

Type of the Paper (Article, Review, etc.)

Human-Robot Interaction through Wearable device - A Wireless Glove System for Teleoperated Control of 7-DoF Robotic Arm

Denis Manolescu¹, Bryan Mutinda¹, and Emanuele Lindo Secco^{2,*}

Affiliation: ¹Robotics Lab, School of Mathematics, Computer Science and Engineering, Liverpool Hope University, UK; 20203547@hope.ac.uk, 23010322@hope.ac.uk

²Robotics Lab, School of Mathematics, Computer Science and Engineering, Liverpool Hope University, UK; seccoe@hope.ac.uk

* Correspondence: seccoe@hope.ac.uk

Abstract

Significance - The ability of any robotic arm system to become tele operational is essential for advancing remote work and various future applications. Existing solutions often require complex setups, specialized training, and expensive hardware, limiting their adaptability and usability. Developing a simplified wearable device that can be rapidly integrated and deployed with these systems can significantly enhance versatility and accessibility for a wider range of tasks. Such devices are particularly crucial in environments where remote control, accuracy, and reliability are paramount, including healthcare for remote surgeries, manufacturing in automating processes, or hazardous environments like safe handling of materials or space exploration.

Aim and Approach - This work presents the design and development of a wireless glove-based control system for remotely operating a robotic arm. The primary objectives are not to focus on extremely precise estimation of the glove posture, rather to create a reliable, responsive, and user-friendly control system that seamlessly fuses user input with robotic motion capabilities. The system offers scalability, customization options, intuitive interaction, and ergonomic comfort to facilitate better human-robot collaboration. The approach combines advanced motion sensing using a *9-DoF Inertial Measurement Unit* (IMU) and real-time data processing to achieve teleoperation control. The IMU estimates roll, pitch, and yaw, directly controlling the translational movements (x, y, z) of the robotic arm's end effector. Depending on button states, these inputs can also switch to control the end effector's rotations. This approach provides instinctive control, enabling users to operate the system effectively with minimal training.

Results - The developed glove demonstrated user compliance, with operators reporting ease of use and intuitive maneuverability during testing. The system is also rapid to deploy, taking an average of 30 seconds to set up and engage. The wireless device successfully captured real-time gesture control data from the IMU and redirected it to the robotic system with an average latency of 185ms. The total latency from IMU data capture to robotic movement was approximately 200ms, faster than the average human reaction and lower than many existing wired glove-based control systems. This low latency, combined with the responsive control, enabled the system to conduct a range of manipulations,

¹ Article Affiliation

Author email address

including pick-and-place, reaching, and grasping objects. The performance of this system demonstrates its potential to enhance efficiency and accuracy in various applications compared to traditional control methods.

Conclusions - This research presents a robust and efficient teleoperation system, offering a viable solution for future advancements in remote robotic arm control. This innovation bridges the gap between the full potential of complex robotic systems and user interaction, unlocking new possibilities across various fields while being able to generate invaluable training data for further supervised machine learning integration.

Keywords: Human-Robot Interaction, Wearable Device, Intuitive Interaction; Human Machine Interface.

Citation:

1. Introduction

The advancement of robotic arm systems continues to profoundly transform each industry, from manufacturing to healthcare and space exploration. These robots offer unparalleled precision, strength and endurance, enabling them to perform tasks that are difficult, dangerous, or even impossible for humans a person. Teleoperation, the capability to remotely control robotic arms, further extends these benefits, allowing them to perform tasks in environments that are otherwise inaccessible or unsafe for human workers [1]. However, the widespread adoption of teleoperation systems faces substantial barriers due to challenges related to complexity, high costs, and the need for specialized training [2] [3]. Historically, robotic remote-control devices have predominantly been designed with bulky, heavy exoskeletons and built for specific robotic configurations. They also require considerable expertise and uncompetitive financial efforts to deploy. To address these limitations, researchers have been exploring the development of more immersive, simplified, and synergetic manipulation devices that can be seamlessly integrated with the existing robotic arm systems [4-7]. But despite the efforts, most of the challenges persist, restricting the use and broader adoption of robotic arms globally [8] [9].

Wearable devices, particularly glove-based designs, have gained significant attention as a promising control interface to facilitate remote robotic arm manipulation [10]. By capturing user hand gestures, like position, orientation, and finger movements, and translating them into mechanical movements, glove-control systems present a more natural and instinctive way of interacting with these robots. These characteristics reduce the learning curve and increase human brain acceptance [11]. Several studies have demonstrated the potential of these devices in teleoperation, showing progressive precision, motion-mimicking compliance, and superior task performance compared to traditional controls [12] [13]. However, integrating wearable controlling features with robotic systems still faces challenges, particularly regarding real-time data processing, latency, and reliable communication between the glove and the robot. Advances in sensor technology, such as the development of 9-DoF (degrees of freedom) IMUs, which combine accelerometers, gyroscopes, and magnetometers for comprehensive motion tracking, have significantly improved the accuracy and responsiveness of these systems.

Furthermore, existing glove-based systems often suffer from limited customization options and a lack of ergonomic comfort, restricting their effectiveness for prolonged or real-world applications [14]. Addressing these constraints is essential for developing a practical and adaptable solution that can be rapidly deployed in any environment.

To solve these limitations, this work introduces a simplified, battery-powered, wireless glove-control system for robotic teleoperation. The device is built and tested on the UFactory xARM-7, using a Raspberry Pi Zero 2W and an Adafruit BNO055 IMU as its central processing hardware. In addition, the proposed system aims to provide an accessible, reliable and flexible solution that can be easily integrated with various robotic arm platforms without requiring extensive modifications or specialized training. By prioritizing user-friendly design, low-latency control, and seamless integration, this project seeks to enhance human-robot collaboration and expand the possibilities for remote manipulation in diverse fields, ultimately contributing to the wider adoption of teleoperated robotic systems in many industries.

The structure of this research is as follows: The subsequent section, Methodology, provides a detailed description of the system design and implementation. This is followed by the Results section, which comprehensively evaluates performance, user experience, and discusses various issues, limitations, and solutions that have been implemented. The Conclusions chapter re-evaluates the accomplishments and their importance while setting future improvement goals. The paper ends with Appendix 1, which present the hardware and software tools used for this work.

2. Materials and Methods

2.1. 3D Design

The control glove is designed using Autodesk Fusion 360, a robust and versatile CAD software known for its capabilities in 3D modelling, simulation, and design optimization. The design process involved creating a detailed digital-twin prototype of the device, ensuring that all components fit together seamlessly, and that the overall ergonomics were optimized for user comfort and functionality (Figures 1 and 2). The glove system is a right-hand device composed of two parts (Figure 1)

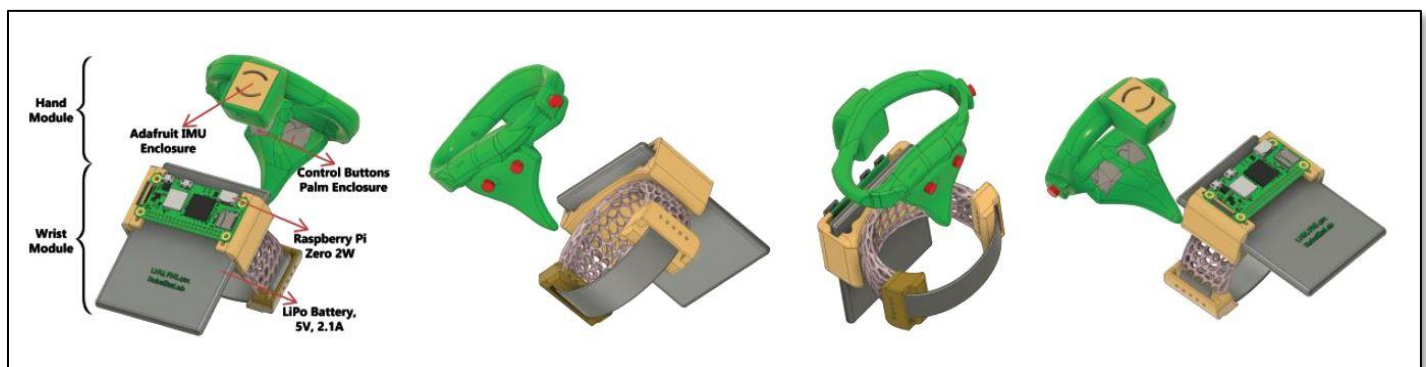


Fig. 1 – The Robotic Arm Control System Glove (RACS). Fusion 360 3D Design with Hardware Integrated

2.1.1 Wrist Module

This module features a custom watch-style design that holds the heaviest components, including the Raspberry Pi Zero 2W microcontroller and the battery pack, while allowing easy access for maintenance or adjustments. The compact module is secured to the user arm with an adjustable 270° Voronoi-pattern bracelet, a biomimicry method to ensure material efficiency, structural strength, and aesthetic appeal. The bracelet is enclosed around the wrist of the operator with an elastic strap, accommodating different arm sizes while ensuring durability, flexibility, and ergonomic comfort.

2.1.2 Hand Module

This shape-optimized part of the glove is built around the user hand. Designed based on the general anatomical shape of the human palm, the module sits naturally in place while allowing easy finger access to two palm-placed embedded buttons. An additional button is located between the thumb and the palm, reachable with a simple finger-flexing movement. These buttons are designed for intuitive use, allowing user to send specific commands to the robotic arm with minimal effort. The back of this module houses the enclosed Adafruit IMU unit, strategically placed to capture

hand motion accurately and ensure the module remains securely fixed to the hand. To minimize spacing, 3mm enamel wires are used for electrical connections, which are discreetly enclosed within the bracelet structure.

Both the wrist and hand modules are connected by a flexible wire tube, ensuring seamless communication and functionality. Further special attention was given to the ergonomic aspects of the glove. The design was iteratively refined to ensure it would be comfortable for prolonged use. These aspects included adjusting the shape and size of the bracelet and enclosures, optimizing the placement of buttons or straps, and ensuring that the glove would not restrict natural hand movements.

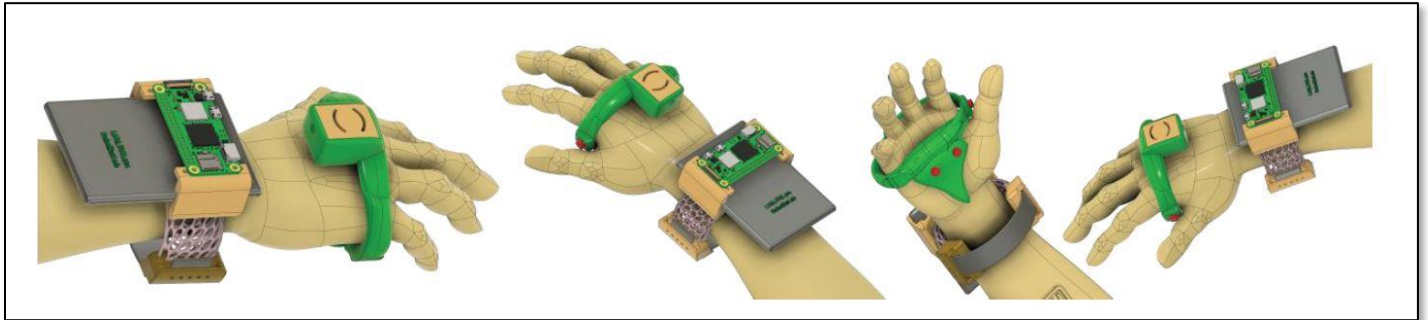


Fig. 2 - 3D Digital-Twin version of the RACS Glove

2.2. System Architecture

The glove-control system is built on three main computational structures: (1) the Glove Central Processing Unit, (2) the Local Computer Processing Unit and (3) the xArm-7 Robotic Arm

2.2.1 The Glove Central Processing Unit

The first primary structure is the device central processing unit, represented by a Raspberry Pi Zero 2W (Figure 3, panel A). This microcontroller was chosen for its balance of performance, size, and power efficiency, making it ideal for wearable applications. The Zero 2W features a 1GHz ARM Cortex-A53 CPU and 512MB of RAM, providing sufficient processing power for handling sensor data and user inputs necessary for this project. It supports various interfaces, including GPIO and I2C, which are essential for integrating different sensors and peripherals. The built-in wireless capabilities of the Zero 2W allow it to send data to the next computational structure, a local PC, ensuring maximum mobility for the glove device.

The microcontroller runs on Debian Linux 12 Bookworm with Desktop and boots from a 128GB SD card built using Raspberry Pi Imager. The initial setup requires an HDMI display and cable with keyboard and mouse to configure the system until the wireless SSH connection is functional. Once set up, SSH is used as a secure backdoor to ensure the Pi system runs smoothly, executes the data pipeline loop scripts, and can be supervised remotely anytime.

Furthermore, the microcontroller handles inputs from the Adafruit BNO055 and user control buttons. This IMU, connected via I2C, provides fast and reliable data about hand orientation and movement using its 9 DoF capabilities. The 9-DoF IMU combines data from three types of sensors: an accelerometer (measuring linear acceleration), a gyroscope (measuring rotational velocity), and a magnetometer (measuring the magnetic field). This fused data delivers a real-time, accurate, comprehensive view of hand motion, essential for precise and responsive control.

2.2.2 Local Computer Processing Unit

The second structure is a local PC, which acts as an intermediary between the Raspberry Pi and the xArm-7 robotic arm (Figure 3, panel B). This computer processes the data stream from Pi received wirelessly via TCP. Then it sends commands to the robot using the xArm-Python-SDK API libraries over an Ethernet connection. By offloading the data processing to the local PC, the Raspberry Pi can focus solely on capturing and transmitting real-time sensor data and button inputs, ensuring optimal performance and responsiveness. This architecture division allows the glove-control device to remain lightweight and mobile as long as a stable Wi-Fi signal is available.

The use of a local PC as an intermediary offers several advantages. First, it allows for more complex data processing and control algorithms to be implemented without overburdening the Pi processor. Second, it provides a flexible and modular architecture, allowing for easy upgrades or modifications to the system without requiring to change the entire hardware chain of the glove system. More in-depth details about the Python algorithm running through the entire system are available on GitHub ([Link](#)).

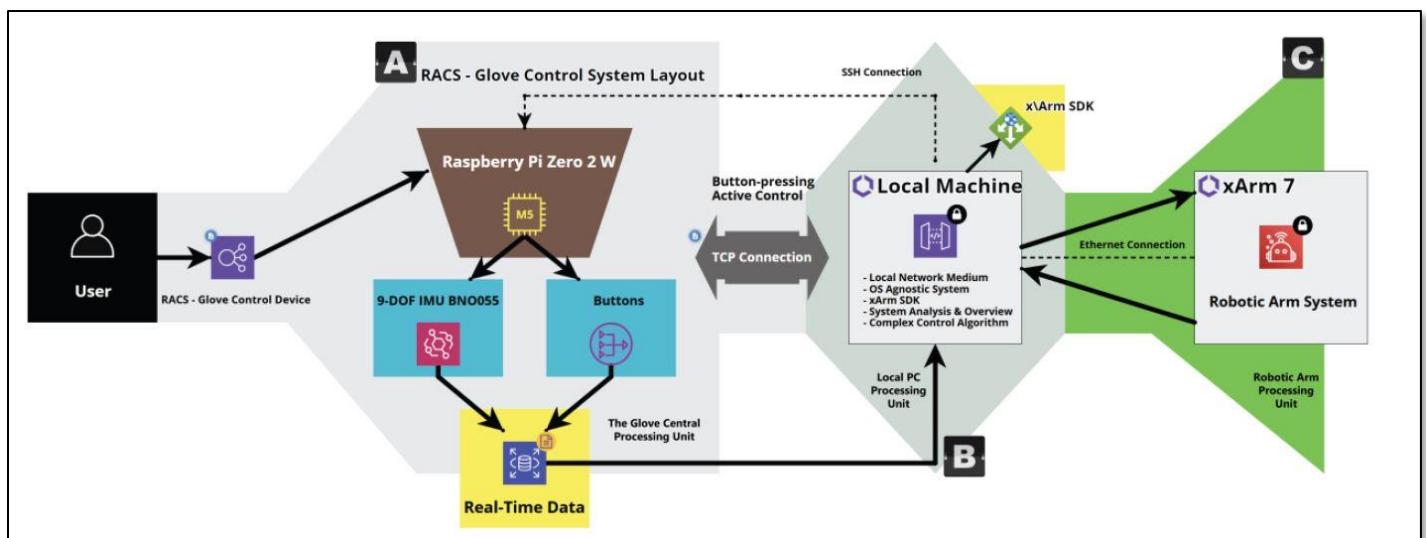


Fig. 3 – System Architecture Pipeline. Hardware and Connections Layout

2.2.3 xArm-7 Robotic Arm

The third data processing structure is the UFactory xArm-7, a versatile robotic arm designed for a variety of applications, including manufacturing, research, and automation (Figure 3, panel C). The arm features seven degrees of freedom, allowing it to perform complex tasks and motions with high flexibility and precision. It has high-torque motors and advanced control algorithms, enabling smooth and accurate operations. Additionally, it offers collision detection features and handy control software. The robust ethernet communication link between the local PC and the xArm-7 ensures low latency and high reliability, facilitating continuous teleoperation control. The xArm 7's modular design allows for easy integration of various end-effectors, making it adaptable to different tasks and environments. In essence, the local PC receives user input from Pi and sends corresponding commands to the xArm-7 using its Python SDK API. More details about the data flow of the system is presented in the next section.

2.3. Data Flow and Command Mapping

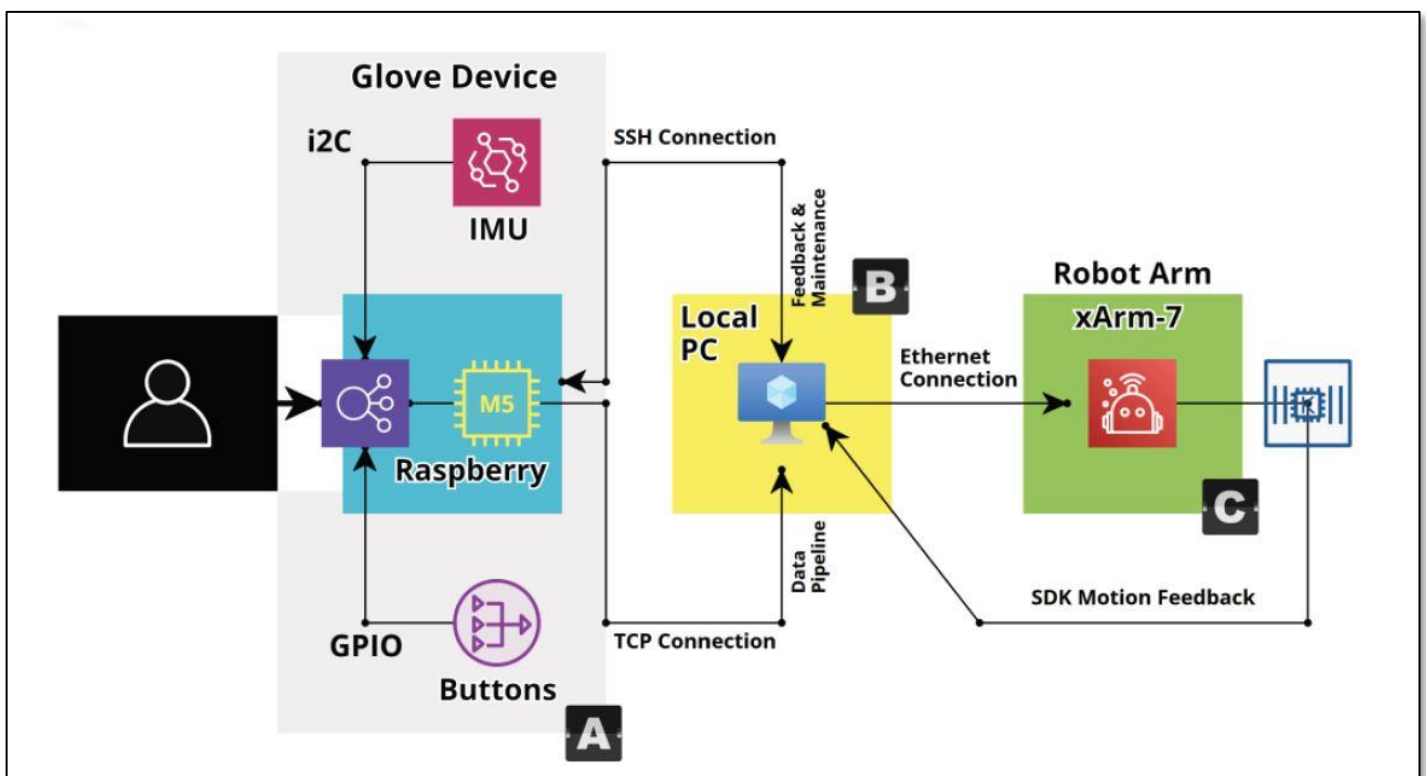
The data flow requires (1) the Data Processing on the Raspberry Pi Zero 2W board, (2) the Data Processing on the Local PC and finally (3) the Command Transmission to the Robotic Arm.

2.3.1 Data Processing on the Raspberry Pi Zero 2W

The Raspberry Pi Zero 2W continuously reads and processes data from the BNO055 IMU and the control buttons. The IMU is set to operate in NDOF mode (Nine Degrees of Freedom), providing comprehensive motion data, including quaternions, which represent the orientation of the operator hand. The Pi board converts these quaternions into Euler angles (yaw, roll, pitch) to facilitate more straightforward interpretation and control. At present we do not approach the system with a sensor fusion strategy, even if we could leverage in the future on Kalman filtering approach or similar ones.

In addition to the IMU data, the microcontroller monitors the state of the control buttons to detect user inputs. A Python script that runs in a close loop on the Raspberry Pi combines the sensor data and button signals to form the basis for controlling the robotic arm further down the pipeline (see *GitHub (Link)*). Once this data is processed, Pi transmits this information wirelessly to the local PC using a TCP connection. The connections layout of the whole system is presented in Figure 4.

2.3.2 Data Processing on the Local PC



Upon receiving the data from the glove microcontroller, the local PC interprets the incoming orientations and button input signals. The computer processes this data to generate precise control commands for the xArm-7 robotic arm. This operation involves translating the Euler angles and button states into XYZ-coordinates and specific movement instructions that the robot end-effector can execute. The PC effectively manages the Cartesian coordinates and ensures the robotic arm moves in strict synchronisation with the mapped hand gestures. The mapping process is defined as follows:

Fig. 4–Connections Layout between the Components

Yaw, Roll, and Pitch to Cartesian Movements - These angle values are mapped to translational movements. The intensity of any rotation dictates the magnitude or speed of the robotic motion:

1. Yaw - rotation around the vertical axis

A significant change in yaw indicates a forward or backward movement along the x-axis. An increase past the threshold in yaw means "forward," while a decrease means "backward."

2. Roll - rotation around the forward axis

A significant change in roll corresponds to moving right or left along the y-axis. An increase in roll means "right," while a decrease means "left."

3. Pitch - rotation around the lateral axis

A significant change in pitch indicates an upward or downward movement along the z-axis. An increase in pitch means "down," while a decrease means "up."

Button Inputs: Button 1 (Cartesian Movement Enable): The Cartesian movement of the robotic arm is only possible when this button is pressed and combined with one of the mapped gestures.

Button 2 (End-Effector Rotation Enable): This push button's functionality is activated in the same way as the previous button. However, when this button is kept pressed, the operator can execute only end-effector rotations. Yaw generates rotation about the z-axis, pitch causes rotation around the y-axis, and roll produces rotation about the x-axis.

The computer continuously updates the current position and orientation of the robot based on these mappings, ensuring that the arm's movements responsively mirror the user's hand gestures.

2.3.3 Command Transmission to the Robotic Arm

Using the xArm Python SDK API, the local computer translates the processed data into executable commands for the xArm-7 robotic arm. These commands are transmitted over an Ethernet connection, ensuring reliable and low-latency communication. With its seven degrees of freedom and advanced control algorithms, the robot executes these commands to accurately perform the desired movements and rotations. For Cartesian movements, any significant change in the Euler angles generated by the glove device moves the robotic arm by 10 mm along the corresponding axis. For rotational movements, changes in the Euler angles rotate the robotic arm by 30 degrees around the corresponding axis. This mapping method ensures that the robotic arm movements closely mimic the hand gestures and orientations captured by the glove device, providing an intuitive and responsive teleoperation experience.

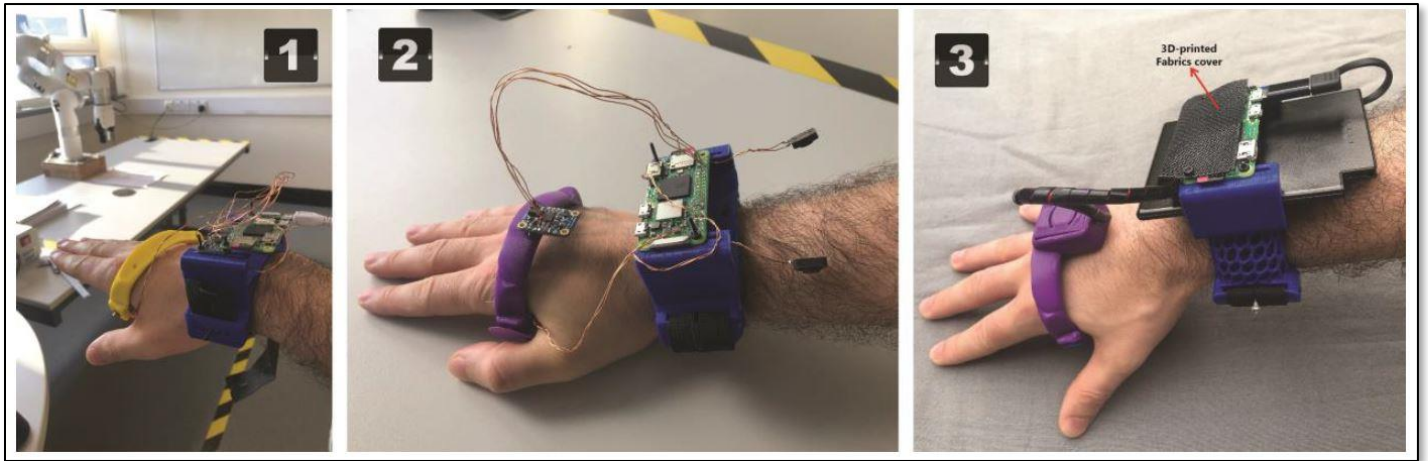


Fig. 5 – From the left to the right panels, the evolution of the RACS three versions design. The current Teleoperation Device is version 3 (right-side, panel 3)

2.4. Testing and Evaluation Criteria

To assess the usability and effectiveness of the developed system, we conducted five first-contact trials with users. During these trials, participants were guided by the system's functionality guidelines as they donned the glove and operated the robotic arm. The evaluation focused on the following criteria:

1. **Ease of Installation:** Was the system straightforward to install on the user's hand? Were there any challenges encountered during this process?
2. **Comfort:** Did the glove provide comfort throughout the operation? If discomfort was experienced, participants were prompted to describe its nature.
3. **Button Responsiveness:** Was the button-press mechanism for controlling the robotic arm responsive, easy to use, and intuitive?
4. **Accuracy of Movements:** Did the robotic arm's movements correspond to the user's intended hand movements?
5. **Task Performance:** Could the system be used to effectively perform manipulation tasks remotely?

Feedback gathered from these trials was instrumental in refining the system to better align with user expectations, particularly regarding ease of use, comfort, and control accuracy.

3. Results & Discussion

The glove design underwent numerous iterations, with various versions testing hand comfort and hardware integration (Figure 5). Users testing the current design reported positive feedback, noting its natural feel, lightweight structure, ease of maneuverability, and intuitive operation (Figure 6). The latest version of the device features 3D-printed fabric coverings for the wiring connections between the Hand and Wrist modules, enhancing both functionality and aesthetics (Figure 5, panel 3).

3.3.1 Performance Analysis and Data Latency

In teleoperation devices, latency data is the most insightful metric for evaluating the responsiveness of the system (Figure 7). During this research, the testing results show the following:

IMU to Pi Latencies: The top-left chart presents the data latency from the IMU to the Raspberry Pi, which aligns with the sensor datasheet and the i2C connection characteristics. The latency between these two devices remains mostly stable at around 0.10ms for standard speed control. It can occasionally spike to 0.15ms when the sensor processes more motion data simultaneously, such as during sudden or accelerated movements. Overall, these results are fully supportive of any teleoperation applications.

Pi to PC to Robotic Arm Latencies: Further down the data pipeline, the latency from the Raspberry Pi to the local PC via a Wi-Fi TCP connection averages around 185ms, with occasional peaks reaching up to 190ms. Although this result might seem high, it is within expected parameters given that the Pi Zero 2W is constrained to 2.4 GHz bandwidth. This noticeable yet manageable bottleneck in the data stream between the glove and the local computer does not pose a significant issue. Additionally, the Ethernet link between the computer and the robot adds approximately 10ms, resulting in a total latency from the IMU sensor to the robotic arm of around 200ms. To reduce the bottleneck latency issue between the microcontroller and the local computer, a device with 5GHz wireless bandwidth could potentially offer superior performance. One example is the BananaPi BMI M4 Zero.

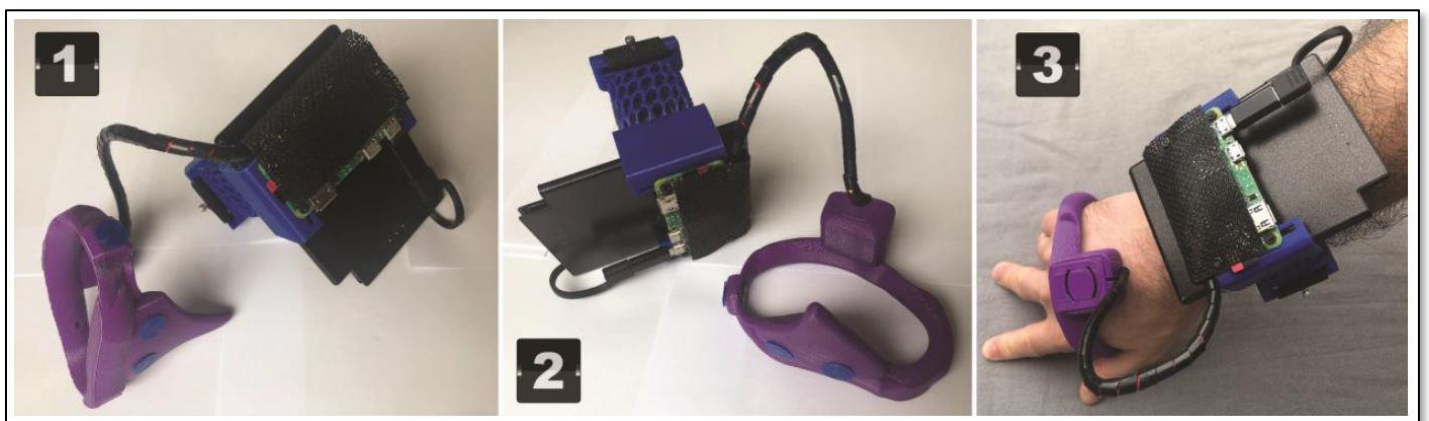


Fig. 6 – Details of the RACS Final versions of the Teleoperation Glove

The final results confirm the capacity of the glove-control system to maintain real-time operations, as the data stream latency remains below typical human reaction time. Given the wireless setup and compact design, the system is fast to deploy and demonstrates responsive and accurate control of the robotic arm during tests, validating its effectiveness for teleoperation. A video demonstration showing the teleoperation capability of the glove is available at this link: [YouTube](#). On the channel two novel demos and videos are also available at the following links

- Youtube demo #1: [Link](#)
- Youtube demo #2: [Link](#)

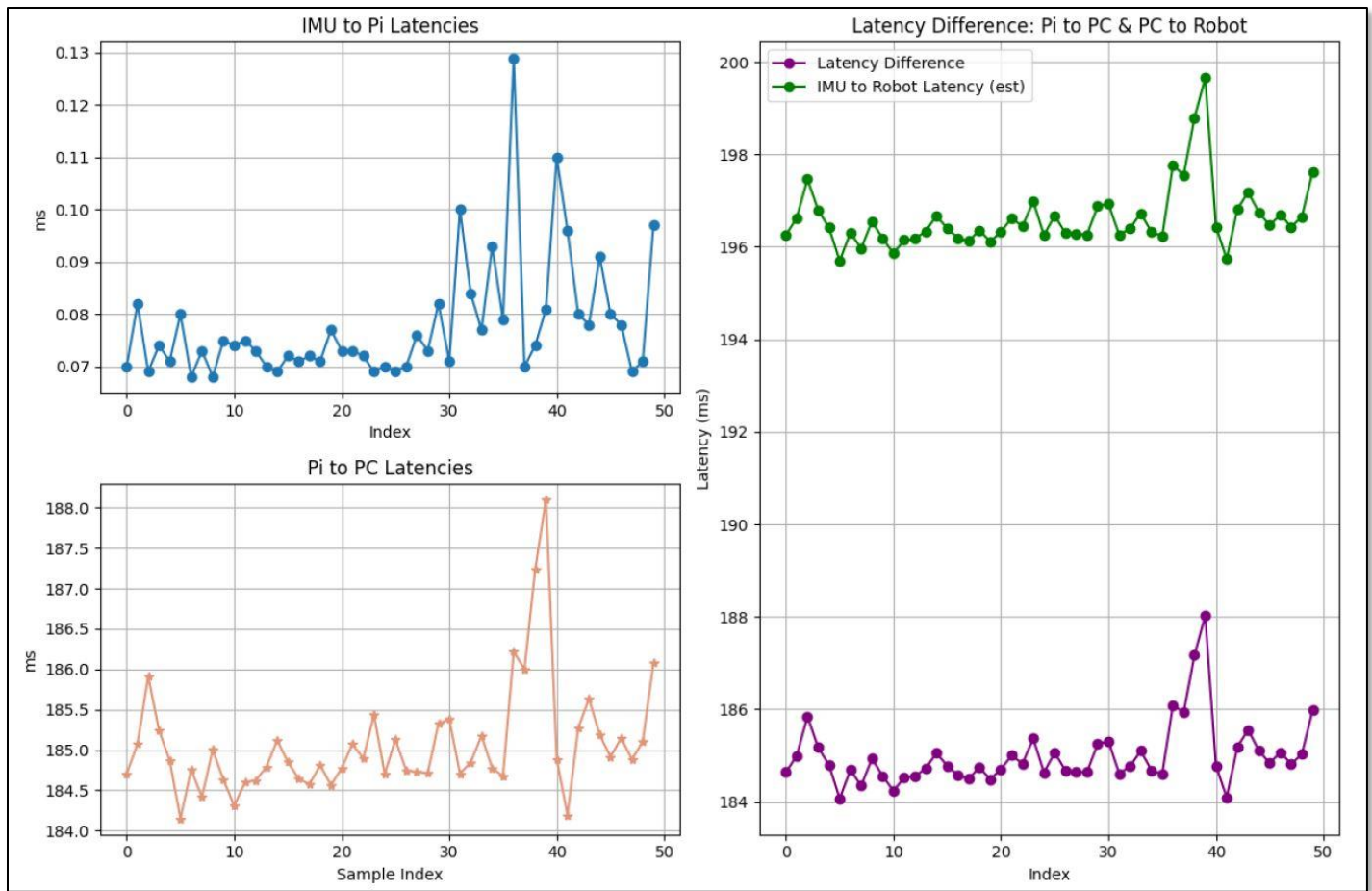


Fig. 7–Latency within the Teleoperation System

4. Conclusion

This research presents the development and successful implementation of a wireless glove-control system for robotic arm teleoperation. The resulting wearable device is a reliable, responsive, user-friendly manipulation tool that seamlessly integrates user input with robotic motion capabilities. The glove design achieves ergonomic comfort, intuitive interaction, and rapid deployment, making it accessible to operators without requiring extensive training.

The presented methodology and architecture serve as a solid foundation for future improvements aimed at achieving higher performance in teleoperation. Future work will focus on optimizing the glove design, exploring additional sensor integrations, and incorporating more powerful microcontrollers. In this context a better estimation of the position and orientation the glove (i.e. the pose) would be beneficial vs other applications. These efforts will focus on enhancing data processing algorithms, improving precision, and reducing latency. Additionally, incorporating haptic feedback and machine learning algorithms for gesture recognition are other key objectives to further improve performance and user experience, expand the capabilities of the system [15-21].

This work may results in benefit on daily interaction of humans with robotic devices: we see room for applications where people with disabilities may have to interact with auxiliary robots at home. We also see perspectives where PT, in a medical and/or rehabilitation environment, may have to teach their patients how to perform rehabilitative movements which a robot can then assist (or resist) towards the patient exercise. Another

aspect which could be explored and integrated with such a system involves telerobotics operation in the medical context (i.e. surgery) and in hazardous context, including space explorations

References

1. Tokatli, O. et al., 2021. Robot-Assisted Glovebox Teleoperation for Nuclear Industry. [Online] Available at: <https://doi.org/10.3390/robotics10030085> [Accessed 2024].
2. Li, S., Rameshwar, R., Votta, A. M. & Onal, C., 2019. Intuitive Control of a Robotic Arm and Hand System With Pneumatic Haptic Feedback. [Online] Available at: <https://ieeexplore.ieee.org/abstract/document/8812751> [Accessed 2024].
3. Wang, F. et al., 2023. Whole-Body Teleoperation Control of Dual-Arm Robot Using Sensor Fusion. [Online] Available at: <https://doi.org/10.3390%2Fbiomimetics8080591> [Accessed 2024].
4. Chen, J. et al., 2017. Development of an Immersive Interface for Robot Teleoperation. [Online] Available at: https://pearl.plymouth.ac.uk/bitstream/handle/10026.1/15865/440170_Print.indd.pdf?sequence=1 [Accessed 2024].
5. Stotko, P. et al., 2020. A VR System for Immersive Teleoperation and Live Exploration. [Online] Available at: A VR System for Immersive Teleoperation and Live Exploration [Accessed 2024].
6. Schwarz, M. et al., 2021. NimbRo Avatar: Interactive Immersive Telepresence. [Online] Available at: NimbRo Avatar: Interactive Immersive Telepresence [Accessed 2024].
7. Iyer, A. et al., 2024. A Versatile Teleoperation System for Robotic Manipulation. [Online] Available at: <https://arxiv.org/html/2403.07870v1> [Accessed 2024].
8. Wang, X., Shen, L. & Lee, L.-H., 2018. Xian Wang. [Online] Available at: <https://arxiv.org/html/2403.11384v1> [Accessed 2024].
9. Penchalaiah, P., Akhil, P. & Ramesh, P., 2023. Robotic Arm Controlled By Robotic Glove; P. Niladri Sai Prasad Reddy; Naveen Babu. [Online] Available at: https://www.indusedu.org/pdfs/IJREISS/IJREISS_4123_41295.pdf [Accessed 2024].
10. Friston, S. et al., 2019. Position-Based Control of Under-Constrained. [Online] Available at: <https://core.ac.uk/download/pdf/226756335.pdf> [Accessed 2024].
11. Park, J., Hwang, I. & Lee, W., 2020. Wearable Robotic Glove Design Using Surface-Mounted Actuators. [Online] Available at: <https://doi.org/10.3389/fbioe.2020.548947> [Accessed 2024].
12. Shadow-Robot-Company, 2022. Shadow Teleoperation System. [Online] Available at: https://www.shadowrobot.com/wp-content/uploads/2022/03/shadow_teleop_technical_specification.pdf [Accessed 2024].
13. Tefera, Y. et al., 2023. Real-time Immersive Remote Telerobotics: Highlighting the Benefit of Humans in the Loop and Applying Machine Learning. [Online] Available at: <https://ceur-ws.org/Vol-3524/short3.pdf> [Accessed 2024].
14. Pulgarin, E. J. L., Tokatli, O., Burroughes, G. & Herrmann, G., 2022. Assessing tele-manipulation systems using task performance for glovebox operations. [Online] Available at: <https://doi.org/10.3389%2Ffrobt.2022.932538> [Accessed 2024].
15. M Innes, EL Secco, [Design of an interactive BB8-like Robot](#), 7th International Congress on Information and Communication Technology (ICICT 2022), Lecture Notes in Networks and Systems (ISSN: 2367-3370), 137-144, 2022, DOI: 10.1007/978-981-19-1607-6

16. EL Secco, DD McHugh, N Buckley, [A CNN-based Computer Vision Interface for Prosthetics' application](#), EAI MobiHealth 2021 - 10th EAI International Conference on Wireless Mobile Communication and Healthcare, 41-59, 2022, DOI: 10.1007/978-3-031-06368-8_3
17. Bilawal Latif, N Buckley, EL Secco, [Hand Gesture & Human-Drone Interaction](#), Intelligent Systems Conference (IntelliSys), 3, 299-308, 2022, DOI: 10.1007/978-3-031-16075-2
18. AT Maereg, EL Secco, TF Agidew, R Diaz-Nieto, A Nagar, [Wearable Haptics for VR Stiffness Discrimination](#), International Workshop on Haptics, Pushing the boundaries of haptic research for Health: Current challenges, European Robotics Forum, Edinburgh, 2017
19. D Goodman, AK Nagar, EL Secco, A Ranasinghe, [Humans Sensitivity Distribution in Perceptual Space by a Wearable Haptic Sleeve](#), 7th EAI International Conference on Wireless Mobile Communication and Healthcare, Vienna, 14-16 November, 2017
20. M Li, EL Secco, Y Zheng, C Dai, P Xiong, G Xu, [Advances in Haptic Feedback for Neurorobotics Applications, Editorial](#), Frontiers in Neuroscience, Neuro Prosthetics, 2023, DOI: 10.3389/fnins.2023.1189749
21. AT Maereg, Y Lou, EL Secco, R King, [Hand Gesture Recognition Based on Near-Infrared Sensing Wristband](#), Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020), 110-117, 2020 - DOI: 10.5220/0008909401100117

Acknowledgments

This work was presented in coursework form in fulfilment of the requirements for the MEng Robotics Engineering for the student D Manolescu under the supervision of EL Secco from the Robotics Laboratory, School of Mathematics, Computer Science and Engineering, Liverpool Hope University.

Funding

There are no sources of funding to declare

Author contributions

Conceptualization, D.M.; methodology, D.M.; software, D.M.; validation, D.M.; writing—original draft preparation, D.M.;

writing—review and editing, D.M., B.M. and E.S.; supervision, E.S.

Conflict of interest

The author(s) declare no conflict of interest.

Data availability statement

Data supporting these findings are available within the article or upon request.

Institutional review board statement

Not applicable.

Informed consent statement

Not applicable.

Sample availability

Physical samples used in the study and parties or prototypes can be accessed in the Robotics Lab, School of Mathematics, Computer Science and Engineering, Liverpool Hope University.

Supplementary materials

The supplementary materials are available in the Appendix 1.

The Python algorithm running through the entire system are available on [GitHub \(Link\)](#).

A video demonstration showing the teleoperation capability of the glove is available at this link [YouTube](#).

Two more vide demonstrations are also reported at the following links

- Youtube demo #1: [Link](#)
- Youtube demo #2: [Link](#)

Additional information

Received: YYYY-MM-DD

Accepted: YYYY-MM-DD

Published: YYYY-MM-DD

Publisher's note: Academia.edu stays neutral regarding jurisdictional claims in published maps and institutional affiliations. All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright: © 2024 copyright by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Appendix 1 – List of Project Tools and Setups

Tools (hardware or software)	Objective	Issues	Observations
UFactory xArm 7 Robotic Arm	Implement low-latency teleoperation control.	-	Reliable, easy to integrate and experiment with.
Raspberry Pi Zero 2W (with battery or charger)	Fast IMU data transition via TCP.	Constrained to 2.4Ghz wireless bandwidth. Needs upgrade to 5Ghz for better performance.	Very fast to boot and stable. Energy efficient, easy to communicate with and to integrate.
USB Hub, Keyboard & Mouse (USB)	Utilities to set up the Raspberry Pi system.	-	
Adafruit BNO055 9-DoF IMU (i2C)	Very fast and reliable, with low latencies.	-	Needs more stress testing for more aggressive data collection.
Local Computer and Network	Used to reduce the data stream stress on the microcontroller.	Needs to have Wi-Fi and ethernet capabilities.	Could be replaced with a Raspberry Pi 5, for a more compact and dedicated solution to teleoperation.
SD Card (128 GB for Pi)	Raspberry Pi Zero 2W utility. 32Gb is minimum recommended.	-	Using Raspberry Pi Imager software for fast flashing and best booting results.
Li-polymer Battery 5V, 2.1A	To keep the system operational for teleoperation control.	Keeps Pi Zero 2W operational for about 2 hours. Probably not enough for some tasks.	Auskang 3.5Ah, 12.95Wh.
Raspberry Pi Imager	Used to install Debian OS on Raspberry Pi SD card	-	Fast and reliable solution
Fusion 360 (CAD Software for 3D design)	To 3D design and build the Glove device at a real scale and to test different aesthetics and design options.	-	Fast, reliable and free solution
Others: - 3D Printer + PLA; - Soldering Station; - Wires; - Push Buttons	Other utilities and consumables for this project.	-	Wires are enamel, 3mm, easy to solder and tough enough for such application. Buttons are flat, with 2-pair pins.