Closing the Adoption Gap: AI-driven, Low-Latency, and Compatible Remote Robotics Laboratories

Dr. Meenakshi Narayan, Miami University

Dr. Meenakshi Narayan is currently an assistant professor and the department coordinator for robotics engineering technology at Miami University, a position she has held since the fall of 2021. She earned her Ph.D. in Mechanical Engineering from the University of Texas at Dallas in December 2020. Dr. Narayan's doctoral research was centered on the development of robust predictive control models designed to enhance patient safety during robotic interventional procedures. Her research interests encompass a wide range of areas, including dynamics systems and control, teleoperation, haptics, data-driven models, signal processing, and artificial intelligence. She applies her expertise in these domains to address challenges in healthcare and industrial automation. At Miami University, Dr. Narayan instructs courses related to electrical engineering technology and is actively involved in the creation of innovative robotics courses for the Bachelor of Technology in Robotics program. These efforts involve collaboration with industry partners and are backed by support from university endowments and funding from the Ohio Department of Higher Education. Her overarching goal is to modernize and invigorate the engineering technology curriculum to prepare students for the demands of the future workforce, aligning with the evolving needs of the industry.

Moataz M. Salem, Miami University
Lokesh Kumar Saharan, Gannon University

Dr. Lokesh Saharan is currently serving as an Assistant Professor of Mechanical Engineering within the Department of Mechanical Engineering at the College of Engineering and Business at Gannon University. He obtained his Ph.D. from the University of Texas at Dallas in December 2017. Dr. Saharan's research interests encompass various fields, including engineering education, rehabilitation robotics, smart materials, soft robotics, and additive manufacturing. His academic contributions include multiple research articles published in peer-reviewed journals and conferences, as well as a book chapter. In addition to his research work, he has actively participated as a reviewer for various peer-reviewed conferences and journals. Before joining Gannon University, Dr. Saharan held the position of Assistant Professor and Department Coordinator for Mechanical Engineering at the University of Texas Permian Basin. During this time, he established and managed the Advanced Manufacturing Center, which received substantial funding of $1.1 million from the Odessa Development Corporation. He also served as a co-principal Investigator for a Department of Education EM-Step grant valued at $750,000. Further enriching his academic journey, Dr. Saharan contributed as an Assistant Teaching Professor at Penn State Behrend during the academic year 2019-20. Here, he played a pivotal role in developing new courses for the biomedical minor within the Mechanical Engineering department. Prior to his tenure in the United States, Dr. Saharan held the position of Assistant Professor (Instruction) in the Mechanical Engineering department at the National Institute of Technology Kurukshetra in India. Throughout his academic and research career, Dr. Saharan has made significant contributions in both teaching and research roles, spanning multiple countries and institutions.

Closing the Adoption Gap: Enabling Remote Robotics Laboratories Through Digital Twins and Latency-Aware Teleoperation

Abstract

The paper presents a work-in-progress educational teleoperation platform designed to support remote and hybrid robotics laboratories by addressing a key challenge in instructional teleoperation systems: end-to-end latency. The platform enables students to control a physical robot remotely while interacting with a synchronized robot digital twin, providing hands-on learning experiences for both distance and in-person learners. The system employs a lightweight, latency-aware predictive control framework based on pre-recorded user motion trajectories and adaptive control. User arm motions are captured through wearable sensing and mapped to joint-level commands that drive both the simulated robot and a physical robot concurrently. When tracking error exceeds a predefined threshold, a prediction algorithm selectively activates, using pre-recorded user commands as reference trajectories to compensate for communication and actuation delays. This approach reduces perceived latency while maintaining stability and simplicity, making it suitable for undergraduate engineering technology laboratories. The platform integrates a digital twin implemented in simulation software with a low-cost physical robot controlled through a microcontroller-based interface. Latency is explicitly measured across user input, simulation, physical actuation, and feedback stages to evaluate synchronization between systems. The implementation was embedded within an undergraduate robotics laboratory course, where students conducted repeated teleoperation trials using arm motion to control both the physical robot and its digital twin. Preliminary experimental results demonstrate a reduction in perceived command-to-actuation lag when prediction is enabled, along with improved tracking accuracy. Educational effectiveness was evaluated through post-lab student surveys included in course assessments, focusing on usability, learning effectiveness, and relevance to workforce preparation and advanced study. Survey results indicate strong student agreement regarding the platform's ease of use, learning value, and feasibility for remote laboratory instruction. While this study does not yet incorporate immersive reality interfaces, computer vision pipelines, or learning-based intent prediction, these components are planned as future extensions. Overall, this work establishes a cost-effective, scalable proof-of-concept for teleoperation-enabled remote laboratories, aligned with engineering technology education objectives related to applied learning, modern tool usage, and equitable access to hands-on robotics education.

Introduction

Robotic teleoperation lets humans control remote robots with live video and haptic feedback, combining human judgment with robotic precision for tasks in healthcare, manufacturing, and education[1,2,3,4,5]. It addresses workforce shortages and safety constraints by enabling experts to operate machines at a distance, particularly in complex or hazardous settings, and supports flexible production where fixed automation struggles[6,7,8,9,10,11]. The central barrier to reliable industrial adoption is total latency, which must be held below the recommended 50 ms for real-time feedback and control[12,13,14]. This paper addresses the challenge of enabling effective remote and hybrid robotics laboratories by developing a latency-aware teleoperation framework that synchronizes user arm motion, a robot digital twin, and a physical robot. Rather than relying on large training datasets or computationally intensive learning models, the work focuses on prediction using pre-recorded user motion data and adaptive control to reduce effective end-to-end latency and improve tracking accuracy. The proposed approach is intentionally lightweight and education-oriented, making it suitable for instructional laboratories and early-stage workforce training. By demonstrating proof-of-concept through simulation and physical experiments embedded within an undergraduate robotics course, this work establishes a cost-effective, scalable foundation for remote and in-person learning environments, with clear pathways for future integration of mixed reality and AI-based learning algorithms.

Existing Low Latency Platforms

Latency is a pervasive constraint in robotic teleoperation, where greater than 50 ms end-to-end delay can induce overshoot and oscillations degrading precision and increasing operator workload[15,16]. The latency accumulates across (1) sensing, (2) networking (3) computation, and (4) actuation, making closed-loop control more error-prone[17,18]. Prior studies have cut latency to under 15 ms, but they usually optimize a single stage such as network or process control rather than the whole pipeline[13,19]. There is little work on reducing total end-to-end latency.

Motivation and Objective

Existing solutions often rely on complex infrastructure or data-intensive learning models, limiting their suitability for instructional environments[20,21,22]. Motivated by these challenges, this work-in-progress paper aims to develop and evaluate a lightweight, latency-aware teleoperation framework that synchronizes user arm motion, a robot digital twin, and a physical robot using prediction based on pre-recorded user commands and adaptive control. The objective is to demonstrate a proof-of-concept remote laboratory platform embedded within an undergraduate robotics course, assess its educational effectiveness through student surveys and experimental validation, and establish a scalable foundation for future integration of mixed reality and AI-based learning models for advanced remote laboratories.

Human-Robot Interaction Platform for Remote Education

This section discusses the setup of an educational platform for remote learning in which users interact with a simulated environment while their commands are executed simultaneously by both a digital-twin robot and a physical robot. Users receive synchronized feedback from the simulated and physical systems, supporting both distance and local learners. Latency at each stage of interaction is explicitly measured to determine total system delay, and predictive methods are applied to reduce these delays and enable real-time operation.

a. Hardware Integration and Teleoperation

This work presents an educational remote-learning platform in which user commands are executed simultaneously by a robot digital twin and a physical robot, enabling synchronized interaction for both distance and in-person learners as shown in Fig. 1. User arm and hand motions are captured using Inertial Measurement Unit (IMU) and force/pressure sensors through a wearable interface, while a standard Arduino Uno microcontroller provides low-level servo control, real-time signal acquisition, and timing synchronization. The physical system uses a Wlkata Mirobot 6-DOF robotic arm; however, experiments are intentionally limited to single-joint motion to simplify analysis and enable precise measurement of latency between the simulated and physical environments. The objective is to verify accurate mapping and execution of joint commands across both systems and to characterize synchronization and delay. On the software side, CoppeliaSim (Edu) implements the robot digital twin with kinematic and joint-level simulation, while Python (via Visual Studio Code) is used for data processing, time-series prediction, and adaptive control. Time-series data consist of pre-recorded joint-angle trajectories derived from user hand motions, which are used to predict future commands and compensate for communication and actuation delays following prior work in[23]. Latency is measured across the user–digital twin, digital twin–physical robot, and physical robot–haptic feedback stages, with predictive methods applied to reduce total delay and enable stable, real-time human–robot interaction.

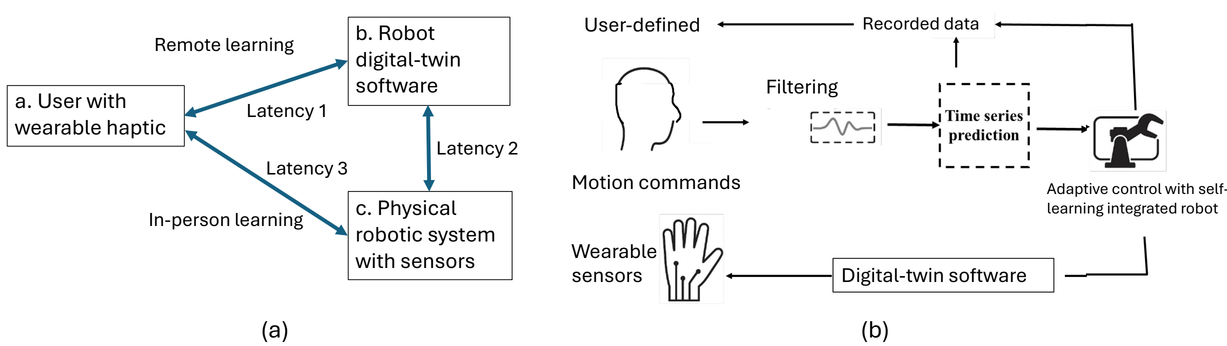


Figure 1: Digital-twin–based educational platform in which users interact with a simulated environment while their commands are executed by both a digital-twin robot and a physical robot.

b. Teleoperation Algorithm with latency Reduction

The flowchart of the proposed teleoperation algorithm is presented in Fig. 2, which models and mitigates latency across the full human–robot interaction loop, from user input to robot actuation and haptic feedback. Rather than attempting to minimize delay at individual system components, the method focuses on reducing perceived and effective latency through prediction and adaptive

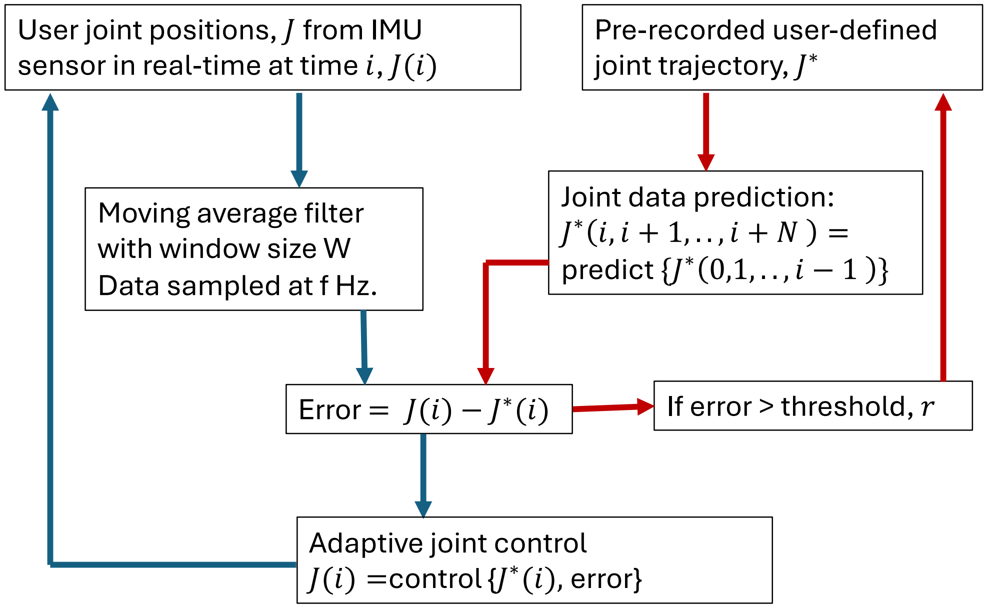
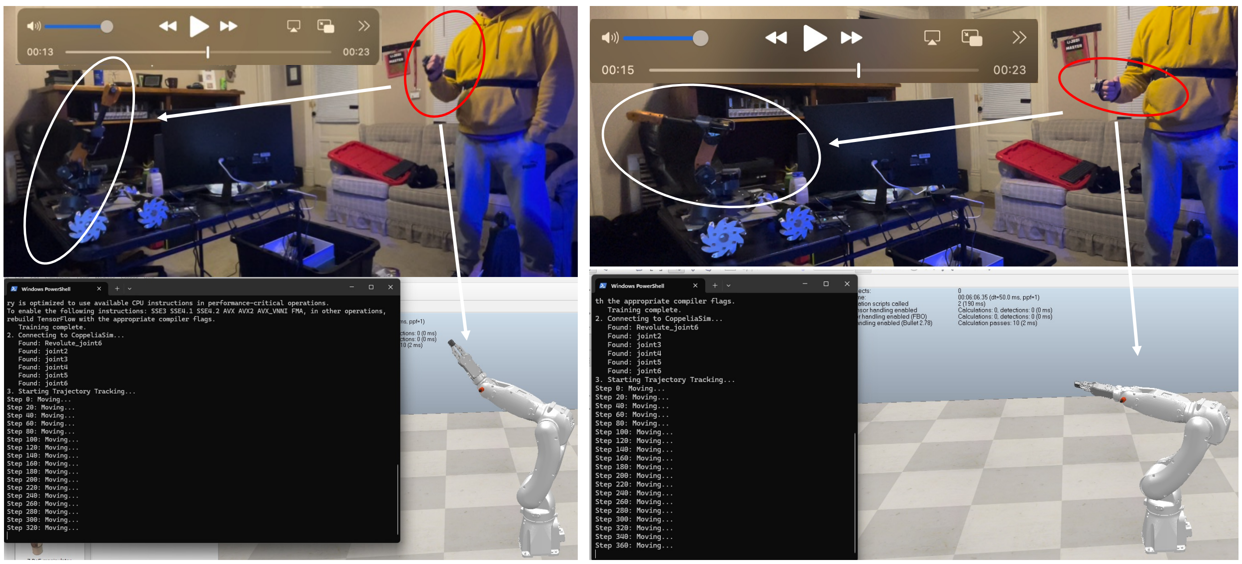


Figure 2: Flowchart of the latency-aware teleoperation algorithm illustrating threshold-based activation of prediction. When the error between the user-defined trajectory and the physical robot trajectory exceeds a predefined threshold, r, predicted joint data from the current time step i to future time steps are used as the reference trajectory for adaptive control. The error between this predicted reference and the actual physical trajectory is then used to compute the control input that drives the robot joint at the next time instant $i + 1$. When the error remains below the threshold, the real-time user trajectory is directly used as the reference input for adaptive control.

control. The algorithm begins with acquisition of user-defined arm motion, which is encoded as joint-level position commands. These commands are streamed simultaneously to a robot digital twin and to the physical robot controller. At each control cycle, the system computes the joint position error as the difference between the desired (target) joint position and the measured joint position of the physical robot. A threshold-based decision logic governs the activation of prediction. When the absolute joint position error exceeds a predefined threshold, r (arbitrarily set to 0.2 in this study), the controller switches from direct command execution to a predictive adaptive control mode. In this mode, the system references pre-recorded user motion trajectories, which are treated as time-series representations of user intent. These trajectories are used to pre-command joint motion, allowing the robot to anticipate incoming user inputs before real-time commands fully propagate through the sensing, computation, and actuation pipeline. Latency is characterized within the algorithm as a timing offset between the desired joint position at time t_2 and the corresponding physical joint response at time t_1 . While this latency metric is not directly used to drive control actions, it serves as an internal diagnostic measure to determine when prediction should be enabled or disabled. Once the joint tracking error falls below the specified threshold, the algorithm automatically returns to standard real-time control, ensuring stability and avoiding unnecessary prediction when synchronization is restored. This flow enables selective use of prediction only during high-error or high-latency conditions, preserving responsiveness during nominal operation. The experimental protocol intentionally used repeated user motion sequences to ensure consistency between real-time commands and pre-recorded trajectories, allowing isolation and evaluation of the prediction mechanism within the control flow. Although this implementation relies on known, repeated user motion patterns, the algorithmic structure is extensible. For real-time and dynamic teleoperation scenarios, the pre-recorded trajectory module can be replaced with learning-based intent prediction models trained on historical user motion data. This extension would allow the same latency-aware control logic to operate under variable and non-repetitive user commands, supporting robust and stable human–robot interaction in real-world applications.



(a) Latency around 1.5 second

(b) Latency around 0.1 second

Figure 3: Screenshot from the recorded simulation and experimental teleoperation videos showing a user controlling the physical robot arm through arm motion while the same motion is simultaneously simulated in CoppeliaSim. (a) Initial timestamps of the teleoperation sequence exhibit high latency between user input and physical robot response. (b) When the prediction algorithm is enabled, pre-recorded reference commands are applied, reducing the average latency between user commands and physical robot execution to approximately 0.1 s.

Experimental setup

To evaluate the proposed teleoperation framework with latency-aware predictive control, a controlled user study was conducted using both a physical robotic system and a simulated digital twin. The experiment involved students performing arm-motion–based teleoperation tasks to control a Wlkata Mirobot 6-DOF robotic arm, while the same commands were executed concurrently in the CoppeliaSim simulation environment.

Each participant first recorded a sequence of arm motions, which served as the pre-recorded reference trajectory for the predictive control algorithm. The motion sequence consisted of three distinct joint positions (up–flat–down), executed continuously. While the motion patterns differed across students, each student repeated the same sequence for four consecutive trials. This design choice ensured consistency across trials and enabled evaluation of algorithmic stability in the absence of machine-learning–based adaptation.

The experiments intentionally avoided learning-based models to isolate the effect of the prediction algorithm using pre-recorded commands. The primary objective was to assess how prediction improves synchronization between user input, the digital twin, and the physical robot under fixed latency conditions. The system architecture allowed the student to visually observe both the physical robot and the simulated robot while performing teleoperation using arm motion.

A screenshot of the experimental setup and simulation interface is shown in Fig. 3, where a student controls the physical robot while the corresponding simulated robot executes the same motion in real time. Initial trials were conducted without prediction to establish baseline latency, followed by trials with prediction enabled.

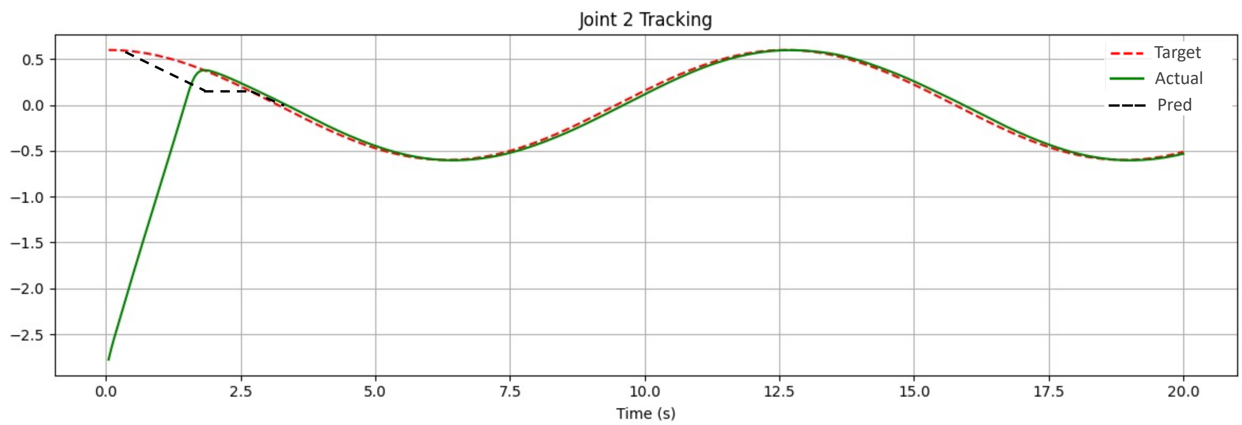


Figure 4: Joint position tracking comparing user-defined arm motion (Target, dotted red) with the physical robot trajectory (Actual, solid green). When the joint error exceeded a threshold of 0.2 during initial timestamps, a prediction algorithm used pre-recorded user motion as the reference for adaptive control, reducing the average tracking error to 2.37% (SD 0.62%). Latency was estimated from the time offset between target and actual joint positions and was mitigated by selectively activating prediction during high-error intervals.

Teleoperation Results

An example frame from the teleoperation experiment during initial operation is shown in Fig. 3(a), where a noticeable lag of approximately 1.5 s is observed between the student's arm motion and the physical robot response. This latency is visually evident when comparing the user motion to the robot position in both the physical workspace and the simulated environment. When the prediction algorithm was activated using the pre-recorded command sequence, the lag between user motion and physical robot execution was significantly reduced. Figure 3(b) illustrates the same experiment under prediction-enabled control, where the average latency decreased to approximately 0.1 s, resulting in visibly synchronized motion between the student's command, the digital twin, and the physical robot. Joint-level performance is further illustrated in the joint position tracking plot shown in Fig. 4. The plot compares the target joint trajectory derived from user arm motion (dotted red), the actual physical robot trajectory (solid green), and the predicted trajectory (dashed black). Large discrepancies between target and actual trajectories occur during

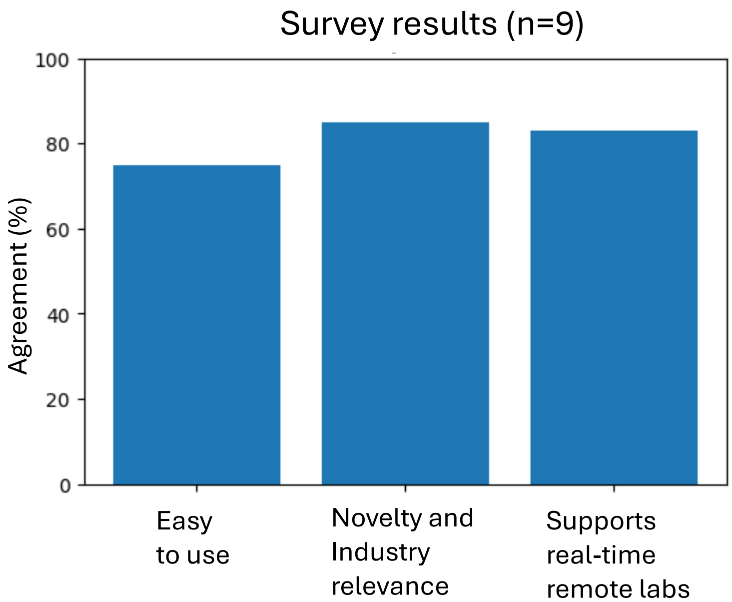


Figure 5: Student survey response analysis on the educational platform.

the initial timestamps due to accumulated latency. When the absolute joint error exceeds a predefined threshold of 0.2, the prediction algorithm replaces the real-time command with the pre-recorded reference trajectory for adaptive control.

Using this approach, the average tracking error between the desired and actual joint position was reduced to 2.37%, with a standard deviation of 0.62% across trials. Latency was computed as the time difference between the target joint position at time t_2 and the corresponding actual joint position at time t_1, measured only when the tracking error fell below the threshold. Although latency was not directly used as a control variable, it served as a diagnostic measure to verify that prediction enabled timely alignment between intended and executed motion. The use of repeated, identical motion sequences ensured that real-time user commands matched the pre-recorded trajectories, allowing reliable evaluation of prediction performance. This experimental design validates the feasibility of the algorithm under controlled conditions and demonstrates its potential to mitigate latency in human-in-the-loop teleoperation.

As this is a work-in-progress study, future extensions will incorporate dynamic user commands, variable network latency, and learning-based intent prediction. For example, in application domains such as soldering laboratories, expert demonstrations, where only a limited set of motion patterns are valid, can be recorded and used to train deep learning models. These models, stored and updated through a cloud-based infrastructure, would enable prediction of real-time user intent under more complex and non-repetitive task conditions.

Student Survey Outcomes and Learning Metrics

To evaluate the educational effectiveness of the proposed teleoperation-based remote laboratory, a short post-lab survey was administered to students (n = 9) enrolled in the Integrated Robotics Systems laboratory course during the Fall 2025 semester. The teleoperation trials described in this paper were embedded as part of a required laboratory activity, and the survey questions were included within the course's end-of-semester evaluation framework. This approach grounds assessment within authentic instructional contexts.

The survey questions were organized into three primary categories, each comprising multiple Likert-scale items (1–4, with 4 representing the highest rating). Category-level scores were computed by averaging student responses within each category and converting the results to percentages for interpretability.

Category 1 (Interface and Responsiveness) focused on usability and perceived latency, including whether the remote lab interface was intuitive and whether students experienced noticeable lag between their commands and visual robot motion. Student responses indicated that the system was generally easy to use, with initial latency noted during early trials but reduced in later prediction-enabled trials.

Category 2 (Skill Development and Career Relevance) evaluated perceived gains in technical skills, relevance to industry and graduate studies, and student awareness of emerging technologies such as mixed reality and artificial intelligence for future expansion. Over 80% of respondents agreed or strongly agreed that the platform enhanced their skills and was relevant to career pathways in automation, robotics, and research-oriented roles.

Category 3 (Learning Effectiveness and Digital Twin Value) assessed the impact of the digital twin on conceptual understanding, flexibility in laboratory practice, and equivalence or improvement of learning outcomes compared to traditional in-person labs. More than 80% agreement was observed in this category, indicating strong student perception that the digital twin improved understanding of robot motion and enabled effective remote laboratory experiences.

From an experimental standpoint, each student conducted four repeated trials of robot motion after initially recording a user-specific sequence of arm movements. While motion sequences varied across students, the same sequence was reused across trials for each individual. This design ensured consistency and allowed evaluation of teleoperation stability in the absence of machine-learning–based adaptation. The results demonstrate that the system can reliably support repeated remote interactions, an important requirement for scalable laboratory instruction.

Conclusion and Future work

As a work-in-progress study, this paper does not yet incorporate learning-based intent prediction, fully immersive mixed-reality interfaces, or AI-driven adaptation, which are essential components of a complete remote laboratory ecosystem. Future work will integrate mixed reality (MR) to provide immersive visualization, spatial awareness, and intuitive interaction with both the digital twin and physical robot, along with AI-based learning algorithms to enable intelligent intent prediction and adaptive control. The platform will be extended to support dynamic user commands and variable latency conditions using prediction and learning models trained on observed user data. For example, in soldering laboratories where task execution typically follows a limited number of valid motion patterns, expert demonstrations can be recorded and used to train deep learning models for real-time intent prediction and guidance. These planned extensions will transform the prototype into a comprehensive remote laboratory platform and further strengthen its alignment with engineering education objectives, including modern tool usage, workforce readiness, applied learning, and equitable access to hands-on engineering technology education.

References

[1] Paola Picozzi, Umberto Nocco, Greta Puleo, Chiara Labate, and Veronica Cimolin. Telemedicine and robotic surgery: a narrative review to analyze advantages, limitations and future developments. *Electronics*, 13(1):124, 2023.

[2] Ziling Chen, Yeo Jung Yoon, Rolando Bautista-Montesano, Zhen Zhao, Ajay Mandlekar, and John Liu. Teleoplab: Accessible and intuitive teleoperation of a robotic manipulator for remote labs. *arXiv preprint arXiv:2509.05547*, 2025.

[3] Claudia González, J Ernesto Solanes, Adolfo Munoz, Luis Gracia, Vicent Girbés-Juan, and Josep Tornero. Advanced teleoperation and control system for industrial robots based on augmented virtuality and haptic feedback. *Journal of Manufacturing Systems*, 59:283–298, 2021.

[4] Abid Haleem, Mohd Javaid, Ravi Pratap Singh, and Rajiv Suman. Telemedicine for healthcare: Capabilities, features, barriers, and applications. *Sensors international*, 2:100117, 2021.

[5] Cosimo Gentile, Giacomo Lunghi, Luca Rosario Buonocore, Francesca Cordella, Mario Di Castro, Alessandro Masi, and Loredana Zollo. Manipulation tasks in hazardous environments using a teleoperated robot: A case study at cern. *Sensors*, 23(4):1979, 2023.

[6] Manufacturing in crisis: Report uncovers key challenges for manufacturers, 2022. URL `https://www.l2l.com/newsroom/manufacturing-in-crisis-report-uncovers-key-challenges-for`

[7] Automate.org. The benefits of remote control capabilities for industrial robots, 2024. URL `https://www.automate.org/industry-insights/benefits-remote-control-capabilities-industr`

[8] Bingjie Tang, Iretiayo Akinola, Jie Xu, Bowen Wen, Ankur Handa, Karl Van Wyk, Dieter Fox, Gaurav S Sukhatme, Fabio Ramos, and Yashraj Narang. Automate: Specialist and generalist assembly policies over diverse geometries. *arXiv preprint arXiv:2407.08028*, 1(2), 2024.

[9] Xiang Zhang, Changhao Wang, Lingfeng Sun, Zheng Wu, Xinghao Zhu, and Masayoshi Tomizuka. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Conference on Robot Learning*, pages 1621–1639. PMLR, 2023.

[10] Teleoperation: a revolution for industry and public works, 2024. URL `https://efa-controls.com/en/teleoperation-a-revolution-for-industry-and-public-works/`.

[11] Sunwook Kim, Ivan Hernandez, Maury A Nussbaum, and Sol Lim. Teleoperator-robot-human interaction in manufacturing: Perspectives from industry, robot manufacturers, and researchers. *IISE Transactions on Occupational Ergonomics and Human Factors*, 12(1-2):28–40, 2024.

[12] Edgar M Hidalgo, Mats Isaksson, Mariadas Capsran Roshan, Thomas H Marwick, Leah Wright, and Gavin Lambert. Evaluating the impacts of network latency, haptics, and ergonomics in a haptically-enabled robot for teleoperated echocardiography. *Computers in Biology and Medicine*, 195:110450, 2025.

[13] Nick Bray, Matthew Boeding, Michael Hempel, Hamid Sharif, Tapio Heikkilä, Markku Suomalainen, and

Tuomas Seppälä. A latency composition analysis for telerobotic performance insights across various network scenarios. *Future Internet*, 16(12):457, 2024.

[14] Ahmad Faiz Fawzan Ahmad Zainudin, Mohd Hairi Mohd Zaman, Mohd Faisal Ibrahim, Mohd Asyraf Zulkifley, and M Marzuki Mustafa. Enhanced teleoperation of robotic arms through master-slave configuration with multithreading. *IEEE Access*, 2025.

[15] Henrikke Dybvik, Martin Løland, Achim Gerstenberg, Kristoffer Bjørnerud Slåttsveen, and Martin Steinert. A low-cost predictive display for teleoperation: Investigating effects on human performance and workload. *International Journal of Human-Computer Studies*, 145:102536, 2021.

[16] Yang Ye, Tianyu Zhou, Qi Zhu, William Vann, and Jing Du. Brain functional connectivity under teleoperation latency: a fnirs study. *Frontiers in Neuroscience*, 18:1416719, 2024.

[17] Sidharth Bhanu Kamtam, Qian Lu, Faouzi Bouali, Olivier CL Haas, and Stewart Birrell. Network latency in teleoperation of connected and autonomous vehicles: A review of trends, challenges, and mitigation strategies. *Sensors (Basel, Switzerland)*, 24(12):3957, 2024.

[18] Muhammad Thoriq, Hendri Maja Saputra, Catur Hilman AHB Baskoro, et al. A systematic literature network analysis of telecommunication system performance in robotic surgery: The role of latency. 2025.

[19] Ahmad Faiz Fawzan Ahmad Zainudin, Mohd Hairi Mohd Zaman, Mohd Faisal Ibrahim, Mohd Asyraf Zulkifley, and M. Marzuki Mustafa. Enhanced teleoperation of robotic arms through master-slave configuration with multithreading. *IEEE Access*, 12:99187–99198, 2024. doi: 10.1109/ACCESS.2024.3417772.

[20] Prajwal Thakur. Cloud-connected model predictive control for autonomous mobile robots in the presence of network latency and message losses. 2025.

[21] Jaziar Radianti, Tim A Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & education*, 147:103778, 2020.

[22] Meng Chu, Ziang Cui, Ailing Zhang, Jiale Yao, Chenyu Tang, Zhe Fu, Arokia Nathan, and Shuo Gao. Multisensory fusion, haptic, and visual feedback teleoperation system under iot framework. *IEEE Internet of Things Journal*, 9(20):19717–19727, 2022.

[23] Meenakshi Narayan and Ann Majewicz Fey. Model-free control for autonomous prevention of adverse events in robotics. *Frontiers in Robotics and AI*, 10:1271748, 2024.