2025 International Conference “ROBOTICS & MEHATRONICS”
5 – 6 November, 2025, Sofia, Bulgaria

A Design-Based Research Approach to Human–Machine Interface for Cobot Teleoperation

Daniel Stoyanov
Institute of Robotics
Bulgarian Academy of Sciences, Acad. Georgi Bonchev str., 1113 Sofia
d.stoynaov@ir.bas.bg

***Abstract*— Collaborative robots (cobots) are increasingly deployed in industrial and research contexts, requiring intuitive human–machine interfaces (HMIs) for teleoperation. However, many existing interfaces are either too complex for non-expert users or lack portability across platforms. This study applies a Design-Based Research (DBR) methodology to the iterative design and evaluation of a modular HMI for cobot teleoperation. The interface was developed in Unity, enabling seamless transfer to mixed reality (MR) and virtual reality (VR) environments. The underlying architecture leverages the lightweight MQTT protocol for fast, reliable, and low-overhead communication between the operator's interface and the robot. Its modular design ensures adaptability to different types of cobots. Pilot experiments were conducted to validate system usability and communication latency. Findings show promising results in terms of responsiveness, modularity, and ease of adaptation.**
***Keywords*—design-based research; collaborative robots; human-machine interface; teleoperation; Unity; MQTT;**

I. INTRODUCTION

Collaborative robots (cobots) are rapidly transforming industrial, service, and research domains due to their ability to operate safely in close proximity with humans. Unlike traditional industrial robots, which are typically confined to cages and require specialized programming skills, cobots are designed to be more adaptable, interactive, and user-friendly. They are now widely used in assembly, quality inspection, logistics, and healthcare, where flexibility and human–robot cooperation are essential [1]. A key challenge, however, lies in designing intuitive human–machine interfaces (HMIs) that allow operators with varying levels of technical expertise to effectively teleoperate cobots in dynamic environments.

Teleoperation remains an important function even in the age of increasing robot autonomy. Operators may need to take control in scenarios where human judgment, adaptability, or contextual awareness is required. Examples include precision assembly tasks, remote inspection in hazardous environments, and collaborative learning where the robot's autonomous decision-making is insufficient. However, current HMI solutions are often constrained by several factors: vendor-specific platforms that limit portability, steep learning curves for novice users, and limited support for emerging technologies such as mixed reality (MR) and virtual reality (VR) [2].

Developing HMIs in Unity offers an advantage here, as the engine natively supports both 2D desktop environments and immersive MR/VR deployments, enabling rapid transfer of interfaces across multiple devices and platforms. This flexibility makes Unity an attractive foundation for modular HMI design aimed at diverse cobot systems.

In addition to usability and immersion, communication performance is critical in teleoperation. High-latency or bandwidth-heavy communication channels can disrupt operator control, leading to inefficiency or even safety risks. For this reason, the present study adopts the MQTT protocol, a lightweight publish–subscribe communication standard optimized for fast and reliable data transfer. Originally designed for low-bandwidth IoT applications, MQTT is well suited to robotics because of its efficiency, scalability, and robustness under variable network conditions [3].

Finally, modularity is essential for ensuring that an HMI can be adapted across different cobot platforms. Many existing solutions are tied to specific hardware or proprietary middleware, which hinders knowledge transfer and slows down innovation. A modular architecture—where robot-specific drivers are separated from the interface logic—not only promotes reusability but also allows researchers and practitioners to rapidly adapt the system to new hardware. This adaptability is critical in research environments, where multiple cobots with diverse specifications (e.g., Franka Emika Panda, UR series, KUKA LBR) are often available.

To address these challenges, this paper applies a Design-Based Research (DBR) methodology to the iterative design and evaluation of a modular, Unity-based HMI for cobot teleoperation. DBR is particularly suitable for this purpose as it emphasizes iterative cycles of design, testing, and refinement within real-world contexts, ensuring that solutions are grounded in both theory and practice. Through this approach, we contribute not only a technical implementation but also design principles for developing adaptable, MR-ready teleoperation systems supported by lightweight communication protocols.

The main contribution of this study is the initial iteration in the development and validation of a prototype interface for the Franka Emika Panda Robot. Pilot experiments validated the feasibility of the system and highlighted strengths such as low-latency communication and responsive Unity-based interaction. At the same time, these tests identified areas for improvement, including expanding MR/VR functionality and conducting larger-scale usability studies. The results confirm the value of the DBR approach in guiding both system development and iterative enhancement.

ACKNOWLEDGE THE FINANCIAL SUPPORT OF THE PROJECT WITH FINANCING AGREEMENT NO. PVU-44 OF 05.12.2024 UNDER PROJECT NO. BG-RRP-2.017-0011 "ECOLOGICAL COLLABORATIVE ROBOTS POWERED BY GREEN HYDROGEN" UNDER THE RECOVERY AND RESILIENCE MECHANISM FOR THE IMPLEMENTATION OF AN INVESTMENT UNDER C2I2 "INCREASING THE INNOVATION CAPACITY OF THE BULGARIAN ACADEMY OF SCIENCES (BAS) IN THE FIELD OF GREEN AND DIGITAL TECHNOLOGIES" FROM THE RECOVERY AND RESILIENCE PLAN, BULGARIA.

II. Related Work

A. Human-Machine Interfaces for Cobots Teleoperation

The development of human–machine interfaces (HMIs) for teleoperation has been studied in a variety of contexts, ranging from industrial robotics to medical and service applications. Traditional HMIs rely on joystick or teach pendant control, which, while robust, require extensive operator training and provide limited situational awareness in complex tasks. More recent research has emphasized the role of graphical user interfaces (GUIs) and web-based dashboards, which improve accessibility but often lack the intuitiveness needed for dynamic teleoperation [4].

Collaborative robots are shaping the future of the HMI in Industry 5.0, shifting from automation to human-centric design [5], that puts people at the center and introduces additional requirements for HMI design. Since cobots are intended to operate in close proximity to humans, interfaces must not only support direct teleoperation but also enable shared autonomy, monitoring, and safety-aware interventions. Studies have highlighted the importance of clear feedback loops—visual, auditory, and haptic—to ensure that operators maintain an accurate mental model of the robot's state during teleoperation [6]. However, many commercially available solutions remain tied to proprietary software stacks, limiting portability across platforms.

B. Lightweight, Communication Protocols in Robotics

Communication is a critical factor in the performance of teleoperation systems. High data transfer overhead can introduce delays that negatively affect task execution and operator trust. Standard robotics frameworks, such as ROS, provide robust communication capabilities but can be resource-intensive in scenarios requiring lightweight, real-time responsiveness.

The MQTT protocol has recently gained attention in robotics for its lightweight publish–subscribe architecture. Originally designed for Internet of Things (IoT) applications, MQTT offers low overhead, efficient message delivery, and the ability to scale across multiple devices and networks. Research has shown that MQTT can achieve lower latency than traditional HTTP-based communication while maintaining high reliability. Its suitability for bandwidth-constrained or distributed systems makes it a strong candidate for cobot teleoperation, where rapid response and flexible integration are required [7].

C. Modularity and Adaptability in Cobots Interfaces

One of the key challenges in teleoperation HMI research is the lack of modularity. Interfaces are often designed for a single robot platform, limiting their reuse in other contexts. Modularity has been identified as a critical design principle in both robotics middleware and user interface design, as it allows researchers and practitioners to adapt systems quickly to new tasks and hardware.

Recent work has explored middleware solutions that decouple robot-specific drivers from higher-level control and visualization logic. By using standardized communication protocols and modular software architectures, these systems aim to support a wide range of robots with minimal reconfiguration. For example, modular Unity-based front ends combined via MQTT to ROS back ends have been proposed to enable portability across multiple cobot platforms [9].

D. Design-Based Research in Robotics Interfaces

Design-Based Research (DBR) has been applied in educational technologies, assistive systems, and increasingly in robotics, where iterative cycles of design, testing, and refinement are essential. Unlike controlled laboratory studies, DBR emphasizes contextualized development and collaboration with end users. In robotics, this means involving operators, engineers, and stakeholders in the design process to ensure that systems are both technically robust and practically usable.

Prior DBR studies in robotics have focused on socially assistive robots, adaptive learning platforms, and AI integration, highlighting the importance of modular architectures and user-centered design. Yet, there is a notable gap in DBR-driven research on teleoperation interfaces for cobots, particularly those designed to be transferable to MR/VR contexts. This study aims to address this gap by combining Unity-based immersive interface design, MQTT communication, and modular adaptability within a DBR framework.

III. Methodology

This study follows a Design-Based Research methodology to guide the design, development, and evaluation of a human–machine interface for cobot teleoperation. DBR is particularly suitable for complex, socio-technical systems such as robotics, as it emphasizes iterative design, real-world testing, and close collaboration between researchers, practitioners, and end-users. The goal is not only to produce a working solution but also to extract generalizable design principles for future implementations.

The methodological framework adopted here is based on the four-phase model proposed by Reeves, consisting of Problem analysis through collaboration with stakeholders; Iterative design and development of solutions; Evaluation of the interventions in practice; Reflection and refinement of design principles.

A. Phase 1: Problem Analysis

The first phase involved identifying the challenges and requirements associated with cobot teleoperation. A literature review was conducted to examine existing HMI solutions, focusing on usability, portability, MR/VR integration, and communication protocols. In addition, informal interviews were carried out with robotics researchers and engineers who regularly work with collaborative robots. From this analysis, several core challenges were identified:

- Usability barriers: Many cobot HMIs require significant technical expertise, limiting accessibility for non-specialists.
- Lack of portability: Vendor-specific platforms are difficult to transfer across cobot types or immersive environments.
- Communication inefficiencies: Existing frameworks often introduce unnecessary overhead, leading to latency during teleoperation.
- Limited modularity: Interfaces are often tightly coupled to a single robot, hindering adaptability.

These challenges provided the foundation for design principles guiding subsequent phases: Unity-based cross-platform development for MR/VR compatibility; MQTT-based lightweight communication for low latency; modular architecture to support multiple cobots platforms.

B. Phase 2: Iterative Design and Development

Following the problem analysis, a prototype HMI was designed in Unity. Unity was chosen because it supports 3D visualization, immersive interaction, and cross-platform deployment to desktop, mobile, web, and MR/VR headsets. The prototype included core functionalities such as robot state visualization and direct teleoperation commands.

To ensure efficient communication, an MQTT broker was implemented as the middleware between Unity and the cobot. Unity clients published teleoperation commands (e.g., joint angles, Cartesian movements, gripper actions), while subscribing to robot state updates (e.g., position, force feedback, error states). The use of MQTT minimized latency and allowed for straightforward extension to additional devices or cobots.

The proposed architecture (Figure 2) was explicitly designed to be modular: robot-specific drivers were abstracted from the Unity interface logic. This enables rapid adaptation of the HMI to different cobots without redesigning the core interface. During development, simulation in Unity was used to test communication flows before deployment to physical robots, reducing risks and development time.

Iterative design cycles included building, testing, and refining the prototype based on feedback from early users (research engineers). Each cycle focused on improving usability, reducing latency, and enhancing modularity.

C. Phase 3: Pilot Evaluation

Pilot experiments were conducted using the Franka Emika Panda cobot in a laboratory setting. Participants included robotics researchers familiar with teleoperation tasks but not necessarily with the specific interface. The evaluation aimed to capture both quantitative and qualitative data:

Quantitative measures:

- Communication latency (measured between Unity commands and cobots response).
- Task success rates (completion of object manipulation tasks under teleoperation).
- Error rates (unexpected robot behaviors or communication failures).

Qualitative measures:

- User perception of usability, intuitiveness, and responsiveness.
- Feedback on interface design, visualization, and workflow.

Although sample size was limited, the pilot tests provided critical insights into system performance and user experience. Results indicated that MQTT communication maintained latency around 50 ms in all experiments (Figure 1), while Unity provided intuitive visualization. Participants emphasized the potential of transferring the interface to MR/VR devices to further improve situational awareness.

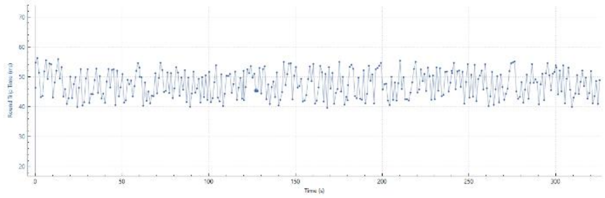

Fig. 1 MQTT Round Tripe Time

D. Phase 4: Reflection and Refinement

The final phase of this iteration involved synthesizing lessons learned from the pilot evaluation and refining the design principles. Key outcomes included:

- Validation of Unity as a flexible platform for both desktop and immersive HMI development.
- Confirmation of MQTT's suitability for lightweight, real-time teleoperation.
- Recognition of the importance of modularity for cross-cobots adaptability.
- Identification of areas for future work, including large-scale usability testing, MR/VR integration, and incorporation of multimodal feedback (e.g., haptics).

By following the DBR cycle, the research not only delivered a functional prototype but also contributed to a broader understanding of how to design adaptable HMIs for cobot teleoperation.

IV. System Architecture

The proposed system architecture was designed to support three overarching requirements: cross-platform transferability, lightweight and responsive communication, and modularity across different cobot platforms. To achieve these goals, the system is structured around three interconnected layers: the Unity-based human–machine interface, the MQTT communication middleware, and the cobot execution layer with modular adapter (Figure 2).

At the front end, the Unity environment functions as the operator-facing interface. Unity was selected because it allows a seamless transition between standard desktop interaction and immersive deployment in mixed or virtual reality. Within this interface, operators are presented with real-time visualizations of the robot's state and task environment, as well as direct mechanisms for issuing control commands. By integrating Unity's rendering capabilities, the interface facilitates a more natural and intuitive form of interaction compared with traditional two-dimensional dashboards.

Communication between the HMI and the robot is managed by an MQTT broker, which mediates the flow of information between Unity and the cobot execution layer. Unlike heavier communication frameworks, MQTT is a lightweight publish–subscribe protocol that reduces both network overhead and latency. In practice, this means that operator commands can be transmitted to the robot with delays consistently around 50 milliseconds, while feedback from the robot is immediately routed back into the Unity interface. The broker-based architecture also enables multiple cobots or monitoring clients to be connected simultaneously, highlighting the scalability of this approach.

The final layer of the system is the cobot execution environment, which is accessed through modular adapters.

Each adapter serves as a translator between the generic MQTT topics and the proprietary instruction set of a particular robot. For instance, an adapter has been developed for the Franka Emika Panda that allows both joint-level and Cartesian teleoperation. Additional adapters can be implemented with relative ease, ensuring that the interface is not bound to a single robot model. This modular design principle isolates robot-specific functionality from the rest of the system, preserving the generality of the Unity interface and the communication layer.
Interaction across these three layers forms a continuous feedback cycle. The operator issues commands in Unity, which are transmitted via MQTT to the appropriate adapter, executed by the robot, and then reported back through MQTT to update the interface. This closed-loop process maintains synchronization between operator input and robot action, while ensuring that the operator is constantly informed about the robot's state and environment.

Taken together, the Unity interface, the MQTT communication layer, and the modular adapters form an architecture that is not only technically efficient but also highly adaptable. The design supports rapid integration with different cobot platforms, ensures low-latency responsiveness necessary for teleoperation, and provides a natural pathway for extension into immersive MR and VR environments. In this way, the architecture balances immediate experimental functionality with long-term flexibility and scalability.

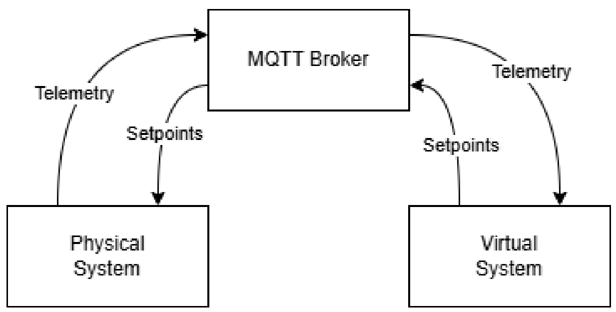

Fig.2 System architecture for modular cobots teleoperation

V. DISCUSSION, CONCLUSIONS AND FUTURE WORK

The design and evaluation of the proposed teleoperation interface highlight the potential of combining Unity, MQTT communication, and modular robot adapters within a design-based research framework. The system was deliberately structured to meet three critical requirements: cross-platform transferability, low-latency communication, and adaptability to different cobot platforms. Each of these requirements was addressed by specific architectural choices that not only functioned well in isolation but also reinforced each other when combined into a coherent whole.
The Unity-based interface played a central role in ensuring usability and flexibility. Unlike traditional teach pendants or text-based programming environments, Unity provided a visual and interactive layer that allowed operators to directly understand and influence robot behavior. This is particularly important in teleoperation contexts, where clear visualization of the robot's state and workspace can significantly reduce operator workload and errors. Moreover, Unity offered a natural pathway toward immersive mixed reality and virtual reality applications. Even though the pilot experiments focused on a desktop environment, the underlying design ensures that the same interface can be transferred to MR or VR headsets with relatively minor adjustments. This opens promising avenues for future work, as immersive visualization is expected to further enhance situational awareness, precision, and overall operator performance.
Equally important was the use of MQTT as the communication backbone. In robotics, communication frameworks are often burdened by complexity or heavy network demands. By contrast, MQTT offered a lightweight and efficient alternative, enabling near real-time responsiveness with measured latencies consistently around 50 milliseconds during pilot tests. For teleoperation, where delays directly impact control quality and user trust, this level of responsiveness is a crucial achievement. Furthermore, MQTT's publish–subscribe architecture supports scalability, meaning that multiple cobots, monitoring systems, or auxiliary devices can be integrated into the same environment without overhauling the communication infrastructure. This flexibility positions the system as a viable solution not only for single-cobot teleoperation but also for more complex scenarios involving multi-robot collaboration.
The third pillar of the architecture, modular adapters, addressed the challenge of hardware dependence. Many existing teleoperation interfaces are tightly bound to specific robots, making them difficult to reuse or extend. By introducing adapters that translate MQTT messages into robot-specific commands, the proposed system isolates hardware dependencies and preserves the generality of the Unity interface and the communication middleware. This design principle proved successful in the case of the Franka Emika Panda and sets the stage for broader applicability to other platforms such as UR cobots, ABB robots. However, realizing this potential will require further empirical testing to ensure that the adapters remain robust and reliable across diverse hardware ecosystems.
While the pilot experiments confirmed the feasibility of the architecture, they also revealed important limitations. The current system was tested only with a small group of research engineers, which limits the generalizability of the findings. Larger-scale studies involving operators with varying levels of experience will be needed to fully assess usability, learning curves, and long-term adoption potential. Furthermore, the experiments were conducted in a controlled laboratory setting, which does not capture the full complexity and unpredictability of industrial environments. Extending the evaluation to more realistic contexts, such as factory floors or collaborative assembly tasks, will provide richer insights into the system's practical value.
From a methodological perspective, the design-based research approach proved to be a valuable framework for structuring development. By combining theoretical design principles with iterative testing and refinement, DBR ensured that the system evolved in response to both technical requirements and user needs. The pilot evaluation, although limited, provided concrete feedback that informed the refinement of the design principles. Future iterations of the DBR cycle will focus on immersive MR/VR integration and long-term usability assessments, thereby strengthening the system's foundations and broadening its applicability.

In conclusion, the study demonstrates that a modular, Unity-based teleoperation interface supported by lightweight MQTT communication can offer a scalable, adaptable, and user-friendly solution for cobot teleoperation. The architecture successfully addresses the challenges of portability, latency, and hardware independence, while laying the groundwork for future expansion into immersive environments. The pilot results provide initial validation of these design choices, while also pointing to areas requiring further research and refinement. Moving forward, the project will concentrate on extending MR/VR functionality, conducting larger-scale evaluations, and exploring scenarios involving multiple cobots. By following an iterative DBR process, the system is expected to evolve into a robust platform that not only advances academic research but also provides practical solutions for the growing field of collaborative robotics.

ACKNOWLEDGMENT

This research was funded by the European Regional Development Fund under the Operational Program "Scientific Research, Innovation and Digitization for Smart Transformation 2021–2027", Project CoC "Smart Mechatronics, Eco- and Energy Saving Systems and Technologies", BG16RFPR002-1.014-0005.

REFERENCES

[1] M. Dhanda, B. Rogers, S. Hall, E. Dekoninck, "Reviewing human-robot collaboration in manufacturing: Opportunities and challenges in the context of industry 5.0", Robotics and Computer-Integrated Manufacturing, vol. 93, 102937, 2025

[2] D. Gonzalez-Sanchez, "Exploring Extended Reality to Optimize Human-Cobot Interaction", ACM Transactions on Human-Robot Interaction, 14(2), 1–22, 2025

[3] J. Rouillard, J.M. Vannobel , "Multimodal Interaction for Cobots Using MQTT", Multimodal Technologies and Interaction, 7(8), 78, 2023

[4] S. Young, J. Peschel, "Review of Human-Machine Interfaces for Small Unmanned Systems with Robotic Manipulators", IEEE Transactions on Human-Machine Systems, Vol. 50, pp.131-143, 2020

[5] D. Mourtzis, J. Angelopoulos,N. Panopoulos, The Future of the Human–Machine Interface (HMI) in Society 5.0. Future Internet, 15(5), 162, 2023

[6] D. Rea, S. Seo, "Still Not Solved: A Call for Renewed Focus on User-Centered Teleoperation Interfaces", Frontiers in Robotics and AI, 2022

[7] T. Luu, Q. Nguyen, T. Tran, "Enchansing real-time robot teleoperation with immersive virtual reality in industrial IoT network", The international Journal of Advanced Manufacturing Technology, 2025

[8] M. Singh, J. Kapukotuwa, "Unity and ROS as Digital and Communication Layer for Digital Twin Application: Case Study of Robotic Arm in Smart Manufacturing Cell", Sensors (Basel), 2024