ECE/COE 1896

Senior Design

Force Feedback Telepresence Robot Conceptual Design

Team#1

Prepared By: Kevin Quigley
Seth Blain
Micah Smith

Table of Contents

Table of Figures

1. Introduction

The Force Feedback controlled Telepresence Robot is a multi-module robotics project aimed at providing a safe resource to be used in dangerous and/or distant work environments. At a basic glance this project can be broken down into four main modules: A force feedback glove (FF-Glove), virtual reality (VR), a camera for remote vision, and a humanoid robotic arm and hand. When fully integrated a user will be able to wear the FF-Glove and VR headset to teleoperate a robotic arm and hand and “feel” what the robotic hand grabs. The user will be able to “see” through the robot's perspective in VR, control the robotic hand as if it was their own, and “feel” the objects that the robotic hand picks up via the force feedback on the glove.

This system will address a couple of needs in the telepresence robotics industry. The first area this system sets out to cover is the cost/accessibility. Many telepresence robots on the market currently cost upwards of $15,000, and that's typically just the cost of the arm itself. This design will include the robotic arm, robot head (neck and camera), as well as the FF-Glove for under $200. Of course, there is also the expense of the VR headset which varies depending on the model; in this case we have elected to use the Oculus Quest 2 as it is the most owned VR headset in 2025 and comes with an average consumer price tag. All the mechanical features of the project will be 3D printed besides bolts and nuts which also helps keep the cost minimal and the project accessible.

Another need in the telepresence robotics industry that this design will address is the number of unique features. Our system utilizes force feedback gloves to allow the user to control the robotic arm and hand using just the FF-Glove. Furthermore, these gloves allow the user to feel objects that the robotic hand grabs. In comparison to telepresence robotic systems on the market, these are rare features. A vast majority of telepresence robotic arms that can be controlled using VR utilize the controllers that come with the VR headset to control the robotic arm. This is done by mapping the arm’s movements to buttons and joysticks on the VR controller which is unintuitive and does not allow the option of force feedback. Furthermore, the robotic hand that is interfaced with these controllers is often a claw, just two planes that can be squeezed together. By utilizing an FF-Glove to control a humanoid robotic hand, the need for the VR controller is obsolete and the user can simply put on the glove and intuitively control the robot appendages as if it's their own.

A major reason why this prototype was proposed was for its vast applications. The most valuable application of this prototype is for occupations that put employees' lives at risk. For example, this prototype could take someone working on a bomb squad out of a potentially life ending scenario but still allow them to utilize their expertise and motor skills to remotely disarm a bomb. Similarly, this prototype could be implemented in mines to remove people from the harms of dangerous gases or to be used by a surgeon who is needed in a different location instantly.

2. Background

This project addresses the problem of unsafe or distant working conditions for tasks requiring fine motor control. Many industries, from laboratory science to construction and industrial maintenance, require humans to operate in hazardous environments where exposure to chemicals, physical danger, or unstable surfaces can cause injury or death. Even with personal protective equipment (PPE) and safety regulations in place, the risks remain high and the costs—both human and economic—are substantial. Our design seeks to reduce these risks by removing the human operator from direct exposure while maintaining the same fidelity and quality of work through a teleoperated robotic arm controlled by a force-feedback glove.

Occupational exposure to hazardous substances and dangerous environments remains a leading cause of premature death and injury worldwide. A study conducted by the National Library of Medicine reported that occupational exposure to hazardous chemicals contributes to more than 370,000 premature deaths annually between 1999 and 2014 [1]. The World Health Organization (WHO) further estimates that globally, nearly 2 million people die each year from work-related causes, with air pollution, chemicals, and occupational exposures being among the largest contributors [2]. In the WHO European Region alone, exposures to chemicals such as lead, pesticides, and industrial particles accounted for over 2 million deaths and 53 million disability-adjusted life years (DALYs) in 2019 [3].

Beyond mortality, chemical exposures often lead to long-term health consequences, including chronic obstructive pulmonary disease (COPD), cancer, cardiovascular disease, and neurological disorders [4]. In the Americas, occupational carcinogens such as asbestos and silica are estimated to cause 200,000 deaths annually [5]. Even short-term exposures can cause acute respiratory distress, neurological impairment, or other severe health issues, highlighting the ongoing need for solutions that reduce human contact with hazardous substances.

Physical accidents also contribute heavily to occupational risk. In the United States alone, 211,640 severe injuries from falls were reported in 2020, along with 805 deaths [6]. The construction industry is particularly vulnerable: in 2022, about one in five workplace deaths occurred in construction, and 38.4% of those deaths were due to falls, slips, and trips [7]. Nonfatal incidents are similarly costly, with injuries from falls to a lower level resulting in a median of 28 days away from work [8]. Slip, trip, and fall incidents cost U.S. businesses tens of billions annually in lost productivity, workers' compensation, and healthcare expenses [9].

Tasks requiring fine motor control amplify these risks. When workers must conduct precise manipulations—such as handling delicate instruments in a laboratory, performing maintenance in confined spaces, or assembling components near hazardous materials—human proximity to danger becomes unavoidable. Protective equipment often reduces dexterity and sensory feedback, while fatigue and reaction time limit human performance, creating an environment where accidents are more likely to occur. A solution that allows humans to conduct such tasks remotely, with full dexterity and feedback, would reduce these risks while maintaining work quality.

Designing such a system requires integrating knowledge across several domains:

- **Robotics and Teleoperation:** Remote manipulators must accurately translate human motion into robotic motion. Challenges include latency, precision, and achieving stable force feedback.
- **Human Factors and Neuroscience:** Effective systems must account for operator fatigue, perception of delay, cognitive load, and the importance of intuitive, ergonomic control interfaces.
- **Health and Safety Sciences:** Understanding toxicology, exposure thresholds, and the long-term impacts of hazardous environments underscores the value of reducing human contact.
- **Mechanical and Control Engineering:** Manipulator design, sensor integration, and safe feedback loops are critical to achieving the fine control required.
- **Regulations and Standards:** Systems must comply with safety guidelines such as OSHA regulations and international standards for hazardous material handling.

Several existing technologies demonstrate the promise and limitations of teleoperation. For example, bomb disposal robots such as the Remotec ANDROS MarkV-A1 and India's DRDO "Daksh" remove humans from immediate danger but are bulky, specialized, and lack fine motor fidelity [10,11]. In research contexts, haptic teleoperation systems have been tested for space and hazardous environments. Studies show that while force feedback improves accuracy under low latency conditions, its benefits diminish—and may even harm performance—under high communication delays [12].

Recent advances, such as the Gesture Articulated Meta Operative Robotic Arm (GAMORA), demonstrate how virtual reality, digital twins, and real-time force feedback can achieve millimeter-scale accuracy in biohazard laboratories [13]. Yet even these cutting-edge systems remain expensive, specialized, and not easily adaptable across industries. Latency, cost, and operator fatigue continue to limit widespread adoption.

Despite decades of incremental improvements in workplace safety and robotics, current solutions often fail to balance safety, fidelity, adaptability, and cost-effectiveness. PPE and safety training reduce risk but cannot eliminate it. Specialized robots protect workers in narrow domains but cannot generalize to the many industries where fine motor work in hazardous environments is required. High-end teleoperation systems demonstrate technical feasibility but remain prohibitively expensive or limited by latency challenges.

Our design aims to bridge these gaps by creating a telepresence robotic system that:

- Preserves fine motor fidelity through force-feedback control.
- Removes the operator from direct exposure to hazards.
- Provides intuitive, ergonomic control that reduces fatigue.
- Is modular and adaptable across multiple industries.
- Prioritizes cost-effective components to enable broader adoption.

By combining advances in haptic teleoperation, control systems, and human-centered design, our solution can help protect workers from hazardous environments while maintaining or even improving the quality of work performed.

3. System Requirements

This project is designed to create an immersive experience for a single user. The user is intended to put on the force-feedback glove and Oculus Quest 2, then upon opening an app in the Quest 2 the user can connect to the telepresence robot. Once connected to the robot the user can "see" through the point of view (POV) of the robot by accessing the live video feed captured by the robot's "head". From there, the user can intuitively move their head, arm, and hand and the robot will automatically mirror all their movements. Below are the functional requirements and non-functional requirements for each module to ensure that the user has an experience as mentioned previously.

3.1 Robotic Arm

3.1.1 Functional Requirements

The robotic arm must have six degrees of motion. The robotic arm must closely mirror the motion of the human user with as little latency as possible. The robot will be able to pick up small objects (< 5 pounds) with enough strength to lift it off of a surface. When the robot has something gripped in its hand, it must be able to communicate that with the glove to restrict the user's motion, creating haptic feedback. The robot will be powered by a wired connection to a standard AC wall outlet. Since there will be 11 servos motors to power in total, a highly-rated power supply is required. The MG996R motors we've selected have a stall current of 2.5A each, so an AC/DC adapter rated for over 30A at 6V is required.

3.1.2 Non-Functional Requirements

The arm does not need to move in ways that a human arm cannot.

3.2 Force Feedback Glove

3.2.1 Functional Requirements

The force feedback glove must be a standalone device. This means that the device can work wirelessly and uses its own power supply, such as a battery. There will be 5 motors on the glove, each with a stall current of 2.5A. For safe operation, the battery must be rated for 15A to factor in overhead. The circuitry on the glove must be enclosed or electrically isolated to prevent exposure of electrical signals to the user. The glove must track one degree of motion for each individual finger (this angle is measured at each knuckle). The glove must also be able to restrict the closing motion of each individual finger separately upon a digital signal. Communication between the glove and robotic arm will be entirely wireless, utilizing Bluetooth or Wi-Fi to increase the range with which the system can be used.

3.2.2 Non-Functional Requirements

The glove should be comfortable so as not to interfere with the users' experience and allow for prolonged use. It should be made from a material that is not itchy or too warm. The glove should weigh less than 3 pounds, so it is not burdensome to wear.

3.3 Virtual Reality Headset

3.3.1 Functional Requirements

The virtual reality headset must be able to be worn by the user to see the view of the robot. The headset must be able to be calibrated with the glove in order to properly track the position of the user's hand. The headset must send the position of the glove to the robotic arm for robotic arm movement calculations. The headset must send its orientation to the camera system to ensure mirrored movements. The communication between the VR headset and the camera system must be wireless, so that it can be controlled remotely.

3.3.2 Non-Functional Requirements

The VR headset should be comfortable so as not to interfere with the users' experience. The VR headset should be able to be bought and used as is, i.e. no modification required.

3.4 Robotic Head - Remote Camera Vision

3.4.1 Functional Requirements

The camera must move to mirror the movements of the user of the VR headset. The camera will be housed on an artificial neck which will be controlled with 3 servos motors, allowing it to move on all axes. Any latency between the user's head movement and the response from the camera must be minimal, to enhance the effect that the user is at the location of the robot. Communication between the headset and camera controller will be wireless, so it can be operated at a distance. Power for the camera and 3 motors will come from the same or similar adapter as to what will power the robotic arm. The camera base will remain stationary where the robotic arm is set up, so wireless power is not required. A smaller power supply could be used since there are only 3 servos to be powered.

3.4.2 Non-Functional Requirements

All physical parts for the physical neck can be 3D printed, allowing for better accessibility and cost of the design (except for electronic components and bolts).

4. Design Constraints: Standards and Impacts

4.1 Design Constraints

4.1.1 Time

One of the primary constraints for this project is the limited time that we have to design, prototype and test a haptic feedback glove integrated with a telepresence robotic arm in a VR environment. We have about 12 weeks to finish this project, while also having to balance the rest of our academic workload. This is a constraint which we have to take into heavy consideration when scheduling out what overall milestones have to be done at what times during the semester as well as individual progress on each of our modules. By planning out a realistic schedule for our project with individualized goals for each of our team members, we will ensure that we can stay on track with our progress. Delays in component shipping or long troubleshooting processes could slow down our progress so planning enough time to work around these challenges will ensure our success.

4.1.2 Manpower

Limited manpower is another crucial constraint in this project. This project has three main parts and each of those parts has a hardware and software component to them. Due to the design of this course, we have a group of 3 people to complete the design, create, and thoroughly test the whole project which may limit the full functionality of the end product. The goal is to have a fully functional haptic feedback glove integrated with a robotic arm, but due to our limited manpower, the precision of our glove tracking robotic arm movement may not reach the highest level of accuracy or responsiveness that would be possible with a larger team. Limited manpower may require us to prioritize certain features of the project while simplifying aspects of the design to ensure that we have the required functionality finished on time. It is crucial that we all stay on the same page over the course of this project and allocate tasks based off each other's strengths in order to ensure that we reach our goals.

4.1.3 Financial

Financial limitations also play a major role in constraining our project. For our senior design project, we are given a $200 budget to support all of our expenses. A high-quality haptic feedback glove and robotic arm system would normally cost far more than this budget due to the high price of precise motors and sensors which allow the system to work. In fact, most commercially available haptic feedback VR gloves cost thousands of dollars so we will have to carefully manage our budget in order to stay under $200. Fortunately, our team already owns VR headsets and we have access to many of the components necessary for completing this project. Additionally, we have identified cheap alternatives to many otherwise expensive parts. These factors will aid us in staying under the budget and allow us to create functional system without any additional financial assistance.

4.1.4 Mechanical

The physical and mechanical limitations of the haptic feedback glove and robotic arm cause significant constraints on this project. The main limitation that arises is the range of motion of the robotic arm. There will be a several different users of this project and the robotic arm we are using has a fixed size. This means that the motion of the robotic arm cannot physically mirror the exact movement of the user due to the difference in limb length, flexibility, and natural motion path. The robotic arm will also be constrained to the speed of the motors meaning that if you move your arm too quickly, the robotic arm will not be able to move at the same speed due to the mechanical limits of the arm. As a result, the software implemented in the robotic arm will have to translate the user's motion and move the robotic arm to the closest position possible to that of the user at a safe speed, rather than a perfect one-to-one motion.

Additionally, the glove itself must remain light and comfortable enough to be worn for an extended period of time while still housing servo motors, wiring, etc. Excessive stiffness or bulk could significantly reduce the rand of motion and overall usability of the glove.

4.1.5 Power

To be able to grip and lift up even small objects, high-torque servo motors are required. These high-torque motors are also required for the force feedback on the glove to accurately simulate the sensation of holding onto an object and restrict the user's movement. High-torque servo motors draw more current than normal servo motors do and therefore require a more powerful supply. If the motors are not supplied with sufficient current, they will not operate at their maximum capacity and won't be able to lift as much weight. We are constrained by the size of power supply required to run the 11 motors on the robotic arm. We need a powerful AC/DC converter that can take wall power and supply it to the motors. A large battery could also accomplish this, but the financial constraints of the project prohibit us from affording a battery that would be capable of this load.

4.1.6 Communication

This system relies on seamless communication between multiple devices which leads to significant constraints. The ESP32 microcontroller on the haptic feedback glove will send wrist and finger movement information wirelessly via bluetooth directly to the ESP32 on the robotic arm which will be communicating back with haptic feedback signals so the user can "feel" any objects they touch. Simultaneously, the Quest 2 controller mounted on the glove and the Meta Quest 2 headset will be communicating position information with the Unity application running on the PC to handle motion tracking. The PC must then send the headset information to the Raspberry Pi in the camera system and controller information to the robotic arm so it can calculate how to move the arm. This layered communication system introduces several different challenges. The ESP32s will be subject to latency and interference which could prevent smooth mirroring of the user's movement. Since there are so many layers of communication, it is essential that we send data efficiently in order to reduce latency and have the best mirroring of movements possible.

4.2 Impacts in Non-Technical Contexts

4.2.1 Environmental

The adoption of a telepresence robotic arm could reduce the need for human presence in hazardous environments such as chemical plants, nuclear facilities, or remote field sites. By enabling remote operation, unnecessary travel to distant or unsafe job locations could be minimized, lowering fuel consumption and associated greenhouse gas emissions. Additionally, since the robot can withstand repeated exposure to harsh conditions without the need for extensive protective gear or safety infrastructure, fewer disposable protective materials (e.g., respirators, suits, gloves) may be required. However, the environmental impact of manufacturing and powering robotic systems must also be considered. Responsible material sourcing, energy-efficient operation, and compliance with sustainability standards (e.g., RoHS, WEEE) will be important to mitigate negative impacts.

4.2.2 Public Health

The primary public health impact of this project lies in reducing worker exposure to harmful environments. By removing humans from direct contact with toxic chemicals, high-radiation zones, or structurally unsafe worksites, risks of occupational illness and injury can be substantially lowered. This technology could also support emergency response efforts, such as handling hazardous spills or conducting rescues in structurally compromised buildings, thereby reducing risk for first responders. A potential negative impact could be increased sedentary behavior among operators, as remote work of this type requires prolonged screen time and limited physical activity, but this can be mitigated through ergonomic design and usage guidelines.

4.2.3 Global, Cultural, and Societal

Globally, telepresence robotics could expand access to specialized labor and expertise across borders, allowing highly skilled workers to contribute to tasks in remote or developing regions without physically relocating. Culturally, this could help balance disparities between regions with advanced technical infrastructure and those that lack it. Societal benefits include improved worker safety, enhanced efficiency, and the ability to maintain operations in industries critical to infrastructure and health. On the other hand, adoption of such technologies may face cultural resistance in workplaces that value in-person, manual labor traditions. Careful integration, training, and demonstration of benefits will be important to achieve broad acceptance.

4.2.4 Diversity, Equity, and Inclusion

This system could broaden participation in technical and hazardous work by enabling people who may be physically unable to enter dangerous environments to contribute effectively through remote operation. For example, individuals with mobility impairments or health restrictions may still be able to work in fields such as manufacturing, research, or hazardous materials handling using the telepresence system. By lowering the physical barriers to participation, this technology could support more inclusive workforce opportunities. However, equitable access must be considered—if the system is prohibitively expensive or limited to certain organizations, it could exacerbate inequities by concentrating safer jobs in wealthier companies or nations.

4.2.5 Welfare and Safety

At its core, this design directly enhances worker welfare and safety by reducing exposure to physical, chemical, or environmental hazards. Industries such as mining, firefighting, disaster recovery, and laboratory research stand to benefit significantly. For example, workers could conduct inspections in environments with high structural collapse risks without ever being physically present. At the same time, safety standards will need to address risks posed by the robot itself—such as unintended motion or failure—that could endanger nearby humans. Developing safeguards and compliance with occupational safety standards (e.g., OSHA regulations) will be critical.

4.2.6 Economic

From an economic perspective, the ability to remove humans from dangerous worksites can lead to reduced healthcare costs, fewer worker compensation claims, and less downtime due to injuries or fatalities. On a community level, this could mean more consistent employment and fewer economic disruptions from workplace accidents. At the corporate level, companies may see reduced training and insurance costs while maintaining high productivity. Nationally, widespread adoption could contribute to improved worker retention, less economic loss due to occupational hazards, and greater competitiveness in industries that rely on precision work under hazardous conditions. Conversely, there is a risk of job displacement in industries where robotic systems may replace rather than augment human labor. Thoughtful implementation strategies—emphasizing remote operation rather than automation-only approaches—will help balance economic benefits with workforce stability.

5. Conceptual Design

Since there are distinct modules to this solution it is easiest to describe the critical functions that a customer will gain from each module and then lastly how everything will communicate.

A single FF-Glove that has one point of tracking per finger as well as one restrictive force per finger. The FF-Glove is expected to work in place of the Oculus Quest 2 controller. An Oculus Quest 2 VR headset that is ready to work in this system after the installation of an app from the Steam library and an external driver. A webcam camera that displays a 1080p live video feed to the user in VR. The webcam is expected to mirror the movements of the users head to provide a realistic view of the robot's POV. Then there will be a humanoid robotic arm that has 6 degrees of freedom and individually controlled finger movements. The humanoid robotic arm will mirror the movements of the user's arm and hand in all directions. When the robotic arm grabs something, it will detect that and restrict the movement of the fingers on the glove, providing the force feedback.

In terms of communication between modules, there must be a wireless connection between the robot (webcam and robotic arm) and the user wearable items (FF-Glove and VR headset).

5.1 Design Concepts

Below are two different conceptual design ideas of the system that each effectively complete the functional description mentioned above, in section 5. As mentioned previously in this document, there are four distinct modules (FF-Glove, VR, Camera Vision, and Robotic Arm) in this solution regardless of the design concept. To emphasize the most significant differences between each design concept there will be subsections for each of these modules detailing the specific implementation of them.

5.1.1 Design Concept 1

5.1.1.1 Overview

Design Concept 1 focuses on utilizing as much opensource hardware mechanical designs as possible following the overall system described above.

5.1.1.2 Force-Feedback Glove Implementation

In this design concept we elected to use an open-source FF-Glove to minimize any unnecessary hardware design that would detract from the ECE focus of this project. The open-source FF-Glove to be used in this design concept is from LucasVR. This glove utilizes five identical nodes that handle finger tracking and force-feedback for each individual finger. Signals coming in and out of each of these nodes are processed by an ESP32.

To expand upon one of the nodes on the glove we must understand the components within it. The three components being used in each node include a servo motor, potentiometer, and a spring-loaded retractable string. The end of the spring-loaded string is attached to the user's fingertip and spring is wrapped around the potentiometer. When the user curls or "closes" their finger the string is pulled and hence rotates the potentiometer to record the angle of your finger measured from the knuckle (location of potentiometer). Then when the user opens that finger up again the spring automatically retracts the string and rotates the potentiometer back into an angle indicative of your finger's position. Lastly, the servo motor is attached directly to the string/potentiometer to apply the force feedback. A circuit diagram of the whole glove can be seen in Figure 1 below.

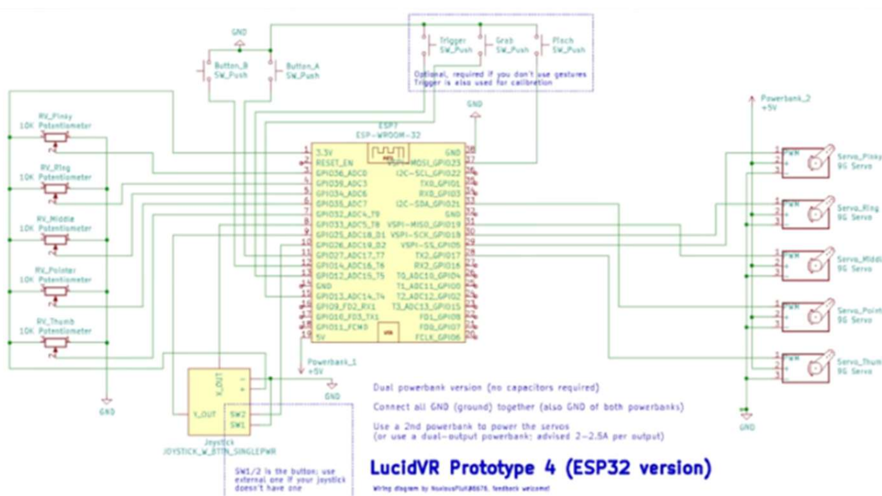

Figure 1: Circuit Diagram of LucidVR Prototype 4 FF-Glove

To track the gloves position in 3D space a controller from the Oculus Quest 2 is mounted on the back of the glove. This is a clever way of utilizing the 3D tracking of the

Quest 2 rather than implementing something like an IMU (inertial motion unit) to calculate the user's hand in space.

Since the glove utilizes an esp32 microcontroller the firmware for the signal processing is all done in c++ through the Arduino IDE and can be ran either using the Bluetooth feature built into the esp32 or using a USB cable for serial communication. The analog signal received from each of the potentiometers is converted to an angle and concatenated into a string. The serial monitor continuously updates with the updated readings. Then Lucas from LucasVR helped create an app on Steam called OpenGloves. This app allows the user to select their glove by the name associated with the microcontroller and essentially tells Steam VR that the microcontroller is the Oculus Quest controller. Once the FF-Glove is in VR the app interprets the data from the serial monitor and moves your virtual fingers to match the same position as your hands in real life.

5.1.1.2.1 Force-Feedback Glove Power Design

Since the glove will be worn by the user and designed to be extremely mobile, it needs to be completely standalone and not connected to any stationary object via wires. With that in mind, we will power the glove using batteries so that it can be worn comfortably and able to move freely. The glove has 5 MG996R high torque servo motors which have a maximum stall current of 2.5A each. For all 5 motors to run simultaneously, the power needs to be able to supply up to 12.5A with room for safety overhead.

5.1.1.3 Virtual Reality Implementation

For this design concept an Oculus Quest 2 will be the VR headset of choice. This is because two of the three members of the group own a Quest 2 which is beneficial for budgeting and development purposes. Another factor that contributed to this being the favored headset is because it is the most owned VR headset in 2025 so it helps us with our goal of creating a cost-effective system for other users.

The user will use the VR headset to be able to see the real-life environment where the robotic arm is operating. A camera will send a live feed of the robot's "view" and stream that directly to the headset the user is wearing. This will provide an immersive experience and allow the user to feel like they are actually present in the environment where the robot is working, but out of harm's way. This also provides a more accurate and realistic vision of the robot's surroundings than an augmented reality render would, for example.

5.1.1.4 Camera Vision/ Head Tracking

This module focuses on the robot's "neck" and "eyes". The goal of this module is to produce a live camera feed from the robot's POV and steam it directly into VR. Furthermore, the camera will be able to move in 3 degrees of motion to mirror the movement of the user's head to provide a realistic viewing experience.

The design of this module consists of using a 1080p USB webcam as the eyes of the robot. Interfacing with this camera will be a Raspberry Pi 4 Model B. This specific model of Raspberry Pi was selected as it has USB3 ports which will be used to directly capture the video feed from the webcam. Also, the Pi has an H.264 encoder built in which makes it easy to work with the 1080p picture in 30fps. This will help streamline the process of iteratively building the camera vision module itself and integrating it with other modules. Once all modules are integrated, software will be developed to integrate 60fps for a smoother viewing experience in VR.

To produce the video feed captured on the Raspberry Pi in VR we will be using WebRTC (Web Real-Time Communication). WebRTC is an open-source project and collection of standards that enables real-time communication, including audio, video, and data exchange. Software will be constructed to utilize the captured video feed as a WebRTC peer source to be displayed on a server/streamer. From there the Oculus Browser supports WebXR and supports modern web APIs needed for WebRTC which will allow the incoming video to be placed on a WebXR layer. This essentially allows us to create an entirely browser-based VR "app". In virtual space the app can be locked to the user's head so the video feed is always directly in front of their face as if the video feed is their actual vision.

Up to this point the user can see through the POV of the robot at a fixed camera angle. To allow the user to move their head freely and have the camera's angle change to match, an external mechanism is required. For simplicity the mechanism will be referred to as the robot's "neck". The neck must account for 3 degrees of movement (pitch, yaw, roll) in order to replicate any movements made by a human head. Figure 2 below demonstrates the aforementioned degrees of freedom and their relation to the human head.

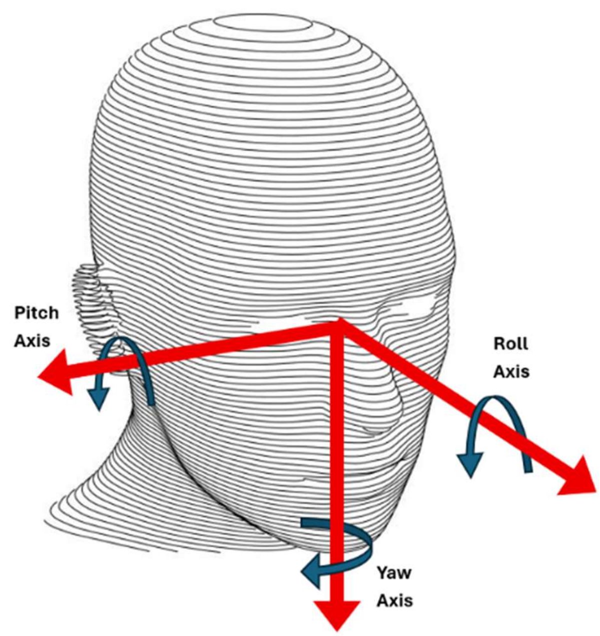

Figure 2: Degrees of Freedom in Reference to the Human Head

In order to achieve these degrees of freedom a tripod-like structure will be used. Figure 3, Figure 4, and Figure 5 below display the full neck mechanism.

Figure 3: Robotic Neck Mechanism - Side View

Figure 4: Robotic Neck Mechanism - Front View

Figure 5: Robotic Neck Mechanism - Exploded View

Based on the figures above a few observations can be made. Firstly, attached to the neck are 3 MG 996R Servo motors; Two in the front attached to the two front-most arms of the tripod structure, and one located in the back that faces down. Also, the third arm of the tripod at the back is a passive element to provide a pivot point for the head plane.

Taking a closer look at the two servos towards the front of the neck that are attached to two of the three tripod legs, we can identify how the roll and pitch axes of motion will be handled. When both servos pull the arms down or up at the same rate pitch can be achieved (view Figure ... below). On the other hand, if they are moved to differing angles roll is achieved (view Figure ... below). Attached at the top of each tripod arm is a universal joint which translates the rotational motion from the servos into linear motion. See Figure 6 below to view the universal joint mechanism.

Figure 6: Robotic Neck Mechanism - Universal Joint

The third servo motor, located at the rear of the neck, faces down and drives a 24-tooth gear. This acts as a planetary gear revolving around the large stationary gear of the baseplate that the neck rests on. This is what allows the neck mechanism to achieve motion around the yaw axis.

5.1.1.4.1 Camera Vision/ Head Tracking Power Design

The "neck" design utilizes 3 MG996R high torque servo motors. The maximum stall current for these motors is 2.5 A at 6V, leading to a maximum current draw of 7.5A for the neck. The robotic arm itself will be driven by a power supply that will plug into the wall. Since the camera will be situated with the robotic arm and stationary apart from its own independent 3-axis movement. Since it will not be mobile, we will also power the neck with the power supplies that receive power from a wall outlet. The neck will have its own power supply so as not to increase the load on the already high current power supply used for the robotic arm.

5.1.1.5 Robotic Arm

5.1.1.5.1 Robotic Arm Mechanical Design

This module focuses on the robotic arm, extending from the shoulder all the way down to its fingertips. The arm is based on an open-source design with a few modifications to reduce time spent on mechanical design, and instead keep the project centered on electrical and computer engineering aspects.

This arm has 6 degrees of freedom, allowing it to closely mirror the movements of the gloved user. Its movements are concentrated at three points: the shoulder, elbow, and hand/wrist.

The shoulder joint has 3 servos as seen in figure 7 with each operating on a distinct axis of motion.

1. Lateral raise/lower - lifting the arm out to the side
2. Forward/backward raise/lower - moving the arm in front of the body

3. Rotation – twisting the arm toward or away from the body

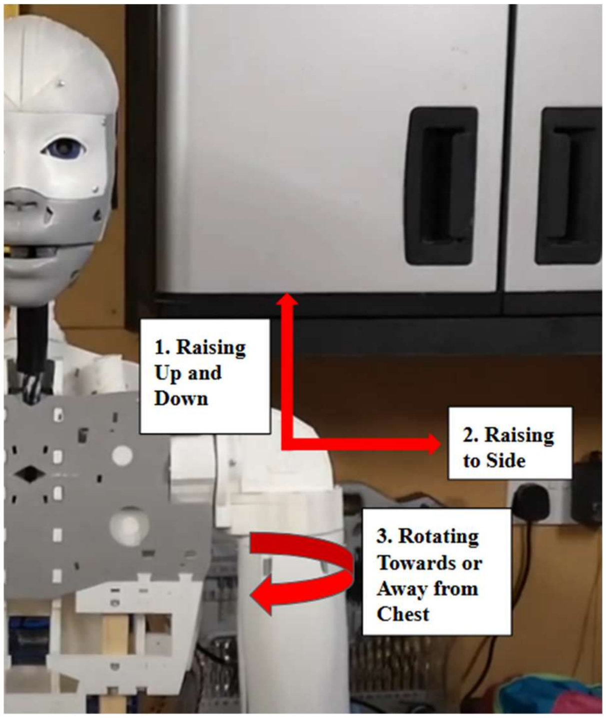

Figure 7: InMoov Robotic Arm - Shoulder Rotation Diagram

There is a servo in the bicep which allows elbow movement as seen in Figure 7. In the current open source version, there is one servo in the wrist which allows it to rotate, but we are designing a new forearm which allows the wrist to bend forward and backward which will incorporate one more servo. In the forearm, there are also 5 other servos which control the fingers by connecting to them with a braided fishing line, allowing the arm to grip and hold objects.

Figure 8: InMoov Robotic Arm and Hand

With these combined 11 servos, the arm has 6 degrees of freedom and individual finger control allowing it to have versatile movements and closely mimic the human arm.

For each joint, excluding the fingers, the potentiometer is removed from the servo motor and placed at the mechanical joint in order to accurately read what the joint angle. Otherwise the servo would be limited to 180 degrees and for many of our joints, there is wider range of motion. This allows us to accurately track arm positions without limiting range of motion.

5.1.1.5.2 Robotic Arm Software Design

The first design Figure 9 option regarding how the robotic arm moves is using software to interpret the force-feedback glove's position and use that information to decide how to move the robotic arm. The glove which we are creating will have a mount on it for a Meta Quest 2 controller. By using the Unity game engine and OpenXR (a standardized pipeline used to translate between the Unity and the headset). Using Unity, we will write a C# script which will grab the position and rotation of the quest controller relative to the VR headset wirelessly from the headset. This will give us three coordinates for the positioning of the controller and 3 angles based off the rotation of the controller. This will be streamed over to the stm32 which will use the position and rotational data to move the robotic arm.

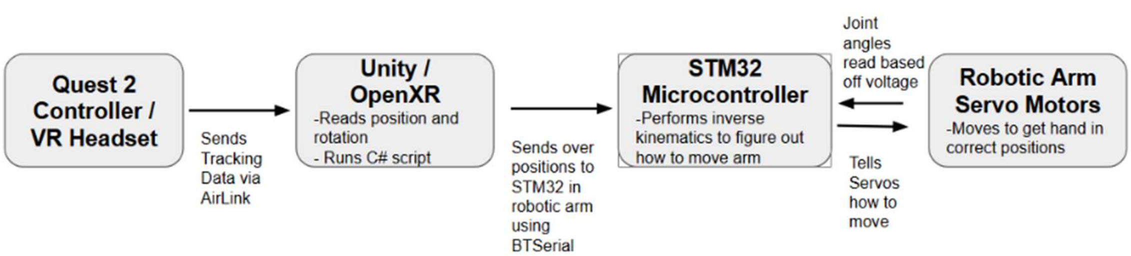

Figure 9: Design Concept 1 Robotic Arm Control

Using the rotational data of the Quest controller, we will know the orientation of the wrist of the glove and thus be able to mirror that position onto the robotic arm. In order to correctly move the robotic arm, however, we have to perform inverse kinematics (IK) which involves figuring out join angles/movements in order to get our robotic arm in the right position. Since we know the end position of the hand, but we have to figure out how to get there. We will use a method called cyclic coordinate descent (CCD) which is an iterative algorithm used to solve inverse kinematics by converging to a coordinate or in this case an angle.

CCD in this case will work as follows:

1. For each joint starting from the wrist up until the shoulder, compute the vector v1 from that joint to the end-effector (the robotic hand)

2. Compute the vector v2 from the joint to the target end position of the end-effector
3. Rotate join slightly to reduce the angle between v1 and v2
4. Repeat until the end-effector is close enough to the target point

5.1.1.5.3 Robotic Hand Software Design

Moving the robotic hand is much simpler as we will wireless send over the finger positions from the glove over to the robotic arm using the Bluetooth feature of the stm32s on the two devices. The servo motors in the forearm will rotate to curl the robotic fingers in order to mirror the position of the glove.

In order to control haptic feedback, the robotic hand must be able to "feel" objects. This will be done using a hall effect sensor and a small magent which are separated by a a soft pad material in each finger. When a robotic finger presses against something, the magnet will move closer to the sensor which will detect a stronger magnetic field. The STM32 will detect the change in voltage and send this information to the FF-Glove to restrict the user's hands. This can be seen in Figure 10.

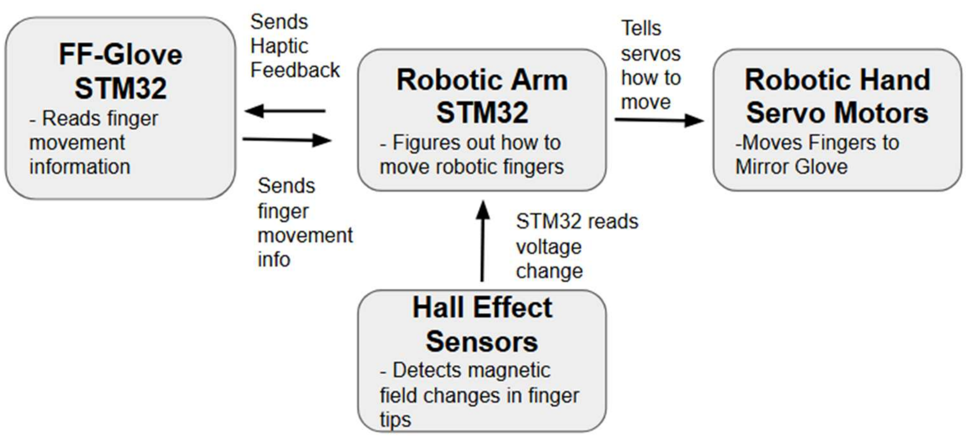

Figure 10: Robotic Hand Control

5.1.1.5.4 Robotic Arm Power Considerations

This design for the robotic arm includes 11 servo motors. The purpose of each of these motors is described in section 1.1.2.5.1 above. The servo motors we are using are MG996R high torque servo motors. The motors need to have a high torque since the arm will be grabbing and picking up objects of at least a few pounds (hopefully more) and need to be able to handle that load. The running current for these motors is between 500-900 mA at 6V, and the stall current is 2.5A at the same voltage. Since all 11 motors will be used simultaneously, a power source with a highly rated current is required. The maximum current draw may even require a second power source. Since this current is so high and the robot will be stationary, we will use a power source that can be plugged into a wall rather than running it off of batteries. While this limits the portability of the robotic arm, it will handle the high current better and prevent the need to constantly be replacing drained batteries.

5.1.2 Design Concept 2

5.1.2.1 Overview

Design Concept 2 is very similar to Design Concept 1 with a few differences. The modules that remain the same as in Design Concept one includes: The VR headset/implementation and camera vision/ head tracking. On the other hand, the main differences in this design concept involve using more hardware solutions to simplify the software that is needed in the project as well as using less open-source designs.

5.1.2.2 Force-Feedback Glove Implementation

In this Design Concept rather than using the open-source FF-Glove from LucasVR an FF-Glove designed by Kevin would be used. Briefly, the differences between these two gloves are the hardware choices. In Lucas's open-source FF-Glove the fingers are tracked using potentiometers and the force-feedback is applied using high torque servo motors to restrict the string that runs along each finger. In Kevin's design each finger is tracked by using a small neodymium magnet and a linear hall effect sensor while force feedback is applied by using solenoids. Figure 11 and Figure 12 below demonstrate what this design looks like on the user's hand.

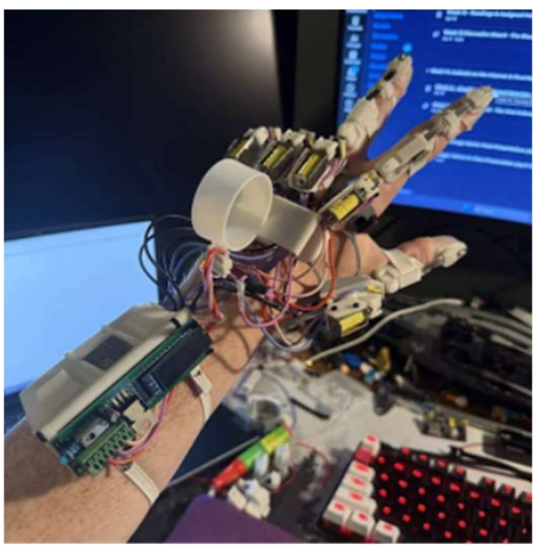

Figure 11: Kevin's Force Feedback Glove - Backhand View

Figure 12: Kevin's Force Feedback Glove - Front-hand View

The design consists of five "finger-nodes", each node contains ziptie tensioners (on the finger tip and along the length of the finger), a finger tracking sensor (hall effect) and force feedback (solenoid). Then, the finger nodes are placed on the backplate which is fastened to your hand using Velcro strips. By placing finger-nodes on the backplate using a 3mm bolt as a pivot axel the finger nodes can move freely in any direction (allows for the user to splay their fingers). Towards the back of the hand there are guide nodes for the wires and steel zipties so that nothing hits the users hands. On top of the guide nodes is a mount for a Quest 2 controller to be utilized for tracking your hands position in space in VR. Attached to the wrist is a power supply system and esp32 case. Withing the power supply system there is a 5V regulator to convert the 12V supply from the battery pack to a usable voltage for the esp32 and linear hall effect sensors. While there is a 5V regulator there is also a normal port which allows the 12V solenoids to be connected directly to the 12V supply. The esp32 attached to this unit runs the code and can communicate with the PC either over serial communication or over Bluetooth under the name kevs-glove".

5.1.2.2.1 Force-Feedback Glove Power Design

Another design could be to make the glove wired instead of battery-powered to reduce its weight, but that would make the glove much less mobile and diminish the user's experience.

5.1.2.3 Virtual Reality Implementation

The virtual reality implementation for this design concept will remain the same as in Design Concept 1. This means that an Oculus Quest 2 headset will be used for development as it is what is accessible to the group.

5.1.2.4 Camera Vision/ Head Tracking

The camera vision/ head tracking design will remain the same as in Design Concept 1. This is because we have already purchased a webcam and the mechanism to control the robot's head movement with 3-degrees of freedom has been built and works as expected.

5.1.2.4.1 Camera Vision/ Head Tracking Power Design

See section 5.1.1.4.1.

5.1.2.5 Robotic Arm

5.1.2.5.1 Robotic Arm Mechanical Design

See section 5.1.1.5.1.

5.1.2.5.2 Robotic Arm Software Design

The robotic arm itself remains the same for this design concept 2. What differs is how the arm is controlled (See Figure 13). Instead of VR tracking, we would extend the glove to have an exoskeleton wrapping around the rest of the user's arm. Potentiometers would be attached at the joints of the wrist, elbow, and shoulder, mimicking the placement of the potentiometers of the robotic arm . The readings of the potentiometers would be sent from the esp32 in the glove directly to the esp32 in the robotic arm. Based off these readings, the robotic arm would know exactly how to move in order to be in the same position as the user.

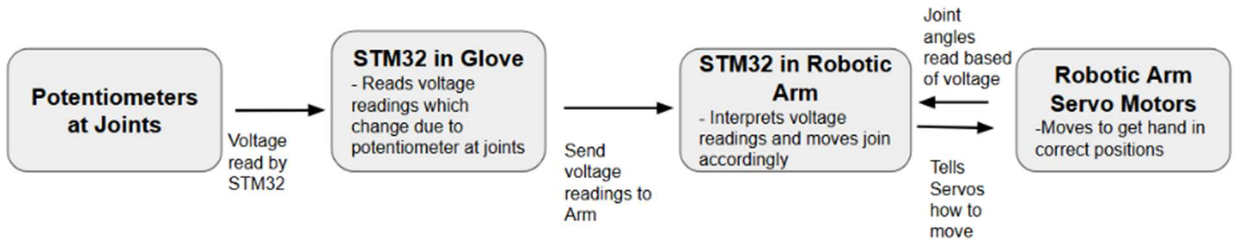

Figure 13: Design Concept 2 Robotic Arm Control

5.1.2.5.3 Robotic Arm Power Considerations

The mechanical design of the robotic arm would stay the same for design 2, but for increased mobility the arm would run off batteries instead of a wall plug adapter. This way, the arm could be moved around much easier making it quicker to switch between tasks. Since the design would still incorporate 11 servo motors, the battery needed to run this would have to be large and likely expensive. There would be a tradeoff of cost if we opted to power the arm wirelessly. This would also lead to the neck being powered wirelessly since it would have to be mobile as well.

5.2 Selected Design Concept

5.2.1 Force Feedback Glove Design

When comparing the options between design concept 1 and design concept 2 for the FF-Glove we decided that the glove created by LucasVR would be the more ideal option for this project. This is because there is more documentation and rigorous testing done on this glove in comparison to the one that Kevin designed. Since the software implementation of both these gloves work and are very similar to each other the main thing that separated the options was the hardware implementation. When comparing the hardware implementations we decided that the servo motors used to apply force feedback in LucasVR's design can hold up longer than the solenoids in Kevin's design as solenoids can get hot when activated for extended periods of time. Another small factor that contributed to this decision was that solenoids can lose functionality if they are stored in a cold climate. This is potentially problematic as we will be working on the project mostly during autumn and winter months and transporting parts between group members houses and Benedum.

5.2.2 Camera Vision/ Head Tracking Design

The design for the camera vision module remained the same for both design concepts. More details on this implementation can be found in section 5.1.1.4.

5.2.3 Robotic Arm Software Design

After evaluating both design concepts, we have decided to go with design concept 1. By taking advantage of the Quest 2 controller's advanced tracking system, we can get a more precise position, especially considering the mechanical constraints. One of our main mechanical constraints was that the robotic arm will not be the same size as the user's arm. By using design concept 1, we are moving the hand to the exact location relative to the user instead of following the angle of the user's joints which would lead to a different relative position. Additionally, design concept 1 is far less invasive than design concept 2 which would require the user's whole arm to be covered. This means that the glove will be much more comfortable, light, and easy to use for the user. Considering our manpower and time constraints, having to design a whole exoskeleton like in design concept 2 for the arm would be another mechanical design which would be very time consuming. With design concept 1, however, we lean into a software solution which would lean into the ECE concentration of this project. When considering the communication constraints, design concept 2 requires a whole new layer of wiring and communication, whereas concept 1 has a much simpler flow from the already mounted quest controller, to the pc, to the robotic arm. Design concept 1 is a cleaner, less invasive, more ECE concentrated solution leading to it being our design of choice for the robotic arm software control.

5.2.4 Selected Hardware Design

After considering both design concepts, we have decided to move forward with all hardware and power designs laid out in conceptual design 1. In terms of power management, design concept 1 meets the goals we have set for ourselves for the project. For the glove, having it be mobile by being wireless is a big priority for our group. Therefore, we have selected to go for the battery powered approach to the design. The robotic arm and camera neck do not have mobility as a big priority, so making sure the motors are sufficienctly powered is the most important aspect of that design. To ensure proper power distribution, we have elected to go for the plug-in approach to support a higher range of current and ensure the motors are running at their optimum capacity. While the 2nd conceptual design does offer the capability of more flexibility and options in terms of powering everything, design 1 aligns with the vision we have for the system and will be the most optimal solution for our problem.

5.3 Sustainability Considerations

Sustainability is a very important factor in our force feedback telepresence robot design which we have kept in mind when designing our system. To minimize power consumption, we are using ESP32 microcontrollers, which are relatively low-powered devices which are more than capable of completing the tasks we need them to do while also enabling wireless communication. We will also be sure to write software which is efficient with optimized computations and data transmission to further reduce the energy consumption during operation.

We are also being wary of electronic waste when building this product. Before ordering parts, we are checking our individual inventory as well as around the ECE department for any parts that we need to avoid unnecessary waste. Additionally, most of our parts are individually replaceable so if anything in the system is not working, only one part has to be replaced so there will be minimal waste.

In order to stay economically sustainable, in addition to using parts already owned, we are using VR headsets and controllers we already own in our design to bring down the cost. We considered economic sustainability when deciding to use design concept 1 because for modules such as the robotic arm software, concept design 1 used a software solution using the controllers we already owned instead of having to purchase additional hardware to build an exoskeleton which would track the user's arm movements. Additionally, this design concept will be lighter, comfortable, and non-invasive making it accessible to more users.

As far as long-term usability, our modular design will be reliant on standardized communication protocols and use widely supported software platforms, such as Unity and OpenXR, to ensure that our system can be maintained and upgraded without high economic or environmental costs in the future.

6. System Test and Verification

6.1 Performance Criteria

6.1.1 Positional Accuracy of Robotic Hand

Positional accuracy will be based on how close the robotic hand gets to the target user-relative location as reported by Quest 2 controller tracking. The metric used to measure this accuracy will be the root mean square error (RMSE) between the target position and where the robotic hand actually is. We will target a RMSE of maximum error $\leq$ 5.0 cm.

6.1.2 Load Bearing of Robotic Arm

Load bearing will be based on the amount of weight the arm is able to grip and lift off of a surface. We will test this by trying to lift a small weight and slowly increase the weight of the load until the arm can't lift it up anymore. We will target a load-bearing weight of 5 pounds.

6.1.3 Torque Applied by Force-Feedback Motors on Glove

Another critical performance criteria for our project will be the force-feedback that the motors on the glove provide to simulate the sensation of grabbing onto an object.

6.2 Software Systems

6.2.1 Robotic Arm Unit Test

- Input end target position and test how close the robotic arm gets to moving there
- Compare robotic arm and target position using RMSE

6.2.2 Kinematics Correctness and Reachability

- This test aims at producing a formal analysis of the robotic arm and hand's inverse kinematics model.
- The commanded viewpoint/pose from the Quest tracking falls outside joint limits of the robotic arm/hand.

6.2.3 Network Degradation / Latency Robustness

- This test is designed to measure how predictable the degradation of the system is when there is various levels of lag in the system. To perform this test, we will forcibly introduce different latencies (50, 100, 200, 400ms) and do a full system test at each level.

- The system is expected to degrade gracefully as more latency is introduced. The code should freeze motion of the arm to prevent dangerous actuation if latency is too high.

6.2.4 Quest 2 Controller Input Verification

- Verify that STM32 receives controller coordinates and correctly extracts values

6.2.5 Glove → Robotic Arm End-to-End Positional Accuracy

- Test the full pipeline: Glove moves → PC receives and passes on tracking info → Robotic Arm mirrors glove movement
- Compare robotic arm and user's arm position using RMSE

6.2.6 Robotic Arm → Robotic Haptic Feedback

- Test the full pipeline: Robotic hand finger tips touch object → Robotic arm STM32 sends haptic feedback to glove → Glove restricts user
- Measure time from object detection until glove restriction

6.3 Hardware Systems

6.3.1 Power Management

- Verify regulator outputs with no load, then with representative loads.
- Use an inline meter to log current draw of arm and glove during idle, motion, and grip.
- Confirm isolation between logic (ESP32, sensors) and motor power rails.

6.3.2 Motor Control (Arm & Glove)

- Test each driver channel individually with a test PWM signal.
- Incrementally connect servos/DC motors: verify direction, range of motion, and current at rest vs. under load.
- Log stall current to validate fusing and regulator capacity.

6.3.3 Sensors & Tracking

- For glove: confirm finger bend sensors and IMU/position tracking give stable outputs.
- For robotic arm: verify the hall sensors are triggered when the fingers grip around an object.
- Validate calibration routines (neutral position, scaling factors).

6.3.4 Force Feedback

- Simulate object contact on arm → verify glove motors restrict finger motion proportionally.

- Test adjustable resistance levels and safety cutoffs (overcurrent, sudden obstruction).

7. Team

Our team is composed of three members; Kevin Quigley, Micah Smith, and Seth Blain. Both Kevin and Micah are computer engineers and Seth is an electrical engineer. Due to there being multiple modules of this project, where each module has both hardware and software aspects, we have delegated work from each so that Kevin and Micah focus most on the software side of things and Seth on the hardware side.

7.1 Kevin Quigley

As mentioned above, each member of the group will handle different parts of each module of the project. I have been designated to work on the software side of the FF-Glove as well as the mechanical design and software design of the camera vision module.

For the software of the FF-Glove I will be responsible for integrating the on-glove esp32 with the finger tracking (potentiometers) and force feedback (servo motors) hardware aspects as well as interfacing the glove with the open source OpenGloves app to get the gloves into virtual space.

The camera vision module has more complexity to it as it includes both the mechanical design and software construction. The mechanical design includes using CAD software to design 3D printable parts which assemble to create the robot's "neck" and "head". Specific details about the mechanical design and functionality can be found in section 1.1.2 Design Concept 1. The software aspect of the camera vision module includes building a program to get tracking information from the VR headset about its position and angle. Then using this tracking information to control the three servo's angles on the head assembly to mirror the VR user's head position. Furthermore, software will also need to be constructed to broadcast the live camera feed captured on the Raspberry Pi 4 to a server via interfacing with WebRTC.

7.1.1 Skills learned in ECE coursework

Throughout the design and construction process of my modules I will be applying knowledge and skills learned across multiple ECE courses. The basis of building the hardware of the FF-Glove, robotic arm, and camera vision modules will be built on knowledge obtained in both ECE 0101 (Linear Circuits and Systems) and ECE 0102 (Microelectronic Circuits). After the construction of the hardware I will be utilizing the knowledge learned in ECE 0301 (ECE Problem Solving With C++), ECE 0302 (Data Structures and Algorithms), ECE 1140 (Systems and Project Engineering), and ECE 1145 (Software Construction and Evolution) to build my software for my respective modules.

Writing code to run on an ESP32 is most easily done in the Arduino IDE using C++. This is where knowledge from ECE 0301 and ECE 1140 comes into play. For example, I learned how to write in C++ from ECE 0301 and in ECE 1140 I applied C++ in unison with Python to control an Arduino to complete my part of the group project. Moreover, I will be responsible for integrating multiple modules of this project which will use different coding languages so applying knowledge from ECE 0302 about how to structure data and optimize algorithms will help tremendously. Lastly ECE 1145 has taught me how to utilize test driven development to ensure any code I write is working as expected as I develop it which is almost necessary for a project like this with so many interconnected parts and code.

Beyond technical skills, working as a team efficiently and effectively is fundamental to completing this project. ECE 1145 was a semester long project that required me to work in a group of seven to work towards a common goal, building up our individual modules to work together in the end. This taught me how to communicate ideas with a large group as well as how to collaborate with others who are working on different cogs of the same machine which is an invaluable skill to have for this project.

7.1.2 Skills learned outside ECE coursework

Outside of ECE coursework I have skills in using CAD software, designing parts meant to be 3D printed, and working with Unity/VR. I have been working in Autodesk Inventor since Sophomore year of high school and have since used this skill in various ECE projects across multiple classes like ECE 1145 and Junior Design. This skill is valuable as I will have to design various hardware components mainly for the robot's neck mechanism as well as a replacement forearm for the InMoov robot to add one more degree of freedom.

I will have to acquire knowledge about how to use WebRTC which will entail setting up a web address to stream a constant live video feed which I have never done before. To do this I will first read through the documentation provided by WebRTC, then reference materials in articles and YouTube videos if I have further questions.

7.2 Seth Blain

As the sole electrical engineer on the project, the module I am tackling involves the hardware components of our system. I will interface all motors and sensors from all components with the microcontrollers that they use to communicate. I will also manage power for each component of our system and make sure electronics are powered properly and safely at full operation. I will guarantee that all power supplies will provide enough current for the many servos motors we are incorporating into the design. I will make large power buses to safely distribute energy and keep temperature in check as well.

Since the interfacing and power management of hardware components is fairly simple and involves choosing and implementing commercial devices, I have expanded my module to allow space for non-trivial electrical engineering design. I will design a custom motor control circuit for the 11 servos that will control the robotic arm and hand. I will design custom circuitry that will allow for the motor to move in both directions, receive sufficient current to carry loads, and interface easily with the software side of the project. I will design a custom PCB that takes power from a wall plug AC/DC adapter and distributes it to the motors and communicates effectively with the ESP32 microcontroller. I will also design a small PCB for the glove to deliver power to all components with proper connections and integrate a battery holder to make replacing it easy.

7.2.1 Skills learned in ECE coursework

I have developed many skills throughout the ECE coursework that will be applicable and help me complete this module. The class and lab for ECE 101 and 102 taught my the basics of circuit design that will be the foundation of the motor control circuit. My project work in CyberPhyscial systems will be beneficial as I have experience in communicate over different platforms to interface with a physical system. Also, I learned project and PCB skills in Junior Design that I will apply to this project and get stronger at throughout.

7.2.2 Skills learned outside ECE coursework

I have hands-on experience gained through 3 co-ops, an internship, and personal projects that will help me complete this module. I have experience with remotely controlled robotics through a project I did in high school. This not only gave me the skills to interface motors with microcontrollers but also how to interface hardware wirelessly using Bluetooth. I understand both the hardware and software aspects of this type of interface. I have experience in rapid prototyping and user-testing, which will allow us to quickly iterate through the mechanical design we will be required to do for the glove. Lastly, I have worked on projects where power management was a main factor (LED systems which also draw high currents). This provides me with the necessary skills to ensure our system will be properly powered at every component.

In order to be entirely successful in producing a successful module I will need to push my abilities further than what I've learned in ECE classes. Although I have done some simple motor control labs in ECE 101 and 102, I will need to understand motor control circuitry on a more complex level. I will need to do research on how to design circuit components such as H-bridges, power busses, etc. I will begin research online looking at general guidelines for motor control design, then look at other similar hobby projects that implement similar types of motor control. I also know Mark has experience in this type of hardware, so will use him as a resource frequently to keep me on the right track.

7.3 Micah Smith

As a computer engineer, I have been designated to work on the software behind the control of the robotic arm as well as the communication between the robotic arm and the haptic feedback glove.

For the glove, I will be responsible for handling the communication between the glove and the PC. This involves sending the position and orientation data from the Quest 2 controllers so the system can determine how to move the robotic arm to best reflect the user's motion.

For the robotic arm, I will be handling the communication between the PC, which has the glove position information, and the ESP32 micronconttller in the robotic arm. I will write control software which will intepret the position information and translate it to motor commands, allowing smooth mirrored movements in the arm.

I am also responsible for handling the direct communication between the FF-Glove and robotic arm, using two ESP32 modules. The two systems will have to be constantly exchanging data between each other, with the glove sending its movements to the robotic arm and the robotic arm sending any haptic feedback back to the glove. This bidirectional communication is crucial to an immersive user experience.

7.3.1 Skills learned in ECE coursework

Throughout the design and implementation process, I will be applying numerous skills that I've acquired through my ECE coursework. For the hardware aspect of the FF-Glove and robotic arm, I will mainly be applying circuit knowledge which I learned in ECE 0101 (lienar Circuits and Systems) and ECE 0102 (Microelectronic Circuits) as well as the circuit design skills which I obtained through ECE 1895 (Junior Design Fundamentals).

For the software modules of the FF-Glove and robotic arm, which will be my main focuses, I will be programming the ESP32s using the Arduino Uno IDE and C++. I have taken numerous classes which revolved around C++ including ECE 0301 (ECE Problem Solving with C++) and ECE 0302 (Data Structures and Algorithms) which has given me a strong base in C++. Additionally, ECE 1140 (Systems and Project Engineering) and ECE 1145 (Software Construction and Evolution) have taught me valuable lessons in managing a large software project and best practices. I also have experience working with the Arduino IDE and programming microcontrollers through ECE 1895 as well as ECE 0202 (Embedded Processors and Interfacing)

One of the most important parts of senior design is having good communication and teamwork skills which I have acquired through numerous classes in the ECE department. ECE 1140 and ECE 1145 were both semester long projects which taught me how to work on a software development team and best practices of managing a code base. These team and communication skills will be essential to ensuring my team can learn from each other and meet our high expectations for our final demonstration.

7.3.2 Skills learned outside ECE coursework

In order to be successful, there are several skills which I will need to learn to be successful in completing my module. Although I have experience programming microcontrollers, this project requires low-latency wireless communication between multiple ESP32s as well as other devices. I will be researching how to do ensure reliable, fast communication between such modules. Additionally, I will have to learn about inverse kinematics in order to translate the glove's position and orientation to directions for the robotic arm's movements. This involves learning inverse kinematics algorithms and writing code which will best move the arm in a smooth, natural manner to mirror the user. Another skill which I will learn is how to use Unity to interact with both the controllers and the ESP32 on the robotic arm. I have no previous experience with Unity so I will be learning how to integrate it with external hardware which I plan on doing through extensive research and testing.

8. Schedule and Budget Plan

8.1 Project Schedule

Week of 9/15:

- **Seth:** Work on one design concept for the hardware module

- **Micah:** Work on module's design concept for robotic arm control and communication with glove.
- **Kevin:** Work on one design concept for the camera vision module

Week of 9/22:

- **Seth:** Complete conceptual design document. Do research on motor control circuitry and other similar hobby projects. Have initial block-diagram design of motor control circuit. Spec out power supplies for all components.
- **Micah:** Complete conceptual design document. Test communication between Unity and Meta Quest 2 Controllers.
- **Kevin:** Complete conceptual design document. Begin working on hardware design of camera vision module (Robot's neck).

Week of 9/29:

- **Seth:** Complete block diagram of motor control circuitry. Begin breadboarding to verify power and control work properly and start modeling the circuit in PCB software. Finalize power supply spec'ing and get order placed.
- **Micah:** Figure out communication full communication between Quest controllers -> Unity -> ESP32. Begin writing code for controlling robotic arm.
- **Kevin:** Set up firmware for communication between FF-Glove/ ESP-32/ Oculus Quest. Finish plan for implementing the code to control the robot's neck (3 degrees of freedom).
- **Everyone:** Build robotic hand/arm. Construct FF-Glove.

Week of 10/6:

- **Seth:** Finish breadboard model of circuit and PCB schematic design. Start laying out PCB. If power supplies arrive, test that they can run all motors as expected. ***Out of town October 9-16.***
- **Micah:** Testing robotic arm movement and continue development of arm control code.
- **Kevin:** Have code for robot's neck working without mirroring user's head movement. For example, be able to tell the neck a specific angle/ rotation to display and it will move there. Get tracking data from the Quest headset into Unity. Display video feed from the webcam into Unity.

Week of 10/13:

- **Seth:** ***Out of town October 9-16.*** If power supplies arrive, test that they can run all motors as expected. Finish PCB layout and have teammates or another qualified peer verify the design. Order PCBs
- **Micah:** Finish robotic arm movement control with set coordinates to move to, and begin integrating communication with glove and Quest Controller tracking.

- **Kevin:** Integrate the sub-modules to have the robot's neck mirror the user's head movement. This includes getting the tracking information for the Quest's position in Unity and then to the Raspberry Pi.

Week of 10/20:

- **Seth:** While waiting for the PCB to arrive, start integrating the other components of the project. When PCB arrives, assemble and test it ASAP. If there is an issue with the board, fix the design and have it re-reviewed, and order the new version. Design and layout PCB for the glove and order it. This will be a small PCB with not as much assembly, so this will still give time to test it and integrate later.
- **Micah:** Finish communicating glove position and movements to the robotic arm and having it move to position.
- **Kevin:** Display video feed from the webcam into VR via WebXR/WebRTC and lock the videos position to the user's head (video will always be right in front of the user's eyes)

Week of 10/27:

- **Seth:** Integrate PCBs into the physical design of the robotic arm and begin testing its real operation. Assemble and test PCB for glove.
- **Micah:** Begin integrating feedback loop of robotic arm movement and current position of glove for smooth movement.
- **Kevin:** Integrate the FF-Glove finger tracking and force feedback with the robotic hand.

Week of 11/3:

- **Seth:** Do quantitative performance tests on the PCBs to calculate power consumption and efficiency. Integrate PCB into the glove.
- **Micah:** Finish integrating robotic arm movement with glove and begin working on haptic feedback loop.
- **Kevin:** Begin testing on the camera vision module. (Head tracking and streaming video.

Week of 11/10:

- **Seth:** Begin testing on the system as a whole to give us time to fix things. Make spares of both PCBs for redundancy and start the poster for expo (or final presentation).
- **Micah:** Finish haptic feedback loop between glove and robotic arm hand .
- **Kevin:** Test whole system and correct any possible errors that could occur.

Week of 11/17:

- **Seth:** Final testing and final presentation.
- **Micah:** Final testing and create final presentation .
- **Kevin:** Final testing and final presentation.

Week of 11/24: THANKSGIVING BREAK

- **Seth:** Finish final presentation and practice it.
- **Micah:** Touch ups on final presentation .
- **Kevin:** Work on final presentation. Practice final presentation.

Week of 12/1: DESIGN PRESENTATION & EXPO

- **Seth:** Work on final document and present to class and at expo.
- **Micah:** Work on final document and present to class and at expo.
- **Kevin:** Working on final document and present to class at the expo.

Week of 12/8:

- **Seth:** Finish final document.
- **Micah:** Finish final document
- **Kevin:** Finish final document.

8.2 Project Budget

This project requires many different components as we have multiple hardware aspects for our design. Generally, we decided to spend less than $60 on the initial order so that we could save more than half of the allotted budget for post-prototyping purchases. This decision was made as we know there will be more expensive hardware components that we will need later, such as servo motors for example. Furthermore, spending less than half of the budget on the initial order gives us more money towards a "rainy day" fund in case there are unforeseen components that we need to purchase.

After deciding to limit ourselves to $80 for the initial order we decided that the most needed components to purchase would include a webcam, various sized bolts, springs, and hall effect sensors.

The webcam we decided to acquire in our initial order was the EMEET 1080p Webcam as it boasted a 90 degree FOV and a 1080p resolution at 30fps or 60fps all for $30. This was a necessary item to begin prototyping as it is the main component of the camera vision module.

Other items in the initial order were all needed to assemble the robotic arm and hand from the open source project, InMoov, that we decided to utilize in our selected design concept (Design Concept 1). Since building the robotic arm and hand are high on our priority list so we can begin writing code and testing it these items were necessary to purchase at this stage of the design process. The various sized bolts included in the order consisted of sizes M3x4, M3x12, and M3x16. Next the InMoov instructions called for springs which are used in the robotic hand, these were purchased off of amazon and included enough to build two hands if time permits. Lastly, hall effect sensors were purchased as that is the main component that is used to detect "touch" in the robotic hand's finger tips. All together these parts cost us $29.33.

In summary, the initial order cost us $30 for the webcam and $29.33 for the parts needed to assemble the robotic arm and hand which brings us to a total of $59.33. The initial order came out to be just shy of the $60 we initial planned on utilizing for this purchase which leaves us in a good spot as we have $140.67 to use for the remainder of the semester.

Another important thing to mention in regard to budget are the items that members of the group already own that will be used towards the completion of this project. Firstly, the most expensive item on the list is an Oculus Quest 2 which retails for $299. This is a necessary component of the project as it is what allows the user to see through the eyes of the robot in virtual space. Another item that we will use is a Raspberry Pi 4 Model B which retails for $45. This is the device which will be used in the robot's head to interface with the webcam and three servo motors controlling its motion.

8.3 Minimum Standard for Project Completion

The minimum standard for project completion will include a robotic hand with 5 servo motors to individually control the five fingers. The robotic hand will be attached to a robotic arm (wrist to shoulder) with 6 degrees of freedom. We will have a webcam attached to a robotic neck which is actuated by 3 servo motors to achieve 3 degrees of freedom. This camera will display 1080p image at 30fps (60 preferred) to the VR headset worn by the user.

Next, a VR headset (Oculus Quest 2) must be implemented into the system. At a minimum this implementation will utilize the OpenGloves software to port our FF-Glove into VR which tracks the glove in 3D space as well as tracks the finger movements. Furthermore, the VR implementation will be able to allow the user to open the web address from the Oculus browser to view the live camera feed.

The FF-Glove must utilize force feedback and respond by pulling on the user’s fingers when the sensor is triggered on the robotic arm. Even if the user can still move their fingers, there should be a noticeable sensation of pulling back on them. The glove must have enough battery power to supply all 5 motors and the ESP32. Worst case scenario is that a wired solution is required for the glove’s operation. This is not ideal as it would limit the user’s mobility but would serve as a minimum accomplishment.

8.4 Final Demonstration

For the final demonstration we plan on displaying the features that align with the minimum standards for completion as well as a few tests that show the limits of our system. The demonstration will be a live test where one member of the group will put on the VR headset and FF-Glove to teleoperate the robot.

The team member conducting the demonstration will be sitting in a chair with a wall in between them and the robot. This is to show how the user would tele-operate the robot without directly seeing the robot in front of them while also allowing instructors to see the movements of both the robot and user simultaneously. Moreover, a projector will be utilized to display the user’s view from within the VR headset.

To demonstrate the accuracy of the user’s mirrored movements in the robot a series of different sub-demos will be conducted. First, the user will wave and move each finger individually. Then they will move their head around in all directions (up, down, left, right, tilting left/right) to display the mirrored movements of the robot’s head/neck (camera vision module).

To demonstrate the system as a whole, the user will have to tele-operate the robot and accomplish various tasks. These tasks will include picking up simple objects, stacking objects, and pouring a drink. The simple objects to pick up will be a cube, ball, and screwdriver. Next, the user will stack three cubes on top of each other. Lastly, the user will pick up a cup of water and pour it into another cup.

9. References

1. National Library of Medicine. (2014). *Occupational exposure to hazardous chemicals and associated mortality*. National Institutes of Health.
2. World Health Organization. (2021, September 17). *Almost 2 million people die from work-related causes each year*. https://www.who.int/news/item/17-09-2021-who-ilo-almost-2-million-people-die-from-work-related-causes-each-year
3. World Health Organization, Regional Office for Europe. (2023). *Chemical safety: Fact sheet*. https://www.who.int/europe/news-room/fact-sheets/item/chemical--safety
4. World Health Organization, Regional Office for Europe. (2023). *Chemical safety: Data and statistics*. https://who-sandbox.squiz.cloud/en/health-topics/environment-and-health/chemical-safety/data-and-statistics
5. Pan American Health Organization/World Health Organization. (2014, April 28). *PAHO/WHO warns of health risks from occupational exposure to chemical carcinogens*. https://www.paho.org/en/news/28-4-2014-pahowho-warns-health-risks-occupational-exposure-chemical-carcinogens
6. Centers for Disease Control and Prevention, National Institute for Occupational Safety and Health. (2021, December 13). *About falls*. https://www.cdc.gov/niosh/falls/about/index.html
7. U.S. Bureau of Labor Statistics. (2024, May 16). *A look at falls, slips, and trips in the construction industry*. TED: The Economics Daily. https://www.bls.gov/opub/ted/2024/a-look-at-falls-slips-and-trips-in-the-construction-industry.htm
8. U.S. Bureau of Labor Statistics. (2021, November 18). *Fatal and nonfatal falls, slips, and trips in the construction industry*. TED: The Economics Daily. https://blsmon1.bls.gov/opub/ted/2021/fatal-and-nonfatal-falls-slips-and-trips-in-the-construction-industry.htm
9. Business & Legal Resources. (2015, October 26). *The costs of slip, trip, and fall hazards*. https://safety.blr.com/workplace-safety-reference-materials/white-papers/employee-safety/slips-and-falls/The-costs-of-slip-trip-and-fall-hazards/
10. Wikipedia contributors. (2023). *MarkV-A1*. In *Wikipedia*. https://en.wikipedia.org/wiki/MarkV-A1
11. Wikipedia contributors. (2023). *DRDO Daksh*. In *Wikipedia*. https://en.wikipedia.org/wiki/DRDO_Daksh
12. Müller, J., Schmidt, A., & Rossi, D. (2023). *Impact of haptic feedback in high latency teleoperation*. *Proceedings of the ACM Symposium on Human-Robot Interaction*. ACM Digital Library.
13. Maddahi, A., Rastegarpanah, A., & Patel, R. V. (2025). *GAMORA: Gesture Articulated Meta Operative Robotic Arm*. *arXiv preprint arXiv:2501.01234*. https://arxiv.org/abs/2501.01234