

# Optimizing Robot Programming: Mixed Reality Gripper Control

Maximilian Rettinger\*, Leander Hacker, Philipp Wolters, Gerhard Rigoll  
Technical University of Munich



Fig. 1. Illustration of the physical and virtual robot arm. The transparent shape represents the working envelope. The boxes on the left and right are used for the user study tasks.

**Abstract**—Conventional robot programming methods are complex and time-consuming for users. In recent years, alternative approaches such as mixed reality have been explored to address these challenges and optimize robot programming. While the findings of the mixed reality robot programming methods are convincing, most existing methods rely on gesture interaction for robot programming. Since controller-based interactions have proven to be more reliable, this paper examines three controller-based programming methods within a mixed reality scenario: 1) *Classical Jogging*, where the user positions the robot’s end effector using the controller’s thumbsticks, 2) *Direct Control*, where the controller’s position and orientation directly corresponds to the end effector’s, and 3) *Gripper Control*, where the controller is enhanced with a 3D-printed gripper attachment to grasp and release objects. A within-subjects study ( $n = 30$ ) was conducted to compare these methods. The findings indicate that the *Gripper Control* condition outperforms the others in terms of task completion time, user experience, mental demand, and task performance, while also being the preferred method. Therefore, it demonstrates promising potential as an effective and efficient approach for future robot programming. Video available at <https://youtu.be/83kWr8zUFIQ>.

## I. INTRODUCTION

Mixed reality (MR) [18] allows users to augment the physical environment with virtual elements, offering promising opportunities in fields ranging from healthcare [12,29,30]

to ordnance disposal [27,28]. For robot programming, this potential has been explored for several years [1,5,25] since traditional programming approaches are unintuitive, highly skill-demanding, or time-consuming [21,36]. For instance, in online programming, the robot is moved to a position via a teach pendant by jogging per cycle, which is tedious and time-consuming [19]. In offline programming, the task is created on an independent computer, which provides features like task simulation and collision detection but is time-consuming, demanding, and requires a complete digital model of the workspace [20]. Another example is walk-through programming, where the robot’s end effector is physically moved to the desired positions while the robot controller records these movements [6]. A concept based on this is programming by demonstration [13], in which the robot not only imitates the movements but also learns the tasks via machine learning to generalize these for similar situations [26]. MR robot programming offers a solution to the limitations of these conventional methods since this technology provides an efficient hybrid solution that combines online and offline programming. It allows users to overlay the physical robot environment with a virtual robot replica, as depicted in Figure 1, enabling them to program the virtual model first and simulate its movements. This process helps identify potential problems, such as collisions, early on. Once

\* e-mail: maximilian.reettinger@tum.de

the immersive simulation runs smoothly, the physical robot can reliably execute the programmed tasks without having to create a detailed digital model of the environment.

Research findings indicate that MR programming is more effective and efficient than conventional methods [8,24]. An analysis of recent publications in this field shows that the interaction is performed using gesture control in most cases [7,11,14,22,22]. This involves the use of a remote manipulation interaction technique for setting waypoints [8,24,34]. However, this method is less accurate than direct manipulation due to hand instability, inaccurate tracking systems, and difficulty mapping the user's movements to virtual objects [16,17]. In addition, studies have shown that the performance of gesture-based interaction is not as good as controller-based interaction [9,15] while using tangible objects leads to better results [2,3,28,33].

In this paper, we investigate the potential of controller-based MR robot programming by comparing the following three methods: 1) *Classical Jogging*, in which the user moves the virtual robot using the thumbsticks and buttons of a controller, 2) *Direct Control*, in which the position and orientation of the controller (6-DoF) is transferred to the virtual end effector, and 3) *Gripper Control*, which extends the *Direct Control* method with a mechanical gripper identical to that of the robot, enabling gripping and releasing of physical objects. These three methods are evaluated in a study ( $n = 30$ ) to determine their effectiveness and efficiency. A fast and uncomplicated robot programming solution for people with no previous experience or knowledge in human-robot interaction can accelerate the widespread introduction of robot technologies in small and medium-sized companies [6].

## II. RELATED WORK

Yang *et al.* [35] compared a video see-through MR robot programming system with an online programming method. The MR method used an Oculus Rift HMD and a physical handheld controller for interaction. In the online method, participants could choose between manual leading and pendant teaching. Both methods were evaluated in a within-subjects study ( $n = 16$ ) in which participants completed two complex programming tasks. The results showed that participants were faster, set fewer waypoints, and had less cognitive load when using the MR method. Gadre *et al.* [8] implemented an MR interface for robot programming using a Microsoft HoloLens with gesture interaction and compared it to a conventional offline 2D programming method (monitor, keyboard, and mouse). Participants had to perform two pick-and-place tasks in a within-subjects study ( $n = 20$ ). The analysis showed that the MR method was faster, easier, more natural, and less work-intensive than the conventional method. Quintero *et al.* [24] investigated an MR robot programming method by comparing it with a traditional kinesthetic teaching method. In a study ( $n = 10$ ), the participants performed free space and contact surface trajectory tasks. The interaction was performed via hand tracking using a Microsoft HoloLens and MYO armband. Results of a study showed that the MR method took less

time, performed better, and required less physical effort but more mental effort. Pizzagalli *et al.* [23] compared the two MR methods, AR (Microsoft HoloLens) and VR (Oculus Quest), with the online programming method teach pendant. Both MR methods were operated via gesture interaction. The results of a within-subjects study ( $n = 21$ ) showed that both MR methods were faster, easier to use, and less demanding for users than the conventional programming method. Dengxiong *et al.* [4] proposed a self-supervised 6-DoF grasp pose detection system through an MR teleoperation framework that efficiently learns useful grasp strategies from human demonstrations without requiring explicit grasp pose annotations. Thus, this method does not require precise grasp pose monitoring, which offers advantages in constrained environments. In real-world experiments, the proposed system demonstrated the ability to successfully grasp unknown objects after only three demonstrations.

## III. METHODS

In this section, we describe our novel robot programming approaches. As an apparatus for three methods, we use the Meta Quest 3<sup>1</sup> MR head-mounted display (HMD) as an interface. An Annin Robotics AR4 open-source robotic arm<sup>2</sup>, which is a desktop-sized 6-DoF industrial robot. A TCP relay server, running on a notebook, serves as the communication interface between the two components. In addition, a self-designed and 3D-printed controller extension is described in more detail in Section III-C.

All three programming methods have the same basic functions for simplifying usability. This includes rendering the virtual robot based on the specified end effector position and orientation by performing real-time inverse kinematics calculations. All methods use a waypoint-based programming approach, where users define key positions and orientations that form the robot's trajectory. If the movements of the end effector overstretch the physical joint limits, the corresponding joint is visually highlighted. The robot's working envelope is visualized via a transparent shape that looks like a sphere so the user can assess its range. Furthermore, the trajectory of the end effector is calculated and visualized using geometric path planning. It is also possible to simulate this trajectory during programming, which has the advantage that the user can safely see the movements at close range and does not have to maintain a safe distance as required for a physical robot. The virtual and physical robot bases are synchronized, so the movements in the augmented simulation match those of the physical robot. This precise synchronization is executed by inserting the 3D-printed gripper extension into a 3D-printed opening (see Figure 2) on the robot base. As the orientation and positioning of this opening relative to the physical robot and the tracked gripper controller are known, the virtual model is calibrated by executing a function with a controller button. The software and its functions are implemented with Unity<sup>3</sup>.

<sup>1</sup><https://www.meta.com/de/quest/quest-3/>

<sup>2</sup><https://www.anninrobotics.com/>

<sup>3</sup><https://unity.com/>

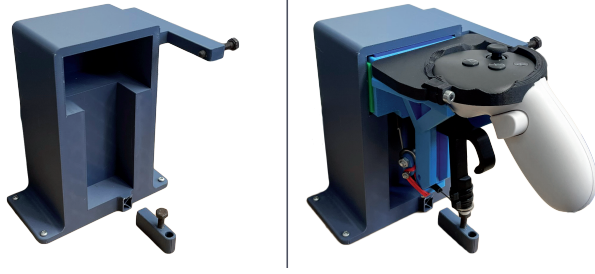


Fig. 2. Construcion for synchronization of the virtual and physical robot bases. It is executed by inserting the 3D-printed gripper extension into the 3D-printed opening case.

#### A. Classical Jogging

The *Classical Jogging* condition is used as a baseline since it reflects the conventional teach pendant method [19]. This allows the user to control the virtual end effector in two separately selectable modes. In Position Mode, the left joystick controls the end-effector's X- and Y-coordinates, while the right joystick controls its Z-coordinate. In Rotation Mode, the left joystick changes the pitch and yaw of the end-effector, while the right joystick modifies the roll angle. Modes can be switched using a controller button, while another button is used to program the defined waypoints. The gripper state can be changed using the grip trigger of a controller.

#### B. Direct Control

The *Direct Control* method allows the virtual end effector to follow the position and orientation of the Meta Quest controller in real-time. The waypoint can be captured with the controller button when the user reaches the desired position, orientation, and grasp status. As in the *Classical Jogging* condition, the state of the gripper can be changed by pressing the controller's grip trigger.

#### C. Gripper Control

In the *Gripper Control* method, as in the *Direct Control* method, the end effector's real-time position and orientation correspond with the controller's. The way objects are gripped and released differs here, as the Meta Quest controller is modified with a 3D-printed gripper extension shown in Figure 3. This extension is a tangible interface that transmits gripper states and provides haptic feedback when interacting with target objects. It allows the user to grip, move, and release physical objects while programming. This approach offers a conceptual advantage, although it has not been investigated in this contribution and is currently only a potential benefit, as the haptic force feedback may assist in handling fragile or deformable objects. All this is possible as the design of the 3D-printed gripper is identical to that of the servo gripper of the AR4 robot. The Gripper Controller is operable with a single hand, and its jaws are interchangeable for diverse pick-and-place tasks.

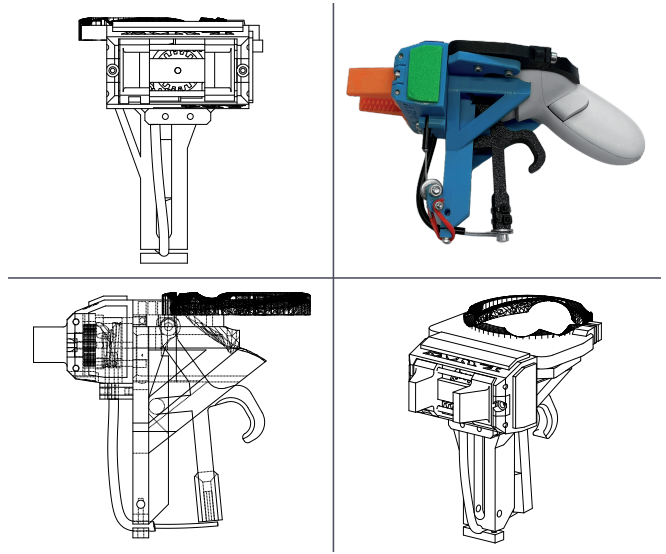


Fig. 3. Visualization of the gripper used for condition *Gripper Control*.

### IV. EVALUATION

#### A. Study Design

We conducted a within-subjects study design to prevent variations caused by individual differences and to obtain a final evaluation of all programming methods. Using the three methods (*Classical Jogging*, *Direct Control*, and *Gripper Control*) all participants completed the three levels of the independent variable. As dependent variables, we collected the participants' programming duration, subjective ratings for assessing the User Experience using the short version of the User Experience Questionnaire (UEQ-S) [31], the Workload using the Raw NASA Task Load Index Questionnaire (NASA-TLX) [10], and specific questions for the feasibility, natural usability, precision, and overall assessment. In all three methods, participants completed two tasks to assess their performance based on programming errors (e.g., collisions, incorrect grab/release of objects, or skipped programming steps). These errors were captured through software data logging and documented observations by the study supervisor.

#### B. Procedure

Initially, participants were briefed about the study and received a safety introduction (for the mixed reality and robotics hardware). Subsequently, they were asked to sign a consent form and provide demographic information. A video tutorial (8 : 40 min.) was presented to each participant to explain the functionality of the system and the three programming methods in an easy-to-understand and consistent manner. Participants then completed the three programming methods following the identical procedure for each method. First, they had five minutes to familiarize themselves with the system and the programming method by moving the robot arm into various positions and orientations, operating the gripper, and programming the robot states. Afterward, there were two programming tasks with increasing levels



of difficulty. We used self-developed “target objects”, which were designed to be easy to pick up and place down by employing self-correcting geometry like big chamfers (see Figure 4).

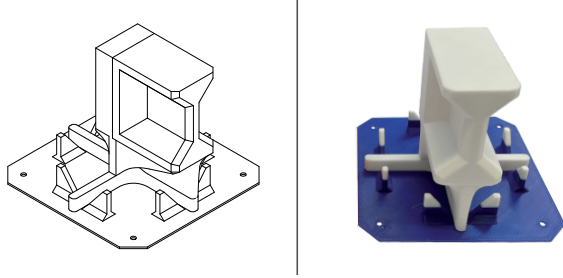


Fig. 4. Target objects that were used for the two study tasks due to their self-correcting geometry.

**Task 1:** The first task (pick-and-place) was designed to be relatively simple, involving the action of grabbing a “target object” and dropping it into a cardboard box. This task was chosen to familiarize the participants with the programming method and to test their ability to carry out movements.

**Task 2:** The second task involved swapping two “target objects”. There were three platforms, with one “target object” on both the left and right platforms. Participants could choose which “target object” to temporarily place on the third empty platform in the middle to complete the task. This procedure was more complex, as it required three grabbing and placing operations and, therefore, more programming steps. The collision risk was also higher.

After completing the two tasks, participants completed a questionnaire that assessed their experience with the programming method. This procedure was repeated for each programming method. Upon completing the third programming method, participants completed an additional questionnaire that focused on their overall experience of all conditions. To avoid “order effects” that might influence participants’ performance, we counterbalanced the order of the three conditions according to the Latin Square. A live stream was used to monitor the participants’ perspectives during the study, providing real-time feedback and assistance in case of problems. With a time limit of 5 minutes for each task, the study took approximately 60 minutes.

### C. Participants

The study was conducted with 30 participants (25 male, 5 female), aged between 19 and 38 years ( $M = 25.17, SD = 4.29$ ). All were academics, most from a technical field. On a scale ranging from 1 = “strongly disagree” to 5 = “strongly agree”, they rated themselves as experienced in robot programming ( $M = 2.63, SD = 1.73$ ) and MR ( $M = 2.93, SD = 1.70$ ).

## V. RESULTS

For parametric statistical data analysis ( $\alpha = 0.05$ ), one-way repeated measures analysis of variance (ANOVA) and post-hoc paired t-tests with Bonferroni-Holm correction were

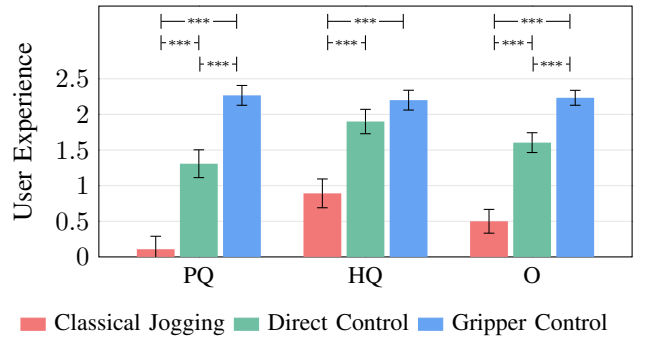


Fig. 5. The means of the UEQ ratings for each condition. Scales are (PQ) Pragmatic Quality, (HQ) Hedonic Quality, and (O) Overall, ranging from -3 to +3. Error bars represent the standard error.

used. Degrees of freedom were corrected in cases where Mauchly’s Test indicated that the sphericity assumption was violated. Effect sizes were reported using  $\eta_p^2$  (Partial Eta Squared): small ( $> .01$ ), medium ( $> .06$ ), and large ( $> .14$ ).

### A. Task Completion Time

The statistical analysis of the duration (in seconds) required by the participants to complete the two tasks in the three programming methods is as follows.

**Duration task 1:** The Greenhouse-Geisser ( $\epsilon = .54$ ) correction was used since Mauchly’s Test indicated a violation of the sphericity ( $\chi^2(2) = 53.612, p < .001$ ) assumption. This resulted in  $F(1.08, 31.307) = 54.477, p < .001, \eta_p^2 = .653$ . The post-hoc analysis showed that participants completed the first task significantly faster in the conditions *Gripper Control* ( $M = 22.84, SD = 10.33, p < .001$ ) and *Direct Control* ( $M = 27.23, SD = 17.64, p < .001$ ) compared to *Classical Jogging* ( $M = 122.53, SD = 74.84$ ).

**Duration task 2:** For the duration of the more complex task, a Greenhouse-Geisser ( $\epsilon = .70$ ) correction was also applied due to a sphericity ( $\chi^2(2) = 15.68, p < .001$ ) violation showing a significant effect  $F(1.4, 40.594) = 52.615, p < .001, \eta_p^2 = .645$ . The *Gripper Control* ( $M = 74.12, SD = 28.49$ ) and *Direct Control* ( $M = 109.50, SD = 57.00$ ) conditions were significantly ( $p < .001$ ) faster than *Classical Jogging* ( $M = 228.97, SD = 102.52$ ). The Methods *Gripper Control* and *Direct Control* duration also differed significantly ( $p = .029$ ).

### B. User Experience

Participants rated the eight items of the UEQ-S questionnaire using a 7-point Likert scale, ranging from -3 = “horribly bad” to 3 = “extremely good”. Four of these items represent the pragmatic quality, and the other four are the hedonic quality (see Figure 5).

**Pragmatic quality** showed a significant effect  $F(2, 58) = 36.971, p < .001, \eta_p^2 = .560$ . It involves factors such as perspicuity, efficiency, and dependability, so the interaction quality was significantly ( $p < .001$ ) higher in the *Gripper Control* ( $M = 2.27, SD = 0.76$ ) and *Direct Control* ( $M = 1.31, SD = 1.07$ ) conditions compared to the *Classical*

*Jogging* ( $M = 0.11, SD = 1.0$ ) conditions. The *Gripper Control* was also significantly ( $p < .001$ ) better than the *Direct Control* condition.

**Hedonic quality** also showed a significant main effect  $F(2, 58) = 34.023, p < .001, \eta_p^2 = .540$ . This quality includes stimulation and novelty, so the pleasure or fun of the product was significantly ( $p < .001$ ) better rated for the *Gripper Control* ( $M = 2.20, SD = 0.76$ ) and *Direct Control* ( $M = 1.90, SD = 0.93$ ) conditions compared to the *Classical Jogging* ( $M = 0.89, SD = 1.11$ ).

**Overall** mean scores showed a clear difference in the statistical analysis  $F(2, 58) = 48.857, p < .001, \eta_p^2 = .628$ . The *Gripper Control* ( $M = 2.23, SD = 0.58$ ) condition received significantly ( $p < .001$ ) better ratings than the *Direct Control* ( $M = 1.60, SD = 0.76$ ), and *Classical Jogging* ( $M = 0.50, SD = 0.91$ ). *Direct Control* was also significantly ( $p < .001$ ) better than *Classical Jogging*.

### C. Workload

Perceived workload ratings (where lower ratings indicate a positive assessment and higher ratings a negative one) are presented in Figure 6 and statistically analyzed as follows. Excluding the NASA-TLX items, **Physical demand**, **Temporal demand**, and **Performance**, the repeated measures ANOVA showed an effect on the remaining items.

**Mental demand**  $F(2, 58) = 37.826, p < .001, \eta_p^2 = .566$  was significantly the lowest for condition *Gripper Control* ( $M = 26.50, SD = 20.48$ ) compared to *Direct Control* ( $M = 38.67, SD = 23.56, p = .001$ ) and *Classical Jogging* ( $M = 57.83, SD = 24.02, p < .001$ ). An effect ( $p < .001$ ) was also between *Direct Control* and *Classical Jogging*.

**Effort** Mauchly's Test indicated that the assumption of sphericity had been violated ( $\chi^2(2) = 7.796, p = .02$ ), so degrees of freedom are corrected using Huyn-Feldt estimation of sphericity ( $\epsilon = .845$ ). Results showed a significant effect  $F(1.689, 48.987) = 29.844, p < .001, \eta_p^2 = .507$ . Compared to method *Classical Jogging* ( $M = 55.17, SD = 23.43$ ), participants had to work significantly less with methods *Gripper Control* ( $M = 25.83, SD = 22.13$ ) and *Direct Control* ( $M = 31.17, SD = 23.22$ ) to accomplish their level of performance.

**Frustration**  $F(2, 58) = 7.416, p = .001, \eta_p^2 = .204$  was significantly lower with the methods *Gripper Control* ( $M = 17.17, SD = 24.02, p = .002$ ) and *Direct Control* ( $M = 19.33, SD = 15.19, p = .006$ ) compared to *Classical Jogging* ( $M = 34.33, SD = 24.49$ ).

**Overall** mean scores of all six NASA-TLX items showed a clear difference in the statistical analysis  $F(2, 58) = 17.947, p < .001, \eta_p^2 = .382$ , as the workload for the *Gripper Control* ( $M = 25.69, SD = 15.62$ ) and *Direct Control* ( $M = 29.56, SD = 13.63$ ) methods was significantly ( $p < .001$ ) lower than *Classical Jogging* ( $M = 38.94, SD = 15.55$ ).

### D. Specific Questions

The statistical analysis of the specific questions, which were rated from 1 = "strongly disagree" to 7 = "strongly agree", showed significant effects.

**Q1:** "I was able to fulfill the tasks without any problems."  $F(2, 58) = 10.140, p < .001, \eta_p^2 = .259$ . The *Gripper Control* ( $M = 6.27, SD = 0.83$ ) condition significantly achieved the best results compared to *Direct Control* ( $M = 5.07, SD = 1.44, p < .001$ ) and *Classical Jogging* ( $M = 5.13, SD = 1.14, p < .001$ ).

**Q2:** "The programming technique felt natural to use."  $F(2, 58) = 28.498, p < .001, \eta_p^2 = .496$ . Conditions *Gripper Control* ( $M = 6.20, SD = 1.06, p < .001$ ) and *Direct Control* ( $M = 5.57, SD = 1.22, p < .001$ ) were rated significantly better than *Classical Jogging* ( $M = 3.73, SD = 1.51$ ).

After completing all three programming methods, the participants evaluated them using the following questions.

**Q3:** "I was able to program the robot precisely." since Mauchly's Test indicated a violated assumption of sphericity ( $\chi^2(2) = 13.033, p = .001$ ), the degrees of freedom are corrected using Huyn-Feldt ( $\epsilon = .757$ ) resulting in  $F(1.515, 43.935) = 11.574, p < .001, \eta_p^2 = .285$ . The post-hoc test indicated that *Gripper Control* ( $M = 6.37, SD = 0.77$ ) was significantly better than *Direct Control* ( $M = 5.03, SD = 1.22, p < .001$ ) and *Classical Jogging* ( $M = 5.13, SD = 1.43, p < .001$ ).

**Q4:** "Overall, I would rate the method.",  $F(2, 58) = 21.253, p < .001, \eta_p^2 = .423$ . The ratings of the condition *Gripper Control* ( $M = 6.27, SD = 0.98$ ) were significantly higher compared to *Direct Control* ( $M = 5.40, SD = 1.13, p = .010$ ) and *Classical Jogging* ( $M = 4.17, SD = 1.58, p < .001$ ). Also, the conditions *Direct Control* and *Classical Jogging* revealed an effect ( $p < .001$ ).

### E. Task Performance

**Performance task 1:** The problem-free completion of task 1 was most frequently achieved with condition *Gripper Control* (70.0%), followed by *Direct Control* (46.7%) and method *Classical Jogging* (26.7%). As illustrated in Figure 7, there is a significant difference in the number of errors made by the participants  $F(2, 58) = 6.911, p = .002, \eta_p^2 = .192$ . *Gripper Control* ( $M = 0.33, SD = 0.55$ ) had significantly less errors compared to *Classical Jogging* ( $M = 1.13, SD = 1.01$ ). *Direct Control* ( $M = 0.70, SD = 0.75$ ) revealed no statistical difference.

**Performance task 2:** With condition *Gripper Control* (23.3%), task 2 was performed most often without any problems, followed by *Direct Control* (6.7%) and *Classical Jogging* (3.3%). The number of programming errors showed a statistical effect  $F(2, 58) = 30.910, p < .001, \eta_p^2 = .516$ . Compared to *Direct Control* ( $M = 3.83, SD = 2.29, p < .001$ ) and *Classical Jogging* ( $M = 4.33, SD = 2.23, p < .001$ ), *Gripper Control* ( $M = 1.53, SD = 1.57$ ) showed a significantly lower number of errors.

## VI. DISCUSSION

The user study results are consistent with those of earlier works, in which the conventional method performed worst [8,24,35]. It should be noted that the *Classical Jogging* method, which serves as a baseline, has certain advantages

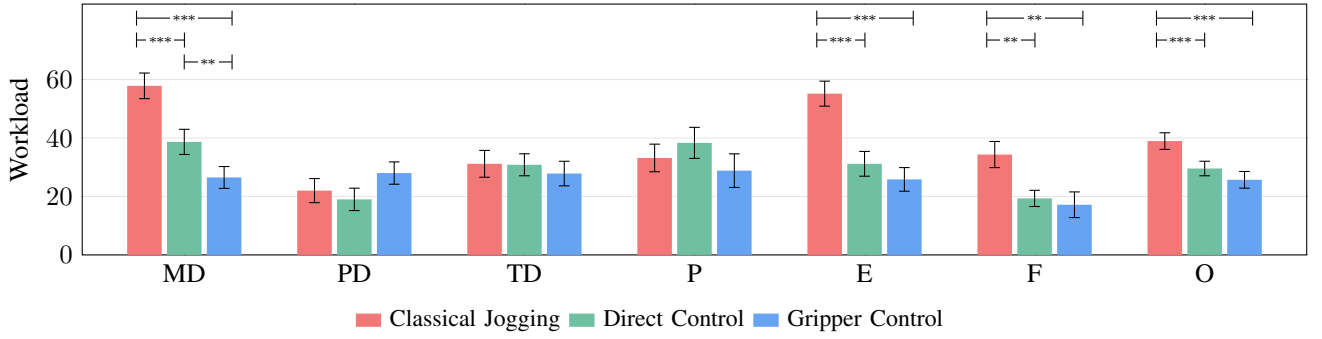


Fig. 6. Mean scores of the NASA-RTLX ratings ranging from 0 [very low] to 100 [very high]. The scales are (MD) Mental demand, (PD) Physical demand, (TD) Temporal demand, (P) Performance, (E) Effort, (F) Frustration, and (O) Overall. Error bars indicate the standard error.

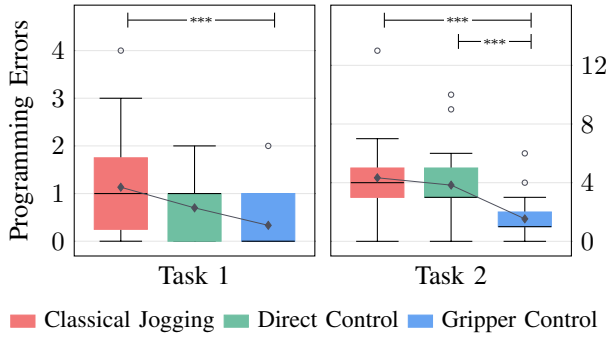


Fig. 7. Errors that occurred during the two robot programming tasks. The connected diamonds represent the mean values, and the dots are the outliers.

(e.g., augmented representation, path planning, trajectory simulation) compared to traditional methods such as the teach pendant. In spite of this, setting the position and orientation using separated axes proved to take much longer than 6-DOF interactions, as multiple mode changes and perspective changes were needed to define a robot pose.

In contrast, the *Direct Control* method performs significantly better due to its controller-based 6-DOF interaction, as shown by the results of the UEQ and NASA RTLX questionnaires and the recorded programming errors. Since this method differs from the *Classical Jogging* method only in its interaction technique, the results can only be attributed to the fact that moving the controller to the desired position requires less effort than moving it with the joysticks, which is supported by more natural usability (Q2). Compared to the existing hand-tracking-based MR approaches, the controller-based approach is more suitable due to its haptics and better tracking accuracy [32].

Overall, the results strongly support the use of the *Gripper Control* method. It is the most efficient since its completion time of the more complex and, therefore, more meaningful task 2 was significantly the fastest. The overall results of the UEQ questionnaire are also better and the mental effort is significantly the lowest. This method is also the most efficient, as evidenced by the lowest number of programming errors in the relevant task 2 and the results of the specific questions Q1 and Q3.

Our study has some limitations. As mentioned by Yang *et al.* [35] regarding rendering quality, the imperfect occlusion between augmented and physical objects can lead to errors in depth perception, e.g., rendering virtual objects completely when parts of them are already inside or behind a physical object. In addition, using the gripper and controller buttons simultaneously can be challenging for people with small hands. The manual calibration is imperfect and can lead to misalignment between the digital twin and the real robot, which occurs when the user briefly removes the HMD from the head. This could be improved by a computer vision-based calibration, e.g., object or marker recognition. However, it is currently not possible to access the real-time pass-through video of the Quest 3.

A promising direction for future work would involve improving the *Gripper Controller*, which holds potential for enhancements in both the construction (e.g., increased stiffness, reduced mechanical friction) and ergonomics (e.g., a longer trigger for more gripping force, or the use of a push interaction instead of a trigger). Additionally, the controller extension can be designed for the attachment of different robot arm extensions.

## VII. CONCLUSIONS

This article presents three controller-based methods for MR robot programming. These are 1) *Classical Jogging*, where the virtually augmented end effector is moved using the thumbsticks of two controllers, 2) *Direct Control*, in which the position of the end effector directly matches that of the controller, and 3) *Gripper Control*, which builds upon *Direct Control* by extending the controller with an interactive 3D-printed gripper, allowing intuitive programming by directly gripping, moving, and placing physical objects. A within-subjects study ( $n = 30$ ) was conducted to compare these methods. Participants completed two tasks for each method. The results demonstrate that the *Gripper Control* condition is the fastest, has the best user experience, the lowest mental demand, and the best subjective and objective task performance. Therefore, the *Gripper Control* method has the potential to significantly improve future robot programming.

## REFERENCES

- [1] J. Aleotti, S. Caselli, and M. Reggiani. Leveraging on a virtual environment for robot programming by demonstration. *Robotics and Autonomous Systems*, 47(2):153–161, 2004. Robot Learning from Demonstration.
- [2] Lal Lila Bozgeyikli and Evren Bozgeyikli. Tangiball: Foot-enabled embodied tangible interaction with a ball in virtual reality. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 812–820, 2022.
- [3] Sébastien Cuendet, Engin Bumbacher, and Pierre Dillenbourg. Tangible vs. virtual representations: when tangibles benefit the training of spatial skills. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, NordiCHI '12*, page 99–108, New York, NY, USA, 2012. ACM.
- [4] Xiwen Dengxiong, Xueting Wang, Shi Bai, and Yunbo Zhang. Self-supervised 6-dof robot grasping by demonstration via augmented reality teleoperation system. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7819–7826, 2024.
- [5] Hongchao Fang, Soh Khim Ong, and Andrew Yeh-Ching Nee. Robot programming using augmented reality. In *2009 International Conference on CyberWorlds*, pages 13–20, 2009.
- [6] Federica Ferraguti, Mattia Bertuletti, Mattia Gambazza, and Matteo Ragaglia. High-velocity walk-through programming for industrial applications. *Robotics and Computer-Integrated Manufacturing*, 81:102505, 2023.
- [7] Felipe Fronchetti, Miles Popiela, Rodrigo Spinola, and Shawn Brixey. Roboart: Artistic robot programming in mixed reality. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 1192–1193, 2024.
- [8] Samir Yitzhak Gadre, Eric Rosen, Gary Chien, Elizabeth Phillips, Stefanie Tellex, and George Konidaris. End-user robot programming using mixed reality. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2707–2713, 2019.
- [9] Elisa Gusai, Chiara Bassano, Fabio Solari, and Manuela Chessa. Interaction in an immersive collaborative virtual reality environment: A comparison between leap motion and htc controllers. In Sebastiano Battiato, Giovanni Maria Farinella, Marco Leo, and Giovanni Gallo, editors, *New Trends in Image Analysis and Processing – ICIAP 2017*, pages 290–300, Cham, 2017. Springer International Publishing.
- [10] Sandra G. Hart. Nasa-task load index (nasa-tlx); 20 years later. volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.
- [11] Jan Krieglstein, Gesche Held, Balázs A. Bálint, Frank Nägele, and Werner Kraus. Skill-based robot programming in mixed reality with ad-hoc validation using a force-enabled digital twin. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11612–11618, 2023.
- [12] Sing Chun Lee, Bernhard Fuerst, Keisuke Tateno, Alex Johnson, Javad Fotouhi, Greg Osgood, Federico Tombari, and Nassir Navab. Multi-modal imaging, model-based tracking, and mixed reality visualisation for orthopaedic surgery. *Healthcare Technology Letters*, 4(5):168–173, 2017.
- [13] Matthew B. Luebbers, Connor Brooks, Carl L. Mueller, Daniel Szafr, and Bradley Hayes. Arc-lfd: Using augmented reality for interactive long-term robot skill maintenance via constrained learning from demonstration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3794–3800, 2021.
- [14] Simone Macciò, Alessandro Carfi, and Fulvio Mastrogiovanni. Mixed reality as communication medium for human-robot collaboration. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2796–2802, 2022.
- [15] Alexander Masurovsky, Paul Chojeci, Detlef Runde, Mustafa Lafci, David Przewozny, and Michael Gaebler. Controller-free hand tracking for grab-and-place tasks in immersive virtual reality: Design elements and their empirical study. *Multimodal Technologies and Interaction*, 4(4), 2020.
- [16] D. Mendes, F. M. Caputo, A. Giachetti, A. Ferreira, and J. Jorge. A survey on 3d virtual object manipulation: From the desktop to immersive virtual environments. *Computer Graphics Forum*, 38, 2019.
- [17] Daniel Mendes, Filipe Relvas, Alfredo Ferreira, and Joaquim Jorge. The benefits of dof separation in mid-air 3d object manipulation. *VRST '16*, page 261–268, New York, NY, USA, 2016. ACM.
- [18] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. In Hari Das, editor, *Telem manipulator and Telepresence Technologies*, volume 2351, pages 282 – 292. International Society for Optics and Photonics, SPIE, 1995.
- [19] Emma C Morley and Chanan S Syan. Teach pendants: how are they for you? *Industrial Robot: An International Journal*, 22(4):18–22, 1995.
- [20] Pedro Neto and Nuno Mendes. Direct off-line robot programming via a common cad package. *Robotics and Autonomous Systems*, 61(8):896–910, 2013.
- [21] Pedro Neto, J. Norberto Pires, and A. Paulo Moreira. Cad-based off-line robot programming. In *2010 IEEE Conference on Robotics, Automation and Mechatronics*, pages 516–521, 2010.
- [22] Mikhail Ostanin, Stanislav Mikhel, Alexey Evlampiev, Valeria Skvortsova, and Alexandr Klimchik. Human-robot interaction for robotic manipulator programming in mixed reality. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2805–2811, 2020.
- [23] Simone Luca Pizzagalli, Yevhen Bondarenko, Vladimir Kuts, Eoin O’Connell, Niall Murray, and Tauno Otto. Ros-based augmented and virtual reality path planning interface for industrial robotic arms: a preliminary assessment. In *Proceedings of the 2023 6th International Conference on Robot Systems and Applications, ICRSA '23*, page 38–48, New York, NY, USA, 2024. ACM.
- [24] Camilo Perez Quintero, Sarah Li, Matthew KXJ Pan, Wesley P. Chan, H.F. Machiel Van der Loos, and Elizabeth Croft. Robot programming through augmented trajectories in augmented reality. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1838–1844, 2018.
- [25] Anu Rastogi, Paul Milgram, David Drascic, and Julius J. Grodski. Telerobotic control with stereoscopic augmented reality. In *Stereoscopic Displays and Virtual Reality Systems III*, volume 2653, pages 115 – 122. International Society for Optics and Photonics, SPIE, 1996.
- [26] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(Volume 3, 2020):297–330, 2020.
- [27] Maximilian Rettinger and Gerhard Rigoll. Defuse the training of risky tasks: Collaborative training in xr. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2022.
- [28] Maximilian Rettinger and Gerhard Rigoll. Touching the future of training: investigating tangible interaction in virtual reality. *Frontiers in Virtual Reality*, 4, 2023.
- [29] Maximilian Rettinger, Julia Steinhilber, Annika Hackenberg, Lisa Lehr, Niklas Müller, Matthias Schöffel, Sonja Pandit, Julia Mayer, Christopher Holzmann-Littig, Gerhard Rigoll, and Christoph Schmäderer. Potential of digital technologies in counteracting long-standing deficits in hemodialysis machine training. *Scientific Reports*, 15(1), 2025.
- [30] Maximilian Rettinger, Julia Steinhilber, Annika Hackenberg, Lisa Lehr, Philip Neugebauer, Niklas Müller, Gerhard Rigoll, and Christoph Schmäderer. Optimizing medical device training: The role of multi-user vr and expert guidance. In *Conference on Computer-Supported Cooperative Work and Social Computing, CSCW Comp. ACM*, 2024.
- [31] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. Design and evaluation of a short version of the user experience questionnaire (ueq-s). *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, 4:103, 01 2017.
- [32] Inês Soares, Ricardo B. Sousa, Marcelo Petry, and António Paulo Moreira. Accuracy and repeatability tests on hololens 2 and htc vive. *Multimodal Technologies and Interaction*, 5(8), 2021.
- [33] Peng Wang, Xiaoliang Bai, Mark Billingham, Shusheng Zhang, Dechuan Han, Mengmeng Sun, Zhuo Wang, Hao Lv, and Shu Han. Haptic feedback helps me? a vr-sar remote collaborative system with tangible interaction. *International Journal of Human-Computer Interaction*, 36(13):1242–1257, 2020.
- [34] Wenhao Yang, Qinqin Xiao, and Yunbo Zhang. An Augmented-Reality Based Human-Robot Interface for Robotics Programming in the Complex Environment. Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability:V002T07A003, 06 2021.
- [35] Wenhao Yang, Qinqin Xiao, and Yunbo Zhang. HaR<sup>2</sup> bot: a human-centered augmented reality robot programming method with the awareness of cognitive load. *Journal of Intelligent Manufacturing*, 35(5):1985–2003, 2024.
- [36] Chen Zheng, Jiajian Xing, Zhanxi Wang, Xiansheng Qin, Benoît Eyraud, Jing Li, Jing Bai, and Yicha Zhang. Knowledge-based program generation approach for robotic manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 73:102242, 2022.