Haptic Feedback Glove

Benjamin Tures
Nicholas Minton

CONCEPT OF OPERATIONS

REVISION – 1
19 September 2020

CONCEPT OF OPERATIONS
FOR
Haptic Feedback Glove

TEAM 64

APPROVED BY:

Nicholas Minton 9/19/2020

Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	9/6/2020	Nicholas Minton	BT IG NM	Initial ConOps
1	9/19/2020	Nicholas Minton		Revision based on feedback

Table of Contents

List of Tables

No table of figures entries found.

List of Figures

1. Executive Summary

Recent advancements in virtual reality are tightening the gap between virtual and physical perception. One area with the potential to strengthen this connection is haptic feedback technology. To capitalize on the improvements of graphics, there is a need for more immersive ways to interact with the virtual environment. Haptic feedback technology aims to mimic forces felt when interacting with physical objects. The goal of this project is to give the hand realistic feedback by allowing for greater depth while interacting with the virtual space.

2. Introduction

2.1. Background

The origins of haptic feedback are rooted in the goal of allowing remote operation and handling of objects. This is accomplished by using the sense of touch as a way to transmit information. This is also known as kinesthetic feedback. Remote operation had an early application in the handling of radioactive materials while keeping humans at a safe distance during the aftermath of WWII [1]. Kinesthetic feedback is most commonly found in controllers. One example is a vibration in a video game controller indicating a player has crashed into a wall in a driving game. A more practical example is the joystick of aircraft controls vibrating to inform the pilot his engine is about to stall [2].

Currently, haptic feedback for virtual reality is still in its infancy. While some great advancements have been made, most devices are either bulky and unwieldy, or only apply feedback on only a few joints, which heavily reduces effectiveness. For instance, current industry leaders in haptic technology use hydraulic force to provide tactile and force feedback. While the feedback provided is immersive, the hydraulic pads only cover a small part of the hand, and to provide feedback to the entire hand would make the glove heavier and bulkier. So, the goal of this project is to solve these two problems: to reduce size and weight of the glove, as well as maximizing the amount of feedback the hand receives.

Aside from being bulky, many modern solutions require complex mechanical systems that drive up the price of the overall glove. The solution proposed will have a much lower overhead cost, allowing for more attractive prices for industries and consumers. This will lead the way to further development of VR technologies as increased accessibility will provide more exposure to haptic feedback solutions.

2.2. Overview

This project is a continuation of a previous project focusing on the design of a haptic glove. The project aims to provide research in flexibility, precision, and feedback while completing the proof of concept started by last year’s senior design team. There are three fundamental goal areas for this project:

1. Maintain dexterity of hand
2. Track the 6 degrees of freedom in the hand

3. Generate a resistive force feedback to each finger

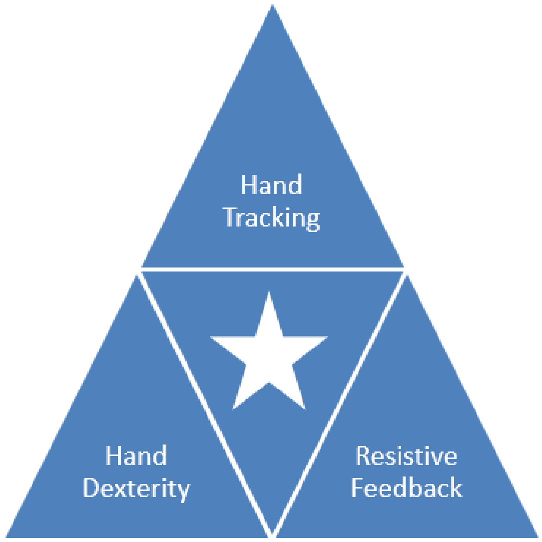


Figure 1: Triangle of Optimal Results

The kinesthetic reaction subsystem in the previous 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.

Our project and haptic glove solution will focus on reusing and building on the subsystems created by our predecessors. This will allow us the best chance to have a usable product by the end of the year. As detailed below, our work will focus on improving glove structure, sensor placement, electrostatic brake design, and the user interface.

2.3. Referenced Documents and Standards

[1] R. J. Stone, “Haptic feedback: A Potted History, from Telepresence to Virtual Reality,” in Workshop on Haptic Human-Computer Interaction,ser. LNCS, S. Brewster and R. Murray-Smith, Eds., vol. 2058. Glas-gow, UK: Springer-Verlag Berlin Heidelberg, 2000, pp. 1–7

[2] *HAPTICS TECH*, 5 Dec. 2016, hapticstech.wordpress.com/.

[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 follow the original requirements of the project set by the previous team, which are 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 goal of this project is to be used alongside modern virtual reality programs to allow the user to interact with the virtual environment in an immersive way that feels natural to use. The user will use our haptic glove design, with its sensors and electrostatic brakes, to simulate the sense of touch in the virtual environment, by simulating the passive, reactive force of a real world object when touching a virtual object, allowing the user to use their sense of touch to navigate around the virtual environment. Our glove will work similarly to other virtual input devices used today, however our glove will send information of virtual interaction to the microcontroller, which will then process the information and supply the voltage needed for that interaction from the power supply to each haptic module.

The constraints of this system include:

- Budget for Tracking Accuracy:

  Though there are more accurate position trackers available to track something like the tip of each finger, our budget prevents us from going that route. An alternative which fits in this constraint is a single 3D position tracker placed on the wrist paired with sensors to find relative locations of fingers.

- Cables:

  Due to the amount of electrostatic brakes needed and precision of voltage needed to allow for precision feedback, cables connecting to each module will be needed to supply the needed voltages. This is a constraint because the need to give force feedback contradicts the goal of retaining user dexterity and virtual reality immersion.

3.3. System Description

MCU and Hand Tracking Assembly:

The movement of the fingers, hand, and wrist will be tracked by strategically placed flex sensors. This data will be sent to the MCU where it will be processed. The necessary voltage requirements to be applied by the Power Supply Assembly will be determined by the MCU based on the user's hand position relative to a virtual object in the User Interface. This data will be sent by the MCU to the Power Supply Assembly to communicate the necessary voltages to be applied to the electrostatic brakes in order to achieve a desired force feedback.

Power Supply and Distribution Assembly:

This subsystem will interface with a standard 120 VAC RMS wall outlet. The AC supply will be multiplied and rectified into a DC voltage within the required operating level for each module (0-1300 VDC). The MCU and Hand Tracking Assemblies will communicate how much voltage for the required force and when that voltage must be applied.

Graphical User Interface:

The User Interface consists primarily of the virtual environment that the user will be interacting with. This environment will include a virtual model of the user's hand that will replicate the positioning of the user's physical hand based on information received from the Glove Assembly. By providing the user with visual feedback the effectiveness of the haptics implemented in the glove will be reinforced, thus improving the user experience. The virtual environment will also include any virtual models that the user will be interacting with. The User Interface will use its tracking and models with the MCU Assembly to determine when a user should experience haptic feedback and the level of feedback necessary.

Glove Assembly:

The glove assembly is critical for achieving the desired outcome of a functional and usable system. One limitation of the existing, partially completed glove, is a structure not sturdy enough to withstand the force between the electrostatic brakes and the fingers. The glove will provide a structure to withstand the force over repeated use. The glove will impose minimal restrictions on the user's motion. The glove will hold pairs of conducting metal strips attached at each force point of the fingers and near the wrist for the electrostatic brakes. These metallic strips will be separated by a dielectric insulator. When the hand in the virtual environment interacts with an object a voltage will be applied across the strips to provide a stopping force calculated by the microcontroller. At a high level, this design is a primarily mechanical component. However, integration with the hand tracking sensors, electrostatic brakes, and its proximity to the end user makes it one of the most complex problems.

3.4. Modes of Operations

- Tracking without Interaction

  The Haptic feedback glove will function as a controller for the virtual environment. Position of the hand and fingers will be mirrored and displayed in the virtual space without applying any force feedback.

- Tracking with Hard Interaction

  The hard interaction mode will provide maximum force feedback through the haptic glove based on interactions in the virtual environment. This will occur with hand tracking and the stopping force will only be applied to the joints indicated by the MCU as currently interacting with a hard object.

- Tracking with Soft Interaction

  Soft haptic feedback will provide a suggestive force to the given joint. This intends to simulate a soft object which would not fully stop the finger. This will occur with hand tracking and the resistive force will only be applied to the joints indicated by the MCU as currently interacting with a soft object.

- Standby

  In standby, the system will not receive or transmit information. It is only waiting for the user to change to an operational mode.

3.5. Users

Our haptic glove is targeted to be used in TAMU's virtual reality lab. The resulting product of our project will demonstrate haptic feedback to users of any experience level in virtual reality. While we plan to only make one glove, the design will be replicable for use in other labs or applications.

3.6. Support

Each glove will be accompanied by a manual which describes how to set up the device and enumerates common problems with their solutions. While the user is in the virtual space, we hope the user will require little to no instructions as we aim to mimic real life interactions.

4. Scenarios

4.1. Space Exploration

Potential use for the space industry in training and sustaining operations. Virtual environments can allow for increased training exercises, since the user can better experience a space-like environment. With our haptic glove, these exercises will increase the user's spatial awareness of the objects around them. There are existing sites such as the Sonny Carter Training facility that simulate many situations in zero gravity. Sites such as Sonny Carter require a large investment of capital and are restrained in areas that they may be built. With further development of VR technologies like the Haptic Glove, the space industry is closer to having the ability to simulate space-like situations at a much lower cost and in many locations.

4.2. Medical

Giving doctors a virtual space to interact with patients has potential to benefit both parties. The use of a realistic, virtual space for training could give medical students experience before they even work with a real patient for the first time.

Recent improvements in speed and reductions to latency (e.g. 5G networks) have made performing surgeries in real time from a remote location a possibility. Accurate physical feedback given to a surgeon's hands paired with the improvements in network speed could make this possibility a reality. Other medical applications could be in supplying virtual environments for physical therapy or immersive sensory environments for the blind.

4.3. Design

Computational models and software are popularly used in design processes because of their ability to accurately simulate physical properties. Often the user interfaces paired with these models are not intuitive to use. They also lack in giving users a hands-on experience.

Pairing current design software with virtual reality and haptic feedback to the hands could give realistic hands on design experience to users. Bridging the gap between difficult design interfaces and computer simulations would also allow users to focus on learning in their subject rather than wasting time learning how to operate software. This technology would also eliminate time spent acquiring physical parts for design and eliminate the cost associated with new designs.

5. Analysis

5.1. Summary of Proposed Improvements

- The glove will utilize a low-footprint exoskeleton that will provide support for the electrostatic braking plates.

- The movement sensors will be placed in specific locations on the exoskeleton to allow for precise tracking of the hand and finger locations.

- The electrostatic plates will be thin and flexible allowing for a more immersive experience for the user.

- The design, particularly the shape and size, of the electrostatic plates will allow for forces of up to 15.5 N to be emulated.

- End caps will be placed on the tip of each finger and mounted on the exoskeleton to provide more strategic leverage for the electrostatic plates.

5.2. Disadvantages and Limitations

The electrostatic braking concept requires that the haptic feedback glove provides voltages of up to 1200 V. This places several restrictions upon the implementation of the design, particularly in the power supply system. To produce an adequate voltage the glove must draw power from a standard outlet and then the AC voltage must be stepped up to the required level via a separate voltage multiplier circuit. The braking systems will also need to operate at very low currents in order to remain safe to the user at high voltages. The need to use standard AC wall power also limits the user’s mobility as the gloves will be tethered to the power supply and will only be able to travel as far as the power cable permits. The voltage multiplier circuit needed to provide the high voltages necessary will consume a lot of space and weight on the glove due to the large capacitance requirement.

5.3. Alternatives

The most common alternative to our haptic feedback design is a hydraulically powered glove. While hydraulics allows for greater force to be achieved, it is heavier and bulkier than the electrostatic brake concept. Other competitors in haptic feedback gloves provide vibration or an array of several small touch points. These are better at providing the sense of touch across your skin. However, when used alone, they do not provide much resistance to the movement on fingers, so they are less effective at providing the sensation of gripping or supplying stopping force.

Another alternative would be to utilize a motor-pulley system. This would involve having wires connected to the exoskeleton of the glove at the fingertips and joints. The other end of the wires would be connected to small motors that would wind or unwind the wire to simulate the desired force. The motors would need to be mounted on the glove around the wrist and the wires would extend to their contact points.

Haptic Feedback Glove

Ben Tures
Nicholas Minton

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 0
20 September 2020

FUNCTIONAL SYSTEM REQUIREMENTS
FOR
Haptic Feedback Glove

PREPARED BY:

Team 64 9/20/2020

Author Date

APPROVED BY:

Nicholas Minton 9/20/2020

Project Leader Date

John Lusher, P.E. Date

Andrew Miller Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/20/20	Nicholas Minton	NM BT	Initial Release

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

The haptic feedback glove is a practical tool to emulate real-space object interaction, allowing for greater immersion in virtual environments. It is useful in a wide range of virtual reality applications due to the versatility of the virtual space. The glove has been designed to test the validity of electrostatic braking as a lightweight solution to haptic feedback technology. A design that uses thin metal plates as the electrostatic element is seen in Figure 1.

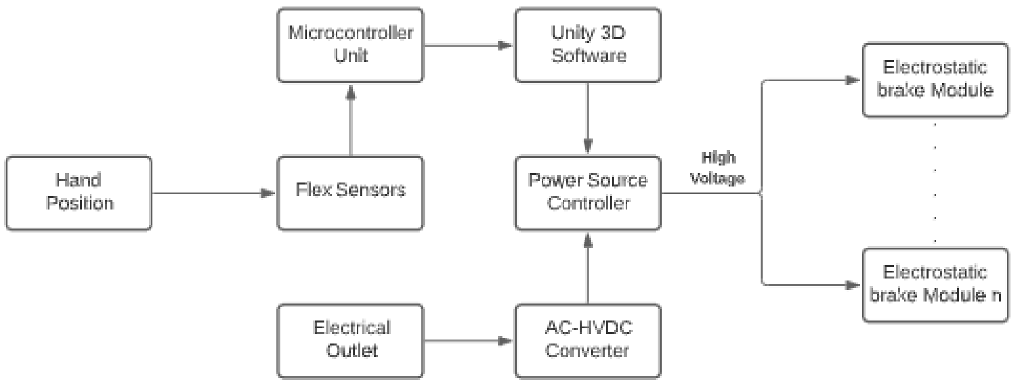


Figure 1. Haptic Glove Core System Interconnect Diagram

Sensors shall record telemetry for the position of the hand and fingers that will be used to locate the user's hand in the virtual space. The virtual environment shall detect when the user's hand is interacting with a virtual object and communicate to the MCU the required force to be simulated at each finger. When it is necessary to simulate a force the MCU shall communicate to the power supply what specific voltages will be supplied to the electrostatic brakes. The levels of force the electrostatic brakes simulate should be enough to provide the user with an immersive virtual reality experience.

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 Nicholas Minton 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).

Name	Subsystem
Nicholas Minton	MCU & Hand tracking
Ben Tures	GUI

Table 1: Team Subsystem Roles

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

[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

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

[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] System Requirements. https://docs.unity3d.com/Manual/system-requirements.html, last accessed on 09/19/20.

[10] System Requirements. https://www.vive.com/us/support/wireless-adapter/category_howto/vive-wireless-system-requirements.html last accessed on 09/19/20.

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 for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

3.1. System Definition

The purpose of the Haptic Feedback Glove is to complete development from the previous designed glove. The Haptic Feedback glove aims to borrow and optimize the design from last year's senior design team. The Haptic Glove will integrate electrostatic brakes at sensors to allow a user to interact with a virtual environment. The goal of this system is to track the motion of the user's hand and finger positions while providing kinesthetic feedback based on interactions in the virtual environment.

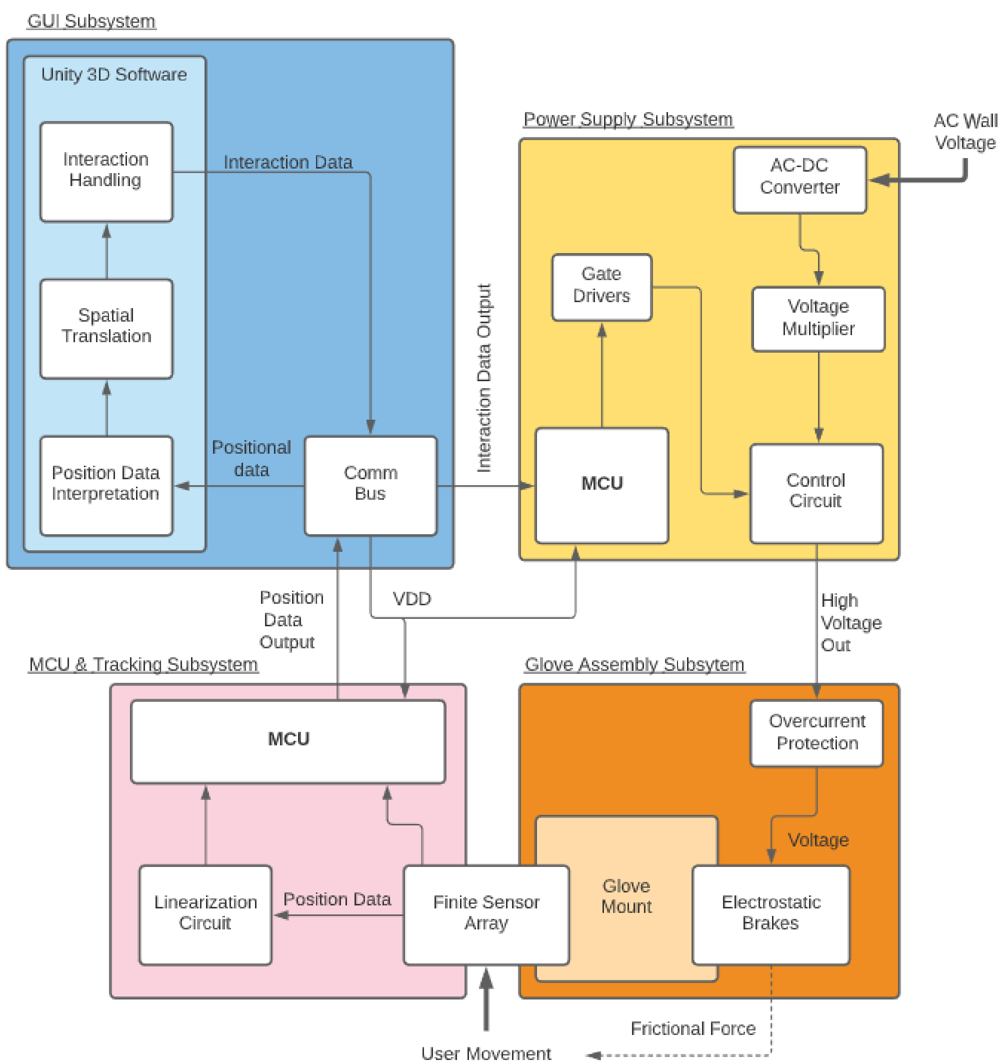


Figure 2. Block Diagram of System

The haptic feedback glove solution consists of the following four subsystems: Glove Assembly, MCU & Hand Tracking, Virtual Environment, and Power Supply. The primary input to our system is the movement of the user's right hand. That movement is interpreted by the finite sensor array and MCU. The positional data outputted by the MCU shall inform the GUI subsystem of the hands position along with the finger movement. The GUI shall handle the positional data and classify the interaction in the virtual environment. The interaction classification will determine the amount of voltage the power supply applies to the electrostatic brakes. The system can be thought of as a feedback loop. The output (force applied by the electrostatic brakes) affects the input (the user's movement).

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Electrostatic Brake

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

> *Rationale: To achieve the goal of immersion in the virtual environment we need multiple points of force feedback on each finger.*

3.2.1.2. Finger Tracking

The flex sensors shall detect finger movement at the metacarpophalangeal (MCP) and distal interphalangeal (DIP) joint. These joints shall be tracked with an accuracy of ±5°. For wrist and hand movement, the IMU shall measure acceleration, velocity, and orientation with an accuracy of ±2.5°.

3.2.1.3. Graphical User Interface

The Haptic Feedback Glove GUI shall classify each finger as being in one of three modes: Interacting with soft objects, interacting with hard objects, and in standby. This hardness classification shall determine the level of force applied to the finger.

> *Rationale: The classification of individual fingers will allow for the user to interact with the environment in a more realistic way. e.g. Only applying force to two fingers when the user picks up an object between their index finger and thumb.*

3.2.1.4. Glove Exoskeleton

The Haptic Feedback Glove shall retain structure under forces from electrostatic brakes listed in 3.2.1.1 of these documents. To retain structure, each point where force is applied shall have rigid attachment between glove and electrostatic brake.

> *Rationale: One area where our team is focusing on improving from last year is giving the glove a more rigid structure so that it can withstand more force without deforming.*

3.2.1.5. 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. Mass Located on Hand

The mass of the glove assembly, electrostatic brakes, and hand tracking subsystems combined shall be no more than 170 grams (g).

> *Rationale: 30 (10 brakes x 3strips each) stainless steel strips weigh approximately 42.2 g. Wax paper dielectric will weigh approximately 20 g. The 12 flex sensors will weigh 16 g. The IMU will weigh 4 g. An additional 80 g is reserved for the 3D printed glove exoskeleton.*

3.2.2.2. Mass Located off Hand

The mass of the hand tracking system shall be no more than 70 g. The mass of the power supply shall be no more 1.5 kg.

> *Rationale: Mass of the HTC VIVE, both PCB's and MCU's, and power supply.*

3.2.2.3. Size

The Size from fingertip thimble to base of electrostatic brakes should be between 17 cm and 20 cm

3.2.2.4. Mounting

The mounting information for the Haptic Feedback Glove shall be captured in the Haptic Feedback Glove ICD.

3.2.3. Electrical Characteristics

3.2.3.1. Inputs

- The presence or absence of any combination of the input signals in accordance with ICD specifications applied in any sequence shall not damage the Haptic Feedback Glove, reduce its life expectancy, or cause any malfunction, either when the unit is powered or when it is not.

- No sequence of command shall damage the Haptic Feedback Glove, reduce its life expectancy, or cause any malfunction.

> *Rationale: By design, should limit the chance of damage or malfunction by user/technician error.*

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.

> *Rationale: This is a requirement specified by our customer due to constraints of their system in which the Search and Rescue System is integrating.*

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). This will be detailed in the Haptic Feedback Glove ICD

3.2.3.1.3 Power Consumption

(a) The power supply shall consume no more than 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 will 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 MOSFET transistor resistance, and current limiting resistance will be the only resistances seen by the power supply.*

(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:

$$I_{leakage} = \frac{Q}{kP}$$

$$k = 3.4$$

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

$$C_{strip} = \frac{\epsilon_r \epsilon_o A}{d} = \frac{3.4 * \left(8.85 \times 10^{12} \frac{F}{m}\right) * (11\ cm^2)}{13 \mu m} = 25.461\ \mu F$$

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

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

$$P = 6.24 \times 10^{-19}\ A * 12500\ V = 7.8 \times 10^{-15}\ W$$

(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. The Flex resistors have an input of 3.3 V and at maximum flex will consume 13 mW.

3.2.3.2. Outputs

3.2.3.2.1 Force Output

The force feedback output shall apply a minimum of 4 N of force to the tip of each finger. The force feedback output shall also apply a minimum of 9 N of force on the lower distal joint of finger each finger. This force feedback shall vary in magnitude depending on the virtual object's hardness classification.

3.2.4. Environmental Requirements

The Haptic Feedback Glove shall be designed for indoor use. Testing shall occur in what is considered typical operating conditions.

> *Rationale: The intended user of VR equipment is someone with a computer and access to power. This will be indoors in typical operating conditions for nearly all situations.*

3.2.4.1. Pressure (Altitude)

The Haptic Feedback Glove shall be able to function properly at all altitudes ranging from sea level to 20,000 ft.

3.2.4.2. Thermal

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

> *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. 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 will provide protection against short circuits throughout the system. Additionally, there shall be over current protection on the braking system to ensure there is not a dangerous amount of current near the user's hand.

Rationale: Having over current protection on both subsystems is a safety precaution. There would need to be two concurrent failures for a dangerous situation to occur.

4. Support Requirements

4.1.1. System Support Requirements

The system requirements for supporting the software to interface with the Haptic Feedback Glove are as follows: [9]

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

4.1.2. Virtual Reality Requirements

The system shall 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: [10]

- 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
- OS: Windows 7 SP1, Windows 8.1, Windows 10 or later

Appendix A: Acronyms and Abbreviations

MCU	Microcontroller Unit
UI	User Interface
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
ICD	Interface Control Document
MCP	Metacarpophalangeal
DIP	Distal Interphalangeal
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
VR	Virtual Reality
mA	Milliamp
μA	Microamp
mW	Milliwatt
V	Volts
N	Newton
ft	Feet
cm	Centimeter
lbs	Pounds
μm	Micrometer
Hz	Hertz

Appendix B: Definition of Terms

Fingers	In this document, fingers refer to each of the five jointed parts of the hand. This includes the thumb

Haptic Feedback Glove

Ben Tures
Nicholas Minton

INTERFACE CONTROL DOCUMENT

REVISION – 2
24 November 2020

INTERFACE CONTROL DOCUMENT
FOR
Haptic Feedback Glove

PREPARED BY:

Team 64

Author Date

APPROVED BY:

Nicholas Minton

Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	9/20/20	Nicholas Minton		Draft Release
2	11/24/2020	Nicholas Minton	NM, BT	MCU Comm Protocol edits

Table of Contents

List of Tables

List of Figures

1. Overview

The following ICD will detail the methods used by each subsystem of the Haptic Feedback Glove and how they meet the criteria provided in the FSR. To ensure that each subsystem will operate properly when applied to the Haptic Feedback Glove detailed descriptions of the physical, electrical, and communication interfaces will be presented. These sections will break apart each subsystem to provide a detailed overview of what is to be accomplished.

2. References and Definitions

2.1. References

[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.

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

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

[7] *Unity Documentation*. https://docs.unity3d.com/Manual/index.html, last accessed on 09/20/20.

2.2. Definitions

MCU	Microcontroller Unit
UI	User Interface
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
ICD	Interface Control Document
MCP	Metacarpophalangeal
DIP	Distal Interphalangeal
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
VR	Virtual Reality
mA	Milliamp
μA	Microamp
mW	Milliwatt
V	Volts
N	Newton
ft	Feet
cm	Centimeter
lbs	Pounds
μm	Micrometer
Hz	Hertz

3. Physical Interface

3.1. Weight

3.1.1. Glove Assembly

For purposes of the physical interface control, the glove assembly system includes the electrostatic brakes, flex sensors, and exoskeleton. The total mass of the glove assembly will be no more than 140 g. The breakdown is seen as follows.

3.1.2. Exoskeleton

The exoskeleton will weigh no more than 80 g. The exoskeleton will be 3D printed and attached at multiple points of the hand and fingers. Each finger will have two mounting locations: one at the tip of the finger and the second at the base of the finger.

3.1.3. Hand Tracking Sensors

The hand tracking system is composed of 19 flex sensors and 1 IMU. The combined weight of all the sensors, wires shall not exceed 28 g (24 g for sensors and wires, 4 g for IMU).

3.1.4. Electrostatic Brakes

The electrostatic braking system will be composed of 30 stainless steel electrode strips. There will be two brakes per finger. Each brake Each brake consists of three stainless steel strips. The total mass of these steel electrodes will be no more than 50 g. The calculation is as follows:

$$Area_{sheet} = 127\ cm * 15.24\ cm$$

$$Weight_{sheet} = 226.796\ g$$

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

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

An extra 21 g will account for mass of the dielectric film.

3.1.5. 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 than 1.5 kg, excluding cabling.

3.1.6. Hand Tracking MCU

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. Dimension of Glove Assembly

The base glove to be used will measure approximately 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 25.4 µm thick. Each flex sensor measures 5 x 0.8 x 0.22 cm. The IMU measures 2.38 x 2.38 x 0.1 cm.

3.2.2. Dimension of Power Supply

The power supply circuitry will be placed in a 3D printed enclosure that measures 30 x 20 x 10 cm.

3.2.3. Dimension of Sensor MCU

The MCU and peripheral PCB for the Glove Assembly sensors will measure 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 Brakes

The electrostatic brakes shall be mounted at two locations on each finger. The base of each brake will be secured at the wrist. One brake will be fixed on the tip of each finger by a thimble and the second brake will be secured by a mount on the base of the finger. The mount at the base of the finger will hold in

4. Thermal Interface

The Power Supply will be the only source of measurable heat emission. Since the power supply will operate at no more than 3W, the system shall provide no more than 5°C of heat. A fan may be included in the power supply housing to ensure cooling through the device and maintain little temperature deviation.

5. Electrical Interface

5.1. Primary Input Power

All power to be used by the haptic feedback glove will be supplied by a standard Type B electrical outlet providing single-phase 120 V RMS at 60 Hz. This input will be used for peripheral hardware and to supply power to the voltage control and multiplier circuits, which will convert the AC voltage into a high DC voltage. The DC voltage will be used in the electrostatic braking system to generate the simulated forces.

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 the output voltage to be easily controlled and variable, of which shall be applied as 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.

$$Vo = Vhv * D$$

5.2.3. Digital IMU Sensor

The digital IMU sensor will communicate to the MCU by the I2C 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

5.3.1. 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 Virtual Environment on the host device. There will be no physical interfacing elements on the glove itself.

5.3.2. Digital Interaction

The digital portion of the user control interface shall be the Virtual Environment on the host device that allows for calibration and setup of the device. This will be performed using simple mouse and keyboard commands. The user will be informed when the glove is active and tracking. Once the glove is active and tracking, the glove itself will become the interfacing element allowing interaction 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. Commands will be sent from the Host Device to the MCU to place the electrostatic brakes in the required mode of operation.

6.3. MCU Communication Protocol

6.3.1. Hand Tracking MCU Communication Protocol

The data being sent from the hand tracking MCU to the host computer will be sent as lines of data each containing 13 comma-separated floating-point numbers. These floating-point values will describe the rotation data being measured by the glove sensors. The first three data points in the communication stream will be reserved for the roll, pitch and yaw values of the hand as measured by the IMU. Data from the flex sensors will be placed in the remaining 10 floating-point numbers, with 2 data points reserved for each individual finger. The two data points will contain the middle and proximal joint rotation angles. In the data stream the values for the fingers will be ordered: 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:

ProximalAngle, DistalAngle

6.3.2. Power Supply MCU Communication Protocol

Data to be sent from the host device to the power supply MCU will be sent as lines of 10 comma-separated values. These values will contain the percent of force required to be applied by each electrostatic braking plate. The data will be sorted by finger in the following order: Thumb, Index, Middle, Ring and Small with two values for each finger. Force data will be divided into two joints based on how the electrostatic brake plates are connected. Each finger will have a data value for the Proximal and the maximum value of the Middle and Distal joints.

Haptic Feedback Glove

Ben Tures
Nicholas Minton

EXECUTION AND VALIDATION PLAN

REVISION – 1
11 November 2020

EXECUTION AND VALIDATION PLAN
FOR
Haptic Feedback Glove

PREPARED BY:

Team 64 09/20/1998

Author Date

APPROVED BY:

Nicholas Minton 09/20/1998

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/20/2020	Nicholas Minton	NM BT	Rev 0
1	11/24/2020	Nicholas Minton	NM BT	End of Semester status

Table of Contents

List of Tables

List of Figures

No table of figures entries found.

1. Fall 2020 Execution Plans

1.1. Fall 2020 High Level Execution Schedule

Task	24-Aug	7-Sep	21-Sep	5-Oct	19-Oct	2-Nov	16-Nov
Research and Planning							
Write ConOps							
Identify Subsystem Modifications							
Write FSR							
Write ICD							
Midterm Presentation							
Order Project Parts							
Manufacture Glove Support and Braking System							
Develop Glove Sensor MCU							
Interface Glove Sensors with MCU							
Interface Braking System with HV Power Supply							
Develop Virtual Space							
Subsystem Troubleshooting							
Final Presentation							
Final Report							
Final Demo							

Table 1: Fall 2020 Execution Schedule

1.2. Subsystem Execution Schedule

Task	24-Aug	7-Sep	21-Sep	5-Oct	19-Oct	2-Nov	16-Nov
Develop Exoskeleton 3D Model							
Request Machining and 3D Printing							
Assemble Individual Finger							
Test Single Finger							
Assemble Remaining Fingers							
Validation							

Table 2 Glove Assembly Schedule

Task	24-Aug	7-Sep	21-Sep	5-Oct	19-Oct	2-Nov	16-Nov
Learn C# and Unity							
Develop Virtual Environment							
Develop Spatial Tracking							
Develop Force Feedback							
MCU Communication							
Validation							

Table 3: Virtual Environment Subsystem Schedule

Task	24-Aug	7-Sep	21-Sep	5-Oct	19-Oct	2-Nov	16-Nov
Download and Learn ALTIUM							
Design PCB							
Order MCU and IMU							
Program MCU							
Fabricate PCB							
Demonstrate Output with Flex Sensor and IMU							
Validation							

Table 4: MCU Subsystem Schedule

2. Validation Plans

2.1. Glove Assembly

Task #	Description	Relevant Requirement	Status
1	Index Finger brake module glides correctly	Movement is not inhibited by exoskeleton	✓
2	Middle Finger brake module glides correctly	Movement is not inhibited by exoskeleton	
3	Ring Finger brake module glides correctly	Movement is not inhibited by exoskeleton	
4	Pinky Finger brake module glides correctly	Movement is not inhibited by exoskeleton	
5	Thumb brake module glides correctly	Movement is not inhibited by exoskeleton	
6	Index Finger flex sensor is secured	Movement is not inhibited by exoskeleton	✓
7	Middle Finger flex sensor is secured	Movement is not inhibited by exoskeleton	
8	Ring Finger flex sensor is secured	Movement is not inhibited by exoskeleton	
9	Pinky Finger flex sensor is secured	Movement is not inhibited by exoskeleton	
10	Thumb flex sensor is secured	Movement is not inhibited by exoskeleton	
11	Hand stretching flex sensor on pinky is secured	Movement is not inhibited by exoskeleton	
12	Hand Stretching flex sensor on index finger is secured	Movement is not inhibited by exoskeleton	
13	Mounted Electrostatic brake can withstand 1300 V	No shorts occur during voltage applied for hard object	
14	Index finger mounts withstand 9 N Force	Glove is to retain shape when force is applied	
15	Middle Finger mounts withstand 9 N Force	Glove is to retain shape when force is applied	
16	Ring Finger mounts withstand 9 N Force	Glove is to retain shape when force is applied	
17	Pinky Finger mounts withstand 9 N Force	Glove is to retain shape when force is applied	
18	Thumb mounts withstand 9 N Force	Glove is to retain shape when force is applied	

Table 5: Glove Assembly Validation Plan

2.2. MCU and Hand Tracking

Task #	Description	Relevant Requirement	Status
1	Measure Flex Resistor values	The sensors should have a resistance between 20k and 120kΩ	✓
2	Pair and demo Integrator Circuit with flex resistor	Flex sensors will accuratley measure bending	✓
3	IMU X-axis rotation	Mapping should beaccurate to less than 2.5 degrees	✓
4	IMU Y-axis rotation	Mapping should beaccurate to less than 2.5 degrees	✓
5	IMU Z-axis rotation	Mapping should beaccurate to less than 2.5 degrees	✓
6	MCU I2C Protocol	MCU interfaces with IMU	✓
7	MCU Calibration Test	Calibrate maximum bend resistance data before use	✓
8	Send data to host	MCU establish communication with host	✓
9	Send resistance data to host	Must send accurate angle data based on bend in flex resistor	✓

Table 6: MCU Validation Plan

2.3. Graphical User Interface

Task #	Description	Relevant Requirement	Status
1	Verify Sensor MCU Communication	Virtual Environment Can Communicate with Glove Peripherals	Pass
2	Verify Power Supply MCU Communication	Virtual Environment Can Communicate with Glove Peripherals	Pass
3	Detects Thumb Rotation	Virtual Environment Can Interpret MCU Data	
4	Detects Index Rotation	Virtual Environment Can Interpret MCU Data	Pass
5	Detects Middle Rotation	Virtual Environment Can Interpret MCU Data	
6	Detects Ring Rotation	Virtual Environment Can Interpret MCU Data	
7	Detects Small Rotation	Virtual Environment Can Interpret MCU Data	
8	Detects Change in Hand Location/Orientation	Virtual Environment Can Interpret IMU Data	Pass
9	Detects Thumb Collision	Virtual Environment Provides Appropriate Feedback	Pass
10	Detects Index Collision	Virtual Environment Provides Appropriate Feedback	Pass
11	Detects Middle Collision	Virtual Environment Provides Appropriate Feedback	Pass
12	Detects Ring Collision	Virtual Environment Provides Appropriate Feedback	Pass
13	Detects Small Collision	Virtual Environment Provides Appropriate Feedback	Pass
14	Virtual Environment Can Recieve Improper Data	In the Event of Faulty Data the Virtual Environment Operated Properly	
15	Applies Appropriate Level of Force	Virtual Environment Provides Appropriate Feedback	

Table 7: GUI Validation Plan

Haptic Feedback Glove

Ben Tures
Nicholas Minton

SUBSYSTEM REPORT

REVISION – 0
23 November 2020

SUBSYSTEM REPORT
FOR
Haptic Feedback Glove

PREPARED BY:

Team 64 11/23/2020

Author Date

APPROVED BY:

Nicholas Minton 11/23/2020

Project Leader Date

John Lusher, P.E. Date

Andrew Miller Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	11/23/20	Nicholas Minton	NM BT	Initial Release

Table of Contents

List of Tables

No table of figures entries found.

List of Figures

1. Introduction

The haptic feedback glove is a practical tool to emulate real-space object interaction, allowing for greater immersion in virtual environments. The system is broken down into 3 Subsystems: glove assembly, hand tracking, and the virtual environment. While each subsystem has been designed, our plan for integration has been delayed because of unexpected results from the electrostatic brakes.

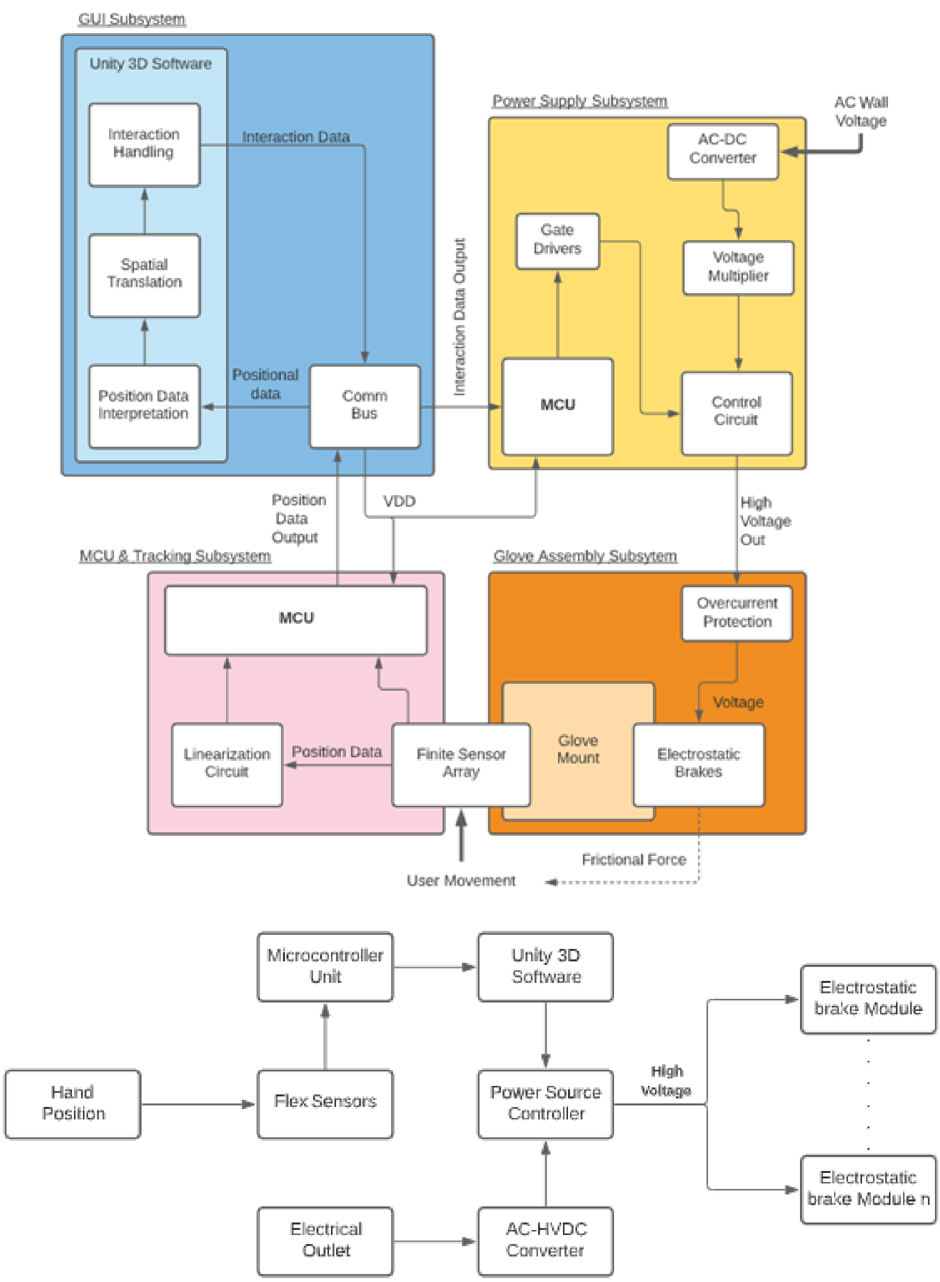


Figure 1. Haptic Glove Core System Interconnect Diagram

2. Hand Tracking Subsystem Report

2.1. Subsystem Introduction

The hand tracking subsystem is designed to provide hand and finger location data to the virtual environment. The subsystem is meant to track finger rotation as well as x, y, and z movement of the hand. Testing has confirmed that these positions can be tracked accurately.

2.2. Subsystem Details

The hand tracking subsystem is made of an array of sensors, an MCU, and voltage divider circuits for each resistor. The sensors include two bend resistors for each finger and a single IMU. The IMU is a spark fun MPU6050. The bend resistors measure rotation of each finger at the proximal and middle joints. The IMU tracks the orientation of the hand and is attached at the base of the palm.

Figure 2: Hand Tracking Setup

2.3. Subsystem Validation

2.3.1. Bend Resistors

The bend resistors were tested on hand for rotation about the proximal and middle joint of the index finger. The measured rotation data was taken with the resistors on hand and read from the serial port my MCU is writing to. During this test, I lined each joint of my finger up with a protractor to assure an accurate actual rotated angle was accomplished.

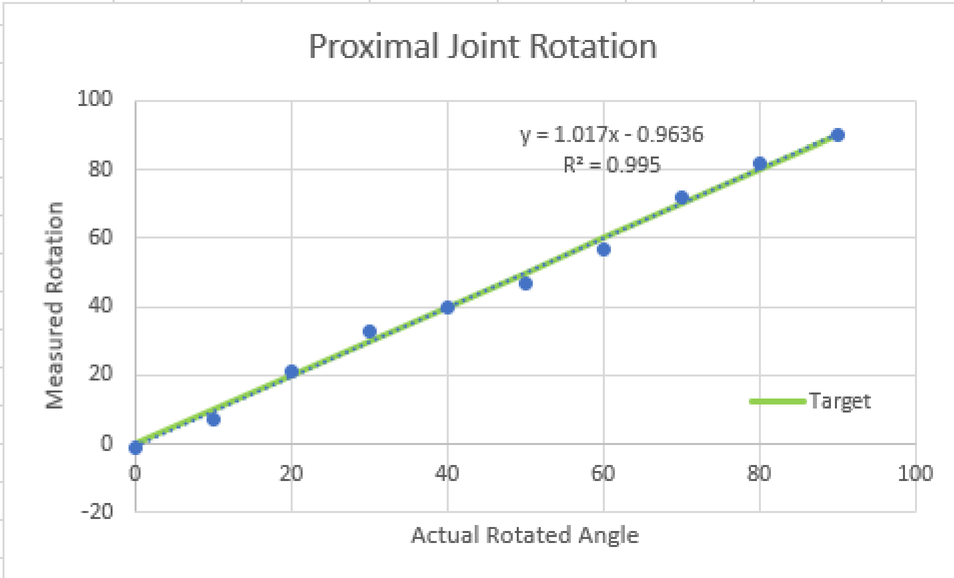


Figure 3: Proximal Joint Rotation

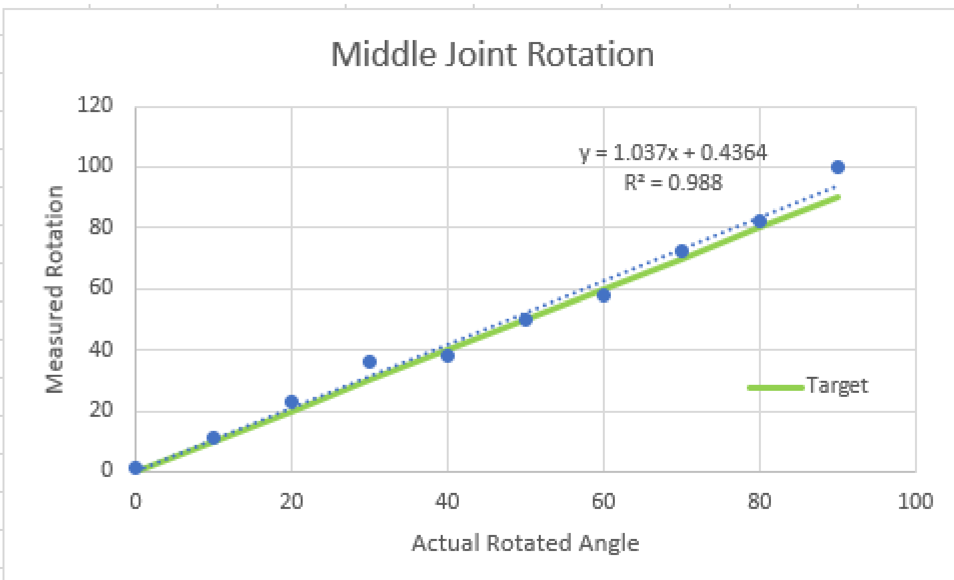


Figure 4: Middle Joint Rotation

2.3.2. Inertial Measurement Unit (IMU)

The first item to validate for the IMU was establishing the I2C protocol connection. This is done by connecting the SCL and SDA pins of the IMU and MCU. Then running a simple script that checks for an I2C device at each possible byte address (See appendix B).

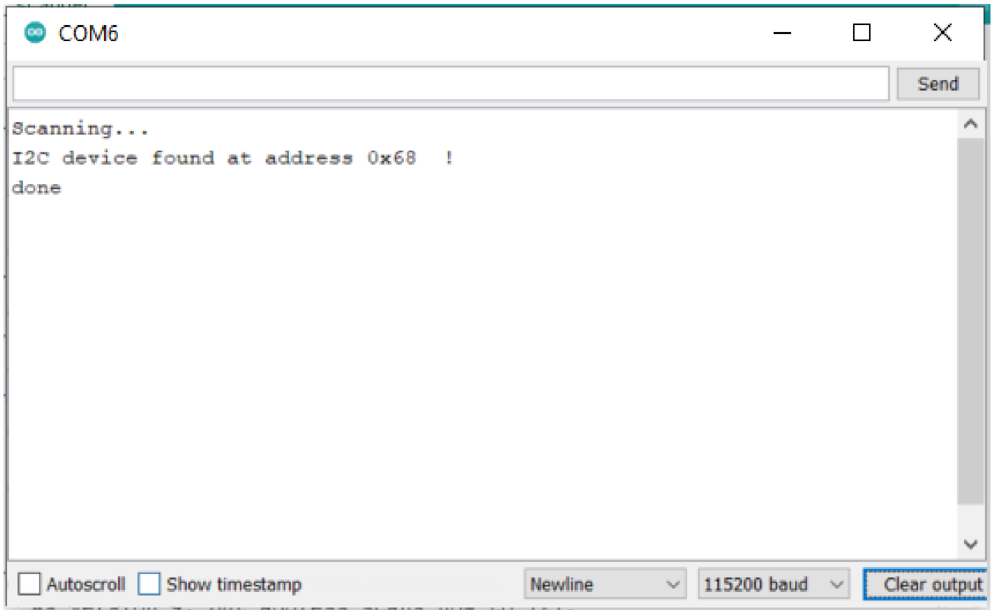


Figure 5: I2C Device Connection

The next step for the IMU was calculating values for rotation in pitch, roll, and yaw (y, x, z). To get these values, we utilize data from the IMU's gyroscope and accelerometer. The gyroscope measurement for change in degrees per second is multiplied by elapsed time to get the current angle in each direction. IMU gyroscope have errors in each direction. This Error could be found and corrected for by taking note of the returned values while the IMU is stationary. Our device had a gyroscope error of -2, -1.95, -0.6 in the x, y, and z directions respectively. The accelerometer's measurement of acceleration due to gravity in each direction is then converted into angles in the x, y and z direction.

$$\theta_{x,accel} = tan^{-1}\left(\frac{Accel_y}{\sqrt{{Accel_x}^2 + {Accel_z}^2}}\right) * \frac{180}{\pi}$$

A similar calculation can be done for the y direction's angle. Roll and pitch is then calculated using a weighted average of the accelerometer and gyroscope measured angle. Yaw's calculation relies exclusively on the gyroscope calculation. Therefore, it is incredibly important to accurately account for the gyroscope error in the z direction. The code detailing this calculation can be found in the appendix B. Results from IMU angle calculations can be seen in the figure below. I account the slight deviation in the measured values to be from the difficulty of accurately being able to rotate my hand to the precise orientation.

IMU Data	x	y	z
Target	0	0	0
Measured	-0.5	3	-4
Target	-90	0	0
Measured	-90.71	-4.87	1.62
Target	0	-90	0
Measured	3.64	-90.06	3.47
Target	0	0	-90
Measured	-8.43	-5.2	-90.36

Figure 6: IMU Results

Last, the pitch, roll, and yaw data is combined with the bend resistor data and written to the serial port as detailed in the ICD. The formatted output is shown below.

```
Roll, Pitch,   Yaw,    proximal°, middle°
0.18,-1.86,-1.71,-15.00,3.00
0.17,-1.85,-1.71,-10.00,2.00
0.17,-1.86,-1.71,-14.00,3.00
0.18,-1.83,-1.71,-11.00,3.00
0.18,-1.84,-1.71,-13.00,3.00
0.18,-1.85,-1.71,-13.00,3.00
0.19,-1.86,-1.71,-15.00,3.00
0.18,-1.85,-1.71,-11.00,3.00
0.17,-1.86,-1.71,-10.00,3.00
0.17,-1.87,-1.71,-13.00,2.00
```

Figure 7: Hand Tracking MCU Output

2.4. Subsystem Conclusion

The hand tracking subsystem met the requirements detailed in the FSR. Once the values were passed into the virtual environment, it was easy to reconfirm that values were being calculated accurately by passing the eye test. The next steps for this subsystem rely in continued integration with the virtual environment. The HTC Vive tracker is the final component in hand tracking subsystem. This tracker is designed to interface with unity and should be relatively easy to plug and play. I anticipate integration with the force feedback and scaling up to all five fingers will reveal areas of need for fine tuning the tracking. An alternative that could be worth looking into is motion tracking through video detection.

3. Virtual Environment Subsystem Report

3.1. Subsystem Introduction

The virtual environment subsystem has been designed in Unity. Inputs from the hand tracking subsystem are operated on to allow the user to interact with objects in the virtual space. IN our environment, there are two types of objects: rigid and soft. The physics engine and coded calculation translate information to an output that will control the power supply and further the electrostatic brakes.

3.2. Subsystem Details

Where possible, portions of this subsystem have been borrowed from the 2019 – 2020 senior design team. In order to be effective at making changes to the code, time was spent understanding the interconnects and structure of last year's code. The key job of this subsystem is providing a calculation for force to be used for feedback to the user's hand. Another important piece is rendering a realistic human hand. The second goal was unable to be accomplished, so visuals in this report will be that of the highlighted hand game object (the green outline in the below figures). The virtual environment consists of a hand and a table with a few items for the user to interact with on it.

3.3. Subsystem Validation

The first item to validate for the virtual environment was the physics of the hand game object.

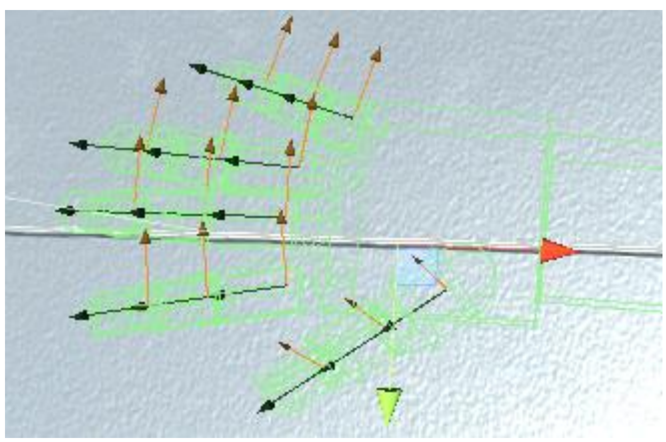


Figure 8: Hand Game Object

The hand can be operated with sliders manipulating the pitch, roll, and yaw of the overall hand or individual finger joints. When a portion of the finger intersects (or tries to intersect) with another game object, a force is returned that is split between the joints of the finger and hypothetical arm. As of now, this is simply being split up evenly between the finger's middle joint, proximal joint and the arm. An edge test case for validating the hands correct physical properties was forcing the hand to rotate into a rigid object. This causes maximum force to be applied on all finger joints. When the hand is rotated too far, it unrealistically snaps back into place. This behavior is acceptable because feedback would indicate to the user that they should stop trying to push through a rigid object.

Another Item to test was pushing a finger into a soft object that was placed on top of a rigid object. An interesting behavior was that quickly after pushing into the soft object, the force became the same as the rigid object underneath it. This models well real-world conditions when soft objects are atop rigid ones well. Lastly, half force was detected when the finger intersected a soft object on its own.

The next portion was to verify was proper handling of inputs from the hand tracking subsystem. Below you can see the unity regurgitating the hand tracking data and printing it to the console.

```
[09:35:32] -44.18,-67.64,-6.69,19.00,9.00
UnityEngine.Debug:Log(Object)
[09:35:32] -44.18,-67.64,-6.69,25.00,9.00
UnityEngine.Debug:Log(Object)
[09:35:32] finger_index_distal interacted with TABLE_Folding which is rigid (Instance)
UnityEngine.Debug:Log(Object)
```

Figure 9: Unity Console Output

The last line of the console output also demonstrates the integrated unity/hand tracking system identifying when the index finger interacts with the table. Below are visuals which visually show the virtual hand reacting to hand tracking subsystem input.

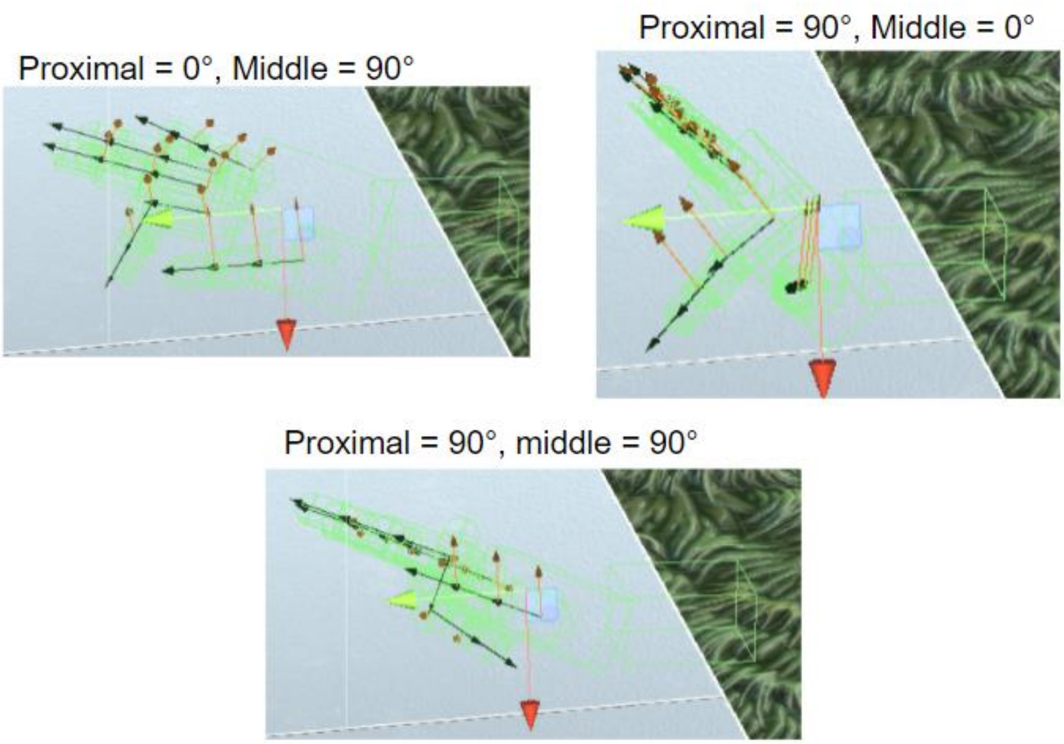
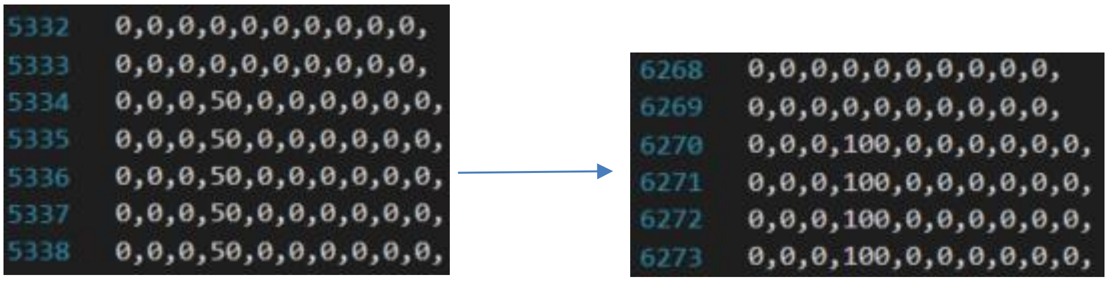


Figure 10: Unity with Hand Tracking Visual

The last point of validation was to show the output to be given to the power control MCU. As seen in the demo video, the index finger was depressed into a soft object, brought back up and then the process repeated for a rigid object. The below results were taken from a text file that was written to by unity as opposed to the serial port the MCU will read from.



```
5332  0,0,0,0,0,0,0,0,0,0,
5333  0,0,0,0,0,0,0,0,0,0,
5334  0,0,0,50,0,0,0,0,0,0,
5335  0,0,0,50,0,0,0,0,0,0,
5336  0,0,0,50,0,0,0,0,0,0,
5337  0,0,0,50,0,0,0,0,0,0,
5338  0,0,0,50,0,0,0,0,0,0,
```

```
6268  0,0,0,0,0,0,0,0,0,0,
6269  0,0,0,0,0,0,0,0,0,0,
6270  0,0,0,100,0,0,0,0,0,0,
6271  0,0,0,100,0,0,0,0,0,0,
6272  0,0,0,100,0,0,0,0,0,0,
6273  0,0,0,100,0,0,0,0,0,0,
```

Figure 11: Output for Power Control

3.4. Subsystem Conclusion

The above validation proves unity subsystem is in an operational state. The major piece of work left on this subsystem is rendering a realistic looking hand. Though we hoped to have this complete by the end of this semester, it should be noted that interfaces with the hand tracking and power supply MCU's are complete. The last step of work before the hand tracking and virtual environment subsystems are fully integrated is interfacing the HTC Vive tracker. Therefore, we can say that the virtual environment subsystem is ready for integration.

4. Glove Assembly/Electrostatic Brakes Subsystem Report

4.1. Subsystem Introduction

The Electrostatic Braking subsystem is designed to provide resistive haptic feedback for the Haptic Glove. This subsystem is comprised of a high voltage DC power supply and the electrodes that will generate the feedback force. The DC power supply supplies large DC voltages to the electrodes which will become attracted to one another by electrostatic forces.

4.2. Power Supply Details

The power supply is comprised of a 4 stage Cockroft-Walton voltage multiplier and an output voltage control circuit for each finger. A schematic of the multiplier circuit is shown below. The multiplier is able to generate the large DC voltage required to create electrostatic force between the electrodes. Safety is a concern with the high voltage generated by the power supply as the electrodes will be mounted very near the user's hands. The Cockroft-Walton multiplier works well for this application, because the impedance of the pump capacitors limits the output current. A challenge with integrating the multiplier circuit is the potential for a short circuit fault. During early testing of the brake assembly the dielectric between the electrodes tore and shorted the high voltage DC output to ground. This short damaged the power supply

thus limiting the output voltage significantly. Minimizing the chance of a short circuit fault would be a key consideration when designing the electrodes for the braking system.
The output voltage is managed by a control circuit and an MSP432 microcontroller. The microcontroller outputs a control that can alter the output voltage of the control circuit by modifying its duty cycle. Being able to select from a range of different voltages at the output allows for different levels of forces to be simulated leading to a more immersive experience for the user.

4.3. Electrostatic Brakes Details

The electrostatic brakes themselves are comprised of two electrodes separated by a thin dielectric. The power supply supplies a DC voltage to one electrode, while the other is tied to the common ground. This voltage differential should be large enough that the plates are then pulled together. Once actuated, the plated remain separated by a thin dielectric material, in this case a polyimide tape was selected. Previous iterations of this project believed that wax paper would be ideal for this dielectric as it possessed more favorable mechanical properties, however to maximize the possible force generated polyimide was selected for its more favorable electrical properties (relative permittivity, breakdown voltage, etc.). The electrostatic force pulling the parallel plated toward one another multiplied by the coefficient of friction of the dielectric would translate to the maximum shear force that would be experienced if one were to move the plates.
To integrate the electrostatic brakes a 3D-Printed structure was designed. This structure would allow the shear generated by the brakes to be translated to the user’s hand. The structure provides points for the electrodes to be mounted on top of the user’s hand so to not impede regular movement. To maximize brake actuation time, the structure was designed to maintain minimal unpowered distance between the two electrodes of the braking subsystem. When fully integrated this structure would likely be fixed outside of a protective glove for the safety of the user in the event of an unexpected fault in the braking subsystem.

4.4. Subsystem Validation

The power supply used was already validated by the previous iteration of the project, so the focus was placed on validation of the braking subsystem.
To design an appropriate electrode the following formula for the electrostatic compression force was considered.

$$F_{\mathrm{N}} = \frac{\varepsilon_0 A}{2} \frac{\varepsilon_r V^2}{d^2} = \frac{\varepsilon_0 A}{2} \varepsilon_r E^2$$

Figure 11: Electrostatic Compression Force Equation

Firstly, the constraints of the system were considered. The maximum output of the power supply (V) was measured to be near 1350 VDC setting the best-case scenario for the force generated. The next constraint was the size of the electrostatic brake electrodes (A). For our planned design using small enough electrodes to fit over each finger on the user’s hand would allow the mounting mechanism to be lightweight and unimposing for the user. An electrode overlap area of 19.5 cm^2 was then selected as the initial test value. The parameter with the most area for experimentation was the selection of the dielectric material. The relative permittivity of the dielectric material would decide the value for epsilon-r in the above equation. The thickness of the selected dielectric material would also set the minimum distance between the plates when in operation (d). Minimizing this distance was critically in the design process as the compression force was inversely related to the square of this distance.

With the listed parameters considered the first validation test was prepared with an overlap area of 19.5 cm^2 at a voltage of 1305 VDC and double layer of polyimide for the dielectric to prevent shorting (50 µm separation). During this test no shear force was able to be measured. Another test was run with the same configuration, but with only one layer of polyimide (25 µm separation). Again, no force was able to be measured as they were too small. These results were then compared with the following calculated values.

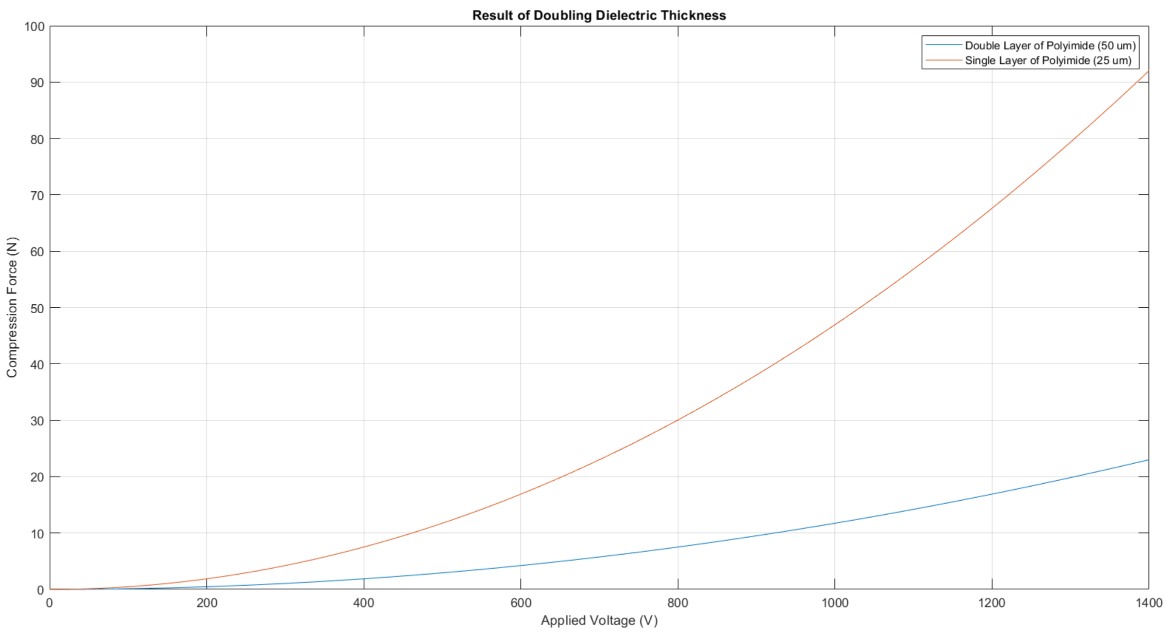


Figure 12: Theoretical Result of Doubling Dielectric Thickness

The calculated values were much higher than what was being observed with the testing setup. The initial method to adhere the polyimide to the electrode was to wrap non-adhesive polyimide film around the electrode and then to tape it. This was identified as potentially detrimental to the electrostatic force by leaving an air gap between the electrode and dielectric. To rectify this issue Kapton tape was used to adhere the dielectric material directly to the electrodes thus minimizing the air gap/distance when the electrodes were in operation.

The tests were repeated with this new dielectric, but the results were the same: no generated force.

In an attempt to generate a measurable force, the overlap area of the electrodes was considered. While increasing the size of the electrodes would modify the overall structure of the glove it was deemed a necessary tradeoff to consider in order to achieve the defined operation of the glove. It can be seen in the following figure that increasing the overlap area of the electrodes should lead to an increase in the compression force.

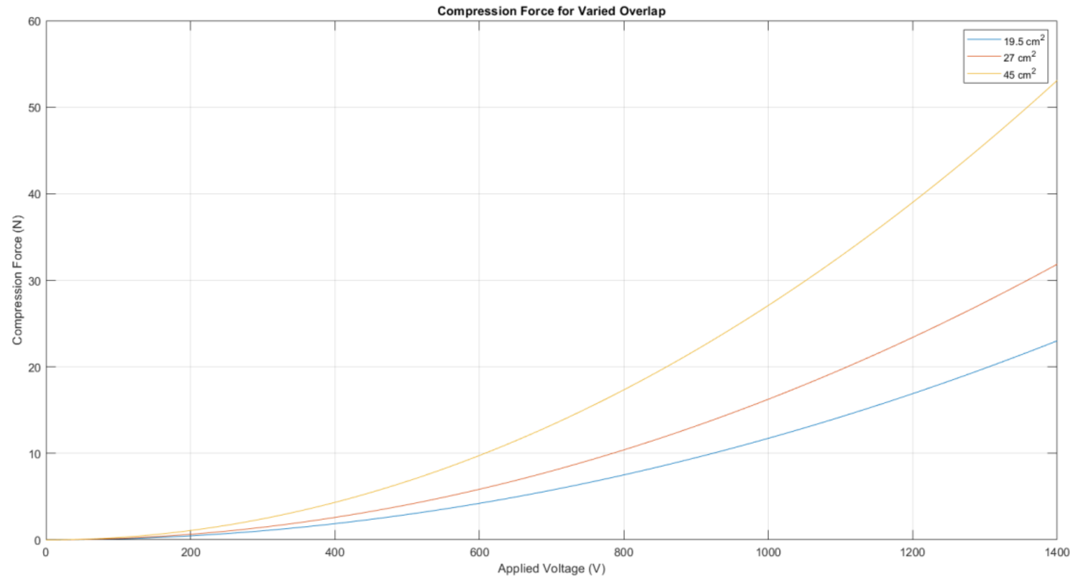


Figure 13: Voltage Response of Varied Electrode Overlap Areas

The three overlap sizes in the above figure were tested empirically. The overlap of 27 cm^2 saw the same results of the 19.5 cm^2 overlap. The 45 cm^2 overlap could be seen to flex 0.5 cm when the power was applied but the compression force was not enough to hold the plates together and so force was insignificant for the application. The electrodes will need further development to create an appropriate level of force for the application.

The mounting assembly designed for the ideal plate size was also tested. The 3D-Printed assembly allowed for free movement of the hand with minimal resistance. The maximum distance measured between the plates was 0.003 meters. The mounting system for one finger can be seen in the figure below.

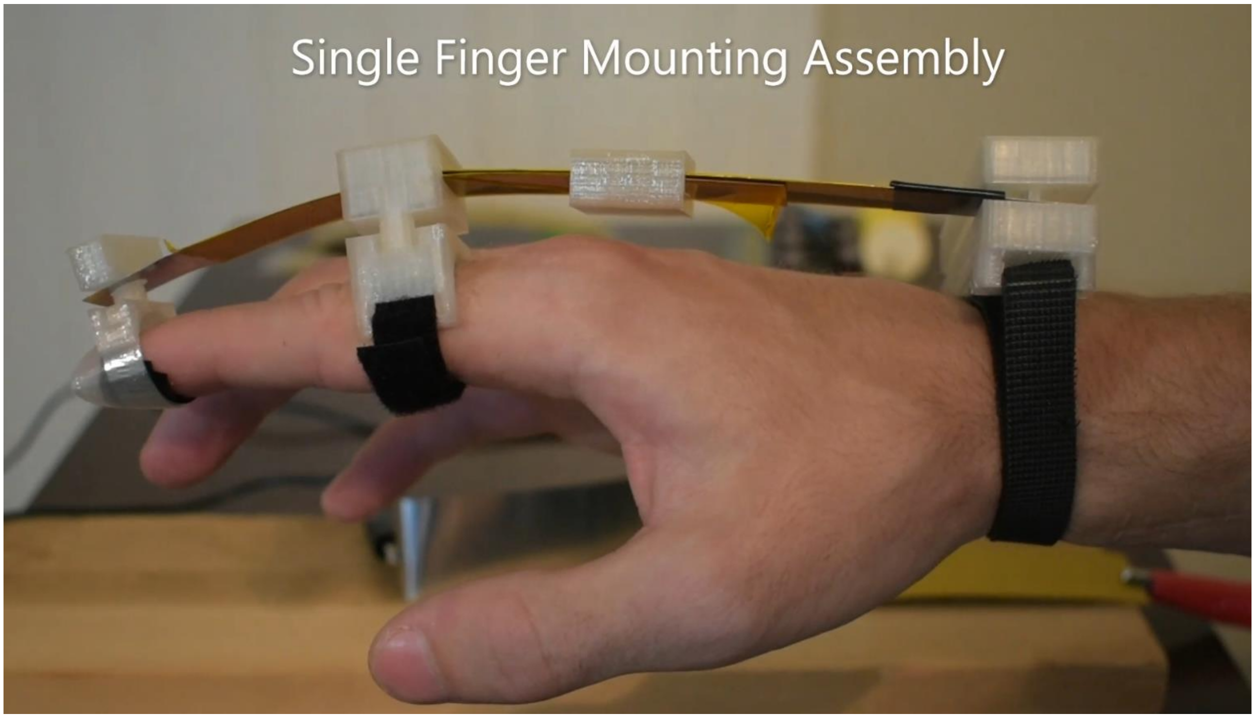


Figure 14: Electrode Mounting Assembly

4.5. Subsystem Conclusion

The power supply and mounting assemblies were shown to work correctly. The electrostatic brakes still need to be significantly improved before they are integrated into the full haptic glove. The operation of the electrostatic brakes is critical to the operation of the haptic glove and has been given full priority in further development of this project. Future modifications to be considered for the glove assembly would be to select a dielectric coating rather than an adhesive film so as to increase the surface area that comes in contact with the electrode. Modifying the output of the power supply is also in consideration, but will require further research. It has been proven by other studies that generation of large electrostatic forces is possible with a small electrode overlap area, so further refinement of the haptic glove is possible.

Appendix A: Acronyms and Abbreviations

MCU	Microcontroller Unit
UI	User Interface
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
ICD	Interface Control Document
MCP	Metacarpophalangeal
DIP	Distal Interphalangeal
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
VR	Virtual Reality
mA	Milliamp
μA	Microamp
mW	Milliwatt
V	Volts
N	Newton
ft	Feet
cm	Centimeter
lbs	Pounds
μm	Micrometer
Hz	Hertz

Appendix B: Code

I: I2C Address finder

```
// This sketch tests the standard 7-bit addresses
// Devices with higher bit address might not be seen properly.
//

#include <Wire.h>

void setup() {
  Wire.begin();

  Serial.begin(9600);
  while (!Serial); // Leonardo: wait for serial monitor
  Serial.println("\nI2C Scanner");
}

void loop() {
  int nDevices = 0;

  Serial.println("Scanning...");

  for (byte address = 1; address < 127; ++address) {
    // The i2c_scanner uses the return value of
    // the Write.endTransmisstion to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    // Serial.println("\ncheckpoint1");
    byte error = Wire.endTransmission();
    // Serial.println("\ncheckpoint2");
    if (error == 0) {
      Serial.print("I2C device found at address 0x");
      if (address < 16) {
        Serial.print("0");
      }
      Serial.print(address, HEX);
      Serial.println("  !");

      ++nDevices;
    } else if (error == 4) {
      Serial.print("Unknown error at address 0x");
      if (address < 16) {
        Serial.print("0");
      }
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0) {
```

```
    Serial.println("No I2C devices found\n");
  } else {
    Serial.println("done\n");
  }
  delay(5000); // Wait 5 seconds for next scan
}
```

II: Hand Tracking code

```
/*
   Combined and edited by Nick Minton
   Sources:
   Arduino and MPU6050 Accelerometer and Gyroscope Sensor Tutorial
   by Dejan, https://howtomechatronics.com


*/
#include <Wire.h>
/******************************************************************************
Create a voltage divider circuit combining a flex sensor with a 47k resistor.
- The resistor should connect from A0 to GND.
- The flex sensor should connect from A0 to 3.3V
As the resistance of the flex sensor increases (meaning it's being bent), the
voltage at A0 should decrease.

Development environment specifics:
Arduino 1.6.7
******************************************************************************/
//const bool TESTING = true;
const int sensorPin[2] = {A0, A1}; // Pins connected to voltage divider output
const int n_sensors = sizeof(sensorPin)/sizeof(sensorPin[0]);
float bend_angles[n_sensors];

// Measure the voltage at 5V and the actual resistance of your
// 47k resistor, and enter them below:
const float VCC = 4.98; // Measured voltage of Ardunio 5V line
const float R_DIV = 47500.0; // Measured resistance of 47k resistor

// Upload the code, then try to adjust these values to more
// accurately calculate bend degree.
const float STRAIGHT_RESISTANCE[2] = {32500.0, 23500.0}; // resistance when straight
const float BEND_RESISTANCE[2] = {48000.0, 56000.0}; // resistance at 90 deg

const int MPU = 0x68; // MPU6050 I2C address
float AccX, AccY, AccZ;
float GyroX, GyroY, GyroZ;
float accAngleX, accAngleY, gyroAngleX, gyroAngleY, gyroAngleZ;
```

```
float roll, pitch, yaw;
float AccErrorX, AccErrorY, GyroErrorX, GyroErrorY, GyroErrorZ;
float elapsedTime, currentTime, previousTime;
int c = 0;
float temp = 0;
void setup() {
  Serial.begin(19200);
  for(int i = 0; i < n_sensors; i++){
    pinMode(sensorPin[i], INPUT);
  }
  Wire.begin();                      // Initialize comunication
  Wire.beginTransmission(MPU);       // Start communication with MPU6050 // MPU=0x68
  Wire.write(0x6B);                  // Talk to the register 6B
  Wire.write(0x00);                  // Make reset - place a 0 into the 6B register
  Wire.endTransmission(true);        //end the transmission
  /*
  // Configure Accelerometer Sensitivity - Full Scale Range (default +/- 2g)
  Wire.beginTransmission(MPU);
  Wire.write(0x1C);                  //Talk to the ACCEL_CONFIG register (1C hex)
  Wire.write(0x10);                  //Set the register bits as 00010000 (+/- 8g full scale range)
  Wire.endTransmission(true);
  // Configure Gyro Sensitivity - Full Scale Range (default +/- 250deg/s)
  Wire.beginTransmission(MPU);
  Wire.write(0x1B);                   // Talk to the GYRO_CONFIG register (1B hex)
  Wire.write(0x10);                   // Set the register bits as 00010000 (1000deg/s full scale)
  Wire.endTransmission(true);
  delay(20);
  */
  // Call this function if you need to get the IMU error values for your module
  calculate_IMU_error();
  delay(20);
}
void loop() {
  // === Read acceleromter data === //
  Wire.beginTransmission(MPU);
  Wire.write(0x3B); // Start with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true); // Read 6 registers total, each axis value is stored in 2
registers
  //For a range of +-2g, we need to divide the raw values by 16384, according to the datasheet
  AccX = (Wire.read() << 8 | Wire.read()) / 16384.0; // X-axis value
  AccY = (Wire.read() << 8 | Wire.read()) / 16384.0; // Y-axis value
  AccZ = (Wire.read() << 8 | Wire.read()) / 16384.0; // Z-axis value
  // Calculating Roll and Pitch from the accelerometer data
  accAngleX = (atan(AccY / sqrt(pow(AccX, 2) + pow(AccZ, 2))) * 180 / PI) - 0.58; // AccErrorX
~(0.58) See the calculate_IMU_error()custom function for more details
  accAngleY = (atan(-1 * AccX / sqrt(pow(AccY, 2) + pow(AccZ, 2))) * 180 / PI) + 1.58; //
AccErrorY ~(-1.58)
  // === Read gyroscope data === //
```

```
  previousTime = currentTime;        // Previous time is stored before the actual time read
  currentTime = millis();            // Current time actual time read
  elapsedTime = (currentTime - previousTime) / 1000; // Divide by 1000 to get seconds
  Wire.beginTransmission(MPU);
  Wire.write(0x43); // Gyro data first register address 0x43
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true); // Read 4 registers total, each axis value is stored in 2
registers
  GyroX = (Wire.read() << 8 | Wire.read()) / 131.0; // For a 250deg/s range we have to divide
first the raw value by 131.0, according to the datasheet
  GyroY = (Wire.read() << 8 | Wire.read()) / 131.0;
  GyroZ = (Wire.read() << 8 | Wire.read()) / 131.0;
  // Correct the outputs with the calculated error values
  GyroX = GyroX + 2; // GyroErrorX ~(-2)
  GyroY = GyroY + 1.95; // GyroErrorY ~(-1.95)
  GyroZ = GyroZ + 0.6; // GyroErrorZ ~ (-0.6)
  // Currently the raw values are in degrees per seconds, deg/s, so we need to multiply by
sendonds (s) to get the angle in degrees
  gyroAngleX = gyroAngleX + GyroX * elapsedTime; // deg/s * s = deg
  gyroAngleY = gyroAngleY + GyroY * elapsedTime;
  yaw =  yaw + GyroZ * elapsedTime;
  // Complementary filter - combine acceleromter and gyro angle values
  roll = 0.96 * gyroAngleX + 0.04 * accAngleX;
  pitch = 0.96 * gyroAngleY + 0.04 * accAngleY;

  //Bend resistor Calcs
  for (int i = 0; i < n_sensors; i++) {
    bend_angles[i] = calculate_bend(i);
  }
  // Adjust valuses to be sent to unity
//  temp = pitch;
//  pitch = yaw;
//  yaw = temp;
  // Print the values on the serial monitor
  Serial.print(roll);
  Serial.print(",");
  Serial.print(pitch);
  Serial.print(",");
  Serial.print(yaw);
  for (int i = 0; i < n_sensors; i++) {
    Serial.print(",");
    Serial.print(bend_angles[i]);
  }
  Serial.println("");

  //calculate_IMU_error();
}
void calculate_IMU_error() {
```

```
  // We can call this funtion in the setup section to calculate the accelerometer and gyro data error. From here we will get the error values used in the above equations printed on the Serial Monitor.
  // Note that we should place the IMU flat in order to get the proper values, so that we then can the correct values
  // Read accelerometer values 200 times
  while (c < 200) {
    Wire.beginTransmission(MPU);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 6, true);
    AccX = (Wire.read() << 8 | Wire.read()) / 16384.0 ;
    AccY = (Wire.read() << 8 | Wire.read()) / 16384.0 ;
    AccZ = (Wire.read() << 8 | Wire.read()) / 16384.0 ;
    // Sum all readings
    AccErrorX = AccErrorX + ((atan((AccY) / sqrt(pow((AccX), 2) + pow((AccZ), 2))) * 180 / PI));
    AccErrorY = AccErrorY + ((atan(-1 * (AccX) / sqrt(pow((AccY), 2) + pow((AccZ), 2))) * 180 / PI));
    c++;
  }
  //Divide the sum by 200 to get the error value
  AccErrorX = AccErrorX / 200;
  AccErrorY = AccErrorY / 200;
  c = 0;
  // Read gyro values 200 times
  while (c < 200) {
    Wire.beginTransmission(MPU);
    Wire.write(0x43);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 6, true);
    GyroX = Wire.read() << 8 | Wire.read();
    GyroY = Wire.read() << 8 | Wire.read();
    GyroZ = Wire.read() << 8 | Wire.read();
    // Sum all readings
    GyroErrorX = GyroErrorX + (GyroX / 131.0);
    GyroErrorY = GyroErrorY + (GyroY / 131.0);
    GyroErrorZ = GyroErrorZ + (GyroZ / 131.0);
    c++;
  }
  //Divide the sum by 200 to get the error value
  GyroErrorX = GyroErrorX / 200;
  GyroErrorY = GyroErrorY / 200;
  GyroErrorZ = GyroErrorZ / 200;
  // Print the error values on the Serial Monitor
  Serial.print("AccErrorX: ");
  Serial.println(AccErrorX);
  Serial.print("AccErrorY: ");
  Serial.println(AccErrorY);
```

```
  Serial.print("GyroErrorX: ");
  Serial.println(GyroErrorX);
  Serial.print("GyroErrorY: ");
  Serial.println(GyroErrorY);
  Serial.print("GyroErrorZ: ");
  Serial.println(GyroErrorZ);
}
float calculate_bend(int pin_n) {
  // Read the ADC, and calculate voltage and resistance from it
  int flexADC = analogRead(sensorPin[pin_n]);
  float flexV = flexADC * VCC / 1023.0;
  float flexR = R_DIV * (VCC / flexV - 1.0);
  //Serial.println("Resistance: " + String(flexR) + " ohms");

  // Use the calculated resistance to estimate the sensor's
  // bend angle:
  float angle = map(flexR, STRAIGHT_RESISTANCE[pin_n], BEND_RESISTANCE[pin_n],
                   0, 90.0);
  //Serial.println("Bend: " + String(angle) + " degrees");
  //Serial.println();
  return(angle);
}
```