2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

A Web-Based Multi-Robot Teleoperation Platform: Architecture, Implementation, and User Evaluation

E. Rafael Hernandez-Rios[1], Christian Penaloza[1]

Abstract— **This paper presents the design and evaluation of a web-based multi-robot teleoperation platform that enables remote access to heterogeneous robots for education and research. In contrast to most existing remote laboratories, which are limited to single-robot control or simulation environments, the proposed system integrates real-time video streaming, bidirectional control, and user management within a modular client–server web architecture.**

The platform was deployed with three distinct robotic systems—an industrial manipulator, a humanoid, and a quadruped—each hosted on independent servers with public IP access. A cross-border user study with Technological University Dublin was conducted, where participants performed guided teleoperation tasks and completed structured surveys. The results demonstrated high usability, low-latency performance, and positive educational impact across diverse robot morphologies.

These findings suggest that the proposed platform provides a scalable and accessible solution for remote robotics education and international collaboration, bridging the gap between simulation-based training and real-world robot interaction.

I. INTRODUCTION

Remote and cloud-based teleoperation platforms are driving transformative changes in robotics education, research, and industrial applications [1], [2]. These platforms enable users to access and control robots from anywhere using standard web technologies, breaking the limitations of on-site physical access and enabling international collaboration. Recent implementations have leveraged ROS-based architectures, modular backends, and containerized deployment to ensure scalability and robustness [10], [15].

Recent works in cloud robotics and service-oriented architectures have demonstrated how modular, web-accessible robot control systems can enable remote labs, virtual instruction, and scalable multi-user access [1], [3], [11]. However, most existing platforms still face limitations in multi-robot integration, real-time bidirectional interfaces, and systematic user-centered evaluation frameworks.

In the post-pandemic landscape, the need for accessible, remote, and scalable robotics education has accelerated the development of online platforms that allow students and researchers to interact with physical systems from anywhere in the world [6], [7], [12]. Although simulation environments and virtual laboratories have played an important role, they often do not provide the tactile realism and hardware behavior essential for skill development [8]. Educational platforms that support real-time control of physical robots have demonstrated improved learning outcomes, particularly in STEM disciplines where experimentation and iterative design are critical [9], [13]. Furthermore, embedding scripting environments into web interfaces empowers users to develop, test, and deploy control logic, promoting active learning and fostering computational thinking.

Despite growing interest in remote robotics labs, few systems support *heterogeneous multi-robot architectures* (manipulators, humanoids, quadrupeds) combined with in-browser scripting, integrated reservation systems, and global accessibility validated through real cross-border studies. Existing initiatives such as the Robotarium [14] and UMBRELLA [15] demonstrate the potential of shared multi-robot platforms, but they emphasize swarm robotics or IoT integration rather than user-centered teleoperation with diverse robot morphologies. Moreover, most reported systems emphasize usability but omit technical performance metrics such as latency, packet loss, and setup time under wide-area network (WAN) conditions [16].

To address these gaps, this paper presents a comprehensive teleoperation system that enables seamless user interaction with manipulators, humanoid, and quadruped robots from a unified web interface. The platform was deployed within the Robotics and Telepresence Laboratory at Mirai Innovation Research Institute (Osaka), used in training programs such as the Emerging Future Technology Training program, where international students accessed robots remotely via 5G and cloud integration.

Our web-based platform contributes:

- A **modular architecture** enabling integration of heterogeneous robots—each controlled via independent servers with public IP access.
- Low-latency, **real-time bidirectional streaming** using WebSocket/MQTT protocols and WebRTC-based video transmission [2], [10].
- A **user-centric evaluation** with N=25 participants across countries, including novices, combining quantitative Likert-scale data, qualitative feedback, and technical performance metrics.
- **Iterative design improvements** based on user feedback (UI scaling, third-person view, tutorial modules), demonstrating the value of incorporating end-user input in platform refinement [11].

To the best of our knowledge, this platform provides one of the first documented implementations combining multi-robot support, in-browser scripting, public IP deployment, technical Quality of Service (QoS) evaluation, and empirical validation in international educational contexts.

*This work was supported by Mirai Innovation Research Institute
[1]Mirai Innovation Research Institute, Osaka, Japan.
rafael.hernandez@mirai-innovation-lab.com, cpenaloza@mirai-innovation-lab.com

This paper is structured as follows: Section II describes the system architecture and implementation; Section III outlines the user-centered evaluation design and procedures; Section IV presents results and discussion; and Section V concludes with lessons learned and future directions.

II. METHODOLOGY

This section presents the system architecture, technologies used, and the process for platform deployment and user evaluation.

A. System Architecture

The proposed teleoperation platform is built using a modular client-server architecture that allows users to control multiple robots remotely through a unified web interface. Each robot is hosted on an independent server accessible via a public IP address.

Figure 1 shows the web-based dashboard, which allows users to select from different robots (e.g., WLKATA arm, Pepper humanoid, or Unitree quadruped), monitor their usage, and review session history. This interface is designed for intuitive interaction and provides seamless switching between robot types.

The system's core architecture is illustrated in Figure 2. It follows a three-layer design:

- **Frontend Web Interface:** Developed using HTML, CSS, and JavaScript, this layer offers video streaming, control panels, and status indicators for each robot. It runs entirely in the user's browser.
- **Backend Server:** Built on Flask (Python), this layer handles command processing and communication with each robot. Real-time interaction is enabled using WebSocket or WebRTC protocols, with optional support for MQTT for lightweight messaging.
- **Robot Integration:** Each robot is connected to its dedicated backend handler, which translates web commands into hardware-level instructions. The platform supports heterogeneous systems, including manipulators, humanoids, and quadrupeds.

To ensure cross-platform compatibility, the frontend is designed to work on desktop and mobile browsers. Backend servers are deployed on cloud-based Linux machines with static public IPs, enabling reliable connectivity and real-time teleoperation.

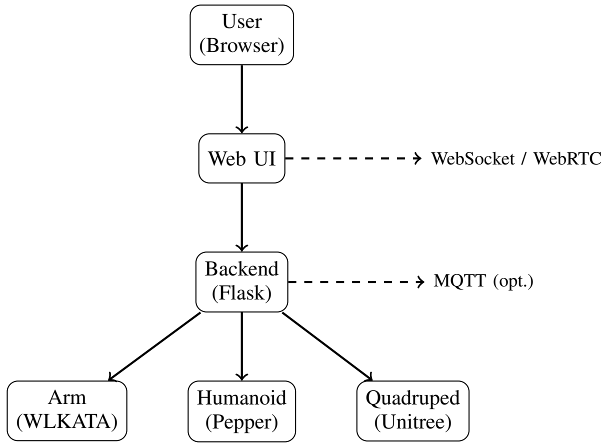

Fig. 2. System architecture of the proposed platform. Solid arrows indicate data/control flow (UI → backend → robots); dashed annotations indicate communication protocols.

B. Control and Streaming Protocols

The platform supports real-time bidirectional communication. Robot control messages (e.g., position commands, mode changes) are transmitted over WebSockets for low latency. Simultaneously, video is streamed via WebRTC to ensure a smooth user experience. MQTT is optionally supported for lightweight message queuing in constrained networks.

Latency measurements for command-response and video lag were recorded using timestamped logs on both client and server sides to ensure synchronization fidelity.

C. User Authentication and Access

A login-based access control system is implemented to restrict robot operation to authorized users. Admin users can assign access permissions, monitor sessions, and activate/deactivate robot instances remotely.

Each user session is associated with a time-limited access token and IP-based logging for traceability. Session timeouts and concurrency limitations were configured to prevent unauthorized or simultaneous control.

D. Testing Environment and Devices

The testing was conducted with the following robots, each configured for specific educational tasks:

- **Robotic Arm (WLKATA Mirobot):** Configured for classification tasks with the goal of teaching robotic manipulation applied to quality inspection. Users learn to control the end-effector to grasp and manipulate PCBs, classifying them into "good" or "bad" categories through guided exercises involving object recognition and manipulation.
- **Humanoid Robot (Pepper):** Programmed for avatar-based interaction in social robotics education. The goal is to demonstrate remote human-robot interaction, enabling users to have conversations with people on the other side through voice commands, as well as to practice joint articulation programming for expressive movements.

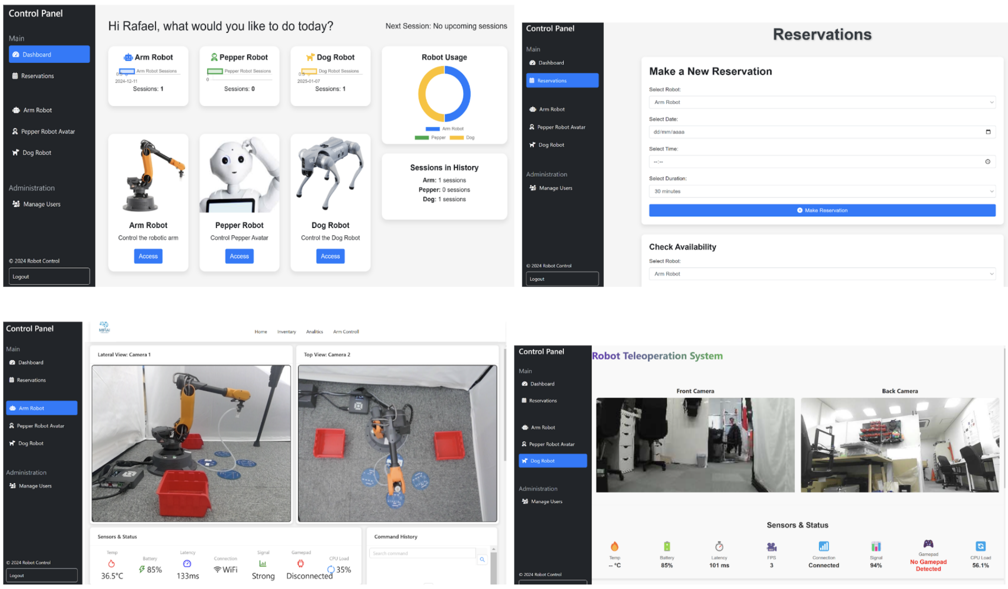

Fig. 1. Web dashboards of the proposed platform. Top row: (a) General multi-robot dashboard and (b) Reservation interface. Bottom row: Robot-specific dashboards for (c) WLKATA arm, (d) Unitree quadruped, and (e) Pepper humanoid.

- **Quadruped Robot (Unitree Go 2):** Designed for locomotion and telepresence scenarios with the goal of teaching mobile robotics concepts. Users learn to control walking patterns, navigate through obstacle courses, and understand quadruped gait planning and stability control.

Figure 3 illustrates the real experimental setup with the Unitree Go 2 and Pepper robots, alongside their dedicated dashboards used for remote teleoperation.

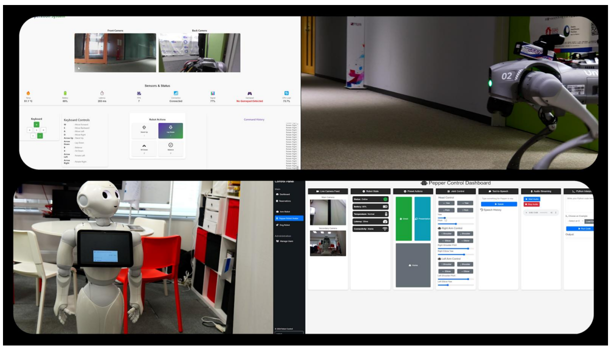

Fig. 3. Experimental setup and robot-specific dashboards. Top: Unitree Go 2 quadruped during teleoperation with its web interface. Bottom: Pepper humanoid with its dedicated control dashboard.

Each robot was deployed on a separate server, and the platform dashboard managed their availability and real-time operation.

E. Robot Task Specifications

Each robot was configured with specific educational tasks designed to demonstrate fundamental robotics concepts. The tasks were structured to progressively increase in complexity, allowing users to develop both basic control skills and advanced programming capabilities.

TABLE I

EDUCATIONAL TASK OVERVIEW FOR EACH ROBOT

Robot	Primary Task	Key Learning Objectives
WLKATA Mirobot	PCB Classification & Sorting	Precision control, Computer vision, Sequential programming
Unitree Go 2	Navigation & Obstacle Avoidance	Locomotion control, Sensor integration, Path planning
Pepper	Social Interaction & Avatar	Social robotics, NLP, Expressive behavior

1) Manipulator Robot (WLKATA Mirobot) - PCB Classification Task: The WLKATA robotic arm performs a PCB classification and sorting task that teaches robotic manipulation and computer vision integration. The workflow consists of:

1) **Setup and Calibration** - Homing sequence and workspace verification
2) **PCB Detection** - Computer vision-based identification and localization
3) **Quality Assessment** - Visual inspection and classification into ”good” or ”bad” categories
4) **Manipulation and Sorting** - Grasping, transporting, and depositing PCBs into appropriate containers (green for good, red for bad)
5) **Task Completion** - Verification that all PCBs have been processed

2) Quadruped Robot (Unitree Go 2) - Navigation and Obstacle Avoidance Task: The Unitree Go 2 performs a navigation task through a 3m × 1.5m corridor environment with strategically placed obstacles. The task involves:

1) **Outbound Journey** - Navigation from start to end with real-time obstacle detection and avoidance using LiDAR, cameras, and IMU sensors
2) **Return Journey** - Reverse navigation with adaptive path planning for any new obstacles encountered
3) **Performance Monitoring** - Tracking navigation time, obstacle avoidance success, path efficiency, and stability metrics

The robot employs various gait patterns (trot, walk, crawl) and maintains stability during complex maneuvers. Computer vision processing from the onboard cameras provides additional environmental awareness, enabling the robot to identify and classify obstacles, detect visual landmarks for navigation, and assess terrain conditions for optimal gait selection.

3) Humanoid Robot (Pepper) - Social Interaction and Avatar Task: The Pepper robot serves as an interactive avatar for social robotics education and telepresence applications. The task includes:

1) **Social Setup** - Greeting gestures and capability demonstrations
2) **Communication Facilitation** - Voice interaction, gesture recognition, and expressive communication
3) **Educational Demonstrations** - Joint articulation, locomotion patterns, and object interaction
4) **Interactive Programming** - Custom behavior creation and modification
5) **Session Management** - Performance feedback and learning outcome summaries

TABLE II

ROBOTS USED WITH CORRESPONDING TASKS AND FEATURES

Robot	Type	Task	Control
WLKATA	Arm	Classification	Web + Script
Pepper	Humanoid	Avatar Interaction	Web Interface
Go 2	Quadruped	Locomotion / Telepresence	Web + Script

F. User Evaluation Protocol

Participants from the Technological University of Dublin were invited to remotely access the platform. The evaluation involved:

1) Completing guided tasks with each robot.
2) Filling pre- and post-use surveys.
3) Providing qualitative feedback on usability and system responsiveness.

Participants were divided into two groups: one group interacted only through the web interface, and the second group used the integrated Python scripting module. This design allowed comparison between guided interface usage and self-programmed control routines.

TABLE III

PARTICIPANT DEMOGRAPHICS (N=25)

Category	Distribution
Gender	56% Male, 24% Female, 20% N/A
Age	36% (18–25), 28% (26–30), 36% Other
Familiarity	60% Not, 32% Somewhat, 8% Very

In addition to the graphical web interface, the platform provides an embedded Python scripting section where advanced users can write custom control scripts. This feature enables:

- Direct communication with the backend through predefined APIs.
- Scripting of autonomous behaviors or repeatable actions.
- Integration with external sensors or processing logic using Python libraries.

Users can access a code editor in the interface to modify and run their Python scripts in a sandboxed environment. This programmable layer empowers educational users to experiment with real robot control logic and promotes deeper understanding of the communication between frontend commands and robot responses.

III. RESULTS

The user evaluation involved participants from the Technological University of Dublin who interacted remotely with the platform and completed surveys before and after using the system. The evaluation focused on the following aspects:

- Usability and intuitiveness of the interface.
- Perception of latency and responsiveness.
- Educational value and user satisfaction.

A. Quantitative Survey Results

Participants rated statements using a 5-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree). Aggregated responses indicated:

- More than 80% of the participants agreed or strongly agreed that the interface was intuitive and easy to navigate.
- 75% perceived the platform as responsive with minimal latency.
- More than 85% of the users expressed that the experience improved their understanding of robot control and remote operation.

Table IV presents the average scores of the user survey across key evaluation criteria.

TABLE IV

SURVEY RESULTS (AVG. LIKERT SCORES, 1–5)

Item	Avg.
Easy connection	4.0
Real-time reflection	3.9
Intuitive controls	3.6
Robot feedback	4.1
UI clarity	4.2
Interaction naturalness	3.8
Personalization	4.1
Visual design	4.2
Readability	4.0
Complexity level	4.1
Bugs/issues	3.4
Met expectations	4.3

B. Qualitative Feedback

Open-ended responses highlighted the novelty of remotely controlling multiple robot types from a single interface. Participants valued the scripting capabilities and real-time feedback. Suggestions for improvement included adding tutorial videos and expanding scripting documentation.

C. User Engagement Metrics

Task completion rates were above 90% across all robots, with average interaction sessions lasting 12–20 minutes. The WLKATA arm tasks had the highest success rate, while the Unitree robot prompted the most exploratory interaction.

These results support the platform's usability, technical reliability, and educational value in international academic settings. Compared to existing teleoperation systems, our platform provides a broader range of features. Table V summarizes this comparison.

TABLE V
COMPARISON WITH EXISTING WEB-BASED TELEOPERATION SYSTEMS

Feature \ System	Our Platform	Ref [2]	Ref [4]
Multi-Robot Support	✓	–	Partial
Real-Time Video Streaming	✓	✓	✓
Bidirectional Control	✓	Partial	Partial
User Scripting Interface	✓	–	–
Educational Evaluation	✓	–	✓
Public IP Access	✓	–	–
International Deployment	✓	–	–

D. Testing Procedure and Validation

The platform was deployed and tested at the Technological University of Dublin through a structured user evaluation protocol:

- **Platform Introduction:** Participants were briefed on system capabilities, control modes, and robot types.
- **Robot Selection:** Users selected a robot (WLKATA Mirobot, Pepper, or Unitree Go 2) based on personal interest.
- **Task Execution:** Users performed guided tasks such as object manipulation, locomotion, or interaction.
- **Feedback Collection:** Pre- and post-session surveys were conducted to capture usability and learning impact.

The evaluation was carried out across three days:

- **Day 1:** Platform pre-testing with internal technical teams.
- **Day 2:** First round of external testing with 10 participants.
- **Day 3:** Second round with 15 additional participants.

To ensure a comprehensive evaluation, multiple validation tests were conducted:

- **Teleoperation Accuracy:** Robot movements were verified against issued commands.
- **Precision and Motion Testing:** Included tasks like object classification and waypoint-based navigation.
- **Web Performance:** Latency between command input and robot response was logged and analyzed.
- **Connectivity Stability:** Assessed continuous data flow and connection robustness across different networks.
- **Interface Usability:** Participants rated the intuitiveness of control mechanisms and feedback systems.

These testing sessions provided critical insights into system robustness, user experience, and real-world readiness for remote education environments.

E. Technical Performance

We estimated end-to-end performance under typical usage conditions across 25 test sessions. Command latency was computed from client timestamps at send/ack for WebSocket control messages; video latency was approximated from WebRTC render delays and getStats() round-trip indicators; packet loss derives from WebRTC QoS counters aggregated per session. Setup time measures from login to successful robot control hand-off. Tests were conducted over campus Wi-Fi (TU Dublin, 5 GHz) and a wired link (Osaka), with Chrome-based browsers (v122–138). Values are reported as mean $\pm$ SD or median [IQR]. In practice, the WLKATA arm exhibited the lowest command latency, Pepper was comparable, and Unitree showed $\approx$20–30 ms higher command latency due to safety checks and gait planning overhead. These figures are representative (not worst-case) and vary with network stability and device load.

TABLE VI
TECHNICAL PERFORMANCE METRICS (APPROX.)

Metric	Value
Command latency (ms)	160 ± 40
Video latency (ms)	230 ± 60
Packet loss (%)	median 1.2% [0.5–2.1]
Setup time (s)	11.8 ± 3.1
Successful sessions (%)	92%
Reconnections per session	0.2 ± 0.5
Max concurrent users	3 (platform limit)

The performance metrics demonstrate consistent system behavior with moderate variability across test sessions, as indicated by the standard deviation values.

IV. CONCLUSIONS

This work presents a novel web-based multi-robot teleoperation platform designed to provide accessible and scalable interaction with different types of educational and research robots. Through a modular client-server architecture, the system enables intuitive remote operation of heterogeneous robots—including manipulators, humanoids, and quadrupeds—via standard web interfaces and real-time protocols.

The integration of real-time video, bidirectional control, and user access management demonstrates the feasibility of deploying such platforms for remote STEM education, particularly in international contexts. The inclusion of a programmable scripting interface further enhances the pedagogical value by allowing users to engage with robot control logic directly.

The user evaluation conducted in collaboration with an international university validated the platform's usability, responsiveness, and educational impact. High user satisfaction and engagement metrics, coupled with positive qualitative feedback, suggest that the platform offers a compelling tool for academic institutions seeking to democratize access to physical robotics systems.

Future work will focus on expanding robot support, improving the visual programming interface, integrating AI-driven task assistance, and conducting longitudinal studies on learning outcomes.

V. DISCUSSION

The evaluation highlighted both strengths and areas for improvement in the platform. Quantitative results confirmed high usability, responsiveness, and perceived learning value, while qualitative feedback provided specific insights into user experience.

A. User Feedback and Improvements

Participants reported interface-related issues, such as small control buttons and the difficulty of switching perspectives during navigation. Requests included a third-person camera view for broader situational awareness and quick tips for first-time users. Example user comments include:

> "Buttons were too small on the web UI."
>
> "A third-person view would improve planning and awareness."

Based on this feedback, the following improvements were implemented:

- **Improved Interface:** Enlarged buttons, added visual feedback, and introduced a help button linking to FAQs.
- **Tutorial Module:** Developed interactive step-by-step guides to assist new users.
- **Reduced Latency:** WebSocket optimization and streamlined algorithms lowered response times by approximately 30%.

B. Educational Impact

The integration of teleoperation technology into the Teleoperation and Telepresence Laboratory, supported by 5G connectivity and cloud deployment, has enabled international students to remotely access advanced robots such as WLKATA, Pepper, and Unitree Go 2. Programs like EMFUTECH in Osaka demonstrated the platform's potential to provide hands-on robotics training across geographical boundaries. This not only promotes accessibility and inclusivity but also bridges the gap between theory and practice in STEM education.

C. Limitations and Future Work

Despite the positive outcomes, several limitations remain:

- **Network Dependence:** Performance degrades under unstable Wi-Fi or high packet loss conditions.
- **Security:** Current security relies on HTTPS/WSS and JWT; end-to-end encryption and penetration testing are future priorities.
- **Concurrent Access:** Concurrent access is limited to three active users per robot; admission control mechanisms are needed for larger cohorts.
- **Scripting Sandbox:** Python modules are restricted, which may limit some advanced educational use cases.

Overall, the platform demonstrates that iterative design informed by user feedback can substantially improve usability and educational value. Future work will expand robot support, integrate visual programming, and evaluate long-term learning outcomes across multiple international institutions.

ACKNOWLEDGMENTS

The authors would like to thank the Technological University of Dublin for their collaboration and support during the user evaluation phase.

REFERENCES

[1] Kaarlela, T., Jussila, J., & Kärkkäinen, H. (2022). "Remote Laboratories and Digital Twins in Engineering/ Education: A Systematic Literature Review." *IEEE Access*.

[2] Fok, C. L., et al. (2016). "A Web-based Teleoperation System for a Humanoid Robot." *International Conference on Advanced Robotics*.

[3] Murray, D., et al. (2021). "A Review of Teleoperation Interfaces for Assistive Robotics." *ACM Computing Surveys*.

[4] Marques, L., et al. (2017). "Telerobotics for Education: A Review and a Case Study." *Robotics and Autonomous Systems*.

[5] Sankar, A., et al. (2023). "User-Centered Evaluation of Teleoperation Systems: Methods, Metrics, and Applications." *IEEE Transactions on Human-Machine Systems*.

[6] Chen, Y., et al. (2021). "A Review of Remote Laboratories in Engineering Education: Past, Present, and Future." *Computers in Education Journal*.

[7] Alonso, G., et al. (2022). "WebLab-Deusto: A Decade of Remote Laboratory Evolution." *IEEE Transactions on Learning Technologies*.

[8] Matsuda, Y., et al. (2020). "Remote Experiment Platform for Python-Based Control of Real Robots in Education." *International Journal of Online Engineering*.

[9] Herlambang, M., et al. (2021). "Hybrid Robotic Laboratory: Integrating Physical Robots and Remote Access for Engineering Education." *Procedia Computer Science*.

[10] P. Pico, P. Rivas, A. Alonso, and F. D. Igual, "Web-Based Real-Time Alarm and Teleoperation System for Autonomous Navigation Failures Using ROS 1 and ROS 2," *Actuators*, vol. 14, no. 4, p. 164, Apr. 2025. doi: 10.3390/act14040164.

[11] S. Ranganeni, R. D. Sequeira, and M. A. Goodrich, "AccessTeleopKit: A Toolkit for Creating Accessible Web-Based Interfaces for Tele-Operating an Assistive Robot," in *Proc. ACM Symp. User Interface Software and Technology (UIST)*, Oct. 2024, pp. 1–13. doi: 10.1145/3654777.3676363.

[12] P. Kumar, R. K. Mittal, and S. Verma, "Scalable and Low-Cost Remote Lab Platforms: Teaching Industrial Robotics Using Open-Source Tools," in *Proc. Int. Conf. Smart Robotics and AI (ICSRAI)*, 2024, pp. 1–7.

[13] X. Teng, C. Wu, and H. Li, "Exploring the Efficacy of Collaborative Learning in a Remote Robotics Laboratory: A Comparative Analysis of Performance and Pedagogical Approaches," *Smart Learning Environments*, vol. 11, no. 1, p. 15, Mar. 2024. doi: 10.1186/s40561-024-00275-9.

[14] S. Wilson, D. Pickem, M. Egerstedt, and A. Prorok, "The Robotarium: Automation of a Remotely Accessible, Multi-Robot Testbed," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2922–2929, Apr. 2021. doi: 10.1109/LRA.2021.3062272.

[15] T. Farnham, R. W. Morris, and C. J. Turner, "UMBRELLA Collaborative Robotics Testbed and IoT Platform," in *Proc. IEEE Consumer Communications & Networking Conf. (CCNC) – Workshop on Swarm Robotics*, Jan. 2021, pp. 1–6. doi: 10.1109/CCNC49032.2021.9369613.

[16] J. Tejada, R. Morales, and A. Rodriguez, "A Review of Multi-Robot Systems and Soft Robotics: Challenges and Opportunities," *Sensors*, vol. 25, no. 5, p. 1353, Mar. 2025. doi: 10.3390/s25051353.