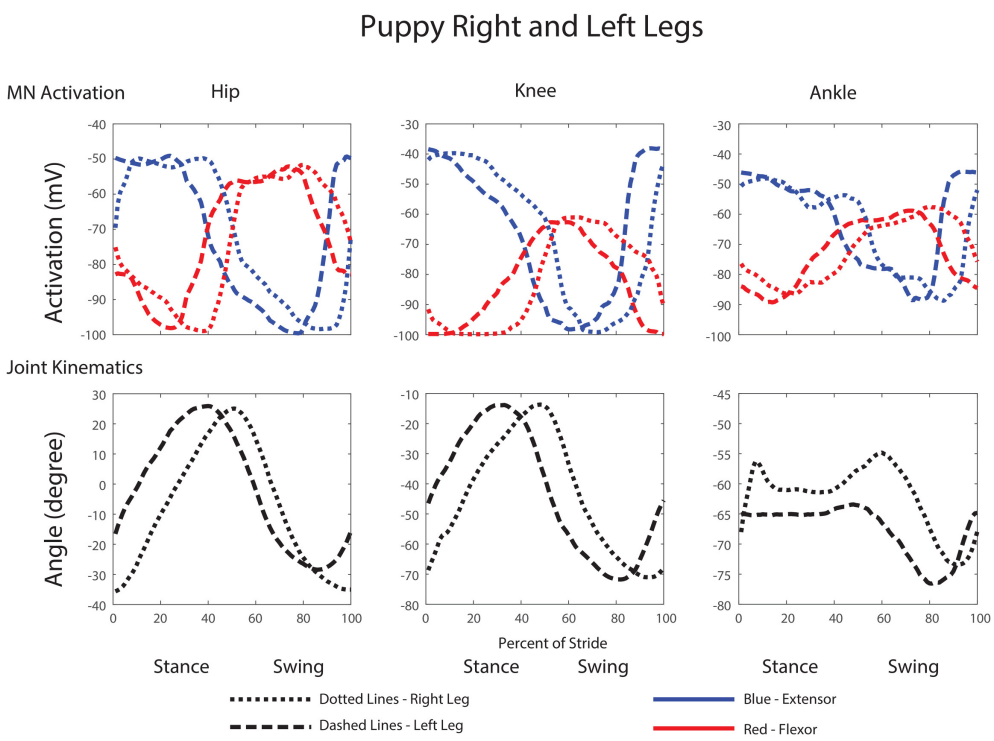


FIGURE 11 | Comparison of motor neuron activations, muscle tensions, and joint kinematics between the left and right legs of the robot.

Puppy's gait was asymmetrical, and one possible explanation for the asymmetry could be differences in ankle motion. It is noted that the left ankle maintains a more flexed position than the right, especially during stance. This difference could be a result of a problem in the robot controller at the low level, turning the MN activations into actuator pressures in an uneven manner. Another explanation is that the controller is such that when a phase delay occurs, it continues to occur based on the overall kinematics and dynamics of the system. This could be determined through more extensive testing of the robot in different initial conditions and determining if the lag always occurs on the same side of the robot.

4.2. Future Work

Future work in controller development will be explored in several areas. First, we will improve our training method in several ways. To do this, we will perform optimization on a physics-based simulation or the walking robot. The neural system could be trained to provide greater stability and/or matching of animal kinematics. This would enable the system to learn low-level feedback pathways that are able to make the subtle corrections necessary for the simulation to produce repetitive, self-supporting walking that more closely matches that of the animal. The second method would require more animal data, using kinematics and dynamics for a series of steps in the training. These series would have different motor neuron profiles for each step, and the optimizer could adjust the feedback pathways to better match the step by step information, and not just the averaged data.

Though the developed controller is able to produce walking with only feedback from muscles, animals take advantage of significantly more sensors while walking. Walking can be made more robust and able to handle more diverse situations such as large perturbations or obstacle avoidance by adding more sensors to the control system. Currently, Puppy is equipped with sensors on the bottom of the feet, which are able to sense ground contact and force in a single direction. Inclusion of these sensors in the walking control system would add redundancy to ground detection and would likely result in more stable behaviors. These could act as ground contact sensors, similar to those used in Klein and Lewis (2012).

We are also in the processes of redesigning the front legs to more accurately reflect the anatomy of the dog (Fischer and Blickhan, 2006). Upon completion of the front legs, we will be able to apply the same training process to produce forward walking in the front legs, and then begin to explore processes which affect inter-leg coordination similar to the work performed in simulation in Hunt et al. (2015a). By working with the physical robot, we will be able to more accurately observe the roles that mechanical interactions play in inter-leg coordination.

This robot and other such biorobots controlled by synthetic nervous systems offer advantages for further researching neural control of locomotion and movement. With our robot, we will be able to test more detailed neurological models of locomotion by replicating experiments which explore how the elimination of different sensory signals can cause specific effects in locomotion. For example, we can adjust the relative strengths of inter-leg

pathways similar to those performed in Talpalar et al. (2013), and observe if similar hopping motions result. Additionally, we can perform experiments which attempt to mimic diseases and their effect on the nervous system. We can then perform experiments in the robot, observe the effects on locomotion, and use the results to inform better models of the disease. We can additionally perform a variety of interventions to overcome deficits caused by the disease without risk to an animal.

5. CONCLUSION

This manuscript presents a robot controlled by a synthetic nervous system built from the known connectivity of mammalian locomotor systems. We demonstrate that the neural controller effectively adapts the robot's stepping on a step-by-step basis and maintains rhythmic walking. This research platform, consisting of the robot, its hardware control system, and its synthetic nervous system, will serve as a useful launching point for studying more complex behaviors as well as the role of different sensory signals in locomotion. The computational method for setting parameter values in a synthetic nervous system based on desired behavior is also presented. This method is significantly faster and more reliable than manual tuning, and has been effective for both a rat simulation and the Puppy robot described here. We believe that the method presented here will prove useful to other researchers attempting to explore the use of neural controllers for other simulated models and robotic systems.

AUTHOR CONTRIBUTIONS

AH: Developed the control system layout and training methods. Performed data collection and analysis. Drafted the manuscript. NS: Performed detailed CPG analysis and developed the software algorithms for performing training. Provided significant revising of the manuscript. RQ: Provided robot support and control development guidance. Provided revisions for important intellectual content.

FUNDING

This work was supported by DARPA M3 Grant DI-MISC-81612A and by the NASA Office of the Chief Technologist, Grant Number NNX12AN24H.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00018/full#supplementary-material

REFERENCES

Akay, T., and Büschges, A. (2006). Load signals assist the generation of movement-dependent reflex reversal in the femur-tibia joint of stick insects. *J. Neurophysiol.* 96, 3532–3537. doi: 10.1152/jn.00625.2006

Akay, T., McVea, D. A., Tachibana, A., and Pearson, K. G. (2006). Coordination of fore and hind leg stepping in cats on a transversely-split treadmill. *Exp. Brain Res.* 175, 211–222. doi: 10.1007/s00221-006-0542-3

Akay, T., Tourtellotte, W. G., Arber, S., and Jessell, T. M. (2014). Degradation of mouse locomotor pattern in the absence of proprioceptive sensory feedback. *Proc. Natl. Acad. Sci. U.S.A.* 111, 16877–16882. doi: 10.1073/pnas.1419045111

Amrollah, E., and Henaff, P. (2010). On the role of sensory feedbacks in Rowat-Selverston CPG to improve robot legged locomotion. *Front. Neurorobotics* 4:113. doi: 10.3389/fnbot.2010.00113

Andrada, E., Mampel, J., Schmidt, A., Fischer, M. S., Karguth, A., Witte, H., et al. (2013). From biomechanics of rats' inclined locomotion to a climbing robot. *Int. J. Design Nat. Ecodyn.* 8, 191–212. doi: 10.2495/dne-v8-n3-192-212

Arbas, E. A., and Calabrese, R. L. (1987). Ionic conductances underlying the activity control heartbeat in the medicinal leech. *J. Neurosci.* 7, 3945–3952.

Aschenbeck, K. S., Kern, N. I., Bachmann, R. J., and Quinn, R. D. (2006). "Design of a quadruped robot driven by air muscles," in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006* (Pisa: IEEE), 875–880.

Ayers, J. (2004). Underwater walking. *Arthropod Struct. Dev.* 33, 347–360. doi: 10.1016/j.asd.2004.06.001

Ayers, J. L., and Selverston, A. I. (1979). Monosynaptic entrainment of an endogenous pacemaker network: a cellular mechanism for von Holst's magnet effect. *J. Comp. Physiol.* 129, 5–17. doi: 10.1007/BF00679907

Bässler, U., and Büschges, A. (1998). Pattern generation for stick insect walking movements-multisensory control of a locomotor program. *Brain Res. Rev.* 27, 65–88. doi: 10.1016/S0165-0173(98)00006-X

Bicanski, A., Ryczko, D., Knuesel, J., Harischandra, N., Charrier, V., Ekeberg, Ö., et al. (2013). Decoding the mechanisms of gait generation in salamanders by combining neurobiology, modeling and robotics. *Biol. Cybern.* 107, 545–564. doi: 10.1007/s00422-012-0543-1

Boaventura, T., Medrano-Cerda, G. A., Semini, C., Buchli, J., and Caldwell, D. G. (2013). "Stability and performance of the compliance controller of the quadruped robot HyQ," in *IEEE International Conference on Intelligent Robots and Systems* (Tokyo), 1458–1464. doi: 10.1109/iros.2013.6696541

Brown, T. G. (1914). On the nature of the fundamental activity of the nergous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *J. Physiol.* 48, 18–46. doi: 10.1113/jphysiol.1914.sp001646

Büschges, A., and Borgmann, A. (2013). Network modularity : back to the future in motor control. *Curr. Biol.* 23, 936–938. doi: 10.1016/j.cub.2013.09.021

Chen, J., Friesen, W. O., and Iwasaki, T. (2011). Mechanisms underlying rhythmic locomotion: body-fluid interaction in undulatory swimming. *J. Exp. Biol.* 214(Pt 4):561–574. doi: 10.1242/jeb.048751

Cofer, D. W., Cymbalyuk, G., Reid, J., Zhu, Y., Heitler, W. J., and Edwards, D. H. (2010). AnimatLab: a 3D graphics environment for neuromechanical simulations. *J. Neurosci. Methods* 187, 280–288. doi: 10.1016/j.jneumeth.2010.01.005

Cohen, A. H., Ermentrout, G. B., Kiemel, T., Kopell, N., Sigvardt, K. A., and Williams, T. L. (1992). Modelling of intersegmental coordination in the lamprey central pattern generator for locomotion. *Trends Neurosci.* 15, 434–438. doi: 10.1016/0166-2236(92)90006-T

Colborne, G. R., Walker, A. M., Tattersall, A. J., and Fuller, C. J. (2006). Effect of trotting velocity on work patterns of the hind limbs of Greyhounds. *Am. J. Veter. Res.* 67, 1293–1298. doi: 10.2460/ajvr.67.8.1293

Crespi, A., Badertscher, A., Guignard, A., and Ijspeert, A. J. (2005). AmphiBot I: an amphibious snake-like robot. *Robot. Auton. Syst.* 50, 163–175. doi: 10.1016/j.robot.2004.09.015

Crespi, A., Karakasiliotis, K., Guignard, A., and Ijspeert, A. J. (2013). Salamandra Robotica II: an amphibious robot to study salamander-like swimming and walking gaits. *IEEE Trans. Robot.* 29, 308–320. doi: 10.1109/TRO.2012.2234311

Dasgupta, S., Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2015). Distributed recurrent neural forward models with synaptic adaptation

and CPG-based control for complex behaviors of walking robots. *Front. Neurorobotics* 9:10. doi: 10.3389/fnbot.2015.00010

Daun-Gruhn, S., Rubin, J. E., and Rybak, I. A. (2009). Control of oscillation periods and phase durations in half-center central pattern generators: a comparative mechanistic analysis. *J. Comp. Neurosci.* 27, 3–36. doi: 10.1007/s10827-008-0124-4

Ekeberg, Ö. (1993). A combined neuronal and mechanical model of fish swimming. *Biol. Cybern.* 69, 363–374. doi: 10.1007/BF01185408

Ekeberg, Ö., Blümel, M., and Büschges, A. (2004). Dynamic simulation of insect walking. *Arthropod Struct. Dev.* 33, 287–300. doi: 10.1016/j.asd.2004.05.002

Ekeberg, O., and Grillner, S. (1999). Simulations of neuromuscular control in lamprey swimming. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 354, 895–902. doi: 10.1098/rstb.1999.0441

Ekeberg, Ö., and Pearson, K. G. (2005). Computer simulation of stepping in the hind legs of the cat: an examination of mechanisms regulating the stance-to-swing transition. *J. Neurophysiol.* 94, 4256–4268. doi: 10.1152/jn.00065.2005

Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E. I., Ornelas-Rodriguez, M., and Sotelo-Figueroa, M. (2016). Design of spiking central pattern generators for multiple locomotion gaits in hexapod robots by christiansen grammar evolution. *Front. Neurorobotics* 10:6. doi: 10.3389/fnbot.2016.00006

Fischer, M. S., and Blickhan, R. (2006). The tri-segmented limbs of therian mammals: kinematics, dynamics, and self-stabilizationa review. *J. Exp. Zool. A Comp. Exp. Biol.* 305, 935–952. doi: 10.1002/jez.a.333

Fischer, M. S., and Lilje, K. E. (2011). *Dogs in Motion*. Essen: VDH Service GmbH.

Fischer, M. S., Schilling, N., Schmidt, M., Haarhaus, D., and Witte, H. (2002). Basic limb kinematics of small therian mammals. *J. Exp. Biol.* 205(Pt 9):1315–1338.

Geertsen, S. S., Stecina, K., Meehan, C. F., Nielsen, J. B., and Hultborn, H. (2011). Reciprocal IA inhibition contributes to motoneuronal hyperpolarisation during the inactive phase of locomotion and scratching in the cat. *J. Physiol.* 589(Pt 1):119–134. doi: 10.1113/jphysiol.2010.199125

Hägglund, M., Dougherty, K. J., Borgius, L., Itohara, S., Iwasato, T., and Kiehn, O. (2013). Optogenetic dissection reveals multiple rhythmogenic modules underlying locomotion. *Proc. Natl. Acad. Sci. U.S.A.* 110, 11589–11594. doi: 10.1073/pnas.1304365110

Hill, A. (1970). *First and Last Experiments in Muscle Mechanics*. London, UK: Cambridge University Press.

Hooper, S. L., Guschlbauer, C., Blümel, M., Rosenbaum, P., Gruhn, M., Akay, T., et al. (2009). Neural control of unloaded leg posture and of leg swing in stick insect, cockroach, and mouse differs from that in larger animals. *J. Neurosci.* 29, 4109–4119. doi: 10.1523/JNEUROSCI.5510-08.2009

Hunt, A. J. (2015). *Neurologically Based Control for Quadruped Walking*. Ph.D. thesis, Case Western Reserve University.

Hunt, A., Schmidt, M., Fischer, M., and Quinn, R. D. (2014). "Neuromechanical simulation of an inter-leg controller for tetrapod coordination," in *Biomimetic and Biohybrid Systems,* eds A. Duff, N. F. Lepora, A. Mura, T. J. Prescott, and P. F. M. J. Verschure (Milan: Springer International Publishing), 142–153.

Hunt, A. J., Schmidt, M., Fischer, M. S., and Quinn, R. D. (2015a). A biologically based neural system coordinates the joints and legs of a tetrapod. *Bioinspiration Biomimetics* 10:055004. doi: 10.1088/1748-3190/10/5/055004

Hunt, A. J., Szczecinski, N. S., Andrada, E., Fischer, M. S., and Quinn, R. D. (2015b). "Using animal data and neural dynamics to reverse engineer a neuromechanical rat model," in *Biomimetic and Biohybrid Systems - Living Machines 2015*, eds S. P. Wilson, P. F. M. J. Verschure, A. Mura, and T. J. Prescott (Barcelona: Springer International Publishing), 211–222.

Iida, F., and Ijspeert, A. J. (2016). "Biologically inspired robotics," in *Springer Handbook of Robotics,* eds B. Siciliano and O. Khatib (Gewerbestrasse: Springer International Publishing), 2015–2034.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Ijspeert, A. J. (2014). Biorobotics: using robots to emulate and investigate agile locomotion. *Science (New York, N.Y.)* 346, 196–203. doi: 10.1126/science.1254486

Ijspeert, A. J., Hallam, J., and Willshaw, D. (1999). Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adapt. Behav.* 7, 151–172. doi: 10.1177/105971239900700202

Inada, H., and Ishii, K. (2004). Bipedal walk using a central pattern generator. *Int. Cong. Series* 1269, 185–188. doi: 10.1016/j.ics.2004.05.129

Iwasaki, T., and Zheng, M. (2006). Sensory feedback mechanism underlying entrainment of central pattern generator to mechanical resonance. *Biol. Cybern.* 94, 245–261. doi: 10.1007/s00422-005-0047-3

Jankowska, E. (1992). Interneuronal relay in spinal pathways from proprioceptors. *Prog. Neurobiol.* 38, 335–378. doi: 10.1016/0301-0082(92)90024-9

Jankowska, E. (2008). Spinal interneuronal networks in the cat: elementary components. *Brain Res. Rev.* 57, 46–55. doi: 10.1016/j.brainresrev.2007.06.022

Jindrich, D. L., and Full, R. J. (2002). Dynamic stabilization of rapid hexapedal locomotion. *J. Exp. Biol.* 205(Pt 18):2803–2823.

Karakasiliotis, K., Thandiackal, R., Melo, K., Horvat, T., Mahabadi, N. K., and Tsitkov, S. (2016). From cineradiography to biorobots : an approach for designing robots to emulate and study animal locomotion. *J. R. Soc. Interface* 13, 1–22. doi: 10.1098/rsif.2015.1089

Klein, T. J., and Lewis, M. A. (2012). A physical model of sensorimotor interactions during locomotion. *J. Neural Eng.* 9:46011. doi: 10.1088/1741-2560/9/4/046011

Lewinger, W. A., and Quinn, R. D. (2010). "A Hexapod walks over irregular terrain using a controller adapted from an insect's nervous system," in *IEEE International Conference on Intelligent Robots and Systems* (Taipei), 3386–3391. doi: 10.1109/iros.2010.5650200

Li, W., Szczecinski, N. S., Hunt, A. J., and Quinn, R. D. (2016). "A neural network with central pattern generators entrained by sensory feedback controls walking of a bipedal model," in *Biomimetic and Biohybrid Systems: 5th International Conference, Living Machines*, eds N. F. Lepora, A. Mura, M. Mangan, P. F. M. J. Verschure, M. Desmulliez, and T. J. Prescott (Edinburgh, UK: Springer International Publishing), 144–154.

Loeb, E., Brown, E., and Cheng, J. (1999). A hierarchical foundation for models of sensorimotor control. *Exp. Brain Res.* 126, 1–18. doi: 10.1007/s002210050712

Lundberg, A. (1981). "Half-centres revisited," in *Regulatory functions of the CNS: Principles of Motion and Organization - 28th International Congress of Phsiological Sciences* (Budapest), 155–167.

Markin, S. N., Klishko, A. N., Shevtsova, N. A., Lemay, M. A., Prilutsky, B. I., and Rybak, I. A. (2010). Afferent control of locomotor CPG: insights from a simple neuromechanical model. *Ann. N. Y. Acad. Sci.* 1198, 21–34. doi: 10.1111/j.1749-6632.2010.05435.x

McCrea, D., Pratt, C., and Jordan, L. M. (1980). Renshaw cell activity and recurrent effects on motoneurons during fictive locomotion. *J. Neurophysiol.* 44, 475–488.

McCrea, D. A., and Rybak, I. A. (2008). Organization of mammalian locomotor rhythm and pattern generation. *Brain Res. Rev.* 57, 134–146. doi: 10.1016/j.brainresrev.2007.08.006

McVea, D. A., Donelan, J. M., Tachibana, A., and Pearson, K. (2005). A role for hip position in initiating the swing-to-stance transition in walking cats. *J. Neurophysiol.* 94, 3497–508. doi: 10.1152/jn.00511.2005

Meyrand, P., Simmers, J., and Moulins, M. (1994). Dynamic construction of a neural network from multiple pattern generators in the lobster stomatogastric nervous system. *J. Neurosci.* 14, 630–644.

Mortin, L. I., and Stein, P. S. (1989). Spinal cord segments containing key elements of the central pattern generators for three forms of scratch reflex in the turtle. *J. Neurosci.* 9, 2285–2296.

Pearson, K. G. (2008). Role of sensory feedback in the control of stance duration in walking cats. *Brain Res. Rev.* 57, 222–227. doi: 10.1016/j.brainresrev.2007.06.014

Perkel, D. H., and Mulloney, B. (1974). Motor pattern production in reciprocally inhibitory neurons exhibiting postinhibitory rebound. *Science (New York, N.Y.)* 185, 181–183.

Pinsker, H. M. (1977). Aplysia bursting neurons as endogenous oscillators. I. Phase-response curves for pulsed inhibitory synaptic input. *J. Neurophysiol.* 40, 527–543.

Pratt, C. A., and Jordan, L. M. (1987). Ia inhibitory interneurons and Renshaw cells as contributors to the spinal mechanisms of fictive locomotion. *J. Neurophysiol.* 57, 56–71.

Pratt, G. A., and Williamson, M. M. (1995). "Series elastic actuators," in *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1 (Pittsburgh, PA: IEEE), 399–406. doi: 10.1109/iros.1995.525827

Prochazka, A., Gillard, D., and Bennett, D. J. (1997). Positive force feedback control of muscles. *J. Neurophysiol.* 77, 3226–3236.

Raibert, M., Blankespoor, K., Nelson, G. M., and Playter, R. (2008). "BigDog, the Rough-Terrain quadruped robot," in *17th World Congress, The International Federation of Automatic Control* (Seoul), 6–9. doi: 10.3182/20080706-5-kr-1001.01833

Rollinson, D., Ford, S., Brown, B., and Choset, H. (2013). "Design and modeling of a series elastic element for snake robots," in *ASME Dynamic Systems and Control* (Palo Alto, CA).

Rostro-Gonzalez, H., Cerna-Garcia, P. A., Trejo-Caballero, G., Garcia-Capulin, C. H., Ibarra-Manzano, M. A., Avina-Cervantes, J. G., et al. (2015). A CPG system based on spiking neurons for hexapod robot locomotion. *Neurocomputing* 170, 47–54. doi: 10.1016/j.neucom.2015.03.090

Russell, A., Orchard, G., and Etienne-Cummings, R. (2007). "Configuring of spiking central pattern generator networks for bipedal walking using genetic algorthms," in *2007 IEEE International Symposium on Circuits and Systems*, 1525–1528. doi: 10.1109/ISCAS.2007.378701

Rutter, B. L., Taylor, B. K., Bender, J. A., Blümel, M., William, A., Ritzmann, R. E., et al. (2011). "Descending commands to an insect leg controller network cause smooth behavioral transitions," in *Intelligent Robots and Systems (IROS 2011)* (San Francisco, CA), 215–220. doi: 10.1109/iros.2011.6095030

Rybak, I. A., Shevtsova, N. A., and Kiehn, O. (2013). Modelling genetic reorganization in the mouse spinal cord affecting left-right coordination during locomotion. *J. Physiol.* 591(Pt 22):5491–5508. doi: 10.1113/jphysiol.2013.261115

Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H. (2013a). Walknet, a bio-inspired controller for hexapod walking. *Biol. Cybern.* 107, 397–419. doi: 10.1007/s00422-013-0563-5

Schilling, M., Paskarbeit, J., Hoinville, T., Hüffmeier, A., Schneider, A., Schmitz, J., et al. (2013b). A hexapod walker using a heterarchical architecture for action selection. *Front. Comput. Neurosci.* 7:126. doi: 10.3389/fncom.2013.00126

Seok, S., Wang, A., Yee, M., Chuah, M., Hyun, D. J., Lee, J., et al. (2015). Design principles for energy-efficient Legged locomotion and implementation on the MIT cheetah robot. *IEEE/ASME Trans. Mechatr.* 20, 1117–1129. doi: 10.1109/TMECH.2014.2339013

Spröwitz, A. T., Ajallooeian, M., Tuleu, A., and Ijspeert, A. J. (2014). Kinematic primitives for walking and trotting gaits of a quadruped robot with compliant legs. *Front. Comput. Neurosci.* 8:27. doi: 10.3389/fncom.2014.00027

Szczecinski, N. S., Brown, A. E., Bender, J. A., Quinn, R. D., and Ritzmann, R. E. (2014). A neuromechanical simulation of insect walking and transition to turning of the cockroach *Blaberus discoidalis*. *Biol. Cybern.* 108, 1–21. doi: 10.1007/s00422-013-0573-3

Szczecinski, N. S., Chrzanowski, D. M., Cofer, D. W., Terrasi, A. S., Moore, D. R., Martin, J. P., et al. (2015). "Introducing MantisBot: hexapod robot controlled by a high- fidelity, real-time neural simulation," in *IEEE International Conference on Intelligent Robots and Systems* (Hamburg, DE), 3875–3881. doi: 10.1109/iros.2015.7353922

Szczecinski, N. S., Hunt, A., and Quinn, R. (2017). Design process and tools for dynamic neuromechanical models and robot controllers. *Biol. Cybern.* 111, 105–127. doi: 10.1007/s00422-017-0711-4

Talpalar, A. E., Bouvier, J., Borgius, L., Fortin, G., Pierani, A., and Kiehn, O. (2013). Dual-mode operation of neuronal networks involved in left-right alternation. *Nature* 500, 85–88. doi: 10.1038/nature12286

Thorson, I., and Caldwell, D. (2011). "A nonlinear series elastic actuator for highly dynamic motions," in *IEEE International Conference on Intelligent Robots and Systems* (San Francisco, CA), 390–394. doi: 10.1109/iros.2011.6094913

Tryba, A. K., Peña, F., and Ramirez, J.-M. (2006). Gasping activity *in vitro*: a rhythm dependent on 5-HT2A receptors. *J. Neurosci.* 26, 2623–2634. doi: 10.1523/JNEUROSCI.4186-05.2006

von Twickel, A., Büschges, A., and Pasemann, F. (2011). Deriving neural network controllers from neuro-biological data: implementation of a single-leg stick insect controller. *Biol. Cybern.* 104, 95–119. doi: 10.1007/s00422-011-0422-1

Zhong, G., Shevtsova, N. A., Rybak, I. A., and Harris-Warrick, R. M. (2012). Neuronal activity in the isolated mouse spinal cord during spontaneous deletions in fictive locomotion: insights into locomotor central pattern generator organization. *J. Physiol.* 590, 4735–4759. doi: 10.1113/jphysiol.2012.240895

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Hunt, Szczecinski and Quinn. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 12 April 2017
doi: 10.3389/fnbot.2017.00020

A Neurocomputational Model of Goal-Directed Navigation in Insect-Inspired Artificial Agents

Dennis Goldschmidt[1,2]*, *Poramate Manoonpong*[3] *and Sakyasingha Dasgupta*[4,5]

[1] Bernstein Center for Computational Neuroscience, Third Institute of Physics – Biophysics, Georg-August University, Göttingen, Germany, [2] Champalimaud Neuroscience Programme, Champalimaud Centre for the Unknown, Lisbon, Portugal, [3] Embodied AI and Neurorobotics Lab, Centre of BioRobotics, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark, [4] IBM Research, Tokyo, Japan, [5] Riken Brain Science Institute, Saitama, Japan

Despite their small size, insect brains are able to produce robust and efficient navigation in complex environments. Specifically in social insects, such as ants and bees, these navigational capabilities are guided by orientation directing vectors generated by a process called path integration. During this process, they integrate compass and odometric cues to estimate their current location as a vector, called the home vector for guiding them back home on a straight path. They further acquire and retrieve path integration-based vector memories globally to the nest or based on visual landmarks. Although existing computational models reproduced similar behaviors, a neurocomputational model of vector navigation including the acquisition of vector representations has not been described before. Here we present a model of neural mechanisms in a modular closed-loop control—enabling vector navigation in artificial agents. The model consists of a path integration mechanism, reward-modulated global learning, random search, and action selection. The path integration mechanism integrates compass and odometric cues to compute a vectorial representation of the agent's current location as neural activity patterns in circular arrays. A reward-modulated learning rule enables the acquisition of vector memories by associating the local food reward with the path integration state. A motor output is computed based on the combination of vector memories and random exploration. In simulation, we show that the neural mechanisms enable robust homing and localization, even in the presence of external sensory noise. The proposed learning rules lead to goal-directed navigation and route formation performed under realistic conditions. Consequently, we provide a novel approach for vector learning and navigation in a simulated, situated agent linking behavioral observations to their possible underlying neural substrates.

Keywords: path integration, artificial intelligence, insect navigation, neural networks, reward-based learning

OPEN ACCESS

Edited by:
Mehdi Khamassi, Université Pierre et Marie Curie, France

Reviewed by:
Nicolas Cuperlier, Université de Cergy-Pontoise, France
Andrew Philippides, University of Sussex, UK

***Correspondence:**
Dennis Goldschmidt
dennis.goldschmidt@neuro.fchampalimaud.org

Received: *13 December 2016*
Accepted: *24 March 2017*
Published: *12 April 2017*

Citation:
Goldschmidt D, Manoonpong P and Dasgupta S (2017) A Neurocomputational Model of Goal-Directed Navigation in Insect-Inspired Artificial Agents. Front. Neurorobot. 11:20. doi: 10.3389/fnbot.2017.00020

1. INTRODUCTION

Social insects, including ants and bees, have evolved remarkable behavioral capabilities for navigating in complex dynamic environments, which enable them to survive by finding vital locations (e.g., food sources). For example, desert ants are able to forage and find small, sparsely distributed food items in a featureless environment, and form stereotyped and efficient routes

between their nest and reliable food sources (Collett, 2012; Mangan and Webb, 2012; Collett and Cardé, 2014; Cheng et al., 2014). These navigational behaviors not only rely on sensory information, mainly from visual cues, but also on internal memories acquired through learning mechanisms (Collett et al., 2013). Such learned memories have shown to be based on orientation directing vectors, which are generated by a process called path integration (PI) (Wehner, 2003).

1.1. Vector Navigation in Social Insects

In PI, animals integrate angular and linear ego-motion cues over time to produce an estimate of their current location with respect to their starting point. This vector representation is called the home vector (HV) and is used by social insects to return back to the home on a straight path. Many animals have been shown to apply PI, including vertebrate (Etienne and Jeffery, 2004) and invertebrate species (Srinivasan, 2015). While PI has mainly been observed in homing behavior, it can also serve as a scaffold for spatial learning of food sources (Collett et al., 1999, 2013). Indeed, experiments have shown that desert ants are capable of forming such memories by using their path integrator (Schmid-Hempel, 1984; Collett et al., 1999). Such memory is interpreted as a so-called global vector (GV), because the vector origin is fixed to the nest (Collett et al., 1998). If the ant is forced to take a detour during a foraging trip, the deviation from the GV is compensated by comparing the GV with the current PI state (Collett et al., 1999). Another example of vector memory is the waggle dance of honeybees (De Marco and Menzel, 2005; Menzel et al., 2005), in which the distance and direction to a goal are encoded by the duration and direction of the dance, respectively. After returning from a successful foraging run, insects re-apply this vector information in subsequent foraging runs (Capaldi et al., 2000; Wolf et al., 2012; Fernandes et al., 2015).

Although PI plays a key role in navigating through environments where visual cues, such as landmarks, are abundant, it also influences navigational behaviors in cluttered environments (Bühlmann et al., 2011). If an ant follows a learned GV repeatedly, it learns the heading directions at local landmarks along the path (Collett and Collett, 2009). These heading directions are view-based from the visual panorama surrounding the ant (Graham and Cheng, 2009; Narendra et al., 2013), and vector-based with additional information about the path segment length (Collett and Collett, 2009, 2015). The latter vector memories are also termed local vectors, because their retrieval is linked to local landmarks instead of global location with respect to the nest (Collett et al., 1998). Besides spatial learning of locations and routes, searching patterns of desert ants have also shown to be influenced by PI (Bolek and Wolf, 2015; Pfeffer et al., 2015).

1.2. Neural Substrates of Social Insect Navigation

Neural substrates of social insect navigation have yet to be completely identified, but previous findings of neural representations of compass cues and visual sceneries may provide essential information about how PI and vector learning is achieved in neural systems (Duer et al., 2015; Plath and Barron, 2015; Seelig and Jayaraman, 2015; Weir and Dickinson, 2015). In particular, neurons in the central complex, a protocerebral neuropil in the insect brain, have shown to be involved in visually guided navigation.

The main sensory cue for PI in social insects is derived from the linear polarization of scattered sunlight (Homberg et al., 2011; Lebhardt et al., 2012; Evangelista et al., 2014). Specialized photoreceptors in the outer dorsal part of the insect eye detect certain orientations of linear polarization, which depend on the azimuthal position of the sun. A distinct neural pathway processes polarization-derived signals leading to neurons in the central complex, which encode azimuthal directions of the sun (Heinze and Homberg, 2007). In a recent study, Seelig and Jayaraman (2015) placed a fruit fly tethered on a track ball setup in a virtual environment and measured the activity of neurons in the central complex. They demonstrated that certain neurons in the ellipsoid body, which is a toroidal subset in the central complex, encode for the animal's body orientation based on visual landmarks and angular self-motion. When both visual and self-motion cues are absent, this representation is maintained through persistent activity, which is a potential neural substrate for short-term memory in insects (Dubnau and Chiang, 2013). A similar neural code of orientations has been found in the rat limbic system (Taube et al., 1990). These so-called head direction (HD) cells are derived from motor and vestibular sensory information by integrating head movements through space. Thus, neural substrates of allothetic compass cues have been found in both invertebrate and vertebrate species. These cues provide input signals for a potential PI mechanism based on the accumulation of azimuthal directions of the moving animal as previously proposed by Kubie and Fenton (2009).

1.3. Computational Models of Vector-Guided Navigation

Because spatial navigation is a central task of biological as well as artificial agents, many studies have focused on computational modeling of such behavioral capabilities (see Madl et al., 2015 for review). Computational modeling has been successful in exploring the link between neural structures and their behavioral function, including learning (Bienenstock et al., 1982; Oja, 1982), perception (Salinas and Abbott, 1995; Olshausen and Field, 1997), and motor control (Todorov and Jordan, 2002). It allows for hypotheses about the underlying mechanisms to be defined precisely and their generated behavior can be examined and validated qualitatively and quantitatively with respect to experimental data.

Most models of PI favor a particular coordinate system (Cartesian or polar) and reference frame (geo- or egocentric) to perform PI based on theoretical and biological arguments (Vickerstaff and Cheung, 2010). While some models (Müller and Wehner, 1988; Hartmann and Wehner, 1995) include behavioral data from navigating animals in order to argue for their proposed PI method, others (Wittmann and Schwegler, 1995; Haferlach et al., 2007; Kim and Lee, 2011) have applied neural network models to investigate possible memory mechanisms for PI. Despite the wide variety of models, only a few of these models

have been implemented on embodied artificial agents (Schmolke et al., 2002; Haferlach et al., 2007) and in foraging tasks similar to the ones faced by animals in terms of distance and tortuosity of paths (Lambrinos et al., 1997, 2000). Furthermore, while some vertebrate-inspired models (Gaussier et al., 2000; Jauffret et al., 2015) offer underlying spatial learning mechanisms based on place and view cells, many insect-inspired models have not linked PI and navigational capabilities to spatial learning and memory. A notable exception is a recent model based on the *Drosophila* brain show impressive results to generate adaptive behaviors in an autonomous agent, including exploration, visual landmark learning, and homing (Arena et al., 2014). However, the model has not been explicitly shown to be scalable for long-distance central-place foraging as observed in social insects.

Kubie and Fenton (2009) proposed a PI model based on the summation of path segments with HD accumulator cells, which are individually tuned to different HDs and hypothesized to encode how far the animal traveled in this direction. These summated path vectors are then stored in a fixed memory structure called shortcut matrix, which is used for navigating toward goals. Although this model is based on HD cells and therefore presented as for mammalian navigation, recent findings in *Drosophila melanogaster* (Seelig and Jayaraman, 2015) demonstrate that similar HD accumulator cells can also be hypothesized for insect navigation. Similar HD accumulator models have been applied for chemo-visual robotic navigation (Mathews et al., 2009) and PI-based homing behavior (Kim and Lee, 2011).

Cruse and Wehner (2011) presented a decentralized memory model of insect vector navigation to demonstrate that the observed navigational capabilities do not require a map-like memory representation. Their model is a cybernetical network structure, which mainly consists of a PI system, multiple memory banks and internal motivational states that control the steering angle of a simulated point agent. The PI system provides the position of the agent given by euclidean coordinates, which are stored as discrete vector memories when the agent finds a food location. To our knowledge, this model is the first and only modeling approach which accounts for behavioral aspects of insect vector navigation. However, although they introduce a learning rule for so-called quality values of stored vectors in a more recent version of the model (Hoinville et al., 2012), their model does not account for how the navigation vectors are represented and learned in a neural implementation.

1.4. Our Approach

Inspired by these findings, in this paper, we present a novel model framework for PI and adaptive vector navigation as observed in social insects. The framework is applied as closed-loop control to an artificial agent and consists of four functional subparts: (1) a neural PI mechanism, (2) a reward-modulated learning rule for vector memories, (3) random search, and (4) an adaptive action selection mechanism. Here, the artificial agent primarily enables us to provide the necessary physical embodiment (Webb, 1995) in order to test the efficacy of our adaptive navigation mechanism, without a detailed reverse engineering of the insect brain.

Based on population-coded heading directions in circular arrays, we apply PI by accumulating speed-modulated HD signals through a self-recurrent loop. The final home vector representation is computed by local excitation-lateral inhibition connections, which projects accumulated heading directions onto the array of output neurons. The activity of these neurons encodes the vector angle as the position of maximum firing in the array, and the vector length as the amplitude of the maximum firing rate in the array. The self-localization ability of PI allows social insects to learn spatial representations for navigation (Collett et al., 1999). We design a reward-modulated associative learning rule (Smith et al., 2008; Cassenaer and Laurent, 2012; Hige et al., 2015) to learn vector representations based on PI. This vector, called global vector, connects the nest to a rewarding food location. Vectors are learned by associating the PI state and a reward received at the food location given a context-dependent state. This association induces weight changes in plastic synapses connecting the context-dependent unit to a circular array of neurons, which represents the vector. The context-dependent unit activates the vector representation in the array, and therefore represents a motivational state for goal-directed foraging. Using the vector learning rule, the agent is able to learn rewarding locations and demonstrate goal-directed navigation. Because of the vector addition of global and inverted home vector in the action selection mechanism, it can compensate for unexpected detours from the original trajectory, such as obstacles (Collett et al., 1999, 2001).

Taken together, our model is a novel framework for generating and examining social insect navigation based on PI and vector representations. It is based on plausible neural mechanisms, which are related to neurobiological findings in the insect central complex. Therefore, we provide a computational approach for linking behavioral observations to their possible underlying neural substrates. In the next section, we will describe the proposed model for reward-modulated vector learning and navigation. The results section will provide detailed descriptions of our experimental setups and simulation results. Finally, conclusions and implications of our model with respect to behavioral and neurobiological studies are discussed in Section 4.

2. MATERIALS AND METHODS

In this paper we propose an insect-inspired model of vector-guided navigation in artificial agents using modular closed-loop control. The model (see **Figure 1A**) consists of four parts: (1) a neural PI mechanism, (2) plastic neural circuits for reward-based learning of vector memories, (3) random search, and (4) action selection. The neural mechanisms in our model receive multimodal sensory inputs from exteroceptive and proprioceptive sensors to produce a directional signal based on a vector (see **Figure 1B**). This vector is represented by the activity of circular arrays, where the position of the maximum indicates its direction and the amplitude at this position indicates its length. We evaluate our model in simulation using a two-dimensional point agent as well as a hexapod walking robot (see Supplementary Material for details).

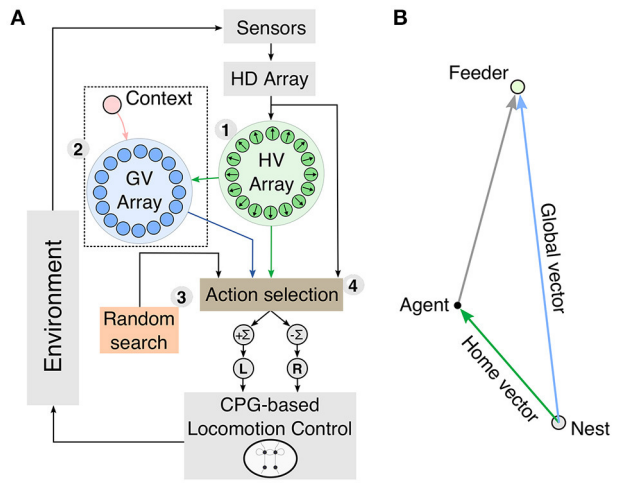


FIGURE 1 | Schematic diagram of the modular closed-loop control for vector navigation. (A) The model consists of a neural path integration (PI) mechanism (1), reward-modulated vector learning (2), random search (3), and action selection (4). Vector information for guiding navigation is computed and represented in the activity of circular arrays. The home vector (HV) array is the output of the PI mechanism and is applied for homing behavior and as a scaffold for global vector (GV) learning. These three vector representations and random search are integrated through an adaptive action selection mechanism, which produces the steering command to the CPG-based locomotion control. **(B)** Spatial representation of the different vectors used for navigation. The HV is computed by PI and gives an estimate for the current location of the agent. In general, GVs connect the nest to a rewarding location. Using vector addition, the agent is able to compute, how to orient from its current location toward the feeder.

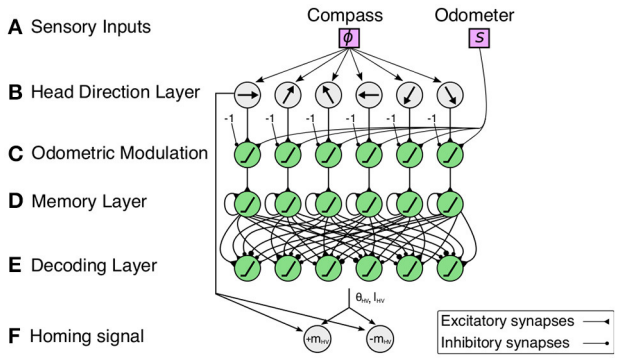


FIGURE 2 | Multilayered neural network of the proposed path integration (PI) mechanism. (A) Sensory inputs from a compass sensor (ϕ) and odometer (s) are provided to the mechanism. **(B)** Neurons in the head direction (HD) layer encodes the sensory input from a compass sensor using a cosine response function. Each neuron encodes a particular preferred direction enclosing the full range of 2π. Note that the figure depicts only six neurons for simplicity. **(C)** An odometric sensory signal (i.e., walking speed) is used to modulate the HD signals. **(D)** The memory layer accumulates the signals by self-recurrent connections. **(E)** Cosine weight kernels decode the accumulated directions to compute the output activity representing the home vector (HV). **(F)** The difference between the HV angle and current heading angle is used to compute the homing signal (see Equation 11).

2.1. Path Integration (PI) Mechanism for Home Vector (HV) Representation

The PI mechanism (**Figure 2**) is a multilayered neural network consisting of circular arrays, where the final layer's activity pattern represents the HV. Neural activities of the circular arrays represent population-coded compass information and rate-coded linear displacements. Incoming signals are sustained through leaky neural integrator circuits, and they compute the HV by local excitatory-lateral inhibitory interactions.

A) Sensory inputs

The PI mechanism receives angular and linear cues as sensory inputs. Like in social insects, angular cues are derived from allothetic compass cues. We employ a compass sensor which measures the angle ϕ of the agent's orientation. In insects, this information is derived from the combination of sun- and skylight compass information (Wehner, 2003). In desert ants, it has been found that linear cues are derived from the strides taken by the animal during the journey (Wittlinger et al., 2006, 2007). For our model, we assume that such odometry is translated into an estimate of the animal's walking speed. For the embodied agent employed here (i.e., a hexapod robot), the walking speed is computed by accumulating steps and averaging over a certain time window. These step counting signals are derived from the motor signals. The input signals for the angular component ϕ and the linear component s have value ranges of

$$\phi \in [0,\ 2\pi), \tag{1}$$

$$s \in [0,\ 1]. \tag{2}$$

B) Head direction layer

The first layer of the neural network is composed of HD cells with activation functions

$$x_i^{\mathrm{HD}}(\phi(t)) = \cos(\phi(t) - \phi_i), \tag{3}$$

$$\phi_i = \frac{2\pi i}{N},\ i \in [0,\ N-1], \tag{4}$$

where the compass signal $\phi(t)$ is encoded by a cosine response function with N preferred directions $\phi_i \in [0, 2\pi)$. The resolution is determined by $\Delta\phi = \frac{2\pi}{N}$ and the coarse encoding of variables, here angles, by cosine responses allows for high accuracy and optimized information transfer (Eurich and Schwegler, 1997). Coarse coding has been shown to be present in different sensory processing in the insect brain, including olfactory (Friedrich and Stopfer, 2001) and visual processing (Wystrach et al., 2014). Furthermore, it has been shown that polarization-sensitive neurons in the anterior optic tubercle of locusts exhibit broad and sinusoidal tuning curves of 90–120° (Heinze et al., 2009; Heinze and Homberg, 2009; el Jundi and Homberg, 2012). Head-direction cells in the central complex of *Drosophila melanogaster* were shown to have activity bump widths of 80–90° (Seelig and Jayaraman, 2015). However, their measurements are based on calcium imaging data, which is only an approximation of the neuron's firing rate.

C) Odometric modulation of head direction signals

The second layer acts as a gating mechanism (G), which modulates the neural activity using the odometry signal s ($\in [0, 1]$). Therefore, it encodes in its activity, the traveled distances of the agent. The gating layer units decrease the HD activities by a constant bias of 1, so that the maximum activity is equal to zero. A positive speed increases the signal linearly. The gating activity is defined as follows:

$$x_i^{\mathrm{G}}(t) = f\left(\sum_{j=0}^{N-1} \delta_{ij} x_j^{HD}(t) - 1 + s\right), \tag{5}$$

$$f(x) = \max(0, x), \tag{6}$$

where δ_{ij} is the Kronecker delta, i.e., first layer neurons j and second layer neurons i are connected one-to-one. Forward speed signals have been found in the central complex of walking cockroaches (Martin et al., 2015).

D) Memory layer

The third layer is the so-called memory layer (M), where the speed-modulated HD activations are temporally accumulated through self-excitatory connections:

$$x_i^{\mathrm{M}}(t) = f\left(\sum_{j=0}^{N-1} \delta_{ij} x_j^{G}(t) + (1-\lambda) x_i^{\mathrm{M}}(t - \Delta t)\right), \tag{7}$$

where λ is a positive constant defined as the integrator leak rate, which indicates the loss of information over time. A leaky integrator has previously been applied by Vickerstaff (2007) to explain systematic errors in homing of desert ants (Müller and Wehner, 1988). If the leak rate is equal to zero, the accumulation of incoming directional signals is unbounded, which is not biologically plausible. As such, any path integration system based on linear integration therefore bounds the natural foraging range of the animal in order to exhibit accurate path integration (Burak and Fiete, 2009).

E) Decoding layer

The final and fourth layer decodes the activations from the memory layer to produce a vector representation, i.e., the HV, which serves as the output of the mechanism referred to as PI state:

$$x_i^{\mathrm{PI}}(t) = f\left(\sum_{j=0}^{N-1} w_{ij} x_j^{\mathrm{M}}(t)\right) \tag{8}$$

$$w_{ij} = \cos(\phi_i - \phi_j) = \cos\left(\frac{2\pi(i-j)}{N}\right), \tag{9}$$

where w_{ij} is a cosine kernel, which decomposes the projections of memory layer actitivities of the jth neuron to the ith neuron's preferred orientation. While a cosine synaptic weight kernel is biologically implausible, it is reasonable to assume that an approximate connectivity could arise from forming local-excitation lateral-inhibition connections (e.g., mexican-hat connectivity). An example of such a connectivity formed by cell proximity could be the ring architecture of head-direction-selective neurons in the ellipsoid body of the central complex (Seelig and Jayaraman, 2015; Wolff et al., 2015). The resulting HV is encoded by the average position of maximum firing in the array (angle θ_{HV}) and the sum of all firing rates of the array (length l_{HV}). We calculate the position of maximum firing using the population vector average given by:

$$\theta_{HV}(t) = \arctan\left(\frac{\sum_{i=0}^{N-1} x_i^{\mathrm{PI}}(t)\sin(2\pi i/N)}{\sum_{i=0}^{N-1} x_i^{\mathrm{PI}}(t)\cos(2\pi i/N)}\right), \tag{10}$$

where the denominator is the x coordinate of the population vector average, and the numerator is the y coordinate. See **Figure 3** for example output activities of the decoding layer neurons.

F) Homing signal

To apply the HV for homing behavior, i.e., returning home on a straight path, the vector is inverted by a 180° rotation. The difference between the heading direction ϕ and the inverted HV direction $\theta_{HV} - \pi$ is used for steering the agent toward home. The agent applies homing by sine error compensation, which defines the motor command:

$$m_{HV}(t) = l_{HV}(t)\sin\big(\theta_{HV}(t) - \phi(t) - \pi\big). \tag{11}$$

This leads to right ($m_{HV} < 0$) and left turns ($m_{HV} > 0$) for negative and positive differences, respectively, and thereby decreasing the net error at each step. The underlying dynamical behavior of this sine error compensation is defined by a stable and an unstable fixed point (see Supplementary Marterial). This leads to dense searching behavior around a desired position, where the error changes rapidly (Vickerstaff and Cheung, 2010).

2.2. A Reward-Modulated Learning Rule for Acquiring and Retrieving Vector Memories

We propose a heterosynaptic, reward-modulated learning rule (Smith et al., 2008; Cassenaer and Laurent, 2012; Hige et al., 2015) with a canonical form to learn vector memories based on four factors (see **Figure 4**): a context-dependent state, an input-dependent PI state, a modulatory reward signal, and the vector array state. Like the HV, GV memories are computed and represented in circular arrays. The context-dependent state, such as inbound or outbound foraging, activates the vector representation, and thus retrieves the vector memory. The association between the PI-based state and the reward signal modulates the plastic synapses connecting the context unit (presynaptic) with the vector array units (postsynaptic). The associated information is used by the agent on future foraging trips to steer toward the rewarding location. The received reward is an internally generated signal based on food reward due to visiting the feeder.

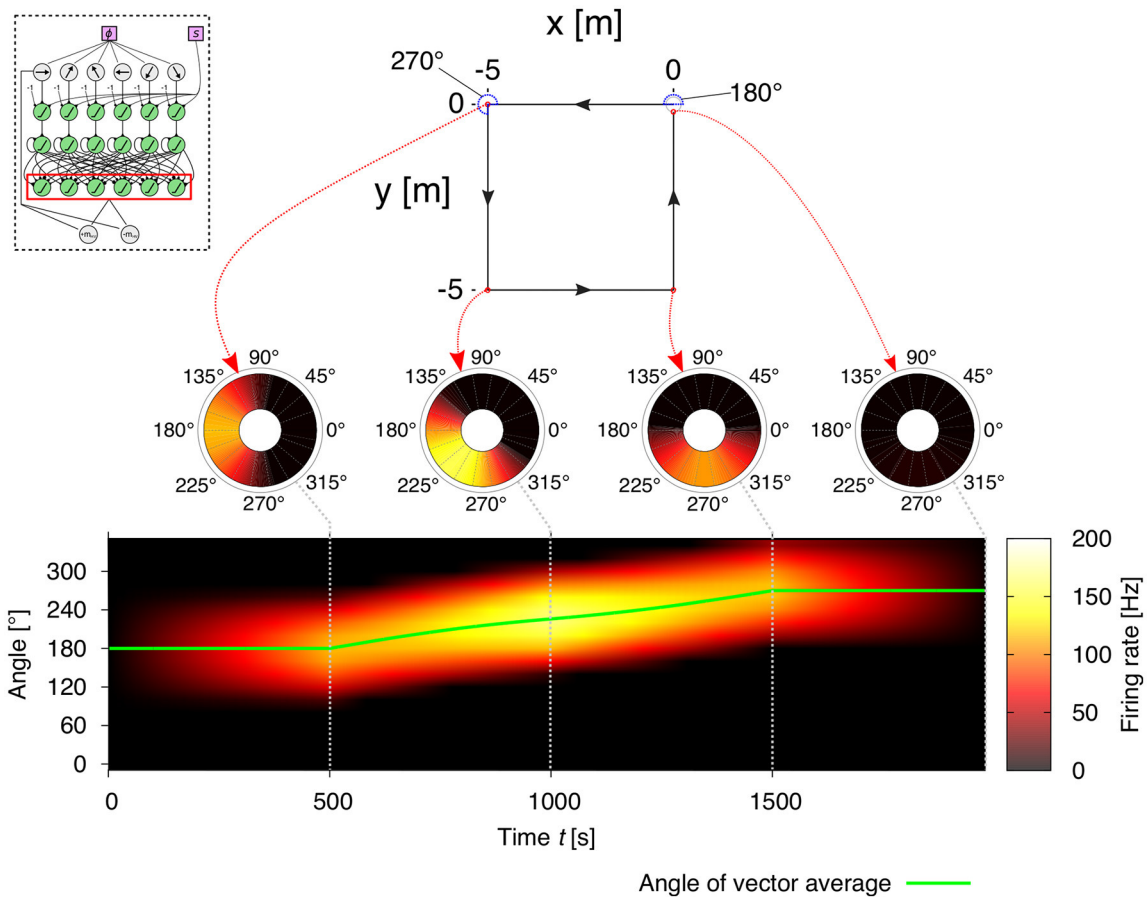


FIGURE 3 | Example of vector representations based on the neural activities of the decoding layer (see Figure 2E) in the path integration (PI) mechanism for a square trajectory. The agent runs for 5 m in one of the four directions (180°, 270°, 0°, 90°), thus finally returning to the starting point of its journey. The coarse encoding of heading orientations lead to a correct decoding of memory layer activities. Thus, the activities of the decoding layer in the PI mechanism (see inlay) represent the home vector (HV), where the position of the maximum firing rate is the angle and the amplitude of the maximum firing rate is the length of the vector. Note that, as the agent returns to the home position, the output activities are suppressed to zero resulting from the elimination of opposite directions.

The context-dependent unit (see **Figure 4**) is a unit that represents the agent's foraging state, i.e., inward or outward. Here we apply a simple binary unit given by:

$$\sigma(t) = \begin{cases} 1 & \text{if outward trip,} \\ 0 & \text{if inward trip.} \end{cases} \tag{12}$$

The context-dependent unit projects plastic synapses onto a circular array that represents the GV. The GV array has the same number of neurons, thus the same preferred orientations as the PI array. In this way, each neuron $i \in [0, N-1]$ has a preferred orientation of $\frac{2\pi i}{N}$. The activity x_i^{GV} of the GV array is given by:

$$x_i^{GV}(t) = w_i^{GV}(t)\sigma(t), \tag{13}$$

where w_i^{GV} are the weights of the plastic synapses. For these synapses, we apply a reward-modulated associative learning rule given by:

$$\Delta w_i^{GV}(t) = \mu^{GV} r(t)\sigma(t)\left(x_i^{PI}(t) - x_i^{GV}(t)\right), \tag{14}$$

$$w_i^{GV}(t+\Delta t) = w_i^{GV}(t) + \Delta w_i^{GV}(t), \tag{15}$$

where $\mu^{GV} = 2$ is the learning rate, and $x_i^{PI}(t)$ is the PI activity in the direction $i = \frac{2\pi i}{N}$. The weights are therefore only changed when the agent forages outbound, because for the inward trip we assume that the agent returns to the home on a straight path. This is in accordance with behavioral data indicating that ants acquire and retrieve spatial memories based on internal motivational states, given by whether they are on an inward or outward trip (Wehner et al., 2006). The food reward $r(t)$ at the feeder is given by:

$$r(t) = \max(0, 1 - 5d(t)) \tag{16}$$

where $d(t)$ is the agent's distance to the feeder, which we computed directly using the positions of the agent and feeder, given that the reward is physically bound to the location of the

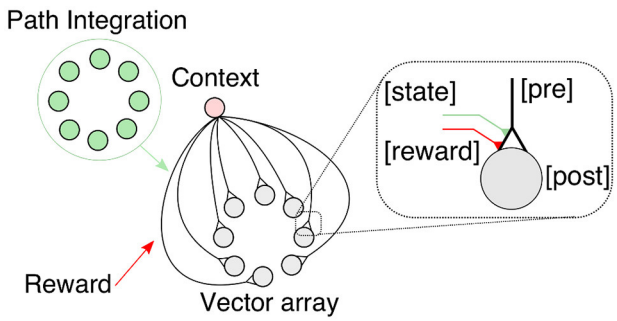


FIGURE 4 | Canonical vector learning rule involves associations of path integration (PI) states with context-dependent and reward signals. Global vector memories are acquired and expressed by this learning circuit. The home vector array activities are associated with the food reward given an active foraging state (outward journey). For details, see text below.

food. Due to the delta rule-like term $x_i^{PI}(t) - x_i^{GV}(t)$, the weights w_i^{GV} approach same values as the activities of the PI state at the rewarding location. Thus, the weights represent the static GV to the rewarding location (feeder). After returning back home, the agent applies the angle θ_{GV} of the GV to navigate toward the feeder using error compensation. The motor signal of the GV:

$$m_{GV}(t) = l_{GV}(t) \sin\big(\theta_{GV}(t) - \phi(t)\big), \tag{17}$$

is applied together with the homing signal m_{HV} and random search m_ε, where l_{GV} is the length of the GV. We model the random search by the agent as a correlated Gaussian random walk, which has been previously used to study animal foraging (Bovet and Benhamou, 1988). Therefore, m_ε is drawn from a Gaussian distribution $\mathcal{N}$ (mean, S.D.):

$$m_\varepsilon(t) \in \mathcal{N}(0, \varepsilon(t)), \tag{18}$$

with an adaptive exploration rate $\varepsilon(t)$ given by:

$$\varepsilon(t) = \sigma(t) \exp\big(-\beta(t)\nu(t)\big), \tag{19}$$

where $\nu(t)$ is an estimate for the average food reward received over time and $\beta(t)$ is the inverse temperature parameter. The exploration rate is thus zero for inward trips, because the agent applies path integration to reach its home position on a straight path. We define ν by the recursive formula:

$$\nu(t) = r(t) + \gamma\nu(t - \Delta t), \tag{20}$$

where $\nu(t)$ is a lowpass filtered signal of the received food reward $r(t)$ with discount factor $\gamma = 0.995$. Convergence of goal-directed behavior is achieved for ε below a critical value, which depends on the choice of β. We assume that ϵ and ν are based on a probability distribution with fixed mean. We derive a gradient rule, which leads to minimization of the Kullback-Leibler divergence between the distribution of $\epsilon(\nu)$ and an optimal exponential distribution (see Supplementary Material for a derivation). The learning rule is given by:

$$\Delta\beta(t) = \mu_\beta\left(\frac{1}{\beta(t)} + \mu_\nu \nu(t)\varepsilon(t)\right), \tag{21}$$

$$\beta(t + \Delta t) = \beta(t) + \Delta\beta(t), \tag{22}$$

where $\mu_\beta = 10^{-6}$ is a global learning rate, $\mu_\nu = 10^2$ is a reward-based learning rate. The adaptation of beta is characterized by small changes scaling with the square root of time, while the term containing $\nu(t)$ allows for exploitation of explored food rewards to further decrease ε through β. In ecological terms, such exploitation of sparse distributed resources is crucial for the survival of an individual as well as the whole colony (Biesmeijer and de Vries, 2001; Wolf et al., 2012; Bolek and Wolf, 2015).

The final motor command Σ in our action selection mechanism is given by the linear combination:

$$\Sigma(t) = (1 - \varepsilon(t))\big(\sigma(t)m_{GV}(t) + m_{HV}(t)\big) + m_\varepsilon(t), \tag{23}$$

where outward trips are controlled by the balance of random walk and global-vector guided navigation depending on the exploration rate ε, while inward trips are controlled solely by the homing signal m_{HV}. The combination of the two sinusoidals is equivalent to a phase vector (phasor) addition resulting in a phasor, which connects the current position of the agent with the learned feeder location (see Supplementary Material for a derivation).

3. RESULTS

Using the proposed model embedded as a closed-loop control into a simulated agent, we carried out several experiments to validate the performance and efficiency in navigating the agent through complex and noisy environments. We will further demonstrate that the generated behaviors not only resemble insect navigational strategies, but can also predict certain observed behavioral parameters of social insects.

3.1. Path Integration (PI) in Noisy Environments

It has been shown, both theoretically and numerically, that PI is inherently prone to error accumulation (Benhamou et al., 1990; Vickerstaff and Cheung, 2010). Studies have focused on analyzing resulting errors from using certain coordinate systems to perform PI (Benhamou et al., 1990; Cheung and Vickerstaff, 2010; Cheung, 2014). Here we apply a system of geocentric static vectors (fixed preferred orientations) and analyze the effect of noise on the resulting error. How can noise be characterized in PI systems? Both artificial and biological systems operate under noisy conditions. Artificial systems, such as robots employ a multitude of sensors which provide noisy measurements, and generate motor outputs that are similarly noisy. Rounding errors in their control systems can be an additional source of noise.

In animals, noise is mainly attributed to random influences on signal processing and transmission in the nervous system, including synaptic release and membrane conductance by ion channels and pumps (see Stein et al., 2005 for review).

In order to validate the accuracy of the PI mechanism, we measure the positional errors of the estimated nest position with respect to the actual position over time. In the following experiments, we averaged positional errors over 1,000 trials with trial duration $T = 1,000$ s (simulation time step $\Delta t = 0.1$ s). In each trial, the agent randomly forages out from the nest and when the trial duration T is reached, the agent switches to the inward state and only applies the path integration mechanism for homing (see **Figure 5A** for example trajectories). After trial duration T, the mean distance of the agent from the nest is 9.3 ± 5.0 m. The radius of the nest the agent has to reach for successful homing is set to 20 cm. **Figure 5B** shows the distribution of positional errors for three different correlated, sensory noise levels (1, 2, and 5%). The distribution of errors follows a two-dimensional Gaussian distribution with mean 0.0 (nest) and width $\langle \delta r \rangle$.

In population coding, neural responses are characterized by correlated or uncorrelated noise (Averbeck et al., 2006, see **Figure 5C** for examples). In the uncorrelated case, fluctuations in one neuron are independent from fluctuations in the other neurons. Correlated noise is described by fluctuations which are similarly expressed across the population activity, and therefore leads to a shift of the observed peak activity. Here, we numerically analyze the effects of correlated and uncorrelated noise on the accuracy of the proposed PI mechanism. Correlated noise is here defined as a shift $\delta\phi$ of the peak activity, i.e., fully correlated noise, such that the compass input to the PI mechanism is given by:

$$\phi_{noisy}(t) = \phi(t) + \delta\phi, \tag{24}$$

where $\delta\phi$ is drawn from a Gaussian distribution $\mathcal{N}(0, 2\pi\,\zeta_{sens})$ with sensory noise level ζ_{sens}. Uncorrelated noise, also referred to as neural noise, is defined by adding fluctuations δx_i^{HD} to the activities of the HD layer, which are drawn from a Gaussian distribution $\mathcal{N}(0, \zeta_{neur})$ with neural noise level ζ_{neur}.

Figure 5D shows the effect of different degrees of sensory noise on the performance of PI for a fixed number of 18 neurons per layer averaged over 1000 trials. For noise levels up to 5% (equal to 18°), the observed mean position error increases only slowly and nonlinearly with values below 0.4 m demonstrating that our PI mechanism is robust for sensory noise up to these levels.

In **Figure 5E**, we show mean position errors for different levels of uncorrelated noise. Similar to sensory noise, the errors first increase slowly and nonlinearly for noise up to 2%, while for noise larger than 5%, errors increase linearly. In comparison with sensory noise levels, uncorrelated noise leads to larger errors due to a more dispersed peak activity. However, for noise levels up to 2%, mean position errors are well below 0.2 m indicating robustness of our PI mechanism with respect to uncorrelated noise. Given this apparent similar nature of correlated and uncorrelated noise, we only applied sensory, correlated noise for the following experiments of this study.

In **Figure 6**, we varied the number of neurons in the circular arrays of the PI mechanism for three different sensory noise level (0, 2, and 5%). Note that the errors for 0% noise arise from the accuracy limit given the number of neurons. While the mean position error is significantly higher for 6 and 9 neurons, it achieves a minimal value for 18 neurons. For larger system sizes, the error only changes minimally. This is again mainly due to the coarse coding of heading directions. Interestingly, the ellipsoid body of the insect central complex contains neurons with 16–32 functional arborization columns (called wedges, see Wolff et al., 2015). The numerical results here might point toward an explanation for this number, which efficiently minimizes the error.

Besides errors resulting from random noise, there are also systematic errors observed in navigating animals. Both invertebrate and vertebrate species exhibit systematic errors in homing behavior after running an L-shaped outward journey (see Etienne and Jeffery, 2004 for review). Müller and Wehner (1988) have examined such errors in desert ants by measuring the angular deviation with respect to the angle of the L-shaped course (see **Figure 7**). In order to show that our mechanism is able to reproduce these errors, we fit our model against the desert ant data from Müller and Wehner (1988) using the leak rate λ (Equation 7) of the PI memory layer as control variable. Using a leak rate of $\lambda \approx 0.0075$ resulted in angular errors most consistent with behavioral data. Leaky integration producing systematic errors is an idea that has been previously proposed (Mittelstaedt and Glasauer, 1991; Vickerstaff and Cheung, 2010). Thus, here our mechanism is not only performing accurately in the presence of random noise, but it also reproduces behavioral aspects observed in animals.

In **Table 1**, we compare the accuracy and efficiency with other state-of-the-art PI models. Haferlach et al. (2007) apply less neurons than our model, but we achieve a better performance in terms of positional accuracy with larger sensory noise (values taken from **Figure 9**). Note that our model achieves similar accuracy, when using six neurons (see **Figure 6**). The model by Kim and Lee (2011) applies 100 neurons per layer leading to a fairly small positional error despite of 10% uncorrelated noise (**Figure 6A**, $N_1 = 100$ neurons). However, both models apply straight paths before homing, which results in smaller path integration errors compared to random foraging as observed in insects. Furthermore, many desert ant species were measured to freely forage average distances of 10–40 m depending on the species (Muser et al., 2005), whereas some individuals travel even up to multiple hundred meters (Buehlmann et al., 2014). Our foraging time has been adjusted for realistic foraging distances, and if we reduce the foraging time in our model, we achieve similarly small positional errors as previous models. Furthermore, behavioral data measured in desert ants (Merkle et al., 2006) revealed that path integration errors are approximately 1–2 m depending on foraging distance. The median values are taken from Figure 3B in Merkle et al. (2006) and reflect the error between the endpoint of an ant's inward run and the correct position of the nest. These larger errors compared

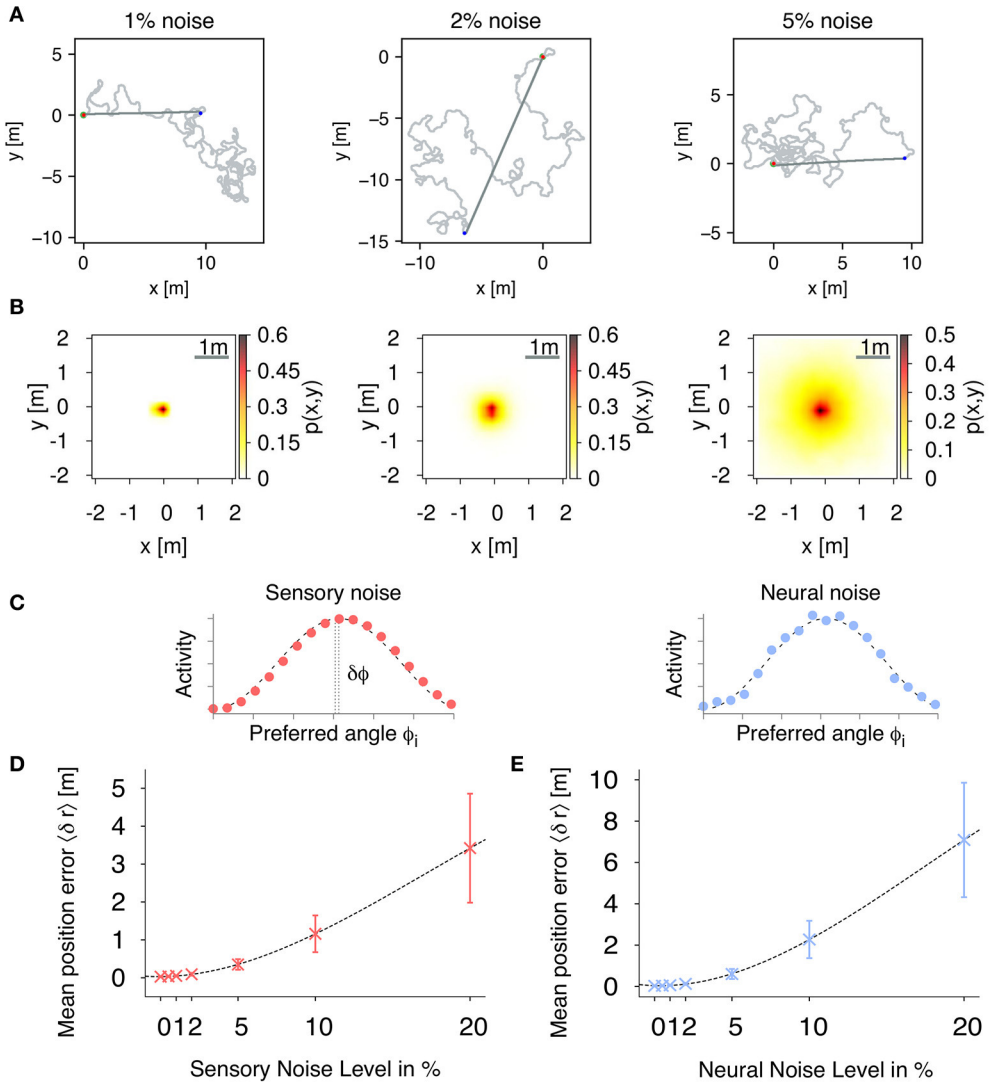


FIGURE 5 | Path integration (PI) accuracy under the influence of external noise. (A) Example trajectories of the simulated agent during random foraging (light gray) and homing behavior (dark gray) for different sensory, correlated noise levels: 1, 2, and 5%. The red point marks the starting point at the nest, and the blue point indicates the return, when the agent switches to its inward state. Using only path integration, the agent successfully navigates back to the nest with a home radius (green circle) of 0.2 m. **(B)** We evaluate the accuracy of the proposed PI mechanism by using the mean positional error averaged over each time step during each trial. Distribution of positional errors for different sensory, correlated noise levels: 1, 2, and 5%. **(C)** Examples of population-coded HD activities with correlated and uncorrelated noise. Filled dots are activities of individual neurons, while the dashed line is a cosine response function. **(D)** Mean position errors $\langle \delta r \rangle$ ($\pm$ S.D.) in PI with respect to fully correlated, sensory noise levels averaged over 1,000 trials (fixed number of 18 neurons per layer). **(E)** Mean position errors $\langle \delta r \rangle$ ($\pm$ S.D.) in PI with respect to uncorrelated, neural noise levels averaged over 1,000 trials (fixed number of 18 neurons per layer).

to model accuracies are likely due to noise accumulation in sensing, neural processing and motor control, although it is difficult to determine an exact quantification. Nonetheless, ants are able to reliably navigate by falling back to other strategies, such as searching behavior or visual homing.

3.2. Global Vector (GV) Learning and Goal-Directed Navigation

In the previous section, we proposed a reward-modulated associative learning rule for GV learning. In order to test the performance of our insect-inspired model applying this learning rule, and to validate the use of learned vector representation in goal-directed navigation, we carried out several experiments under biologically realistic conditions. We apply the PI mechanism with $N = 18$ neurons per layer and a sensory noise level of 5%. In the first series of experiments, a single feeder is placed with a certain distance L_{feed} and angle θ_{feed} to the nest. The agent is initialized at the nest with a random orientation drawn from a uniform distribution on interval $[0, 2\pi)$. In this naïve condition, the agent starts to randomly search in the environment. If the agent is unsuccessful in locating the feeder after a fixed time t_{forage}, it turns inward and performs homing

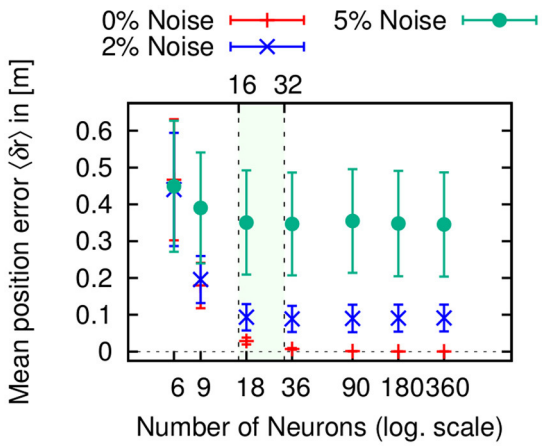


FIGURE 6 | Mean positional errors $\langle \delta r \rangle$ ($\pm$ S.D.) in path integration (PI) with respect to number of neurons per layer averaged over 1,000 trials for three different sensory noise level (0, 2, and 5%). In all three cases, the error reaches a minimum plateau between 16 and 32 neurons (colored area), which corresponds to the number of functional columns in the ellipsoid body of the insect central complex (Wolff et al., 2015).

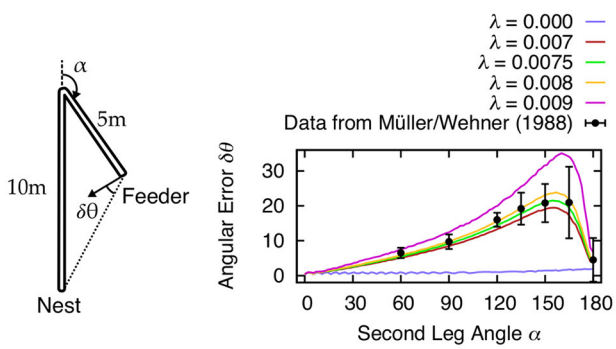


FIGURE 7 | Systematic errors $\delta\theta$ of desert ant homing are reproduced by leaky integration of path segments. Müller and Wehner (1988) tested the ants how accurate they return to the nest after following the two connected, straight channels with 10 and 5 m length to the feeder (sketch modified from Müller and Wehner, 1988). The second channel angle α was varied in 2.5° intervals for the simulation results. In our model, the leak rate λ in the self-recurrent connections is used to fit the behavioral data (Müller and Wehner, 1988). We found that values $\lambda \approx 0.0075$ accurately describe the observed systematic errors in desert ants.

behavior using only the PI mechanism. If the agent however finds the feeder, the current PI state is associated with the received reward, and stored in the weights to the GV array. The agent returns back home after the accumulated reward surpasses a fixed threshold. Each trial lasts a fixed maximum time of $T = \frac{3}{2} t_{forage}$, before the agent is reset to the nest position. On subsequent foraging trips, the agent applies the learned vector representation and navigates along the GV, because the exploration rate is decreased due to the previous reward. If the agent finds the feeder repeatedly, the learned GV stabilizes and the exploration rate decreases further.

Figure 8 demonstrates such an experiment for a feeder with a distance of $L_{feed} = 10$ m and angle $\theta_{feed} = 90°$ from the nest.

TABLE 1 | Comparison of existing path integration (PI) models in terms of accuracy and efficiency.

Model	Neurons	Noise [%]	Error [m]	Foraging dist. [m]
Haferlach et al., 2007	6	3	0.46 ± 0.18	≤ 5
Kim and Lee, 2011	100	10	0.018 ± 0.002	≤ 5
Our model	18	5	0.351 ± 0.140	9.3 ± 5.0
	18	10	1.160 ± 0.484	9.3 ± 5.0
	18	5	0.070 ± 0.037	5 ± 3
Cataglyphis fortis	–	–	median=1.27 ($N = 51$)	5
(Merkle et al., 2006)	–	–	median=2.45 ($N = 53$)	10
	–	–	median=2.47 ($N = 50$)	20

In **Figure 8A**, we show the trajectories of the agent during five trials. The trial numbers are color-coded (see colorbox). During the first trial, the agent has not visited the feeder yet and returns home after $t_{forage} = 2,000$ s of random search. During the second trial (see yellow-colored trajectory), the agent finds the feeder and learns the GV representation from the PI state (see **Figure 8B**). Here the red dotted line indicates the correct angle $\theta_{feed} = 90°$ to the feeder, while the cyan-colored line is the average angle estimated from the synaptic strengths of the GV array. In doing so, the agent is able to acquire an accurate vector representation (**Figure 8B**) resulting in stable trajectories toward the goal for the final three trials, which is again due to a low exploration rate (**Figure 8C**). The repeated visits to the feeder decrease the exploration rate due to the received reward (red line). In the final two trials, the agent navigates to the feeder on a stable trajectory (i.e., low exploration rate) demonstrating that the learning rule is robust for goal-directed navigation in noisy environments. Note, that the reward signal peak is decreased for the final two trials, because the agent does not enter the reward area centrally. Furthermore, switching the context unit to the inbound state is determined by the accumulated amount of reward over time. As such, smaller, but broader reward signals give a similar accumulated reward than a bigger and sharper signal.

In **Figure 9**, we simulated 100 learning cycles with different randomly generated environments, each consisting of 100 consecutive trials. The feeders are randomly placed by sampling from a uniform distribution $\mathcal{U}$ as follows:

$$r_{feed} = (r_{max} - r_{min})\sqrt{n_1} + r_{min}, \tag{25}$$

$$\theta_{feed} = 2\pi n_2, \tag{26}$$

$$n_1, n_2 \in \mathcal{U}(0, 1), \tag{27}$$

where r_{feed} is the distance from the nest to a feeder and θ_{feed} is the angle with respect to the x axis. We chose the $r_{min} = 1$ m and $r_{max} = 40$ m to be the bounds, in which the feeders can be placed. The density is determined by how many feeders will be placed within these bounds. Here, we generated 50 feeders for each environment. In **Figure 9A**, we show the mean exploration rate, and the running averages of mean homing and goal success rates with respect to trials (foraging time $t_{forage} = 1,000$ s, averaged over 100 cycles). Note that the foraging time has been

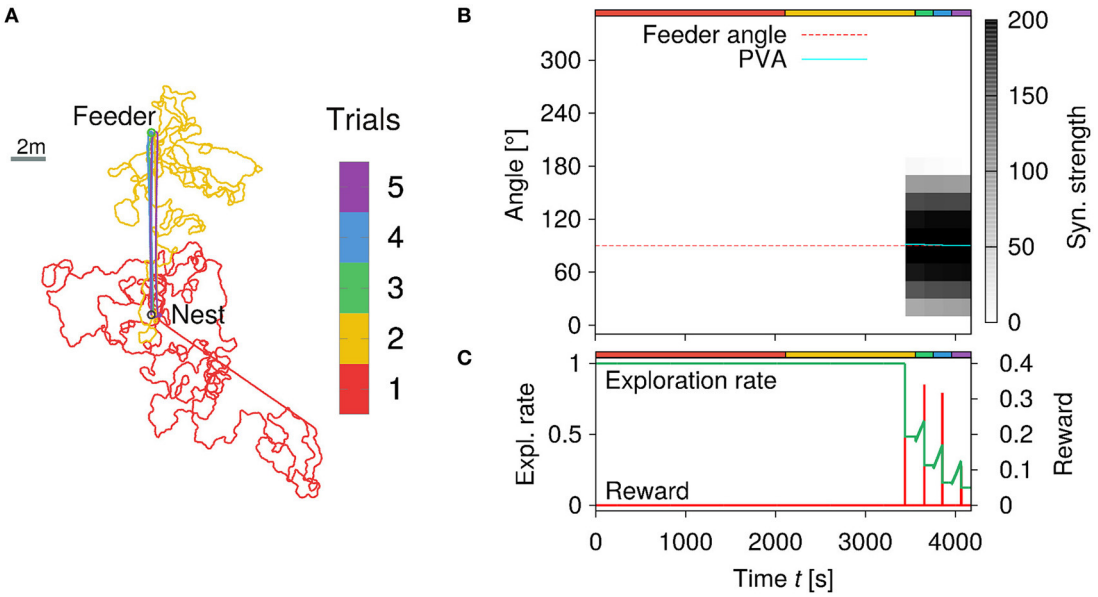


FIGURE 8 | Learning walks of the simulated agent for a feeder placed L_{feed} = 10 m away from the nest. (A) Trajectories of the agent for five trials with a feeder in 10 m distance and 90° angle to the nest. Each trial number is color-coded (see colorbar). Inward runs are characterized by straight paths controlled only by PI. See text for details. **(B)** Synaptic strengths of the GV array changes due to learning over time (of the five trials). The estimated angle θ_{GV} (cyan-colored solid line) to the feeder is given by the position of the maximum synaptic strength. **(C)** Exploration rate and food reward signal with respect to time. The exploration rate decreases as the agent repeatedly visits the feeder and receives reward.

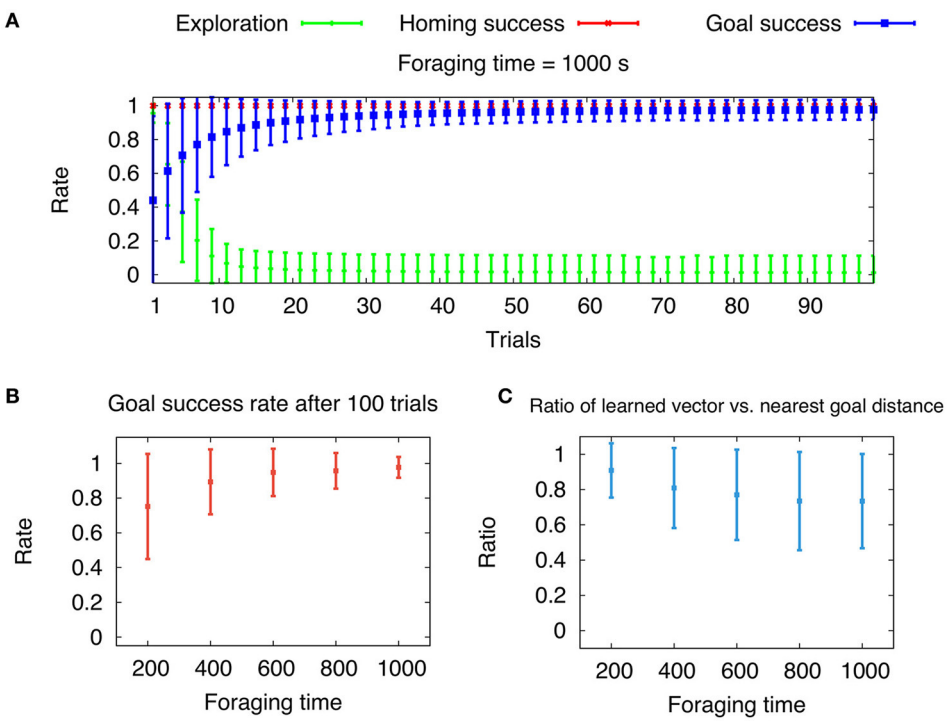


FIGURE 9 | Longer foraging durations during global vector (GV) learning increase the average goal success rate, but decrease the ratio of learned global vector and nearest feeder distance. (A) Mean exploration rate and running mean goal success and homing rate (± S.D.) with respect to trials averaged over 100 cycles of randomly generated environments (foraging time t_{forage} = 1,000 s). Goal success is defined by whether a feeder was visited per trial. The homing rate is determined by the agent's return to the nest within the given total trial duration T. **(B)** Mean goal success rate after 100 trials with respect to foraging time t_{forage} averaged over 100 cycles. **(C)** Mean ratio of learned GV distance and nearest feeder distance with respect to foraging time t_{forage} averaged over 100 cycles.

reduced compared to **Figure 8**, because the random environment contain multiple, not just a single feeder. This leads to a higher probability of finding a feeder and for the learning algorithm to converge. During the 100 trials, learning converges on average within the first 20 trials given by a low mean exploration rate. Like in the previous experiment, the agent reaches the feeder in every trial after convergence is achieved. This is indicated by the goal success approaching one. Average homing success is one for every trial, which results from sufficient searching behavior and the given total time T. The convergence of the learning process is dependent on the foraging time, because longer time allow for longer foraging distances, and thus larger search areas. Therefore, we varied the foraging time $t_{forage} = 200, 400, 600, 800$, and $1,000$ s and measure the mean goal success rate after 100 trials averaged over 100 cycles (**Figure 9B**). Note, that in contrast to naturalistic learning in ants, our agents reduces the exploration rate to zero leading to pure exploitation of the learned global vector. Ants live in environments with rather sparse, dynamic food sources, thus their exploitation of learned vector memories is rather flexible. Nevertheless, our results indicate that for longer foraging times, the mean goal success rate approaches one and its variance decreases. However, by measuring the averaged ratio of learned vector and nearest feeder distance, we show that this ratio decreases for larger foraging times (**Figure 9C**). Thus, there is a trade-off with respect to convergence and reward maximization, leading to an optimal foraging time. Desert ants have been shown to increase their foraging times up to a certain value, after which it saturates (Wehner et al., 2004). This adaptation of foraging time might be indicated by the trade-off resulting from our model. Furthermore, we encourage the reader to see the **Supplementary Video** of path integration and global vector learning performed by a simulated hexapod robot.

4. DISCUSSION

Social insects, such as bees and ants, use PI-based vector memories for guiding navigation in complex environments (Collett et al., 1998, 1999; De Marco and Menzel, 2005; Collett and Collett, 2015). Here, we proposed a novel computational model for combining PI and the acquisition of vector memories in a simulated agent. We have shown that a computational model based on population-coded vector representations can generate efficient and insect-like navigational behaviors in artificial agents. These representations are computed and stored using a simple neural network model combined with reward-modulated associative learning rules. Thus, the proposed model is not only accounting for a number of behavioral aspects of insect navigation, but it further provides insights in possible neural mechanisms in relevant insect brain areas, such as the central complex. In the following, we will discuss certain aspects of our model juxtaposing it with neurobiological findings in insects. Furthermore, we provide comparisons to other state-of-the-art models of vector-guided navigation (Kubie and Fenton, 2009; Cruse and Wehner, 2011).

4.1. Head-Direction (HD) Cells and Path Integration (PI)

A main property of the PI mechanism of our model is that it receives input from a population of neurons, which encode for allothetic compass cues. Here, we apply a cosine response curve for coarse encoding of orientations. Such a mechanism was previously applied by other models (Haferlach et al., 2007; Kim and Lee, 2011). Neurons in the central complex of locusts contain a population-coded representation of allothetic compass cues based on the skylight polarization pattern (Heinze and Homberg, 2007). Similarly, central complex neurons in the *Drosophila* brain encode for heading orientations based on idiothetic self-motion and visual landmarks. Seelig and Jayaraman (2015) measured the fluorescent activity of genetically expressed calcium sensors indicating action potentials, while the fly was tethered on an air-suspended track ball system connected to a panoramic LED display. Any rotation of the fly on the ball is detected and fed back by corresponding motions of the visual scene on the display. The activity of 16 columnar neurons, which display the full circular range, generates a single maximum, which moves according to the turns of the fly on the ball. Interestingly, even though the representation is generated by visual stimuli, it can be accurately maintained solely by self-motion cues over the course of several seconds in the dark. A recent study on dung beetles (el Jundi et al., 2015), which navigate completely unaffected by landmarks, has shown that celestial compass cues are encoded in the central complex revealed by electrophysiological recordings. Taken together, it is likely that the central complex of social insects contains a similar neural coding of polarization- and landmark-based compass cues. Not only is the central complex function and anatomy highly conserved across insect species, but behavioral experiments on ants and bees also suggest the central role of using polarization and landmark cues for navigation. Our model further predicts allothetic goal-direction cues to be involved in PI mechanisms. Such neural representations have yet to be observed in experiments, ideally by applying the tethered track ball setup described in Seelig and Jayaraman (2015). A recent study has developed such a system for the use in desert ants (Dahmen et al., 2017), providing a powerful tool for future investigation of underlying neuronal mechanisms by combining this technology with electrophysiological recordings.

In our model, we assume that the agent's walking speed is neurally encoded as a linear signal that modulates the amplitude of HD activities by an additive gain. A similar, so-called gater mechanism has been applied in a model by Bernardet et al. (2008). Such linear speed signals have recently been found to be encoded by neurons in the rat's medial entorhinal cortex (Kropff et al., 2015) as well as the cockroach central complex (Martin et al., 2015). This shared encoding mechanism indicates the necessity of linear velocity components for accurate PI (Issa and Zhang, 2012). The temporal accumulation of speed-modulated HD signals in our model is achieved by a self-recurrent connection. Biologically, these recurrent connections can be interpreted as positive feedback within a group of neurons with the same preferred direction. Since our model applies PI as a scaffold for spatial learning, we apply this simplified

accumulation mechanism to avoid random drifts observed in more complex attractor networks (Wang, 2001), which were applied in previous PI models (Touretzky et al., 1993; Hartmann and Wehner, 1995). We were also able to test the leaky-integrator hypothesis (Mittelstaedt and Glasauer, 1991) by fitting a single leakage parameter to observed behavioral data from desert ants (Müller and Wehner, 1988). The leakage parameter decreases the self-recurrent connection weight for leaky integration.

A HV representation is computed by using a cosine weight kernel, which was also used in Bernardet et al. (2008). Such a connectivity acts on each represented direction by adding the projections from other directions, respectively. This leads to the formation of an activity pattern with a single maximum across the population. The angle of the represented vector is readout by averaging the population vectors, while the distance is encoded by the amplitude of the population activity. We show that such a readout of a population-coded vector is sufficient to generate robust homing behavior in an artificial agent. Furthermore, it allows for accurate localization required for spatial learning of locations.

The extensive numerical analysis of noise affecting the accuracy of our PI mechanism leads to two predictions. First, PI accuracy seems to follow a similar function with respect to the noise levels for both the fully correlated and uncorrelated random fluctuations. While uncorrelated noise could be further filtered depending on the system size N, decorrelation of sensory input noise could be achieved by adding inhibitory feedback as shown in a model by Helias et al. (2014). Second, we varied the number of neurons N per layer for different levels of fully correlated noise, which predicts an accuracy plateau between 16 and 32 neurons where the accuracy will not increase for larger systems. This indicates that such a number of partitions for representing orientation variables is efficient and accurate enough. Interestingly, most prominent neuropils of the central complex exhibit a similar number of functional columns (Wolff et al., 2015). The central complex has been shown to be involved in sky compass processing (Heinze and Homberg, 2007), spatial orientation (Seelig and Jayaraman, 2015), and spatio-visual memory (Neuser et al., 2008; Ofstad et al., 2011). Its columnar and reverberating connectivity further supports the functional role of integrating orientation stimuli. These evidences suggest that the proposed circular arrays representing navigation vectors might be encoded in the central complex. We conclude that further experiments are needed to unravel how PI is exactly performed in the insect brain by closely linking neural activity and circuitry to behavioral function.

4.2. Reward-Modulated Vector Acquisition and the Role of Motivational Context

PI provides a possible mechanism for self-localization. As such, it has been shown experimentally that social insects apply this mechanism as a scaffold for spatial learning and memory (Collett et al., 2013). Here we propose a reward-modulated associative learning rule (Smith et al., 2008; Cassenaer and Laurent, 2012; Hige et al., 2015) for acquiring and storing vector representations. The acquisition and expression of such vector memories depend on the context during navigation. For GVs, the context is determined by the foraging state, which we model as a binary unit. Indeed, behavioral studies on desert (Wehner et al., 2006) and wood ants (Fernandes et al., 2015) have shown that expression of spatial memories is controlled by an internal state in a binary fashion. The association of the context with a reward signal, received at the feeder, drives synaptic weight changes corresponding to the difference between the current PI state and the respective weight. As this difference is minimized, the weights converge toward values representing the PI state when the reward was received at the feeder. Thus, like the HV, GVs are population-encoded with the angle determined by the position of the maximum activity and the length determined by the amplitude of the activity. To our knowledge, this is the first model that applies such a neural representation to perform vector-guided navigation. Previous models, such as Kubie and Fenton (2009); Cruse and Wehner (2011), do not provide possible underlying neural implementations of the PI-based stored information used for navigation. The HD accumulator model (Kubie and Fenton, 2009) argued that vector information is stored in so-called shortcut matrices, which are subsequently used for navigating toward goals. Similarly, the Cruse and Wehner model (Cruse and Wehner, 2011) stored HVs as geocentric coordinates in the activity of two neurons. Although it has been argued that this representation is biologically plausible, it is unlikely that persistent activity can explain global vector memories which are expressed over several days (Wehner et al., 2004). Furthermore, representing a two-dimensional variable requires at least three neurons, because firing rates are strictly positive. As such, existing models offer sufficient mechanisms in order to generate vector-guided navigation, they neither seem biologically plausible nor provide any explanations how such information is dynamically learned during navigation.

Our proposed encoding of GVs is validated by recent findings from a behavioral study on wood ants (Fernandes et al., 2015). The authors carried out a series of novel experimental paradigms involving training and testing channels. In the training channel, ants were trained to walk from their nest to a feeder at a certain distance, before they were transferred to the testing channel. There, they measured the expression of vector memories by observing the behavior. The authors showed that vector memories are expressed by successful association of direction and distance, therefore such memories might be encoded in a common neural population of the insect brain. The acquisition of vectors were rapid after 4–5 training trials, which corresponds to the rapid vector learning shown by our model during learning walks (**Figure 8**). However, the study mainly examined the expression of homeward vector memories which are not included in our model, because here the agent applies PI for homing. Recent work by Fleischmann et al. (2016) investigates landmark learning and memory during naturalistic foraging in the desert ant species *Cataglyphis fortis*. Like other desert ants, they spent the initial weeks of their lifetime inside the nest, before spending about a week foraging repeatedly for food to bring back to the nest. By placing controlled, prominent landmarks around the nest, the authors could measure the foraging routes of individual, marked ants. They also measured the accuracy

of landmark-guided memories by transferring inward running ants right before they entered the nest. Their results show that ants initially forage only within a short distance and duration, but more experienced foragers increase their average foraging range and duration. Furthermore, they paths become straighter and they are more successful in finding food (also shown in another desert ant species; Wehner et al., 2004). Taken together, their results indicate that landmark learning and memory is a gradual process. Our model does not model landmark guidance during foraging, but it provides a simple strategy that could support this gradual learning mechanism. Specifically, it could provide the agent with a directional bias, by which the agent can learn visual routes toward rewarding food sources (Ardin et al., 2016). Finally, possible interactions between path integration and landmark-based memories has been recently shown in behavioral experiments (Wystrach et al., 2015), and as such, a complete neural model of naturalistic foraging behavior remains to be future work.

Two major higher brain areas in social insects exhibit experience-dependent plasticity due to foraging activity: the mushroom bodies (Yilmaz et al., 2016) and the central complex (Schmitt et al., 2016). The mushroom bodies are paired neuropils known to be involved in olfactory learning and memory (Owald and Waddell, 2015), as well as visual learning in discrimination tasks (Vogt et al., 2014). Studies on the central complex across various insect species have revealed its role in visual object localization (Seelig and Jayaraman, 2013) and visual learning (Liu et al., 2006), motor adaptation (Strauss, 2002), spatio-visual memory (Neuser et al., 2008; Seelig and Jayaraman, 2015; Ofstad et al., 2011), as well as polarization-based compass (Heinze and Homberg, 2007). A common coding principle in the central complex appears to be the topological mapping of stimuli within the full azimuthal circle (Plath and Barron, 2015). Both higher brain neuropils involve the functional diversity of multiple neuropeptides and neurotransmitters (Kahsai et al., 2012). The short neuropeptide F is a likely candidate influencing the foraging state, as it has been shown to regulate feeding behavior and foraging activity after starvation (Kahsai et al., 2010). Based on this evidence, we conclude that the population-coded vector memories described by our model are likely to be found in the central complex. Nonetheless, we do not exclude the possibility of possible interactions between the central complex and the mushroom bodies involved in spatial learning and navigation, which is supported by recent findings on novelty choice behavior in *Drosophila* (Solanki et al., 2015).

We proposed a novel computational model for PI and the acquisition and expression of vector memories in artificial agents. Although existing vertebrate and invertebrate models (Kubie and Fenton, 2009; Cruse and Wehner, 2011) have followed a similar approach of implementing vector-guided navigation, here we provide plausible neural implementations of the underlying control and learning mechanisms. Tested on a simulated agent, we show that the proposed model produces navigational behavior in the context of realistic closed-loop body-environment interactions (Webb, 1995; Seth et al., 2005; Pfeifer et al., 2007). In our previous work, we applied this approach to study adaptive locomotion and climbing (Manoonpong et al., 2013; Goldschmidt et al., 2014; Manoonpong et al., 2014), goal-directed behavior (Dasgupta et al., 2014) and memory-guided decision-making (Dasgupta et al., 2013). Although our model does not reproduce the full repertoire of insect navigation, it has shown to be sufficient in generating robust and efficient vector-guided navigation. Besides behavioral observations, our model also provides predictions about the structure and plasticity of related neural circuits in the insect brain (Haberkern and Jayaraman, 2016). We discussed our findings in the context of neurobiological evidences related to two higher brain areas of insects, the central complex and the mushroom bodies. We therefore conclude that our model offers a novel computational model for studying vector-guided navigation in social insects, which combines neural mechanisms with their generated behaviors. This can guide future behavioral and neurobiological experiments needed to evaluate our findings.

AUTHOR CONTRIBUTIONS

Conceived and designed the experiments: DG, SD, and PM. Performed the experiments: DG. Analyzed the data: DG, SD, and PM. Contributed reagents/materials/analysis tools: DG and SD. Wrote the paper: DG, SD, and PM.

FUNDING

This research was supported by Centre for BioRobotics (CBR) at University of Southern Denmark (SDU, Denmark). DG was supported by the Fundação para a Ciência e Tecnologia (FCT). PM was supported by Bernstein Center for Computational Neuroscience II Göttingen (BCCN grant 01GQ1005A, project D1) and Horizon 2020 Framework Programme (FETPROACT-01-2016—FET Proactive: emerging themes and communities) under grant agreement no. 732266 (Plan4Act). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

ACKNOWLEDGMENTS

We thank Florentin Wörgötter at the Department of Computational Neuroscience in Göttingen, where most of this work was conducted. DG and SD thank Taro Toyoizumi and his lab members at RIKEN BSI for fruitful discussions. We thank James Humble for comments on the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00020/full#supplementary-material

Supplementary Video | Path integration and global vector learning in a simulated hexapod robot.

REFERENCES

Ardin, P., Peng, F., Mangan, M., Lagogiannis, K., and Webb, B. (2016). Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLoS Comput. Biol.* 12:e1004683. doi: 10.1371/journal.pcbi.1004683

Arena, P., Patanè, L., and Termini, P. S. (2014). *A Computational Model for the Insect Brain*. Cham: Springer International Publishing.

Averbeck, B. B., Latham, P. E., and Pouget, A. (2006). Neural correlations, population coding and computation. *Nat. Rev. Neurosci.* 7, 358–366. doi: 10.1038/nrn1888

Benhamou, S., Sauvé, J.-P., and Bovet, P. (1990). Spatial memory in large scale movements: efficiency and limitation of the egocentric coding process. *J. Theor. Biol.* 145, 1–12. doi: 10.1016/S0022-5193(05)80531-4

Bernardet, U., Bermúdez i Badia, S., and Verschure, P. F. M. J. (2008). A model for the neuronal substrate of dead reckoning and memory in arthropods: a comparative computational and behavioral study. *Theor. Biosci.* 127, 163–175. doi: 10.1007/s12064-008-0038-8

Bienenstock, E., Cooper, L., and Munro, P. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2, 32–48.

Biesmeijer, J. C., and de Vries, H. (2001). Exploration and exploitation of food sources by social insect colonies: a revision of the scout-recruit concept. *Behav. Ecol. Sociobiol.* 49, 89–99. doi: 10.1007/s002650000289

Bolek, S., and Wolf, H. (2015). Food searches and guiding structures in north african desert ants, cataglyphis. *J. Comp. Physiol. A* 201, 631–644. doi: 10.1007/s00359-015-0985-8

Bovet, P., and Benhamou, S. (1988). Spatial analysis of animals' movements using a correlated random walk model. *J. Theor. Biol.* 131, 419–433. doi: 10.1016/S0022-5193(88)80038-9

Buehlmann, C., Graham, P., Hansson, B. S., and Knaden, M. (2014). Desert ants locate food by combining high sensitivity to food odors with extensive crosswind runs. *Curr. Biol.* 24, 960–964. doi: 10.1016/j.cub.2014.02.056

Bühlmann, C., Cheng, K., and Wehner, R. (2011). Vector-based and landmark-guided navigation in desert ants inhabiting landmark-free and landmark-rich environments. *J. Exp. Biol.* 214, 2845–2853. doi: 10.1242/jeb.054601

Burak, Y., and Fiete, I. R. (2009). Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput. Biol.* 5:e1000291. doi: 10.1371/journal.pcbi.1000291

Capaldi, E. A., Smith, A. D., Osborne, J. L., Fahrbach, S. E., Farris, S. M., Reynolds, D. R., et al. (2000). Ontogeny of orientation flight in the honeybee revealed by harmonic radar. *Nature* 403, 537–540. doi: 10.1038/35000564

Cassenaer, S., and Laurent, G. (2012). Conditional modulation of spike-timing-dependent plasticity for olfactory learning. *Nature* 482, 47–52. doi: 10.1038/nature10776

Cheng, K., Schultheiss, P., Schwarz, S., Wystrach, A., and Wehner, R. (2014). Beginnings of a synthetic approach to desert ant navigation. *Behav. Process.* 102, 51–61. doi: 10.1016/j.beproc.2013.10.001

Cheung, A. (2014). Animal path integration: a model of positional uncertainty along tortuous paths. *J. Theor. Biol.* 341, 17–33. doi: 10.1016/j.jtbi.2013.09.031

Cheung, A., and Vickerstaff, R. (2010). Finding the way with a noisy brain. *PLoS Comput. Biol.* 6:e1000992. doi: 10.1371/journal.pcbi.1000992

Collett, M. (2012). How navigational guidance systems are combined in a desert ant. *Curr. Biol.* 22, 927–932. doi: 10.1016/j.cub.2012.03.049

Collett, M., and Cardé, R. T. (2014). Navigation: many senses make efficient foraging paths. *Curr. Biol.* 24, R362–R364. doi: 10.1016/j.cub.2014.04.001

Collett, M., Chittka, L., and Collett, T. S. (2013). Spatial memory in insect navigation. *Curr. Biol.* 23, R789–R800. doi: 10.1016/j.cub.2013.07.020

Collett, M., and Collett, T. S. (2009). The learning and maintenance of local vectors in desert ant navigation. *J. Exp. Biol.* 212, 895–900. doi: 10.1242/jeb.024521

Collett, M., Collett, T. S., Bisch, S., and Wehner, R. (1998). Local and global vectors in desert ant navigation. *Nature* 394, 269–272.

Collett, M., Collett, T. S., and Wehner, R. (1999). Calibration of vector navigation in desert ants. *Curr. Biol.* 9, 1031–1034. doi: 10.1016/s0960-9822(99)80451-5

Collett, T., and Collett, M. (2015). Route-segment odometry and its interactions with global path-integration. *J. Comp. Physiol. A* 201, 617–630. doi: 10.1007/s00359-015-1001-z

Collett, T., Collett, M., and Wehner, R. (2001). The guidance of desert ants by extended landmarks. *J. Exp. Biol.* 204, 1635–1639. doi: 10.5167/uzh-690

Cruse, H., and Wehner, R. (2011). No need for a cognitive map: decentralized memory for insect navigation. *PLoS Comput. Biol.* 7:e1002009. doi: 10.1371/journal.pcbi.1002009

Dahmen, H., Wahl, V. L., Pfeffer, S. E., Mallot, H. A., and Wittlinger, M. (2017). Naturalistic path integration of cataglyphis desert ants on an air-cushioned lightweight spherical treadmill. *J. Exp .Biol.* 220, 634–644. doi: 10.1242/jeb.148213

Dasgupta, S., Wörgötter, F., and Manoonpong, P. (2013). Information dynamics based self-adaptive reservoir for delay temporal memory tasks. *Evolv. Syst.* 4, 235–249. doi: 10.1007/s12530-013-9080-y

Dasgupta, S., Wörgötter, F., and Manoonpong, P. (2014). Neuromodulatory adaptive combination of correlation-based learning in cerebellum and reward-based learning in basal ganglia for goal-directed behavior control. *Front. Neural Circ.* 8:126. doi: 10.3389/fncir.2014.00126

De Marco, R., and Menzel, R. (2005). Encoding spatial information in the waggle dance. *J. Exp. Biol.* 208, 3885–3894. doi: 10.1242/jeb.01832

Dubnau, J., and Chiang, A.-S. (2013). Systems memory consolidation in drosophila. *Curr. Opin. Neurobiol.* 23, 84–91. doi: 10.1016/j.conb.2012.09.006

Duer, A., Paffhausen, B. H., and Menzel, R. (2015). High order neural correlates of social behavior in the honeybee brain. *J. Neurosci. Methods* 254, 1–9. doi: 10.1016/j.jneumeth.2015.07.004

el Jundi, B., and Homberg, U. (2012). Receptive field properties and intensity-response functions of polarization-sensitive neurons of the optic tubercle in gregarious and solitarious locusts. *J. Neurophysiol.* 108, 1695–1710. doi: 10.1152/jn.01023.2011

el Jundi, B., Warrant, E. J., Byrne, M. J., Khaldy, L., Baird, E., Smolka, J., et al. (2015). Neural coding underlying the cue preference for celestial orientation. *Proc. Natl. Acad. Sci. U.S.A.* 112, 11395–11400. doi: 10.1073/pnas.1501272112

Etienne, A. S., and Jeffery, K. J. (2004). Path integration in mammals. *Hippocampus* 14, 180–192. doi: 10.1002/hipo.10173

Eurich, C. W., and Schwegler, H. (1997). Coarse coding: calculation of the resolution achieved by a population of large receptive field neurons. *Biol. Cybernet.* 76, 357–363. doi: 10.1007/s004220050349

Evangelista, C., Kraft, P., Dacke, M., Labhart, T., and Srinivasan, M. V. (2014). Honeybee navigation: critically examining the role of the polarization compass. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 369:20130037. doi: 10.1098/rstb.2013.0037

Fernandes, A. S. D., Philippides, A., Collett, T. S., and Niven, J. E. (2015). The acquisition and expression of memories of distance and direction in navigating wood ants. *J. Exp. Biol.* 218, 3580–3588. doi: 10.1242/jeb.125443

Fleischmann, P. N., Christian, M., Müller, V. L., Rössler, W., and Wehner, R. (2016). Ontogeny of learning walks and the acquisition of landmark information in desert ants, *Cataglyphis fortis*. *J. Exp. Biol.* 219, 3137–3145. doi: 10.1242/jeb.140459

Friedrich, R. W., and Stopfer, M. (2001). Recent dynamics in olfactory population coding. *Curr. Opin. Neurobiol.* 11, 468–474. doi: 10.1016/S0959-4388(00)00236-1

Gaussier, P., Joulain, C., Banquet, J. P., Leprêtre, S., and Revel, A. (2000). The visual homing problem: an example of robotics/biology cross fertilization. *Robot. Auton. Syst.* 30, 155–180. doi: 10.1016/S0921-8890(99)00070-6

Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2014). Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots. *Front. Neurorobot.* 8:3. doi: 10.3389/fnbot.2014.00003

Graham, P., and Cheng, K. (2009). Ants use the panoramic skyline as a visual cue during navigation. *Curr. Biol.* 19, R935–R937. doi: 10.1016/j.cub.2009.08.015

Haberkern, H., and Jayaraman, V. (2016). Studying small brains to understand the building blocks of cognition. *Curr. Opin. Neurobiol.* 37, 59–65. doi: 10.1016/j.conb.2016.01.007

Haferlach, T., Wessnitzer, J., Mangan, M., and Webb, B. (2007). Evolving a neural model of insect path integration. *Adapt. Behav.* 15, 273–287. doi: 10.1177/1059712307082080

Hartmann, G., and Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biol. Cybernet.* 73, 483–497. doi: 10.1007/bf00199541

Heinze, S., Gotthardt, S., and Homberg, U. (2009). Transformation of polarized light information in the central complex of the locust. *J. Neurosci.* 29, 11783–11793. doi: 10.1523/JNEUROSCI.1870-09.2009

Heinze, S., and Homberg, U. (2007). Maplike representation of celestial e-vector orientations in the brain of an insect. *Science* 315, 995–997. doi: 10.1126/science.1135531

Heinze, S., and Homberg, U. (2009). Linking the input to the output: new sets of neurons complement the polarization vision network in the locust central complex. *J. Neurosci.* 29, 4911–4921. doi: 10.1523/JNEUROSCI.0332-09.2009

Helias, M., Tetzlaff, T., and Diesmann, M. (2014). The correlation structure of local neuronal networks intrinsically results from recurrent dynamics. *PLoS Comput. Biol.* 10:e1003428. doi: 10.1371/journal.pcbi.1003428

Hige, T., Aso, Y., Modi, M. N., Rubin, G. M., and Turner, G. C. (2015). Heterosynaptic plasticity underlies aversive olfactory learning in drosophila. *Neuron* 88, 985–998. doi: 10.1016/j.neuron.2015.11.003

Hoinville, T., Wehner, R., and Cruse, H. (2012). "Learning and retrieval of memory elements in a navigation task," in *Biomimetic and Biohybrid Systems*, Vol. 7375, *Lecture Notes in Computer Science*, eds T. Prescott, N. Lepora, A. Mura, and P. Verschurepages (Berlin; Heidelberg: Springer), 120–131.

Homberg, U., Heinze, S., Pfeiffer, K., Kinoshita, M., and el Jundi, B. (2011). Central neural coding of sky polarization in insects. *Philoso. Trans. R. Soc. Lond. B Biol. Sci.* 366, 680–687. doi: 10.1098/rstb.2010.0199

Issa, J. B., and Zhang, K. (2012). Universal conditions for exact path integration in neural systems. *Proc. Natl. Acad. Sci. U.S.A.* 109, 6716–6720. doi: 10.1073/pnas.1119880109

Jauffret, A., Cuperlier, N., and Gaussier, P. (2015). From grid cells and visual place cells to multimodal place cell: a new robotic architecture. *Front. Neurorobot.* 9:1. doi: 10.3389/fnbot.2015.00001

Kahsai, L., Carlsson, M., Winther, Å., and Nässel, D. (2012). Distribution of metabotropic receptors of serotonin, dopamine, gaba, glutamate, and short neuropeptide f in the central complex of drosophila. *Neuroscience* 208, 11–26. doi: 10.1016/j.neuroscience.2012.02.007

Kahsai, L., Martin, J.-R., and Winther, Å. M. E. (2010). Neuropeptides in the drosophila central complex in modulation of locomotor behavior. *J. Exp. Biol.* 213, 2256–2265. doi: 10.1242/jeb.043190

Kim, D., and Lee, J. (2011). Path integration mechanism with coarse coding of neurons. *Neural Process. Lett.* 34, 277–291. doi: 10.1007/s11063-011-9198-5

Kropff, E., Carmichael, J. E., Moser, M.-B., and Moser, E. I. (2015). Speed cells in the medial entorhinal cortex. *Nature*. 523, 419–424 doi: 10.1038/nature14622

Kubie, J. L., and Fenton, A. A. (2009). Heading-vector navigation based on head-direction cells and path integration. *Hippocampus* 19, 456–479. doi: 10.1002/hipo.20532

Lambrinos, D., Kobayashi, H., Pfeifer, R., Maris, M., Labhart, T., and Wehner, R. (1997). An autonomous agent navigating with a polarized light compass. *Adapt. Behav.* 6, 131–161. doi: 10.1177/105971239700600104

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., and Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robot. Auton. Syst.* 30, 39–64. doi: 10.1016/S0921-8890(99)00064-0

Lebhardt, F., Koch, J., and Ronacher, B. (2012). The polarization compass dominates over idiothetic cues in path integration of desert ants. *J. Exp. Biol.* 215, 526–535. . doi: 10.1242/jeb.060475

Liu, G., Seiler, H., Wen, A., Zars, T., Ito, K., Wolf, R., et al. (2006). Distinct memory traces for two visual features in the drosophila brain. *Nature* 439, 551–556. doi: 10.1038/nature04381

Madl, T., Chen, K., Montaldi, D., and Trappl, R. (2015). Computational cognitive models of spatial memory in navigation space: a review. *Neural Netw.* 65, 18–43. doi: 10.1016/j.neunet.2015.01.002

Mangan, M., and Webb, B. (2012). Spontaneous formation of multiple routes in individual desert ants (*Cataglyphis velox*). *Behav. Ecol.* 23, 944–954. doi: 10.1093/beheco/ars051

Manoonpong, P., Dasgupta, S., Goldschmidt, D., and Wörgötter, F. (2014). "Reservoir-based online adaptive forward models with neural control for complex locomotion in a hexapod robot," in *2014 International Joint Conference on Neural Networks (IJCNN)* (Beijing), 3295–3302.

Manoonpong, P., Parlitz, U., and Wörgötter, F. (2013). Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines. *Front. Neural Circ.* 7:12. doi: 10.3389/fncir.2013.00012

Martin, J. P., Guo, P., Mu, L., Harley, C. M., and Ritzmann, R. E. (2015). Central-complex control of movement in the freely walking cockroach. *Curr. Biol.* 25, 2795–2803. doi: 10.1016/j.cub.2015.09.044

Mathews, Z., Lechón, M., Calvo, J., Dhir, A., Duff, A., Bermúdez i Badia, S., et al. (2009). "Insect-like mapless navigation based on head direction cells and contextual learning using chemo-visual sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009* (St. Louis, MO), 2243–2250.

Menzel, R., Greggers, U., Smith, A., Berger, S., Brandt, R., Brunke, S., et al. (2005). Honey bees navigate according to a map-like spatial memory. *Proc. Natl. Acad. Sci. U.S.A.* 102, 3040–3045. doi: 10.1073/pnas.0408550102

Merkle, T., Knaden, M., and Wehner, R. (2006). Uncertainty about nest position influences systematic search strategies in desert ants. *J. Exp. Biol.* 209, 3545–3549. doi: 10.1242/jeb.02395

Mittelstaedt, M.-L., and Glasauer, S. (1991). Idiothetic navigation in gerbils and humans. *Zoologis. Jahrbuch Physiol.* 95, 427–435.

Müller, M., and Wehner, R. (1988). Path integration in desert ants, cataglyphis fortis. *Proc. Natl. Acad. Sci. U.S.A.* 85, 5287–5290. doi: 10.1073/pnas.85.14.5287

Muser, B., Sommer, S., Wolf, H., and Wehner, R. (2005). Foraging ecology of the thermophilic australian desert ant, melophorus bagoti. *Aust. J. Zool.* 53, 301–311. doi: 10.1071/ZO05023

Narendra, A., Gourmaud, S., and Zeil, J. (2013). Mapping the navigational knowledge of individually foraging ants, myrmecia croslandi. *Proc. R. Soc. Lond. B Biol. Sci.* 280:20130683. doi: 10.1098/rspb.2013.0683

Neuser, K., Triphan, T., Mronz, M., Poeck, B., and Strauss, R. (2008). Analysis of a spatial orientation memory in drosophila. *Nature* 453, 1244–1247. doi: 10.1038/nature07003

Ofstad, T. A., Zuker, C. S., and Reiser, M. B. (2011). Visual place learning in drosophila melanogaster. *Nature* 474, 204–207. doi: 10.1038/nature10131

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *J. Math. Biol.* 15, 267–273. doi: 10.1007/BF00275687

Olshausen, B. A., and Field, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vis. Res.* 37, 3311–3325.

Owald, D., and Waddell, S. (2015). Olfactory learning skews mushroom body output pathways to steer behavioral choice in drosophila. *Curr. Opin. Neurobiol.* 35, 178–184. doi: 10.1016/j.conb.2015.10.002

Pfeffer, S., Bolek, S., Wolf, H., and Wittlinger, M. (2015). Nest and food search behaviour in desert ants, cataglyphis: a critical comparison. *Anim. Cogn.* 18, 885–894. doi: 10.1007/s10071-015-0858-0

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Plath, J. A., and Barron, A. B. (2015). Current progress in understanding the functions of the insect central complex. *Curr. Opin. Insect Sci.* 12, 11–18. doi: 10.1016/j.cois.2015.08.005

Salinas, E., and Abbott, L. (1995). Transfer of coded information from sensory to motor networks. *J. Neurosci.* 15, 6461–6474.

Schmid-Hempel, P. (1984). Individually different foraging methods in the desert ant cataglyphis bicolor (hymenoptera, formicidae). *Behav. Ecol. Sociobiol.* 14, 263–271. doi: 10.1007/BF00299497

Schmitt, F., Stieb, S. M., Wehner, R., and Rössler, W. (2016). Experience-related reorganization of giant synapses in the lateral complex: potential role in plasticity of the sky-compass pathway in the desert ant *Cataglyphis fortis*. *Dev. Neurobiol.* 76, 390–404. doi: 10.1002/dneu.22322

Schmolke, A., Mallot, H., and Neurowissenschaft, K. (2002). "Polarization compass for robot navigation," in *The Fifth German Workshop on Artificial Life* (Lübeck), 163–167.

Seelig, J. D., and Jayaraman, V. (2013). Feature detection and orientation tuning in the *Drosophila* central complex. *Nature* 503, 262–266. doi: 10.1038/nature12601

Seelig, J. D., and Jayaraman, V. (2015). Neural dynamics for landmark orientation and angular path integration. *Nature* 521, 186–191. doi: 10.1038/nature14446

Seth, A. K., Sporns, O., and Krichmar, J. L. (2005). Neurorobotic models in neuroscience and neuroinformatics. *Neuroinformatics* 3, 167–170. doi: 10.1385/NI:3:3:167

Smith, D., Wessnitzer, J., and Webb, B. (2008). A model of associative learning in the mushroom body. *Biol. Cybernet.* 99, 89–103. doi: 10.1007/s00422-008-0241-1

Solanki, N., Wolf, R., and Heisenberg, M. (2015). Central complex and mushroom bodies mediate novelty choice behavior in drosophila. *J. Neurogenet.* 29, 30–37. doi: 10.3109/01677063.2014.1002661

Srinivasan, M. (2015). Where paths meet and cross: navigation by path integration in the desert ant and the honeybee. *J. Comp. Physiol. A* 201, 533–546. doi: 10.1007/s00359-015-1000-0

Stein, R. B., Gossen, E. R., and Jones, K. E. (2005). Neuronal variability: noise or part of the signal? *Nat. Rev. Neurosci.* 6, 389–397. doi: 10.1038/nrn1668

Strauss, R. (2002). The central complex and the genetic dissection of locomotor behaviour. *Curr. Opin. Neurobiol.* 12, 633–638. doi: 10.1016/S0959-4388(02)00385-9

Taube, J., Muller, R., and Ranck, J. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. I. description and quantitative analysis. *J. Neurosci.* 10, 420–435.

Todorov, E., and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* 5, 1226–1235. doi: 10.1038/nn963

Touretzky, D., Redish, A., and Wan, H. (1993). Neural representation of space using sinusoidal arrays. *Neural Comput.* 5, 869–884. doi: 10.1162/neco.1993.5.6.869

Vickerstaff, R. J. (2007). *Evolving Dynamical System Models of Path Integration*. Ph.D. thesis, University of Sussex.

Vickerstaff, R. J., and Cheung, A. (2010). Which coordinate system for modelling path integration? *J. Theor. Biol.* 263, 242–261. doi: 10.1016/j.jtbi.2009.11.021

Vogt, K., Schnaitmann, C., Dylla, K. V., Knapek, S., Aso, Y., Rubin, G. M., et al. (2014). Shared mushroom body circuits underlie visual and olfactory memories in Drosophila. *eLife* 3:e02395. doi: 10.7554/eLife.02395

Wang, X.-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends Neurosci.* 24, 455–463. doi: 10.1016/S0166-2236(00)01868-3

Webb, B. (1995). Moving the frontiers between robotics and biology using robots to model animals: a cricket test. *Robot. Auton. Syst.* 16, 117–134. doi: 10.1016/0921-8890(95)00044-5

Wehner, R. (2003). Desert ant navigation: how miniature brains solve complex tasks. *J. Comp. Physiol. A* 189, 579–588. doi: 10.1007/s00359-003-0431-1

Wehner, R., Boyer, M., Loertscher, F., Sommer, S., and Menzi, U. (2006). Ant navigation: one-way routes rather than maps. *Curr. Biol.* 16, 75–79. doi: 10.1016/j.cub.2005.11.035

Wehner, R., Meier, C., and Zollikofer, C. (2004). The ontogeny of foraging behaviour in desert ants, cataglyphis bicolor. *Ecol. Entomol.* 29, 240–250. doi: 10.1111/j.0307-6946.2004.00591.x

Weir, P. T., and Dickinson, M. H. (2015). Functional divisions for visual processing in the central brain of flying *Drosophila*. *Proce. Natl. Acad. Sci. U.S.A.* 112, E5523–E5532. doi: 10.1073/pnas.1514415112

Wittlinger, M., Wehner, R., and Wolf, H. (2006). The ant odometer: stepping on stilts and stumps. *Science* 312, 1965–1967. doi: 10.1126/science.1126912

Wittlinger, M., Wehner, R., and Wolf, H. (2007). The desert ant odometer: a stride integrator that accounts for stride length and walking speed. *J. Exp. Biol.* 210, 198–207. doi: 10.1242/jeb.02657

Wittmann, T., and Schwegler, H. (1995). Path integration – a network model. *Biol. Cybernet.* 73, 569–575. doi: 10.1007/BF00199549

Wolf, H., Wittlinger, M., and Bolek, S. (2012). Re-visiting of plentiful food sources and food search strategies in desert ants. *Front. Neurosci.* 6:102. doi: 10.3389/fnins.2012.00102

Wolff, T., Iyer, N. A., and Rubin, G. M. (2015). Neuroarchitecture and neuroanatomy of the drosophila central complex: a gal4-based dissection of protocerebral bridge neurons and circuits. *J. Comp. Neurol.* 523, 997–1037. doi: 10.1002/cne.23705

Wystrach, A., Dewar, A. D., and Graham, P. (2014). Insect vision: emergence of pattern recognition from coarse encoding. *Curr. Biol.* 24, R78–R80. doi: 10.1016/j.cub.2013.11.054

Wystrach, A., Mangan, M., and Webb, B. (2015). Optimal cue integration in ants. *Proc. R. Soc. Lond. B Biol. Sci.* 282:20151484. doi: 10.1098/rspb.2015.1484

Yilmaz, A., Lindenberg, A., Albert, S., Grübel, K., Spaethe, J., Rössler, W., et al. (2016). Age-related and light-induced plasticity in opsin gene expression and in primary and secondary visual centers of the nectar-feeding ant *Camponotus rufipes*. *Dev. Neurobiol.* 76, 1041–1057. doi: 10.1002/dneu.22374

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Goldschmidt, Manoonpong and Dasgupta. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 18 October 2016
doi: 10.3389/fnbot.2016.00012

Closed-loop Robots Driven by Short-Term Synaptic Plasticity: Emergent Explorative vs. Limit-Cycle Locomotion

Laura Martin, Bulcsú Sándor and Claudius Gros*

Institute for Theoretical Physics, Goethe University Frankfurt, Frankfurt am Main, Germany

We examine the hypothesis, that short-term synaptic plasticity (STSP) may generate self-organized motor patterns. We simulated sphere-shaped autonomous robots, within the LPZRobots simulation package, containing three weights moving along orthogonal internal rods. The position of a weight is controlled by a single neuron receiving excitatory input from the sensor, measuring its actual position, and inhibitory inputs from the other two neurons. The inhibitory connections are transiently plastic, following physiologically inspired STSP-rules. We find that a wide palette of motion patterns are generated through the interaction of STSP, robot, and environment (closed-loop configuration), including various forward meandering and circular motions, together with chaotic trajectories. The observed locomotion is robust with respect to additional interactions with obstacles. In the chaotic phase the robot is seemingly engaged in actively exploring its environment. We believe that our results constitute a concept of proof that transient synaptic plasticity, as described by STSP, may potentially be important for the generation of motor commands and for the emergence of complex locomotion patterns, adapting seamlessly also to unexpected environmental feedback. We observe spontaneous and collision induced mode switchings, finding in addition, that locomotion may follow transiently limit cycles which are otherwise unstable. Regular locomotion corresponds to stable limit cycles in the sensorimotor loop, which may be characterized in turn by arbitrary angles of propagation. This degeneracy is, in our analysis, one of the drivings for the chaotic wandering observed for selected parameter settings, which is induced by the smooth diffusion of the angle of propagation.

Keywords: closed-loop robots, short-term synaptic plasticity, limit cycles, sensorimotor loop, self-organized locomotion, compliant robot

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark, Denmark

Reviewed by:
Malte Schilling, Bielefeld University, Germany
Ralf Der, Max Planck Institute for Mathematics, Germany
Georg Martius, Institute of Science and Technology Austria, Austria
Keyan Ghazi-Zahedi, Max Planck Institute for Mathematics in the Sciences, Germany

****Correspondence:***
Bulcsú Sándor
sandor@itp.uni-frankfurt.de

Received: *09 August 2016*
Accepted: *03 October 2016*
Published: *18 October 2016*

Citation:
Martin L, Sándor B and Gros C (2016) Closed-loop Robots Driven by Short-Term Synaptic Plasticity: Emergent Explorative vs. Limit-Cycle Locomotion. Front. Neurorobot. 10:12. doi: 10.3389/fnbot.2016.00012

1. INTRODUCTION

It has been argued (Pfeifer et al., 2007; Aguilar et al., 2016) that "robophysics," defined as the pursuit of the discovery of biologically inspired principles of self generated motion, may constitute a promising road for eventually achieving life-like locomotor abilities. Distinct principles such as predictive information (Ay et al., 2008), surprise minimization (Friston, 2011), chaos control (Steingrube et al., 2010), empowerment (Salge et al., 2014), homeokinesis (Der and Martius, 2012), cheap design (Montúfar et al., 2015), and curiosity (Frank et al., 2014) have been studied in this

context. Behavior, resulting from guided self organization (Prokopenko, 2009) or autonomous adaption (Chiel and Beer, 1997), may be generated in addition through suitable synaptic (Der and Martius, 2015; Der, 2016) and intrinsic (Sándor et al., 2015) plasticity rules.

Here we point out, that complex dynamics may be generated through a transient plasticity mechanism widely present in the brain. Short-term synaptic plasticity (STSP) (Fioravante and Regehr, 2011; Regehr, 2012) is an activity induced transient modulation of the synaptic efficiency, which may lead either to facilitating or to depressing behavior lasting from a few hundred to a few thousand milliseconds. STSP has been argued, besides others, to be relevant or causal for working memory (Barak and Tsodyks, 2014), for the facilitation of time sequences of alternating neural populations (Carrillo-Reid et al., 2015), for motor control in general (Nadim and Manor, 2000), and for the sculpting of rhythmic motor patterns (Jia and Parker, 2016) in particular. Plasticity mechanisms similar to STSP have also been shown to allow for stable gaits (Toutounji and Pasemann, 2014) in neural networks which are distinctively simpler than the ones used conventionally for bio-inspired controllers (Schilling et al., 2013).

In this study we use the LPZRobots physics simulation package (Der and Martius, 2012) for the investigation of the spherical three-axis robot illustrated in **Figure 1**. This robot is driven exclusively by STSP, with locomotion coming to a stillstand both in the absence of synaptic plasticity and when the feedback from the environment is cut off, e.g., when the gravitational constant is set to zero. We find a surprisingly large palette of self-organized motion primitives, which includes a chaotic phase. The locomotion observed is flexible, in all modes, readjusting seamlessly to disturbances like the collision of the robot with obstacles.

The capability of STSP to have a large impact on locomotion can be traced back in our analysis to the destabilizing effect short-term synaptic plasticity may have on attracting states of the controlling network, inducing attractor-to-attractor transitions within timescales of the order of a few hundred milliseconds. We corroborate this findings by short-circuiting the sensori-motor loop, viz by taking out the environment. Transitions between distinct limit cycles within the full sensori-motor loop are found in addition in the chaotic mode.

2. MATERIALS AND METHODS

2.1. Tsodyks-Markram Model with Full Depletion

The way neurotransmitters are released through the synaptic cleft may change transiently upon repeated presynaptic activity (Tsodyks and Markram, 1997), both for excitatory (Wang et al., 2006) and for inhibitory (Gupta et al., 2000) synapses. Physiologically this is, on the one side due to an increase of the Ca-concentration $u \in [1, U_{max}]$ within the presynaptic bulge, facilitating the release of the respective neurotransmitter, and, on the other side, due to the decrease of the number $\varphi \in [0, 1]$ of available vesicles of neurotransmitters. We use here with

$$\begin{aligned} \dot{u} &= \frac{U(y) - u}{T_u}, & U(y) &= 1 + (U_{max} - 1)y \\ \dot{\varphi} &= \frac{\Phi(u, y) - \varphi}{T_\varphi}, & \Phi(u, y) &= 1 - \frac{uy}{U_{max}} \end{aligned} \tag{1}$$

a modified version of the original Tsodyks-Markram model (Tsodyks and Markram, 1997; Hennig, 2013), in which the the Ca-concentration u and the number of vesicles φ of a given synapse relax to target values $U = U(y)$ and $\Phi = \Phi(u, y)$, determined in turn by the level $y \in [0, 1]$ of the presynaptic activity. A prolonged maximal presynaptic activity $y \equiv 1$ would lead with $\varphi \to 0$ to a full depletion of the reservoir of vesicles.

The dynamics of the full depletion model (1) is determined by the relaxation time constants T_u and T_φ, and by the maximal level U_{max} of the Ca concentration. For $U_{max} = 1$ a monotone depression is present, whereas $U_{max} > 1$ initially generates facilitation by a fast calcium influx, being annulled later on by the depletion of neurotransmitters. Overall, the synaptic efficiency is proportional to $u\varphi$, viz to the number of vesicles and to the release probability (which in turn is assumed to be proportional to u). We use $T_u = 300$ ms and $T_\varphi = 600$ ms, together with either

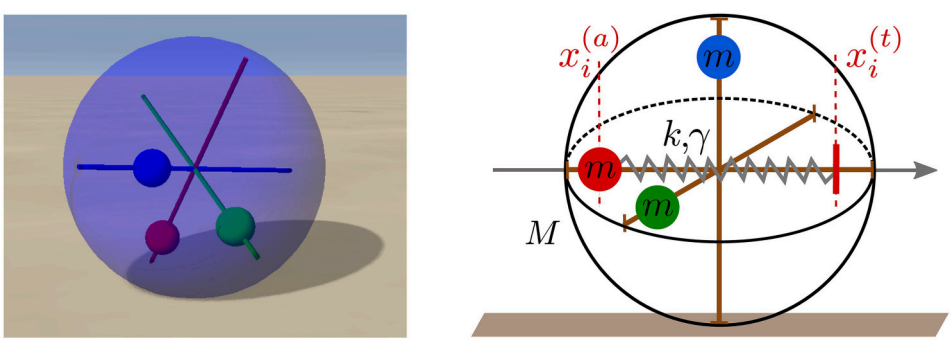


FIGURE 1 | Left: A snapshot of the spherical robot from the LPZRobots simulation environment (Martius et al., 2013). The three weights (red, green, and blue) can move along the respective rods without interference. **Right**: A sketch of the robot with the three perpendicular rods together with the three weights of mass m. The red vertical dashed lines show the actual position $x_i^{(a)}$ and a putative target position $x_i^{(t)}$ of the red weight along its rod. A damped spring with spring constant k and damping γ then pulls the weight toward the target position, which is given in turn by the output of a controlling neuron (compare **Figure 2**).

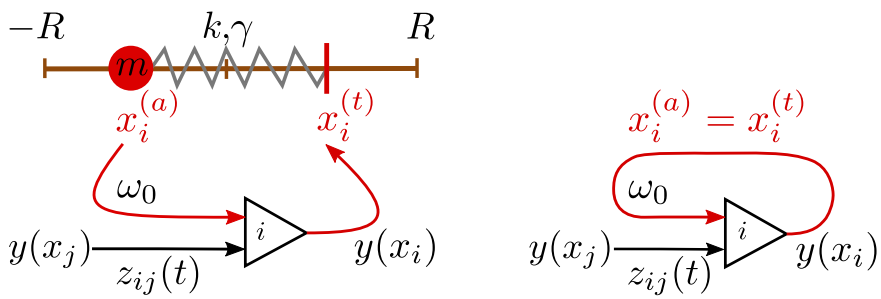


FIGURE 2 | Left: Sketch of the sensorimotor loop of the three-axis spherical robot illustrated in Figure 1. The three weights $i = 1, 2, 3$ with masses m are each controlled by a single neuron. The excitatory input $w_0(x_i^{(a)} + pR)/(2pR)$ of neuron i is proportional to the proprio-sensory measurement of the actual position $x_i^{(a)} \in [-R, R]$ of the i-th mass ($p \in [0, 1]$). The neuron also receives inhibitory inputs $-z_0\varphi_j u_j y(x_j)$ from the other two neurons ($j \neq i$). The output $y(x_i)$ of the i-th neuron determines via $x_i^{(t)} = pR[2y(x_i) - 1]$ the target position of the i-th mass. **Right**: A network of (three) neurons having the identical topology as the one of the three-axis spherical robot, but with the feedback of the environment short-cut by identifying the actual position $x_i^{(a)}$ with the target position $x_i^{(t)}$.

$U_{max} = 1$ or $U_{max} = 4$. These values are within the typical range of what is physiologically observed (Gupta et al., 2000; Wang et al., 2006).

2.2. The Robot

The movement of robot illustrated in **Figure 1** is induced by the relative gravitational pull of the three weights, together with the rolling friction and angular momentum conservation. The individual neurons $i = 1, 2, 3$ are modeled as rate-encoding leaky integrators,

$$\dot{x}_i = -\Gamma x_i + \frac{w_0}{2pR}\left(x_i^{(a)} + pR\right) - z_0 \sum_{j \neq i} u_j \varphi_j y(x_j),$$
$$y(x_j) = \frac{1}{1 + \exp(-a x_j)}, \tag{2}$$

where x_i and $y(x_i)$ are the respective membrane potentials and firing rates. Γ is the relaxation rate, R the diameter of the robot, $p \in [0, 1]$ a rescaling factor, $x_i^{(a)} \in [-R, R]$ the sensory reading of the actual position of the weight on the rod, $w_0 > 0$ the weight of excitatory input and $z_0 > 0$ the magnitude of the inter-neural inhibitory connections. We note that the variables of the STSP, u_j and φ_j, as described by Equation (1), depend only on the presynaptic activity and can hence be attributed altogether to the presynaptic neuron. For the slope of the sigmoidal $a = 0.4$ has been selected. The weight of the excitatory input w_0 is not modulated here by short-term synaptic plasticity, corresponding to a direct sensory reading.

We selected with $p = 1/2$ a reduced range for the target position $x_i^{(t)}$,

$$x_i^{(t)} = pR\left[2y(x_i) - 1\right], \qquad x_i^{(t)} \in [-pR, pR]. \tag{3}$$

This choice allows to avoid dynamic overshooting of the weight when accelerated from its actual to the target position. The force accelerating the weight is calculated by the LPZRobots package by simulating a damped spring:

$$m\ddot{x}_i^{(a)} = -k(x_i^{(a)} - x_i^{(t)}) - \gamma \frac{d(x_i^{(a)} - x_i^{(t)})}{dt} + F_i, \quad x_i^{(a)} \to x_i^{(t)}, \tag{4}$$

where k is the spring constant and γ the damping. Centrifugal and other induced forces, F_i, act additionally in Equation (4) on the individual weights. The complete setup of the three-neuron network is illustrated in **Figure 2**.

2.3. Simulation parameters

The LPZRobots simulation environment (Der and Martius, 2012) is an interactive simulator based on the ODE (Open Dynamic Engine) (Smith, 2005). LPZRobots contains rigid body dynamics in terms of a library of basic primitive objects, such as spheres and cuboids, as well as a variety of joints, sensors and surface materials.

We used *roughness* = 0.8, *slip* = 0.01, *hardness* = 40 and *elasiticity* = 0.5 for the collision and friction properties together with *friction* = 0.3 (the rolling friction coefficient), *gravity* = −9.81 (the gravitational constant) and *noise* = 0 (for the global noise level). All parameters are in SI units. For the stepsize of the physical simulation *simstepsize* = 0.001 was used (corresponding to a millisecond). With *controlinterval* = 1 one ensures that the controller, viz Equation (2), is updated as often as the physics of the environment.

The robot itself has a diameter of $2R = 0.5$, a mass off $M = 1$ and a *motorpowerfactor* = 120. The parameters for the damped oscillator (Equation 4) are $m = 1$, $k = m * motorpowerfactor$ and $\gamma = 2\sqrt{k * m}$ (critical damping). The relaxation rate for the membrane potential entering Equation (2) has been set to $\Gamma = 20$, retaining the bare excitatory and inhibitory weights, w_0 and z_0, as free simulation parameters.

3. RESULTS

3.1. Emergent Limit-Cycle Locomotion

In **Figure 3** we present the stability regions for the various regular movement patterns found, with respective close-ups given in

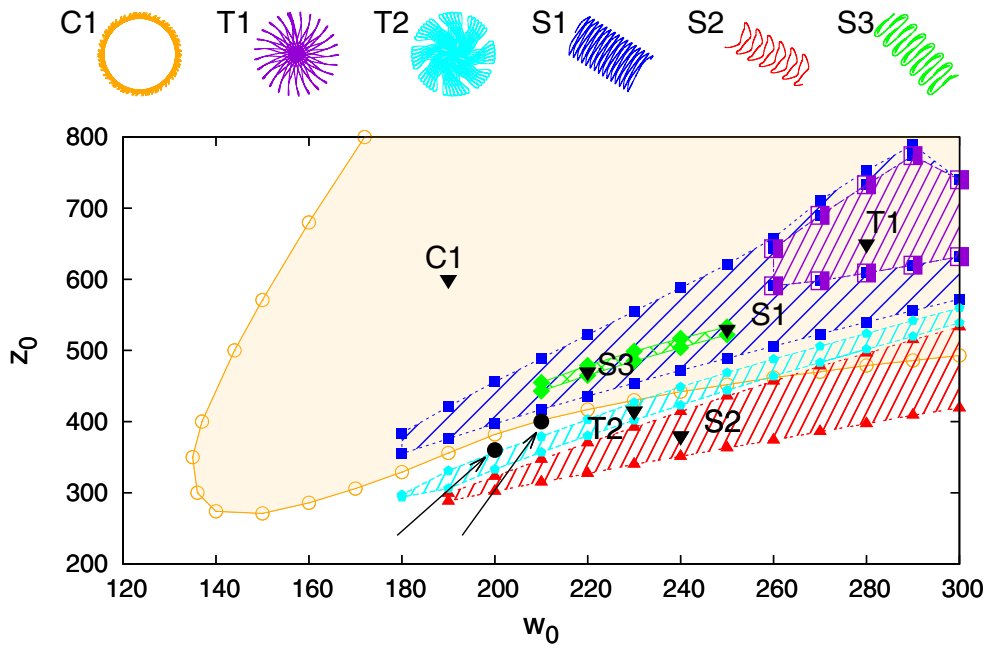


FIGURE 3 | Phase diagram for $U_{max} = 1$ in the parameter plane of excitatory (w_0) and inhibitory (z_0) synaptic weights. On the top the different types of identified regular motion patterns are illustrated, tagged respectively with black triangles in the respective regions of stability (shaded areas). Close-up trajectories are given in **Figure 4**; for a comparison see also Supplementary Video 1. Examples of two parameter settings, (200, 360) and (210, 400), for which chaotic behavior is observed are indicated by black filled circles (at the tip of the respective arrows).

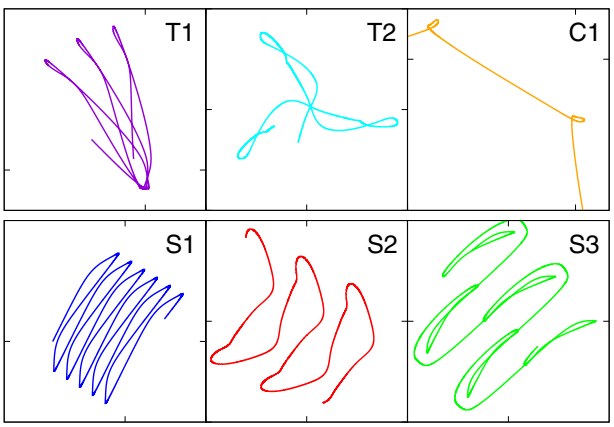


FIGURE 4 | A close-up of the trajectories in the plane of locomotion, for the parameters (w_0, z_0) tagged as black triangles in the phase diagram presented in Figure 3. T1: (280, 650), T2: (230, 415), C1: (190, 600), S1: (250, 530), S2: (240, 380), S3: (220, 470).

Figure 4. The results are for $U_{max} = 1$ (depressing short-term synaptic plasticity without Ca dynamics) and for the parameters specified in Section 2.3. They are obtained by adiabatically continuing stable states along a grid until stability is lost. Without STSP only a globally attracting fixpoint corresponding to a motionless robot is present. We note that regular motion arises for a wide range of bare excitatory (w_0) and inhibitory (z_0) synaptic weights. z_0 needs however to be larger than w_0.

All motion patterns observed are self-organized. There is no objective function (Gros, 2014), such as a maximal velocity, to be optimized. This implies that the quantitative features of the individual motion patterns change smoothly within their respective stability regions, and that one can identify the observed regular movement patters as stable limit cycles in the sensorimotor loop (Sándor et al., 2015). Fast switching between motion primitives would be possible by a putative overarching controller, since more than one limit cycle may be stable for given synaptic weights w_0 and z_0. Interactions between robots or with external obstacles might also lead to the automatic selection of another coexisting mode (see for instance Supplementary Video 1).

It is evident that the body plan of the robot examined here tends to produce meandering motion pattern. T1 and T2 are sun- and star-like movements with small (T1) and large (T2) processing angles (compare **Figure 4**; "T" stands for torus in phase space). There is, in addition, a (nearly pure) circular motion, C1, and three types of forward snake-like meandering motion patters, S1, S2, and S3. From these S3 partly overlaps with itself. These modes are characterized by distinct motion patterns of the three weights, as shown in **Figure 5**, as measured by their positions along their respective rods. The differences between the distinct modes are in part qualitative, in terms of the time sequences in which the three neurons are subsequently active, and in part only quantitative. The difference between T1 and S1 is, in this respect, that the up-times of the two active neurons are symmetric for S1, but not for T1. A spontaneous symmetry

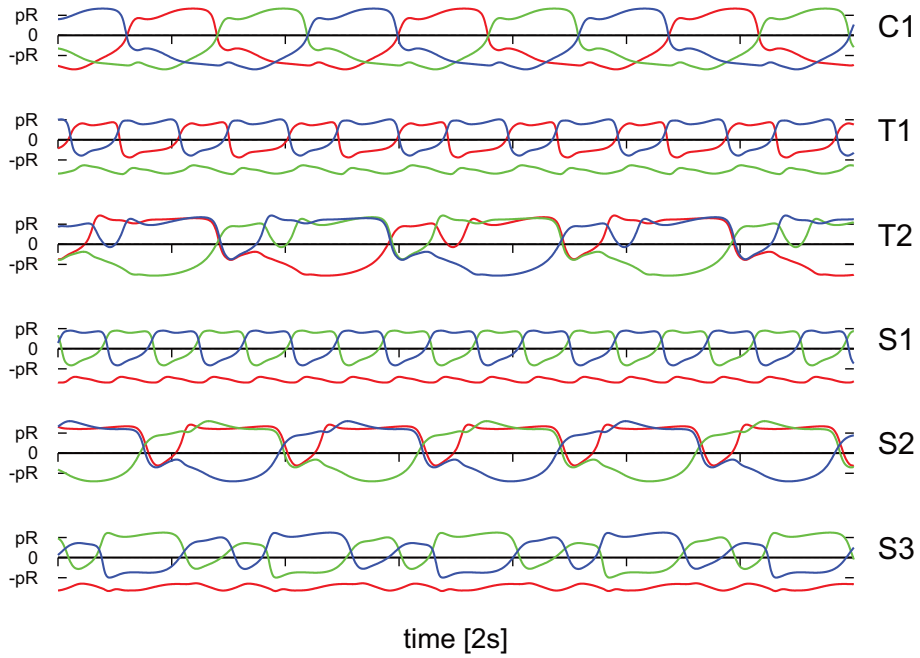


FIGURE 5 | The positions $x_i^{(a)}$ of the three weights as a function of time, compare Figure 1, along the corresponding rods. The modes and parameters are identical to the ones presented in **Figure 4**. Time is measured in units of 2 s.

breaking can be furthermore observed in case of T1, S1, S2, S3, for which two weights always have alternating dynamics, the third one showing a qualitatively different behavior. In contrast to that, the time-series of the C1 and T2 modes reveals the symmetrical but phase shifted oscillation of the three weights. Note that the positions of the weights may overshoot the interval $[-pR, pR]$ for the target positions $x_i^{(t)}$, both due to inertia and due to the additional gravitational pull. Motion patterns similar to the ones shown in **Figure 4** have been observed in a self-organized two-wheeled robot in the frozen mode (Der and Martius, 2013).

3.2. Chaotic Modes Allowing for Explorative Behavior

The dynamics of the robot takes place in a phase space combining the internal variables, of both body and controller, with the ones of the environment. The stability regions of the individual limit cycles presented in **Figure 3** will therefore be bounded, generically, by a suitable bifurcation, such as a supercritical Hopf bifurcation or a fold bifurcation of limit cycles (Gros, 2015; Sándor et al., 2015). Alternatively, a transition to chaos may occur. It is on the other side also possible that chaotic attractors emerge from previously unstable manifolds and that the stability region of chaotic and stable manifolds overlap.

Close to a chaotic phase long transients may occur, which makes it difficult to study systematically the exact extend of the chaotic region. In **Figure 3** we have indicated however a few representative combinations of parameters, for which stable chaos is observed both in the limit of long simulations times and for a wide range of stepsizes of the ODE simulator. No regular motion patterns can be observed in the screenshots presented in **Figure 6**. We have also evaluated the long-time behavior of the square of the covered real-space distance,

$$d^2(\tau) \;=\; \langle \big(\mathbf{x}(t+\tau) - \mathbf{x}(t)\big)^2 \rangle_t \,. \tag{5}$$

We found diffusive transport $d \sim \sqrt{\tau}$ for the chaotic mode and ballistic transport $d \sim \tau$ for the forward meandering modes S1, S2, and S3. Both as expected.

It has been observed, that chaotic locomotion of an embodied system may be considered as a basic explorative behavior, both of the environment and of the own motor pattern (Steingrube et al., 2010; Shim and Husbands, 2012). As a test of this hypothesis we have set our three-rod robot into a restricted playground containing movable objects in the form of blocks, which can be pushed, to a certain extend, over the ground. A screenshot is presented in **Figure 6**. One can observe, that the robot stays for a while close to the object, bumping around, and retracting in part a trajectory having a shape similar to the one generated by a C1 limit cycle. This is possible, as the set of parameters $(w_0, z_0) = (210, 400)$ considered is located close to but outside the C1-stability region. The C1 limit cycle is hence only weakly unstable in the chaotic phase. The active exploration of the environment, occurring here when bumping into obstacles, gives the robot hence access to otherwise unstable locomotion options. The overall behavior may be interpreted alternatively in terms of non-representational sensorimotor knowledge (Buhrmann and Di Paolo, 2014). For a longer simulation see the Supplementary Videos.

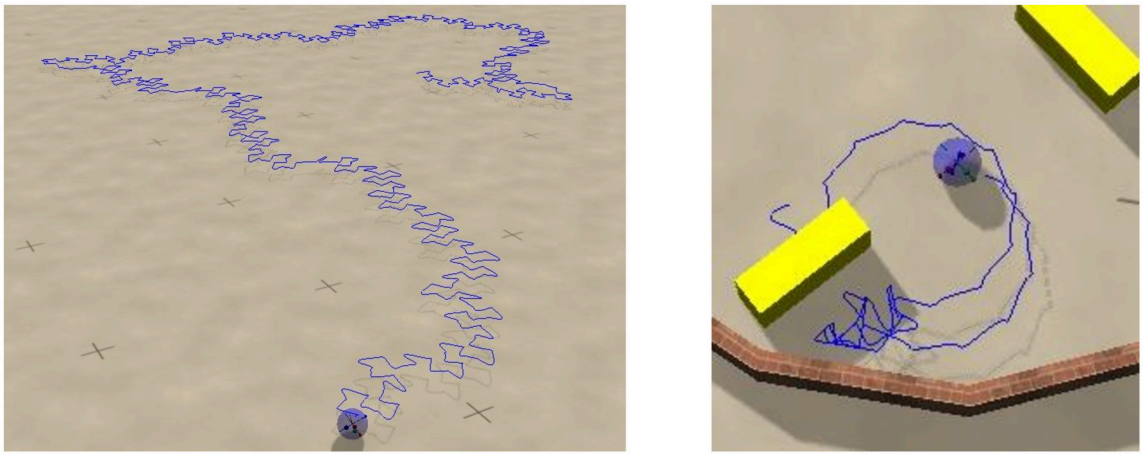


FIGURE 6 | Screenshots of the sphere robot in a chaotic mode; $U_{max} = 1$ and $(w_0, z_0) = (210, 400)$. The blue lines retrace the past trajectory. The short-time motion of the robot is close to the one of the S2 mode, which is here an unstable attractor (compare **Figure 4**). **Left**: In open space. **Right**: In a closed environment allowing for the interaction with movable objects (yellow blocks). The circular sections correspond to unstable C1 limit cycles. A close-up to the dynamics and a longer simulation in the maze can be seen in Supplementary Videos 2, 3 respectively.

In the movie presented in the Supplementary Material one can observe, furthermore, that the robot is pushing the blocks around in a seemingly "playful" manner (see Supplementary Video 3). A remarkable behavior, in our view, considering that the sphere robot disposes of a mere total of three controlling neurons. We note, that this complex behavior results from the interplay of the autonomous dynamics, as resulting from the inter-neural short-term synaptic plasticity, with environmental feedback.

3.3. Embodiment Shaping the Intrinsic Dynamics

One can consider the controlling 3-neuron network in isolation by identifying the sensory reading $x_i^{(a)}$ for the actual position of the weight along the rod with the respective target position $x_i^{(t)}$, viz by setting $x_i^{(a)} = x_i^{(t)}$ in Equation (2). The resulting network contains a self-excitatory coupling w_0 together with all-to-all inhibition with a bare synaptic strength z_0. The short-term synaptic plasticity then induces an autonomous activity, as illustrated in **Figure 7**, which is topologically equivalent to the C1 mode. This equivalence becomes even more pronounced when suspending the robot in air, which can be achieved in turn by simply removing gravity from the physics simulation (bottom time-series in **Figure 7**). One can hence consider the C1 mode as the driver for the observed physical motion.

The isolated 3-neuron network has, however, only a single stable limit cycle. Numerically integrating the isolated network for parameters settings (w_0, z_0) corresponding to the six modes of **Figure 5**, as well as for chaotic states, we find always an identical sequential activation of the three neurons illustrated in **Figure 7**, with only slight changes in the overall shape. It is hence clear, that the other modes T1, T2, S1, S2, and S3, as well as the chaotic behavior, do result from the closed-loop feedback of the environment. The interaction of the environment with the intrinsic dynamics then results in the emergence of alternative types of locomotion.

3.4. Stability with Respect to Noise

We present in **Figure 8** an analysis of the stability of the various modes found, with respect to noise in the sensory readings, where the level of the noise is given by the relative standard deviation σ of the sensory readings $x_i^{(a)}$. Comparing with the phase diagram, as presented in **Figure 3**, one notices that first modes to disappear, T1 and S3, are the ones with small stability regions in the phase diagram. Ramping up the noise level the T1 and S3 modes turn respectively, above their corresponding critical noise levels, into C1 and S1 modes. The other modes, including the chaotic phase, are in contrast very stable with respect to noise.

3.5. Autonomous Mode Switching

We present in **Figure 9** the phase diagram obtained when using $U_{max} = 4$ for the maximal Ca-level entering Equation (1). Within the range of (w_0, z_0) scanned we find four out of the six modes observed for $U_{max} = 1$ (compare **Figure 3**). The range of inhibitory weights z_0 for which stable locomotion is found is rescaled down, in addition, with respect to the $U_{max} = 1$ case. Interestingly we found a chaotic state at (180, 80) which lies just inside the stability region of the C1 mode.

We did let the robot evolve within the borders of a simple maze, as shown in **Figure 10** and Supplementary Video 4. Most of the time the robot is in the chaotic state, which is the dominant mode for the parameters used, namely $(w_0, z_0) = (180, 80)$ and $U_{max} = 4$. Intermittently, after colliding with a wall, the robot switches to the coexisting C1 mode. The radius of the stable C1 limit cycle in real-world coordinates is however so large, for $(w_0, z_0) = (180, 80)$, that it does not fit into the maze.

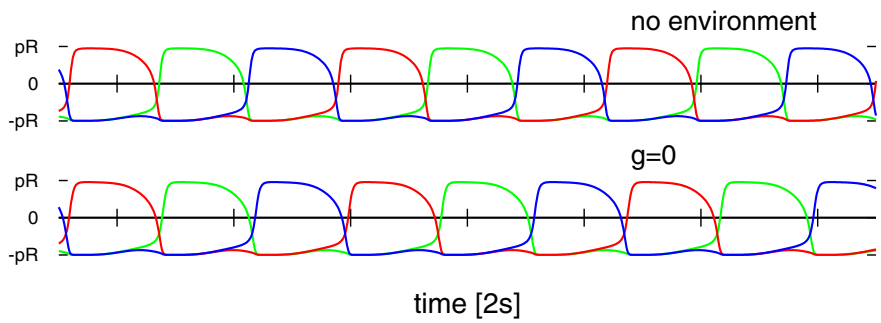


FIGURE 7 | Time series of the target positions $x_i^{(t)}$ for $U_{max} = 1$ and $(w_0, z_0) = (190, 600)$, which correspond to the C1 mode shown in Figures 4, 5. Top: For a numerical simulation of the isolated network obtained when setting $x_i^{(a)} = x_i^{(t)}$ in Equation (2). **Bottom**: For the 3-rod robot suspended in air (with the gravity constant g set to zero). Note that both time-series are very similar but not identical.

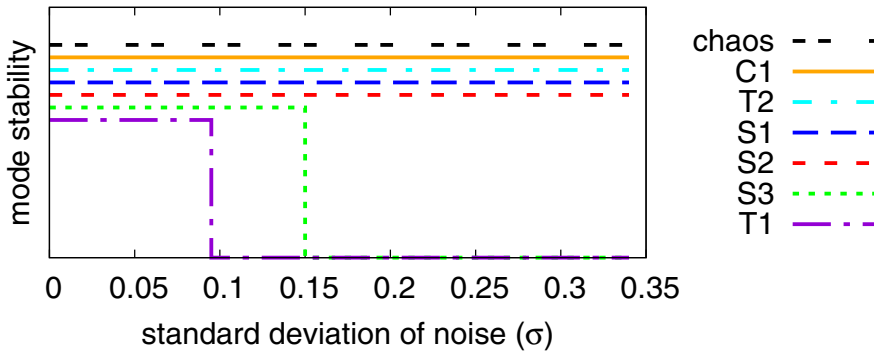


FIGURE 8 | Stability analysis of the modes found for $U_{max} = 1$, compare Figure 3, against a noise term Δx in the sensory reading, defined by $x_i^{(a)} \rightarrow x_i^{(a)}(1 + \Delta x)$. Finite and zero values along the y-axis indicate stability and instability (the displacements along the y-axis are only for avoiding overcrowding). The noise Δx is normal-distributed with standard deviation σ. Once T1 and S3 become unstable, when adiabatically increasing the noise level, their respective basins of attraction merge with the attracting regions of C1 and S1.

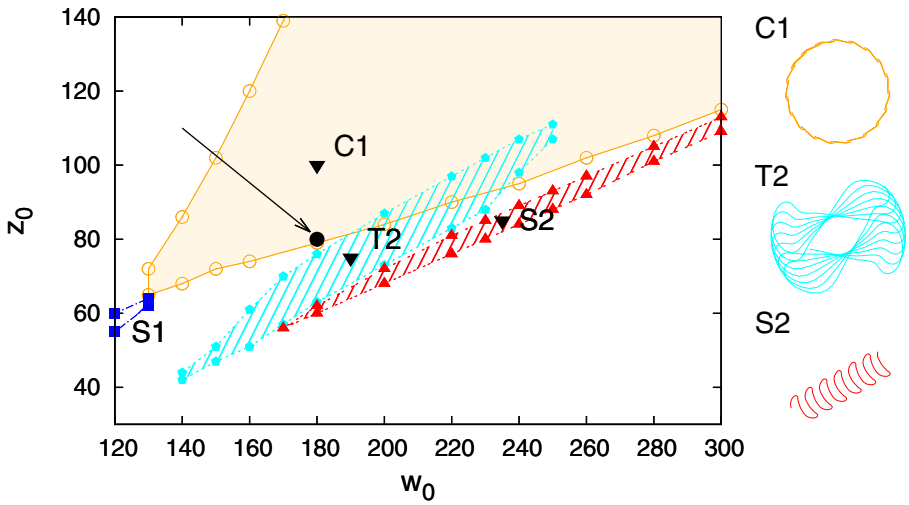


FIGURE 9 | The phase diagram obtained when using $U_{max} = 4$ in the STSP rules (Equation 1), with the naming of the modes corresponding to the ones used for the $U_{max} = 1$ phase diagram presented in Figure 3. At $(w_0, z_0) = (180, 80)$ there is chaotic state, as indicated by the arrow, coexisting with the C1 mode. The extent of the chaotic phase has not been examined in depth. On the right the traces are shown for the three dominant modes C1, T2, and S2.

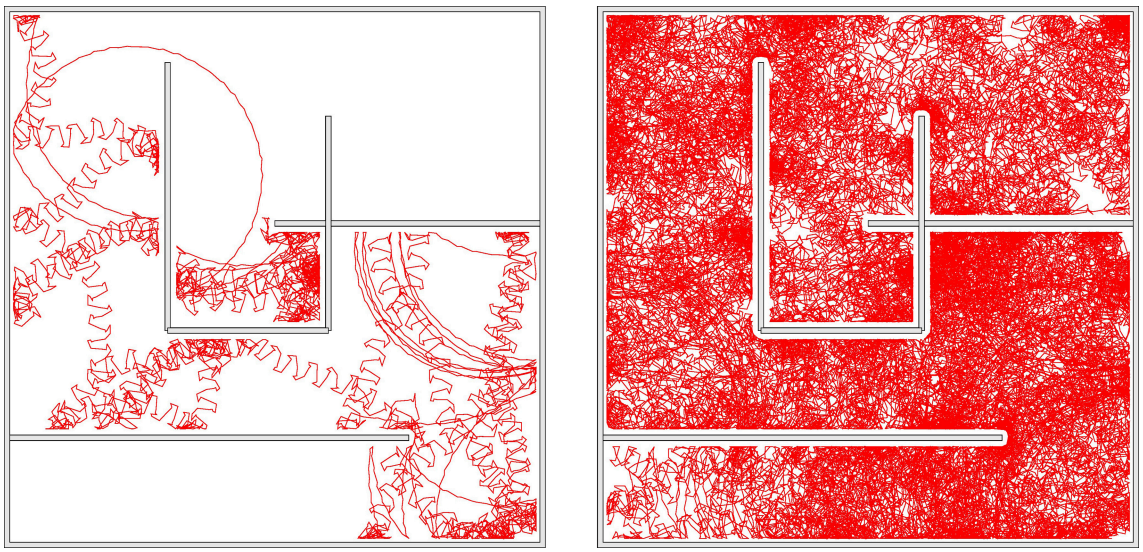


FIGURE 10 | The trace of the robot in a maze for a simulation time of 83 (left) and 1000 (right) min, respectively. The robot may remain stuck occasionally in corners, but not forever. The parameters are $U_{max} = 4$ and $(w_0, z_0) = (180, 80)$, corresponding to the chaotic mode indicated by the arrow in **Figure 9**. Bumping against the wall the robot sometimes turns up in the C1 mode, which is a coexisting stable limit cycle. The radius of the C1 mode is however, in this case, so large, that it does not fit as a whole into the maze. Also note that the chaotic mode is locally akin to the here unstable S2 mode, and that it changes the overall direction only on a relatively large scale.

The robot hence continues exploring. We have obtained similar results when using a $U_{max} = 1$ chaotic mode.

A screenshot of a trajectory in open space is presented in **Figure 11**. One notices, that the $U_{max} = 4$ and $(w_0, z_0) = (180, 80)$ chaotic mode wanders around aimlessly in much smother manner, than the $U_{max} = 1$ chaotic mode shown in **Figure 6**. This is the result of topologically different attractor structures, as seen in the phase space of internal variables (see the Supplementary Materials). Different types of chaos are indeed known to exist (Wernecke et al., 2016).

The autonomous mode switching observed for the regular motion primitives can also be seen in Supplementary Video 1. For a detailed discussion of the possible switching scenarios see the Supplementary Materials.

3.6. Switching between Degenerate Unstable Limit Cycles

In **Figure 12** we compare for the two chaotic modes, realized for $U_{max} = 1$ and for $U_{max} = 4$ respectively, the time series for the positions of the weights along the rods. One observes, that the movements of the weight is qualitatively similar, on short time scales, to an S2 mode (compare **Figure 5**, see also Supplementary Video 3). It is interesting, in this context, that the S2 mode has two types of degeneracies.

- Continuous. The S2 mode may propagate in any direction. There is hence a continuous manifold of attractors in the combined phase of controller, body and environment. Outside the actual region of stability this manifold contains either unstable limit cycles or limit cycle relicts (Gros, 2009).
- Discrete. There is a spontaneous symmetry breaking in the S2 mode, with two weights having identical but phase shifted movement patterns along their respective rods, which are qualitatively different to the trajectory of the third weight (see **Figure 5**).

For the $U_{max} = 4$ chaotic mode we did not observe discrete mode switching, in above sense, which however occurs frequently for the $U_{max} = 1$ mode (see **Figure 12**). The chaotic meandering observed for the $U_{max} = 4$ chaotic mode, as evident in **Figure 11**, is hence a consequence of a smooth diffusion of the angle of propagation on the manifold of unstable S2 limit cycles (or limit cycle relicts Linkerhand and Gros, 2013). In the phase space of the neural activity (as shown in Supplementary Figure 5), the trajectory corresponds to a chaotic phase diffusion along a limit cycle (Wernecke et al., 2016). This process is determinstic and not due to numerical errors, as we have checked by systematically reducting the stepsize used for the numerical integration. Noise is absent.

4. CONCLUSIONS

We have shown here, that a robot controlled by only a very limited number of neurons, three in our case, may show complex behavior which may be interpreted as explorative or playful. This is possible when locomotion results from self-organizing processes in the sensorimotor loop. The driving control dynamics, for which we have considered here short-term synaptic plasticity, then adapts itself seemingless to the physical requirements. No central controller is needed to detect an external object (Rai et al., 2014), or to switch direction

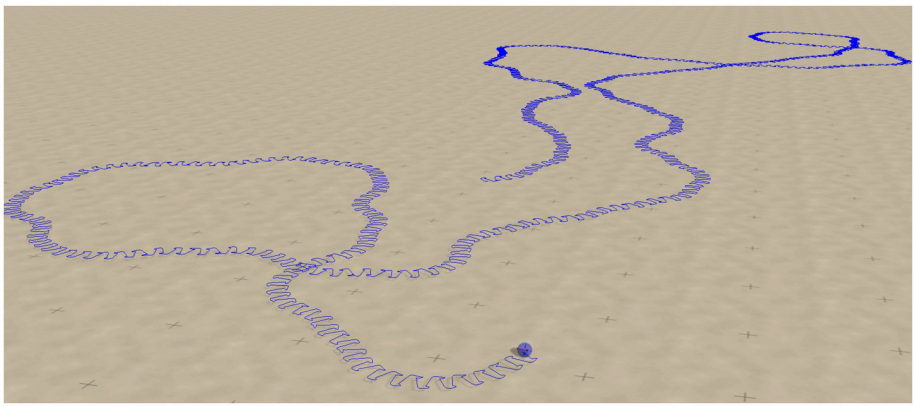


FIGURE 11 | Screenshot of the sphere robot in a chaotic mode for $U_{max} = 4$ and $(w_0, z_0) = (180, 80)$, indicated by the arrow in Figure 9. The blue line retraces the past trajectory. Note that the chaotic wandering is substantially smoother than the one observed for the $U_{max} = 1$ case (compare **Figure 6**).

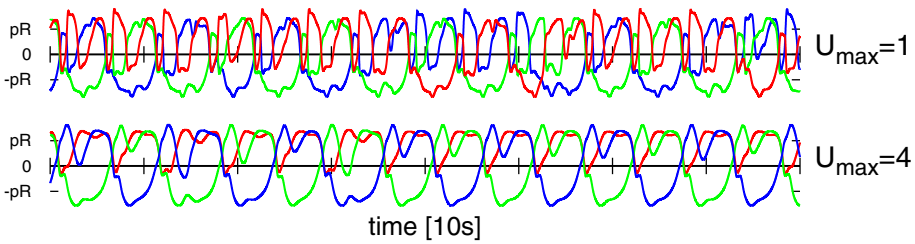


FIGURE 12 | As a function of time the positions of the three weights, compare Figure 1, along the corresponding rods. Top: For the $U_{max} = 1$ chaotic mode with $(w_0, z_0) = (210, 400)$ shown in **Figure 6**. **Bottom**: For the $U_{max} = 4$ chaotic mode $(w_0, z_0) = (180, 80)$ shown in **Figure 11**. Both modes are locally akin to an S2 mode, albeit with substantial fluctuations (e.g., compare the bottom curvatures of the green line for $U_{max} = 4$, see also **Figure 5**). Note that phase slips do occur for the case of $U_{max} = 1$, but not for $U_{max} = 4$.

when colliding with it. Stable and unstable limit cycles, together with chaotic attractors, arise in the phase space of internal (control and robot body) variables. These attractors form continua in the space of physical location and overall propagation direction, with the chaotic locomotion transitioning between unstable limit cycles. Transitions may either be between different types of regular locomotion, bounded circular or propagation meandering modes, or between the directions of unstable propagating limit cycles.

We note that the formation of a continuum of attractors is possible, whenever internal and external variables can be separated, such that internal variables span an independent subset of the phase space of the dynamical system. Here, the position of the robot (on the ground plane, in the absence of obstacles) acts as an external variable, all the other variables being independent of it. The limit cycles and chaotic attractors, living in the subspace of internal variables, exist thus for all position vectors, generating a continuous degeneracy of locomotion modes. The interactions with other robots and obstacles then results in a transient breakdown of this degeneracy, which is restored instantaneously with the termination of physical contact. Within this context, higher order control mechanisms would correspond to an external-variable dependent feedback, shaping the attractors either intermittently or slowly (with respect to the internal dynamics), thus leading possibly to the emergence of transiently stable attractors.

Our result, that the three-rod robot switches spontaneously between a continuous set of attractors, in the chaotic state, can be seen as a realization of chaotic wandering (Tsuda, 2001), which has been argued in turn to occur in the brain in the form of self-organized instabilities (Friston et al., 2012), viz as transient-state dynamics (Gros, 2007). There is furthermore a close relation to the concept of attractor metadynamics (Gros et al., 2014), which denotes the either induced or spontaneous switching between attracting sets.

The here simulated robot is furthermore compliant both on the level of control and actuators, showing a highly flexible response. The actuators are implemented by specifying a target position for a limb, here a moving weight on a rod. The force acting on the weight then results from the interplay between the internal driving, provided by a damped spring (between the actual and the target position), with the physical restoring forces acting on the weights, which in turn depend on the body dynamics determined by the interaction with the ground, obstacles and other robots (Floreano et al., 2014).

The isolated controlling network (realized in the limit of infinitely strong actuators) can be interpreted in addition

as a central pattern generator (Steingrube et al., 2010), having a single intrinsic limit-cycle attractor. The open-loop control incorporates however the feedback of the environment through the induced forces. We find here, that the resulting embodiment (Cangelosi et al., 2015) does morph the driving dynamics of the central pattern generator not only quantitatively, but also qualitatively, giving rise to a vast array of modes which differ in part topologically from the dynamics of the underlying central pattern generator. We believe that this dynamical systems approach of the locomotion of simple robots has not been fully exploited yet, having many interesting features and applications in store for the field of neurorobotics.

AUTHOR CONTRIBUTIONS

The experiments were conceived and designed by CG, BS, and LM, performed mainly by LM with BS adding some data. The data was analyzed by CG, BS, and LM, most of the plots produced by LM. The manuscript was mostly written by CG, with BS adding some paragraphs and revising it with LM.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2016.00012

REFERENCES

Aguilar, J., Zhang, T., Qian, F., Kingsbury, M., McInroe, B., Mazouchova, N., et al. (2016). A review on locomotion robophysics: the study of movement at the intersection of robotics, soft matter and dynamical systems. *arXiv:1602.04712*.

Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *Eur. Phys. J. B* 63, 329–339. doi: 10.1140/epjb/e2008-00175-0

Barak, O., and Tsodyks, M. (2014). Working models of working memory. *Curr. Opin. Neurobiol.* 25, 20–24. doi: 10.1016/j.conb.2013.10.008

Buhrmann, T., and Di Paolo, E. (2014). "Non-representational sensorimotor knowledge," in *International Conference on Simulation of Adaptive Behavior* (Castellon: Springer), 21–31. doi: 10.1007/978-3-319-08864-8_3

Cangelosi, A., Bongard, J., Fischer, M. H., and Nolfi, S. (2015). "Embodied intelligence," in *Springer Handbook of Computational Intelligence* (Berlin; Heidelberg: Springer), 697–714. doi: 10.1007/978-3-662-43505-2_37

Carrillo-Reid, L., Lopez-Huerta, V. G., Garcia-Munoz, M., Theiss, S., and Arbuthnott, G. W. (2015). Cell assembly signatures defined by short-term synaptic plasticity in cortical networks. *Int. J. Neural Syst.* 25:1550026. doi: 10.1142/S0129065715500264

Chiel, H. J., and Beer, R. D. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends Neurosci.* 20, 553–557. doi: 10.1016/S0166-2236(97)01149-1

Der, R. (2016). In search for the neural mechanisms of individual development: behavior-driven differential hebbian learning. *Front. Robot. AI* 2:37. doi: 10.3389/frobt.2015.00037

Der, R., and Martius, G. (2012). *The Playful Machine: Theoretical Foundation and Practical Realization of Self-Organizing Robots*, Vol. 15. Berlin; Heidelberg: Springer Science & Business Media. doi: 10.1007/978-3-642-20253-7

Der, R., and Martius, G. (2013). "Behavior as broken symmetry in embodied self-organizing robots," in *Advances in Artificial Life, ECAL 2013* (Cambridge, MA: MIT Press), 601–608. doi: 10.7551/978-0-262-31709-2-ch086

Der, R., and Martius, G. (2015). Novel plasticity rule can explain the development of sensorimotor intelligence. *Proc. Natl. Acad. Sci. U.S.A.* 112, E6224–E6232. doi: 10.1073/pnas.1508400112

Fioravante, D., and Regehr, W. G. (2011). Short-term forms of presynaptic plasticity. *Curr. Opin. Neurobiol.* 21, 269–274. doi: 10.1016/j.conb.2011.02.003

Floreano, D., Ijspeert, A. J., and Schaal, S. (2014). Robotics and neuroscience. *Curr. Biol.* 24, R910–R920. doi: 10.1016/j.cub.2014.07.058

Frank, M., Leitner, J., Stollenga, M., Förster, A., and Schmidhuber, J. (2014). Curiosity driven reinforcement learning for motion planning on humanoids. *Front. Neurorobot.* 7:25. doi: 10.3389/fnbot.2013.00025

Friston, K. (2011). What is optimal about motor control? *Neuron* 72, 488–498. doi: 10.1016/j.neuron.2011.10.018

Friston, K., Breakspear, M., and Deco, G. (2012). Perception and self-organized instability. *Front. Comput. Neurosci.* 6:44. doi: 10.3389/fncom.2012.00044

Gros, C. (2007). Neural networks with transient state dynamics. *N. J. Phys.* 9:109. doi: 10.1088/1367-2630/9/4/109

Gros, C. (2009). Cognitive computation with autonomously active neural networks: an emerging field. *Cogn. Comput.* 1, 77–90. doi: 10.1007/s12559-008-9000-9

Gros, C. (2014). "Generating functionals for guided self-organization," in *Guided Self-Organization: Inception*, ed M. Prokopenko (Berlin; Heidelberg: Springer), 53–66. doi: 10.1007/978-3-642-53734-9_3

Gros, C. (2015). *Complex and Adaptive Dynamical Systems: A Primer*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-319-16265-2

Gros, C., Linkerhand, M., and Walther, V. (2014). "Attractor metadynamics in adapting neural networks," in *International Conference on Artificial Neural Networks* (Berlin; Heidelberg: Springer), 65–72. doi: 10.1007/978-3-319-11179-7_9

Gupta, A., Wang, Y., and Markram, H. (2000). Organizing principles for a diversity of gabaergic interneurons and synapses in the neocortex. *Science* 287, 273–278. doi: 10.1126/science.287.5451.273

Hennig, M. H. (2013). Theoretical models of synaptic short term plasticity. *Front. Comput. Neurosci.* 7:45. doi: 10.3389/fncom.2013.00045

Jia, Y., and Parker, D. (2016). Short-term synaptic plasticity at interneuronal synapses could sculpt rhythmic motor patterns. *Front. Neural Circuits* 10:4. doi: 10.3389/fncir.2016.00004

Linkerhand, M., and Gros, C. (2013). Generating functionals for autonomous latching dynamics in attractor relict networks. *Sci. Rep.* 3:2042. doi: 10.1038/srep02042

Martius, G., Der, R., and Ay, N. (2013). Information driven self-organization of complex robotic behaviors. *PLoS ONE* 8:e63400. doi: 10.1371/journal.pone.0063400

Montúfar, G., Ghazi-Zahedi, K., and Ay, N. (2015). A theory of cheap control in embodied systems. *PLoS Comput. Biol.* 11:e1004427. doi: 10.1371/journal.pcbi.1004427

Nadim, F., and Manor, Y. (2000). The role of short-term synaptic dynamics in motor control. *Curr. Opin. Neurobiol.* 10, 683–690. doi: 10.1016/S0959-4388(00)00159-8

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Prokopenko, M. (2009). Guided self-organization. *HFSP J.* 3, 287–289. doi: 10.1080/19552068.2009.9635816

Rai, A., Meier, F., Ijspeert, A., and Schaal, S. (2014). "Learning coupling terms for obstacle avoidance," in *2014 IEEE-RAS International Conference on Humanoid Robots* (Piscataway, NJ: IEEE), 512–518. doi: 10.1109/HUMANOIDS.2014.7041410

Regehr, W. G. (2012). Short-term presynaptic plasticity. *Cold Spring Harb. Perspect. Biol.* 4:a005702. doi: 10.1101/cshperspect.a005702

Salge, C., Glackin, C., and Polani, D. (2014). "Empowerment–an introduction," in *Guided Self-Organization: Inception*, ed M. Prokopenko (Berlin; Heidelberg: Springer), 67–114. doi: 10.1007/978-3-642-53734-9_4

Sándor, B., Jahn, T., Martin, L., and Gros, C. (2015). The sensorimotor loop as a dynamical system: how regular motion primitives may emerge from self-organized limit cycles. *Front. Robot. AI* 2:31. doi: 10.3389/frobt.2015.00031

Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H. (2013). Walknet, a bio-inspired controller for hexapod walking. *Biol. Cybern.* 107, 397–419. doi: 10.1007/s00422-013-0563-5

Shim, Y., and Husbands, P. (2012). Chaotic exploration and learning of locomotion behaviors. *Neural Comput.* 24, 2185–2222. doi: 10.1162/NECO_a_00313

Smith, R. (2005). *Open Dynamics Engine*. Available online at: http://ode.org/ode-latest-userguide.html

Steingrube, S., Timme, M., Wörgötter, F., and Manoonpong, P. (2010). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nat. Phys.* 6, 224–230. doi: 10.1038/nphys1508

Toutounji, H., and Pasemann, F. (2014). Behavior control in the sensorimotor loop with short-term synaptic dynamics induced by self-regulating neurons. *Front. Neurorobot.* 8:19. doi: 10.3389/fnbot.2014.00019

Tsodyks, M. V., and Markram, H. (1997). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proc. Natl. Acad. Sci. U.S.A.* 94, 719–723. doi: 10.1073/pnas.94.2.719

Tsuda, I. (2001). Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behav. Brain Sci.* 24, 793–810. doi: 10.1017/S0140525X01000097

Wang, Y., Markram, H., Goodman, P. H., Berger, T. K., Ma, J., and Goldman-Rakic, P. S. (2006). Heterogeneity in the pyramidal network of the medial prefrontal cortex. *Nat. Neurosci.* 9, 534–542. doi: 10.1038/nn1670

Wernecke, H., Sándor, B., and Gros, C. (2016). Partially predictable chaos. *arXiv:1605.05616*.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2016 Martin, Sándor and Gros. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 30 January 2017
doi: 10.3389/fnbot.2017.00003

ReaCog, a Minimal Cognitive Controller Based on Recruitment of Reactive Systems

Malte Schilling[1]* *and Holk Cruse*[2]

[1] Center of Excellence Cognitive Interaction Technology, Bielefeld University, Bielefeld, Germany, [2] Department of Biological Cybernetics and Theoretical Biology, Bielefeld University, Bielefeld, Germany

It has often been stated that for a neuronal system to become a cognitive one, it has to be large enough. In contrast, we argue that a basic property of a cognitive system, namely the ability to plan ahead, can already be fulfilled by small neuronal systems. As a proof of concept, we propose an artificial neural network, termed reaCog, that, first, is able to deal with a specific domain of behavior (six-legged-walking). Second, we show how a minor expansion of this system enables the system to plan ahead and deploy existing behavioral elements in novel contexts in order to solve current problems. To this end, the system invents new solutions that are not possible for the reactive network. Rather these solutions result from new combinations of given memory elements. This faculty does not rely on a dedicated system being more or less independent of the reactive basis, but results from exploitation of the reactive basis by recruiting the lower-level control structures in a way that motor planning becomes possible as an internal simulation relying on internal representation being grounded in embodied experiences.

Keywords: reactive system, cognitive system; internal model, motor planning, internal simulation, neural networks, attention

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark, Denmark

Reviewed by:
Yulia Sandamirskaya, University of Zurich, Switzerland
Yoonsuck Choe, Texas A&M University, USA
Michail Maniadakis, Foundation for Research & Technology – Hellas, Greece

***Correspondence:**
Malte Schilling
mschilli@techfak.uni-bielefeld.de

Received: *18 September 2016*
Accepted: *11 January 2017*
Published: *30 January 2017*

Citation:
Schilling M and Cruse H (2017) ReaCog, a Minimal Cognitive Controller Based on Recruitment of Reactive Systems. Front. Neurorobot. 11:3. doi: 10.3389/fnbot.2017.00003

INTRODUCTION

Over the last years more and more findings in neuroscience have shown that higher level cognitive capabilities cannot be detached from the functioning of lower level sensorimotor control systems (van Duijn et al., 2006; Barsalou, 2008) which is the core idea of embodied cognition as a field. It is assumed that cognition recruits the underlying sensorimotor systems (Anderson, 2010). Intensively studied examples controlled by such sensorimotor, or reactive, systems are insects. Already a lot is known about their structure and properties of their sensorimotor systems (Menzel et al., 2007; Cruse et al., 2009) which allows to build well performing biologically inspired systems (Pfeifer et al., 2007; Ijspeert, 2014). But it is still unclear if all the crucial properties are understood that are required to form the basis for a cognitive system. Do the known principles allow to leverage the sensorimotor control systems toward cognition?

A basic problem concerns what, after all, is meant by the term "cognition." Definitions cover various ideas, reaching from Maturana and Varela (1981) "life is cognition" (which would include even bacteria to be cognitive systems), Engel et al. (2013) who note that "cognition is action." Other authors avoid the problem of a short definition, which almost inevitably includes comparatively simple systems, by listing a collection of phenomena to characterize cognitive systems (e.g., Khlentzos and Schalley, 2007; Menzel et al., 2007). The most important faculties generally agreed as to characterize a cognitive system are attention, awareness, emotion, learning, specific aspects

of memory, language as well as thinking, reasoning, planning ahead, decision making, volition, Theory of Mind or even subjective feelings and consciousness (for another list proposed by Langley et al. (2009, see Discussion). In this article, we will not enter this discussion but focus on basic properties discussed by several authors as to be crucial for a cognitive system, namely the ability to invent new behaviors and the ability to plan ahead the latter being required to test the feasibility of the new invention.

Lower level behaviors, often termed reactive or automatic, controlled by "reactive systems," require procedural elements ensuring survival and allowing for basic behavioral abilities, e.g., locomotion, feeding, object avoidance. The combination of such controllers may also be suited to guide seemingly more complex behaviors (e.g., navigation). These controllers constitute the procedural memory of the system. Exploiting the loop through the world (Brooks, 1989) even a "hard-wired" memory system allows for adaptation to changing environments as will be illustrated in the second section (Reactive Walker). In reactive systems many of these procedures (or "action-perception circuits," Pulvermüller and Garagnani, 2014) can be active at the same time, but they may also compete amongst each other for controlling the system (Brooks, 1989). Therefore, a crucial ability for each behaving system—including reactive systems—is the ability to select one among different possible actions. This architecture is inspired by earlier authors as Arbib (1998), Brooks (1991b), and Minsky (1986).

Reactive systems, by definition, do not belong to the field of cognition. However, many authors (e.g., Newell, 1994; Anderson, 2010; Glenberg and Gallese, 2012) argue that cognition in all known systems is strongly based on and is intimately connected with a functional reactive system. Even more, as proposed by Barsalou (2008) and others, reactive (or behavior-based) systems having internal states (as introduced in the second section, Reactive Walker) plus being embodied are basic requirements for a system to become a cognitive one. As already noted briefly above, there is indeed strong support showing that neuronal elements forming cognitive properties are tightly intertwined with the reactive system itself and a functional separation is not possible. For example, planning of a movement is interpreted in this view as a mental enactment of the movement (Jeannerod, 2001; Hesslow, 2002). This view is supported as brain regions that formerly were assumed as being highly specialized, for example the motor area, are also activated during language processing or perception (Feldman and Narayanan, 2004; Buccino et al., 2005; Pulvermüller, 2005; Jeannerod, 2006; Pulvermüller and Garagnani, 2014). More generally, Gallese and Lakoff state that "a key aspect of human cognition is. the adaptation of sensory-motor brain mechanisms to serve new roles in reason and language, while retaining their original function as well." (Gallese and Lakoff, 2005, p. 456). This is supported by behavioral research showing that behavioral and cognitive processes are functionally related insofar as both processes seem to apply the same structuring principles and seem to have access to memory in a structurally similar way (e.g., Jeannerod and Decety, 1995; Cross et al., 2006; Barsalou, 2008; Barsalou et al., 2012).

What distinguishes a reactive system from a cognitive one? A key feature that might be suited for a distinction between reactive, or behavior-based, systems, and cognitive systems is that the former are restricted to apply their procedural memory elements (or internal representations, or internal models) only in the context in which the latter have been acquired (Wilson, 2008). For example, a specific movement (e.g., grasping a specific type of prey) is stored as a (congenital or learned) procedural memory. The content of this memory element may also be considered as a model of that movement, which can—in a reactive system—only be triggered by a specific stimulus, the specific prey. In contrast, cognitive systems are able to modify their behaviors and thereby may come up with solutions for a novel task (Glenberg and Gallese, 2012). A novel task is considered here a task in which, in the current context, none of the existing procedural memory elements can be applied to solve the problem, as none of the available procedures are able to deal with the actual situation or to predict the resulting consequences. Therefore, to approach a cognitive level, one has to search for systems that are creative, i.e., able to alter their procedural memory elements or to compose them in a new way allowing the system to handle such a novel tasks. This characterization agrees with the statement of Limongelli et al. (1995) "cognition is the ability to relate different unconnected pieces of information in new ways and apply the resulting knowledge in an adaptive manner." Taking a broader view, Anderson (2010), in his massive redeployment hypothesis, states that "neural reuse" is a fundamental principle not only applied in evolutionary time scales but also for solving current problems by a cognitive system. Thus, in this article we will focus on a system that is able to find solutions for novel tasks.

What are the prerequisites to find a solution to a current problem? One way to find new solutions is to apply a search strategy based on simple trial and error. But trial and error is a risky approach and generally quite slow. As an alternative, "internal trial-and-error" could be applied. This means that in addition to the ability to modify the procedures and their composition, such systems are able to anticipate consequences of new actions which enables the agent to decide based on these predictions (Hesslow, 2002). These aspects have already been captured by McFarland and Bösser (1993) who indeed define cognition as the faculty to plan ahead. Planning ahead allows to verify the feasibility of new solutions before execution. Therefore, planning ahead is the second basic property of our system. The ability to predict requires internal models, or internal representations.

Because our system is characterized here as to search for new solutions by exploiting the already existing memories (or internal models) in a flexible way, i.e., not only in a specific context, but in different contextual situations, an organizational scheme is required that allows for compositionality and modulation of specific parameters. In the third section (Motor Planning) we will provide a simple solution for this problem.

Following the view proposed by Barsalou (2008), Glenberg and Gallese (2012) and others, our approach is to start with a non-trivial reactive system that is then equipped with the ability to plan ahead. To this end, we will consider a system with a complex enough body (i.e., having a considerable number of extra degrees of freedom), but an arguably simple controller, which—in order to comply with biological

constraints—is based on elements forming an artificial neural network.

Using a system able to control autonomous behavior and using a complex, non-trivial body, we follow a whole-systems approach. We take the embodiment approach literally insofar as our system is constructed in such a way that it is currently used to control a simulated robot in a dynamical simulation environment, but will be transferred to a physical robot in a next step. Thus, we deal with really executable behaviors rather than with more abstract approaches on a dynamical systems level or systems that operate on a symbolic level. Application of such purely high-level approaches may bear the danger that serious problems occurring at a lower level may be overlooked (Brooks, 1991a; Verschure and Althaus, 2003).

Taken together, we focus on a system that allows for the ability to plan ahead (McFarland and Bösser, 1993) relying on intersnal representation (Steels, 2003) that are grounded in embodied experiences (Gallese and Lakoff, 2005). In this way, we follow the proposal of Feynman, who stated that we can only understand a system when we are able to create it (in Hawking, 2001; p. 83). We start with a decentralized, reactive neuronal network controller (Dürr et al., 2004) for a complex hexapod robot which is expanded by a holistic body model represented by a "hard-wired" recurrent neural network (RNN) and used for inverse kinematics (Schilling et al., 2012). Based on a reactive structure the robot allows for walking in an unpredictable environment.

We will further enable the robot to cope with situations for which the reactive system does not offer a solution. In this case, a "cognitive expansion" shall allow the system to search for a new solution to this problem. The search space is not only characterized by the 18°C of freedom (DoF) of the robot, but is expanded by the fact that the controller being embodied heavily depends on the "loop through the world," i.e., depends on the unpredictable properties of the environment. Further, the complexity of the situation is increased as behavioral elements to be selected show various time dependencies. To cope with such situations, the system first has to search for a behavioral element normally not used in the current context. The search space is large and not continuous. So, gradient descent methods are not applicable. The search for new solutions is based on (i) a somatotopic heuristic, (ii) noise applied to part of the cognitive expansion network as well as (iii) tests for physical feasibility of the solution proposed, first by internal simulation, second by performing the behavior in reality. For internal simulation, we exploit the property of the body model used here, which means that the same model cannot only be used as an inverse model, but also as a predictive model. Therefore, this body model can be used for motor planning applying an internal simulation to test newly selected behavioral elements.

The results show that the cognitive expansion requires only a small number of neurons coupled by a quite simple connectivity. This simple network shows basic properties required for a cognitive system and can be used as a scaffold for later introduction of further properties. In addition, capabilities like showing attention or emotions, might be found as properties emerging from such an architecture as discussed in Cruse and Schilling (2013).

The article is structured in the following way. The second section (Methods and Material) is divided in three parts. In section Background and Previously Developed Models. Reactive Walker—the Walknet (Reactive Walker) the simple control system for a hexapod walker is introduced which is biologically inspired from studies on the walking of insects. In section Motor Planning: from Walknet to reaCog (Motor Planning) the cognitive expansion is presented including an example that illustrates how the basic reactive system is recruited for planning. This will be followed by a more detailed explanation of the control architecture and the experiment setup (section Cognitive Expansion). Simulation results will be presented, on the one hand, for an example scenario (section Results) explaining our approach. On the other hand, a series of simulations shall demonstrate how the approach deals with disturbed walking. While there is no similar robotic architecture which applies behaviors out of context and realizes recruitment as internal simulation, we will present a brief overview on related work and discuss differences and implications (section Related Work). In the Discussion we will analyze the properties of the complete system, discuss them and briefly turn toward the question as to how aspects of higher-level phenomena being listed above may emerge in our system (Discussion and Conclusions).

MATERIALS AND METHODS

Background and Previously Developed Models. Reactive Walker—The Walknet

Biological Model of Insect Walking

The example we choose as a reactive basis and which will briefly be explained in the following concerns a hexapod (insect-like) walking system (see review Schilling et al., 2013b for details). The task to walk over a non-predictable substrate—possibly cluttered with obstacles of varying size and holes—is by no means a trivial one. The walker has six legs each equipped with three joints. Therefore, the controller has to deal with 18°C of freedom (DoF). As body position in space is defined by only six DoFs (three for position in space, three for orientation) there are 12 DoFs free to be decided upon by the controller which means that the controller has to make these 12 (respectively 18) decisions in a sensible way at any moment of time while dealing with an unpredictable environment. As a first step, the walker is only using tactile sensors situated in the legs (and possibly the antennae Schütz and Dürr, 2011) measuring contact with external objects, and with proprioceptors measuring position, torques and velocities of joints.

The walking system to be described in the following is based on behavioral (and to some extent neurophysiological) studies on insects, in particular stick insects (Schilling et al., 2013b). At first, we briefly describe the essentials of the earlier version, Walknet, and will then introduce expansions.

Experiments on the walking stick insect have shown that the neuronal system is organized in a decentralized way (Wendler, 1968; Bässler, 1983; Cruse, 1990). Derived from these results, a model has been proposed in which each leg is attributed a separate controller (Dürr et al., 2004; for a review Schilling

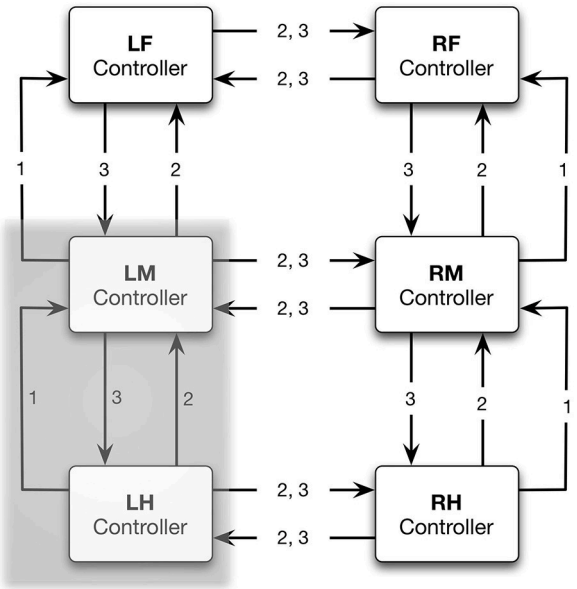


FIGURE 1 | General architecture of the reactive controller Walknet. The complete system consists of one controller for each leg (LF/RF left/right front leg, LM/RM left/right middle leg, LH/RH left/right hind leg). Coordination rules (1,2,3) act between neighboring legs, prolonging, or shortening the stance phase. Each leg controller contains several modules, a Swing-net and a Stance-net, to control swing and stance movement, respectively. In **Figure 2**, the shaded section is depicted in more detail.

et al., 2013b). **Figure 1** sketches the approximate anatomical arrangement of the controllers and the numbering of the legs. These single leg controllers are assumed to be situated in the thoracic ganglia (for a review see Bässler and Büschges, 1998). **Figure 2** shows details of the controllers as used in Walknet for the left middle leg (LM_leg) and the left hind leg (LH_leg). A single leg controller mainly consists of several movement primitives that reflect the leg movement consisting of two phases. These are the stance movement, during which the leg maintains ground contact and is retracted to propel the body forward, while supporting the weight of the body, and the swing movement where the leg is lifted off the ground and moved in the direction of walking, to touch down at the location where the next stance should begin. The movement primitives controlling stance and swing are realized in the leg controller (**Figure 2**) as several modules, or procedures, each containing artificial neurons forming a local, in general, recurrent neural network (RNN). These modules might receive direct sensory input and provide output signals that can be used for driving motor elements. The two most important procedural elements in our example are the Swing-net, responsible for controlling a swing movement, and the Stance-net controlling a stance movement (**Figure 2**, for swing: see Dürr et al., 2004; Schumm and Cruse, 2006, for stance: Schmitz et al., 2008; Schilling et al., 2012). The end positions used during forward walking are stored in the procedures for the swing and stance movement, i.e., the Swing-net and Stance-net respectively (in **Figure 2** they are part of the gray rounded boxes called Swing-net and Stance-net. Swing is triggered when the stance-end-position is reached, Stance movement is triggered by ground contact).

Following Maes (1990) the overall activation of a procedural element is controlled by a motivation unit (represented by yellow circles in the Figures) that gates to what extent the corresponding procedural element contributes to the control of the leg. In the network, these units forming rate coded, non-spiking neurons with leaky integrator, i.e., low pass, dynamics. They have a piecewise linear activation function (from 0 to 1) and control the strength of the output of the corresponding procedure (in a multiplicative way). Here we deal with a very simple motivation unit network that, initially, consists of just two units, the motivation units for the two procedural elements used in forward walking, Swing-net and Stance-net. Each motivation unit is reinforcing itself (not shown in **Figure 2**) and at the same time inhibiting the other motivation unit, forming a winner-take-all (WTA) net and allowing only one behavior to be active at any given time (**Figure 2**). Secondly, sensory signals control the behavior selection by influencing the motivation units and thus initiate behavioral transitions. When the leg touches the ground toward the end of a swing movement, the ground contact causes switching to stance movement by activating the motivation unit Stance. Correspondingly, during forward walking, reaching a given posterior position activates the motivation unit Swing. As an extension, we introduced backward walking. In this case, new swing and stance procedures are introduced including their motivation units (**Figure 3**). Swing_toBack behavior stores the target for the swing movement to the back. As for forward walking, a memory element is required representing the stance end position (for details see Schilling et al. (2013a) and explanation of the Stance movement below).

Furthermore, a leg controller must also take into account the interaction with the other legs. Part of these interactions are mediated directly by the body and through the environment, making explicit computations superfluous (see, e.g., the local positive velocity feedback approach Schmitz et al., 2008). While the physical coupling through the environment is important, it is not sufficient. In addition, the controllers of neighboring legs are coupled via a small number of channels transmitting information concerning the actual state of that leg (e.g., swing, stance) or its position (i.e., values of joint angles). These coordination rules were derived from behavioral experiments on walking sticks (Cruse, 1990). In **Figure 1** the channels are numbered 1–3. These coordination rules influence the length of the stance movement by influencing the transition from stance to swing movement, i.e., they change the value of the PEP. In **Figure 2** only one connection is shown, influence # 1, which suppresses the start of a swing movement of the anterior leg during the swing movement of the posterior leg (for details see Schilling et al., 2013b).

Beyond the motivation units that are directly controlling a procedural element, there are also motivation units (**Figure 3**, yellow circles) that are arranged to form some kind of hierarchical structure. Units which belong to the procedural nets controlling the left middle leg show positive connections to a motivation unit termed Leg_LM and this is correspondingly

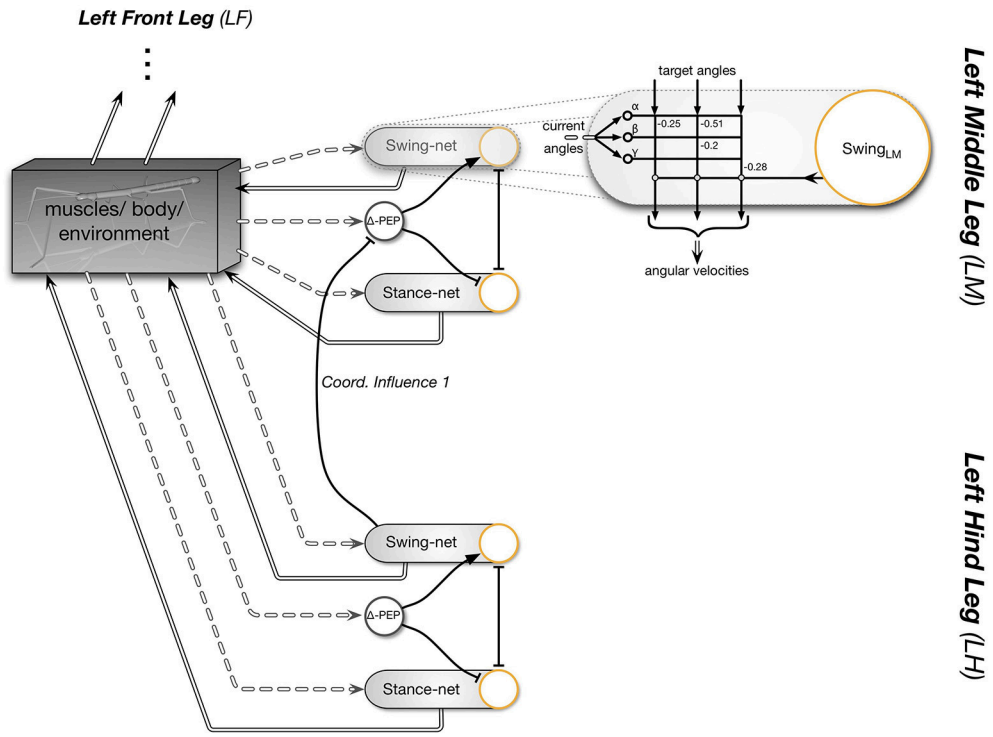


FIGURE 2 | Interactions between two leg controllers, left middle leg and left hind leg. This figure details the shaded area of **Figure 1**. The left side indicates the interaction with the environment mediated through the body. Each leg controller contains several modules: a Swing-net and a Stance-net to control swing and stance movement, respectively, each equipped with a motivation unit (depicted by yellow circles). Connections with an arrow indicate positive ("excitatory") influences, connections ending with a T-shaped ending indicate negative ("inhibitory") influences. On the right, one sub-module (Swing-net) is shown in more detail, as it is implemented as a neural network (numbers refer to weights). Target angles serve as an input to the neural network and are stored in the component. Each of the three neural units inside the Swing-net controls the movement of one leg joint. Only one coordination influence is shown in the diagram. In this case, coordination influence 1 (see **Figure 1**) is acting between the hind and the middle leg. While the hind leg is in swing, the posterior extreme position (PEP) of the anterior leg is shifted backwards and therefore the stance movement is prolonged (Δ-PEP). For further details see Schilling et al. (2013b).

the case for all six legs (only two legs are depicted in **Figure 3**). These six "leg units" are in turn connected to a unit termed "walk" in **Figure 3**. This unit serves the function of arousing all units possibly required when the behavior "walk" is activated.

In the case considered here, the motivation unit network, a recurrent neural network, can adopt different stable states, or attractors, forming different overlapping ensembles. For example, all "leg" units and "walk" are activated during backward walking and during forward walking, but only one of the two units termed "forward" and "backward" and only 12 of the 24 end position memories are active in either case. The network is therefore best described as forming a heterarchical structure (for details see Schilling et al., 2013a). Such an "internal state" adopted by the network protects the system to respond to inappropriate sensory input. For instance, as a lower-level example, depending on whether a leg is in swing state, or in stance state, a given sensory input can be treated differently: stimulation of a specific sense organ (not depicted in **Figures 2–3**, but see Schilling et al., 2013b) leads to a levator reflex when in swing, but not during stance. In other words, the motivation unit network can be considered to act as a top-down attention controller. On higher levels, further internal states could be distinguished, as for example walking, standing still or feeding (for a more detailed discussion on how such a heterarchical network can be structured and learned see Cruse and Schilling, 2010).

The heterarchical structure sketched in **Figure 3** comprises a simple realization of neural reuse as proposed in Anderson's massive redeployment hypothesis (Anderson, 2010) as specific procedures are used in different behavioral contexts.

The system as described so far is a slightly expanded version of the earlier Walknet that represents a typical case of an embodied controller (1st order embodiment, c.f. Metzinger, 2006, 2014): Kinematic and dynamic simulations as well as tests on robots have shown that this network can control walking at different velocities, producing different insect gaits including the continuous transitions between the so called wave gait, tetrapod gait and the tripod gait, negotiating curves (Kindermann, 2002), climbing over obstacles (Kindermann, 2002; Dürr et al., 2004), and over very large gaps (Bläsing, 2006), and coping with leg loss (Schilling et al., 2007). Thus, Walknet exhibits a free gait controller where the gaits emerge from a strictly decentralized architecture. Application of this

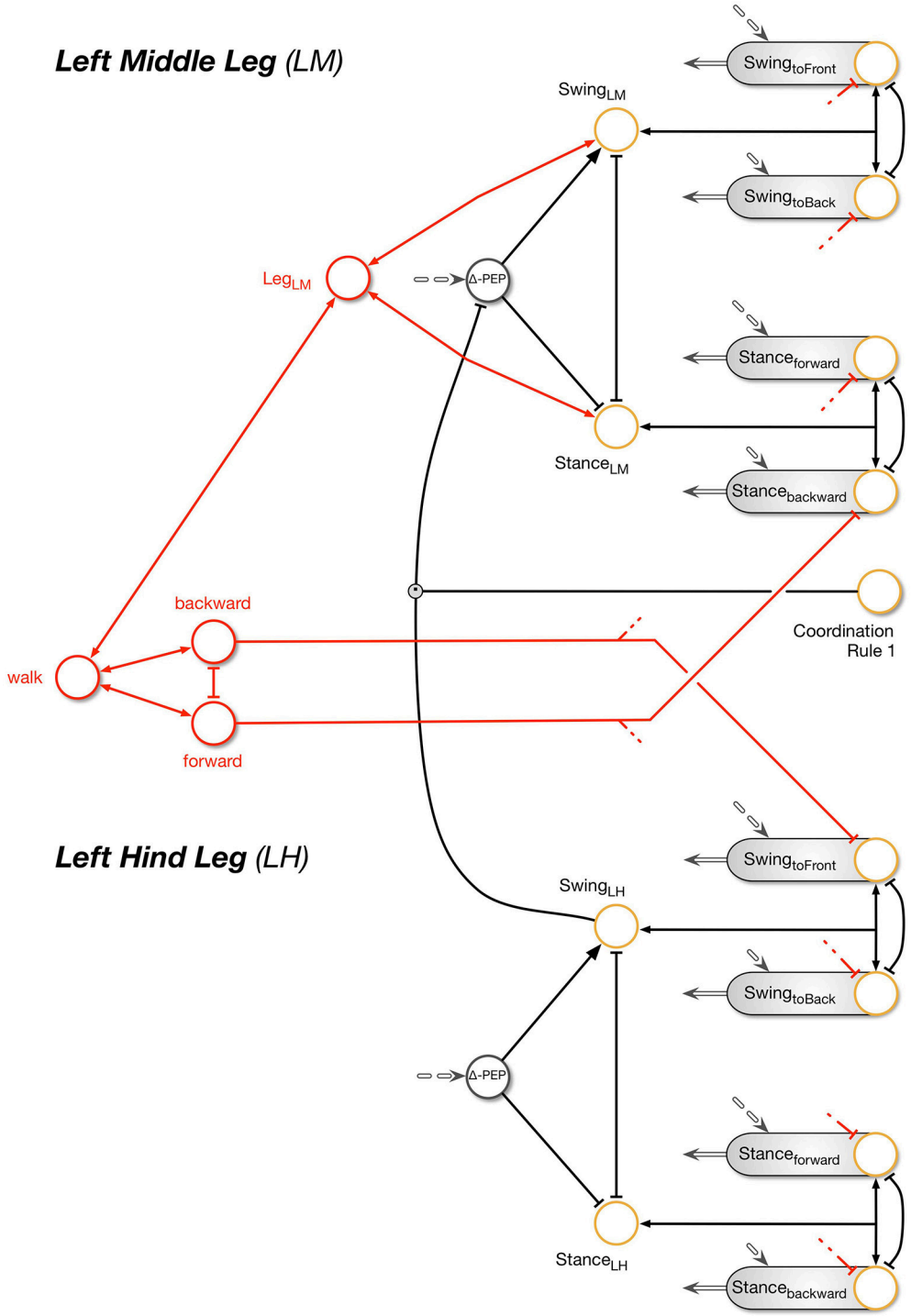


FIGURE 3 | The extended Walknet. Compared to the version shown in **Figure 2**, the ability to walk backwards has been introduced (not all details are shown here.). Each procedural element is equipped with a motivation unit (yellow circle). In addition, the coordination influences (only rule # 1 is depicted) can now be modulated by a motivation unit (yellow circle, coordination Rule # 1). Further motivation units are introduced (red connections and units) being arranged in a heterarchy—again only a fraction of the network is shown (see also **Figure 2**).

decentralized approach allows for a dramatic simplification of the computation by exploiting the loop through the world (including the own body). For example, trajectories of swing movements are not explicitly given, but result from the cooperation between the Swing-net and the "loop through the world," i.e., the sensor readings describing the current position of the leg joints. This

structure allows for immediate adaptation of swing trajectories to unpredictable disturbances. Similarly, the spatio-temporal patterns of leg movement ("gaits") are not explicitly specified but result from decentralized local coordination rules and the coupling of the legs via the substrate (see review Schilling et al., 2013b). This network has been tested in dynamic simulation (Schilling et al., 2013a,b) and applied to the robot Hector (Schneider et al., 2011; Paskarbeit et al., 2015). As will be shown in section Motor Planning: from Walknet to reaCog (Motor Planning), this modular structure is a crucial condition to allow recombination of procedural elements as required by a cognitive system.

Walknet with a Body Model

The control of the stance movement is a complex task which requires the coordination of multiple legs and joints. While local embodied approaches can deal with quite complex walking scenarios and disturbances (Schmitz et al., 2008), a purely embodied approach relying on the coupling through the body itself and local leg controllers has shown to become insufficient in other cases (Schilling et al., 2012). For example, stick insects are able to negotiate curves which can be very tight (Dürr, 2005; Dürr and Ebeling, 2005). In the case of curve walking, the different legs are producing quite different movements and are taking over different roles as there is, for example, a differentiation between inner and outer legs. To better cope with such problems, we apply an internal model of the body for the control of the stance movement (Schilling et al., 2012).

Body models are used for three different purposes [for a recent, comprehensive review see Morasso et al. (2015)]. First, inverse models have been applied (e.g., Wolpert and Kawato, 1998) to compute motor commands for given goal positions of an end-effector. The second task concerns the ability to predict the position of the end-effector when motor commands are known but not yet executed (Wolpert and Flanagan, 2001; Webb, 2004). In this case the body model is used as a forward model, for instance to overcome sensory delays. Third, even simple animals as insects use a high number of sensors, for example to measure joint positions or load. In order to exploit this redundancy (e.g., to improve inexact or even missing sensor data), the different sensory inputs have to be fused which requires a body model (Makin et al, 2008). Used for visual perception, the body model, mirroring the observed movement, is strongly related to mirror systems as found in animals (Rizzolatti et al., 1996) and in humans (Rizzolatti, 2005), and might be linked to the understanding of others (Loula et al., 2005).

Whereas, in other approaches usually an individual model has been required for each task and each behavioral element (Wolpert and Kawato, 1998), we use one simple holistic recurrent neural network that can cope with all three tasks. The body model used copes with the at least 18°C of freedom of the insect body (six legs of 3°C of freedom each).

The complexity of the six-legged walker is distributed in the body model into interacting submodels (see **Figure 4**, Schilling and Cruse, 2007). On the lowest level, each leg is represented as a detailed model of all the leg segments and connecting joints [**Figure 4B**, right; for details see (Schilling, 2011; Schilling et al., 2012)]. These leg models are integrated on a higher level in a model of the central body, where each leg is only represented by a vector pointing from the body segment toward the tip of the leg (**Figure 4B**, left; for details see Schilling and Cruse, 2012; Schilling et al., 2013a). As this network is based on the principle of pattern completion, any input vector given to the network—may it correspond to the input required for a forward model, an inverse model, or a sensor fusion model—provides an output that, after relaxation, leads to a coherent body state. This means that in any case the kinematics represent a geometrically correct body position. Next, we will explain how this body model can be integrated into the architecture of Walknet.

Figure 5 illustrates how the body model is integrated into the network. As depicted in this figure, the internal body model comprises an independent system, which may receive sensory input and/or motor commands. In turn, it provides sensory

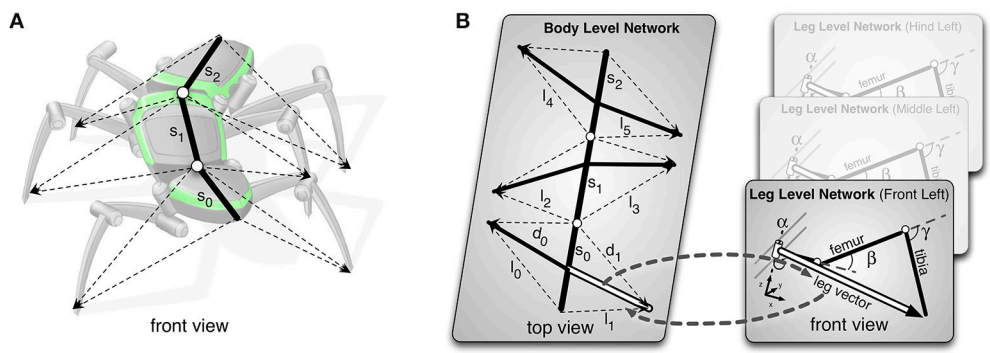


FIGURE 4 | The body model. (A) illustrates how the body model (black) represents the body of the robot (gray). **(B)** The Mean of Multiple Computation (MMC) body model for the six-legged walker is divided into two layers. The lower layer contains six networks, each representing one leg (for details see Schilling et al., 2012). The upper layer represents the body and the six legs, which are only represented by bold vectors pointing toward the tip of each leg as shown in **(B)**, left. On this level the leg is described with reference to the respective body segment. Both layers are connected via the shared leg vectors (marked by the double-lined vectors of the left front leg) and are implemented as recurrent neural networks.

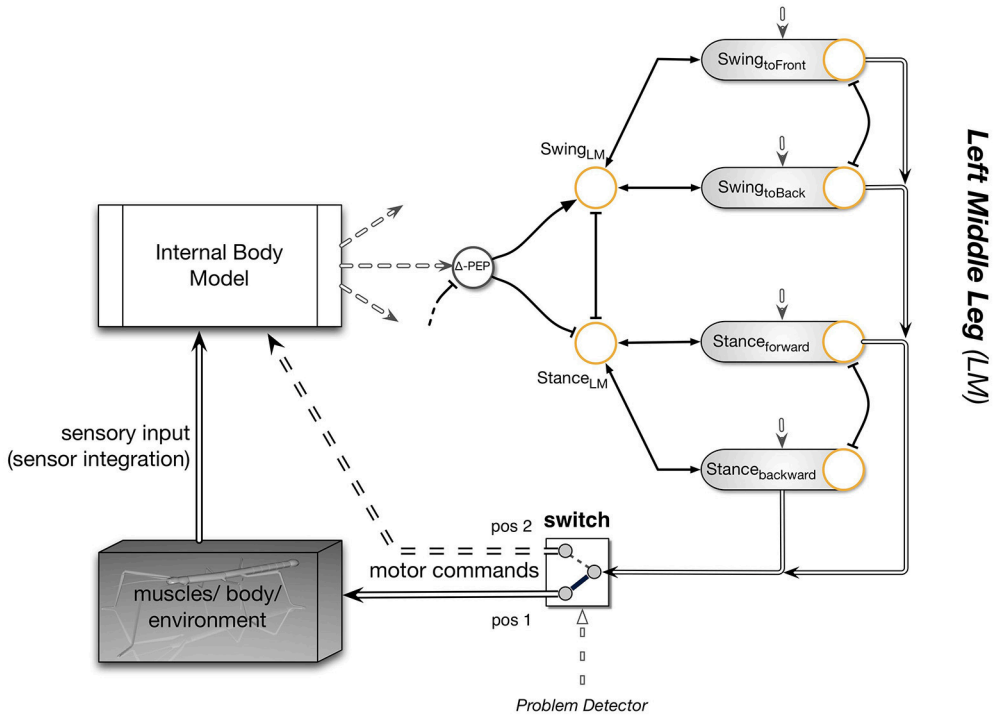


FIGURE 5 | The first step to reaCog: Walknet expanded by an internal body model. Only a part of Walknet as shown in **Figures 1**, **2** is depicted (the left middle leg). During normal behavior, the Internal Body Model (upper left) serves perception. The body provides proprioceptive input (e.g., joint angles from the legs) that is integrated within the body model to form a coherent sensory experience. With the switch in position 1, the network represents a reactive controller. If the system runs into a problem, the switch is flipped from position 1 to position 2 and the motor control (double-lined arrows entering the switch on the right) is routed not to the body anymore, but instead to the body model (dashed double line). This circuit is used for internal simulation and predicts the sensory consequences of the action. The body model is now driven by the motor commands predicting the sensory consequences instead of integrating them. For further explanations see text.

signals or motor commands to the reactive structure Walknet. The body model can be used for controlling the motor output of the stance behavior in complex walking scenarios. In this case it is part of the reactive controller (in **Figure 5** the switch has to take position 1). Using the body model as an inverse model, movement of the legs during stance can easily be controlled by applying the passive motion paradigm (Mussa-Ivaldi et al., 1988). Like a simulated puppet, the internally simulated body is pulled by its head in the direction of desired body movement (**Figure 5**, sensory input). As a consequence, the stance legs of the puppet follow that movement in an appropriate way and the changes of the simulated joint angles can be used as commands to control the actual joints. Therefore, if such a body model is given, that represents the kinematical constraints of the real body, we obtain an easy solution of the inverse kinematic problem, i.e., for the question how the joints of legs standing on the ground have to be moved in concert to propel the body (for details and application for the control of curve walking see Schilling et al., 2012, 2013a).

In the next section we will introduce a fundamental expansion termed "cognitive expansion." The complete network, as we will argue, shows how cognitive properties can emerge from a system heavily relying on reactive structures, why we will call this network reaCog.

Motor Planning: from Walknet to Reacog

The General Idea

To be able to implement the faculty to plan ahead, the neuronal system has to be equipped with a representation of parts of the environment (Schilling and Cruse, 2008; Marques and Holland, 2009). As it has been argued that, as seen from the brain's point of view, the body is the most important part of the environment (Cruse, 2003), a neural representation of the own body is the first step to take. Later, this body model of course has to be extended to include aspects of the environment as are tools extending the body, objects to be handled or an environment to interact with, for example obstacles to be climbed over or to be circumvented.

As mentioned the body model introduced in the previous section can be also used for prediction. Therefore, the body model will be applied to allow the system for being capable of planning ahead through internal simulation.

The basic idea that will be detailed in this section is simple. In short, we will apply the following two-step procedure. If a problem occurs, which means that the ongoing behavior cannot be continued when using only the existing reactive controller, the behavior will be interrupted. The system will then try to come up with new behaviors by recombining the existing procedural elements in a new way, i.e., not envisaged in the current context. A procedural element is characterized by a section of the network

that can be controlled by a motivation unit (as shown in **Figure 3**, red and yellow circles). The properties of the new combination will then be tested by using the internal body model instead of the real body, the former now exploiting its faculty to serve as a forward model. If the new combination turns out to be successful, it will be applied to control the behavior. If not, the system will search for another new combination.

For better illustration, we will use the following example: Imagine the case that one—say the left hind leg—has been moved far to the rear and now receives the signal to start a swing movement, i.e., to lift the leg off the ground. If the two neighboring legs—the left middle leg and the other, right, hind leg—accidentally are positioned far to the front, lifting the left hind leg might lead the body falling to the rear (**Figures 6A,B**).

Interruption of Behavior

To avoid tumbling over backwards, the system must be able to detect that it is running into trouble. Therefore, one or several systems are necessary that are able to detect that there is a problem. While there are different biologically plausible solutions (e.g., using load sensors as found in the insects), we chose as a simple approach a stability sensor which is activated in case the leg would be lifted,. In the example scenario this detector becomes activated immediately after the motivation unit swing of the hind left leg becomes activated, i.e., before the animal would fall backwards onto the lifted leg.

If a problem has been detected by any detector the system must (i) interrupt the ongoing behavior and (ii) be able to change from the state "perform behavior" to the state "simulate behavior." To this end, we have introduced a switch as shown in **Figure 5**. By moving the switch from position 1 to position 2, the output of the leg controller—which is normally (position 1 of the switch) routed to the motor output to influence the body—is now instead routed directly to the body model. Thereby the position of the real body is kept fixed, i.e., the ongoing behavior is interrupted (Hesslow, 2002) is providing a biological account for this decoupling which has also been found in insects (Bläsing and Cruse, 2004), but the internal body model can perform the movements determined by the reactive controller. As in the case of actively moving the body, the output signals of the body model, in particular the angular values describing the position of the leg joints, are given to the reactive procedures. In this way the loop is closed and the system can internally simulate the behavior by moving the body model instead of the real body. Note that modules of the reactive procedures as are Swing-net and Stance-net are still active as is the case in Walknet. 2.2.3 Coming up with a new solution.

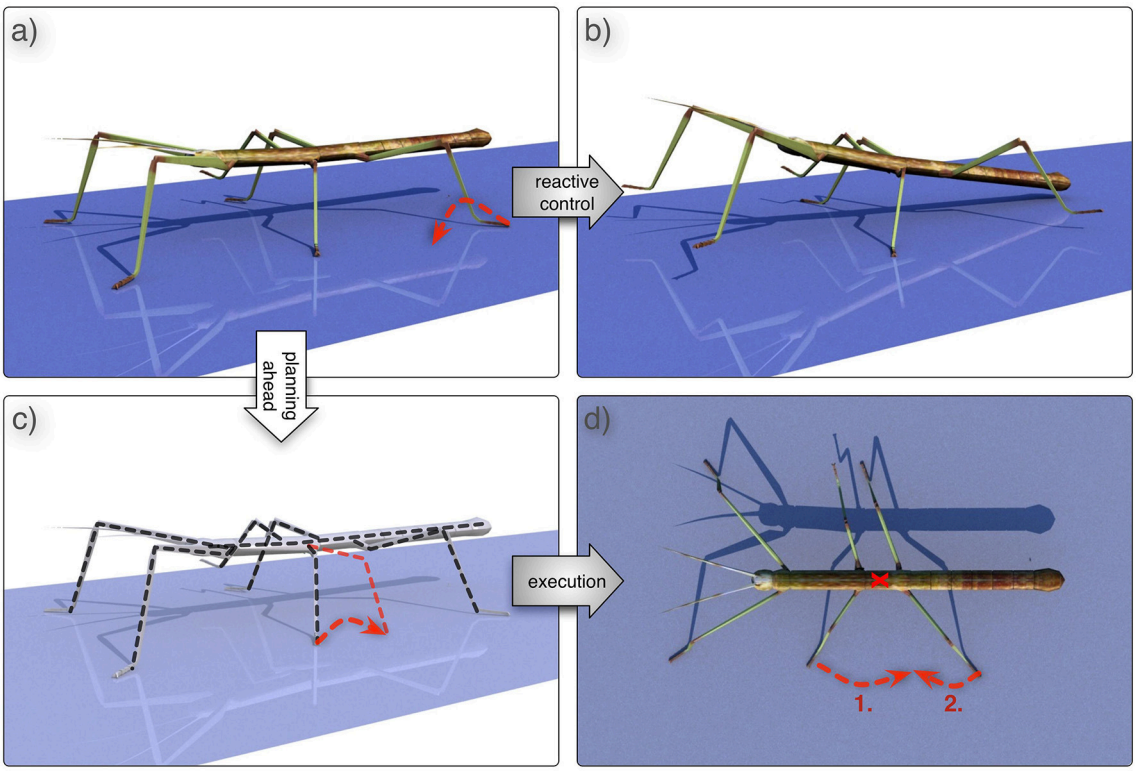


FIGURE 6 | A problem and a possible solution. (A) shows a posture in which the animal would fall over when trying to lift the left hind leg (dashed red arrow), because the anterior, middle, leg and the other hind leg are too far to the front. The result is depicted in **(B)**. However, the problem detector detects a problem before the left hind leg is actually lifted, the cognitive system should start searching for a solution through mental simulation **(C)**. The system might come up with the idea to perform a backward swing with its left middle leg and afterwards proceed walking. After successful testing in simulation **(C)**, the plan can be executed in reality **(D)**, i.e., first swinging the middle leg backwards and then swinging the hind leg to the front while continuing normal walking.

This switch given, it appears of course not very interesting to simulate exactly the behavior which has just led to the problem. Instead, it is necessary to test new, currently not available solutions. Therefore, the signal from the problem detectors is not only used to move the switch, but also to start the search for a new solution. To allow for this faculty, reaCog requires a further fundamental expansion.

The main idea is that for internal simulation a new behavioral element has to be selected. This new behavioral element may be selected also from procedures not belonging to the current context. How is this solved by reaCog? In **Figure 7**, the upper, left part (i.e., without SAL net, WTA net, and RTB net) shows a simplified version of the network as presented in **Figure 5**. The expansion depicted at the right side enables the system finding "new solutions" and then testing their qualification to solve the problem. This expansion—that we will call "cognitive expansion" or, as will be motivated in Section Discussion and Conclusions), "attention system"—contains three additional layers, a spreading activation layer (SAL, red circles), a winner-take-all layer (WTA, green circles) and a remember-tested-behavior (RTB, blue circles) layer with identical number of units each. In addition, there is a small network termed Global Phases (lower part of **Figure 7**).

At the bottom, Global Phases, the structure is illustrated that organizes the temporal sequence of finding a behavior as

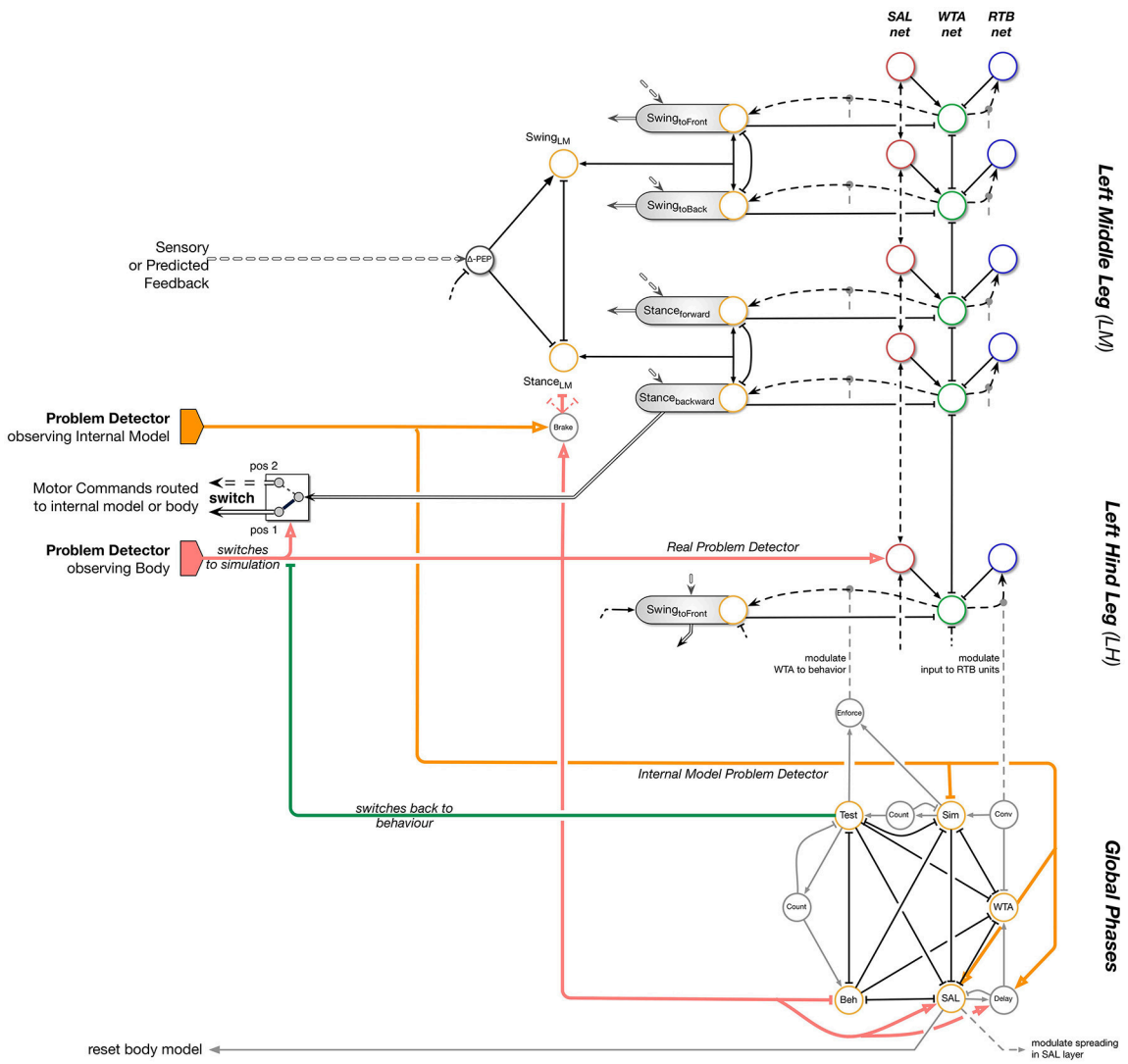


FIGURE 7 | ReaCog: Walknet plus cognitive expansion. This figure shows an extension of the Walknet structure presented in **Figure 5**. The motivation unit structure (yellow, e.g., Swing, Swing_toFront) is replicated on the right side, termed attention system, in three ways. There is a Spreading-Activation-Layer (SAL, red circles), the WTA layer (green circles), and the remember-tested-behavior (RTB units, blue circles) layer. The problem detector (red and yellow, the latter for the internal model) not only activates the switch, but also the spreading activation layer (SAL; red arrows) The activated spreading activation layer units activate their partner units in the WTA network. The winner of the WTA is activating the corresponding motivation unit (dashed black arrows) and the corresponding motor program will be carried out using internal simulation. Note that the connections within the WTA layer are not completely depicted.

a solution to a novel problem. Additional units (gray circles) show temporal properties and are used to organize the switching between stages as explained in the text. Units "count" represent a specific time delay.

Cognitive Expansion

In the following we will explain the function of the cognitive expansion as depicted in **Figure 7**. The goal of the cognitive expansion network is to search for a new procedural element that allows for a solution of the current problem. The first step is to look for behavioral elements existing in the memory, which are, however, not activated in the current context. As will be explained, only such procedural elements can be selected that can be activated by a motivation unit. Second, the possible contribution of this additional memory element will be tested by internal simulation.

How is this done? The units of the SAL (**Figure 7**, red circles) receive input from morphologically neighboring problem detectors (in **Figure 7**, one example is depicted by a bold, red circle). Neighboring units are connected by positive weights. In this way, an activation arising from a problem detector is spread over the SAL roughly corresponding to a circular wave starting at the position of the unit excited by the problem detector. Further, there is noise added to the units of the spreading activation layer. The middle layer is representing a winner-take-all network. The units of the WTA layer (**Figure 7**, green circles) are activated by the corresponding partner units in the SAL layer. In addition, already active behavioral elements, i.e., their active motivation units, are inhibiting their counterparts in the WTA-layer (**Figure 7**, black solid line with T-shaped end). In this way, currently active behaviors are prevented from being selected for testing in internal simulation. Through the winner-take-all process the units are inhibiting each other in a way that only one unit remains active when the network settles. For the third, the right hand layer, there is again a one-to-one connection to the WTA-layer. These RTB units (**Figure 7**, blue circles) store which of the WTA units have already been tested in an earlier internal simulation run.

The different procedural elements of Walknet and their motivation units are anatomically arranged in a way that this arrangement coarsely reflects the morphological ordering of the legs (**Figure 1**, left). Consequently, the motivation units of neighboring legs as well as the partner units of the Spreading Activation layer (SAL) and of the winner-take-all (WTA) layer are neighboring, too, and thus form some kind of somatotopical map. Thus, the problem detector is not only signaling the problem, but in addition also carries some information where the problem occurred. In this way, the search for a new behavior is not purely random, but follows some heuristics,—there is some probability that a solution may be found morphologically near the cause of the problem—which may accelerate the searching process.

What is the functional role of these three additional layers forming an expansion that we will call "cognitive expansion" or, as will be motivated later in the discussion (Section Discussion and Conclusions), "attention system"? Assume that in our example (**Figure 5**) the problem detector situated in the left hind leg has been activated (**Figure 7**, bold red arrow, starting at the left). As noted earlier, this signal moves the switch from position 1 to position 2 to route the motor output to the body model instead of the body itself. Thereby the ongoing behavior is interrupted. In addition this signal activates one (or several) neighboring units of the Spreading Activation layer. **Figure 8** illustrates the sequential activation of WTA layer, and RTB layer.

The winning WTA unit activates its motivation unit and as a consequence, the corresponding—new—procedural element. After the WTA net has made its decision and has activated the motivation unit of a procedure normally not used in the actual context, simulation using the internal body model will be started to test the contribution of this new procedure. Note that therefore a problem detector is also required inside the internal model which functions in the same way, i.e., it observes static stability of the (internally simulated) body (**Figure 7**, bold yellow arrows).

If during the internal simulation no problem detector becomes active, the procedure appears to be a suitable solution for the given problem. Thus, the solution is found following a search

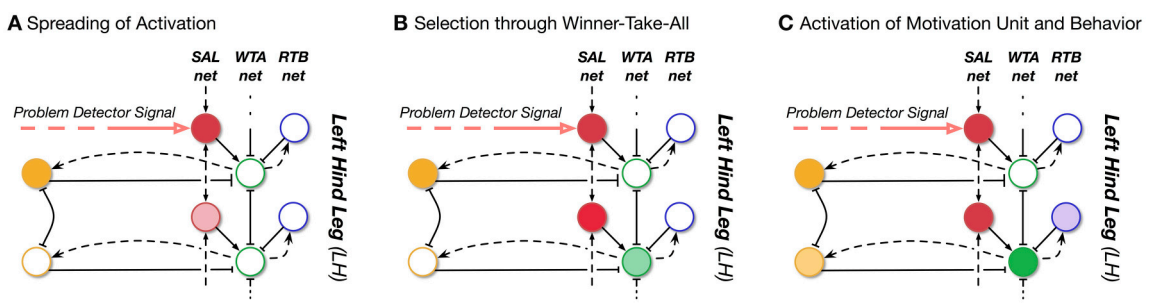


FIGURE 8 | Illustration of the sequential changes of activation of SAL, WTA, and RTB units. When a problem occurs, the problem detector, on the one hand stops the execution of current behavior (not shown). On the other hand, it induces activity in the spreading activation layer (SAL, red) which indicates where the problem occurred. The activation is spreading vertically in the SAL. Each SAL unit excites its corresponding WTA unit. Importantly, currently active motivation units (yellow) inhibit the WTA units (green units). The WTA units compete among each other producing one winning unit which in turn activates the corresponding motivation unit and behavior. The units in the RTB layer (blue) represent which behavior has been active once during the simulation process and will inhibit a future activation during the WTA selection process.

driven by a heuristic including noise (given to the SAL units). As a next step, this solution is tested for being mechanically appropriate. In this case the switch is set back to position 1 and the corresponding behavior will then be applied in reality. By setting back the switch the real body will provide the sensory input. As the winning WTA unit is still active (thus representing a short term memory), the newly selected procedure will be executed. If, however, already during the internal simulation this "new solution" has proven not to be a solution—defined by a problem detector of the internal model becoming active—the search for a solution will be continued further. To this end, the internal model will be reset to the current state of the body. Then, the SAL net will continue the spreading of its activations and a new behavior will be selected by the WTA-net. In this way the procedure will be repeated until a solution is found.

When the new solution is tested in reality, there are still two possibilities to be considered. If the realization of the proposed solution is successful, behavior continues. However, the solution may also turn out not to be realizable. This might for example happen because the body model does not simulate the physical properties of the body (and the environment) well enough. In this case a—possibly different—problem detector will be activated by starting again a new search procedure, with the internal body model being reset to the current real state of the body as given through the sensors.

In the remainder of this section, the structure that controls the temporal sequences sketched above is explained in detail. As indicated in the lower part of **Figure 7**, the complete procedure is controlled by five specific motivation units, Beh, SAL, WTA, SIM, and Test forming the center of the Global Phases network. These units are coupled via mutual inhibition (not depicted in **Figure 7**) and in part by transient, i.e., high-pass like, units (**Figure 7**, gray units and connections in the lower part).

During normal, i.e., reactively controlled walking the motivation unit "Beh" is active, thereby inhibiting the other four motivation units. If a problem is detected, the problem detector is activated which in turn inhibits the ongoing behavior (motivation unit "Beh") and activates the "SAL" motivation unit. In addition, the switch is moved to bypass the physical body (the switch might be realized by further mutually coupled motivation units, not shown in **Figure 7**) and the current forward movement of the robot is inhibited for some time that corresponds to duration of about one step of the leg (i.e., 100 iterations). This allows sufficient time to test movements before starting to continue forward walking. After a given time required for sensible spreading of activations (e.g., two iterations, triggered by the "Delay" unit shown in gray in **Figure 7**), the SAL motivation unit is inhibited and the WTA motivation unit is activated instead. The relaxation of the WTA net may require a variable number of iterations. A simple solution is to introduce one unit observing the convergence of the WTA-network ("Relax"). This unit is activated as soon as the first unit of the WTA layer has reached a given threshold, representing the winning unit.

Only after a winner is detected ("Relax" in **Figure 7**), the "WTA" motivation unit is inhibited and the simulation is started (motivation unit "SIM"). If, after a given time of internal simulation (we use 400 iterations which equals 4 s or about three to four step cycles), no problem occurred, the motivation unit "Test" will be activated instead to start the real behavior. If during the test of the real behavior the problem occurs again or a new problem is detected (in contrast to the situation during simulation), the behavior is inhibited and the "SAL" motivation unit is activated again. If however the behavioral test is successful, too, the motivation unit "Beh" is activated (and the motivation unit "Test" inhibited) to allow continuation of the normal behavior. In contrast, if during simulation a problem is detected, the simulation is interrupted (motivation unit "SIM" is inhibited) and instead the motivation unit "SAL" is excited again to search for a new "idea." The temporal order of activation of the different motivation units of the Global Phases network is controlled by dedicated connections running in parallel to the mutual inhibitory connections (**Figure 7**, gray) of all these units,

Importantly, each internal simulation has to start from the real situation, i.e., the situation that led to the problem. Therefore, the internal body model as well as the control system have to be reset to this state before a new internal simulation is started. This reset is triggered during the "SAL" stage. As the body did not actively move during internal simulation, the current posture and sensor readings can be used to reset the internal body model. It takes the reactive part of the control system only a couple (one or two) iterations to converge to the original state. It turned out that the internal state does not have to be stored explicitly.

The complete procedure controlled by the Global Phase network corresponds to what has been termed "incubation" and "verification" (Helie and Sun, 2010), and is similar to the "note-assess-guide" strategy or "metacognitive loop" as introduced by Anderson et al. (2006). In a mathematical analysis applied for example to logic reasoning systems the latter authors could show that introduction of such a strategy indeed improves the behavior of the complete system. The complete period, during which the body is fixed and the body model is used for internal simulation, may correspond to what Redish (2016), referring to Tolman, has termed "vicarious trial an error."

RESULTS

Simulation Results for the Example Scenario

In this section, we will show a dynamic simulation of the reaCog system. The example illustrates the faculty of reaCog to find new solutions to a current problem using its capabilities for planning ahead. (In this study there is no physical robot used yet, but it is represented by a dynamic simulation.). **Figure 6** shows an awkward posture. This configuration can become problematic as the left hind leg is already very far to the back and cannot move further back. Therefore, in this situation the left hind leg has to produce a swing movement. If the position of the left middle leg and right hind leg are positioned very far to the front, lifting the left hind leg can lead to instability, because the center of mass is placed quite far to the rear, between the hind legs. A sensible solution in our paradigm (**Figure 6**) might be the activation of the Swing_toBack module of the left middle leg: A backward step

of the anterior middle leg might allow this leg to take over the body weight, and—as a consequence—afterwards allow lifting of the left hind leg. Thereby, continuation of walking may become possible.

In normal walking the reactive part of the controller is not ending up in such a strange posture. Therefore, we had to introduce an external disturbance to make the system tumble. To this end, the placements of the left middle leg and right hind leg will be changed in a way that during swing movement the target position is pushed further to the front (by a third of a step length). Such a strong change might occur in insects when climbing over irregular ground. When there is no foothold, the insects are starting searching movements to the anterior in order to find a foothold (Dürr and Krause, 2001; Bläsing and Cruse, 2004; Schütz and Dürr, 2011) which can be quite far to the front. This does not pose a problem for the stick insect as stability is strongly supported through the ability to attach the feet to the ground. As the robot cannot use this method, he has to find another solution (for example the one sketched in **Figure 6**).

In the following, with help from **Figures 9**, **10**, we will explain how the system deals with this intervention. **Figure 9** (middle panel) shows a footfall pattern which illustrates the swing movements of the legs over time. A leg which is in swing phase is marked as a black (or red) bar. For the medium velocity chosen a gait is emerging that can be seen in the stepping pattern in the left part of the figure. From a tripod-like starting posture the robot converges more toward a fast tetrapod-like gait (at about 500 iterations). The lower part of **Figure 9** shows still images of the dynamic simulation (see **Supplementary Material Videos 1**, **2**), whereas the upper part provides a top view of the robots' (or internal models') configuration. The upper part shows four specific snapshots of the posture of the walker (top view) facing to the right. Only legs in stance phase, i.e., legs which support stability are depicted.

For the same run, **Figure 10** illustrates the position of each leg over time. The position is plotted on the ordinate showing the movement of the leg (green lines, swing movements during forward walking are pointing upwards; stance movements are going into the opposite direction).

The jumps in the position of the legs are due to the switching from the real robot to the internal model required to reset the internal model. Colors are used as in **Figure 9**. For further explanations see text.

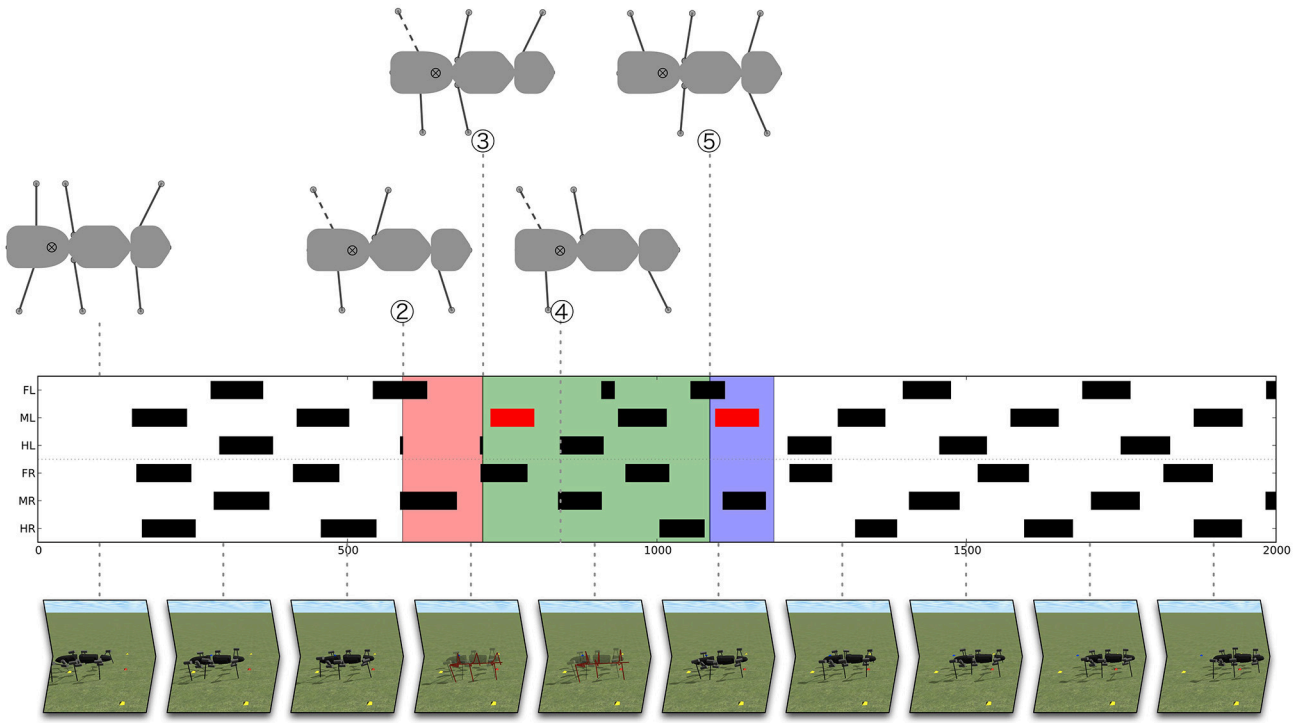


FIGURE 9 | Solving the problem illustrated in Figure 6: Foot fall patterns. The middle panel shows the footfall pattern of the hexapod over time (black/red bars indicate swing movement of the leg). The upper panel shows some critical configurations of the robot (or, during internal simulation, the configuration of the internal model). The robot is walking from left to right. In three cases, the left hind leg is shown as a dashed line indicating that it is supposed to start a swing movement. The lower panel illustrates the behavior by screen shots taken from the **Supplementary Material Videos 1**, **2**. The robot starts with a tripod-like leg configuration and converges to a fast tetrapod gait. The problem is detected at (#2). The section highlighted red represents an unsuccessful internal simulation [ending in an unstable configuration again as shown in (#3)]. The second internal simulation, highlighted green [starting at (#3)], turns out to be successful and solves the problem (backswing of the left middle leg, depicted by red bars, (#4) shows the new posture before the start of the forward swing movement of the left hind leg). Highlighted blue is the application of this solution to the robot (starting at (#5) which shows the robot posture at the beginning of the backward swing movement of the left middle leg). This final test is successful, too, and the robot continues to walk (⊗ indicates center of mass).

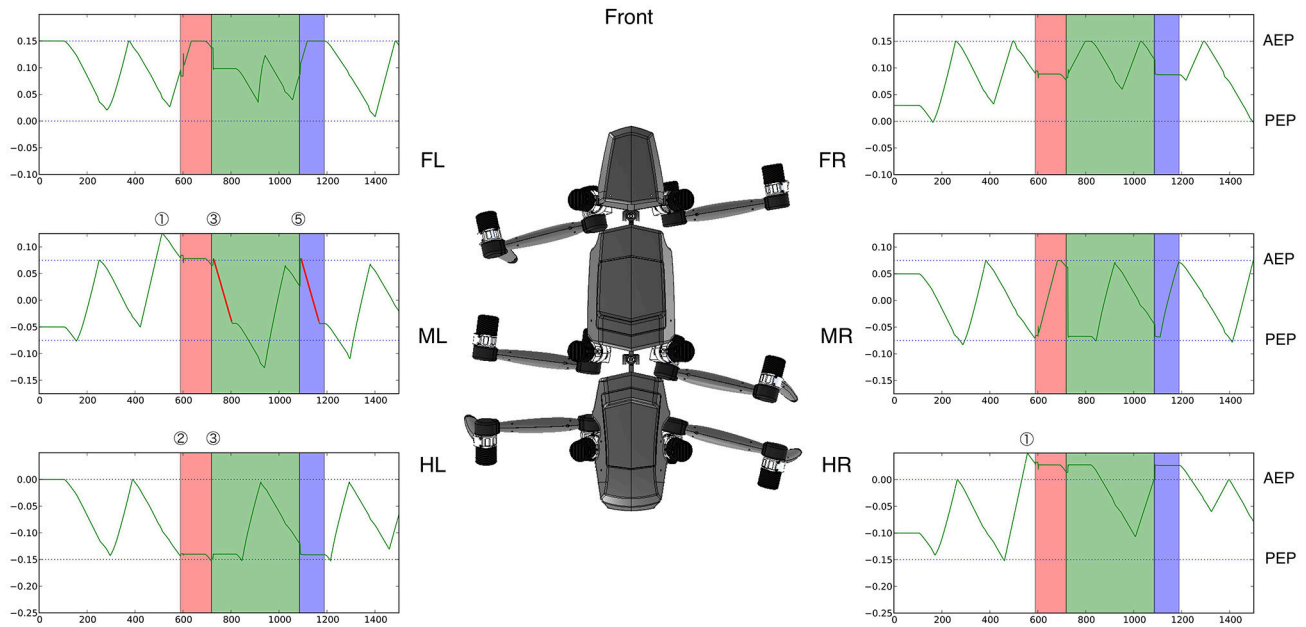


FIGURE 10 | Solving the problem: Position of the individual legs over time. Green lines show the position of each leg over time—positive values are toward the front of the walker. Ordinate is given in cm with the origin fixed to the COM of the robot. The blue dashed lines indicate the average extreme positions: The Anterior Extreme Position (AEP) is the target position for the swing movement and is fixed during forward walking. The Posterior Extreme Position (PEP) indicates the position at which a leg controller initiates a swing movement on average and switches from stance to swing (note that the coordination rules act on the PEP and shift the PEP forward or rearward to organize the overall behavior which is not shown in the figure). Shortly after the left middle and right hind leg performed swing movements that point very far to the front of the working range (#1), the walker becomes unstable (#2) when trying to lift the left hind leg. Therefore, internal simulations are started (highlighted in green and red) during which motor commands are routed to the internal body model, the leg positions of which are shown. First (highlighted red), an unsuccessful behavior is tested: a stance movement which has initially no effect as the agent is stopped. But when the agent accelerates again (after 100 iterations) the problem is still present and the agent becomes instable (#3). As a second trial, a backward swing movement of the middle left leg is tested via internal simulation (green highlighted area; the swing movement in the unusual direction is plotted in red). Afterwards (#5) the solution found is tested on the real robot (highlighted in blue) showing that walking continues successfully.

As mentioned, we forced the robot into an awkward posture in such a way that the swing movement of the left middle and right hind leg moved very far to the front of their working range, i.e., beyond their normal AEP. Next, the left hind leg marked by a dashed line in **Figure 9** is supposed to start a swing movement. The center of mass would then not be supported anymore by the left middle leg and right hind leg [**Figures 9**, **10** (2), after 580 iterations]. Therefore, the system would tumble backwards.

As a consequence, the problem detector is activated, which stops the overall movement of the robot and triggers the cognitive expansion which then starts motor planning. In the example shown in **Figures 9**, **10** the robot first selects a stance movement in the left hind leg (due to the somatotopical neighborhood, see **Figure 7**, in SAL layer). This stance movement is then applied in internal simulation.

As a result, an unsuccessful internal simulation can be observed (highlighted in red) (2)–(3), which is interrupted when the left hind leg should be lifted, because this action would again lead to an instable configuration of the internal body model [see upper panel, (3)]. Note that during the time highlighted in red (and green, see below) the robot is not moving. Only the internal model is used to provide predictions of the movements.

As a consequence, a second iteration of the cognitive expansion is invoked (this section is highlighted green, as it turns out to be successful): First, activation is further spread in the SAL layer. Second, a behavior is selected in the WTA layer which has not yet been tested. And third, the behavior is applied as internal simulation.

For this second internal simulation, the internal body model and control system have to be reset initially. To this end, it turned out to be sufficient to update, first, the internal model with the values from the real robot structures (this is the starting condition required for the internal simulations; see **Figure 10**, at the border of the red and green section, the position of the leg in the internal model jumps back to the original position of the robot leg). Second, as the control system is behavior-based it depends on the sensor state represented by the current position of the robot. This state can be enforced onto the control system so that the system converges back to its behavioral state.

In the simulation run shown, the behavior selected next is a backward swing movement of the left middle leg (depicted in **Figure 10** by a red line for the position of the left middle leg; correspondingly, in **Figure 9** the swing movement backwards is shown as a red bar). As illustrated in the parts highlighted in green, again the forward movement of the body is interrupted for

some time. During this time the newly selected behavior is tested by internal simulation. When the system starts to accelerate again, the left middle leg now being placed further to the rear helps to support the robot. When the left hind leg starts to swing, the left middle leg is ready to take over and to support the body (shown in **Figure 9** in the upper panel in the body posture at #4 at around 800 iterations). The internal simulation runs further for a given time (here we used additional 300 iterations) in order to guarantee that normal walking can be continued.

When the internal simulation was successful the behavior selected (which is still stored in the WTA layer) will be applied on the (simulated) physical system (see #5 and blue area in **Figures 9**, **10**). This part is still regarded as a test of the behavior. This test is necessary because internal simulation and robot can of course lead to slightly different results which over time might accumulate. For example, in **Figure 9** the behavior of the right middle leg differs between internal simulation and testing the behavior on the robot. The right middle leg is very close to its posterior extreme position and on the verge of starting a swing movement. In both cases, the robot is standing still and not supposed to move further forward. But in the case of the robot (highlighted blue), a small passive movement would be sufficient to initiate a swing movement. Nonetheless, as can be seen from the footfall pattern, the application on the robot is also successful and the system converges to a stable gait pattern. This stresses the robustness of the underlying control approach and highlights how important it is that planning and control are tightly intertwined. In the blue area and beyond, **Figure 10** shows the movements of the leg of the real robot. Immediately after the new behavior has been induced, one can observe how the phases of the individual leg controllers are rearranged. For example, the right front leg is forced to make an early swing movement after the right middle leg has finished its swing movement (see Schilling et al., 2013b). But already after a very short time, a single step of the robot, a stable tetrapod-like gait emerges (as can be seen in **Figure 9**).

The example illustrates the faculty of reaCog to activate behavioral elements out of context in order to find a solution to a current problem. As illustrated, the system (reaCog plus robot) manifests an impressive stable behavior, although various deviations from normal walking behavior can be observed during the complete process.

Simulation Series on Disturbed Walking

For a more quantitative evaluation of the performance of the reaCog architecture we performed two additional series of simulations to illustrate the contributions of the different parts of the system. On the one hand, there is the underlying reactive and biological inspired control system (based on Walknet Schilling et al., 2013a). On the other hand, when running into stability problems the cognitive expansion has been introduced which can take over in order to reconfigure the posture in a way that allows to continue stable walking.

Following the approach presented above in detail, we again used the repositioning of a leg during swing movement which means that, for a selected swing movement, the target position is shifted to the front. This represents a quite natural example disturbance as the insects are often climbing through twigs that do not provide many footholds. As a consequence, insects perform searching movements that may shift the end position of the swing movements further to the front.

As a first series of simulations, after a randomly chosen point in time (during the first 10 s of walking) one leg is selected randomly for which the next swing movement is shifted to the front (about 5 cm which equals a third of a complete step length). In this way, different legs are affected in different walking situations. We ran 100 different simulations, therefore each leg was targeted multiple times and in the different stages of the 10 s of walking. As a result, when only one leg is targeted the reactive control system showed to be sufficient and the walker never got unstable independent of which leg was shifted. For all simulations, walking continued for at least 5 more seconds after the disturbance. In most cases, already after one subsequent step the control system has established again a stable walking pattern. Only for an early change in a front leg this requires two stepping cycles. Stability is accomplished mainly through compensating the leg shift. While the shifting of the target position would prolong the next step for the respective leg, the local coordination influences force the leg into an earlier lift-off in order to compensate. Detailed results are provided as **Supplemental Data 1** in Supplementary Material which show for each of the different legs (front, middle, and hind leg) a single run as an example. As can also be seen in the data, the walking pattern emerges quite early in the first or the second step.

As a more severe disturbance, we performed a series of simulations in which two legs were targeted. Again, after a randomly chosen point in time (during the first 10 s of walking) two legs are selected randomly for which the next swing movement is shifted to the front (about 5 cm which equals a third of a complete step length). We performed 100 simulation runs with all kind of combinations between legs multiple times. As already discussed for the example shown above (Section Simulation Results for the Example Scenario), in this case the reactive biologically inspired control system may run into unstable situations that require to stop the walking behavior to avoid that the robot would topple over. In the following we provide results on for how many cases the system struggled with stability and how the cognitive expansion was able to deal with those situations. Overall, there are eight instable situations which were caused by a disturbance of a middle and the diagonal hind leg (a case as described in detail above, Section Simulation Results for the Example Scenario). For these eight simulation runs the cognitive expansion had to take over and has found a solution in all instances. The system always became instable when the other (non-disturbed) hind leg tried to initiate a swing movement. Interestingly, different solutions have been found. On the one hand, a rearrangement of the legs could be observed in a way that one leg was moved backwards and unload the non-disturbed hind leg which afterwards was able to initiate a swing movement. This was accomplished either through moving backwards the anterior middle leg or the contra lateral hind leg. On the other hand, we observed two cases in which the slowing down of the walking speed of the complete system was already sufficient to solve the problem as during the slowing down a swing movement

could be terminated which provided enough support for the walker.

These results show that the cognitive expansion is able to find different suitable solutions. Note, that the solution disrupts the coordination pattern of all the legs. Only together with the reactive system and the coordination rules the system is able to select a movement which enables stable ongoing walking. In some instances the system discarded solutions which we, on a first guess, would have assessed as possible solutions, but which later-on run into conflicts.

RELATED WORK

In this section, we will compare reaCog as a system with related recent approaches in order to point out differences. While there are many approaches toward cognitive systems and many proposals concerning cognitive architectures, we will concentrate on models that, like reaCog, consider a whole systems approach. First, we will deal with cognitive architectures in general. Second, we will briefly present relevant literature concerning comparable approaches in robotics, because a crucial property of reaCog is that it uses an embodied control structure to run a robot.

Models of Cognitive Systems

Models of cognitive systems generally address selected aspects of cognition and often focus on specific findings from cognitive experiments (e.g., with respect to memory, attention, spatial imagery; review see Langley et al. (2009), Wintermute (2012). Duch et al. (2008) introduced a distinction between different cognitive architectures. First, these authors identified symbolic approaches. As an example, the original SOAR (State, Operator, and Result; Laird, 2008) has to be noted, a rule-based system in which knowledge is encoded in production rules that allow to state information or derive new knowledge through application of the rules. Second, emergent approaches follow a general bottom-up approach and often start from a connectionist representation. As one example, following a bottom-up approach, Verschure et al. (2003) introduced the DAC (Distributed Adaptive Control) series of robot architectures (Verschure et al., 2003; Verschure and Althaus, 2003). These authors initiated a sequence of experiments in simulation and in real implementation. Verschure started from a reflex-like system and introduced higher levels of control on top of the existing ones which modulated the lower levels and which were subsequently in charge on longer timespans (also introducing memory into the system) and were integrating additional sensory information. The experiments showed that the robots became more adapted to their environment exploiting visual cues for orienting and navigation etc., (Verschure et al., 2003). Many other approaches in emergent systems concentrate on perception, for example, the Neurally Organized Mobile Adaptive Device (NOMAD) which is based on Edelman (1993) Neural Darwinism approach and demonstrates pattern recognition in a mobile robot platform (Krichmar and Snook, 2002). Recently, this has gained broader support in the area of autonomous mental development (Weng et al., 2001) and has established the field of developmental robotics (Cangelosi and Schlesinger, 2015). A particular focus in such architectures concerning learning is currently not covered in reaCog. In general, as pointed out by Langley et al. (2009), these kinds of approaches have not yet demonstrated the broad functionality associated with cognitive architectures (and—as in addition mentioned by Duch et al. (2008)—many of such models are not realized and are often not detailed enough to be implemented as a cognitive system). ReaCog realizes such an emergent system but with focus on a complex behaving system that, in particular, aims at higher cognitive abilities currently not reached by such emergent systems. The third type concerns hybrid approaches which try to bring together the advantages of the other two paradigms, for example ACT-R (Adaptive Components of Thought-Rational, Anderson, 2003). The, in our view, most impressive and comprehensive model of such a cognitive system is presented by the CLARION system (review see Sun et al., 2005; Helie and Sun, 2010) being applied to creative problem solving. This system is detailed enough so that it can be implemented computationally. Applying the so called Explicit-Implicit Interaction (EII) theory and being implemented in the CLARION framework, this system can deal with a number of quantitatively and qualitatively known human data, by far more than can be simulated by our approach as reaCog, in contrast, does not deal with symbolic/verbal information. Apart from this aspect, the basic difference is that the EII/CLARION system comprises a hybrid system as it consists of two modules, the explicit knowledge module and the implicit knowledge module. Whereas, the latter contains knowledge that is not "consciously accessible" in principle, the explicit network contains knowledge that may be accessible. Information may be redundantly stored in both subsystems. Mutual coupling between both modules allows for mutual support when looking for a solution to a problem. In our approach, instead of using representational differences for implicit and explicit knowledge to cope with the different accessibility, we use only one type of representation, that, however, can be differently activated, either being in the reactive mode or in the "attended" mode. In our case, the localist information (motivational units) and the distributed information (procedural networks) are not separated into two modules, but form a common, decentralized structure. In this way, the reaCog system realizes the idea of recruitment as the same clusters are used in motor tasks and cognitive tasks. Whereas, we need an explicit attention system, as given in the spreading activation and winner-take-all layer, in the CLARION model decisions result from the recurrent network finding an attractor state.

Many models of cognition take, quite in contrast to our approach, as a starting point the anatomy of the human brain. A prominent example is the GNOSIS project (Taylor and Zwaan, 2009). It deals with comparatively fine-grained assumptions on functional properties of brain modules, relying on imaging studies as well as on specific neurophysiological data. While GNOSIS concentrates mainly on perceptual, in particular visual input, the motor aspect is somewhat underrepresented. GNOSIS shows the ability to find new solutions to a problem, including the introduction of intermediate goals. Although an attention system is applied, this is used for controlling perception, not for supporting the search, as is the case in reaCog. Related to this, the search procedure—termed non-linguistic reasoning—in

GNOSIS appears to be less open as the corresponding network is tailored to the actual problem to avoid a too large search space. In our approach, using the attention system, the complete memory can be used as substrate for finding a solution.4.2 Cognitive Robotic Approaches

The approaches introduced in the previous section are not embodied and it appears difficult to envision how they could be embodied (Duch et al., 2008). Following the basic idea of embodied cognition (Brooks, 1989; Barsalou, 2008; Barsalou et al., 2012) embodiment is assumed as being necessary for any cognitive system. Our approach toward a minimal cognitive system is based on this core assumption. Robotic approaches have been proposed as ideal tools for research on cognition as the focus cannot narrowed down to a singular cognitive phenomenon, but it is required to put a unified system into the full context of different control processes and in interaction with the environment (Pezzulo et al., 2012).

ReaCog as a system is clearly embodied. The procedures cannot by themselves instantiate the behavior, but require a body. The body is a constitutive part of the computational system, because the sensory feedback from the body is crucially required to activate the procedural memories in the appropriate way. The overall behavior emerges from the interaction between controller, body and environment. In the following, we will review relevant embodied robotic approaches.

Today, many robotic approaches deal with the task of learning behaviors. In particular, behaviors should be adaptive. This means, a learned behavior should be transferable to similar movements and applicable in a broader context. Deep learning approaches have proven quite successful in such tasks e.g., Lenz et al. (2015), but many require large datasets for learning. Only recently Levine et al. (2015) presented a powerful reinforcement learning approach in this area. In this approach, the robot uses trial-and-error during online learning to explore possible behaviors. This allows the robot to quickly learn control policies for manipulation skills and has shown to be effective for quite difficult manipulation tasks. When using deep learning methods it is generally difficult to access the learned model. In contrast to reaCog such internal models are therefore not well suited for recruitment in higher-level tasks and planning ahead. In particular, there is no explicit internal body model which could be recruited. Rather, only implicit models are learned and have to be completely acquired anew for every single behavior.

In the following, two exciting robotic examples tightly related to our approach will be addressed in more detail. The approach by Cully et al. (2015) aims at solving similar tasks as reaCog for a hexapod robot. It also applies as a general mechanism the idea of trial-and-error learning when the robot encounters a novel situation. In their case these new situations are walking up a slope or losing a leg. There are some differences compared to reaCog. Most notably, the testing of novel behaviors is done on the real robot. This is possible as the trial-and-error method is not applying discrete behaviors. Instead, central to the approach by Cully et al. (2015) is the idea of a behavioral parametrization which allows to characterize the currently experienced situation in a continuous, low dimensional space. A complete mapping toward optimal behaviors has been constructed in advance offline (Mouret and Clune, 2015). This pre-computed behaviors are exploited when a new situation or problem is encountered. As the behavioral space is continuous, the pre-computed behavior can be used to adapt for finding a new behavior. Further, there is no explicit body model that is shared between different behaviors. Instead, the memory approximates an incomplete body model, as it contains only a limited range of those movements which are geometrically possible. In contrast, reaCog, using its internal body model, allows to exploit all geometrically possible solutions and is not constraint to search in a continuous space, as illustrated by our example case, where a single leg is selected to perform completely out of context.

While there is only a small number of robotic approaches dealing with explicit internal simulation, most of these are using very simple robotic architectures with only a very small number of degrees of freedom [for example see Svensson et al. (2009) or Chersi et al. (2013)]. It should further be mentioned that predictive models are also used to anticipate the visual effects of the robot's movements (e.g., Hoffmann, 2007; Möller and Schenck, 2008). With respect to reaCog the most similar approach has been pursued by Bongard et al. (2006). These authors use a four-legged, eight DoFs robot which, through motor babbling—i.e., randomly selected motor commands—learns the relation between motor output and the sensory consequences. This information is used to distinguish between a limited number of given hypotheses concerning the possible structure of the body. Finding the best fitting solution, one body model is selected. After the body model has been learned, in a second step the robot learns to move. To this end, the body model was used to perform different simulated behaviors and was only used as a forward model. Based on a reward given by an external supervisor and an optimizing algorithm, the best controller (sequence of moving the eight joints) was then realized to run the robot. Continuous learning allows the robot to register changes in the body morphology and to update its body model correspondingly. As the most important difference, Bongard et al. (2006) distinguish between the reactive system and the internal predictive body model. The central idea of their approach is that both are learned in distinct phases one after another. In reaCog the body model is part of the reactive system and required for the control of behavior. This allows different controllers driving the same body part and using the same body model for different functions (e.g., using a limb as a leg or as a gripper, Schilling et al., 2013a, **Figure 10**). In addition, different from our approach, Bongard et al. (2006) do not use artificial neural networks (ANN) for the body model and for the controller, but an explicit representation because application of ANN would make it "difficult to assess the correctness of the model" (Bongard et al., 2006, p. 1119). ReaCog deals with a much more complex structure as it deals with 18 DoFs instead of the only eight DoFs used by Bongard et al. (2006) which makes an explicit representation even more problematic.

Different from their approach, we do not consider how the body model and the basic controllers will be learned, but take both as given (or "innate"). While the notion of innate body representations is controversial (de Vignemont, 2010), there is at least a general consensus about that there is some form

of innate body model (often referred to as the body schema) reflecting general structural and dynamic properties of the body (Carruthers, 2008) which is shaped and develops further during maturation. This aspect is captured by our body model that encodes general structural relations of the body in service for motor control, but may adapt to developmental changes. While currently only kinematic properties are applied, dynamic influences can be integrated in the model as has been shown in Schilling (2009).

A further important difference concerns the structure of the memory. Whereas, in Bongard's approach one monolithic controller is learned to deal with eight DoFs and producing one specific behavior, in reaCog the controller consists of modularized procedural memories. This memory architecture allows for selection between different states and therefore between different behaviors.

DISCUSSION AND CONCLUSION

A network termed reaCog has been proposed that is based on the reactive controller Walknet equipped with decentrally organized behavioral modules, or procedures, all connected to motivation units, and a body model. The motivation units form a network that represents a heterarchical architecture allowing for the realization of various internal states. These states result from parallel activation of elements as well as competitive selection between elements.

The body model can be used as an inverse model for controlling motor output, as a forward model for internal simulation of behavior, and it can be exploited to improve erroneous sensor data ("sensor fusion"). Whereas, the reactive part uses the ability of the body model to function as an inverse model, the cognitive expansion exploits the internal body model to be used as a forward model and thereby as a tool for internal simulation of behavior. Internal simulation is used for finding a new solution for a problem detected by a problem detector. To this end, a three-layered network has been introduced that selects a new, currently not used module of the procedural memory, the contribution of which will then be tested through internal simulation. If this simulation turns out to be successful, i.e., shows a solution for the current problem, the corresponding behavior will be executed in reality. Thus, motor planning is possible using an extremely small expansion, a network consisting of essentially six units plus three parallel layers of units connected in a simple way.

In reaCog, there is no explicit, separate planner as used in hybrid systems. Rather, the ability to plan ahead relies on exploiting the reactive basis by operating on it much like a parasite operates on its host, that is, by only controlling the functioning of the reactive part. In other words, the cognitive expansion does not represent a separate planner, but organizes the activity of the reactive part, which is, during planning, not connected with the motor output.

Thus, constitutive elements of reaCog are (1) embodiment, (2) a decentralized organization of various procedures arranged in a heterarchical architecture, (3) a holistic body model allowing for pattern completion that is used in reactive behavior and can be recruited for planning ahead, and (4) a small network, called cognitive expansion, that enables the otherwise reactive system to become—in the sense of McFarland and Bösser (1993)—a cognitive one. We are not aware of any other neuronal approach that covers all these properties. Although the network represents a simple architecture, in the following we will argue that properties often attributed to "higher" brains can be found in reaCog, too, thereby approaching the question concerning the basic neuronal requirements of such higher level phenomena.

Before entering into this discussion, one important aspect missing in the current version of reaCog has to be noted. There is long term memory represented by the procedures in the form of "species memory" (Fuster, 1995). There is short term memory as a new solution is stored until the corresponding behavior is executed. There is however no faculty yet to transform the content of this short term memory into a long term memory. The ability to store such a newly acquired procedure as a long term memory would of course be an advantageous property. To gain this capability, the sensory situation accompanying the occurrence of a "problem" should be able to directly elicit activation of the procedure found to solve the problem.

When discussing the properties of a network like reaCog, a crucial aspect concerns the notion of emergence. The rational behind searching for emergent properties is the assumption that many "higher level" properties are not based on dedicated neuronal systems specifically responsible for the respective properties. Rather, emergent properties arise from the cooperation of lower-level elements and are characterized as to require levels of description other than those used to describe the properties of the elements. In the remainder, such emergent properties will, where appropriate, be related to the requirements posed by Langley et al. (2009) supporting the idea that reaCog provides a minimal functional description for some of those requirements.

According to Langley et al. (2009), a cognitive system should show the following properties: (1) Storing motor skills and covering the continuum from fully reactive, closed-loop behavior to (automatic) open-loop behavior; (2) Emergent properties resulting from the cooperation between different independent modules; (3) Long term memory and short term memory; (4) Attention to select sensory input; (5) Decisions on the lower level and "choice" on the higher level; (6) Predictions of possible actions; (7) Problem solving and planning of actions in the world; (8) Recognition and categorization of sensory input; (9) Remembering and episodic memory; (10) Application of symbols and reasoning; (11) To support reasoning, relationships between beliefs have to be realized; (12) Interaction and communication, including representation of verbal symbols; (13) Reflection and explanation (metareasoning); (14) Confronting the interactions between body and mind.

Requirements (1) and (2) are properties of Walknet. Above, we already argued that requirements (3), (6), and (7) can be found in reaCog, too. Below we will argue that also requirements (4) and (5) are fulfilled, but not aspects (8)–(14).

As the cognitive expansion of the reactive network allows the complete system—using psychological terms to describe

its function—to "focus" or "concentrate" or "attend" on a specific behavior, we have already earlier termed this expansion "attention system" supporting Langley et al.'s issue (4). Its ability to focus on specific memory elements may correspond to what sometimes has been termed "spot light" (Baars and Franklin, 2007) referring to the observation that the content of only one memory element becomes aware at a given moment in time. Recall, that selection of a specific procedure via the WTA network of the attention system does not mean that the other procedures are suppressed. The cognitive expansion network does not prohibit parallel activation of procedures. This requirement is in line with current developments in the area of cognitive systems research as pointed out by Duch et al. (2008). Inspired by the way how brains are organized, these authors propose, first, that cognitive systems in the future should incorporate a mechanism to focus attention, which is realized in reaCog through simple local competition as realized in the WTA structures. And second, that a neural network-like spreading activation mechanism is required in order to broaden search and follow associations, which is given in the spreading activation layer.

The fifth aspect of Langley et al. (2009) is concerned with action selection on lower levels and "choice" of behavior on a higher level. Action selection is indeed a crucial property of the network. On a lower level, within a given behavioral context—in our case walking—specific procedures compete via local WTA connections. For instance, a leg controller has to decide when to perform swing or stance movements. On an intermediate level, a decision can be made between, for example, forward walking and backward walking. On an even higher level, reaCog, exploiting the cognitive expansion, can select one specific behavioral element to be activated in addition to the currently active units. Therefore, Langley et al.'s requirement (5) is covered, too.

Thus, reaCog shows action selection not only on the reactive level, but also on the cognitive level, whereby the decisions based on internal simulation (or imagined action, "mental" action, or "probehandeln" according to Freud (1911) are not determined strictly by the sensorily given situation. Even if an external observer had the ability to monitor the internal states of the agent controlled by reaCog, the behavior of the agent could not be predicted by this observer. This is the case because, due to the noise in the SAL network, there is a stochastic element contributing to the decision. On the other hand, the final decision is not purely random, because the proposals made by the attention system are tested for feasibility via the internal simulation and are to some extent guided by the somatotopic structure of the SAL network. The proposal is further tested by performing the behavior in reality. In this way, this process of finding a new solution may be viewed as to be based on a Darwinian procedure, starting with an, in part, stochastic "mutation," followed by a selection testing the proposal for "fitness."

It has been stated that in a cognitive system, in order to address memory elements out of context ("global availability," e.g., Dehaene and Changeux, 2011), these elements have to be represented independently, i.e., not embedded in reactive structures. In reaCog, procedures are not represented independently, but are always represented within their context. Nonetheless, the functioning of the cognitive expansion allows to integrate them in another context. In other words, in reaCog, the procedures are globally available. Therefore, global availability may not require procedures being stored independent of any context (or "amodally"). Thus, reaCog represents a concrete architecture showing how global availability might be established in a neural system without requiring independent representation.

There is a group of related terms addressing a fundamental principle of brains. These are the "massive redeployment hypothesis" (Anderson, 2010), the "neural recycling theory" (Dehaene, 2005), the "shared circuits model" of Hurley (2008) and Gallese's "neural exploitation" (Gallese and Lakoff, 2005), summarized by Anderson (2010) by the term "neural reuse." Neural reuse means that a system is able to exploit existing components that do something useful to support a new task, either in the evolutionary time frame or by learning (Anderson, 2010, p. 250). In other words, neural reuse states that existing elements are used for other purposes. ReaCog models neural reuse of two kinds as listed by Anderson. One type, already applied in the current version of reaCog, corresponds to the use of the same procedural elements for both motor control and planning. Here reuse corresponds to the case of having been installed in evolutionary time scales. The second type addressable in reaCog concerns the reuse of procedural elements as a result of learning the integration of a given procedure in a new context as described above, which is, in reaCog, currently only given in the form of short term memory. But the ability to transfer this information into long term memory is a major focus for future work.

Although the structure of reaCog is far away from any morphological similarity to mammalian brains, functionally reaCog shows some similarity and may, therefore, in spite of its simplicity, be considered as a scaffold helpful for the understanding of properties of higher brains. To this end, taking a more abstract view, one might ask whether higher level properties characterized by using psychological terminology might be attributed to reaCog. As noted earlier, in reaCog emergent properties can already be observed at lower levels (e.g., production of different gaits) but they can also be found at higher levels, thereby supporting Langley et al.'s second requirement. Above, we had already used one such higher level term, attention. It has been argued (Cruse and Schilling, 2013, 2015) that further emergent properties as are intentions and emotions might be attributed to a system as reaCog, too, at least on the functional level. When adding some further procedures, reaCog might even be equipped with basic aspects required for Access Consciousness as well as Reflexive Consciousness (Cruse and Schilling, 2013, 2015).

Taken together, Langley et al.'s (Langley et al., 2009) requirements for cognitive systems (1)–(7) are well covered by reaCog. To conclude, we will briefly address the remaining issues (some of which have already been mentioned above): The capability to categorize sensory input, [Langley et al.'s issue (8)] is not given in reaCog as we focus mainly on the motor aspects. As mentioned, learning will be the focus of future work and

will address episodic memory (9). Other aspects would require further extension: Langley et al.'s issues (10)–(12) refer to the ability to use (verbal) symbols, a property not given in reaCog. However, a way has been sketched how this might be possible (for a first step toward this property see (Schilling and Spranger, 2010; Cruse and Schilling, 2013; Schilling and Narayanan, 2013), based on ideas of Steels and Belpaeme (2005) and Narayanan (1999).

Langley et al.'s issue (13) concerns Reflection and explanation. This property is not realized in reaCog and may also depend on the ability to apply symbolic knowledge. Issue (14), the property of cognitive systems to "confront the interactions between body and mind" addresses the property of having phenomenal experience, and is not found in reaCog, too (for a discussion of this matter see Cruse and Schilling, 2013, 2015). In summary, a number of emergent properties can be observed in reaCog, including Langley et al.'s issues (1)–(7). Issues (8)–(13) require further expansions.

In this article we focus on the situation that there is a problem which requires immediate solution and as a consequence, immediate internal simulation. As in our situation the body model is needed for simulation, the former cannot be used for controlling other behaviors at the same time. In other words, the body position has to be kept constant during internal simulation. In the following we briefly mention three cases which do not comply with this situation. In the first case internal simulation is not required. In this simple case the network is equipped with reactive procedures that allow for unspecific, general responses in case a problem detector is activated. An ubiquitous example is given by freezing behavior without active search for a solution, hoping that the problem will disappear on its own. Another example might be a procedure that allows emitting a general alarm signal that activates conspecifics. As a second case, one might think of situations that allow to postpone the search for a solution. In this case the normal behavior can be continued until a situation is given that allows to use the internal model without getting into conflict with current behavior. This case would at least require a short term memory to store the problem situation so that this could later be reactivated to start internal simulation, an expansion not yet implemented in the current version of reaCog. As a third, more complex case there might be a network that is able to control any behavior and at the same time, run an internal simulation. Whereas, for the second case a comparatively simple expansion of reaCog would suffice, the latter case appears to be much more demanding. It might, for example, require a second internal model plus the corresponding circuit to control both models independently.

The term "cognition" as used here, is based on the simple definition proposed by McFarland and Bösser (1993), i.e., the faculty of being able to plan ahead. This faculty is achieved here by using a reactive system plus introduction of a "cognitive expansion." As discussed above, such a system appears to be suited to form a basis on which further emergent properties may be realized, properties that are often listed as being required for a system termed cognitive as are Langley et al's requirements (8)–(14), for example. If this view is correct, these properties need not necessarily be explicitly included in such a definition, but appear to result from a system based on reactive structures plus the critical capability of planning ahead, underlining the power of McFarland and Bösser (1993) clear-cut definition of cognition.

AUTHOR CONTRIBUTIONS

Conceptualization, methodology and writing—MS and HC. Software, investigation and simulation—MS.

ACKNOWLEDGMENTS

This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology "CITEC" (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG), by the EC project EMICAB, the Wissenschaftskolleg zu Berlin, the ICSI, and the DAAD (PostDoc grant to MS).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00003/full#supplementary-material

Supplemental Data 1 | Showing example runs from additional simulations.

Supplementary Video 1 | AwkwardPosture_InternalSimulation.mp4. Video showing the example case of testing different behaviors out of context in internal simulation and afterwards applying a successful solution to the robot.

Supplementary Video 2 | AwkwardPosture_InternalSimulation_LegPositions.mp4. Video showing the example case of testing different behaviors out of context in internal simulation and afterwards applying a successful solution to the robot. Showing the leg positions over time.

REFERENCES

Anderson, M. L. (2003). Embodied cognition: a field guide. *Artif. Intell.* 149, 91–130. doi: 10.1016/S0004-3702(03)00054-7

Anderson, M. L. (2010). Neural reuse: a fundamental organizational principle of the brain. *Behav. Brain Sci.* 33, 254–313. doi: 10.1017/S0140525X10000853

Anderson, M. L, Oates, T., Chong, W., and Perlis, D. (2006). The metacognitive loop I: enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. *J. Exp. Theor. Artif. Intell.* 18, 387–411. doi: 10.1080/09528130600926066

Arbib, M. A. (1998). "Schema theory," in *The Handbook of Brain Theory and Neural Networks*, ed M. Arbib (Cambridge, MA: MIT Press), 830–834.

Baars, B. J., and Franklin, S. (2007). An architectural model of conscious and unconscious brain functions: global workspace theory and IDA. *Neural Netw.* 20, 955–961. doi: 10.1016/j.neunet.2007.09.013

Barsalou, L. W. (2008). Grounded cognition. *Annu. Rev. Psychol.* 59, 617–645. doi: 10.1146/annurev.psych.59.103006.093639

Barsalou, L. W, Spivey, M., McRae, K., and Joanisse, M. (2012). *The Human Conceptual System. The Cambridge Handbook of Psycholinguistics.* New York, NY: Cambridge University Press.

Bässler, U. (1983). *Neural Basis of Elementary Behavior in Stick Insects*. Berlin; Heidelberg; New York, NY: Springer.

Bässler, U., and Büschges, A. (1998). Pattern generation for stick insect walking movements—multisensory control of a locomotor program. *Brain Res. Rev.* 27, 65–88. doi: 10.1016/S0165-0173(98)00006-X

Bläsing, B. (2006). Crossing large gaps: a simulation study of stick insect behavior. *Adapt. Behav.* 14, 265–285. doi: 10.1177/105971230601400307

Bläsing, B., and Cruse, H. (2004). Stick insect locomotion in a complex environment: climbing over large gaps. *J. Exp. Biol.* 207, 1273–1286. doi: 10.1242/jeb.00888

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science* 314, 1118–1121. doi: 10.1126/science.1133687

Brooks, R. A. (1989). A robot that walks; emergent behaviors from a carefully evolved network. *Neural Comput.* 1, 253–262. doi: 10.1162/neco.1989.1.2.253

Brooks, R. A. (1991a). "Intelligence without reason," in *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91),* eds J. Myopoulos and R. Reiter (Sydney, NSW; San Mateo, CA: Morgan Kaufmann publishers Inc.), 569–595.

Brooks, R. A. (1991b). Intelligence without representation. *Artif. Intell. J.* 47, 139–159.

Buccino, G., Riggio, L., Melli, G., Binkofski, F., Gallese, V., and Rizzolatti, G. (2005). Listening to action-related sentences modulates the activity of the motor system: a combined TMS and behavioral study. *Brain Res. Cogn. Brain Res.* 24, 355–363. doi: 10.1016/j.cogbrainres.2005.02.020

Cangelosi, A., and Schlesinger, M. (2015). *Developmental Robotics: From Babies to Robots*. Cambridge, MA: MIT Press/Bradford Books.

Carruthers, G. (2008). Types of body representation and the sense of embodiment. *Conscious. Cogn.* 17, 1302–1316. doi: 10.1016/j.concog.2008.02.001

Chersi, F., Donnarumma, F., and Pezzulo, G. (2013). Mental imagery in the navigation domain: A computational model of sensory-motor simulation mechanisms. *Adapt. Behav.* 21, 251–262. doi: 10.1177/1059712313488789

Cross, E. S, Antonia, F. F., and Grafton, S. T. (2006). Building a motor simulation *de novo*: observation of dance by dancers. *Neuroimage* 31, 1257–1267. doi: 10.1016/j.neuroimage.2006.01.033

Cruse, H. (1990). What mechanisms coordinate leg movement in walking arthropods? *Trends Neurosci.* 13, 15–21. doi: 10.1016/0166-2236(90)90057-H

Cruse, H. (2003). The evolution of cognition: a hypothesis. *Cogn. Sci.* 27, 135–155. doi: 10.1016/S0364-0213(02)00110-6

Cruse, H., Dürr, V., Schilling, M., and Schmitz, J. (2009). "Principles of insect locomotionm" in *Cognitive Systems Monographs*, eds P. Arena and L. Patanè (Berlin: Springer), 43–96.

Cruse, H., and Schilling, M. (2010). "Getting cognitive," in *The Neurocognition of Dance,* eds B. Bläsing, M. Puttke, and T. Schack (London: Psychology Press), 53–74.

Cruse, H., and Schilling, M. (2013). How and to what end may consciousness contribute to action? Attributing properties of consciousness to an embodied, minimally cognitive artificial neural network. *Front Psychol* 4:324. doi: 10.3389/fpsyg.2013.00324

Cruse, H., and Schilling, M. (2015). "Mental states as emergent properties. From walking to consciousness," in *Open Mind*, eds T. Metzinger and J. Windt (Frankfurt am Main: MIND Group). doi: 10.15502/9783958570436

Cully, C. A. J., Tarapore, D., and Mouret, J-B. (2015). Robots that can adapt like animals. *Nature* 521, 503–507. doi: 10.1038/nature14422

Dehaene, S. (2005). "Evolution of human cortical circuits for reading and arithmetic: the neuronal recycling hypothesis," in *From Monkey Brain to Human Brain*, eds S. Dehaene, J. R. Duhamel, M. Hauser, and G. Rizzolatti (Cambridge, MA: MIT Press), 133–157.

Dehaene, S., and Changeux, J. P. (2011). Experimental and theoretical approaches to conscious processing. *Neuron* 70, 200–227. doi: 10.1016/j.neuron.2011.03.018

de Vignemont, F. (2010). Body schema and body image — pros and cons. *Neuropsychologia* 48, 669–680. doi: 10.1016/j.neuropsychologia.2009.09.022

Duch, W., Oentaryo, R. J., and Pasquier, M. (2008). "Cognitive architectures: where do we go from here?," in *Frontiers in Artificial Intelligence and Applications,* Vol. 171, eds P. Wang, B. Goertzel, and S. Franklin (IOS Press), 122–136.

Dürr, V. (2005). Context-dependent changes in strength and efficacy of leg coordination mechanisms. *J. Exp. Biol.* 208, 2253–2267. doi: 10.1242/jeb.01638

Dürr, V., and Ebeling, W. (2005). The behavioural transition from straight to curve walking: Kinetics of leg movement parameters and the initiation of turning. *J. Exp. Biol.* 208, 2237–2252. doi: 10.1242/jeb.01637

Dürr, V., and Krause, A. (2001). "The stick insect antenna as a biological paragon for an actively moved tactile probe for obstacle detection," in *Climbing and Walking Robots – From Biology to Industrial Applications. Proceedings 4th International Conference. Climbing and Walking Robots* (Karlsruhe: CLAWAR 2001).

Dürr, V., Schmitz, J., and Cruse, H. (2004). Behaviour-based modelling of hexapod locomotion: linking biology and technical application. *Arthropod Struct. Dev.* 33, 237–250. doi: 10.1016/j.asd.2004.05.004

Edelman, G. M. (1993). Neural Darwinism: selection and reentrant signaling in higher brain function. *Neuron* 10, 115–125. doi: 10.1016/0896-6273(93)90304-a

Engel, A. K, Maye, A., Kurthen, M., and König, P. (2013). Where's the action? The pragmatic turn in cognitive science. *Trends Cogn. Sci.* 17, 202–209. doi: 10.1016/j.tics.2013.03.006

Feldman, J., and Narayanan, S. (2004). Embodied meaning in a neural theory of language. *Brain Lang.* 89, 385–392. doi: 10.1016/S0093-934X(03)00355-9

Freud, S. (1911)."Formulierung über die zwei prinzipien des psychischen geschehens," in *Gesammelte Werke, Bd. VIII*, 230–238.

Fuster, J. M. (1995). *Memory in the Cerebral Cortex*. Cambridge, MA: MIT Press.

Gallese, V., and Lakoff, G. (2005). The brain's concepts: the role of the sensory-motor system in conceptual knowledge. *Cogn. Neuropsychol.* 22, 455–479. doi: 10.1080/02643290442000310

Glenberg, A. M., and Gallese, V. (2012). Action-based language: A theory of language acquisition, comprehension, and production. *Cortex* 48, 905–922. doi: 10.1016/j.cortex.2011.04.010

Hawking, S. W. (2001). *The Universe in a Nutshell*. New York, NY; Toronto, ON: Bantam Books.

Helie, S., and Sun, R. (2010). Incubation, insight, and creative problem solving: a unified theory and a connectionist model. *Psychol. Rev.* 117, 994–1024. doi: 10.1037/a0019532

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends Cogn. Sci. (Regul. Ed).* 6, 242–247. doi: 10.1016/S1364-6613(02)01913-7

Hoffmann, H. (2007). Perception through visuomotor anticipation in a mobile robot. *Neural Netw.* 20, 22–33. doi: 10.1016/j.neunet.2006.07.003

Hurley, S. (2008). The shared circuits model (SCM): how control, mirroring, and simulation can enable imitation, deliberation, and mindreading. *Behav. Brain Sci.* 31, 1–22. discussion: 22–58. doi: 10.1017/S0140525X07003123

Ijspeert, A. J. (2014). Biorobotics: using robots to emulate and investigate agile locomotion. *Science* 346, 196–203. doi: 10.1126/science.1254486

Jeannerod, M. (2001). Neural simulation of action: a unifying mechanism for motor cognition. *Neuroimage* 14 (1 Pt 2), S103–S109. doi: 10.1006/nimg.2001.0832

Jeannerod, M. (2006). *Motor Cognition — What Action Tells the Self*. Oxford: University Press.

Jeannerod, M., and Decety, J. (1995). Mental motor imagery: a window into the representational stages of action. *Curr. Opin. Neurobiol.* 5, 727–732. doi: 10.1016/0959-4388(95)80099-9

Khlentzos, D., and Schalley, A. (2007). "Mental categories in natural languages," in *Mental States: Language and Cognitive Structures*, Vol. 2, ed A. C. S. Drew Khlentzos (Philadelphia, PA: John Benjamins Pub. Co.), 1–10.

Kindermann, T. (2002). Behavior and adaptability of a six-legged walking system with highly distributed control. *Adapt. Behav.* 9, 16–41. doi: 10.1177/105971230200900103

Krichmar, J. L., and Snook, J. A. (2002). "A neural approach to adaptive behavior and multi-sensor action selection in a mobile device," in *Robotics and Automation, Proceedings. ICRA'02. IEEE International Conference on* (Washington, DC: IEEE Conference on Robotics and Automation), 3864–3869.

Laird, J. E. (2008). Extending the soar cognitive architecture. *Front. Artif. Intell. Appl.* 171, 224–235.

Langley, P., Laird, J. E., and Rogers, S. (2009). Cognitive architectures: research issues and challenges. *Cogn. Syst. Res.* 10, 141–160. doi: 10.1016/j.cogsys.2006.07.004

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *Int. J. Rob. Res.* 34, 705–724. doi: 10.1177/0278364914549607

Levine, S., Wagener, N., and Abbeel, P. (2015). Learning contact-rich manipulation skills with guided policy search. ArXiv Preprint ArXiv:1501.05611.

Limongelli, L. Boysen, S, T., and Visalberghi, E. (1995). Comprehension of cause-effect relations in a tool-using task by chimpanzees (pan troglodytes). *J. Comp. Psychol.* 109:18. doi: 10.1037/0735-7036.109.1.18

Loula, P F. S., Harber, K., and Shiffrar, M. (2005). Recognizing people from their movement. *J. Exp.Psychol.* 31, 210–220. doi: 10.1037/0096-1523.31.1.210

Maes, P. (1990). "A bottom-up mechanism for behavior selection in an artificial creature," in *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats* (Cambridge, MA: MIT Press).

Makin, T. R, Holmes, N. P., and Ehrsson, H. H. (2008). On the other hand: Dummy hands and peripersonal space. *Behav. Brain Res.* 191, 1–10. doi: 10.1016/j.bbr.2008.02.041

Marques, H. G., and Holland, O. (2009). Architectures for functional imagination. *Neurocomputing* 72, 743–759. doi: 10.1016/j.neucom.2008.06.016

Maturana, H.R., and Varela, F. J. (1981). *Autopoiesis and Cognition: The Realization of the Living*. Dordrecht: Reidel.

McFarland, D., and Bösser, T. (1993). *Intelligent Behavior in Animals and Robots*. Cambridge, MA: MIT Press.

Menzel, R., Brembs, B., and Giurfa, M. (2007). "Cognition in invertebrates," in *Evolution of Nervous Systems in Invertebrates,* ed J. H. Kaas (Oxford: Academic Press), 403–442.

Metzinger, T. (2006). Different conceptions of embodiment. *Psyche* 12.

Metzinger, T. (2014). "First-order embodiment, second-order embodiment, third-order embodiment," in *The Routledge Handbook of Embodied Cognition*, ed L. A. Shapiro (London: Routledge), 272–286.

Minsky, M. (1986). *Society of Mind*. New York, NY: Simon and Schuster.

Möller, R., and Schenck, W. (2008). Bootstrapping cognition from behavior — a computerized thought experiment. *Cogn. Sci.* 32, 504–542. doi: 10.1080/03640210802035241

Morasso, C. P. M., Mohan, V., Rea, F., and Zenzeri, J. (2015). Revisiting the body-schema concept in the context of whole-body postural-focal dynamics. *Front. Hum. Neurosci.* 9:83. doi: 10.3389/fnhum.2015.00083

Mouret, J-B., and Clune, J. (2015). Illuminating search spaces by mapping elites. ArXiv Preprint ArXiv:1504.04909.

Mussa-Ivaldi, F A, Morasso, P., and Zaccaria, R. (1988). Kinematic networks distributed model for representing and regularizing motor redundancy. *Biol. Cybern.* 60, 1–16.

Narayanan, S. (1999). "Reasoning about actions in narrative understandingm" in *Proceedings International Joint Conference on Artifical Intelligence, Reasoning about Actions in Narrative Understanding* (Stockholm: IJCAI'1999. Morgan Kaufmann Press).

Newell, A. (1994). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Paskarbeit, J., Schilling, M., Schmitz, J., and Schneider, A. (2015). "Obstacle crossing of a real, compliant robot based on local evasion movements and averaging of stance heights using singular value decomposition," in *IEEE International Conference on Robotics and Automation (ICRA)* (Seattle), 3140–3145. doi: 10.1109/ICRA.2015.7139631

Pezzulo, G. L., Barsalou, W., Cangelosi, A., Fischer, M. H., McRae, K., and Spivey, M. J. (2012). Computational grounded cognition: a new alliance between grounded cognition and computational modeling. *Front. Psychol.* 3:612. doi: 10.3389/fpsyg.2012.00612/abstract

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Pulvermüller, F. (2005). Brain mechanisms linking language and action. *Nat. Rev. Neurosci.* 6, 576–582. doi: 10.1038/nrn1706

Pulvermüller, F., and Garagnani, M. (2014). From sensorimotor learning to memory cells in prefrontal and temporal association cortex: a neurocomputational study of disembodiment. *Cortex* 57, 1–21. doi: 10.1016/j.cortex.2014.02.015

Redish, A. D. (2016). Vicarious trial and error. *Nat. Rev. Neurosci.* 17, 147–159. doi: 10.1038/nrn.2015.30

Rizzolatti, G. (2005). The mirror neuron system and its function in humans. *Anatomy Embryol.* 210, 419–421. doi: 10.1007/s00429-005-0039-z

Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cogn. Brain Res.* 3, 131–141. doi: 10.1016/0926-6410(95)00038-0

Schilling, H. M. T., Schmitz, J., and Cruse, H. (2013b). Walknet, a bio-inspired controller for hexapod walking. *Biol. Cybern.* 107, 397–419. doi: 10.1007/s00422-013-0563-5

Schilling, M. (2009). "Dynamic equations in MMC networks: construction of a dynamic body model," in *Proceedings of the 12th International Conference on Climbing and Walking Robots (CLAWAR)* (Istanbul), 1047–1054.

Schilling, M. (2011). Universally manipulable body models — dual quaternion representations in layered and dynamic MMCS. *Auton. Robots* 30, 399–425. doi: 10.1007/s10514-011-9226-3

Schilling, M., and Cruse, H. (2007). "Hierarchical MMC networks as a manipulable body model," in *Proceedings of the International Joint Conference on Neural Networks* (*IJCNN 2007*) (Orlando, FL).

Schilling, M., and Cruse, H. (2008). "The evolution of cognition – from first order to second order embodiment," in *Modeling Communication with Robots and Virtual Humans,* eds I. Wachsmuth and G. Knoblich (Berlin: Springer), 77–108.

Schilling, M., and Cruse, H. (2012). What's next: Recruitment of a grounded predictive body model for planning a robot's actions. *Front. Psychol.* 3:383. doi: 10.3389/fpsyg.2012.00383

Schilling, M., Cruse, H., and Arena, P. (2007). Hexapod walking: An expansion to walknet dealing with leg amputations and force oscillations. *Biol. Cybern.* 96, 323–340. doi: 10.1007/s00422-006-0117-1

Schilling, M., and Narayanan, S. (2013). "Communicating with executable action representations," in *AAAI Spring Symposium: Designing Intelligent Robots* (Stanford, CA).

Schilling, M., Paskarbeit, J., Hoinville, T., Hüffmeier, A., Schneider, A., Schmitz, J., et al. (2013a). A hexapod walker using a heterarchical architecture for action selection. *Front. Comput. Neurosci.* 7:126. doi: 10.3389/fncom.2013.00126

Schilling, M., Paskarbeit, J., Schmitz, J., Schneider, A., and Cruse, H. (2012). "Grounding an internal body model of a hexapod walker — control of curve walking in a biological inspired robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012* (Vilamoura).

Schilling, M., and Spranger, M. (2010). "Embodied posture verbs: emergence of a vocabulary for describing postures," in *Extended Abstract at Conference on Conceptual Structure, Discourse and Language (CSDL)* & *Embodied and Situated Language Processing (ESLP) 2010* (San Diego, CA).

Schmitz, S. J. A., Schilling, M., and Cruse, H. (2008). No need for a body model: positive velocity feedback for the control of an 18-DOF robot walker. *Appl. Bionics Biomech. Special Issue Biol. Inspired Robots* 5, 135–147. doi: 10.1080/11762320802221074

Schneider, A., Paskarbeit, J., Schäffersmann, M., and Schmitz, J. (2011). "Biomechatronics for embodied intelligence of an insectoid robot," in *Proceedings of the 4th International Conference on Intelligent Robotics and Applications (ICIRA 2011)* (Aachen), 1–11.

Schumm, M., and Cruse, H. (2006). Control of swing movement: Influences of differently shaped substrate. *J. Comparat. Physiol.* 192, 1147–1164. doi: 10.1007/s00359-006-0147-0

Schütz, C., and Dürr, V. (2011). Active tactile exploration for adaptive locomotion in the stick insect. *Philos. Trans. R. Soc. Lond,. B Biol. Sci.* 366, 2996–3005. doi: 10.1098/rstb.2011.0126

Steels, L. (2003). Intelligence with representation. *Philos. Trans.* 361, 2381–2395. doi: 10.1098/rsta.2003.1257

Steels, L., and Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behav. Brain Sci.* 28, 469–489. doi: 10.1017/S0140525X05000087

Sun, R., Slusarz, P., and Terry, C. (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychol. Rev.* 112, 159–192. doi: 10.1037/0033-295X.112.1.159

Svensson, H., Morse, A. F., and Ziemke, T. (2009). "Representation as internal simulation: a minimalistic robotic model," in *Proceedings of the Thirty-First Annual Conference of the Cognitive Science Society*, ed N. Taatgen and H. van Rijn (Austin TX: Cognitive Science Society Austin), 2890–2895.

Taylor, L. J., and Zwaan, R. A. (2009). Action in cognition: the case of language. *Lang. Cogn.* 1, 45–58. doi: 10.1515/LANGCOG.2009.003

van Duijn, M., Keijzer, F., and Franken, D. (2006). Principles of minimal cognition: casting cognition as sensorimotor coordination. *Adapt. Behav. Anim. Animats Softw. Agents Robots Adapt. Syst.* 14, 157–170. doi: 10.1177/105971230601400207

Verschure, P. F. M. J., and Althaus, P. (2003). A real-world rational agent: Unifying old and new AI. *Cogn. Sci.* 27, 561–590. doi: 10.1207/s15516709cog2704_1

Verschure, P. F. M. J., Voegtlin, T., and Douglas, R. J. (2003). Environmentally mediated synergy between perception and behaviour in mobile robots. *Nature* 425, 620–624. doi: 10.1038/nature02024

Webb, B. (2004). Neural mechanisms for prediction: Do insects have forward models? *Trends Neurosci.* 27, 278–282. doi: 10.1016/j.tins.2004.03.004

Wendler, G. (1968). "Ein Analogmodell der Beinbewegungen eines laufenden Insekts," in *Kybernetik 1968, Beihefte Zu "elektronischen Anlagen*, Bd. 18 (München: Oldenbourg), 68–74.

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., et al. (2001). Autonomous mental development by robots and animals. *Science* 291, 599–600. doi: 10.1126/science.291.5504.599

Wilson, M. (2008). "How did we get from there to here? An evolutionary perspective on embodied cognition," in *Handbook of Cognitive Science: An Embodied Approach,* eds P. Calvo and A. Gomila (Amsterdam: Elsevier), 375–393.

Wintermute, S. (2012). Imagery in cognitive architecture: representation and control at multiple levels of abstraction. *Cogn. Syst. Res.* 19, 1–29. doi: 10.1016/j.cogsys.2012.02.001

Wolpert, D. M., and Flanagan, J. R. (2001). Motor prediction. *Curr. Biol.*11, 729–732. doi: 10.1016/S0960-9822(01)00432-8

Wolpert, D. M., and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Netw.*11, 1317–1329. doi: 10.1016/S0893-6080(98)00066-5

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Schilling and Cruse. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

HYPOTHESIS AND THEORY
published: 08 March 2017
doi: 10.3389/fnbot.2017.00012

Motor-Skill Learning in an Insect Inspired Neuro-Computational Control System

*Eleonora Arena[1], Paolo Arena[1,2], Roland Strauss[3] and Luca Patané[1]**

[1] Dipartimento di Ingegneria Elettrica, Elettronica, e Informatica, University of Catania, Catania, Italy, [2] National Institute of Biostructures and Biosystems, Rome, Italy, [3] Institut für Zoologie III (Neurobiologie), University of Mainz, Mainz, Germany

In nature, insects show impressive adaptation and learning capabilities. The proposed computational model takes inspiration from specific structures of the insect brain: after proposing key hypotheses on the direct involvement of the mushroom bodies (MBs) and on their neural organization, we developed a new architecture for motor learning to be applied in insect-like walking robots. The proposed model is a nonlinear control system based on spiking neurons. MBs are modeled as a nonlinear recurrent spiking neural network (SNN) with novel characteristics, able to memorize time evolutions of key parameters of the neural motor controller, so that existing motor primitives can be improved. The adopted control scheme enables the structure to efficiently cope with goal-oriented behavioral motor tasks. Here, a six-legged structure, showing a steady-state exponentially stable locomotion pattern, is exposed to the need of learning new motor skills: moving through the environment, the structure is able to modulate motor commands and implements an obstacle climbing procedure. Experimental results on a simulated hexapod robot are reported; they are obtained in a dynamic simulation environment and the robot mimicks the structures of *Drosophila melanogaster*.

Keywords: insect brain, insect mushroom bodies, spiking neural controllers, learning, goal-oriented behavior

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark Odense, Denmark

Reviewed by:
Sakyasingha Dasgupta, IBM Research-Tokyo, Japan
Björn Brembs, University of Regensburg, Germany

***Correspondence:**
Luca Patané
lpatane@dieei.unict.it

Received: *01 September 2016*
Accepted: *20 February 2017*
Published: *08 March 2017*

Citation:
Arena E, Arena P, Strauss R and Patané L (2017) Motor-Skill Learning in an Insect Inspired Neuro-Computational Control System. Front. Neurorobot. 11:12. doi: 10.3389/fnbot.2017.00012

1. INTRODUCTION

Recent results and experiments performed on insects shed light on their highly developed learning and proto-cognitive capabilities enabling them to adapt extremely well to their natural environment (Menzel and Giurfa, 1996; Liu et al., 1999; Tang and Guo, 2001; Chittka and Niven, 2009). Modeling insect brains is an increasingly important issue for the design of learning and control strategies to be applied on autonomously walking robots. Within the insect brain an important paired neuropil with higher control functions are the mushroom bodies (MBs), recently used to model different behavioral functions (Smith et al., 2008; Arena et al., 2013c). Studies on bees and flies identified the MBs as a relevant area for associative learning and memory in odor conditioning experiments (Menzel and Muller, 1996; Menzel, 2001; Scherer et al., 2003; Liu and Davis, 2006). MBs are also involved in behaviors depending on other sensory modalities, like vision (Liu et al., 1999; Menzel, 2001; Tang and Guo, 2001), other types of learning such as choice behaviors (Tang and Guo, 2001; Gronenberg and Lopez-Riquelme, 2004; Brembs, 2009) and, as recently introduced, also in the improvement of gap-climbing tasks (Pick and Strauss, 2005; Kienitz, 2010).

MBs receive olfactory input from the antennal lobes via projection neurons. The latter run in the medial antennal lobe tract, provide input to the MB calyces and continue on to the lateral

horn (LH). The mediolateral and the lateral antennal lobe tracts emerge from the antennal lobes as well, but bypass the calyces and project directly to the LH. The LH region controls inborn behavior, whereas the MBs are thought to be involved in learnt behavior. Analysing the interaction between the different neural structures we investigated the emergence of interesting neural activities responsible for specific behaviors in insects, including flies, like attention, expectation, delayed-match to sample tasks, and others (Arena et al., 2012a,b, 2013b).

Major dynamical aspects characterizing the locust olfactory system were already outlined in Mazor and Laurent (2005). Here a principal component analysis on the firing rate of a population of PNs revealed different attractors for different odors. These attractors show two transients and one fixed point, but transients are most significant for an efficient odor classification. This addressed for the first time the importance of transient dynamics to explain and understand neural coding and information processing in the MBs. Following these results, we hypothesized that the role of transient dynamics is relevant for the sensory information coding extending the results obtained in locust olfactory system to the fruit fly. This hypothesis well match with the organization properties of the MBs discussed in Nowotny et al. (2003, 2005). Their model is based on spiking neurons and synaptic plasticity, distributed through different layers. The model is able to show consistent recognition and classification of odors. In the study of Nowotny and colleagues, MBs are assumed to be multi-modal integration centers, combining olfactory and visual inputs. As in our current model, the capabilities are independent of the type and the source of information processed in the MBs.

Wessnitzer and co-authors investigated the interaction between MBs and antennal lobes (ALs) and proposed a computational model for non-elemental learning (Wessnitzer et al., 2012). Different levels of learning and reinforcement mechanisms were considered at the stage of the KCs to create a coincidence detector and non-elemental learning. Reward mechanisms are commonly considered for the creation of aversive and appetitive olfactory memories (Schwaerzel et al., 2003) and the role of dopamine is relevant in *Drosophila* (Waddell, 2013). We here extended this scheme to memorize specific parameters involved in the motor-skill learning process. On the basis of fruit fly brain structures and on hypotheses related to information processing and learning mechanisms MBs are a structure able to adapt and memorize relevant parameters involved in motor learning. This improves the fly's capabilities when it is trained in repeating a task like climbing over a chasm. Therefore, a simplified computational model of the MB neuropile was developed using a pool of spiking neurons representing the so-called Kenyon Cells (KCs).

The computational model proposed in this paper for motor learning takes the biological characteristics of the MBs into account and, on the basis of the previously introduced hypotheses, arrives at a neuro-computational structure similar to a Liquid State Machine (LSM) proposed by Maass et al. (2002). The information embedded in the dynamical neural lattice is transferred to the lower motor layers by extrinsic MB neurons that have been modeled as read-out maps.

One fundamental difference between the proposed model and the LSM is the presence of local connectivity among the neurons within the liquid layer. This element of our model deserves particular attention: in fact, the structure configures as a locally connected recurrent neural network which is fairly similar to the Cellular Neural/Nonlinear Network (CNN) structure (Manganaro et al., 1999), a paradigm already used for the generation of complex dynamics and for controlling artificial locomotion (Arena et al., 1999) and perception phenomena (Arena et al., 2009). The other important characteristics of the proposed model is related to the hardware implementation: in fact there are a number of analog/logic VLSI CNN-based chips available which implement digitally programmable analog computers characterized by high computational speed and analog, parallel computation capability, typically used for high frame rate visual microprocessors (Rodríguez-Vázquez et al., 2008). However, the suitable adaptation of the MB structure modeled in this paper as a CNN architecture via the suitable addition of trainable read-out maps would allow for the possibility to adopt a well-assessed reference hardware for the real time implementation of the proposed approach.

From the modeling perspective, the developed structure links two main ideas: high parallelism in brain processing and Neural Reuse (Anderson, 2010). According to the first-mentioned, sensory pathways run in parallel and concur to form abstract schemes of the environmental state, useful for motor actions or abstract decisions. The Neural Reuse approach, on the other hand, states that the same neural structure can be concurrently exploited for different tasks. The insect MBs were already addressed as centers where such characteristics could be found, and the control structure herewith introduced makes a step forward to derive an efficient computational model directly useful as a robot behavioral controller (Arena and Patané, 2014).

2. MOTOR-SKILL LEARNING IN INSECTS

Among the different forms of neural adaptation encountered in animals, motor-skill learning is a fundamental capability needed to survive in dynamically changing environments and also to cope with accidental impairments of animal's limbs.

Motor-skill learning can be defined as the process to acquire precise, coordinated movements needed to fulfill a task. Due to the importance of this capability,sensory-motor conditioning was one of the earliest types of associative learning found in cockroaches and locusts. It has been demonstrated in the ventral nerve cord of insects (Horridge, 1962) and is probably ubiquitous in moving animals (Byrne, 2008; Dayan and Cohen, 2011).

The motor-skill learning system incrementally improves the motor responses by monitoring the resulting performance: this process guides the adaptive changes. By exploiting the involved sensory motor loops, agents apply operant strategies during motor learning: when a movement is performed, sensory feedback is used to evaluate its accuracy (Brembs and Heisenberg, 2000; Broussard and Karrardjian, 2004).

In insects there are different examples of motor learning processes that adapt motor schemes to specific tasks. For

instance, honeybees can adapt the antennal movements to an obstacle after a prolonged presentation of this obstacle. Furthermore, the use of an outside rewarding mechanism dramatically speeds-up the learning process (Erber et al., 1997).

Other insect behaviors involving motor learning were reported by Mohl (1993); he investigated the relevance of proprioception during flight in locust. In an interesting paper on *Drosophila* motor-skill learning capabilities (Wolf et al., 1992),a series of conditions has been identified for proper motor-skill learning.

First, the fly has a desired target to reach; to fulfill this aim, a number of motor programs are activated in a random sequence. Efference copies of the motor programs are compared with references and if, for a given motor behavior, a meaningful correlation is found, this is applied. Other studies in this direction were performed on bumblebees (Chittka, 1998) and butterflies (Lewis, 1986).

Behavioral studies on insects confirmed that they are able to show sophisticated and adaptive motor-control strategies requiring the joint coordinated activity among the limbs. A particularly suitable experimental setup to inspect motor learning capabilities is the behavioral paradigm of gap crossing, first described by Blasing and Cruse (2004) and Blasing (2006) in relation to stick insects, by Pick and Strauss (2005) for *Drosophila* and in Goldschmidt et al. (2014) where the coackroach capabilities were considered. Flies with a body length of typically 2.5 mm (and with their wings clipped to disable flight) can cross gaps of up to 4.3 mm when fully exploiting their biomechanical limits. Direct observation and high-speed video analysis of the gap climbing procedure (see Pick and Strauss, 2005 and videos supplied) outlined that flies first visually estimate the gap width via parallax motion generated while approaching the gap. Then, if they consider the gap as being surmountable, they initiate the climbing procedure by combining and successively improving, through several attempts, a number of parameters for climbing. The hind legs are placed as close as possible near the proximal edge; the middle legs are attached to the proximal side wall of the gap and arrange the body horizontally; the front legs stretch out to attach to the opposite gap side. Then the middle legs are detached from the proximal side, swing over and are attached to the distal side surface of the gap. Finally, the hind legs are detached and the fly moves toward the other side. These experiments clearly show that several parameters are modified from their nominal values (for normal walking) and also combined together in several successive phases to maximize the climbing performance.

Later it was shown by Kienitz (2010) that flies improve their climbing abilities when they iteratively climb over gaps of the same width. The short-term improvements after 24 training trials within 1 min were seen in tests 20 min after training; they are missing in plasticity mutants. Rescue of plasticity in the MBs was sufficient to restore the motor-learning capacity. The finding that plasticity in MBs is a prerequisite for motor learning will be taken as our working hypothesis for the development of the proposed computational model. Experiments on gap crossing were also performed with stick insects (Blasing and Cruse, 2004; Blasing, 2006). In these works the authors outlined the role of single leg movements, searching reflexes, and coordination mechanisms as important to fulfill the task. A model of gap crossing behavior was implemented extending a previously developed bio-inspired network Walknet (Cruse et al., 1998), to reach simulated results comparable with the biological experiments. Here the gap crossing issue was considered as an extension of normal walking behavior with only limited modifications. In our work we reached a similar conclusion though starting from quite different models. In fact the CPG for normal walking is maintained whereas only a parameter adaptation was introduced to efficiently implement climbing.

The climbing capabilities of other insects like cockroaches were also considered to develop experiments on obstacle climbing and gap crossing using hexapod robots (Goldschmidt et al., 2014). The presence of an actuated joint in the robot body was exploited to improve the capabilities of the system to face with complex situations including gaps and obstacles (Goldschmidt et al., 2014; Dasgupta et al., 2015). In Pavone et al. (2006), the sprawled posture was a key element for solving the obstacle-climbing issue. In other cases the presence of spoked legs is a simple and efficient solution to improve power efficiency and walking capabilities in presence of obstacles (Moore et al., 2002). In some cases hybrid legged and wheeled robots try to take the advantages of both solutions (Arena et al., 2010).

Whereas, these approaches exploit the mechanical structure, other strategies instead consider primarily the adaptive capabilities of the control structure. For instance, for solving the antenna motor control problem, in Krause et al. (2009) an echo-state network was applied to generate the antenna movements in a simulated stick insect robot. The network was able to store specific trajectories and to reproduce them creating smooth transitions between the different solutions available, depending on the control input provided.

Distributed recurrent neural networks, working as reservoirs, were also used in Dasgupta et al. (2015) to create a forward model needed to estimate the ground contact event in each leg of a walking hexapod robot. The prediction error has been used to improve the robot walking capabilities for different types of terrains.

Our approach belongs to this last type of strategies, since it takes into account primarily the adaptive capabilities of a recurrent spiking network to solve a specific motor learning issue.

In fact, in our work, we considered only obstacle climbing scenarios because our *Drosophila*-like hexapod robot does not contain body joints (i.e., as exploited in Dasgupta et al., 2015 to facilitate also gap crossing); on the other hand it is unfeasible to include in the robot the adhesive capabilities of fly leg tips. Moreover, we assumed that the same computational structure as that one involving the MBs for gap climbing tasks is also involved in obstacle climbing. In the proposed example the external information used to characterize the scenario to be faced, was reduced to the obstacle height (e.g., acquired through a simple visual processing method) in order to learn the set of parameters that allow to fulfill the climbing task.

In particular, the MB intrinsic neurons are here modeled as a spiking network working as a reservoir, able to generate a rich, input-driven dynamics that is projected to other neural centers

using read-out maps that work as MB extrinsic neurons. An important added value obtained through the learning process consists in allowing a generalization of the learned data: in fact the network can generate the suitable output signals also for input patterns not included in the learning set by interpolating the memorized functions.

3. MODELING MOTOR-SKILL LEARNING

3.1. Known and Hypothesized Biological Functions

Tasks related to motor-skill learning need a specialization of motor functions to optimize performance.To fulfill this aim, a strategy for searching for the most suitable system parameters to be applied for modulating the leg trajectories is envisaged. The generation of pseudo-random parameters constrained only by the insect's body parameters is the initial step needed to improve the ongoing solution iteratively by trial and error. The searching process will produce a subset of successful attempts used to improve the overall system performance, storing the new set of suitable parameters evaluated on the basis of an internal reward function.

In insects, thoracic ganglia can be in charge for the generation of these trials (Horridge, 1962), but MBs should mediate the selection process consisting in a statistical shaping and in the final choice of the successful parameters that modulate the basic behaviors (Kienitz, 2010). Such learning processes are the basic ingredients for the implementation of a short-term working memory.

A neuro-control block scheme model is shown in **Figure 1** where the main elements involved in the proposed model of motor-skill learning are depicted. Plasticity and learning is ubiquitous in the model due to the complexity of the brain functions but for the aim of the proposed work we focused our attention only on specific parts. Therefore, we considered all the interconnections to be fixed except the synaptic output of the MBs, as will be discussed in details in Sections 3.2 and 3.3, in relation to the motor system (CPG). Plasticity and learning inside other blocks, including the visual sensory and pre-processing system, are not treated in this work.

The central complex (CX) is an excitatory center responsible for behavior activations on the basis of visual and mechanosensory inputs. The input signals are here processed through a series of substructures: the protocerebral bridge (PB), the fan-shaped body (FB), and the ellipsoid body (EB) (Hanesch

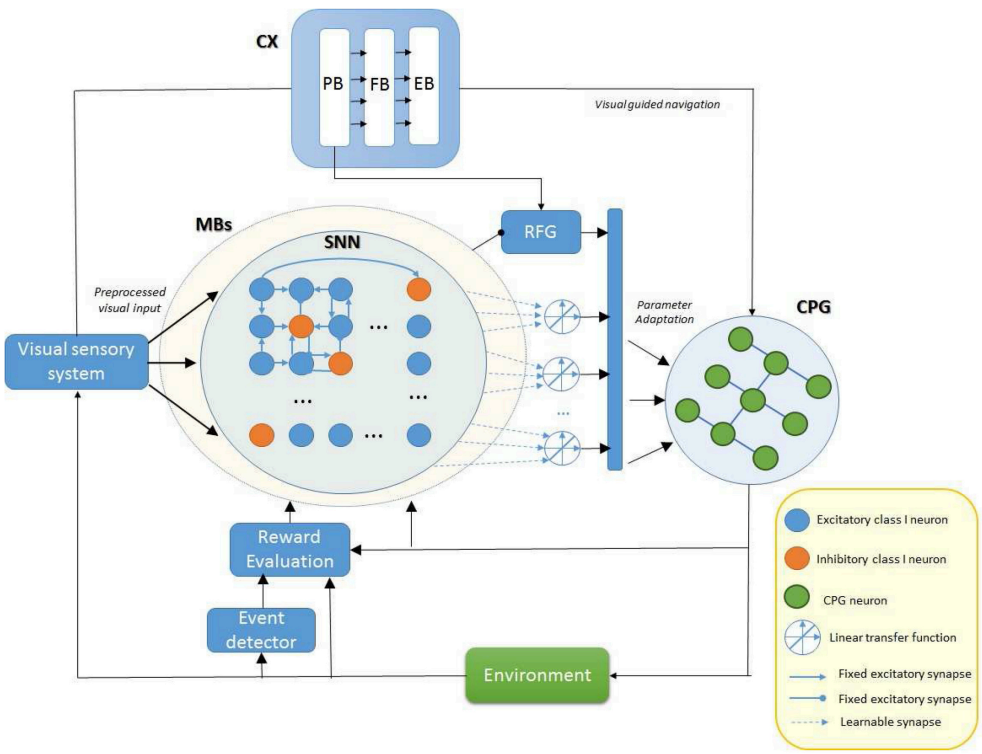


FIGURE 1 | Block diagram illustrating the role of different fly brain neuropils involved in motor control. Our model assumes that the parameter adaptation for the modulation of the ongoing behavior is performed by the MBs that receive reinforcement signals in form of dopaminargic/octopaminergic neuron activity and elaborate the learning process using a spiking neural network (SNN). The robot performs a visually guided navigation that acts on the Central Pattern Generator (CPG) structure to control locomotion. A reinforcement signal is generated for the MBs whereas a Random Function Generator (RFG) is used to include fluctuations in the set of control parameters for the CPG. The SNN function is used to memorize the temporal evolution of the modulated parameters if this improves the final motor behavior during learning.

et al., 1989; Strauss, 2002). Moreover, the PB is directly involved in motor control; it is also responsible for the stabilization of the walking direction (Triphan et al., 2010). On the contrary the MBs seem have an inhibitory effect and are fundamental for the adaptive termination of behaviors (Mronz and Strauss, 2001).

MBs present a large complexity at the level of the calyx, due to the different KC types and their interconnection. From the modeling point of view, KC types could be implemented through different non-linear functions (or dynamical systems). No information is available on the dynamics of these neurons and electrophysiological data are in short supply. On the other side powerful neurogenetic tools are available for the fruit fly which allow for precise manipulations of the nervous system in order to address links among specific neural substrates, their functions and specific behaviors they are responsible for.

Learning in *Drosophila melanogaster* has revealed multiple memory types and phases and recent investigations underlined that not all memory processes occur in MB neurons (Wu et al., 2007; Zhang et al., 2013).

Here we hypothesize that the CX and in particular the PB plays a role in motor learning: it performs adaptation of the motor system parameters shaping the motor behavior while the insect performs a task. The involvement is plausible as the PB seems to control step length for direction (Strauss, 2002; Triphan et al., 2010). This variability is attained in our model (see **Figure 1**) through a random function generator (RFG) which perturbs some relevant leg control parameters. This strategy generates perturbed leg trajectories. On the basis of the expected results, the on-going behavior is evaluated and eventually, MBs receive a reinforcement signal via extrinsic dopaminergic and octopaminergic neurons (Schwaerzel et al., 2003). Memory consolidation occurs overnight. After consolidation, the MBs are assumed to inhibit the perturbation provided by the RFG to allow the memory retrieval. The overall control system designed and implemented, as outlined in the following constitutes a clear example of a bio-inspired embodied, closed-loop neural controller.

3.2. MB Model for Motor Learning: Working Hypotheses

In order to design both a biologically plausible and a computationally feasible model of the MBs, the two following hypotheses were formulated:

- It is possible that different KCs accept different sensorial inputs at the level of the calyx. This assumption regarding different sensory modalities is made in parallel to olfactory learning (Lin et al., 2013).
- Signal processing within the network takes place at two different levels: within the KCs we have a spiking dynamics within locally, randomly connected neurons, whereas, at the level of extrinsic neurons, we have an external learning needed to learn different tasks. This is a working hypothesis, useful, from the one hand, to computationally simplify the model, and, from the other hand, to allow the concept of Neural Reuse to be directly implemented.

The following structural elements can be outlined:

1. Presence of randomly distributed internal connections.
2. Structural and functional correspondence between internal weights, mirroring the connections within the KC lattice, and the output weights, standing for connections among MBs and extrinsic neurons.
3. Possibility of using the same neural lattice concurrently in completely different tasks, following the Neural Reuse paradigm, by separately training different sets of read-out weights. The same network can therefore model a multimodal (and multifunctional) structure, as are the MBs (Arena et al., 2013c). We are hypothesizing that different sets of extrinsic neurons are devoted to map different tasks.

The proposed control scheme has been implemented in a computational model embedded on a robot simulated in a realistic dynamical environment. Referring to **Figure 1**, the robot navigates driven by vision: the heading commands are provided to the locomotion controller through external stimuli. An evaluation procedure assesses the suitability of the performed actions in solving the assigned task. An event detector triggers the evaluation process.

The reinforcement signal is passed to the MBs to evaluate the changes generated by the RFG and used to update a set of motor control parameters. Successful parameter updates, leading to significant improvements in the climbing behavior lead to memory formation. A SNN was considered as a plausible model to generate the long-term memory of the best parameters selected during the learning process and to guarantee interpolation capabilities important for the generation of feasible behaviors in situations similar to those ones encountered during the learning procedure. Finally a selector block determines if either a random trial can be performed or the information stored in the SNN can be used for the motor actions. Among the different kinds of neural networks used for solving problems like navigation (Tani, 1996), multi-link system control (Cruse, 2002) and classification, a lot of interest was devoted to Reservoir computing, which mainly includes two different approaches: Echo State Network (ESN) and LSM (Jaeger, 2001; Maass et al., 2002). In previous studies the idea to use non-spiking Recurrent Neural Networks to model the MBs memory and learning functions was explored (Arena et al., 2013a). The core of the newly proposed architecture, inspired by the biology of MBs', resembles the LSM architecture. It consists of a large collection of neurons, the so called liquid layer, receiving time-varying inputs from external sources as well as recurrent connections from other nodes in the liquid layer. The recurrent structure of the network turns the time-dependent input into spatio-temporal pattern in the neurons. These patterns are read out by linear discriminant units. In the last years LSM are becoming a reference point in replicating brain functionalities. However, there is no guaranteed way to analyze the role of each single neuron activity on the overall network dynamics: the control over the process is very weak. This apparent drawback is a consequence of the richness of the dynamics potentially generated within the liquid layer. The side advantage is that the high dimensional complexity can be concurrently exploited through several projections (the read-out maps) to obtain

non-linear mappings useful for performing different tasks at the same time. The proposed network differs from the structure reported in Arena et al. (2013a) in many aspects: it consists of a lattice of inhibitory and excitatory spiking (instead of non-spiking) neurons with a random connectivity, which is mainly local (instead of non-local). Moreover, the network configuration in Arena et al. (2013a), for solving the motor learning problem, required a much larger network configuration. This could be addressed to the much richer dynamics generated within the SNN (see Section 5.1). Inputs are here provided as currents that, through a sparse connection, reach the hidden lattice (i.e., the liquid layer). Multiple read-out maps, fully connected with the hidden lattice, can be learned considering the error between the network output, collected through an output neuron for each read-out map, and the target signal. The network details are illustrated in the next section.

3.3. Network Structure and Parameters

Following the biological hints, proposed hypotheses and suggestions from the classical LSM paradigm, the MBs' structure involved in motor learning has been modeled as a spiking-based network consisting of three layers: an input layer, a hidden recurrent neural lattice, and an output layer. The input layer behaves like a filter that randomly redirects input stimuli to a reduced number of neurons in the hidden-layer (KCs lattice). The connectivity percentage used in this work is 15% from the input layer to the KC layer.

The hidden layer is a SNN (i.e., the reservoir network), where each unit is an Izhikevich Class I spiking neuron (Izhikevich, 2000) organized in a square topology with toroidal boundary connections. The regular distribution of the neurons in a square-shaped lattice was selected because, for computational reasons, we considered the simplest structure where we can perform distance metrics. The following differential equations describe the model:

$$\begin{aligned} \dot{v} &= 0.04v^2 + 5v + 140 - u + I \\ \dot{u} &= 0.02(-0.1v - u) \end{aligned} \tag{1}$$

following spike-resetting condition:

$$\texttt{if} \;\; v \geq 0.03, \;\; \texttt{then} \begin{cases} v \leftarrow -0.055 \\ u \leftarrow u + 6 \end{cases} \tag{2}$$

Here v is the membrane potential, I is the synaptic current and u is a recovery variable. Izhikevich neural models are well-known in literature for offering a good compromise between biological plausibility and computational efficiency.

Neurons are connected through synapses: here the spike-rate from the pre-synaptic neuron is transformed into a current for the post-synaptic one. The response of the synapses to a pre-synaptic spike is as follows:

$$\varepsilon(t) = \begin{cases} Wt/\tau \exp(t/\tau), & \texttt{if} \;\; t > 0 \\ 0 \;\; , & \texttt{if} \;\; t < 0 \end{cases} \tag{3}$$

where τ is the time constant, t is the time passed since the last spike arrived at the pre-synapse and W is the synaptic efficiency. This last parameter can be modulated by learning. This synaptic model was also used to connect the lattice neurons to the output neurons.

The fraction of inhibitory neurons in the pool is about 10%. The connections within the lattice are represented by a synaptic weight with a random uniform distribution in the range (0.5–1.5), the input weights are equal to 1. The weights of the read-out map are subject to training. The generation of the inter-layer synaptic connectivity depends on a probabilistic function of the distance $d_{i,j}$ between the presynaptic (i) and postsynaptic (j) neurons:

$$P_{ij} = k * C_{i,j} \tag{4}$$

where

$C_{i,j}$	Inhibitory (j)	Excitatory (j)
Inhibitory (i)	0.1	0.4
Excitatory (i)	0.2	0.3

and

$$\begin{aligned} k = 2 \;\; \texttt{if} \;\; d_{i,j} \leq 1 \\ k = 1 \;\; \texttt{if} \;\; 1 < d_{i,j} \leq 2 \\ k = 0 \;\; \texttt{if} \;\; d_{i,j} > 2 \end{aligned} \tag{5}$$

The parameters $C_{i,j}$, reported in the previous table, have been chosen according to Maass et al. (2002). The distance $d_{i,j} = 1$ is calculated, either for horizontal or vertical adjacent neurons, considering the neurons as distributed on a regular grid possessing toroidal boundary conditions. From the relations above it derives that the connectivity realized within the lattice is local; this is an important element that facilitates a potential hardware implementation of the control system where the number of connections is drastically reduced and limited to each neuron neighborhood.

The time constant in Equation (3) was randomly chosen among the values $\tau = 5$, 10, 30, and 50 ms. This variability improves the dynamics potentially shown by the network as will be discussed in the following sections. The values of the synaptic time constant have been chosen to obtain significant dynamics in the simulation time window that is limited to 150 ms.

The output layer consists of a series of output neurons, modeled with a linear transfer function and fully connected with the hidden lattice. The output weights are randomly initialized in the interval $(-1, 1)$ and are subject to learning. The integration step used for the reported simulations was fixed to $dt = 1.5$ ms.

3.4. Learning Mechanism

The time evolution of the target signals that the network need to memorize is generated by shaping the lattice dynamics using read-out maps. An incremental learning rule based on the Least mean square algorithm is adopted to update the synaptic weights of each read-out map. The learning process, resembling the classical delta rule, depends on the lattice activity and on the error

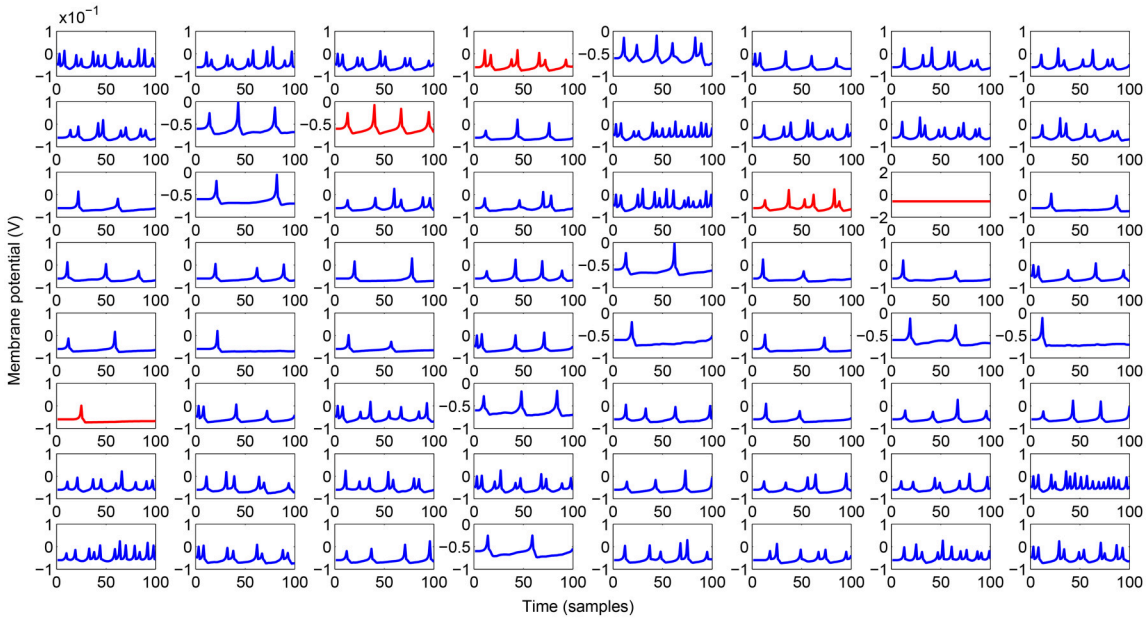


FIGURE 2 | Spiking activity of the lattice while generating the output signal during the testing phase. Inhibitory neurons are outlined in red.

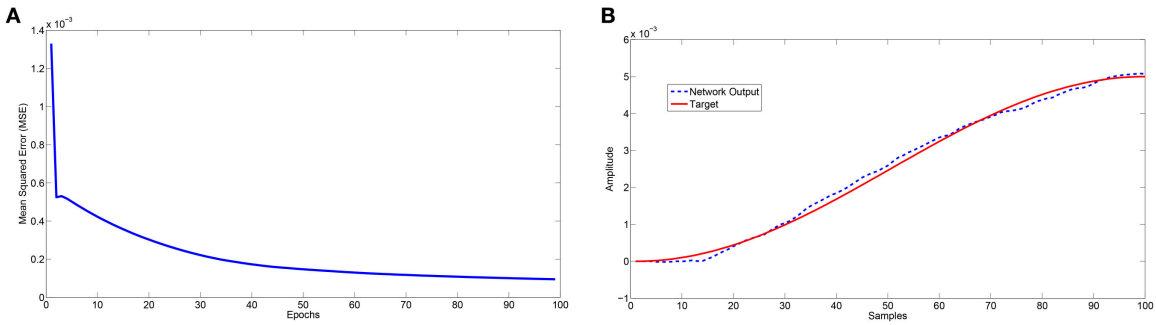


FIGURE 3 | (A) Trend of the mean square error during 100 learning trials (epochs), **(B)** Comparison between the expected output and the network approximation at the end of the 100 learning trials.

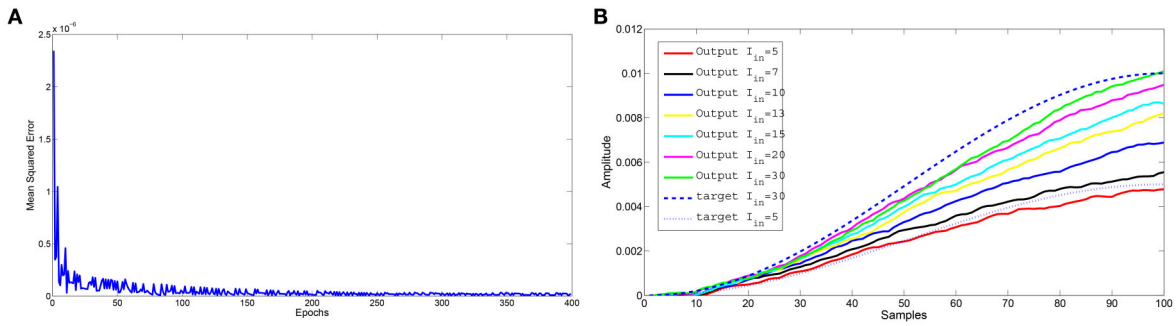


FIGURE 4 | Learning example with two different input-target pairs: input current 5 and 30μA. The interpolation capabilities were tested using different inputs: 5, 7, 10, 13, 15, 20, and 30μA. **(A)** Trend of the mean square error during 400 epochs used to learn the two input-target patterns. For each epoch we provided alternatively either one or the other input-target pair, allowing the learning process to update the read-out weights obtained during the previous learning epoch. **(B)** Comparison between the expected output and the network approximation at the end of the epochs.

between the current output and the desired target. The updating rule of the synaptic weights is here reported:

$$W_{i,j}(t+\delta t) = W_{i,j}(t) + \eta * Z_{i,j}(t) * E(t) \quad (6)$$

where η is the learning rate, $Z_{i,j}(t)$ is the synaptic output of the neuron (i, j) at time t and $E(t)$ is the error between the network output neuron and the desired target. Another possibility consists of cumulating the weight variations during the simulation time window, to finally apply the cumulative result during the last simulation step.

4. SIMULATION RESULTS

The analysed motor learning process consists of adopting a series of perturbations on specific leg control parameters to reach a success in the assigned task. To apply a smooth perturbation, we adopted as target signal, a cosinusoidal function, whose final value corresponds to the parameter to be applied. In the following simulations we adopted a lattice with 8 x 8 neurons that is a good compromise to obtain a considerable variety of internal dynamics. The learning process needs a series of iterations (here called epochs) to successfully store the information in

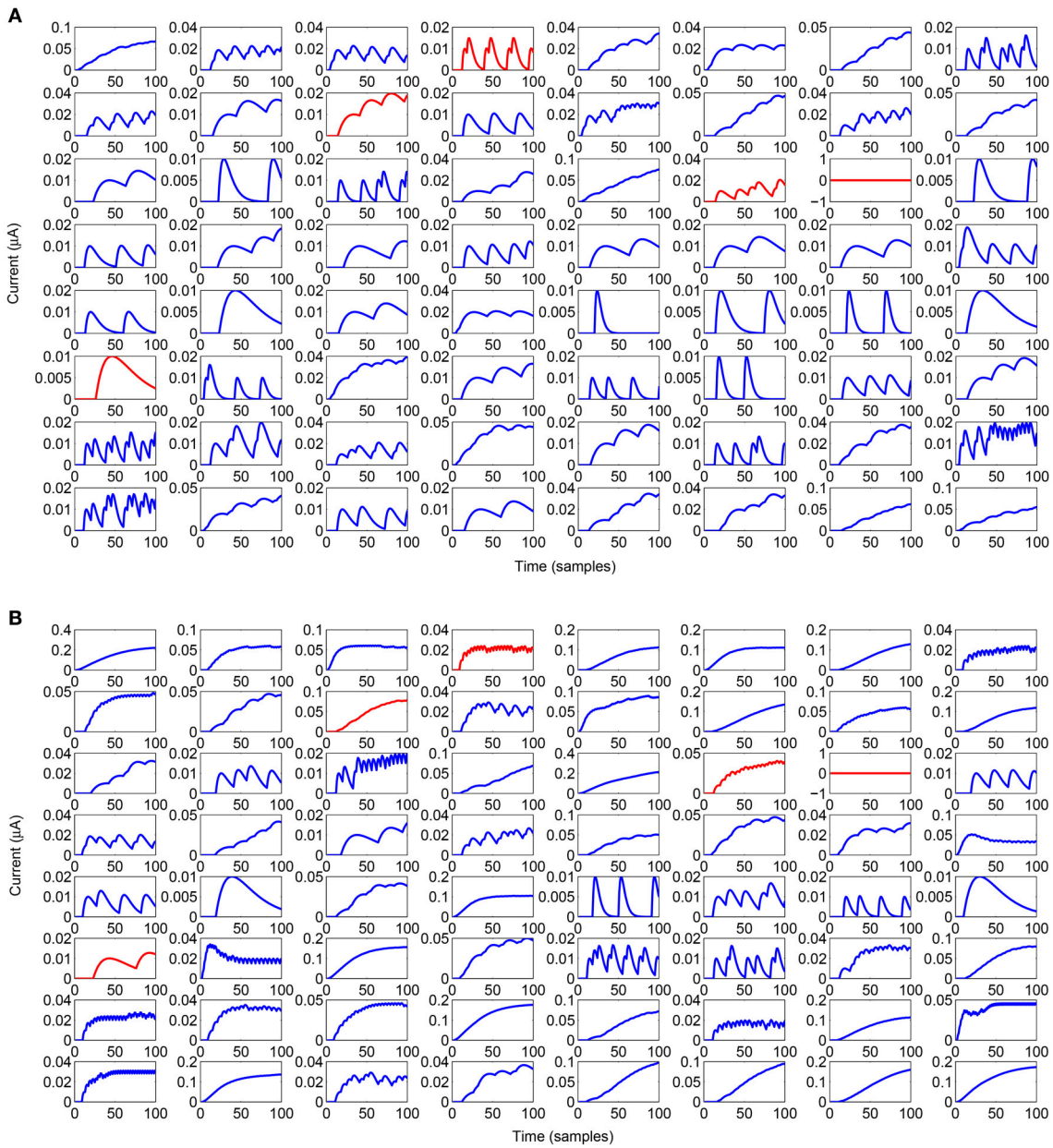


FIGURE 5 | Behavior of the input currents provided to the output neuron from the lattice when the input layer provides a current of 5μA (A) and 30μA **(B)**. Inhibitory neurons are outlined in red.

the read-out maps. In the following analysis we considered 100 epochs with a learning rate $\eta = 0.5$. During each epoch the network is simulated for 100 integration steps. A typical activity of the neural lattice is shown in **Figure 2**. The input given to the network through an external current is related to the information acquired from the environment and, using the learning rule in Equation (6), we can determine the weights of the read-out map in order to follow a target signal as shown in **Figure 3**.

The network allows to interpolate the information acquired during learning as illustrated in **Figure 4**. During the 400 epochs used for the learning phase, two distinct output signals, corresponding to different input currents ($I_{in} = 5$ and $30\mu A$), were learned. During the testing phase, besides the two inputs already used in the learning phase, also other input currents were provided obtaining plausible behaviors that interpolate the dynamics of the two learned target signals.

Figure 5 reports the synaptic activity (Equation 3), in the form of currents generated by the lattice before learning, weighted by the read-out map and summed over the 100 samples for all the spikes emitted by the neurons to reproduce the two target signals. It can be noticed that even a lattice with a limited number of neurons can produce a large variety of dynamics that can be combined by the output neurons. The differences in the synaptic time constant, play a role in increasing the richness of dynamics during the network activity. It is also evident how sensitive the structure is to a change in the input current provided to the lattice; it can generate a drastic change in the temporal evolution of the network dynamics. This allows for a high interpolation capability. The use of spiking networks over nonspiking ones to model nonlinear dynamics is often considered as an additional complication. Our case is an example to the contrary. In fact, in Arena et al. (2013a) nonlinear nonspiking recurrent neural networks were used to model MB activity: the non-spiking recurrent configuration, suitable for solving the motor-learning problem was fixed to 140 non locally connected units, whereas the results presented in this paper were obtained via a network with 64 spiking locally connected neurons in the liquid layer.

5. MOTOR LEARNING: APPLICATION TO CLIMBING

5.1. Learning New Motor Activities in a Stable Locomotion Controller

The insect brain can be considered as a parallel computing architecture where reflexive paths serve the basic needs for survival, whereas learned paths allow the formation of more complex behaviors.

Regarding motor activities in insects, the thoracic ganglia are mainly responsible for the generation of locomotion gaits, and the Central Pattern Generator (CPG) has widely been accepted as being the core unit for locomotion control but its fine-tuning is usually achieved by sensory information. The approach proposed here considers the task of motor learning as that of finding a suitable way for modifying the basic motor trajectories on the single leg joints so as to improve motor-skills in the light of novel conditions imposed by the environment. Using a control approach, we can realize motor-skill learning through a hierarchical adaptive controller, where, when facing novel conditions, some parameters controlling the leg joint trajectories are modulated. These modulations, shaped by the

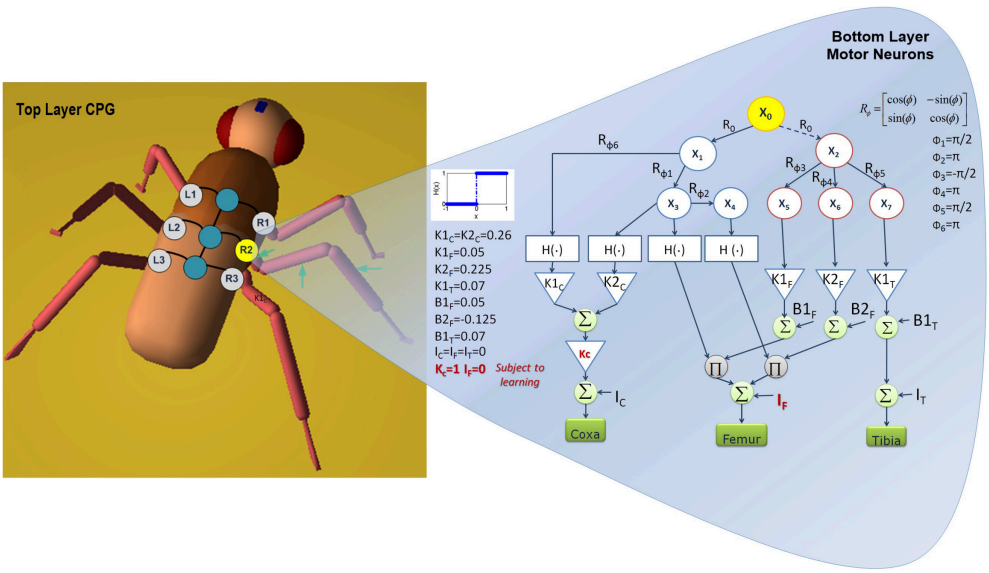


FIGURE 6 | Neural network scheme: the top layer generates a stable gait pattern, whereas the bottom layer is constituted by additional sub-networks generating the specific reference signals for the leg joints. The network devoted to control a middle leg is reported. The parameter adapted during the learning process for the middle legs are indicated in red.

kinematic constrains, realize novel leg trajectories which are then applied to assess their suitability for the task. Once the former locomotion conditions are restored, these modulations are withdrawn and the baseline stable locomotor activity re-emerges. Sets of successful parametric values are retained, so that they can be re-applied whenever similar conditions should be encountered again. The locomotion controller is made up of basically two networks: one is devoted to generate a stable phase displacement among the legs; the other is shaped on the specific kinematic structure of each leg and constituted by several motor neuron structures, as illustrated in **Figure 6**. The basic cell characterizing the CPG architecture is described by the following equations:

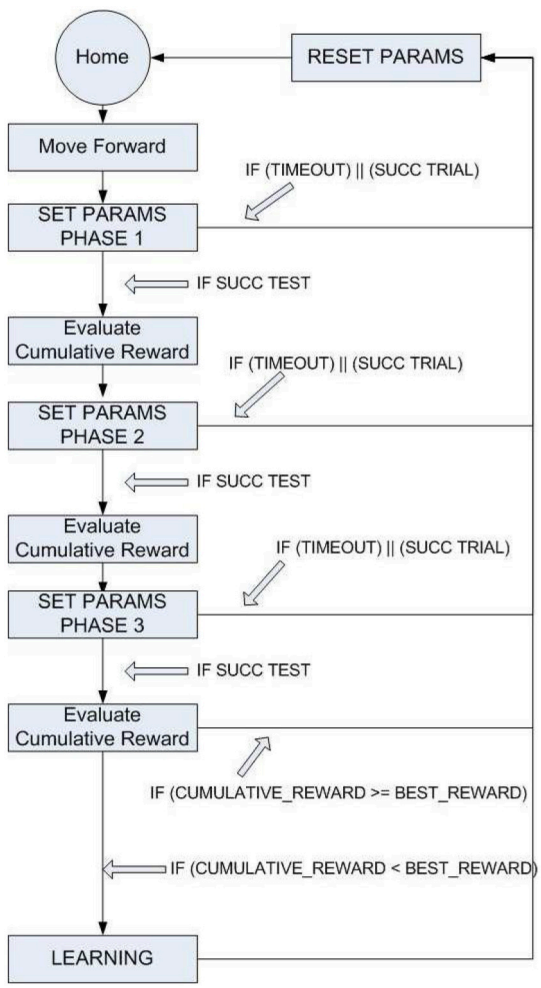


FIGURE 7 | General scheme of the procedure followed to improve the robot motor-skills in a multi-stages task. Starting from Home, an event triggers the request of parameter adaptation for the Step 1 that is tried until a success occurs or a time-out is reached. Within the time-out triggered window, it is possible to evaluate the effectiveness of multiple sets of parameters that persist for about a complete cycle of a leg (i.e., overtime). The success is evaluated by a cumulated reward and, if an improvement is obtained, the parameter evolution is stored in the long-term memory (i.e., read-out maps). The other stages follow the same procedure.

$$\begin{cases} \dot{x}_{1,i} = -x_{1,i} + (i + \mu + \varepsilon)y_{1,i} - s_1 y_{2,i} + i_1 \\ \dot{x}_{2,i} = -x_{2,i} + s_2 y_{1,i} + (i + \mu - \varepsilon)y_{2,i} + i_2 \end{cases} \quad (7)$$

with $y_i = tanh(x_i)$ and the parameters for each cell: $\mu = 0.23, \varepsilon = 0, s_1 = s_2 = 1, i_1 = i_2 = 0$ generate a stable limit cycle (Arena et al., 2005). μ is chosen to approximate the dynamics to a harmonic oscillation. The CPG network is built connecting adjacent cells using links expressing rotational matrices $R(\phi)$, as follows:

$$\dot{x}_i = f(x_i, t) + k \sum_{j \neq i} (R(\phi_{i,j})x_j - x_i) \text{ with } i, j = 1, \cdots, n \quad (8)$$

where the summation involves all the neurons j which are nearest neighbor to the neuron i; n is the total number of cells; $f(x_i, t)$ represents the reactive dynamics of the i-th uncoupled neurons as reported in Equation (7) and k is the strength of the connections. The sum of terms performs diffusion on adjacent cells and induces phase-locking as a function of rotational matrices (Seo and Slotine, 2007). The presence of local connections is an important added value because it reduces the system complexity in view of a hardware implementation. The bottom layer is designed based on the desired kinematic behavior; it is directly correlated to the morphology of the limb. The network controlling one of the middle legs is sketched in **Figure 6**. The CPG neuron identified with the label $R2$ is connected through rotational matrices with different angles to a network of motor neurons arranged in a directed tree graph that uses the same neuron model as CPG. The blocks $H(\bullet)$ are Heaviside functions and are used to distinguish, within the limit cycle, between the stance and swing phases: this allows to associate suitable modulation parameters to each part of the cycle, depending on the morphology of the leg. The signals are finally merged to generate the position control command for the coxa, femur and tibia joints. A detailed discussion on the CPG structure and behaviors is reported in a previous study (Arena E. et al., 2012).

The overall network stability was theoretically proven exploiting tools from partial contraction theory on a network made of nonlinear oscillators with Laplacian couplings. As demonstrated in previous studies, the network for gait control has a diffusive, undirected tree-graph configuration, which guarantees asymptotic phase stability independently of any imposed locomotion pattern (Arena et al., 2011; Arena E. et al., 2012).

The stable phase-locked oscillations generated in that way are passed on to the motor neural network for each leg, whose particular structure controls leg motion while maintaining the imposed phase among the legs. Upon this stable basic locomotor activity, the motor-learning controller is added, whose role is to find suitable modulation of the single-leg motions to learn proper trajectories in the presence of specific needs. Basic motor activities are so disturbed to find new solutions for the leg motions, thus implementing motor-skill learning.

5.2. Climbing Experiment

Motor-skill learning in the presented multi-limb system is applied to improve the robot capabilities in solving different tasks

involving multiple degrees of freedom; here, in fact a fine tuning of parameters is required to modulate the basic cycling behavior in the different legs of the robot.

Among the possible tasks, in the simulation a step-climbing scenario has been considered in the simulation. In nature, insects are continuously faced with uneven terrains and they adapt their motor responses to accomplish tasks like climbing over surmountable objects. Even flying insects, like *D. melanogaster*, show exquisite climbing skills, since searching for food in the near-field and courtship are achieved during walking. The aim of motor learning in our experiments is to improve the climbing capabilities of a simulated robot through the modulation in time of a group of parameters used in the leg motor layer. This simulated scenario is a realistic alternative to the gap climbing scenario used in the biological experiments (Blasing and Cruse, 2004; Pick and Strauss, 2005; Kienitz, 2010; Triphan et al., 2010). In fact, due to the adhesive capability of the fly legs (possessing pulvilli and claws), gap climbing is an affordable task for the real insect, whereas this is extremely difficult for a *Drosophila*-inspired robot that cannot reach the same dexterity as the biological counterpart. In other hexapod robots the presence of an active body joint, inspired by the cockroach, was exploited to improve the system capabilities in gap climbing tasks (Goldschmidt et al., 2014). In our *Drosophila*-inspired robot, due to the absence of this degree of freedom in the body, we considered obstacle-climbing scenarios, which are a challenging task for legged robots that have to improve their climbing capabilities by learning. For a future direct comparison with biological experiments, the new paradigm lends itself for testing real flies. Moreover, the step climbing scenario can be made more demanding by using slippery surfaces which would reduce the advantage of the animal if compared with the robot.

Step climbing for a robot is quite a complex task and should involve an optimization method to adapt the joint movements to different surfaces. To simplify the problem, the task was split into different phases shown in **Figure 7**.

The approaching phase is guided by the visual system that is able to recognize the distance from the obstacle and its height. When the robot's distance from the step is below a threshold, Phase 1 is activated and the parameters of the front legs are adapted using the RFG to modify its movements, in an attempt to find a foot-hold on the step. For sake of simplicity, a subset of parameters available in the adopted CPG was subjected to learning in this phase.

In details, for the coxa joint the bias value, for the femur joint the gain value, for the tibia joint the bias and gain values were selected for learning. This phase leads to a stable positioning of the front legs on the step, with the body lifted off. The extent of the angular motion of the leg joints, caused by the modulation of the parameter profiles, is used as an index of the energy spent

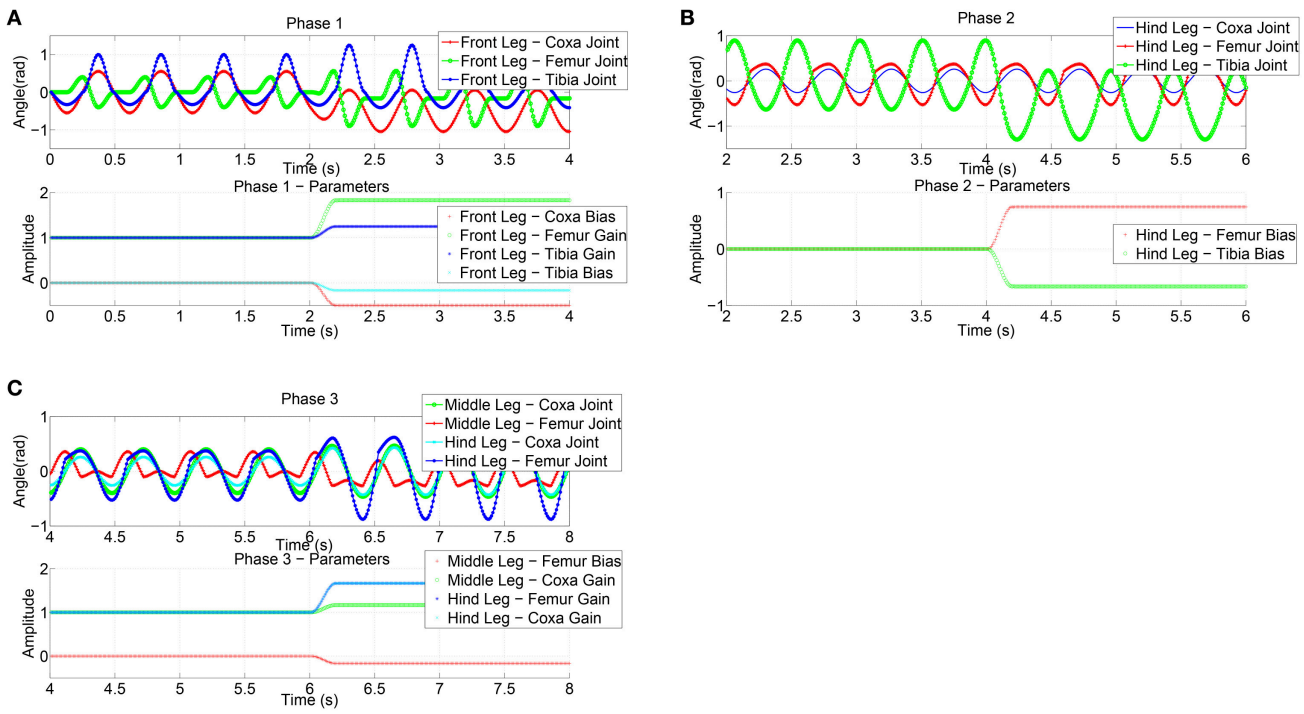


FIGURE 8 | Effects of the parameter adaptation on the leg joints (only the left body side is shown). A limited number of parameters is subject to learning in the three phases of the obstacle climbing procedure: **(A)** in phase 1 only the front legs are involved, **(B)** the hind legs in phase 2, and **(C)** both middle and hind legs in phase 3. The effect of the parameters on the leg joint trajectory is limited to the current phase.

in this task and to define a reward function. The reward value is then compared with the previously found best value and, if an improvement is obtained, the new sets of functions are stored in the SNN readout map. For the considered task we have a single lattice with one input (i.e., step height) and a total of ten readout maps, one for each parameter to be learned for a specific leg joint. The SNN receives as input a normalized value related to the step height and the lattice dynamics generates a spatio-temporal spiking activity that is transformed in a continuous, non spiking signal, through the output synapses that converge on the output neurons, one for each parameter that is subject to learning.

A series of experiments were performed using a step that is insurmountable unless a gait adaptation is introduced: the height of the step is around 0.9 mm, whereby we chose the simulated *Drosophila* body length as 3.2 mm and the average height of the center of mass as to be located at about 1 mm above the ground during forward walking.

The joint angular positions caused by the parameter adaptation in the anterior legs are shown in **Figure 8A**. The subsequent phase is similar: here as relevant parameters to be adapted, the bias of femur and tibia joints of the hind legs are considered to facilitate the climbing of the middle legs. The event considered in this phase to evaluate the success

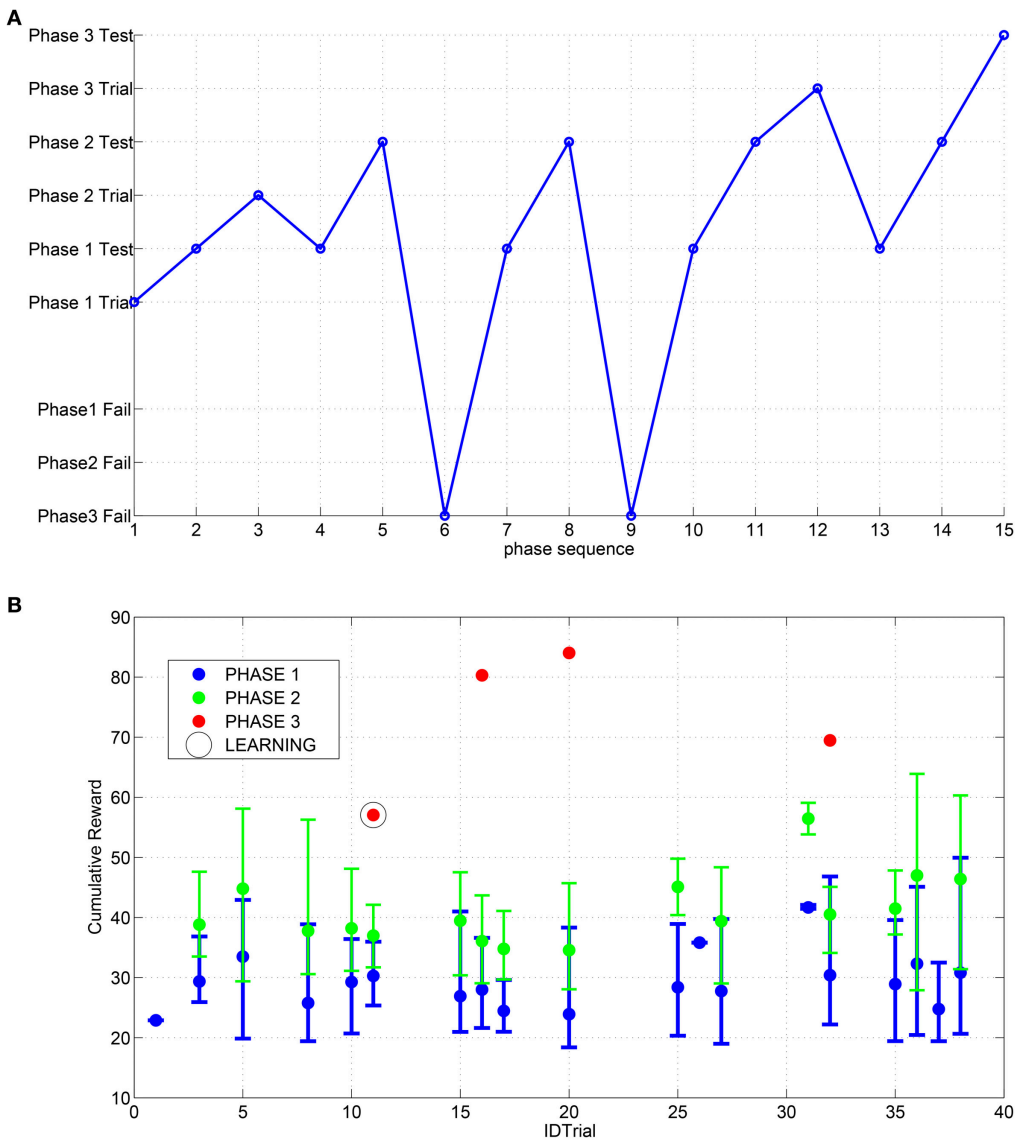


FIGURE 9 | (A) Sequence of events obtained during the searching process for the suitable parameters through the RFG. **(B)** Distribution of the cumulative reward for each trial; the error bars indicate the range of excursion between min and max value and the marker corresponds to the mean value. The learning of the SNN is performed only for the IDTrial 11 that is the same as reported in **(A)** because in the other success event for the third step there are no improvements for the cumulative reward.

and the consequent passage to the successive phase is the horizontal position of the center of mass of the robot with respect to the obstacle. The parameter adaptation results for the second phase are depicted in **Figure 8B**. During the third phase the robot elevates the hind legs on the step: this is achieved by modulating the gain of the coxa and bias of the femur joint for the middle legs and the gain of the coxa and femur joint of the hind legs (**Figure 8C**). In the actual experiments the function adopted to deliver the randomly generated parameter modulation on the joints is a cosinusoid, however other functions, like exponentials, quarter sinusoids, or sigmoids could be used. Actually the function reaches the steady state value in a given time window that is a portion of a stepping cycle.

In the dynamic simulation herewith reported, we adopted an integration time dt = 0.01 s, a stepping cycle of about 1.5 s: these conditions, the parameters reach the steady state within [20–60] integration steps. Looking at the learning process, the RFG generates the new parameters to be tested for the first phase. If the trial is successful the robot is re-placed to the starting position to perform a test: this assures the robustness of this new set of parameters. If the robot succeeds, it can proceed to the second phase, otherwise the parameters are discarded and the first phase is repeated. The trial ends when the robot overcomes the last phase or after a given number of attempts (i.e., 15 events). If this time-out occurs, the parameters just used for the phases are discarded because they are not globally suitable for a complete climbing behavior.

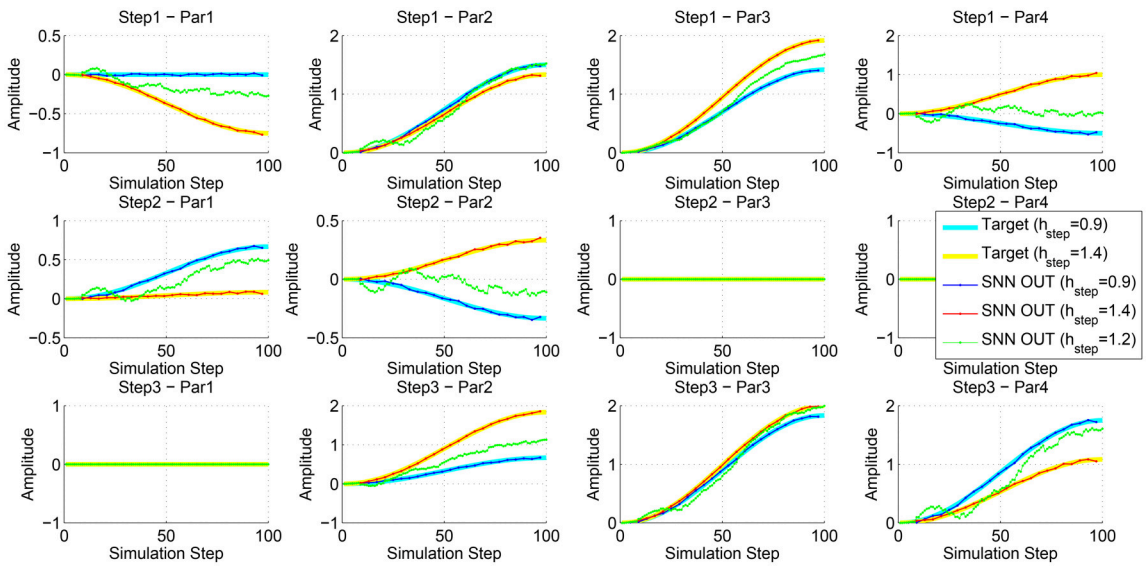


FIGURE 10 | Comparison between the best parameters provided by the RFG and the output of the SNN after the read-out map learning for a step of 0.9 and 1.4 mm. Moreover, the output of the network for the input of an intermediate step height is shown.

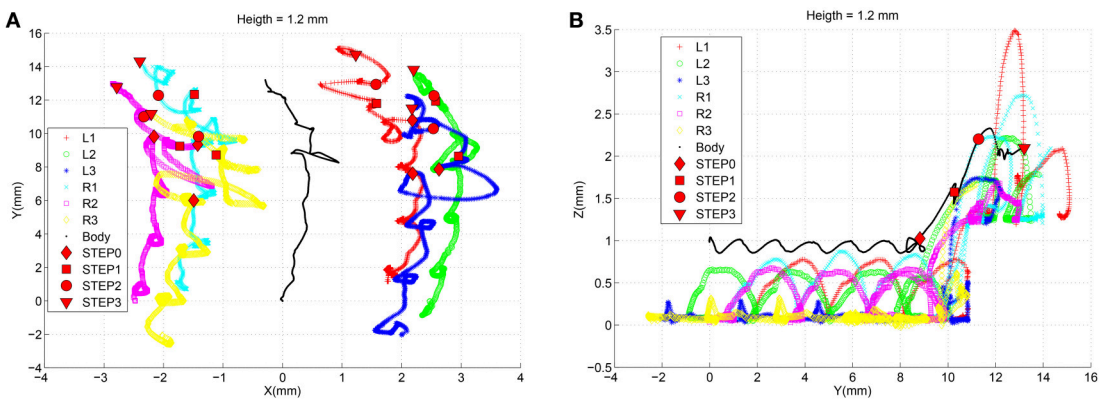


FIGURE 11 | Trajectories followed by the center of mass of the robot and by the tip of each leg during the climbing behavior facing an obstacle of 1.2 mm. A marker is placed in the signals to indicate when the robot completes each phase.

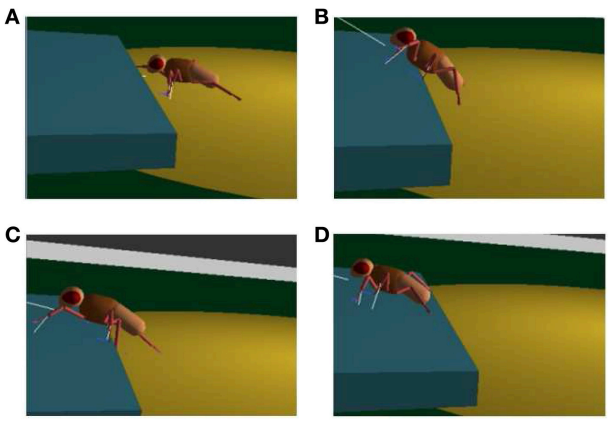


FIGURE 12 | Snapshots of the fly-inspired robot while climbing an obstacle: the robot approaches the obstacle (A); it reaches the step with its front legs **(B)**, middle legs **(C)**, and finally with the hind legs **(D)**.

In **Figure 9A** an example of a trial is reported: the robot succeeds in the first attempt to find a suitable set of parameters to complete the first and second phase, whereas for the third phase a series of failures both in learning and in test are obtained until the final success is reached (see Supplementary Video 1 for a typical sequence of trials with successes and failures). The success in the trial can be followed by a learning process in the SNN depending on the overall reward value obtained. In **Figure 9B** the distribution of the cumulative reward in a campaign is shown. For each trial the success condition for each phase can be reached multiple times until the complete climbing behavior is tested successfully or otherwise a time-out occurs. If the obtained cumulative reward (i.e., sum of the rewards for each phase) after the third phase is lower than the previously obtained values, the parameters are learned by the network.

To evaluate the interpolation capabilities of the network we also performed a series of learning sessions with higher obstacles (i.e., 1.4 mm) and subsequently we tested the robot with a step height never provided during learning (i.e., 1.2 mm). The best-adapted parameters obtained for the two learned step heights are reported in **Figure 10** together with the network response to the new step with an intermediate height. The obtained results were tested with the simulated robot obtaining a success in the climbing behavior as reported in **Figure 11**. This depicts the motion of the robot's center of mass (COM) and of the tips of each leg when climbing a 1.2 mm step. The edge of the step is placed at 11 mm far from the COM home position (along the y axis) (see Supplementary Video 2). Moreover, a series of snapshots outlining the posture of the fly-inspired robot during the climbing task are depicted in **Figure 12**.

To evaluate the generalization capability of the control system, the previously learned system was tested in a different scenario where a stair-like obstacle was introduced. The robot followed the same climbing procedure as described above, repetitively applied for the three stair steps encountered on its path with height 1.3, 1.1, and 0.9 mm, respectively. The detection of each obstacle produces an effect at the motor level on the basis of the parameter adaptation mechanism induced by the SNN. **Figure 13A** shows the trend of the joint position angles for the left-side legs during the whole climbing procedure. The adapted parameters produce changes in the leg movements during the different climbing phases as illustrated in **Figure 13 B** where the dynamics of the robot COM and the leg tip positions are reported (see Supplementary Video 3).

The results obtained were achieved relying only on the adaptive capabilities of the legs acquired during the learning phase. The body structure was considered rigid as in the fruit fly case. Including in the robotic structure active body joints (Dasgupta et al., 2015), mimicking the body of other insects like cockroaches, would only improve the robot capabilities. Therefore, the proposed control strategy can be also applied to other different robotic structures to improve their motor capabilities in fulfilling either obstacle climbing tasks or other similar scenarios affordable for the robot under consideration.

6. REMARKS AND CONCLUSIONS

In this paper a bio-inspired, embodied, closed-loop neural controller has been designed and implemented in a simulated hexapod robot that is requested to improve its motor-skills to face unknown environments. Taking inspiration from the insect brain and in particular from the fruit fly, the following hypotheses were formulated: relevant role of MB neuropiles in the motor learning task; direct transfer of the important role of transient dynamics in the olfactory leaning from the locust to the fly brain and further extension to motor learning; design of a neuro-computational model based on a LSM-like structure for the implementation of obstacle climbing in a simulated hexapod robot.

In details, a computational model for motor-skill learning was developed and realized in a dynamic simulation environment. Inspired by behavioral experimental campaigns of motor learning in real insects, the computational structure consisted in a randomly connected SNN that generates a multitude of nonlinear responses after the presentation of time dependent input signals. By linearly combining the output from the lattice neurons with a weighted function, a reward-based strategy allows to learn the desired target by tuning the weights of a readout map. Looking at MBs in insects, the idea of a pool of neurons enrolled to solve different tasks depending on the specific requested output is next to the concept of Neural Reuse which has a number of biological evidences. The reported results demonstrate that the system can learn, through a reward-driven mechanism, the time evolution of several independent parameters related to the leg movements, to improve the robot climbing capabilities when exposed to the step-climbing task. The robot was also able to deal with step heights never presented before, exploiting the interpolation abilities of the proposed network.

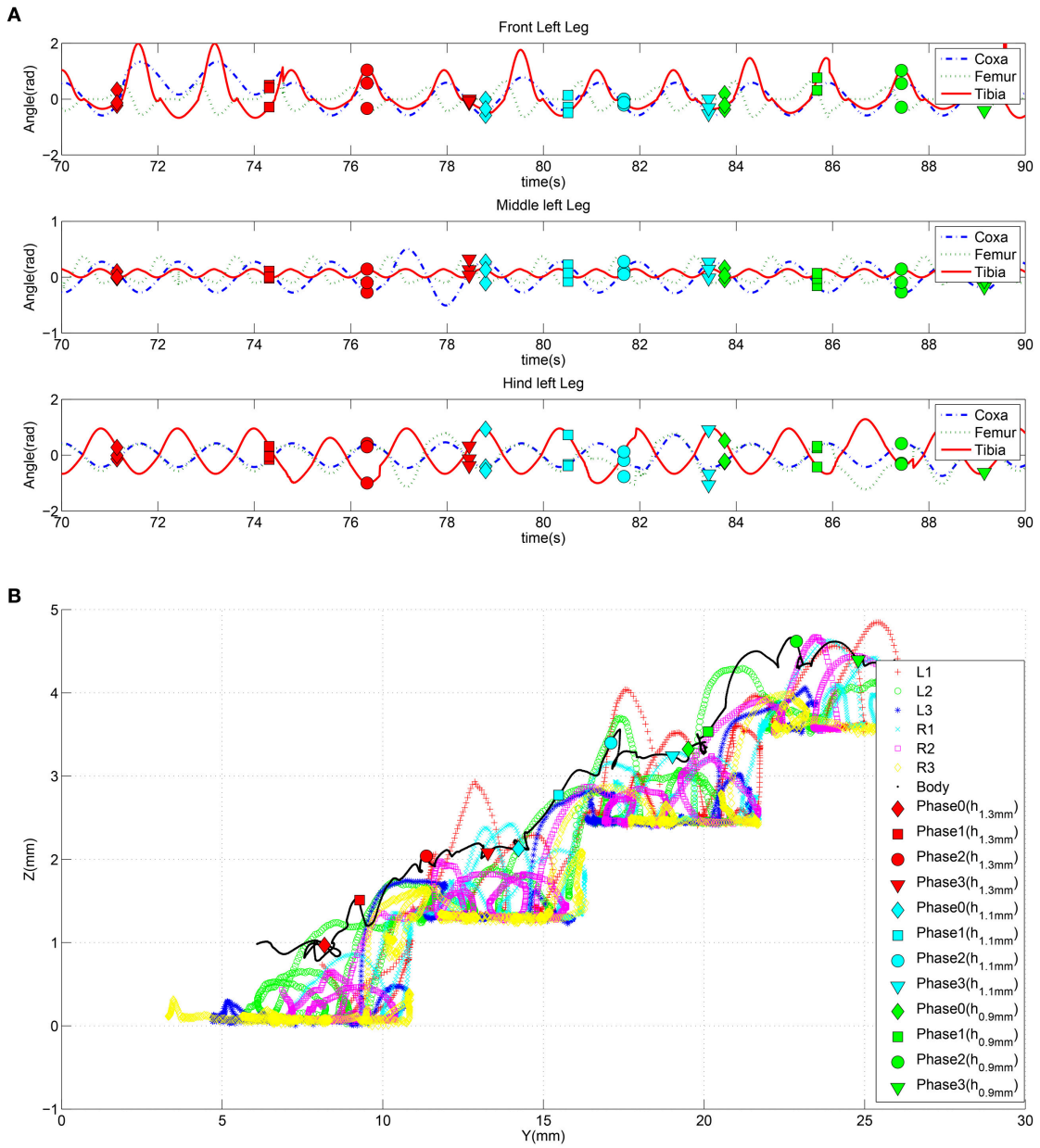


FIGURE 13 | (A) Trend of the joint positions for the left side legs when the robot faces a series of obstacles. **(B)** Trajectories followed by the center of mass of the robot and by the tip of each leg during the climbing behavior facing with multiple obstacles with height 1.3, 1.1, and 0.9 mm, respectively. A marker is placed in the signals to indicate when the robot completes each climbing phase.

AUTHOR CONTRIBUTIONS

All authors listed, have made substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

This work was partially supported by the FIR 2014 Project *Self-reconfigurable, flexible, multi-body, lowcost, modular robot systems for exploration of dynamical, hostile or unknown environments*, and by MIUR Project CLARA (*CLoud Platform for lAndslide Risk Assessment*) and by EC FP7 EMICAB (*Embodied motion intelligence for cognitive autonomous robots*).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00012/full#supplementary-material

REFERENCES

Anderson, M. (2010). Neural reuse: a fundamental organizational principle of the brain. *Behav. Brain Sci.* 33, 245–266. doi: 10.1017/s0140525x10000853

Arena, E., Arena, P., and Patané, L. (2011). "Efficient hexapodal locomotion control based on flow-invariant subspaces," in *18th World Congress of the International Federation of Automatic Control (IFAC), Vol. 44*. (Milan), 13758–13763.

Arena, E., Arena, P., and Patané, L. (2012). "CPG-based locomotion generation in a Drosophila inspired legged robot," in *Biorob 2012* (Rome), 1341–1346.

Arena, P., Caccamo, S., Patané, L., and Strauss, R. (2013a). "A computational model for motor learning in insects," in *International Joint Conference on Neural Networks (IJCNN)* (Dallas, TX), 1349–1356.

Arena, P., De Fiore, S., and Patané, L. (2009). Cellular nonlinear networks for the emergence of perceptual states: application to robot navigation control. *Neural Netw.* 22, 801–811. doi: 10.1016/j.neunet.2009.06.024

Arena, P., De Fiore, S., Patané, L., Pollino, M., and Ventura, C. (2010). "Insect inspired unsupervised learning for tactic and phobic behavior enhancement in a hybrid robot," in *WCCI 2010 IEEE World Congress on Computational Intelligence* (Barcelona), 2417–2424.

Arena, P., Fortuna, L., and Branciforte, M. (1999). Reaction-diffusion CNN algorithms to generate and control artificial locomotion. *IEEE Trans. Circ. Syst. I* 46, 253–260.

Arena, P., Fortuna, L., Frasca, M., and Patané, L. (2005). A CNN-based chip for robot locomotion control. *IEEE Trans. Circ. Syst. I* 52, 1862–1871. doi: 10.1109/tcsi.2005.852211

Arena, P., and Patané, L. (2014). *Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots II: An Insect Brain Computational Model*. Cognitive Systems Monographs, Vol. 21. Dordrecht: Springer.

Arena, P., Patané, L., Stornanti, V., Termini, P., Zaepf, B., and Strauss, R. (2013b). Modelling the insect mushroom bodies: application to a delayed match-to-sample task. *Neural Netw.* 41, 202–211. doi: 10.1016/j.neunet.2012.11.013

Arena, P., Patané, L., and Strauss, R. (2013c). "The insect mushroom bodies: a paradigm of neural reuse," in *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)* (Taormina: MIT Press), 765–772.

Arena, P., Patané, L., and Termini, P. S. (2012a). Learning expectation in insects: a recurrent spiking neural model for spatio-temporal representation. *Neural Netw.* 32, 35–45. doi: 10.1016/j.neunet.2012.02.034

Arena, P., Patané, L., and Termini, P. (2012b). "Modeling attentional loop in the insect mushroom bodies," in *International Joint Conference on Neural Networks (IJCNN 2012)* (Brisbane, QLD), 7–12.

Blasing, B. (2006). Crossing large gaps: a simulation study of stick insect behavior. *Adapt. Behav.* 14, 265–285. doi: 10.1177/105971230601400307

Bläsing, B., and Cruse, H. (2004). Mechanisms of stick insect locomotion in a gap crossing paradigm. *J. Comp. Physiol.* 190, 173–183. doi: 10.1007/s00359-003-0482-3

Brembs, B. (2009). Mushroom bodies regulate habit formation in *Drosophila*. *Curr. Biol.* 19, 1351–1355. doi: 10.1016/j.cub.2009.06.014

Brembs, B., and Heisenberg, M. (2000). The operant and the classical in conditioned orientation of *Drosophila melanogaster* at the flight simulator. *Learn. Mem.* 7, 104–115. doi: 10.1101/lm.7.2.104

Broussard, D. M., and Karrardjian, C. D. (2004). Learning in a simple motor system. *Learn. Mem.* 11, 127–136. doi: 10.1101/lm.65804

Byrne, J. (2008). *Concise Learning and Memory*. London: Academic Press.

Chittka, L. (1998). Sensorimotor learning in bumblebees: long term retention and reversal training. *J. Exp. Biol.* 201, 515–524.

Chittka, L., and Niven, J. (2009). Are bigger brains better? *Curr. Biol.* 19, 995–1008. doi: 10.1016/j.cub.2009.08.023

Cruse, H. (2002). MMC - a new numerical approach to the kinematics of complex manipulators. *Mech. Mach. Theory* 37, 375–394. doi: 10.1016/S0094-114X(01)00080-5

Cruse, H., Kindermann, T., Schumm, M., Dean, J., and Schmitz, J. (1998). Walknet - a biologically inspired network to control six-legged walking. *Neural Netw.* 11, 1435–1447.

Dasgupta, S., Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2015). Distributed recurrent neural forward models with synaptic adaptation and CPG-based control for complex behaviors of walking robots. *Front. Neurorobot.* 9:10. doi: 10.3389/fnbot.2015.00010

Dayan, E., and Cohen, L. (2011). Neuroplasticity subserving motor skill learning. *Neuron* 72, 443–454. doi: 10.1016/j.neuron.2011.10.008

Erber, J., Pribbenow, B., Grandy, K., and Kierzek, S. (1997). Tactile motor learning in the antennal system of the honeybee (*Apis mellifera* L.). *J. Comp. Physiol. A* 181, 355–365.

Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2014). Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots. *Front. Neurorobot.* 8:3. doi: 10.3389/fnbot.2014.00003

Gronenberg, W., and López-Riquelme, G. (2004). Multisensory convergence in the mushroom bodies of ants and bees. *Acta Biol. Hung.* 55, 31–37. doi: 10.1556/ABiol.55.2004.1-4.5

Hanesch, U., Fischbach, K., and Heisenberg, M. (1989). Neuronal architecture of the central complex in Drosophila melanogaster. *Cell Tissue Res.* 257, 343–366.

Horridge, G. A. (1962). Learning of leg position by headless insects. *Nature* 193, 697–698.

Izhikevich, E. (2000). Neural excitability, spiking, and bursting. *Int. J. Bifurcation Chaos* 10, 1171–1266. doi: 10.1142/S0218127400000840

Jaeger, H. (2001). *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks*. GMD Report 148, GMD - German National Research Institute for Computer Science.

Kienitz, B. (2010). *Motorisches Lernen in Drosophila Melanogaster*. Ph.D. thesis, Shaker Verlag, Aachen.

Krause, A., Bläsing, B., Dürr, V., and Schack, T. (2009). "Direct control of an active tactile sensor using echo state networks," in *Human Centered Robot Systems*, eds H. Ritter, G. Sagerer, R. Dillmann, and M. Buss (Berlin; Heidelberg: Springer), 11–21.

Lewis, A. (1986). Memory constraints and flower choice in Pieris rapae. *Science* 232, 863–865.

Lin, A. C., Bygrave, A. M., de Calignon, A., Lee, T., and Miesenböck, G. (2013). Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nat. Neurosci.* 17, 559–568. doi: 10.1038/nn.3660

Liu, L., Wolf, R., Ernst, R., and Heisenberg, M. (1999). Context generalization in *Drosophila* visual learning requires the mushroom bodies. *Nature* 400, 753–756.

Liu, X., and Davis, R. (2006). Insect olfactory memory in time and space. *Curr. Opin. Neurobiol.* 6, 679–685. doi: 10.1016/j.conb.2006.09.003

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560. doi: 10.1162/089976602760407955

Manganaro, G., Arena, P., and Fortuna, L. (1999). *Cellular Neural Networks: Chaos, Complexity and VLSI Processing*. Berlin; Heidelberg: Springer-Verlag.

Mazor, O., and Laurent, G. (2005). Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. *Neuron* 48, 661–673. doi: 10.1016/j.neuron.2005.09.032

Menzel, R. (2001). Searching for the memory trace in a mini-brain, the honeybee. *Learn. Mem.* 8, 53–62. doi: 10.1101/lm.38801

Menzel, R., and Giurfa, M. (1996). Dimensions of cognitive capacity in an insect, the honeybee. *Behav. Cogn.* 5, 24–40.

Menzel, R., and Muller, U. (1996). Learning and memory in honeybees: from behaviour to neural substrates. *Learn. Mem.* 19, 379–404.

Mohl, B. (1993). The role of proprioception for motor learning in locust flight. *J. Comp. Physiol. A* 172, 325–332.

Moore, E., Campbell, D., Grimminger, F., and Buehler, M. (2002). "Reliable stair climbing in the simple hexapod RHex," in *IEEE International Conference on Robotics and Automation* (Washington, DC), 2222–2227.

Mronz, M., and Strauss, R. (2001). "Genetic dissection of the fly's preference for near objects and the ability to abandon previously chosen targets," in *Göttingen Neurobiology Report 2001*, eds N. Elsner and G. Kreutzberg (Stuttgart; New York, NY: Thieme), 524.2.

Nowotny, T., Huerta, R., Abarbanel, H., and Rabinovich, M. I. (2005). Self-organization in the olfactory system: one shot odor recognition in insects. *Biol. Cybern.* 93, 436–446. doi: 10.1007/s00422-005-0019-7

Nowotny, T., Rabinovich, M. I., Huerta, R., and Abarbanel, H. D. (2003). Decoding temporal information through slow lateral excitation in the olfactory system of insects. *J. Comput. Neurosci.* 15, 271–281. doi: 10.1023/A:1025825111088

Pavone, M., Arena, P., Fortuna, L., Frasca, M., and Patané, L. (2006). Climbing obstacle in bio-robots via CNN and adaptive attitude control. *Int. J. Circ. Theory Appl.* 34, 109–125. doi: 10.1002/cta.345

Pick, S., and Strauss, R. (2005). Goal-driven behavioral adaptations in gap-climbing Drosophila. *Curr. Biol.* 15, 1473–1478. doi: 10.1016/j.cub.2005.07.022

Rodríguez-Vázquez, Á., Domínguez-Castro, R., Jiménez-Garrido, F., Morillas, S., Listán, J., Alba, L., et al. (2008). "The eye-ris cmos vision system," in *Analog Circuit Design: Sensors, Actuators and Power Drivers; Integrated Power Amplifiers from Wireline to RF; Very High Frequency Front Ends*, eds H. Casier, M. Steyaert, and A. H. M. Van Roermund (Dordrecht: Springer), 15–32.

Scherer, S., Stocker, R. F., and Gerber, B. (2003). Olfactory learning in individually assayed *Drosophila* larvae. *Learn. Mem.* 10, 217–225. doi: 10.1101/lm.57903

Schwaerzel, M., Monastirioti, M., Scholz, H., Friggi-Grelin, F., Birman, S., and Heisenberg, M. (2003). Dopamine and octopamine differentiate between aversive and appetitive olfactory memories in Drosophila. *J. Neurosci.* 23, 10495–10502. Available online at: http://www.jneurosci.org/content/23/33/10495.long

Seo, K., and Slotine, J. (2007). "Models for global synchronization in CPG-based locomotion," in *Proceedings 2007 IEEE International Conference on Robotics and Automation* (Rome), 281–286.

Smith, D., Wessnitzer, J., and Webb, B. (2008). A model of associative learning in the mushroom body. *Biol. Cybern.* 99, 89–103. doi: 10.1007/s00422-008-0241-1

Strauss, R. (2002). The central complex and the genetic dissection of locomotor behaviour. *Curr. Opin. Neurobiol.* 12, 633–638. doi: 10.1016/S0959-4388(02)00385-9

Tang, S., and Guo, A. (2001). Choice behavior of *Drosophila* facing contradictory visual cues. *Science* 294, 1543–1547. doi: 10.1126/science.1058237

Tani, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. Syst. Man Cybern. B Cybern.* 26, 421–436.

Triphan, T., Poeck, B., Neuser, K., and Strauss, R. (2010). Visual targeting of motor actions in climbing Drosophila. *Curr. Biol.* 20, 663–668. doi: 10.1016/j.cub.2010.02.055

Waddell, S. (2013). Reinforcement signaling in drosophila; dopamine does it after all. *Curr. Opin. Neurobiol.* 23, 324–329. doi: 10.1016/j.conb.2013.01.005

Wessnitzer, J., Young, J. M., Armstrong, J. D., and Webb, B. (2012). A model of non-elemental olfactory learning in *Drosophila. J. Neurophysiol.* 32, 197–212. doi: 10.1007/s10827-011-0348-6

Wolf, R., Voss, A., Hein, S., Heisenberg, M., and Sullivan, G. (1992). Can a fly ride a bicycle? *Philos. Trans. R. Soc. Lond. B* 337, 261–269.

Wu, C. L., Xia, S., Fu, T. F., Wang, H., Chen, Y. H., Leong, D., et al. (2007). Specific requirement of NMDA receptors for long-term memory consolidation in Drosophila ellipsoid body. *Nat. Neurosci.* 10, 1578–1586. doi: 10.1038/nn2005

Zhang, Z., Li, X., Guo, J., Li, Y., and Guo, A. (2013). Two clusters of GABAergic ellipsoid body neurons modulate olfactory labile memory in Drosophila. *J. Neurosci.* 33, 5175–5181. doi: 10.1523/JNEUROSCI.5365-12.2013

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Arena, Arena, Strauss and Patané. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 27 March 2017
doi: 10.3389/fnbot.2017.00016

Morphological Properties of Mass–Spring Networks for Optimal Locomotion Learning

Gabriel Urbain, Jonas Degrave, Benonie Carette, Joni Dambre and Francis Wyffels*

IDLab, Electronics and Information Systems Department, Ghent University – imec, Ghent, Belgium

Robots have proven very useful in automating industrial processes. Their rigid components and powerful actuators, however, render them unsafe or unfit to work in normal human environments such as schools or hospitals. Robots made of compliant, softer materials may offer a valid alternative. Yet, the dynamics of these compliant robots are much more complicated compared to normal rigid robots of which all components can be accurately controlled. It is often claimed that, by using the concept of morphological computation, the dynamical complexity can become a strength. On the one hand, the use of flexible materials can lead to higher power efficiency and more fluent and robust motions. On the other hand, using embodiment in a closed-loop controller, part of the control task itself can be outsourced to the body dynamics. This can significantly simplify the additional resources required for locomotion control. To this goal, a first step consists in an exploration of the trade-offs between morphology, efficiency of locomotion, and the ability of a mechanical body to serve as a computational resource. In this work, we use a detailed dynamical model of a Mass–Spring–Damper (MSD) network to study these trade-offs. We first investigate the influence of the network size and compliance on locomotion quality and energy efficiency by optimizing an external open-loop controller using evolutionary algorithms. We find that larger networks can lead to more stable gaits and that the system's optimal compliance to maximize the traveled distance is directly linked to the desired frequency of locomotion. In the last set of experiments, the suitability of MSD bodies for being used in a closed loop is also investigated. Since maximally efficient actuator signals are clearly related to the natural body dynamics, in a sense, the body is tailored for the task of contributing to its own control. Using the same simulation platform, we therefore study how the network states can be successfully used to create a feedback signal and how its accuracy is linked to the body size.

Keywords: morphological computation, mass–spring networks, morphological control, physical reservoir computing, soft robotics

OPEN ACCESS

Edited by:
Poramate Manoonpong, University of Southern Denmark Odense, Denmark

Reviewed by:
Kohei Nakajima, Kyoto University, Japan
Keyan Ghazi-Zahedi, Max Planck Institute for Mathematics in the Sciences, Germany
Helmut Hauser, University of Bristol, UK

****Correspondence:***
Gabriel Urbain
gabriel.urbain@ugent.be

Received: *23 December 2016*
Accepted: *06 March 2017*
Published: *27 March 2017*

Citation:
Urbain G, Degrave J, Carette B, Dambre J and Wyffels F (2017) Morphological Properties of Mass–Spring Networks for Optimal Locomotion Learning. Front. Neurorobot. 11:16. doi: 10.3389/fnbot.2017.00016

1. INTRODUCTION

Since its very early formulation, control theory has tried to automate increasingly complex systems (Fernández Cara and Zuazua Iriondo, 2003). The first implementations of PID controllers using feedback to regulate non-linear systems only originated in the first part of the twentieth century and were improved considerably, in particular with the progress in aerospace. More recently, with the evolution of the computation power and the advances in machine learning, the focus has evolved

toward the control of highly compliant systems with many degrees of freedom. Passive compliant robots indeed possess dynamical properties closer to animal bodies, whose performances can still not be reached, and show a real advantage for solving complex tasks in noisy human environments.

However, the framework for a theory allowing a deep understanding of such control systems—and hence engineering opportunities—is still under construction. It is largely believed that the concept of morphological computation can partly answer this issue, as it enables more fluent and robust motion control while providing adapted embodied controllers that use the body itself as a computational mean (Paul, 2006; Pfeifer and Bongard, 2006).

Nonetheless, the concept of morphological computation does not have a clear definition as discussed in Müller and Hoffmann (2016). In Füchslin et al. (2013), the authors refer to the first International Conference on Morphological Computation in Venice in 2007, where it was defined as "any process that serves a computational purpose, has clearly assignable input and output states, is programmable (i.e., the behavior can be adapted by varying a set of parameters) and has a sort of teleological embedding." This definition is however rather broad as it also includes every traditional digital computing means. Hereafter, we will restrict our definition to any way of increasing efficiency of computation or control in terms of energy, memory, time, etc. by outsourcing computational tasks to analogical physical systems. This interpretation follows the work of Pfeifer and Bongard (2006) where morphological computation refers to "certain processes are performed by the body that otherwise would have to be performed by the brain" or with the experiments conducted in Hauser et al. (2011). Moreover, it constitutes a fundamental motivation to embodiment which states that steps toward adaptive intelligence do not only come from the controller complexity but also from the interactions with the body and the environment. Broader analysis about the quantification of morphological computation as well as the trade-offs with informational computation include Polani (2011), Zahedi and Ay (2013), Haeufle et al. (2014), Hoffmann and Müller (2014), and Ghazi-Zahedi et al. (2016).

Illustrative applications of morphological computation and embodiment for locomotion are numerous in biology and robotics. For instance, Dickinson et al. (2000) provide an analysis of how animals succeed in efficient locomotion using their muscles not solely as motors but to provide multiple functions varying from brakes to springs and struts. The *passive walker* in McGeer (1990) constitutes an extreme example of an engineered robot exploiting the same concept. This two-legged physical structure is able to walk down a slope in a very natural way without any actuation. This work has been extended later in Collins et al. (2005) to robots with low-power actuators. They show a walking pattern that looks natural and energy efficient compared to traditional stiff controlled robots. In other fields of robotics, we can also cite the works of Iida and Pfeifer (2006) or Degrave et al. (2015), in which dynamical properties of compliant quadruped robots are used to provide low power consumption, to reduce controller computational complexity, and to observe natural transitions between gaits. Examples that clearly benefit from compliance to improve moving can also be found, among others, in Cham et al. (2004) which focuses on hexapod locomotion.

A practical implementation of morphological computation can be inspired from Reservoir Computing (RC). RC denotes a computational framework that enables the approximation of a broad range of dynamical behaviors for which a precise model is not available. RC originates from the domain of recurrent neural networks and is mainly based on the theories of Echo State Networks (ESN) and Liquid State Machines (LSM) as outlined in Lukoševičius and Jaeger (2009). At the time of their introduction, they offered a solution to the training of Recurrent Neural Networks (RNN), which was still considered difficult. They avoided having to train feedback connections and the problems with bifurcations this brings, i.e., the discontinuities in the network outputs observed for some points in the parameter space, by training only the synaptic connections of the readout nodes. The core architecture consists of a randomly connected RNN, the *reservoir*, for which the synaptic weights are sampled from some distribution and then globally rescaled to tune the dynamical regime close to the edge of chaos. RC also resulted in different robotics applications as learning of inverse kinematics of an iCub robot arm from a neural reservoir in Reinhart and Steil (2009) or the creation Central Pattern Generators (CPG) to control human movements in Wyffels et al. (2014) and hexapod locomotion in Dasgupta et al. (2015).

As the reservoir network is constituted of randomly connected non-linear entities, many physical dynamical systems presenting sufficiently complex transformations of their inputs provide similar dynamical properties and can be used as reservoirs. For instance, it has been demonstrated in Hauser et al. (2012) that generic types of physical bodies such as Mass–Spring–Damper (MSD) networks are able to approximate any given time-invariant filter with fading memory and generate adaptive periodic patterns autonomously when a feedback loop is added. This extension of RC is generally referred to as Physical Reservoir Computing (PRC). The expensive step of computing the reservoir transformation is now outsourced to a physical system's natural dynamics. This means that the neuron states will not be explicitly updated digitally anymore, but this computation is transferred to the body's dynamical evolution. Only the readout layer only needs to be engineered, most often using digital computing.

The main advantage of PRC lies in the parallelism of the computations in the physical reservoir and, in the case of robotic locomotion, in the fact that the transformations computed by the robot body are a natural result of the gait. However, PRC is essentially a supervised machine learning technique. By contrast, robotic control is intrinsically a reinforcement learning problem, in which the optimal desired actuator signals are not known *a priori*. In addition, successful reservoir implementations often require the observation of the reservoir state at many different points. In robotics, this implies that for each observation point a sensor needs to be installed.

Numerous applications of PRC have been demonstrated in the past decade. In robotics, highly compliant robot models have been addressed for example to MSD networks in Hauser et al. (2011) (simulation only), tensegrity structures in Caluwaerts et al. (2014) or a real soft robotic platform inspired by an octopus arm in Nakajima et al. (2014, 2015). Closed-loop control of quadruped robot exploiting a spine made with soft material as a reservoir can

be found in Zhao et al. (2013). Simulations or implementations of PRC outside robotics include water ripples in Fernando and Sojakka (2003), electro-optical devices in Larger et al. (2012), or pure optical devices in Brunner et al. (2013) and Vandoorne et al. (2014).

This paper presents two main research objectives. First, we design a small scalable simulation setup to provide empirical compliance studies on the locomotion of MSD networks. To our knowledge, such an analysis does not yet exist and should help to evaluate the potential of compliance for locomotion in terms of robustness, efficiency, and stability. To this end, three main experiments are conducted. The first experiment gives an overview about how increasing the number of nodes in a MSD network leads to more stable locomotion. The second experiment provides an analysis on the optimal frequency range for the setup, and the third experiment explores the maximal reachable speeds for different driving powers and underlines the limitations of the design to get high performance. In the second part, we analyze the computational capacity of a MSD body to generate motor control signals and integrate them as a regulation feedback to a forward controller.

2. OPEN-LOOP CONTROL

2.1. Materials and Methods

To run our experiments and analysis, we designed a MSD network simulator directly implementing mechanical equations using *Python* and *Numpy*.[1] These networks, inspired by Hermans et al. (2014) and Caluwaerts et al. (2013), consist of a set of nodes with mass, connected by spring–damper links which are all actuated separately. The simulation can be performed either in 2D or 3D.

[1]https://github.com/Gabs48/SpringMassNetworks.

2.1.1. Mass Spring Networks

The MSD morphology is presented in **Figure 1**. Each of the N nodes, except those at the end or beginning, is sparsely connected to its closer neighbors by C connections. The total number of springs in the network S can be easily deduced using geometry:

$$S = \left(N - 1 - \frac{C/2 - 1}{2}\right) \cdot \frac{C}{2}. \tag{1}$$

Each node $i \in \{1, \ldots, N\}$ is represented by its mass m_i, whereas the passive parameters for each connection are the spring stiffness k_j and the damper coefficients d_j for $j \in \{1, \ldots, S\}$. In this paper, the notion of compliance will be used. It is defined as the inverse of the stiffness $1/k_j$. If not specified, the default values used in the following experiments are $N = 20$, $C = 3$, $m_i = 1$ kg, $k_j = 100$ N/m, and $d_j = 10$ Ns/m.

In our model, the acceleration, speed, and position of each mass are updated using the force vector $\mathbf{F}_i$ which combines the gravity force, the spring force, the damping force, and the air friction force:

$$\mathbf{F}_i = \mathbf{F}_i^s + \mathbf{F}_i^d + \mathbf{F}_i^g + \mathbf{F}_i^a, \tag{2}$$

where

- $\mathbf{F}_i^s$ is the spring force vector applied on the node i and equals the sum of the $j \in \{1, \ldots, C\}$ connected non-linear springs forces for which the equations can be found in Palm (1999):

$$\mathbf{F}_j^s = -k_j \cdot \frac{\mathbf{l}_j}{l_j} \cdot \left((l_j - l_{j,0}) + \frac{\alpha}{l_{j,0}^2} \cdot (l_j - l_{j,0})^3\right). \tag{3}$$

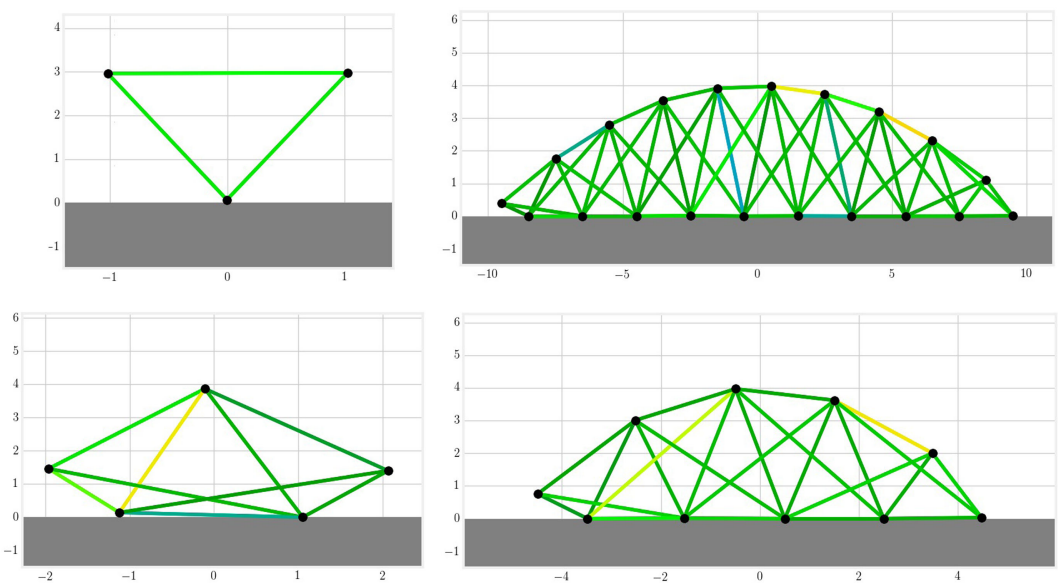


FIGURE 1 | The MSD structures are build automatically with a simple morphology that takes the number of nodes and connections as an input. On the figure above individuals with three, five, ten, and twenty nodes are drawn in a 2D space. Each black circle represents a mass and each line a set of spring and damper in parallel. The colors indicate the current amplitude of actuation.

In this equation, $\mathbf{l}_j$ represents the spring length vector and $l_{j,0}$ its reference length. The variable α is a non-linearity coefficient which will induce a saturation of the spring force for large extension lengths. It also takes inspiration from the work of Hauser et al. (2011) which demonstrates the importance of these non-linearities from a computational consideration.

- $\mathbf{F}_i^d$ is the damper force vector applied on the node i and equals the sum of the $j \in \{1, \ldots, C\}$ connected dampers:

$$\mathbf{F}_j^d = d_j \,.\, \frac{\mathbf{v}_j}{v_j} \,.\, (v_j - v_{j,0}), \tag{4}$$

where $\mathbf{v}_j$ is the vector of extension speed.

- $\mathbf{F}_i^g$ is the gravity force vector:

$$\mathbf{F}_j^g = g \,.\, m_i \,.\, \mathbf{x}_i^y, \tag{5}$$

where g is the gravity constant and equals 9.81 m/s^2.

- $\mathbf{F}_i^a$ represents the drag force induced by air friction. It is assumed proportional to the speed:

$$\mathbf{F}_j^s = -a \,.\, \mathbf{v}_i, \tag{6}$$

where a is the coefficient of air friction and equals 0.1 Ns/m. It has been included to avoid unrealistic models with very high speed. However, these did not occur in the experiments presented in this paper.

The ground reactions are modeled by setting the vertical velocity to zero and the horizontal friction coefficient to infinite. The masses perfectly stick to the ground as soon as they touch it. This is a hard constraint that can impact the nature and the performance of locomotion. However, it simplifies the study of the body influence by assessing perfect friction conditions in every simulation.

2.1.2. Control

To actuate the spring using a control signal, we modulate the reference lengths of the springs $l_{j,0}$. In the simplest and default case, this will be represented by a simple sinusoidal signal like in Hermans et al. (2014):

$$l_j(t) = l_{j,0} \,.\, (1 + \; a_j \,.\, \sin(\omega_j \,.\, t + \phi_j)). \tag{7}$$

It induces a set of tunable parameters $l_{j,0}$, ω^j, ϕ^j for each spring in the simulation.

2.1.3. Physics Solver

The simulation time is discretized using K time steps t_k, and equations are solved numerically using the Verlet algorithm as described in Thijssen (2007). The Verlet integrator leads to more accurate trajectories, especially for periodic oscillations where energy is rigorously conserved due to the time reversibility of this operator. For non-periodic trajectories, one can prove that due to symplecticity, the energy does not drift away and errors remain bounded as demonstrated in Yoshida (1990). Although it is more accurate, the Fourth-Order Runge–Kutta integrator requires four force evaluations per update step and is not symplectic. In our implementation, the update equations are slightly changed in order to take the effect of the ground reactions into account.

2.1.4. Loss Function

The goal is to develop a generic approach to obtain robust locomotion in open loop without prior knowledge about the body dynamics. In the case of simulated MSD networks, this implies the optimization of controller and morphology parameters for each specific network. This can be formulated as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}). \tag{8}$$

where the score function $f(\boldsymbol{\theta})$ and the optimization algorithms are detailed below. Typically, the optimized parameters $\boldsymbol{\theta}$ of the MSD network are the controller amplitude a_j between 0 and 0.25, its frequency between 0 and 10 Hz, its phase ϕ_j between 0 and 2π, and the spring stiffness k_j between 0 and 100 N/m. To synchronize the actuators together and impose the fundamental frequency, the angular speeds ω_j are fixed to the same value. In the case of a MSD with $N = 20$ nodes connected to their six closest neighbors ($C = 6$), this represents a total number of springs $S = 54$ (see equation (1)) and therefore 163 parameters to optimize. Locomotion characterization and evaluation is performed through two performance metrics:

- **Distance traveled** D: the difference between the centers of mass at the end and at the beginning of the simulation. This function is determined by the full locomotion sequence along the simulation.
- **Power efficiency** P: the power dissipation of the non-linear spring actuators can be approximated according to Palm (1999):

$$P = \sum_j k_j \,.\, \frac{a_j^2 \;\; l_{j,0}^2 \left(1 + \alpha^2 a_j^2\right)}{4\pi}, \tag{9}$$

in which a_j are the relative amplitudes, α is the spring non-linearity factor, and $l_{j,0}$ are the reference lengths of the springs.

Using the ratio of distance to power is unsatisfactory, as this could result in robots that consume very little power because they barely locomote. Instead, we will use the following power efficiency score displayed in **Figure 2**:

$$f(\theta) = \tanh\left(\frac{D(\boldsymbol{\theta})}{D_{ref}}\right) \,.\, \tanh\left(\frac{P_{ref}}{P(\boldsymbol{\theta})}\right), \tag{10}$$

in which D_{ref} and P_{ref} are reference values allowing to normalize and homogenize the scores. As it is desirable to operate in the linear regime and avoid saturation of the score, we set them to 3,600 and 100, respectively, following the statistics of the observed distance and power values.

2.1.5. Optimization

The aim is to develop an optimization approach that can be applied to highly compliant physical robots, without any need for an analytical model for the body dynamics. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) as formulated in Hansen (2006) has been selected from a pool of different optimization methods. Indeed, it fits very well for browsing non-convex parameter landscapes with a lot of local minima. In addition, it presents a good convergence speed and requires very few initialization parameters:

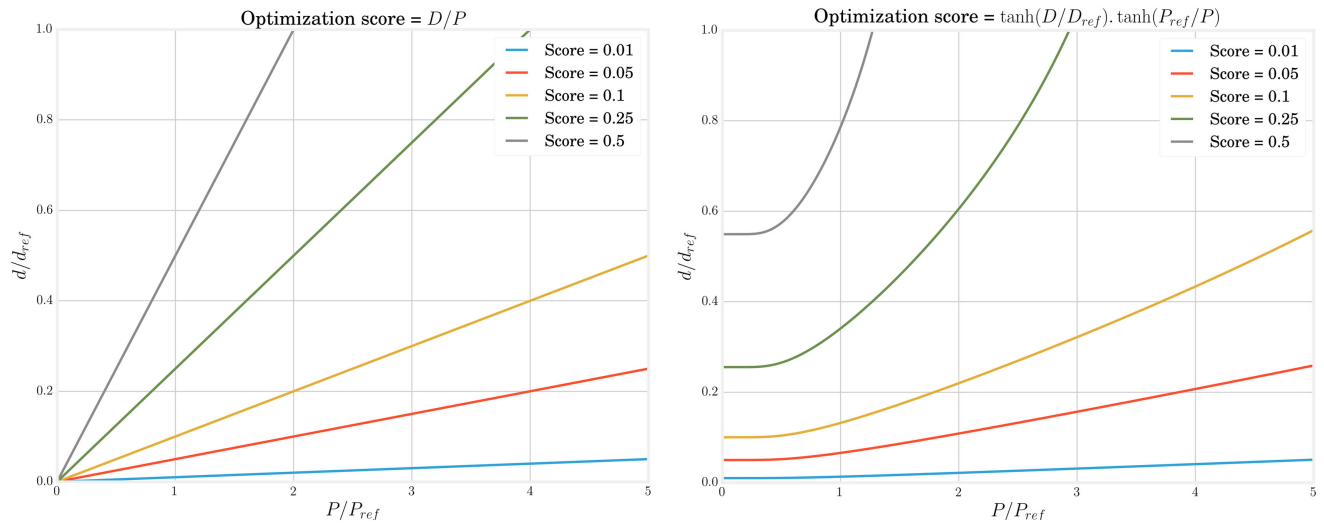


FIGURE 2 | The efficiency score to quantify locomotion quality increases with traveled distance and decreases with dissipated power. However, taking directly the ratio between both metrics (left) could lead to optima close to the origin, i.e., where the body barely moves. Using a hyperbolic tangent (left) solves this problem for small powers but requires to select P_{ref} and D_{ref} carefully to avoid a saturation due to the measure itself.

- The initial parameter distribution is a Gaussian centered in 0.5 and with a SD of 0.2 after normalization of all parameters.
- The population size, the step size, and covariance matrix parameter are set to their default values as recommended in Hansen (2006).
- The iteration number is set to ensure convergence, which will be assessed qualitatively by observing saturation in the score evolution.

2.2. Results

In this section, we assess the influence of the MSD network size and compliance on the best locomotion speed found, on the power consumption, and on the noise robustness in our specific example. Three different experiments are described in this context. In the first one, we increase the number of mass nodes in the network to determine its influence on locomotion efficiency. The second investigates how optimal compliance is related to the morphology parameters and the locomotion frequency. Finally, we discuss how the optimized gait changes when the driving power is constrained.

2.2.1. Morphology Analysis

The choices made during the design of a system can contribute to more efficient and robust behaviors for solving sophisticated tasks. In the case of the MSD setup, we can intuitively assume that increasing the number of nodes will broaden the space of available trajectories, therefore increasing the number of optima at the expense of a longer learning process. It is interesting to note that such a tuning does not necessarily imply an increase of complexity, in the sense of the definition presented in Lungarella and Sporns (2006).

To verify this assumption, we have optimized open-loop locomotion controllers for networks with increasing number of nodes and springs. As mentioned before, this optimization consists in tuning the actuators' amplitudes and phases, the spring constants, and the global frequency of locomotion. Other parameters of the MSD network are set to the same value for all bodies, except for the nodes mass. This is normalized by the number of nodes, such that the total mass of the MSD network (20 kg) remains the same in every simulation and the power levels required for locomotion can be compared.

In order to converge toward stable gaits, we add random acceleration impulses during the simulation. Their value is centered around 10% of the mean absolute acceleration and applied on random nodes 5% of the time. In the CMA-ES algorithm, the number of iterations is tuned specifically for each optimization to ensure convergence, since optimizing small structures will converge faster than larger ones. From each optimization run, the best individual is retained. Each optimization is repeated five times in order to average the results and obtain an estimate of the variability of our observations.

Figure 3 shows the evolution of the averaged best individual score for increasing body size in blue. From left to right, we observe that the scores rapidly decrease for structures of up to five nodes before steadily increasing again. However, the good results in the first part of the curve should be interpreted carefully, taking their robustness to noise into account. To assess this property, we also represented the scores obtained for simulation using the same parameters but without noise on the same figure. We notice that this difference decreases with the number of nodes. This shows that structures with more nodes are more robust to the noise added during the simulation. The evolution of the CMA-ES algorithm represented in **Figure 4** also supports this hypothesis. It shows that the optima of the structures with a small number of nodes are found randomly instead of through convergence of the algorithm, unlike the structures with more nodes. High scores originate from these bodies' reduced stability. This makes them very sensitive to impulse noise as small disturbances can either make them fall over or push them forward. They can therefore rightly be regarded as outliers.

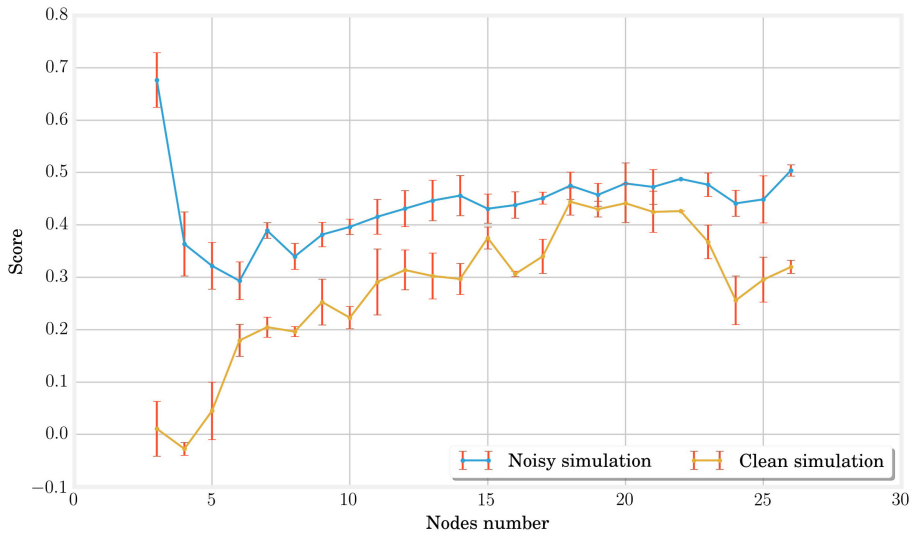


FIGURE 3 | In this graph, the best individuals for CMA-ES optimizations of different MSD networks are plotted in blue. Other simulations without noise are then performed on the same individuals with the same parameters in order to identify the outliers due to noise and qualify the stability of locomotion. The low performance of 3, 4, and 5 nodes MSD structures indicate unstable gaits. For larger structures, the score first increases with the number of nodes but saturates rapidly for networks of more than twenty nodes.

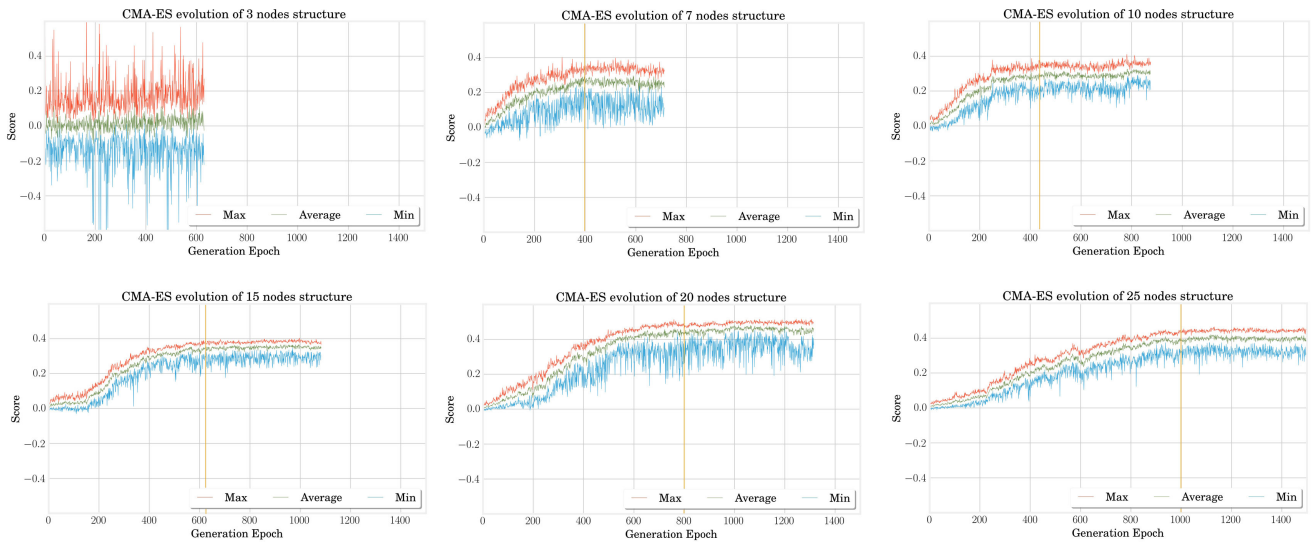


FIGURE 4 | In this graph representing the CMA-ES evolution for different structures, we can qualitatively observe that the convergence time, whose estimation is given by the vertical yellow lines, increases with the MSD network size. This is expected as the problem becomes more complex and the number of optimized parameters is higher as well. When the structure is too simple such as the three nodes (one in the upper left corner), the problem cannot converge and the best results encountered during the exploration are mainly due to the random noise added in simulation.

It is finally interesting to note that the score increases gradually starting from six nodes but quickly saturates. A more detailed analysis in **Figure 5** shows that this is due to better performances in terms of traveled distance, whereas dissipated powers are very similar. However, note that this is achieved at the expense of a longer learning process, as pointed out by the number of epochs represented on the graphs X axis of **Figure 4**.

In conclusion, this experiment points out that increasing the number of nodes and springs in the MSD networks leads to an increased robustness to external noise and better speed performances.

2.2.2. Frequency Range Analysis

In this second set of experiments, we try to evaluate the nature of a link between robot compliance, which is defined by $1/k$, the inverse of spring stiffness, and the optimum efficiency of locomotion.

The resonance frequency of a MSD system with one unique node and spring equals $\sqrt{k/m}$. It ranges from 0.6 to 1.8 Hz for

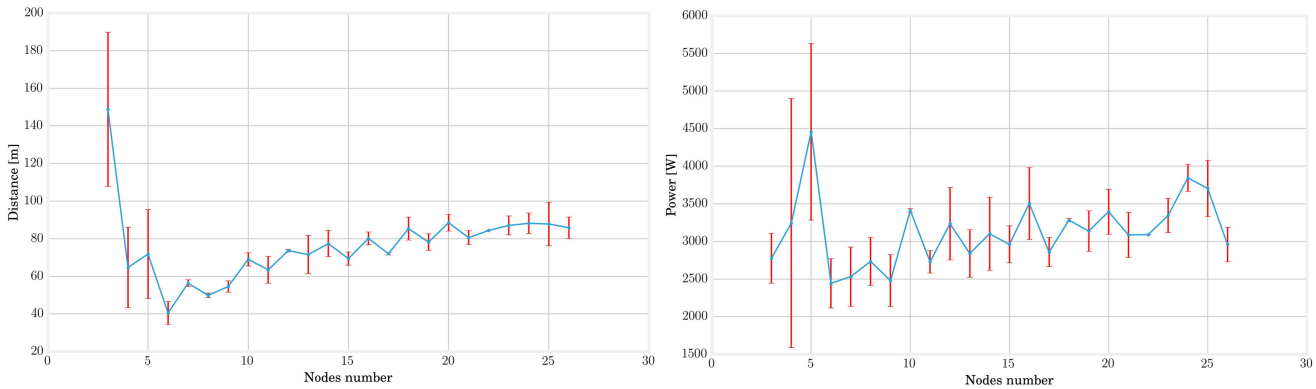


FIGURE 5 | By displaying separately the distance and power components in the loss function of the CMA-ES optimization, we can acknowledge that the observed variation for different nodes number are mainly due to the distance. As expected by the normalization factor, the driving power remains sensibly equals for each structure.

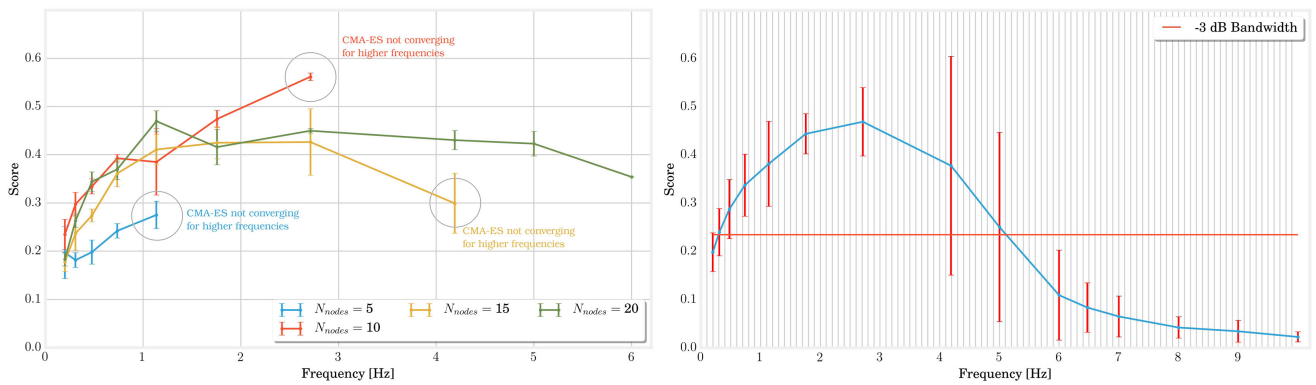


FIGURE 6 | The two graphs in this figure show the evolution of score performances with the fundamental frequency of locomotion. On the left side, we separated the result according to the number of nodes in the structure. By applying a simple CMA-ES convergence assertion, we notice that the operating range can extend to higher frequency for larger structures. On the right side, we combined all structures to determine a −3 dB bandwidth ranging from 0.3 to 5.2 Hz.

the m_i, k_j, and d_j values that we are using in our setup (as a reminder, m_i varies with the number of nodes). There is therefore a bijective function between compliance and resonance frequency. By extension, we can formulate the hypothesis that the resonance frequency of a MSD structure is directly coupled to its compliance. Since they are composed of several masses and springs, we can expect that the bandwidth of the resonance peak will broaden but still appear at the same frequency.

With this assumption, the study of correlation between compliance and locomotion efficiency can be reformulated to focus on the link between actuation frequency and efficiency. Previous work such as Buchli et al. (2006) for robotic systems or McMahon and Cheng (1990) for models of mammalian gaits highlighted such a link: self-learning systems with different morphology properties tuned their actuation frequency to the resonance of the structure to reach optimal performance in locomotion.

In this setup, MSD structures with 5, 10, 15, and 20 nodes were optimized several times by fixing their global frequency to values between 0 and 10 Hz. In **Figure 6** (on the left), we have represented the results for different numbers of nodes. Each optimization corresponds then to a point on the graph. For some of those points, however, the optimization process was not able to converge to a gait that is both stable (whose pattern does not change in time) and robust (allowing external noisy perturbations). In the graph, this failure is particularly true for structures with few number of nodes simulated at high frequencies. A first empirical conclusion is that the robustness of MSD networks at high frequencies increases with the number of nodes. This represents an additional advantage concerning the size of the system along with the discussion from previous section. In terms of score, however, there is no significant difference between the topologies, and their optimal bandwidths are very similar. The optimal scores are a little lower only for the 5 nodes structures, which corroborates the results from the previous experiment. To get a more accurate measure of the bandwidth, it may even be interesting to combine all the results as they possess very similar resonance frequency. This is presented on the right side of **Figure 6** where we can observe that the structure is optimal over a 3 dB bandwidth in the range [0.3; 5.2 Hz]. The large confidence intervals around 4 and 5 Hz are again explained by the absence of convergence for the structures with a low number of nodes.

To sum up, this experiment provides guidance on the choice of compliance values in the design of a MSD network for locomotion. Choosing the global compliance to optimize a robot of a given mass is conditioned by the frequency at which we plan to actuate the robot. Also, structures with more nodes tolerate a broader range of frequencies while keeping stability.

2.2.3. Performance Limits with Constrained Power

So far, we have used a loss function that combines performance with respect to both traveled distance and energy consumption. However, it may be beneficial to analyze them separately in order to understand the limiting factors and to observe what can be the best compromise between them. The following experiment also allows us to qualitatively characterize the gaits of our structures and to observe possible transitions between different modes.

For this purpose, several optimizations have been performed by constraining the power and forcing their saturation to different values. In this way, one can expect to observe what is the maximum distance an individual can reach for a given power. Since we work outside the boundaries of the desirable operating range of the original cost function, we have now increased the reference value D_{ref} to 1,000 m in order to avoid a saturation effect due to the cost function itself.

Figure 7 shows the evolution of the optimal speed as a function of a constrained power budget. The best individuals are in the upper left corner. As might be expected from the conclusions of the previous section, the 3-Hz frequency gives the best results. Concerning the shape of the curve, we can see that the maximum speed increases almost linearly until 15,000 W and starts saturating beyond that.

This saturation highlights the limits of our model. It helps to understand which factors such as the spring saturation, the ground friction, the air drag, or the geometry play a larger role in performance compared to the driving power. It also situates the previous experiments in the non-saturating range, which helps to appreciate their significance better.

Finally, for very low power, an energy increase does not seem to add any improvement and even the opposite happens for frequencies 1 and 4 Hz.

A visual observation of the locomotion is useful to give more insights about the possible gait transitions on this curve. For this purpose, we have produced a series of videos renditions of individual simulations provided in Supplementary Material. A qualitative analysis of those video shows that the most common gaits consist of displacing the whole structure along a wave movement (each node touches the ground a little after the previous one) or locomoting in two steps (the body touches the ground two times per period with a phase difference of 180°). Concerning the high power saturation, a video was made for each point of the 3-Hz curve. It shows that the most energy-consuming individuals present spring extension close to their saturation, which causes a loss of stability of the locomotion. In the same way, videos were produced in the low-power domain for the points on the 4-Hz curve. For the lowest power, a good two-step alternation of contacts between the body and the ground is observed, whereas the phase shifts between the different contacts with the ground are much less synchronous for the following individuals. The same results have been established each of the 5 times the experiment has been conducted. Progressively with increasing power, a two-step approach with robot–ground contacts phase-shifted by 180°comes up again.

In short, we can stress the role of the body design in locomotion through two principal observations: first, a saturation of the spring leading to a degraded operation in high power; second, a qualitative influence of the optimal gait on the performances for a given morphology and power consumption.

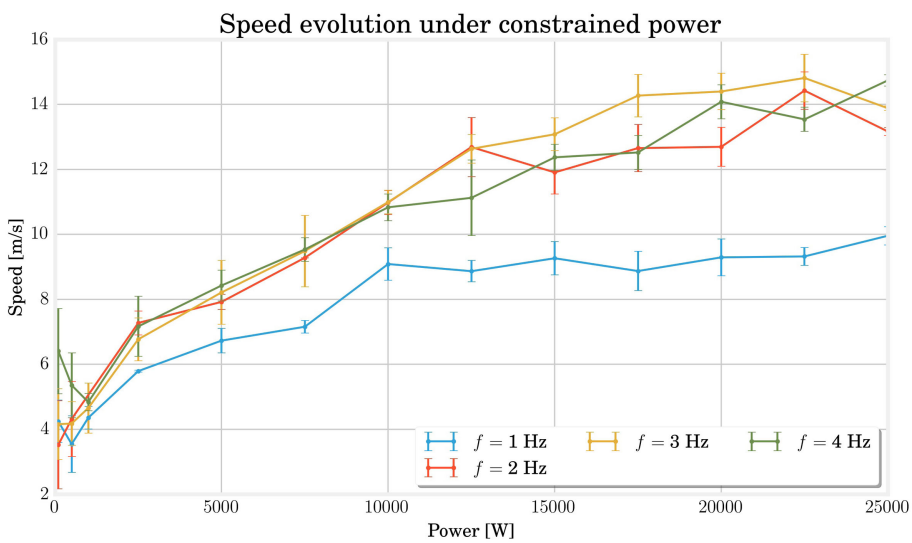


FIGURE 7 | This curves shows the evolution of the maximal speed reached for different constrained power at different frequencies. The saturation effect for high powers demonstrates the physical limits of the structures, while the slight decrease for very low powers indicates a change in locomotion gaits associated with a different efficiency.

3. CLOSED-LOOP CONTROL

3.1. Materials and Methods

The closed-loop control of the MSD network is performed through physical reservoir computing. In this setup, our goal is to reproduce the control signals at a time step t_k using the physical states of the network at times $t_{k-1-n} \ldots t_{k-1}$ only. This is performed by training the weights of a linear combination (the readout).

3.1.1. Setup

The closed-loop system is composed of different elements represented at **Figure 8**:

- The MSD structure that can be perceived as a physical reservoir because of its dynamics and high complexity. For each time step t_k, the system's current state is evaluated using the acceleration vectors $\mathbf{a}[k]$, $\mathbf{a}[k-1]$, and $\mathbf{a}[k-2]$, which comprise both X and Y components of all the nodes. The choice of acceleration, instead of, e.g., speed, is based on the work of Caluwaerts et al. (2013). Trials using integrated quantities such as position or speed instead have also been evaluated but added a drifting error during training. Also, based on the same work, we choose a buffer size of 3 time steps. In our experiments, smaller values led to deteriorated results but larger ones did not show any significant improvements.
- A sensor filter, whose principal role is to model the physical limitations in acceleration sensing. It is composed of an amplitude threshold followed by a low-pass filter. The cutoff frequency at 6 Hz has been chosen very low to eliminate possible oscillations due to our numerical integration method while keeping the locomotion fundamental frequency and its first-order harmonics. At the output of the filter, a vector $\mathbf{x}[k]$ is sent to the next element.

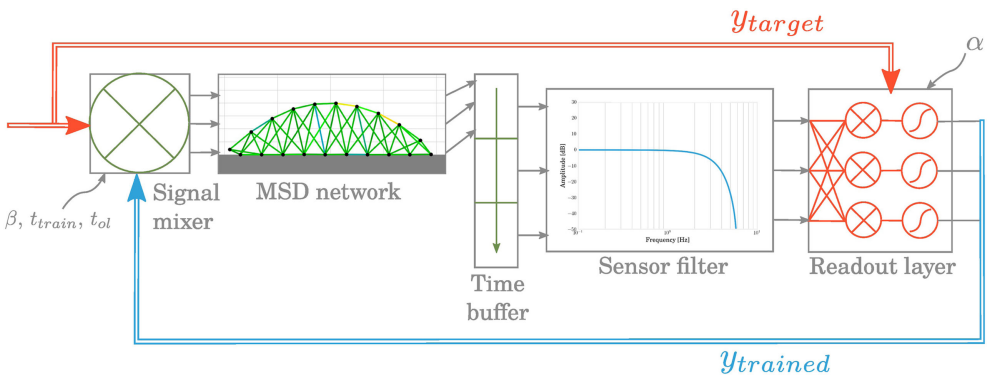


FIGURE 8 | The principal components in the closed-loop learning pipeline consist in a readout layer whose weight matrix is trained at each time step and a signal mixer that gradually integrates the feedback in the actuation signal.

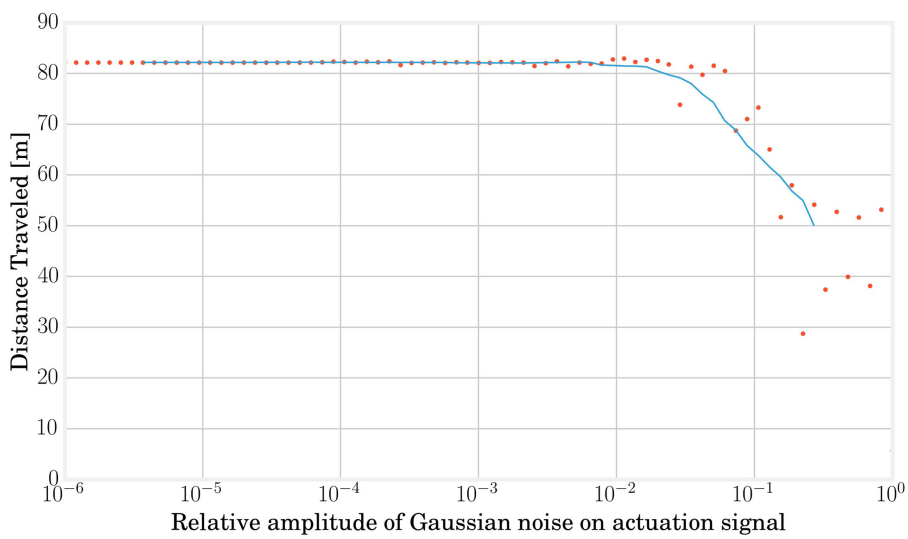


FIGURE 9 | Adding some noise on the actuation signal in open loop can give a hint about the maximum error we can accept on the trained signal in closed loop without damaging the locomotion stability and is helpful to determine the regularization parameter. In this graph, each red point represents a simulation and the blue line shows the average evolution. Performances start decreasing from a relative Gaussian noise of 0.01.

- A readout layer, which computes the actuation signals for the next time step based on the current and previous states of the MSD:

$$\mathbf{y}[k+1] = \mathbf{W}_{\text{out}}^T \,.\, \mathbf{x}[k]. \tag{11}$$

To learn the weights of the output matrix $\mathbf{W}_{out}$, we use the FORCE learning method as in Sussillo and Abbott (2009), whose equations are the following:

$$\mathbf{e}[k+1] = f_{\text{sigmoid}}\left(\mathbf{W}_{\text{out}}^T[k] \,.\, \mathbf{x}[k+1]\right) - \mathbf{y}_{\text{target}}[k+1] \tag{12}$$

$$\mathbf{P}[k+1] = \mathbf{P}[k] - \frac{\mathbf{P}[k] \,.\, \mathbf{x}[k+1] \,.\, \mathbf{x}^T[k+1] \,.\, \mathbf{P}[k]}{1 + \mathbf{x}[k+1]^T \,.\, \mathbf{P}[k] \,.\, \mathbf{x}[k+1]} \tag{13}$$

$$\mathbf{W}_{\text{out}}[k+1] = \mathbf{W}_{\text{out}}[k] - \mathbf{P}[k+1] \,.\, \mathbf{x}[k] \,.\, \mathbf{e}[k+1]^T \tag{14}$$

$$\mathbf{y}_{\text{trained}}[k+1] = f_{\text{sigmoid}}\left(\mathbf{W}_{\text{out}}^T[k+1] \,.\, \mathbf{x}[k+1]\right), \tag{15}$$

where the estimate of the inverse of the correlation matrix $\mathbf{P}$ is initialized to $\mathbf{I}/\alpha$. The sigmoid function added ahead of the readout adds non-linearity in the control signal by saturating for too high values.

- A signal mixer to avoid a brutal transition from open-loop to closed-loop control. Its role is to incorporate gradually the readout output contribution to the target signal. It is defined by three parameters: the open-loop time t_{ol} when the MSD network is run in open-loop mode only; the training time t_{train} in which the contribution of closed-loop signal increases linearly and the percentage β of feedback in the full control signal before switching to closed-loop mode only.

3.1.2. Parameter Tuning

The α parameter of the FORCE learning algorithm plays the role of a regularization variable in the process of learning the $\mathbf{W}_{\text{out}}$ matrix. It must be selected in order to avoid an overfitting that would reduce robustness to undesired forces on the MSD structure but also to ensure a trained signal sufficiently close to the target. This is a major issue since a signal $\mathbf{y}_{\text{trained}}$ with too much noise can easily cause a divergence in the locomotion limit cycle. Tests on signal noise robustness as presented in **Figure 9** allowed to estimate a value of $\alpha = 0.01$ as a good compromise.

The open-loop training and running times can be estimated by analyzing the convergence error of the FORCE algorithm (see **Figure 10**) and are fixed to 12 s of open-loop learning followed by 38 s where the feedback signal is gradually added to the target signal to reach a value of $\beta = 95\%$ before closing the loop. Stopping the training before the actuation signal reaches 100% of feedback avoids convergence to a steady state as discussed in Caluwaerts et al. (2013).

3.2. Results

In order to determine the contribution of the system size in the process of learning its own locomotion gaits, we simulated MSD networks with different numbers of nodes and evaluated the distances traveled over the last 10 s in closed loop. The same simulation was carried out in open loop to provide a reference. The results of these simulations are presented in **Figure 11**. At first sight, it appears that the learning algorithm with its configuration can achieve performances of the same order of magnitude in open and closed loops for the structures between three and twenty-six nodes analyzed in this simulation. However, it is worth noting that MSD with less than 6 nodes already provided non-significant results in open loop.

Alternatively, the study of limit cycles gives an indication of the stability of closed-loop control. In **Figure 12**, we represented the temporal evolution of the internal states $\mathbf{x}_k$ in a 2 coordinate space obtained by PCA. Larger structures lead to smoother limit cycles in closed loop. The limit cycles even diverge from their basin of attraction for very small MSD networks. A simple interpretation is that more nodes lead to more cycles in the physical reservoir, which provides more robust trajectories in the principal

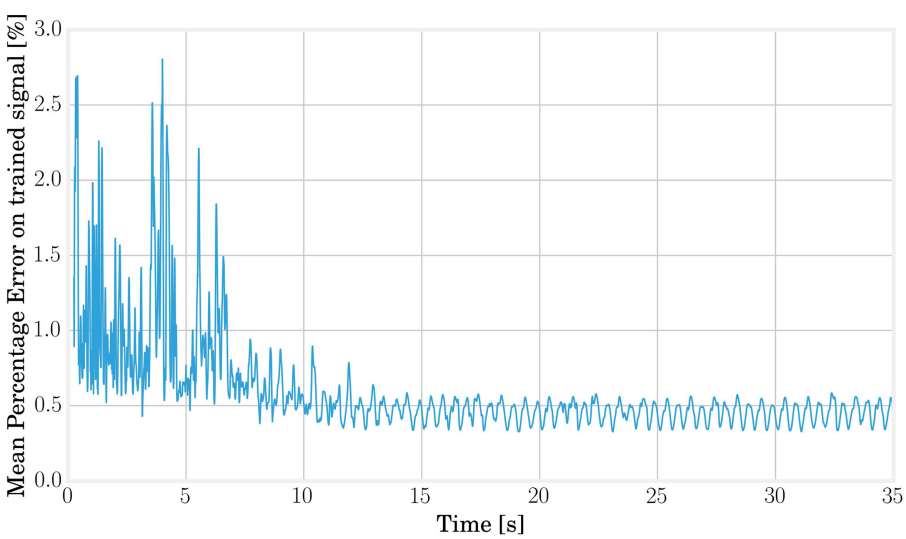


FIGURE 10 | The learning error can be used to estimate the required training time and the maximal rate at which the loop should be closed. From this graph, we can deduce that 12 s of simulation is sufficient to consider the convergence of the readout weights.

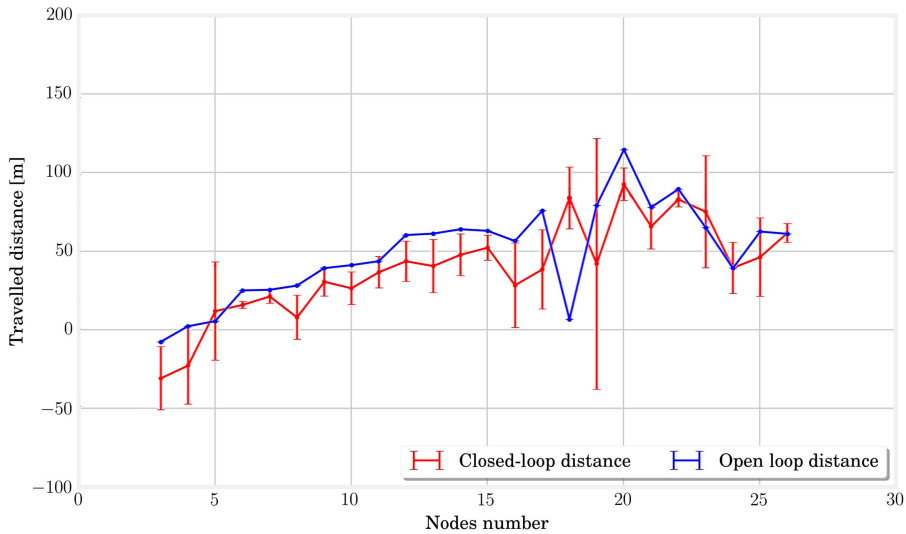


FIGURE 11 | In this picture, we plot the traveled distances for the last 10 s of simulation in open loop in blue and closed loop in red. There is no crucial difference between the two curves, which seems to indicate that the performances in closed loop are close to the one in open loop for all structures.

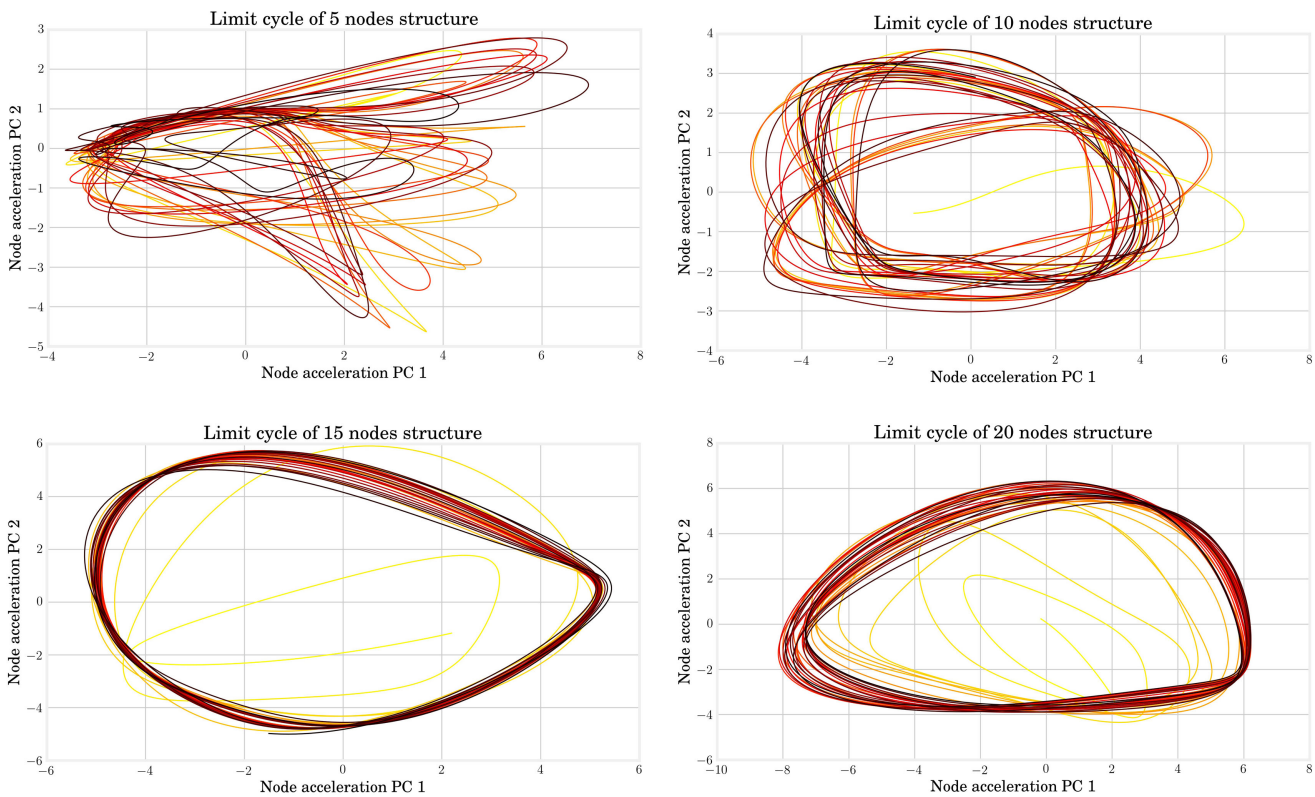


FIGURE 12 | From left to right and top to bottom, the limit cycle during FORCE training are represented for structures with, respectively, 5, 10, 15, and 20 nodes. The color ranges from yellow for the initial seconds of the simulation (which point out the transient effect) to black in the end of the simulation. When the node number is too low, the trained signal can diverge from its basin of attraction.

components reference. This hypothesis is corroborated by analyzing the quality of the generated actuation signals. This can be quantified by plotting the Normalized Root Mean Square Error, as shown in **Figure 13**, which decreases with the number of nodes.

In conclusion, the morphology of MSD bodies has the capability to compute at each time step the next value on the parametric trajectories found in open-loop optimization with a sufficient accuracy for locomotion task. The computation and memory that

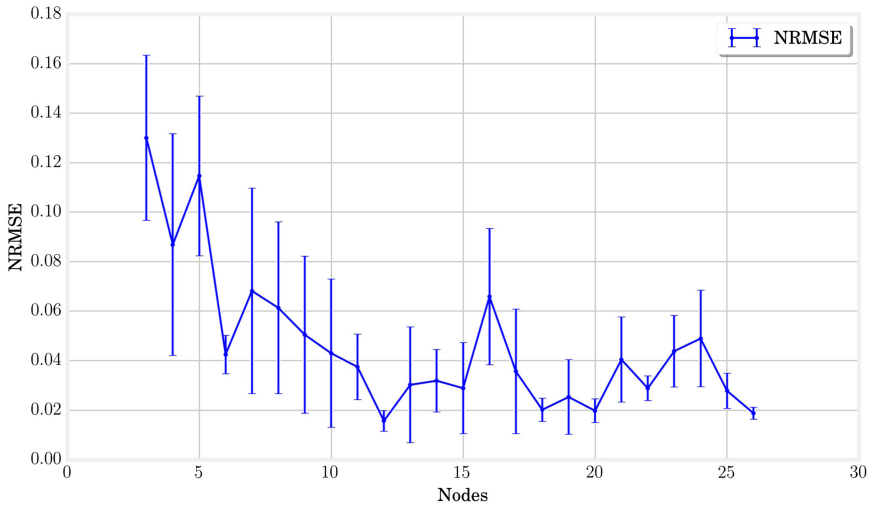


FIGURE 13 | The normalized root mean square error between the trained and target signals is represented for each node number. It indicates that the learning tends to produce better more accurate results with an increasing number of nodes.

was previously embedded in an external controller can be fully distributed in the structure and the readout layer. The size and number of sensor measurements on the structure have a positive effect on the accuracy and stability of the feedback signal.

4. DISCUSSION

In this article, we have tried to study systematically the influence of high-level design choices on the performance of MSD systems. Because of their analytical simplicity and their modularity, those body structures seem indeed adapted to conduct studies on the morphological contribution in the process of locomotion control. This research was divided into two main parts. On the one hand, an open-loop study focused on the benefits of body size to efficiency and stability. A similar analysis was also performed on locomotion frequency and helped to draw conclusions about how compliance can be chosen to increase optimal performance. On the other hand, we aimed at demonstrating the key role of morphology to generate control signals in a completely closed operation mode.

The different trials undertaken in open loop indicated the importance of the structure size to ensure optimal performance in terms of distance traveled and gait stability. Concerning compliance, its relation to the fundamental frequency of locomotion was used to demonstrate a link with the efficiency and to provide a specific suggestion in the design of optimal MSD systems. It has been noted that the frequency response of the different MSD networks shows a bell shape, displaying a degraded score for too high or too low frequencies and that the stability at high frequencies is better for larger structures. Finally, the behavior at different power values has highlighted the limits of the design in reaching high speeds, and a qualitative study has shown the effect of the gait evolution in this phenomenon.

In closed loop, the ability of MSD structures to generate their control signals on the basis of a single, fully connected layer of neurons has been attested. An increase in the size or the number of sensor signals induced a positive influence with regard to the limit cycle stability and the accuracy of the signals generated by the algorithm.

In future work, the main improvement should focus on increasing noise robustness and adaptability on different terrains and facing various obstacles. In this way, the goal is to provide a simple and generic locomotion primitive for complex structures, which learns how to perform actuator synchronization by harvesting the mechanical feedback while taking higher level control inputs such as the locomotion frequency. On the other hand, it would be interesting to generalize our conclusions to both real robots and biologically inspired dynamical models such as quadrupeds and bipeds.

AUTHOR CONTRIBUTIONS

The experiments were conceived by GU, BC, FW, JDegrave, and JDambre and designed by GU and BC. The data were analyzed by GU with help of FW, JDegrave, and JDambre. The manuscript was mostly written by GU, with comments and corrections from FW and JDambre.

FUNDING

The research leading to these results has received funding from the European Unions Horizon 2020 Research and Innovation Programme under Grant Agreement No. 720270 (HBP SGA1).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at http://journal.frontiersin.org/article/10.3389/fnbot.2017.00016/full#supplementary-material.

VIDEO S1 | This video presents several simulation renditions. The different locomotion processes displayed are learned through optimization in open-loop control. It aims at providing the reader a qualitative understanding of the different gait types obtained when constraining the dissipated power at different levels. A discussion on that matter is given in Section 2.2.3 of the related article.

REFERENCES

Brunner, D., Soriano, M. C., Mirasso, C. R., and Fischer, I. (2013). Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* 4, 1364. doi:10.1038/ncomms2368

Buchli, J., Iida, F., and Ijspeert, A. J. (2006). "Finding resonance: adaptive frequency oscillators for dynamic legged locomotion," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing: IEEE), 3903–3909.

Caluwaerts, K., Despraz, J., Isçen, A., Sabelhaus, A. P., Bruce, J., Schrauwen, B., et al. (2014). Design and control of compliant tensegrity robots through simulation and hardware validation. *J. R. Soc. Interface* 11, 20140520. doi:10.1098/rsif.2014.0520

Caluwaerts, K., D'Haene, M., Verstraeten, D., and Schrauwen, B. (2013). Locomotion without a brain: physical reservoir computing in tensegrity structures. *Artif. Life* 19, 35–66. doi:10.1162/ARTL_a_00080

Cham, J. G., Karpick, J. K., and Cutkosky, M. R. (2004). Stride period adaptation of a biomimetic running hexapod. *Int. J. Robot. Res.* 23, 141–153. doi:10.1177/0278364904041323

Collins, S., Ruina, A., Tedrake, R., and Wisse, M. (2005). Efficient bipedal robots based on passive-dynamic walkers. *Science* 307, 1082–1085. doi:10.1126/science.1107799

Dasgupta, S., Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2015). Distributed recurrent neural forward models with synaptic adaptation and cpg-based control for complex behaviors of walking robots. *Front. Neurorobot.* 9:10. doi:10.3389/fnbot.2015.00010

Degrave, J., Caluwaerts, K., Dambre, J., and Wyffels, F. (2015). "Developing an embodied gait on a compliant quadrupedal robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg: IEEE), 4486–4491.

Dickinson, M. H., Farley, C. T., Full, R. J., Koehl, M., Kram, R., and Lehman, S. (2000). How animals move: an integrative view. *Science* 288, 100–106. doi:10.1126/science.288.5463.100

Fernández Cara, E., and Zuazua Iriondo, E. (2003). Control theory: history, mathematical achievements and perspectives. *Bol. Soc. Esp. Mat. Apl.* 26, 79–140.

Fernando, C., and Sojakka, S. (2003). "Pattern recognition in a bucket," in *European Conference on Artificial Life* (Dortmund: Springer), 588–597.

Füchslin, R. M., Dzyakanchuk, A., Flumini, D., Hauser, H., Hunt, K. J., Luchsinger, R. H., et al. (2013). Morphological computation and morphological control: steps toward a formal theory and applications. *Artif. Life* 19, 9–34. doi:10.1162/ARTL_a_00079

Ghazi-Zahedi, K., Haeufle, D. F., Montúfar, G., Schmitt, S., and Ay, N. (2016). Evaluating morphological computation in muscle and dc-motor driven models of hopping movements. *Front. Robot. AI* 3:42. doi:10.3389/frobt.2016.00042

Haeufle, D., Günther, M., Wunner, G., and Schmitt, S. (2014). Quantifying control effort of biological and technical movements: an information-entropy-based approach. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 89, 012716. doi:10.1103/PhysRevE.89.012716

Hansen, N. (2006). "The CMA evolution strategy: a comparing review," in *Towards a New Evolutionary Computation*, eds J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea (Berlin: Springer Berlin Heidelberg), 75–102.

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., and Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biol. Cybern.* 105, 355–370. doi:10.1007/s00422-012-0471-0

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., and Maass, W. (2012). The role of feedback in morphological computation with compliant bodies. *Biol. Cybern.* 106, 595–613. doi:10.1007/s00422-012-0516-4

Hermans, M., Schrauwen, B., Bienstman, P., and Dambre, J. (2014). Automated design of complex dynamic systems. *PLoS ONE* 9:e86696. doi:10.1371/journal.pone.0086696

Hoffmann, M., and Müller, V. C. (2014). "Trade-offs in exploiting body morphology for control: from simple bodies and model-based control to complex bodies with model-free distributed control schemes," in *E-book on Opinions and Outlooks on Morphological Computation*, eds H. Hauser, R. M. Fuchslin, and R. Pfeifer, Chapter 17, 185–194. arXiv preprint arXiv:1411.2276.

Iida, F., and Pfeifer, R. (2006). Sensing through body dynamics. *Rob. Auton. Syst.* 54, 631–640. doi:10.1016/j.robot.2006.03.005

Larger, L., Soriano, M. C., Brunner, D., Appeltant, L., Gutiérrez, J. M., Pesquera, L., et al. (2012). Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express* 20, 3241–3249. doi:10.1364/OE.20.003241

Lukoševičius, M., and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* 3, 127–149. doi:10.1016/j.cosrev.2009.03.005

Lungarella, M., and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Comput. Biol.* 2:e144. doi:10.1371/journal.pcbi.0020144

McGeer, T. (1990). "Passive walking with knees," in *Proceedings, 1990 IEEE International Conference on Robotics and Automation* (Ibaraki: IEEE), 1640–1645.

McMahon, T. A., and Cheng, G. C. (1990). The mechanics of running: how does stiffness couple with speed? *J. Biomech.* 23, 65–78. doi:10.1016/0021-9290(90)90042-2

Müller, V., and Hoffmann, M. (2016). What is morphological computation? On how the body contributes to cognition and control. *Artif. Life*. 23, 1–24. doi:10.1162/ARTL_a_00219

Nakajima, K., Hauser, H., Li, T., and Pfeifer, R. (2015). Information processing via physical soft body. *Sci. Rep.* 5, 10487. doi:10.1038/srep10487

Nakajima, K., Li, T., Hauser, H., and Pfeifer, R. (2014). Exploiting short-term memory in soft body dynamics as a computational resource. *J. R. Soc. Interface* 11, 20140437. doi:10.1098/rsif.2014.0437

Palm, W. J. (1999). *Modeling, Analysis, and Control of Dynamic Systems*. New York: Wiley.

Paul, C. (2006). Morphological computation: a basis for the analysis of morphology and control requirements. *Rob. Auton. Syst.* 54, 619–630. doi:10.1016/j.robot.2006.03.003

Pfeifer, R., and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press.

Polani, D. (2011). "An informational perspective on how the embodiment can relieve cognitive burden," in *2011 IEEE Symposium on Artificial Life (ALIFE)* (Paris: IEEE), 78–85.

Reinhart, R. F., and Steil, J. J. (2009). "Reaching movement generation with a recurrent neural network based on learning inverse kinematics for the humanoid robot iCub," in *9th IEEE-RAS International Conference on Humanoid Robots* (Paris: IEEE), 323–330.

Sussillo, D., and Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron* 63, 544–557. doi:10.1016/j.neuron.2009.07.018

Thijssen, J. (2007). *Computational Physics*. Cambridge University Press.

Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., et al. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* 5, 3541. doi:10.1038/ncomms4541

Wyffels, F., Li, J., Waegeman, T., Schrauwen, B., and Jaeger, H. (2014). Frequency modulation of large oscillatory neural networks. *Biol. Cybern.* 108, 145–57. doi:10.1007/s00422-013-0584-0

Yoshida, H. (1990). Construction of higher order symplectic integrators. *Phys. Lett. A* 150, 262–268. doi:10.1016/0375-9601(90)90092-3

Zahedi, K., and Ay, N. (2013). Quantifying morphological computation. *Entropy* 15, 1887–1915. doi:10.3390/e15051887

Zhao, Q., Nakajima, K., Sumioka, H., Hauser, H., and Pfeifer, R. (2013). "Spine dynamics as a computational resource in spine-driven quadruped locomotion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo: IEEE), 1445–1451.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Urbain, Degrave, Carette, Dambre and Wyffels. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 25 July 2018
doi: 10.3389/fnbot.2018.00037

Modular Neural Mechanisms for Gait Phase Tracking, Prediction, and Selection in Personalizable Knee-Ankle-Foot-Orthoses

Jan-Matthias Braun[1,2]*, *Florentin Wörgötter*[1,2] *and Poramate Manoonpong*[2,3,4]

[1] Computational Neuroscience Group, 3. Physics Institute, Georg-August-University, Göttingen, Germany, [2] Bernstein Focus Neurotechnology, Georg-August-University, Göttingen, Germany, [3] CBR Embodied AI & Neurorobotics Lab, The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark, [4] Bio-inspired Robotics & Neural Engineering Lab, School of Information Science & Technology, Vidyasirimedhi Institute of Science & Technology, Rayong, Thailand

Orthoses for the lower limbs support patients to perform movements that they could not perform on their own. In traditional devices, generic gait models for a limited set of supported movements restrict the patients mobility and device acceptance. To overcome such limitations, we propose a modular neural control approach with user feedback for personalizable Knee-Ankle-Foot-Orthoses (KAFO). The modular controller consists of two main neural components: neural orthosis control for gait phase tracking and neural internal models for gait prediction and selection. A user interface providing online feedback allows the user to shape the control output that adjusts the knee damping parameter of a KAFO. The accuracy and robustness of the control approach were investigated in different conditions including walking on flat ground and descending stairs as well as stair climbing. We show that the controller accurately tracks and predicts the user's movements and generates corresponding gaits. Furthermore, based on the modular control architecture, the controller can be extended to support various distinguishable gaits depending on differences in sensory feedback.

Keywords: artificial neural network, neural orthosis control, adaptation, gait classification, level walking, stair climbing, internal model, model invalidation

OPEN ACCESS

Edited by:
Florian Röhrbein, Technische Universität München, Germany

Reviewed by:
Huanqing Wang, Carleton University, Canada
Xin Luo, Chongqing Institute of Green and Intelligent Technology (CAS), China
Miriam Zacksenhouse, Technion-Israel Institute of Technology, Israel

***Correspondence:**
Jan-Matthias Braun
janbraun@gwdg.de

Received: *23 December 2016*
Accepted: *14 June 2018*
Published: *25 July 2018*

Citation:
Braun J-M, Wörgötter F and Manoonpong P (2018) Modular Neural Mechanisms for Gait Phase Tracking, Prediction, and Selection in Personalizable Knee-Ankle-Foot-Orthoses. Front. Neurorobot. 12:37. doi: 10.3389/fnbot.2018.00037

1. INTRODUCTION

Bipedal gait is inherently unstable (Winter, 1995; Milton et al., 2009) and therefore requires constant balancing and support from the lower limbs. This inherent instability can be amplified by changes in nerve, muscle, or bone status. Consequences can range from limited mobility to a complete inability of effective locomotion. Many different supporting techniques have been developed to regain mobility, e.g., crutches, splints, and prostheses, depending on the physical condition. Here, we focus on a knee-ankle-foot-orthosis (KAFO), a device which is attached to the lower limbs and provides mechanical support to its users.

Selecting and fitting such a supportive device for a patient are performed by professional staff. Based on a given patient's condition, the professional staff determines a device providing the support needed to enable or improve the patient's locomotion. This choice has to take into account the patient's remaining abilities, balancing the patient's need for support against the danger of excessive support which might prevent use of the patient's remaining abilities. In this light, it is

advantageous to maximize the patients' own contribution to train their remaining abilities pursuing the aim to preserve or regain the patients' mobility. This increase in mobility has, for example, been observed in terms of the range of kinematic parameters and walking speed (Irby et al., 2005, 2007).

From the patient's perspective, other factors contribute to the assessment of the chosen device and thus to the actual use or even abandonment of a device. For example, comfort in daily activities, the ability to fast and easily don/doff the device, and cosmetic properties, i.e., how the device alters the self-perception, or the perception of others (Bernhardt et al., 2006; Robinson et al., 2010; McKee and Rivard, 2011), have a high impact on patient satisfaction. Kaufman et al. (1996) presents several studies where abandonment rates "from 60 % to nearly 100 %" were observed. In Phillips and Zhao (1993), from 60 users of lower extremity braces, 35 abandoned their devices. In the list of top reasons, the authors cite whether the user's "personal opinion [was] considered in the selection process." Interestingly, it was not important if there were "alternatives to choose from." A follow-up survey on 250 veterans after 22 months of rehabilitation programs showed that only 16 out of 73 contactable patients were still using their braces. In other words, 78 % had abandoned their devices. In another example, in 31 % of 35 replies, patients expressed that they did not use their brace anymore while "60 % continued to use their wheelchair as their main means of displacement" (Mikelberg and Reid, 1981). Although the studies on device abandonment are from the 1980s and 1990s, they make clear the importance of the patient's opinion concerning the prescribed brace.

Another problem is side effects of orthosis use, which can arise from device limitations or mistrust toward its reliability. Gailey et al. (2008) gives an overview on gait deviations of prosthesis users, including a tendency to favor the intact limb, generating additional stress on the less impaired parts of the body which may induce secondary conditions. There are, for example, (1) degenerative changes (trunk asymmetry, osteoarthritis, and scoliosis), (2) pain (lower back, hip, and/or knee joints), and (3) general deconditioning. These changes can, for example, be observed as asymmetric and slower gaits. Compensatory gait patterns like "increased upper-body lateral sway, ankle plantar flexion of the contralateral foot (vaulting) hip elevation during swing phase (hip hike) or leg circumduction" are listed especially for orthoses with fully extended knees in Yakimovich et al. (2009). Mills et al. (2010) come to the conclusion that "there is a large amount of variability with regard to how patients respond to orthoses." These studies suggest that the patient's gait and device-perception can be improved by individual fitting. As a device-induced gait change surfaces on long timescales (Irby et al., 2005, 2007), Robinson et al. (2010) are speaking about a "lifetime of adjustments," an approach for individual fitting has to facilitate continuous adaptation.

In consequence, to achieve patient acceptance in addition to an optimal medical outcome, one has to consider the user's impression of the device and its fitting as well as the specialist's opinion.

When looking at controller techniques, finite state machine based controllers (FSMs) provide a large fraction of state of the art approaches. In a FSM, the supported gaits are represented by a number of states, which define the control output. Given specific conditions, transitions between these states occur. To achieve adjustment in such an approach, parameters defining the control output or transition conditions may be changed. The number of states and transitions is typically left unchanged, with exceptions like Zlatnik et al. (2002), where a rule database is used. As the complexity and variety of the supported behaviors directly translates to the complexity of the graph, a higher number of states and transitions are needed. The process of designing or extending gait support has to ensure that the controller will not get stuck and that transitions support all possible changes in patient behavior.

Approaches to provide complex behaviors, i.e., more gaits, have to deal with increasing complexity. For example, a controller, consisting of three FSMs for walking, standing and sitting movements, is presented in Varol et al. (2010), where additional transitions switch between these individual FSMs. In Sup et al. (2011), a similar approach handles slopes of varying degrees, using three FSMs for level ground, as well as 5 ° and 10 ° inclinations. These approaches try to reduce FSM complexity by a divide-and-conquer approach, but only tackle sub-problems, e.g., either level and slope walking or level walking, standing, and sitting. A similar approach is gait (FSM) switching based on a Gaussion mixture models (Varol et al., 2008). The switching is based on a history of means and standard deviations in the input channels treated with dimension reduction; inputs are sampled with 1, 000 Hz. The behavior was optimized in terms of increasing the history length, resulting in a switching delay of 430 ms in the testing condition.

Other approaches try to provide adaptive control. Speed estimation (Herr et al., 2002) and slope estimation in standing (Lawson et al., 2011) try to achieve control which adapts to gait and environment by the adjustment of control parameters at runtime. These approaches are limited by the flexibility designed into the underlying state machines.

These controllers select between gaits chosen at design time and are partially able to adapt to environmental changes or walking speed. The included FSM controllers represent predefined gait models, only allowing to fit the behavior to the patient with design-time selected parameters. With more supported movements comes higher complexity in terms of an increase in states and possible transitions, which allow more parameters to be selected, still the designed gait model may not cover every individual gait. This problem of fitting the controller to the patient gets worse in case of orthoses, where the patient's conditions are more variable. The variability in patients conditions results in large variability in remaining abilities and, thus, in large variability in individual need for support. These varying conditions are often met with very individual avoidance or compensation strategies, resulting in unique gaits which can conflict with the general gait model.

In consequence, we identified five important problems, which dominate the success of a device: (i) *individualization* according to the patients' neurological status and remaining motor function. As a complication, orthosis patients can show very differentiated medical conditions which have to be compatible

with the devices' support. (ii) Typically devices have a *specialized design* supporting a reduced set of movements, which limit the patients' mobility. (iii) The *target group* of orthotic devices is typically limited as a consequence of (i) and (ii). (iv) *An asymmetric gait* due to patients' favoring of healthy limbs leading to gait deviations and secondary conditions. (v) *Device acceptance* is strongly subjective and depends on users' opinions toward the device and their role in the selection process.

These problems have so far not been approached with a common concept. Here, we assume, that they can be addressed best on the controller side with a shift of focus on extensive patient fitting and behavior adaptation. Thus, we propose a personalized and patient centered approach, which *individualizes* via training with patient data. With an user interface, patients directly influence the control output, giving them direct feedback on a possible tuning. The patients' gaits are tracked in terms of gait dynamics, i.e., joint-sensor dynamics. This approach of relying on the sensor dynamics makes the controller independent of the actual mechanical structure of the device as well as, for example, the moments or joint angles the patients can apply and maintain. As controller training leads to a high affinity to the gait dynamics of the trained gait, a modular structure is presented. In this modular structure, the number of supported gaits is limited by the systems' ability to differentiate the gaits by their dynamics. These dynamics are determined by the chosen sensors. As the design is not geared toward a specific set of sensors, the modular controller is not explicitly limited toward specific movements. Thus, given suitable data, which signifies the new gait, the extension with a new gait is a simple, formalized process. With this approach, we want to overcome *design limitations* and extend the *target group*. Furthermore, training of *individual* gait with direct patient feedback in the tuning process may lead to a better fitting and understanding of the controller behavior, hopefully leading to more *symmetric gait* and better *device acceptance*.

Taken together, in this study we present gait dependent damping modulation based on gait phase tracking. Gait phase tracking is based on observed gait samples and therefore implements personalized gait support for single gaits. The damping modulation is implemented as a one dimensional mapping from the gait phase to the desired damping, which can be adapted via a simple user interface. Together, an implementation of (a) gait phase tracking paired with (b) suitable damping modulation constitutes a supported gait. The second contribution lies in the selection of a suitable gait from a set of supported gaits, allowing to adapt the controller's behavior to gait changes. The most suitable gait is selected based on the gait dynamics, which is predicted by internal models for the gait dynamics. Thus, for the second contribution, now three components constitute a supported gait: (a) gait phase tracking, (b) suitable damping modulation, and (c) a model to predict the gait's dynamics for gait identification. Gait selection on three such defined gaits, for walking on flat ground, stair climbing, and descending stairs, have been tested on a healthy subject with an orthosis prototype provided by Otto Bock. This prototype applies damping to knee flexion, providing support to the users' body. Based on the tests with this prototype, we provide performance data on the method's ability to linearly track the gait phase, as well as its ability to fast and reliably select a suitable single-gait controller. As our long term goals of the study, we aim to implement a fully adaptive controller with the patient's user-feedback. The feedback mechanism will not only enable the user to influence the devices behavior, but also provide the means to control changes made by an adaptive controller.

2. MATERIALS AND METHODS

In the introduction, we outlined the five general problems we see with current control schemes and our approach to them, like individualization, fitting, and behavior adaptation based on patient gait data with the inclusion of user feedback. Here, we outline the implementation based on the concrete control scheme (**Figure 1**) for gait phase tracking, prediction, and selection.

We present the hardware platform together with the sensors capturing its configuration in section 2.1. In section 2.2, we introduce the single gait controller as the core neural control module. It consists of gait phase tracking, the timing module, and the shaping module which transforms the gait phase into the control output. The single gait controller relies on the user interface to gather user-feedback (in section 2.3), which is only considered here, as it provides the control output as a function of the gait phase. Based on single gait control, section 2.4 presents predicting gait models and the gait-selection module, which will select a single-gait controller in accordance with current motion. As fundamental basis for the analyses, we describe how the segmentation of continuous recordings into steps has been performed in section 2.5. Finally, the experiments underlying this manuscript are presented in section 2.6.

2.1. Hardware

The hardware platform used during controller development and for tests with healthy walkers is based on the Otto Bock C-Leg® hydraulic damper attached to the knee joint of a knee-ankle-foot-orthosis. The damper allows the design of a semi-active orthosis, as it actively manipulates damping of knee-flexion with a motor-controlled valve. The interface allows to position the valve in 100 configurations from effectively free motion (open valve) over high damping to blocked motion (closed valve).

While the overall design as leg splint with the knee damper system restricts the target group, we aim to have the controller as universally applicable as possible. Therefore, we implemented and tested with two hardware models (**Figure 2**). One hardware model has a compliant ankle joint (**Figures 2A,B**). It uses a carbon fiber bar of high stiffness beneath the knee which is directly attached to the foot, thus fixating the ankle. The other one has an orthopedic ankle joint (**Figures 2C,D**). The orthopedic ankle joint allows either free motion or to constraint the range of motion by blocking it or inserting a spring (comparable to the Otto Bock double action joints). Both models' hardware structure is similar to Otto Bock's C-Brace® system, which is equipped with a C-Leg® hydraulic damper itself. As each device was fitted for a different healthy user, the data wouldn't have been directly comparable. For this reason, we only include data from the device with the orthopedic ankle joint in this study, while pointing out that the controller itself is independent of the actual hardware structure.

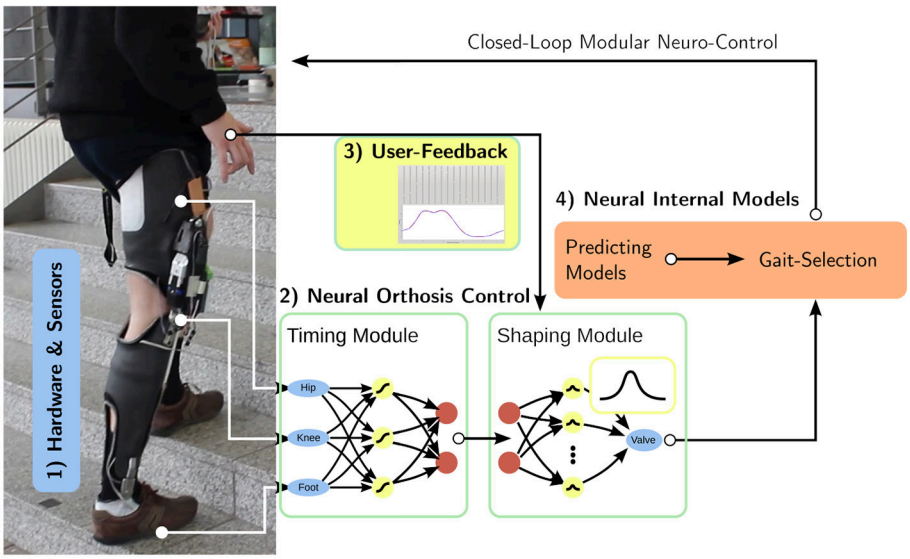


FIGURE 1 | Overview: Based on (1) the hardware platform and its sensors, we develop (2) neural feed-forward gait controllers consisting of a timing and an output shaping module. While the timing unit learns the user's gait from observation, the shaping module is controlled via direct (3) user input, integrating the patient in the control loop. The whole controller consists of several feed-forward controllers which are augmented with predicting gait models. Based on these (4) internal models, the controller selects the correct gait for the current motion.

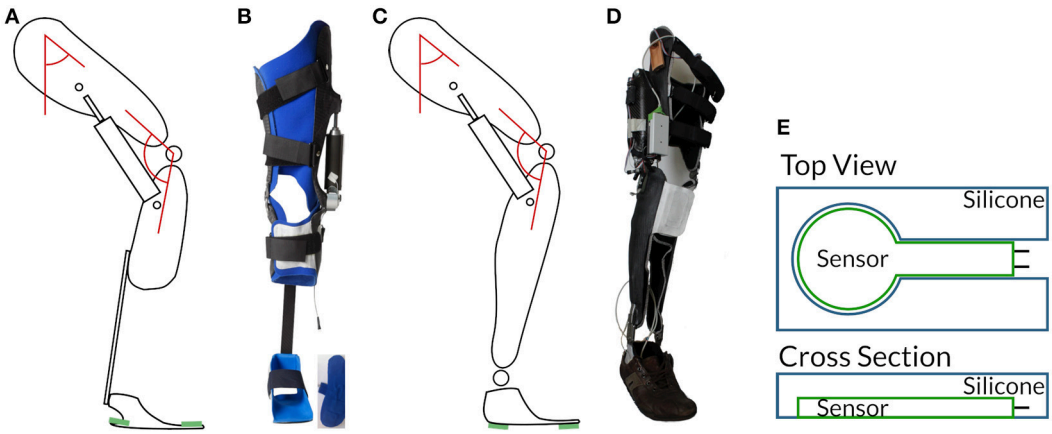


FIGURE 2 | Hardware revisions: In **(A,B)** the first generation with compliant ankle, in **(C,D)** the second with orthopedic ankle joint. The schematics include positions of thigh- and knee-angle sensors indicated by red angles and pressure sensors in green. Force sensing resistors (FSRs) are embedded between the orthosis frame and the shoe. To protect the highly sensitive FSRs against interactions between the orthosis frame and the shoe, we embed them in a silicone plate **(E)**. The applicability to different different hardware layouts shows how versatile the presented controller is.

A data acquisition interface allows sampling of the embedded sensors at 100 Hz. As sensors, we equipped angle sensors at the thigh and the knee-joint, and force sensing resistors (FSRs) in the soles between orthosis frame and shoe. The latter are very sensitive and show a binary switch characteristic due to their measuring range of $\approx 0.1 - 100.0$ N. We therefore embedded them in a silicone layer to reduce noise from interactions between the orthosis frame and the shoe. We localized and fixated all sensors on the device to keep the procedure of device application as simple as possible. Additionally, sensor calibration to achieve full range input signals for the artificial neural networks does not have to be recalculated when reequipping the device.

2.2. Neural Control for Gait Tracking

The application of damping to knee-flexion is a one dimensional control problem, where the controller determines a valve position regulating the desired damping. We assume, that the required damping can be determined from the gait phase, i.e., the configuration of the leg as represented by a

suitable set of sensors. In practical situations, the controller has to cope with huge variances in space and time. Here, the controller is designed as a feed forward controller to achieve an immediate response to sensory inputs $\vec{s}(t)$ in two steps. The *timing unit* estimates the phase of the gait $\varphi(t)$ from the sensory reading $\vec{s}(t)$, while the *shaping unit* determines the damping $c(t) = c(\varphi(t))$ given the phase (**Figure 1**). Separating these two steps allows to independently modify the gait tracking and the desired controllers behavior.

For the time discrete control system we write at time step t:

$$\text{timing unit}: \vec{s}^{\,t} \mapsto \varphi^t, \varphi^t \in [0, 1) \tag{1}$$

$$\text{shaping unit}: \varphi^t \mapsto c^t\left(\varphi^t\right), \text{with} c\,(0) \equiv c\,(1)\,. \tag{2}$$

We chose to implement c with a radial-basis-function network as universal function approximator, as detailed in section 2.3.

The gait progress φ is modeled as a cyclic, angular variable (**Figure 3**), thus capturing the periodicity feature of walking.

The timing unit is implemented using a multi-layer perceptron network with sigmoidal activation function (Nissen, 2003) with four neurons in one hidden layer and two output neurons, representing φ as circular motion in the plane. Thus, the output function is similar to the periodic sensory inputs, which improves learning and accuracy.

$$\hat{\varphi}^t : \vec{s}^{\,t} \mapsto \begin{pmatrix} x_\varphi^t \\ y_\varphi^t \end{pmatrix} = \begin{pmatrix} \cos\left(2\pi\varphi^t\right) \\ \sin\left(2\pi\varphi^t\right) \end{pmatrix}.$$

In case of noisy sensors, a low-pass filter can be applied on top of the output function $\hat{\varphi}$. With reliable sensors, this step is typically not needed. As it only leads to a small delay, it can be applied anyhow.

The gait phase can then be gained using the transformation

$$\varphi^t = \begin{cases} \frac{1}{4} & \text{for } x_\varphi^t = 0 \wedge y_\varphi^t \geq 0 \\ \frac{1}{2\pi} tan^{-1}(y_\varphi^t / x_\varphi^t) & \text{for } x_\varphi^t \neq 0 \\ \frac{3}{4} & \text{for } x_\varphi^t = 0 \wedge y_\varphi^t < 0 \end{cases}, \varphi \in [0, 1).$$

To facilitate network training, sensor calibration is used to map all values to the range $[-1, 1]$. The calibration procedure uses: vertical thigh (0), to 90° flexion (1). The knee angle is mapped from straight (−1) to 90° flexion (1). For the force sensors, thresholds are chosen such that ground contact maps to ≤ 0 and a free foot to 1.

Training data is then segmented into steps (section 2.5) using the ground contact signal. The step duration l_j is determined as the number of samples in the jth step. Then, the desired gait phase φ_i and network output o_i for each sample i are given as

$$\varphi_i = \frac{i}{l_j}, \tag{3}$$

$$\vec{o}_i = \begin{pmatrix} o_1 \\ o_2 \end{pmatrix} = \begin{pmatrix} -sin(2\pi\varphi_i) \\ -cos(2\pi\varphi_i) \end{pmatrix}. \tag{4}$$

We chose the sensors with the aim to capture motion in terms of sensor dynamics, instead of relying on defined events. The *timing module* frees the sensory inputs from time dependencies and thus provides a device, gait, and patient independent description of gait progress. The second part, the *shaping module*, generates the control output. It is augmented by a user interface for direct user-feedback.

2.3. User Defined Output Modulation

The damping function is tailored to the need of the individual user by incorporating user feedback in the shaping of the damping function $c(\varphi)$.

The user interface (**Figure 4**) provides the samples to fit the Radial-Basis-Function network, which provides universal function approximation (Park and Sandberg, 1991; Buhmann, 2003). The sliders represent the applied damping $c(\varphi)$ by values on a grid of supporting points, with the lowest position corresponding to no damping, the highest position to maximum damping. The Radial-Basis-Function network is updated immediately and thus users can immediately experience the effect of their changes. The choice of the Gaussian kernels' widths allows to choose the amount of smoothing of user-input applied during network-training.

We used a network of $n = 10$ equidistant nodes in the interval $[0, 1)$. The Gaussian transfer function had a half-width of $\sigma = \sqrt{\frac{n}{2}}$.

2.4. Gait Recognition

The neural orthosis controller described in section 2.2 is gait specific: its timing and shaping units were designed to estimate

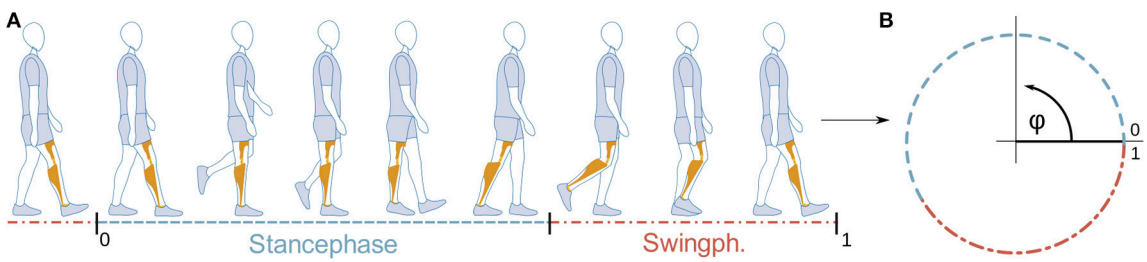


FIGURE 3 | Mapping of gait progress with time **(A)** to the cyclic gait-phase variable φ **(B)**. Restricting $\varphi \in [0, 1)$ allows a direct comparison to classical gait descriptions in the literature. (Illustration **(A)** based on marketing material by Otto Bock).

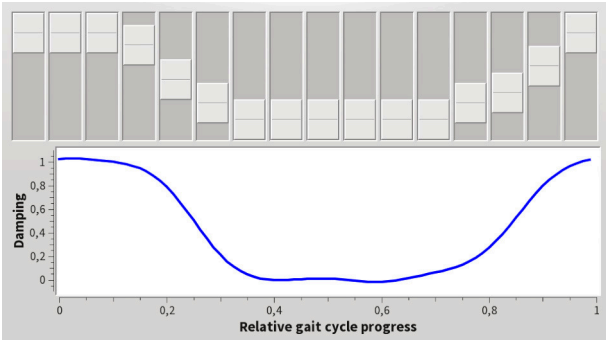


FIGURE 4 | User interface to tune the control output, i.e., the knee damping. The horizontal axis shows the gait phase φ, the vertical the desired damping c. Changing a slider, directly modifies the desired damping at the corresponding gait phase. The system updates its behavior immediately to changed user input. This enables users to tune the device's behavior to their expectations and needs.

the phase of a specific gait and generate the damping appropriate for that gait. To support different gaits, we propose to train different controllers for different gaits, and to activate the proper controller based on model based gait recognition. As long as the gaits can be differentiated with the available sensory information (**Figure 1**), the number of supported gaits is not directly limited. The controller can be extended by providing single-gait controller modules together with internal models to identify the corresponding gait (Braun et al., 2014).

Gait recognition is based on the prediction of sensory input: each gait is associated with a predictor $\mathcal{P}$ that predicts the sensory input $\vec{s}^{\,t+1}$ for the next time step based on a subset of the sensory history $\mathcal{H}_N = \left\{\vec{s}^{\,t}, \vec{s}^{\,t-1}, \ldots, \vec{s}^{\,t-N+1}\right\}$, where N is the history length. A comparison of the prediction of the next time step's sensory reading $\vec{p}^{\;t+1}$ to the actual sensory reading $\vec{s}^{\,t+1}$ in the next time step defines a prediction error. The prediction error allows the decision unit to determine the best fitting gait model by choosing the model with the smallest error within predefined acceptable bounds of error.

To estimate reasonable history lengths N, we assume a minimal step duration T_{step} of

$$T_{step} \gtrapprox 1\,\text{s},$$

for complete steps with the orthosis. Further, we assume a stance to swing duration ratio of $\approx 60:40$ and that gait changes can occur at any time[1]. When the gait changes, the history contains two gaits and will naturally lead to diverging predictions of models trained on a single gait. While this prediction error is critical to achieve a fast invalidation of the old gait, we want the history to contain only one gait swiftly afterwards. To estimate a reasonable scale of the new gait's duration in the transition step, we go for a fraction of 50 % of the swing phase as the smaller gait phase. This translates to around 20 % of the step length. Given the hardware specific sampling frequency of 100 Hz, we determine the maximum number of samples in the history N_{History} to

$$N_{\text{History}} \lessapprox 20\,\text{samples} = \frac{1}{5}\,\text{s}.$$

When the history length is chosen larger, a gait change could stay longer in the history than the above requested 20 % of the step length.

The history length is a trade-off between the prediction's accuracy and the supported frequency of gait switches. The prediction accuracy should increase with a longer history, which can cover more details leading to better discrimination. In contrast, the frequency of supported gait switches will decrease, because data from different gaits in the history will lead to lower quality predictions while conflicting gaits stay in the history. The choice of $T_{\mathcal{H}} = \frac{1}{5}$ s allows several gait switches per step with quite accurate results, as shown below.

2.4.1. Predicting Gait Models

The predicting gait models were implemented like the timing unit of the feed-forward gait controller above. Based on a history of sensor data, a single channel was predicted by the predicting model. To this end, a multi-layer perceptron network with 3 neurons in the hidden layer and one output neuron was trained on recorded gait samples to predict the sensory inputs using a history. The history was implemented as a delay line (**Figure 5**).

First experiments showed that the backpropagation learning algorithm tended to exploit the last sensor reading as a good prediction of the next time step. Ignoring most of the history, this was equivalent to the approximation with a constant value, predicting the next time step's state almost only on the preceding one as the error of the approximation was in many cases of the order of magnitude of the signal change, considering step to step fluctuations and sensor noise. This prediction on only one time step was independent of the trained mode. Thus, this approach had no predictive power which related to the actual gait's dynamics.

As a consequence, only a subset of the history is actually used for prediction. Especially for the predicted channel, the current sensory reading is omitted and only older values are used. Effectively, we have coarsened the history to a grid with a width of $10\Delta t$ (**Figure 5**), to exclude simple models. Besides solving the problem of forwarding of the last reading, this sparse selection reduces the computational complexity of the models greatly.

Therefore, to predict channel i $\vec{p}_i^{\;t+1}$, we use sensory readings of all other channels for t, $t-9$, and $t-19$, but only the readings for $t-9$ and $t-19$ of the predicted channel (**Figure 5**).

2.4.2. Prediction-Based Gait Selection

The part of the controller, which selects the current gait based on the model's predictions, will be called the decision unit, in accordance to previous naming conventions. The selection

[1]If we consider, for example, **Figures 13**, **14**, they show a transition from descending stairs to flat walking. We see that after the heel strike follows a period, where the dynamics seem to mostly follow the stair regime. This is followed by a period where differences to both gaits occur. Here, it is difficult to determine the gait until after the maximum knee flexion, when the gait dynamics converge to those of the flat walking regime. As the flat walking gait is only reached during the extension of the knee, we assume that the new gait is reached during the swing phase. Still, the transition changes the whole step's dynamics.

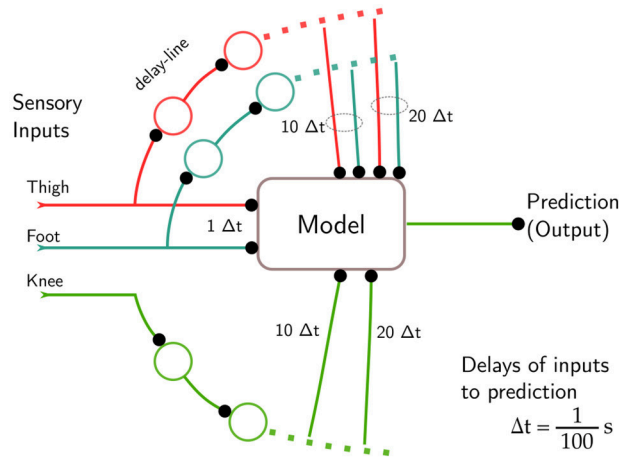


FIGURE 5 | Gait detection tries to minimize the prediction error for sensory inputs $\vec{s}^{\,t}$ by choosing an appropriate model. Delay lines provide a history $\mathcal{H}$ of sensory inputs, which is fed into a feed-forward model predicting a single value in the next time step, here the knee. The delay relative to the predicted signal is given as a multiple of the sampling interval $^1/_{100}$ s. For the predicted channel, the current input is not fed in, as experiments showed a high probability of the network to choose it as the best prediction.

process chooses the gait model which minimizes the prediction error $\vec{e}_j^{\,t} = \left|\vec{p}_j^{\,t} - \vec{s}_2^{\,t}\right|$ for the gait j with the sensory inputs $\vec{s}_2^{\,t}$. The subscript 2 indicating that the ground contact signal has not been used for the calculation of the prediction error. The absolute value was chosen to increase sensitivity to the amplitude of the prediction error, preventing the low pass filter (below) to average out fluctuations.

These prediction errors often occur in relatively short intervals of the step. Thus, we apply post-processing in form of a low pass filter to prolong the time span that the final fitness-measure is usable.

$$\tilde{e}_{i,j}^{t} = (1-\beta)\tilde{e}_{i,j}^{t-1} + \beta e_{i,j}^{t}, \beta = 0.9, \quad i \in \{\text{knee}, \text{hip}\}.$$

Model- (j) and channel (i) -specific thresholds $\theta_{i,j}$ suppress prediction noise in the low pass filtered errors $\tilde{e}_{i,j}^{t}$. These thresholds are chosen for each gait j individually based on the error signal $\tilde{e}_{i,j}^{t}$ for matching gait samples. Remaining prediction errors are counted if they are greater than this threshold $\theta_{i,j}$ (**Figure 6**) for each predicted channel $i \in \{\text{knee}, \text{hip}\}$.

$$f_i^t = \alpha \cdot \begin{cases} f_i^{t-1}, & \text{if } \tilde{e}_i^t < \theta_i \\ \max\left(f_i^{t-1}+1, 2\right), & \text{else} \end{cases}, \alpha \in \{\mathbb{R} | 0 < \alpha < 1\}.$$

This count f_i^t is limited to the range $[0, 2]$ and decays with factor $\alpha = 0.99$. The factor α and the maximum value 2 are chosen such that the value is significant on timescales of steps.

These f_i^t measure the unfitness of the model's predictions per channel and are merged with a gait specific weight γ_j to reflect the importance of the individual channels,

$$f_j^t = \gamma_j f_{thigh}^t + (1-\gamma_j) f_{knee}^t.$$

Finally, all gaits with $f_j^t > 1.1$ are discarded and the gait with the lowest f_j^t, i.e., smallest unfitness, is selected from the remaining gaits. Its feed-forward controller operates the current time step.

2.4.3. Training of Prediciting Models and Selection

Training of predicting models is analogous to the feed-forward controller's timing unit and can use the same recordings. The recordings should reflect the variance in the user's gait and should not be too regular. Then, the perceptron is trained using a backpropagation algorithm.

To improve the performance of the internal models, each model scales the sensory inputs such that typical signals lie in the range $(-1, 1)$. Of course, in addition to optimal working conditions, such a scaling will help to differentiate huge changes in amplitude which might be connected to different gaits. In a converse argument, it supports bad predictions for gaits with too low or too high amplitudes in comparison to a model's training data set.

2.5. Step Segmentation

Segmentation of gait data by step boundaries is needed to create training data as well as for the analysis of the tracking unit's and gait recognition performance. As typical in the literature, the heel-strike marks beginning and end of a step (**Figure 3**), which we determine by flanks of the pressure onset at the heel FSR. Due to the high sensitivity of the sensors and interaction with the orthosis frame and foot, only onsets can be detected and we have to apply filters to compensate varying amplitudes and fluctuations.

The sensory data is assumed to be in the range $[-1, 1]$ with 1 no pressure and -1 high pressure. To improve robustness, we use a hysteresis to detect state changes, changing to ground contact when the sensor goes below 0, and to free heel when > 0.8. Heel-strike detection is implemented with a finite state machine as:

1. Based on the first sample, the state is initialized to *ground contact* or *free heel*.
2. For each sample
   a. If *ground contact* and the current sample is *above threshold for free heel*, then change the state to *free heel*.
   b. If *free heel* and the current sample is *below threshold for ground contact*, then change the state to *ground contact*.
3. Collect the sample numbers of all touch down events (state changes to ground contact) in a list.
4. As the detection reacts to the steepest part of the flank in the FSR signal, we move the touch down event to the preceding sample which is closest to an FSR reading of 0.8 to determine the onset of heel-strike.

This list of events now describes the heel-strikes in the given recording.

2.6. Experiments

2.6.1. Gait Phase Tracking

For single gait support, the following statements hold. (i) We make no assumptions about when and what kind of support

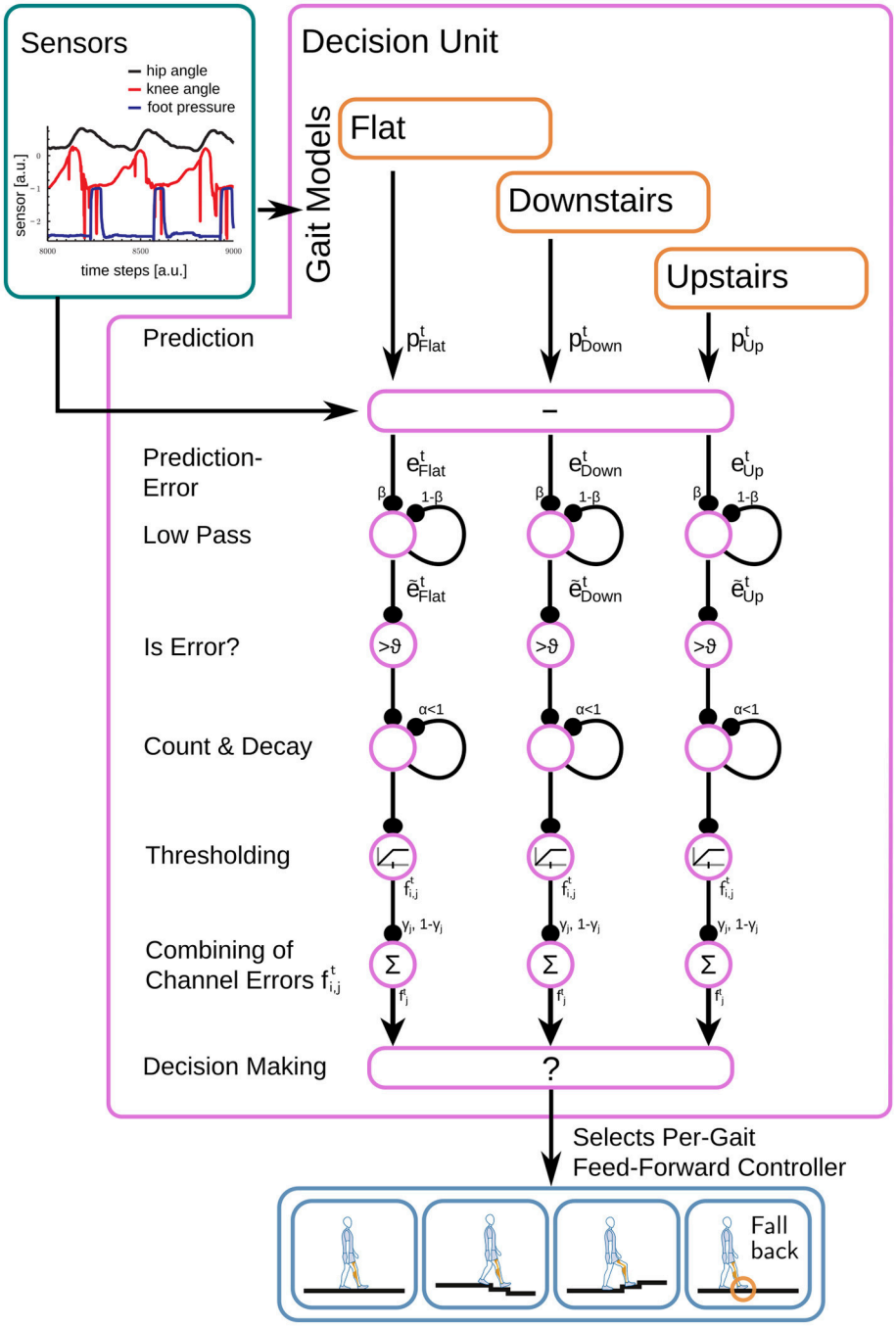


FIGURE 6 | Based on the gait-specific predictions for the next sensory inputs $\vec{p}_j{}^t$ and the actual sensory reading, the decision unit calculates the prediction error $\vec{e}_j^t$ and applies a low-pass filter before an unfitness measure f_j^t is calculated for each gait. At the end, from gaits with f_j^t in an acceptable range, the best fitting one is selected.

the user needs. (ii) The damping function c is smooth (due to the representation as an RBF function). And (iii) the applied damping at knee-flexion is a direct function of the gait phase and thus of the sensory input $c_t = c_t(\varphi_t(\vec{s}_t))$. Thus, the applied damping c only changes when the gait phase φ changes and, in consequence, the controller's ability to apply the desired damping at any gait phase solely depends on the properties of the gait phase φ. Thus, the quality of the gait phase φ determines the quality of the control output.

Ideally, the gait phase φ produces a linear mapping for constant motion velocity, as it guarantees the same detail of control for all phases of gait, i.e., control accuracy does not depend on the gait phase. Thus, we investigate the linearity of the gait progress representation and the timing of the heel strike after training. We compare to the ideal gait phase φ' according to section 2.5, which can only be derived after the heel-strike and therefore has to be acquired for offline processing. To evaluate steps of different duration, we will resample and interpolate each step to 200 samples, leading to an ideal slope of $\Delta\varphi' = \frac{1}{200}$. The deviation of the controller's gait phase to these will be investigated in terms of linearity, monotony, and smoothness.

Furthermore, the timing of the gait phase has to match the timing of the step to provide a useful representation for users, such that the tuning of the control output with the user interface (**Figure 4**) can be done intuitively.

2.6.2. Gait Selection

Due to the controller providing body support at the knee level, it assists in the stance phase, while in the swing phase, all single-gait controllers provide free knee swinging. We therefore argue, that the most important aspect for secure and seamless operation is timely gait switching to prepare for the heel-strike. Thus, to evaluate the accuracy of gait recognition, we check that the controller not only classifies the step correctly but also achieves a correct result prior to heel strike. To quantify correctness and timing, we analyze a walking sequence where a healthy user annotates the intended gait, for example flat walking, stair climbing, and descending stairs. The inclusion of descending stairs requires that we have to deactivate the damping unit for security reasons.

Then, we analyze step by step and measure the time ahead of the heel-strike that the decision unit recognized the step's final gait. **Figure 7** shows how the user's annotations are compared to the controllers' classification: For each step, we compare the controllers' classification against the last valid user annotation. If both match, we measure the duration the correct classification was known and set this duration in relation to the duration of the swing phase to allow the comparison independent of the actual step length. We call this fraction the *range of certainty*. A range of certainty of zero means that the correct gait was not known prior to heel-strike. For a range of certainty of one, the controller was certain of the used gait for the whole swing phase.

Thus, the range of certainty allows to asses if the controller is able to apply the correct gait model during swing phase, where all single-gait models will provide free knee motion. We then analyze the average success rate and range of certainty for all tested gaits, to determine if the presented controller in combination with the sampling frequency of the data acquisition board can react to gait changes. Then, we quantify the controller's ability to differentiate the tested gaits against each other with the selected set of sensors. We conclude with the investigation of gait changes for steps showing conflicts between the user's annotation and the controller's classification.

To access the orthosis controllers accuracy, we take a reaction time into account. At a sampling rate of 100 Hz and step lengths in the experiment between 1.3 and 1.8 s, a *range of certainty* of 3 % guarantees that the orthosis controllers' classification is in time for heel-strike.

3. RESULTS

3.1. Gait Phase Tracking

The experiments conducted aim to show that a trained gait model is able to track gait progress better than a model trained for other gaits. Control quality depends on the smoothness and monotony of the tracked gait phase φ, which we quantify in terms of linearity and the distribution of increments, e.g., discrete slopes. Furthermore, the accuracy of the tracked heel-strike is used as a measure for correct timing.

In a first run, the single gait controllers were trained on runs with 49 steps on even ground and 59 steps on stairs. In a second run, we record the gait phase φ of these two controllers for later comparison to the ideal gait phase φ'. We analyzed 30 steps on even ground and 38 steps climbing stairs of a healthy subject wearing the orthosis. Steps at gait changes were manually removed, due to issues discussed later.

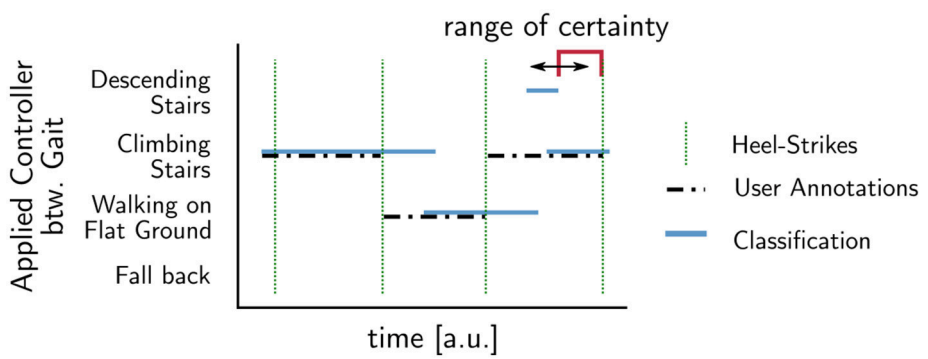


FIGURE 7 | Determination of the time a correct classification result was available before heel-strike: The *range of certainty*, i.e., the fraction of swing phase where user annotations (dotted black) and the system's prediction match. It describes the duration for which the prediction produces a correct result and describes the predictive power of the model for a specific step.

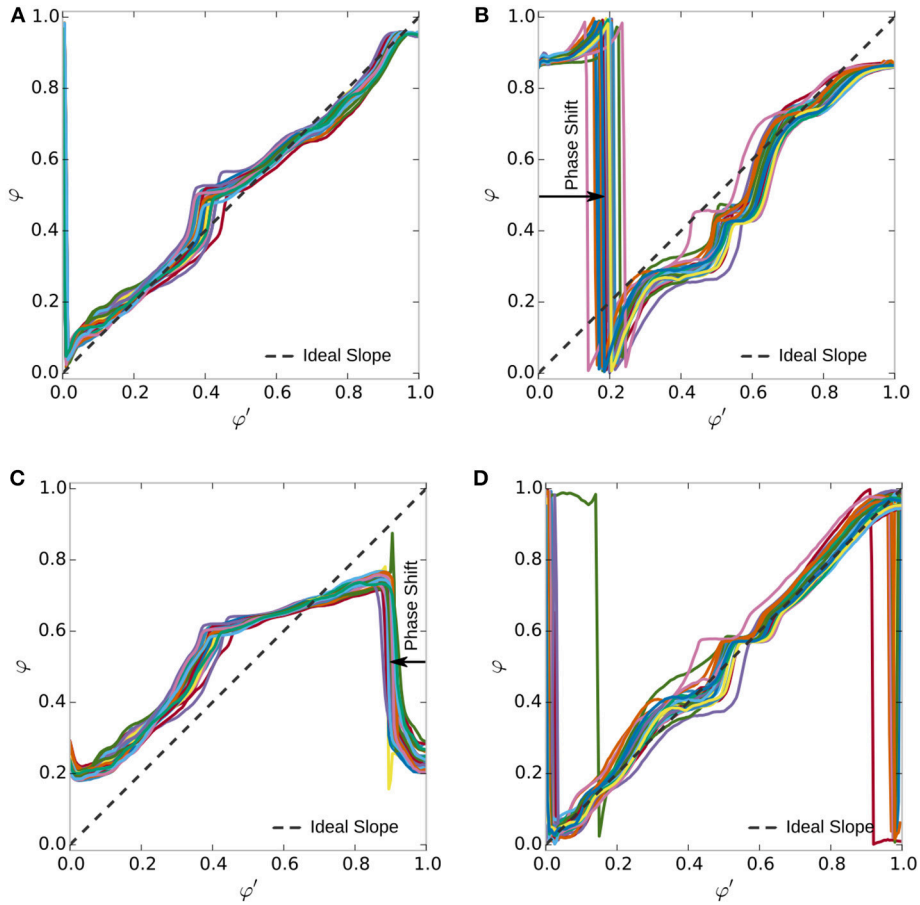


FIGURE 8 | Smoothness of gait phase representation. Colored lines indicate 25 steps on flat ground and 8 on stairs, without gait transitions. Compared are models for walking on flat ground and stair climbing on their native and the opposite terrain. The native models (**A** $r^2 = 0.88$, **D** $r^2 = 0.72$) produce smoother gait phase output in comparison to the unfitting models (**B** $r^2 = 0.33$, **C** $r^2 = 0.06$). The latter expose phase shifts and strong deviations the ideal gait phase φ' indicated by the dashed line. The coefficient of determination (r^2) supports the notion that the native models are generally following the ideal linear relation. The lower number of steps in **(D)** increases the influence of the step segmentation, degrading r^2.

In **Figure 8**, the controller-derived gait phases φ_{Flat} and φ_{Stair} are plotted against the ideal, offline computed gait phase φ'. In the case where the model matches the user's gait (**Figures 8A,D**), the ideal gait phase is approximated well. In the mixed cases (**Figures 8B,C**), where the model does not match the gait, the controller's heel strike has a phase shift against the real event. In addition, the flat ground model on stairs (**Figure 8B**) shows 4 steps with almost constant intervals between the steps. The stair climbing model on flat ground (**Figure 8C**) fails to reproduce the gait phase completely; it only oscillates between 0.2 and 0.8. The r^2 values in **Table 1** support that the native model is close to linear and significantly better than the non-native model.

The accuracy in timing of the heel-strike is accurate only for the trained gait, as the phase shift in **Figure 8** and **Table 1** shows. Considering the sampling frequency of 100 Hz and an average duration of 150–200 samples, the gait phase should progress by $\frac{360\,^\circ}{200} - \frac{360\,^\circ}{150} = 1.8\,^\circ$–$2.4\,^\circ$ per sample. This value matches with the average precision shown in **Table 1**, which is determined by averaging the phase shift indicated in **Figure 8**.

The distribution of increments $\Delta\varphi$ in **Figure 9** supports these observations. When considering the variation of increments around $\Delta\varphi' = \frac{1}{200}$, we considered the interval $\left[\frac{1}{2}\Delta\varphi', 2\Delta\varphi'\right]$. Using this interval, we allow a variation of up to a factor of two in each direction, but do not count extreme or negative increments, as the standard deviation might have. For level walking (**Figure 9A**), the fitting model has 69 % of increments in this interval, while the model for stair climbing only has a fraction of 31 % inside this interval. In the case of steps on stairs (**Figure 9B**), 65 % of the increments are inside for the fitting model and only 40 % for the flat walking model. The histogram for the native models (in red) has its maximum around the optimal slope with lower standard deviation (**Table 1**). For the mixed cases (in blue), the optimal slope is not in the center of the distributions but has a maximum around zero and larger standard deviations. Furthermore, we see the presence

TABLE 1 | Linearity of the graphs in **Figure 8** according to the r^2 values for a linear regression.

		Environment							
		Flat ground				Stairs			
		r^2	Accuracy [°]	Std. dev.	Skew	r^2	Accuracy [°]	Std. dev.	Skew
Model	Flat	0.88	1.8	0.004	3.94	0.33	34.2 ± 3.7	0.033	1.17
	Stair climbing	0.06	−18.0 ± 1.2	0.017	−10.45	0.72	2.5 ± 5.5	0.005	2.02

*Timing accuracy of heel strike based on the heel-strikes' phase shifts as shown in **Figure 8**. The standard deviation and skewness relate to **Figure 9**. For flat ground based on 30 steps and while stair climbing (38 steps for the flat model and 31 steps for the stair climbing model).*

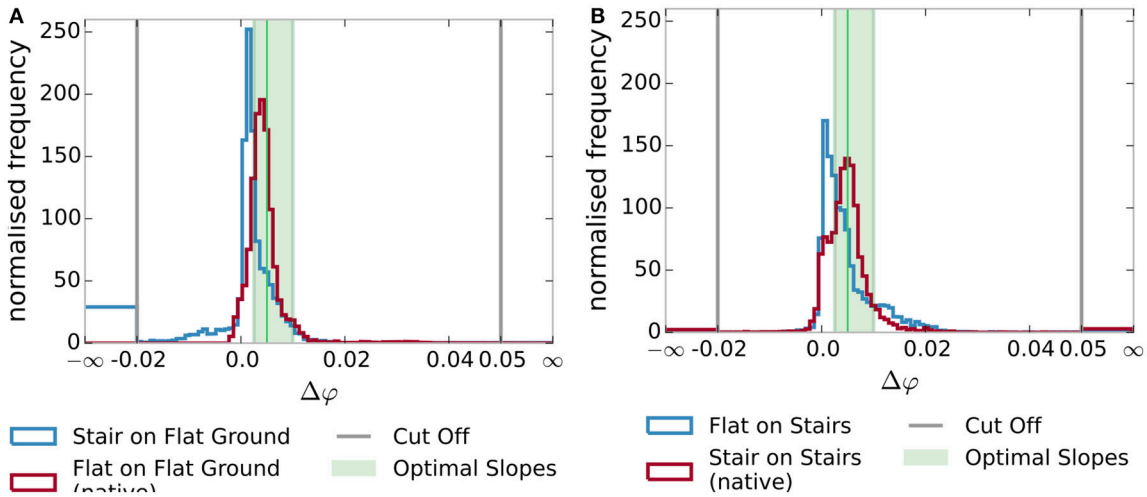


FIGURE 9 | Distribution of gait phase increment sizes $\Delta\varphi$. The native models (in red) have a distribution of increments centered near the ideal linear slope. The non-native models (in blue) show a stronger deviation from the ideal slope. The standard deviation and skewness are noted in **Table 1**. **(A)** Increments on flat ground. **(B)** Increments while stair climbing.

of significant negative changes for the stair climbing model on flat ground and an increase in larger values in the case of the flat ground model on stairs, i.e., less monotony and smoothness.

The ability to apply a damping pattern to steps of varying length is shown in **Figure 10**. As the abstract gait progress φ removes any time dependency from the input, changes in step duration and length are transparently handled. The red bars in **Figure 10** indicate unit lengths: the steps to the right are twice as fast as the ones to the left.

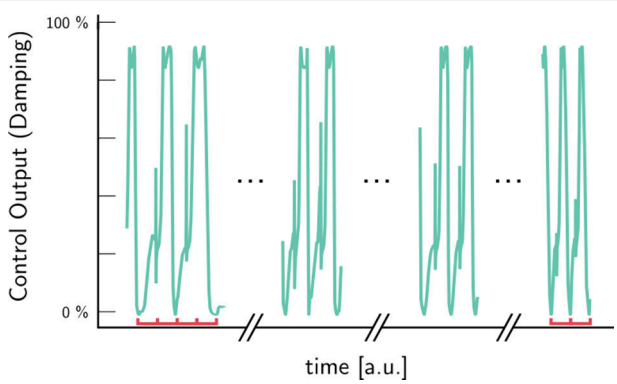


FIGURE 10 | Automatic adaptation to different step-lengths. The abstract gait progress φ removes any time dependency from the input. Step duration and length are transparently handled. The red bars indicate unit lengths: the steps to the right are twice as fast as the ones to the left.

3.2. Gait Selection

In this section, we test the hypothesis that a set of feed-forward single-gait controllers can be combined into to a multi-gait controller that enables the correct feed-forward controller to support a wide range of motions. Therefore, to evaluate the accuracy of gait recognition, we have to show that the gait recognition provides a correct result and that this result is available in time for the controller to react to gait changes.

The experiments include walking on flat ground, stair climbing, and descending stairs performed by a healthy subject. Prior to use, the gait models were trained with 146 steps for walking on flat ground, 35 steps for stair climbing, and 32 steps for descending stairs. The difference in training set sizes is due

to every stair run including steps of flat ground and the gaits on stairs being comparatively exhausting. Three independent recordings with 215 steps were used in the evaluation. These include gait transitions between 81 steps on flat ground, 64 steps mixing flat ground and stair climbing and 70 steps mixing flat ground and descending stairs. As the staircase used in the experiment is comprised of sequences of 5 stairs, each of the mixed runs includes the high number of 36 transition steps.

The development of gait certainty over the swing phase (**Figure 11**) shows that the gait for 83 % of steps was known at toe-off. The fraction of correct classification now increases toward above 94 % at heel-strike. This high accuracy is indicative of the fact, that most steps stem from step-sequences with the same gait. Furthermore, it indicates that many gait changes occur during swing phase.

The final classification accuracy, with ranges of certainty of at least 3 %, are plotted in **Figure 12** as confusion matrix between the user annotations in the rows and the controllers' classification in the columns. Note that the additional column *unknown gait* in the controllers' classifications, which counts cases, where the prediction errors are unacceptably high for all gait models. In these cases, the application of a fall back controller allows safe operation, for example, knee locking on ground contact, although it is most likely less comfortable. In general, the confusion matrix shows high classification rates between 87 % for descending stairs, 95 % for walking on flat ground, and 100 % for stair climbing. Furthermore, we see a number of steps, where the gait recognition was unable to differentiate or even mixing up walking on flat ground and descending stairs. The wrong classified steps are one transition step each for descending stairs and walking on flat ground.

The dynamics of knee and thigh angle (**Figures 13, 14**) show the transition step between descending stairs and walking on flat ground. It is easy to see that these steps are neither similar to one nor the other gait in 2D when plotting the angles over time, or thigh angle against knee angle. For the predicting models, which are working on a higher dimensional history, the dissimilarity is even more drastic. As a consequence, prediction errors are high for all models for this kind of gait transition step. It is to be expected, though, that many of these transitions fall into swing-phase transitions, where highly varying dynamics are possible and the actual control output is not that important for a device supporting mechanically.

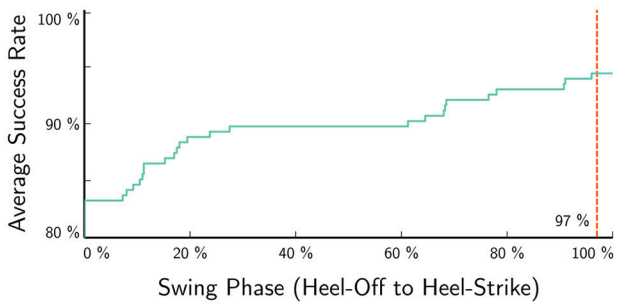


FIGURE 11 | The average success rate increases during swing phase, indicating that the classification matches the user annotation better the smaller the timescale of the prediction is. A high onset of more than 84 % indicates step sequences of unaltered gait. The red line to the right indicates the time, where the controller can still use gait information before heel-strike at ≈ 97 % swing phase (3 % *range of certainty* **Figure 7**), e.g., to successfully apply pre-damping.

4. DISCUSSION

The presented neural mechanisms set out as an adaptive orthosis controller, empowering users to control device behavior.

4.1. Gait Phase Representation

We implemented a neural single-gait controller to *individualize* gait support in terms of (1) the patient's gait dynamics with learning from observation and (2) direct user feedback with

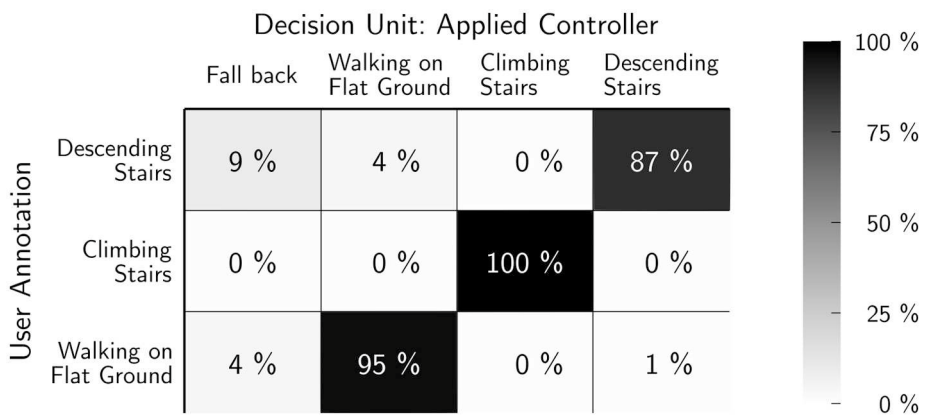


FIGURE 12 | Classification accuracy for a *range of certainty* of at least 3 %. The rows show the user annotation, the columns show the system's output. Note that the latter has the additional category "unkown/fall back" which is selected, when all predicting models generate high errors. Stair climbing has a high accuracy due to its unique phase relation, which leads to a 100 % success rate. Flat walking and stair descent have a higher overlap. Of 215 steps, only 2 transition steps were wrongly categorized (**Figures 13, 14**).

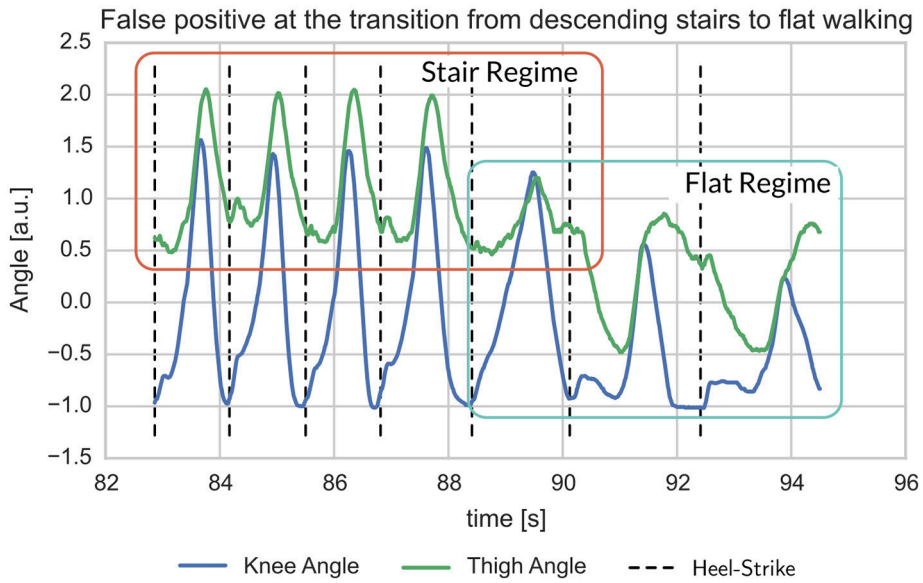


FIGURE 13 | Example for mismatch between user-label and gait detection at the transition from stair descent to flat walking. The transition step clearly deviates from earlier and following steps in that it shows mixed characteristics (**Figure 14**).

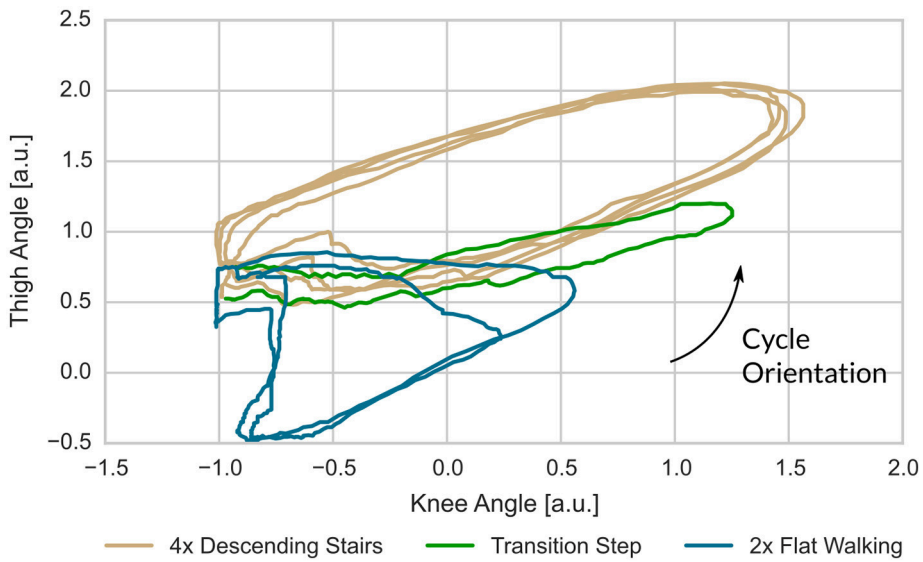


FIGURE 14 | For the label mismatch of **Figure 13**, the plot of thigh- over knee-angle shows that the transition step has quite unique properties. Note, that the predicting gait models work on a history going back 0.2 s, i.e., they are working on a higher dimensional space and thus can easier separate these gaits.

an interface for tuning, placing the patient in the loop. The gait phase abstracts gait dynamics and thereby removes dependencies on remaining abilities, except the ability to initiate motion. Furthermore, the gait phase removes the time domain from the sensory inputs. Thus, it transparently supports gaits of different speeds and step lengths (**Figure 10**) as well as standing; it provides immediate reactions to regular and critical events like stumbling. Variability in the training set enables use in varying environments such that a level walking controller supports even ground as well as slopes of several degrees (up to $\pm 15^\circ$ were tested but not presented here).

The presented user feedback is a minimal implementation, which allows to define an arbitrary damping function c in sufficient detail and allows the user to adapt c at run-time. It allows the users to understand the controller behavior in an experimental way: this way the users can develop an intuition of how changes to c modify the controllers behavior. Furthermore,

it simplifies the mapping from the gait phase to a valve position. Calibration and transformation are not necessary, as the user implicitly deals with these nonlinear operations. From the users' perspective, the user interface allows to define the level of support required. More important, their opinion is directly included in the controller's behavior. This inclusion of the patient's opinion concerns one of the top reasons for device abandonment see (Phillips and Zhao, 1993) and references therein.

Quantitative measurements verify instant reaction to motion, and high accuracy in timing and tracking of the patient's gait. We validated experimentally that the timing unit determines the heel-strike with high accuracy in the order of the sampling frequency. Furthermore, testing under the assumption that the recorded steps were ideally and steadily progressing, the timing unit was shown to approximate a linear progression of gait phase for trained gaits. Our generic approach of function approximation as representation for the control output provides intuitive tuning of the control output.

While the accuracy of gait phase tracking shows that gait models for quite different gaits can be learned, like flat ground walking and stair climbing, it also makes clear, that training leads to specialization of the feed-forward controller. To support movement in different environments, the controller has to be extended with controllers for multiple gaits in such a manner that free motion is possible.

4.2. Gait Selection

Specialization to one gait in the single-gait controller is overcome by a gait selection process based on predicting models. To support a gait, the controller therefore needs (1) a timing module with training samples, (2) the desired output shaping module, and (3) a predicting model which can be trained with the same samples as (1). This modular control approach overcomes *design* problems which typically restrict supported motion and the patient *target group*.

Based on the internal models' prediction errors, the gait selection swiftly chooses a single gait controller with the best fitting dynamics. Eighty-four percent of the steps in our experiments were already correctly identified at heel-off, most of them as part of a sequence of the same gait. But, the $\approx 84\,\%$ steps include at least 50 % of the 72 transition steps. The $\approx 13\,\%$ steps, which are identified between heel-off and heel-strike, indicate that gait recognition has to perform continuous. **Figures 13, 14** indicate that the gait dynamics is not bound to switch at any specific point and shows the flexibility and precision of the presented approach. For example, the initial step after standing phases is typically handled by the stair climbing module, which supports only vertical lift-off.

A fall-back controller, based on the ground contact sensing FSR, enables save operation in cases when gait dynamics are not matched by a model. The requirement for a fall-back controller is especially associated with transition steps, which often are singular events. The use of a history enables swift detection of changes in the motion. But at the same time, a gait change in the history will reduce the precision of predictions. Therefore, the history length not only determines the accuracy of gait prediction, but it also determines the frequency of changes, which can be tracked.

4.3. Advantages and Limitations

The greatest advantages of the presented approach lie in (1) its flexibility, as only the equipped sensors determine which gaits can be differentiated, (2) its implicit support for stumbling support, due to ground contact directly shifting the gait phase toward stance phase, (3) device independence, and (4) independence of remaining abilities, as long as circular motion can be initiated.

A difficulty in the evaluation of the presented approach lies in the handling of transition steps. As gait transitions can seemingly happen anytime, training data will not cover them in all possible variations. This singular nature of transition steps was not captured in the user annotations. Nonetheless, the results show that the controller is able to choose a gait with similar dynamics in many cases. In these cases, the user's annotation describes an intention, but not necessarily provides the best match to gait dynamics. In other words, the annotations are only valid for steps without transitions, for which excellent recognition rates could be seen even with 100 Hz sampling rate and 3 channels. For transition steps, a broad selection of training data will allow to address many transitions. For all other cases, the fallback controller has to provide save operation, i.e., guarantee support in stance phase, which can be achieved with the FSRs.

A general problem is the question of the number of supported movements. While three gaits were sufficient to control all motions during experiment sessions, it is still unclear how many gaits need to be supported for comfortable operation in everyday life. At the same time, support for more gaits might fill gaps in gait transitions as more independent motions are supported.

The presented control approach integrates the user into the tuning process and allows to directly model individual movements. We believe that this approach improves the handling of *gait deviations* and *device acceptance*. Still, the presented experiments have been conducted with a healthy subject. Thus, patient tests have to be undertaken to understand the interaction and consequences for real patients.

For patient tests, the user interface should be simplified. Instead of defining the damping function via a set of function values over a grid of support points, more suitable parameters should be chosen. A promising idea would be to focus on the start and end points of the support periods. Considering these together with the amplitude and the slope should provide an interface which is easy to understand, but even easier to handle.

4.4. Gait Phase Tracking in the Literature

Li et al. (2014) aim for a similar result by gait phase tracking on the contralateral leg. Inference of the controllers internal gait phase is based on the assumption of a constant phase shift to the ipsilateral leg. Besides practical issues with the instrumentation of the contralateral leg which directly impact comfort of use and visual appearance, it is important to note that constant phase shift can only be assumed in non-critical situations. Especially when stumbling or external forces disturb this phase shift, the contralateral leg does not reflect the device's state. The presented

approach always faithfully reflects the ipsilateral state, keeping the procedure of device application to the ipsilateral leg.

The first prosthetic device to reduce its wearers energetic cost of walking were presented in Malcolm et al. (2013) and Mooney et al. (2014). The effect was highest, when device activation was triggered at $\approx$ 43 %. This result indicates that control based on gait progress presents an interesting approach to pursue.

4.5. Multi-Gait Support in the Literature

Besides the here presented prediction error to invalidate gait models, many other approaches (Meyer, 1997; Mazzaro et al., 2005; Ding, 2008; Varol et al., 2008, 2010) are proposed. They are based on, for example, Gaussian mixture models or hidden Markov Models. Unfortunately, all of these studies discuss different selections of gaits, gait variations. For this reason, an actual performance comparison is difficult and would most likely be possible for image sequence based approaches (Mazzaro et al., 2005; Meyer, 1997), which are unfitting for prosthetic devices due to their outside-view on the walker. Furthermore, this study was working with a healthy walker. Still, average success rates between 83 and 94 % are comparable to vision based model invalidation approaches (Meyer, 1997; Mazzaro et al., 2005).

Besides image sequence based approaches, the literature mostly covers active prostheses. Due to space and weight limitations, active prostheses are of higher practical relevance than active orthoses, and therefore more present in the literature. Here, we will not cover technical differences, but focus on the controller.

Lawson et al. (2013) present a prosthesis controller for stair ascent and descent. The FSM architecture prevents the easy inclusion of other gaits and the missing support of level walking omits the region of high model overlap in this study. Sup et al. (2011) presents a hierarchy of FSMs, where one outer FSM with a slope estimator selects from three slope specific FSMs for 0, 5, and 10 °, respectively. While the fixation to slopes is incompatible with the gaits of this study, the addition of parameter estimation would provide beneficial input to the presented controller. A history based Gaussian mixture model differentiates standing and walking in Varol et al. (2008), selecting gait-specific FSM controllers on the fly. This study is based on seven signals sampled at 1, 000 Hz. An offline analysis was performed to reduce the dimensionality of the input for the Gaussian mixture models. In another step, the history length was increased, until the method provided a 100 % success rate. History frames of 50, 100, 200, or 400 samples were tried and finally a window size of 100 samples was selected with an overall delay of 430 ms. Later, this approach was extended to include sitting motion (Varol et al., 2010). The selection of sensors was described as task specific. In this study, the optimal delay was 500 ms.

All in all, the cross section of literature shows unique, and often incomparable gait selections and approaches. A similar approach with instantaneous selection was used in Varol et al. (2008, 2010) to differentiate standing, walking, and sitting motion. In contrast, the dynamics based gait tracking in our approach renders the recognition of standing superfluous. This focus on the device configuration simplifies data processing and needs neither explicit models of the device or gait nor expensive preprocessing. The presented work is based on only 3 sensors sampled at 100 Hz. Further improvements can be implemented with estimators of environmental parameters, sensors which provide differentiating inputs, or higher sampling frequencies. Especially with higher sampling frequencies, extensive optimization of the history and controller parameters, like amplification gains and weights could lead to significant improvements.

In comparison to biologically inspired modeling of modular motor learning and control, our control mechanism implements a partial function of the internal models for motor control proposed by Wolpert and Kawato (1998). The internal models are classified into three types: inverse internal model (the system calculates a motor command from a desired trajectory/state information), forward internal model (the system predicts sensory consequences from efference copies), and integrated internal model (the system integrates both inverse and forward models). In our case here, our shaping module acts as an inverse internal model that translates a user desired damping curve (i.e., desired trajectory) into a proper valve control command (i.e., motor command).

4.6. Outlook

Further optimization is possible with the many parameters in prediction, gait selection. Here, also other machine learning techniques can be employed (e.g., self-organizing learning of an adaptive resonance model Grossberg, 1987) is possible. The application of additional sensors, like torque sensors in the joints or IMUs, can improve differentiation.

Patients tests can show if the desired aims can be reached with the presented approach in real-world scenarios. Therefore, they are very important for future research.

The most interesting aspect is that our approach provides the building blocks for a completely self-learning controller. We demonstrated generalization of gait patterns, adaptation to changes in gait and in the environment as observed via gait changes. The user interface allows a user to adapt the support to individual needs. Still, at the stage presented here, the controller is not fully adaptive to a user in that it neither 1. automatically updates gait patterns, 2. damping output (lifetime of adjustments), nor does it learn new gaits on its own. Nonetheless, the modular structure allows to pursue these advanced aims. Additionally, other procedures (e.g., reinforcement learning or imitation learning) can be employed for offline training, where the subject provides the reward (good or bad) according to a given profile, or for fitting to the pattern of damping in human walking (Nakanishi et al., 2004).

Observation based training can be implemented at runtime, constructing and improving gait models continuously. A simple approach is to continuously add new samples to the training set and update the multi-layer perceptron networks's weights. The classification of recorded steps can be used to create new models, when new observations contradict existing models. In this way, bootstrapping of the controller can consist of a mostly generic model for walking on flat ground and an appropriate

fallback controller. Then, during everyday usage, the controller adapts to the patient and vice-versa, while the patient can always influence the control output. Suggestions for automatic tuning could be generated and tested in accordance with the patient, based on gait quality assessment in the controller. In this way, patients would be empowered to fit their own orthosis, hopefully improving trust into and the general opinion of the device.

ETHICS STATEMENT

The experiment was performed in accordance with the ethical standards laid down by the 1964 Declaration of Helsinki. We followed the relevant guidelines of the Germany Psychological Society according to which this experiment, given the conditions explained above, does not need explicit approval by an Ethics Committee (Document: 28.09.2004 DPG: Revision der auf die Forschung bezogenen ethischen Richtlinien).

AUTHOR CONTRIBUTIONS

J-MB, FW, and PM designed the research. J-MB developed the neural mechanisms. J-MB and PM carried out the experiments. J-MB and PM analyzed data. All authors wrote and reviewed the manuscript.

FUNDING

This research was supported by the BMBF-funded BFNT & BCCN Gottingen with grant numbers 01GQ0810 (project 3A) and 01GQ1005A (project D1), respectively.

ACKNOWLEDGMENTS

We thank our collaborators from Otto Bock, who provided us with the prototype and helped us with many insightful discussions.

REFERENCES

Bernhardt, K. A., Irby, S. E., and Kaufman, K. R. (2006). Consumer opinions of a stance control knee orthosis. *Prosthet. Orthot. Int.* 30, 246–256. doi: 10.1080/03093640600618818

Braun, J.-M., Wörgötter, F., and Manoonpong, P. (2014). "Internal models support specific gaits in orthotic devices," in *Mobile Service Robotics, Proceedings of the International Conference on Climbing and Walking Robots* (Poznan: World Scientific), 539–546.

Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*, Vol. 12. Cambridge: Cambridge University Press.

Ding, T. (2008). "A robust identification approach to gait recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008* (Anchorage, AK), 1–8.

Gailey, R., Allen, K., Castles, J., Kucharik, J., and Roeder, M. (2008). Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use. *J. Rehabil. Res. Dev.* 45, 15–29. doi: 10.1682/JRRD.2006.11.0147

Grossberg, S. (1987). Competitive learning: from interactive activation to adaptive resonance. *Cogn. Sci.* 11, 23–63.

Herr, H., Wilkenfeld, A., and Blaya, J. (2002). "Patient-adaptive prosthetic and orthotic leg systems," in *Proceedings of the 12th Nordic Baltic Conference on Biomedical Engineering and Medical Physics*, 123–128.

Irby, S. E., Bernhardt, K. A., and Kaufman, K. R. (2005). Gait of stance control orthosis users: the dynamic knee brace system. *Prosthet. Orthot. Int.* 29, 269–282. doi: 10.1080/03093640500238915

Irby, S. E., Bernhardt, K. A., and Kaufman, K. R. (2007). Gait changes over time in stance control orthosis users. *Prosthet. Orthot. Int.* 31, 353–361. doi: 10.1080/03093640601076909

Kaufman, K. R., Irby, S., Mathewson, J., Wirta, R., and Sutherland, D. (1996). Energy-efficient knee-ankle-foot orthosis: a case study. *J. Prosthet. Orthot.* 8, 79–85.

Lawson, B, E., Varol, H, A., Huff, A., Erdemir, E., and Goldfarb, M. (2013). Control of stair ascent and descent with a powered transfemoral prosthesis. *IEEE Trans. Neural Syst. Rehabil. Eng.* 21, 466–473. doi: 10.1109/TNSRE.2012.2225640

Lawson, B. E., Varol, H. A., and Goldfarb, M. (2011). "Ground adaptive standing controller for a powered transfemoral prosthesis," in *2011 IEEE International Conference on Rehabilitation Robotics (ICORR)* (Zurich), 1–6.

Li, J., Li, W., Li, C., Hu, H., Guo, H., Yu, S., et al. (2014). "Joint parameter mapping method for the control of knee prosthesis," in *Mobile Service Robotics, Proceedings of the International Conference on Climbing and Walking Robots* (Poznan: World Scientific), 45–52.

Malcolm, P., Derave, W., Galle, S., and De Clercq, D. (2013). A simple exoskeleton that assists plantarflexion can reduce the metabolic cost of human walking. *PLoS ONE* 8:e56137. doi: 10.1371/journal.pone.0056137

Mazzaro, M., Sznaier, M., and Camps, O. (2005). A model (in)validation approach to gait classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1820–1825. doi: 10.1109/TPAMI.2005.210

McKee, P. R., and Rivard, A. (2011). Biopsychosocial approach to orthotic intervention. *J. Hand Ther.* 24, 155–163. doi: 10.1016/j.jht.2010.08.001

Meyer, D. (1997). "Human gait classification based on hidden markov models," in *3D Image Analysis and Synthesis*, 139–146.

Mikelberg, R., and Reid, S. (1981). Spinal cord lesions and lower extremity bracing: an overview and follow-up study. *Spinal Cord* 19, 379–385.

Mills, K., Blanch, P., Chapman, A. R., McPoil, T. G., and Vicenzino, B. (2010). Foot orthoses and gait: a systematic review and meta-analysis of literature pertaining to potential mechanisms. *Br. J. Sports Med.* 44, 1035–1046. doi: 10.1136/bjsm.2009.066977

Milton, J., Cabrera, J. L., Ohira, T., Tajima, S., Tonosaki, Y., Eurich, C. W., et al. (2009). The time-delayed inverted pendulum: implications for human balance control. *Chaos* 19:026110. doi: 10.1063/1.3141429

Mooney, L., Rouse, E., and Herr, H. (2014). Autonomous exoskeleton reduces metabolic cost of human walking during load carriage. *J. Neuroeng. Rehabil.* 11:80. doi: 10.1186/1743-0003-11-80

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robot. Auton. Syst.* 47, 79–91. doi: 10.1016/j.robot.2004.03.003

Nissen, S. (2003). *Implementation of a Fast Artificial Neural Network Library (Fann).* Technical report, Department of Computer Science University of Copenhagen (DIKU). Available online at: http://fann.sf.net

Park, J., and Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Comput.* 3, 246–257.

Phillips, B., and Zhao, H. (1993). Predictors of assistive technology abandonment. *Assist. Technol.* 5, 36–45.

Robinson, V., Sansam, K., Hirst, L., and Neumann, V. (2010). Major lower limb amputation–what, why and how to achieve the best results. *Orthop. Trauma* 24, 276–285. doi: 10.1016/j.mporth.2010.03.017

Sup, F., Varol, H., and Goldfarb, M. (2011). Upslope walking with a powered knee and ankle prosthesis: initial results with an amputee subject. *IEEE Trans. Neural Syst. Rehabil. Eng.* 19, 71–78. doi: 10.1109/TNSRE.2010.2087360

Varol, H., Sup, F., and Goldfarb, M. (2008). "Real-time gait mode intent recognition of a powered knee and ankle prosthesis for standing and walking," in *2nd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics, 2008* (Scottsdale, AZ), 66–72.

Varol, H., Sup, F., and Goldfarb, M. (2010). Multiclass real-time intent recognition of a powered lower limb prosthesis. *IEEE Trans. Biom. Eng.* 57, 542–551. doi: 10.1109/TBME.2009.2034734

Winter, D. A. (1995). Human balance and posture control during standing and walking. *Gait Posture* 3, 193–214.

Wolpert, D. M., and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Netw.* 11, 1317–1329.

Yakimovich, T., Lemaire, E. D., and Kofman, J. (2009). Engineering design review of stance-control knee-ankle-foot orthoses. *J. Rehabil. Res. Dev.* 46, 257–268. doi: 10.1682/JRRD.2008.02.0024

Zlatnik, D., Steiner, B., and Schweitzer, G. (2002). Finite-state control of a trans-femoral (tf) prosthesis. *IEEE Trans. Control Syst. Technol.* 10, 408–420. doi: 10.1109/87.998030

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Braun, Wörgötter and Manoonpong. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 29 November 2016
doi: 10.3389/fnbot.2016.00019

Biomechanical Reconstruction Using the Tacit Learning System: Intuitive Control of Prosthetic Hand Rotation

Shintaro Oyama[1], Shingo Shimoda[2], Fady S. K. Alnajjar[2], Katsuyuki Iwatsuki[1], Minoru Hoshiyama[3], Hirotaka Tanaka[4] and Hitoshi Hirata[1]*

[1] Department of Hand Surgery, Nagoya University School of Medicine, Nagoya, Japan, [2] Brain Science Institute-TOYOTA Collaboration Center, RIKEN, Nagoya, Japan, [3] Brain and Mind Research Center, Nagoya University School of Medicine, Nagoya, Japan, [4] Department of Rehabilitation, Chubu Rosai Hospital, Nagoya, Japan

Background: For mechanically reconstructing human biomechanical function, intuitive proportional control, and robustness to unexpected situations are required. Particularly, creating a functional hand prosthesis is a typical challenge in the reconstruction of lost biomechanical function. Nevertheless, currently available control algorithms are in the development phase. The most advanced algorithms for controlling multifunctional prosthesis are machine learning and pattern recognition of myoelectric signals. Despite the increase in computational speed, these methods cannot avoid the requirement of user consciousness and classified separation errors. "Tacit Learning System" is a simple but novel adaptive control strategy that can self-adapt its posture to environment changes. We introduced the strategy in the prosthesis rotation control to achieve compensatory reduction, as well as evaluated the system and its effects on the user.

Methods: We conducted a non-randomized study involving eight prosthesis users to perform a bar relocation task with/without Tacit Learning System support. Hand piece and body motions were recorded continuously with goniometers, videos, and a motion-capture system.

Findings: Reduction in the participants' upper extremity rotatory compensation motion was monitored during the relocation task in all participants. The estimated profile of total body energy consumption improved in five out of six participants.

Interpretation: Our system rapidly accomplished nearly natural motion without unexpected errors. The Tacit Learning System not only adapts human motions but also enhances the human ability to adapt to the system quickly, while the system amplifies compensation generated by the residual limb. The concept can be extended to various situations for reconstructing lost functions that can be compensated.

Keywords: myoelectric prosthesis, artificial intelligence, biomechanical function reconstruction, motor control, magnetoencephalography, interactive musculoskeletal modeling analysis, muscle, sensory synergy

OPEN ACCESS

Edited by:
Christian Tetzlaff,
Max Planck Institute for Dynamics and Self-Organization, Germany

Reviewed by:
Jing Jin,
East China University of Science and Technology, China
Jan-Matthias Braun,
University of Göttingen, Germany

****Correspondence:***
Shintaro Oyama
oyama.shintaro@c.mbox.nagoya-u.ac.jp

Received: *16 August 2016*
Accepted: *16 November 2016*
Published: *29 November 2016*

Citation:
Oyama S, Shimoda S, Alnajjar FSK, Iwatsuki K, Hoshiyama M, Tanaka H and Hirata H (2016) Biomechanical Reconstruction Using the Tacit Learning System: Intuitive Control of Prosthetic Hand Rotation. Front. Neurorobot. 10:19. doi: 10.3389/fnbot.2016.00019

INTRODUCTION

When we lose a functional part in our body (e.g., reaching out, walking, trunk control, breathing, watching, etc.), we not only lose functional output but also sensory feedback. Every biomechanical movement is a result of computations in the central-nervous-system (CNS) and at the same time, consecutive sensory feedback prediction and modification of motor behavior goes on in the cerebellum, allowing us to accomplish natural motion, and construct changes in response to the external environment (Brooks et al., 2015). Therefore, reconstruction of lost biomechanical function should not only include fine motor control but also dense sensory feedback that precisely, bi-directionally, and with high frequency communicates with the CNS. However, even the most advanced neuromotor reconstruction technology has not accomplished this communication, and lacks appropriate feedback for natural function. Furthermore, construction of a practical and ergonomic mechanical system that adapts to environmental changes within seconds is difficult due to lack of flexibility in current artificial machine learning.

One typical challenge of reconstructing lost function is the functional hand prosthesis. These are widely used in reconstruction on forearm amputees and congenital forearm deficient individuals for restoring their ability to reach and grasp. Among these, body power and myoelectric prostheses are widely used for motor control. In the past, body powered prosthesis were advantageous in cost, intuitiveness and sensory feedback, but not in function. Thus, a great effort was required to accomplish more function and natural movement in myoelectric prosthesis (Ciancio et al., 2016).

Developments in technology over the past few decades has improved control on multiple functions, with a primary focus on minimizing user burden and increasing prosthesis' function. Nevertheless, increasing the number of myoelectric input channels resulted in non-physiological muscle activation that required exhaustive training (Schulz et al., 2005). Target muscle re-innervation (Kuiken et al., 2007) may be one solution, but is too invasive and less beneficial for trans-radial amputees which represent the largest proportion of individuals with upper extremity deficiency (Hahne et al., 2012). The development of pattern recognition and machine learning techniques of electromyography (EMG) signals increased the number of degrees of freedom (DOFs) while keeping the number of utilized electrodes low. However, this technique has a critical limitation of low adaptability to environmental changes (Ciancio et al., 2016).

Meanwhile, a large number of studies have used the brain's plasticity to quickly adapt and reorganize cross-modal sensory integration for sensory feedback reconstruction. Since most of the work focuses on tactile feedback for adjusting grip force, it is still a challenge to reconstruct natural sensory feedback and mimic natural control. Recently, a few studies have reported increased sensory information density by neural implants (Ciancio et al., 2016); however, neurophysiological studies have indicated that position in space is estimated by integrating information from multiple sensory inputs rather than direct input. Moreover, as this integrated feedback is noise-robust, useful and cost-effective, adding appropriate sensory integration may result in better reconstruction (Alnajjar et al., 2015).

In our natural motion learning, we use two different modes, i.e., explicit and tacit learning. The former occurs with learner's awareness, while the latter takes place subliminally. When we perform a motor skill, there is a variety in the status of our neuromotor situation, which is subliminal and highly coordinated to express low dimensional motion. The key to a natural control strategy is management of this inherent redundancy in the musculoskeletal system mediated by a high number of DOFs with low dimensional outputs (Metzger et al., 2012).

Recently, several studies have shown that muscle synergy is like a neural strategy that the CNS has adopted to simplify the control of our redundant musculoskeletal system. Additionally, the importance of integrating environmental inputs into suitable low-dimensional signals before sending them to the CNS for simplified control have been documented (Alnajjar et al., 2015). Yet the neural dynamics inside the CNS have not been investigated in detail. Shimoda introduced a biological self-regulatory adaptive control strategy called "Tacit Learning System" (TLS) for posture control with self-sufficiency. This system is designed for unsupervised acquisition of skills or creation of new behavioral structures for adapting to environmental changes. Signal accumulation is a key factor for "Tacit Learning" in the adaptation process and primitive behaviors composed of several reflex actions are gradually tuned into suitable behaviors for the environment (Shimoda and Kimura, 2010; Shimoda et al., 2012, 2013).

Shimoda and his team have succeeded in controlling 36 DOFs in a humanoid bipedal locomotive robot using this TLS and demonstrated a wide adaptation capability to a redundant motor-skeletal system along with robustness to environmental changes compared to conventional machine learning algorithms (Shimoda et al., 2012). We thus hypothesized that introducing TLS into the biomechanical structure as a subsystem will integrate it with muscle synergy to control implicit motion with adaptation to environmental changes, allowing the user to concentrate on explicit tasks like grasping in myoelectric hand prosthesis. Clinically, a compensatory strategy to the rotation function of a lost wrist, involves using proximal residual limbs to achieve the necessary motion, result in increased burden on users that limit prosthesis usage (Metzger et al., 2012). This rotation function of reaching is an example of implicit motion. Consequently, we performed experiments to evaluate the efficacy of TLS in a prosthesis hand model, by appointing the system to regulate wrist rotation to minimize redundant compensatory motion as a biomimetic regulatory system while performing reaching tasks.

MATERIALS AND METHODS

In this study, a non-randomized experiment was conducted to evaluate efficacy of the TLS and its effects on the central nervous system during the prosthesis control tasks.

Prosthesis Efficacy Evaluation in Bar Relocation Tasks

Seven men and one woman participated after giving informed consent. All participants were below elbow amputees, and experienced users of the conventional one-degree (hand open and close) myoelectric hand prosthesis. **Table 1** shows the participants' demographic data.

Each participant's prosthesis handpiece was exchanged with the TLS handpiece and their remaining arm sockets were used in the trials. The open/close signal detector on the handpiece was connected to sensors in the socket to allow the participants to control hand motions as usual. Since we could not find adequate forearm rotation tasks for hand prosthesis in the past literature, we placed three plastic bars (3 cm diameter and 10 cm length with the central 3 cm part covered with Velcro tape to increase grasp) horizontally on a table. Participants sat in front of the table, reached out to hold the bars, placed them vertically and then back to horizontally three times (**Figure 1**). This exercise counted as one trial. Initialization of the system was performed as a pre-trial. Participants were instructed to stay still for 5 s with their shoulders at 0° flexion, rotation, and abduction, along with elbows at 0° flexion. Subsequently, they were instructed to repeat the trials until the rotational support of hand prosthesis made no more improvement. Twenty trials were done in approximately 10 min, and this was sufficient for every patient to achieve convergence of the parameters. Angles of shoulder joints derived from three goniometers placed on the participant's shoulder (flexion, rotation, and abduction) were monitored and fed back to the TLS. Participants' movements were recorded with a computer vision based human body motion capture tracking system (Section Tacit Learning Handpiece and Data Preprocessing) First person sight video (FPV) (Video 1, 2) recording was performed by the camera (HERO, GoPro, Inc., CA, USA) attached to the prosthesis socket. We conducted descriptive type questionnaires to determine participant satisfaction and how effective the participants felt the system was.

Tacit Learning Handpiece and Data Preprocessing

The system consisted of three goniometer sensors to measure angles of shoulder flexion (θ_1), horizontal flexion (θ_2), rotation (θ_3), and a handpiece with two actuators (rotation and grip) (**Figure 2**). One actuator was for handpiece wrist rotation. Rotation angle θ_r was a desired angle of prosthesis wrist rotation, controlled by a low-level controller embedded in the hardware. The other actuator was for grip with an on-off control provided by surface EMG sensing which is commonly used by commercial prosthesis. When the shoulder angles exceeded pre-defined threshold values θ_t (the value found at unnatural postures), the system tuned the control gain, accumulating extremity joint angles. The control and adaptation laws were defined as follows:

$$\theta_r = k\Theta - \dot{\theta}_r \tag{1}$$

$$k = \int q dt \tag{2}$$

$$q = \begin{cases} \Theta & |\Theta| \geq \theta_t \\ 0 & |\Theta| < \theta_t \end{cases} \tag{3}$$

$$\Theta = k_1\theta_1 + k_2\theta_2 + k_3\theta_3 \tag{4}$$

When a linear combination of residual upper limb joint angles Θ in Expression (4) exceeded the settled threshold angle θ_t in Expression (3), primary reflex modulated rotatory assistance angle θ_r depending on Θ in Expressions (1) and (2). Expression (1) was a speed control component of rotation. A previous mathematical study suggests that biological arm kinematics are optimized by total energy expenditure (Berret et al., 2008), which is positively correlated to the total joint angle Θ. Thus, we determined the control law of system as minimization of Θ. In this experiment, we set $\theta_t = 1$, $k_1 = 0.1$, $k_2 = 0.1$, and $k_3 = 0.5$ as the initial values.

Motion Capture System

Kinematic patterns of the participants' movements were captured with a motion capture system (Workstation 5.2.4, VICON). Twenty-four markers (spheres covered with reflective tape) were attached to various parts of the participant's body and prosthesis prior to the experiment. The motion capture system consisted of six cameras, which tracked and reconstructed the motion of each of the recorded markers in 3D space.

Data Analysis

We focused on the tacit learning rotational control of prosthesis on this study.

Hence, we computed system energy consumption by using the software for Interactive Musculoskeletal Modeling (SIMM, MusculoGraphics, Inc., Santa Rosa, California, USA). It is a graphical software system for developing and analyzing models of musculoskeletal structures, and performs inverse dynamics calculations from motion capture data (Delp and Loan, 1995; Neptune et al., 2008). It creates a musculoskeletal model consisting of representations of bones, muscles, and ligaments by calculating the joint moments. In this study, we used a standardized musculoskeletal model calculated from the participants' body weight, height, and sex. Pre-trial system energy in all participants was normalized as one.

TABLE 1 | Demographic data of participants.

No.	Age (y)	Sex	Side	Duration of myoelectric prosthesis use	Device type
1	51	Male	Right	5 years	Ottobock8E44=6+10S17+10V38
2	40	Male	Right	8 years	Ottobock8E38=9
3	46	Female	Right	12 years	Ottobock8E38=6
4	41	Male	Right	6 months	Ottobock8E44=6+10S17+10V38
5	29	Male	Right	2 months	Ottobock8E44=6+10S17+10V38
6	52	Male	Right	3 years, 6 months	Ottobock8E38=6+10S17
7	33	Male	Left	1 year	Ottobock8E38=6+10S17+10V38
8	74	Male	Right	32 years	Ottobock8E38=6

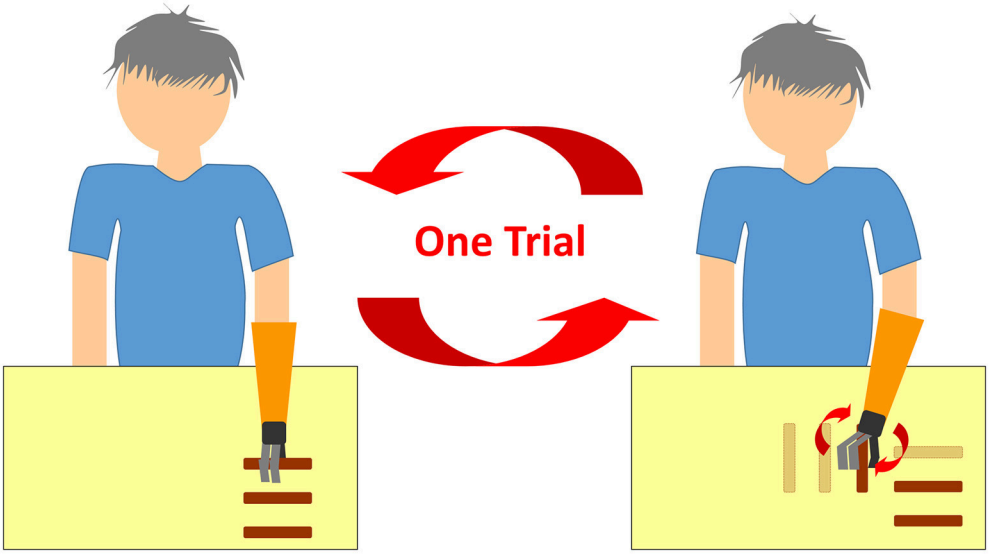


FIGURE 1 | A schematic figure of the trial. After moving three bars vertically, the participants were instructed to place these three bars back to where they were horizontally.

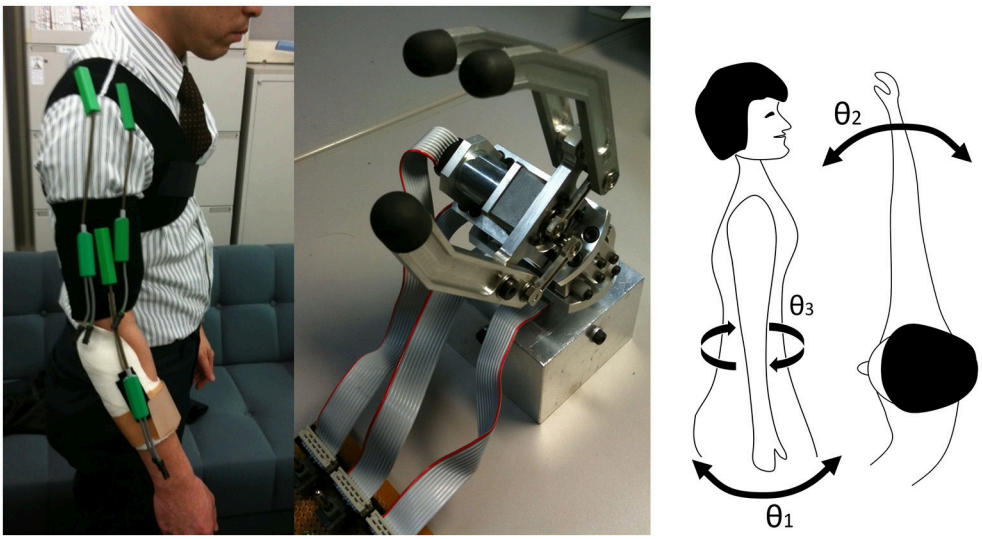


FIGURE 2 | Three goniometers were attached as shown (left) and linked to the handpiece (center) for measuring the upper extremity joint angle ($\theta_1, \theta_2, \theta_3$, right).

RESULTS

All participants successfully completed their assigned tasks. Online video (Video 1, 2, 3) shows participant 3 working on his tasks. "After learning" represents 20 trials after the first one. Adaptation advanced in both wearer and prosthesis in a short while as shown in the videos (Video 1: Without TLS assistance. Video 2: After twenty trials). After 20 trials, the shoulder rotation angle (θ_3) decreased in all participants as shown in **Figure 3**. Total system energy estimated by SIMM decreased in five out of six patients (**Figure 4**). Energy estimation was not possible in participants 7 and 8 due to failure of the motion-capture marker. **Figure 5** shows changes in the actual estimated system energy data during trials in participant 1. The graphs show system energy before and after TLS learning. The compensation rotation angle of shoulder [Θ in Section Tacit Learning Handpiece and Data Preprocessing, Expression (4)] in participant 1 decreased after TLS learning as shown in **Figure 6**. Seven out of eight participants were comfortable with TLS assistance. No participant required special training before the trials. After TLS learning of bar rotational tasks, participant 8 volunteered to open two types of drawers and turn the oven indicator (Video 4). **Figure 7** shows rotation angle of the prosthesis wrist during the tasks. Although TLS did not experience any of these tasks, it provided good

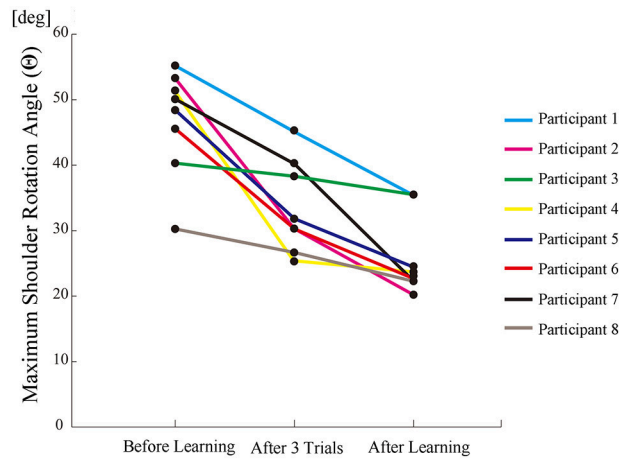


FIGURE 3 | The maximum shoulder rotation angle (as compensation movement for forearm rotation tasks) decreased after trials.

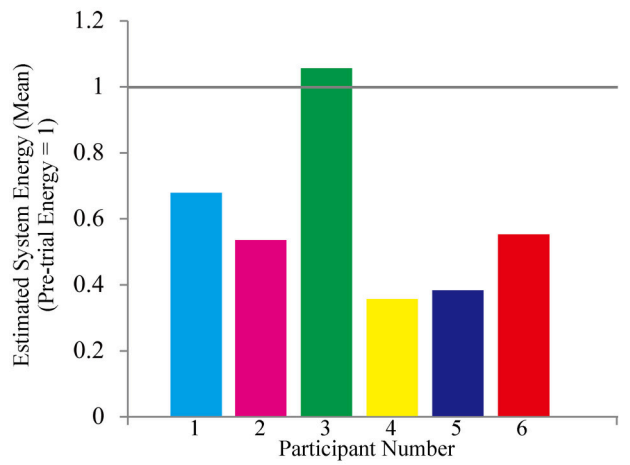


FIGURE 4 | Estimated mean system energy decreased significantly in five out of six participants.

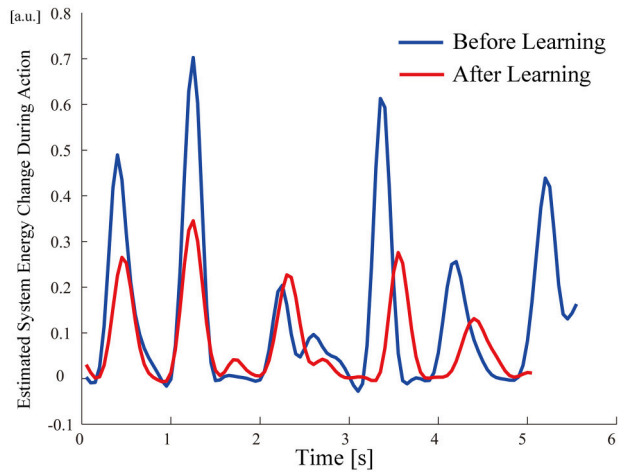


FIGURE 5 | Estimated system energy change during trials before/after TLS learning in participant 1.

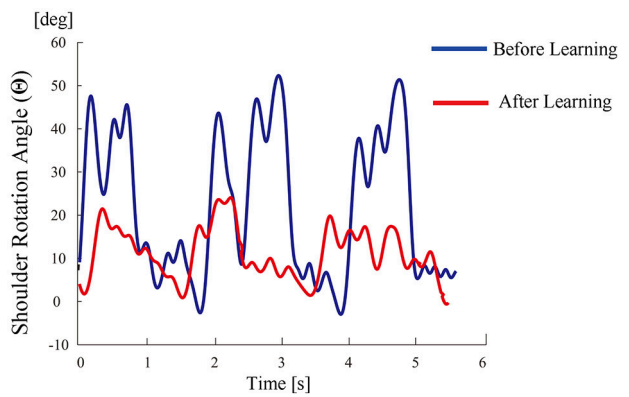


FIGURE 6 | Shoulder rotation angle (as compensation movement for forearm rotation tasks) during tria/FCIs before/after TLS learning in participant 1.

assistance and showed generalized performance for rotational support despite changes in arm posture. This participant was also satisfied with the intuitiveness of TLS support as determined by the descriptive type questionnaires.

DISCUSSION

A good hand prosthesis should reconstruct the original dexterity of human hands. While far from complete, in this endeavor we replicated one of the most complex biomechanical structures. Improvements in EMG signal analysis (Tenore et al., 2009), Targeted Muscular Reinnervation (TMR) including sensory feedback (Kuiken et al., 2007; Ohnishi et al., 2007; Li et al., 2010), brain interface (Yanagisawa et al., 2011), peripheral nerve interface (Navarro et al., 2005), and new training systems (Pilarski et al., 2011) were invented, but these methods required a certain period of special training or special surgery invasions. Furthermore, none of these methods satisfied the contrary demands for intuitiveness, multi-functionality, and cost. Moreover, due to the lack of flexibility in present control methods to adapt with environmental changes including complex nature of the bio-signals, repeated calibration is often required by patients and physiotherapists (Ciancio et al., 2016).

The present work focuses on reconstructing each joint's movement, but not muscle synergy. Reconstruction of muscle synergy does not involve isolating and reorganizing bio-kinematic outputs from residual function, but expanding muscle synergy in a biological way. In other words, an optimization algorithm should be introduced for mimicking human-like motion and finding the natural output from residual limb to cope with. Shimoda described that human kinematic output is unpredictable for machines with a model-based strategy that does not represent certain posture situations (i.e., forearm rotations, elbow flexion/extension) due to intrinsic fragility (Shimoda and Kimura, 2008). To cope with all motions, it is necessary to model all possible posture changes and device control actions

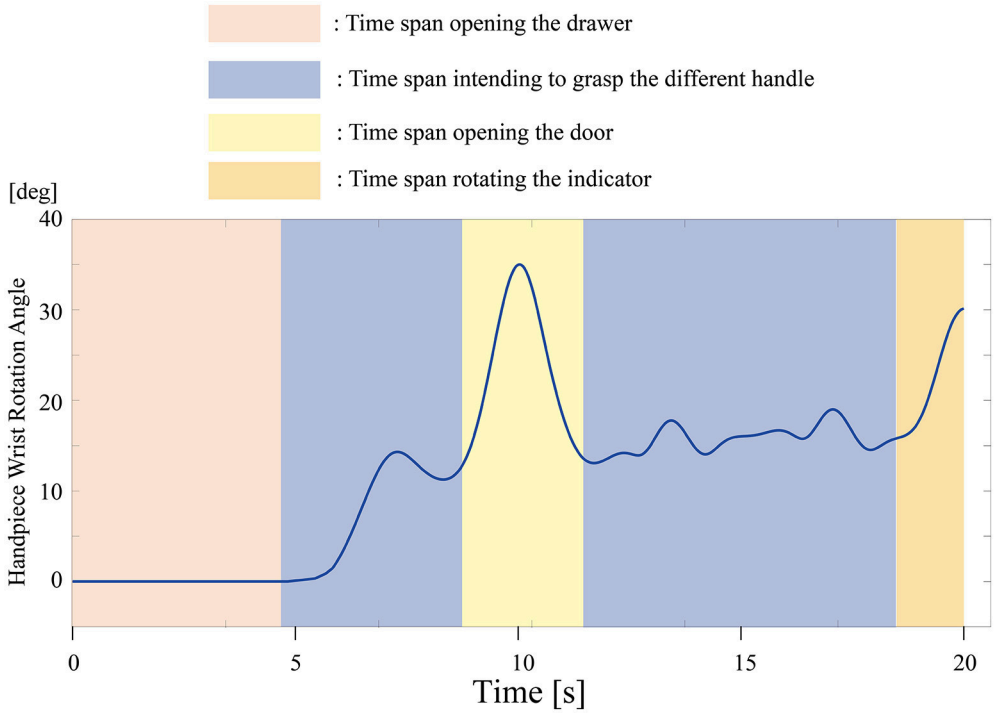


FIGURE 7 | Wrist rotation angle of the prosthesis during the drawer and indicator tasks. Rotational support was efficient, even though tacit learning TLS did not experience the tasks.

in every model. To solve these issues, various types of biomimetic and self-organizing learning methods including artificial neural networks have been proposed, but the capability of current learning methods to adapt to unknown situations is not sufficient in terms of learning speed and the level of generalization.

The "Tacit Learning System" introduced by Shimoda has two main advantages for controlling the prosthesis compared to other control methods: learning speed and a simple, inexpensive system with intrinsic robustness (Shimoda et al., 2012, 2013). Furthermore, as Alnajjar described, this controller has a role in reduction of sensory stimulus dimension. This is called "sensory synergy" in contrast to muscle synergy. They defined "sensory synergy" as "a group of weighted sensory inputs whose function is to provide the quality of the resulting motion as feedback to the CNS through a single synergy recruitment signal in order to facilitate the generation of the next command, thus accelerating the search time for the optimal muscle synergy." In particular, in TLS, the controller modulates sensory synergies contributed by acquired sensory signals and inferred artificial sensory synergies into motor commands. Consequently, activated motor commands of the prosthesis enable intuitive motor control by the wearer and simultaneous confirmation with visual feedback. In short, the output of sensory synergy is used as an input to both the CNS and the TLS, and control signals for the prosthesis device are created through motor synergy that combines signals from the CNS and the prosthesis device (Alnajjar et al., 2015).

Our results from the bar relocation experiment convinced us that this system has high affinity toward the CNS. It was easy to add on the conventional system, required no special training, reduced users' burden and is low-cost. The level of satisfaction was high.

Recently, we reported a case report from a magnetoencephalography study on the effect of the TLS system on CNS. This report showed that the coherence value among sensorimotor-related cortices in the dominant hemisphere increased only while watching a video of oneself using the prosthesis with TLS support and vice versa. This result is no more than a showcase, but we are preparing for a future clinical study evaluating the effect of the "Tacit Learning System" prosthesis on CNS based on the evidence of this basic study.

A limitation of this study is that we tested this system on limited tasks and thus, it is still in the prototype phase currently. We tried several learning motions and determined that various motions could progress the learning in a similar way to that shown in the results. This robustness is justified with the experiments of motion generalizations by the drawer opening task. In cases where less extreme motions were used in the training sessions, the learning speed was slow, and it took many trials to learn the appropriate behaviors. For this study, we choose a simple relocation task to control the learning environment for all participants and to compare the differences in the learning process. The users did not try the system in real life tasks like cooking, housework etc. However, results of additional tasks performed by participant 8 in an additional experiment suggest robustness of the system in different situations. Short battery life is also a concern. The system continuously senses upper limb motions and tries to adjust prosthesis positions at all times, so battery drainage is three to five times greater than

the conventional systems. Setting the threshold adjustment may be a solution. Higher threshold to a TLS support may increase battery life but may result in reduced support, which needs to be considered according to the users' lifestyle.

In summary, we introduced a novel "Tacit Learning System," a self-regulatory strategy in a myoelectric prosthesis, to control wrist rotation and confirmed its efficacy in conventional type myoelectric prosthesis users. We infer that TLS showed the ability to recover the lost function by adjusting compensatory overreaction generated by residual function. Theoretically, it can be used for recovering functions in other situations such as lower limb amputation, palsy in association with functional electric stimulation, or even in ventilation failure if residual function is present.

HUMAN AND ANIMAL RIGHTS AND INFORMED CONSENT

This study was approved by the Ethics Committee of the Nagoya University, School of Medicine (Approval number 2012-0145). All participants gave informed consent before enrolling in the study, and all procedures were performed in accordance with ethical standards of the responsible committee on human experimentation and with the Helsinki Declaration of 1975, revised in 2000 and 2008.

AUTHOR CONTRIBUTIONS

Conceived and designed the experiment: SO, SS, KI, and HH. Performed and analyzed the data for experiment: SO, SS, FA, and KI. "Tacit Learning System" design, settings, and construction: SS and FA. Participant registrations: KI and HT. Wrote the paper: SO, SS, KI, MH, and HH.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2016.00019/full#supplementary-material

REFERENCES

Alnajjar, F., Itkonen, M., Berenz, V., Tournier, M., Nagai, C., and Shimoda, S. (2015). Sensory synergy as environmental input integration. *Front. Neurosci.* 8:436. doi: 10.3389/fnins.2014.00436

Berret, B., Darlot, C., Jean, F., Pozzo, T., Papaxanthis, C., and Gauthier, J. P. (2008). The inactivation principle: mathematical solutions minimizing the absolute work and biological implications for the planning of arm movements. *PLoS Comput. Biol.* 4:e1000194. doi: 10.1371/journal.pcbi.1000194

Brooks, J. X., Carriot, J., and Cullen, K. E. (2015). Learning to expect the unexpected: rapid updating in primate cerebellum during voluntary self-motion. *Nat. Neurosci.* 18, 1310–1317. doi: 10.1038/nn.4077

Ciancio, A. L., Cordella, F., Barone, R., Romeo, R. A., Bellingegni, A. D., Sacchetti, R., et al. (2016). Control of prosthetic hands via the peripheral nervous system. *Front. Neurosci.* 10:116. doi: 10.3389/fnins.2016.00116

Delp, S. L., and Loan, J. P. (1995). A graphics-based software system to develop and analyze models of musculoskeletal structures. *Comput. Biol. Med.* 25, 21–34.

Hahne, J. M., Rehbaum, H., Biessmann, F., Meinecke, F. C., Muller, K.-R., Jiang, N., et al. (2012). "Simultaneous and proportional control of 2D wrist movements with myoelectric signals," in *IEEE International Workshop on Machine Learning for Signal Processing* (Satander: IEEE), 1–6. doi: 10.1109/MLSP.2012.6349712

Kuiken, T. A., Miller, L. A., Lipschutz, R. D., Lock, B. A., Stubblefield, K., Marasco, P. D., et al. (2007). Targeted reinnervation for enhanced prosthetic arm function in a woman with a proximal amputation: a case study. *Lancet* 369, 371–380. doi: 10.1016/S0140-6736(07)60193-7

Li, G., Schultz, A. E., and Kuiken, T. A. (2010). Quantifying pattern recognition-based myoelectric control of multifunctional transradial prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* 18, 185–192. doi: 10.1109/TNSRE.2009.2039619

Metzger, A. J., Dromerick, A. W., Holley, R. J., and Lum, P. S. (2012). Characterization of compensatory trunk movements during prosthetic upper limb reaching tasks. *Arch. Phys. Med. Rehabil.* 93, 2029–2034. doi: 10.1016/j.apmr.2012.03.011

Navarro, X., Krueger, T. B., Lago, N., Micera, S., Stieglitz, T., and Dario, P. (2005). A critical review of interfaces with the peripheral nervous system for the control of neuroprostheses and hybrid bionic systems. *J. Peripher. Nerv. Syst.* 10, 229–258. doi: 10.1111/j.1085-9489.2005.10303.x

Neptune, R. R., Sasaki, K., and Kautz, S. A. (2008). The effect of walking speed on muscle function and mechanical energetics. *Gait Posture* 28, 135–143. doi: 10.1016/j.gaitpost.2007.11.004

Ohnishi, K., Weir, R. F., and Kuiken, T. A. (2007). Neural machine interfaces for controlling multifunctional powered upper-limb prostheses. *Expert Rev. Med. Devices* 4, 43–53. doi: 10.1586/17434440.4.1.43

Pilarski, P. M., Dawson, M. R., Degris, T., Fahimi, F., Carey, J. P., and Sutton, R. S. (2011). Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. *IEEE Int. Conf. Rehabil. Robot.* 2011:5975338. doi: 10.1109/ICORR.2011.5975338

Schulz, S., Pylatiuk, C., Reischl, M., Martin, J., Mikut, R., and Bretthauer, G. (2005). A hydraulically driven multifunctional prosthetic hand. *Robotica* 23, 293–299. doi: 10.1017/S0263574704001316

Shimoda, S., and Kimura, H. (2008). Neural computation scheme of compound control: tacit learning for bipedal locomotion. *SICE J. Control. Meas. Syst. Integr.* 1, 275–283. doi: 10.9746/jcmsi.1.275

Shimoda, S., and Kimura, H. (2010). Biomimetic approach to tacit learning based on compound control. *IEEE Trans. Syst. Man. Cybern.* 40, 77–90. doi: 10.1109/TSMCB.2009.2014470

Shimoda, S., Yoshihara, Y., Fujimoto, K., Yamamoto, T., Maeda, I., and Kimura, H. (2012). "Stability analysis of tacit learning based on environmental signal accumulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura: IEEE), 2613–2620. doi: 10.1109/IROS.2012.6385631

Shimoda, S., Yoshihara, Y., and Kimura, H. (2013). Adaptability of tacit learning in bipedal locomotion. *IEEE Trans. Auton. Ment. Dev.* 5, 152–161. doi: 10.1109/TAMD.2013.2248007

Tenore, F. V. G., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R., and Thakor, N. V. (2009). Decoding of individuated finger movements using surface electromyography. *IEEE Trans. Biomed. Eng.* 56, 1427–1434. doi: 10.1109/TBME.2008.2005485

Yanagisawa, T., Hirata, M., Saitoh, Y., Goto, T., Kishima, H., Fukuma, R., et al. (2011). Real-time control of a prosthetic hand using human electrocorticography signals. *J. Neurosurg.* 114, 1715–1722. doi: 10.3171/2011.1.JNS101421

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2016 Oyama, Shimoda, Alnajjar, Iwatsuki, Hoshiyama, Tanaka and Hirata. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 11 July 2017
doi: 10.3389/fnbot.2017.00028

Obstacle Avoidance and Target Acquisition for Robot Navigation Using a Mixed Signal Analog/Digital Neuromorphic Processing System

Moritz B. Milde[1], Hermann Blum[1†], Alexander Dietmüller[1†], Dora Sumislawska[1], Jörg Conradt[2], Giacomo Indiveri[1] and Yulia Sandamirskaya[1]*

[1] Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland, [2] Neuroscientific System Theory, Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany

Neuromorphic hardware emulates dynamics of biological neural networks in electronic circuits offering an alternative to the von Neumann computing architecture that is low-power, inherently parallel, and event-driven. This hardware allows to implement neural-network based robotic controllers in an energy-efficient way with low latency, but requires solving the problem of device variability, characteristic for analog electronic circuits. In this work, we interfaced a mixed-signal analog-digital neuromorphic processor ROLLS to a neuromorphic dynamic vision sensor (DVS) mounted on a robotic vehicle and developed an autonomous neuromorphic agent that is able to perform neurally inspired obstacle-avoidance and target acquisition. We developed a neural network architecture that can cope with device variability and verified its robustness in different environmental situations, e.g., moving obstacles, moving target, clutter, and poor light conditions. We demonstrate how this network, combined with the properties of the DVS, allows the robot to avoid obstacles using a simple biologically-inspired dynamics. We also show how a Dynamic Neural Field for target acquisition can be implemented in spiking neuromorphic hardware. This work demonstrates an implementation of working obstacle avoidance and target acquisition using mixed signal analog/digital neuromorphic hardware.

Keywords: neuromorphic controller, obstacle avoidance, target acquisition, neurorobotics, dynamic vision sensor, dynamic neural fields

OPEN ACCESS

Edited by:
Christian Tetzlaff,
Max Planck Institute for Dynamics and Self Organization (MPG), Germany

Reviewed by:
Tomas Kulvicius,
University of Göttingen, Germany
Bernd Porr,
University of Glasgow,
United Kingdom

***Correspondence:**
Yulia Sandamirskaya
ysandamirskaya@ini.uzh.ch

† These authors have contributed equally to this work.

Received: *29 November 2016*
Accepted: *22 May 2017*
Published: *11 July 2017*

Citation:
Milde MB, Blum H, Dietmüller A, Sumislawska D, Conradt J, Indiveri G and Sandamirskaya Y (2017) Obstacle Avoidance and Target Acquisition for Robot Navigation Using a Mixed Signal Analog/Digital Neuromorphic Processing System. Front. Neurorobot. 11:28. doi: 10.3389/fnbot.2017.00028

1. INTRODUCTION

Collision avoidance is one of the most basic tasks in mobile robotics that ensures safety of the robotic platform, as well as the objects and users around it. Biological neural processing systems, including relatively small ones such as those of insects, are impressive in their ability to avoid obstacles robustly at high speeds in complex dynamical environments. Relatively simple neuronal architectures have already been proposed to implement robust obstacle avoidance (e.g., Blanchard et al., 2000; Iida, 2001; Rind and Santer, 2004), while probably the most simple conceptual formulation of a neuronal controller for obstacle avoidance is the famous Braitenberg vehicle (Braitenberg, 1986). When such *neuronal* control architectures are implemented on a conventional computer, analog sensor signals are converted and stored in digital variables. A large number of numerical computations are performed then, which are required to model the involved neuronal dynamics in software.

Neuromorphic hardware offers a physical computational substrate for directly emulating such neuronal architectures in real time (Indiveri et al., 2009; Furber et al., 2012; Benjamin et al., 2014; Chicca et al., 2014), enabling low latency and massively parallel, event-based computation. Neuromorphic electronic circuits can implement dynamics of neurons and synapses using digital (Furber et al., 2012) or analog (Benjamin et al., 2014; Qiao et al., 2015) designs and allow for arbitrary connectivity between artificial neurons. The analog implementations of artificial neural networks are particularly promising, due to their potential smaller size and lower power consumption figures than digital systems (for a review see Indiveri et al., 2011; Hasler and Marr, 2013). But these features come at a price of precision and reliability. Indeed, with analog designs, the device mismatch effects (i.e., variation in properties of artificial neurons across the device) have to be taken into account for the development of robust functional architectures (Neftci et al., 2011).

A promising strategy for taking these issues into account is to implement the mechanisms used in *biological neural networks*, which face the same problem of using an unreliable computing substrate that consists of noisy neurons and synapses driven by stochastic biological and diffusion processes. These biological mechanisms include adaptation and learning, but also using *population coding* (Ermentrout, 1998; Pouget et al., 2000; Averbeck et al., 2006) and *recurrent connections* (Wilson and Cowan, 1973; Douglas et al., 1995) to stabilize behaviorally relevant decisions and states against neuronal and sensory noise. In this work, we show that by using the population-coding strategy in a mixed signal analog/digital neuromorphic hardware, it is possible to cope with the variability of its analog circuits and to produce reliably the desired behavior on a robot.

We present a first proof of concept implementation of such a neuromorphic approach to robot navigation. Specifically, we demonstrate a *reactive vision-based* obstacle avoidance strategy using a neurally-inspired event-based Dynamic Vision Sensor (DVS) (Lichtsteiner et al., 2006) and a Reconfigurable On-Line Learning (ROLLS) neuromorphic processor (Qiao et al., 2015). The proposed architecture is event-driven and uses the neural populations on the ROLLS device to determine the steering direction and speed of the robot based on the events produced by the DVS. In the development phase, we use a miniature computer Parallella[1] solely to manage the traffic of events (spikes) between the neuromorphic devices, and to store and visualize data from the experiments. The Parallella board can be removed from the behavioral loop in target applications, leading to a purely neuromorphic implementation. In this paper, we demonstrate the robustness and limits of our system in a number of experiments with the small robotic vehicle "Pushbot[2]" in a robotic arena, as well as in an unstructured office environment.

Several neuromorphic controllers for robots were developed in the recent years, e.g., a SpiNNacker system (Furber et al., 2012) was used to learn sensory-motor associations with robots (Conradt et al., 2015; Stewart et al., 2016), a neural-array integrated circuit was used to plan routes in a known environment (Koziol et al., 2014), three populations of analog low-power subthreshold VLSI integrate-and-fire neurons were employed to control a robotic arm (Perez-Peña et al., 2013). Our system goes along similar lines and realizes a reactive robot navigation controller that uses a mixed signal analog/digital approach, and exploits the features of the ROLLS neuromorphic processor.

In this work we follow a dynamical systems—attractor dynamics—approach to robot navigation (Bicho et al., 2000), which formalizes one of the famous Braitenberg vehicles (Braitenberg, 1986). The neuronal architecture in our work is realized using a number of neuronal populations on the neuromorphic device ROLLS. The dynamical properties of neuronal populations and their interconnectivity allow to process a large amount of sensory signals in parallel, detecting the most salient signals and stabilizing these detection decisions in order to generate robustly closed-loop behavior in real-world unstructured and noisy environments (Sandamirskaya, 2013; Indiveri and Liu, 2015). Here, we demonstrate the feasibility of deployment of a neuromorphic processor for the closed loop reactive control. We found several limitations of the simple Braitenberg-vehicle approach and suggest extensions of the simple architecture that solve these problems, leading to robust obstacle avoidance and target acquisition in our robotic setup.

2. MATERIALS AND METHODS

The experimental setup used in this work consists of the Pushbot robotic vehicle with an embedded DVS camera (eDVS) and the ROLLS neuromorphic processor. A miniature computing board Parallella is used to direct the flow of events between the robot and the ROLLS. **Figure 1A** shows the components of our hardware setup, while **Figure 1B** shows the information flow between different hardware components.

The Pushbot communicates with the Parallella board via a wireless interface for receiving motor commands and for sending address-events produced by the DVS. Using a dedicated WiFi network, we achieve communication latency below 10 ms, which was enough to demonstrate functionality of our system at speeds, possible with the Pushbot.

The ROLLS device is interfaced to the Parallella board using an embedded FPGA, which is used to configure the neural network connectivity on the chip and to direct stimulating events to neurons and synapses in real time. The Parallella board runs a simple program that manages the stream of events between the neuromorphic processor and the robot.

2.1. The ROLLS Neuromorphic Processor

The ROLLS is a mixed signal analog/digital neuromorphic chip (Qiao et al., 2015) that comprises 256 spiking silicon neurons, implemented using analog electronic circuits which can express biologically plausible neural dynamics. The neurons can be configured to be fully connected with three sets of synaptic connections: an array of 256 × 256 non-plastic ("programmable") synapses, 256 plastic ("learnable") synapses that realize a variant of the Spike-Timing-Dependent Plasticity (STDP) rule (Mitra et al., 2009), and 4 additional ("virtual")

[1]https://www.parallella.org
[2]http://inilabs.com/products/pushbot

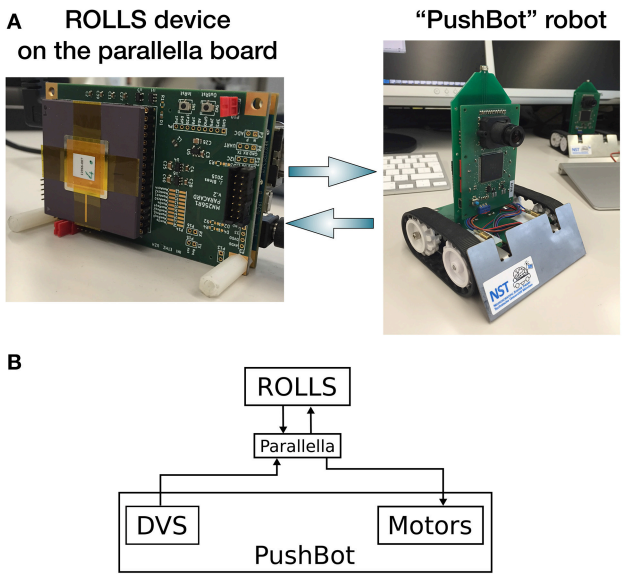


FIGURE 1 | (A) Information flow between the three main components: ROLLS, Parallella, and the Pushbot, in particular, its sensor DVS and two motors. **(B)** The hardware setup used in this work: the neuromorphic processor ROLLS is interfaced to a miniature computer Parallella, which communicates with the Pushbot robot over a dedicated WiFi network.

synapses that can be used to receive external inputs. In this work, only the programmable synapses were used for setting up the neuronal control architecture, as no online-learning was employed for the navigation task.

Figure 2 shows a block diagram of the ROLLS device, in which 256 spiking neurons, implemented using analog electronic circuits (Indiveri et al., 2006), are shown as triangles on the right, and 256 × 256 non-plastic ("programmable") synapses, which can be used to create a neuronal architecture on the ROLLS, as well as 256 "virtual" synapses used to stimulate neurons externally, are shown as white squares. A digital Address Event Representation (AER) circuitry allows to stimulate neurons and synapses on the chip, as well as to read-out spike events off chip; a temperature-compensated digital bias-generator allows to control parameters of analog electronic neurons and synapses, such as the refractory period or membrane time constant.

The programmable synapses share a set of biases that determine their weight values, their activation threshold, and time constants. These three parameters determine the synaptic strength and dynamics of the respective connection between two neurons. A structural limitation of the hardware is that each synapse can only assume one of eight possible weight values (four excitatory and four inhibitory values). This means that in a neuronal architecture, several different populations might have to share weights, which limits the complexity of the architecture. ROLLS consumes ∼4 mW of power in typical experiments, run here. The ROLLS parameters (biases) used in this work are listed in the Appendix (Supplementary Material, Appendix A).

2.2. The DVS Camera

The Dynamic Vision Sensor (DVS) is an event-based camera, inspired by the mammalian retina (Lichtsteiner et al., 2006; Liu and Delbruck, 2010). **Figure 3** shows a typical output of the DVS camera accumulated over 0.5 s (right) from the Pushbot robot driving in the office (left).

Each pixel of the DVS is sensitive to a relative temporal contrast change. If such change is detected, each pixel sends out an *event* at the time in which the change was detected (asynchronous real-time operation). Each event e is a vector: $e = (x, y, ts, p)$, where x and y define the pixel location in retinal reference frame, ts is the time stamp, and p is the polarity of the event. The event polarity encodes whether the luminance of the pixel increased (an "on" event) or decreased (an "off" event). All pixels share a common transmission bus, which uses the Address Event Representation (AER) protocol to transmit the address-events off chip.

The AER representation and asynchronous nature of communication makes this sensor low power, low latency, and low-bandwidth, as the amount of data transmitted is very small (typically, a very small subset of pixels produce events). Indeed, if there is no change in the visual scene, no information is transmitted off the camera. If a change is detected, it is communicated instantaneously, taking only a few microseconds to transfer the data off-chip.

For the obstacle avoidance scenario, important properties of the DVS are its low data rate, high dynamic range, and small sensitivity to lighting conditions (Lichtsteiner et al., 2006). The challenges are noise, inherent in the sensor, its inability to detect homogeneous surfaces, and relatively small spatial resolution (128 × 128 pixels), as well as a limited field of view (60°). New versions of DVS are currently available, which would further improve performance of the system. Moreover, more sophisticated object-detection algorithms for DVS are currently being developed (Moeys et al., 2016).

The embedded version of the DVS (eDVS) camera (Müller and Conradt, 2011) used in this work uses an ARM Cortex microcontroller to initializes the DVS, capture events, send them to the wireless network, and to receive and process commands for motor control of the Pushbot.

2.3. Neuromorphic Robot

The robot used in this work is the mobile autonomous platform Pushbot, which consists of a 10 × 10 cm chassis with two motors driving two independent tracks for propulsion (left and right). The predominant component on the small robot is an eDVS (Section 2.2), which acquires and provides sensory information and controls actuator output, including the robot's motors, through its embedded microcontroller. The sensor's integrated 9 DOF IMU reports changes of velocity and orientation. The robot actuators include a buzzer, two parallel, horizontal forward laser pointers and an LED on top, which all can show arbitrary activation patterns. The Pushbot is powered by 4 AA-batteries, which ensure ∼2 h operation time.

The robot communicates through WLAN at up to 12 Mbps, which allows remote reading of sensory data (including events from the eDVS) and setting velocities with a latency < 10 ms.

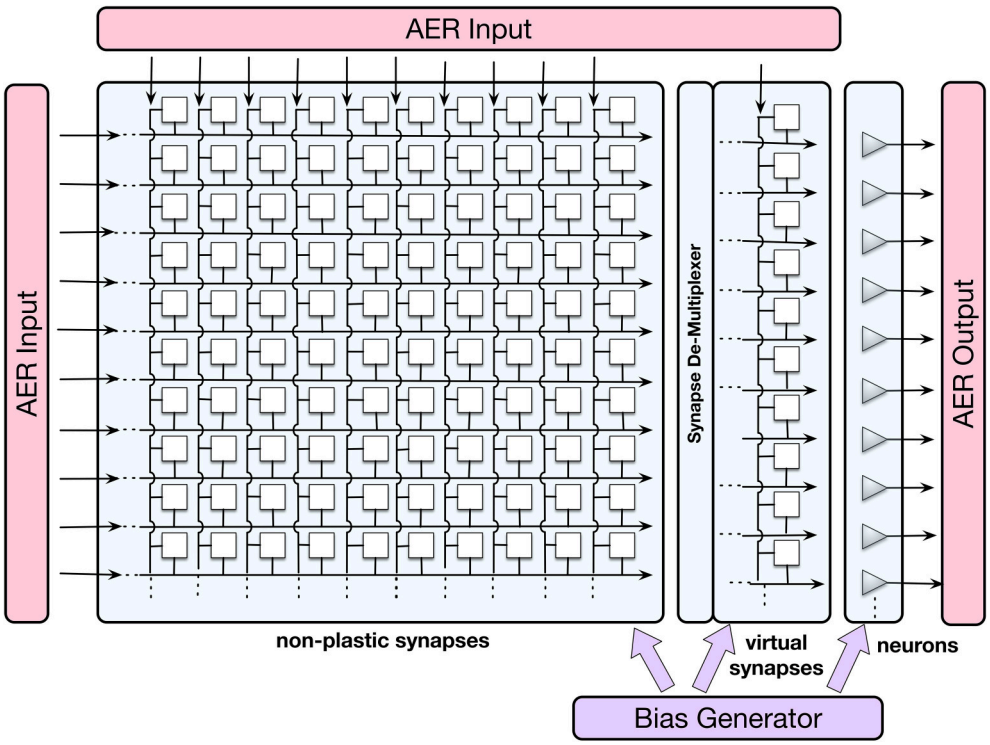


FIGURE 2 | The schematic visualization of neurons (grey triangles), non-plastic and virtual synapses (white squares), as well as input-output interfaces and bias generator of the ROLLS chip. Each neuron on the chip (presynaptic neuron) sends its output spikes to 256 non-plastic synapses, which, if set active, can route these spikes to any of the neurons on the chip (postsynaptic neurons). The connectivity matrix allows for all-to-all connectivity, but also other configurations. AER is a digital Address-Event Representation, used to communicate spikes (it consists of an index of the spike-emitting neuron).

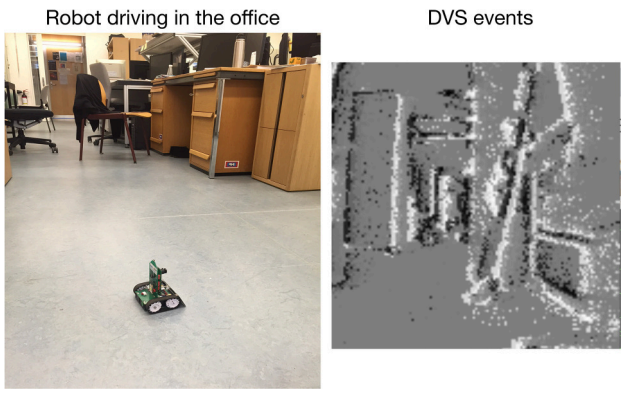


FIGURE 3 | The Pushbot robot driving in the office **(left)** and a visualization of the DVS output **(right)**, accumulated over 0.5 s.

The Pushbot robot is too small to carry the current experimental hardware setup. In principle, however, it is possible to place the ROLLS chip directly on a robot, removing the WiFi latency.

2.4. Spiking Neural Network Architecture

The core of the system presented here is a simple neural network architecture that is realized in the ROLLS device and allows the robot to avoid obstacles and approach a simple target. The "connectionist" scheme of the obstacle avoidance part of the architecture is shown in **Figure 4A**, while the scheme of the target acquisition architecture is shown in **Figure 4B**.

For obstacle avoidance, we configured two neuronal populations of 16 neurons each to represent a sensed obstacle to the right ("obstacle right," or OR) and to the left ("obstacle left," or OL) from the robot's heading direction. Each neuron in the OL and OR populations receives a spike for each DVS pixel that produces an event in the left (right) part of the sensor, respectively (we used the lower half of the sensor for obstacle avoidance). The spiking neurons in the two obstacle populations sum up the camera events according to their neuronal integrate-and-fire dynamics (equations can be found in Appendix B (Supplementary Material)). If enough events arrive from the same neighborhood, the respective neuron will fire, otherwise it will ignore events that are caused by the sensor noise. Thus, the obstacle representing neuronal populations achieve basic filtering of the DVS events. The output spikes of the neuronal populations signal the detection of an object in the respective half of the field of view.

Each of the obstacle detecting neuronal populations is connected to a motor population "drive left, DL" or "drive right, DR" (with 16 neurons per population). Consequently, if an obstacle is detected on the right, the drive left population is stimulated, and vice versa. The drive populations inhibit each other, implementing a winner-take-all dynamics. Thus, a

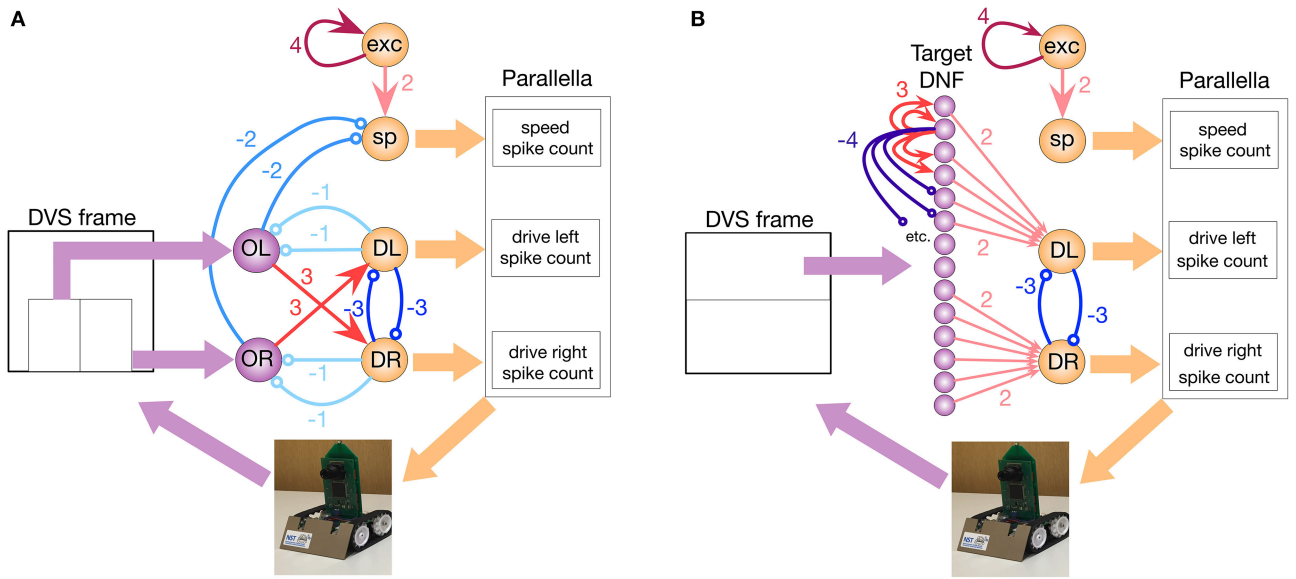


FIGURE 4 | The implemented neuronal architectures for obstacle avoidance and target acquisition. **(A)** Obstacle avoidance: Violet OL and OR circles represent obstacle detecting neuronal populations. Orange DL and DR circles are motor driving populations. sp is the speed-setting population, and exp is the constantly firing population that sets the default speed. Thin arrows show excitatory non-plastic connections realized on the ROLLS chip, whereas colors and numbers show the weights (the exact value of the weight is set by the biases, listed in Table 2) in Appendix B (Supplementary Material). **(B)** Target acquisition: Same notations. The violet line of circles shows a DNF population that represent targets. On the chip, both architectures are realized at the same time.

decision about the direction of an obstacle-avoiding movement is taken and stabilized at this stage by the dynamics of neuronal populations on the chip.

The *drive* populations, in their turn, inhibit both obstacle detecting populations, since during a turning movement of the robot, many more events are generated by the DVS, compared to those generated during translational motion. This inhibition compensates for this expected increase in the input rate, similar to the motor re-afferent signals in biological neural systems (Dean et al., 2009). This modification of the simple Braitenberg vehicle principle is required to enable robust and fast behavior.

The speed of the robot is controlled by a neuronal population, "speed, sp," which receives input from a constantly firing "exc," excitatory population. The latter group of neurons has strong recurrent connections and continually fires when triggered by a transient activity pulse. In an obstacle-free environment, the speed population sets a constant speed for the robot. The obstacle detecting populations OL and OR inhibit the speed population, making the robot slow down if obstacles are present. The decreasing speed ensures a collision-free avoidance maneuver.

These six populations comprise only 96 neurons, and represent all that is needed to implement the obstacle avoidance dynamics in this architecture (**Figure 4A**).

The control signals sent to the robot are, first, the angular velocity, v_a, that is proportional to the difference in the number of spikes per neuron emitted between the two drive populations (Equation 1), and, second, the forward velocity, calculated based on the number of spikes per neuron emitted by the speed population (Equation 2):

$$v_a = c_{turn}\left(\frac{N_{DL}^{spike}}{N_{DL}^{n}} - \frac{N_{DR}^{spike}}{N_{DR}^{n}}\right), \quad (1)$$

$$v_f = c_{speed}\frac{N_{sp}^{spike}}{N_{sp}^{n}}, \quad (2)$$

where N_{XX}^{spike} are the numbers of spikes, obtained from the respective populations [drive left (DL), drive right (DR), and speed (sp)] in a fixed time-window, we used 500 and 50 ms in an improved version); N_{XX}^{n} is the number of neurons in the respective population; and c_{turn} and c_{speed} are turn- and speed-factors (user-defined constants), respectively.

Thus, we used neural population dynamics to represent angular and translational velocities of the robot and used the firing rate of the respective populations of neurons as the control variable.

2.4.1. Dynamic Neural Field for Target Representation

To represent targets of the navigation dynamics, we use a Dynamic Neural Fields (DNFs) architecture as defined in Bicho et al. (2000). DNFs are population-based models of dynamics of large homogeneous neuronal populations, which have been successfully used in modeling elementary cognitive function in humans (Schöner and Spencer, 2015), as well as in implementing cognitive representations for robots (Erlhagen and Bicho, 2006; Bicho et al., 2011; Sandamirskaya et al., 2013). DNFs can be easily realized in neuromorphic hardware by setting a winner-take-all (WTA) connectivity network in a neural population (Sandamirskaya, 2013). Each neuron in a soft WTA network has a positive recurrent connection to itself and to its 2–4

nearest neighbors, implementing the lateral excitation of the DNF interaction kernel. Furthermore, all neurons have inhibitory connections to the rest of the WTA network, implementing the global inhibition of a DNF. These inhibitory connections can be either direct, as used here, or be relayed through an inhibitory population, which is a more biologically plausible structure.

In our architecture, we select 128 neurons on the ROLLS chip to represent visually perceived targets. Each neuron in this population receives events from the upper half of each column of the 128 × 128 sensor frame from the eDVS and integrates these events according to its neuronal dynamics: only events that consistently are emitted from the same column lead to firing of the neuron. The nearby neurons support each other's activation, while inhibiting further neurons in the WTA population (**Figure 4B**).

This connectivity stabilizes localized blobs of most salient sensory events, filtering out sensor noise and objects that are too large (inhibition starts to play role within object representation) or too small (not enough lateral excitation is engaged). Thus, the WTA connectivity stabilizes the target representation. The target in our experiments was a blinking LED of the second robot, which was detected in the DNF realized on the ROLLS. While this target could be easily detected since the blinking LED produces many events, more sophisticated vision algorithms are being developed to pursue an arbitrary target (Moeys et al., 2016).

The target population was divided in three regions: neurons of the DNF that receive inputs to the left from midline of the DVS frame drive the "drive left" population, whereas neurons that receive input from the right half of the DVS frame drive the "drive right" population. We did not connect the central 16 neurons of the target DNF to the drive populations to ensure more smooth target pursue when the target is in the center of the DVS frame (**Figure 4B**).

2.4.2. Combining Obstacle Avoidance and Target Acquisition

The two neuronal populations that ultimately determine the robot's steering direction (DR and DL) sum-up contributions from the obstacle-representing populations and the target-representing WTA population (**Figure 4**). The obstacle contribution is made effectively stronger than the target contribution by setting the ROLLS biases accordingly. Thus, in the presence of an obstacle in the robot's field of view, an obstacle avoidance maneuver is preferred.

Figure 5 shows the connectivity matrix used to configure the non-plastic connections on the ROLLS chip to realize both obstacle avoidance and target acquisition. This plot shows the weights of non-plastic synapses on the ROLLS chip (blue being the negative weights and red the positive weights; the same color code is used for the different weights as in **Figure 4**), which connect groups of neurons (different populations, labeled on the right side of the figure) among each other. Within-group connections are marked with black squared frames on the diagonal of the connectivity matrix. Violet and orange arrows show inputs and outputs of the architecture, respectively.

This connectivity matrix is sent to the ROLLS device to configure the neuronal architecture on the chip, i.e., to "program" the device.

3. DEMONSTRATIONS

We verified the performance of our system in a number of demonstrations, reported next. Overall, over 100 runs were performed with different parameter settings. In the following, we will provide an overview for the experiments and describe a few of them in greater detail to provide intuition of how the neural architecture works. For most experiments, we let the robot drive in a robotic arena with a white background and salient obstacles. We used a tape with a contrastive texture to make the walls of the arena visible to the robot. In four runs, we let the robot drive for several minutes freely in the office.

3.1. Probing the Obstacle Avoidance: A Single Static Obstacle

In the first set of experiments, we let the robot drive straight toward a single object (a colored block 2.5 cm wide and 10 cm high) and measured the distance from the object at which the robot crossed a virtual line perpendicular to the robot's initial heading direction, on which the object is located (e.g., see the distance between the robot and the "cup" object at the last position of the robot in **Figure 6**). We varied the speed factor of the architecture from 0.1 (∼0.07 m/s) to 3.0 (∼1 m/s) and have verified the effectiveness of the obstacle avoidance maneuver. Furthermore, we have increased the turning factor from 0.5 to 1.0 to improve performance at high speeds and have tested color-dependence of the obstacle perception with the DVS. **Table 1** shows results of these measurements. Each trial was repeated 3 times and mean over the trials was calculated.

The table allows to note the following characteristics of the architecture at the chosen parametrization. First, the performance drops at very low speeds (speed factor 0.1), especially for red and yellow objects, due to an insufficient number of DVS events to drive the neuronal populations on ROLLS. Second, there is a trade-off between this effect and the expected decay in performance (in terms of the decreasing distance to the obstacle) with increasing speed. Thus, at a turning factor 0.5, best performance is achieved for the blue object at speed factor 0.5 and for the red object at speed factor 1. Distance to the obstacle can be further increased by increasing the turn factor. Thus, at turn factor 1 and speed factor 1 best performance (i.e., largest distance to the obstacle) can be achieved for both the blue and red objects. Yellow object provides too little contrast to be reliably perceived by the DVS in our set-up.

Figure 6 demonstrates how the neuronal architecture on the ROLLS chip realizes obstacle avoidance with the Pushbot. On the left, an overlay of video frames (recording the top view of the arena) shows the robot's trajectory when avoiding a single obstacle (here, a cup) in one of the runs. Numbers (1–3) mark important moments in time during the turning movement. On the right, summed activity of the neuronal populations on the ROLLS device is shown over time. The same moments in time

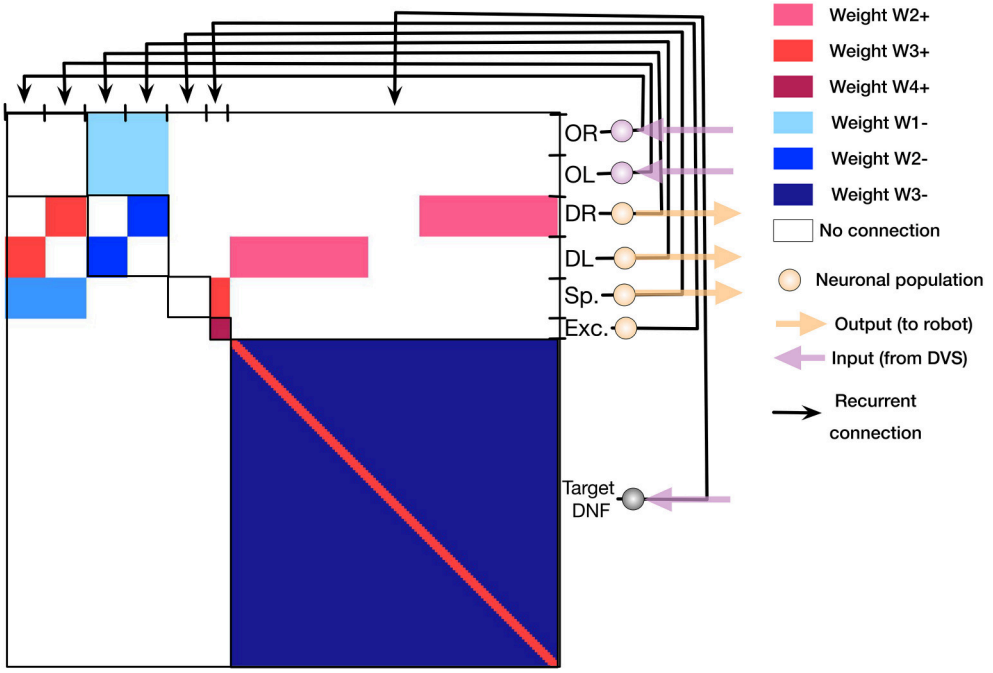


FIGURE 5 | The synaptic connectivity matrix, configured on the ROLLS chip to implement the obstacle avoidance and target acquisition architectures. Colors encode different synaptic weights (red for positive and blue for negative connection weights) of the recurrent connections on the chip.

are marked with numbers as in the left figure. In this case, already the obstacle detecting populations had a clear "winner"—the left population forms an increasing activity bump over time, which drives the "drive right" population, inducing a right turn of the robot. The bottom plot shows the commands that are sent to the robot (speed and angular velocity): the robot slows down in front of the obstacle and turns to the right.

We have performed several further trials, varying the lighting conditions (normal, dark, very dark) and parameters of the architecture. Since the architecture uses the difference in spiking activity, induced by sensory events from the two halves of the visual space, avoiding a single obstacle works robustly, although the camera might miss objects with a low contrast (e.g., yellow block in our white arena). More advanced noise filtering would improve performance. While more extended version of the performed tests will be reported elsewhere, **Figure 7** show results of some of the successful and unsuccessful runs.

3.2. Avoiding a Pair of Obstacles

We repeated the controlled obstacle avoidance experiment with two and three blocks in different positions. Each configuration was tested twenty times without crashes at speed 0.35 m/s (speed factor 0.5).

Figure 8 shows an exemplary run that explains *how* the robot avoids a pair of obstacles. This example is important, since in the attractor dynamics approach to navigation, distance between the two objects determines a decision to move around or between the objects.

Snapshots from the overhead camera are shown on the left of **Figure 8**. Output of the DVS, accumulated in 500 ms time windows around the time when the snapshots were taken[3], is shown in the second column, and the spiking activity of neuronal populations recorded from the ROLLS chip is shown in the two right-most columns. Activity is shown of the obstacle representing left (red) and right (blue) neuronal populations (third column), the left (red) and right (blue) drive populations, and the speed population (gray, forth column). Each of these populations has 16 neurons, dots represent their spikes[4].

At the moment, depicted in the top row of **Figure 8**, the robot senses an obstacle on the right, although the DVS output is rather weak. Note that the neuronal population filters out sensory noise of the DVS and only detects events that cluster in time and in space. The robot turns left, driven by the activated *drive left* population and now the obstacle on the right becomes visible, providing a strong signal to the *right obstacle* population and, consequently, to the *drive left* population (second and third row). Eventually, the obstacle on the right dominates and the robot drives past both obstacles on the left side (forth row).

Thus, with the chosen parametrization of the neuronal network architecture, the robot tends to go around a pair of objects, avoiding the space between them. This behavior could be changed, making the connections between the obstacle representing populations and drive populations stronger. However, for a robot equipped with a DVS, such strategy is

[3]We dropped 80% of DVS events randomly in our architecture; moreover, we only used 5% of all remaining events for plotting.
[4]Only 5% of the ROLLS spikes (every 20th spike) are shown in all our plots.

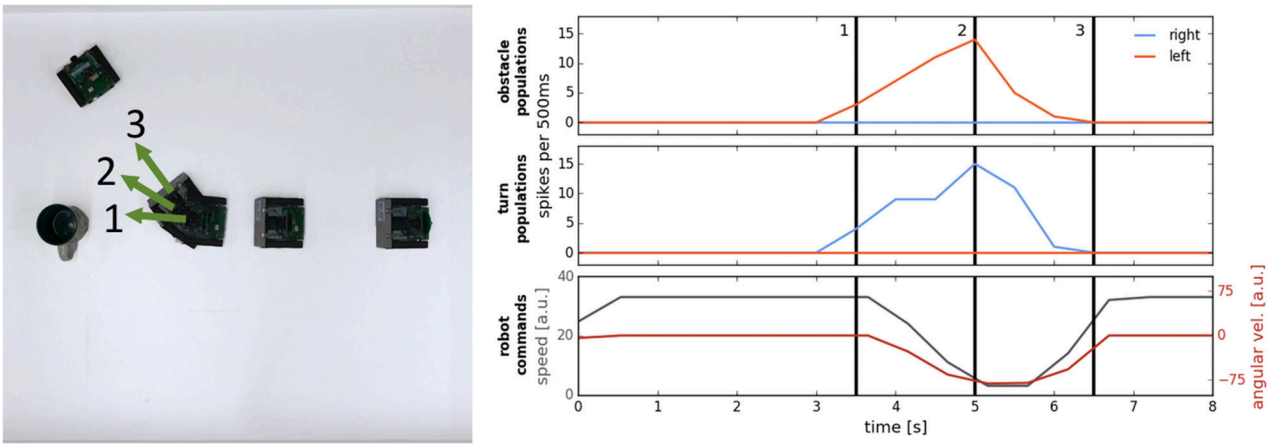


FIGURE 6 | An example of an obstacle avoidance maneuver. **Left:** Overlay of video frames showing the trajectory of the robot. **Right:** activity of the neuronal populations on the chip (Top: the left and right obstacle detecting populations; Middle: the left and right drive populations), and the motor commands, sent to the robot (Bottom plot).

TABLE 1 | Collision avoidance at different speeds: distance to the obstacle when crossing the obstacle-line (mean over 3 trials ± standard deviation in [cm]) at different speed- and turn-factors and for different colors of the obstacle.

Speed/turn	0.1/0.5	0.5/0.5	1/0.5	1/1	1.5/1	2/1	3/1
Blue	7.0 ± 1.0	10.3 ± 0.6	7.7 ± 1.5	19.3 ± 2.1	16.3 ± 3.3	10.8 ± 2.6	0*
Red	0*	2.3 ± 0.6	4.7 ± 0.6	10.7 ± 1.2	9.7 ± 3.5	5.0 ± 1.0	0*
Yellow	0*	0*	0*	7.0* ± 6.1	0*	0*	0*

** signifies trials when a collision happened.*

safer, since for homogeneous objects, the DVS can only sense the edges, where a temporal contrast change can be induced by the robot's motion. The robot thus might miss the central part of an object and avoiding pairs of close objects is a safer strategy. Adaptive connectivity that depends on the robot speed is also feasible.

3.3. Avoiding a Moving Obstacle

In these experiment, the robot is driving straight in the arena while we move an obstacle (a coffee mug) into its path. We repeat this experiment six times with varying speed factors (0.1–2) of the robot. The robot was capable to avoid collisions in all tested cases. In fact, avoiding a moving obstacle is more robust than avoiding a static obstacle because the moving obstacle produces more DVS events than a static one at the same robot speed.

Figure 9 shows how the robot avoids a moving obstacle. The same arrangement of plots was used as in **Figure 8**, described in Section 3.2. The robot was moving with $c_{speed} = 0.5$ (0.35 m/s) here, the cup was moved at ∼0.20 m/s.

3.4. Cluttered Environment

In the following set of experiments, we randomly placed obstacles (8–12 wooden pieces) in the arena and let the robot drive around at an average speed (0.35 m/s). We analyzed the performance of the architecture, suggesting a number of modifications to cope with its limitations.

Figure 10 demonstrates behavior of the obstacle avoidance system in a cluttered environment. In particular, we let the robot drive in an arena, in which 8 obstacles were randomly distributed. The robot successfully avoids obstacles in its way with two exceptions: the robot touches the blue obstacle in the center of the arena, which entered the field of view too late for a maneuver, and also collides with the yellow object, which did not provide enough contrast to produce the required number of DVS events. These collisions point to two limitations of the current setup, which, first, uses single camera with a narrow field of view and, second, drops 80% of events to improve signal to noise ratio (the latter deprives performance for objects with low contrast against the background). Using more sophisticated noise filter would improve visibility of the faint obstacles. Note that we used rather small objects on these trials (blocks of 2 × 5 cm), which posed a challenge for the event-based detection, especially taking into account our very simplistic noise-reduction strategy.

To improve behavior in a cluttered environment, we modified the architecture, adding two more populations on the ROLLS chip, which receive input from the inertia measurement unit of the Pushbot and which suppress obstacle populations when the robot is turning. Moreover, we replaced the homogeneous connections between the obstacle and the drive populations with graded connections that become stronger for obstacles detected in the center than in the periphery of DVS field of view. This allows the robot to make shorter avoidance maneuvers and avoid obstacles in a denser configuration at a higher speed. **Figure 11**

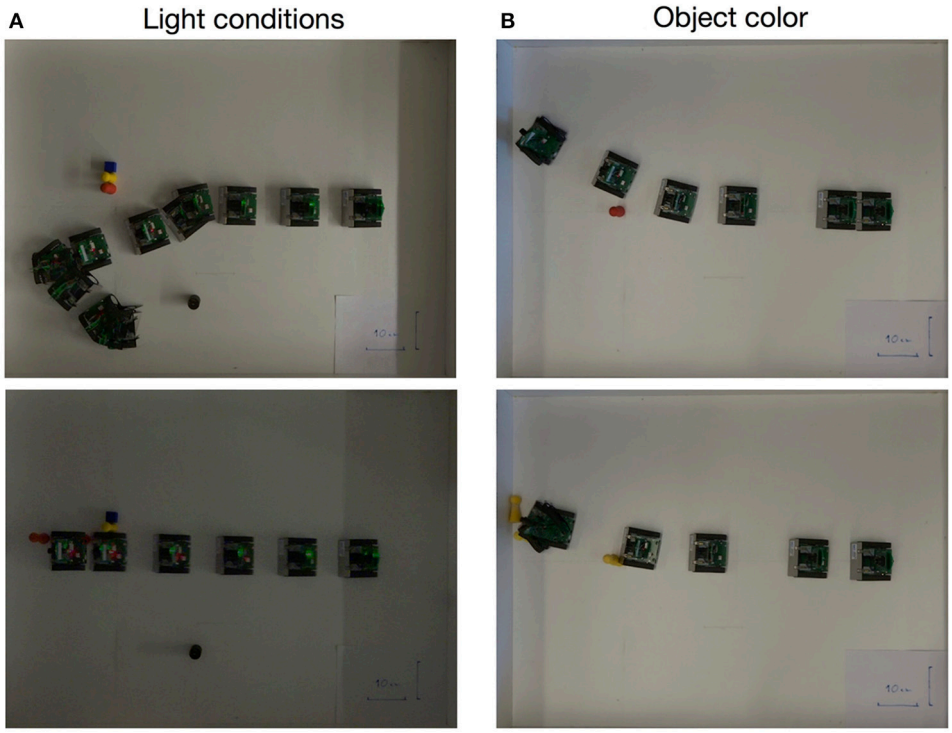


FIGURE 7 | Exemplary experiments showing successful **(Top row)** and unsuccessful **(Bottom row)** obstacle avoidance maneuvers in different light conditions **(A)** and with obstacles of different colors **(B)**.

shows a successful run with the modified architecture. Here, we also changed the sampling mechanism used to calculate the robot commands, replacing a fixed time window with a running average. This allowed us to avoid obstacles in the cluttered environment without collisions at speed as high as 0.5 m/s.

3.5. Variability of Behavior

Since behavior of our robot is controlled by activity of neuronal populations, implemented in analog neuromorphic hardware, the behavior of the robot has some variability, even when exactly the same parameters of the architecture and the same hardware biases are used. Despite this variability, the robot's goal—avoiding obstacles—remains fulfilled. Such variability of behavior can be used as a drive for exploration, which may be exploited in learning scenarios in more complex architectures, built on top of our elementary obstacle avoidance system.

Figure 12 demonstrates variability behavior of our neuronal controller. In the figure, we show three trials, in which the robot avoids a two-blocks configuration, starting from exactly the same position and with the same configuration of the neuronal controller (speed factor 0.5, turn factor 0.5). Mismatch in the neuronal populations implemented in analog neuromorphic hardware, variability of the DVS output, and its dependence on the robot's movements lead to strong differences in trajectories. In particular, in the case shown in **Figure 12**, the trajectories may bifurcate and the robot might avoid the two obstacles on the right, or on the left side.

3.6. Obstacle-Avoidance in a Real-World Environment

Finally, we tried our architecture outside of the arena as well. The robot was placed on the floor in the office and drove around avoiding both furniture and people. The high amount of background activity compared to the arena did not diminish the effectiveness of the architecture: in four 0.5–1.5-min long trials, the robot only crashes once after it maneuvered itself into a dark corner under a table where the DVS sensor could not provide sufficient information to recognize obstacles.

Figure 13 shows an example of the Pushbot robot driving in the office environment. On the left, three snapshots from the video camera recording the driving robot are shown (full videos can be see in the Supplementary Material). The snapshots show the robot navigating the office environment with its task being to avoid collisions. The middle column of plots shows pairs of eDVS events, accumulated over 500 *ms* around the moment in time in the corresponding snapshot on the left, and respective histograms of events from the center region, used for obstacle avoidance. Events above the mid-line of the eDVS field-of-view are shown with transparency to emphasize that they were not used for obstacle avoidance: only events from the region of the eDVS field-of-view between the two vertical lines in **Figure 13** were used.

Histograms below the eDVS plots show the events from this region of the field of view, summed over the eDVS columns. These events drive the obstacle left (red colored part of the

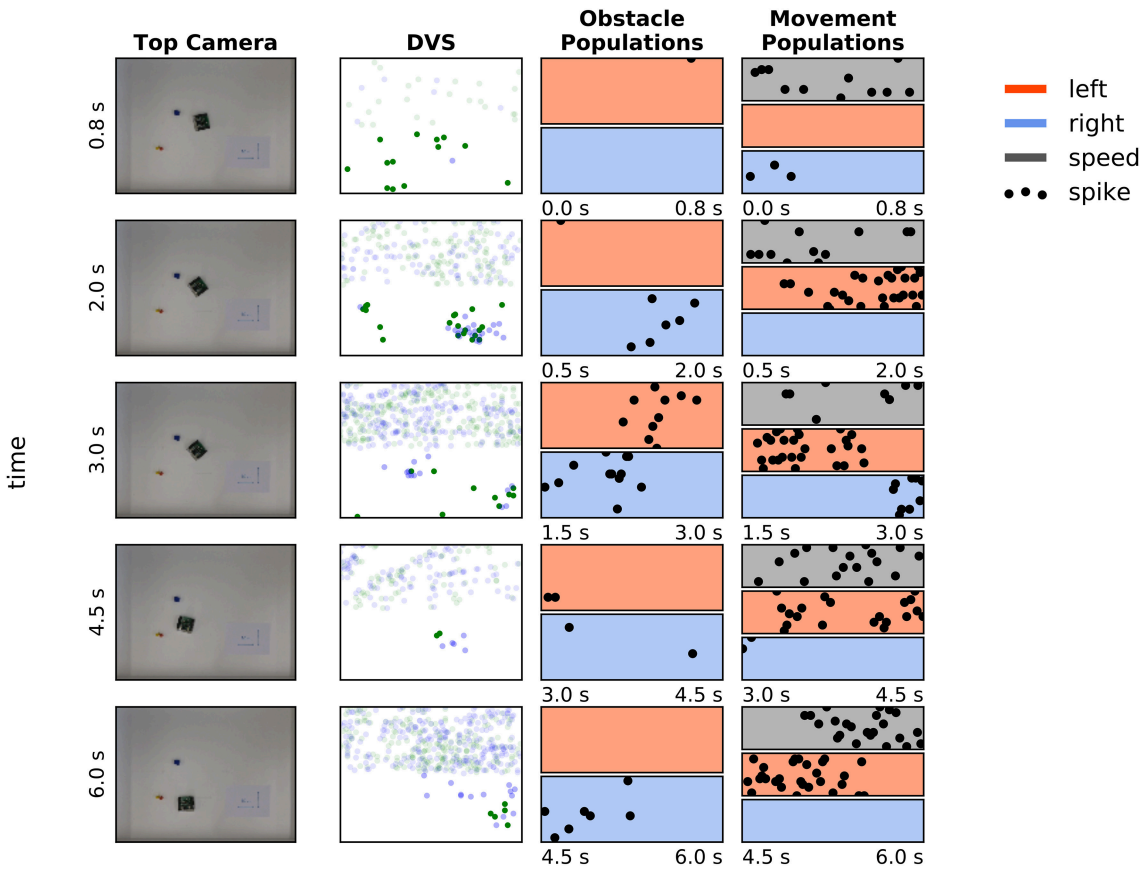


FIGURE 8 | Avoiding a pair of obstacles. **First column:** Snapshots of four moments in time during avoidance of a cup, moved into the robot's trajectory. **Second column:** DVS "frames"—events, accumulated over a 0.5 s time window. Green dots are off events, blue dots are on events. Events in the upper part of the frame were not considered for the obstacle avoidance. **Third column:** Activity of the obstacle representing populations in 0.6–1.5 s before the camera snapshot in the first column was taken (red—left population (n_{OL}), blue – right population (n_{OR}); each population has 16 neurons). **Forth column:** Activity of the drive left (red), drive right (blue), and speed population on the ROLLS chip in the same time as on the plots in column 3.

histogram) and obstacle right (blue part of the histogram) neuronal populations on the ROLLS chip.

The right column shows activity of the neuronal populations on the ROLLS chip over time, as in the previous figures. Black vertical lines mark time moments that correspond to the three snapshots in the left column. These plots allow to see that although the left and right obstacle populations are often activated concurrently, only one of the drive populations (either left or right) is active at any moment, leading to a clear decision to turn in either direction in the presence of perceived obstacles. The speed plot shows that movement of the robot is not very smooth—it slows down and accelerates often based on the sensed presence of obstacles. This behavior is improved in the modified architecture, briefly described in Section 3.4.

When driving around the office, robot faced very different lighting conditions, as can be seen already in the three snapshots presented here. This variation in lighting conditions did not effect obstacle avoidance in most cases, since the DVS is sensitive to relative change of each pixel's intensity, which varies less than the absolute intensity when the amount of ambient light changes. However, in an extreme case, shown in the lower snapshot in **Figure 13**, the robot collided with the metal foot of the chair. This was the only collision recorded.

3.7. Target Acquisition

In addition to obstacle avoidance we also tested target acquisition in ten experiments using a second robot with a blinking LED as target. The robot successfully turns and drives toward the target every time (at speed and turn factors = 0.5). In 8 out of 10 experiments the target is recognized as an obstacle when approached and is avoided; in two experiments, the robot failed to recognize target as obstacle after approaching it.

Obviously, the simple visual preprocessing that we used did not allow us to distinguish the target from obstacles (other than through their position in the upper or lower part of the field-of-view of the DVS). Moreover, we would need an object detection algorithm to detect the target and segregate it from the background. This vision processing is outside the scope of our work, but there is a multitude of studies going in this direction (Moeys et al., 2016) using modern deep/convolutional neural networks learning techniques.

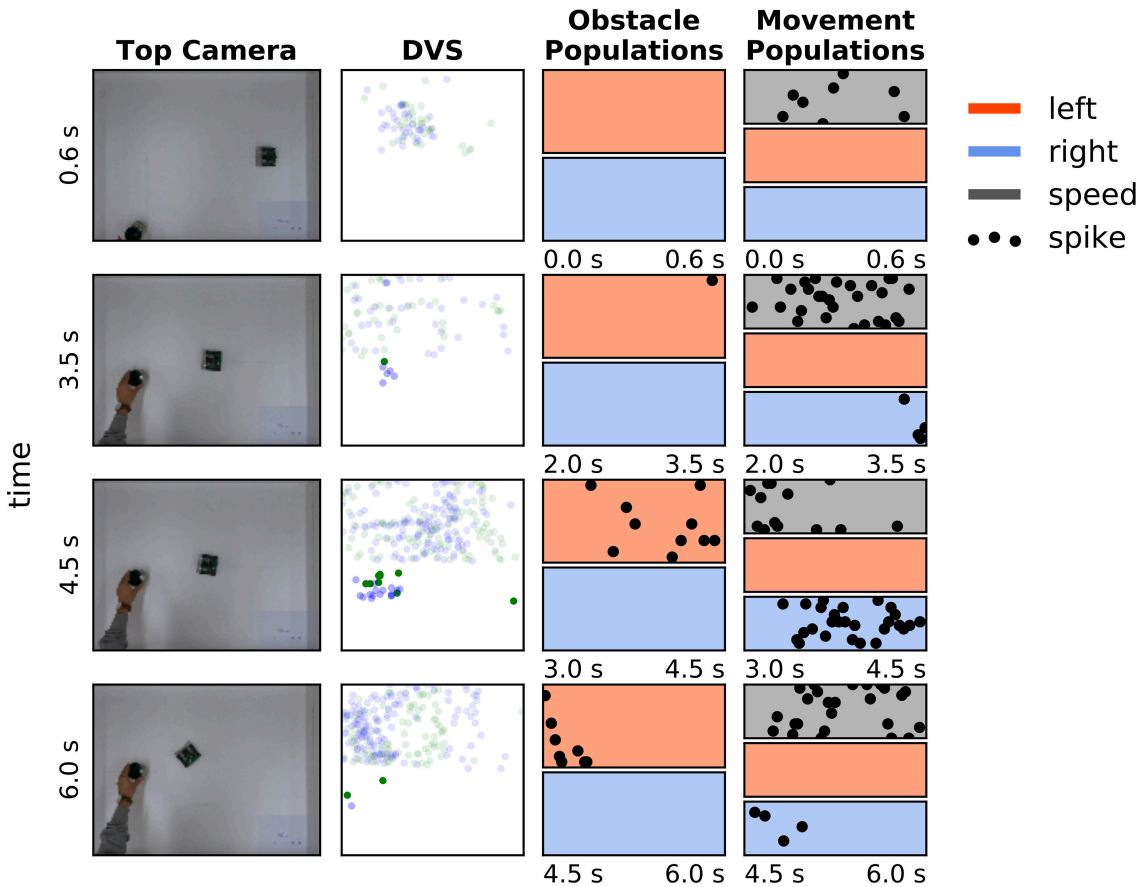


FIGURE 9 | Avoiding a moving obstacle. The same arrangement is used as in **Figure 8**. See main text for the discussion.

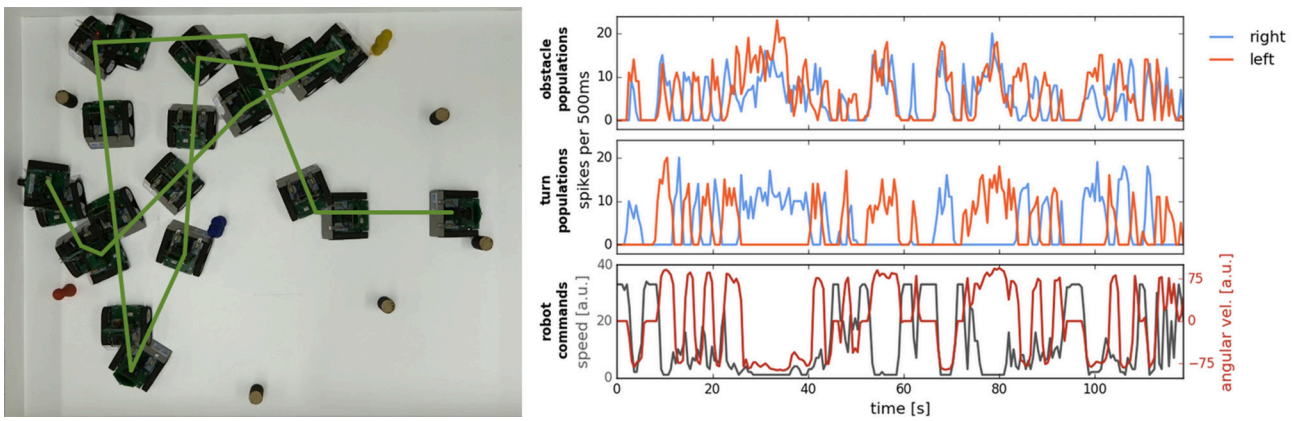


FIGURE 10 | Navigation in a cluttered arena. **Left:** Overlayed frames from the video, recoding the robotic arena from the top. Green line markes the path of the robot. **Right:** Summed activation of neurons in populations on the ROLLS chip over the time of the experiment. Obstacle and turn (**left** and **right**) population are shown, as well as the commands sent to the robot (angular velocity and speed).

Figure 14 shows target acquisition for a static target and demonstrates that the robot can approach the target object. At a short distance, the obstacle component takes over and the robots turns away after approaching the target. The figure shows the overlayed snapshots from the overhead camera, showing how the robot turns toward the second robot, standing on the left side of the image. When getting close to the second robot (∼10 cm), the robot perceives the target as an obstacle, which

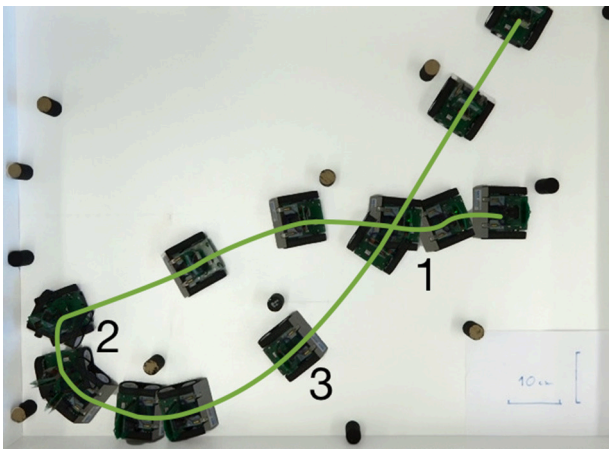


FIGURE 11 | Successful run in a cluttered environment with a modified neuronal architecture. Overlay of the overhead-camera frames.

has a stronger contribution to its movement dynamics and the robot turns away. On the left, the spiking activity of the target representation on the ROLLS chip is shown (raster plot where each dot represents a spike[5]). We can see that the robot perceives its target consistently on the left. After the eighth second, the obstacle contribution on the right becomes dominant and the robot turns left strongly.

Figure 15 shows how the robot can chase a moving target. We have controlled the second Pushbot remotely and have turned its LED on (at 200 Hz, 75% on-time). The LED provided a rather strong (though spatially very small) input to the DVS of the second, autonomously navigating robot. This input was integrated by our target WTA (DNF) population, which, however, also received a large amount of input from the background (in the upper part of the field of view the robot could see behind the arena's walls). Input from the localized LED was stronger and more concise than more distributed input from the background and such localized input was enhanced by the DNF's (WTA's) lateral connections. Consequently, the respective location in the target WTA formed a "winner" (localized activity bump in the DNF terminology) and inhibited the interfering inputs from other locations.

In the figure, four snapshots of the video recording the two robots are shown (top row). The leading robot was covered with white paper to reduce interference from the obstacle avoidance dynamics as the robots get close to each other (the space in the arena and the small size of the blinking LED forced us to put the robots rather close to each other, so that the target robot could be occasionally perceived as an obstacle).

In the second row in **Figure 15**, the summed over 500 ms events of the DVS are shown, around the same time points as the snapshots. Only the upper part of the field-of-view was considered for target acquisition. This part is very noisy, since the robot "sees" outside the arena and perceives objects in the background, which made target acquisition very challenging. Still, the blinking LED provided the strongest input and in most cases the target DNF was able to select its input as the target and suppress the competing inputs from the background—see activity of neurons in the target DNF in the bottom plot.

This last plot shows spiking activity of 215 neurons of the ROLLS chip, used to drive the robot (we don't show the constantly firing n_{exc} population here). We can see that the target DNF (WTA) successfully selects the correct target in most cases, only loosing it from sight twice, as the robot receives particularly many DVS events from the background during turning. The lower part of this raster plot shows activity of the obstacle populations, the drive populations, and the speed population, thus the dynamics of the whole architecture can be seen here.

4. DISCUSSION

This paper presents a neuronal architecture for reactive obstacle avoidance and target acquisition, implemented using a mixed-signal analog/digital neuromorphic processor (Qiao et al., 2015) and a silicon retina camera DVS as the only source of information about the environment. We have demonstrated that the robot, controlled by interconnected populations of artificial spiking neurons, is capable of avoiding multiple objects (including moving objects) at an average movement speed (up to 0.35 m/s with our proof of concept setup). We have also demonstrated that the system works in a real-world office environment, where background clutter poses a challenge for the DVS on a moving vehicle, creating many distracting events. We demonstrated that also the target acquisition neural architecture can cope well with this challenge, which was relevant even in the robotic arena. The distributed DNF representation of the target, supported by lateral interactions of the WTA neuronal population, enabled robust detection and reliable selection of the target against background.

The reactive approach to obstacle avoidance that we adopt in this work has a long history of success, starting with the neurally inspired turtle robot more than half a century ago, as reviewed by Holland (1997). Later, Valentino Braitenberg analyzed a number of hypothetical vehicles, or creatures, that use reactive control to produce complex behaviors (Braitenberg, 1986). His controllers were realized as simple "nervous systems" that directly linked the sensors to the motors of the vehicle. Using similar sensorimotor, or behavioral modules as building blocks, Rodney Brooks developed a behavior-based controller paradigm for roaring vehicles, known as "subsumption architecture" (Brooks, 1991). Although this framework did not scale well for complex tasks and is not ideally suited for online learning methods, this type of controller is at the heart of highly successful real-world robotic systems such as the autonomous vacuum cleaners, and has been adopted, to some extent, in a wide range of impressive controllers for autonomous robots (e.g., Khansari-Zadeh and Billard, 2012).

The dynamical systems approach to robot navigation (Schöner et al., 1995) is an attempt to mathematically formalize reactive control for autonomous robots using

[5]Remember, that only 5% (every 20th) of all spikes from the ROLLS processor are shown.

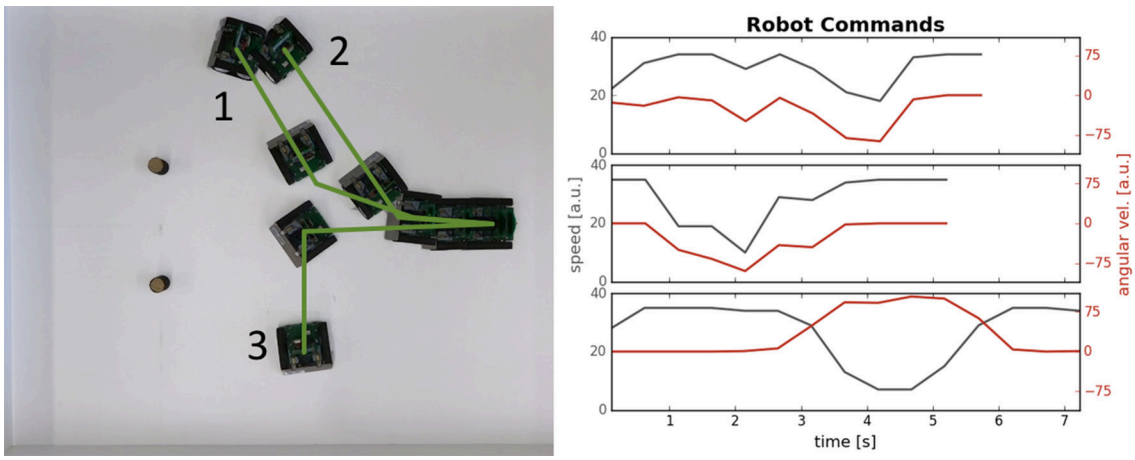


FIGURE 12 | Variability of the robot's behavior. **Left:** Overlay of video camera frames recording the robot, avoiding a pair of obstacles; top view. Three different trials are recorded and overlayed here (trajectories are shown with green lines 1–3). **Right:** Velocity commands, received by the robot from the neuronal architecture (angular velocity and speed) for the three trials (from top to bottom).

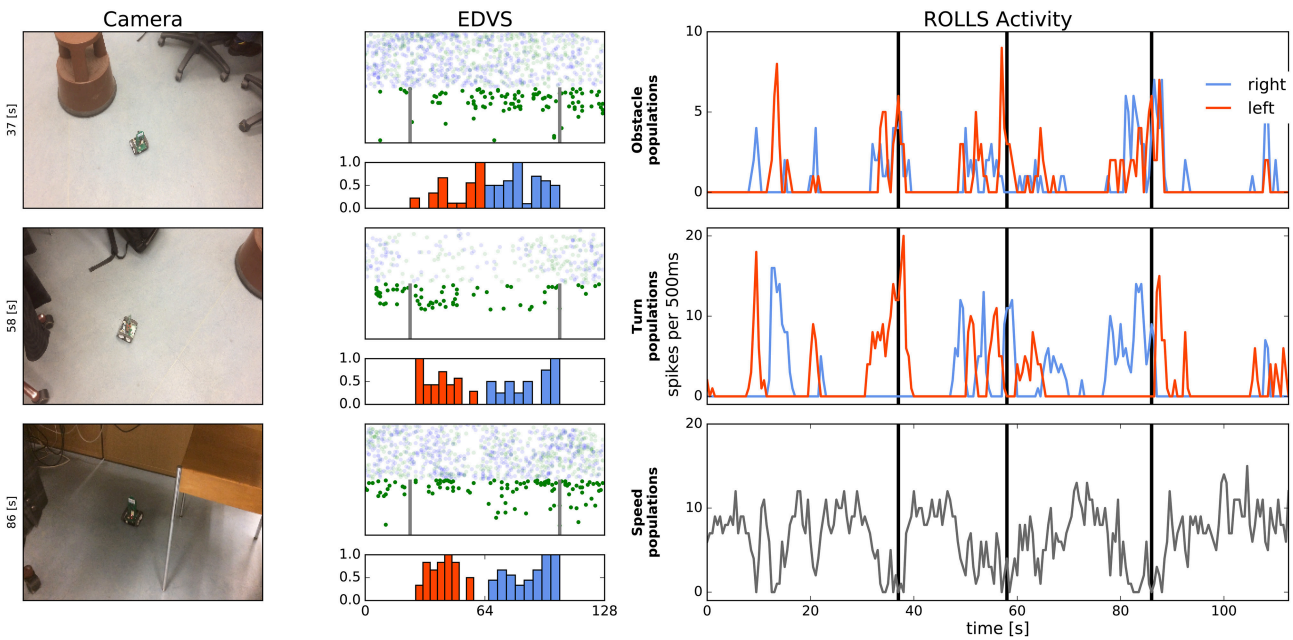


FIGURE 13 | Robot driving in the office environment. **Left:** Snapshots from the video camera showing robot at three time points during the experiment. **Middle:** Events from the DVS camera and histogram of these events, binned over 500 *ms* in columns in the region between two vertical lines, which were used to drive obstacle populations on the ROLLS. Each pair of the eDVS events and histogram corresponds to the time point of the video frame in the **Left** column. Note that 80% of events are randomly dropped here and only "on" events are shown in the relevant region (lower part of the screen). Events above the midline of the image sensor are shown with transparency (these events were not used for obstacle avoidance). **Right:** Activity of the obstacle (left and right), drive (left and right), and speed neuronal populations over time (summed activity across each population). Vertical lines mark time point that correspond to the video frames in the **Left** column.

differential equations that specify attractors and repellors for behavioral variables that control the robot's heading direction and speed (Bicho et al., 2000). In this framework, obstacle avoidance has been integrated with target acquisition and successful navigation in an unknown environment has been demonstrated both for vehicles and robotic arms (Reimann et al., 2011). This approach is similar to another successful reactive approach to obstacle avoidance: the potential field approach (e.g., Haddad et al., 1998), in which the target creates a global minimum in a potential that drives the robot, whereas obstacles create elevations in this potential. However, the use of Cartesian space instead of robot-centered velocity space used in

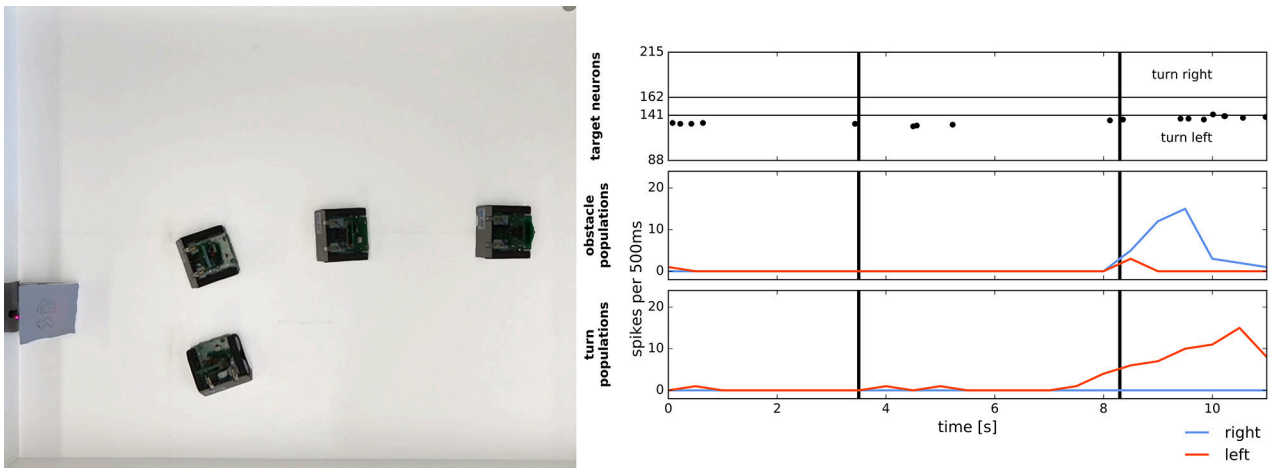


FIGURE 14 | Simple target acquisition: single stationary target. **Left:** Overlay of video frames from the overhead camera. The robot approaches a stationary target on the left-hand side of the arena from right to left. The robot turns left toward the target until it perceives it as an obstacle and makes an obstacle avoidance maneuver. **Right:** Time-course of the spiking activity (raster plot) of the target-representing (WTA) neurons on the ROLLS chip (top plot) and summed (over 500 ms and over populations) activity of neurons in obstacle representing and drive populations on the ROLLS chip. Vertical lines mark time points that correspond to two middle positions of the navigating robot.

this potential field approach makes it prone to getting trapped in local minima.

In mixed-signal analog /digital neuromorphic hardware, the neuronal dynamics is taken care of by the physics of analog electronic circuits, avoiding loosing digital computational resources on simulating them. Thus, neuromorphic implementation of simple biologically inspired obstacle-avoidance architectures can lead to low-latency (on the order of microseconds) and power-efficient (on the order of milliwatts) solutions, analogous to the ones used by insects. In contrast, more conventional obstacle-avoidance systems require a substantial amount of computing resources to process and store sensory data, detect obstacles, and compute motor commands. Neuromorphic implementation of such low-level processing will allow to use analog sensory signals directly, avoiding their digital representation and storage, while at the same time allowing to build complex neural-network based computing architectures, that could be used for solving cognitive tasks, such as task planning, map building, or object recognition.

We consider the work proposed as a first feasibility study, which still has a number of limitations that we will address in our future work. The main limitation is variability of neuronal behavior because of parameter drift on the analog hardware: the parameters of the hardware neural network change the network properties as the experimental setup conditions (temperature, humidity, etc.) change. This is a serious limitation of the hardware used, which makes in challenging to implement complex architectures that have to balance contributions of different behavioral modules (e.g., controlling turning and forward velocities, or obstacle avoidance and target acquisition). We are currently working on algorithms and methods for automatically re-tuning these parameters in a principled fashion with optimization and machine learning techniques. In addition, we are designing new versions of the neuromorphic hardware with on-board stabilization of the chip parameters, and more resources for simplifying the fine-tuning process of the architectures. However, approach employed here—use of populations of artificial neurons in place of single nodes in the architecture—allowed us to generate behavior with the state of the art analog neuromorphic hardware.

Apart from the hardware limitations, our simple architecture currently allows robust obstacle avoidance at moderate speeds (∼0.35 m/s). Since the robot slows down when an obstacle is detected, movement appears to be "jerky." Although the smoothness of the robot movement could be improved by tuning the coupling strength between the obstacle and drive populations, the best solution would involve improving the visual pre-processing stages. In our setup, the DVS detects local contrast changes and produces different amount of events depending on the objects in the environment, but also modulated by the robot translational and rotational movements. Currently we ignore about 80% of all DVS events to remove both noise and to reduce bandwidth. This very basic strategy improves the signal to noise ratio, because the architecture enhances the spatially and temporally coherent inputs and suppresses the effect of random inputs. However, we plan to study a more principled approach to pre-processing and noise reduction, and to investigate other biologically inspired architectures for obstacle avoidance, for example inspired by the fly's EMD (Elementary Motion Detector) (Hassenstein and Reichardt, 1956) or the locust's LGMD (looming detector Lobula Giant Movement Detector) (Gabbiani et al., 2002; Rind and Santer, 2004). We are currently working on neuromorphic implementation of these algorithms (Milde et al., 2016; Salt et al., 2017).

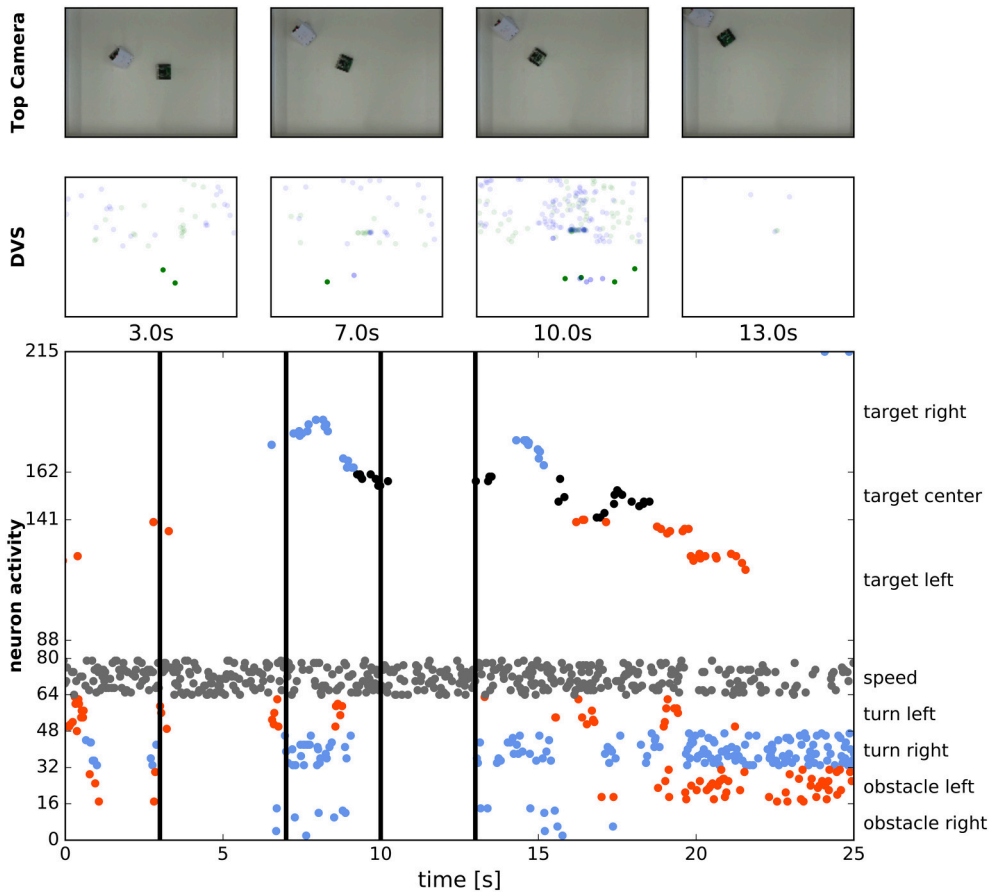


FIGURE 15 | Chasing a moving target. **First row:** Snapshots from the overhead camera showing the robot controlled by the ROLLS chip chasing a manually controlled robot. **Second row:** Summed eDVS events from 500 ms time windows around the same moments. Events from the upper part were used for target acquisition, events from the lower part—for obstacle avoidance. **Bottom row:** Spiking activity of all neurons on the ROLLS chip over the time of the experiment. Vertical line show the moments in time, selected for the first two rows. Red dots are spikes from the "left" populations and blue dots are spikes from the "right" populations.

Moreover, the 500 ms time window that we used to create plots of DVS events and average spiking activity was also used in our controller for counting spikes when calculating motor commands, sent to the robot. In our preliminary experiments on optimizing the controller, we have reduced this time window to 50 ms and, more importantly, replaced it with a sliding-window calculation of the average firing rate of the drive and speed neuronal populations. A more principled solution to this problem would be development of a more direct hardware interface between the spiking neuromorphic processor and the robot's motors, so that spikes can control the motor rotation directly, as suggested by Perez-Peña et al. (2013).

Our target acquisition network can also be further improved: the main strategy will be to introduce target representations in a reference frame that moves with the robot, but has a fixed orientation. Such representation will allow the robot to turn back to a target that has been lost from sight due to an obstacle avoidance maneuver. Furthermore, increasing the strength of lateral interactions in the WTA (DNF) population will allow to stabilize the target representation, allowing it to form a "working memory," which will support target acquisition behavior in cluttered environments. To still make the system reactive and allow it to follow the visible target, control of the strength of lateral interactions will be introduced, increasing their strength when target is being lost from view and decreasing their strength when the target is visible. Detecting the target based on its features perceived with a DVS is a separate topic of ongoing research both in our lab and worldwide (e.g., Lagorce et al., 2015).

Despite of this list of necessary improvements, our neuromorphic architecture is an important stepping stone toward robotic controllers, realized directly in neurally inspired hardware, being the first architecture for closed-loop robot navigation that uses analog neuromorphic processor and minimal preprocessing of visual input, obtained with a silicon retina DVS. Such neuromorphic controllers may become an energy efficient, fast, and adaptive alternative to conventional digital computers and microcontrollers used today to control both low-level and cognitive behaviors

of robots. While *neural network* implementations using the conventional computing architecture are typically time- and energy consuming, implementation of neuronal architecture using analog neuromorphic hardware approaches the efficiency of biological neural networks. Building neuronal models for higher cognitive function using, for instance, the framework of Dynamic Neural Fields (Sandamirskaya, 2013) or the Neuro-Engineering Framework (Eliasmith, 2005), will allow to add more complex behaviors to the robot's repertoire, e.g., finding a particular object, grasping and transporting it, as well as map formation and goal-directed navigation, which is the goal of our current research efforts.

AUTHOR CONTRIBUTIONS

MM: conceptualization of the model, analysis of the results, writing up. HB and AD: implementation of combined obstacle avoidance and target acquisition, experiments, results analysis, writing up; DS: implementation of first version of obstacle avoidance, parameter tuning on the chip, state of the art analysis; JC: support with robotic hardware and middleware, analysis of the results, writing up; GI: support with neuromorphic hardware, and state of the art and result analysis, writing up; YS: conceptualization of the model, development of the architecture, experiment design, analysis of the results, embedding in the literature, discussion of the results, writing, and overall supervision of the project.

FUNDING

Supported by EU H2020-MSCA-IF-2015 grant 707373 ECogNet, University of Zurich grant Forschugnskredit, FK-16-106, and EU ERC-2010-StG 20091028 grant 257219 NeuroP, as well as INIForum and Samsung Global Research Project.

ACKNOWLEDGMENTS

We would like to thank Aleksandar Kodzhabashev and Julien Martel for their help with the software code used in this work. This work has started at the Capo Caccia 2016 Workshop for Neuromorphic Engineering.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00028/full#supplementary-material

REFERENCES

Averbeck, B. B., Sohn, J.-W., and Lee, D. (2006). Activity in prefrontal cortex during dynamic selection of action sequences. *Nat. Neurosci.* 9, 276–282. doi: 10.1038/nn1634

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* 102, 699–716. doi: 10.1109/JPROC.2014.2313565

Bicho, E., Erlhagen, W., Louro, L., and Costa e Silva, E. (2011). Neuro-cognitive mechanisms of decision making in joint action: a human-robot interaction study. *Hum. Move. Sci.* 30, 846–868. doi: 10.1016/j.humov.2010.08.012

Bicho, E., Mallet, P., and Schöner, G. (2000). Target representation on an autonomous vehicle with low-level sensors. *Int. J. Robot. Res.* 19, 424–447. doi: 10.1177/02783640022066950

Blanchard, M., Rind, F. C., and Verschure, P. F. M. J. (2000). Collision avoidance using a model of the locust LGMD neuron. *Robot. Auton. Syst.* 30, 17–38. doi: 10.1016/S0921-8890(99)00063-9

Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press.

Brette, R., and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.* 94, 3637–3642. doi: 10.1152/jn.00686.2005

Brooks, R. A. (1991). New approaches to robotics. *Science (New York, N.Y.)* 253, 1227–1232. doi: 10.1126/science.253.5025.1227

Chicca, E., Stefanini, F., Bartolozzi, C., and Indivei, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* 102, 1367–1388. doi: 10.1109/JPROC.2014.2313954

Conradt, J., Galluppi, F., and Stewart, T. C. (2015). Trainable sensorimotor mapping in a neuromorphic robot. *Robot. Auton. Syst.* 71, 60–68. doi: 10.1016/j.robot.2014.11.004

Dean, P., Porrill, J., Ekerot, C.-F., and Jörntell, H. (2009). The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. *Nat. Rev. Neurosci.* 11, 30–43. doi: 10.1038/nrn2756

Douglas, R. J., Koch, C., Martin, K. A. C., Suarezt, H. H., and Mahowald, M. (1995). Recurrent excitation in neocortical circuits. *Science* 269, 981–985. doi: 10.1126/science.7638624

Eliasmith, C. (2005). A unified approach to building and controlling spiking attractor networks. *Neural Comput.* 17, 1276–1314. doi: 10.1162/0899766053630332

Erlhagen, W., and Bicho, E. (2006). The dynamic neural field approach to cognitive robotics. *J. Neural Eng.* 3, R36–R54. doi: 10.1088/1741-2560/3/3/r02

Ermentrout, B. (1998). Neural networks as spatio-temporal pattern-forming systems. *Rep. Prog. Phys.* 353, 353–430. doi: 10.1088/0034-4885/61/4/002

Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., et al. (2012). Overview of the SpiNNaker system architecture. *IEEE Trans. Comput.* 62, 2454–2467. doi: 10.1109/TC.2012.142

Gabbiani, F., Krapp, H. G., Koch, C., and Laurent, G. (2002). Multiplicative computation in a visual neuron sensitive to looming. *Nature* 420, 320–324. doi: 10.1038/nature01190

Haddad, H., Khatib, M., Lacroix, S., and Chatila, R. (1998). "Reactive navigation in outdoor environments using potential fields," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Leuven), 1232–1237.

Hasler, J., and Marr, B. (2013). Finding a roadmap to achieve large neuromorphic hardware systems. *Front. Neurosci.* 7:118. doi: 10.3389/fnins.2013.00118

Hassenstein, B., and Reichardt, W. (1956). Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus. *Zeitschrift fr Naturforschung B* 11, 513–524. doi: 10.1515/znb-1956-9-1004

Holland, O. (1997). "Gray walter: the pioneer of real artificial life," in *Proceedings of the 5th International Workshop on Artificial Life* (Cambridge: MIT Press), 34–44.

Iida, F. (2001). "Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision," in *Proceedings of the 32nd ISR* (Seoul), 19–21.

Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.* 17, 211–221. doi: 10.1109/TNN.2005.860850

Indiveri, G., Chicca, E., and Douglas, R. J. (2009). Artificial cognitive systems: from VLSI networks of spiking neurons to neuromorphic cognition. *Cogn. Comput.* 1, 119–127. doi: 10.1007/s12559-008-9003-6

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073

Indiveri, G., and Liu, S.-C. (2015). Memory and information processing in neuromorphic systems. *Proc. IEEE* 103, 1379–1397. doi: 10.1109/JPROC.2015.2444094

Khansari-Zadeh, S. M., and Billard, A. (2012). A dynamical system approach to realtime obstacle avoidance. *Auton. Robots* 32, 433–454. doi: 10.1007/s10514-012-9287-y

Koziol, S., Brink, S., and Hasler, J. (2014). A neuromorphic approach to path planning using a reconfigurable neuron array IC. *IEEE Trans. Very Large Scale Integr. Syst.* 22, 2724–2737. doi: 10.1109/TVLSI.2013.2297056

Lagorce, X., Ieng, S. H., Clady, X., Pfeiffer, M., and Benosman, R. B. (2015). Spatiotemporal features for asynchronous event-based data. *Front. Neurosci.* 9:46. doi: 10.3389/fnins.2015.00046

Lichtsteiner, P., Posch, C., and Delbruck, T. (2006). "A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers, 2004–2006* (San Francisco, CA). doi: 10.1109/isscc.2006.1696265

Liu, S.-C., and Delbruck, T. (2010). Neuromorphic sensory systems. *Curr. Opin. Neurobiol.* 20, 288–295. doi: 10.1016/j.conb.2010.03.007

Milde, M. B., Sandamirskaya, Y., and Indiveri, G. (2016). "Neurally-inspired robotic controllers implemented on neuromorphic hardware," in *Proceedings of IEEE International Conference on Biomimetics* (Bremen).

Mitra, S., Fusi, S., and Indiveri, G. (2009). Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *IEEE Trans. Biomed. Circuits Syst.* 3, 32–42. doi: 10.1109/TBCAS.2008.2005781

Moeys, D. P., Corradi, F., Kerr, E., Vance, P., Das, G., Neil, D., et al. (2016). "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on IEEE* (Krakow), 1–8.

Müller, G. R., and Conradt, J. (2011). "A miniature low-power sensor system for real time 2D visual tracking of LED markers," in *Proceedings of 2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011* (Hong Kong), 2429–2434. doi: 10.1109/ROBIO.2011.6181669

Neftci, E., Chicca, E., Indiveri, G., and Douglas, R. (2011). A systematic method for configuring VLSI networks of spiking neurons. *Neural Comput.* 23, 2457–2497. doi: 10.1162/NECO_a_00182

Perez-Peña, F., Morgado-Estevez, A., Linares-Barranco, A., Jimenez-Fernandez, A., Gomez-Rodriguez, F., Jimenez-Moreno, G., et al. (2013). Neuro-inspired spike-based motion: from dynamic vision sensor to robot motor open-loop control through spike-VITE. *Sensors (Basel, Switzerland)* 13, 15805–15832. doi: 10.3390/s131115805

Pouget, A., Dayan, P., and Zemel, R. (2000). Information processing with population codes. *Nat. Rev. Neurosci.* 1, 125–132. doi: 10.1038/35039062

Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Sumislawska, D., Indiveri, G., et al. (2015). A Re-configurable On-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141

Reimann, H., Iossifidis, I., and Schöner, G. (2011). "Autonomous movement generation for manipulators with multiple simultaneous constraints using the attractor dynamics approach," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai), 1050–4729. doi: 10.1109/icra.2011.5980184

Rind, F. C., and Santer, R. D. (2004). Collision avoidance and a looming sensitive neuron: size matters but biggest is not necessarily best. *Proc. Biol. Sci.* 271, 27–29. doi: 10.1098/rsbl.2003.0096

Sandamirskaya, Y. (2013). Dynamic neural fields as a step toward cognitive neuromorphic architectures. *Front. Neurosci.* 7:276. doi: 10.3389/fnins.2013.00276

Sandamirskaya, Y., Zibner, S. K. U., Schneegans, S., and Schöner, G. (2013). Using dynamic field theory to extend the embodiment stance toward higher cognition. *New Ideas Psychol.* 31, 322–339. doi: 10.1016/j.newideapsych.2013.01.002

Salt, L., Indiveri, G., and Sandamirskaya, Y. (2017). "Obstacle avoidance with LGMD neuron: towards a neuromorphic UAV implementation," in *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS* (Baltimore, MD).

Schöner, G., Dose, M., and Engels, C. (1995). Dynamics of behavior: theory and applications for autonomous robot architectures. *Robot. Auton. Syst.* 16, 213–245. doi: 10.1016/0921-8890(95)00049-6

Schöner, G., and Spencer, J. P. (eds.). (2015). *Dynamic Thinking: A Primer on Dynamic Field Theory*. Oxford, UK: Oxford University Press.

Stewart, T. C., Kleinhans, A., Mundy, A., and Conradt, J. (2016). Serendipitous offline learning in a neuromorphic robot. *Front. Neurorobot.* 10:1. doi: 10.3389/fnbot.2016.00001

Wilson, H. R., and Cowan, J. D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13, 55–80. doi: 10.1007/BF00288786

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Milde, Blum, Dietmüller, Sumislawska, Conradt, Indiveri and Sandamirskaya. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: info@frontiersin.org | +41 21 510 17 00

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership