Object-Based Teleoperation Interface for Collaborative Manipulation

2nd Year Internship Report

Interactive Robotics Lab — Tokyo University of science — Katsushika-ku, Tokyo

Toméo Bourin
Autonomous Robotics
FISE 2026
tomeo.bourin@ensta.fr

September 30, 2025

Abstract

In summer 2025, I carried out my internship at the Interactive Robotics Laboratory of Tokyo University of Science. The main objective was to design a mixed-reality framework to control a collaborative robot and explore how such an interface could support research on human–robot interaction.

My work involved building a modular system that connected Unity with the robot's SDK through ROS2, enabling real-time communication between the virtual and physical environments. To make interaction more natural, I designed a digital twin of the robot and introduced an object-based control method.

The framework was successfully tested in collaborative manipulation tasks, showing that the robot could share the effort with a human. A user study confirmed that the object-based method felt more natural and usable. These results provided both a functional prototype for the lab and the basis for future research work.

This internship allowed me to strengthen my skills in robotics software, mixed-reality development, and experimental evaluation. It also gave me valuable experience with scientific writing and teamwork, helping me grow as an engineer and preparing me for future research projects.

Acknowledgments

I would like to express my sincere gratitude to Professor Yoshida for welcoming me into his laboratory and offering me the opportunity to carry out this internship. I am especially thankful to Professor Sasaki, who supervised my work with constant guidance and encouragement throughout the project. I also wish to thank Romain Bornier, a former student, for putting me in contact with the laboratory. Finally, I am grateful to all the members of the lab for their kindness, for sharing their culture, and for giving me countless tips about daily life in Japan. Without them, this internship would have been far less enjoyable and rewarding.

Contents

Acronyms

AI Artificial Intelligence. 4

AIST National Institute of Advanced Industrial Science and Technology. 4

BL Baseline TCP condition. 18–20

CNRS Centre National de la Recherche Scientifique. 4

Cobots Collaborative Robots. 6

EE End Effector. 8

ICRA International Conference on Robotics and Automation. 5

IEEE Institute of Electrical and Electronics Engineers. 5

IRL Interactive Robotics Laboratory. 4, 5

JRL Joint Robotics Laboratory. 4, 5

MR Mixed Reality. 6–9, 11–13, 15, 19, 22

NASA-TLX NASA Task Load Index. 19, 20

OB Object-Based method. 18–20

pHRI physical Human–Robot Interaction. 6

ROS Robot Operating System. 8–10, 12, 13, 15, 16

RPY Roll–Pitch–Yaw. 8

SDK Software Development Kit. 8–10, 12, 14, 15

SII IEEE International Symposium on System Integration. 7, 20

SUS System Usability Scale. 19, 20

TCP Tool Center Point. 7, 13–15, 18

VCP Virtual Control Point. 13–15, 17–19

1 Presentation of the Laboratory

1.1 Identification and Context

My internship was hosted by the Interactive Robotics Laboratory (IRL) at Tokyo University of Science, on the Katsushika campus. The lab was founded recently, in 2022, and has grown quickly thanks to its strong positioning in humanoid and collaborative robotics. Although young, it already benefits from solid institutional support and has started to build a strong reputation abroad.

One of the main reasons for this visibility is its collaboration with the Centre National de la Recherche Scientifique (CNRS), the National Institute of Advanced Industrial Science and Technology (AIST) and the Joint Robotics Laboratory (JRL) in Tsukuba. This partnership gives the IRL access to a much larger research network, combining expertise from Japan and Europe, and provides a fertile ground for the exchange of methods and ideas.

1.2 Research Focus and Organization

Research at the IRL revolves around the idea of creating robots that are both precise and able to interact naturally with humans. Three areas structure the work: advanced control methods (either model-based or Artificial Intelligence (AI)-driven), the design of new force sensors for accurate manipulation, and autonomous systems able to anticipate human intentions.

The group itself is relatively small, about fifteen members, but very diverse. Students at different levels work side by side with postdoctoral researchers. The supervisors, Dr. Yoshida and Dr. Sasaki, coordinate both the scientific direction and daily activities. Weekly meetings ensure everyone can share their progress, but beyond formal gatherings, the atmosphere is friendly and supportive. With several francophones and other international students, cultural exchanges are a natural part of daily life in the lab.

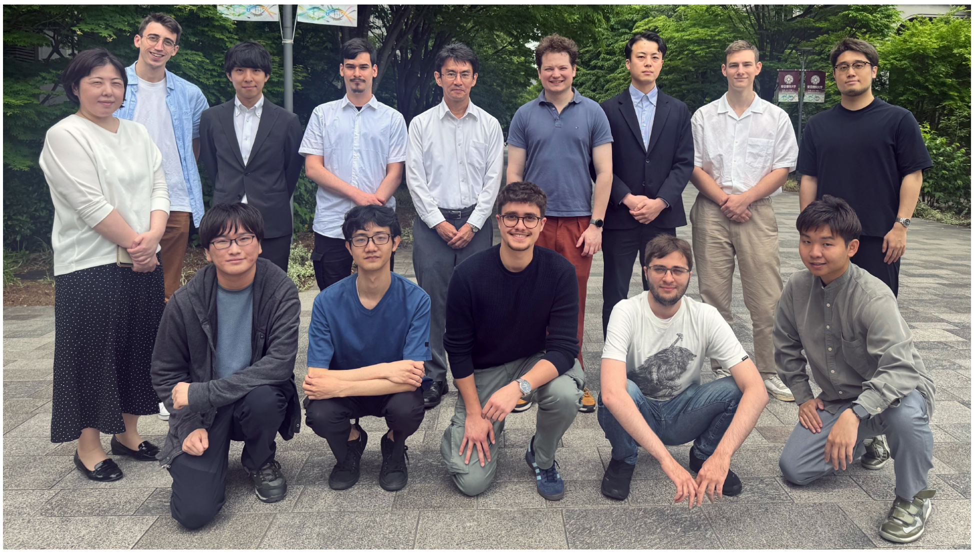


Figure 1: Summer 2025 Interactive Robotics Laboratory members

1.3 Missions and International Position

The laboratory's mission is to move from theory to practice by testing algorithms on real platforms. Among the robots used are Kawasaki's humanoid Kaleido and collaborative arms such as the UFactory and Franka Robotics systems. These experiments anchor the research in concrete applications and help connect fundamental robotics concepts with industrial needs.

From left to right: UFactory xArm6, Kawasaki Kaleido, and Franka Emika Panda.

Figure 2: Robots used at the IRL.

International collaboration is also central. The IRL maintains projects with teams in France and Canada, and its members are active in major conferences like the Institute of Electrical and Electronics Engineers (IEEE) and the International Conference on Robotics and Automation (ICRA). These activities strengthen its visibility on the global stage and provide opportunities for young researchers to present their work.

Looking ahead, the laboratory is determined to expand. It is actively seeking new students and doctoral candidates, from Japan as well as from abroad. Thanks to its close link with the JRL, it has the means to take part in ambitious projects on humanoid autonomy and human–robot cooperation. In just a few years, the IRL has positioned itself as a promising and fast-growing hub in robotics research.

2 Issues and Problem Definition of the Internship

2.1 Initial Definition and Expectations

At the beginning, the internship had a rather simple mission: to develop a framework allowing the xArm6 cobot to be controlled with a Mixed Reality (MR) headset controller. The goal was pragmatic: to provide a “ready-to-use” tool that could later serve as a starting point for another project if the opportunity appeared.

This task was expected to take around two months and included several implicit conditions. The framework had to be reactive — since control needed to be done in real time — and above all, clear, well-structured, and properly documented. The lab insisted on this point, because the value of the tool depended less on its technical sophistication than on how easily it could be reused by others. Behind this set of requirements, there was also another objective: to test my motivation and my ability to work independently, in a field that was partly new to me.

2.2 Scientific Context and Project Evolution

After about a month, the first phase was completed, and the direction of the project changed. To understand how this framework could be useful, many scientific articles on teleoperation and human–robot collaboration were reviewed.

These works showed that while teleoperation — controlling a robot from a distance — has been widely studied, its integration in physical Human–Robot Interaction (pHRI) scenarios is still relatively unexplored. Yet this is exactly where Collaborative Robots (Cobots) become relevant. Designed to work close to humans, they are intentionally less powerful than traditional industrial robots, mainly for safety reasons (see Fig. 3). Their role is therefore not to lift heavy loads by themselves, but they could be used to assist humans in handling objects that are bulky, heavy, or awkward to manipulate alone.

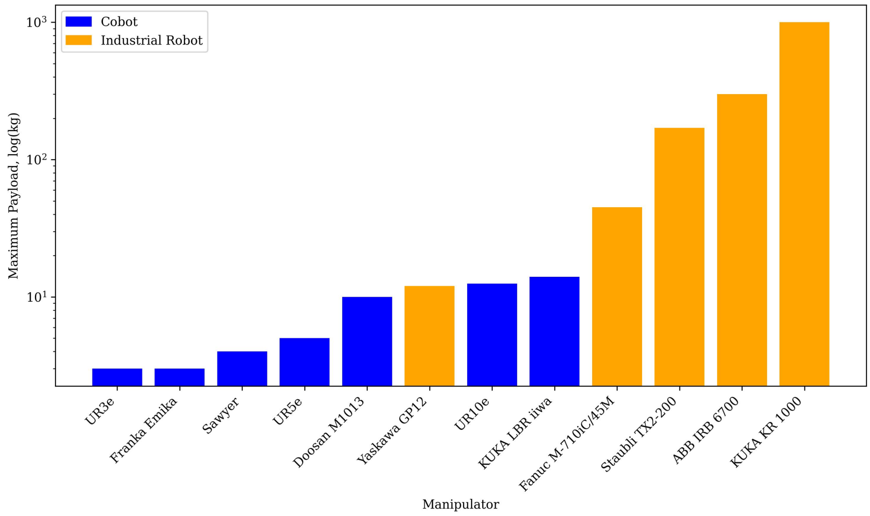


Figure 3: Comparison of different payload capacities.

However, Cobots are also far from being autonomous for these kinds of collaborative tasks; they cannot yet decide how to share the effort or coordinate naturally. For this reason, the concept of a team — combining a remote teleoperator, a cobot, and a local human providing most of the force — becomes essential. Yet, in such situations, the real challenge is now the way the teleoperator communicates the task. Most existing control approaches rely on the Tool Center Point (TCP), fixed at the end of the robot's tool. While robust, this method feels indirect and unintuitive when the goal is to orient or move a cumbersome object together with a human.

Thus, to support this vision, a different control strategy is needed. Instead of commanding the robot only through its TCP, the operator should be able to interact directly with the object. Shifting the control point onto the object itself makes the collaboration more natural and intuitive. After discussions with my supervisor, the project was therefore redefined: to design a MR interface that allows the robot's control point to be dynamically shifted to a virtual point directly placed on the 3D model of the manipulated object. This approach would enable the teleoperator to intuitively guide the robot's assistance, while the local human provides the physical strength.

This reorientation transformed the internship into a true research project. The new objective was not only to build a functional prototype but also to aim for a scientific contribution, with the prospect of submitting a paper to the international conference IEEE International Symposium on System Integration (SII) 2026 in Cancún.

2.3 Issues and Added Value

The issues were now very different from those imagined at the beginning. At first, the mission was not intended to bring visibility to the lab or to be part of a collaborative project. It was more about anticipating a possible future need and saving time for the postdoctoral researcher. With the reorientation, the project became fully aligned with the lab's main strategy: studying human–robot interactions. This time, external visibility was indeed at stake, through a possible scientific publication and through the public release of the code, accessible for other researchers to use.

For the lab, the main contributions were twofold: a concrete prototype ready to use, and a scientific basis opening the door to future research. For me, the experience was a turning point. It was necessary to go beyond an engineering mindset and adopt the logic of scientific research: reviewing literature, defining a problem, and learning how to write an academic paper. This step also provided the opportunity to contribute to another project on teleoperation of the xArm, which later led to inclusion as a co-author of a publication.

Finally, the lasting value of this internship lies in the fact that the work, even though functional, is still open. The framework works, but many aspects can still be improved — smoother control, more intuitive interaction, better ergonomics. This means the project does not end with my internship but can serve as a solid basis for future research.

3 Description of the Internship activities

3.1 Implementation of the Initial Control Framework

From the beginning, the project was designed with a modular architecture. This approach seemed the most practical, as it allows each functional block to be separated (MR input, data conversion, robot control) while keeping the possibility to reuse or replace them later. To reach this goal, Robot Operating System (ROS) 2 was chosen as middleware, since it provides a clear structure of nodes and topics and makes communication between different components easier, even if they are written in different languages.

A simple and progressive roadmap was then defined. The first step was to understand the robot Software Development Kit (SDK): what kind of commands it accepts and in which format. The second step was to explore Unity and determine what data is returned by the MR controllers of the HTC Vive Pro 2. Finally, the last step was to connect these two worlds — Unity and the robot SDK — using a ROS2 pipeline.

3.1.1 Understanding and Validating the xArm6 SDK

The SDK provided with the robot is a Python library that allows a WiFi connection to the xArm6 and the transmission of commands. The principle is simple: the robot expects a direct Cartesian position expressed in its local reference frame, with coordinates X, Y, Z in millimeters and Euler angles Roll–Pitch–Yaw (RPY) in degrees. With these six values, it moves its End Effector (EE) to the desired location. Besides, the gripper is controlled separately. The SDK takes a value between `-1` and `85`, which corresponds to the opening in millimeters: `-1` closes it fully, while a value close to `85` opens it completely.

The initial reasoning was straightforward: take the coordinates and orientation from the MR controller and send them directly to the SDK. For the gripper, a button on the controller could be used with three states: opening, closing, or static.

Based on this idea, a first ROS2 node was built with a clear interface: it takes as input a position X, Y, Z and R, P, Y angles, transmitted as `Twist` messages, and a simple `Float32` (`-1`, `0`, `1`) for the gripper control.

To validate this node, a second, simpler test node was implemented, which replaced MR inputs with keyboard commands. This step allowed checking several essential points:

- the **accuracy** of the movements,
- the **responsiveness** of the system (no visible delay),
- the correct execution of **translations, rotations, and gripper commands**.

This validation confirmed that the software foundation was reliable and that the SDK responded correctly to commands. With this base secured, the next step was to connect the framework with MR data through Unity.

3.1.2 Integration with Unity

To interface the robot with MR, Unity was selected, as it is the leading engine for MR environments. Originally designed for video games (interactive scenes, real-time physics, animation), Unity is now widely used in robotics for prototyping interfaces, simulating environments, and developing immersive applications.

Unity was a new tool for this project, and no ready-to-use documentation existed to send a MR controller state to ROS2 in real time. Many different resources (partial documentation, developer forums, sample code) were combined to learn how to collect the right data, process it, and publish it to ROS. Unity works wit

(a) Unity's Frame

(b) Robot's Frame

Figure 4: Comparison of coordinate frames.

As explained earlier, the SDK requires a Cartesian pose (X_r, Y_r, Z_r) in millimeters and Euler R, P, Y angles (in degrees). To convert Unity data to the robot's frame, both an axis permutation and a correction for handedness are needed. The final conversion is:

$$X_r = Z_u, \quad Y_r = -X_u, \quad Z_r = Y_u$$

For orientation, Unity provides quaternions (q_x, q_y, q_z, q_w). The robot SDK, however, expects Euler angles (ϕ, θ, ψ) in the R, P, Y convention. To avoid sign mistakes, the conversion was done in two steps:

1. build Unity's rotation matrix $R_u(q)$ from the quaternion,
2. apply the permutation matrix P and extract Euler angles from the resulting robot rotation matrix R_r:

$$P = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad R_r = P\,R_u$$

From $R_r = [r_{ij}]$, the R, P, Y angles are extracted (convention $R = R_z(\psi)\,R_y(\theta)\,R_x(\phi)$):

$$\phi = \operatorname{atan2}(r_{32},\, r_{33}), \quad \theta = -\arcsin(r_{31}), \quad \psi = \operatorname{atan2}(r_{21},\, r_{11})$$

where ϕ is the roll, θ the pitch, and ψ the yaw. These values are then converted to degrees for the SDK.

For the gripper, the HTC Vive Pro 2 controller includes a touchpad returning two float values: $(\hat{x}, \hat{y}) \in [-1, 1]^2$, representing the thumb's position. At each iteration, the value of $\hat{y}$, multiplied by a gain k, is accumulated to control the gripper opening (in mm), clipped to the robot's limits $[-1, 85]$:

$$\texttt{Opening}(t_{k+1}) = \texttt{clip}(\texttt{Opening}(t_k) + k \cdot \hat{y},\ -1,\ 85)$$

The resulting value is published to the ROS node `ee_pose_controller`, which calls the SDK functions to move the robot and control the gripper over WiFi.

With this architecture, the first framework was complete: one Unity→ROS node handling controller states, coordinate conversion (with permutation and minus sign), and gripper input, and one robot-side node communicating with the SDK.

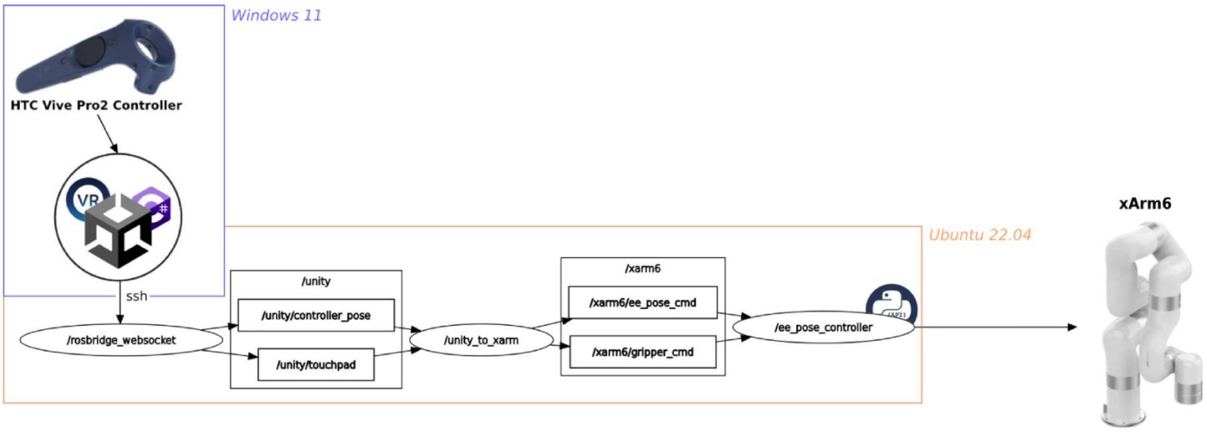


Figure 5: Pipeline of the first control framework

3.1.3 First Framework Limitations and Initial Tests

With both nodes connected, the first version of the framework was ready for testing. However, several important limitations quickly appeared.

The most significant issue came from the absolute mapping strategy. In this mode, the controller position in Unity was directly mapped to the robot's position in the real world. While simple, this approach forced the operator into uncomfortable postures: for example, moving "backwards" or staying constantly standing to cover the robot's full range of motion. Even more critical, when starting the program, the robot would immediately try to reach the initial position of the controller, without considering its environment. This behavior created a real risk of collision if the operator had not prepared the scene carefully.

Another difficulty was the calibration of offsets. To make the virtual and physical spaces correspond, manual adjustments were needed so that the operator, standing with their arm at rest, would match the robot's initial position. These offsets were sensitive to the room setup and had to be reconfigured each time the environment changed, making the solution not easily generalizable and not viable for long-term use.

To tackle those issues, Prof. Sasaki found a scientific paper describing an alternative control method based on relative motion.

3.1.4 Introduction of Relative Control

To address the limits of absolute mapping, a new strategy was explored. Instead of forcing the robot to track the absolute position of the MR controller, this method transmits only the translations and rotations of the hand if a Trigger button is pressed. Inspired by the *Robot Telekinesis* technique [1] , the operator holds one or two controllers to create a virtual plane in space, which is mirrored at the robot's end effector (see Fig. 6). When the user moves or rotates this plane, the robot follows these deltas in real time. A key advantage is the possibility of clutching: the user can deactivate control, change posture, then resume without disturbing the robot. This makes interaction more natural, reduces fatigue, and avoids unsafe initializations.

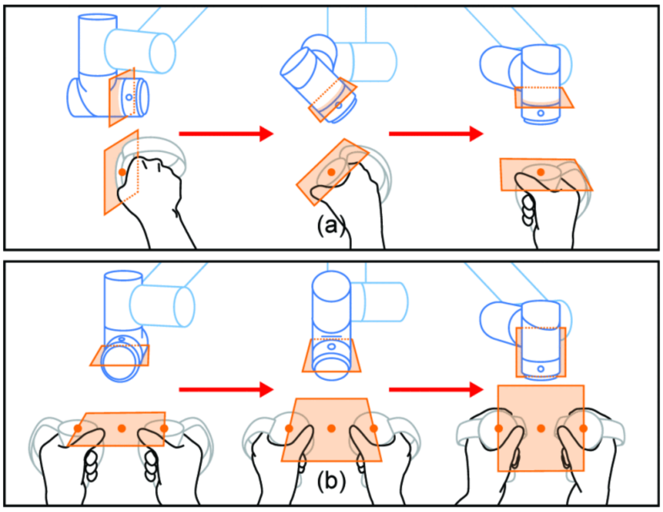


Figure 6: Illustration of the relative motion, with (a) one or (b) two hands.[1]

Thus, after considering the benefits of this method, the absolute motion was replaced with relative motion in about one week. At that point, the framework was close to completion, but one major issue remained: the lack of fluidity. With the HTC Vive Pro 2, the update rate was limited to around 5 Hz. Such latency made smooth interaction impossible and forced the operator to focus constantly to avoid mistakes. To address this, a new setup was tested with the Meta Quest 3, a standalone headset that performs its spatial computations without a PC. After adapting both the Unity scene and the code, the frequency rose to nearly 70 Hz, which represented a significant improvement.

In the end, the first framework reached its main objective: delivering a modular and documented solution capable of controlling the xArm6 in real time with a MR controller.

3.2 Research Context and Prior Work

As the first framework was completed, a new question arose: what could be done next? The system was working, but it was still only a tool. To find an interesting direction, scientific papers in robotics and teleoperation were explored. At first, this task was not easy. Articles are dense and sometimes difficult to connect to the project. Step by step, however, reading methods improved: starting with the abstract, introduction, and figures

---

[1] This illustration was extracted from [1].

to quickly determine if the work was relevant. It also became clear that each paper does not attempt to solve everything but focuses on one part of a larger problem. This approach made it easier to follow the ideas and understand how researchers build on each other's work.

One key discovery was the research on multi-arm manipulation. Many teams attempt to increase the payload of robots by using two or more arms simultaneously. This works in theory but reduces manipulability and makes the system harder to control. Instead of adding more robots, a different idea emerged: imagining a team composed of a robot and a human, where the

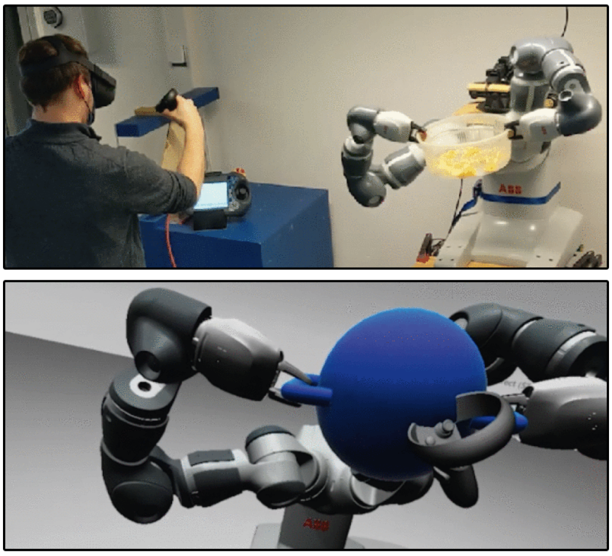


Figure 7: Control of a dual-arm YuMi via a MR interface.[2]

This idea seemed promising, but the limit was clear: robots are still far from autonomous enough to work like this alone. Therefore, we need a teleoperator to guide them. And for this teleoperator to be effective, we must provide both an interface and a control method. Looking deeper into the literature, it was found that object-based control and MR had been proposed, but they were rarely tested in real physical human–robot interaction tasks. This gap provided the direction for the next stage of the internship and for the related article.

3.3 Proposed Method

3.3.1 MR Interface Design

The goal of the MR interface was to create a representation of the robot and its environment that was as close as possible to reality. A key point was to keep a 1:1 scale for the robot, so that the operator could correctly feel the size and the reachable workspace. To avoid collisions, the closest objects around the robot were also modeled in the virtual scene, such as the table, the power box, and the computer screen (see Fig. 8).

To achieve a real-time digital twin, a publisher was added to the `xarm_ctrl` ROS2 node to send the six joint values and the gripper state from the SDK. A C# script in

---

[2] This illustration was extracted from [2].

Unity received these values and updated the 3D robot model continuously. Thanks to my previous experience with Unity–ROS connections, this step was completed in about one day. The result was a system where movements of the real robot were reflected almost instantly in the virtual scene, with a delay of only 10–20 ms, which felt imperceptible to the user.

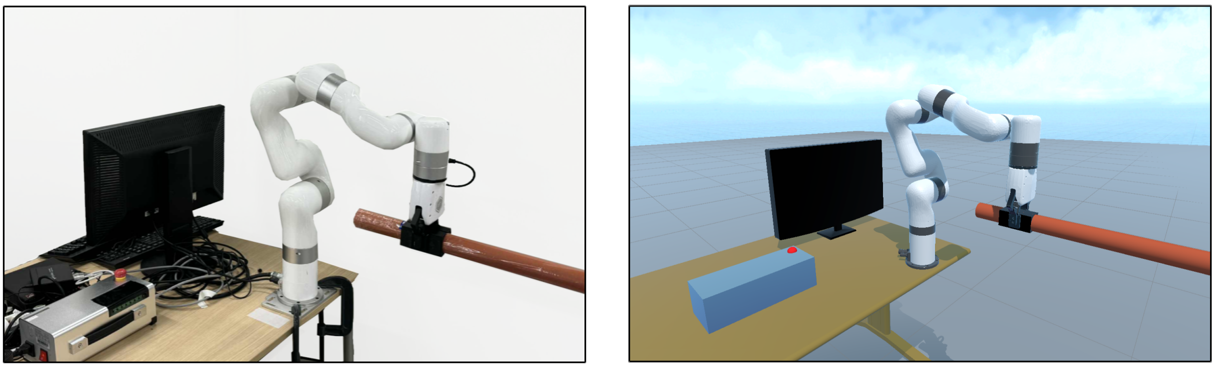


Figure 8: Comparison between the real and virtual robot manipulating an elongated object.

3.3.2 Object-Based Control Strategy

To go further, an object-based control strategy was implemented. The idea is that instead of directly commanding the TCP of the robot, the operator defines and manipulates a Virtual Control Point (VCP) located on the object itself.

Implementation in Unity. The first step was to design a way for the operator to select a VCP in the virtual scene. When the MR controller approaches the virtual object, its position is projected radially onto the object's surface. If the distance between the controller and the object is smaller than a predefined threshold, the system registers the closest point as the VCP (See Fig. 9). This mechanism establishes the offset between the robot's TCP and the chosen VCP. This logic was implemented in a C# script inside Unity. Once this was functional, a new Unity publisher was added that continuously sent the position and orientation of the VCP to ROS2.

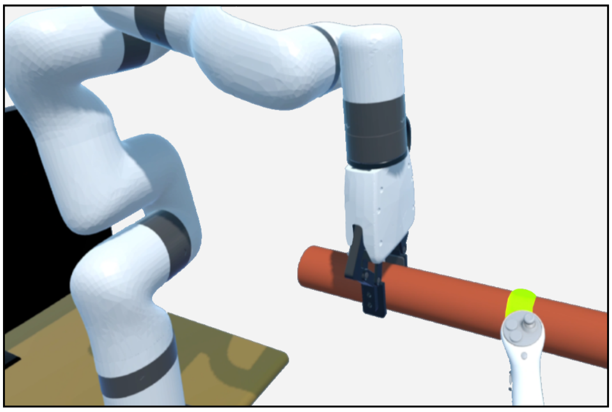


Figure 9: Selection of the Virtual Control Point.

Mathematical formulation. The SDK of the xArm6 accepts only the pose of the robot's TCP. However, the control method is now designed around the VCP defined on the manipulated object. The objective is therefore: given the desired pose of the VCP in the world frame and its offset to the TCP, compute the corresponding pose of the TCP that will realize this motion.

Let us define the TCP A and the VCP M:

$$\boldsymbol{T}_A^w = \begin{bmatrix} \boldsymbol{R}_A & \boldsymbol{p}_A^w \\ \boldsymbol{0}^\top & 1 \end{bmatrix}$$

$$\boldsymbol{T}_M^w = \begin{bmatrix} \boldsymbol{R}_M & \boldsymbol{p}_M^w \\ \boldsymbol{0}^\top & 1 \end{bmatrix}$$

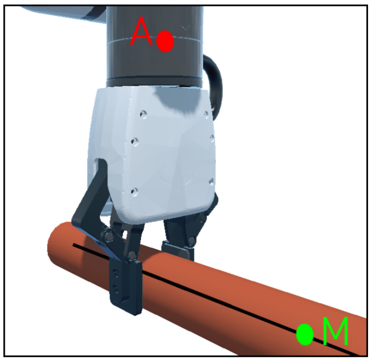


Figure 10: Geometric relation of A and M.

Here $\boldsymbol{T}_A^w$ is the TCP pose in the world frame $\{w\}$, while $\boldsymbol{T}_M^w$ is the desired VCP pose in the world frame. The relative pose of M with respect to A is expressed as

$$\boldsymbol{T}_M^A = \begin{bmatrix} \boldsymbol{R}_{AM} & \boldsymbol{p}_{AM}^A \\ \boldsymbol{0}^\top & 1 \end{bmatrix}.$$

From rigid-body kinematics, the relation between VCP and TCP is

$$\boldsymbol{T}_M^w = \boldsymbol{T}_A^w \boldsymbol{T}_M^A.$$

Solving for $\boldsymbol{T}_A^w$ gives

$$\boldsymbol{T}_A^w = \boldsymbol{T}_M^w \left(\boldsymbol{T}_M^A\right)^{-1}.$$

Since the object is rigidly grasped by the robot, the TCP and the VCP always share the same orientation. i.e. $\boldsymbol{R}_A = \boldsymbol{R}_M = \boldsymbol{R} \Leftrightarrow \boldsymbol{R}_{AM} = \boldsymbol{I}$

$$\boldsymbol{T}_M^A = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{p}_{AM}^A \\ 0 & 1 \end{bmatrix} \quad \Leftrightarrow \quad \left(\boldsymbol{T}_M^A\right)^{-1} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{p}_{AM}^A \\ 0 & 1 \end{bmatrix}.$$

The TCP position can then be written as

$$\boxed{\boldsymbol{p}_A^w = \boldsymbol{p}_M^w - \boldsymbol{R}\,\boldsymbol{p}_{AM}^A}$$

This means that: knowing the desired VCP pose $(\boldsymbol{p}_M^w, \boldsymbol{R})$ and the offset $\boldsymbol{p}_{AM}^A$, we can compute the corresponding TCP pose $(\boldsymbol{p}_A^w, \boldsymbol{R})$, which is then sent to the robot's inverse kinematics solver through the SDK.

3.3.3 Final Framework

At this stage, all the components were integrated into a complete framework (see Fig. 11). It is divided into two main parts: the remote side, where the operator interacts with the MR interface, and the local side, where the robot executes the commands. Both sides are connected through ROS2 messages over a wireless network.

On the remote side, the user wears the *Meta Quest 3* headset and manipulates the MR controller. Unity runs on a Windows 11 PC and generates the MR scene. Two types of data are produced:

- the **controller position** and trigger state,
- the **Virtual Control Point position**, computed by Unity when the trigger is pressed near the object.

These data are published as ROS2 messages and transmitted to the local side.

On the local side, the ROS2 framework is organized into three main functional blocks:

- **Unity Data Interpreter:** This node subscribes to the incoming Unity messages and converts the controller and VCP positions into usable coordinates in the robot frame.
- **Next Position Solver:** This node calculates the aimed robot state. In the case of object-based control, it applies the mathematical transformation to derive the corresponding TCP pose from the VCP pose.
- **xArm Controller:** This node communicates directly with the Python xArm SDK, and sends the final joint commands to the robot in real time.

The architecture is fully modular: each ROS2 node has a well-defined role and exchanges information only through messages. This means that Unity or the robot can be replaced with minimal changes, as long as the messages respect the same structure. The digital twin is continuously updated because the current robot state is also sent back from the SDK to Unity, ensuring that the operator always sees the virtual copy of the robot.

3.3.4 Technical Evaluation

The purpose of the technical evaluation was twofold: first, to verify that the proposed framework can be applied to the collaborative manipulation of objects that exceed the robot's maximum payload; and second, to confirm that the VCP method, together with the mathematical formulation developed earlier, correctly translates the desired VCP pose into the expected TCP motion. To test this, we used an aluminum ladder with a mass of 6.89 kg, heavier than the nominal payload of the xArm6 (5 kg). In the experiment, the robot grasped one extremity of the ladder while a human operator held the other side, representing the targeted use case: the robot alone cannot carry the full load, but it can share the effort with a person.

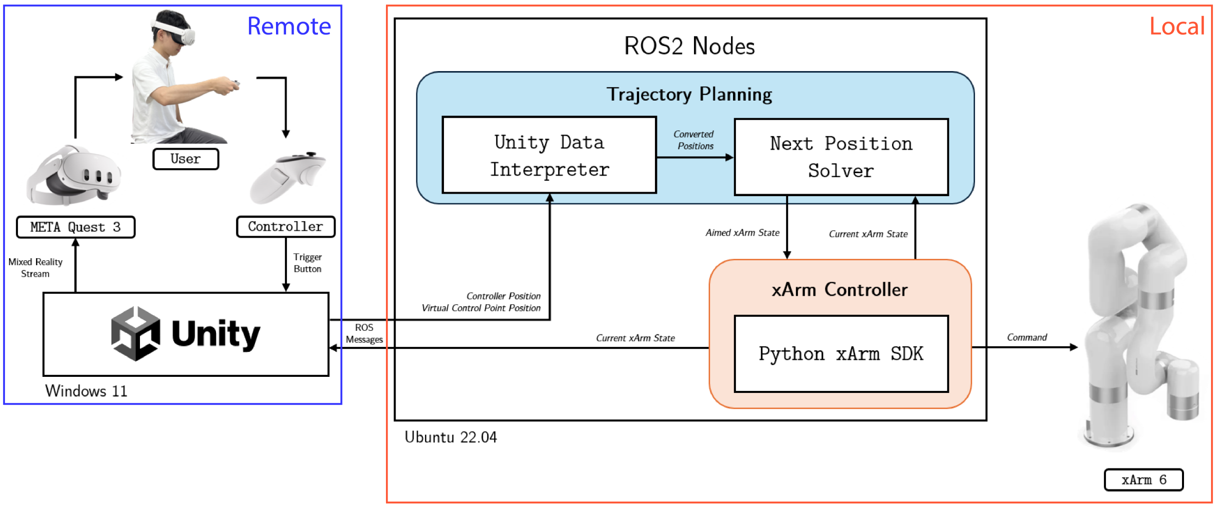


Figure 11: Final framework architecture, showing the communication between Unity, ROS2 nodes, and the xArm6 robot.

Experimental Setup. To record the motion of the ladder, three reflective markers were attached close to its geometric center. A motion capture system (OptiTrack V120 Trio) tracked the markers in real time (see Fig. 12). Only the motion of the ladder was recorded: the individual trajectories of the robot and the human were not measured. During the tests, the robot executed predefined trajectories, while the human followed passively to support and balance the object. The environment was kept free of obstacles and workspace constraints to simplify the analysis.

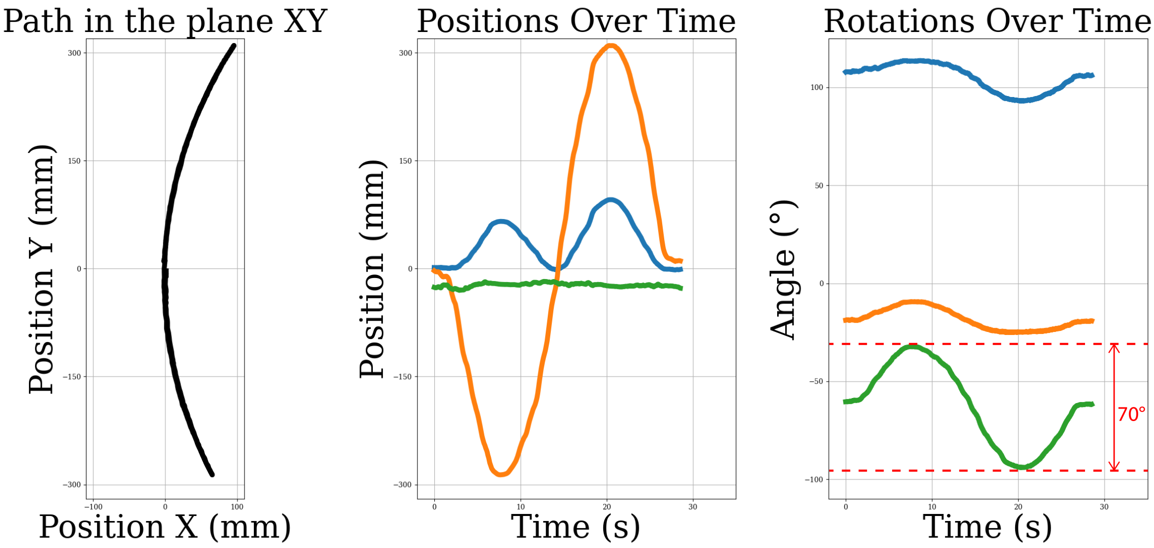
Figure 12: Configuration of the motion capture system.

Experimental Protocol. Three different rotation tasks around the vertical Z-axis were tested to study how the system behaves under different pivot constraints:

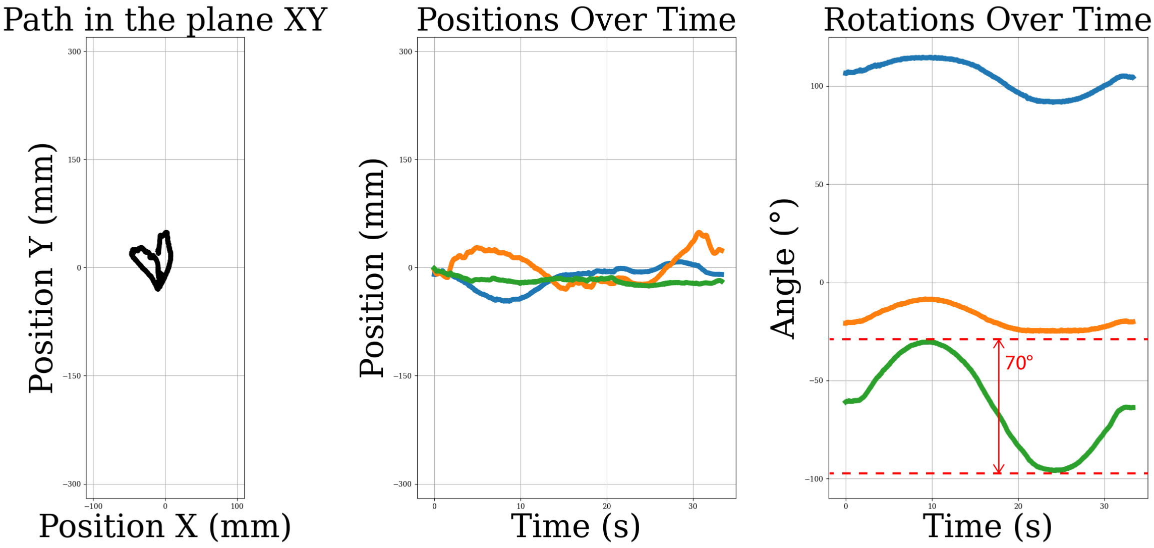
- **Motion 1:** Rotation of 35° around the point grasped by the robot.
- **Motion 2:** Rotation of 35° around the point where the motion capture markers were located.
- **Motion 3:** Rotation of 20° around the point held by the human operator. (The angle was limited by the range of the xArm6.)

In practice, this means that the same rotational input was applied, but with three different choices of VCP. At rest, we also measured the share of the ladder's weight supported by the human operator. Using a scale, it was 3.70 kg, which means the robot was holding roughly half of the total mass.

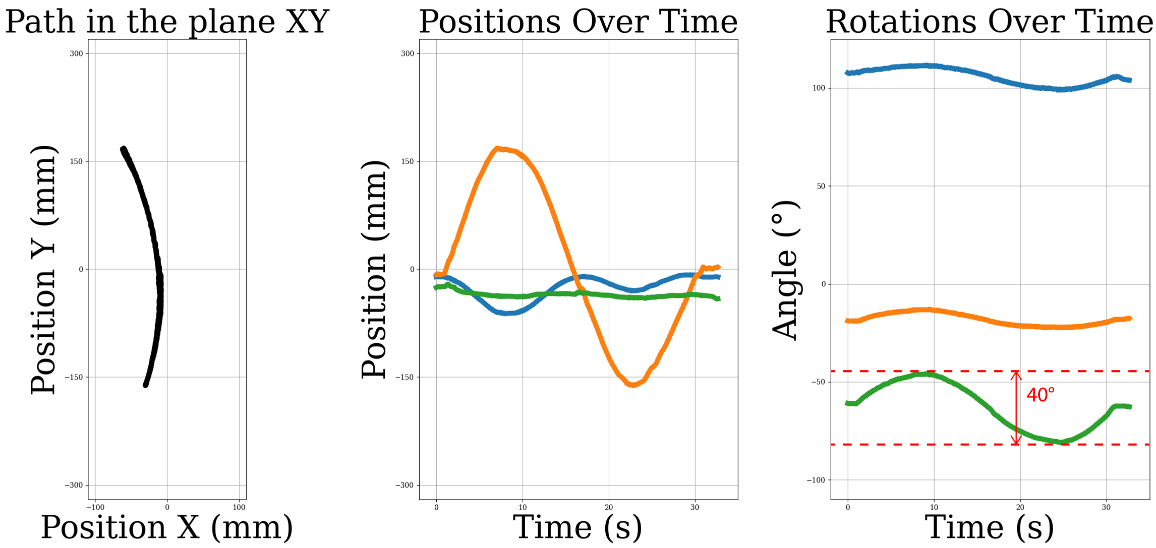
Results. The results of the motion capture are shown below. Each graph presents the XY trajectory of the ladder, as well as the time evolution of the position and the rotation.



Motion 1.



Motion 2.



Motion 3.

Figure 13: Results from the three motions (Blue : X-axis, Orange : Y-axis, Green : Z-axis).

For Motion 1 and Motion 3, where the ladder was rotated around one extremity, the XY trajectory followed a smooth arc and the main rotational component was clearly along the Z-axis. For Motion 2, where the rotation was defined around the middle of the ladder, the data showed almost no displacement and a dominant rotation along the Z-axis. This confirmed that the marker position was a good approximation of the rotational center.

Most importantly, the amplitude of the measured rotations confirmed that the system reproduced the commanded VCP motions: a 35° input corresponded to a measured rotation of 70° on the graphs, and a 20° input corresponded to 40°. This validated that the framework correctly translated the VCP commands into robot motions, even in the collaborative setup with a human operator.

The trajectories were not perfectly clean, since the ladder was not rigidly fixed to the robot and the human operator could slightly bias the motion. Still, the results demonstrated that the framework was capable of producing different and well-controlled pivoting behaviors depending on the chosen VCP. This confirmed both the feasibility of the proposed method and its potential for heavy-object co-manipulation.

3.3.5 User Study

After the technical evaluation showed that the framework could reproduce the expected VCP motions with a heavy object, the next step was to study how real users would experience the system. The goal of this user study was therefore to evaluate the usability of the object-based control method in comparison with a more classical baseline, where the robot is controlled directly through its TCP.

Procedure and Task. Each participant tested both control methods: the Baseline TCP condition (BL) and Object-Based method (OB). To avoid learning or fatigue effects, the order of presentation was balanced (some started with BL, others with OB). Before the trials, participants received a short tutorial and a 5-minute practice phase for each method. The actual task consisted of aligning a physical object, held by the robot, with virtual target poses displayed in the MR headset. A human assistant held the other extremity of the object only to counterbalance its weight, but without guiding the motion (see Fig. 14). Participants then performed twelve placement tasks per condition, with the same fixed sequence of target poses to ensure consistency.

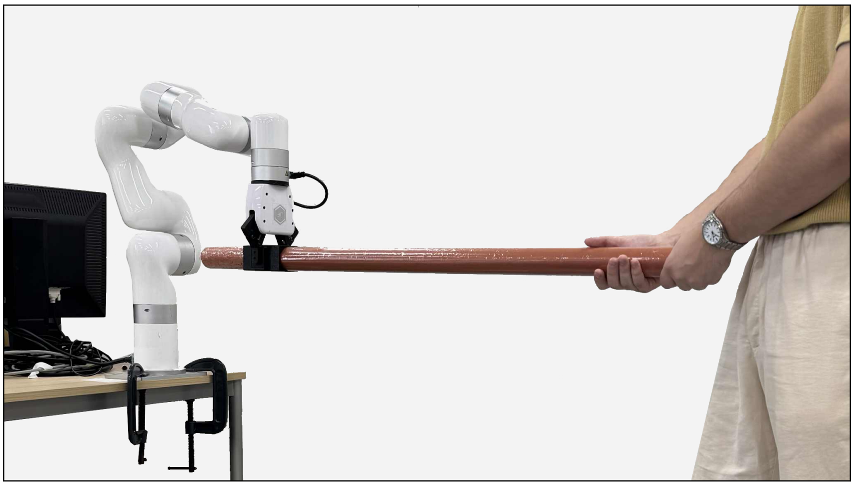


Figure 14: Evaluation setup with a local user passively holding the object.

Measurement. Several types of data were collected during the sessions:

- Object position and orientation, recorded continuously.
- Log events such as trigger presses and control point selections.
- Task completion time, measured as the interval between the appearance of a target pose and the moment the alignment was achieved.

In addition, after each condition participants filled out two questionnaires: the NASA Task Load Index (NASA-TLX) [3], which measures perceived workload across several dimensions, and the System Usability Scale (SUS) [4], a standardized scale to evaluate usability.

Results. Seven volunteers from Tokyo University of Science participated in the study. All were healthy adult males and had no prior experience with robot teleoperation.

Task performance. The OB method produced longer completion times on average (197.33 s ± 25.60) compared to the BL method (171.65 s ± 43.99), an increase of 15%. This can be explained by the fact that OB sometimes required participants to move around physically, for example standing up to reach the object and activate a new VCP. In contrast, the BL method used a single fixed control point, so users could remain seated. Trigger usage was also higher in OB (50 ± 15 activations) compared to BL (45 ± 4), which can be explained by the infinite number of possible control points in OB, leading participants to fine-tune their inputs for greater precision.

Subjective workload and usability. The NASA-TLX results showed that perceived workload remained low in both cases, with average scores below the midpoint of the scale. However, the SUS clearly favored the OB method: participants rated it at an average of 84 ± 12.71 out of 100, compared to 74 ± 16.85 for BL (see Fig. 15). Both scores are above the commonly accepted usability threshold of 68, but OB was consistently judged as more intuitive and pleasant to use.

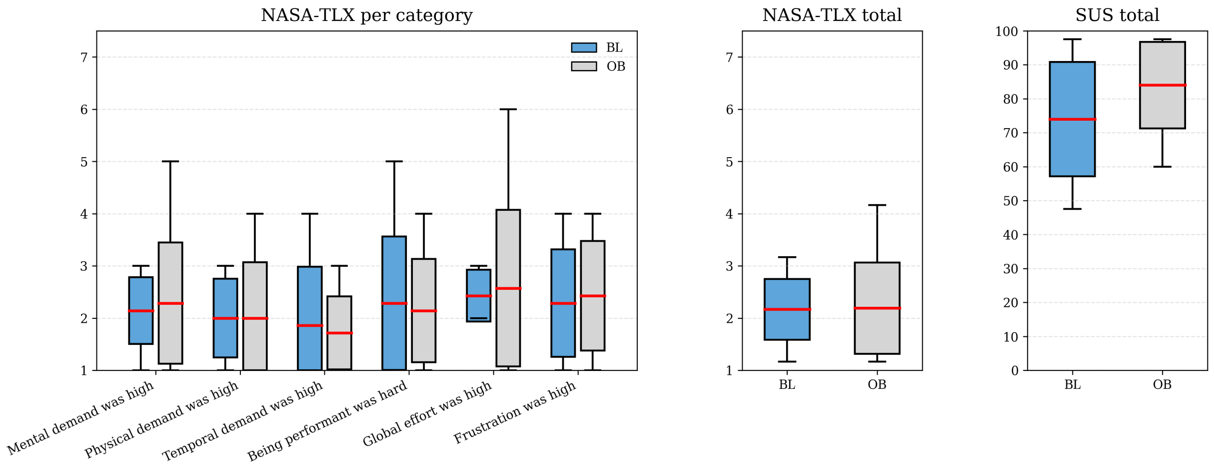


Figure 15: Comparison of Workload (NASA-TLX) and Usability (NASA-TLX) between BL and OB

Overall, these results showed that while OB requires slightly more time and effort, it was perceived as more usable and offered greater flexibility for collaborative manipulation tasks. This confirmed the potential of the object-based approach for practical teleoperation scenarios.

3.4 First Experience with Scientific Writing

When the project took a new direction, the supervisor suggested trying to publish the results at the *SII 2026* conference. This represented a completely new challenge, as it was the first time writing a scientific article. At the beginning, it was not clear how to start. Unlike technical reports, papers are not only about describing what was done, but also about explaining why it matters, connecting it to previous research, and showing the novelty.

The first step was to build the structure of the paper. Following the supervisor's advice, the classic format was used: introduction, method, experiments, results, and conclusion. Even if this structure looks simple, each section has its own style and difficulty. The introduction must be clear and motivating; the method must be precise without being too heavy; and the results should be presented in a way that convinces the reader. Writing became a continuous cycle of review and feedback. A first version was drafted, then corrected by the supervisor, often with requests to be more precise or to better explain the motivation. Sometimes the text seemed clear, but for an external reader it was not, which required rethinking how ideas were expressed.

Looking back, this first experience with scientific writing was one of the most valuable parts of the internship. It showed how research is shared, how ideas are debated, and how to communicate results in a professional way. Even if the paper is still not published, there is now greater confidence to contribute to other articles in the future.

4 Conclusion

This internship represented a real step into the world of research. It started with a simple technical goal, but the project quickly evolved into a broader challenge about how humans and robots can cooperate. Adaptation was required, along with learning new tools and viewing the work from a much larger perspective.

For the laboratory, the contribution was not only a functional control framework but also a starting point for future studies on MR and object-based control. For the author, the experience was transformative. It provided skills to analyze scientific literature, design and test experiments, and write in a more rigorous way. It also revealed that research is not only about solving problems but about discussing ideas, receiving feedback, and improving step by step. This mindset, very different from pure engineering, fostered growth in both confidence and independence.

Looking ahead, this experience confirmed a strong interest in continuing research. It provided the motivation to carry out the graduation project in a research laboratory and to possibly pursue further studies toward a PhD.

Appendix

Framework Repository

The full framework developed during this project is available on GitHub at the following address: `https://github.com/bourinto/Object-Based-Teleoperation-Interface`

Demonstration Video

A short video presenting both the demonstration and the technical evaluation can be accessed at: `https://youtu.be/zhlKmO18nR8`

References

[1] J. H. Lee, Y. Kim, S.-G. An, and S.-H. Bae, "Robot telekinesis: Application of a unimanual and bimanual object manipulation technique to robot control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9866–9872.

[2] F. Kennel-Maushart, R. Poranne, and S. Coros, "Multi-arm payload manipulation via mixed reality," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 11 251–11 257.

[3] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of empirical and theoretical research," in *Advances in Psychology*, P. A. Hancock and N. Meshkati, Eds. North-Holland, Jan. 1988, vol. 52, pp. 139–183.

[4] J. R. Lewis and J. Sauro, "Item benchmarks for the system usability scale," *Journal of Usability Studies*, vol. 13, no. 3, pp. 158–167, May 2018.

ENSTA Bretagne

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :
At the end of the internship, please return this report via mail or email to:

ENSTA Bretagne – *Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE*
☎ 00.33 (0) 2.98.34.87.70 / stages@ensta-bretagne.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name* Tokyo University of Science

Adresse / *Address* 6-3-1, Niijuku, Katsusika-ku, Tokyo 125-8585 Japan

Tél / *Phone (including country and area code)* +81-3-5876-1734

Nom du superviseur / *Name of internship supervisor* Eiichi YOSHIDA

Fonction / *Function* Professor

Adresse e-mail / *E-mail address* eiichi.yoshida@rs.tus.ac.jp

Nom du stagiaire accueilli / *Name of intern* Toméo BOURIN

II - EVALUATION / *ASSESSMENT*

Veuillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
Please attribute a mark from ***A (excellent)*** *to* ***F (very weak).***

MISSION / *TASK*

- La mission de départ a-t-elle été remplie ? *Was the initial contract carried out to your satisfaction?* [✓ A] B C D E F

- Manquait-il au stagiaire des connaissances ? *Was the intern lacking skills?* ☐ oui/*yes* ☒ non/*no*

  Si oui, lesquelles ? / *If so, which skills?*

ESPRIT D'EQUIPE / *TEAM SPIRIT*

- Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / *Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)* [✓ A] B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* The student was very conscientious.

COMPORTEMENT AU TRAVAIL / *BEHAVIOUR TOWARDS WORK*

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?

Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?

[✓ A] B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* The student actively participated in the work and was eager to learn new things.

INITIATIVE – AUTONOMIE / *INITIATIVE – AUTONOMY*

Le stagiaire s'est –il rapidement adapté à de nouvelles situations ? [✓ A] B C D E F
(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

Did the intern adapt well to new situations? [✓ A] B C D E F
(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* The student autonomously handled various problems they encountered.

CULTUREL – COMMUNICATION / *CULTURAL – COMMUNICATION*

Le stagiaire était-il ouvert, d'une manière générale, à la communication ? [✓ A] B C D E F
Was the intern open to listening and expressing himself /herself?

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* The student was able to communicate with his colleagues and work collaboratively.

OPINION GLOBALE / *OVERALL ASSESSMENT*

❖ La valeur technique du stagiaire était : [✓ A] B C D E F
Please evaluate the technical skills of the intern:

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?
Would you be willing to host another intern next year? ☒ oui/*yes* ☐ non/*no*

Fait à ______, le ______
In Tokyo, on September 4, 2025

Signature Entreprise 吉田英一 Signature stagiaire
Company stamp ______ *Intern's signature*

Merci pour votre coopération
We thank you very much for your cooperation