



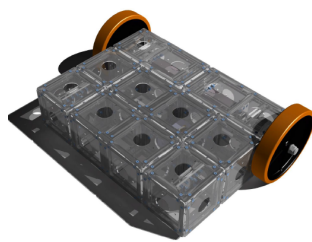
Build-A-Bot: Developing A Software Platform For A Modular Mobile Robot

Khaled Kassem
TU Wien
Vienna, Austria
khaled.k.kassem[at]tuwien.ac.at

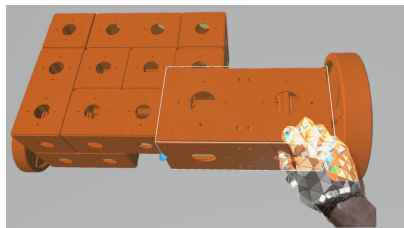
Galya Pavlova
TU Wien
Vienna, Austria
e01308054[at]student.tuwien.ac.at

Sebastian Schlund
TU Wien
Vienna, Austria
sebastian.schlund[at]tuwien.ac.at

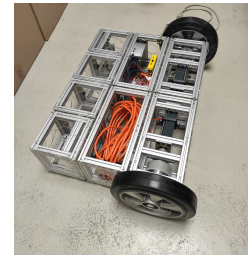
Florian Michahelles
TU Wien
Vienna, Austria
florian.michahelles[at]tuwien.ac.at



(a) 3D representation of the modular robot in CAD



(b) A user interacting with the simulated modular robot



(c) The physical prototype is in early production, in parallel to the software development

Figure 1: Three stages of developing a reconfigurable modular mobile robot.

ABSTRACT

This work aims to create a software platform that can manage dynamic changes in the configuration and functionalities of a modular robot that is currently in development, simulate its final appearance and behavior in augmented reality, and evaluate its usability through a user study.

We developed a software platform for a modular collaborative robot where users can add, remove, and swap modules without code changes. The software platform displays possible functions based on the current configuration of modules. To enable 3D interaction with the robot’s digital twin in real-time, we utilized an AR environment with the HoloLens. We conducted a user study with 28 participants without prior knowledge of robotics to evaluate the software’s usability and user experience for non-experts.

The study results indicate that the software platform was well-received and user-friendly, with the digital twin in an AR environment providing a realistic robot simulation. Participants’ subjective feedback on usability, user experience, and cognitive workload

of different software components was collected, and the analysis showed that the platform is suitable for non-experts.

The results showed that our design of the platform and its UI is well-accepted and easy to use, shown by a SUS score of 74.29. We show how we built our software platform, as well as the simulation in AR. Moreover, we propose the practice of using AR simulation to design the software platform before future physical prototype development, in order to test different scenarios, and possibly inform the physical design. Finally, we recommend that future work explores usability with a more diverse set of non-expert users, as well as different tasks.

ACM Reference Format:

Khaled Kassem, Galya Pavlova, Sebastian Schlund, and Florian Michahelles. 2023. Build-A-Bot: Developing A Software Platform For A Modular Mobile Robot. In *The International Conference on the Internet of Things (IoT 2023)*, November 07–10, 2023, Nagoya, Japan. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3627050.3627054>

1 INTRODUCTION

The Internet of Things (IoT) has revolutionized various industrial, educational, and experimental domains by enabling the deployment of smart and interconnected devices. Among these advancements, modular robots have emerged as a promising solution in modern industrial processes due to their flexibility and customization capabilities. These modular robots consist of interchangeable mechatronic modules that can be rapidly reconfigured on the fly,



This work is licensed under a Creative Commons Attribution International 4.0 License.

IoT 2023, November 07–10, 2023, Nagoya, Japan
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0854-1/23/11.
<https://doi.org/10.1145/3627050.3627054>

eliminating the need for multiple specialized robots for different functionalities [55].

However, the complexity of modular robots results in challenges in programming and controlling them effectively. Traditional robot programming methods are demanding and expensive, requiring diverse skill sets, from different disciplines [14, 35]. This complexity hinders the involvement of non-expert end-users, limiting the usability and adoption of these robots.

To address this challenge, our research presents a novel software platform tailored to the unique requirements of modular robots, without a need for reprogramming. Our software platform can dynamically adjust to changes in shape, size, and functionality that result from changing the configuration of a modular robot. Additionally, using the concept of digital twins, our platform allows users to interact with a simulated version of the modular robot in Augmented Reality (AR). Our approach facilitates the testing and deployment of Plug-N-Play modules, enabling rapid testing of reconfiguration and the resulting robot's functionality and morphology.

In this paper, we focus on the design, implementation, and evaluation of our software platform, with the primary objective of empowering non-expert users to seamlessly reconfigure and control the modular robot without the need for extensive programming knowledge. We address the following research questions:

- **RQ1:** How can we develop real-time software that effectively manages a modular mobile robot without the need for complex reprogramming?
- **RQ2:** To what extent is this software user-friendly and understandable for non-expert users in various environments?
- **RQ3:** What aspects of the software platform and interface may introduce additional cognitive workload for users?

To assess the usability and user experience of our software platform, we conducted a user study involving non-expert users. Study participants interacted with the modular robot in an AR environment, simulating the still-in-development physical prototype of the modular robot. The results of the study demonstrate the high level of acceptance and ease-of-use provided by our approach. The rest of this paper is structured as follows: Section 2 provides a review of related work on robotics in IoT and digital twins, emphasizing user-friendly robot programming approaches. Section 3 outlines the conceptual design and architecture of our software platform, followed by details of the user study in Section 4. In Section 5, we present the findings of the user study, followed by a discussion of the implications of our work in Section 6. Finally, Section 8 concludes the paper, highlighting the potential impact and future research directions for promoting the accessibility of modular robots within the IoT landscape.

This work offers the following contribution:

- We develop a software platform capable of handling configuration changes of the modular robot
- We test the functionality, usability, user experience, and workload resulting from users interacting with our platform in AR

- We offer recommendations for future work based on insights from our findings

2 RELATED WORK

This section provides an overview of prior research in the domains of robot programming, software platforms for robots, robot simulation, digital companions, and the integration of AR in industry.

2.1 Robot Programming

The complexity of robot software stems from the need to control various sensors and actuators in real-time, often amidst uncertainty and noise. Traditional robot programming requires specialized skills, making it challenging for many end-users [35, 59]. Researchers have explored solutions to make programming more accessible, such as end-user programming [15, 23, 42] and intuitive interfaces like voice-based programming [22]. The use of crowdsourcing for usability studies has also been investigated [43]. Specific applications include teleoperating a robotic arm using Robot Operating System (ROS) and HoloLens [62], and similar software relying on ROS and Oculus [40].

2.2 Software Platforms for Robots

The ROS has become a prominent platform for robot development, serving as the backbone for various robots [7, 18, 26, 36, 38, 39, 48, 61]. Research in reconfigurable robots has led to different categorizations and the use of ROS in reconfiguration [2, 11, 13, 24, 30, 46, 50].

2.3 Robot Simulation

Simulation environments like Gazebo and Unity are vital for robot development and testing. Gazebo is widely used for simulating diverse robots [1, 3–5, 9, 25, 39, 54, 56, 58], while Unity offers support for virtual and augmented reality applications [6, 21, 29, 44, 44, 52]. Collaborative map generation for simultaneous localization and mapping (SLAM) has also been proposed [51].

2.4 Augmented Reality

AR has been leveraged to enhance various aspects of robotics, including navigation, teleoperation, maintenance, and safety [28, 32, 63]. In medical robotics, AR has found applications in surgery, training, and rehabilitation [12, 31, 37, 41, 45]. The use of AR in robot programming has led to systems for defining virtual obstacles and tasks, reducing programming time, and improving program quality [17, 42].

Recent advancements have focused on the integration of AR with workplace learning and collaboration. This includes the development of shared virtual platforms for training [16], controlling smart factories through spatially-correct AR representations [27], and enhancing workplace learning [33]. Visualization of interactions between autonomous machines [34] and the creation of expert digital companions for working environments [53] are also notable contributions. The use of Head-mounted Display (HMD) devices like Microsoft HoloLens has further enabled natural human-robot interaction [31, 57, 60], and systems for supervising and coordinating multiple robots [47].

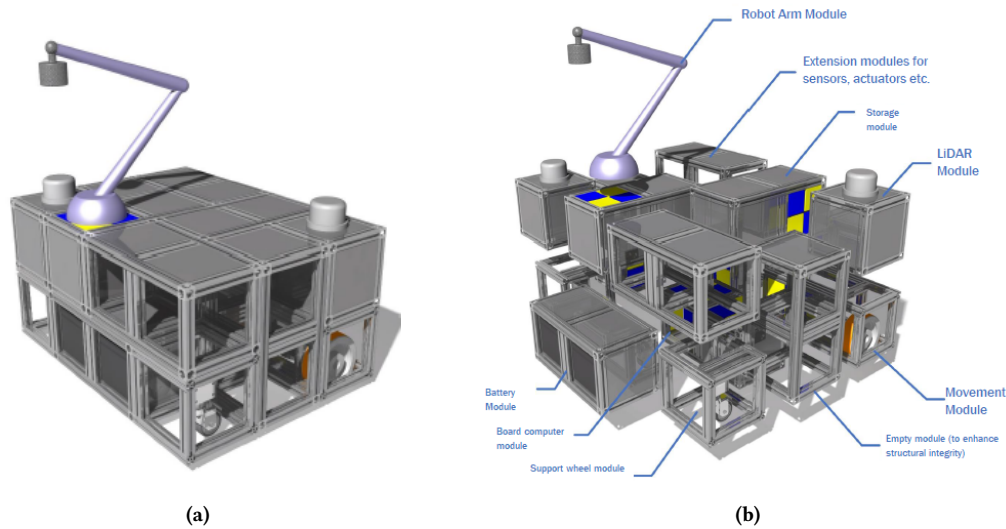


Figure 2: The modular robot. a) One possible configuration. b) Exploded view of the configuration showing the different modules used.

3 IMPLEMENTATION

In our work, the virtual robot acts as a placeholder for the in-development modular robot physical prototype. Therefore, adjusting the shape and dimensions of the different modules making up the robot was important to match the planned real counterparts.

3.1 Virtualization

To use the 3D models of the robots' parts in ROS, they need to be converted from STEP format to Unified Robotics Description Format (URDF), which is compatible with ROS. This is achieved by using SOLIDWORKS¹ in conjunction with the SOLIDWORKS to URDF Exporter plugin² to generate a single URDF file for each robot part. This simplifies the import process in both Unity and ROS, as the parts can be imported separately. Furthermore, having separate URDF files facilitates easy reconfiguration of the robot by adding or removing parts as needed.

After import, default scripts and Unity GameObjects are removed, and each part is saved as a prefabricated object, or a *prefab*. These prefabs serve as templates for creating new GameObjects with nested subparts in a hierarchical Parent-Child relationship. For the AR setup, menus are created using prefabs from the Mixed Reality Toolkit (MRTK)³ Toolbox in Unity, with scripts added for user and hand interactions, and a *ConstraintManager* to disable scaling and rotation of the parts.

The assembled robot parts form the body of the virtual robot, which is then placed inside an AR environment using a HMD (the Microsoft HoloLens). AR allows real-time interaction between the real and virtual components of the setup, allowing for the augmentation of the physical space where the robot exists with additional virtual 3D objects and audio. A virtual 3D representation of the

¹<https://www.solidworks.com/>

²http://wiki.ros.org/sw_urdf_exporter

³<https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2>
room is then created to allow the robot to navigate around objects in the room, such as furniture and other obstacles.

When using HoloLens with Unity, the camera position is not fixed, leading to varying positions of the robot in the 3D room each time the application is started. To solve this problem, we use a single Azure Spatial Anchors (ASA), which provides a fixed 3D position in the real world. On startup, we set the robot's position at the origin of the ASA, and create an empty GameObject *world* at the anchor's position to act as a coordinate system for localization and navigation in ROS. This allows us to transform the robot's position from ROS coordinate system to Unity using the *world* GameObject as it matches the coordinates of the map. Without the *world* coordinate system, we would need to store the initial robot's position in Unity on startup and recalculate the global position when a new position is received, making the process more complex and inefficient.

3.2 Control

Figure 3 presents the software architecture of the robot control system enabled with AR. The AR engine is developed using Unity and the MRTK to operate on the HoloLens. The right side illustrates the robotic backbone using ROS, responsible for simulating sensors, navigation planning and execution, and path planning for the robotic arm using libraries such as MoveIt [10] and Gazebo [25]. The HoloLens renders the user interface in the middle and allows the user to interact with the digital robot. The ROS TCP Connector package facilitates real-time data exchange between the HoloLens and ROS.

3.3 System Modes

We decided to design the prototype for typical tasks that a modular robot could perform. The system has three distinct modes, each with its specific functionality:

- **Configure Mode** - In this mode, the user can add or remove parts to the robot. It allows for the customization of the robot

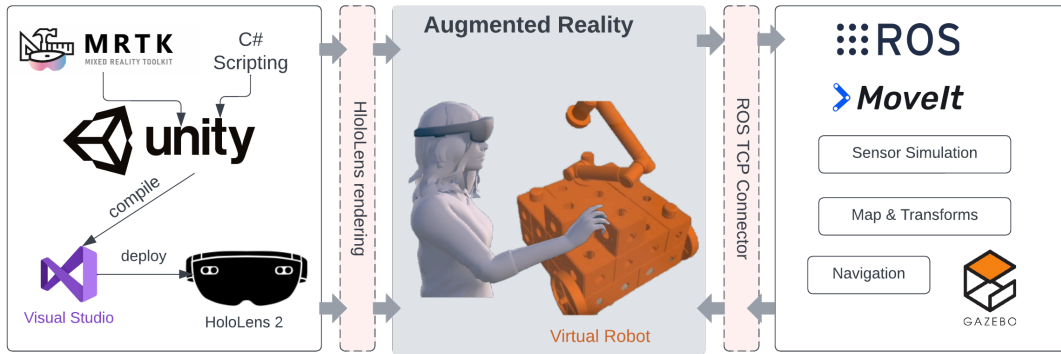


Figure 3: Overview of the software architecture.

to fit specific needs. This mode supports the simulation of reconfiguring the robot on the fly.

- **Move Mode** - This mode activates the navigation stack in ROS. When the user selects this mode, the robot initiates movement and attempts to stop in front of the user. This mode supports visual assessment of the robot's size and possible path in relation to surrounding physical objects, potential obstacles, and its environment.
- **Set Arm Goal Mode** - In this mode, the user specifies the goal for the robotic arm, which is displayed as a white sphere. This triggers the MoveIt package in ROS to plan and execute the arm's movement to the designated location. This mode is to support visualizing the reach of the current configuration of the robot, to assess the feasibility of arm-related tasks.

These modes provide users with various ways to interact with the robotic system and accomplish their intended goals. The system offers three modes, each of which can be activated through a button on the main menu displayed in Figure 4. The left palm facing the camera triggers the opening of the main menu. In addition, the system provides users with audio information and an information panel displayed below the buttons panel in Figure 4.



Figure 4: Overview of the supported modes - Configure (left), Move (middle) and Set Arm Goal (right).

4 USER STUDY

This section outlines the user study conducted to evaluate the usability and user experience of the software architecture proposed. We describe the procedure and the measures used to assess workload, usability, and user experience.

4.1 Participants

The study involved 28 participants (4 female, 24 male) aged between 20 and 35 ($M = 23$, $SD = 3.49$), who were university students (Computer Science, Software Engineering) and staff members with no expertise in robotics. One additional participant was excluded from the evaluation due to an interruption during the study procedure.

4.2 Procedure

The participants signed a consent form for data usage and were familiarized with the software and usage of the HoloLens, followed by the official HoloLens Tips application to learn the hand gestures and handling of the HoloLens. The participants started the software and completed a series of tasks:

- **Configure** - Participants were asked to build a robot with specific dimensions using blocks of their choice. The robot's width had to be 5 blocks and its length had to be 3 blocks.
- **Move** - Participants were instructed to send the robot to three different positions of their choosing. Paths to those positions could include obstacles such as chairs if desired.
- **Arm Goal** - Participants were asked to set three different goals for the robot's arm, with the goals being freely chosen by the participants.

To eliminate order effects, the task sequence was randomized for each participant. After completing each task, participants were asked to fill out the corresponding questionnaires. The study took each participant between 45 and 55 minutes to complete.

4.3 Measures

To evaluate the user interaction in AR, participants completed a questionnaire consisting of NASA Task Load Index (NASA-TLX) [19], User Experience Questionnaire (UEQ)-S [49], and System Usability Scale (SUS) [8]. NASA-TLX assessed workload,

while the short version of UEQ covered the user experience-related aspects of the study. Participants also rated their previous AR experience on a 5-point Likert scale from "None" (1) to "Expert" (5) and answered open-ended questions. We asked the participants about what they found most interesting about the software prototype, as well as what they would change about the software if given the opportunity. Finally, we asked them about any additional feedback they would like to share.

The perceived usability of the software was measured using SUS. The user experience is evaluated using the UEQ, which provides a measure for both pragmatic and hedonic qualities. The benchmarks used for comparison are derived from [20], which are based on the long version of the UEQ. The workload evaluation of three tasks is measured using NASA-TLX.

5 RESULTS

SUS. We started by calculating the SUS scores. Figure 5 displays the computed scores for each participant. The baseline average score of 68, is indicated by the green line in figure 5. 29% (8 participants) of the scores fell below the average baseline, with the lowest score of 55, while 29% (8 participants) gave grade A scores (above 80.3).

UEQ-S. The hedonic quality, with a mean of 1.64 and a standard deviation of 0.80, is classified as *excellent*, placing it within the top 10% of results. In contrast, the pragmatic quality has a mean of 0.86 and a standard deviation of 0.89, categorizing it as *below average*. Nonetheless, the overall evaluation is deemed *above average*, with a mean score of 1.25 and a standard deviation of 0.70. Figure 6 displays the computed average scores for each quality, as well as the overall score.

Item (3), which addresses the perceived efficiency of the software, received the lowest score of 0.3, categorized as *neutral* (below 0.8). There is a statistically significant moderate positive correlation between item (3) and the average SUS score, as well as SUS statement 1 (frequency) and statement 9 (confidence), and a moderate negative correlation with statement 8 (cumbersomeness). These correlations suggest that the perceived efficiency of the software has an impact on the overall usability experience and the user's frequency and confidence in using the software. Item (2) - perceived software complexity - lies slightly below the *neutral* score. The software complexity and the usability score (calculated SUS score) show a statistically significant strong correlation.

The two hedonic qualities with the highest scores are Item (5) and (6), which address excitement and interest in using this software. A Spearman's correlation test shows no statistically significant correlation between these items and the participants' experience. However, there is a statistically significant positive correlation between experience with AR and the overall hedonic property, meaning that the novelty effect is not due to unfamiliarity with the HoloLens, but can be traced back to the software.

NASA-TLX. The *Configure* task has the highest workload, except for the Performance scale. The *Move* task has the lowest workload in all scales, and the temporal demand is similar for all three tasks. The perceived performance is also similar for all tasks, with the *Arm Goal* task having the highest demand. The Effort score for the *Configure* task is notably lower than the highest score, which is 20.

A Friedman Test was performed for each of the three modes. The perceived performance is the same for each task ($\chi^2 = 1.188, p = .552$), but there is a statistically significant difference in the perceived mental, physical and temporal demand, as well as the effort and frustration, depending on which task is performed. The highest cognitive workload is confirmed to associate with the "Configure" task as evidenced by Table 1. The "Move" task has the lowest workload, with no significant difference compared to the "Arm Goal" task in overall workload.

Open-ended Questions. Participants expressed positive feedback about the software in open-ended questions, particularly in regard to the reconfiguration of the robot and the similarity to real-world experience. Some participants mentioned the virtual LEGO-like feel of the placing and snapping of blocks, the ability to view from all angles, declarative controlling, and the potential for the software to become an intuitive control system. On the other hand, the main criticism was for occasional errors or delays in rendering and visualizing the 3D objects after manipulation. One participant suggested adding an undo functionality.

6 DISCUSSION

We created a novel system that can dynamically manage a reconfigurable modular mobile robot, using ROS and Unity, and supporting six types of modules (caster, wheel, arm, LiDAR, and two sizes of chassis blocks) with up to 30 instances of each.

The usability results show that the overall perceived usability of the software in AR is above average, with an average SUS score of 74.29. However, it is important to note that 29% of the participants gave a score below the average baseline. This result, combined with the responses from the open-ended questions, indicates potential areas for improvement, such as improving simulation fidelity and reducing delay.

Performance issues and the need for more information about the robot were reported, which should be addressed in future work. The study results show that the reconfiguration of robot parts is the most demanding task with the highest cognitive load. However, this task is less frequent in real-life production scenarios compared to working with the robotic arm or moving the robot to new positions. Nevertheless, the high cognitive load of the configure task, compared to the other two tasks, indicates that one possible improvement would be investigating how to present the interface for this task more intuitively. Future work could attempt to apply

Factor	χ^2	p	Kendall's W
Mental Demand	36.400	< .001	0.650
Physical Demand	43.438	< .001	0.804
Temporal Demand	12.849	0.002	0.229
Performance	1.188	0.552	0.021
Effort	35.843	< .001	0.640
Frustration	28.714	< .001	0.513
Overall	33.817	< .001	0.604

Table 1: Friedman Test results for each NASA-TLX workload subscale.

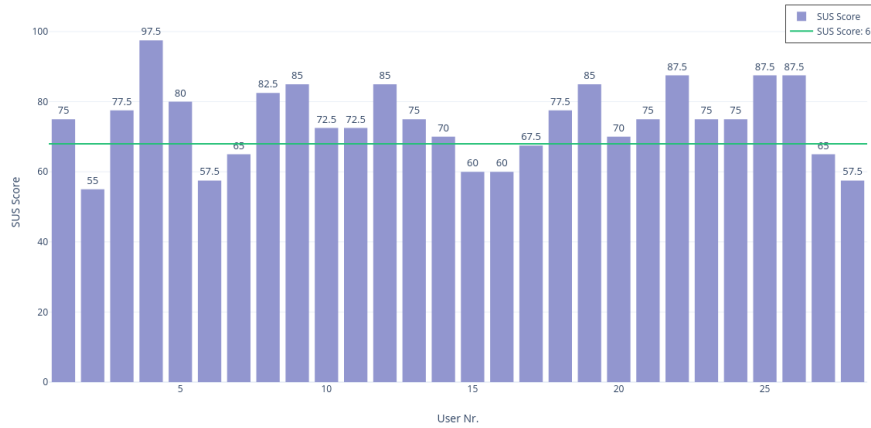


Figure 5: The calculated SUS score for each study participant (purple bar) with the SUS score of 68 as benchmark (green line).

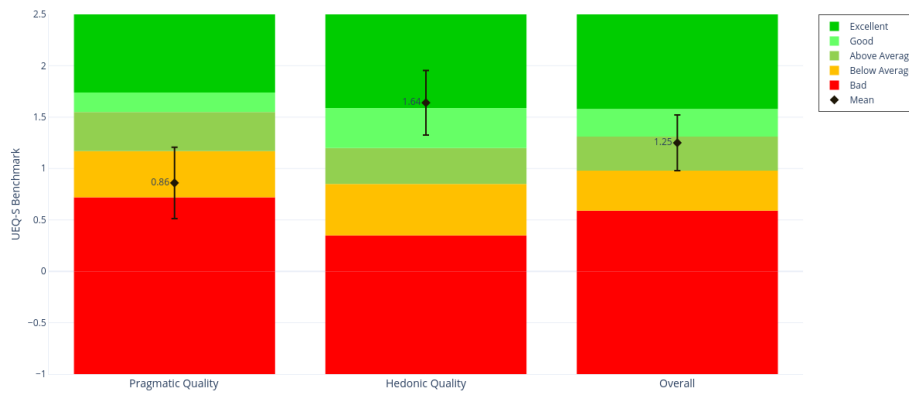


Figure 6: The mean UEQ score (with 95% confidence margins) for pragmatic quality (left), hedonic quality (middle) and overall (right). Each bar shows the color-coded benchmarks, presented by [20].

insights from the domain of user interface (UI) and user experience (UX) to reach this goal. The interaction with the robotic arm and navigation of the robot showed low cognitive demand, and the declarative controlling was seen as a positive aspect.

On the other hand, the same number of non-expert participants gave scores above grade A, indicating that the software was highly usable for them. The 99% confidence interval shows that the true mean score is likely between grades C and B, which is also above the average benchmark. Despite reporting they would use the software less frequently in the future due to lack of experience and their limited availability of the HoloLens, some participants found the software exciting and interesting to use. The software could be used without the need for large changes or extensive training.

The simulation is done in ROS, while HoloLens is primarily used for visualization. While functional and usable by non-experts,

one limitation of this approach is that it requires maintaining the two frameworks concurrently, which requires higher effort from the software maintenance aspect compared to a single system. An additional challenge is the simulation of the physics, kinematics, mass, and gravity distributions of the used modules. This would add a processing overhead, but would ultimately improve the fidelity of the simulation.

7 LIMITATIONS

The study was conducted with a group composed of computer science students and technical university staff. This means that the user sample could have a higher affinity to AR-based applications than the general non-expert population of potential users. To mitigate this effect we carried out the correlation analysis with

AR experience, but the underlying effect of higher affinity to technology should be acknowledged. To address this, future work should look into a more diverse set of participants, in order to reinforce the generalizability of our findings.

8 CONCLUSION AND OUTLOOK

The software developed successfully enabled non-expert users to interact with a digital twin-inspired simulation of the in-development robot in a mixed reality environment. The software currently supports three interactions: reconfiguring the robot by adding or removing modules, sending navigation commands, and setting a goal for the robotic arm (RQ1). To make our code available to the scientific community, we hosted it in a GitHub repository that we can share with other researchers upon request.

To evaluate the user experience and usability of the software (RQ2), a study with 28 participants was conducted. The results indicate that the software is user-friendly, understandable, and accessible to non-experts without prior knowledge of robotics or AR. The self-rated workload of the study participants was used to compare the three different modes of interaction. The results suggest that reconfiguring the robot parts has the highest cognitive workload (RQ3), while interacting with the robot's navigation is the most efficient and least cognitively demanding task.

One promising direction for research is examining how this approach compares to other approaches (e.g. block-based programming approach [59]) in helping non-experts utilize the configurable robot. Comparing the simulation to the real-world robot would also be beneficial. Questions regarding how similar the simulation is to the real-world robot, as well as how accurate the navigation and LiDAR simulation are, should also be studied. One potential direction of research is how such a tool can be used to inform or adjust the ongoing construction of the physical prototype. Another direction should examine how our findings generalize to a larger and more diverse sample of non-expert participants with more varied backgrounds.

In summary, this work proposes a novel way of simulating robots by combining ROS with an AR device. Although still in the early phases, this software has proven to be an intriguing and convenient way of human-robot interaction. It has the potential to become a low-cost training tool, a new and more realistic way of teleoperation, or a part of the prototyping process for new robots. We provide the source code and documentation, for the scientific community to build on this work.

REFERENCES

- [1] Ilya Afanasyev, Artur Sagitov, and Evgeni Magid. 2015. ROS-Based SLAM for a Gazebo-Simulated Mobile Robot in Image-Based 3D Model of Indoor Environment. In *Advanced Concepts for Intelligent Vision Systems*. Springer International Publishing, 273–283.
- [2] Hossein Ahmadzadeh, Ellips Masehian, and Masoud Asadpour. 2015. Modular Robotic Systems: Characteristics and Applications. *Journal of Intelligence and Robotic Systems* Under press (06 2015). <https://doi.org/10.1007/s10846-015-0237-8>
- [3] Murat Aksu, John L. Michaloski, and Frederick M. Proctor. 2018. Virtual Experimental Investigation for Industrial Robotics in Gazebo Environment (*ASME International Mechanical Engineering Congress and Exposition, Vol. Volume 2: Advanced Manufacturing*). <https://doi.org/10.1115/IMECE2018-87686>
- [4] Mark Allan, Uland Wong, P. Michael Furlong, Arno Rogg, Scott McMichael, Terry Welsh, Ian Chen, Steven Peters, Brian Gerkey, Moraan Quigley, Mark Shirley, Matthew Deans, Howard Cannon, and Terry Fong. 2019. Planetary Rover Simulation for Lunar Exploration Missions. In *2019 IEEE Aerospace Conference*. 1–19. <https://doi.org/10.1109/AERO.2019.8741780>
- [5] Rachmad Andri Atmoko and Daoguo Yang. 2018. Online Monitoring & Controlling Industrial Arm Robot Using MQTT Protocol. In *2018 IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robotics)*. 12–16. <https://doi.org/10.1109/ROBIONETICS.2018.8674672>
- [6] Christoph Bartneck, Marius Soucy, Kevin Fleuret, and Eduardo B. Sandoval. 2015. The robot engine – Making the unity 3D game engine work for HRI. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 431–437. <https://doi.org/10.1109/ROMAN.2015.7333561>
- [7] Wendwosen Bellete Bedada, Rawan Kalawoun, Ismayil Ahmadi, and Gianluca Palli. 2020. A Safe and Energy Efficient Robotic System for Industrial Automatic Tests on Domestic Appliances: Problem Statement and Proof of Concept. *Procedia Manufacturing* 51 (01 2020), 454–461. <https://doi.org/10.1016/j.promfg.2020.10.064>
- [8] John Brooke. 1996. SUS: a quick and dirty' usability. *Usability Evaluation In Industry* (1996), 189.
- [9] Denis Chikurtev. 2020. Mobile Robot Simulation and Navigation in ROS and Gazebo. In *2020 International Conference Automatics and Informatics (ICAI)*. 1–6. <https://doi.org/10.1109/ICAI50593.2020.9311330>
- [10] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. 2014. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. <https://doi.org/10.48550/ARXIV.1404.3785>
- [11] J. Denavit and R. S. Hartenberg. 1955. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Journal of Applied Mechanics* 22, 2 (June 1955), 215–221. <https://doi.org/10.1115/1.4011045>
- [12] Ryan Dickey, Neel Srikishen, LarryI Lipshultz, PhilippeE Spiess, RafaelE Carrion, and Tariq Hakky. 2015. Augmented Reality Assisted Surgery (ARAS): A Urologic Training Tool. *Asian Journal of Andrology* 18 (12 2015). <https://doi.org/10.4103/1008-682X.166436>
- [13] Anubhav Dogra, Sakshay Mahna, Srikanth Sekhar Padhee, and Ekta Singla. 2022. Unified modeling of unconventional modular and reconfigurable manipulation system. *Robotics and Computer-Integrated Manufacturing* 78 (2022), 102385. <https://doi.org/10.1016/j.rcim.2022.102385>
- [14] Markus Ehrenmann, Oliver Rogalla, Raoul Daniel Zöllner, and Rüdiger Dillmann. 2001. Teaching Service Robots Complex Tasks: Programming by Demonstration for Workshop and Household Environments.
- [15] Krzysztof Foit and Grzegorz Cwika. 2017. The CAD drawing as a source of data for robot programming purposes – a review. *MATEC Web of Conferences* 94 (2017), 05002. <https://doi.org/10.1051/mateconf/20179405002>
- [16] Markus Funk, Mareike Kritzler, and Florian Michahelles. 2017. HoloCollab: A Shared Virtual Platform for Physical Assembly Training Using Spatially-Aware Head-Mounted Displays. In *Proceedings of the Seventh International Conference on the Internet of Things (IoT '17)*. Association for Computing Machinery, New York, NY, USA, 1–7. <https://doi.org/10.1145/3131542.3131559>
- [17] Andre Gaschler, Maximilian Springer, Markus Rickert, and Alois Knoll. 2014. Intuitive robot tasks with augmented reality and virtual obstacles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. <https://doi.org/10.1109/icra.2014.6907747>
- [18] Peng Guo, Haichao Shi, Shijie Wang, Liansheng Tang, and Zipeng Wang. 2022. An ROS Architecture for Autonomous Mobile Robots with UCAR Platforms in Smart Restaurants. *Machines* 10 (09 2022), 844. <https://doi.org/10.3390/machines10100844>
- [19] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9)
- [20] Andreas Hinderks, Martin Schrepp, and Jörg Thomaschewski. 2018. A Benchmark for the Short Version of the User Experience Questionnaire. In *International Conference on Web Information Systems and Technologies*. <https://doi.org/10.5220/0007188303730377>
- [21] Ahmed Hussein, Fernando Garcia, and Cristina Olaverri Monreal. 2018. ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation. <https://doi.org/10.1109/ICVES.2018.8519522>
- [22] Tudor B. Ionescu and Sebastian Schlund. 2022. Programming cobots by voice: a pragmatic, web-based approach. *International Journal of Computer Integrated Manufacturing* 36, 1 (Nov. 2022), 86–109. <https://doi.org/10.1080/0951192x.2022.2148754>
- [23] Yuya kaneshige, Satoru Satake, Takayuki Kanda, and Michita Imai. 2021. How to Overcome the Difficulties in Programming and Debugging Mobile Social Robots?. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. ACM. <https://doi.org/10.1145/3434073.3444674>
- [24] Yeon Kang, Donghan Kim, and Kwangjin Kim. 2019. URDF Generator for Manipulator Robot. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*. 483–487. <https://doi.org/10.1109/IRC.2019.00101>
- [25] N. Koenig and A. Howard. 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Vol. 3. 2149–2154 vol.3. <https://doi.org/10.1109/IROS.2004.1389727>

- [26] Jason Kreitz, Moonyoung Lee, Hyun SubPark, Paul Y. Oh, and Jun Ho Oh. 2020. Implementing ROS Communications for Sensor Integration with the RB5 Collaborative Robot. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. 0378–0383. <https://doi.org/10.1109/CCWC47524.2020.9031202>
- [27] Mareike Kritzler, Markus Funk, Florian Michahelles, and Wolfgang Rohde. 2017. The Virtual Twin: Controlling Smart Factories Using a Spatially-Correct Augmented Reality Representation. In *Proceedings of the Seventh International Conference on the Internet of Things (IoT '17)*. Association for Computing Machinery, New York, NY, USA, 1–2. <https://doi.org/10.1145/3131542.3140274>
- [28] Ryotaro Kuriya, Takeshi Tsujimura, and Kiyotaka Izumi. 2015. Augmented reality robot navigation using infrared marker. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 450–455. <https://doi.org/10.1109/ROMAN.2015.7333607>
- [29] Yunwang Li, Sumei Dai, Yong Shi, Lala Zhao, and Minghua Ding. 2019. Navigation Simulation of a Mecanum Wheel Mobile Robot Based on an Improved A* Algorithm in Unity3D. *Sensors* 19 (07 2019), 2976. <https://doi.org/10.3390/s19132976>
- [30] Jinguo Liu, Xin Zhang, and Guangbo Hao. 2016. Survey on Research and Development of Reconfigurable Modular Robots. *Advances in Mechanical Engineering* 8 (08 2016). <https://doi.org/10.1177/1687814016659597>
- [31] Zhanat Makhataeva and Huseyin Atakan Varol. 2020. Augmented Reality for Robotics: A Review. *Robotics* 9, 2 (2020). <https://doi.org/10.3390/robotics9020021>
- [32] Zhanat Makhataeva, Altay Zhakatayev, and Huseyin Atakan Varol. 2019. Safety Aura Visualization for Variable Impedance Actuated Robots. In *2019 IEEE/SICE International Symposium on System Integration (SII)*. 805–810. <https://doi.org/10.1109/SII.2019.8700332>
- [33] Bernhard Mandl, Marius Stehling, Thomas Schmiedinger, and Martin Adam. 2017. Enhancing Workplace Learning by Augmented Reality. In *Proceedings of the Seventh International Conference on the Internet of Things (IoT '17)*. Association for Computing Machinery, New York, NY, USA, 1–2. <https://doi.org/10.1145/3131542.3140265>
- [34] Simon Mayer and Florian Michahelles. 2017. HoloInteractions: Visualizing Interactions between Autonomous Cognitive Machines. In *Proceedings of the Seventh International Conference on the Internet of Things (IoT '17)*. Association for Computing Machinery, New York, NY, USA, 1–2. <https://doi.org/10.1145/3131542.3140278>
- [35] U. Nehmzow, O. Akanyeti, Christoph Weinrich, Theocharis Kyriacou, and S.A. Billings. 2007. Robot programming by demonstration through system identification. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE*. <https://doi.org/10.1109/iros.2007.4399087>
- [36] Sameul Noh, Jiyoung Park, and Junhee Park. 2020. Autonomous Mobile Robot Navigation in Indoor Environments: Mapping, Localization, and Planning. 908–913. <https://doi.org/10.1109/ICTC49870.2020.9289333>
- [37] Renz Ocampo and Mahdi Tavakoli. 2019. Visual-Haptic Colocation in Robotic Rehabilitation Exercises Using a 2D Augmented-Reality Display. In *2019 International Symposium on Medical Robotics (ISMR)*. 1–7. <https://doi.org/10.1109/ISMR.2019.8710185>
- [38] Miguel Oliveira, Afonso Castro, Tiago Madeira, Eurico Pedrosa, Paulo Dias, and Vitor Santos. 2020. A ROS framework for the extrinsic calibration of intelligent vehicles: A multi-sensor, multi-modal approach. *Robotics and Autonomous Systems* 131 (2020), 103558. <https://doi.org/10.1016/j.robot.2020.103558>
- [39] Diego Osorio-Sanchez, Hugo G. Gonzalez-Hernandez, Rogelio Bustamante-Bello, and Jorge A. Reyes-Avendano. 2021. ROS-Gazebo and MATLAB Simulations of a Multi-purpose Ground Vehicle. In *The 4th International Conference on Electronics, Communications and Control Engineering*. ACM. <https://doi.org/10.1145/3462676.3462692>
- [40] Lorenzo Peppoloni, Filippo Brizzi, Carlo Alberto Avizzano, and Emanuele Ruffaldi. 2015. Immersive ROS-integrated framework for robot teleoperation. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 177–178. <https://doi.org/10.1109/3DUI.2015.7131758>
- [41] Patrick Pessaux, Michele Diana, Luc Soler, Tullio Piardi, Didier Mutter, and Jacques Marescaux. 2014. Towards cybernetic surgery: robotic and augmented reality-assisted liver segmentectomy. *Langenbeck's archives of surgery / Deutsche Gesellschaft fur Chirurgie* 400 (11 2014). <https://doi.org/10.1007/s00423-014-1256-9>
- [42] T. Pettersen, J. Pretlove, C. Skourup, T. Engedal, and T. Lokstad. 2003. Augmented reality for programming industrial robots. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* IEEE Comput. Soc. <https://doi.org/10.1109/ismar.2003.1240739>
- [43] Daria Piacun, Tudor B. Ionescu, and Sebastian Schlund. 2021. Crowdsourced Evaluation of Robot Programming Environments: Methodology and Application. *Applied Sciences* 11, 22 (Nov. 2021), 10903. <https://doi.org/10.3390/app112210903>
- [44] Jonathan Platt and Kenneth Ricks. 2022. Comparative Analysis of ROS-Ubuntu3D and ROS-Gazebo for Mobile Ground Robot Simulation. *Journal of Intelligent & Robotic Systems* 106 (12 2022). <https://doi.org/10.1007/s10846-022-01766-2>
- [45] Long Qian, Anton Deguet, and Peter Kazanzides. 2018. ARssist: Augmented Reality on a Head-Mounted Display for the First Assistant in Robotic Surgery. *Healthcare Technology Letters* 5 (08 2018). <https://doi.org/10.1049/htl.2018.5065>
- [46] Francisco Ramos, Andres Vazquez, Raúl Fernández, and Alberto Olivares-Alarcos. 2018. Ontology based design, control and programming of modular robots. *Integrated Computer-Aided Engineering* 25 (02 2018), 1–20. <https://doi.org/10.3233/ICA-180569>
- [47] Frank Regal, Christina Petlowany, Can Pehlivanurk, Corrie Van Sice, Chris Suarez, Blake Anderson, and Mitch Pryor. 2022. AugRE: Augmented Robot Environment to Facilitate Human-Robot Teaming and Communication. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 800–805. <https://doi.org/10.1109/RO-MAN53752.2022.9900721>
- [48] Casper Schou and Ole Madsen. 2017. A plug and produce framework for industrial collaborative robots. *International Journal of Advanced Robotic Systems* 14 (07 2017), 17298814171747. <https://doi.org/10.1177/1729881417171747>
- [49] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. 2017. Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence* 4 (01 2017), 103. <https://doi.org/10.9781/ijimai.2017.09.001>
- [50] Ming-Chuan Shiu, Hou-Tsan Lee, Feng-Li Lian, and Li-Chen Fu. 2008. Design of the Octobot Self-Reconfigurable Robot. *IFAC Proceedings Volumes* 41, 2 (2008), 15630–15635. <https://doi.org/10.3182/20080706-5-KR-1001.02643> 17th IFAC World Congress.
- [51] Abbas Sidaoui, Imad H. Elhajj, and Daniel Asmar. 2019. Collaborative Human Augmented SLAM. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2131–2138. <https://doi.org/10.1109/IROS40897.2019.8968027>
- [52] Enrico Sita, Csongor Horvath, Trygve Thomessen, Péter Korondi, and Anthony Pipe. 2017. ROS-Ubuntu3D based system for monitoring of an industrial robotic process. 1047–1052. <https://doi.org/10.1109/SII.2017.8279361>
- [53] Janick Spirig, Kimberly Garcia, and Simon Mayer. 2022. An Expert Digital Companion for Working Environments. In *Proceedings of the 11th International Conference on the Internet of Things (IoT '21)*. Association for Computing Machinery, New York, NY, USA, 25–32. <https://doi.org/10.1145/3494322.3494326>
- [54] Aaron Staranowicz and Gian Luca Mariottini. 2011. A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. In *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments (Heraklion, Crete, Greece) (PETRA '11)*. Association for Computing Machinery, New York, NY, USA, Article 56, 8 pages. <https://doi.org/10.1145/2141622.2141689>
- [55] Kasper Stoy and David Brandt. 2013. Efficient enumeration of modular robot configurations and shapes. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE*. <https://doi.org/10.1109/iros.2013.6696972>
- [56] Sohan Suvarna, Dibayan Sengupta, Pavankumar Koratikere, and Rajkumar S. Pant. 2019. Simulation of Autonomous Airship on ROS-Gazebo Framework. In *2019 Fifth Indian Control Conference (ICC)*. 237–241. <https://doi.org/10.1109/INDIANCC.2019.8715570>
- [57] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. 1–33. <https://doi.org/10.1145/3491102.3517719>
- [58] Kenta Takaya, Toshinori Asai, Valeri Kroumov, and Florentin Smarandache. 2016. Simulation environment for mobile robots testing using ROS and Gazebo. In *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*. 96–101. <https://doi.org/10.1109/ICSTCC.2016.7790647>
- [59] David Weintrop, Afsoon Afzal, Jean Salac, Patrick Francis, Boyang Li, David C. Shepherd, and Diana Franklin. 2018. Evaluating CoBlox. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. <https://doi.org/10.1145/3173574.3173940>
- [60] Nele Witte, Fien Buelens, Glen Debar, Bert Bonroy, Wout Standaert, Fernando Tarnogol, and Tom Van Daele. 2022. Handheld or head-mounted? An experimental comparison of the potential of augmented reality for animal phobia treatment using smartphone and HoloLens 2. *Frontiers in Virtual Reality* 3 (12 2022). <https://doi.org/10.3389/frvir.2022.1066996>
- [61] Rui Xu, Lu Qian, and Xingwei Zhao. 2022. Development of dual-arm mobile robot platform based on ROS. *Cobot* 1 (01 2022), 4. <https://doi.org/10.12688/cobot.17457.1>
- [62] Chung Xue Er Shamaine, Yuansong Qiao, John Henry, Ken McNeven, and Niall Murray. 2020. RoSTAR: ROS-based Telerobotic Control via Augmented Reality. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSp)*. 1–6. <https://doi.org/10.1109/MMSp48831.2020.9287100>
- [63] A.W.W. Yew, S.K. Ong, and A.Y.C. Nee. 2017. Immersive Augmented Reality Environment for the Teleoperation of Maintenance Robots. *Procedia CIRP* 61 (2017), 305–310. <https://doi.org/10.1016/j.procir.2016.11.183> The 24th CIRP Conference on Life Cycle Engineering.