Glove for Real-time Imitation and Performance (GRIP)

Rian Lowery, Christopher Marcoccia, Kane McAgy, and Griseld Xhengo

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Glove for Real-time Imitation and Performance (GRIP) aims to narrow the gap between pilots and the environments their remote vehicles explore. Rather than using controls such as a joystick or preprogrammed subroutines to control an apparatus on the remote vehicle, the user will be equipped with a glove that takes positional data of their hand and wirelessly transmits that data to a bionic hand, which will be an extremity on the vehicle. With this system, pilots will be able to have more intricate interactions with the environment than current systems, allowing us to further satisfy our curiosity.**

***Index Terms* — Robotics, embedded systems, wireless communication, real-time systems**

I. Introduction

In 1953, photographer Dimitri Rebikoff developed the first remote observation vehicle (ROV) in order to capture photos underwater. Since then, ROVs have become a staple in the exploration of highly hazardous environments, each being meticulously designed to explore a variety of places from the depths of our oceans to the vastness of space. However, not all ROVs are just for surveying; some are equipped with tools to manipulate their surroundings, such as grabbers, cutters, and sample containers. Most of these tools, particularly grabbers, are controlled by pilots via joysticks. While they offer precise control, their overall functionality can still be quite limiting and sometimes overwhelming.

The Glove for Real-time Performance and Imitation (GRIP) aims to address these limitations by enabling direct motion replication between a human operator and a robotic hand. Rather than relying on indirect control inputs, GRIP allows users to control a bionic hand directly with their own, via a specially designed glove that tracks their hand's movements. The project aims to create a seamless interface between human intent and mechanical execution.

The project is motivated by applications in hazardous terrain, one example being the handling of chemicals and remote exploration. Additionally, GRIP has potential applications in prosthetics, where the data from the glove could be used to help map gestures to the electrical signals from the brain, providing more accurate signal processing for prosthetic gestures.

II. System Overview

The GRIP system is comprised of two major subsystems communicating with each other wirelessly, those subsystems being the Glove system and Robotic Hand system. The system is designed to prioritize low latency and reliability while maintaining a cost-effective and scalable architecture.

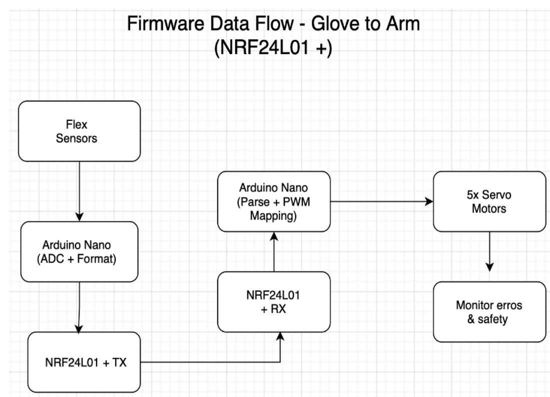


Figure 2.1

A. Glove Subsystem

The Glove system uses flex sensors, which are variable resistors that increase in resistance the more you bend them, to measure the position of the user's fingers. This is done by using a typical voltage divider circuit, the output of which is connected to the MCU. Whenever the resistance of the flex sensor changes, the voltage will change accordingly, and the Arduino Nano will detect it and process the data. This data is then converted into angles for the servo. The values are then transmitted to the robotic hand subsystem via the wireless transceiver module. The glove is responsible for capturing the user's hand movements. Flex sensors are designed and positioned along each finger to measure bending angles, manufacturing analog signals proportional to said bend angles for finger motion. Signals are processed by an embedded microcontroller that converts them into digital representations of joint positions.

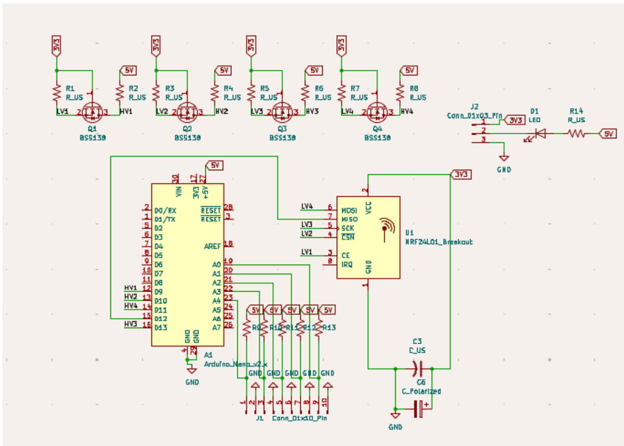


Figure 2.2

B. *Robotic Hand Subsystem*

The robotic hand subsystem is relatively simple compared to the glove. This system receives the angles from the glove and drives the servos to the desired position. The servos have their own power source, as the Arduino is unable to provide enough current for them. This also reduces the noise substantially, which allows for cleaner signals.

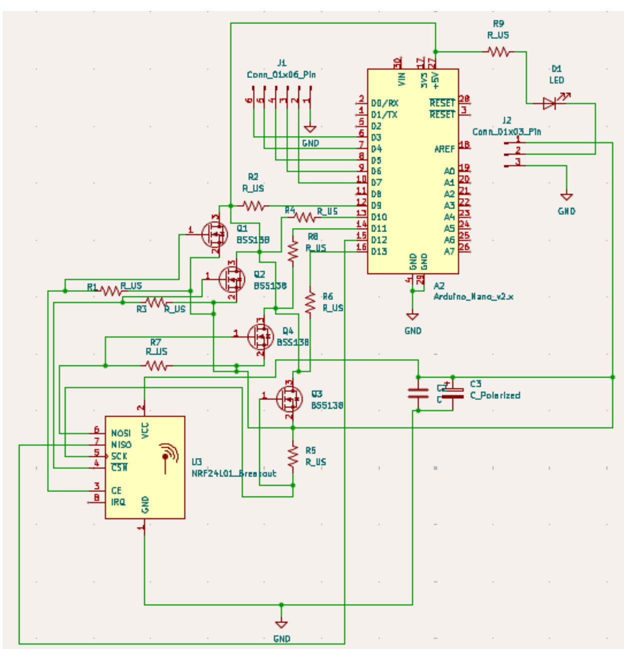


Figure 2.3

C. *System Workflow*

1) Data acquisition
2) Signal processing and calibration
3) Wireless Transmission via transceiver
4) Data validation and interpretation
5) Mechanical real-time actuation

III. System Components

This section gives an outline of the components that come together to make up the whole GRIP system.

A. MCU

The center of this project is going to be a version of the Arduino Nano that we are designing and assembling ourselves. There will be some minor differences in the components used, but the functionality will remain the same. The MCU will handle all of the code and calculations for the project. This will include receiving the values from the flex sensors and making them useful for the motors. Since we are using two main PCBs, the other MCU will take those values from the flex sensors and then turn those and make sure the motors turn according to what the glove is doing. The MCUs will also use the five-volt pin to send power to the wireless communication module on the PCBs, since portability will be important.

B. Wireless Communication

In order to communicate between the two subsystems, the nRF24L01 transceiver will be used. This is a common component used for RF systems and is fully compatible with Arduino. However, the Arduino cannot produce enough current for the nRF, so the five volts from the Arduino will be converted into 3.3V via the LM3670. This will not only produce the desired 3.3V signal, but also the current needed for the nRF to function, with peak current draw being around 120mA. The frequency will be in the range of 2.4 - 2.5 GHz.

C. Logic Level Shifter

Since the MCU is a five-volt device and the wireless communication is a 3.3-volt device, there needs to be a logic level shifter in between the two in order for them to communicate. The BSS138, a MOSFET device, is utilized in a bidirectional logic level shifter circuit in order for the Arduino and nRF to communicate properly.

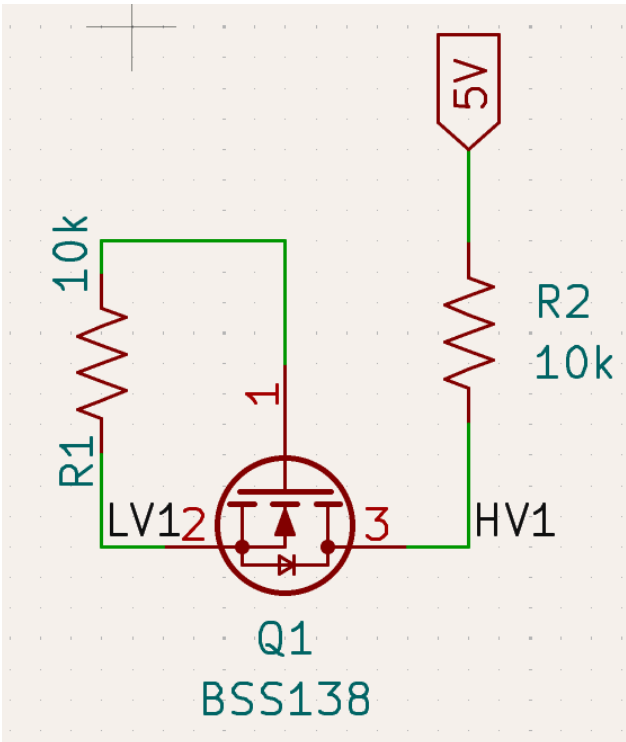


Figure 3.1

D. Motors

The motors for this project needed to be more sturdy than a normal servo motor since they need to be able to move the fingers and hold an accurate position under a load. MG996R servo motors were selected for their small size and high torque. They also have a generous functional voltage range of 5-7.2V, with 6 volts being the most optimal for torque. They meet all the requirements we have for the motors. However, using 5 servos at once has a very large current requirement. Each motor has the potential to stall at the same time, which will result in a maximum potential current draw of 15A.

E. Batteries

Since this project will need to be portable, the power supplies need to be small and portable as well. The motors are the most demanding component, so they will have their own power supply in the form of an NiMH RC battery. This will allow them to get the power they need, and the battery can be mounted with wires going to the motors. The two subsystems will use standard 5V 2.4A USB power banks.

F. Voltage Regulator

In order for the MCU and the nRF24 to communicate properly, we need to be able to acquire a 3.3V source that can provide enough current for the nRF. For this purpose, the LM3670 was chosen. With an efficiency of 90-95%, its ability to convert the 5V from the Arduino with minimal losses is ideal for the design. However, it is a noisier option compared to other regulators that we viewed, but by filtering the output with a set of capacitors, we are able to reduce the noise substantially.

IV. Design Objectives and Specifications

The design of GRIP is mapped by a few key objectives to maximize effective performance and usability. Primarily, the idea is to achieve accurate replication between human operators and a robotic hand. The system is designed to have an intuitive feel, allowing users to control the robotic hand with ease.

A. Design Goals

The main tasks of the system include achieving real-time motion replication, helping to minimize latency, maintaining high accuracy, and, above all, ensuring safe operations. These objectives are critical for enabling intuitive user movements and reliability in the system performance.

B. Performance Specifications

Meeting the goals of the project required a series of objectives to complete, with the following specifications in mind:

- Comms latency less than 250 milliseconds
- System battery to be at least 1.5 hours
- Finger response below 300 milliseconds
- Packet loss below 0.5% within a 5-meter range
- Transmitter range of at least 10m

Safety is also a critical consideration among the specifications. The system must ensure all movements remain within mechanical limits in order to prevent damage to hardware components or the operator.

C. Design Challenges

Realistically achieving specifications required balancing several competing factors: communications, system complexity, and power consumption. One example of this is reducing the latency required with transmission protocols in the software, all while keeping the accuracy required with careful sensor calibration and signal filtering. Another example is our wireless communication.

There are two models of the nRF24, one with an external antenna and one with a trace antenna. The external antenna offers vastly more range at the cost of a slower bit rate. Since we require the quickest speeds possible, we made the decision to use the trace antenna model, as it provides the fastest speeds. Furthermore, should we wish to have more range for the system, these problems will be much simpler to solve, rather than increasing the bitrate for the antenna nRF.

V. Hardware Design

A. Glove Hardware Design

The glove subsystem needs the flex sensor technology to capture finger motion. The sensors are flexible, lightweight, and interchangeable. When bent, the resistance increases, which is used to vary the voltage going through them. The sensors connect to an Arduino Nano Microcontroller, which processes the analog signals into digital data. For the MCU to communicate with the nRF, there needs to be a logic level shifter on the PCB. This is because the Arduino is a 5-volt device and the transceiver is a 3.3-volt device. The 5-volt signal is sent through a series of MOSFETs to the transceiver. The data is then sent through the transceiver to the transceiver on the hand subsystem. Since the Arduino is going to be powering the rest of the components on the PCB using the 5 volts it outputs from one of its pins, there also needs to be a regulator that converts the voltage to 3.3 volts. We used the LM3670 regulator for this.

B. Robotic Hand Hardware Design

The hand subsystem is very similar to the glove hardware-wise. The minor differences come from the components requiring a different setup. Since the flex sensors are absent, the 5V pin on the Arduino will only be used for the level shifters and the input of the regulator. The same nRF24 is used as a receiver on this subsystem and receives the data. The logic level shifter then makes sure the MCU and the receiver can communicate with each other. The board for the Arduino, nRF, and all parts necessary for communication between them are on one board, while the motors and their power source reside on another board

C. Motor Board Design

Since the motors require a large amount of current and exact voltage, we decided to put the connectors for them on their own PCB and run the signal to the main hand PCB with wires and connectors. The current for each motor would be around 3 amps, and the voltage from the battery should be no more than 6 volts, or we risk damaging the motors. Since the current will be so high, special attention was paid to the size of the traces in order to make sure there is no bottleneck for the current and the motors all work properly. There is also a large capacitor of 4700 microfarads that is added at the power terminal to handle voltage spikes. Without the capacitor, the servos can brown out and start rotating without stopping, which was seen in testing before the addition of the capacitor.

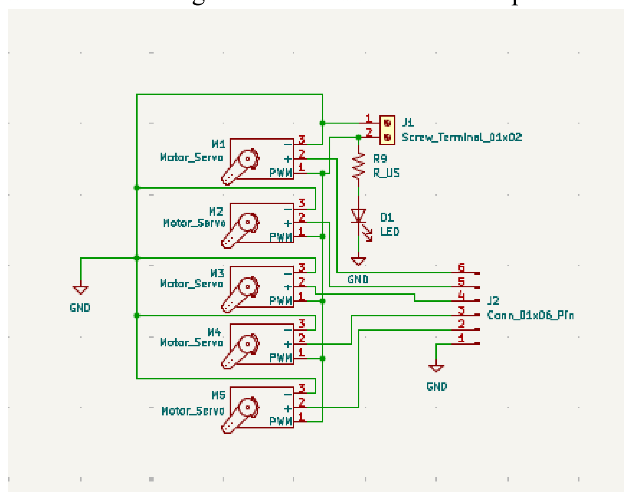


Figure 5.1

D. Regulator

The regulator was a simple LM3670 that is on its own PCB that connects to the main PCBs with a connector to allow for easy swapping and keep the board footprint small for mounting. There was troubleshooting required since it was not outputting the required 3.3 volts from the input of 5 volts. The inductor on the regulator had to have the appropriate maximum current for the system since the thermal limit would lower the output voltage.

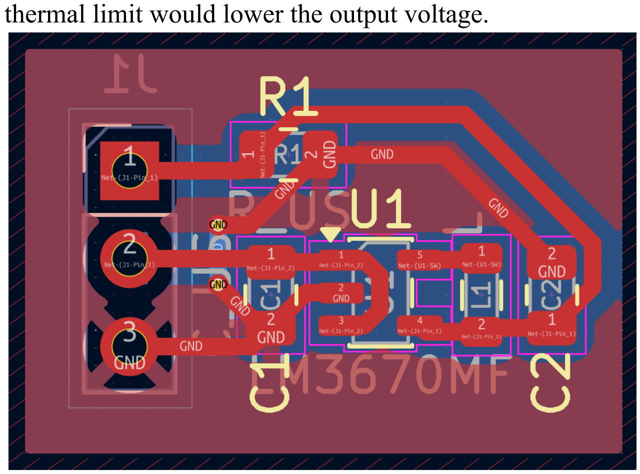


Figure 5.2

VI. Software Design

A. Glove-Side Signal Acquisition and Processing

The glove subsystem is responsible for capturing and processing real-time finger motion using flex sensors connected to an Arduino Nano microcontroller. Each flex sensor produces an analog voltage proportional to the degree of finger bending. These signals are sampled using the microcontroller’s 10-bit analog-to-digital converter (ADC).

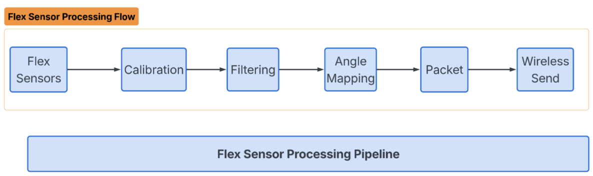


Figure 6.1

To ensure consistent performance across different users, a calibration process is implemented. Minimum and maximum bend values are recorded and used to normalize the raw ADC readings into a standardized range. The normalized values are then mapped to joint angles between 0 and 180 degrees, corresponding to the operating range of standard servo motors.

To improve signal stability, light filtering is applied to reduce small fluctuations caused by electrical noise and sensor variability. The filtering is carefully tuned to minimize jitter while maintaining low-latency response, ensuring smooth motion tracking without introducing significant delay.

B. Wireless Communication Protocol

Processed joint angle data is transmitted wirelessly using the nRF24L01 2.4 GHz transceiver. The module operates in a low data rate mode of 250 kbps to improve communication stability and range while maintaining low latency.

A custom packet structure is implemented to efficiently transmit multi-finger data. Each packet consists of a start byte for synchronization, a sequence number for tracking packet order, and one byte per finger representing joint angles. This compact structure minimizes transmission overhead and ensures fast data transfer.

The nRF24L01 module provides built-in hardware cyclic redundancy check (CRC) validation. This ensures that corrupted packets are automatically rejected at the hardware level before reaching the software processing stage, improving overall system reliability.

C. Receiver-Side Validation and Processing

On the receiver side, incoming packets are first validated to ensure data integrity and synchronization. The start byte is checked to confirm correct packet alignment, followed by hardware-level CRC validation performed by the transceiver.

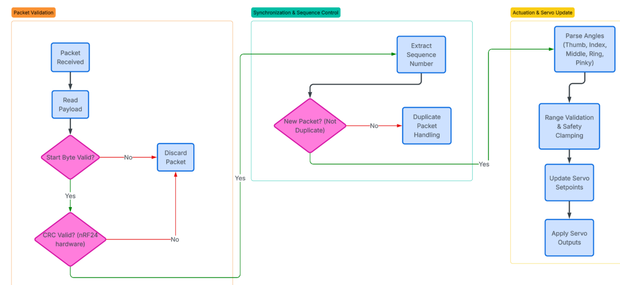


Figure 6.2

After validation, the sequence number is extracted to detect duplicate or out-of-order packets. Duplicate packets are ignored to prevent repeated application of the same motion command, which could introduce instability or jitter in the system.

If a packet fails validation, it is discarded, and the system maintains the last valid actuator state. This failsafe mechanism prevents corrupted or missing data from affecting physical motion, ensuring stable and predictable system behavior.

D. Servo Control and Actuation

Validated joint angle data is converted into pulse-width modulation (PWM) signals to drive servo motors. The system uses hardware timer-based PWM generation to ensure precise and stable pulse timing.

The servo control loop operates at approximately 50 Hz, corresponding to a 20-millisecond refresh cycle. This update rate is sufficient to capture natural human finger motion while maintaining smooth and responsive actuation.

To further enhance motion stability, light smoothing is applied to the servo command signals. This reduces minor fluctuations caused by sensor noise or wireless timing variations. Additionally, safety clamping ensures that all commanded angles remain within the mechanical limits of the servos, preventing potential hardware damage.

E. Software Technology Selection and Justification

The software architecture was designed with a focus on deterministic timing, reliability, and simplicity. Table I summarizes the technologies considered and the final selections.

Component	Options Considered	Selected	Justification
Programming Language	Python, C++	C++	Deterministic timing, low-level hardware access
Wireless Protocol	Bluetooth, WiFi, nRF24	nRF24	Low latency, hardware CRC, simple packet control
Control Architecture	Interrupt-based, Polling Loop	Polling Loop	Predictable timing and simpler synchronization
Servo Control Method	Software PWM, Hardware Timer	Servo Library (Hardware Timer)	Stable pulse timing with minimal overhead

Figure 6.3

VII. Performance Evaluation

The performance of the GRIP system was tested through a series of tests designed to measure latency, accuracy, communication reliability, and motion stability. All tests were conducted under normal operating conditions to simulate real-world usage scenarios.

A. Latency Analysis

The performance of the GRIP system was evaluated by measuring end-to-end latency from finger motion input to servo actuation. The system achieves a measured latency of approximately 12 to 18 milliseconds under normal operating conditions. Latency contributions are divided across several stages, including sensor processing, wireless transmission, receiver validation, and servo actuation. Signal processing and wireless transmission introduce only a few milliseconds of delay, while the primary contributor to total latency is the servo refresh cycle, which operates at 50 Hz (20 ms period). Due to the asynchronous nature of command arrival relative to the servo update cycle, the effective actuation delay varies within this range. Despite this, the system remains well within acceptable limits for real-time human motion tracking.

B. System Update Rate

The system operates at a stable update rate of approximately 50 Hz, corresponding to a 20-millisecond control loop. This rate is sufficient for capturing natural human finger motion, which typically occurs below 10 Hz. By sampling at a higher rate than the motion frequency, the system achieves smooth and continuous tracking without noticeable lag. This ensures that the robotic hand closely replicates user movements in real time.

C. Wireless Data Integrity

Reliable wireless communication is achieved through the use of hardware-level CRC validation provided by the nRF24L01 module. This mechanism ensures that corrupted packets are automatically rejected before entering the processing pipeline. Testing confirmed that invalid packets do not propagate to the actuation stage, preventing unintended or unsafe motion. This contributes significantly to the overall robustness of the system.

D. Motion Stability

Motion stability was evaluated by observing the system under continuous operation. After tuning the signal filtering and control pipeline, the system demonstrated smooth and stable motion without visible jitter.

A balance between filtering and responsiveness was achieved to ensure that noise reduction does not introduce excessive delay. This tradeoff is critical for maintaining both stability and real-time performance.

The responsiveness of the GRIP system is further illustrated in Fig. 7.1. The figure represents the timing relationship between input motion, signal processing, wireless communication, and servo actuation. It provides a visual representation of how the system maintains real-time performance through a continuous control pipeline.

As shown in Fig. 7.1, the system operates in a periodic loop driven primarily by the servo refresh rate of about 50 Hz. Each cycle begins with sensor data acquisition, where the flex sensors capture the current finger positions. These analog signals are then converted into digital values by the MCU's converter and immediately processed to determine the corresponding joint angles.

Following processing, the joint angle data is packaged into a compact wireless packet and transmitted to the robotic hand subsystem. The transmission occurs within a few milliseconds, contributing minimally to the total system delay. Upon reception, the data is validated and passed to the control logic responsible for generating a pulse with modulation signals.

The final stage shown in Figure 7.1 is servo actuation, where the signals are applied to the motors. Due to the nature of the servo control cycle, actuation occurs at fixed intervals. This introduces a bounded delay that depends on when the command arrives relative to the update cycle. This explains the latency variation between 12 and 18 ms.

The figure also highlights the overlap between system stages. Indicating that sensing, communication, and actuation are not sequential but instead operate in a pipeline manner. This overlapping behavior is critical for achieving smooth and continuous motion as it allows new input data to be processed while previous commands are still being executed.

Overall, Figure 7.1 demonstrates that the GRIP system maintains a stable and efficient real-time control loop where each subsystem contributes to overall performance. The visualization reinforces that system latency is dominated by actuator timing rather than communication or processing delays.

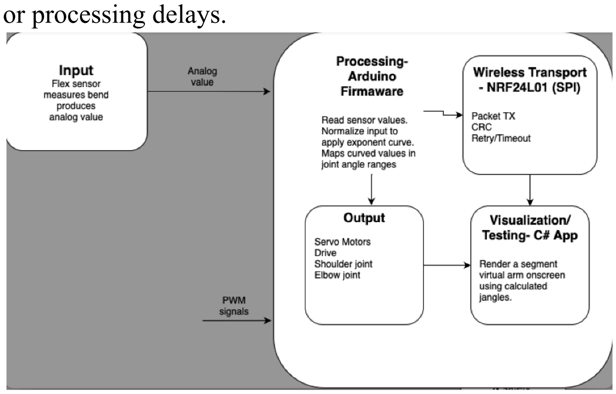


Figure 7.1

E. Packet Loss Handling

The system includes a failsafe mechanism to handle packet loss or validation failure. In such cases, the system maintains the last valid servo position instead of applying new data. This approach prevents sudden or erratic motion caused by missing or corrupted packets. As a result, the system maintains stable behavior even under imperfect wireless conditions.

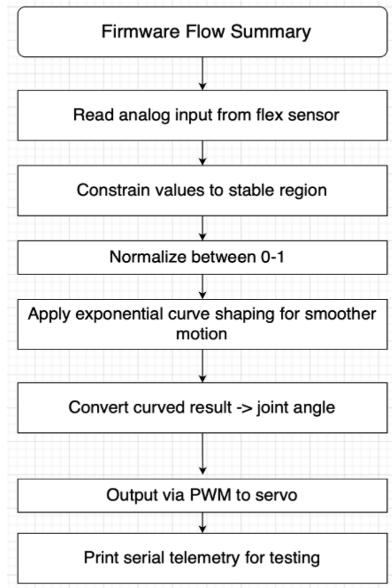


Figure 7.2

VIII. Results / Discussion Section

Overall, the GRIP system meets its design objectives for low-latency, reliable wireless control. The system achieves real-time performance while maintaining data integrity and stable actuation. The primary limitation in system responsiveness is the servo refresh cycle, which defines the minimum achievable latency.

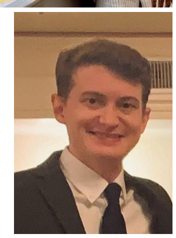
The GRIP system demonstrates real-time motion replication with minimal latency and high reliability. The measured latency of approximately 12 to 18 milliseconds is well within acceptable limits for real-time human interaction. One of the most significant findings is that the primary limitation in system responsiveness is not wireless communication or processing time, but the servo refresh rate. This highlights an important design consideration for future improvements. The system shows strong robustness against communication errors. The use of packet handling ensures that corrupted data does not propagate to the actuation stage. While the project performs well overall, areas can be improved, such as positional accuracy, reducing power consumption, and enhancing mechanical design for greater durability.

IX. Conclusion

In conclusion, the Glove for Real-time Imitation and Performance successfully demonstrates a low-latency system that implements robotic control using wearable input. The system presently provides accurate motion replication, an expandable interchangeable architecture, and wireless communication. Future improvements can be expanded into full arm integration, enhanced precision, and advanced control algorithms. GRIP lays the foundation for next-generation human interaction and shows the potential for a system that can allow pilots of ROVs to have more meaningful interactions in the environment in which they are deployed.

The Team

Rian Lowery - A 22-year-old who is graduating with a bachelor's in computer engineering at the University of Central Florida in the spring of 2026. Currently working at Siemens Energy as an IT intern who plans to have a permanent position in the future.



Christopher Marcoccia - A 24 year-old graduating with a bachelor's in Electrical Engineering. Currently working in theme park control systems and animatronics, with the goal of pursuing a career in prosthetics.

Kane McAgy - A 24 year-old Electrical Engineering Student on the general track, career goals to work on transmission controls or signals at a large firm.

Griseld Xhengo - is a senior at the University of Central Florida and is planning on graduating with a bachelor's in Electrical Engineering in the spring of 2026. After graduation, he plans on beginning his career in electrical engineering.

Figure List

Acknowledgement

The authors wish to acknowledge the assistance and support of the IMS Steering Committee.

References

[1] IEEE Standards Association, "IEEE Robotics and Automation Standards," 2023.
[2] Arduino, "Arduino Nano Documentation," Arduino.cc
[3] Nordic Semiconductor, "nRF24L01 Product Specification"
[4] Servo Motor Datasheet (MG996R or your model)
[5] Flex Sensor Datasheet (Spectra Symbol)