Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina 2026

# Bilateral haptic teleoperation bridging YARP and ROS 2 ecosystems

Santos-García, Á.*, Łukawski, B., Copaci, D., Victores, J. G., Balaguer, C.

*RoboticsLab, Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911, Leganés, Spain.*

**Resumen**

Trabajos previos establecieron una arquitectura de teleoperación para brazos robóticos sobre YARP y ROS 2. El presente trabajo extiende dicha arquitectura incorporando retroalimentación háptica para cerrar un bucle de control bilateral. Un dispositivo Phantom Omni transmite comandos de posición al robot y recibe fuerzas de contacto del entorno remoto. La integración comprende tres capas: YARP para la abstracción hardware del dispositivo háptico, ROS 2 como backbone de comunicación, y una capa de control específica por plataforma, que emplea la pila de control cartesiano para el robot humanoide TEO y el protocolo EGM de ABB para el brazo colaborativo GoFa. Para evitar calibración espacial explícita entre marcos de referencia, se adopta una estrategia de control diferencial basada en desplazamientos incrementales. La retroalimentación de fuerzas se implementa mediante un nodo dedicado que evalúa la penetración del extremo del robot en regiones de restricción virtual y genera fuerzas de restitución proporcionales, cerrando el bucle bilateral de forma independiente a la plataforma objetivo.

*Palabras clave:* Sistemas de control de movimiento, Percepción y sensorización, Tecnologías robóticas, Manipuladores robóticos, Telerrobótica

**Abstract**

Previous works from the group established a teleoperation architecture for robot arm manipulators built on YARP and ROS 2. The present work extends that architecture by incorporating haptic feedback to close a bilateral control loop. A Phantom Omni device serves as the operator interface, transmitting position commands to the robot while rendering contact forces from the remote environment. The integration is structured into three layers: YARP for hardware abstraction of the haptic device, ROS 2 as the communication backbone, and a platform-specific control layer employing the Cartesian control stack for the TEO humanoid robot and ABB's EGM protocol for the GoFa collaborative arm. To avoid the need for explicit spatial calibration between coordinate frames, a differential control strategy based on incremental displacements is adopted. Force feedback is implemented through a dedicated node that evaluates end-effector penetration into virtual constraint regions and generates proportional restoring forces, closing the bilateral loop independently of the target platform.

*Keywords:* Motion control systems, Perception and sensing, Robotics technology, Robots manipulators, Telerobotics

## 1. Introduction

Robot teleoperation has historically been a challenge for control engineering. The main difficulty lies in achieving a balance between two opposing objectives, being one the controller and the device to be commanded. The information provided by sensors and/or cameras does not provide enough information to command the device, as the absence of physical interaction cues forces the operator to rely solely on indirect observations. Even more so when the environment of the robot and the operator differ greatly.

This is where haptic feedback becomes of great importance. Providing the operator with information about the forces exerted by the robot in the remote environment allows for more intuitive and precise control, from the detection of contacts and obstacles to the perception of resistances and inertias. However, incorporating this type of feedback is not trivial, as it can compromise the stability of the system.

In this context, a system has been developed using YARP for hardware abstraction and ROS 2 for communication between the haptic device and the robot, with nodes implemented in C++ for TEO and in Python for the ABB GoFa.

*Corresponding author: alvarosang2003@gmail.com

The remainder of this document is organized as follows. Section 2 reviews the existing work related to the present topic. Section 3 introduces the hardware and software components used, while section 4 goes deeper into their implementation. Section 5 presents the experimental validation carried out on both robot platforms, combining simulated and real environments, and section 6 gathers the main conclusions. Finally, the acknowledgments and references are included at the end of the document.

## 2. Background

Robot teleoperation refers to the remote operation of a robotic system by a human operator through a master device that maps the operator's motion onto the slave robot. While early systems relied on position tracking and visual feedback as the primary means of situational awareness, these proved insufficient for tasks involving physical interaction with the environment (Hokayem and Spong, 2006). This motivated the incorporation of haptic feedback to transmit interaction forces back to the operator, enabling more intuitive and precise manipulation. Teleoperation systems have been explored with a variety of master devices, from low-cost generic controllers to dedicated haptic interfaces capable of rendering rich force information, each presenting different trade-offs between cost and feedback fidelity (Łukawski et al., 2023; Calzada et al., 2024).

The challenge of maintaining stability while transmitting force feedback has driven extensive research on bilateral control architectures, where passivity-based approaches and impedance control have emerged as dominant paradigms due to their theoretical guarantees in the presence of communication delays (Tian et al., 2025). The Geomagic Touch device, formerly known as PHANToM Omni, has become a standard haptic platform in this context (Silva et al., 2009). Its use has been demonstrated in bimanual humanoid teleoperation configurations as well as in more demanding scenarios involving legged manipulators with model-predictive whole-body controllers, where the trade-off between update rate and haptic transparency becomes critical (Cheng et al., 2022).

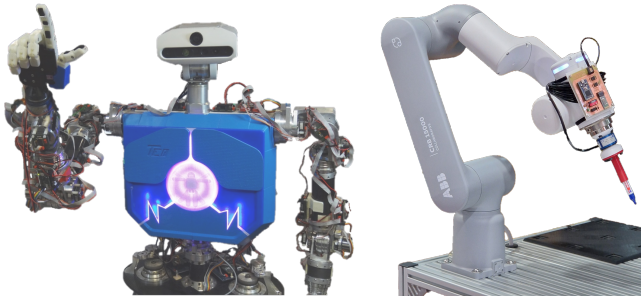
From the software perspective, YARP has been extensively adopted in humanoid robotics for its distributed and hardware-agnostic design, while ROS and its successor ROS 2 have become dominant in the broader robotics community (Metta et al., 2006; Quigley et al., 2009). Bridging both ecosystems enables the reuse of existing YARP-based device interfaces alongside modern ROS 2 components, complemented with virtual reality interfaces that extend the operator's situational awareness beyond what force feedback alone can provide (Łukawski et al., 2025b,a). At the robot controller level, protocols such as ABB's Externally Guided Motion (EGM) enable low-latency real-time Cartesian streaming, providing the tight control loops required for transparent haptic teleoperation (Łukawski et al., 2026).

Prior work from our group has also explored cognitive robot skill learning and haptic rendering techniques for compliant constraints in simulation (Fernandez-Fernandez et al., 2023; Montesino et al., 2025), providing further motivation for the haptic integration presented here.

## 3. Materials

### 3.1. Robots

TEO is a full-sized humanoid robot platform developed at Universidad Carlos III de Madrid (Pérez Martínez et al., 2010). With 28 degrees of freedom distributed across its limbs, torso and neck, together with modular grippers and a sensor suite including force-torque, inertial and RGBD sensors, it has been applied to tasks ranging from bimanual manipulation to human-robot interaction. In this work, either of its right 6-DoF arms serves as the primary manipulation platform. The second robot is the ABB GoFa CRB 15000-5, a commercial 6-DoF collaborative arm whose joint-level torque sensors enable collision detection, while an onboard safety controller continuously monitors velocity and applied forces at the tool center point. Both platforms are shown in Figure 1.



(a) TEO humanoid (b) GoFa collaborative robot

Figure 1: Robot platforms.

### 3.2. Tools

Two tools are used in the experimental setup. The first is a custom pen-shaped end-effector instrumented with a JR3 force-torque sensor (Łukawski et al., 2026), providing six-axis wrench measurements during contact. The second is the Phantom Omni, a desktop haptic device that delivers three-axis force feedback to the operator through a stylus-based interface, enabling the user to perceive interaction forces while issuing motion commands. Both are depicted in Figure 2.

(a) Pen tool (b) Phantom Omni haptic device

Figure 2: Tools.

## 4. Proposed architecture

### *4.1. YARP, ROS 2 and EGM-based controllers*

The proposed architecture consists of three middleware layers that separate hardware abstraction, communication, and real-time control, as illustrated in Fig 3.

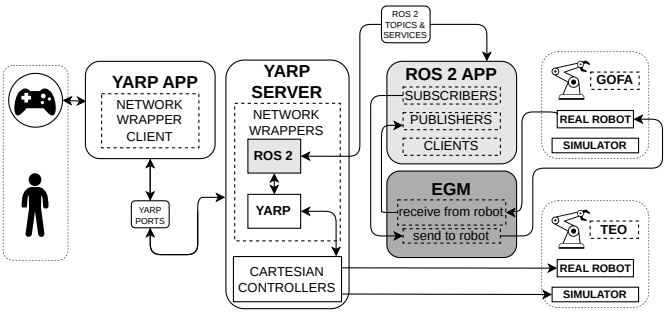



Figure 3: Proposed system architecture.

At the lowest layer, YARP (Yet Another Robot Platform) handles hardware abstraction and device management (Metta et al., 2006). Following a distributed architecture based on modular components called *devices*, YARP encapsulates specific hardware functionalities behind standardized C++ interfaces. In this work, a dedicated YARP device driver interfaces the Geomagic Driver (Phantom Omni) haptic device through its native library, exposing position and orientation readings while accepting force commands to be rendered back to the operator.

At the intermediate layer, ROS 2 serves as the communication backbone between the haptic interface and the different robot platforms (Quigley et al., 2009). ROS 2 structures distributed robotic applications into independent processes known as *nodes*, which exchange typed messages over *topics* (publish/-subscribe) or through *services* (request/response). A YARP–ROS 2 bridge translates the methods into standard message types such as `geometry_msgs/Pose`, exposing the haptic device as a topic while streaming force feedback through the corresponding topics.

At the highest layer, the architecture branches depending on the target robot platform. For TEO, ROS 2 cartesian nodes backed by YARP drivers receive the incoming pose commands. For the ABB GoFa, a dedicated ROS 2 package leverages the Externally Guided Motion (EGM) interface, which enables low-latency real-time cartesian streaming at the control level.[1] In both cases, the interaction forces measured at the end effector are routed back through the same communication pipeline to be rendered on the haptic device, closing the bilateral control loop.

### *4.2. YARP haptic sensor interface*

The haptic sensor interface is implemented as a set of YARP devices following the standard client-wrapper-driver pattern (Metta et al., 2006), as illustrated in Fig. 4. The GeomagicDriver wraps the native OpenHaptics library of the Geomagic Touch device, running an asynchronous scheduler callback that reads stylus position, gimbal orientation and button states, while applying Cartesian forces or joint torques to the actuators. The `HapticDeviceWrapper` runs as a periodic thread that publishes the device state and listens for incoming force feedback over YARP ports. Finally, the HapticDeviceClient provides a transparent remote interface, connecting to the wrapper through state, feedback and RPC ports, and exposing the `IHapticDevice` interface to any YARP application.[2]

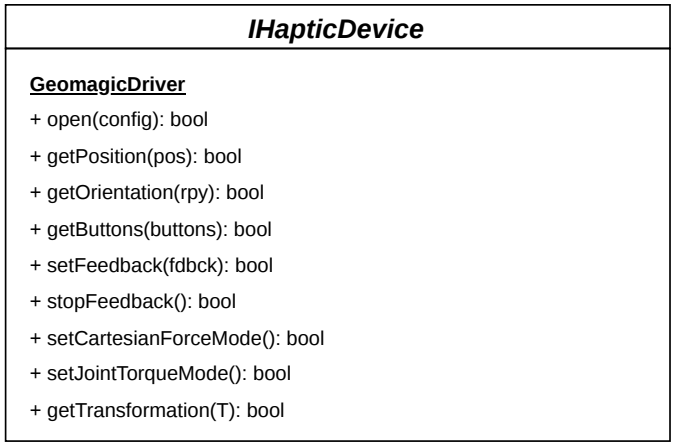



Figure 4: Unified Modeling Language (UML) class diagram of the haptic sensor interface.

### *4.3. YARP–ROS 2 bridge*

The `IHapticDevice` interface is exposed over ROS 2 through a Network Wrapper Server (NWS) that runs as a periodic thread, publishing the device state at a fixed rate and accepting force feedback commands through a subscription callback.[3] The NWS also exposes a service interface to manage force mode, stop feedback, query the current mode, get/set transformations, and query maximum force. Table 1 summarizes the full mapping.

Table 1: `IHapticDevice` mapped to ROS 2 interfaces.

| topic / service | message type | role |
|---|---|---|
| `state/pose` | Pose | publisher |
| `state/buttons` | Int32MultiArray | |
| `state/force_feedback` | Wrench | |
| `state/transform` | Transform | |
| `feedback` | JointState | subscriber |
| `set_force_mode` | SetBool | service |
| `stop_feedback` | Trigger | |
| `state/force_mode` | Trigger | |
| `state/max_force_feedback` | GetMaxFeedback | |
| `state/transform` | GetTransformation | |
| `set_transformation` | SetTransformation | |

[1] `https://github.com/FLo-ABB/ABB-EGM-Python`
[2] `https://github.com/robotology/yarp-devices-haptic`
[3] `https://github.com/robotology/yarp-devices-ros2`

### 4.4. Haptic controller application

The proposed controller architecture and the previous components are materialized in the overall scheme presented in Figure 5. A collection of distributed processes, both in the YARP realm as devices and in the ROS 2 ecosystem as nodes, are combined and coordinated through their corresponding middlewares, bridging the haptic input device with the target robot platforms.

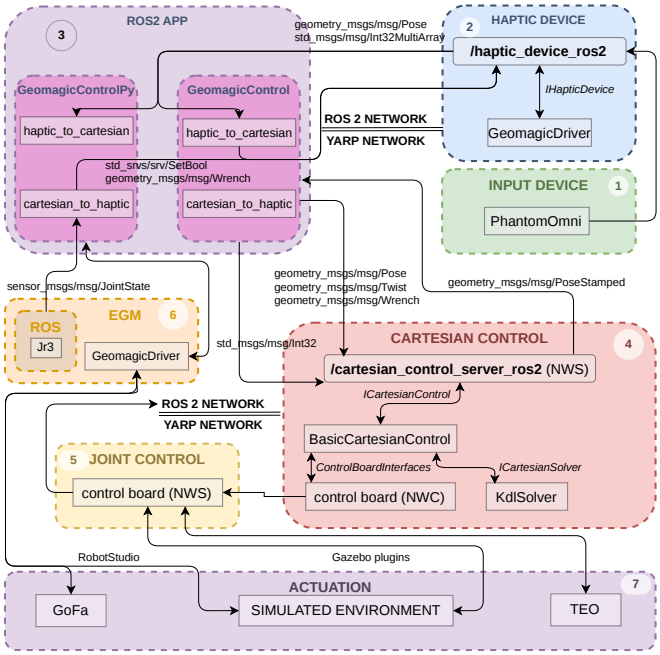



Figure 5: Teleoperation architecture for the haptic device.

The YARP–ROS 2 NWS bridge exposes the Phantom Omni haptic device over the ROS 2 network, publishing device state on dedicated topics and accepting force feedback through a subscription, as detailed in Table 1. This bridge constitutes the common entry point for all downstream ROS 2 nodes regardless of the target platform, decoupling the hardware-specific YARP layer from the application logic and allowing the same haptic device to drive different robot platforms without any modification to the underlying driver.

The main ROS 2 application node subscribes to the pose and button state topics published by the NWS and is responsible for continuously computing Cartesian commands to be forwarded to the controlled robot. On startup, the node performs an initialisation step in which it captures the current pose of both the haptic device and the robot end effector as fixed reference frames. This initial capture is essential to the differential control strategy: rather than mapping absolute haptic device coordinates onto the robot workspace, which would require precise spatial calibration between both systems, the node computes the incremental displacement between the current haptic pose and the captured reference, and applies that displacement on top of the robot's initial configuration. This approach makes the system robust to arbitrary initial positions of both the operator and the robot, and allows the operator to reposition freely between teleoperation sessions by pressing the first button of the Phantom Omni, which triggers a new reference capture without restarting the node. The second button provides an additional runtime control to fully enable or disable motion capture, allowing the operator to reposition the haptic stylus without inadvertently commanding the robot. Each incoming haptic pose is transformed into an incremental Cartesian displacement expressed in the robot base frame through a chain of KDL rigid-body transformations, with a fixed rotation parameterised at launch via three angles (roll, pitch and yaw) that aligns the sensor coordinate frame with that of the robot, compensating for any mechanical misalignment between the haptic device mounting and the robot base.

At this point the architecture branches depending on the target platform, as illustrated in Figure 5. For TEO, the computed pose is published over the ROS 2 network and consumed by the Cartesian NWS, which acts as the ROS 2-facing entry point of the YARP-based Cartesian control stack. This NWS communicates with the BasicCartesianControl implementation, which provides a generic Cartesian controller suitable for any kinematically described robot and manages an instance of the KDL-based Cartesian solver that performs the inverse kinematics calculations required to convert the incoming end-effector pose into joint-space references. The resulting joint commands are forwarded through the NWC counterpart of the joint controller down to the actual robot actuators or their simulated equivalents in Gazebo. Prior to initiating the streaming loop, the application node configures the Cartesian NWS through a parameter service call, setting the streaming command mode to pose so that the controller interprets incoming messages as absolute Cartesian pose targets. For the ABB GoFa, the YARP–ROS 2 and Cartesian control layers are bypassed entirely; Cartesian poses are transmitted directly over UDP using ABB's Externally Guided Motion (EGM) interface, a proprietary real-time streaming protocol that grants external processes direct access to the robot's motion controller at a low-latency control cycle (250 Hz). This removes the overhead introduced by the ROS 2 middleware and the YARP device stack, providing tighter timing guarantees that are particularly relevant for haptic teleoperation, where communication delays directly degrade the transparency and stability of the force-feedback loop. On startup, the EGM connection is established and the robot's current Cartesian state is retrieved and stored as the initial reference pose, mirroring the initialisation procedure used in the TEO path.

The force-feedback path is handled by a dedicated node that closes the bilateral loop independently of the target platform. For TEO, this node subscribes to the end-effector pose published by the Cartesian NWS. For the ABB GoFa, where no Cartesian NWS is present, the pose is derived directly from the EGM state stream. In both cases, the node evaluates whether the TCP has entered a virtual spherical constraint region defined by a configurable centre point and radius. When outside the constraint, a zero force vector is sent to preserve a transparent teleoperation feel. When penetration is detected, a restoring force proportional to the depth is computed, scaled by a configurable gain, capped at a maximum admissible value, and directed outward along the radial axis. The resulting force vector is published on the haptic device NWS feedback topic, which forwards it to the Geomagic Driver.

## 5. Experiments

The following section describes the validation experiments carried out on both robot platforms. For TEO, a simulated environment based on Gazebo Classic is used. For the ABB GoFa, validation is performed both in simulation using RobotStudio and on the real robot.

### *5.1. Teleoperation with simulated TEO*

To validate the system with TEO, the environment is prepared by launching the YARP server and sourcing the required ROS 2 packages, which include the Cartesian Control stack together with its YARP–ROS 2 bridge, the haptic device NWS package, and the developed application. Gazebo Classic is then launched with the TEO model, followed by the YARP device and Cartesian Control, the haptic device YARP–ROS 2 bridge, and finally the application nodes.

In a first phase, the teleoperation channel is validated. Moving the Phantom Omni causes pose commands to be sent to the Cartesian control server, and the right arm of TEO replicates the motion in the simulator. It is also verified that communication with the parameter server operates correctly, as the node configures the streaming mode to pose through a service call prior to entering the control loop. The button functionality is additionally tested: pressing the second button disables motion capture, halting the transmission of commands to the robot, and pressing it again resumes teleoperation. The node logs confirm each of these events.

In a second phase, the force-feedback node is launched, simulating a virtual spherical constraint in front of the TEO arm. As the robot TCP penetrates the defined region, the computed restoring force increases proportionally to the penetration depth and is sent to the haptic device. It is verified that the haptic device feedback mode is correctly configured as joint torque through the corresponding service, since in this mode the device directly receives a force vector per axis, as opposed to the Cartesian mode which is not suitable for this application. The operator physically perceives the resistance on the Phantom Omni stylus as the TCP approaches the virtual boundary.

Finally, to simplify deployment, a master node integrating the functionality of both nodes into a single process is used, and the same tests are repeated with equivalent results, reducing the complexity of the system launch.

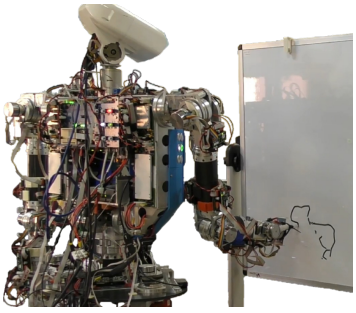


Figure 6: TEO performing a drawing task (Fernandez-Fernandez et al., 2023).

### *5.2. Teleoperation with ABB GoFa*

For the GoFa validation, RobotStudio is used as the simulation environment, running on Windows. Since the ROS 2 and YARP packages of the application run on Linux, WSL2 in mirrored networking mode is employed, enabling direct communication between the Linux environment and RobotStudio without additional network configuration. The USB Ethernet adapter connected to the Linux machine is natively accessible in this mode, also enabling communication with the haptic device running on a separate Linux host. Both machines are connected to the same local network. After sourcing the application ROS 2 packages, the YARP server and the corresponding bridge are started to handle communication with the haptic device.

In RobotStudio, the RAPID program is loaded, positioning the GoFa at a predefined initial configuration and enabling EGM communication, leaving the controller in a waiting loop until the external start signal is received. Once the EGM connection is established, the application node retrieves the current Cartesian state of the robot and stores it as the initial reference pose, mirroring the initialisation procedure used with TEO. Moving the Phantom Omni transmits Cartesian commands directly to the GoFa controller via UDP, and the simulated robot replicates the motion in real time within RobotStudio.

For the validation of the force-feedback channel in simulation, and given that in the real scenario this channel would be fed by readings from the JR3 force-torque sensor mounted on the end effector (Łukawski et al., 2026), an equivalent approach to the one used with TEO is implemented: the initial TCP position is taken as a reference, and a growing restoring force is generated as the robot advances along the Z axis beyond that reference, simulating contact with a surface. The joint torque mode of the haptic device is again verified to be correctly activated through the corresponding service, and the operator perceives the increasing force on the stylus as the robot penetrates the virtual contact region.

(a) Real robot (b) Simulated robot

Figure 7: ABB GoFa performing a drawing task with force feedback.

Beyond simulation, the teleoperation system was additionally validated on the real ABB GoFa robot. The same ROS 2 and YARP packages are used, with EGM communication established directly with the physical controller. The operator drives the robot arm through a drawing task using the Phantom Omni, while the force-feedback channel renders the contact forces measured by the JR3 sensor mounted on the end effector (Łukawski et al., 2026). Figure 8 shows a sequence of the task performed on the real setup.

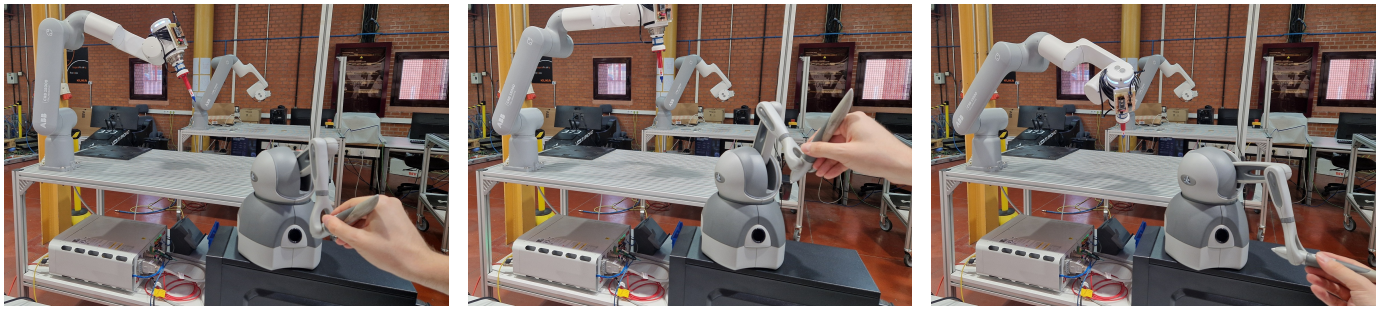


Figure 8: Drawing teleoperation task with ABB GoFa.

## 6. Conclusions

This work expands on existing teleoperation libraries developed in RoboticsLab by incorporating haptic feedback into a bilateral control loop across heterogeneous robot platforms. A layered architecture bridging YARP and ROS 2 has been proposed and validated on two distinct systems, demonstrating its flexibility and reusability.

The main challenge encountered was the translation of motion between two independently referenced coordinate frames. The differential control strategy adopted addresses this by computing incremental displacements relative to a captured initial pose on each side, eliminating the need for explicit spatial calibration between the operator and robot workspaces. Supporting both TEO, routed through the full Cartesian control stack, and the ABB GoFa, commanded directly via EGM over UDP, required a platform-agnostic initialisation design that was successfully shared across both paths.

Force feedback was validated in simulation using a virtual spherical constraint, with the operator physically perceiving contact boundaries through the Phantom Omni stylus. On the real ABB GoFa, the force-feedback channel was driven by readings from the JR3 force-torque sensor mounted on the end effector, and the operator successfully performed a drawing task under haptic guidance. Future work will focus on extending real robot validation to TEO and on evaluating bilateral loop stability under realistic contact conditions. The source code is publicly available online.[4]

## Acknowledgments

This research has been financed by "FotoArt5.0-CM, Laboratorios inteligentes para la ciencia del futuro" (TEC-2024/TEC-308) and "iRoboCity2030-CM, Robótica Inteligente para Ciudades Sostenibles" (TEC-2024/TEC-62), both funded by "Programas de Actividades I+D en Tecnologías de la Comunidad de Madrid"; "ROBOASSET: Intelligent robotic systems for assessment and rehabilitation in upper limb therapies" (PID2020-113508RB-I00, financed by AEI/10.13039/501100011033); "iREHAB: AI-powered Robotic Personalized Rehabilitation" (DTS22/00105, financed by Instituto de Salud Carlos III (ISCIII)); "ASEPEYO-UC3M: FASE IV" serious game-based rehabilitation program; and EU structural funds.

## References

Calzada, A., Łukawski, B., Victores, J. G., Balaguer, C., 2024. Teleoperation of the robot TIAGo with a 3D mouse controller. In: Simposio de Robótica, Bioingeniería y Visión por Computador. pp. 133–138.

Cheng, J., Abi-Farraj, F., Farshidian, F., Hutter, M., 2022. Haptic teleoperation of high-dimensional robotic systems using a feedback MPC framework. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 6197–6204.
DOI: 10.1109/IROS47612.2022.9981290

Fernandez-Fernandez, R., Victores, J. G., Balaguer, C., 2023. Deep robot sketching: an application of deep Q-learning networks for human-like sketching. Cognitive Systems Research 81, 57–63.
DOI: 10.1016/j.cogsys.2023.05.004

Hokayem, P. F., Spong, M. W., 2006. Bilateral teleoperation: An historical survey. Automatica 42 (12), 2035–2057.
DOI: 10.1016/j.automatica.2006.06.027

Łukawski, B., Montesino, I., Oña, E. D., Victores, J. G., Balaguer, C., Jardón, A., 2025a. Towards the development of telepresence applications with TIAGo and TIAGo++ using a virtual reality headset. In: Int. Conf. on Autonomous Robot Systems and Competitions (ICARSC). pp. 192–197.
DOI: 10.1109/ICARSC65809.2025.10970173

Łukawski, B., Olano Díaz, A., González Villasante, A., Oña, E. D., Victores, J. G., Jardón, A., 2026. External force-torque sensor feedback on the abb gofa collaborative robot. In: Simposio de Robótica, Bioingeniería, Visión por Computador y Automática Marina.

Łukawski, B., Rebollo, M., Gilabert, Á., Victores, J. G., Balaguer, C., Jardón, A., 2025b. YARP cartesian controller layers over ROS 2 for teleoperation and web applications. In: XLVI Jornadas de Automática.
DOI: 10.17979/ja-cea.2025.46.12252

Łukawski, B., Victores, J. G., Balaguer, C., 2023. A generic controller for teleoperation on robotic manipulators using low-cost devices. In: XLIV Jornadas de Automática.
DOI: 10.17979/spudc.9788497498609.785

Metta, G., Fitzpatrick, P., Natale, L., 2006. YARP: yet another robot platform. International Journal of Advanced Robotic Systems 3 (1), 43–48.
DOI: 10.5772/5761

Montesino, I., Bachiller, A., Victores, J. G., Balaguer, C., Jardon, A., 2025. Quasi-god object and geodesically restricted 6-DOF haptic forces for compliant constraints and low frequency simulation. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 13942–13948.
DOI: 10.1109/IROS60139.2025.11245986

Pérez Martínez, C., Pierro, P., Martinez, S., Pabon, L., Arbulú, M., Balaguer, C., 2010. RH-2: an upgraded full-size humanoid platform. In: Mobile Robotics: Solutions and Challenges. pp. 471–478.
DOI: 10.1142/9789814291279_0058

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., 2009. ROS: an open-source robot operating system. In: ICRA workshop on open source software. Vol. 3. p. 5.

Silva, A. J., Ramirez, O. A. D., Vega, V. P., Oliver, J. P. O., 2009. PHANToM OMNI haptic device: Kinematic and manipulability. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA). pp. 193–198.
DOI: 10.1109/CERMA.2009.55

Tian, J., Zhou, Y., Yin, L., Alqahtani, S., Tang, M., Lu, S., Wang, R., Zheng, W., 2025. Control structures and algorithms for force feedback bilateral teleoperation systems: A comprehensive review. Computer Modeling in Engineering & Sciences 142 (2), 973.
DOI: 10.32604/cmes.2024.057261

[4] https://github.com/alvarosang14/pruebas_geomagic