Bidirectional Tactile Interface: Control and Perception Strategies

Gabriel Paffi
Reconfigurable Robotics Lab, EPFL

***Abstract*—Tactile interfaces, integral to human-computer interaction, enable users to perceive and interact with virtual or remote environments through tactile feedback. This semester project presents the development and implementation of a bidirectional interface utilizing ROS2 communication between two modules equipped with Arduino, enabling bidirectional control and enhanced movement perception with motor control. The system features load cells to improve tactile perception and define the working mode of the interface. Accuracy and delay measurements on tactile interfaces were conducted to evaluate system performance. Future work will focus on integrating torque current control to enable direct force control with the interface.**

I. Introduction

In today's rapidly advancing field of robotics, the integration of interactive surfaces presents a promising avenue for enhancing teleoperation systems. Haptic interfaces are essential in enhancing human-computer interaction by allowing users to perceive and manipulate virtual or remote objects through tactile feedback [1], [2]. For instance, the DLR Bimanual Haptic Device [3] integrates two lightweight robotic arms to optimize workspace and force capabilities, providing high transparency and precise control in remote and virtual environments. Another study explores the feasibility of using haptic foot interfaces for controlling robotic assistants in laparoscopic surgery, emphasizing haptic-shared control strategies to alleviate coordination load [4]. Another example is Sigma 7 [7], a haptic interface developed by Force Dimension with force and torque feedback at the end-effector for aerospace and medical applications. Unlike conventional interfaces such as joysticks, interactive surfaces offer richer tactile feedback, improving user interaction and control. On the design side of the tactile interface, some research leads to origami-inspired structures, known for their compliance and lightweight properties, which hold significant promise for safe human-robot collaboration [5], [6]. The design of such interfaces is manufactured to provide three degrees of freedom force feedback in a compact platform, providing an adapted and safe approach to bidirectional communication with tactile interfaces.

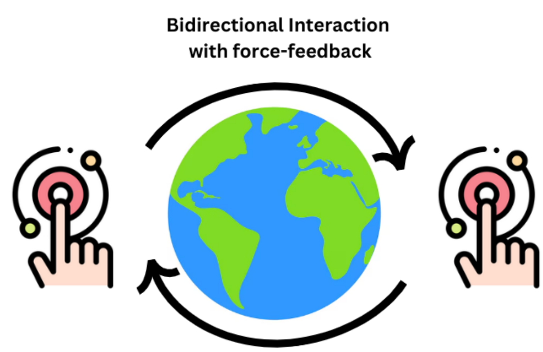
Building upon the existing literature, several scientific challenges emerge in the development of haptic interfaces. The primary challenge lies in achieving bidirectional interaction capabilities within haptic interfaces, particularly when integrated into origami-inspired structures with force-torque feedback. This entails developing mechanisms for communication and interaction between two users, enabling both parties to perceive and respond to tactile feedback in real-time.

Overcoming this challenge requires advancements in bidirectional communication protocols between two interfaces, control, and perception strategies with load cell integration to facilitate synchronized bidirectional interactions while maintaining the structural integrity and functionality of the origami-based robotic systems.

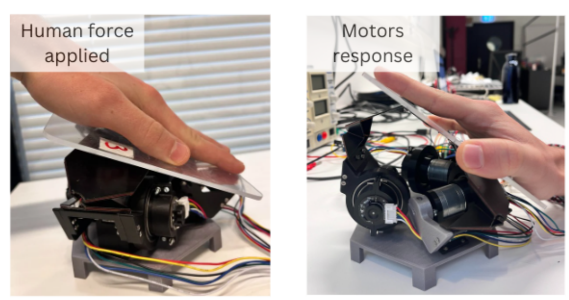
Our work addresses this gap by developing a haptic user interface with ROS2 communication between two Arduino-equipped modules, allowing for bidirectional control and enhanced movement sensation. The integration of load sensors into our system will enable torque-control feedback to enhance the realism of tactile feedback, enabling users to experience real-time force interactions.

Our contributions encompass:

- Bidirectional control strategy for surface-based tactile interface
- Perception of interaction through force sensing layer
- Experimental validation of the concept



a.



b.

Fig. 1. Bidrectional communication : (a.) represents the concept of the communication, (b.) is the realization of this communication with tactile interfaces with force feedback.

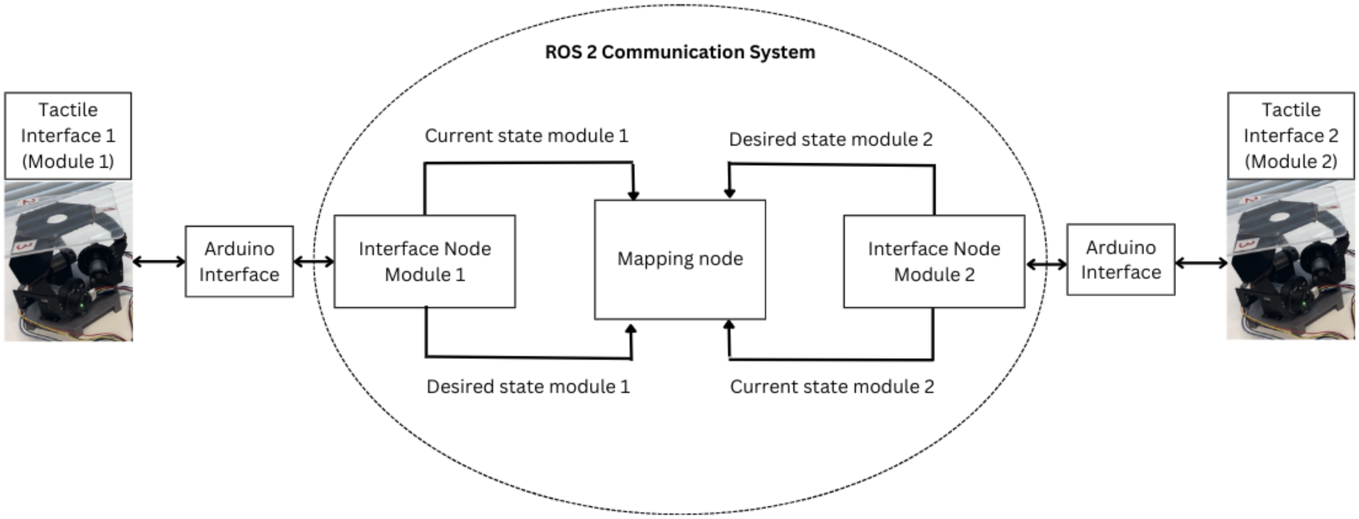


Fig. 2. Schematic of the ROS2 Communication: Arduino reads the current value of encoders of module 1, and communicates to module 2 and inversely via nodes to achieve bidirectional communication.

II. Bidirectional tactile interface

In the first part, I worked on the hardware part of the system, with two tactile interfaces inspired by origami robots with loads cells integrated. I also integrated an Arduino interface and ROS2 to establish communication between these two interfaces to enable bidirectional communication.

A. Interface Hardware

In this project, I focused on an origami-based haptic interface integrated with load cells for tactile interaction. These interfaces will enable bidirectional control and force feedback in our system (figure 3). The interface with origami design enables accuracy, control, and force feedback with its three degrees of freedom [5].

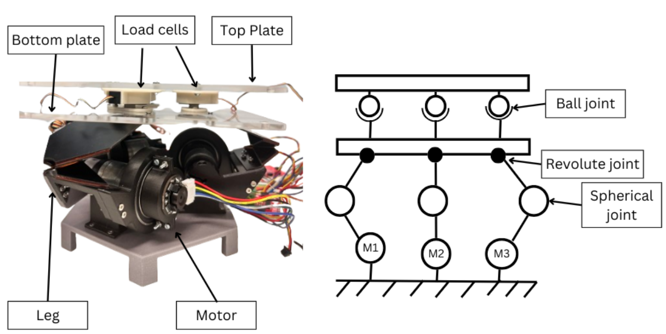


Fig. 3. Model Kinematics of the haptic interface with load cells implementation.

B. Arduino Interface

The Arduino Interface enables us to extract data from the module through a board such as the value of the encoder as a bit value connected to three motors and then transform it into a degrees value. It will also be used to send and receive data to ROS2 nodes. At the initialization, we implement a setup code for the tactile interface to go from a zero position ($\theta = (\theta_0, \theta_1, \theta_2,) = (0,0,0)$) to an initial position ($\theta = (20, 20, 20)$) where θ_i represents the position of the motor i in degree (figure 4).

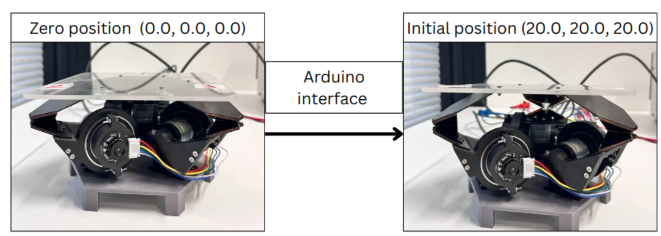


Fig. 4. The Arduino interface module initializes the setup loop to move from a zero position to an initial/working position.

C. Bidirectional communication using ROS2

Robot Operating System 2 (ROS2), serves as a robust framework in the construction of robotic systems. Featuring a flexible and modular architecture, it is fit for bidirectional communication which I want to implement in the tactile interface system.

1) Interface Node: This node facilitates communication between the Arduino Interface and the Mapping Node (figure 2). The node reads the message from the Arduino board that describes the position of a tactile interface through the 3-angle position of motors, communicated by the Arduino Interface, and publishes it as the current state ("Current_state_module_x"). Additionally, it subscribes to another topic named ("Desired_state_module_x") to encode the new desired state onto the Arduino board. This communication node will be used for the two tactile interfaces and can be applied to numerous tactile interfaces. The reading and encoding are achieved through the Arduino Class created with a serial library.

2) Mapping Node: This node facilitates communication between the two Interface Nodes (figure 2). The node subscribes to the current state of an interface and publishes a desired state for another interface. In our case, we aim for bidirectional communication, so the mapping node subscribes to the current state of the tactile interface 1 and publishes to the desired state of the tactile interface 2, and symmetrically the tactile interface 2 communicates the desired state to the tactile interface 1. To achieve asymmetric movement between the two modules, I create a function within the mapping node. This function takes the current state of an interface, the limit inferior for motors position ($\theta_1 = 0, \theta_2 = 0, \theta_3 = 0$), and the limit superior for motors position ($\theta_1 = 40, \theta_2 = 40, \theta_3 = 40$), and checks if the current state is within the limit on height, pitch, and roll of the second interface. It then returns the opposite position based on the maximum position and the minimum position. For a given current state of the tactile 1 ($\theta_{11}, \theta_{12}, \theta_{13}$)interface and the current limitations of our system, we have the following desired state for the tactile interface 2 :

$$\theta_{2i} = \begin{cases} \text{limit\_pos\_inf}[i] & \text{if } \theta_{1i} > \text{limit\_pos\_sup}[i] \\ \text{limit\_pos\_sup}[i] & \text{if } \theta_{1i} < \text{limit\_pos\_inf}[i] \\ \text{limit\_pos\_sup}[i] - \theta_{1i} & \text{else} \end{cases}$$

3) Launching ROS2 Node Simultaneously with a Launch File: To streamline the process, I have developed a ROS2 launch file that allows for the simultaneous execution of various nodes by calling them through a single file. This file can be easily customized based on the modules to be controlled and the desired mode of operation. For debugging purposes, we have the option to run individual nodes through separate terminals, providing a clearer insight into the process.

III. Control Strategies

In this project, I investigated two modes of interaction between the interfaces: a leader-follower mode and a bidirectional mode. While the ultimate goal was to implement the bidirectional mode, the examination of the leader-follower mode provided valuable insights into the dynamics required for its realization.

A. Leader-Follower mode

The Leader-Follower mode involves one tactile interface moving independently while the second tactile interface follows by mimicking the opposite movement of the first interface through the mapping node. Essentially, one interface can be controlled manually while the other automatically mirrors its actions, requiring only one interface to be actively controlled (figure 5).

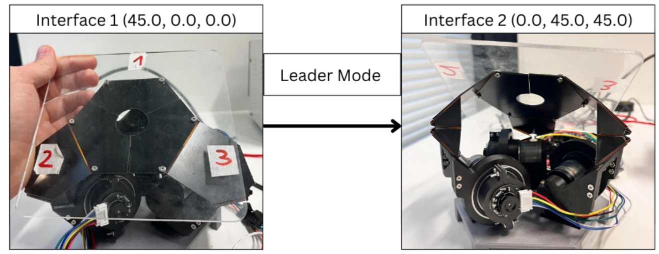


Fig. 5. Leader-Follower mode: the communication goes from the tactile interface 1 to the tactile interface 2: interface 1 sends the current encoders values, that is published as current state, and interface 2 subscribes to the desired state from the mapping node.

B. Bidirectional mode

The bidirectional mode involves the two tactile interfaces moving dependently on each other. Initially, the two interfaces will stabilize around their initial position. Once one of the interfaces moves, the second one will move in the other direction. With this process, putting a hand on each interface allows us to feel the pressure established on the first interface by a counter pressure on the other interface. In this mode, both interfaces are actuated, ensuring synchronized movement as they respond to each other's actions (figure 6).

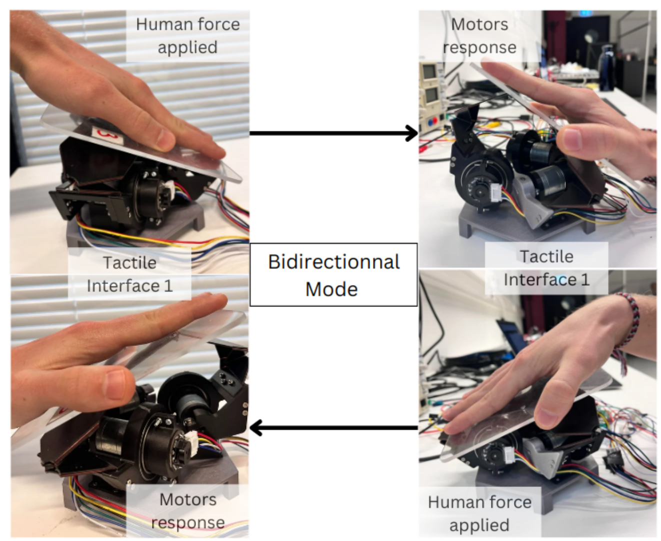


Fig. 6. Mutual mode: the communication goes from interface 1 to interface 2 and from interface 2 to interface 1: the interface 1 sends the current state and the interface 2 subscribes to the desired state from the mapping node, the same message is sent from module 2 to module 1 with in this case the current state of the interface 2.

IV. Force Perception

In this section, we will delve into the integration of load cells on the tactile interfaces. Load cells will enable the perception of interaction forces, providing precise measurements that will be added to our current position control.

A. Mechanical Integration

To implement the load cells into our system, we designed two new parts for each load cell. The coupler (figure 7) will be used to attach the load cells on the top acrylic plate (figure 3). The activator (figure 7) will modulate a contact point between the sensor and the bottom acrylic plate (figure 3) to apply pressure to the load cell. One of the key challenges in this implementation was to optimize the sensor activator height to ensure contact with the sensor pressure surface.

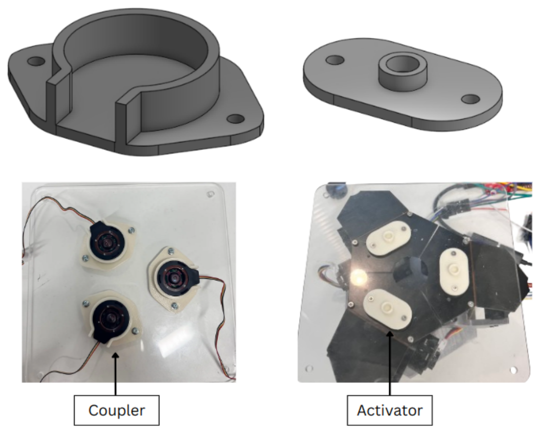


Fig. 7. Sensors Coupler and Activator: On the left, we have the coupler that will secure the load cell to the acrylic plate, and on the right, we have the activator that will make contact with the load cell.

B. Electronics integration

The integration of the load cells was accomplished using a MUCS (Multi-Use Communication System), which allows for communication across multiple I2C protocols. We then connected the load cells to breakout boards, which were then connected to the MUCS. The MUCS, in turn, connects directly to the main board.

The I2C (Inter-Integrated Circuit) protocol is a serial communication protocol that enables multiple devices to communicate with each other using just two wires: one for data (SDA) and one for the clock (SCL). This protocol is widely used in embedded systems due to its simplicity and efficiency in facilitating communication between various components like sensors, microcontrollers, and other peripherals. By leveraging the MUCS, we can interface with several I2C devices simultaneously, streamlining the data collection and integration process for our load cells.

C. Implementation

I then implement force reading into our Arduino Interface to read the actual force values of each load cell. Additionally, I implemented a system to decide on which mode we want the system to work. If the load cells on only one of the two tactile interfaces detects force, it indicates leader-follower mode, where the interface experiencing the force would lead the movement, and the other interface would perform the opposite movement. If both tactile interfaces load cells detect forces, the system would switch to bidirectional mode, where each interface could control the other and receive feedback from one other.

V. Results

In this section, I characterize the accuracy and the delay of our haptic interface. I focus on the leader mode with inverse positioning for the rest of the study. I use the Vicon system with Vero IR cameras to measure the displacement of the top plate of each haptic interface with precision. The experiment has been done with a frame rate of 300 Hz. The tactile interface 1 is the leader and is controlled by hand, the tactile interface 2 is the follower that will react thanks to DC power supply (4.5 Volts). Once the measurements have been done, I have treated the data to obtain three measurements on the height z, the rotation in x, and the rotation in y of each haptic interface.

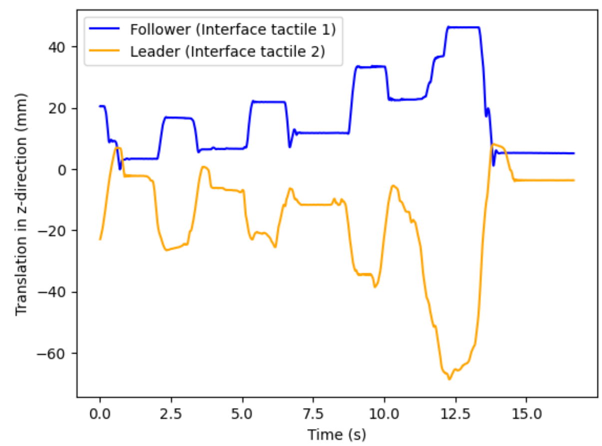


Fig. 8. Plot of the displacement in z-direction from the two tactile interfaces as a function of time from Vicon system.

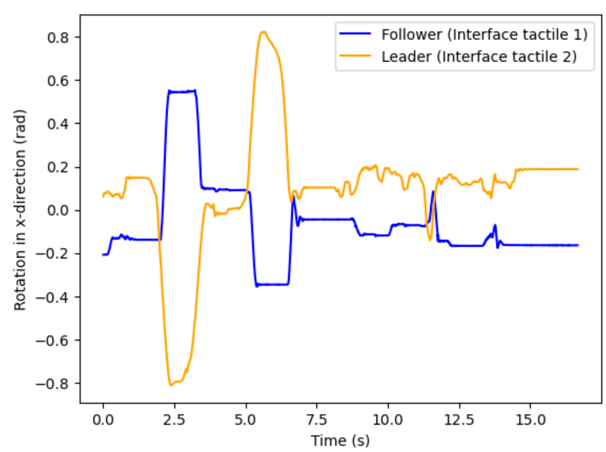


Fig. 9. Plot of the rotation in x-direction from the two tactile interfaces as a function of time from Vicon system.

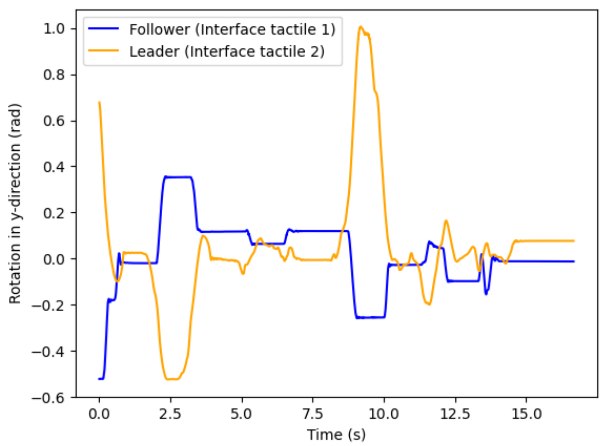


Fig. 10. Plot of the rotation in y-direction from the two tactile interfaces as a function of time from Vicon system.

From figure 8, figure 9 and figure 10, we can observe the asymmetrical displacement and rotation between the two interfaces confirming the operational integrity of our ROS system. We can also observe a saturation for the follower interface on displacement and rotation compared to the leader interface. For example, in Figure 8, we can observe multiple flat regions for the follower module, whereas the leader module can fluctuate freely. Another point to mention is the delay between the two interfaces. In figure 10, we observe a different time for the peak value for the leader (8.5 seconds) compared to the follower (8.75 seconds). Based on these plots I have computed the error of our system for each direction. I obtain an average error of 7.31 % for the displacement in the z-direction, 4.10% for the rotation in the x-direction, and 12.47 % for the rotation in the y-direction.

I also compute the mean delay in the z-direction by taking the difference times for the follower and the leader to reach some peaks (minimum local and maximum local). I determine the average delay with 4 peaks of figure 8 on the following interval of time : [2,4], [4.5, 7], [7.5, 10], and [10, 14], and I obtain an average delay of 0.18 seconds (5.55Hz) and a maximum delay of 0.63 seconds (1.59Hz).

VI. Discussion and Conclusion

In this project, I have implemented bidirectional communication on two origami-inspired interfaces through ROS communication. I integrated load cells to enable force perception between the two tactile interfaces and then measured the performance of the system with two main characteristics: accuracy and delay. The results of the experiment on the control position show the effective operation of the ROS communication between the two tactile interfaces but also some limitations of the system.

First, the error between the two interfaces can explained by:

- the actual limitations on motors and encoders that I impose in the mapping node to constraints the system in the reachable zone of each tactile interface.
- the hardware limitations on both DC motors and encoders.
- the human error in calibration and system assembly

Secondly, the delay between the two interfaces can be explained by:

- the delay in the communication on ROS determined by the ROS frequency that we choose at 10 Hz.
- the delay in communication between the Arduino board and the motors.

The current limitations and future enhancement of the system are:

- Force perceptions between the interfaces with closed-loop force control.
- Remote control of tactile interfaces through IP Protocols.

To address the first challenge, we want to compute the torque applied on each tactile interface. These calculated torques can be linked to the torque of the motor of the complementary interfaces. This approach will enable each module to react in term of force instead of motor positioning.

Based on figure 11 I already started to implement in the system the torque in roll and pitch based on the force applied in the z-direction with these 3 equations :

$$F_z = F_{z1} + F_{z2} + F_{z3}$$

$$\tau_{roll} = -F_{z2} * l_{y2} + F_{z3} * l_{y3}$$

$$\tau_{pitch} = +F_{z1} * l_{x1} - (F_{z2} + F_{z3}) * l_{x2}$$

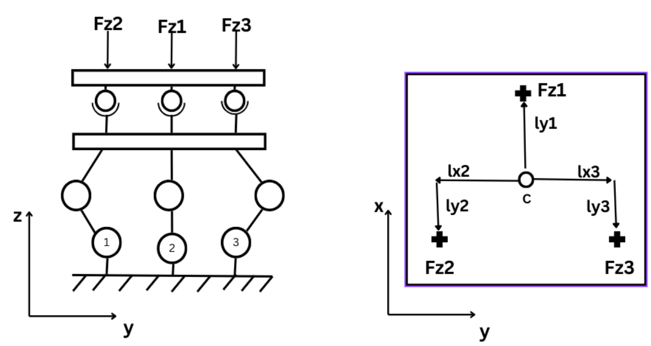


Fig. 11. Kinematics System description with Force-Torque relation, front view on the left of the figure and top view on the right of the figure

To address the second challenge, we explored the control of one module from another through a network. To achieve this, we integrated a segment of code into the mapping node using Python's socket library, allowing it to function as both a server and a client. This enables the sending of messages between computers via the IP protocol. We modified our mapping node to continue subscribing to the current_state topic from the interface node while also publishing messages via IP address (figure 12).

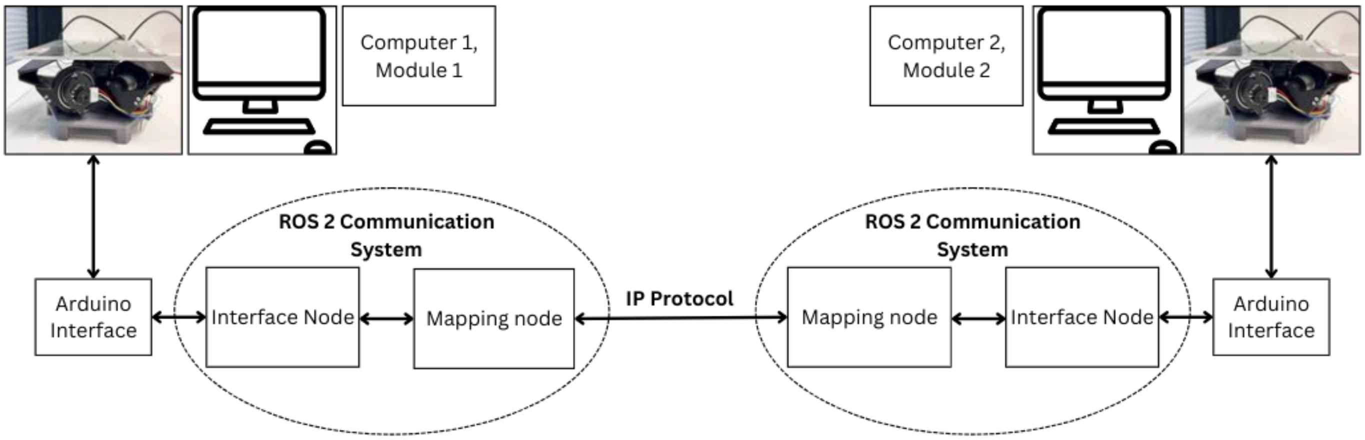


Fig. 12. Schematic of the ROS2 Communication with IP Protocol across multiple computers.

The developments achieved in this project will advance the precision and responsiveness of haptic interfaces. By integrating torque control mechanisms, the system will enable more accurate force feedback, improving the user's interaction with virtual or remote environments. Furthermore, the implementation of IP protocols will enhance communication between modules, supporting remote applications. The key objective is to facilitate bidirectional remote communication between two or multiple interfaces. These enhancements will contribute to the progress of haptic technology, with potential applications in fields such as remote interactions, virtual reality, and robotics.

VII. Acknowledgement

I extend my sincere gratitude to Mr. Alexander Schlusser from the Reconfigurable Robotics lab for his invaluable mentorship, resources, and guidance throughout this project. His expertise and support have been instrumental in this project. I extend my sincere gratitude to Professor Jamie Paik for allowing me to work in this laboratory and on this project.

References

[1] M. Kennedy, "The role of collaborative robotics in assistive and rehabilitation applications," *Science Robotics*, vol. 8, Oct. 2023.

[2] A. Kalinowska, P. M. Pilarski, and T. D. Murphey, "Embodied communication: How robots and people communicate through physical interaction," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, pp. 205–232, May 2023.

[3] T. Hulin, K. Hertkorn, P. Kremer, S. Schatzle, J. Artigas, M. Sagardia, F. Zacharias, and C. Preusche, "The dlr bimanual haptic device with optimized workspace," in *2011 IEEE International Conference on Robotics and Automation*, 2011.

[4] T. Hulin, K. Hertkorn, P. Kremer, S. Schatzle, J. Artigas, M. Sagardia, F. Zacharias, and C. Preusche, "Enabling four-arm laparoscopic surgery by controlling two robotic assistants via haptic foot interfaces," *The International Journal of Robotics Research*, vol. 42, pp. 475–503, May 2023.

[5] S. Mintchev, M. Salerno, A. Cherpillod, S. Scaduto, and J. Paik, "A portable three-degrees-of-freedom force feedback origami robot for human-robot interactions," Nature Machine Intelligence, vol. 1, pp. 584–593, 12 2019.

[6] J.-L. Huang, Z. Zhakypov, H. Sonar, and J. Paik, "A reconfigurable interactive interface for controlling robotic origami in virtual environments," *The International Journal of Robotics Research*, vol. 37, pp. 629–647, 2018.

[7] "Sigma.7 interface by force dimension," accessed: 2024-04-07. [Online]. Available: https://www.forcedimension.com/products/sigma

[8] ROS2 Humble Documentation, https://docs.ros.org/en/humble/index.html

[9] Hybrid Force-Impedance Control for Fast End-Effector Motions, IEEE ROBOTICS AND AUTOMATION LETTERS, vol. 8, no. 7, July 2023.

[10] FIT0521 DC Motor Datasheet, DFROBOT, https://www.dfrobot.com/product1617.html

[11] ROS2 Parameters and Launch Files, https://foxglove.dev/blog/how-to-use-ros2-parameters
https://github.com/foxglove/tutorials/tree/main/ros2/params

[12] Bridging Nodes Across Multiple Computers on ROS, Robotics Back-End, https://www.youtube.com/watch?v=nRi8qJ-Wk8k

[13] Sparkfun Qwiic Scale, load sensor implementation, https://github.com/sparkfun/Qwiic_Scale