HAPTIC FEEDBACK GLOVE

Jonathan Arze
Luciano Brignone
Taylor Gage
Nicole Khoury

CONCEPT OF OPERATIONS

REVISION – Final
04/28/20

CONCEPT OF OPERATIONS
FOR
Haptic Feedback Glove

TEAM ⟨40⟩

APPROVED BY:

Jonathan Arze Date

John Lusher II, P.E. Date

Andrew Miller Date

Revision History

Revision	Date	Author(s)	Description
Draft	09/14/19	JA	Created
A	09/16/19	JA	Cleaned up text for submission
B	09/24/19	JA	Updated scope for 10 joints
C	09/30/19	JA	Changed Modes of Operations
D	12/04/19	JA	Updated wording, fixed scope, added some information about GUI
E	04/25/20	JA	Added ethical considerations section
F	04/27/20	JA	Updated wording to reflect correct scope and deliverables
G	04/28/20	JA	Added ethical consideration
Final	04/28/20	JA	Final commit

Contents

List of Tables

List of Figures

1 Executive Summary

Haptic feedback is one of the largest barriers to bridging the gap between virtual and physical connection. Though more solutions to real-world problems are being developed in the virtual space, our ability to understand information and data is limited by the depth to which we can interact with it. The goal of this project is to enable users to interact with virtual environments in a way that is natural, intuitive, and immersive. By using natural hand movements, users can interface with a virtual environment via precise, omnidirectional finger and wrist tracking. Likewise, the computer communicates with the user by applying resisting forces on the fingers to simulate natural stimulation. By using a two-way communication line between the virtual and physical spaces, a plethora of opportunities open for new types of computational interaction.

2 Introduction

2.1 Background

Telepresence and telemanipulation have been in the minds of researchers since the early 1950's. Original designs essentially utilized a master-slave design in which the master robot simply mirrored the slave robot used by an operator [1]. However, even in its early stages, it was clear that imitation was not enough; the operator needed some sort of kinesthetic feedback to be able to effectively perform tasks remotely.

The idea of kinesthetic feedback, though costly to produce, slowly began to grow. In the 1960's, the U.S. military used haptic feedback in their pilot training simulations to provide vibration and movement to simulate the feeling in the cockpit of an actual plane. However, development of this technology largely stalled until the 1990's and focused primarily on small vibrations in response to user input more as a communication method than as sensory feedback. However, in 1993, an MIT student constructed a "Three Degree of Freedom Force-Reflecting Haptic Interface" [2], a device that delivered much more precise feedback than had been previously demonstrated. Over the past few decades, many breakthroughs have been made in device technology, fabrication techniques, and computational capabilities that make haptic feedback more feasible than ever before.

Current solutions to this problem are either large and unwieldy, or provide little force at too few joints to be effective. For example, the current industry leader in haptic glove technology uses hydraulic pumps to provide both tactile and force feedback. While this solution is undoubtedly innovative, it is unlikely it will ever be able to maintain full usage of the hand. Other attempts at solving this problem have used pulleys or other exoskeletal devices that provide little stopping force, and also greatly increase the operational size of the hand. Not only does the need for extra-bodily awareness break immersion, it also requires one of two things from the user:

- The user must learn new strategies and techniques for interacting with the environment, or
- The user must consciously compensate for atypical obstacles that are not rendered in the virtual environment.

These two considerations yield the problem to be solved by this project: to design a haptic feedback device that both minimizes physical size and maximizes sensory feedback. The kinesthetic reaction subsystem in our design is inspired by research performed at the Advanced Interactive Technologies Lab at ETH Zurich [3]. Though only a proof of concept, their design boasts a light, form-fitting solution to the force feedback problem using electrostatic brake technology.

2.2 Overview

This project aims to use the latest research in the field of haptics to provide a previously unexplored level of flexibility, precision, and feedback for interaction in a virtual environment. There are three foundational goal areas for this project:

1. Maintain dexterity of the hand

2. Track all degrees of freedom in the hand
3. Generate resistive force feedback to each finger

Each goal can be considered equally valuable, such that no goal should be compromised for another. As such, the vision for this project is to achieve the optimal quality of all three areas as shown in Figure 1 below.

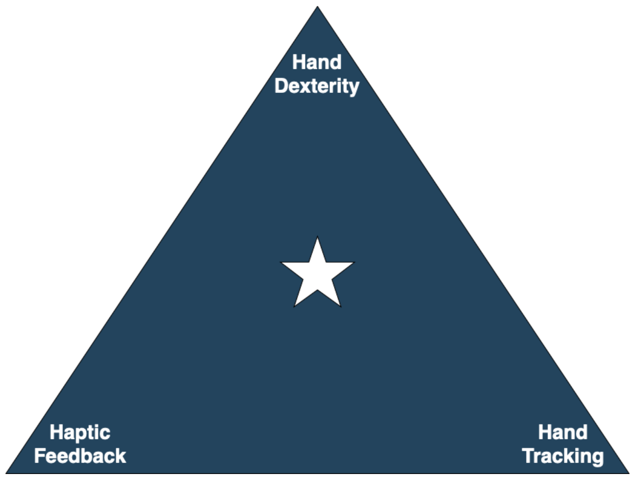


Figure 1: Optimal Balance of Goal Areas

The project will have several features in order to achieve these goals. The glove will be able to communicate the position of the fingers to Unity's 3D software for both ease of use and widespread implementation. Unity will translate the position into the virtual space and will relay any virtual interaction events to the power controller. The power controller will be able to independently enable and vary the voltage applied to each of the fourteen feedback modules. A high-level view of data flow can be found in Figure 2.

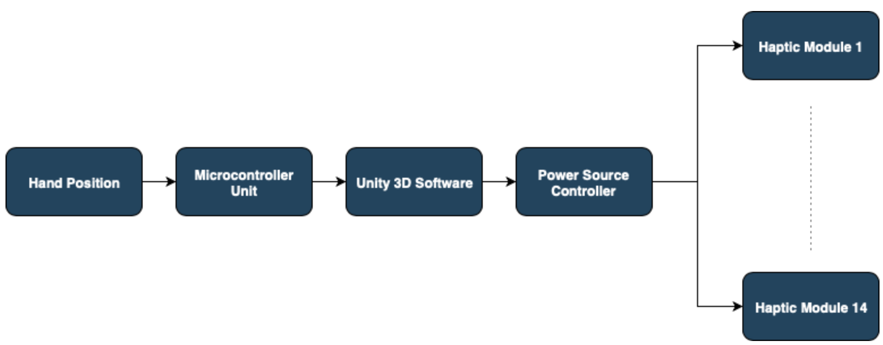


Figure 2: High-Level Flow of Information

2.3 Referenced Documents and Standards

[1] B. Hannaford. "Kinesthetic Feedback Techniques in Teleoperated Systems". In: *Control and Dynamic Systems – Advances in Theory and Applications*. Ed. by C.T. Leondes. Academic Press, 1991. Chap. 1.

[2] T. Massie. "Design of a Three Degree of Freedom Force-Reflecting Haptic Interface". Massachusetts Institute of Technology, May 1993.

[3] R. Hinchet et al. "DextrES: Wearable Haptic Feedback for Grasping in VR via a Thin Form-Factor Electrostatic Brake". In: *UIST 2018*. 2018.

3 Operating Concept

3.1 Scope

The deliverables of this project shall be as follows,

- A glove that accurately and precisely tracks finger and wrist movement.
- To design a modular feedback system that is capable of applying a moderate amount of force to 10 individual joints of the fingers.
- A microcontroller unit to process glove sensor data and transmit to a host computer.
- A microcontroller unit to process virtual interaction data received from a host computer and independently control 10 force feedback outputs.
- A power supply capable of independently and dynamically supplying sufficient voltage to all 10 feedback modules based on a control input.
- A 3D environment that visualizes hand tracking data, calculates feedback forces, and transmits information about interactions in the virtual environment to the microcontroller unit.

3.2 Operational Description and Constraints

The intention of this project is to be used alongside current virtual reality programs and hardware to enable the user to interact with virtual environments in a way that is natural, intuitive, and immersive. Apart from the headset provided by their current system, users will use our haptic glove, containing sensors and an electrostatic brake design, to allow the user to navigate and interact with the virtual environment. As with any virtual system, the user will have to calibrate the glove so it will be able to perform optimally. After the initial set up, the glove will perform the same functions as any other virtual input device; however, our glove will have to communicate with the user interface to translate the position into the virtual environment and send the information of the virtual interactions to a microcontroller. The microcontroller will have to process this information and provide an adequate amount of voltage from the power supply to each individual haptic module.

The constraints of this system include:

- Tracking Accuracy:
  As expected with any tracking device, the glove might have slight precision inaccuracy due to our method of tracking the hand.
- Cables:
  Since we are creating the glove to allow almost full immersion to a virtual environment, our electrostatic brakes will need to have a series of cables connected to each module to supply the correct amount of voltage to the glove.
- Spreading of fingers:
  Our glove will not be able to provide resistance to the spreading of fingers.

3.3 System Description

The Haptic glove is compromised of the following four main subsystems:

1. Microcontroller Unit and Hand Tracking:
   The microcontroller unit (MCU) will receive and send signals to the corresponding subsystems. The movement of the hand and fingers will be processed by a set of sensors and resistors at different positions on the hand and then send this data to the MCU. The MCU transmits the information of the hand/finger positioning to the user interface. Data sent from the user interface is sent to the MCU that will then send information to the power distribution system.

2. Power Distribution:
   This subsystem will receive information from the host computer and convert the 120 volts available from the electrical outlet to the correct amount of voltage needed to each haptic module in the glove. Each module will need between 0 to 1300 volts to operate. Different levels of applied voltage equate to different force feedback levels. Alternating the feedback levels allows the illusion of different hardness in the virtual environment.

3. User Interface:
   The user interface will consist of a virtual environment to interact with as well as the I/O communication with the MCU. The virtual environment will primarily consist of a kinematically correct model of the hand that precision finger tracking from the MCU shall be mapped to. This ensures body ownership and gripping confidence, even if the user's hand is not directly visible. The user interface will also be in charge of calculating and outputting any interaction variables, including which feedback modules to affect and how much force to apply, to the power controller.

4. Glove Design:
   The electrostatic brakes of the glove will be fabricated from elastic metal plates that will be positioned along each joint of the finger. These plates are separated by an insulator. When the hand in the virtual environment comes into contact with a virtual object, the correct voltage from the power distribution system will be applied between the plates. This will clamp the plates to prevent the movement of the joints. Each module will need to be strategically placed so that the electrostatic attraction of the other modules don't interact with each other.

3.4 Modes of Operations

The Haptic Feedback Glove shall be able to track intersections with hard and soft object. Force feedback will be adjusted accordingly with each virtual object.

- Tracking without Interactions:
  The Haptic Feedback Glove will be able to function as a simple controller without any force feedback. This can be thought of as simply reflecting physical movements in the virtual environment.

- Tracking with Hard Interaction:
  With a hard interaction turned on, the Haptic Feedback Glove will provide maximum force to any given joint based on interactions in the virtual environment. This will occur in conjunction with hand tracking, and will only be applied to the joints indicated by the MCU as currently interacting with a hard object.

- Tracking with Soft Interaction:
  With a soft interaction turned on, the Haptic Feedback Glove will provide suggestive force to any given joint based on interactions in the virtual environment. This is intended to simulate a soft object that does not fully stop the finger, but provides some resistive force. This will occur in conjection with hand tracking, and will only be applied to the joints indicated by the MCU as currently interacting with a hard object.

- Standby:
  In standby, the system will not be receiving or transmitting information; it will be waiting for the user to change the mode to operational.

3.5 Users

Traditionally, virtual devices use joysticks as a mean for the user to interact with the virtual environment, but with the Haptic Feedback Glove, users are able to be fully immersed in their environment. This glove is a product targeting companies and people who have virtual devices and wish to blur the lines between reality and simulation. Providing force feedback to each segment of the finger allows the user to grip and feel different shapes present in virtual space; ultimately creating a natural feeling for the user.

3.6 Support

Each haptic glove will include a user manual in which will consist of detailed descriptions on how to set up the glove and how different subsystems communicate between each other. Furthermore, the user manual will provide the needed documentation to help troubleshoot any problems that may arise from setting up the system as well as contact information.

4 Scenarios

4.1 Medical

The applications of virtual environments in the medical field are being explored, with uses in medical training, mental health studies, and surgery. There are potential benefits to both the doctor and the patient. Live training and simulations are specific aspects that would be greatly enhanced by the introduction of haptic feedback.

To be able to have the sensation of feeling tools and various surroundings makes it easier to accurately train medical personnel for different situations that may not otherwise be conducive to training. For example, a medical resident could train for any surgery, from hip replacements to heart transplants, without any added risk. The more realistic the experience feels the more benefit it can have.

Additionally, there are applications in mental health. Virtual environments are being used to treat patients with Post Traumatic Stress Disorder, anxiety, phobias, and depression. The idea is that the patient can come into contact with different scenarios in a safe and controlled environment. Haptic feedback is an important aspect of this immersion experience, and would greatly increase the effectiveness of these treatments.

4.2 Military

Virtual environments allow the military to include a wide range of training simulations. Without haptic feedback systems, the current setups lack important aspects that would be able to make these simulations much more realistic. The introduction of this feedback would increase the user's spatial awareness of objects and create a more effective training space.

One main downfall of current virtual environment technology is that the hand tracking is done via the camera in the headset. This means that for the tracking to work, the user must be looking at their hands. This limits options in simulations and does not allow the user to work with their hands while they are not in view. A haptic feedback system would allow full hand and finger tracking at all times, as well as guide the user through actions while they are looking somewhere else.

4.3 Sports

Virtual training environments are being used in sports to analyze player performance. The potential for virtual practice sessions increases when the player has the ability to actually feel the ball they are holding. Trying to throw a perfect spiral is difficult when the user cannot feel the ball in their hands. With a haptic feedback glove, practicing in a virtual environment becomes much more beneficial.

Football, soccer, and basketball are some of the sports with teams already utilizing virtual environments, either for fan interaction or practice purposes. With the added realism of a haptic feedback glove, the potential for all of these markets increases exponentially.

5 Analysis

5.1 Summary of Proposed Improvements

The best feedback glove being developed currently is a design by Haptx. Their design uses pneumatic methods to provide force and haptic feedback. One advantage to this is high-resolution haptic feedback for the fingers and palm of the hand. Though this provides the user with unique haptic sensations, it comes at the price of limiting the free movement of the hand. Instead, they opt to reduce the fingers to a basic, single gripping motion for easier force feedback control.

Our electrostatic brake design seeks to remedy this reduction in hand movement by applying force feedback to two separate locations on each finger, one at the fingertip to restrict distal and middle finger movement, another on the longest segement of the finger to restrict proximal movement. The thumb will also have two feedback points, one for the middle segment and one on the tip of the thumb, for a total of ten feedback points for the entire hand. This will allow the user to grip and feel the shape of a variety of objects, such as spheres and boxes, as opposed to just single motion gripping. Another added improvement this brings is the return of lateral finger movement and maintained dexterity of the hand. Since we shall have the ability to leverage two movement points per finger we will design the system to allow for much more natural, free motion. And lastly, our system will apply different voltages across the electrostatic brakes depending on the firmness of the object held. This, in turn, will allow the user a greater variety of force feedback, letting them squeeze softer objects but not harder ones. All of these aforementioned improvements will result in a greatly increased sense of immersion, as we seek to blur the line between simulation and reality further and further.

5.2 Disadvantages and Limitations

The biggest disadvantage to our design is the high levels of voltage needed to produce the range of force feedback we want. For softer forces around 4 newtons or 0.9 pounds, we will apply a voltage of about 500 volts. For stronger forces of up to 15.5 newtons or 3.5 pounds, a voltage of near 1200 volts will be necessary. These voltages are quite high; however, our system will operate at very low currents, meaning that there is no risk to the user, and the overall power used by our design will be minimal.

A limitation that we face is how far the glove can travel. Because of the high voltages mentioned above, our system will require a steady, large voltage source to maintain power. Since computer power outputs are usually limited to around 5 volts we will be using a standard wall outlet and boosting the voltage up to the levels we need. This results in needing to be tethered to a location, limiting the ability of the user to roam freely.

5.3 Alternatives

There are some alternatives to our electrostatic braking mechanism, though we found that the approach we decided on offers the most advantages. Some of the other options we had considered included wires controlled by motors, mechanical braking mechanisms, and electromagnetic clamping designs. Starting with the motorized wiring idea, the issues we faced were finding motors small enough to fit on the glove discreetly while also providing enough torque for our desired level of feedback. Also, this approach would have only accounted for one restriction point on the tip of each finger, which would again reduce the degrees of freedom available for the user's hand. Next, we considered a mechanical braking solution which would utilize small motors to clamp down on the wire as opposed to retracting it, reducing the size of motor needed to produce a similar stopping power. Again though, the motors this idea would require were still too large and heavy, though it would allow the fingers to move more than the previous revision. Finally, the last alternative we considered was electromagnetic clamping, which would function the same as the mechanical braking solution but using small electromagnets instead of motors to reduce the size and weight. The trade-off with this idea was the interactions the magnetic fields of each finger would have on each other, along with the effects on surrounding components. When compared to the alternatives listed here, our electrostatic braking system combines the best of each while minimizing the drawbacks.

5.4 Ethical Considerations

There is a donor that is concerned with the real-world applications of such an immersive virtual reality experience. The donor wants to implement an overall time constraint for regular participants, limiting interaction to 7 hours per month. The fear is that possible desensitization from certain experiences, most notably violent video games, is heightened due to the realistic nature of this simulation.

Issue and solution: Limiting the leisure activities of a consenting adult turns into a very complicated problem. It escalates into situations beyond the control of the company or its research participants. If exposure is limited to 7 hours per month, would special considerations be made for individuals with a history of mental health issues? The applications of this system are invaluable and there is not a blanket limit that can be imposed on all users. If the system develops to a point where it is inside the user's home, there would be no way to enforce such a restriction without violating privacy rights. After ensuring that users are 18 or over and are made aware of risks (psychological and electrical), our only responsibility is to the function of the system.

HAPTIC FEEDBACK GLOVE

Jonathan Arze
Luciano Brignone
Taylor Gage
Nicole Khoury

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – Final
04/28/20

FUNCTIONAL SYSTEM REQUIREMENTS
FOR
Haptic Feedback Glove

TEAM ⟨40⟩

APPROVED BY:

Jonathan Arze	Date
John Lusher II, P.E.	Date
Andrew Miller	Date

Revision History

Revision	Date	Author(s)	Description
Draft	09/24/19	JA	Created
A	09/29/19	JA	Initial Release
B	12/04/19	JA	Updated wording and scope definition
Final	04/28/20	Updated wording and references. Final commit	

Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose and Scope

The purpose of the Haptic Feedback Glove is to improve on the limitations of previous devices designed. The Haptic Feedback Glove will allow the user to interact with a virtual environment, achieved by integrating with the biological nervous system. The goal of this system is to provide a new and intuitive way to interact with virtual data that maintains the dexterity of the hand while also providing kinesthetic feedback to the user about their interactions in the virtual environment. The graphical user interface (GUI) for this system shall include one or more virtual environments created in Unity 3D software in which to calibrate the glove and interact with virtual objects. Figure 1 below shows the high-level flowchart of information in this project.

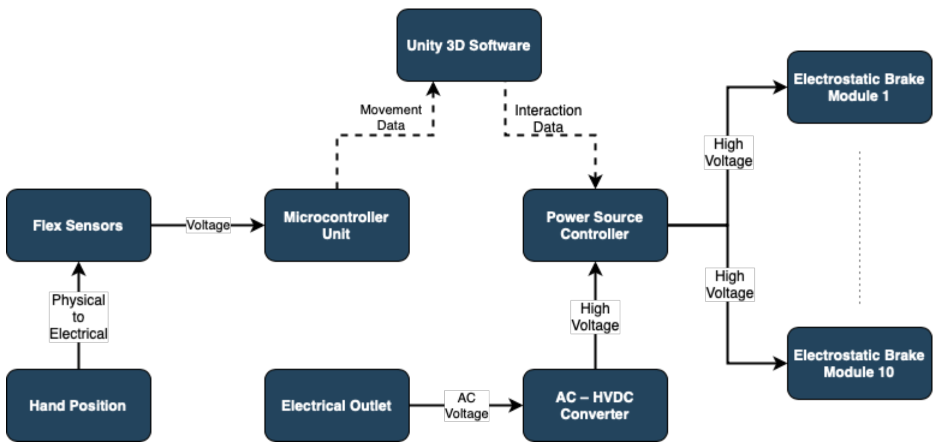


Figure 1: High-Level Flowchart of the Haptic Feedback Glove

The Haptic Feedback Glove shall be able to record and transmit kinematic data for each finger and the wrist in order to recreate the position of the hand in a virtual environment. This data shall be sent by the microcontroller unit (MCU) to a host computer running Unity 3D, and the host computer shall render the positioning of the user's hand. The host computer shall then communicate data about interactions between the rendered hand and other objects in the virtual environment to the MCU of the power source controller. This controller shall enable independent electrostatic brake (feedback) modules, and may determine the voltage applied to the module.

The following definitions differentiate between requirements and other statements.

Shall:	This is the only verb used for the binding requirements.
Should/May:	These verbs are used for stating non-mandatory goals.
Will:	This verb is used for stating facts or declaration of purpose.

1.2 Responsibility and Change Authority

Each team member is in charge of their subsystem; however, there will be cooperation between team members on all subsystems. Each member will have their own responsibilities and deadlines he/she will be required to meet. The project manager (Jonathan Arze) will make sure that each of the member's deadlines and the team's deadlines are met. Since our project is a faculty-sponsored project, any changes must be approved by the majority of the team and the sponsor (Andrew Miller).

Team Member	Subsystem	Responsibility
Jonathan Arze	Graphical User Interface	Project Manager
Luciano Brignone	MCU and Hand Tracking	Treasurer
Taylor Gage	Power Supply	Secretary
Nicole Khoury	Electrostatic Brake Modules	Editor

Table 2: Table of Team Member Responsibilities

2 Applicable Reference Documents

2.1 Applicable Documents

[1] "IEEE Standard for High-Voltage Testing Techniques". In: *IEEE Std 4-2013 (Revision of IEEE Std 4-1995)* (May 2013), pp. 1–213. DOI: 10.1109/IEEESTD.2013.6515981.

[2] "IEEE Recommended Practices for Safety in High-Voltage and High-Power Testing". In: *ANSI/IEEE Std 510-1983* (1983), pp. 1–19. DOI: 10.1109/IEEESTD.1983.81973.

[3] U. Nanda and S. K. Pattnaik. "Universal Asynchronous Receiver and Transmitter (UART)". In: *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*. Vol. 01. 2016, pp. 1–5.

[4] "IEEE Draft Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats". In: *IEEE P1451.2/D20, February 2011* (2011), pp. 1–28.

2.2 Reference Documents

[5] Laura Sbernini Giovanni Saggio Francesco Riillo and Lucia Rita Quitadamo. "Resistive flex sensors: a survey". In: (2015).

[6] *MSP432P401R, MSP432P401M SimpleLink™ Mixed-Signal Microcontrollers*. Texas Instruments. 2019.

[7] *MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.0*. InvenSense. 2012.

[8] N. Waghamare and R. Argelwar. "High Voltage Generation by using Cockcroft-Walton Multiplier". In: *International Journal of Science, Engineering and Technology Research* Volume 4 (Issue 2 Feb. 2015).

[9] *Map of College Station - Texas*. https://www.usclimatedata.com/map.php?location=USTX2165, last accessed on 09/26/19.

[10] *System Requirements*. https://unity3d.com/unity/system-requirements, last accessed on 10/01/19.

[11] *System Requirements*. https://developer.vive.com/resources/knowledgebase/vive-specs/, last accessed on 04/28/20.

2.3 Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings, or other documents that are invoked as "applicable" in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except Interface Control Documents (ICDs) that have their relevant documents considered to be incorporated as cited.

3 Requirements

3.1 System Definition

The Haptic Feedback Glove shall be worn on the user's right hand. It shall be used to supplement his/her experience in a virtual environment. When holding or interacting with virtual objects, the user shall experience kinesthetic feedback that shall change based on varying materials.

The amount of voltage supplied to the braking modules will vary the friction force of the feedback. This will allow the user to differentiate between hard and soft objects, and may simulate objects with various malleability. Flex sensors and an inertial measurement unit (IMU) on the glove shall allow for full finger and wrist tracking to allow the user to intuitively interact with the UI. This shall allow hand tracking without the use of a camera, enabling the user to handle objects in the virtual environment without looking directly at them.

Figure 2 shows the functional block diagram of the full system. This shows the interaction between the four subsystems, and how integration will ultimately be taking place. For hand tracking, position data shall be collected from user movement via the flex sensors and IMU, and sent to the host device to be rendered in Unity 3D. The type of virtual object interaction shall determine what voltage is sent from the power supply to the braking modules. These high voltages shall be sent to the braking modules, which each may be equipped with over-current protection measures. The user shall feel the force feedback as an output.

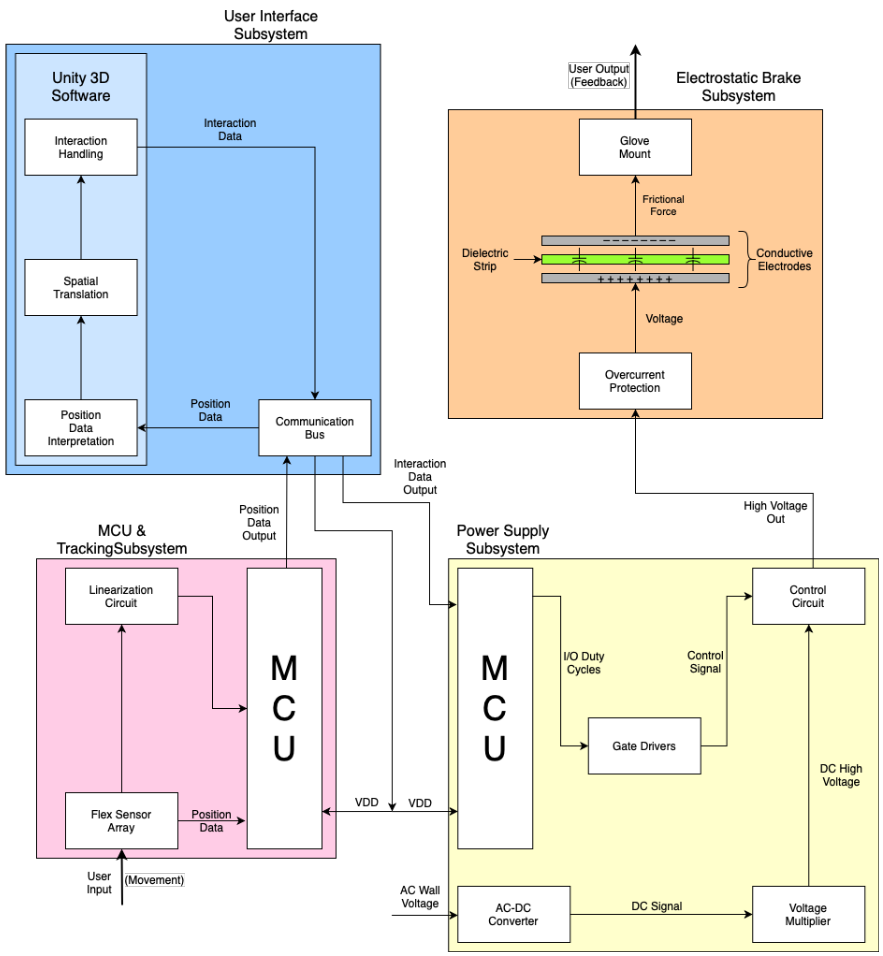


Figure 2: Full System Functional Block Diagram

3.2 Characteristics

3.2.1 Functional/Performance Requirements

3.2.1.1 Electrostatic Brake The electrostatic brake system shall provide at least 9 newtons (N) of force to the larger, lower finger joint, and at least 4 N of force to the smaller, upper finger joints. The braking system will have 10 points of force feedback; 2 for each finger of the right hand.

3.2.1.2 Finger Tracking The flex sensors used for finger tracking shall detect finger movement of each joint with an accuracy of $\pm 5°$. For the wrist and hand movement, the IMU shall measure acceleration, velocity, and orientation with an accuracy of $\pm 2.5°$.

3.2.1.3 Power Supply The power supply shall provide an adjustable high voltage of at least 4 volts (V) and not exceeding 1360 V by utilizing a high voltage rail generated by a Cockcroft-Walton voltage multiplier configuration [8]. These voltages shall be individually controlled and applied to each of the 10 points of force feedback. The power dissipation of the supply shall not exceed 3 watts (W), and the maximum current draw across any singular output to a control point shall not be greater than 80 microamps (μA).

3.2.2 Physical Characteristics

3.2.2.1 Weight Located on Hand The mass of the electrostatic braking system shall be no more than 68 grams (g). The weight of the hand tracking system shall be no more than 48 g. The weight of the cloth glove is approximately 15 g.

Rationale: 30 stainless steel strips will weigh approximately 42.2 g. Wax paper dielectric will weigh approximately 21 g. The 19 flex sensors will weigh approximately 24 g. The IMU will weigh approximately 4 g. An additional 55 g are reserved for the 3D printed guides, Velcro straps, and wires for the braking and hand tracking system.

3.2.2.2 Weight Located off Hand The weight of the hand tracking system shall be no more than 70 g. The wight of the power supply shall be no more than 1.5 kilogram (kg).

Rationale: The weight of both PCBs and MCUs for the hand tracking and power supply.

3.2.2.3 Size The Haptic Feedback Glove shall measure approximately 22 cm from the wrist to the top of the fingertips and 15 cm from index finger to fifth digit.

Rationale: These measurements were taken from the product specifications of the heat-resistant glove that was purchased. This glove is a standard one-size-fits-all knit glove, and it shall be outfitted with the flex sensors and braking systems.

3.2.3 Electrical Characteristics

3.2.3.1 Inputs

3.2.3.1.1 Supply Voltage Level The input voltage level for the Haptic Feedback Glove shall be +120 V from a wall outlet and +5 V from computer/device connecting to the MCUs. The minimum voltage required to operate the flex sensors and the IMU device is 2.7 V with a maximum of 3.3 V.

3.2.3.1.2 Communication Protocols The system will communicate between separate subsystems with already established serial communication protocols: I^2C, Serial Peripheral Interface (SPI), and Universal Asynchronous Receiver Transmitter (UART).

3.2.3.1.3 Power Consumption

(a) The power supply shall consume no more then 3 W in order to ensure very low heat and low current operation. The maximum current flowing through any given output voltage branch will not exceed 500 μA.

Rationale: The low voltage controls of the power supply shall be protected from the high voltage rail via a designated power MOSFET gate driver to ensure no rush current is seen by the microprocessor. Because of this, the only resistance seen by the power supply shall be that of the MOSFET transistors used to control the power output as well as the current limiting resistors.

(b) The maximum power of the braking system shall be equal to or less than 625 milliwatts (mW), assuming an initial overlap area of 11 cm^2 and an initial distance between electrodes of 13 μm.

Rationale:

$$I_{leakage} = \frac{Q}{k\rho}$$

$$k = 3.4$$

$$\rho = 1.5 \times 10^{17}\,\Omega\,\text{cm}$$

$$C_{strip} = \frac{\epsilon_r \epsilon_0 A}{d} = \frac{3.4 * 8.85 \times 10^{-12}\,\text{F/m} * 11\,\text{cm}^2}{13\,\mu\text{m}} = 25.461\,\mu\text{F}$$

$$Q = CV = 25.461\,\mu\text{F} * (1250V * 10) = 0.318\,\text{C}$$

$$I_{leakage} = \frac{0.318\,\text{C}}{3.4 * 1.5 \times 10^{17}\,\Omega\,\text{cm}} = 6.24 \times 10^{-19}\,\text{A}$$

$$P = 6.24 \times 10^{-19}\,\text{A} * 12\,500\,\text{V} = 7.8 \times 10^{-15}\,\text{W}$$

(c) The maximum power consumption of the hand tracking subsystem shall not be more than 25 mW.

Rationale: The IMU with 3.3 V input and all sensors enabled will consume 12 mW; while the flex sensors with an input of 3.3 V and at maximum flex position will consume 13 mW.

3.2.3.2 Outputs

3.2.3.2.1 Force Output The feedback output of the Haptic Feedback Glove shall be at minimum 4 N of force on the upper finger joints and 9 N of force on the lower finger joints. This shall cause the user to feel at least 0.9 pounds (lbs) of force on the upper finger joints and/or 2.02 lbs of force on the lower finger joints when interacting with virtual objects.

This force feedback shall vary in magnitude depending on the type of virtual object and its perceived stiffness or malleability. The force output of the system shall not exceed 15 N, or 3.4 lbs, of force.

3.2.4 Environmental Requirements

This device shall be designed only for indoor use. Though the device may operate under more extreme environmental conditions, laboratory testing shall occur in what is to be considered typical operating conditions (see Appendix B).

3.2.4.1 Pressure (Altitude) This device shall be designed and tested at an elevation of 328 feet (ft) [9], though any operational changes due to altitude pressure may be adjusted for with system calibration.

Rationale: There are no pressure sensitive components in this design, and the device is not intended for use in extreme environments.

3.2.4.2 Thermal The maximum range of operation for this device shall be 80°C, though performance may become unstable at such temperatures. The device shall be verified in typical operating conditions of 24 ± 1°C.

Rationale: All components are rated for at least the above value, but since this device is not intended for use in extreme conditions, testing will occur in typical conditions.

3.2.4.3 External Contamination The haptic feedback glove shall be resistant to normal indoor contaminants (dust, particles, etc.), though occasional cleaning may be necessary in order to maintain the dexterity of the glove.

3.2.4.4 Water The device should be kept away from water or any other liquids. This device is not rated for any water resistance. This device contains high voltage electronics that, if exposed to water, may pose a risk of serious injury to the user and to others. Do not expose this device to water under any circumstance.

3.2.5 Failure Propagation

The power supply shall have overcurrent protection to ensure all outgoing current returns through the wall outlet. This shall provide protection against short circuits throughout the system. Additionally, there shall be overcurrent protection on the braking system as an extra precautionary measure. There would need to be two concurrent failures of both protection systems for a dangerous situation to occur.

4 Support Requirements

4.1 System Requirements

The system requirements for the software to interface with the Haptic Feedback Glove are as follows: [10]

- OS: Windows 7 SP1+, macOS 10.12+, Ubuntu 16.04+
- Graphics card with DX10 (shader model 4.0) capabilities.
- CPU: SSE2 instruction set support.

4.2 Virtual Reality Requirements

The system may include support for a virtual reality scene utilizing the HTC VIVE and a VIVE Tracker. The system requirements for the Haptic Feedback Glove to operate in a virtual reality environment are as follows: [11]

- Graphics: NVIDIA® GeForce® GTX 1060 or AMD Radeon™ RX 480, equivalent or better.
- Processor: Intel® Core™ i5-4590 or AMD FX™ 8350, equivalent or better
- Memory: 4 GB RAM or more
- Video out: HDMI 1.4, DisplayPort 1.2 or newer
- USB ports: 1x USB 2.0 or better port
- Operating system: Windows 7 SP1, Windows 8.1 or later, Windows 10

4.3 Provided Support

This system shall include all the necessary hardware to operate the device, excluding the host computer, as well as the software containing calibration and sample environments (see system requirements above).

Appendix A Acronyms and Abbreviations

MCU	Microcontroller Unit
UI	User Interface
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
ICD	Interface Control Document
mA	Milliamp
μA	Microamp
mW	Milliwatt
V	Volts
N	Newton
ft	Feet
cm	Centimeter
lbs	Pounds
μm	Micrometer

Appendix B Definition of Terms

Typical Operating Conditions	Typical operating conditions shall be defined as an indoor, temperature and humidity controlled room in the range of $22°C$ - $26°C$, and with humidity levels between 30% - 50%. Infrared or other forms of natural radiation are not within the bounds of typical operating conditions. Operation outside of the defined typical conditions may produce unexpected results or damage the device.

HAPTIC FEEDBACK GLOVE

Jonathan Arze
Luciano Brignone
Taylor Gage
Nicole Khoury

INTERFACE CONTROL DOCUMENT

REVISION – Final
04/28/20

INTERFACE CONTROL DOCUMENT
FOR
Haptic Feedback Glove

TEAM ⟨40⟩

APPROVED BY:

Jonathan Arze Date

John Lusher II, P.E. Date

Andrew Miller Date

Revision History

Revision	Date	Author(s)	Description
Draft	09/26/19	JA	Created
A	09/29/19	JA	Initial Release
B	12/04/19	JA	Updated wording to reflect changes in operation
C	04/25/20	JA	Added Communication Protocols and updated wording in Sections 6 and 5.3
Final	04/28/20	JA	Updated wording and references. Final commit.

Contents

List of Tables

List of Figures

1 Overview

The Interface Control Document (ICD) for the Haptic Feedback Glove will provide more detail on how the subsystems in the Concept of Operations and the Functional System Requirements will be produced. The ICD will include physical descriptions of the various elements and the electrical interface, power, and placement of sensors and electrostatic brake modules. This document will show how each system will be implemented together to translate hand movement into a virtual environment and express virtual interaction with kinesthetic feedback.

2 References and Definitions

2.1 Applicable Documents

[1] "IEEE Standard for High-Voltage Testing Techniques". In: *IEEE Std 4-2013 (Revision of IEEE Std 4-1995)* (May 2013), pp. 1–213. DOI: 10.1109/IEEESTD.2013.6515981.

[2] "IEEE Recommended Practices for Safety in High-Voltage and High-Power Testing". In: *ANSI/IEEE Std 510-1983* (1983), pp. 1–19. DOI: 10.1109/IEEESTD.1983.81973.

[3] U. Nanda and S. K. Pattnaik. "Universal Asynchronous Receiver and Transmitter (UART)". In: *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*. Vol. 01. 2016, pp. 1–5.

[4] "IEEE Draft Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats". In: *IEEE P1451.2/D20, February 2011* (2011), pp. 1–28.

2.2 Reference Documents

[5] Laura Sbernini Giovanni Saggio Francesco Riillo and Lucia Rita Quitadamo. "Resistive flex sensors: a survey". In: (2015).

[6] *MSP432P401R, MSP432P401M SimpleLink™ Mixed-Signal Microcontrollers*. Texas Instruments. 2019.

[7] *MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.0*. InvenSense. 2012.

[8] *Unity Documentation*. https://docs.unity3d.com/Manual/index.html, last accessed on 04/28/20.

[9] *Energia Language Reference*. https://energia.nu/reference/, last accessed on 04/28/20.

2.3 Definitions

MCU	Microcontroller Unit
IMU	Inertial Measurement Unit
GUI	Graphical User Interface
UI	User Interface
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
HV	High Voltage
PCB	Printed Circuit Board
kg	Kilogram
g	Gram
mA	Milliamp
μA	Microamp
mW	Milliwatt
cm	Centimeter
μm	Micrometer
Hz	Hertz

3 Physical Interface

3.1 Weight

3.1.1 Electrostatic Brake

The electrostatic braking system will be composed of 30 stainless steel electrode strips. These strips will be sectioned off into 2 different braking systems for each finger. The total mass of the steel electrodes will be no more than 50 g. The calculation is the following:

$$Area_{sheet} = 127cm * 15.24cm = 1935.48cm^2$$

$$Weight_{sheet} = 226.796g$$

$$Area_{strips} = 10 * (18cm * 1cm) + 20 * (9cm * 1cm) = 360cm^2$$

$$Weight_{strips} = \frac{360cm^2 * 226.796g}{1935.48cm^2} = 42.2g$$

An extra 21 g will account for the weight of the dielectric film. Another 55 g will account for the binding mechanisms, 3D-printed gliders and Velcro. This brings the total weight of the electrostatic braking system to 118.2 g. This does not include the weight of the knit glove. The 55 g of binding mechanisms include those that will be used for the hand tracking system as well.

3.1.2 Power Supply

The power supply will be an off-glove solution, sending the output voltages to the electrostatic brakes via insulated cabling. It will be created out of very small passive components as well as transistors, therefore the weight of the power supply will be no more then 1.5 kg, excluding cabling.

3.1.3 Hand Tracking

The hand tracking system is composed of 19 flex sensors and 1 IMU. The combined weight of all the sensors, wires, and 3D-printed materials shall not exceed 48 g: 24 g for the sensors and wires, 4 g for IMU, and 20 g for 3D-printed material. This does not include the weight of the glove. The hand tracking system will connect to an MCU and PCB; although it will not be placed on the glove, the weight of the MCU is 29 g and the weight of the PCB is 41 g.

3.2 Dimensions

3.2.1 Glove and Electrostatic Brake

The Haptic Feedback Glove will consist of a heat-resistant glove that measures 22 x 15 cm. The dimensions of the stainless steel electrodes will be 18 x 1 x 0.0127 cm for the middle plate and 9 x 1 x 0.0127 cm for the 2 outer plates. The dielectric that will be placed on the outside plates will cover the entire length of both sides of each electrode and will be 25.4μm thick.

3.2.2 Power Supply

The power supply shall consist of a voltage multiplication circuit, 10 individual switching outputs, a microcontroller, and the logic circuitry with which to control the output switching. All of the components shall be contained within a custom 3D-print housing in order to protect the circuit from the outside environment, as well as the user from exposure to high voltages. This housing shall be a 30 x 20 x 10 cm rectangular prism.

3.2.3 Hand Tracking

The 19 flex sensors and the IMU will be mounted on the heat-resistant glove. Each flex sensor will measure 5 x 0.8 x 0.22 cm. The IMU used to sense hand movement and positioning will measure 2.38 x 2.38 x 0.1 cm. The MCU and PCB will not be placed on the hand and will be 9.5 x 5.7 x 2.54 cm and 8.2 x 5.7 x 1.31 cm respectively.

3.3 Mounting Locations

3.3.1 Electrostatic Brake

The electrostatic brakes shall be mounted onto the finger with 3D-printed guides which shall allow the metal strips to glide smoothly when not braking. These guides shall also serve to isolate the upper joint braking system from the lower joint braking system. The overlap areas of the 2 joint systems shall be placed on top of one another, and it is important that the exposed metal strips do not touch.

Additionally, the strips shall be secured to the fingers using thin Velcro straps. There shall be 2 Velcro straps on the thumb and 3 Velcro straps on the 4 remaining fingers. There shall be additional Velcro straps on the back of the hand to further secure the ends of the metal strips.

3.3.2 Power Supply

The power supply shall be positioned off of the glove in a box containing all of the voltage boosting as well as voltage controls. A standard wall outlet cable shall feed into the box to provide power for the system, and a series of cables shall exit the supply to provide power to the electrostatic braking mechanisms on the glove.

3.3.3 Hand Tracking

The hand tracking sensors shall be strategically mounted on every segment of the finger and between the fingers to sense changes of movement and spreading. The base of the sensor shall not move, however the end of the sensor shall be placed on a 3D-printed slider to allow the sensor to bend without causing damage to it. On the glove there shall be 3 sensors on each digit, 2 on the thumb, 1 between each of the fingers (spreading of hand) and 1 from the middle of hand to the thumb (to measure movement of the thumb) for a total of 19. The IMU shall be placed in the middle of the glove to detect the rotation of the hand.

4 Thermal Interface

The power supply will be the only source of measurable heat emission. Operating at no more than 3 watts (W), the system shall provide no more than 5 degrees Celsius (C) of heat. A fan may be included inside of the power supply housing to ensure proper airflow and cooling across the design, minimizing any temperature variation.

5 Electrical Interface

5.1 Primary Input Power

The primary input power of the system shall be controllable, switching DC voltage. It shall take in the standard single-phase 120 V, 60 Hz output from an American wall outlet and output a series of steady DC switching voltages to the rest of the system. These voltages shall be controllable using a fast switch, allowing a programmable voltage setting at each electrostatic brake on the glove by changing the output duty cycle.

5.2 Signal Interfaces

5.2.1 Analog Flex Sensors

The analog flex sensors will communicate to the MCU by outputting a specific voltage range for each sensor. This voltage range – between 0 V and 5 V – will be linearized and mapped by the MCU to accurately determine the degree of movement of each finger. This information will then be sent out to other subsystems.

5.2.2 Power System MCU

The output signal from the user interface as well as the finger tracking will be sent to the power controller MCU. This data is then parsed into the 10 separate control signals required to individually operate control points across the glove. Each of these signals will translate directly to the duty cycle of the switching transistor, which in turn changes the duty cycle of the output transistor. This will allow for an easily controllable, variable output voltage which shall be utilized to apply variable force feedback across the glove. The formula to calculate the output voltage is the voltage across the high voltage line multiplied by the duty cycle.

$$V_O = V_{HV} * D$$

5.2.3 Digital IMU Sensor

The digital IMU sensor will communicate to the MCU by the I^2C protocol. The sensor will send data from the gyroscope and accelerometer to the MCU. The MCU will interpret and linearize this data and transmit to the host device.

5.2.4 Host Computer

The host computer will connect to the hand tracking MCU and the power controller MCU via a USB cable to transmit and receive information on both ends.

5.3 User Control Interface

The user control interface is split into two types of interactions with the device.

Physical Interaction The physical portion of the user control interface shall be simply wearing and using the glove. By using natural hand movements to adjust an array of variable resistors, the user is effectively controlling the UI on the host device. There will be no physical interfacing elements on the glove itself.

Digital Interaction The digital portion of the user control interface shall be the GUI on the host device that allows for calibration and setup of the device. This will be performed using simple mouse and keyboard commands, then the user will be informed when the glove is active and tracking.

Once the glove is active and tracking, the glove itself becomes the interfacing element to interact with the digital component of the device. There will be three types of interaction surfaces in the virtual environment: none, soft, and rigid. These three correspond to different feedback percentages, namely 0, 50%, and 100%, respectively.

6 Communications / Device Interface Protocols

6.1 Host Device

The host device will communicate with both the hand tracking and the power controller MCUs on the Haptic Feedback Glove using Universal Asynchronous Receiver-Transmitter (UART) protocol via USB 3.0. This will also provide 5V to the MCUs.

Both the hand tracking MCU and the host device will transmit data as soon as it is available. This means that it falls on the receiver to manage their input stream effectively. The receiving device may need to flush the input buffer every time that it reads data so as to have the most recent data available to it.

The hand tracking MCU will transmit with a baud rate of 115,200, while the host computer will transmit interaction data to the power controller with a baud rate of 9,600.

6.2 Device Peripheral Interface

The connection between the MCUs and computer will be handled through a serial port using UART. Analog and digital pins on the MCUs will be used to communicate with sensors and power switching circuits.

6.3 Communication Protocols

As previously stated, there are two different digital communication channels in the Haptic Feedback Glove: hand tracking and power controller. The protocols for these two channels will be as follows:

6.3.1 Hand Tracking Protocol

The data being transmitted from the hand tracking MCU to the host computer shall be in the format described below. Each line of data shall contain 21 comma-separated floating-point numbers containing rotation data. The first three data points shall consist of yaw, pitch, and roll of the hand. The subsequent 19 data points shall consist of rotation values for each finger. The order of the joint rotations shall be organized in groups of four, organized by finger in the order as follows:

Index, Middle, Thumb, Ring, Small

For each finger, the order of data will move from the base of the finger towards the tip, meaning that for a normal finger the data will be in the following order:

ProximalX, ProximalY, MiddleY, DistalY

For the thumb, data will be organized differently, and shall follow the order:

ProximalX, ProximalY, ProximalZ, DistalY

The diagram in Figure 1 breaks down this confusing protocol visually.

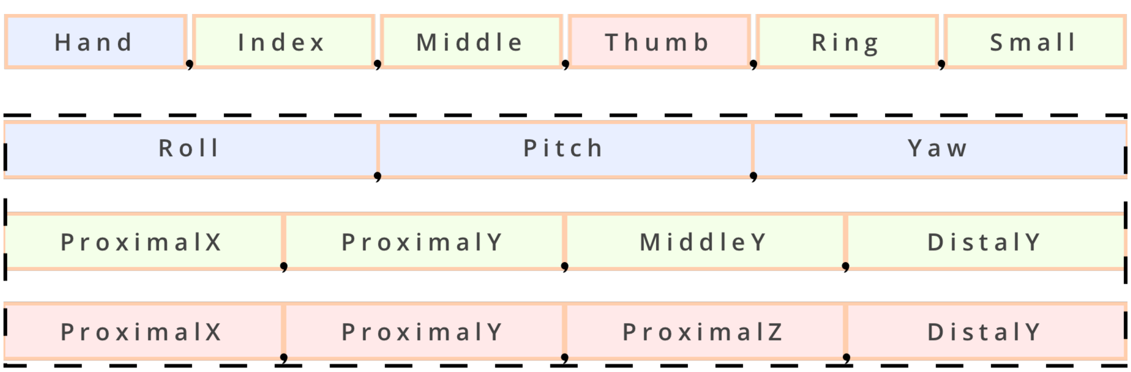


Figure 1: Data String Visual Breakdown

6.3.2 Power Controller Protocol

The data being transmitted from the host computer to the power controller MCU shall be in the format described below. Each line of data shall contain 10 comma-separated integers representing the percentage of force being applied to each point of contact. The data points shall be paired by finger in the order as follows:

Thumb, Index, Middle, Ring, Small

For each finger, the order of data will move from the base of the finger towards the tip. Since there are only two connection points on each finger, the value for the distal connection shall be the maximum value between the forces applied to the middle and distal joints. Since the thumb does not have a middle joint, the second value shall simply be the force applied to the distal joint. This means that the order of data for each finger is as follows:

$$\text{Proximal},\ \max(\text{Middle, Distal})$$

For example, if the user encloses a ball in his index finger and thumb, effectively applying 50% of full force to the tips of both fingers and 60% to the middle index, proximal index, and proximal thumb joints, then the command string following this protocol would be

$$60, 50, 60, 60, 0, 0, 0, 0, 0, 0$$

Likewise, if the user were to press the tips of their fingers against the surface of a rigid table, the command string folloing this protocol would be

$$0, 100, 0, 100, 0, 100, 0, 100, 0, 100$$

HAPTIC FEEDBACK GLOVE

Jonathan Arze
Luciano Brignone
Taylor Gage
Nicole Khoury

EXECUTION & VALIDATION PLAN

REVISION – Final
04/28/20

EXECUTION & VALIDATION PLAN FOR Haptic Feedback Glove

TEAM ⟨40⟩

APPROVED BY:

Jonathan Arze — Date

John Lusher II, P.E. — Date

Andrew Miller — Date

Revision History

Revision	Date	Author(s)	Description
Draft	09/28/19	JA	Created
A	09/30/19	JA	Initial Release
B	12/04/19	JA	Updated Validation Plan and added date map
C	04/28/20	JA	Updated validation and execution plans to match Spring 2020 progress
Final	04/28/20	JA	Final Commit

1 Fall 2019 Execution Plans

1	2	3	4	5	6	7	8	9	10	11	12
8/26	9/02	9/09	9/16	9/23	9/30	10/07	10/14	10/21	10/28	11/04	11/11

Table 2: Date equivalent for week numbers

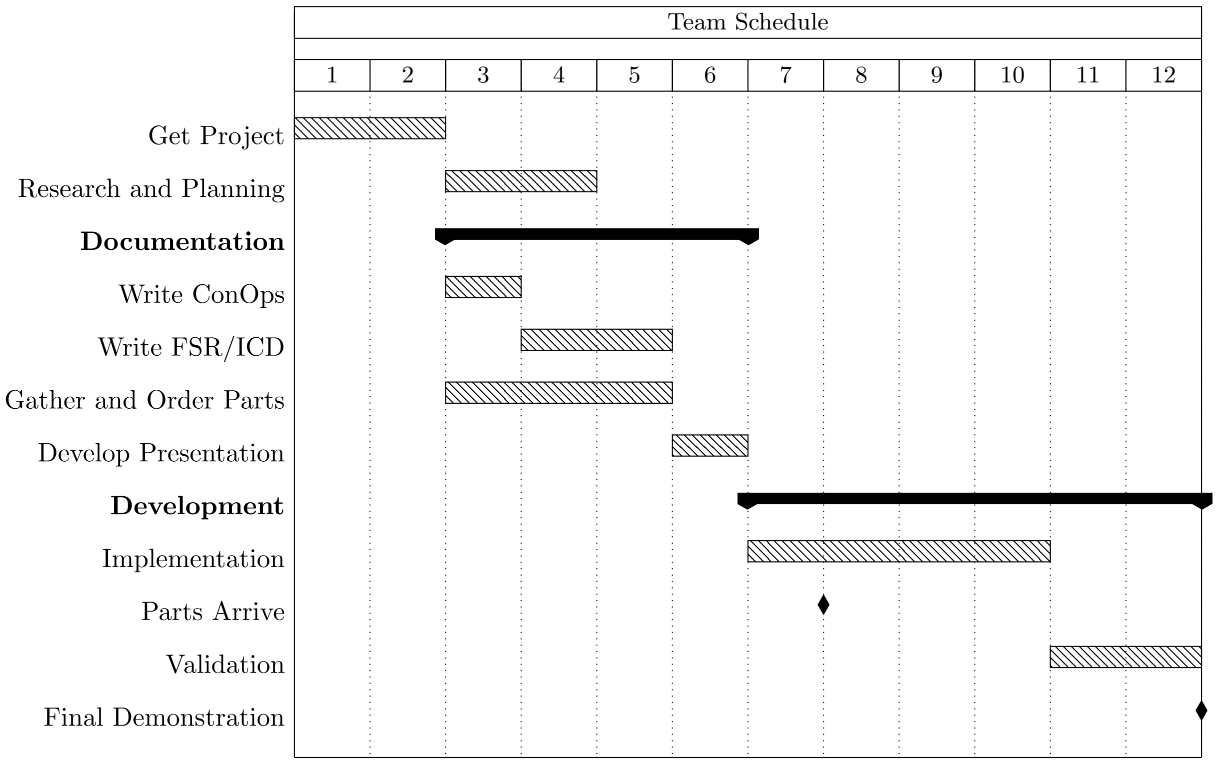
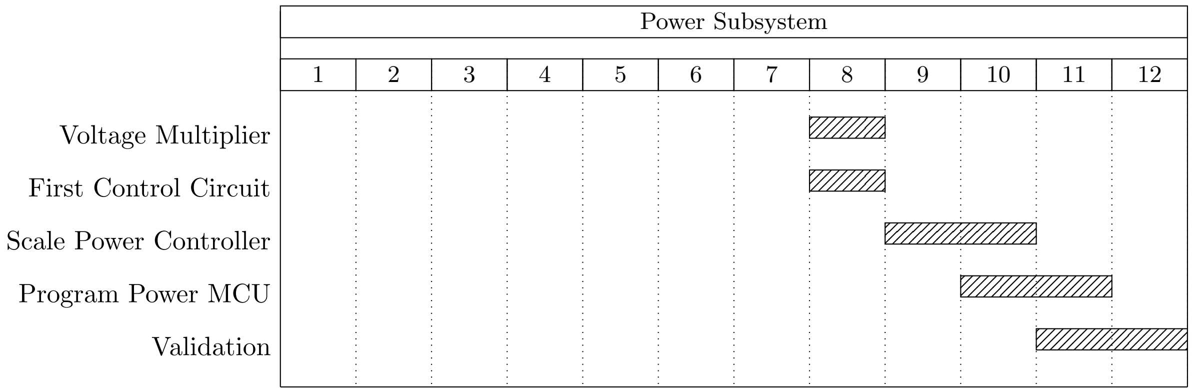
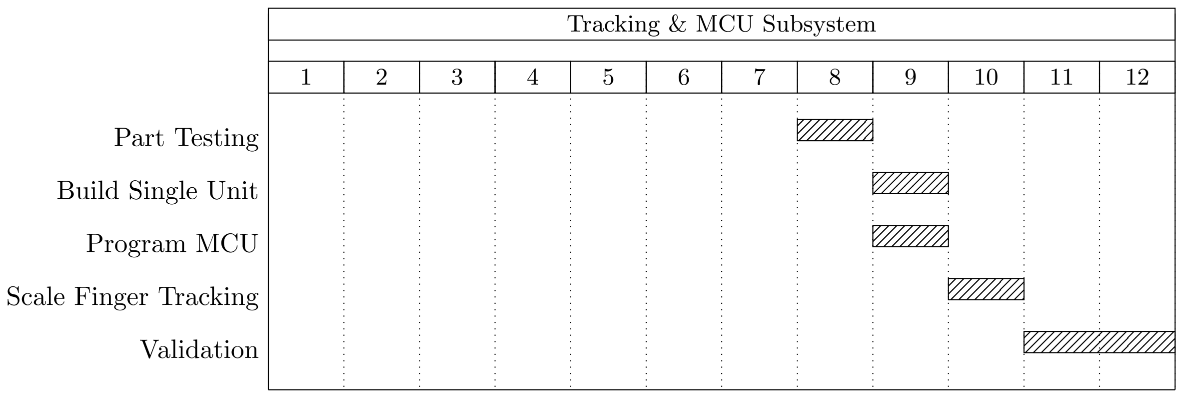
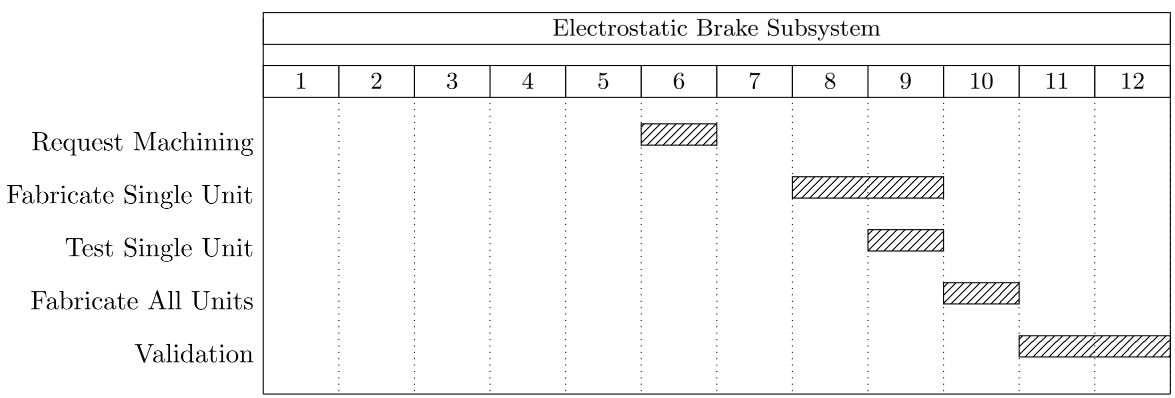
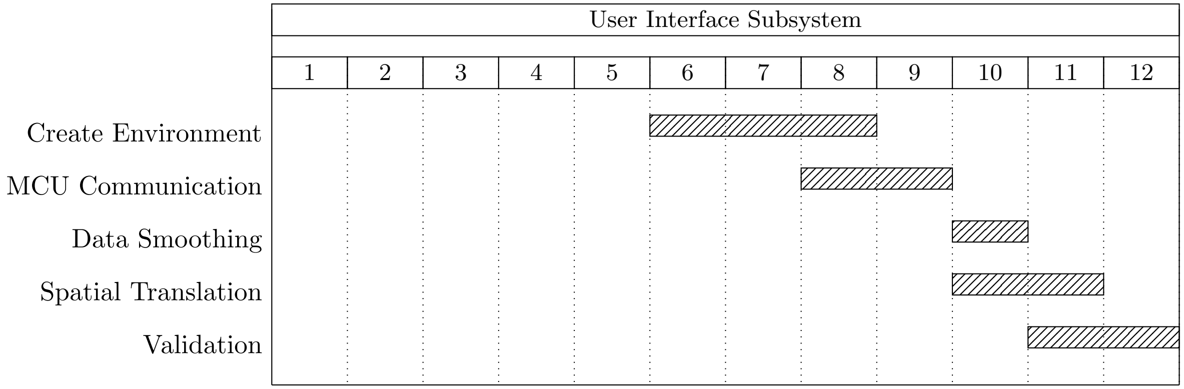


Figure 1: Fall 2019 Execution Plan


Power Subsystem
1
2
3
4
5
6
7
8
9
10
11
12
Voltage Multiplier
First Control Circuit
Scale Power Controller
Program Power MCU
Validation


Tracking & MCU Subsystem
1
2
3
4
5
6
7
8
9
10
11
12
Part Testing
Build Single Unit
Program MCU
Scale Finger Tracking
Validation


Electrostatic Brake Subsystem
1
2
3
4
5
6
7
8
9
10
11
12
Request Machining
Fabricate Single Unit
Test Single Unit
Fabricate All Units
Validation


User Interface Subsystem
1
2
3
4
5
6
7
8
9
10
11
12
Create Environment
MCU Communication
Data Smoothing
Spatial Translation
Validation

2 Spring 2020 Execution Plan

<table>
<tr><th></th><th>Feb 5</th><th>Feb 19</th><th>Feb 25 (checkpoint)</th><th>March 4</th><th>March 11</th><th>March 18</th><th>April 2 (Demo)</th></tr>
<tr><td>Electrostatic Braking Modules</td><td>Eliminate voltage arcing; increase force feedback</td><td>Iterate design for space efficiency; integrate with power system</td><td rowspan="4">Functional Prototype</td><td>Integrate onto glove</td><td rowspan="4">Reevaluation and Iterative Development</td><td>Safety improvements and final integration</td><td>Measure final forces</td></tr>
<tr><td>Power Supply</td><td>Design PCB</td><td>Order PCB/ start Host protocols</td><td>Validate PCB/Host interactions</td><td>Final integration, safety improvements, debug code</td><td>Wrap up validation of specs</td></tr>
<tr><td>Hand Tracking and MCU</td><td>Design Exoskeleton, Order new parts</td><td>Coms with host computer, calibrate sensors</td><td>Print exoskeleton, full integration, PCB</td><td>Reprint glove modifications, final mounting</td><td>Polish design, finalize validation</td></tr>
<tr><td>Graphical User Interface</td><td>Serial communication interface, map data to hand</td><td>Create and clean interaction logic, thumb and finger spread</td><td>Integrate into VR</td><td>Polish demo environment, debug, measure final latency</td><td>Usability testing and immersion observations</td></tr>
</table>

Figure 2: Spring 2020 Execution Plan

Unfortunately, there were several unforseen issues with printing that caused massive delays and pipelining problems; this really gave insight to the project planning side of engineering, and helps demonstrate why anticipating problems such as printing delays can help avoid pipelining. Because there were issues with printing, hardware could not be assembled, and because hardware could not be assembled, software could not be integrated and validated. There was also an issue accessing virtual reality equipment in the lab, which was a stretch goal anyway.

Also note that this schedule appears incomplete due to the premature ending of our project as a result of 2020's COVID-19 pandemic.

Electrostatic Braking Validation Plan

Task	Description	Requirements	P/F
Index Finger	1st joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Index Finger	2nd joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Middle Finger	1st joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Middle Finger	2nd joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Ring Finger	1st joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Ring Finger	2nd joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Small Finger	1st joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Small Finger	2nd joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Thumb	1st joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Thumb	2nd joint brake module glides correctly	Movement is not inhibited by gliders	Pass
Dielectric	Can withstand 300 V	Sending 300 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 350 V	Sending 350 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 400 V	Sending 400 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 450 V	Sending 450 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 500 V	Sending 500 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 550 V	Sending 550 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 600 V	Sending 600 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 700 V	Sending 700 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 750 V	Sending 750 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 800 V	Sending 800 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 850 V	Sending 850 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 900 V	Sending 900 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 950 V	Sending 950 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 1000 V	Sending 1000 V through dielectric plate does not cause a short	Pass
Dielectric	Can withstand 1300 V	Sending 1300 V through dielectric plate does not cause a short	Pass

Power Validation Plan

Test	Description	Test Requirements	P/F
1	Test resistors before soldering to ensure correct value	Tested resistance must match advertised within 1%	Pass
2	Test resistors after soldering for correct value	Tested resistance must match advertised within 1%	Pass
3	Ensure no shorts between IC socket pins after soldering	Use diode mode on multimeter to ensure no connections between pins	Pass
4	Ensure no shorts after soldering GND of IC socket to GND of power	Use diode mode on multimeter to ensure no connections between pins	Pass
5	Test IC socket GND connection with test voltage	Voltage applied to GND = 6 V, Voltage measured at IC socket GND pin = 6 V	Pass
6	Ensure no shorts after soldering VCC of IC socket to VCC of power	Use diode mode on multimeter to ensure no connections between pins	Pass
7	Test IC socket VCC connection with test voltage	Voltage applied to VCC = 6 V, Voltage measured at IC socket VCC pin = 6 V	Pass
8	Test IC turn on threshold	Ensure Vin = 2.6 produces output with VCC = 9 V	Pass
9	Control Signal Fidelity Check	Check low output waveform matches input waveform, square wave input	Pass
10	IC turn on voltage controller check (low voltage) INVERTED	HV = 25 V. Connect Logic in A to 0 V across 2k Ohms (0 uA), and connect Out A to gate of Q1. When Logic A = 0 V ouput of Q1 should be 0.2 V ± 50%	Pass
11	IC turn off voltage controller check (low voltage) INVERTED	HV = 25 V. Connect Logic in A to 2.6 V across 2k Ohms (105 uA), and connect Out A to gate of Q1. When Logic A = 2.6 V ouput of Q1 should be 25 V ± 15%	Pass
12	Measure voltage drop accross drain resistor series of Q1 with Q1 on (Low voltage) INVERTED	HV = 25 V, voltage at drain of Q1 should drop to around 0.9% HV (0.225 V)	Pass
13	IC turn off voltage check HIGH VOLTAGE TEST (INVERTED)	Connect Logic in A to 0 V across 2k Ohms (0 uA), and connect Out A to gate of Q1. When Logic A = 0 V ouput of Q1 should be <0.1% HV	Pass
14	IC turn on voltage check HIGH VOLTAGE TEST (INVERTED)	Connect Logic in A to 2.6 V across 2k Ohms (105 uA), and connect Out A to gate of Q1. When Logic A = 2.6 V ouput of Q1 should be HV ± 5%	Pass
15	Ensure no shorts after soldering HV source live socket	Use didoe mode on multimeter to ensure no shorts between live and nuetral	Pass
16	Ensure no shorts after soldering HV source ground socket	Use didoe mode on multimeter to ensure no shorts between live and nuetral	Pass
17	IC Duty Test	Apply 0%-100% Duty cycle to each input and verify outputs match.	Pass
18	Test diode voltage drop for each diode before soldering new leads	Tested voltage drop should be <0.6 V	Pass
19	Test diode voltage drop for each diode after soldering new leads	Tested voltage drop should be <0.6 V	Pass
20	Test film capacitor values before soldering new leads	Tested capacitance should be within 5% of 5 uF	Pass
21	Test film capacitor values after soldering new leads	Tested capacitance should be within 5%	Pass

22	Test diode voltage drop for each diode after soldering to board	Tested voltage drop should be <0.6 V	Pass
23	Test film capacitor value after soldering to board	Tested voltage drop should be <0.6 V	Pass
24	Test each electrolytic capactifor value before soldering to board	Tested capacitance should be within 10% of 27 uF	Pass
25	Test each electrolytic capactifor value after soldering to board	Tested capacitance should be within 10% of 27 uF	Pass
26	Recheck connections to ensure no unwanted shorts	Use diode mode on multimeter to ensure no unwanted shorts	Pass
27	Ensure no shorts with power cord connection after modification	Use didoe mode on multimeter to ensure no shorts between live and nuetral, as well as any other surrounding componants	Pass
28	Test power cord GND connection with test voltage	Voltage applied to GND of power cord = 6 V, Voltage measured at GND terminal = 6 V	Pass
29	Test power cord live connection with test voltage	Voltage applied to live of power cord = 6 V, Voltage measured at live terminal = 6 V	Pass
30	Test GFCI adapter	**Use self trip test on GCFI, it MUST PASS SELF TEST.**	Pass
31	Test first stage of HV rectification/mulitplication (wall socket input)	Tested output voltage should be within 14 %of 170V (DC)	Pass
32	Test second stage of HV multiplication (wall socket input)	Tested output voltage should be within 14% of 340V (DC)	Pass
33	Test third stage of HV mulitplication (wall socket input)	Tested output voltage should be within 14% of 510V (DC)	Pass
34	Test fourth stage of HV multiplication (wall socket input)	Tested output voltage should be within 14% of 680V (DC)	Pass
35	Test fifth stage of HV multiplication (wall socket input)	Tested output voltage should be within 14% of 850V (DC)	Pass
36	Test sixth stage of HV mulitplication (wall socket input)	Tested output voltage should be within 14% of 1020V (DC)	Pass
37	Test seventh stage of HV mulitplication (wall socket input)	Tested output voltage should be within 14% of 1190V (DC)	Pass
38	Test eighth stage of HV multiplication (wall socket input)	Tested output voltage should be within 14% of 1360V (DC)	Pass
39	Observe heat change at each componant	The temperature increase across any single componant should be no greater then 2 degrees celcius	Pending
40	Test loaded output voltage	Loaded output drop must be under 5% of HV line	Pass
41	Test clock divider program at 10-20 Hz	Program internal clock divider to produce output switching at 10-20 Hz adjustable	Pass
42	Test PWM code	Program pins to produce 3 V output at adjustable duty cycle (verify with oscilloscope)	Pass
43	Short Test	Short live and ground of control circuit output, voltage multiplier circuit must still operate as intended after.	Pass
44	High voltage short test	Short live and ground of control circuit output, control circuit must still operate as intended after.	Pass

45	Vout1 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
46	Vout1 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
47	Vout1 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
48	Vout1 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
49	Vout1 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
50	Vout1 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
51	Vout1 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
52	Vout1 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
53	Vout1 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
54	Vout1 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
55	Vout2 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
56	Vout2 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
57	Vout2 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
58	Vout2 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
59	Vout2 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
60	Vout2 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
61	Vout2 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
62	Vout2 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
63	Vout2 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
64	Vout2 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
65	Vout3 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
66	Vout3 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
67	Vout3 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
68	Vout3 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
69	Vout3 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
70	Vout3 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass

71	Vout3 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
72	Vout3 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
73	Vout3 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
74	Vout3 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
75	Vout4 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
76	Vout4 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
77	Vout4 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
78	Vout4 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
79	Vout4 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
80	Vout4 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
81	Vout4 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
82	Vout4 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
83	Vout4 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
84	Vout4 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
85	Vout5 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
86	Vout5 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
87	Vout5 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
88	Vout5 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
89	Vout5 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
90	Vout5 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
91	Vout5 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
92	Vout5 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
93	Vout5 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
94	Vout5 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
95	Vout6 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
96	Vout6 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass

97	Vout6 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
98	Vout6 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
99	Vout6 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
100	Vout6 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
101	Vout6 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
102	Vout6 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
103	Vout6 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
104	Vout6 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
105	Vout7 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
106	Vout7 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
107	Vout7 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
108	Vout7 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
109	Vout7 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
110	Vout7 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
111	Vout7 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
112	Vout7 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
113	Vout7 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
114	Vout7 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
115	Vout8 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
116	Vout8 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
117	Vout8 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
118	Vout8 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
119	Vout8 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
120	Vout8 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
121	Vout8 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
122	Vout8 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass

123	Vout8 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
124	Vout8 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
125	Vout9 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
126	Vout9 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
127	Vout9 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
128	Vout9 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
129	Vout9 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
130	Vout9 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
131	Vout9 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
132	Vout9 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
133	Vout9 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
134	Vout9 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
135	Vout10 HV Voltage Selection Test 10%	Adjust duty cycle to 10%, output should be 130 V ± 5%	Pass
136	Vout10 HV Voltage Selection Test 20%	Adjust duty cycle to 20%, output should be 260 V ± 5%	Pass
137	Vout10 HV Voltage Selection Test 30%	Adjust duty cycle to 30%, output should be 390 V ± 5%	Pass
138	Vout10 HV Voltage Selection Test 40%	Adjust duty cycle to 40%, output should be 520 V ± 5%	Pass
139	Vout10 HV Voltage Selection Test 50%	Adjust duty cycle to 50%, output should be 650 V ± 5%	Pass
140	Vout10 HV Voltage Selection Test 60%	Adjust duty cycle to 60%, output should be 780 V ± 5%	Pass
141	Vout10 HV Voltage Selection Test 70%	Adjust duty cycle to 70%, output should be 910 V ± 5%	Pass
142	Vout10 HV Voltage Selection Test 80%	Adjust duty cycle to 80%, output should be 1040 V ± 5%	Pass
143	Vout10 HV Voltage Selection Test 90%	Adjust duty cycle to 90%, output should be 1170 V ± 5%	Pass
144	Vout10 HV Voltage Selection Test 100%	Adjust duty cycle to 100%, output should be 1300 V ± 5%	Pass
145	All Vout HV Voltage Selection Test 10%	Adjust duty cycle to 10%, outputs should be 130 V ± 5%	Pending
146	All Vout HV Voltage Selection Test 20%	Adjust duty cycle to 20%, outputs should be 260 V ± 5%	Pending
147	All Vout HV Voltage Selection Test 30%	Adjust duty cycle to 30%, outputs should be 390 V ± 5%	Pending
148	All Vout HV Voltage Selection Test 40%	Adjust duty cycle to 40%, outputs should be 520 V ± 5%	Pending

149	All Vout HV Voltage Selection Test 50%	Adjust duty cycle to 50%, outputs should be 650 V ± 5%	Pending
150	All Vout HV Voltage Selection Test 60%	Adjust duty cycle to 60%, outputs should be 780 V ± 5%	Pending
151	All Vout HV Voltage Selection Test 70%	Adjust duty cycle to 70%, outputs should be 910 V ± 5%	Pending
152	All Vout HV Voltage Selection Test 80%	Adjust duty cycle to 80%, outputs should be 1040 V ± 5%	Pending
153	All Vout HV Voltage Selection Test 90%	Adjust duty cycle to 90%, outputs should be 1170 V ± 5%	Pending
154	All Vout HV Voltage Selection Test 100%	Adjust duty cycle to 100%, outputs should be 1300 V ± 5%	Pending

MCU & Hand Tracking Validation Plan

Task	Description	Requirements	P/F
Sensor Resistors	Get the correct resistors for the sensors	The sensors should have a resistance between 20k to 120k	Pass
Index Finger	1st segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Index Finger	2nd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Index Finger	3rd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Middle Finger	1st segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Middle Finger	2nd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Middle Finger	3rd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
ring Finger	1st segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
ring Finger	2nd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
ring Finger	3rd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Small Finger	1st segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Small Finger	2nd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Small Finger	3rd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Thumb	1st segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Thumb	2nd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Thumb	3rd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	Pass
Thumb & Index	the space between the fingers is mapped correctly with <= +/- 5* accuracy	The mapping should be less than 5 degrees off the actual angle	Pass
Index & Middle	the space between the fingers is mapped correctly with <= +/- 5* accuracy	The mapping should be less than 5 degrees off the actual angle	Pass
Middle & Ring	the space between the fingers is mapped correctly with <= +/- 5* accuracy	The mapping should be less than 5 degrees off the actual angle	Pass
Ring & Small	the space between the fingers is mapped correctly with <= +/- 5* accuracy	The mapping should be less than 5 degrees off the actual angle	Pass
PCB - Index Finger	1st segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	Pass
PCB - Index Finger	2nd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	Pass

PCB - Index Finger	3rd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - Middle Finger	1st segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	Pass
PCB - Middle Finger	2nd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	Pass
PCB - Middle Finger	3rd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - ring Finger	1st segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - ring Finger	2nd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - ring Finger	3rd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - pinky Finger	1st segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - pinky Finger	2nd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - pinky Finger	3rd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - thumb Finger	1st segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - thumb Finger	2nd segement of finger is mapped correctly	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - thumb Finger	3rd segement of finger is mapped correctly with PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - Thumb & Index	the space between the fingers is mapped correctly with <= +/- 5* accuracy with the PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - Index & Middle	the space between the fingers is mapped correctly with <= +/- 5* accuracy with the PCB	The mapping should be less than 5 degrees off the actual angle	Pass
PCB - Middle & Ring	the space between the fingers is mapped correctly with <= +/- 5* accuracy with the PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB - Ring & Pinky	the space between the fingers is mapped correctly with <= +/- 5* accuracy with the PCB	The mapping should be less than 5 degrees off the actual angle	In Pogress
PCB	PCB must work correctly	The PCB should not interfere with any data	In Pogress

IMU	Y-axis rotation	The mapping should be less than 2.5 degrees off the actual angle	Pass
IMU	X-axis rotation	The mapping should be less than 2.5 degrees off the actual angle	Pass
IMU	Z-axis rotation	The mapping should be less than 2.5 degrees off the actual angle	Pass
MCU	I2C protocol	Test if it works with IMU	Pass
MCU	Calibration Test	Must calibrate the sensors before use	Pass
MCU to host	send data to host	Must send accurate data to host without any problems	Pass
MCU to host	send resistor data to host	Must send accurate degree data to host without any problems for all the sensors	In Pogress

GUI Validation Plan

Test	Description	Test Requirements	P/F
Thumb Distal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Thumb Middle Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Index Distal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Index Middle Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Index Proximal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Middle Distal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Middle Middle Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Middle Proximal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Ring Distal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Ring Middle Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Ring Proximal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Small Distal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Small Middle Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Small Proximal Joint	Spawn block on joint and listen for beginning of interaction	Registered start of interaction in the next frame	Pass
Interaction Dynamics - Register End			Pass
Thumb Distal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass

Thumb Middle Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Index Distal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Index Middle Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Index Proximal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Middle Distal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Middle Middle Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Middle Proximal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Ring Distal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Ring Middle Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Ring Proximal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Small Distal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Small Middle Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Small Proximal Joint	Despawn block and listen for end of interaction	Registered end of interaction in the next frame	Pass
Thumb Distal Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation	Resolves within 3° for 5 frames within 250ms	Fail*
Thumb Middle Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*
Thumb Proximal Joint – Y Axis	Apply target rotation of 45 degrees (°) on the y-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*

Thumb Proximal Joint – X Axis	Apply target rotation of 5 degrees (°) on the x-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*
Thumb Proximal Joint – Z Axis	Apply target rotation of 5 degrees (°) on the z-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Index Distal Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Index Middle Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Index Proximal Joint – Y Axis	Apply target rotation of 45 degrees (°) on the y-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Index Proximal Joint – Z Axis	Apply target rotation of 5 degrees (°) on the z-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Middle Distal Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Middle Middle Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*
Middle Proximal Joint – Y Axis	Apply target rotation of 45 degrees (°) on the y-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*
Middle Proximal Joint – Z Axis	Apply target rotation of 5 degrees (°) on the z-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Ring Distal Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Ring Middle Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*
Ring Proximal Joint – Y Axis	Apply target rotation of 45 degrees (°) on the y-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass

Ring Proximal Joint – Z Axis	Apply target rotation of 5 degrees (°) on the z-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Small Distal Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Small Middle Joint	Apply target rotation of 45 degrees (°) and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Fail*
Small Proximal Joint – Y Axis	Apply target rotation of 45 degrees (°) on the y-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Small Proximal Joint – Z Axis	Apply target rotation of 5 degrees (°) on the z-axis and listen for actual rotation to resolve to this rotation within 0.1 degrees (°)	Resolves within 3° for 5 frames within 250ms	Pass
Spatial Translation – Bounds Max Check			Pass
Thumb Distal Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Thumb Middle Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Thumb Proximal Joint – Y Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Thumb Proximal Joint – X Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Thumb Proximal Joint – Z Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Index Distal Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Index Middle Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass

Index Proximal Joint – Y Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Index Proximal Joint – Z Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Middle Distal Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Middle Middle Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Middle Proximal Joint – Y Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Middle Proximal Joint – Z Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Ring Distal Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Ring Middle Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Ring Proximal Joint – Y Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Ring Proximal Joint – Z Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Small Distal Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Small Middle Joint	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Small Proximal Joint – Y Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass
Small Proximal Joint – Z Axis	Apply target rotation of joint maximum + 5 degrees (°)	Resolves to joint maximum	Pass

Spatial Translation – Bounds Min Check			Pass
Thumb Distal Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Thumb Middle Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Thumb Proximal Joint – Y Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Thumb Proximal Joint – X Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Thumb Proximal Joint – Z Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Index Distal Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Index Middle Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Index Proximal Joint – Y Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Index Proximal Joint – Z Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Middle Distal Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Middle Middle Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Middle Proximal Joint – Y Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Middle Proximal Joint – Z Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass

Ring Distal Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Ring Middle Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Ring Proximal Joint – Y Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Ring Proximal Joint – Z Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Small Distal Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Small Middle Joint	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Small Proximal Joint – Y Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Small Proximal Joint – Z Axis	Apply target rotation of joint minimum - 5 degrees (°)	Resolves to joint minimum	Pass
Data Reception	Send several sets of sample data over serial to host computer and compare	Sample data is the same on both sides	Pass
Too Much Data	Send sample data with too many data points	Correctly flags data as incorrectly formatted	Pass
Too Little Data	Send sample data with too few data points	Correctly flags data as incorrectly formatted	Pass
Position Comparison	Compare sent data with resulting hand position	Hand position matches sent data	Pass
Continuous Stream Test	Test the continuous stream of live data from the glove	Hand visually matches glove positioning	Pass
Data Transmission	Send several sets of sample data over serial to power controller and compare	Sample data is the same on both sides	Pass
Data Formatting	Trigger different sets of interactions and compare to expected results	Correctly formats data strings	Pass

*** This test was not a Functional System Requirement, but rather to help give information about the functionality of the system. Failing this test does not indicate a failure to complete system requirements.**

HAPTIC FEEDBACK GLOVE

Jonathan Arze
Luciano Brignone
Taylor Gage
Nicole Khoury

FINAL SUBSYSTEM REPORT

REVISION – Final
04/28/20

FINAL REPORT
FOR
Haptic Feedback Glove

TEAM ⟨40⟩

APPROVED BY:

Jonathan Arze Date

John Lusher II, P.E. Date

Andrew Miller Date

Revision History

Revision	Date	Author(s)	Description
Draft	11/24/19	JA	Created
A	12/04/19	JA	Initial Release
B	04/25/20	JA	Updated each subsystem for Fall 2020
Final	04/28/20	JA	Final Commit

Contents

List of Tables

List of Figures

References

[1] 3DHaupt. *Rigged Hands*. `https://www.turbosquid.com/3d-models/hands-rigged-3ds-free/932376`, last accessed on 12/03/19. 2015.

1 Introduction

The purpose of the Haptic Feedback Glove is to improve on the limitations of previous devices designed. The Haptic Feedback Glove will allow the user to interact with a virtual environment, achieved by having an improved integration with the biological nervous system. The goal of this system is to provide new and intuitive ways to interact with virtual data that maintains the dexterity of the hand while also providing kinesthetic feedback to the user about their interactions in the virtual environment. The user interface (UI) for this system shall include one or more virtual environments created in Unity 3D software in which to calibrate the glove and interact with virtual objects. Figure 1 below shows the high-level flowchart of information in this project.

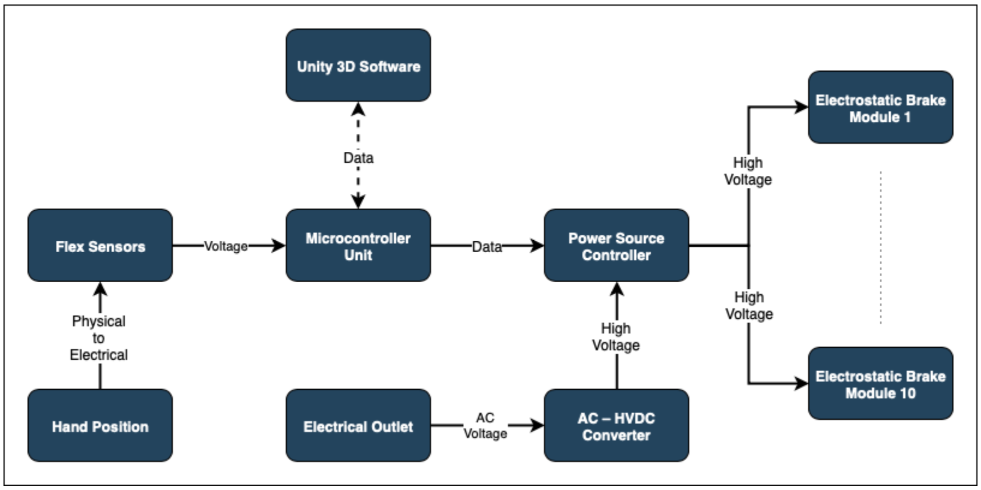


Figure 1: High-Level Flowchart of the Haptic Feedback Glove

The Haptic Feedback Glove shall be able to record and transmit kinematic data for each finger and the wrist in order to recreate the position of the hand in a virtual environment. This data shall be sent by the microcontroller (MCU) to a host computer running Unity 3D, and the host computer shall render the positioning of the user's hand. The host computer shall then communicate data about interactions between the rendered hand and other objects in the virtual environment back to the MCU. The MCU shall then transmit the interaction data to the power source controller. This controller shall enable independent electrostatic brake (feedback) modules, and may determine the voltage applied to the module.

The host computer may implement some machine learning algorithms to assist in the interpretation of position data. The glove may also be coupled with a virtual reality tracking puck to allow for spatial tracking integrated with the HTC Vive.

The following definitions differentiate between requirements and other statements.

Shall:	This is the only verb used for the binding requirements.
Should/May:	These verbs are used for stating non-mandatory goals.
Will:	This verb is used for stating facts or declaration of purpose.

2 Electrostatic Braking Module Subsystem

2.1 Subsystem Introduction

The purpose of the electrostatic braking subsystem is to provide the user with physical feedback to correspond with objects being interacted with in the virtual environment. Varying voltages simulate interactions with objects of different materials. Upon completion, individual finger joints will be locked in place to prevent the user from moving past the boundary of the virtual object.

To be able to provide forces to individual joints of the finger, there are 2 modules fitted for each finger. This enables more accurate feedback and a more realistic way of interacting with the virtual objects.

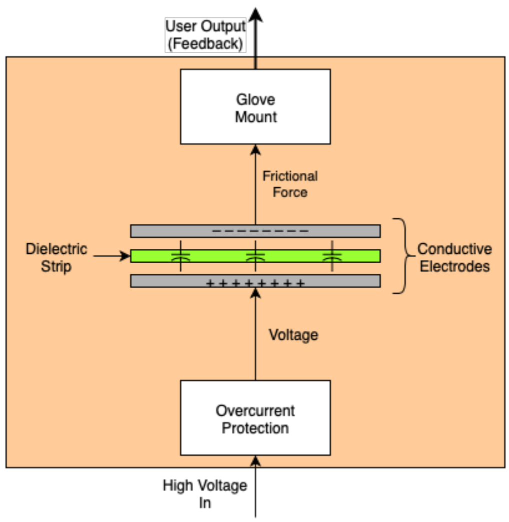


Figure 2: Electrostatic Brake Subsystem Diagram

2.2 Capacitor Hardware

The large capacitors that are placed on the fingers of the glove serve to provide the tactile feedback to the user. The metal is flexible and light enough that it does not inhibit the user's natural finger movement. Polyimide film was chosen as the dielectric between the plates because of its high breakdown voltage.

2.2.1 Operation

The capacitors that each braking module consists of are made from 3 strips of stainless steel shim. For the purpose of mounting the capacitors to the glove, each strip is 1 cm wide. The 2 outer plates are coated in wax paper, which serves as the dielectric. After voltage is applied, the outer plates are forced inward and clamp the middle plate in place. This provides a suggestive force to a specific joint on the user's finger. The voltage can be applied to either the outer plates or the middle plate, but for safety purposes it is applied to outer plates.

Figure 3: Capacitor System Design

The separation between the plates is the largest barrier in terms of generating force feedback. The possible force is inversely proportional to the square of the distance. After taking into account the thickness of the wax paper, the force is greatly reduced. This is why a double parallel plate capacitor was used, doubling the overlap area to compensate for the separation.

2.2.2 Validation

There were several design iterations before the final double parallel plate capacitor was decided on. The goal was to maximize the overlap area while preventing voltage arcing.

The original design was connected to the power supply and the grounded plate was measured for voltage changes. The cause of the voltage arcing was determined to be dielectric breakdown because the polyimide film that had been previously used was too thin and prone to breakage. This caused the ground plate to build up a high positive voltage, which led to the arcing. Changing to wax paper allowed the dielectric plates to be handled more without possible ripping.

The increased thickness of the wax paper solved the issue of voltage arcing due to dielectric breakdown, but the increased thickness was detrimental to the force feedback. Several design prototypes were created in an attempt to maximize overlap area of the plates to compensate for this.

Figure 4 shows a large double parallel plate capacitor design. This was created with 3 thicker, inflexible steel plates. The middle plate was able to glide freely between the outer 2 plates, which were covered with wax paper. This design provided substantial perceptible force feedback, but would be too large to place directly on the glove. While mounting these plates to the forearm would have been possible, the mechanical structure required to transfer the force feedback from the forearm to the finger would have been too involved.

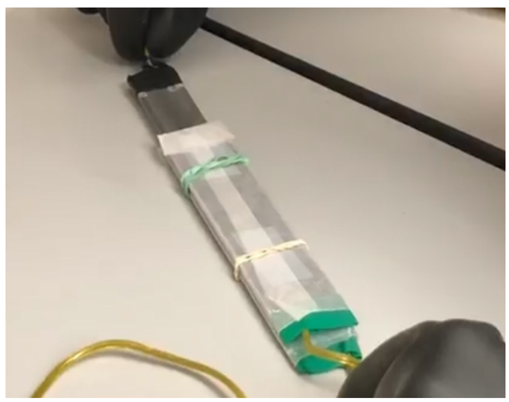


Figure 4: Initial Double Parallel Plate Capacitor Prototype

Figure 5 is a force diagram of the cylindrical prototype that was created. It consisted of a hollow steel tube that was covered in wax paper conductive metal tape. The wax paper served as the dielectric between the conductive tape and the steel tube. The outer tube had to be made out of something flexible so that it could collapse inward onto the tube when voltage was applied. The collapsing would prevent the inner tube from sliding freely, locking the finger in place. This design was tested and found to be able to withstand maximum voltage from the power supply. The cylindrical design increased the overlap area in relation to the original mechanism, but would require forearm mounting similar to the initial double parallel plate capacitor (seen in Figure 4). Ultimately, there needed to be a flexible, light design that could be mounted directly on top of the hand.

2.3 Force Feedback

The maximum voltage produced by the power supply is 1305 V, therefore a maximum voltage of 1300 V is used for force calculation purposes. Due to size constraints of the glove, overlap area is not a highly variable factor. While keeping the overlap area to a maximum of 9 cm^2 and varying the applied voltage, the relationship between force feedback and voltage can be seen from the following equation and Figure 6.

$$\epsilon_r \epsilon_0 A \left(\frac{V}{d}\right)^2 \tag{1}$$

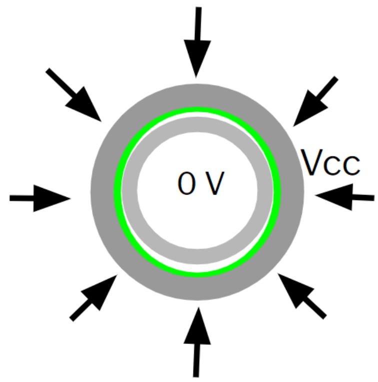


Figure 5: Cylindrical Capacitor Prototype

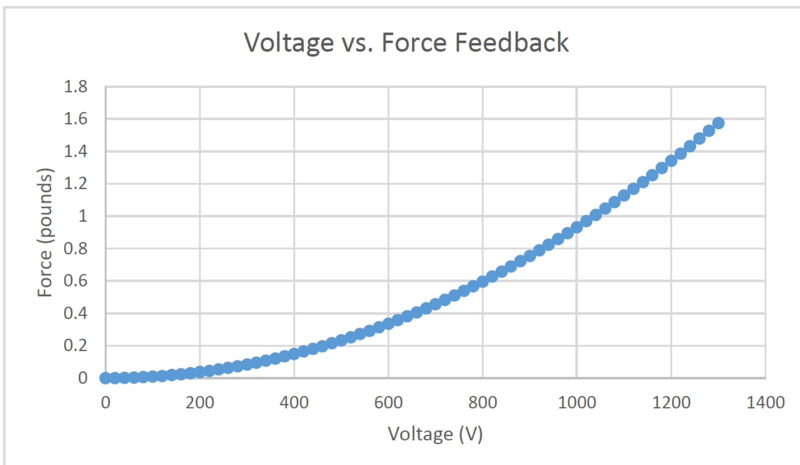


Figure 6: Force Feedback

The air gap between the plates needs to be almost nonexistent. With the thickness of the dielectric, any additional separation is detrimental to observable force feedback. Several different methods were used to minimize plate separation, with the final design consisting of a double parallel plate capacitor. This effectively doubles the overlap area in an attempt to offset the thickness of the wax paper.

2.3.1 Validation

Validation was conducted on the feedback of the braking modules by integrating with the power system and measuring plate interaction with video recordings. Manual pull tests were also performed to evaluate force resistance.

2.4 Glove Hardware

The final iteration of the glove design utilized 3D printed gliding mechanisms (Figure 7) to keep the plates close together. The outer 2 dielectric plates extend to the back of the hand, while the middle plate is secured to the tip of the finger. This allows the middle plate to glide freely with any movement of the fingers.

The Velcro straps allow the glove to be fitted to different hands, and the mechanisms are secured once the user is wearing the glove. More 3D printed devices were designed and made in order to integrate the braking modules with the hand tracking. These devices would have replaced the Velcro straps.

Due to the nature of the stainless steel, the wires could not be soldered on. Instead, holes were made in the bottom of the plates to thread wire through. The wire was twisted and soldered to itself for structural security. The connections on the outer plates were covered with wax paper and electrical tape, and electrical tape on the middle plate.

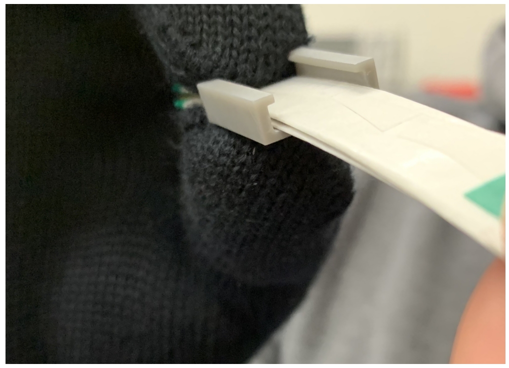


Figure 7: Glider Mechanism

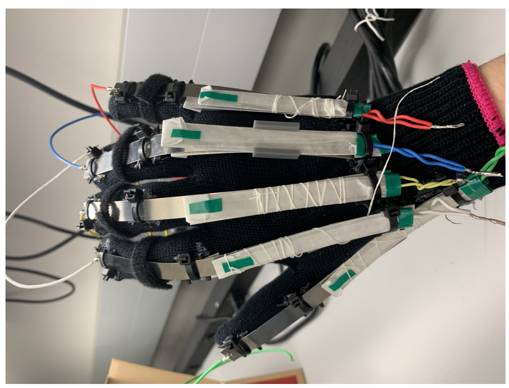


Figure 8: Glove Design

2.4.1 Validation

The gliding mechanism for the capacitor plates was important to troubleshoot to ensure system functionality. It had to be a tight enough fit to ensure that the plates were kept as close together as possible, but if it was too small it could tear the wax paper dielectric, causing a very dangerous short. Precise tools were used to gather the plate measurements, and each set of braking modules had to be fabricated in the same way to ensure that they would all fit in the same gliders. Figure 9 shows an image from the .stl file of the final design.

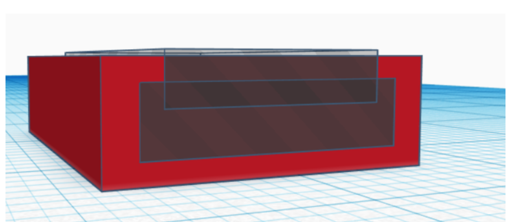


Figure 9: Final Glider Design File

2.5 Subsystem Conclusion

The braking system has demonstrated force feedback with several different design prototypes. Ultimately the design chosen was the most flexible and easiest to mount directly onto the desired point of contact. The main concern with the braking modules is reducing and standardizing this separation to increase force feedback. This mechanism works for its intended purpose, and further testing and glove design iterations will optimize it.

2.5.1 Future

The future of this subsystem requires fitting all the braking modules with the 3D printed gliders and integrating with the hand tracking subsystem. The braking system is fully integrated with the power supply, and full system testing can begin once the braking and hand tracking systems can function cohesively on a single glove.

3 Power Subsystem

3.1 Subsystem Introduction

This power subsystem is directly responsible for taking the digital force interaction output data from the GUI subsystem and translating it into an analog voltage. Using this data, 10 outputs are individually controlled to each produce a chosen voltage anywhere from 0 volts to 1300 volts, approximately. The purpose of this is to turn digital interactions into real-world force feedback through the application of the electrostatic braking modules. This power subsystem is made possible by the combination of 3 smaller subsystems, which are the voltage multiplier, voltage control circuit, and voltage control software. Each of these individual systems received functional testing and validation to ensure that it would provide the necessary 10 uniquely controlled outputs.

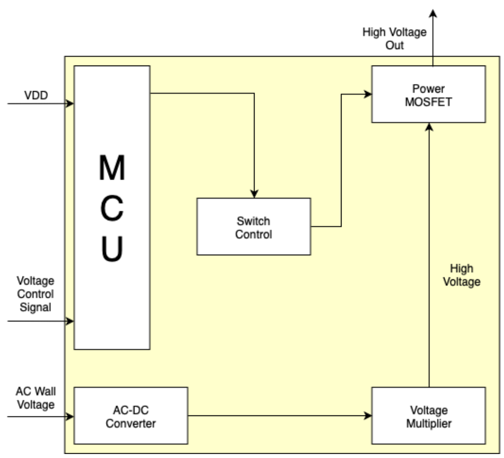


Figure 10: Power Subsystem Diagram

3.2 Voltage Multiplier

3.2.1 Design

The power system is designed to draw its power from a standard American wall outlet, which provides an AC voltage of around 120 V at 60 hertz. The electrostatic breaks require a either a DC or square wave voltage going from 0 V up to as high as 1300 V for maximum force feedback. This change from AC to DC, as well as the increase from 120 V to a much higher voltage were the immediate challenges that the power system had to overcome.

For these reasons, I built an AC to DC converter that also serves as a voltage multiplier. My system takes the 120 V AC and outputs roughly 1300 V DC. The topology that I decided on constructing is known as a Cockcroft-Walton Voltage Multiplier. This design consists of capacitors and diodes in an alternating ladder configuration which causes the reference voltages of each capacitor to be equal to the total charge of the capacitor that precedes it. What this means is that the first capacitor will charge to around 170 V, the peak voltage of the wall outlet sine wave. The second capacitor will then charge to 170 V above the charge of the first capacitor, resulting in 340 V. The third capacitor will then charge 170 V above that 340 V, and so on. This can result in extremely high voltages being achieved in a very few amount of parts, and due to the diode configuration, a DC output. The first figure below is a simulation layout of my design, which shows the output reaching 1350 V DC in ideal settings. The following figure is an image fully constructed PCB which houses both the voltage multiplier as well as the control circuits. The voltage multiplier can be seen in the upper left corner, constructed with a maroon capacitors and one large yellow capacitor.

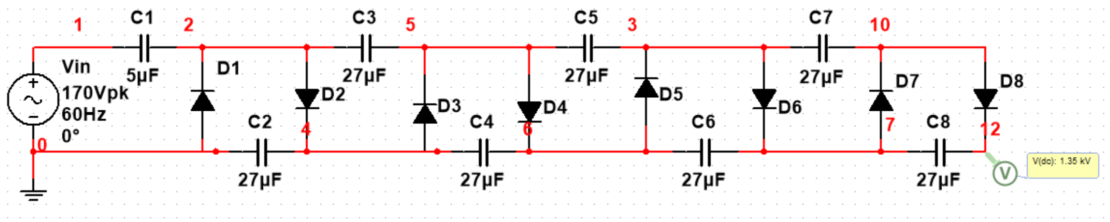


Figure 11: Voltage Multiplier Schematic

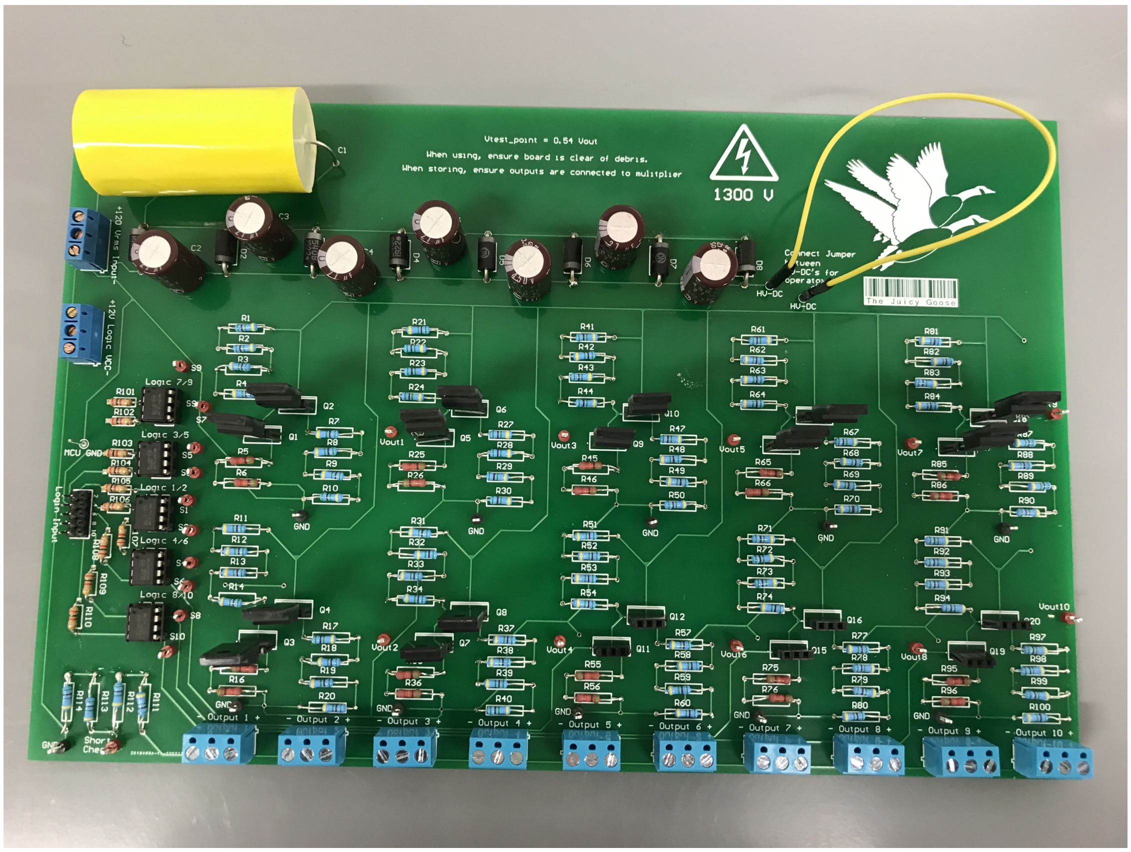


Figure 12: Photo of Subsystem PCB

Some of the immediate benefits of my design include cost efficiency and safety. Due to the charging characteristics of the capacitors, as previously explained, the voltage seen across any one component is only the peak value of the input voltage. This means that my parts only had to be rated for 170 V minimum. For safety and practicality reasons I chose ratings of 450 V for my components, which are exponentially cheaper then components rated for 1500 V, which is what would normally be needed to produce such a large voltage. The other benefit I mentioned is safety, which is a top priority for our entire project. Though the Cockcroft-Walton Multiplier has many commonly agreed advantages, there is one thing that most designers consider a drawback, that being the current supply limitations. Since the circuit consists of only capacitors and diodes, any current greater then a few milliamps causes the output voltage to "sag", or drop, quite dramatically. While this is bad for more power consuming designs, it is actually perfect for our low power solution. Our glove is designed so that the electrostatic breaks never make a connection, meaning that nearly no current should ever flow. Should any accidental shorts occur with the external environment, the low current dependencies of my multiplier will come into effect and the voltage will naturally drop, discharging the capacitors to nearly 0 V within moments.

3.2.2 Operation

Once the voltage multiplier is plugged into the wall outlet and outlet power is turned on using a switch, the final capacitor is charged to around 1305 V in under 3 seconds. Once this is complete the multiplier is ready to provide a high-voltage DC line to the rest of the system. Upon turning off the outlet voltage, if unloaded, the voltage multiplier will take around 24 hours to discharge to safe levels. When fully engaged with the 10 outputs though, the discharge only takes around 4 minutes before the voltage across the capacitors is no longer a concern.

3.2.3 Validation

The voltage multiplier was validated with numerous component and functionality tests. As the circuit was constructed, each diode and capacitor was measured before and after soldering in order to ensure there was no component damage. Each hole in the PCB was also checked to ensure continuity where it was meant to be, and an open circuit where it was not. Once the circuit was fully constructed, the 120 V input from the wall outlet was applied. The voltage on each of the capacitors was then measured in reference to system ground and recorded for error calculations. The multiplier performed as expected with a final loaded output of 1286 V, with non-idealities and a very slight current draw keeping it from achieving the full 1350 V as seen in simulations. This 4.7% difference is negligible, as the voltage dependant force feedback plateaus between 1200 V and 1400 V, meaning that all voltages in between result in the same amount of force applied with the braking system. The temperature of the system out-preformed expectations, with the temperature of any given component never exceeding 0.2 Celsius above ambient temperatures.

Another test that was performed was a short-test, in which the live output of the final capacitor was directly shorted to the system ground through a cable. This resulted in a momentary spark, followed by the charge on the capacitors dropping to nearly 0 V. The circuit was then powered back on to ensure functionality was maintained. Each time this test was performed, component values were also remeasured to confirm no damage was done. At the end of the testing no component received any measurable damage and provided validation that the system is satisfactorily robust.

3.3 Voltage Control Circuit

3.3.1 Design

The other challenge mentioned with this subsystem was the control method for the high DC voltage. Because the method of force feedback is voltage dependant, and not current dependant, a high side switch is absolutely necessary. A high side switch is a switch which is placed before the load in a circuit, meaning that no voltage is seen by the load until the switch is closed. This is in contrast to a low side switch, in which the load constantly sees the supply voltage regardless of the state of the switch, but no current passes through the load until the switch closes. This is especially important due to the very high voltage nature of the power supply. This proved to be quite difficult to design, since high side switching is done almost exclusively with PNP channel MOSFET transistors. These PNP transistors are nearly never manufactured to withstand greater then 600 V, and are exponentially more expensive then their NPN channel counterparts when achieving these higher voltage standards.

After days of constantly redesigning the control circuit, an affordable, effective schematic was achieved. This schematic uses 2 NPN channel MOSFET transistors to switch the 1305 V supply on and off. This is made possible by configuring the 2 transistors in cascade format. Firstly the gate of the output MOSFET is connected to the high voltage supply with a large resistance of 20.4 Mega-Ohms. The drain of the output MOSFET is then directly connected to the high voltage supply. The source of the output transistor is then connected to ground through another 20.4 Mega-Ohm resistor configuration. This results in the output transistor always being in the on state, allowing the voltage to pass through to the output. The second MOSFET, the control transistor, has its drain directly connected to the gate of the output transistor. The source of the control transistor is connected to ground through a 101 kilo-Ohm transistor configuration. The gate of the control transistor is connected to an inverting gate driver, which takes the logic level input from the micro-controller and translates it into voltage and current used to turn the control transistor on and off. The purpose for these resistor values is to limit the current flow to around 66 microamps at all times, orders of magnitude smaller then what the voltage multiplier can sustain so that there is safe, assured functionality. Also the 101 kilo-Ohm resistor chain is used to create a voltage divider. This divider drops the 1300 V keeping the output on down to 6 V, turning it off. The result is a switch that is naturally off while the gate driver is active, supplying no voltage to the output, but that can be turned on through the use of the control transistor. A schematic is provided below for reference, followed by a picture of the constructed circuit. This exact circuit can be seen replicated in Figure 11, in which all 10 of these output circuits are built into the same PCB as the voltage multiplier.

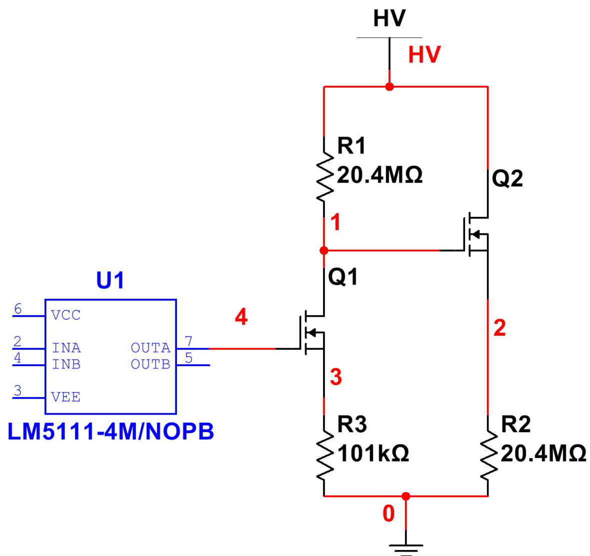


Figure 13: Control Circuit Schematic

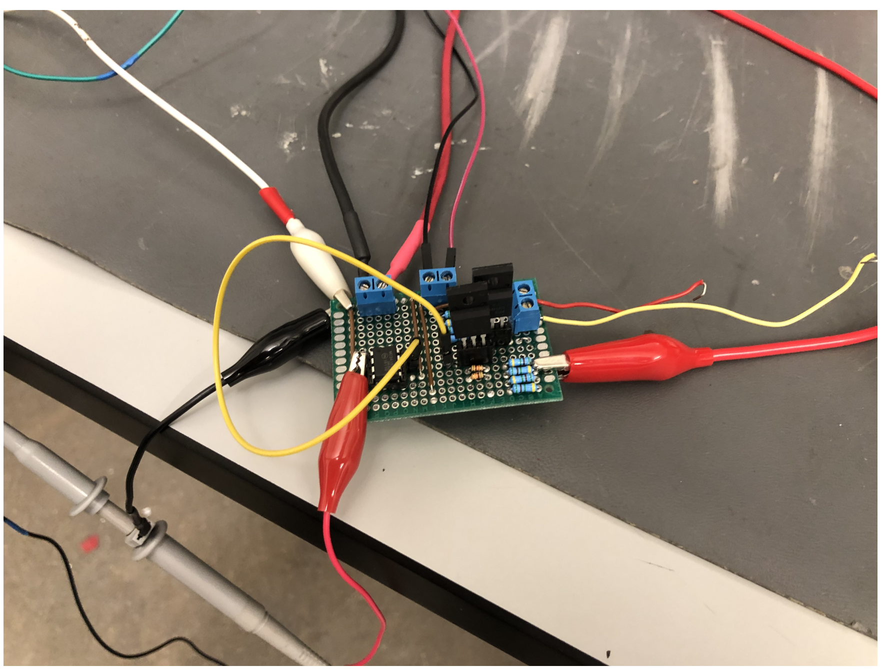


Figure 14: Photo of Constructed Control Circuit

One last challenge that presented itself with the control circuit during the latter half of our development was the fact that if a short were to occur across the output circuit, the power MOSFET would be overloaded by current, causing a burnout and the output to be forced high permanently. To combat this, a final 20.4 Mega-Ohm resistor chain was added across the collective ground point for the grounded electrostatic brake plate. This was to ensure that even in the event of a short between the two plates, the power outputs could still be controlled and deactivated.

3.3.2 Operation

In its natural state, the inverting gate driver supplies the voltage necessary to turn the control transistor on, allowing current to flow through the 20.4 Mega-Ohm and 101 kilo-Ohm resistances to ground. This acts as a resistor divider, lowering the gate voltage of the output transistor to around 6 V, below the 9 V turn on threshold voltage, resulting in nearly no voltages being seen on the output (approx 2 V). Once a turn on signal from the microcontroller is sent, the inverting gate driver turns off, which in turn turns off the control transistor. With the control transistor off the voltage at the gate of the output transistor is equal to the high voltage supply and the output voltage of the circuit is equal to the high voltage supply.

3.3.3 Validation

The validation for the control circuit once again consisted of multiple component and functionality tests. Each series of resistors was measured before and after soldering to confirm fidelity. The gate driver was initially validated separately, with each of its input and outputs tested with turn-on voltages ranging from 2.5 V to 5 V. The first of the functionality tests was turning on and off a lower voltage of around 25 V with a 3 V control signal which was given to the gate driver. Once this was confirmed the test was then repeated with the high DC voltage from the voltage multiplier, using the same 3 V control signal on the gate driver. All of these tests were successful, resulting in a fully functional control circuit capable of turning on and off the high voltage supply rapidly.

3.4 Voltage Control Software

3.4.1 Design

The method of control for the variable output voltages is standard pulse width modulation. This involves switching the MOSFETs on and off at a set frequency while adjusting the on time to produce a duty cycle ranging from 1 when the signal is always high and 0 when the signal is always low. This duty cycle inversely scales the output voltage, the formula for this can be seen below.

$$V_{out} = V_{in} * (1 - Duty)$$

The selection of the duty cycle is to be read from the data being received from the GUI subsystem, but during testing and validation this data was entered manually using the microcontroller programming software Energia.

3.4.2 Operation

The software is designed so the user may input a comma separated list of duty cycles in percentage. Using a 100 millisecond period this is then translated into on and off times for each individual control circuit. This was achieved using the Energia IDE for the MSP430 microcontroller. Using this period results in the micro-controller producing a signal at a frequency of 10 Hz and lets the user directly choose 10 duty cycles to achieve the corresponding output voltages. As previously mentioned these voltage selections will be automated by the GUI subsystem to apply variable force feedback to individual control circuits upon further improvement. The code that was developed is given here for reference.

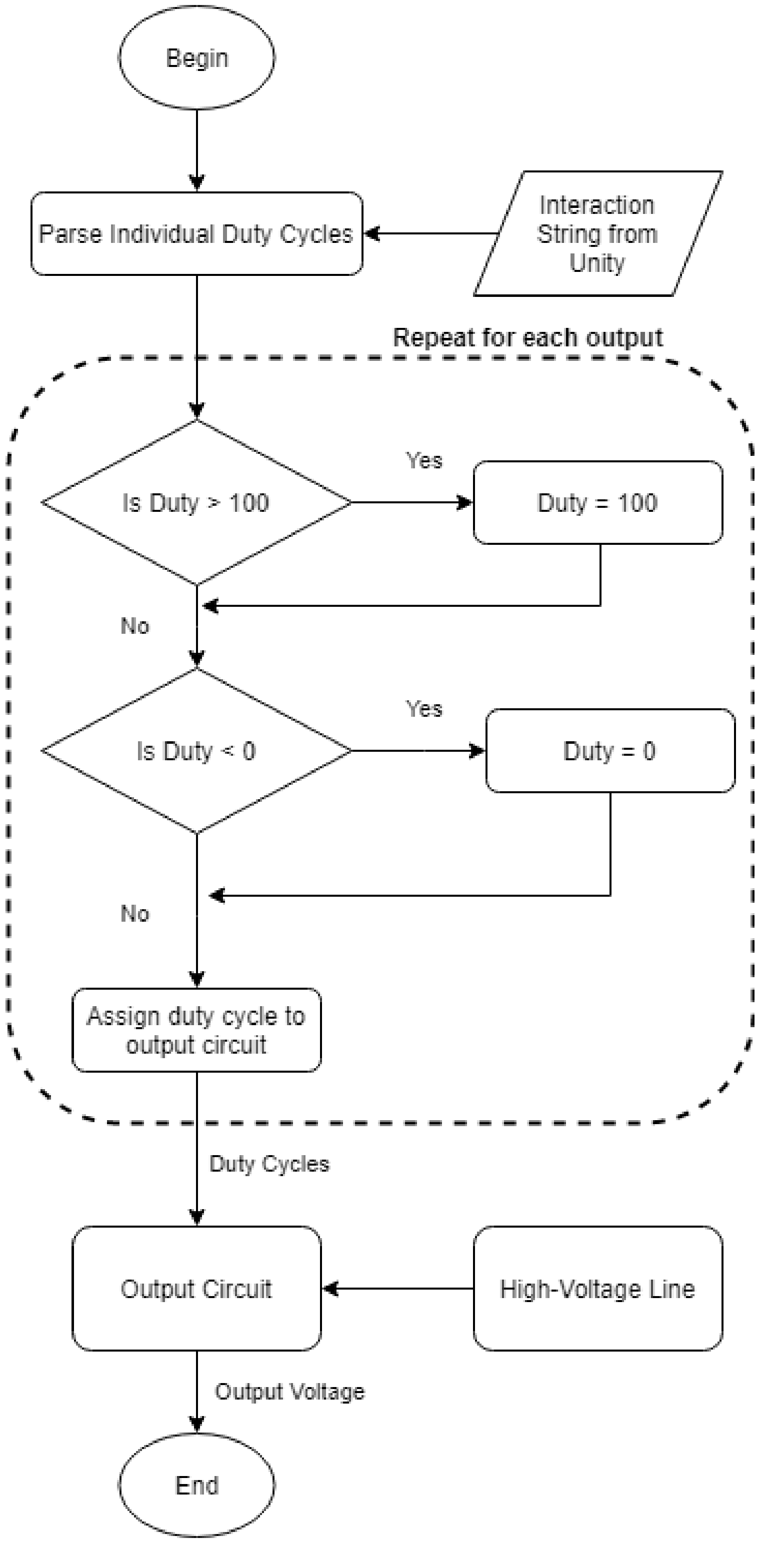


Figure 15: Control Software Flowchart

3.4.3 Validation

The validation for the control software was much simpler than for the hardware. An oscilloscope was used to probe the output pins of the MSP430, which provided the measurements for the signal amplitude, frequency, and duty cycle. I manually set each of the 10 output pins on the microcontroller to various duty cycles, incrementing from 0% to 100% in steps of 10%, measuring each of the aforementioned parameters at each pin. The amplitude remained a steady 3 V and the frequency was 10 Hz with a standard deviation of 0.2 Hz. The duty cycle was very accurate, always falling within 3% of the expected value. The next figure depects the actual vs expected duty cycle output for each duty setting.

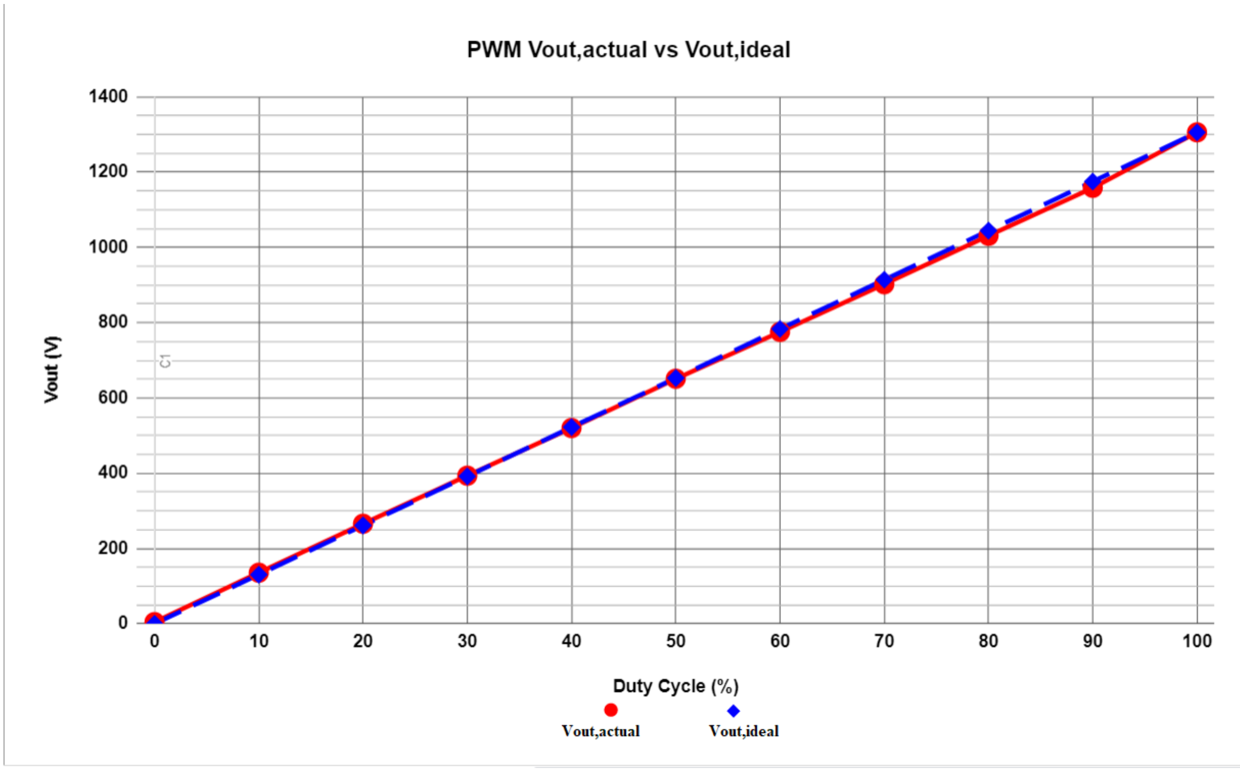


Figure 16: Duty Cycle vs Vout vs Vout,ideal

3.5 Total Subsystem Validation

Once all of the individual smaller systems had been completed and validated they were then combined to form the entire power subsystem. Power is supplied from a wall outlet and multiplied, which is connected to the control circuit, which is connected to the microcontroller. For validation, the circuit was powered on and the duty cycle was changed from 0% to 100% in increments of 10%. The average output voltage was then measured on each individual output, with at least 100 data points being taken for each duty cycle for the best overall average. The results matched the expected outputs with a largest error of 5%. A graph of the duty cycle vs output voltage is provided below.

On top of this the short between plates test was also re-ran. In this test the wire for the live plate was connected to the ground plate, simulating a worst case scenario short. In this scenario the MOSFET saving resistor chain designed into the control circuits preformed as expected and limited the current to 66 microampsp, allowing for the safe disconnection of the high voltage line. After the short tests were ran the initial functionality test was repeated to insure no damage had occurred. The subsystem preformed perfectly even after 10 shorting events, proving once again how robust it is.

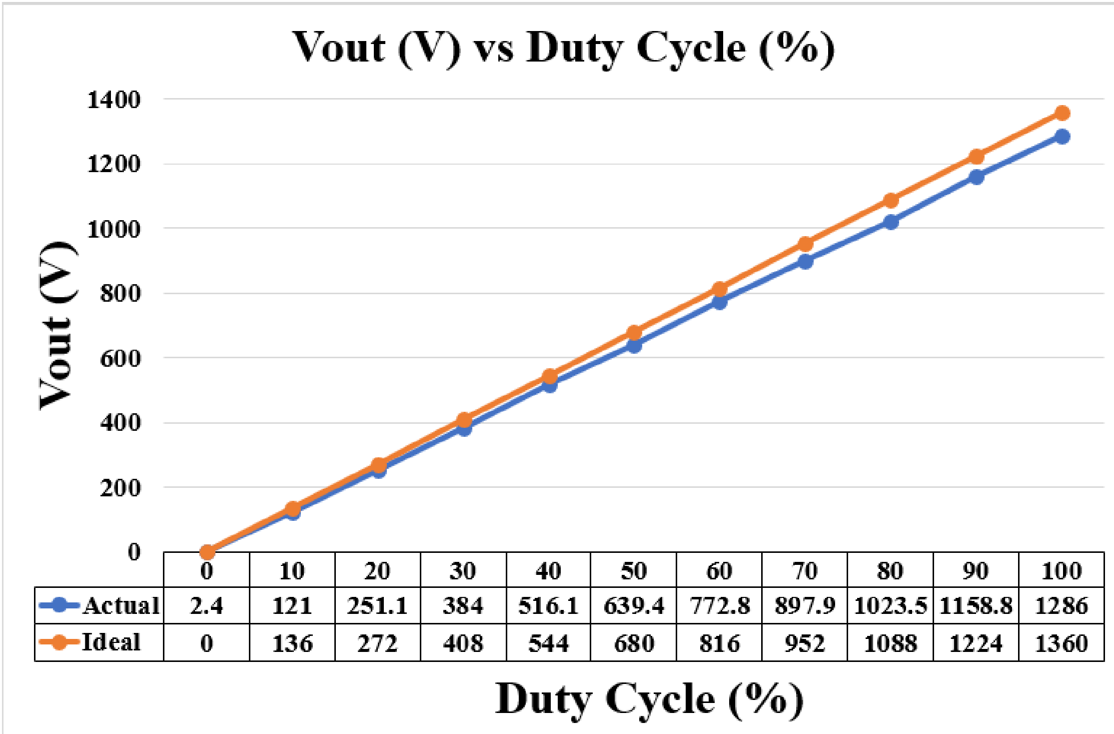


Figure 17: Vout,actual vs Vout,ideal

3.6 Subsystem Conclusion

The power supply subsystem exceeded expectations, providing a very low loss conversion from an AC input to a high voltage DC output and an easy to understand, accurate method of control in order to produce 10 very customized output voltages. The subsystem met all of the criteria set at the beginning of the design process and was built with the intent to make modular integration with any other subsystem very straightforward. The singular PCB allows for a very compact, portable design which removes any set up or tear down time when in use. Finally the program that was designed to allow user control is very modular as well, allowing for the possibility of future expansions with more or less outputs depending on usage case.

4 MCU and Hand Tracking Subsystem

4.1 Subsystem Introduction

The MCU and Hand Tracking subsystem is the initial step in bridging the gap between the digital and physical realms for the user. By tracking the movement and positioning of the hand, this information can be broadcast to the Graphical User Interface Subsystem by a serial communication between the MCU and the host device. The host device then sends information to the MCU of the Power Subsystem. Since the purpose of the subsystem is tracking of the hand, it is crucial that each sensor in the subsystem is operating correctly. Not only is it required to individually test each sensor to ensure it performs as the manufacturer described, but to also validate that the sensors are operating in specific boundaries for the purpose of tracking the user's hand.

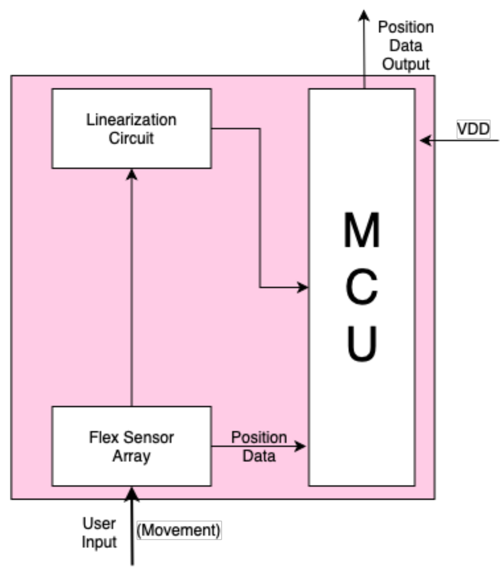


Figure 18: MCU Subsystem Diagram

4.2 Flex Sensor

4.2.1 Operation

For the movement of fingers, Sparkfun's 2.2" flex sensors were used. This sensor consists of conductive ink printed on a flexible base forming a resistor. As the sensor is bent (in the appropriate direction) the conductive layer is stretched, reducing the cross section. This increase in length and reduction of cross section causes the resistance of the sensor to increase.

4.2.2 Flex Sensor Hardware

Unfortunately, flex sensors operate on an exponential basis. Changes from $0°$ to $15°$ can result in an increase of $3,000\Omega$, while changes from $70°$ to $85°$ result in an increase of $11,000\Omega$ - shown in Figure 19.

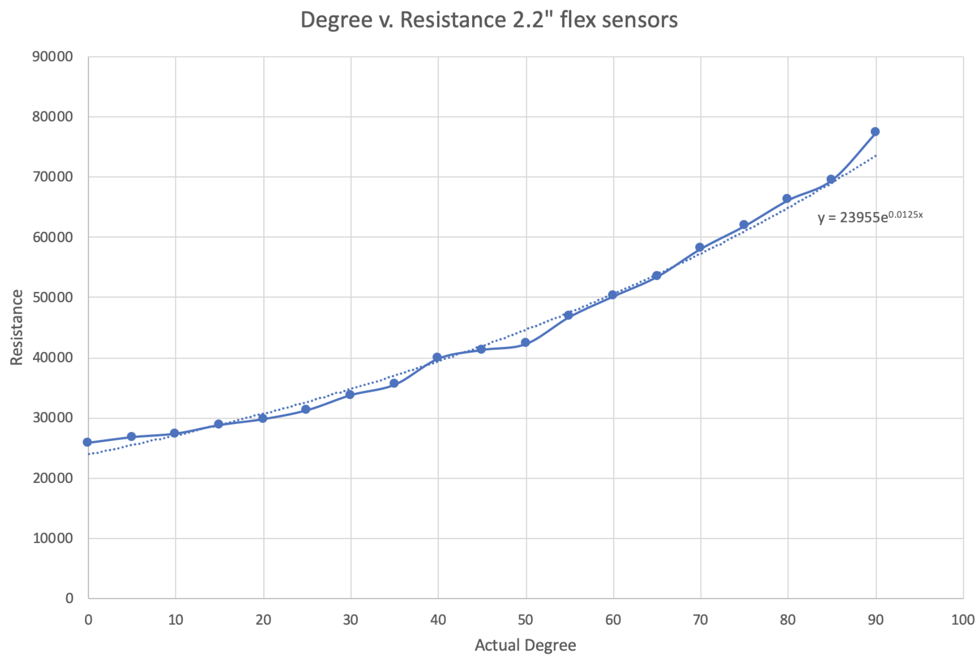


Figure 19: Degree v. Resistance for flex sensors

To linearize the voltage readings, the system needs a measuring resistor (R_m) in a voltage divider configuration and op-amp with negative feedback. The R_m needs to be chosen specifically for the sensors range of motion. For example, a flex sensor that is flat and then bent to 90° (mimicking the bending of a finger) will have a different R_m to a sensor between the fingers (ranging from 120° to 180°). The addition of the op-amp with negative feedback is so the data can be stable and further increase linearization of the voltage range from the voltage divider - shown in Figure 20.

$$V_{out} = \frac{V_{in}}{1 + \frac{R_m}{R_s}} \tag{2}$$

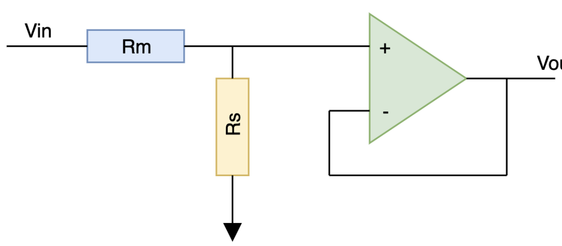


Figure 20: Voltage divider circuit with op-amp for linearization

However, to get the best (R_m) for that specific sensor we need to obtain the sensors lowest and highest resistance, which in this case its the amount of bending the sensor will experience. We then increase the R_m and find what voltage output it will have in the system with a $5V$ input. By doing this for each sensor, we guarantee that the sensor is in its ideal operating range. Figure 21 shows that for sensor 6, the ideal (R_m) is $54,000\Omega$. This resistor takes the highest voltage difference between lowest and highest resistance value from sensor 6.

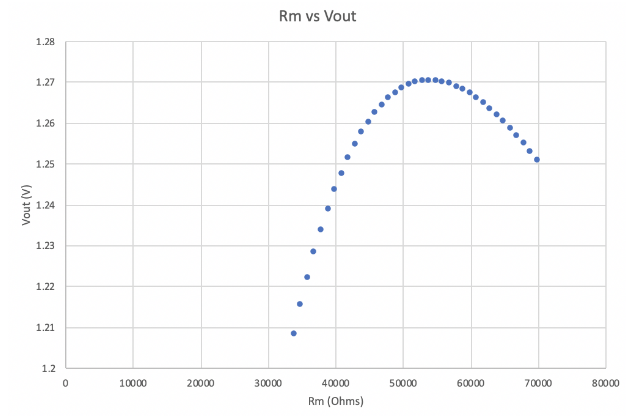


Figure 21: Rm vs. Vout

4.2.3 MCU Programming of Flex Sensors

The MSP432 was used as this subsystem's MCU because of its multiple analog pins and the ability to add booster packs that can double the analog pins if needed (i.e. another glove). The MCU reads the corresponding analog pins that the sensors are connected to. The pins return integer values from 0 to 1023, therefore a function was created to convert this reading to a voltage. Originally, the first 5 seconds of the program is the calibration portion in which the user flexes his/her hand so that the program can record the maximum and minimum voltages for each sensor. After the calibration, any movement by the user goes to a mapping function, specific to that sensor, and returns the degree of flexion of that individual joint. Currently however, we have the maximum and minimum thresholds for each sensor programmed in the computer. This allows us to control to a higher degree the voltage output and increases our accuracy of the sensor.

Since the data coming from the circuit of the flex sensor is already linear (from the linear circuit and R_m), a linear mapping function that outputs a degree given a certain voltage. The mapping is only for degree changes of 0° to 90° - for joints - and 120° to 180° - for spreading of fingers. knowing the initial and final possible degree, the code divides the difference between the user's maximum and minimum voltage change (for that specific flex sensor) by the amount of degree change on that finger. Thus, this function can accurately track changes of flexion.

After computing the corresponding degrees, the MCU outputs the information in a stream of comma-separated values for the user interface. The hand tracking MCU will transmit with a baud rate of 115,200 to ensure that every movement of the user is tracked and there is the least amount of latency.

4.2.4 Validation

The flex sensors were validated by placing each sensor on a hinge, increasing the angle by 5°, and comparing the actual angle to the output of the program.The recorded performance, as shown in Figure 22, is almost linear and under the ± 5° difference between actual angle and program output.

The flex sensors accuracy is quite impressive, but for us to reach this point we had to test multiple setups to guarantee the sensors are working optimally. The first design change came form the placement of the sensor between the fingers. Through validation, we realized that if the sensor was place directly between the fingers, the sensors readings will not be accurate (especially for the higher degrees). We then tried putting the sensor perpendicular to the two fingers and so the sensor would form an arc above the two fingers. Although this seemed to have fixed the issue, the placement of this sensor could possibly wear down since there is no guidance for the arc. The constant spreading of the fingers would ultimately wear the sensor down and it would break. This led to our final design for the sensor between fingers: place the sensor above the two fingers and parallel to the hand. The base of the sensor will be held in place by one of the gliders, while the tip of the sensor will be free to move along the other finger's glider. Additionally, between the two knuckles of the hand a $3D$ printed semicircle (with the curve facing the wrist) should be placed there and the middle of the sensor should be touching this curve. This gives the sensor guidance on where to bend and

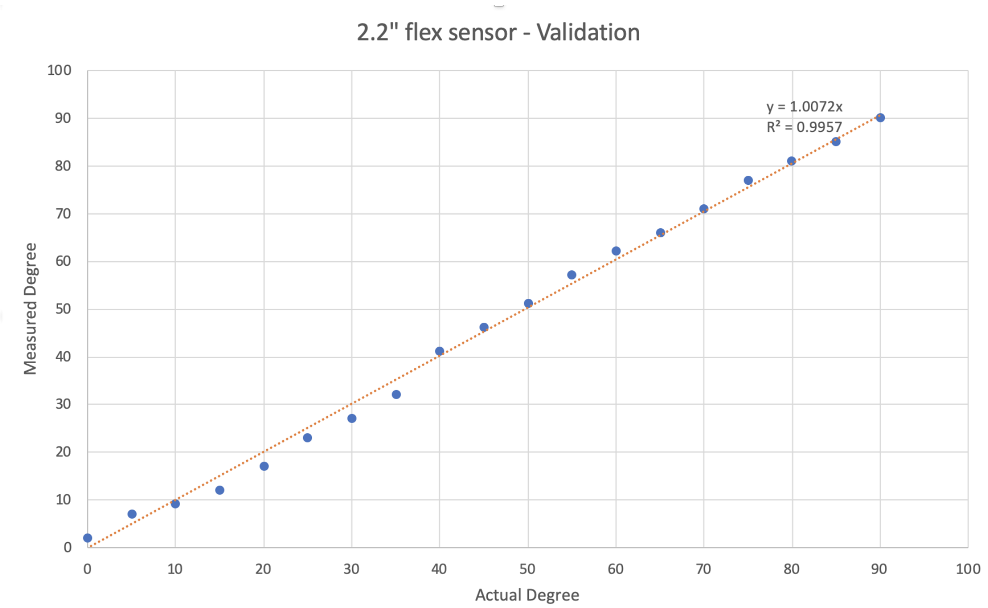


Figure 22: Flex sensor validation

ensures that the sensor will not break after repeated use.

The other design change comes from the placement of the sensor that maps the base of the thumb movement. Originally, we have planned on a sensor going across the palm. For quick reference, the movement I am referring to is when one tries to touch their thumb with their small finger. Notice the area of the palm that is bent. Unfortunately, the turning radius of this sensor exceeded what the manufacturer had designed. This caused sensors to break with only a few tests. To combat this issue, we had the sensor placed across the top of the hand going down to the side of the thumb. This still enables us to track that movement and it reduces the wear of the sensor. See Figure 23.

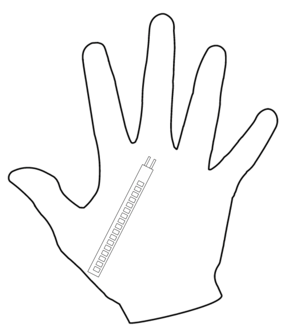


Figure 23: Placement of flex sensor

The last major key decision that was made was to join the gliders of the flex sensors to the gliders of the electrostatic brakes (for reference see Figure 27). By combining both gliders, not only do we reduce the size and the weight of the glove, but we also make it more robust and gives the user more freedom of movement. If the sensors were placed below the electrostatic brakes, then there was a higher chance of the ground plate touching the top of of the conductive ink from the sensor. When this happens, the sensor is shorted and no change in resistance is observed.

4.3 IMU Sensor

4.3.1 Operation

The IMU sensor is a critical component in determining the movement of the hand. The IMU used in this subsystem is the MPU6050, which is a $6-DOF$ IMU sensor (gyroscope and accelerometer). The IMU is used to measure the pitch, roll, and yaw of the hand. The sensor sends this information to the MCU via I^2C protocol.

4.3.2 MCU Programming of IMU Sensors

The sensor has its own corresponding library to interpret the data. Calibration of the sensor requires that the user puts his/her hand on a flat surface for 10 seconds. The library for the sensor includes a calibration portion as well, but every sensor has its own factory offset which needs to be compensated by the 10 second calibration test. After this initial calibration test, the user can move his/her hand freely and the program output will accurately display the changes of degrees.

4.3.3 Validation

Similar to the flex sensor validation, the IMU sensor was placed on a hinged platform. Here the platform is increasing in angle by 5° at a time and recording the actual degree of change to that of the program's output.

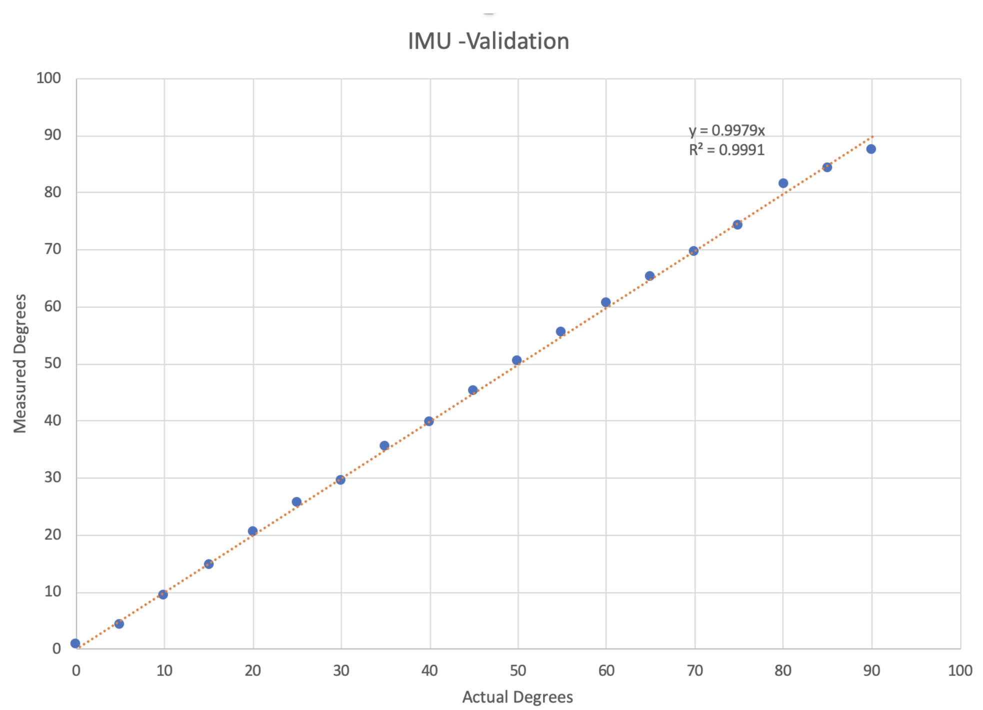


Figure 24: IMU validation

The performance of the IMU, shown in Figure 24, is astonishingly close to the real angle and under the ±2.5° difference. One thing to note is that when the sensor reaches 80° to 90°, the accuracy of the sensor decreases. Although this difference is less than 2.5°, the reason for this decrease in accuracy is because of the library coding. The library is coded so that as the values approach 90° it will begin to fall back down to 0° if the user continues to move in that direction. This means that if the IMU was flipped upside down, it will read 0° instead of 180°. Because of this coding scheme, the accuracy in the last 10° decreases to 1.5° off instead of the average 0.5°.

4.4 Hand Tracking Glove

Each flex sensor will be placed on every joint of the hand, between the fingers (to measure finger spreading), and on the palm of the hand (see Figure 23). Figure 25 is a picture of the glove with 5 flex sensors and the IMU.

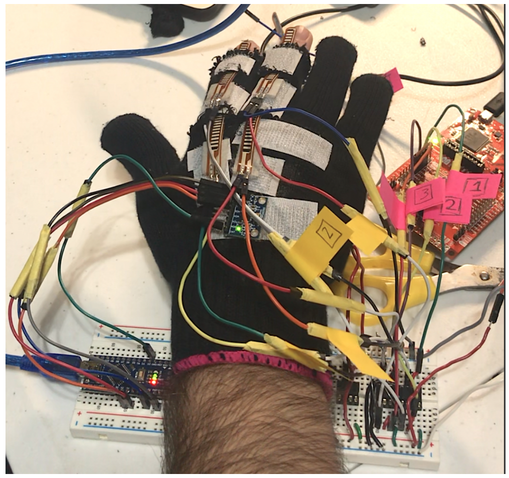


Figure 25: Glove with 5 flex sensors and IMU sensor

The figure shown above contains 2 flex sensors on the index finger, 2 sensors on the middle finger, 1 flex sensor between the index and middle finger, and the IMU sensor at the top of the hand. One thing to mention is that in Figure 25 one can see an Ardunio Nano being used for the IMU sensor. This was due to an error from the MSP432 code. After reading the library of the IMU and making the appropriate changes to the code, I was able to integrate the IMU to the MSP432.

Once we have verified that all the smaller systems in were working correctly, they were combined and scaled to form the entire hand tracking subsystem. To reduce size and make it easier to troubleshoot, we created a printed circuit board (PCB) that houses the hardware linerization circuits and created 3D printed sliders that mounts the flex sensors (see Figure 26 and 27). Movement from the user was accurately tracked (see section 4.2.4 and 4.3.3) and outputted from the MCU to the host device. For the future of this project, refer to section 4.5.

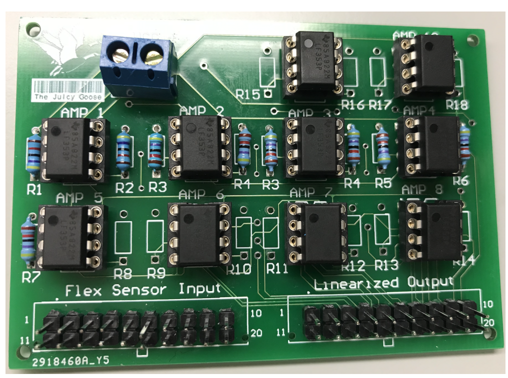


Figure 26: PCB of linearization circuit

4.5 Subsystem Conclusion

The hand tracking subsystem works as expected, providing reliable and accurate data for the rest of the system to interpret. The subsystem met all requirements set at the beginning of the design process, and was

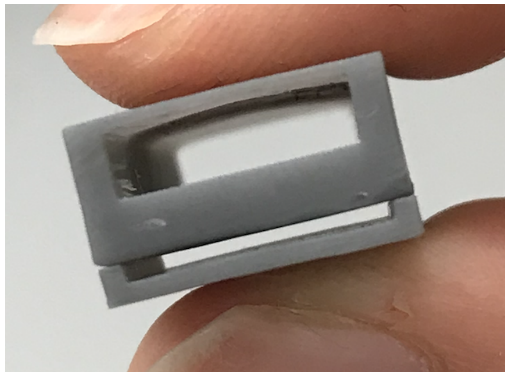


Figure 27: Gliders

built to make sure it was easy to understand and integrate with the other subsystems. The linearized flex sensors attained an accuracy of less than 5°, and the IMU successfully maintained the desired accuracy of less than 2.5°. Each of the sensors operate as expected. The use of 3D printed gliders with the PCB makes it easy for the user to setup and use. The program is straightforward and permits scaling, allowing future expansions to be added with ease. Although more testing needs to be done with all the sensors connected to the PCB, the subsystem works correctly and provides all the necessary information to support the other subsystems.

4.5.1 Key Learnings

Flex sensors are not as accurate as one expects them to be. But by researching how these sensors could provide accurate data, I learned how to linearize an exponential curve using hardware components. Additionally, learning how to use the MSP432 with the Energia IDE and libraries associated with the MPU6050, I learned how to program MCUs and use its various features to solve the task at hand. Apart from the physical components, I also learned how to communicate my ideas and present them in a proper manner, so the audience could understand the information I am conveying. The presentations helped me improve my public speaking abilities and working with a team on a large project made me realize the importance of communication.

4.5.2 Future

For the future of this project, connecting all of the sensors to the PCB and verifying that it outputs the correct information to the host device is crucial. We have tested all the connections from the PCB and ensured that it is correctly setup, we have calibrated all of the sensors with correct R_m resistor, but the last bit of testing is needed to create a full hand prototype.

5 Graphical User Interface Subsystem

5.1 Subsystem Introduction

This subsystem acts as the user's primary interface with the device by providing a GUI to send commands to the MCU. Even beyond this, however, the goal of this subsystem is to act as the bridge between the digital and physical realms for the user. By utilizing two-way serial communication with the MCU to visualize hand tracking data and send interaction events, we are able to achieve this. The subsystem makes use of the Unity engine for rendering a virtual environment in real-time and simulating convincing physics. The subsystem was tested and verified through usage tests, as well as a series of unit tests to simulate extreme or border cases. These helped to verify that each portion of the subsystem operated as expected within reasonable boundaries.

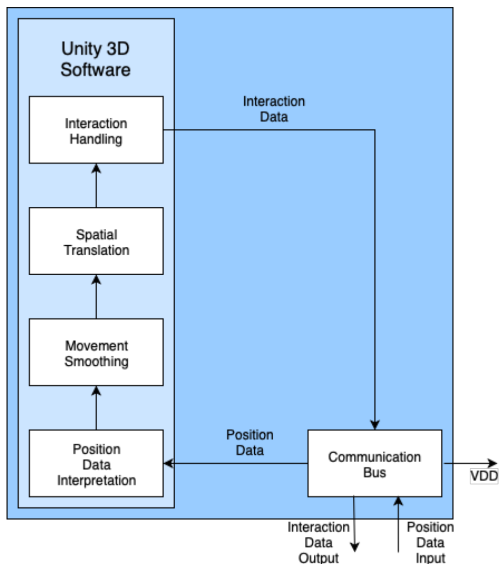


Figure 28: GUI Subsystem Diagram

Each subsection below describes the design of a specific feature of this subsystem, as well as its validation plan.

5.2 Virtual Hand

The foundation of this subsystem lies in rendering a believable recreation of physical reality for the user to interact with. In order to achieve this, we needed to create a virtual hand with which to interact with the virtual environment.

5.2.1 Model and Rig

The model and hand were found for free use online [1], but the bone rig needed to be redesigned to fit the needs of our system. Each finger bone was represented by a bone with inverse kinematic properties for movement constraints. Figure 29 demonstrates how by manipulating the bones in the hand rig, the hand model reflects the changes, allowing us to control the hand rendering via bone movements.

In order to optimize the physics calculations performed on the hand, the collision mesh is composed of a collection of capsules and boxes attached to different bones on the rig. The collision mesh allows Unity to simulate the physical presence of the hand in the virtual environment, and allows the virtual hand to interact with the world.

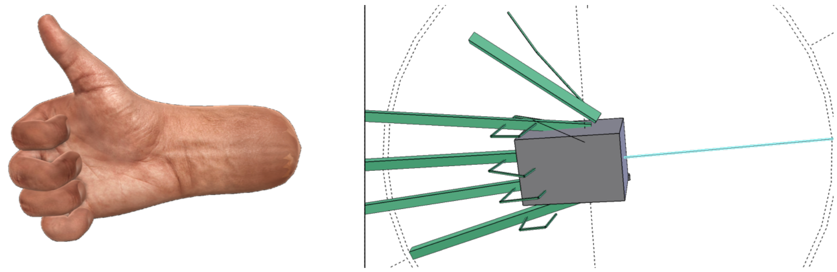


Figure 29: Comparison of the render and rig of a Gig 'Em

5.2.2 Kinematics

In order to appropriately control the virtual hand, the model was imported into Unity and outfitted with a ConfigureableJoint component for each of the joints tracked by the MCU. These were then programmatically loaded with configurable inverse kinematic boundaries and parameters in order to simulate realistic hand movements in the physics engine. Table 2 shows the approximated kinematic boundaries.

	X	Y	Z
Thumb Proximal	$-15°$ to $15°$	$0°$ to $90°$	$-10°$ to $10°$
Thumb Middle	Locked	$0°$ to $90°$	Locked
Thumb Distal	Locked	$0°$ to $90°$	Locked
Digit Proximal	Locked	$-5°$ to $90°$	$-10°$ to $10°$
Digit Middle	Locked	$0°$ to $90°$	Locked
Digit Distal	Locked	$0°$ to $90°$	Locked

Table 2: Inverse kinematic boundaries for each joint type on each rotational axis

5.2.3 Validation

For this feature, only the kinematics can be validated. 30 unit tests were written specifically to test the boundaries of each joint's rotation. In order to achieve this, a target rotation of the boundary max $\pm 5°$ was applied to the joint under test, testing the maximum and minimum boundary conditions respectively. If the rotation resolved to the kinematic boundary and not the target rotation, the test passed. 30/30 unit tests passed.

5.3 Spatial Translation

Spatial Translation refers to taking an input of rotation values for any joint and translating them into a combination of forces on the virtual hand joints discussed above.

5.3.1 Controller Scripts

In order to facilitate the appropriate movement of each finger, a series of controller scripts were utilized. Each script is designed to isolate only the methods related to that script's intended object to ensure complete validation. Figure 30 depicts the control tree layout used to control each joint's movement.

5.3.2 Mechanism

As mentioned in Section 5.2.2, each finger bone is outfitted with a ConfigureableJoint component in Unity. The ConfigureableJoint component is a generic joint component that allows for flexible scripting and configuration. In this case, the ConfigureableJoint component is important so that some of the joints can rotate on multiple axis, while still allowing us to apply forces to the joint in order to facilitate movement rather than directly changing the rotational value of each finger. This is important so that we avoid clipping,

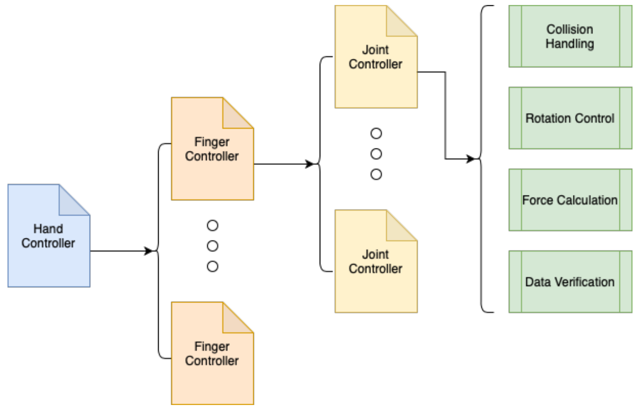


Figure 30: Flow diagram of controller scripts demonstrating modularity, readability, and ease of access

as well as simplifying the process of calculating the forces on each finger. It also utilizes the built-in physics engine, which means that we can tap into the Unity API for interacting with the virtual world.

Each joint has a linear multiplier parameter h that increases the amount of force applied to achieve the target rotational value. At each physics step, the controller sets the target angular velocity equal to $h \times \Delta R$, where ΔR is the difference between the target and actual rotational values of the joint. The value of h was chosen experimentally by testing different values of h and using the largest value that did not cause the feedback loop to become unstable.

5.3.3 Virtual Reality Tracking

One of the "stretch goals" put in place by our sponsor was to have the glove be integrated with an HTC VIVE Tracker to allow for free-space movement in a room using virtual reality. While we were unable to fully integrate into a scene due to an unforseen lack of access to development equipment, a VR scene integrating the VIVE Tracker was successfully created with a static prototype attached to it. Expanding this to include the hand prototype would be trivial, as the object with the rigged hand, collision boxes, interaction pads, and relevant scripts are all self-contained within Unity and would simply need to be attached to the VIVE Tracker object in a scene.

5.3.4 Validation

In order to validate the spatial translation feature, both observation tests and unit tests were applied. Observation tests consisted of creating a GUI that allows the user to choose any joint and manually set the target rotation using a slider. This validated that each joint was able to be individually and independently controlled using user input (which will eventually be communicated through the MCU).

In addition to the test environment for manual control, to test the glove using serial communication was also created. This environment allows for full functionality test of the glove, and is designed to be minimal. This is the environment used to validate the serial communication between the glove, the host, and the power controller.

The unit tests written for this feature are designed to test both the actual translation of the finger as well as the resolution speed and accuracy. To set the parameters for this test, we recorded video of a finger closing at 240 frames per second. By evaluating the number of frames it took to achieve a certain target rotation, we estimated that the virtual fingers should be able to resolve to a target rotation of 45° within 50ms and stay there for the following 5 frames. The fingers are given an error allowance of 3° because the way Unity calculates physics frames is asynchronous to real time, so at that speed the maximum offset from one physics frame would be approximately 3°.

For these unit tests, only 12/21 tests passed. Since is a stretch goal for this subsystem, this does not reflect the functionality of this feature. This is clearly not due to the operational capacity of the spatial translation by way of the observation tests, so the resolution rate could certainly be the issue. More likely, however, is that there was a design problem in the unit tests, causing primarily y-axis tests to fail. Further validation is needed in order to determine the root cause of this issue, but it does not appear to be a significant issue, as use tests showed the hand to be very responsive to user movement. This could be seen as becoming an issue later on when fully integrated with a virtual reality headset because too much latency can cause a disruption of body ownership, causing the user to feel disconnected with the device and break the immersion.

5.4 Interaction Handling

The key to achieving haptic feedback is knowing when and where to apply the feedback. These two variables are calculated and communicated in this subsystem using the systems shown below.

5.4.1 Interaction Pads

In order to better facilitate interaction detection, the virtual hand includes what we refer to as interaction pads. These, as the name implies, are pads on the fingers that trigger an event whenever they detect a collision. They only act as a trigger and do not affect the collision mesh of the hand. The left view on Figure 31 shows the interaction pads on the palmar side of the hand.

These interaction pads exploit two important features of our device.

- Haptic feedback is only available on the y-axis.
- The feedback force is passive, meaning that feedback only relies on magnitude, not direction.

The use of detection areas also function as an optimization tool. Instead of performing potentially hundreds of calculations per update frame to determine the average direction of the force based on normal vectors and overlap points, we can simply trigger an interaction event and focus processing power more on handling those events.

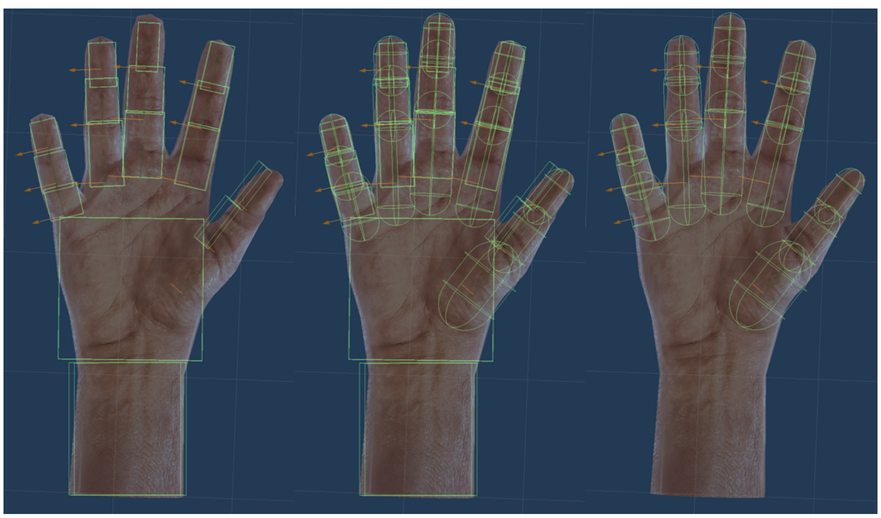


Figure 31: Three different views of colliders attached to the virtual hand

Left: Interaction pads (box colliders). **Center:** Both layers. **Right:** Collision mesh colliders (capsule colliders).

5.4.2 Force Calculations

Currently, our functional system requirements only call for three modes of feedback: Hard, Soft, and Off. In order to facilitate this, each object in the virtual environment is assigned a physic material with it's associated feedback "hardness". When an interaction event is triggered, the hardness of the material is transmitted as well, allowing us to translate this into a useful value for the MCU or for debugging. This appropriately fulfills these requirements, but is easily expanded upon in the future. For example, to have an

API for the glove to be used in other applications, it would be trivial to require a "hardness value" between 0.0 and 1.0. For simplicity and focus on design, however, the current two materials that exist are "rigid" and "soft", applying 100% and 50% feedback, respectively.

5.4.3 Validation

To validate the interaction handling feature of this subsystem, we utilized observation tests as well as 28 unit tests. The observation tests simply printed the name of the bone interacting with an object, the name of the interacting object, and the "hardness" of that object to the console.

The unit tests validated the specific function of each joint. Half of these unit tests validate the proper evocation of the interaction event on each interaction pad, and the other half validate the proper evocation of the disengagement event on each interaction pad when the interaction stops. This was performed by spawning an object with a given hardness in the trigger region and listening for the collision event or the disengagement event. 28/28 unit tests passed.

5.5 Serial Communication

Serial communication is necessary in order to send/receive data from the MCUs. Our system uses simple UART protocol to communicate between the external devices and the host device that is running this subsystem. An important note for this requirement is that synchronized read/write is unnecessary because the input/output data is independent.

A major change from the initial design of this portion of the project was the decision to allow the host device to communicate with both the hand tracking MCU and the power controller MCU rather than forcing communication between the two. This greatly simplified the process of writing code for the microcontrollers, but added some extra burden to the host computer. To handle this, there are now two different serial communication classes: one to receive and parse information from the hand tracking system, and one to transmit to the power controller. Each is included as a script on an empty SerialManager object in the scene with links to the hand from which to send/receive data.

5.5.1 Hand Tracking Controller

This controller makes the connection with the hand tracking MCU to receive a constant stream of input data. As of right now, serial port is manually inserted, but it would be trivial to expand to an API to add a dropdown menu to the Unity inspector window for developers to select which serial port the glove is being controlled from. Figure 33 demonstrates the flow of the serial communication class for hand tracking. The controller sends data to the HandController assuming the communication protocol layed out in the ICD between the hand tracking MCU and the host computer. This also means, however, that it would be trivial to reorganize transmission to allow for expansion or change of protocol. Right now, the controller is set to begin when initialized and close the stream whenever the simulation is closed.

5.5.2 Interaction Controller

The interaction serial controller handles data transmission to the power controller. This includes constructing the output string and transmitting it via UART. This process is fairly simple; there is no handshaking or other formalities between the systems to worry about. Constructing the command string consists of a simple for-loop representing the force for each string as described in the ICD. Again, the serial port is manually saved, but could easily be detached to allow for better API inclusion.

5.5.3 Validation

Before we were ready to integrate, serial communication was validated using an Arduino setup to communicate both ways. Once we were ready to integrate some of our prototype pieces, the system was further validated using several manual tests. In order to facilitate this, both systems were connected separately. The hand tracking subsystem was manually validated using the following measures:

- Sending several command strings and comparing the interpreted input to the actual input.

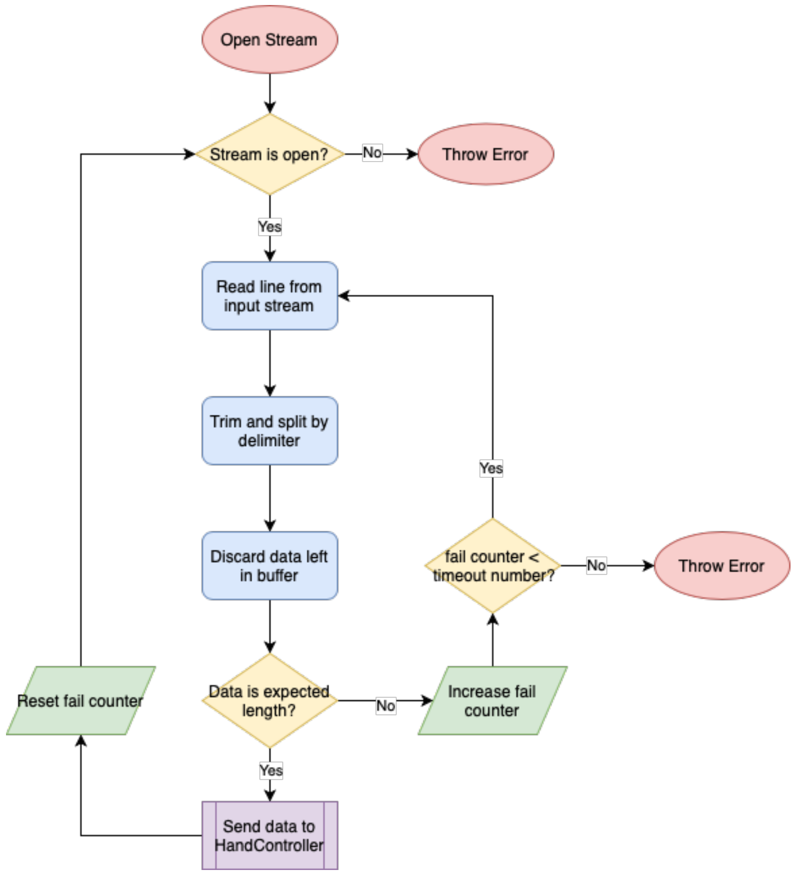


Figure 33: Flow Diagram of data flow for hand tracking serial class

- Sending command strings with incorrect number of data points.
- Checking sent command strings against final hand positioning.
- Fully integrating hand tracking prototype and visually comparing the positioning of the hand to the positioning of the glove.

Thankfully, the input data validation is performed once it arrives at the HandController, which is validated separately in Section 5.3.4.

To validate the power controller subsystem, really the only validation that needs to occur is

- Sending several command strings and comparing the interpreted input to the actual input.
- Correct command strings are sent

There is no data verification to be done; the command string pulls directly from the HandController, which is validated separately.

5.6 Subsystem Conclusion

This subsystem operates as expected and satisfies all relevant functional system requirements. Integration with other subsystems was seamless and operated exactly as expected, requiring no further tweaking of the original design. Virtual environment dynamics, including a fully-rigged virtual hand Unity prefab, are all in place and operate as expected. Communication between the hand tracking MCU and the host computer has successfully been established, and the incoming movement data is correctly translated into the rotation of individual joints on the virtual hand. Interactions in the virtual environment are successfully being captured and transmitted to the power source controller using the protocol described in the ICD.

5.6.1 Validation Summary

Table 3 below shows a summary of the validation performed to date on this subsystem. As you can see, the most crucial tests are either passed or pending. Pending tests required hardware to be assembled in order for validation tests to be run.

Test	Pass/Fail	Test Type
Interaction Boundaries	Pass	Unit Test
Interaction Logic	Pass	Manual
Interaction Output	Pending	N/A
Movement Boundaries	Pass	Unit Test
Movement Logic	Pass	Observation
Movement Parsing	Pending	N/A
Serial Transmission	Pass	Manual
Serial Reception	Pass	Manual
Joint Resolution Speed	Mixed	Unit Test

Table 3: Summary of GUI Subsystem Validation

Note that the Joint Resolution Speed was intended to be a measure of how fast the virtual finger can resolve to the correct rotational value. This was supposed to be tested by assessing whether the joint could rotate half of its maximum rotational value within 5 frames of 100ms. Since the joints were all tuned the same but had mixed results, I suspect that there was a problem with the unit test written, but validating it was not high priority since the resolution speed was not defined as a functional system requirement.

5.6.2 Key Learnings and Decisions

Ultimately, in developing the visualization of the hardware of this system, there were several key decisions and changes that were made as a result of learned information. For example, the decision to utilize primitive collision boxes came as a result of learning more about the processing of 3D meshes and how computationally intensive using a complex mesh for collisions would be. Making use of the various limitations of our system to save computational power was a huge learning curve, but ultimately greatly enhanced the design in the end. Since Unity primarily uses C# for scripts, that meant learning a new programming language and expanding my list of known languages. The most obvious learning curve was to interact with hardware, both digital and analog, using software; a really valuable skill that will undoubtedly prove useful in the future.

5.6.3 Future

Unfortunately, due to unexpected circumstances, the project was forced to a premature ending. This meant that several system-wide tweaks were left undone. Overall, this subsystem is almost completely finished, but since software is one of the last systems to be tweaked, this means there are a few tasks left to be done, namely:

- Validation of joint rotation delta between physical and virtual hands
- Gathering data on latency between glove and host computer
- Digital calibration and setup
- User experience testing

Though there were several declared goals that were left after our premature finish, there were several stretch goals that would be easily achievable in the future if someone were to pick up this project. Some stretch goals and potential future improvements might be,

- Full VR integration

- Haptic Feedback Glove API
- More polished UI and hand model
- Digital movement smoothing and efficiency improvements
- Normalize joint resolution speeds

HAPTIC FEEDBACK GLOVE

Jonathan Arze
Luciano Brignone
Taylor Gage
Nicole Khoury

FINAL SYSTEM REPORT

REVISION – Draft
04/25/20

FINAL REPORT
FOR
Haptic Feedback Glove

TEAM ⟨40⟩

APPROVED BY:

Jonathan Arze Date

John Lusher II, P.E. Date

Andrew Miller Date

Revision History

Revision	**Date**	**Author(s)**	**Description**
Draft	04/25/20	JA	Created

Contents

List of Tables

List of Figures

1 Overview

The goal of this project is to create a Haptic Feedback Glove that combines the dexterity and force feedback of current technology. Users will experience further immersion into a virtual environment with variable force feedback, allowing them to distinguish between objects of different materials. Figure 1 below shows the high-level flowchart of information in this project.

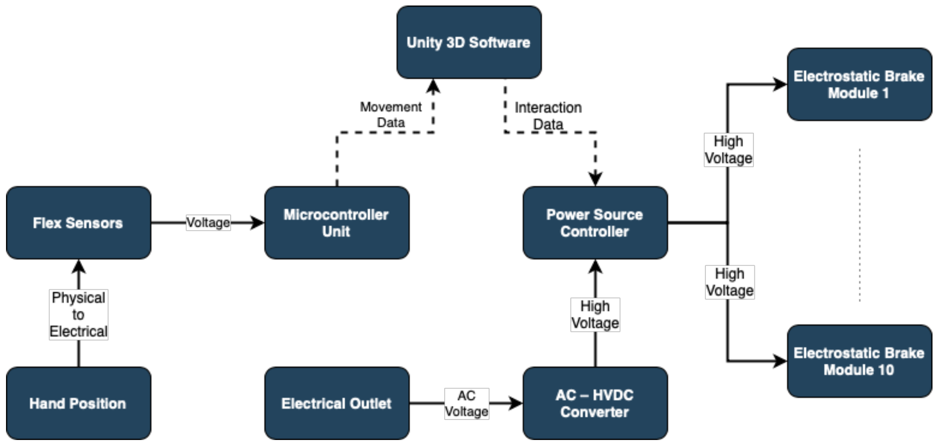


Figure 1: High-Level Flowchart of the Haptic Feedback Glove

The Haptic Feedback Glove will be able to track the user's hand and finger movements in order to represent the hand in the virtual environment, using the spatial data to communicate about object interactions. Applying a suggestive force to the user's finger joints will allow them to experience tactile feedback in the environment, further elevating their experience.

2 Development Plan and Execution

As described in the overview, this system is easy to break down into four separate subsystems: the hand tracking, the host computer software, the power converter, and the force feedback electrostatic braking modules. During the first half of this capstone project, our team focused on individually-assigned subsystems and validating their performance. The second half, however, was focused on integrating these subsystems into one, complete subsystem. The sections below will explore in more detail the development plan used to create this system, and will primarily focus on the second half of the capstone project.

2.1 Design Plan

The first challenge in imagining a design for a haptic feedback glove was researching what is currently available on the market, exploring what research is being done, and deciding what the focus of our device would be. Having seen several large, unwieldy devices, our team decided to pursue a previously experimental route with a focus on maintaining the dexterity of the hand. This meant that our solution had to be lightweight and natural-feeling, yet still provide a firm force feedback against the fingers to indicate to the user when they have interacted with an object in a space they do not occupy. The research that most closely resembled these criteria was out of EPFL and ETH Zurich utilizing electrostatic brakes placed on the hand using frictional force to apply force feedback to each finger.

The four subsystems described above are obvious choices for this system. The biggest challenge would be designing this system while still remaining within our moderate budget. Flex resistors were chosen to track hand movement because they provided enough accuracy for our needs while minimizing mechanical components. To create a large voltage potential needed to operate the plates, we needed something that was expandable but cost efficient. After looking at several designs, we chose to use a Cockcroft-Walton Voltage Multiplier to create the high voltage, then utilize several small switching circuits to control the potential difference on each finger line. For the electrostatic brakes, we wanted to minimize size and rigidity, but maximize capacitance (to convert voltage potential into frictional force). Ultimately, after several design iterations, we landed on using a double plate capacitor with a layer of wax paper as a dielectric. We would later find that the electric field breakdown of wax paper was not high enough, so we were forced to sacrifice some capacitance for another layer of wax paper. Lastly, our biggest design choice with the software was the request from our sponsor to make it compatible with Unity and SteamVR 2.0. With these choices made, the design plan was fairly straightforward: each subsystem would have a prototype built demonstrating the function of one or more modules.

2.2 Execution

Each subsystem is modular, so can be scaled to as many inputs and outputs as we need. Because of this, once one module of each subsystem has been designed and validated, the only challenge to scaling is creating more and repeating validation. It also means that we could begin integration before scaling the full system because once full integration with a few modules was validated, the system was functionally guaranteed to work at full integration with all modules. In terms of full system integration, our execution plan followed the schedule shown in Figure 2. Unfortunately, execution was interrupted by unforseeable circumstances, as displayed by the incomplete plan.

	Feb 5	Feb 19	Feb 25 (checkpoint)	March 4	March 11	March 18	April 2 (Demo)
Electrostatic Braking Modules	Eliminate voltage arcing; increase force feedback	Iterate design for space efficiency; integrate with power system	Functional Prototype	Integrate onto glove	Reevaluation and Iterative Development	Safety improvements and final integration	Measure final forces
Power Supply	Design PCB	Order PCB/ start Host protocols		Validate PCB/Host interactions		Final integration, safety improvements, debug code	Wrap up validation of specs
Hand Tracking and MCU	Design Exoskeleton, Order new parts	Coms with host computer, calibrate sensors		Print exoskeleton, full integration, PCB		Reprint glove modifications, final mounting	Polish design, finalize validation
Graphical User Interface	Serial communication interface, map data to hand	Create and clean interaction logic, thumb and finger spread		Integrate into VR		Polish demo environment, debug, measure final latency	Usability testing and immersion observations

Figure 2: Spring 2020 Execution Plan

During integration, there were a few design problems that had to be fixed, such as voltage arcing through the dielectric as described before. Thankfully, the time to fix issues such as this were incorporated into the execution plan, so we were not set behind. Much of the integration lay in writing new code for the MCUs to handle serial communication for the bulk of the control layers. Hand tracking integrated seamlessly with the Unity software, but the power controller encountered a few bugs that were unable to be solved before the end of our design window. The power converter and electrostatic brakes integrated with no problems as well. New mounts were designed and printed for integrating the flex sensors and electrostatic brakes onto the glove. Meanwhile, we had PCBs printed to facilitate the expansion of power and hand tracking modules.

Had the project not come to a premature halt, the following steps would have been to finish construction of the glove, finish debugging the host to power controller software, and make minor tweaks as necessary. Again, since this system is so modular, having a few modules successfully integrate, we expected that building out the remaining modules would add no new problems to the system, despite budgeting time for reevaluation and iterative development along the way.

2.3 Validation Plan

Unfortunately, the end of our development time occurred just before we were able to completely integrate our system. Therefore, the validation plan the partially-integrated system is shown below. The Haptic Feedback Glove was tested for operation as a hand tracking system, as well as a multi-output force feedback system. The system functionalities that were able to be validated include:

- hand tracking,
- interaction detection,
- communication between host computer and the microcontrollers,
- outputting different force feedback levels from all 10 outputs.

3 Critical System Data

3.1 Electrostatic Brakes Data

The electrostatic brakes work together with other subsystems to provide the tactile feedback to the user. This is achieved by increasing or decreasing the voltage to each module. Varying voltages simulate interactions with objects of different materials. The table below shows different design iterations and their corresponding capacitance.

Design	Measured Capacitance
Double Plate Capacitor (Wax Paper)	0.931 nF
Cylindrical Capacitor (Wax Paper)	0.534 nF
Cylindrical Capacitor (Plastic Film)	0.322 nF
Double Plate Capacitor (x2 Wax Paper)	0.319 nF
Original Parallel Plate Capacitor (Polyimide)	0.153 nF

Table 2: Design and Measured Capacitance

Through trial and error, we concluded that the double plate capacitor wrapped twice with wax paper yielded the best results. Figure 3 shows the force produced by varying the voltage levels.

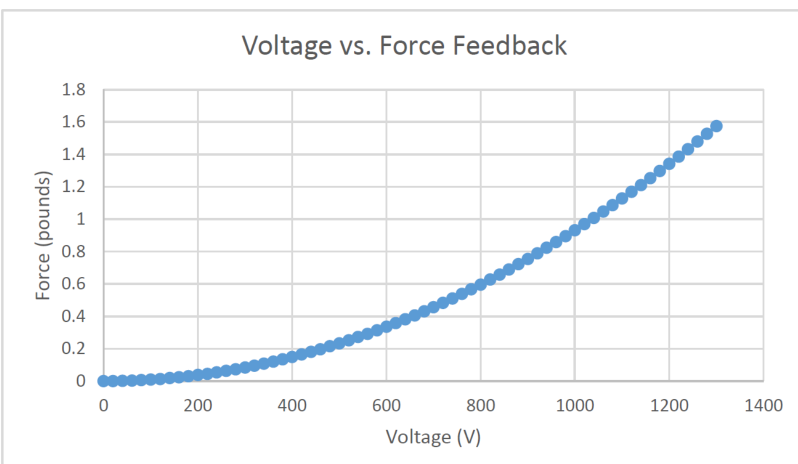


Figure 3: Force Feedback

Due the current global situation, we were unable to further test the forces produced. The graph shown is the expected results from the electrostatic brakes. This data is important because it shows that based on the placement and area of overlapping material, we are still able to generate a respectable amount of force feedback.

3.2 Power Supply Data

The power data is responsible for taking the digital force interaction output data from the GUI subsystem and translating it into an analog voltage. Using this data, 10 outputs are individually controlled to each produce a chosen voltage anywhere from 0 volts to 1300 volts, approximately. Figure 4 depicts that by increasing the duty cycle, the output voltage is increased linearly. This data is crucial so that the electrostatic brakes are able to apply the correct amount of force to the user.

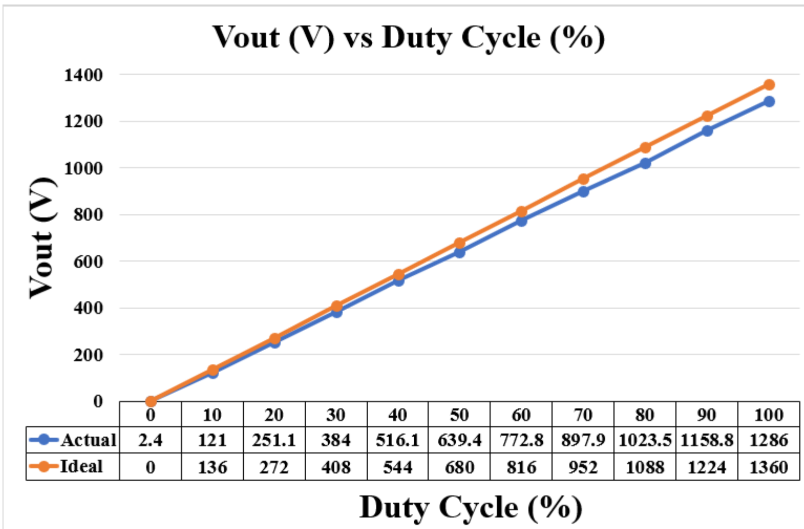


Figure 4: Vout,actual vs Vout,ideal

At 100% duty cycle, we are outputting $1286V$, meaning the system can supply the necessary voltage to the electrostatic brakes. Additionally, at each 10% increase, the voltage output is extremely close to what the ideal output should be, verifying the robustness and accuracy of the power supply.

3.3 Sensor Data

Flex sensors and an IMU sensor work hand-by-hand to provide the system with user data. These sensors are read at a baud rate of 115,200 and the data is sent to an MCU that then passes the corresponding degree of change to the host device. Figure 5 shows how the increasing the voltage (bending of finger) increases the degree output of the program. The data also shows how linear the circuit is, therefore the mapping from voltage to degree is very accurate.

We need the data from the user to be as accurate as possible. If the system were not as accurate as it is, it would be difficult for the user to completely immerse with the virtual environment.

3.4 GUI Data

During the operation mode of the system, the GUI is receiving data from the sensors and outputting data to the power system. The subsystem makes use of the Unity engine for rendering a virtual environment in real-time and simulating convincing physics. As data is received, the data is visualized in real-time in the virtual environment. If the user comes in contact with an object in in the virtual environment, triggered by the virtual pads on the fingers, in relays this information to the power MCU. The data that is sent contains 10 comma-separated integers representing the percentage of force being applied to each point of contact. For instance, if the user grabs a ball with his/her middle finger and thumb, then the command string following

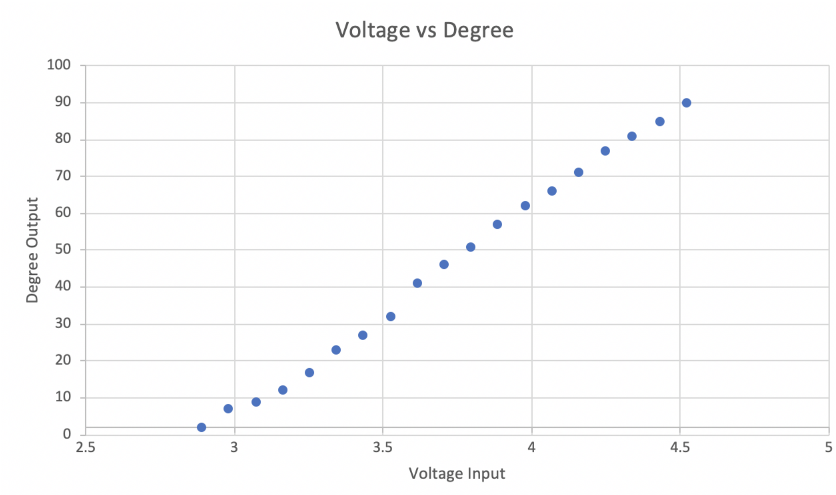


Figure 5: Voltage vs Degree

this would be

$$60, 50, 0, 0, 60, 60, 0, 0, 0, 0$$

Furthermore, if we were able to further validate the integration between the hand tracking data and the GUI, we could determine any latency between the glove and the virtual hand and the degree of accuracy the virtual finger is displaying versus how much is truly being bent.

4 Conclusion

The Haptic Feedback Glove demonstrated great potential to overtake current virtual environment control methods in terms of immersion and accuracy. From design to partial execution and validation, the project was taken from concept to reality in a matter of months.

4.1 Key Decisions

The majority of the design choices were made at the subsystem levels by their respective engineers. Choices such as the overall functionality, and extra safety design considerations, on the other hand, were made by the team as a whole. Starting with the functionality, we had to decide what level of force feedback would be appropriate, as well as the best method of application for this force. We eventually decided on a double-sided electrostatic braking mechanism applied to the second joint of each finger as well as the fingertip in order to best maintain our desired force feedback while also minimizing the size of the glove. Secondly, with safety considerations, we had multiple rounds of design iterations and discussions about how best to apply up to 1300 V so closely to the human hand without putting the user in danger. Eventually, we decided to have multiple emergency shutoff switches across the power system and even in the software. These included a switch on the power outlet feeding the power system, a user input to deactivate the glove, and an isolated wire that could be removed to force a shutdown.

4.2 Learnings

Throughout the two semesters that we have had senior design, our team has grown greatly in both skill and confidence as engineers. From technical designs and project management to teamwork and conflict resolution, every aspect of our work ethic was in some way improved.

With respect to technical "hard" skills, many tools were learned and utilized in order to create our project. Many members of the team became much more fluent in various coding software, including Unity for a very fleshed out and intuitive user interface and Energia for microcontroller protocols. Beyond this, development software such as Altium and various 3-D printing tools were crucial in developing the compact PCBs and component housings that were required to create the glove. Moving to hardware, things such as linearization circuits and intricate sliding mechanisms were developed to produce very precise and robust measurements for finger tracking. Even the fundamentals of power system design and capacitance force calculations were greatly improved throughout our team, as was required to produce the very effective high-voltage, low-current power supply and the elegantly designed electrostatic brakes. These were not the only type of skills that saw improvement though, as the "soft" skills of our team were bolstered as well.

The main change in this regard was our confidence in public speaking. Though it is usually a daunting task for students such as ourselves to present our work in front of people who often have years of experience over us, the environment of senior design was such that we never felt scared to share our successes, as well as our failures. We learned to answer questions and take criticisms with composure and a mindset of improvement, rather than defensiveness. All of these things further contributed to the improvement of our abilities to coherently and precisely describe designs, events, and data in front of our peers.

4.3 Future Work

4.3.1 To Be Done

The spread of COVID-19 and the subsequent statewide quarantine resulted in our team not being able to completely build and validate our project. Though this was outside of our control, we were very much on track to have our completed product in the following weeks, which would have consisted of data collection across the project. The power subsystem and electrostatic brake subsystem had been completely integrated, and quantitative testing was next on the agenda in order to record the forces our system was capable of applying. Moving forward the exact voltage to force feedback curve would need to be measured systematically and validated to meet our specifications as laid out in the FSR. The hand tracking subsystem and the GUI subsystem were also integrated, with hard data such as packet drop rate, update speed, and exact precision still needing to be collected.

The gap between the GUI and the power subsystem is currently purely software bugs. The MSP430 microcontroller used by the power system currently handles manually input strings of duty cycles to translate to voltage outputs, but stalls when trying to read in data through the USB port from the host computer. This is the only barrier to full system integration. Moving forward this would need to be debugged and validated to work between various computers, and a full system validation would need to take place.

4.3.2 Possible Development

There are a few things that could be considered to better build upon the electrostatic brake glove design. These include increasing the size of the brake strips in order to translate more force per voltage, which would in turn allow for a lesser voltage to be generated from the power supply. Another possible development would be to implement a fuse system at each output in order to maximize safety measurements, though this would become very expensive very fast at such high operating voltages. A final development that could be foreseeable would the the application of Bluetooth or WiFi communication to allow for a less tethered experience. This would remove the connection between the user and their computer, but the limiting factor would be the length of the power supply cables, which must use a stationary wall outlet for operation.