2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Mixed Reality-Based, Immersive, Semi-Autonomous Robotic Telemanipulation for the Execution of Peg-In-Hole Tasks

Shifei Duan, Francesco De Pace, Zhe Wang and Minas Liarokapis

***Abstract*— Semi-autonomy in telemanipulation frameworks has the potential to reduce user cognitive load while preserving human perceptual oversight and decision-making capabilities. However, existing semi-autonomous telemanipulation systems are heavily dependent on calibration and hardware configurations, making rapid deployment difficult. Moreover, existing VR-based telemanipulation systems lack intuitive interaction mechanisms, requiring users to manage complex control interfaces. To address these limitations, we introduce an intuitive and immersive semi-autonomous robotic telemanipulation system that leverages a mixed reality (MR) headset with minimal hardware requirements. Requiring only CPU processing and coarse calibration procedures, the system combines human perception with autonomous control strategies through natural hand tracking and finger gestures to achieve precise, reliable task execution. To validate this approach, we conducted thorough evaluations involving complex peg-in-hole tasks and compared performance with and without the proposed control strategy. The results highlight that our system demonstrates robust performance, and the proposed control strategy further enhances its stability and effectiveness.**

I. Introduction

As one of the most significant applications in modern manufacturing, autonomous robotic assembly has attracted considerable research attention due to its potential to provide efficient and reliable solutions for complex tasks. Peg-in-hole tasks, such as inserting bolts into slots and fitting parts together, represent challenging benchmarks for robotic assembly operations. While autonomous solutions typically rely on object pose estimation algorithms to determine the position and orientation of components, achieving high accuracy requires advanced hardware, complex algorithms, extensive training data, and precise calibration. In contrast, humans excel at performing such tasks through natural dexterity and spatial awareness. Therefore, robot telemanipulation emerges as a key technology that not only keeps human operators safe from hazardous environments but also leverages human perception in robot operations. However, telemanipulation remains challenging when high dexterity is required, placing a significant cognitive workload on users. Thus, semi-autonomous telemanipulation systems represent a promising approach to address these challenges by combining human perception and dexterity with robot capability for repeatable and reliable task execution. Among state-of-the-art telemanipulation interfaces, traditional approaches typically rely on RGB video feeds, which provide visual feedback but lack crucial depth information [1]–[3]. This limitation often impedes precise positioning and orientation control of the robotic end-effector (EE). To address these perceptual challenges, Virtual Reality (VR) offers a compelling solution by creating immersive three-dimensional representations of the robot's workspace [4]. By enhancing spatial awareness and environmental perception, VR interfaces enable operators to execute tasks with improved accuracy and efficiency [5], leading to increased adoption of advanced VR technologies in telemanipulation research. Although various VR devices for robotic telemanipulation have been extensively evaluated in previous research [4], the recent release of advanced Mixed Reality (MR) platforms, such as the Apple Vision Pro, has sparked renewed interest in robotic telemanipulation applications. Researchers are leveraging its high-resolution displays, multiple sensors, and powerful processing capabilities to deliver immersive and interactive experiences for robotic operators [6], [7]. Despite these technological advances, telemanipulation systems remain fundamentally dependent on continuous human intervention, requiring operators to put

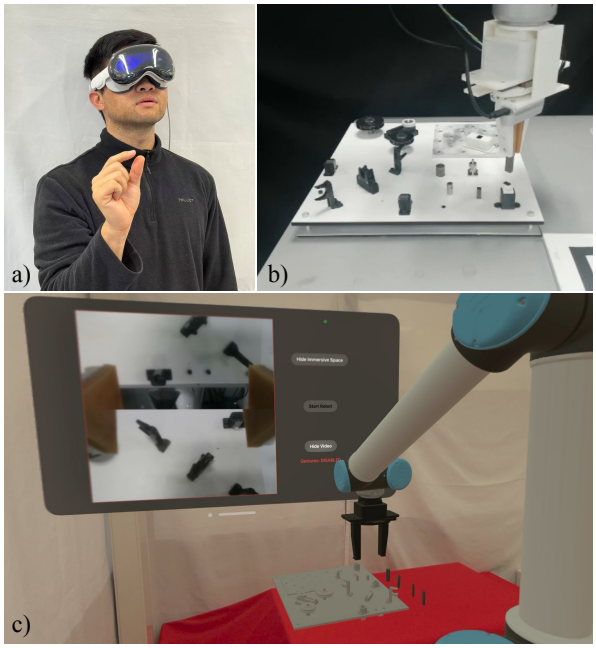

Fig. 1. Subfig. a) presents the user wearing the mixed reality headset for the execution of a peg-in-hole task with the proposed telemanipulation system. Subfig. b) presents the real robot system executing the assembly task. Subfig. c) presents the mixed reality environment in the headset.

Shifei Duan and Zhe Wang are with the New Dexterity research group, The University of Auckland, New Zealand {sdua078, zwan341}@aucklanduni.ac.nz

Francesco De Pace is with the Competence Industry Manufacturing 4.0 (CIM4.0), Turin, Italy francesco.de_pace@cim40.com

Minas Liarokapis is with the New Dexterity research group, The University of Auckland, New Zealand and the National Technical University of Athens, Greece liarokapis@mail.ntua.gr

considerable physical effort and maintain sustained concentration throughout the process. To address these challenges, human-robot collaboration has gained significant attention in telemanipulation research, focusing on alleviating user cognitive load while enhancing operational efficiency. Multiple studies [8]–[11] have corroborated the effectiveness and versatility of semi-autonomous telemanipulation approaches in handling various complex manipulation tasks, with peg-in-hole insertion being particularly representative of these challenges.

Extensive research has focused on advanced vision-based methods to address peg and hole localization challenges. Vision-based approaches can be broadly categorized into model-based and learning-based methods. Model-based methods have been developed to estimate the pose of novel objects, as demonstrated by [12] and [13]. While model-based methods provide reliable pose estimation, they are constrained by their dependency on prior templates.

Consequently, with the advancement of deep learning in computer vision, learning-based methods have also gained prominence [14]–[17]. However, most existing approaches have limitations: they struggle to handle scenarios with multiple holes present, and positional uncertainty often remains a significant challenge that needs to be addressed. Recent advances in reinforcement learning (RL) have introduced new possibilities for the execution of peg-in-hole tasks. RL-based policies can directly integrate multi-modal sensor data and generate control commands [18]. Nevertheless, these methods typically require substantial amounts of training data. To address this limitation, [19] proposed a 6D pose estimation approach for target holes using less annotated images to achieve accurate socket pose estimation. Despite achieving acceptable success rates, these methods still face common challenges: they require considerable training data, precise calibration procedures, and high-accuracy sensors to guarantee reliable performance.

Therefore, this work presents a mixed reality-based immersive semi-autonomous robotic telemanipulation system for precise control of robotic manipulators. The proposed system addresses existing limitations by combining human decision-making with robotic capabilities through adaptive control strategies that reduce user cognitive load while enhancing task execution efficiency. Our approach eliminates the requirement for sustained human concentration in telemanipulation while providing the decision-making capabilities that autonomous systems lack. This approach also enables reliable performance in the execution of complex peg-in-hole tasks while maintaining low-cost implementation without requiring extensive training data, high-performance computing hardware, or additional sensor configurations.

II. Problem Description

The peg-in-hole problem studied in this paper, we formulate it as a grasp-and-place task. The conventional solution utilizes cameras to detect the poses of both the peg and hole, and then plans a trajectory to autonomously complete the task. However, this approach requires highly accurate calibration of both the camera and robot, which requires extensive setup time in practical applications. Moreover, pose estimation relies heavily on computational resources and trained models, making it computationally expensive. Additionally, the grasping procedure introduces inevitable positional and orientational variations to the object as the gripper makes physical contact and secures its hold. These variations are generally unpredictable and challenging to quantify using vision-based systems. Such uncertainties can lead to misalignment in the execution of peg-in-hole tasks, resulting in failed insertions.

To address these challenges, we propose an approach based on mimicking human behavior by bringing humans into the loop for solving peg-in-hole insertion tasks. When humans attempt to insert a peg into a hole, they continuously adjust the peg's pose as they align it with the hole and move it downward toward the target. Inspired by this observation, we identify the insertion process as the critical stage and separate the overall task into three components: detection, alignment estimation, and insertion strategy. The detection component identifies the shapes and edges of the peg and hole. The alignment estimation component estimates the relative positioning error between the peg and hole using camera pixel coordinates. Finally, the insertion strategy component executes the insertion task through a semi-autonomous approach that combines autonomous robotic insertion with telemanipulated adjustments. A MR device provides immersive visualization and an intuitive telemanipulation interface, allowing users to easily comprehend the task requirements and make real-time adjustments to the robot's EE pose as needed.

III. Methodology

A. Hardware and Framework

The hardware setup comprises a 6-DoF serial manipulator (UR10) from Universal Robots equipped with a parallel jaw gripper and two RealSense D435i RGB cameras mounted on the robot EE. The manipulator and cameras are connected to an Ubuntu 20.04 PC running the Robot Operating System (ROS) [20], which manages the ROS node architecture responsible for trajectory planning and RosBridge communication [21]. The control interface utilizes an Apple Vision Pro (AVP) MR system that connects to the same network via Wi-Fi. The MR application is developed in visionOS and programmed in Swift, with communication between the AVP and ROS environment occurring through WebSocket protocols, as illustrated in Fig. 2.

B. Robot telemanipulation

The MR user interface integrates three core components: (i) a Digital Twin (DT) representation of the robot system, constructed from the system's URDF specifications, (ii) the virtual models of the task workspace including the peg-in-hole assembly board and components, and (iii) real-time video feedback for direct observation of the physical robot EE, which is shown in Fig. 1. When the user gazes at the video feed interface as detected by AVP sensors, they can control the robot EE through specific hand gestures:

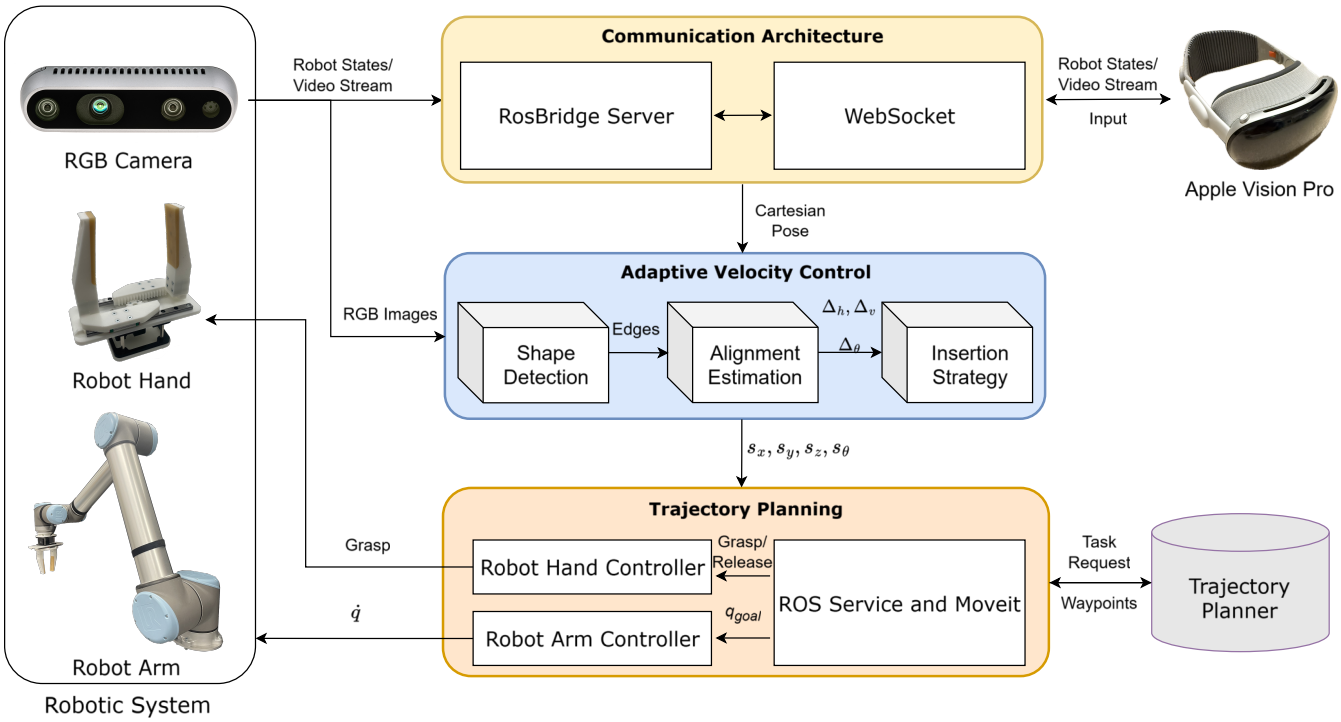

Fig. 2. The framework architecture of MR-based robotic telemanipulation with an adaptive velocity control strategy.

pinch-and-drag motions provide translational control, while simultaneous pinching with both hands followed by rotational hand movements controls the EE's orientation. The spatial displacement vectors from user hand movements are transformed and mapped in real-time from the AVP coordinate frame to the robot EE motion commands. During the final insertion phase, z-axis motion is constrained to ensure controlled descent, requiring the operator to manage only XY-plane positioning and z-axis rotation for overall alignment. This telemanipulation framework is implemented using the Universal Robots UR modern driver [22], ROS Control packages [23], and Cartesian controllers [24].

C. Peg and Hole Detection

The system utilizes a hybrid computer vision approach implemented in OpenCV [25]. For circular peg detection, it combines the Hough circle transform with contour-based ellipse fitting. For rectangular peg identification, it uses the probabilistic Hough line transform and morphological edge detection. The dual-camera setup employs Region-of-Interest (ROI) segmentation with camera-specific parameters: the upper camera detects the horizontal tangent line of the peg's upper arc, and the lower camera detects the lower arc, while both cameras detect the vertical tangent line of the peg's side. To ensure smooth and robust real-time performance, an adaptive state machine implements temporal consistency filtering through a 10-frame moving average window, supported by multi-threaded parallel processing and pre-allocated memory buffers. Hole detection uses the same approach as peg detection but with a different ROI corresponding to the hole's location in the image frame.

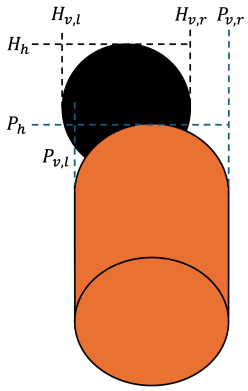

Fig. 3. Tangent line detection for the peg and the hole from single camera.

D. Peg-to-Hole Alignment Estimation

To align the peg with its corresponding hole, the system first localizes the detected tangent lines within the image frame using pixel coordinates. Specifically, vertical tangent lines are identified by their y-coordinates, and horizontal tangent lines by their x-coordinates. The alignment error, or variation, is then calculated using a cross-camera tangent line comparison method. The algorithm computes the horizontal variation by comparing the peg's tangent position, $P_h^{(i)}$ from camera i, with the hole's tangent position, $H_h^{(i)}$ from the other camera, leveraging the stereo vision setup to establish correspondence. Figure 3 shows the tangent line detection diagram for the peg and the hole from one of the cameras,

TABLE I

KEY PARAMETERS OF THE ADAPTIVE VELOCITY CONTROL (AVC) STRATEGY

Parameter	Value	Description
s	0.1	Velocity reduction factor during insertion
$t_{\text{threshold}}$	2.0 s	Time before restoring nominal z-velocity
λ	0.1	Exponential scaling parameter

and the horizontal variation is calculated as follows:

$$\Delta_h = \left| |P_h^{(0)} - H_h^{(1)}| - |P_h^{(1)} - H_h^{(0)}| \right| \tag{1}$$

For vertical alignment, the left and right vertical tangent boundaries are processed separately. The system calculates the disparities between the corresponding vertical tangents of the peg (left $P_{v,l}^{(i)}$ and right $P_{v,r}^{(i)}$) and the hole (left $H_{v,l}^{(i)}$ and right $H_{v,r}^{(i)}$) across both cameras. The minimum value is selected to determine the best match:

$$\begin{aligned} \delta_{v,l} &= \min\{|P_{v,l}^{(0)} - H_{v,l}^{(1)}|,\ |P_{v,l}^{(1)} - H_{v,l}^{(0)}|\}, \\ \delta_{v,r} &= \min\{|P_{v,r}^{(0)} - H_{v,r}^{(1)}|,\ |P_{v,r}^{(1)} - H_{v,r}^{(0)}|\}. \end{aligned} \tag{2}$$

The final vertical variation is then calculated using a cross-camera side-line comparison:

$$\Delta_v = |\delta_{v,l} - \delta_{v,r}| \tag{3}$$

For non-circular peg and hole orientation alignment, the angular variation between the peg tangent line and the hole tangent line is calculated as follows:

$$\Delta_\theta = \arccos(|\cos(\theta_p - \theta_h)|) \tag{4}$$

This approach provides a symmetric quantification of the alignment error, where smaller horizontal, vertical, and angular disparities correspond to better peg-to-hole alignment.

E. Adaptive Velocity Control-Based Insertion Strategy

To enhance control precision, the system implements an Adaptive Velocity Control (AVC) strategy that dynamically adjusts the robot EE velocity based on both user input and visual feedback. Table I provides the key parameters of the AVC strategy. When the user provides directional input, the z-axis EE velocity is reduced by a scaling factor s_z. If no input is provided within the specified time limit ($t_{\text{threshold}} = 2.0$ s), the velocity returns to its nominal rate, allowing brief pauses without excessively slowing task execution. Concurrently, the algorithm calculates the horizontal and vertical disparities, Δ_h and Δ_v, between the peg and the hole. These disparities are used to dynamically adjust X and Y scaling factors s_x and s_y, which enable finer control in the X-Y plane as the peg approaches the hole. Additionally, the rotational velocity of the robot EE is modulated by a scaling factor s_θ to achieve precise orientation alignment. Each scaling factor is calculated independently as follows:

$$f(\Delta) = s + (1 - s) \cdot \frac{e^{\lambda\Delta} - 1}{e^{\lambda\Delta_0} - 1} \tag{5}$$

where $s = 0.1$ is the minimum scaling factor, ensuring that the robot motion never completely stops even when the variation is small. The parameter $\lambda = 0.1$ controls the rate of the exponential variation-to-velocity mapping, resulting in a gradual decrease in velocity as the peg approaches the hole center. Δ_0 is the initial variation value. By introducing Δ_v, Δ_h, and Δ_θ, the scaling factors s_x, s_y, and s_θ are computed, respectively. This approach integrates human perceptual decision-making with computer vision assistance: as the peg approaches the target hole, indicated by decreased disparities Δ_h and Δ_v, the EE velocity is automatically reduced proportionally in the corresponding direction. This velocity modulation provides the user with increased reaction time and enables finer manual adjustments during the critical final alignment phase. The algorithm is detailed in Algorithm 1.

Algorithm 1: Adaptive Velocity Control Strategy

Input: Directional input vector $\vec{\Delta} \in \mathbb{R}^n$
Output: Scaling factor $\vec{\Delta}_{\text{scaled}} \in \mathbb{R}^4$
$s_{\text{normal}} \leftarrow 1$;
$t_{\text{elapsed}} \leftarrow 0$;
$t_{\text{threshold}} \leftarrow$ time limit;
$s \leftarrow$ velocity reduction factor;
$\Delta_h \leftarrow$ horizontal variation;
$\Delta_v \leftarrow$ vertical variation;
while *system active* **do**
  Acquire $\vec{\Delta}$ from Apple Vision Pro;
  $\Delta_h, \Delta_v \leftarrow$ compute_disparities();
  $s_x \leftarrow f(\Delta_h)$;
  $s_y \leftarrow f(\Delta_v)$;
  $\vec{\Delta}_{\text{scaled},x} \leftarrow \vec{\Delta}_x \cdot s_x$;
  $\vec{\Delta}_{\text{scaled},y} \leftarrow \vec{\Delta}_y \cdot s_y$;
  if *peg is non-circular* **then**
    $\Delta_\theta \leftarrow$ angular variation;
    $s_\theta \leftarrow f(\Delta_\theta)$;
    $\vec{\Delta}_{\text{scaled},\theta} \leftarrow \vec{\Delta}_\theta \cdot s_\theta$;
  if $\|\vec{\Delta}_{\text{scaled}}\| > 0$ **then**
    $s_z \leftarrow s$;
    $t_{\text{elapsed}} \leftarrow 0$;
  else
    $t_{\text{elapsed}} \leftarrow t_{\text{elapsed}} + \Delta t$;
    if $t_{\text{elapsed}} > t_{\text{threshold}}$ **then**
      $s_z \leftarrow s_{\text{normal}}$;
  $\vec{\Delta}_{\text{scaled},z} \leftarrow \vec{\Delta}_z \cdot s_z$;
return $(\vec{\Delta}_{\text{scaled},x}, \vec{\Delta}_{\text{scaled},y}, \vec{\Delta}_{\text{scaled},z}, \vec{\Delta}_{\text{scaled},\theta})$

IV. USER STUDY

A. Experiments

This study aims to assess the effectiveness of the system and conduct a comparative analysis between two modalities: (1) Partial Insertion Strategy Mode, where the proposed insertion strategy was applied only in the z-axis, and (2)

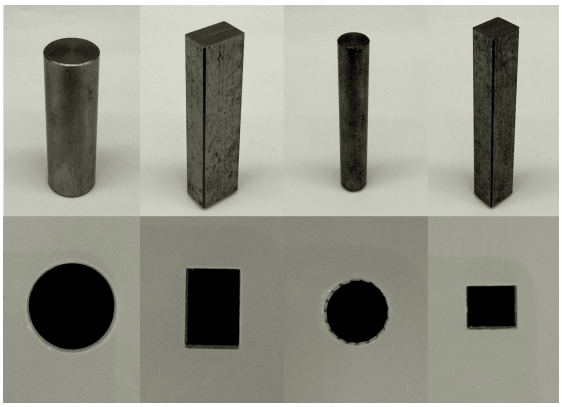

Fig. 4. Left to right: four different pegs and the corresponding holes for the peg-in-hole task: Peg1, Peg2, Peg3, and Peg4.

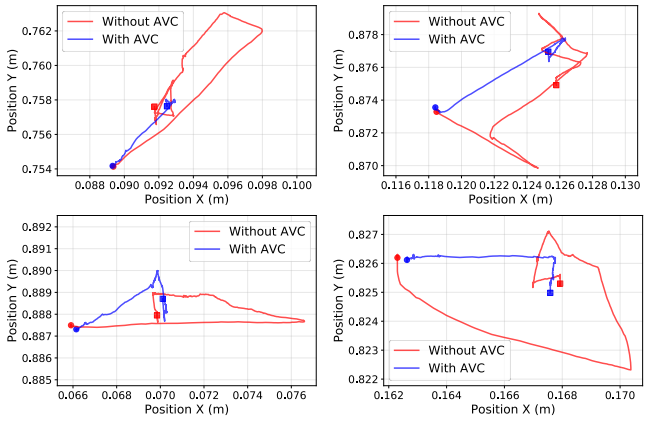

Fig. 5. Robot EE trajectories during the peg-in-hole task for four different pegs with and without AVC.

Full Insertion Strategy Mode, where the insertion strategy was applied in all three axes (x, y, and z axes). Both modalities employed the same z-axis insertion strategy to ensure consistent vertical positioning, while the full strategy mode additionally incorporated xy-plane adjustments to compensate for lateral misalignments during the insertion process. The task utilizes a task board designed by the National Institute of Standards and Technology (NIST) for the Robot Grasping and Manipulation Competitions [26], which involves four metal pegs consisting of two cylindrical pegs with different diameters and two rectangular pegs with distinct dimensions, together with their corresponding target insertion holes, as illustrated in Fig. 4. The task objective is to achieve precise insertion of all four metal pegs into their designated holes within a 0.2 mm tolerance. The pegs and task board were pre-positioned at known locations within the robot's reference frame, with corresponding virtual models accurately registered in the MR environment. However, precise calibration of individual hole and peg positions was not performed, resulting in minor positional discrepancies that needed to be compensated for during insertion.

To evaluate and compare the performance of the proposed modalities, the robot EE trajectories, task execution times,

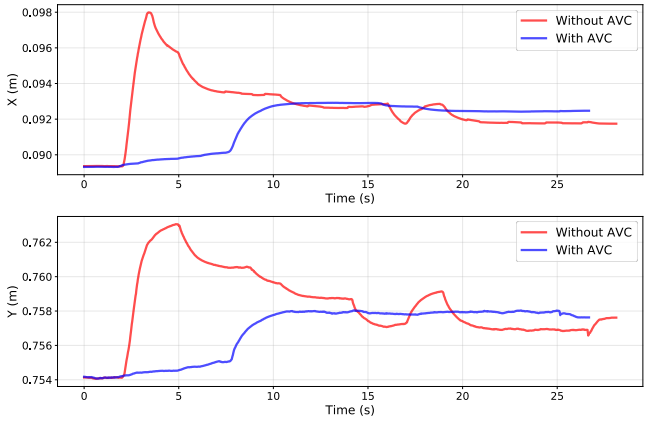

Fig. 6. Robot EE displacement in the x-axis and y-axis during the insertion task for Peg1 with and without AVC.

and failure counts were recorded. Additionally, subjective data were collected to assess: (i) general user information, (ii) simulator sickness using the Simulator Sickness Questionnaire (SSQ) [27], (iii) usability via the System Usability Scale (SUS) [28], (iv) workload through the NASA-TLX questionnaire [29], and (v) user preference using the Single Ease Question (SEQ) [30]. To ensure a fair comparison, the peg and robot initial positions remained constant throughout the experiment, as did the robot speed limitation. Before performing the actual task with each modality, participants were given one practice attempt. The practice session involved inserting the large square peg (Peg2) into a 3D-printed hole located on the opposite side of the NIST board. During the formal evaluation, each participant performed a single recorded trial per modality, with counterbalanced modality order and randomized peg order to mitigate learning effects. The overall experimental procedure was as follows:

- Participants receive a briefing on the research objectives and experimental procedure.
- General background information is collected via participant questionnaire.
- Participants complete the SSQ assessment, followed by a practice session with one modality.
- The Main experimental task is executed using the assigned modality.
- Post-task assessments conducted using the SUS, NASA-TLX, and SEQ questionnaires.
- Steps 3-5 repeated for alternative modality condition.
- Structured interview conducted to gather qualitative feedback.

V. Results and Discussion

Five volunteers participated in the user tests after providing written informed consent, which included details about the study's objectives and the confidentiality of their data in line with the Declaration of Helsinki [31]. The group consisted of 3 males and 2 females, aged 27 to 40 years old (33.2±4.76), all of whom volunteered without any form of compensation. Their familiarity with robotics and MR was assessed using

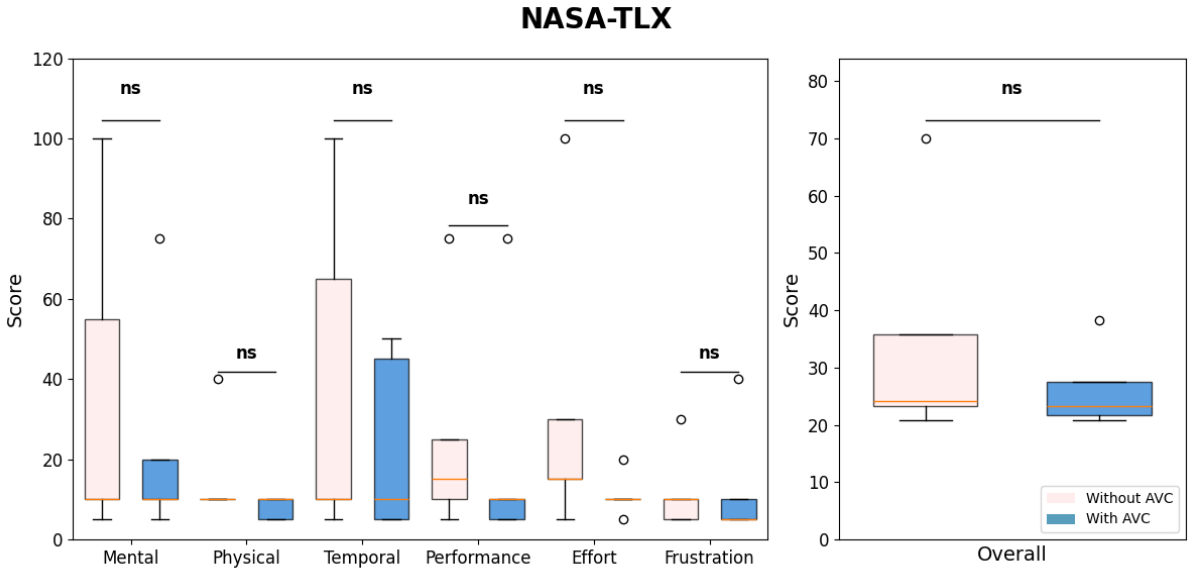

Fig. 7. The NASA-TLX dimensions.

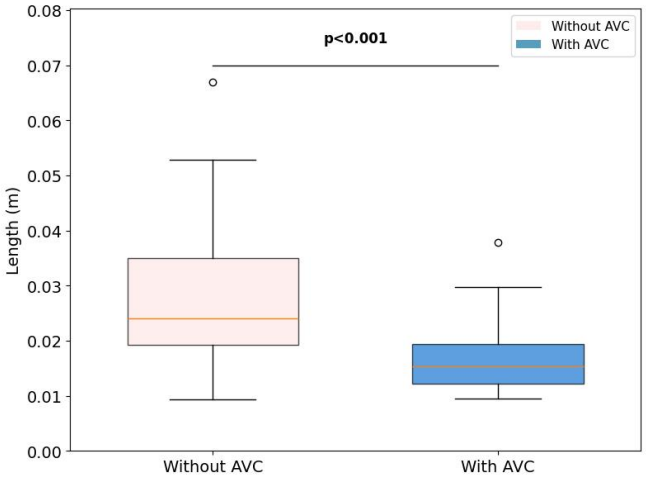

Fig. 8. Trajectory length.

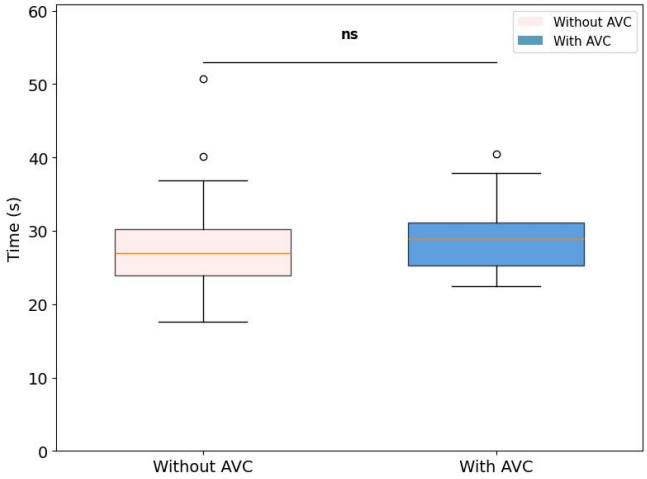

Fig. 9. Execution time.

a custom questionnaire on a 1-5 scale (1 = never used and 5 = everyday usage), revealing a moderate understanding of robotics (3±1.58) and MR (3.2±1.30), alongside limited experience in programming robotic arms (1.6±1.34) and using MR headsets (1.8±0.83).

Figure 5 illustrates the robot EE trajectories during peg-in-hole task execution with and without the AVC strategy, while Figure 6 presents the corresponding EE displacement along the x-axis and y-axis. The results clearly demonstrate that without the AVC strategy, the robot EE exhibits significant overshooting and oscillations during the final adjustment phase. This unstable behavior occurs because direct mapping between the user's hand movements and robot EE motion becomes difficult to coordinate precisely, particularly during fine manipulation tasks requiring precise control and subtle adjustments. In contrast, when the AVC strategy is employed, the system effectively stabilizes the robot EE motion, significantly reducing oscillations and overshooting behavior. This results in smoother trajectory control and more precise positioning as the peg approaches the target hole.

Data distributions for trajectory length and execution time were assessed using the Shapiro-Wilk test, presenting non-uniform distributions ($p < 0.05$). Regarding trajectory length, a two-way non-parametric Aligned Rank Transform (ART) test [32] with Group and Peg as within-subject factors showed a significant effect of Group ($F(1, 28) = 14.04$, $p < 0.001$). The multifactor contrast test procedure [33] showed statistically significant differences ($p < 0.001$) between the modality with AVC (0.002±0.014) and that without AVC (0.016±0.007), indicating that the AVC strategy provided shorter trajectories than without it. No other effects were detected. No statistically significant differences were found in execution time between

TABLE II

COMPARISON OF SUCCESS RATES, SUS AND NASA-TLX SCORES ACROSS OTHER MODALITIES

Study	Modality	Interface	Success Rate↑	SUS↑	NASA-TLX↓
[11]	Vision-based Fully Autonomous	-	68.75%	-	-
[35]	Kinesthetic Teaching	-	100%	59.25	48.17
[35]	Pure Telemanipulation	SpaceMouse	91.6%	41.75	63.6
[11]	Pure Telemanipulation	HTC Vive	91%	69.75	43.17
[35]	Pure Telemanipulation	HoloLens 2	93%	51	60
[11]	Semi-Autonomous Telemanipulation	HTC Vive	93%	82.87	24.78
Our work	**Semi-Autonomous Telemanipulation w/o AVC**	**Apple Vision Pro**	**95%**	**78.5**	**33.59**
Our work	**Semi-Autonomous Telemanipulation w. AVC**	**Apple Vision Pro**	**100%**	**92.29**	**18.99**

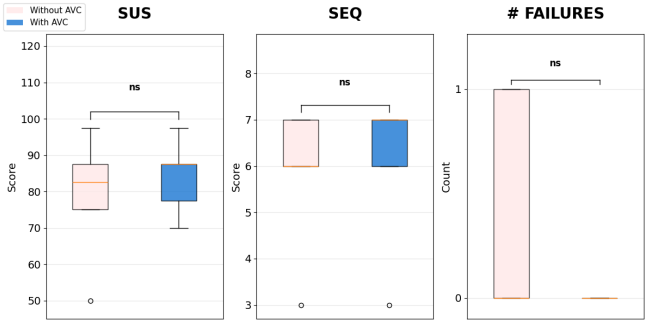

Fig. 10. The three diagrams represent the SUS, SEQ, and number of failures, respectively.

the two modalities across all pegs, as shown in Fig. 9, so both modalities achieve comparable efficiency in task completion time.

Figure 7 presents the average NASA-TLX values for weighted dimensions categorized by workload type, including p-values from pairwise statistical tests. No statistically significant differences were observed across workload dimensions; the Wilcoxon signed-rank tests failed to demonstrate statistically significant differences between conditions (all $p > 0.05$), potentially due to insufficient sample size. Nevertheless, effect size analysis using Cohen's d revealed meaningful practical differences. Notable large effect sizes were found for Physical workload ($d = 0.83$) and Effort ($d = 0.80$), along with a medium effect size for Overall workload ($d = 0.55$). Similarly, no statistically significant differences emerged in SUS and SEQ outcomes from Fig. 10, yet both modalities demonstrated similar usability and acceptability when employing AVC ($SUS_{w/} = 84.0 \pm 10.5$, $SEQ_{w/} = 6.0 \pm 1.7$) compared to the modality without AVC ($SUS_{w/o} = 78.5 \pm 17.9$, $SEQ_{w/o} = 5.8 \pm 1.6$). Based on [34], users experienced excellent usability (SUS > 80) with the AVC implementation, while the modality without AVC still maintained acceptable experience levels. Regarding the number of failures (Fig. 10), the modality with AVC provides a more reliable solution. These results indicate that both modalities achieve good levels of usability and performance, although the modality employing the AVC strategy produces shorter trajectories. However, larger sample sizes are needed to establish statistical significance.

We selected prior works for comparison based on different levels of autonomy for similar assembly tasks, as shown in Table II. The experimental tasks in [11], [35] are similar to ours. These works also introduced a fully autonomous baseline that utilizes a vision-based method to localize the peg and hole, and then execute the task without user intervention. Our work demonstrates considerable performance compared with other solutions, although the results are preliminary due to the limited sample size.

In summary, both modalities achieved comparable efficiency levels. The AVC strategy was associated with shorter trajectory lengths, suggesting a tendency to reduce unnecessary movements during task execution. Although no statistically significant differences were observed for execution time, workload, usability, or acceptability metrics due to the limited sample size, effect size analysis indicated potential practical benefits, particularly in reducing physical workload and effort demands. Overall, these findings provide preliminary evidence that the proposed semi-autonomous telemanipulation system demonstrates stable and efficient performance in peg-in-hole tasks, and that the AVC strategy may contribute to improved stability and efficiency. Further validation with a larger sample size is required to confirm these trends and assess their statistical robustness.

VI. CONCLUSIONS

This study introduced a semi-autonomous robotic telemanipulation system by implementing an adaptive velocity control strategy. It leverages the Apple Vision Pro mixed reality headset to provide a unique and immersive interaction experience. Additionally, the system was evaluated through a comprehensive user study, and the results highlight its reliability and effectiveness. The findings indicate that the system delivers an immersive and intuitive user experience while successfully executing accurate peg-in-hole assembly tasks with enhanced precision and efficiency. However, a larger sample size is required to thoroughly compare the proposed modalities, and the adaptive velocity control strategy could be further optimized through systematic experiments to determine optimal parameters for balancing semi-autonomous telemanipulation control.

References

[1] B. Bejczy, R. Bozyil, E. Vaičekauskas, S. B. K. Petersen, S. Bøgh, S. S. Hjorth, and E. B. Hansen, "Mixed reality interface for improving mobile manipulator teleoperation in contamination critical applications," *Procedia Manufacturing*, vol. 51, pp. 620–626, 2020.

[2] E. Triantafyllidis, C. Mcgreavy, J. Gu, and Z. Li, "Study of multimodal interfaces and the improvements on teleoperation," *IEEE Access*, vol. 8, pp. 78 213–78 227, 2020.

[3] A. Yew, S. Ong, and A. Nee, "Immersive augmented reality environment for the teleoperation of maintenance robots," *Procedia Cirp*, vol. 61, pp. 305–310, 2017.

[4] R. Hetrick, N. Amerson, B. Kim, E. Rosen, E. J. de Visser, and E. Phillips, "Comparing virtual reality interfaces for the teleoperation of robots," in *2020 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 2020, pp. 1–7.

[5] S.-J. Park, "A study on sensor-based upper full-body motion tracking on hololens," *Journal of The Korea Society of Computer and Information (JKSCI)*, vol. 26, no. 4, pp. 39–46, 2021.

[6] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang, "Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 12 248–12 255.

[7] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.

[8] S. Duan, F. De Pace, F. P. Sanches, H. Jiang, and M. Liarokapis, "Semi-autonomous, virtual reality based robotic telemanipulation for the execution of peg-in-hole assembly tasks," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2024, pp. 351–358.

[9] A. S. Alharthi, O. Tokatli, E. Lopez, and G. Herrmann, "Towards semi-autonomous robotic arm manipulation operator intention detection from force data," *IEEE Access*, 2024.

[10] D. Min, H. Yoon, and D. Lee, "A semi-autonomous telemanipulation order-picking control based on estimating operator intent for box-stacking storage environments," *Sensors*, vol. 25, no. 4, p. 1217, 2025.

[11] S. Duan, F. De Pace, and M. Liarokapis, "Comparing semi-autonomous strategies for virtual reality based remote robotic telemanipulation: On peg-in-hole tasks," *IEEE Robotics and Automation Letters*, vol. 11, no. 3, pp. 2562–2569, 2026.

[12] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, "Megapose: 6d pose estimation of novel objects via render & compare," *arXiv preprint arXiv:2212.06870*, 2022.

[13] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[14] J. C. Triyonoputro, W. Wan, and K. Harada, "Quickly inserting pegs into uncertain holes using multi-view images and deep network trained on synthetic data," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5792–5799.

[15] M. Nigro, M. Sileo, F. Pierri, K. Genovese, D. D. Bloisi, and F. Caccavale, "Peg-in-hole using 3d workpiece reconstruction and cnn-based hole detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4235–4240.

[16] R. Haugaard, J. Langaa, C. Sloth, and A. Buch, "Fast robust peg-in-hole insertion with continuous visual servoing," in *Conference on Robot Learning*. PMLR, 2021, pp. 1696–1705.

[17] W. Gao and R. Tedrake, "kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[18] Z. Zhang, Y. Wang, Z. Zhang, L. Wang, H. Huang, and Q. Cao, "A residual reinforcement learning method for robotic assembly using visual and force information," *Journal of Manufacturing Systems*, vol. 72, pp. 245–262, 2024.

[19] K. Zhang, C. Wang, H. Chen, J. Pan, M. Y. Wang, and W. Zhang, "Vision-based six-dimensional peg-in-hole for practical connector insertion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1771–1777.

[20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3. Kobe, Japan, 2009, p. 5.

[21] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Robotics research: The 15th international symposium ISRR*. Springer, 2016, pp. 493–504.

[22] T. T. Andersen, "Optimizing the universal robots ros driver." 2015.

[23] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke *et al.*, "ros_control: A generic and simple control framework for ros," *The journal of open source software*, vol. 2, no. 20, pp. 456–456, 2017.

[24] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4568–4575.

[25] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[26] J. A. Falco, "Iros 2019 robotic grasping and manipulation competition: Manufacturing track," *National Institute of Standards and Technology (NIST), Accessed*, vol. 7, 2020.

[27] H. Walter, R. Li, J. Munafo, C. Curry, N. Peterson, and T. Stoffregen, "Apal coupling study 2019," 2019.

[28] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[29] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.

[30] T. Rotolo, "The Single Ease Question," https://trymata.com/blog/2015/03/04/measuring-task-usability-the-single-ease-question/, accessed: 22/01/2024.

[31] "Declaration of Helsinki," https://tinyurl.com/7s6djxpn, accessed: 22/01/2024.

[32] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins, "The aligned rank transform for nonparametric factorial analyses using only anova procedures," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2011, pp. 143–146.

[33] L. A. Elkin, M. Kay, J. J. Higgins, and J. O. Wobbrock, "An aligned rank transform procedure for multifactor contrast tests," in *The 34th annual ACM symposium on user interface software and technology*, 2021, pp. 754–768.

[34] J. R. Lewis and J. Sauro, "Item benchmarks for the system usability scale." *Journal of Usability Studies*, vol. 13, no. 3, 2018.

[35] A. Smith and M. Kennedy III, "An augmented reality interface for teleoperating robot manipulators," *arXiv preprint arXiv:2409.18394*, 2024.