



OPEN ACCESS

EDITED BY

Dioselin Gonzalez,
Independent Researcher, Seattle, WA,
United States

REVIEWED BY

Jean Elsner,
Technical University of Munich, Germany
Ping-Hsuan Han,
National Taipei University of Technology,
Taiwan

*CORRESPONDENCE

Soroosh Mortezaipoor,
✉ soroosh.mortezaipoor@tuwien.ac.at

RECEIVED 16 June 2025

ACCEPTED 26 August 2025

PUBLISHED 03 October 2025

CITATION

Mortezaipoor S, Ghazanfari M, Vasylevska K,
Vonach E and Kaufmann H (2025) Safety for
mobile encountered-type haptic devices in
large-scale virtual reality.
Front. Virtual Real. 6:1648019.
doi: 10.3389/frvir.2025.1648019

COPYRIGHT

© 2025 Mortezaipoor, Ghazanfari, Vasylevska,
Vonach and Kaufmann. 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.

Safety for mobile encountered-type haptic devices in large-scale virtual reality

Soroosh Mortezaipoor*, Mohammad Ghazanfari,
Khrystyna Vasylevska, Emanuel Vonach and Hannes Kaufmann

VR and AR Research Unit, Institute of Visual Computing and Human-Centered Technology, Faculty of Informatics, TU Wien, Vienna, Austria

Mobile robots are becoming more common in Virtual Reality applications, especially for delivering physical interactions through Encountered-Type Haptic Devices (ETHD). However, current safety standards and definitions of collaborative robots (Cobots) do not sufficiently address situations where an immersed user shares the workspace and interacts with a mobile robot without seeing it. In this paper, we explore the specific safety challenges of using mobile platforms for ETHDs in a large-scale immersive setup. We review existing robotic safety standards and perform a risk assessment of our immersive autonomous mobile ETHD system CoboDeck. We demonstrate a structured approach for potential risk identification and mitigation via a set of hardware and software safety measures. These include strategies for robot behavior, such as pre-emptive repositioning and active collision avoidance, as well as fallback mechanisms. We suggest a simulation-based testing framework that allows evaluating the safety measures systematically before involving human subjects. Based on that, we examine the impact of different proposed safety strategies on the number of collisions, robot movement, haptic feedback rendering, and noise resilience. Our results show considerably improved safety and robustness with our suggested approach.

KEYWORDS

virtual reality, mobile robots, safety, haptics, interaction

1 Introduction

With the recent advancements in quality and field of view of Head-Mounted Displays (HMDs), Virtual Reality (VR) environments are becoming increasingly immersive. In addition to visual immersion, haptic feedback can further enhance the user's sense of presence by allowing physical interaction with virtual objects (Sallnäs et al., 2000). However, placing a passive proxy object to overlap with the virtual one is not enough, as continuously aligning the objects is difficult. A specific class of haptic technologies known as Encountered-Type Haptic Devices (ETHDs) utilizes robots to position a proxy object that allows users to touch and interact with a corresponding virtual object as if it were physically present (Yokokohji et al., 1996; 2005). Compared to passive proxy objects, ETHDs provide more dynamic feedback and a wider range of simulation possibilities (Araujo et al., 2016). They can deliver high-fidelity tactile sensations without adding external load or restricting movement, unlike wearable haptic devices, which can reduce user comfort by introducing additional gear on the user's body (de Tinguy et al., 2020; Horie et al., 2021).

Previously, ETHDs relied mostly on stationary robotic arms to create haptic feedback within a limited volume (Mercado et al., 2021; Vonach et al., 2017). To improve scalability and enable large-scale haptic interaction, ETHDs have recently been integrated with mobile platforms, such as quadcopters (Yamaguchi et al., 2016; Hoppe et al., 2018; Abtahi et al., 2019; Abdullah et al., 2018) and wheeled robots (Suzuki et al., 2020; Wang et al., 2020). Although these mobile ETHDs (mETHDs) can be highly maneuverable, they cannot provide the forces necessary to simulate sturdier objects, like walls. For that, more powerful devices with substantial payloads should be employed, such as lower-tier industrial robotic arms. Robotic arms with reasonable payload are very heavy; thus, mobile robotic platforms able to provide enough stability and movability as a base for such arms need to be massive. Moreover, at a large scale, the robot needs to be autonomous to minimize the response delay when providing haptics. However, the high weight of such mobile robots makes collision risks more severe for the user.

Ultimately, the human-robot interaction (HRI) should be safe. Therefore, collaborative robots, also known as cobots, were introduced. This group of robots has advanced safety features built in, such as power and force limitations, and an emergency stop at contact. Nevertheless, in typical interactive industrial scenarios, cobots do not move while the user physically interacts with them and stop or move very slowly if the user just shares the same space. These safety measures add complexity and strongly restrict the possible interactions.

In a haptic VR scenario, the user needs to be physically collocated and touch the robot that provides the real-time haptic feedback. In a large-scale setup, such as a design studio, the VR user would need to freely walk and interact with the virtual content, requiring a mobile robot to quickly react to her intention to interact: (see Figure 1A). Apart from this, the HMD is effectively blindfolding the user from the physical world. Sharing space with any mobile robot the VR user cannot see makes her vulnerable to collisions and other hazards during close proximity interaction due to the robot's size and shape, its possible corners and edges, or any instability due to the load's weight, shape, and inertia: (see Figure 1B).

Although there are works on safety in HRI covering various user-aware path-planning algorithms and different sensors to avoid unwanted collisions, these efforts are largely focused on traditional

HRI settings with limited interactions (see Section 2.1). The combination of industrial experiences with safety research has led to the development of safety standards and industry norms. However, existing safety protocols are only partially applicable and sometimes not sufficient for the challenges posed by any ETHDs near a VR user. For example, a common safety measure in HRI is the maintenance of a substantial distance between humans and robots in a shared workspace. The industrial norms call for the robot to stop its operation or proceed at a considerably reduced speed if a human enters its space. However, neither is suitable for mETHDs.

We review the relevant studies and industrial standards further in Section 2. There seems to be no research that systematically addresses the safety of users in large-scale VR setups with mETHDs from design to complete implementation. This paper seeks to fill this gap by (1) conducting a comprehensive risk analysis of an mETHD utilizing a mobile robot carrying a robotic arm as an example case, identifying various safety risks associated with such systems. Based on the result of this risk analysis (2) we propose safety measures at both hardware and software levels, tailored specifically for close proximity HRI in mETHDs. Furthermore, (3) we describe a testing approach that enables systematic testing and validation of safety protocols. Using the implemented testing framework, we define a series of test cases to (4) evaluate and validate our proposed safety-oriented robot behaviors, demonstrating their effectiveness in enhancing safety in large-scale VR applications involving mobile autonomous robots. Although our mobile platform and robotic arm form a single unit, here we primarily focus on the mobile platform's safety, which is underexplored in research. We developed our safety approach for a special case of a ground-based heavy mobile robot, where safety is critical. However, it can be applied to other types of mobile robots like vacuum cleaners, suspended robots, and drones.

2 State of the art

2.1 Human-robot interaction and safety

Early studies in HRI mostly recommended segregating robot workspaces from human operators to prevent accidents

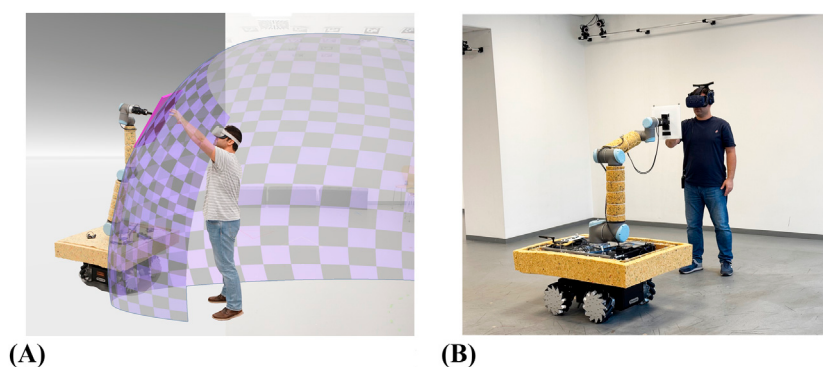


FIGURE 1
Close-proximity human-robot interactions in an mETHD system. **(A)** Overlays the virtual object that the user is interacting with in VR. **(B)** Shows the physical human-robot collaborative workspace. The user remains unaware of the position of the robot as she is immersed in VR.

(Engelberger, 1985). However, some early works suggest basic safety considerations for close collaboration between robots and humans. For example, Heinzmann and Zelinsky (1999) introduced simple and basic safety mechanisms for robotic manipulators, aiming to create “human-friendly robots” capable of operating safely near human users. For shared workspaces, Baerveldt (1992) introduced basic localization mechanisms to stop the robot when a human approaches, as well as emergency stop buttons mounted on robots to handle hardware or software failures. While these early safety measures are useful in general HRI contexts, they are very basic and fall short in applications involving more dynamic and close interactions.

Physical safety is recognized as absolutely necessary for human-robot collaboration (HRC). Ensuring physical safety requires careful consideration in the design of the robot itself (Giuliani et al., 2010; Casals et al., 1993), the environment (Michalos et al., 2015), and the control systems governing robot movements (Sisbot et al., 2007; Kulić, 2006; Haddadin et al., 2008; De Santis et al., 2008). Alami et al. (2006) reviewed general safety principles and challenges in physical HRI, while De Santis et al. (2008) provided a comprehensive review on safety standards, injury criteria, mechanical and control techniques, and planning for safe robotic operations. Haddadin et al. (2008) also examined strategies for collision detection and mitigation, experimentally validating techniques to reduce injury risk from physical contact. Haddadin et al. (2009) highlighted risks for humans interacting with industrial robots by systematically testing various industrial robots colliding with crash test dummies to document possible injuries from uncontrolled robot movements. Beluško et al. (2016) proposed visual work instructions to support safe industrial robot operation, enhancing both safety and productivity but potentially reducing immersion in VR applications, such as ETHDs. When considering autonomous mobile robots safety becomes even more critical due to the mobility and dynamic nature of these systems (Hata et al., 2019).

Beyond physical safety, the user’s feeling of security when interacting with a robot known as perceived safety, is vital in HRC. Factors influencing perceived safety include comfort, trust, and a sense of control (Akalın et al., 2022). Various methods to measure perceived safety, such as questionnaires, physiological assessments, behavioral analysis, and direct feedback devices, have been proposed (Rubagotti et al., 2022; Menolotto et al., 2024). Importantly, these factors may differ in VR applications, where users are immersed and unaware of a robot in close proximity, yet still required to interact with them.

According to ISO 10218 (ISO, 2011a), collaborative robots must undergo a risk assessment before being used. Although risk assessment and mitigation largely rely on expert knowledge, recently various tools and methods have been proposed to support risk assessment and reduction of cobots (Huck et al., 2021). Askarpour et al. (2016) introduced SAFER-HRC, a safety analysis method based on formal verification techniques to explore errors between human operators and robots. Chemweno et al. (2020) conducted a review of the state-of-the-art in risk assessment processes and safety safeguards. Semi-automated risk assessment methods are often effective in static environments, such as production lines with well-defined tasks. However, these approaches do not cover VR simulations, where users should

dynamically and continuously interact with robots to haptically experience the elements of virtual environments (VEs).

Zacharaki et al. (2020) conducted a comprehensive review of prior research in this field and proposed a systematic method for ensuring safety in HRC. In a more recent survey by Lasota et al. (2017), the authors categorized existing related works into safety through control, motion planning, prediction, and consideration of psychological factors. While these studies inspired our safety design, we optimize specifically for VR settings, where the user is effectively blindfolded by HMDs. Moreover, in applications such as mETHDs, physical contact between the user and the robot is intended and unavoidable. Therefore, unlike conventional HRC contexts, mETHDs require collision avoidance algorithms that distinguish between safe, intentional interactions and undesired contact.

2.2 Industry norms

As robots become more common in manufacturing, industry experts and organizations are actively developing safety standards for HRC applications. These efforts have led to the publication of various types of documents in the International Standards Organization (ISO) and the American National Standards Institute (ANSI). A brief comparison of related standards used in shaping our presented safety guideline is listed in Table 1.

The ISO 10218 series (ISO, 2011a; ISO, 2011b) provides guidelines for the safe deployment of industrial robots. It includes fundamental aspects such as robot safety requirements, risk assessment, and safety protocols to prevent accidents in industrial settings. The first part focuses on the safety requirements related to the robot itself, including specifications for design, construction, and integration to ensure safe operation in industrial environments. The second part extends this by emphasizing the safety of the overall robotic system, including the robot, its environment, and human interaction.

In addition to ISO 10218, ISO/TS 15066 offers specialized guidance on collaborative robot systems, emphasizing risk assessment and safety prerequisites such as power and force limitations and speed monitoring (ISO, 2016). A subsequent analysis discussing applications, requirements, and efforts to address the limitations of this technical specification was published in a report by Marvel from the National Institute of Standards and Technology (NIST) (Marvel, 2017). ISO/TS 15066 defines four types of HRC operations: Safety-rated Monitored Stop (SMS), Hand Guiding (HG), Speed and Separation Monitoring (SSM), and Power and Force Limiting (PFL). In SMS the robot halts its motion while a human is in the workspace. HG allows cobots to move under direct control from an operator. SSM permits cobots to move concurrently with the operator as long as they maintain a pre-defined distance apart. PFL requires cobots to have force feedback to detect contact with humans and react accordingly. Under this standard, when a human operator is in the shared workspace, the cobot must either remain stationary (SMS and HG) or move very slowly (SSM and PFL). Adhering to these constraints in VR might significantly reduce the performance of cobots in delivering haptic feedback as mETHDs. Furthermore, in scenarios where robots interact with visually isolated VR users, they must be able to respond quickly to avoid

TABLE 1 Overview of safety standards in Human-Robot Interaction: summarizing key details and differences across standards governing safety in industrial and collaborative robotics, personal care robots, and control systems.

Standard	Domain	Summary of coverage	Focus
ISO 10218-1 (International)	Industrial Robotics	Safety requirements for industrial robots and systems	Emphasizes guidelines for hardware and software design and construction of safe industrial robots
ISO 10218-2 (International)	Industrial Robotics	Safety for the integration of robots into specific systems	Safety of the whole robotic environment including integration and installation aspects, workspace requirements, and operator's safety
ISO/TS 15066 (International)	Collaborative Robotics	Guidelines for assessing risks in collaborative robot apps	Addresses risk mitigation measures like power/speed limitations (active measures) and rounded edges for contact areas (passive measures)
ISO 13482 (International)	Personal Care Robotics	Safety requirements for personal care robots	Focuses on safety for physical interaction between non-industrial robots and humans including speed limits and stop mechanisms
ANSI/RIA R15.08 (USA)	Autonomous mobile robots	Safety standards for autonomous mobile robots in industrial settings	Addresses Navigation, obstacle detection, and collision avoidance for safe autonomous mobile robot use
EN ISO 13849-1 (Europe)	Control Systems	Design of safety functions in control systems	Outlines guideline to design safety-related parts in control systems that can achieve the required performance level
EN ISO 13849-2 (Europe)	Control Systems	Test and validation of safety functions in control systems	specifies the procedures for the validation of safety-related parts in control systems including analysis and testing

collisions. This requires rapid movements conflicting with the speed limitations imposed by the standard.

For personal care robots, ISO 13482 (ISO, 2014) outlines safety requirements for non-industrial environments, addressing collision avoidance, impact force thresholds, and fault tolerance to ensure safe interaction. However, the standard is not suitable for VR applications, as it primarily focuses on healthcare robots, which have different characteristics and functions.

The European EN ISO 13849 series (ISO, 2023; 2012) focuses on the safety of control systems, delineating performance levels for safety-related aspects and offering guidance on designing and validating safety functions. The general risk assessment procedure for machinery is provided by ISO 12100 (ISO, 2010), forming the basis for our risk assessment.

In the United States, the Robotic Industries Association (RIA) issues the ANSI/RIA R15.08 standard series, which provides guidelines for the safe deployment of autonomous mobile robots in industrial environments (Robotic Industries Association, 2020; 2023). This series of standards emphasizes human safety through key principles such as navigation safety, obstacle detection, emergency stop systems, and the use of sensors to prevent collisions. While highly relevant for autonomous mobile robots used in mETHDs, its focus is on traditional industrial settings, not on the unique challenges posed by VR users interacting with mobile robots. As a result, while ANSI/RIA R15.08 can serve as a foundational reference, VR applications demand more stringent safety measures.

2.3 Safety in ETHDs in VR

Safety considerations in VR are crucial because users are isolated from the physical environment, making them vulnerable to various hazards like collisions (Kang and Han, 2020; Kanamori et al., 2018). The addition of cobots further heightens these safety challenges. One primary application of robots in VR is in ETHDs. Safety has been considered a critical factor in the design of many ETHDs. For example, Vonach et al.

(2017) and Mercado et al. (2021) outlined safety measures for their robotic arm haptic platforms, including speed and torque limits, emergency switches, and human oversight. Similarly, Dai et al. (2022) implemented speed and force limitations to minimize collision risks. Zhou and Popescu (2025) proposed a redirection strategy for static and dynamic objects to ensure safety by avoiding the user colliding with a moving ETHD. In addition to that, they argued that safety is enforced because users interact with the ETHD using a handheld stick rather than their bare hands. While this is valid, it may be less applicable in scenarios where direct haptic feedback with bare hands is required.

In mETHDs, the likelihood of unintended collisions increases. Common safety measures, such as emergency stop mechanisms with dead-man switches and visual emergency warnings to alert users of potential collisions, are discussed in related works (Abtahi et al., 2019; Yixian et al., 2020). More recent works proposed using an overseer equipped with an emergency stop button to monitor the entire system and intervene in case of safety risks (Gomi et al., 2024; Onishi et al., 2022). These mechanisms are further detailed and evaluated in our proposed guidelines. Some mETHDs incorporate hardware-specific safety mechanisms. For instance, Abtahi et al. (2019) developed a critical landing mechanism for drones, while Bouzbib et al. (2020) designed dedicated stop mechanisms for moving columns. Although effective, such solutions are highly tailored to specific hardware and are, therefore, unsuitable as general safety guidelines.

Using bumpers and soft pads around the robot, similar to what we proposed in this paper, is suggested by Weng et al. (2025). In this category, Horie et al. (2021) proposed designing safe end-effectors by covering their edges with soft materials to minimize the risk of harsh collisions with the user's body or the robot itself. However, this approach is not always practical, particularly when the end-effector must simulate hard materials such as stone or concrete.

Guda et al. (2020) tackled safety in ETHDs by developing a robot placement algorithm that ensures collision-free interaction between the user and the robot. This algorithm enables the end-effector to reach most areas of interest while maintaining user safety. However,

it is designed specifically for robotic arms and does not extend to mobile platforms.

Integrating visual feedback to increase user awareness and perceived safety has been suggested in the literature (Kästner and Lambrecht, 2019; Oyekan et al., 2019). At the same time, displaying the robotic ETHDs in VR might disrupt the user's immersion and distract from the simulated experience (Medeiros et al., 2021). For instance, Mercado et al. (2022) explored various visual feedback methods to raise user awareness of ETHDs, designing 18 safety techniques and establishing eight evaluation criteria to balance immersion and perceived safety. Similarly, Horie et al. (2021) introduced a 'safety guardian' that visualizes the robotic arm and end-effector in VR to prevent collisions. Hoshikawa et al. (2024); Gomi et al. (2024) suggested robot visualization to enhance safety for VR users, but both reported breaks in immersion in their studies. A review of visual feedback methods to improve awareness in VR, along with evaluation criteria for ETHD applications, is presented in (Garcia, 2021). We have also adopted this technique to alert users in emergency situations, despite its potential to affect immersion.

Bouzbib and Bailly (2022) presented a structured approach for identifying a wide spectrum of failures and potential risks in ETHDs, accompanied by proposed solutions to mitigate them. Although safety risks are included within the discussed failures, the focus is more on outlining initial mitigation actions rather than developing a comprehensive safety guideline specifically for ETHDs.

To summarize, there are a number of works that partially addressed safety in ETHD design. However, the literature still lacks a comprehensive guideline or an in-depth exploration of safety for mETHDs in VR. This work aims to bridge that gap by proposing safety considerations specifically tailored for VR applications, with a focus on mETHDs.

3 Safety requirements analysis

The ISO/TS 15066:2016 emphasizes the importance of conducting a comprehensive risk analysis for collaborative robotic systems to ensure safe human-robot interaction. In line with these guidelines, we analyzed possible risks in large-scale VR systems with mETHDs and human actors, taking the CoboDeck as a case example.

CoboDeck is an immersive VR autonomous mETHD system, where VR users can freely walk and interact in a large virtual environment. The employed collaborative robot RB-Kairos consists of an omnidirectional mobile platform with a robotic arm, each controlled by a separate onboard computer unit. It moves in real space and provides user interactions with virtual objects. For that purpose, the robot presents a physical prop in the real environment that corresponds to the position and orientation of a virtual counterpart. The robot's autonomous operation is controlled over the Robot Operating System (ROS) by several state machines called "behaviors", between which the robot switches depending on the situation. Unity 3D simulates the virtual space, where the user can interact with the presented virtual objects naturally by hand, unaware of the robot's position or actions. For further details on the framework architecture and implementation, please refer to (Mortezapoor et al., 2023).

The robot is equipped with a variety of sensors and subsystems to perceive its own state and surroundings, i.e., Light Detection and Ranging (LiDAR) sensors, an Inertial Measurement Unit (IMU), and internal sensors for odometry, several localization subsystems, cameras, and many more. As required by ISO 13849-1, the robot's key systems are redundant, so the robot has multiple ways to perform a particular function if one system fails. As a cobot, it also has some common safety measures, such as remote emergency stop buttons and LiDAR-based obstacle detection. As they are implemented by the manufacturer, they will not be discussed in this work.

To perform the analysis in a systematic way, one researcher created an initial document, collecting possible hazards from their own experience, related work, and interviews with four experts regarding Cobodeck. First, the actors involved in the interaction scenario were identified: the user, robot, and overseer. Then the functions and risks of the actors were listed. In a large table, the variations of possible risk-associated situations were listed and assessed for the severity of the risk as well as the probability of its occurrence. On this basis, the four experts individually extended the document with comments, additional hazards, and their initial subjective assessment applied to Cobodeck, which the researcher, as a meta-expert, merged into one document. Finally, the four experts plus the meta-expert finalized the document in an extensive group discussion, identifying over 100 individual hazardous scenarios. The hazards include unintended contact between the user and robot that could lead to an injury, particularly in sensitive regions (Park et al., 2019), but also situations that could lead to damage to equipment or the environment. For each actor, we summarize the most notable identified risks below.

3.1 System's actors and their risks

3.1.1 VR user

The VR User, or *user* for short, is an individual immersed in the virtual environment via wearing a tracked HMD and moving freely within the physical space, without seeing it. One of the primary safety risks is an unexpected or too quick user's motion to which the robot cannot appropriately react or might overreact, e.g., avoid the user in the wrong direction. The user may experience sudden physiological symptoms, such as cybersickness (LaViola, 2000) due to VR exposure. Delays or irregularities in visualization or tracking data may also lead to unexpected user behavior. Faulty tracking or incorrect localization of the user might also increase the collision risk. In such scenarios, timely intervention may be necessary to mitigate potential risks to the user's health, prevent injuries, or avoid further damage.

3.1.2 Overseer

An overseer is an individual responsible for monitoring the system from outside and ensuring safety during operation. She monitors both the physical workspace and the system's operational information on the monitors. The information includes the robot's current behavior, the VR scene with the user, and the robot's motion plan. In the event of a hazardous situation, the overseer has to intervene and stop the user and the robot.

One potential risk for the overseer is the misinterpretation of the workspace situation. Given the complexity of the monitoring task, there is also a chance of missing critical parameter changes or failing to intervene in time. Furthermore, although the overseer should remain outside the workspace, entering this space poses additional risks. In such a case, she could only be detected by the robot's LiDAR sensors. If the sensors malfunction or lack sufficient resolution, a collision could occur. Other risks include the overseer's unexpected illness, power failure for the stationary computers, or low battery levels in the wireless emergency stop device. Although cobots are designed to fail safely with emergency stop if the device becomes inactive or disconnected, there is still a delayed detection risk.

3.1.3 Robot

The robot refers to an autonomous mobile platform that supports movement in any direction without rotation and features a robotic arm for serving props for interaction with the user. The robot can reach speeds of up to 3 m/s in all directions, with acceleration and deceleration rates of $\pm 3 \text{ m/s}^2$, allowing for rapid motions. Unlike stationary robotic arms, the mobile platform has a new set of risks during operation that we focus on.

A key risk associated with the robot is the potential for uncontrolled loss of traction. Rapid velocity or acceleration changes can cause the robot to slide if ground grip is insufficient or tip over due to a sudden redistribution of momentum, especially with the robot arm stretched out, which increases the risk of injury. Sensor and subsystem failures pose additional risks. Delays or inaccuracies in data from the LiDAR sensors or the localization subsystem can result in incorrect robot behavior. For instance, the robot's delayed response to changes in the user's or its own positions, or an obstacle, might lead to collisions. The limited resolution of LiDAR sensors can interfere with the robot's situational awareness. Failing to detect the legs of a chair or dirt on the sensor might create a non-existent obstacle. A low battery level can render critical functionalities unreliable or inoperative, posing further safety risks. Moreover, the computers responsible for controlling the robot's behavior could crash, leading to system-wide

failures. Errors in the robot's behavior engine may also result in unpredictable reactions, creating a dangerous situation where only overseers can intervene in time to prevent harm.

Furthermore, the props presented by the robot also introduce risks. They may detach and fall if not securely mounted or if their weight exceeds the robot arm's capacity. A loose prop in motion could be propelled towards nearby individuals, posing a significant hazard. Furthermore, if the robot moves with its articulated arm outstretched, it may risk injuring users or damaging itself through collisions with environmental elements.

3.2 Risk assessment matrix

We analyzed the apparent dangers with a risk matrix—a table with the probability of occurrence on one axis and the potential extent of damage on the other. This matrix is a visual tool to summarize the risk evaluation. Note that the system was analyzed in its basic functional state with general sense and common safety features implemented, such as the cobot intrinsic safeties. Therefore, they are excluded from the table. [Table 2](#) shows a shortened version of the resulting matrix, where several scenarios were combined to fit in this paper. The probability of occurrence is divided into six levels, from “Frequent” to “Inconceivable”. We specified that “Frequent” corresponds to the occurrence of this task or scenario 7 to 10 times within 1 h of normal operation. “Probable” 4 to 6 times and “occasional” 0 to 3 times in 1 h. “Imaginable” is a situation that can occur, but is not probable within 1 h of normal operation, while “Unlikely” should not be possible in 100 runs. “Inconceivable” describes the likelihood of situations that should never occur. In addition, we defined the extent of possible damage ranging from “catastrophic” (user would be injured or objects would be irreparably damaged) through “serious” (damage to the robot, prop, or room, which can be repaired) and “minor” (does not need repair) to “insignificant” (needs no further attention). The combination of the probability and severity of the damage leads to the evaluation of the risk severity: critical, high, medium, and low.

TABLE 2 Shortened Risk matrix: An overview of the potential extent of damage and probability of occurrence of identified risks of the Cobodeck.

		Extent of damage			
		Catastrophic	Serious	Minor	Insignificant
Probability of occurrence	Frequent	Incorrect localization of robot or user		Uncontrolled loss of traction of robot	
	Probable		LiDARs resolution not sufficient		Power failure or low battery level of devices
	Occasional	User moves unpredictably or fast		Communication lags or errors	User experiencing cybersickness
	Imaginable		Overseer misinterprets situation	Unexpected activation of emergency stop	Overseer in scene
	Unlikely	Failure of various sensors	Robot may tip over	Failure of VR glasses	
	Inconceivable	User's hands touch robot unexpectedly			Failure of prop

The risk severity is color coded: critical - red, high - orange, medium - yellow, and low - green.

The goal of the risk assessment is not only to identify the risks but also to plan for the actions that will reduce the risks' severity to As Low As Reasonably Practicable (ALARP) (Jones-Lee and Aven, 2011; Langdalen et al., 2020).

For instance, there is an imaginable hazard of the robot driving over the user's foot. Providing metal-reinforced builders' shoes would reduce the extent of damage from "high" to "low". For a more complex case, consider interaction. To facilitate interaction, the robot positions itself frontally to the user and positions a prop within the arm's reach of a person. The maximum distance to enable any interaction with extended robot and human arms would be 188 cm with low or "imaginable" injury probability. Reducing the distance increases the possible interaction range has a trade-off of increasing risk. Standing directly next to each other, the interaction volume would be the largest possible, but the risk of serious injuries or unintentional collisions gets high or "catastrophic" since sensitive areas such as the head and neck are within the action radius of the robot arm. Consider, the robot's footprint with its arm parked has a radius of 77 cm, but increases to a maximum of 147 cm if the arm is extended horizontally. Therefore, adding to a safety distance 50 cm reduces the volume for interaction, but enables the user to leave the action radius simply by pulling back the arms. This way, the sensitive areas of the body will be outside the interactive volume by default, and the risk of the user touching the robot will be reduced from high to low. Similarly to the examples above, we analyzed all identified risk-associated scenarios to create a number of strategies for risk mitigation in the form of specific safety measures discussed in this paper.

4 Safety measures

A group of risks could be addressed on the hardware level, such as LiDAR resolution and communication. We particularly considered the detection of thin objects such as chair legs. Therefore, we selected devices with a wide field of view (270°) and fine resolution (0.5°). Similarly, we facilitated the communication with wireless multi-channel WiFi six supporting throughput up to 4.8 Gbit/s. For timely delivery of haptic feedback over a large area, the robot's speed and acceleration need to be much higher than stipulated in existing safety guidelines. We made sure to have an option for fast movements of the robotic platform with speeds up to 3 m/s and acceleration up to 3 m/s^2 . The speed's tradeoff, as well as other means for safety, are discussed in more detail below. Since the speed of the robot poses a risk of tipping over, especially so if the robotic arm is extended, in situations that are not emergency-triggered, the arm should be folded and parked before fast motion is performed.

4.1 Mechanical protection: safety bumper and impact sensors

Fast motion and the potential risk of losing grip on the floor surface might lead to unintended contact, such as brushing by or a low-impact collision. Our RB-Kairos robot is environment-aware via LiDAR sensors but has no means to either detect or soften a collision. Another concern is the protection of the robot's enhancing

devices and sensors and their connections to the physical interfaces of the robotic platform, in the event of unintended contact.

To serve these purposes, we implemented safety bumpers with two primary features. One feature is a high-sensitivity rapid impact detection system, ensuring that the robot can take immediate action to minimize potential harm. Depending on the situation, this could involve activating emergency stop or redirection of movement, to prevent further contact. The second feature is the composition: the equipment should have a rigid protection case while obstacles and the user require impact-dampening cushioning with soft, yet dense and elastic material. It should be also compact and light enough to avoid affecting the acceleration, traction, and floor grip of the robot due to the extra weight. A composite foam is an example of an ideal material for that.

For our RB-Kairos robot (see Figure 2A), we created a suspended wood and foam frame around the platform. The wooden part provides stability and protects the wired connections on the robot (see Figure 2B). On it, we mounted equally distributed arrays of push buttons for impact sensing (see Figure 2C). To ensure the reliable work of the sensors, we attached small plates of thin 2 mm plastic plates over them at a distance created with springs to guarantee that on impact more than one sensor will be triggered. Those were, in turn, covered with a 5 cm thick composite foam as a cushioning layer (see Figure 2C). The robot with bumpers is depicted in Figure 2D. The impact sensors were then connected to a PJRC Teensy 4.1 microcontroller attached via USB to the mobile platform's computer. The controller processes the triggering of the sensors in any of the sensor clusters and triggers an emergency stop on the robotic platform.

4.2 Tracking redundancy: co-localization accuracy and reliability

Accurate and precise co-localization of both the user and the robot is essential for haptic interaction within the shared space. Localization issues, such as delayed or damaged data, can lead to erratic or unintended robot behavior, affect the user's experience, and introduce safety risks, especially from the robot's side. To address these risks, robust localization strategies are required.

To enhance reliability and mitigate the risks, we propose employing multiple localization systems operating in parallel, with their data inputs fused through a Kalman filter. This approach not only compensates for the inherent errors of individual subsystems but also ensures fault tolerance, providing a safety net in case one localization method fails. Such data fusion ensures safer and more seamless interactions within VR environments.

In CoboDeck, we employed several localization solutions, with their outputs fused using an Extended Kalman Filter (Moore and Stouch, 2016) to ensure precise robot positioning. The first solution utilizes the robot's dual LiDAR sensors combined with a pre-scanned map of the environment for localization, implemented through Adaptive Monte Carlo Localization (AMCL) (Fox et al., 1999). The second solution involves a scalable, low-cost marker-based GPU tracking system, where large markers are mounted on the ceiling to provide sub-centimeter precise tracking (Podkosova et al., 2016). The third solution uses HTC Vive Lighthouse tracking

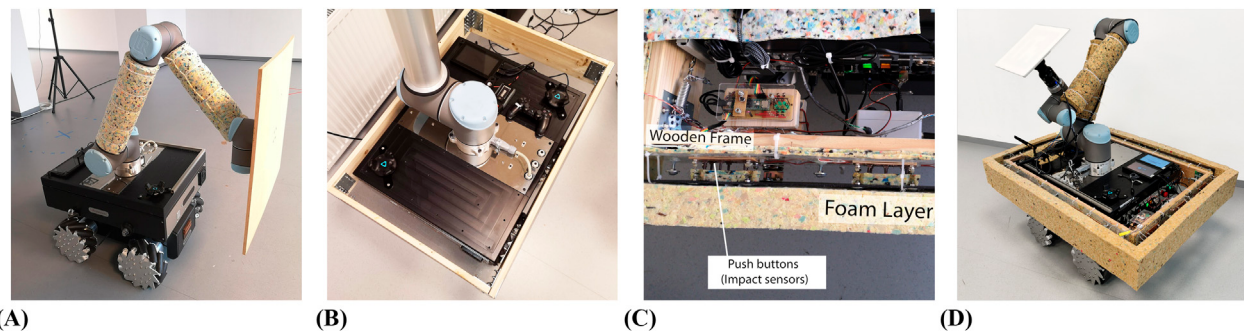


FIGURE 2

An RB-Kairos robot with safety bumpers and impact sensors. (A) The robot in its original shape without safety bumpers. (B) The wooden frame around the robot is a solid bumper part. (C) The safety bumpers' top view, the impact sensors array placed between the wooden frame and the external foam layer. (D) The robot with complete safety bumpers.

(HTC and Valve, 2016) with four base stations and two HTC Vive V2 trackers attached diagonally onto the mobile platform to provide submillimeter accuracy. Note that each solution can also be replaced as needed, as the Vive tracking will be replaced within the next iteration of the project. The Extended Kalman Filter is a fail-safe localization mechanism working at 30 Hz that accounts for different precision and noise levels in the localization solutions enhancing the final localization precision and robustness of the robotic system. Due to the real-time rendering requirement for high-frequency tracking (minimum 90 Hz), the user currently cannot profit from such fusion and should rely on the best available tracking solution that is constantly checked for quality.

4.3 Positioning strategy for large scale setups

For large-scale haptic VR systems, robot positioning dynamics affect both user safety and system efficiency. The robot should not remain at the site of its last interaction until the next one. At first, it would be too close to the user, creating a potentially dangerous situation. This situation will be addressed in the next section. Then

later, it might be too far from the next interaction point, introducing a substantial response delay, requiring the use of higher speeds for positioning with subsequent issues. Given the user's freedom to move in any direction and distance, a strategically positioned robot nearby might minimize response time and ensure a smooth user experience while maintaining the safety of the workspace.

To tackle this challenge, we propose the robot to “Chase” the user. Under this strategy, the robot follows the user, maintaining a short but safe distance, making itself available for haptic interaction on short notice. We suggest a user-centric weighted *chase space* consisting of two arches as shown in Figure 3A for the robot's positioning near the user. The formula for the weights is outlined in Equation 1 below:

$$W(r, \theta) = \begin{cases} \sin\left(\frac{r - d_{\min}}{d_{\max} - d_{\min}} \times \pi\right) \times \cos\left(|\theta| + \frac{\pi}{4}\right), & d_{\min} \leq r \leq d_{\max}, \frac{\pi}{6} \leq |\theta| \leq \frac{3\pi}{4} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The function $W(r, \theta)$ returns the weight of a point in the polar coordinate system centered at the user's position where r represents the radial distance from the origin, and θ denotes the angular position relative to a user's torso forward direction, ranging from

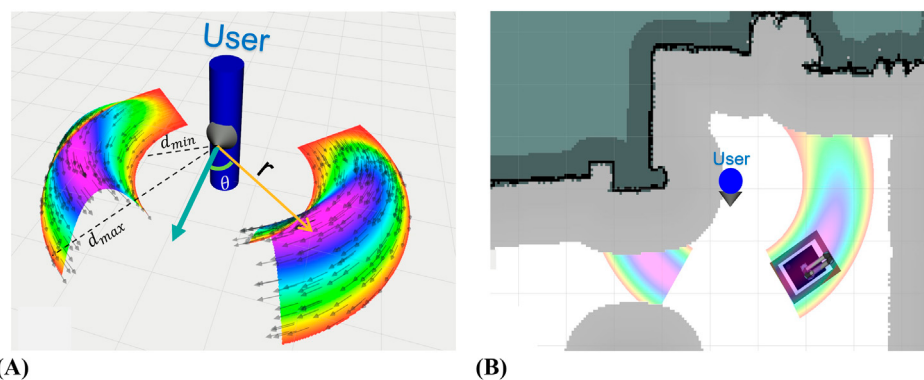


FIGURE 3

Chase space. (A) Full chase space on both sides of the user. The elevation and coloring depict the weight distribution, where higher/purple is preferable. The arrows depict the robot's orientation. (B) Chase space in use excludes areas occupied by obstacles or their buffer zones.

30° to 135°, and −135° to −35° respectively. Here, d_{min} and d_{max} should be set according to the specifications of the robot and the environment. The absolute minimum of d_{min} value is the sum of the robot's bounding circle radius (see Figure 6B for an example) and the radius of the space needed for an idle user (see the blue circle around the user in Figure 6C). We suggest using d_{min} no less than 150% of the minimum value. For d_{max} , we suggest using $d_{min} + 1.5$ m or more to have a sufficient amount of options.

When in use, the parts of the space that are obstructed by obstacles (static or dynamic) should be excluded from consideration, as shown in Figure 3B. This approach avoids interference with the user's free movement and safety, but also ensures that the robot is consistently positioned optimally for serving the interaction.

Implementing the "Chase" strategy has multiple benefits. We showed in Mortezapoor et al. (2023) that it reduces the total distance traveled by the robot, which in turn conserves battery life. It also significantly improves the response time for haptic interactions. Moreover, this strategy enhances safety - the robot actively maintains a safe distance from the user and does not obstruct the user's direct walking path, thus reducing the risk of collisions. Through this user-aware re-positioning, the robot not only becomes more efficient and safe in its role but also contributes to a more seamless haptic VR experience.

4.4 Fast collision prevention strategy

In haptic VR, the user is in close proximity, and interaction dynamics and the environment itself result in a variety of possible collision scenarios. In the basic scenario, the VE is static and the objects' positions are known and remain unchanged. The robot can be strategically positioned in safe zones when not interacting with the user and positioned inside the virtual obstacle during interaction. These safe positions would need to be reevaluated after each VE or workspace change. However, when the robot travels to or from the interaction, unexpected collisions still can occur, necessitating a fast response and repositioning.

In contrast, dynamic environments, such as a design studio with dynamic behaviors and content, present a more complex challenge. The robot must adapt in real time to new elements and layout alterations. Alternatively, the content might be dynamic. The user might open the door she touched a second ago and try walking through it while the door simulation robot is still blocking her path. In such dynamic scenes, it is critical for the robot to instantly and safely reposition itself in response to the user's actions, prioritizing her safety over everything else.

To achieve instant reaction, the regular safety protocols and the robot's operational parameters should be readjusted and extended. This includes decreasing the standard obstacle buffer size to gain more space for navigation around obstacles or increasing acceleration limits for the agile reaction. To save time for navigation planning, there also should be precalculated options to escape and reestablish the minimum safe distance from the user.

In our proposed strategy, the potential escape positions away from the user and her potential walking path should be computed concurrently and prioritized based on real-time assessments of the workspace situation (see Figure 4). Factoring priorities include the current extended occlusion map (obstacles and their buffer zones),

position, and orientation of both the user and the robot. Figure 4A provides a top-down view of a scenario with an idle robot near an approaching user. Here, the small red arrows show 19 possible escape positions in a constellation shaped to correspond best to the space and the robot's shape and maneuverability.

Although other constellation shapes and sizes are possible, in the case of Cobodeck and its real workspace, this version performed the best. This constellation comprises nine regions (see Figure 4B) with different priorities, from which the highest priority goes to R1 and the lowest to R9. The high-priority regions allow the robot to have more time to empty the direct walking path while not driving too far from the user. These are picked to be far enough from the current position of the robot to allow high initial acceleration, and enough space for deceleration for a quick escape. The escape positions' accessibility depends on static and dynamic obstacles in the workspace, leaving only feasible positions for path planning. When masked with obstacles or their buffer zones, regions with lower priorities are taken, although they might lead to further distances to the user after the escape is performed. When multi-part regions are available (R2, R3, and R7), the robot prefers the sub-region that is further from the user's position. This applies to the escape destinations within each region in general as well, when multiple inner-region escape destinations exist. For an omnidirectional robot, the orientation alignment to the robot's current orientation leads to a faster escape, as rotation takes time.

Upon detecting a potential collision, the robot escapes from the user to the highest-priority escape position. In Figure 4C, this process is visualized, showing how some escape positions are excluded due to obstacles.

4.5 Actions at autonomous operation limits

Despite the measures implemented to enable the robot's autonomous risk management, it may encounter situations beyond its resolution capabilities. Scenarios such as extremely constrained maneuvering spaces or unexpected functional delays present such challenges. The "Inform" strategy is designed to bridge the gap between autonomous robot functionality and the need for human intervention or awareness with the following sequence of means:

Robot visualization in VR: Disclosing the robot's physical position enables the user to adjust her actions to resolve an issue without leaving VR. However, the immersion might be compromised.

Notification to the overseer: In parallel, the overseer alert ensures that complications are promptly addressed by a trained professional.

Human-safe robot recovery action: A predefined robot action can be executed when standard autonomous responses are insufficient, but the situation might still be resolved without the need for the overseer's intervention. For instance, the robot's recovery action can be to stay still until an escape route becomes available as the user steps out of the way.

Emergency robot stop: This feature is for scenarios with an immediate threat to the user's safety and when a rapid halt preserves equipment, as in the case of a severe malfunction. It might go beyond the human-triggered functionality with

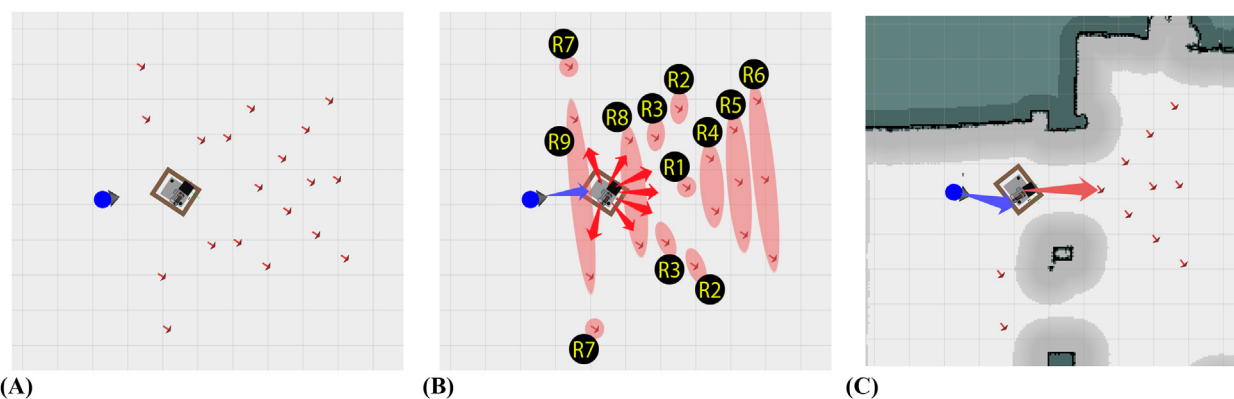


FIGURE 4

Escape direction calculation. The small red arrows around the robot indicate possible escape positions. (A) The robot in the escape situation with all possible escape points available. (B) If the user walks toward the robot (blue arrow), the escape positions are prioritized based on escape regions (lower region number gains higher priority). (C) Escape situation with occlusions: the robot moves toward the best available escape position within the escape region with highest priority.

programmatic triggers, e.g., based on self-test results. The stop can be canceled only by the overseer.

Notification for VR disengagement: Some situations might require an alert for the user to exit the VR momentarily by removing the goggles. This measure applies when immediate user awareness of the real-world environment is necessary and should be complemented by prior instructions.

4.6 System health monitoring

In close human-robot interaction, especially if the user cannot see the robot, the presence of an external expert observer (overseer) is indispensable. The overseer should be situation-aware, monitoring the state of the robot's and user's hardware and software systems, and understand user-triggered responses.

From the robot's side, the overseer needs to have an overview of the status of the robot's onboard computer, internal inter-components network connection, ROS Master process, and the function and data flow of the sensors as well as subsystems such as localization, navigation, and the arm's Inverse Kinematic (IK) planner and joint synchronization. From the user's side of the system, the overseer needs to ensure the healthy function of the VR computer and its operating system, network connection, ROS data synchronization, VR hardware, backend, and application. In parallel, the overseer needs to constantly monitor the physical workspace, the robot, and the user.

To accomplish this complex task and for better visibility, we suggest a portable screen or, even better, standalone AR glasses to allow seeing the workspace with overlaid necessary information, as we presented the concept in Mortezapoor et al. (2025). The status monitoring solutions should operate independently of the robot's onboard or user's computers, to ensure continuous and reliable observation, should any part of the system fail. Additionally, this monitoring solution can also have semi-automatic capabilities. In certain scenarios, such as major failures, it can have the authority to

trigger a hardware-level emergency stop of the robot and notification of the user, thereby providing an additional layer of safety.

4.7 Pre-deployment testing

Before the safe operation is established, a real user cannot be collocated with the robot due to health concerns. However, the user's behavior can be simulated. Similarly, the robot's software solutions should be extensively tested in simulation before deploying to the real robot. For mobile robots, this also saves battery-life and decreases the amount of real tests that are necessary for simulation results verification.

To confirm that the robot behaves as expected and fulfills its safety and performance requirements from the end user's viewpoint, we chose a black-box testing paradigm (Beizer and Wiley, 1996). For that, the user behavior is defined for a test under various scenarios and only the robot's responses are monitored. The integration between the user and robot subsystems should be seamless and the same for both real and simulated operations. This means that the robot responses can be tested with real or simulated users and robots in all possible combinations with minimal differences. The safety testing should start with a full simulation of both and progress towards full reality.

We defined a template shown in Table 3 (more details in Supplementary Appendix 1), for the creation of flexible and comprehensive test scenarios, that include all necessary parameters, and to tailor testing conditions to specific cases. The template is easily translatable into code, enabling efficient implementation. This way, the system's performance can be systematically evaluated, even under extreme or unusual conditions, maximizing failure coverage and enhancing overall robustness. In addition, test-specific criteria can be introduced depending on the scenario, e.g., a criterion specifying maintenance of a minimum safe distance (e.g., 1.5 m) during robot-user interactions. The test passes the general safety

TABLE 3 Test case definition template.

Field	Description
Test Category	Testing target environment (e.g., simulation/physical)
Description	Test case overview, its primary purpose, including essential considerations
Expected Hazards	Potential hazards that could arise in the defined scenario
Pre-conditions	Necessary settings of the workspace before starting the test
Disturbance ❖	Set of disturbances applied to the testing environment in sequential runs
User Behavior ❖	Set of user behaviors, including user trajectories, interaction points, and engagement duration in sequential runs
Steps to Perform	Step-by-step actions to perform to simulate the testing scenario
Assertion Data Fields	Additional data to be collected during and after the test
Acceptance Criteria	Criteria to determine the outcome of the test runs

Fields with ❖ are lists of items and can contain multiple values that might represent different test conditions.

requirements if no robot-user collisions occur for either actor and the robot avoids collisions with the workspace.

Based on the defined template, we implemented a Unity-based testing framework. The framework provides test case definition, user behavior replay and simulation, and scenario repetition for data collection and analysis. Similar to VR setup with ROS synchronization, it allows testing in both simulated and real-world environments. The user properties and trajectories combined together with environmental parameters according to the test case template, are stored in JSON files, allowing for consistent and reproducible test executions. Repeatable tests and result logging further aid in optimizing subsystems and improving system performance. The major components of the testing framework are:

- *User Behavior Manager* defines user behavior for test cases. They include the interaction positions in VE, the user's walking paths and patterns. The paths can be generated in various shapes (straight, arches, zig-zags, etc.), hand-drawn trajectories, real walking recordings, or recording modifications. For testing haptic responses, passive hand positions or recorded real hand movements can be used.
- *Test Runner* is responsible for executing the test cases. It loads the JSON files containing the tests and their variations, and iterates through the test cases, driving the simulated user along the predefined paths of each test case. During execution, the Test Runner monitors the testing criteria, enabling systematic evaluation of the robot's performance under various conditions. When a discrepancy between the expected and actual behavior of the robot is detected, it logs incidents in detail. This includes storing both numerical data and visual snapshots from the current scene in Unity, providing a comprehensive basis for further analysis and troubleshooting.
- *Test Scene Decorator* prepares the testing environment by setting chosen space limits and positioning virtual objects

within the scene according to the test case. If adjustments to the scene are necessary once the test sequence is in progress, the Decorator can pause the simulation, reposition objects as needed, and then resume testing. That is particularly useful for challenging scenarios, such as placing the robot in confined spaces or other complex environments that test its ability to navigate and perform safely.

- *Disturbance Generator* handles disturbances simulating possible real-world conditions. It is a software agent within the robotic system controlled from the test environment. The generator replicates various disturbances based on our observations to rigorously test the platform's safety under adverse circumstances. The disturbances include inconsistencies in localization data, traction loss, data jumps, latency issues, component failures, and even operating system crashes.
- *Simulation Pace Synchronization*: Many robotic simulation platforms, such as Gazebo (Koenig and Howard, 2004), operate with and report a simulation-time/real-time ratio that can vary. Ideally, this ratio is 1, but it may decrease based on hardware availability and the complexity of the simulated environment and should be synchronized within the testing setup. The Test Runner incorporates this simulation speed information, adjusting the pace of the simulated user accordingly to ensure synchronization with the robotic system's operations. This adjustment is critical for maintaining accurate and reliable testing results, particularly in complex simulations where timing discrepancies could otherwise lead to misleading conclusions.

The design of the testing framework with a wide parameterization range ensures that the robotic system can be rigorously tested in a wide range of scenarios. The high level of customization and flexibility allows the tests to closely mimic real-world user behaviors and interactions, enhancing the reliability and applicability of the testing outcomes.

5 Evaluation

In this section, we perform the evaluation of the proposed safety measures. For that, we implemented the test cases that cover the majority of the critical risks. They can also serve as practical examples demonstrating how such a system can be systematically tested and assessed using the proposed methods and tools.

The first set of tests addresses the effectiveness of our proposed strategies in contrast to their absence. The second set tests the system's robustness to signal disturbances that might reflect malfunctions or external interference on the most critical localization system. Due to a high number of tests, we tested our system in a fully simulated setup.

5.1 Technical setup

Our testing environment replicates the lab space. The robotic platform was simulated using a detailed 3D model of the robot

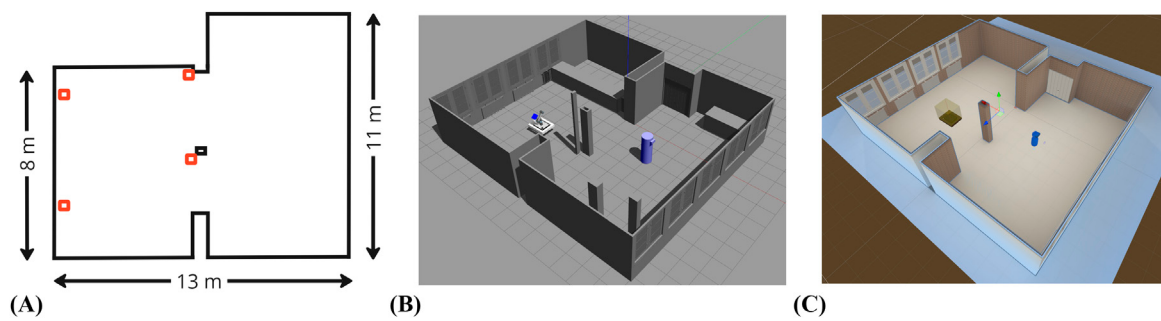


FIGURE 5

The simulation workspace for testing based on the accessible real space. (A) The layout of the accessible physical workspace. (B) The simulation environment of the robotic system. (C) The testing framework environment.

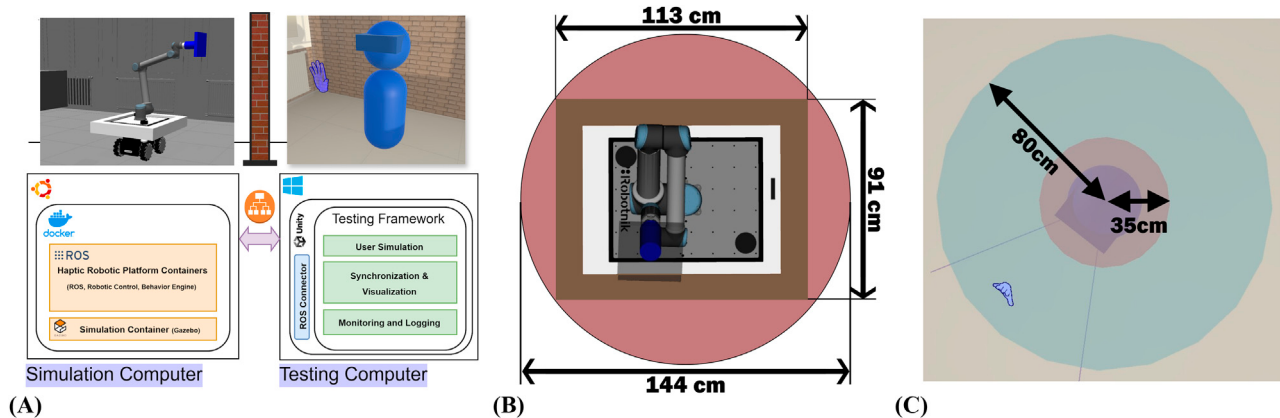


FIGURE 6

Test environment architecture and key parameterization: (A) Deployment schema of the testing and simulation environments. (B) The footprint and the minimum in-place turning space of the simulated robot. (C) The footprint of the simulated user, with the red circle depicting the user's collision space, and the blue circle indicating the Inform activating space.

provided by the manufacturer. The robot simulation employed the Gazebo engine and user simulation and test execution were done in Unity on a separate PC, as shown in Figure 5.

The testing runs across two PCs connected via LAN for synchronized operation. The robot simulation computer (Ubuntu 22.04, Intel Core i9 14700KF, 128 GB RAM, Nvidia RTX 4090) manages the robotic simulation in Gazebo, maintaining a real-time factor of 1.0 for most of the test execution. The robotic system runs ROS Melodic in a Docker environment with GPU pass-through for optimized performance. The testing computer (Windows 11, Intel Core i7 9900K, 32 GB RAM, Nvidia RTX 2080Ti) runs the Unity-based testing framework. Figure 6A illustrates the system deployment.

Robotnik RB-KAIROS with a UR10 robotic arm and Mecanum wheels, supporting omnidirectional driving. The robot's footprint forms a 113 cm by 91 cm rectangle, as shown in Figure 6B. A minimal enclosing circle with a 77 cm radius around the robot defines the minimum space required for in-place rotation. The safety strategies were implemented in FlexBe Engine (Schillinger et al., 2016) as behaviors running on the onboard computer.

5.2 Evaluation's system settings

To systematically evaluate the system's safety performance, we defined standardized test conditions, including user behavior, robot behaviors, initial positions, and evaluation metrics.

Initial Positions: For consistency, the robot and user always started from predefined positions with a 9-m separation. This ensured the robot had sufficient time to initialize and select appropriate behaviors and start following the user.

User Behavior: For consistency in user behavior across all test cases, we employed a predefined set of trajectories leading through a sequence of 100 interaction points randomly distributed across the simulated workspace. The interaction points were generated randomly within the test environment and filtered to ensure that their orientation allowed sufficient space for the user and robot approach in at least 90% of the cases. The remaining 10% included positions where successful delivery within interaction time was not due to physical constraints, such as narrow spaces near walls or corners. While the primary evaluation objective is serving haptic interaction and performance, it is also important to address the failure cases.

At each point, the user first remains stationary for 12 s, simulating the visual exploration of an object. Then, the interaction is initiated and lasts for 20 s, during which the robot positions itself and delivers the haptic feedback. After the interaction, the user waits 5 s simulating decision-making before moving to the next point. The user moves between points at a translational speed of approximately 1.2 m/s and an angular speed of $\frac{2\pi}{3}$ rad/s in a human-like walking pattern.

Chase Settings: In all of our tests, the robot had the Chase behavior active to ensure the desired distance between the robot and the user without active interaction. The value of the environment-dependent Chase parameters d_{min} and d_{max} (see Equation 1; Figure 3A) were set to 2 m and 5 m respectively. This implies that the preferred distance between the robot and the user in this behavior was 3.5 m in our testing environment.

Escape Settings: To trigger Escape, we set a minimum distance threshold of 2.1 m between the user's and the robot's positions when the robot is in front of the user within the frontal missing arc shown in Figure 4A, or 1.7 m at any other angle. Escape is disabled for the interaction duration, but was initiated right after the interaction finished. This ensures a clear user's direct walking path if the object is removed after the interaction.

Inform Settings: The Inform alert about a potential hazard is handled in Unity. It is activated from FlexBe when the Escape behavior fails, or when any part of the robot enters a 0.80 m space around the user (see the blue circle in Figure 6C). To emulate the reaction time to the Inform alert, based on our observations from the physical system, the test user was programmed to do a full stop within 1 s and stay still while the Inform behavior is active. The user could then continue her trajectory once the safety was restored.

Test Data Collection: During each test run, we collected the robot's localization data, traveled distance, user-robot and robot-environment collisions, and a number of triggered Inform and Escape behaviors, successful and not haptic interactions, and various timings. We also recorded the duration when the distance between the robot and the user was less than 1 m. This 1 m threshold is critical, especially when the robot is positioned orthogonally to the user, as its bumper corners may pose a collision risk (see Figures 6B,C).

To measure the system's haptic performance, the number of successful haptic prop serves and the haptic response time were recorded, indicating how the system responded to haptic requests in each of the test variations. These parameters and test acceptance criteria (defined in Section 4.7) provided an informative evaluation of both the safety and performance of the system.

5.3 Safety strategies impact tests

These tests examine how Escape and Inform behaviors, implementing corresponding strategies, enhance user safety. First, we evaluate the system without them and then progressively introduce the behaviors to analyze their individual and combined effects in the following conditions:

1. Without safety behaviors ("No Safety"): The system operates only with the Chase behavior, maintaining user proximity without Chase or Inform. This condition serves as a baseline.
2. Inform's impact ("Inform Only"): The Inform behavior is added, allowing the system to warn the user of potential proximity hazards. This test assesses the Inform effectiveness in preventing collisions and its influence on system performance.
3. Escape's impact ("Escape Only"): The Escape behavior is enabled without Inform. This test evaluates Escape's impact in maintaining safe distances and its limitations without user warnings.
4. With Safety Behaviors ("Inform and Escape"): Both safety behaviors are active for evaluation of their combined effectiveness for safety.

5.3.1 Results

Table 4 presents the results of the conducted tests. The test durations varied, with the "No Safety" case taking 96.03 min "Escape Only" reduced it by 4% (91.99 min) while "Inform Only" increased runtime by 15% (110.46 min). The "Inform and Escape" condition had a runtime of 97.09 min, closely matching the "No Safety" case, indicating that both behaviors have minimum impact on task performance time.

The robot's traveled distance was 1288.17 m in the "No Safety". With "Inform Only", it increased by 21% (1564.41 m), and with "Escape Only", it reached the highest value at 1910 m (+48%). With "Inform and Escape", the robot traveled 1822.40 m, falling between the "Inform Only" and "Escape Only" conditions but still noticeably higher than the "No Safety". Notably, the traveled distance did not directly correlate with test duration.

The bar chart in Figure 7A depicts the number of collisions in the tests. No user-robot collisions occurred in tests where Inform was active ("Inform Only" and "Inform and Escape"). In contrast, "Escape Only" resulted in eight collisions, while "No Safety" recorded 70 collisions. No collisions between the robot and the environment were observed in any of the tests.

Critical proximity duration (time spent within 1 m of the user) was significantly reduced by safety behaviors. In the "No Safety" condition, it was 119.76 s "Inform Only" reduced it to 1.81 s, while "Escape Only" lowered it further to 0.40 s. With both behaviors active "Inform and Escape", no critical proximity events were recorded. The distribution is shown in Figure 7B.

Throughout the test runs, the logged average distance between the robot and the user was 2.61 m (SD 1.08 m) in "No Safety", and 2.56 m (SD 1.05 m) in "Inform Only". This is below our preferred distance of 3 m between the robot and the user. The distance increased to an average of 2.98 m (SD 1.00 m) in "Escape Only" and reached a mean of 3.10 m (SD 1.14 m) in the "Inform and Escape" condition. Figure 8B illustrates the distribution of the user-robot distance.

The number of successfully served haptic interactions and the average haptic response time (time between initiation and actual haptic interaction) remained consistent across all four conditions. In the "Inform and Escape" condition, a maximum of 96 out of 100 haptic requests were served successfully, while a minimum of 92 out of 100, occurred in the "Inform Only" condition. The haptic response time showed slight variations, with the shortest mean response time of 7.06 s (SD 0.85 s) recorded in the "Inform Only" condition, and the longest mean response time of 7.44 s

TABLE 4 Results of the evaluation of the safety strategies impact test cases.

	Safety strategies impact test cases			
	No safety	Inform only	Escape only	Inform and escape
Time to complete the test in min	96.03	110.46	91.99	97.09
Robot's total traveled distance in m	1288.17	1564.41	1910	1822.40
* Number of robot-user collisions	70	0	8	0
* Number of robot-environment collisions	0	0	0	0
* Total duration within critical distance in s	119.76	0.40	1.81	0
Robot-user distance in m, Mean (SD)	2.61 (1.08)	2.56 (1.05)	2.98 (1.00)	3.10 (1.14)
Number of successful haptic serves	95	92	93	96
Haptic response time in s, Mean (SD)	7.21 (1.17)	7.06 (0.85)	7.30 (1.21)	7.44 (1.07)
Traveled distance per request in m, Mean (SD)	2.36 (0.01)	2.37 (1.06)	2.77 (0.01)	2.74 (1.86)
Number of Inform behavior activations	0	128	0	44
Duration in Inform in s, Mean (SD)	0	4.21 (0.05)	0	0.41 (0.30)
Total duration in Inform in s	0	539.67	0	17.5
Number of Escape behavior activations	0	0	267	262

The parameters with * are the primary safety-related parameters. The parameters reported with Mean and SD, correspond to interactions and those without them to the whole test case.

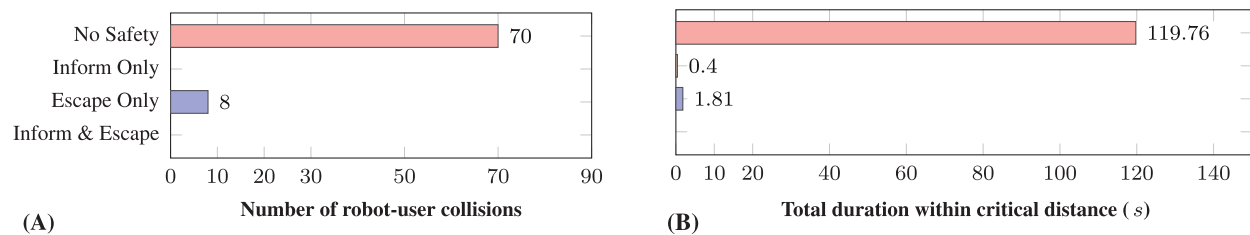


FIGURE 7 Safety-related parameters of the test case runs. (A) The number of collisions between the robot and the user. (B) The total duration of critical proximity ($<1m$ distance) between the user and the robot.

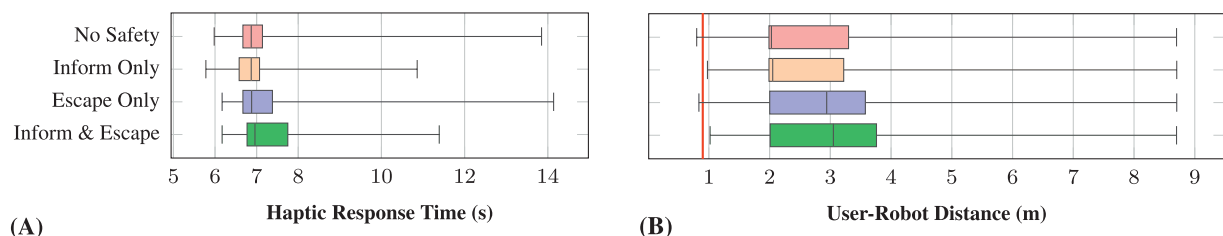


FIGURE 8 Safety Strategies Impact Tests Results. (A) Haptic response time, and (B) the distance between the robot and the user compared to the baseline. The red line in (B) indicates the collision distance between the user and the robot.

(SD 1.07 s) in “Inform and Escape” condition. The distribution of haptic response times across the test cases is shown in Figure 8A.

The Inform behavior activated 128 in “Inform Only”, compared to 44 in “Inform and Escape”, a 66% reduction. Additionally, the total duration of Inform signals was significantly higher in “Inform

Only” (539.67 s) compared to “Inform and Escape” (17.5 s), showing that the Escape behavior reduces reliance on user warnings. The Escape behavior was activated 267 times in “Escape Only” and 262 times in “Inform and Escape”, showing consistent activation frequency across these cases.

5.3.2 Discussion

In the “No Safety” condition, the system experienced 70 collisions between the user and the robot over a 96 min operation time. Introducing Inform behavior alone eliminated all user-robot collisions and reduced the duration of critical proximity from nearly 2 min to less than a second. However, this came at the cost of activating Inform 128 times, with each activation averaging 4.21 s. The system required the user to fully stop and detach from the VE for a total of 8.5 min with one or more Informs per interaction. In reality, frequent warnings have a notable impact on user presence and overall system performance perception, due to many interruptions reminding about the physical world.

In contrast, “Escape Only” had less impact on VR presence, as no warnings were issued. This setup allowed the user to operate uninterrupted. However, while Escape drastically reduced the number of user-robot collisions considerably (from 70 to 8) and the total critical proximity duration (from 2 min to <2 s), it did not fully prevent collisions. These improvements came with a 48% increase in the robot’s traveled distance compared to the “No Safety” condition, as the robot had to move significantly more to avoid collisions. With “Inform Only”, the robot traveled 21% more compared to the “No Safety” condition, reflecting the increased operational cost for enhanced safety.

The “Inform and Escape” condition provided the best safety-performance balance. It eliminated collisions and critical proximity events while maintaining a completion time similar to the “No Safety” condition. Inform was triggered two-thirds less than in the “Inform Only” case, due to Escape proactively preventing warnings. Inform activations averaged under half a second, reducing total activation time to 17.5 s compared to 539.67 s in the “Inform Only” condition. However, Escape was activated 262 times, increasing the robot’s traveled distance by 41% compared to “No Safety” and 16% compared to “Inform Only”.

Despite increased robot travel in all safety conditions, haptic response time remained consistent. In the “Inform and Escape” case, response time increased by only 3%, confirming that extra movement primarily occurred during idle periods. Notably, this condition also achieved the highest number of successful haptic serves (96 out of 100), suggesting that improved positioning and repositioning enhanced interaction performance.

5.4 Disturbance resilience tests

Testing under environmental disturbances is important for validating system robustness in real-world conditions, where sensor noise and localization errors are common. For that, the haptic system in the “Escape and Inform” condition was tested with the noise as disturbance and followed the methodology described in [section 4.7](#). The noise was applied to:

- *User localization data*: Evaluates the system’s response to distorted user tracking data, represented by sudden jumps in the user’s position and orientation away from the predefined trajectory and haptic interaction position.
- *Robot localization data*: Assesses system robustness against distorted robot tracking data, presented as position and orientation jumps similar to user localization noise.

The types of applied noise are detailed in [Table 5](#), selected based on our experience with the physical setup. The noise was randomized within defined ranges and independently injected into the localization data streams and did not interfere with the subsystems’ internal workflow.

5.4.1 Results

In this group of tests, the system in the “Escape and Inform” condition without noise is the comparison baseline. The [Supplementary Appendix 2](#) provides a full summary of the test results. The system successfully served the haptic prop in 94–97 out of 100 interaction points, maintaining service performance despite localization noise. The measured test durations did not deviate much from the baseline (97.09 min) staying in the 97.1 s–101.55 s range across the noise conditions. The primary factor here was user behavior, which remained consistent across all tests, and noise with safety behaviors had minimum impact on the completion time.

Traveled distance per interaction also remained consistent (mean 2.74–3.13 m, SD 1.55–2.16 m). The true robot’s traveled distance remained similar to the baseline (mean 1786.91 m, SD 49.53 m). However, in conditions with translational noise injection to the robot’s data stream, the testing framework reported traveled distances significantly and abnormally higher than the baseline. We obtained the actual traveled distances from ROS logs and used them to cross-check for the framework’s report (see [Supplementary Appendix 2](#)). This discrepancy is an artifact of the black-box testing framework and noise implementation, which calculated the traveled distance based on noisy localization data. The confusion of the robot’s localization itself should be tested as well but would require a different implementation and hybrid testing.

Similarly, no true collisions occurred in any of these test case variations, although the test logs indicated collisions between the user and the robot in some variations. Further inspection of the scene captures confirmed that these were artifacts caused by the noisy data. Similarly, brief instances of critical proximity (1 m or less) in cases with translational noise were verified as artifacts and not genuine events (see [Supplementary Appendix 2](#)).

Escape activations were generally consistent with the baseline, as shown in [Figure 9B](#), with fluctuations below 10%. Inform activations varied, increasing when noise affected the robot’s localization, particularly in the wide translational noise case. However, it decreased when noise was applied to the user, where the wide translational noise case recorded half as many activations as the baseline (see [Figure 9A](#)). Interestingly, when noise was applied to the user, the average duration and total active time of Inform were consistently higher than the baseline, except for the total duration in the orientation-wide noise case. In contrast, when noise was applied to the robot, Inform durations generally aligned with or fell below the baseline ([Figure 10](#)).

5.4.2 Discussion

This test category further demonstrated the system’s ability to maintain user safety under varying conditions, even in the presence of localization noise. The robot did not collide with the environment in the tests, even when subjected to translational and orientational localization noise. This resilience stems from the robot-centric

TABLE 5 Noise types and magnitudes for localization disturbance test variations and the explanation of their possible source and impact on the system.

Noise type	Magnitude	Explanation
Translational	[−0.2 m, 0.2 m]	Represents the most common noise type. These errors typically arise from inaccuracies in localization systems calibration
Translational wide	[−1 m, 1 m]	Wide localization errors are less frequent and occur due to factors like hardware unplanned relocation (collisions) or software failures
Orientalional	[−30°, +30°]	Minor directional inaccuracies in the complex localization typically occur after sudden changes in the direction, requiring a short amount of time for the system to readjust
Orientalional wide	[−120°, +120°]	Major directional inaccuracies in the localization system are caused by failing sensors or issues in sensor data fusion, potentially leading to unpredictable behaviors of both the user and the robot
Translational and Orientalional wide (Combined wide)	[−1 m, 1 m] and [−120°, +120°]	Combined noise typically arise from severe sensors or data fusion problems, as well as network and connection issues mostly experienced in the robot

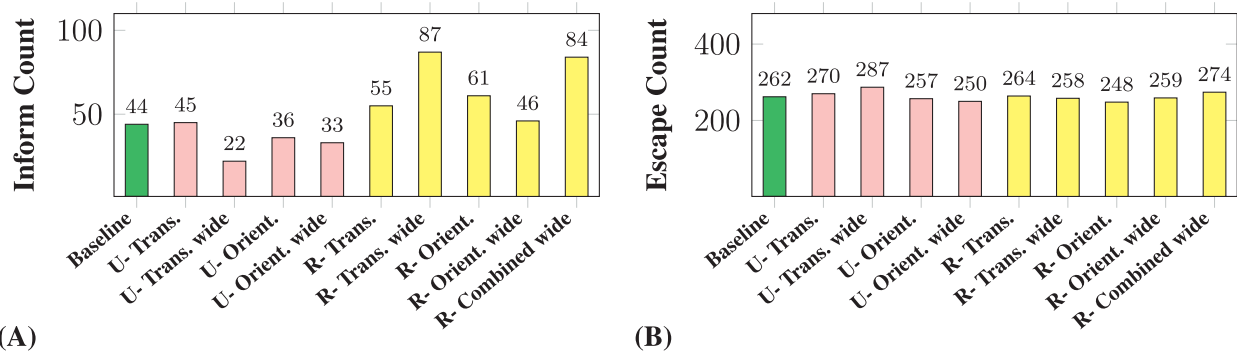


FIGURE 9

The number of Inform and Escape behaviors activations in each test condition. In these charts, green color is used for the baseline, pink for the user (U), and yellow for the robot (R). **(A)** The number of Inform activations. **(B)** The number of Escape activations.

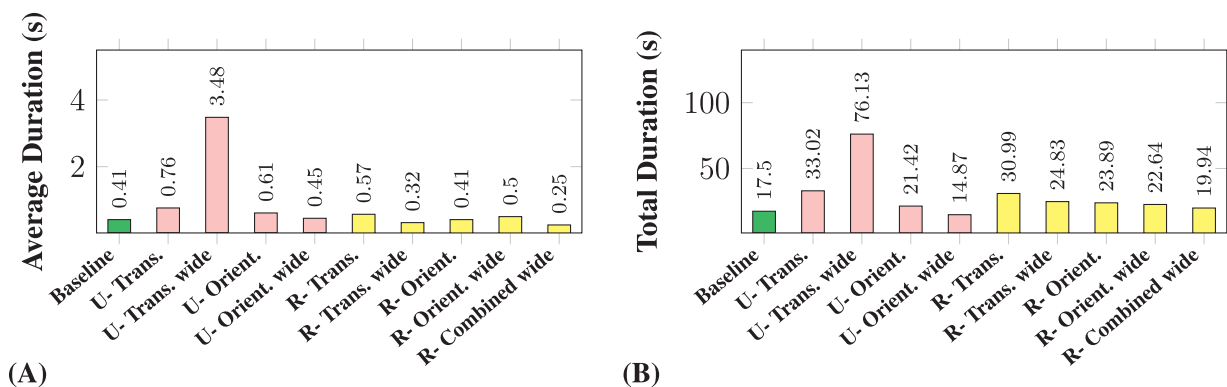


FIGURE 10

Inform activations in each test condition. In these charts, green color is used for the baseline, pink for the user (U), and yellow for the robot (R). **(A)** The average time Inform behavior stayed active per condition. **(B)** The total duration of active Inform in test conditions.

LiDAR-based localization and obstacle detection, which does not rely on global localization data. While localization noise can disrupt navigation, it does not impair the robot's ability to detect and avoid obstacles during movement. Indeed, the collisions are typically a result of fast user movements toward the robot rather than the robot's navigation errors.

An interesting anomaly was observed when the user experienced wide translational noise. In this case, the number of Inform activations dropped to exactly half of the baseline, yet the mean and total durations of activations were significantly higher than those of other test cases in this category. Additionally, this test condition recorded the highest number of Escape activations among

all scenarios. This behavior can be attributed to the wide translational noise (up to 1 m), which caused Inform to remain active for extended periods and prompted the system to enter Escape behavior more frequently. Despite these unusual patterns, none of the tests, including the wide translational noise case, resulted in any collisions between the user and the robot, maintaining user safety throughout.

Such unique cases underscore the utility of the proposed testing framework, which allows developers to identify, investigate, and elaborate on outlier scenarios by defining closer and more specific test cases. This capability helps explore a broader range of situations and system responses to identify unforeseen hazards that could compromise safety. Another use case of the testing outcome is the collection of the training data for the overseer, who needs to interpret the observed situation correctly.

6 Discussion

While safety concerns have been partially addressed in several previous works on custom ETHD systems, e.g., (Vonach et al., 2017; Mercado et al., 2021), the discussion remains open, especially for the use of mobile robots. Based on our experience, we proposed an approach to integrate a constellation of safety measures into the design of a mETHD system. As a groundwork, a thorough initial risk analysis reveals potential hazards, which have to be addressed with suitable safety measures. The risk analysis has to be revisited repeatedly until all risks are at least at the ALARP level (Jones-Lee and Aven, 2011; Langdalen et al., 2020).

In a mETHD system, the robot and user are closer, often in direct contact, than the safety standards suggest, and the user cannot see the robot. So developing such a system, the possibility of failure of every safety measure should be assumed. Fundamentally, multiple hardware safeties should be present as an indispensable backup solution to cover the worst-case scenario. A cobot is usually already equipped with a number of mitigating mechanisms on a hardware level, like joint torque sensors or an emergency stop. However, some of them can also create a hazard if triggered untimely. Thus, we specify the inclusion of a human overseer with a standard remote emergency stop, crucial to be able to immediately halt the robot from a distance when necessary. Additionally, we propose placing contact sensors on the mobile platform, creating a three-way emergency stop mechanism.

One critical aspect of hardware-level safety is ensuring that the robot's acceleration and velocity are sufficient for rapid pre-emptive collision avoidance maneuvers. Another essential feature is the robot's capability of omnidirectional mobility, enabling it to react and reposition efficiently without requiring in-place rotation before movement. These capabilities were key factors in selecting the robot used in CoboDeck, as it can navigate omnidirectionally and maintain speeds comparable to a typical human walking in VR.

We analyzed which other parts of the robot might be dangerous for the user and could be addressed with physical means. Inspired by the level of safety of soft robots (Raatz et al., 2015), we proposed to employ soft materials in areas where user contact is possible, even if improbable. For example, in CoboDeck, we equipped the robot with a suspended wooden frame with an extra foam layer, as well as foam matting around the robotic arm, as shown in Figure 2. We also

considered what and how the robot can sense with redundant sensors, and strongly recommend enhancing the localization system's reliability and confidence.

Nevertheless, the best safety is to prevent the worst case in principle. In this paper, we proposed a set of behavior strategies for mETHDs: "Chase", "Escape", and "Inform". These general strategies systematically address the performance and safety of such systems and could be customized for the specifics of the individual robots. Software-level safety measures involve real-time system monitoring, dynamic user response, and dedicated safety robot behaviors. For instance, we proposed the Chase behavior for repositioning during the robot's idle state. The robot actively moves to areas that are less probable to be explored by the user from where they currently stand (see Section 4.3; Figure 3). This approach could be extended to other systems like those in Bouzbib et al. (2020) to be considered in path-planning strategies to move components to safer positions when not in use for interactions.

In scenarios when users take unexpected paths, the robotic system must respond promptly with controlled behaviors, such as the Escape strategy we proposed, explained in Section 4.4 and depicted in Figure 4. For robots lacking omnidirectional movement, Escape behavior design must ensure a fast response without requiring an in-place rotation, as delays here could compromise safety. Since Escape behavior is dependent on both the robot's mobility and the environment's geometry, it should be tailored to each specific system. Additionally, variations in VR applications may lead to different user behaviors, necessitating further customization of the robot's movements. In the instance of Zoomwalls (Yixian et al., 2020), it would be recommendable to consider the response of the system if the VR user acts in an unpredictable way, such as walking backward.

When everything fails and the robot cannot prevent a collision autonomously, fallback strategies are crucial. These include providing the user with visual or auditory alerts, such as the Inform behavior we proposed in Section 4.5. Inform helps maintain user awareness during unexpected situations by temporarily interrupting the VR immersion to ensure safe interaction with the robotic system.

We suggest implementing the strategies as a state machine with its modular approach without tight coupling. It allows modification of the strategies for the specifics of each robot. The proposed strategies are also applicable for non-omnidirectional grounded robots, but they should consider the specific robot orientation and, for bulky robots, as in Suzuki et al. (2020), the air resistance. For mETHDs based on vacuum cleaners (Wang et al., 2020), where stepping onto the robot is a hazard, the robot needs to be either covered by a proxy object (physical solution) or the distances to the user should be increased. For ceiling-mounted mobile props (Bouzbib et al., 2020), the chasing position should be far enough from the user to account for a possible fall during movement and stop for the interaction, to allow the user to close in from a safe distance.

Consequently, our safety strategies generalize to various mobile grounded, suspended, or flying robots, except body-grounded ones, as our use of 2D distances and positioning is not applicable to robots attached to the user. Yet, the safety risk analysis, physical means, and redundant sensors have to be customized for each individual case. Nevertheless, once defined, some solutions might be reused on a

similarity basis. Likewise, the testing should also consider specifics in the requirements elicited during risk analysis, the use-case specifics, and hardware limitations.

Implementation of a close-to-reality simulation system with an integrated easy-to-adapt testing framework makes evaluation of the mETHD safety fast, repeatable, low-cost, and reliable. Such a framework allows testing various scenarios involving sensor noise, delays, and other unwanted factors that can be challenging to replicate in a physical system. The evaluation section demonstrated how this testing approach can identify the system's vulnerabilities prior to deployment, ensuring that the employed safety measures are validated under realistic and potentially adverse conditions. Thorough pre-deployment testing ensures that mETHD's software controls are reliable, safe, and ready for real-world use.

However, for the final deployment of the haptic VR system, the testing cannot be limited to virtual tests only. It should be followed by semi-virtual tests with a real robot and a simulated user. This step will help to identify the physical details that could not be simulated, such as floor-to-wheel traction, wireless connectivity, actual battery life, and reflective surfaces or bad lighting in the working volume that might affect the system. Next, the real user motion replay can address the behavioral specifics, such as unexpected actions or speed alteration, without endangering them. In our ongoing research, we are using a small four-wheeled omnidirectional robot with a make-shift mannequin on top to test the user detection mechanisms and collision reaction. It moves according to a pre-recorded real user trajectory and can trigger a haptic request. After this step, the transition should be made to live testing with real users to confirm the resilience under final working conditions.

7 Limitations

This paper has several limitations that were partially or not addressed. One limitation is that we did not cover the use of the robotic arm and its user-aware dynamic (re)positioning when rendering large virtual objects. The interactions with stationary or body-anchored robotic arms have been covered in the existing literature (Mercado et al., 2021; Vonach et al., 2017; Horie et al., 2021). Therefore, this work focused on the mobile platform that can support such an arm and greatly increase the interaction range.

Our risk analysis and parameterization are mostly based on our existing Cobodeck system. Therefore, it generalizes only to a certain extent and should be reevaluated for other systems. Environment geometry, hardware specifics, and the user's tasks within a system also might have a significant impact on the risk assessment and the full set of safety measures and parameterization. Taking this into account, we placed more emphasis on developing strategies than on specific guidelines. Strategies themselves have to be adapted to the specifics of the work environment and robots' characteristics. For instance, Escape in narrow spaces will likely work for a vacuum cleaner but might need adaptations for a fast and heavy platform like RB Kairós.

Another limitation lies in the black-box testing approach. While effective for many testing scenarios, this approach is limited to external observation and injected interferences that sometimes pose an issue for interpretation. For instance, when localization noise was introduced, some observed results required cross-checks with the

internal ROS logging system. While false positives are generally more acceptable than false negatives in safety-critical applications, in some cases, a grey-box testing approach might be necessary.

Discrepancies between the simulated and the real-world robot behavior are also a limitation. For instance, the omnidirectional wheels in simulation tools like Gazebo cannot be accurately modeled due to complex interactions with the floor. As a result, simplified models are used, which can cause notable differences in behavior compared to the real world. We observed that the robot's real starting and stopping behavior slightly differed from the simulation, which did not account for occasional slipping. This underlines that simulation tests cannot fully replace real-world tests and such discrepancies should be observed for specific environments and robots to be reflected in parameterization accordingly.

Similarly, computational load and power differences between the simulation and the physical robot might occur. While the simulation runs on a dedicated, powerful testing PC, the real robot must manage processing and hardware control with weaker computers due to limited power, which can lead to performance variations. These differences are individual to each system and should be verified and parametrized accordingly to minimize the gaps between simulation and reality. We partially addressed this issue with synchronization of the testing framework with the simulation. However, minor adjustments might be necessary for the real robot.

8 Conclusion and future work

Mobile robots can be utilized in diverse applications, including VR for fully or semi-autonomously delivering haptic feedback as mETHDs, aiming to provide seamless user experiences. However, existing robotic solutions and standards are not made to share a workspace with a blindfolded VR user, who touches virtual objects, while unknowingly interacting with a real, solid robot. The rapid speed of development in robotics and VR systems has outpaced the evolution of safety standards. Moreover, the conventional definition of collaborative robots and systems falls short of addressing the unique requirements of these environments. This paper focused on the safety challenges of ETHDs with mobile platforms and the methodology to resolve them.

Starting with research and industry standards, we review the existing safety requirements and regulations. On the example of our immersive autonomous mETHD system CoboDeck, we demonstrated how a system should undergo risk analysis and discussed the approaches for risks mitigation. Beyond what current standards suggest, we identified unique conditions that must be considered for haptic VR. We discuss the necessary fallback plans and redundant subsystems that improve the system's reliability. Moreover, we proposed the distance maintenance and haptic response optimization strategy "Chase" and safety-specific strategies "Escape" and "Inform". We also presented and discussed the testing approach for the mETHD systems with a testing framework and proposed a test case definition template. Based on it, we defined test cases that cover the most critical risks and evaluated the safety impact of the proposed strategies. This way we provided an example of the pre-deployment testing and discussed its contributions and

limitations to the improvement of the mETHD systems. Our approach to the safety assessment with the simulated pre-deployment testing enables safer, faster, and more systematic verification before tests with real users.

In our future work, we plan to further address a number of safety aspects for mETHD. The safety of combined and parallel movements of the mobile robot and the robotic arm during interactions with large virtual objects (walls, buildings) requires further research. We also consider machine learning for the prediction of user's actions. Finally, our ultimate goal is to generalize the safety guidelines for mETHDs, particularly concerning movement and rotational maneuvers in close proximity to users.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

SM: Writing – original draft, Writing – review and editing. MG: Writing – original draft, Writing – review and editing. KV: Writing – original draft, Writing – review and editing. EV: Writing – original draft, Writing – review and editing. HK: Funding acquisition, Project administration, Supervision, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This research was funded in whole or in part by the Austrian Science Fund (FWF) [10.55776/F77]. For open access purposes, the authors have applied a CC BY public copyright license to any accepted manuscript version arising

from this submission. Open access funding provided by TU Wien University of Technology.

Conflict of interest

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.

Generative AI statement

The author(s) declare that Generative AI was used in the creation of this manuscript. The text is fully human-written and has been checked with Grammarly AI tools for correctness.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frvir.2025.1648019/full#supplementary-material>

References

- Abdullah, M., Kim, M., Hassan, W., Kuroda, Y., and Jeon, S. (2018). "Hapticdrone: an encountered-type kinesthetic haptic interface with controllable force feedback: example of stiffness and weight rendering," in *2018 IEEE haptics symposium (HAPTICS)*, 334–339. doi:10.1109/HAPTICS.2018.8357197
- Abtahi, P., Landry, B., Yang, J. J., Pavone, M., Follmer, S., and Landay, J. A. (2019). "enBeyond the force: using quadcopters to appropriate objects and the environment for haptics in virtual reality," in *Proceedings of the 2019 CHI conference on human factors in computing systems* (Glasgow Scotland Uk: ACM), 1–13. doi:10.1145/3290605.3300589
- Akalin, N., Kristofferson, A., and Loutfi, A. (2022). Do you feel safe with your robot? Factors influencing perceived safety in human-robot interaction based on subjective and objective measures. *Int. J. Human-Computer Stud.* 158, 102744. doi:10.1016/j.ijhcs.2021.102744
- Alami, R., Albu-Schaeffer, A., Bicchi, A., Bischoff, R., Chatila, R., De Luca, A., et al. (2006). "Safe and dependable physical human-robot interaction in anthropic domains: state of the art and challenges," in *2006 IEEE/RSJ international conference on intelligent robots and systems*, 1–16. doi:10.1109/IROS.2006.6936985
- Araujo, B., Jota, R., Perumal, V., Yao, J. X., Singh, K., and Wigdor, D. (2016). "Snake charmer: physically enabling virtual objects," in *Proceedings of the TEI '16: tenth international conference on tangible, embedded, and embodied interaction* (New York, NY, USA: Association for Computing Machinery), 218–226. doi:10.1145/2839462.2839484
- Askarpour, M., Mandrioli, D., Rossi, M., and Vicentini, F. (2016). "SAFER-HRC: Safety analysis through formal vERification in human-robot collaboration," in *Computer safety, reliability, and security*. Editors A. Skavhaug, J. Guiochet, and F. Bitsch (Cham: Springer International Publishing), 283–295. doi:10.1007/978-3-319-45477-1_22
- Baerveldt, A.-J. (1992). "Cooperation between man and robot: interface and safety," in *[1992] proceedings IEEE international workshop on robot and human communication (IEEE)*, 183–187.
- Beizer, B., and Wiley, J. (1996). Black box testing: techniques for functional testing of software and systems. *IEEE Softw.* 13, 98. doi:10.1109/MS.1996.536464
- Beluško, M., Hegedűs, M., and Fedorko, G. (2016). Creating visual work instructions to ensure safe and fluent operation of the Semi-automatic Production lines. *Open Eng.* 6. doi:10.1515/eng-2016-0097
- Bouzbib, E., and Bailly, G. (2022). "Let's meet and work it out: understanding and mitigating encountered-type of haptic devices failure modes in VR," in *2022 IEEE conference on virtual reality and 3D user interfaces (VR)*, 360–369. doi:10.1109/VR51125.2022.00055
- Bouzbib, E., Bailly, G., Haliyo, S., and Frey, P. (2020). Covr: a large-scale force-feedback robotic interface for non-deterministic scenarios in vr. *arXiv Prepr. arXiv:2009.07149*, 209–222. doi:10.1145/3379337.3415891
- Casals, A., Villà, R., and Casals, D. (1993). "A soft assistant arm for tetraplegics," in *Rehabilitation technology* (Amsterdam, Netherlands: IOS Press), 103–107.
- Chemweno, P., Pintelon, L., and Decre, W. (2020). Orienting safety assurance with outcomes of hazard analysis and risk assessment: a review of the ISO 15066 standard for collaborative robot systems. *Saf. Sci.* 129, 104832. doi:10.1016/j.ssci.2020.104832

- Dai, S., Smiley, J., Dwyer, T., Ens, B., and Besancon, L. (2022). Robohapalytics: a robot assisted haptic controller for immersive analytics. *IEEE Trans. Vis. Comput. Graph.* 29, 451–461. doi:10.1109/tvcg.2022.3209433
- De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A. (2008). An atlas of physical human-robot interaction. *Mech. Mach. Theory* 43, 253–270. doi:10.1016/j.mechmachtheory.2007.03.003
- de Tinguy, X., Howard, T., Pacchierotti, C., Marchal, M., and Lécuyer, A. (2020). “WeATaViX: wearable actuated TAngibles for VIRTual reality eXperiences,” in *Haptics: science, technology, applications: 12th international conference, EuroHaptics 2020, Leiden, the Netherlands* (Berlin, Heidelberg: Springer-Verlag), 262–270. doi:10.1007/978-3-030-58147-3_29
- Engelberger, J. F. (1985). *Robotics in practice: management and applications of robotics in industry*. American Management Assoc., Inc.
- Fox, D., Burgard, W., Dellaert, F., and Thrun, S. (1999). Monte carlo localization: efficient position estimation for mobile robots. *AAAI/IAAI 1999*, 343–349. doi:10.1109/ROBOT.1999.772544
- Garcia, V. R. M. (2021). *Contribution to the study of usability and haptic feedback of encountered-type haptic displays*. Ph.D. thesis (Rennes, France: INSA de Rennes).
- Giuliani, M., Lenz, C., Müller, T., Rickert, M., and Knoll, A. (2010). Design principles for safety in human-robot interaction. *Int. J. Soc. Robotics* 2, 253–274. doi:10.1007/s12369-010-0052-0
- Gomi, R., Suzuki, R., Takashima, K., Fujita, K., and Kitamura, Y. (2024). “Inflatable shape-changing mobile robots for large-scale encountered-type haptics in vr,” in *Proceedings of the 2024 CHI conference on human factors in computing systems*, 1–14.
- Guda, V. K., Chablat, D., and Chevallereau, C. (2020). “enSafety in a human robot interactive: application to haptic perception,” in *Virtual, augmented and mixed reality. Design and interaction*. Editors J. Y. C. Chen and G. Fragomeni (Cham: Springer International Publishing), 562–574. doi:10.1007/978-3-030-49695-1_38
- Haddadin, S., Albu-Schaffer, A., De Luca, A., and Hirzinger, G. (2008). “Collision detection and reaction: a contribution to safe physical human-robot interaction,” in *2008 IEEE/RSJ international conference on intelligent robots and systems*, 3356–3363. doi:10.1109/IROS.2008.4650764
- Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. (2009). Requirements for safe robots: measurements, analysis and new insights. *Int. J. Robotics Res.* 28, 1507–1527. doi:10.1177/0278364909343970
- Hata, A., Inam, R., Raizer, K., Wang, S., and Cao, E. (2019). “Ai-based safety analysis for collaborative mobile robots,” in *2019 24th IEEE international conference on emerging technologies and factory automation (ETFA)* (IEEE), 1722–1729.
- Heinzmann, J., and Zelinsky, A. (1999). A safe-control paradigm for human-robot interaction. *J. Intelligent robotic Syst.* 25, 295–310. doi:10.1023/a:1008135313919
- Hoppe, M., Knierim, P., Kosch, T., Funk, M., Futami, L., Schneegass, S., et al. (2018). “Vrhapticdrones: providing haptics in virtual reality through quadcopters,” in *Proceedings of the 17th international conference on Mobile and ubiquitous multimedia*, 7–18.
- Horie, A., Saraiji, M. Y., Kashino, Z., and Inami, M. (2021). “EncounteredLimbs: a room-scale encountered-type haptic presentation using wearable robotic arms,” in *2021 IEEE virtual reality and 3D user interfaces (VR)*, 260–269. doi:10.1109/VR50410.2021.00048
- Hoshikawa, Y., Fujita, K., Takashima, K., Fjeld, M., and Kitamura, Y. (2024). Redirecteddoors+: Door-opening redirection with dynamic haptics in room-scale vr. *IEEE Trans. Vis. Comput. Graph.* 30, 2276–2286. doi:10.1109/tvcg.2024.3372105
- HTC and Valve (2016). Htc vive virtual reality system.
- Huck, T. P., Münch, N., Hornung, L., Ledermann, C., and Wurll, C. (2021). Risk assessment tools for industrial human-robot collaboration: novel approaches and practical needs. *Saf. Sci.* 141, 105288. doi:10.1016/j.ssci.2021.105288
- ISO (2010). *ISO 12100:2010 safety of machinery — general principles for design — risk assessment and risk reduction*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/51528.html>.
- ISO (2011a). *ISO 10218-1:2011 robots and robotic devices — safety requirements for industrial robots — part 1: robots*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/51330.html>.
- ISO (2011b). *ISO 10218-2:2011 robots and robotic devices — safety requirements for industrial robots — part 2: robot systems and integration*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/41571.html>.
- ISO (2012). *ISO 13849-2:2012 safety of machinery — safety-Related parts of control systems — part 2: validation*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/53640.html>.
- ISO (2014). *ISO 13482:2014 robots and robotic devices — safety requirements for personal care robots*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/53820.html>.
- ISO (2016). *ISO/TS 15066:2016 robots and robotic devices — collaborative robots*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/62996.html>.
- ISO (2023). *ISO 13849-1:2023 safety of machinery — safety-related parts of control systems — part 1: general principles for design*. Geneva, Switzerland: International Organization for Standardization. Available online at: <https://www.iso.org/standard/73481.html>.
- Jones-Lee, M., and Aven, T. (2011). Alarp—What does it really mean? *Reliab. Eng. and Syst. Saf.* 96, 877–882. doi:10.1016/j.res.2011.02.006
- Kanamori, K., Sakata, N., Tominaga, T., Hijikata, Y., Harada, K., and Kiyokawa, K. (2018). “Obstacle avoidance method in real space for virtual reality immersion,” in *2018 IEEE international symposium on mixed and augmented reality (ISMAR)* (IEEE), 80–89.
- Kang, H., and Han, J. (2020). Safexr: alerting walking persons to obstacles in mobile xr environments. *Vis. Comput.* 36, 2065–2077. doi:10.1007/s00371-020-01907-4
- Kästner, L., and Lambrecht, J. (2019). “Augmented-reality-based visualization of navigation data of mobile robots on the microsoft hololens-possibilities and limitations,” in *2019 IEEE international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mechatronics (RAM)* (IEEE), 344–349.
- Koenig, N., and Howard, A. (2004). “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *IEEE/RSJ international conference on intelligent robots and systems (sendai, Japan)*, 2149–2154.
- Kulić, D. (2006). “Safety for human-robot interaction,”. Ph.D. thesis (Vancouver, Canada: University of British Columbia). doi:10.14288/1.0080744
- Langdalen, H., Abrahamsen, E. B., and Selvik, J. T. (2020). On the importance of systems thinking when using the alarp principle for risk management. *Reliab. Eng. and Syst. Saf.* 204, 107222. doi:10.1016/j.res.2020.107222
- Lasota, P. A., Fong, T., and Shah, J. A. (2017). A survey of methods for safe human-robot interaction. Delft, Netherlands: *Found. Trends® Robotics* 5, 261–349. doi:10.1561/23000000052
- LaViola, J. J. (2000). A discussion of cybersickness in virtual environments. *SIGCHI Bull.* 32, 47–56. doi:10.1145/333329.333344
- Marvel, J. A. (2017). “Sensors for safe, collaborative robots in smart manufacturing,” in *2017 IEEE SENSORS*, 1–3. doi:10.1109/ICSENS.2017.8234264
- Medeiros, D., Dos Anjos, R., Pantidi, N., Huang, K., Sousa, M., Anslow, C., et al. (2021). “Promoting reality awareness in virtual reality through proxemics,” in *2021 IEEE virtual reality and 3D user interfaces (VR)* (IEEE), 21–30.
- Menolotto, M., Komaris, D.-S., O’Sullivan, P., and O’Flynn, B. (2024). “Assessing trust in collaborative robotics with different human-robot interfaces,” in *2024 IEEE international instrumentation and measurement technology conference (I2MTC)* (IEEE), 1–6.
- Mercado, V. R., Howard, T., Si-Mohammed, H., Argelaguet, F., and Lécuyer, A. (2021). “Alfred: the haptic butler On-Demand tangibles for object manipulation in virtual reality using an ETHD,” in *2021 IEEE world haptics conference (WHC)*, 373–378. doi:10.1109/WHC49131.2021.9517250
- Mercado, V. R., Argelaguet, F., Casiez, G., and Lécuyer, A. (2022). Watch out for the robot! designing visual feedback safety techniques when interacting with encountered-type haptic displays. *Front. Virtual Real.* 3, 928517. doi:10.3389/frvir.2022.928517
- Michalos, G., Makris, S., Tsarouchi, P., Guasch, T., Kontovrakis, D., and Chrysosolouris, G. (2015). Design considerations for safe human-robot collaborative workplaces. *Procedia CIRP* 37, 248–253. doi:10.1016/j.procir.2015.08.014
- Moore, T., and Stouch, D. (2016). “A generalized extended kalman filter implementation for the robot operating system,” in *Intelligent autonomous systems 13*. Editors E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi (Cham: Springer International Publishing), 335–348.
- Mortezapoor, S., Vasylevska, K., Vonach, E., and Kaufmann, H. (2023). “CoboDeck: a large-scale haptic VR system using a collaborative Mobile robot,” in *2023 IEEE conference virtual reality and 3D user interfaces (VR)*, 297–307. doi:10.1109/VR55154.2023.00045
- Mortezapoor, S., Ghazanfari, M., Vonach, E., Hoffmann, J. P., and Kaufmann, H. (2025). “Ar monitoring system for user safety in vr encountered-type haptics,” in *2025 IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW)*, 316–318. doi:10.1109/VRW66409.2025.00074
- Onishi, Y., Takashima, K., Higashiyama, S., Fujita, K., and Kitamura, Y. (2022). “Waddlewalls: room-scale interactive partitioning system using a swarm of robotic partitions,” in *Proceedings of the 35th annual ACM symposium on user interface software and technology*, 1–15.
- Oyekan, J. O., Hutabarat, W., Tiwari, A., Grech, R., Aung, M. H., Mariani, M. P., et al. (2019). The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans. *Robotics Computer-Integrated Manuf.* 55, 41–54. doi:10.1016/j.rcim.2018.07.006
- Park, M. Y., Han, D., Lim, J. H., Shin, M. K., Han, Y. R., Kim, D. H., et al. (2019). Assessment of pressure pain thresholds in collisions with collaborative robots. *PLoS one* 14, e0215890. doi:10.1371/journal.pone.0215890
- Podkosova, I., Vasylevska, K., Schoenauer, C., Vonach, E., Fikar, P., Bronederk, E., et al. (2016). “Immersivedeck: a large-scale wireless vr system for multiple users,” in *2016 IEEE 9th workshop on software engineering and architectures for realtime interactive systems (SEARIS)*, 1–7. doi:10.1109/SEARIS.2016.7551581

- Raatz, A., Blankemeyer, S., Runge, G., Bruns, C., and Borchert, G. (2015). Opportunities and challenges for the design of inherently safe robots. *Soft robotics Transferring theory Appl.*, 173–183. doi:10.1007/978-3-662-44506-8_15
- Robotic Industries Association (2020). ANSI/RIA R15.08-1-2020 industrial Mobile robots - safety requirements - part 1: requirements for the industrial Mobile robot. Washington, DC, United States: Robotic Industries Association. Available online at: https://webstore.ansi.org/standards/ria/ansir15082020?srsId=AfmBOoq65r8TFuADecY0gtBc4kgi4ootHTRonK_BYjrlxDka-2oeRfT.
- Robotic Industries Association (2023). ANSI/RIA R15.08-2-2023 industrial Mobile robots - safety requirements - part 2: requirements for IMR system(s) and IMR application(s). Ann Arbor, MI, United States: Robotic Industries Association. Available online at: https://webstore.ansi.org/standards/ria/ansia3r15082023?srsId=AfmBOo8rRn2hdgsGXQiWmg0Pyq71U0U-ptETHYdSZgLBL_Vcy0DvBr.
- Rubagotti, M., Tusseyeva, I., Baltabayeva, S., Summers, D., and Sandygulova, A. (2022). Perceived safety in physical human-robot Interaction—a survey. *Robotics Aut. Syst.* 151, 104047. doi:10.1016/j.robot.2022.104047
- Sallnäs, E.-L., Rasmus-Gröhn, K., and Sjöström, C. (2000). Supporting presence in collaborative environments by haptic force feedback. *ACM Trans. Computer-Human Interact.* 7, 461–476. doi:10.1145/365058.365086
- Schillinger, P., Kohlbrecher, S., and Von Stryk, O. (2016). “Human-robot collaborative high-level control with application to rescue robotics,” in *2016 IEEE international conference on robotics and automation (ICRA)* (IEEE), 2796–2802.
- Sisbot, E. A., Marin-Urias, L. F., Alami, R., and Simeon, T. (2007). A human aware mobile robot motion planner. *IEEE Trans. Robotics* 23, 874–883. doi:10.1109/tro.2007.904911
- Suzuki, R., Hedayati, H., Zheng, C., Bohn, J. L., Szafir, D., Do, E. Y.-L., et al. (2020). “RoomShift: room-scale dynamic haptics for VR with furniture-moving swarm robots,” in *Proceedings of the 2020 CHI conference on human factors in computing systems* (New York, NY, USA: Association for Computing Machinery), 1–11. doi:10.1145/3313831.3376523
- Vonach, E., Gatterer, C., and Kaufmann, H. (2017). “VRRobot: robot actuated props in an infinite virtual environment,” in *2017 IEEE virtual reality (VR)*, 74–83. doi:10.1109/VR.2017.7892233
- Wang, Y., Chen, Z. T., Li, H., Cao, Z., Luo, H., Zhang, T., et al. (2020). “MoveVR: enabling multi-form force feedback in virtual reality using household cleaning robot,” in *Proceedings of the 2020 CHI conference on human factors in computing systems* (New York, NY, USA: Association for Computing Machinery), 1–12. doi:10.1145/3313831.3376286
- Weng, Y.-H., Han, P.-H., Chang, K.-N., Lin, C.-Y., Lin, C.-H., Ng, H. Y., et al. (2025). Hit around: substitutional moving robot for immersive and exertion interaction with encountered-type haptic. *IEEE Trans. Vis. Comput. Graph.* 31, 3569–3579. doi:10.1109/tvcg.2025.3549556
- Yamaguchi, K., Kato, G., Kuroda, Y., Kiyokawa, K., and Takemura, H. (2016). “A non-grounded and encountered-type haptic display using a drone,” in *Proceedings of the 2016 symposium on spatial user interaction - SUI '16* (New York, New York, USA: ACM Press), 43–46. doi:10.1145/2983310.2985746
- Yixian, Y., Takashima, K., Tang, A., Tanno, T., Fujita, K., and Kitamura, Y. (2020). “Zoomwalls: dynamic walls that simulate haptic infrastructure for room-scale vr world,” in *Proceedings of the 33rd annual ACM symposium on user interface software and technology*, 223–235.
- Yokokohji, Y., Hollis, R. L., and Kanade, T. (1996). “What you can see is what you can feel - development of a visual/haptic interface to virtual environment,” in *Proceedings of the IEEE 1996 virtual reality annual international symposium*, 46–53.
- Yokokohji, Y., Muramori, N., Sato, Y., and Yoshikawa, T. (2005). Designing an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behaviors. *Int. J. Robotics Res.* 24, 717–729. doi:10.1177/0278364905057123
- Zacharaki, A., Kostavelis, I., Gasteratos, A., and Dokas, I. (2020). Safety bounds in human robot interaction: a survey. *Saf. Sci.* 127, 104667. doi:10.1016/j.ssci.2020.104667
- Zhou, Y., and Popescu, V. (2025). Dynamic redirection for safe interaction with ethd-simulated virtual objects. *IEEE Trans. Vis. Comput. Graph.* 31, 2923–2932. doi:10.1109/tvcg.2025.3549169