Universität Osnabrück
Institut für Informatik
AG Technische Informatik
Prof. Dr.-Ing. Mario Porrmann

Master's Thesis

Mixed Reality Environment for Robot-in-the-Loop-Testing using Digital Twins

Jonas Kittelmann

Betreuer: Prof. Dr.-Ing. Mario Porrmann
M.Sc. Philipp Gehricke

Osnabrück, Day Month 2025 M 00

Abstract

Hier sollte in einem Abstract kurz der Inhalt der Arbeit erläutert werden.
Zuerst auf deutsch.

Then an abstract of the thesis in english should follow.

Contents

1 Introduction

This chapter motivates the topic, frames the problem, states the objectives, and outlines the structure of the thesis.

1.1 Motivation

Mobile robotic systems are becoming increasingly integral to a diverse range of fields, including industry, research, and education. The development of applications for these complex systems fundamentally relies on robust processes for implementation, testing, and evaluation. A significant challenge in this process is bridging the divide between the controlled, flexible environment of pure software simulation and the high-fidelity, but often costly and hazardous, environment of real-world physical testing.

While initial validation can be effectively performed in simulated environments to gain early insights into system behavior, these virtual tests often fail to capture the full complexity of real-world dynamics. Consequently, physical field testing with actual hardware is crucial to uncover challenges and performance issues that are not apparent in a purely virtual setting. This creates a "testing gap" between the two methodologies. Pure simulation lacks physical realism, while physical testing is expensive, potentially dangerous, and difficult to conduct in a repeatable manner.

To bridge this gap, modern robotics development has embraced hybrid testing paradigms. The "X-in-the-Loop" (XitL) approach, particularly Robot-in-the-Loop (RitL) simulation, offers a powerful solution by integrating a physical robot with a simulated virtual environment. This method allows for the safe and repeatable testing of high-level algorithms on real hardware. The core technology enabling these frameworks is the Digital Twin, a virtualized, synchronized counterpart of a physical system that facilitates real-time monitoring and dynamic interaction. Furthermore, Mixed Reality (MR) technologies, including Augmented Reality (AR) and Virtual Reality (VR), are increasingly used to enhance human-robot interaction by providing intuitive interfaces for visualizing data and controlling the robot within these hybrid environments. This thesis is motivated by the need to create an advanced, high-fidelity platform that integrates these concepts to provide a more effective and scalable solution for the testing and validation of mobile robotic applications.

1.2 Structure of the Thesis

An overview of the following chapters.

2 Background and Related Work

Your text on the fundamentals and a review of existing literature and work.

2.1 Robot-in-the- Loop-Testing

The validation and verification of modern Robotic and Autonomous Systems (RAS) is a significant challenge due to their inherent complexity [AMV23]. This complexity arises from integrating many different fields, including software, mechanical, and electrical engineering [AMV23]. A central difficulty is testing the connection between the system's software ("cyber") and its hardware ("physical") aspects at the same time [AMV23]. To address this, the "X-in-the-Loop" (XitL) simulation paradigm acts as a crucial bridge, connecting the flexibility of pure software simulation with the high fidelity of full-scale physical testing [Bra+19].

A foundational XitL method is Hardware-in-the-Loop (HIL) simulation, which is a critical technique for testing complex mechatronic products [Mih+22; Bra+19]. The main principle of HIL is creating a closed loop between the real hardware being tested and a real-time computer simulation that represents the rest of the system or its operational environment [Mih+22]. This setup effectively "tricks" the hardware into behaving as if it were operating in the complete, real system, allowing for thorough testing across a wide range of virtual scenarios [Bra+19]. The primary reason for using HIL is to shorten development cycles and prevent costly or dangerous failures by making exhaustive testing possible before the system is deployed on the actual physical "plant" [Mih+22]. Because of this, HIL is essential in safety-critical industries like automotive, aerospace, and robotics, where real-world testing can be too expensive, dangerous, or even impossible [Bra+19].

Building on the HIL concept, Robot-in-the-Loop (RitL) simulation is a specialized paradigm where the hardware under test is a complete robotic system, such as an uncrewed vehicle [Mih+22]. RitL is directly related to HIL, replacing parts of a pure simulation with the physical robot to increase the test's fidelity [Hu05]. As illustrated in Figure **??**, a typical RitL configuration has the robot's real actuators operating in the physical world while its perception is fed by a simulated virtual environment [Hu05; Mih+22]. This creates a hybrid reality where, for instance, the robot might use "virtual sensors" to see objects in the simulation while using its real motors as HIL actuators to move physically [Hu05]. To keep the physical and virtual worlds synchronized, the real robot often has a "virtual counterpart" in the simulation whose state is updated as the physical robot moves and acts [Hu05].

Figure 2.1: Conceptual diagram of a Robot-in-the-Loop (RitL) simulation, showing the real robot interacting with a virtual environment. The real robot's actions affect its virtual counterpart, and the virtual environment provides sensor data back to the real robot.

The main benefit of RitL is the ability to conduct repeatable testing of high-level software, like navigation algorithms, using the dynamics of real hardware [Mih+22]. This avoids the cost, complexity, and danger of a full physical testbed [Mih+22]. RitL is therefore invaluable for safely validating system performance in a lab. This is especially important for safety-critical applications or for testing in environments that are hard to replicate, such as a Mars rover mission or a large-scale robot swarm [Hu05; Mih+22].

The RitL paradigm has been widely applied to validate complex autonomous systems, especially in the automotive industry through Vehicle-in-the-Loop (VitL) testing. The "Dynamic Vehicle-in-the-Loop" (DynViL) architecture, for instance, integrates a real test vehicle with the high-fidelity CARLA simulator, which is built on the Unreal Engine [DSR+22]. As shown in Figure **??**, this approach allows a physical vehicle on an empty track to be stimulated by sensor data from a complex virtual world [DSR+22]. This enables the safe and repeatable testing of automated driving functions in scenarios that would be dangerous to replicate physically [DSR+22]. A similar "Vehicle-in-Virtual-Environment" (VVE) method also uses CARLA and the Unreal Engine to create a closed loop where the real vehicle's motion is tracked and reflected in the virtual world, allowing its control systems to react to simulated events [Cao+23].

Figure 2.2: An example of a Vehicle-in-the-Loop (VitL) setup, illustrating a real car on a test track connected to a high-fidelity simulator like CARLA that generates virtual traffic and sensor data. Adapted from Drechsler et al. (2022) and Cao et al. (2023).

These examples show a clear trend towards using powerful, game-engine-based simulators to test autonomous vehicles. This approach increases safety and efficiency, and it bridges the gap between pure simulation, which lacks real vehicle dynamics, and physical testing, which is often not repeatable [Cao+23]. Furthermore, some frameworks extend this concept to directly stimulate the vehicle's sensors. For instance, a Radar Target Simulator (RTS) can feed artificial radar echoes from a virtual scene to the vehicle's real radar sensor, allowing for end-to-end validation of the entire perception and control pipeline [Die+21].

The versatility of the X-in-the-Loop concept is shown by its application in other domains. In marine robotics, a VIL framework was developed to test the long-term autonomy of a robot swarm. In this system, a real Autonomous Surface Vehicle (ASV) interacts with many simulated underwater sensor nodes to validate complex cooperative behaviors without the high logistical cost of deploying a full physical swarm [Bab+20]. In the aerospace domain, the RFlySim platform uses an FPGA-based HIL system to create a high-fidelity simulation for safely testing UAV autopilot systems in a lab, which reduces the need for expensive and risky outdoor flights [Dai+21].

The evolution of these methods points towards an even more integrated future. This includes the introduction of "Scenario-in-the-Loop" (SciL) frameworks that aim to completely blur the lines between real and virtual testing [Sza+21]. These next-generation systems rely on creating comprehensive digital twins of the entire test environment and coordinating a mix of real and virtual objects to run complex, mixed-reality test scenarios [Sza+21].

2.2 Digital Twins

A key concept that underpins many modern "X-in-the-Loop" frameworks is the Digital Twin (DT). The DT serves as the virtual counterpart to a physical system, acting as a virtual copy that enables real-time monitoring and simulation [AA23]. The core idea is to create a digital information model of a physical system that stays linked to it throughout its entire lifecycle [Gri17].

This model, first called the "Information Mirroring Model," has three primary components: the physical product, its corresponding virtual model, and the data connection that links them [Gri17; AA23]. A conceptual diagram of this structure is shown in Figure **??**. The virtual model is more than a simple geometric shape; it is often a complex simulation that can model a system's mechanical, electrical, and software properties [Len+21]. The data connection is bi-directional, allowing sensor data to flow from the real world to update the virtual model. In turn, the virtual model can send commands back to control or optimize the physical system [Gri17; Len+21].

This continuous, synchronized data exchange is the defining feature of a Digital Twin [AA23].

Figure 2.3: The foundational concept of a Digital Twin, illustrating the three core components: the physical entity, the virtual model, and the bi-directional data connection that links them. Adapted from Grieves et al. (2017).

In robotics, Digital Twins are often built using specialized simulation software. The Robot Operating System (ROS) typically acts as the middleware connecting the virtual simulation to the physical hardware. For instance, Stączek et al. used the Gazebo simulator with ROS to create a DT of a factory floor, which they used to test and optimize the navigation algorithms of a mobile robot in narrow corridors [Stą+21]. A similar approach was taken by R. Singh et al., who developed a DT of a greenhouse in Gazebo to validate a deep learning-based harvesting robot [Sin+24b]. Other simulators like CoppeliaSim and Webots are also common. Magrin et al. used CoppeliaSim and ROS to create a DT as a learning tool for mobile robot design [Mag+21], while Marques et al. used Webots for an Automated Guided Vehicle (AGV), synchronizing it via an OPC-UA server [Mar+24]. These examples show a common method of using traditional simulators to model robot kinematics and sensor feedback for closed-loop testing.

More recently, modern game engines have become a popular choice for creating Digital Twins, as they offer higher-fidelity graphics and more realistic physics. The Unity game engine, in particular, is used to build powerful, performant, and visually rich virtual environments. This can be seen in Figure **??**. For example, Pérez et al. used Unity3D to develop a DT of a multi-robot cell that included an immersive Virtual Reality (VR) interface for virtual commissioning and operator training [Pér+20].

Figure 2.4: A comparison of robotic Digital Twin environments, showing a typical view from a traditional simulator (left) versus a high-fidelity visualization from a modern game engine like Unity (right).

The technical capabilities of these engines are also a key factor. Yang et al. used Unity and its integrated NVIDIA PhysX engine to simulate the complex physics of a UAV. They also demonstrated how to create virtual sensors, such as a LiDAR, directly within the game engine using its raycasting API [Yan+20]. The performance and reliability of these game engine-based frameworks have also been a focus of research. Kwon et al. developed a safety-critical DT in Unity and ROS 2, achieving a low data-transmission latency of just 13 ms for predicting collisions [Kwo+25]. Similarly, M. Singh et al. created a Unity and ROS-based DT and conducted extensive performance validation, reporting a communication latency of 77.67 ms and a positional accuracy of 99.99% [Sin+24a].

2.3 Mixed Reality

Beyond the testing paradigm and the digital twin, the way a user interacts with the system is another critical part of a modern robotics framework. Virtual, Augmented, and Mixed Reality (VAM) have become promising technologies for improving the information exchange between humans and robots [Wal+23]. In robotics, Augmented Reality (AR) is an especially powerful tool for enhancing Human-Robot Interaction (HRI) by integrating 3D virtual objects into a real-world environment in real-time [MV20].

The relationship between these technologies is formally described by the foundational "Reality-Virtuality (RV) Continuum" concept, first introduced by Milgram and Kishino [MK94; Ska+21; MV20]. As illustrated in Figure **??**, this continuum is a scale that is anchored by a purely real environment at one end and a completely virtual one at the other [MK94; Ska+21; MV20].

Figure 2.5: The Reality-Virtuality Continuum, illustrating the spectrum from a real environment to a completely virtual one. Mixed Reality (MR) encompasses both Augmented Reality (AR) and Augmented Virtuality (AV). Adapted from Milgram & Kishino (1994) [MK94] and Walker et al. (2023) [Wal+23].

A conventional Virtual Reality (VR) environment represents the continuum's virtual endpoint. In VR, the user is totally immersed in and can interact with a completely synthetic world [MK94]. This approach is very useful for HRI research, as it allows for testing interactions with virtual robots in scenarios that might be unsafe or too expensive for physical hardware [Wal+23]. The general term Mixed Reality (MR) describes the entire spectrum between the two extremes, where real and virtual worlds are merged within a single display [MK94]. MR is made up of two main categories: Augmented Reality (AR) and Augmented Virtuality (AV) [MK94; MV20]. AR involves augmenting a real environment with virtual objects [MK94]. This allows HRI researchers to place a robot's 3D data and intentions directly into the user's physical space [Wal+23]. The opposite is Augmented Virtuality (AV), where a primarily virtual world is enhanced with elements from the real world, like live video feeds [MK94].

2.3.1 A Sub-Concept

Text...

3 Requirements

3.1 Requirements Analysis

Summarize stakeholder needs, constraints, and quality attributes (performance, fidelity, usability, portability).

3.2 Derivation from VERA Limitations

Translate the identified gaps of VERA into concrete functional and non-functional requirements.

4 Concept and Implementation

The Mixed Reality Environment for Robot-in-the-Loop Testing.

4.1 System Architecture

4.1.1 Integration of Unity, ROS 2, and EMAROS

Describe the components, their responsibilities, and deployment setup.

4.1.2 Communication Concept

Topics, message types, timing, synchronization, and time bases.

4.2 Development Milestones

4.2.1 Base Integration

Initial data flow, messaging bridge, and minimal scenes.

4.2.2 Realization of the Digital Twin

Modeling, coordinate frames, sensors, and actuators.

4.2.3 Interaction Scenarios

Scenario catalog and scripting.

4.2.4 VR Integration

Devices, interaction techniques, and UX considerations.

4.2.5 Pure Virtual Model

Implementation and use cases without the physical counterpart.

5 Evaluation

5.1 Methodology

5.1.1 Test Scenarios

Define representative scenarios and justification.

5.1.2 Metrics

Latency, frame rate, synchronization accuracy, and resource usage.

5.2 Execution and Results

5.2.1 Base Integration Performance

Report baseline measurements and analysis.

5.2.2 Scenario-dependent Performance

Discuss performance under different interaction and load conditions.

5.3 Discussion

Interpret results, limitations, threats to validity, and implications.

6 Conclusion

A summary of your work and discussion of future research directions.

List of Figures

Listings

Bibliography

[AA23] K. K. Alnowaiser and M. A. Ahmed. “Digital Twin: Current Research Trends and Future Directions”. In: *Arabian Journal for Science and Engineering* 48.2 (2023), pp. 1075–1095.

[AMV23] Hugo Araujo, Mohammad Reza Mousavi, and Mahsa Varshosaz. “Testing, Validation, and Verification of Robotic and Autonomous Systems: A Systematic Review”. In: *ACM Transactions on Software Engineering and Methodology* 32.2 (2023). ISSN: 1049-331X. DOI: `10.1145/3542945`. URL: `https://doi.org/10.1145/3542945`.

[Bab+20] Author Babic et al. “Vehicle-in-the-Loop in Marine Robotics”. In: *Proceedings*. Update with full citation. 2020.

[Bra+19] Author Brayanov et al. “X-in-the-Loop Simulation Overview”. In: *Proceedings*. Update with full citation. 2019.

[Cao+23] Author Cao et al. “Vehicle-in-Virtual-Environment method”. In: *Journal/-Conference* (2023). Update with full citation.

[Dai+21] Author Dai et al. “RFlySim: FPGA-based HIL for UAVs”. In: *Proceedings*. Update with full citation. 2021.

[Die+21] Author Diewald et al. “Radar Target Simulator for End-to-End Validation”. In: *Proceedings*. Update with full citation. 2021.

[DSR+22] Maikol Drechsler, Varun Sharma, Fabio Reway, et al. “Dynamic Vehicle-in-the-Loop: A Novel Method for Testing Automated Driving Functions”. In: *SAE International Journal of Connected and Automated Vehicles* 5 (June 2022), pp. 1–14. DOI: `10.4271/12-05-04-0029`.

[Gri17] Michael Grieves. *Origins of the Digital Twin Concept*. White paper; update with full citation. 2017.

[Hu05] Xiaolin Hu. “Applying robot-in-the-loop-simulation to mobile robot systems”. In: *2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 506–513. ISBN: 0-7803-9178-0. DOI: `10.1109/ICAR.2005.1507456`.

[Kwo+25] Author Kwon et al. “Safety-critical DT in Unity + ROS 2”. In: *Journal/-Conference* (2025). Update with full citation.

[Len+21] Author Leng et al. “Digital Twin Technologies Review”. In: *Journal/Conference* (2021). Update with full citation.

[Mag+21] Author Magrin et al. “DT for Mobile Robot Design with CoppeliaSim”. In: *Proceedings*. Update with full citation. 2021.

[Mar+24] Author Marques et al. “Webots-based AGV with OPC-UA Synchronization”. In: *Journal/Conference* (2024). Update with full citation.

[Mih+22] Author Mihalic et al. “Hardware-in-the-Loop for Mechatronics”. In: *Journal/Conference* (2022). Update with full citation.

[MK94] P. Milgram and F. Kishino. “A Taxonomy of Mixed Reality Visual Displays”. In: *IEICE Transactions on Information Systems* E77-D.12 (1994), pp. 1321–1329.

[MV20] Z. Makhataeva and H. A. Varol. “Augmented Reality for Robotics: A Review”. In: *Robotics* 9.2 (2020), p. 21.

[Pér+20] Author Pérez et al. “Unity3D DT for Multi-robot Cell with VR”. In: *Proceedings*. Update with full citation. 2020.

[Sin+24a] M. Singh et al. “Performance Validation of Unity + ROS DT”. In: *Journal/-Conference* (2024). Update with full citation.

[Sin+24b] R. Singh et al. “Greenhouse DT for Harvesting Robot in Gazebo”. In: *Proceedings*. Update with full citation. 2024.

[Ska+21] Richard Skarbez et al. “Revisiting the RV Continuum”. In: *Journal/Conference* (2021). Update with full citation.

[Stą+21] Author Stączek et al. “Factory Floor DT with Gazebo and ROS”. In: *Proceedings*. Update with full citation. 2021.

[Sza+21] Author Szalay et al. “Scenario-in-the-Loop (SciL) Frameworks”. In: *Journal/Conference* (2021). Update with full citation.

[Wal+23] Author Walker et al. “VAM for HRI”. In: *Journal/Conference* (2023). Update with full citation.

[Yan+20] Author Yang et al. “Unity + PhysX for UAV and Virtual Sensors”. In: *Proceedings*. Update with full citation. 2020.

Declaration of Authorship

I hereby declare that I have written this thesis independently and without unauthorized assistance. I have not used any sources or aids other than those indicated. All passages taken verbatim or in spirit from the works of other authors have been properly acknowledged and cited.

hesisDate

hesisAuthor