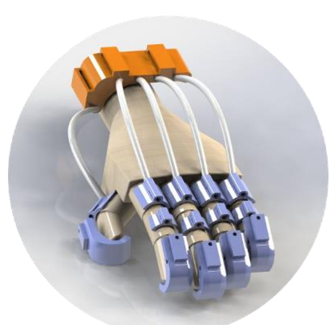
Fourth Year Design Project Final Report

Virtual Reality Haptic Glove for Upper-limb Rehabilitation: Final Design & Implementation



Prepared for
Prof. James Tung and Prof. Sanjeev Bedi
Department of Mechanical and Mechatronics Engineering
Faculty of Engineering
Waterloo, ON

Group 18
April 3rd, 2017

Prof. James Tung and Prof. Sanjeev Bedi
Department of Mechanical and Mechatronics Engineering
University of Waterloo
Waterloo, Ontario
N2L 3G1

Dear Prof. James Tung and Prof. Sanjeev Bedi,

Fourth year design project group 18 is thrilled to submit the attached report entitled **"Virtual Reality Haptic Glove for Upper-limb Rehabilitation: Final Design & Implementation"**. The purpose of the report is to show how the team created the final design as well as implementation details of the various systems involved in the final solution. The report also covers the project schedule and budget.

Our team is composed of five Mechatronics engineering students who have a common guiding vision of creating a new digital glove platform for therapy, so that the recovery process will be significantly improved. We would like to thank Prof. James Tung and Prof. Sanjeev Bedi for their support and guidance throughout the project. We would also like to thank the stroke survivors who took the time to respond to our Facebook survey on their opinions about the disability.

This report was entirely written by members of group 18 and has not received credit at any other academic institutions. All information that are not created by the team are listed in the reference section of this report.

Sincerely,

Sang Min Shin	Si Te Feng	Hernan Gomez	Ben Lambert	Sunaal Mathew
ID: 20415552	ID: 20474492	ID: 20412951	ID: 20463715	ID: 20454829

Table of Contents

List of Figures

List of Tables

Executive Summary

Happen seeks to improve patient engagement and recovery by providing the patient with tactile and force feedback in a virtual game environment designed for physical and recreational therapy. Patients suffering from brain injury, such as stroke, as well as other neurological hand disabilities, can benefit from this type of multi-sensory therapy and recover their hand functions to perform activities of daily living. Current solutions are either not sufficiently engaging, such as conventional therapy with physical objects, or lack the complete integration of the senses that promote recovery - vision, sound and touch. In addition, patients rely entirely on the therapist to provide engagement during rehabilitation which can be physically and mentally exhausting for the therapist.

Happen has successfully designed and produced a functional hand rehabilitation system that includes a complete cable-based stepper motor force feedback system, an electrotactile haptic feedback system, and a custom virtual reality game and futuristic environment in which a rehabilitation patient can enjoy gamified exercising. This system is functional for one hand, provides adjustable haptic feedback to five different fingertips, has redundant safety mechanisms built into each system, consumes less than 100 watts, and allows a user to explore a virtual space larger than 0.125 m^3.

The project budget total was estimated to be 2,100 CAD. The project cost 1081.09 CAD between September and December 2016, and total expenses to date come out to 2983.90 CAD. Third party funding was achieved through Velocity, Engineers of the Future Trust Award, Mechatronics Symposium Technical Content Award, and the department of Mechatronics Engineering. In total these funds come out to 4026.03 CAD. While the team was late in their completion of the final system integration and validation stage, the project was successfully completed and demonstrated at the Mechatronics Capstone Symposium 2017.

1 Introduction

This section introduces the project through discussion of the background, problem statement, design constraints, and design criteria. This section also discusses an overall design review.

1.1 Background

In the Western world, stroke is one of the leading causes of adult disability and one of the most common causes of death in the world [1]. Patients who have had a stroke experience a variety of motor, sensory, and cognitive disabilities depending on the magnitude and severity of the problem, [1]. In particular, a significant portion of patients have impaired upper-limb motor function following the stroke and have difficulty in independently performing activities of daily living [1].

According to studies, of a 100 people who have a stroke, approximately 75% of stroke patients are unable to recover completely – 15% are reported to die due to the stroke [1]. Patients who suffered a stroke may live with long-term disabilities or complications that lead to serious social and economic impacts [1]. Stroke costs the Canadian economy $3.6 billion a year due to physician services, hospital costs, lost wages, and decreased productivity [1]. These numbers will continue to rise with an aging global community.

For recovery, often intense and consistent physiotherapy is required. However, several challenges exist with these conventional therapy methods such as the lack of patient engagement and motivation, monotonous repetitive therapy that may not encourage motor learning, lack of adequate feedback, poor measurement of participation and recovery, the need for the constant presence of therapists or caregivers, and the limited resources.

1.2 Problem Statement

A more immersive rehabilitation program – in addition to conventional therapy – needs to be developed such that it will stimulate sensory and motor feedback and encourage faster recovery in patients through increased and motivating participation in therapies. There needs to be therapy and technology that allows patients to engage in immersive, independent and repetitive training to encourage recovery and restore functional loss while performing activities of daily living.

1.3 Design Constraints

The constraints for the project are the following:

- Must be functional for at least one hand
- Must be able to simulate the visual, force and tactile sensations from the interaction with at least one virtual object
- Must individually control at least 2 different haptics actuation points
- Must provide levels of force and tactile feedback to the user based on user's interaction
- Maximum actuation force produced must not exceed 30 [N] - approximately 3 [kg] mass
- Must have at least one safety mechanism in each of the systems - mechanical, hardware and software
- Must not cause any damage or harm to the user in any way
- Energy consumption must not exceed 100 [W]
- Virtual workspace must be a cube of at least 0.5 [m] on each side - this is the approximate reach of an average person
- Must provide the user with at least one performance metric relating to movement kinematics

Note that these constraints were taken from the Final Design Proposal Report submitted to the MTE 481 course [2].

1.4 Design Criteria

The criteria for the project are the following:

- Highest spatial resolution of tactile sensations
- Minimum response time or delay of system and components
- Minimum resistance to the user's natural biomechanics
- Minimum engineering risk
- Highest performance reliability
- Highest degrees of freedom

Note that these criteria were taken from the Final Design Proposal Report submitted to the MTE 481 course [2].

1.5 Design Review

1.5.1 Constraints Analysis

The overall design fulfilled most of the constraints set for the project.

The fulfilled constraints include, but are not limited to, the one hand functionality, the set limitations set on both force and energy outputs, safety, and simulation for at least one virtual object. However, the design was unable to provide the user with reliable performance metrics related to movement kinematics. The design implemented a LeapMotion to track the hand position and posture. Ideally, the data provided by the LeapMotion was to be used to calculate performance metrics; however, the data reliability was sometimes questionable as the Leap Motion did not always accurately determine the hand's position or posture. This was especially troublesome when there were any forms of obstruction between the LeapMotion and the user's hand. For future development, a reliable motion tracking system must be developed.

1.5.2 Criteria Analysis

When considering the alternative solutions for each system, the criteria were used to determine which was the most appropriate. Note that all the solutions considered were initially thought to fulfill all applicable constraints.

For the force feedback system, the following solutions will be discussed: pull-wire and mechanical exoskeleton. The mechanical exoskeleton was designed to use rigid links and small DC motors to provide torques on the joints that would apply force directly to the fingers. This solution would provide reliable force feedback with a minimum time delay. Furthermore, it had the potential to be able to integrate a reliable motion tracking system. The pull-wire solution used small cables that ran along the arm to the finger through a series of tubes that were actuated through a series of motors in a base unit on the side. The pull-wire solution had significant benefits. Compared to the mechanical exoskeleton, the obstruction to the user's biomechanics, had a lower engineering risk, and safety as it was designed to only provide reactive forces.

When designing the system to provide tactile feedback, there were several solutions that were considered. The two that will be discussed are the mechanical haptic actuators and the electrotactile system. To provide tactile feedback, mechanical haptic actuators were activated to provide vibration that would vary in frequency and pattern depending on the surface. The electrotactile system was designed to provide electrical stimulation that would directly activate the nerve fibers within the hand. Depending on the frequency, amplitude, and pattern this technique is theoretically able to provide sensations like pressure, vibration, and skin stretch. Both solutions had varying strengths. The mechanical haptic actuators had a high performance reliability, was low risk and cheaper, but the electrotactile solution was deemed to be much more appropriate based on most of the other criteria. This is because the electrotactile solution minimized the spatial requirement and physical obstruction on the user's hand, had a minimal time delay, and provided a much higher spatial resolution. Most of all, the electrotactile system had a significantly larger range in potential stimulation.

For the motion tracking system, many solutions were considered including the LeapMotion and inertial measurement units (IMUs). For the IMU sensors, they would have been built into the glove and the position and orientation of the hand and fingers could be theoretically tracked after an initial calibration and processing. However, the IMU sensors had some issues associated with the level of engineering risk and performance reliability. First, if the user paused the game at all, the system would lose track of the hand and would need recalibration. Also, the efforts required to calculate finger positions from the raw data would have been troublesome as each users could vary significantly. The LeapMotion was implemented as it was simpler to use, had a lower risk, and an assumed acceptable level of accuracy. Furthermore, it did not require any additional components to be built into the glove thereby reducing any potential obstruction.

1.5.3 Overall Design

The overall design was a combination of four systems: an electrotactile system, a pull-wire system, LeapMotion for tracking, and a virtual reality environment. These systems are further discussed in detail in the later sections.

The overall design provided the user with visual, audio, tactile, and force feedback. The user was required to wear both the Oculus Rift and glove. The Oculus Rift provided the user with visual and audio stimulus of the developed virtual reality environment and game. Using the LeapMotion mounted on the front of the Oculus Rift, the user's hand was tracked for both position and posture. The data provided by the LeapMotion was used to generate and visualize the user's hand in the virtual reality environment. When the user interacted with a virtual object, the software system was designed to send a message to an Arduino via a serial communication line. This Arduino controlled a series of stepper motors that controlled the pull-wire system to provide varying levels of force depending on the interaction. At the same time, this Arduino relayed the message through another serial line to Teensy that controlled the electrotactile system to provide different patterns of stimulation. For the time being, the stimulation was limited to the fingertips.

The system performed adequately; however, there is a significant room for improvement. These improvements include developing a more reliable motion tracking method and increasing the resolution of both the electrotactile and pull-wire systems.

2 Final Design of Pull Wire Hand Exoskeleton

This section discusses the final design of the of the pull wire mechanism responsible for creating force feedback. First, an overview is provided about the final design and its subsystem, this is followed by a section that explores the divergence of this design from the original design. Next, there is a section dedicated to manufacturing and commissioning of the different parts of the system. Last, there is section dedicated to the testing and performance assessment of the system.

2.1 Final Design Details

The final design of the force feedback system consists in the assembly of three main subsystems: the pull wire lock mechanism, the pull wire assembly and the hand exoskeleton.

2.2 Pull Wire Lock Mechanism

The pull wire lock mechanism has the main function of locking the pull wire into place. A diagram showing the main components of the lock mechanism is shown in Figure 1. When the hand is flexed, the motion is transferred to the pull wire through the hand exoskeleton causing the rotation of the shaft and the pull wire to unwind from the spool. As the motor is connected to the shaft as well, if energized it will apply a holding torque, preventing the unwinding of the pull wire and thus the flexion of the hand. It is important to note that the third vertical plate shown in the diagram, was meant to be the encoder mounting plate and the final design doesn’t use it.

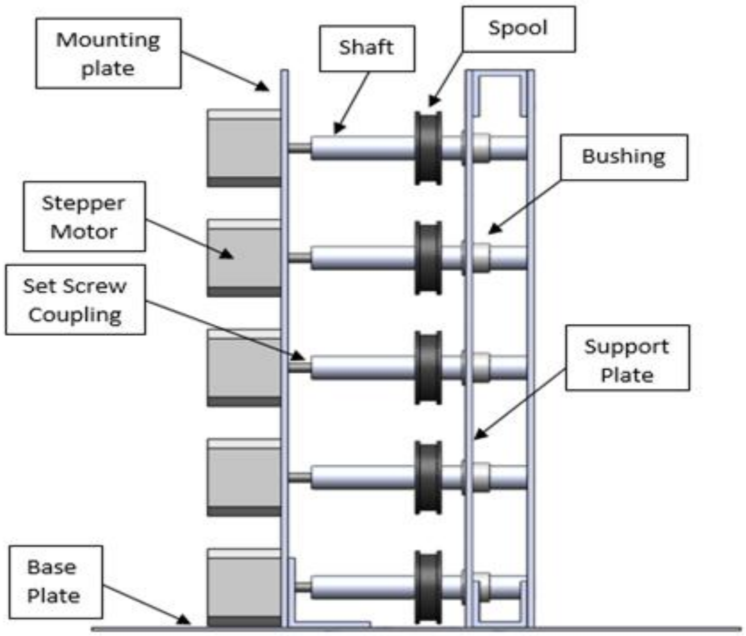


Figure 1 Pull wire lock mechanism

The following table provides details about the components of the pull wire lock mechanism.

Table 1 Pull wire lock mechanism components

Component	Description
Stepper Motor	Mercury Motor SM42BYG01125, 12V DC, 0.33 A, holding torque: 23 N.cm
Base Plate	12 in x 12 in, 1/8 in thick, material: aluminum
Shaft	½ in diameter, material: Delrin
Spool	1.5 in diameter, material: PLA (3D printed)
Bushing	Low friction, dry contact, material: nylon
Mounting and	Mounting Plate: 12 in x 3 in, 1/8 in thick, material: aluminum, mill machined
Support Plates	Support Plate: 12 in x 2 in, 1/8 in thick, material: aluminum, mill machined

2.3 Pull Wire Assembly

The pull wire assembly consist of two separate strings (monofilament fishing line) that are connected to the same spool. One half of this assembly is contained in the tabletop device of the force feedback system and is shown in Figure 2. The purpose of string #1 is to transfer the force from the spring (k = 0.57 lbs/in) to the shaft and the purpose of string #2 is to transfer the force from the motor to the hand to prevent its flexion.

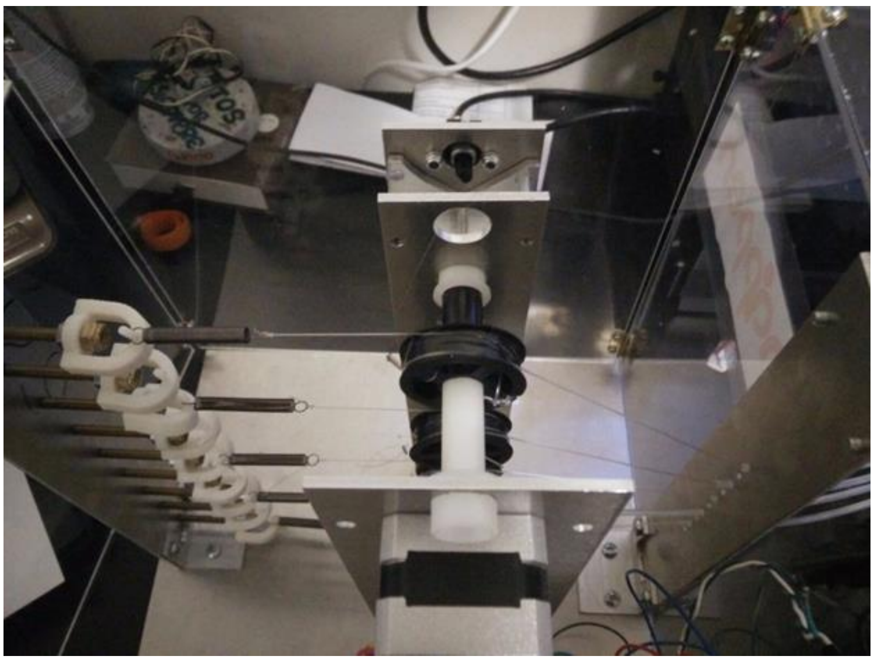


Figure 2 String assembly in tabletop device

The other half of the assembly consist of the rest of string #2 that is not contained inside the tabletop device. This half of the assembly consist of the string portion that is inside of the 70 cm long Teflon tubing and that runs through the hand exoskeleton as shown in Figure 3. This string is anchored to the distal part of each finger in the exoskeleton using a screw to clamp the string.

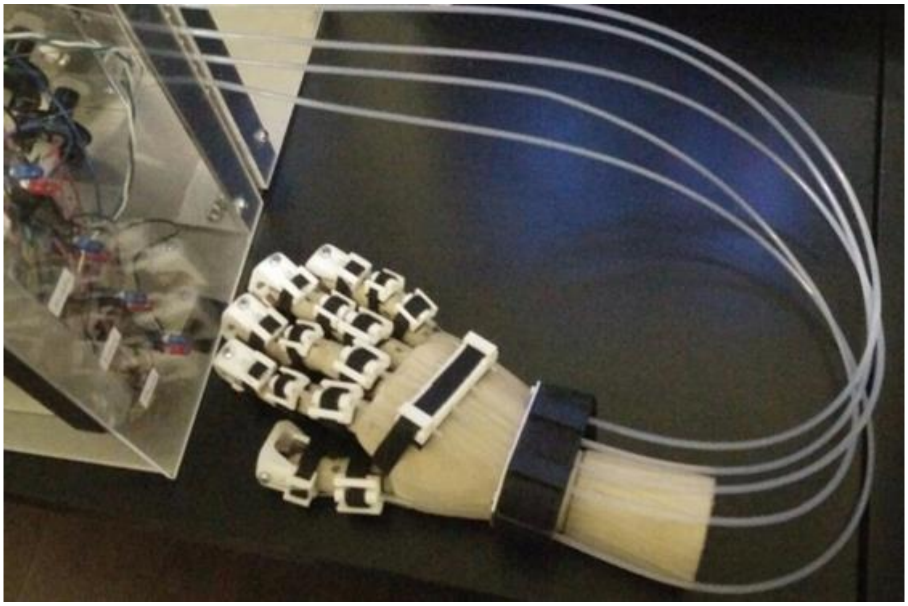


Figure 3 String assembly in hand exoskeleton

2.4 Hand Exoskeleton

The hand exoskeleton consists of 16 individual jigs as shown in Figure 4. These jigs have the function of conducting each of the five pull wires from the wrist to the tip of the fingers using segments of Teflon tubing. A cushion made of neoprene foam was used as the interface between the exoskeleton and the hand of the user. The individual jigs were made using fused deposition modeling (3D printed) with ABS. Velcro straps were used to secure the jigs into the hand of the user.

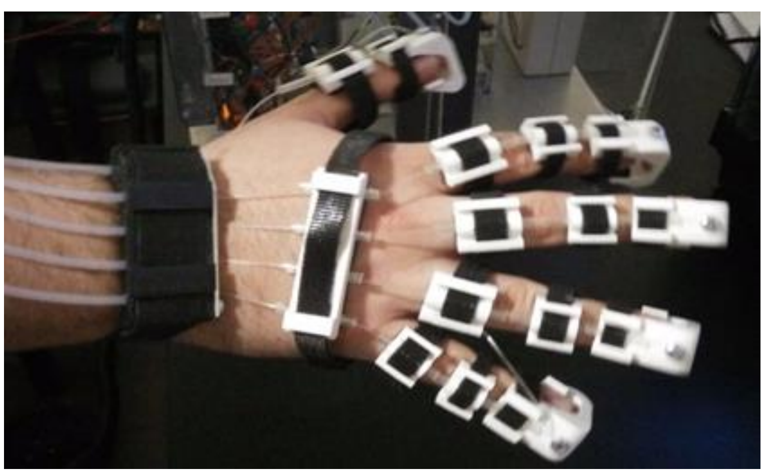


Figure 4 Hand exoskeleton

2.5 Modifications from Original Design

The only modification from the original design is related with motion tracking. Originally, the pull wires were going to be use for both, force feedback and hand motion tracking as indicated in Figure 5. But, due to the use of leap motion, the final design did not require hand motion tracking through the pull wires. This eliminated the need of encoders and the additinonal pull wires per finger.

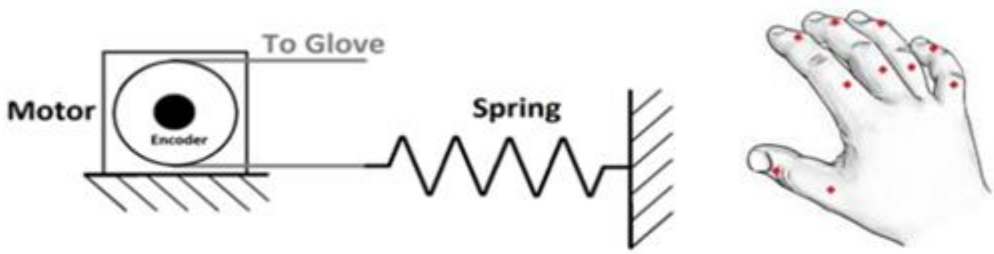


Figure 5 Use of encoders and two pull wire anchor points per finger in original design

The final design still has the capability of mounting encoders in the shafts and adding a secondary shaft and spring per finger in the tabletop device, this is shown in the Figure 6.

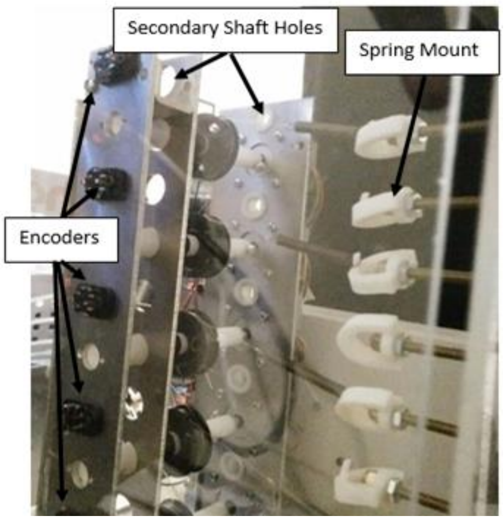


Figure 6 Final design with features of the original design

2.6 Manufacturing

Most of the manufacturing work consisted in machining the frame of the tabletop device which is shown in Figure 7. The frame consists of 1/8 in thick aluminum plates, L-shaped and U-shaped aluminum brackets. For increased accuracy and ensure perfect alignment, all the holes were made using a milling machine that provide accurate position tracking while drilling the holes.

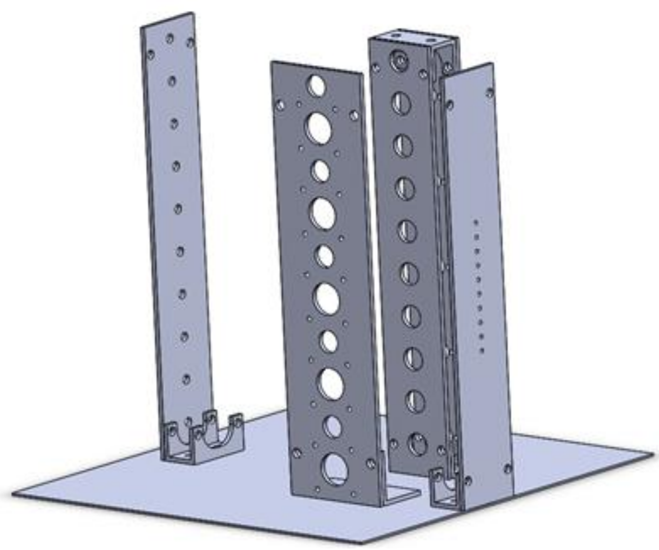


Figure 7 Tabletop device frame

The rest of the manufacturing consisted in the manual assembly of all the components. The following table summarizes the assembly methods used for the different components of the system.

Table 2 Assembly methods used

Assembly Method	Interfaces Assembled
Screws into tapped holes	Tabletop device frame, casing, motor-plate
Set screw coupling	Motor-shaft
Press fitted	Shaft-spool, tubing-plate & exoskeleton, bushings-plates
Fishing knot	Spring-string, string spool
Adhesive	Magnetic strip-casing, foam cushion & tubing-exoskeleton

2.7 Commissioning

The only components of the force feedback system that were commissioned were the 3D printed parts which include: complete hand exoskeleton, spring mounts and spools. The parts were made through fused deposition modeling using ABS plastic in the 3D Printing Center at SDC in University of Waterloo.

.

2.8 Testing and Performance

For consistency purposes the performance testing of the force feedback system was performed with the device placed in a mannequin hand as shown in Figure 8. The performance of the system was evaluated by the effort required to flex each of the 3 joint in the little finger of the mannequin when the lock system is activated. To test the proximal joint restriction, only this joint was flexed. To test the middle joint, the proximal was flexed first followed by the flexion of the middle joint. And last, to test the distal joint, the proximal joint was flexed, then the middle and then followed by the distal joint. This motion in the joints mimics the natural flexion of the hand. The flexions consisted in manually and gently flexing the mannequin finger according to the joint being studied.

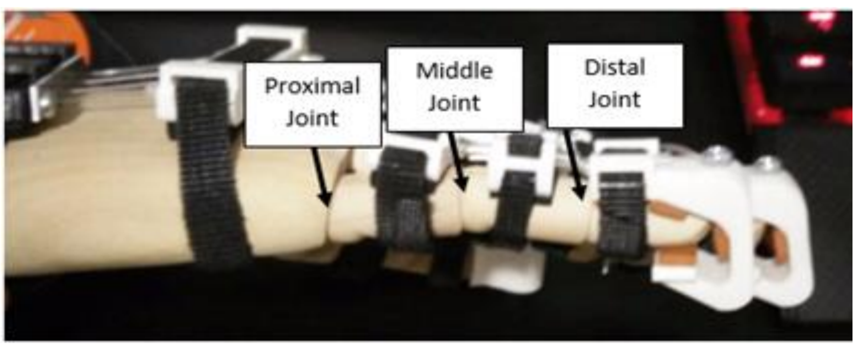


Figure 8 Joints highlighted in the mannequin hand with the device

Another important point is that the test was only done in the little finger of the mannequin as there is no friction with the other fingers and similar results are expected in the other fingers. Also, the tension in the pull wire was adjusted so the spring extends approximately 6 mm (0.134 pounds of tension) when the finger is not flexed.

One variable in the test was whether the spool was rolled back when activated or not. The roll back was decided to be 18 degrees or 10 steps of the stepper motor, this value was selected because after 15 steps of roll back it was observed some slipping in the stepper motor due to the high tension created in the pull wire.

The pass and fail criteria in the test consist of the following: fail if the joint can partially or fully be flexed with a gentle manual push in the finger in the mannequin hand and pass if the finger doesn't flex with the gentle push. The results are summarized in Table 3.

Table 3 Force feedback system test

Test	Spool rollback	Observation
Proximal Joint	No	Fail, the joint can flex nearly 90 degrees
Middle Joint	No	Pass, further finger flexion not possible
Distal Joint	No	Pass, further finger flexion not possible
Proximal Joint	Yes	Fail, the joint can flex approximately 45 degrees
Middle Joint	Yes	Pass, further finger flexion not possible
Distal Joint	Yes	Pass, further finger flexion not possible

Even with fail trials in both proximal joint tests, the use of rollback produced acceptable results for the use of the force feedback system with virtual reality.

3 Final Design of Electrotactile System

This section discusses the final design of the electrotactile system that is used to provide tactile feedback to the user. First, the final design of the electrotactile system and its comprising sub-systems are described in detail, followed by a brief description of the prototyping process from the initial design to the final implemented system. Next, design modifications are discussed in detail based on certain design criteria such as electrical safety. This section also discusses manufacturing for the final design, commissioning and lastly, testing and performance for the electrotactile system.

3.1 Final Design Details

The final design of the electrotactile system comprises of 4 subsystems - the Power Module, the Microcontroller Module, the Electrotactile Controllers and the Electrodes.

3.1.1 Power Module

The power module consists of a battery supply, a high voltage converter and a low voltage regulator. Table 4 summarizes the components selected for each of the above-mentioned systems in the power module.

Table 4 Summary of final components selection for Power Module Design

`	System	Component	Specifications	Manufacturer
1	Battery supply	LiPo Battery Pack	Voltage: 11.1 [V] Capacity: 3300 [mAh] Discharge: 30C	Turnigy [3]
2	High Voltage Converter	DC-DC Boost Converter	Input voltage: 8-16 [V] Input current: 5 [A] (max) Output voltage: 45-390 [V] Output current: 0.2 [A] (max)	Qianson [4]
3	Low Voltage Regulator	5V/3.3V Power Regulator	Input voltage: 40 [V] (DC max) Output voltage: 5 or 3.3 [V]	Sparkfun [5]

The battery supply provides the input power to the power module and should have sufficient capacity to supply the energy requirements for the electrotactile system. The selection of the 11.1[V] LiPo battery allows for the electrotactile system to be independent of the wall power supply, which is a requirement for electrical safety, given the scope of this project. The maximum output power required by electrotactile system can be calculated as follows:

Maximum Output Voltage supplied to Electrotactile Controller = 390 [V]

Maximum pulse width of stimulation = 500 [uS]

Maximum current peak of stimulation = 3.3 [mA]

Maximum frequency of stimulation = 200 [Hz]

Maximum DC value of current stimulation = (500 [uS] / (1/200[Hz])) * 3.3 [mA] = 0.33[mA]

Maximum DC value of current required from Lipo Battery = (390 [V] * 0.33 [mA])/(11.1[V]) = 11.59 [mA]

Calculated battery life given selected battery = 3300 [mAh] / 11.59 [mA] = **284.72 hours**

Hence, the theoretical battery life calculated is more than sufficient for the scope of this project, where the electrotactile device is expected to function for at least 10 hours (approximately a day of use before requiring a recharge of the Lipo battery).

The high voltage converter converts the input power from the battery supply into a higher voltage, which can be adjusted between 40-390 [V]. For the electrotactile device, the selected output voltage for the converter is 350 [V]. The input supply is rated at 11.1[V], which is within the input voltage range of the selected high voltage converter.

In order to power the Microcontroller Module, a constant 5[V] power supply is required. Since the selected power supply is rated at 11.1[V], a 5[V] voltage regulator is used to convert the 11.1[V] to 5[V].

3.1.2 Microcontroller Module

The controller for the electrotactile system was designed according to constraints imposed by the needs of the electrotactile electrode controller boards. A list of constraints formulated for the design of the controller is given below.

<u>Controller Design Constraints</u>

1. Digital to analog converter
2. Clock frequency of at least 1Mhz
3. Physical dimensions equal to or smaller than designed electrotactile controller modules
4. Support for simple programming interface such as Arduino IDE
5. Must control, per module,
    a. 2 digital outputs
    b. 2 analog inputs
    c. 1 analog output

The digital to analog converter was required for control of the analog input voltage which is used to set a current level on the current regulator. Due to expecting the minimum pulse width of the stimulation waveforms to be on the order of microseconds, a minimum clock frequency was selected to be 1 Mhz. A device was desired which would come in package smaller than the electrotactile controller boards so that one mechanical enclosure was required for both controller and electrode control modules. Additionally, support for the Arduino IDE, or other similar interface, was required as minimal project time was allocated for embedded system implementation. Lastly, due to each board requiring two stimulation current direction controls, one analog current level control, and two analog inputs for reading voltage and current from each board, the controller system was required to support five inputs/outputs for each electrode controller module. To support this busy interface problem, it was assumed that bidirectional analog multiplexer systems would be included in the design, between the controller and the electrode control boards, allowing for serial control of the inputs/outputs and a reduction in the number of GPIO required of a controller system. Three third-party complete microcontroller systems were evaluated according to these constraints: the Arduino Micro, the Pololu A-Star 32U4 Micro, and the Teensy LC.

The Teensy LC, shown below in Figure 9, was selected for the final design due to being the only microcontroller with an integrated digital to analog converter.

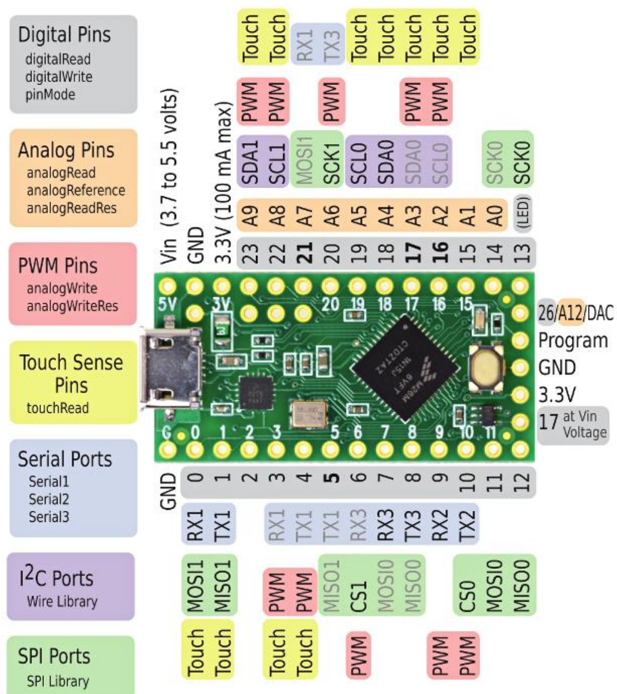


Figure 9 Teensy LC pinout diagram [6]

3.1.3 Electrotactile Controllers

The electrotactile controllers are responsible for providing the stimulation waveform to the electrodes with regulated current. Each electrode can be controlled by one controller, and the final prototype includes the use of 5 electrodes and hence 5 electrotactile controllers. The final designed and manufactured electrotactile controllers are shown in Figure 10 below.

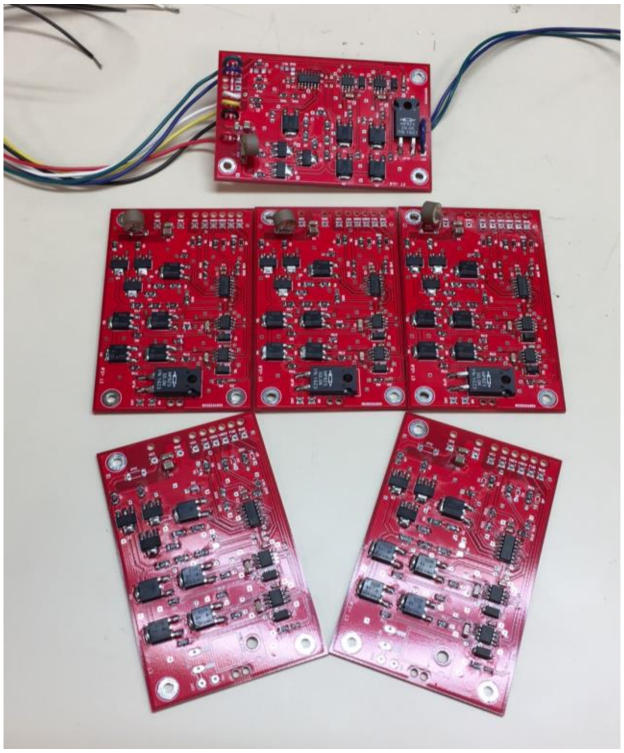


Figure 10 Fabricated and populated electrotactile controllers

The final design parameters and their corresponding values for electrotactile stimulation are shown in Table 5. These parameters are used to guide the design of the electrotactile controller, the electrodes, as well as the selection of the components.

Table 5 Final design parameters and corresponding values for electrotactile stimulation

No.	Design Parameter	Value	Units
1	Current Amplitude	0-3.3	mA
2	Duty Period (tp)	20-500	us
3	Voltage	350	V
4	Inner Electrode Diameter	1	mm
5	Outer Electrode Diameter	10	mm
6	Bandwidth (BW)	0-200	Hz

As discussed in the initial design report, the electrotactile controller design can consist of 3 main sections - Voltage to Current Converter, H-Bridge Switching system and Output Sensors.

Voltage to Current Converter: The first section is the Voltage to Current Converter design, which is a combination of a Voltage Controlled Current Source (VCCS) and a Current Mirror. This design is leveraged to deliver constant regulated current to the skin across the electrodes irrespective of the skin impedance (as long as the skin impedance is below the maximum output resistance threshold of the current mirror). Figure 11 shows the final schematic design of the voltage to current converter. The value of R2 is set to 1K [Ohm], where to drive 1 [mA] of current from VEE (350V from power module), VIN has to be set to 1 [V]. Since the VIN is limited to maximum of 3.3 [V] (maximum output voltage from microcontroller), the current can only be driven to a maximum of 3.3 [mA], which is also the maximum stimulation current possible. The output of voltage to current converter is VDD, which is the regulated supply voltage connected to the H-Bridge Switching System.

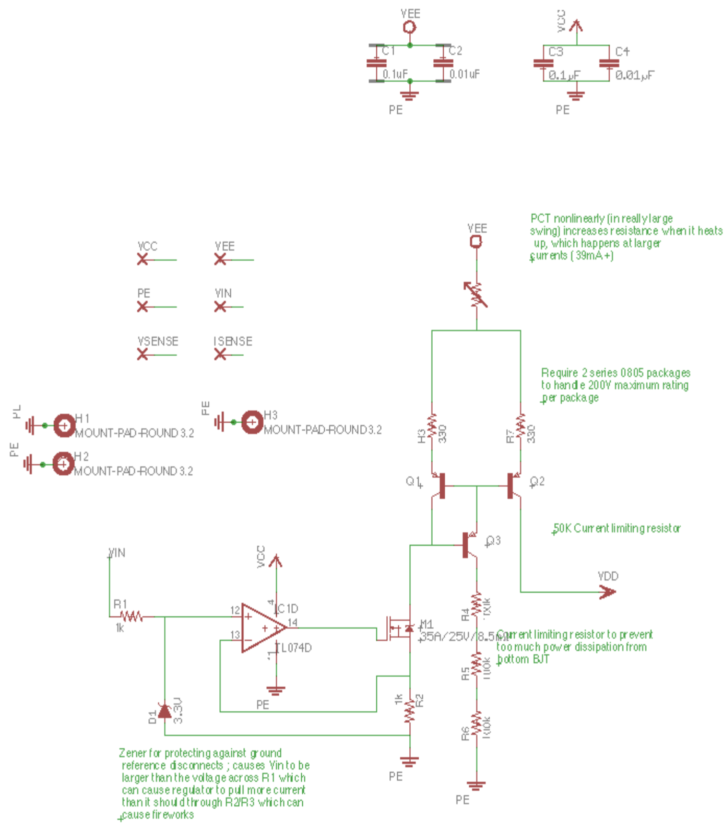


Figure 11 Final schematic design of voltage to current converter

<u>H-Bridge Switching System:</u> The second section is the switching system based on a H-bridge design, which is used to switch the direction of the stimulation current flow through the skin; this is used to toggle between anodic stimulation, cathodic stimulation and also no stimulation. Figure 12 shows the final schematic design for the H-bridge Switching System. The selection of the resistors, capacitors and diodes, and their associated electrical connections, are based on the datasheet recommendations for the associated IC HIGH/LOW side drivers in the H-Bridge Switching System.

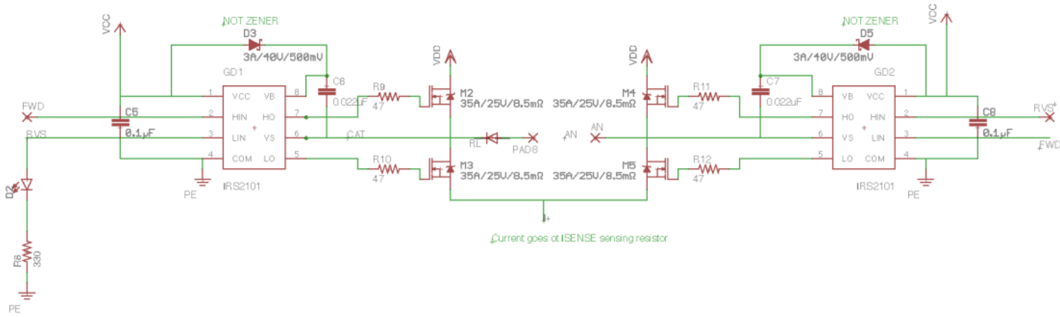


Figure 12 Final schematic design of the switching System

__Output Sensors:__ The output sensors consist of current and voltage sensing circuits to measure the output stimulation current passing through the skin as well as the output voltage across the electrodes. Operational amplifiers are used to design the output sensors, in addition to resistors and Zener diodes (for voltage protection at the analog inputs of the microcontroller). Figure 13 shows the final schematic design for the output sensors, where ISENSE and VSENSE corresponds to the current and voltage sensors respectively.

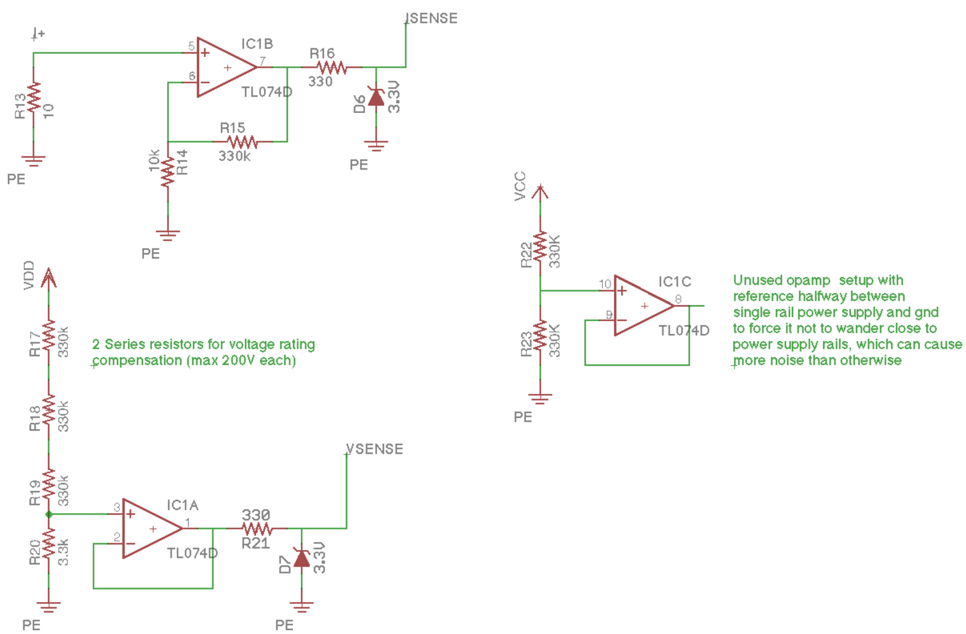


Figure 13 Final schematic design of the output sensors for current and voltage

The final board design of the electrotactile controller is designed using Eagle 8.0, and is shown in Figure 14. Following PCB design guidelines, the analog and digital sections of the design are separated, with majority of the analog ground being on the left-hand side while majority of the digital ground is on the right-hand side. The board design has 3 grounded mounting holes, strain relief holes and required wire pads. The inputs to the controller are located on the top half of the board (8 wire pads), while the outputs are located on the bottom of the board (2 wire pads). The final design is a 2-layer board, with both the top and bottom copper planes assigned as ground pours to maximize heat dissipation and allow for shortest path to ground for traces and components connected to ground. The placement of components, such as the MOSFETS in the H-bridge Switching System, are done such that the trace length and cross-overs of traces are kept to a minimum.

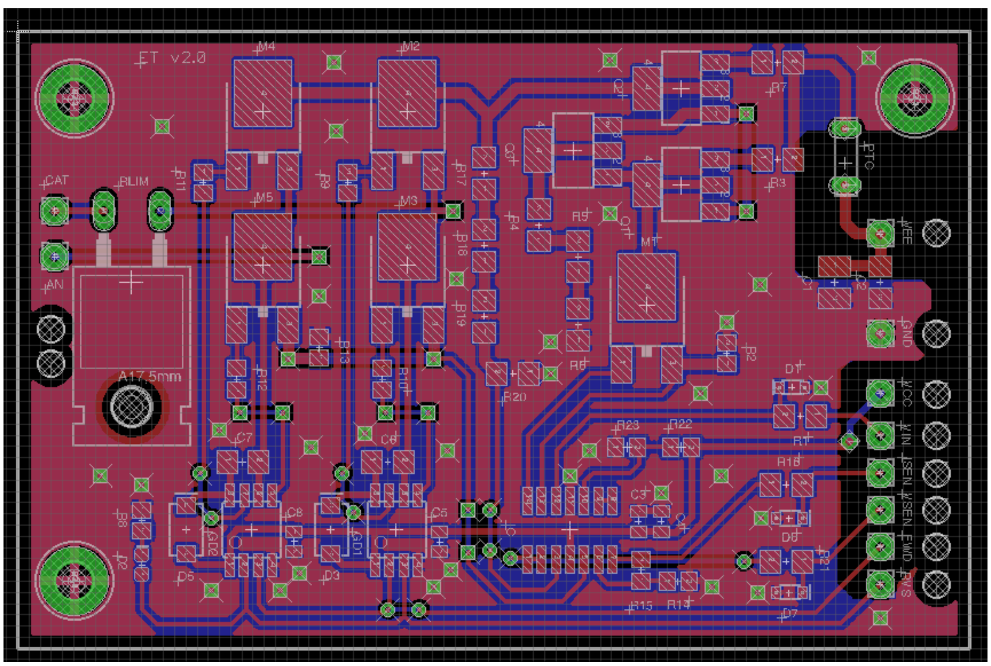


Figure 14 Final board design of the electrotactile controller PCB

Table 6 shows a summary of the components selected for the final design of the electrotactile controller PCB.

Table 6 Summary of final components selected for the Electrotactile Controller PCB

No.	Component	Description	Function	Selection Criteria
1	NDF04N60ZH (ON Semiconductor)	N-Channel Power MOSFET [7] (Package: DPAK)	Voltage to Current Converter, H-bridge Switching System	In the OFF state, the MOSFET used should be able to handle high voltage dropped across it. The breakdown voltage of the selected MOSFET (Drain-Source) is 600 [V], which is above the maximum voltage value of 390 [V] supplied by the high voltage converter in the power module. In addition, this component is rated for 83[W] in power dissipation, which is well above the maximum power dissipated. Hence, the selection of this MOSFET is justified.
2	FZT560TA (Diodes Incorporated)	High Voltage PNP Transistor [8] (Package: SOT223)	Voltage to Current Converter	The collector-emitter breakdown voltage of this selected PNP transistor is 500 [V], which is determined to be sufficient for use in the voltage to current converter since the maximum voltage supplied by the power module is 390 [V]. Also, this component has a maximum power rating of 2[W]. Hence, the selection of this PNP transistor is justified.
3	LM324DT (STMIcroelectronics)	General Purpose Amplifier 4 Circuit [9] (Package: 14-SO)	Voltage to Current Converter, Output Sensors	This operational amplifier provides a wide supply range (3~320 [V] or +/-1.5 to 15 [V]). This dual operation provides sufficient flexibility in designing the supply current waveform for electrotactile stimulation. In addition, the maximum control voltage required in order to provide a maximum of 3.3 [mA] in stimulation current, is 3.3 [V]. Hence, the LM358AN can provide the required control input voltage due to it's high supply range, and also has low voltage offset and input current requirements.
4	IRS2101STRPBF (Infineon Technologies)	IC DRIVER HIGH/LOW SIDE [10] (Package: 8-SOIC)	H-Bridge Switching System	These IC Driver chips are selected to drive the HIGH and LOW gates of the MOSFETs in the H-bridge, which allow for toggling the direction of stimulation current between cathodic and anodic stimulation. The IC requires a input voltage supply of any value between 10~20 [V], which can be satisfied since the Lipo Battery selected in the Power Module can supply is rated for 11.1 [V]. In addition, the IC driver can handle a maximum high side voltage of 600 [V], which is sufficient since the maximum possible high side voltage is 390 [V].
5	B59774B115A70 (EPCOS TDK)	PTC Resettable Fuse 550V [11](Through Hole Radial, Disc)	Electrical Safety for components	The resistance of this circuit protection device increases significantly after a certain threshold or current-trip value, which is 32 [mA] for this component. This is sufficient since if the current rises above a 32 [mA] due to a mechanical or electrical failure, the fuse will protect the remaining components on the

				circuit. The selected PTC is also rated for 500[V], which is sufficient since the maximum output voltage from the power module is 390 [V].
6	MP925-50.0K-1% (Caddock Electronics Inc.)	RES 50K OHM 25W 1% [12] (Package: T0220)	Electrical Safety for User	In order to ensure that the user is protected at all times, this high power resistor is used in series with the electrodes to ensure that the maximum possible current that can pass through the skin is approximately 8 [mA] (390[V]/50[kOhm] ~ 8[mA]). The maximum possible power dissipation in this current limiting resistor is given by: (8 ^ 2) [mA] * 50[kOhm] = 3.2 [W]. Hence, the device is rated for 25 [W], which provides a safety factor of 7.8 approximately.

3.1.4 Electrodes

Due to the use of electrodes for stimulation, and the irregular design constraints of the electrotactile system, custom printed circuit boards were designed as stimulatory electrodes for use on the surface of the fingertips. The final electrode design consisted of a set of exposed tinned-copper electrode rings with a lead-free finish. During the operation of the electrotactile system, the polarity of rings alternates, allowing for two directions of stimulation as current passes through the epidermal layer of skin tissue. Two oppositely oriented light-emitting diodes allow for the independent indication of either current direction. A capacitor between these two diodes absorbs initial high voltage transients when contact with the electrode is first made, protecting the low-voltage rated diodes. The assembled electrode is shown Figure 15, along with the layout diagram for inspection of greater detail.

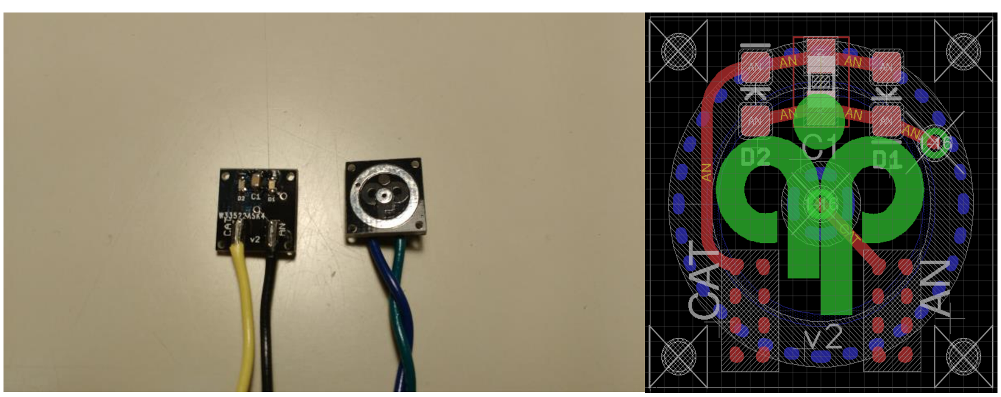


Figure 15 Happen custom electrode assembled design and layout

3.1.5 Mechanical Design for Electrotactile System

Due to the potentially dangerous electrical conditions that a person could be subjected to while in contact with an electrode controller PCB, it was necessary to design an enclosure that would protect someone from being in contact with a high voltage from the 360 [V] source. The PCB designs for the electrotactile controllers also included plated mounting holes connected to the ground plane of the board; these mounting holes allow for the ground plane of the board to be connected to the potential level of the human skin that an electrotactile board enclosure would be in contact with. The purpose of this design mechanism is to ensure that the potential of the human skin does not wander too far from the potential of the ground plane voltage of the boards, which could prevent unwanted electrostatic discharge effects from occurring during temporary contact with the enclosures or PCBs. The SolidWorks design of the pod top cover is shown in Figure 16.

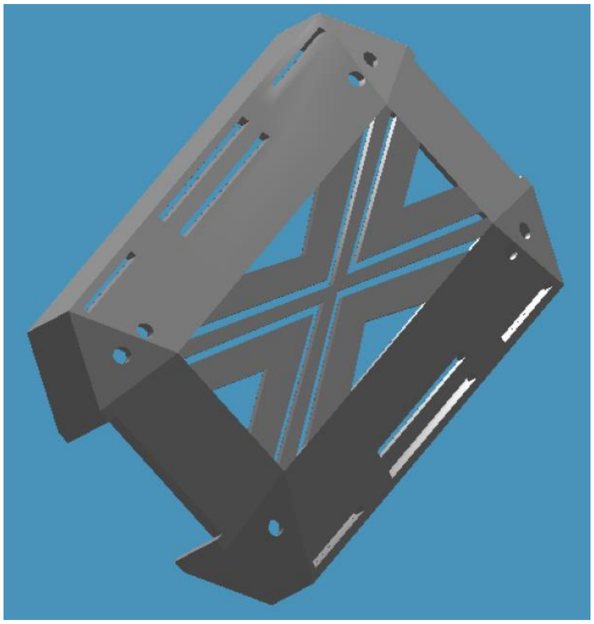


Figure 16 Pod top cover design for the electrotactile controller and microcontroller module

In order to limit the length of free floating wires as much as possible, the team has decided to make the controller pods arm mounted. The wearable pods consists of two electrically connected armbands with three pods each. Each armband is elastically connected with a strip of 1 [inch] elastic band for fitting on various arm thicknesses. This attachment includes 6 pods, where 5 pods enclose 5 electrotactile controllers (one controller for each electrode). The sixth pod contains the microcontroller module and electrical connections from the other pods. All 6 pods consist of the same design with a top cover and a bottom plate. As a result of this arrangement, only 5 connections were needed between the wearable and the base station.

There are four holes in the pods that allow single M3 screws going from the bottom plate, through the PCB mounting holes, to the top cover, where it is fastened with a nut. The triangular cut-outs on the top cover facilitates with heat dissipation as well as allowing the team members to see the led status on the PCB for debugging purposes.

3.2 Design Lifecycle

The design lifecycle for the electrotactile system is well represented by the following product lifecycle management diagram Figure 17.

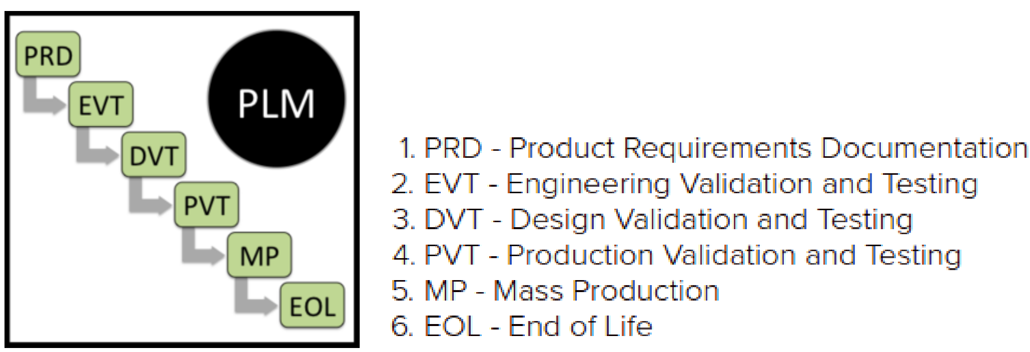


Figure 17 Product lifecycle

In reference to the product lifecycle, there were four discrete designs which were completed, in sequence, through the product lifecycles: the original design, a complete system design on a breadboard, a first PCB design, and a second PCB design.

The original design existed in the PRD and EVT stages of the PLM process, during which the product requirements were specified and subsystems were built and validated separately. In this stage the switching circuits were designed and tested for the current direction control module, and the current regulation function was validated.

The breadboard design existed through the EVT and DVT stages, with initially separate validation and testing of subsystems and then integrated functional validation. At the end of this stage the controller was integrated with electrodes and the electrotactile system was validated to have satisfied design requirements.

The PCB iterations built off the successfully validated breadboard design and were carried through DVT, PVT, and MP stages. The first PCB design was the first system to be made with surface mount components; the operation of these components had not been validated before and in the production validation stage, it was discovered that a few resistors, BJTs, and MOSFETs were out of spec and the power ratings on the devices were being exceeded, causing the system to fail validation. For this reason, the system returned to the DVT stage and began a second iteration of PCB design. This iteration was completed with high power rating components and successfully passed the product validation stage. These boards became the final design and entered the mass production cycle.

3.3 Modifications from Original Design

This section discusses the various modifications made from the original design for the electrotactile system proposed in the final design proposal report submitted earlier in the 4A term. The functional areas of change include electrical safety, control system design, current regulation, switching regulation and electrode design

3.3.1 Electrical Safety Design Modifications

The original design for the electrotactile system had no current limiting resistor at the output of the system to the electrodes. Hence, theoretically, for the absolute worst case scenario, assuming a short from the high voltage supply to the skin, the maximum current that can be drawn is 200 [mA] (maximum output current from power module). In order to provide guaranteed electrical safety to the user, a 50 [kOhm] resistor rated at 25 [W] is connected in series with the electrodes to limit the maximum possible current that can pass through the skin to approximately 8 [mA] (rounding up from 7.8 [mA]). The calculations for the maximum permissible current and power dissipation can be found below:

Maximum current through skin = 390 [V]50 [k] = 7.8 [mA] 8 [mA] (rounding up)
Maximum current through skin = (8 [mA])2*50 [k] = 3.2 [W]

The "let-go" current range is observed to fall between 9-30 [mA] DC, which is the current range in which an individual may lose control of his/her muscles [13].Since the current is limited to 8 [mA] DC in the worst possible case, our final design ensures to be below this "let-go" range and hence protect from the user from harmful shocks or injuries in case the system malfunctions due to a mechanical or electrical failure.

In addition to providing electrical protection for the user, a PTC (Positive Temperature Coefficient) based resettable fuse is used to provide overcurrent and short-circuit protection for the electrotactile system. This was not present in the original design and is added in the final design connected between the high voltage supply (VEE) from the power module and the current mirror in the voltage to current converter. The selected PTC has a trip current of 32 [mA] and an operating current of 16[mA], which is sufficient to protect the components selected for the final design. The PTC functions by changing from a low-resistance to a high resistance state in response to an overcurrent, which is referred to as "tripping" of the overcurrent protection device [14].

Figure 18 below shows an example of how the resistance of PTC rises sharply close to the "trip-point", where the increasing current leads to increasing temperature, and hence, increase in resistance of the PTC.

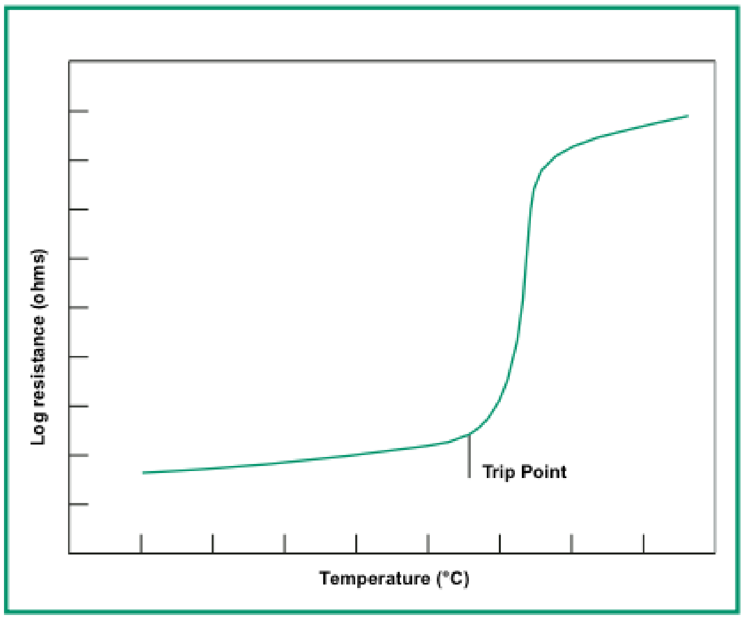


Figure 18 Response of PTC device with respect to temperature [14]

The final design also involved 2 additional electrical safety design considerations. Firstly, a 2 [A] fast-blow glass fuse (enclosed in a cylindrical fuse protection containers) is connected in series with the 11.1 [V] Lipo battery supply in the power module. This ensures that input power supply is protected from overcurrent due to short-circuit. The second design for electrical safety involves the use of a relay to electrically isolate the ground from the stepper motor controller system and the ground from the electrotactile system. Since the stepper motor system is powered by a wall power supply, it is essential to isolate the ground connections of the 2 systems during integration.

3.3.2 Control System Design Modifications

The original electrotactile controller module design operates in open-loop; the system receives an analog input voltage and accordingly regulates a high voltage supply to produce a proportionate magnitude of unidirectional current. Due to the inability of this described system to be able to monitor the magnitude of current that might potentially travel through living human tissue, it was desired to provide some form of feedback as an output from the electrode controller boards. This feedback allows for an input generating system to tune its generated input in order to achieve an output stimulation in the range of acceptable parameters. The feedback parameters required for design were chosen according to Ohm's law; due to the nature in which the impedance between the two electrode poles could change while in contact with live human skin tissue, it was desired to be able to tell exactly how large this impedance would be. This required the measurement of two variables, voltage across the electrode poles and current flowing from one pole to the other.

Analog voltage and current measuring circuits were required to convert the high voltage source for the electrodes to a low voltage input to the 3.3v microcontroller for measurement and processing. Due to the need for these measurement circuits to be located on the electrode control boards, it was necessary to design simple custom measurement circuits. These circuits are shown in Figure 19.

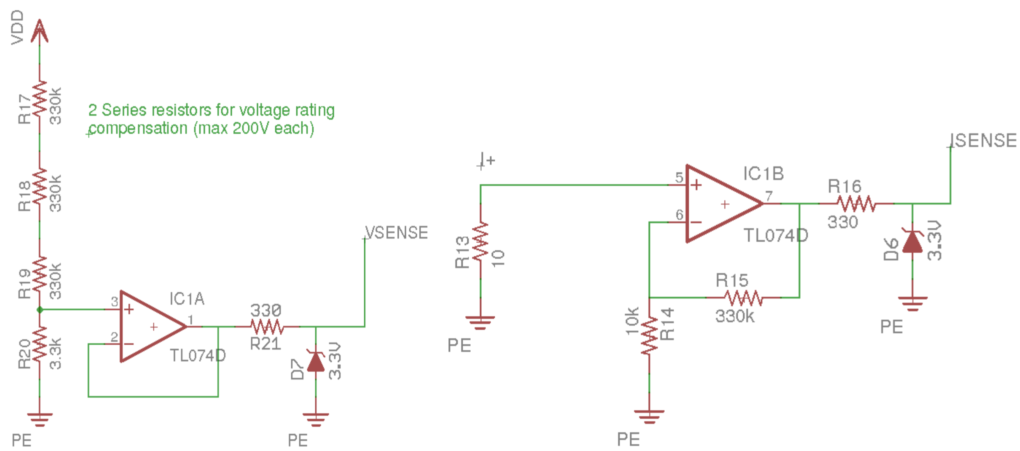


Figure 19 Voltage and current measuring circuits

The voltage measurement system was designed to be a simple resistor divider with an op-amp buffer stage and a regulating Zener diode circuits for protecting a microcontroller from being exposed to high voltages while measuring the electrode voltage. Resistor values for the divider were chosen according to an expected 360 [V] at the input and the maximum accepted 3.3 [V] at the output. The following equation was used to select resistor values.

$$V_{sense}=R_{20} * V_{DD} / (R_{20}+ R_{19} + R_{18}+ R_{17})$$

Arbitrary resistor values of 3.3 [kΩ] and 330 [kΩ] were chosen to achieve a V_{sense} with a maximum output of 1.2 [V].

The current measuring circuit was designed according to the following equation.

$$I_{sense}=R_{13}(R_{15}/R_{14}+1)I_{+}$$

The input current magnitude was designed to be limited to 8 [mA]; readily available resistors shown in the diagram were chosen to limit the output voltage at 8 [mA] to

2.72 [V]. This measurement circuit also includes a Zener diode voltage regulation circuit for microcontroller GPIO voltaging limiting at 3.3 [V].

In addition to these measurement circuits being designed, an isolated input signal mechanism was required such that electrical isolation could be achieved between the stepper motor controller, which coordinates stimulation signals with the VR system and is powered from a wall power supply, and the electrotactile microcontroller which is powered from a battery. Due to the availability and simple control required for relay breakout boards, a system such as the one below in Figure 20 was implemented.

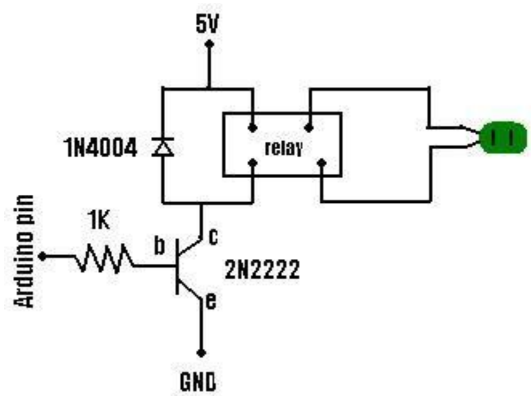


Figure 20 Relay isolated control scheme [15]

This isolation system is operated by the stepper motor control system when a stimulation is desired in the electrotactile system; the electrotactile system sends a ‘turn-on’ stimulus signal to itself which is modulated by the relay to produce stimulation at desired points in time when a user is grabbing a virtual object inside the virtual environment.

3.3.3 Current Regulation Design Modifications

The original design of the voltage to current converter is shown in Figure 21. In the original design, large resistance values were used at R2 and R3, specifically 10k

[Ohm] each. The intention of using large resistance values in the original design was to limit the current passing through the current mirror. However, with the addition of the 50k [Ohm] (25[W]) current limiting resistor in series with the electrodes across the skin, the current is limited to 8[mA] at the output and electrical safety for the user is satisfied. Hence, large resistance values at R2 and R2 are not required in the final design.

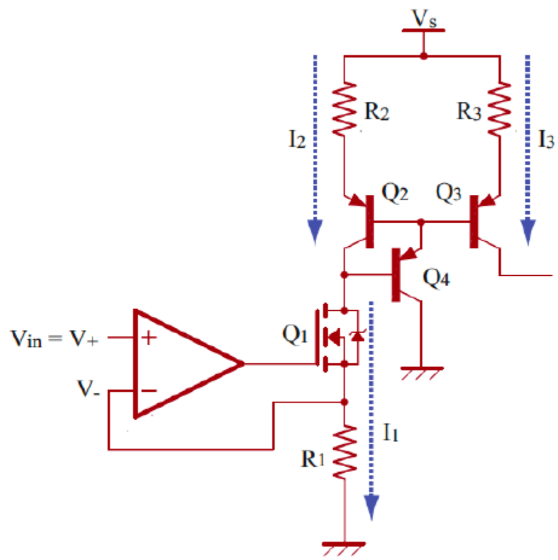


Figure 21 Original design for the voltage to current converter [16]

In the final design (as shown in Figure 22), the values of R2 and R3 are changed to 330 [Ohms] respectively (R3 and R7 in Figure 22), which also ensures that there is no significant voltage drop across the resistors and power dissipation in R2 and R3 is also reduced. The value of R1 (R2 in Figure 22) is also changed from 100 [Ohms] to 1k [Ohms] to limit the driving current in the current mirror to 3.3 [mA], since the maximum possible voltage for VIN is 3.3 [V]. This is due to the fact that the microcontroller cannot produce more than 3.3 [V] at it’s analog output pin, which is connected to VIN. A Zener diode rated for a reverse breakdown voltage of 3.3 [V] is also added at the non-inverting input of the op-amp to ensure that voltage level does not exceed 3.3 [V], if the microcontroller ground and the analog ground disconnect due to some mechanical/electrical fault.

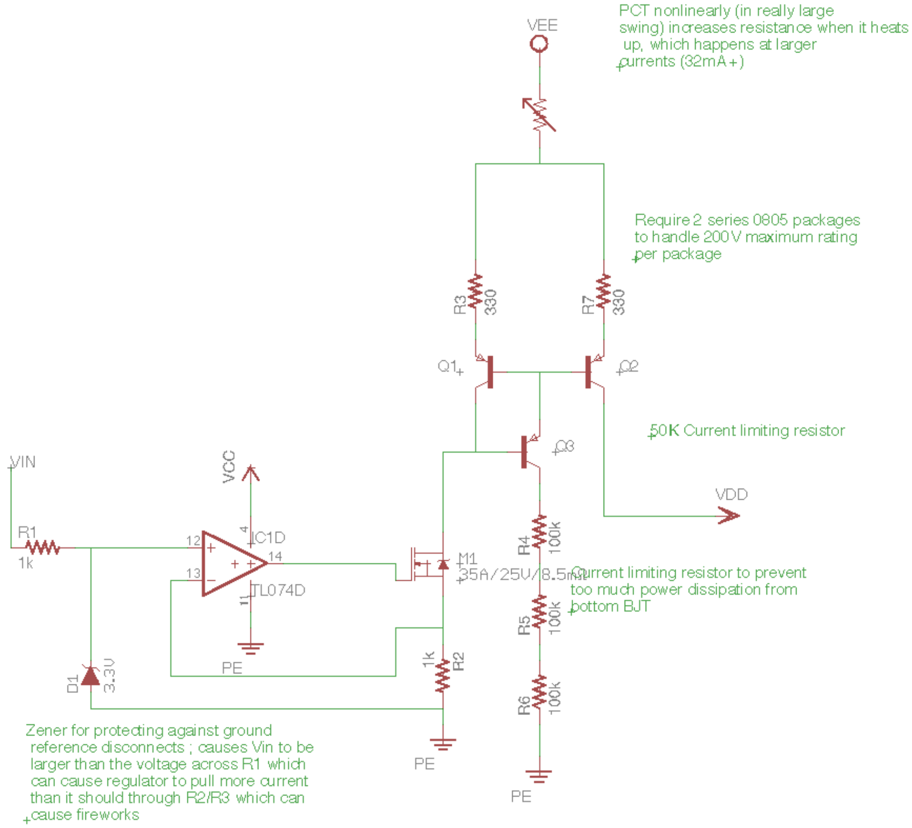


Figure 22 Final design for the voltage to current converter

As seen in the final design in Figure 22 above, another change made is the addition of 3 current limiting resistors (R4, R5, R6) at the collector of the transistor Q3. These resistors are 100k [Ohm] each, limiting the current to 1.3 [mA] when the output is high impedance. This design change provides 2 advantages compared to the original design with no resistors connected to the collector of Q3. Firstly, these resistors limit the power dissipation in the transistors when the load is high impedance, protecting these transistors from exceeding their power rating of 2 [W]. The second advantage is that the current-limiting resistors also limit the energy loss when the load is high impedance, where the skin is not in contact with the electrodes and the load is an open-circuit. Hence, this design modification makes the electrotactile system more power efficient.

3.3.4 Switching System Design Modifications

The switching system is responsible for alternating the direction in which the current travels through the cathode and anode poles of the electrodes being driven by the electrode control system. The major design challenge in the design of the switching system was the conversion from CMOS 3.3V logic levels form a microcontroller to the 360V control levels required for switching the MOSFETS. The original switching system was designed per Figure 23 below.

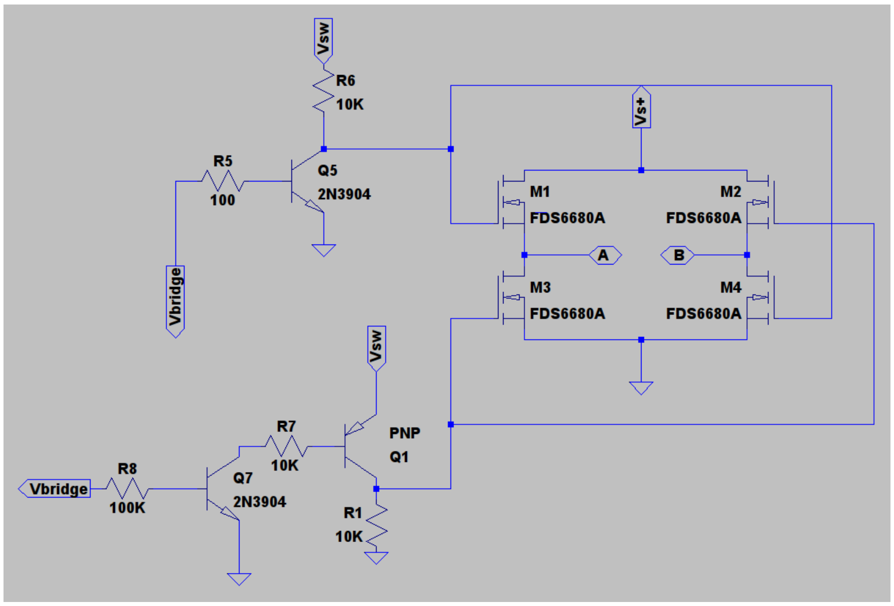


Figure 23 Original design switching system

This system was proven to be flawed, as the single NPN BJT gate driver circuits would only function as low side drivers, not high side drivers. Additionally, the Sziklai configuration switches at the bottom of the figure were represented as low and high side switches when they would realistically only be used for driving the high side MOSFETs.

In the breadboard design stage, these switching mechanism flaws were fixed towards a proper functional design, producing the circuit visualized in Figure 24. Snubber circuits were appended to the design between the Sziklai outputs and the gates of the high side MOSFETs to reduce the turn off time but "snub" or attenuate the turn-on spike magnitudes by increasing the turn-on time.

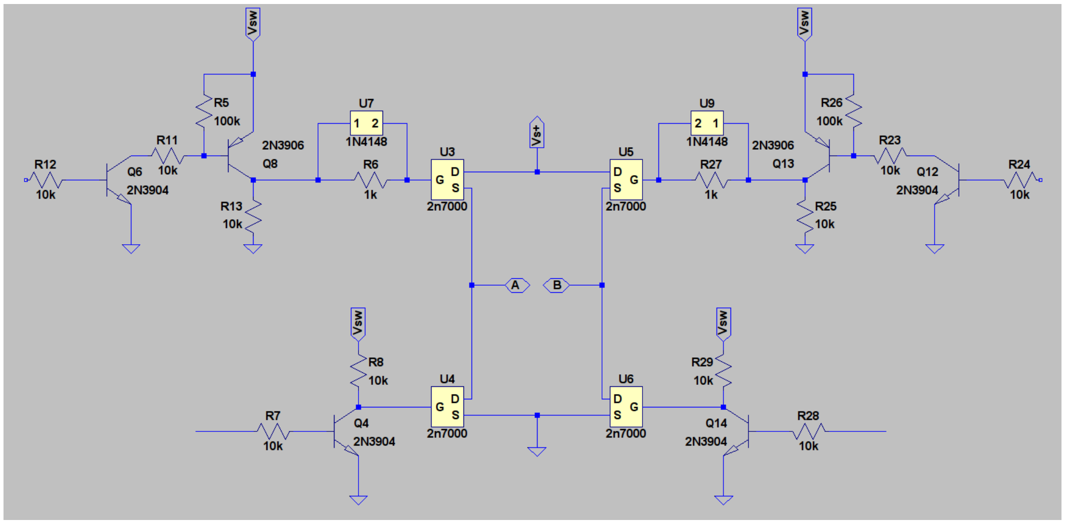


Figure 24 Sziklai configuration switches with snubber drives

While in simulation these systems functioned appropriately, the Sziklai switches had to be supplied with a voltage that was greater than the voltage being dropped across the load between A and B in the Figure, otherwise the gate-source voltage of the MOSFETs would not rise above 0 [V] and they would not turn on. To solve this problem, a custom voltage bootstrapping circuit was designed to raise the Sziklai driving voltage above the load driving voltage. This circuit is shown in Figure 25 and its output is shown below in Figure 26.

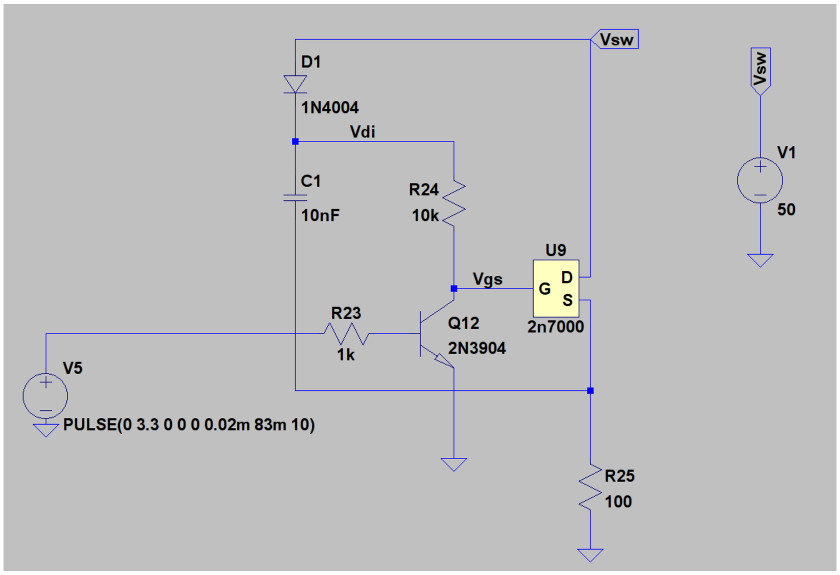


Figure 25 Bootstrap circuit

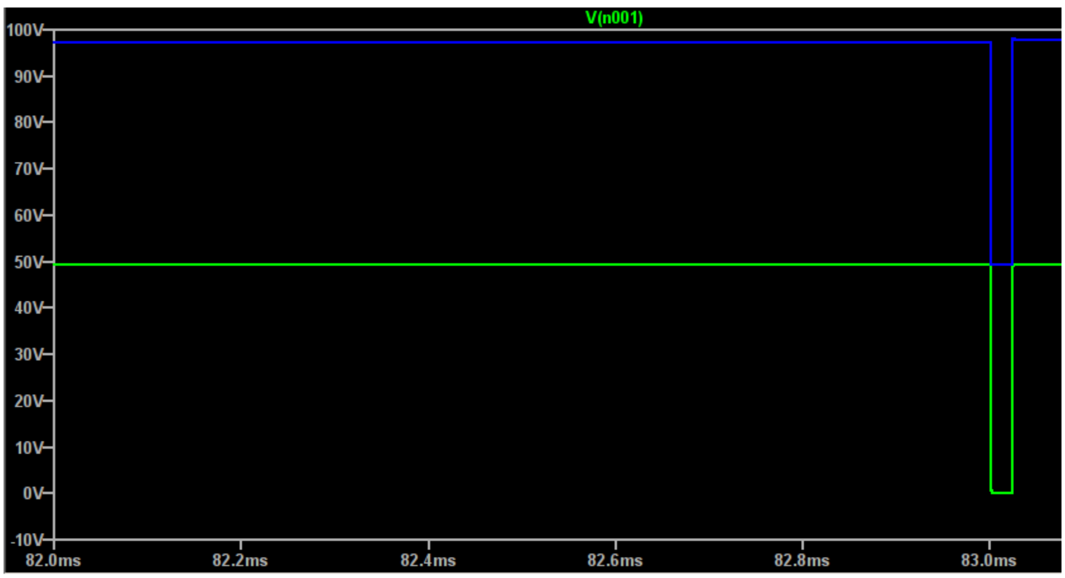


Figure 26 Bootstrap circuit output across load (Green) and bootstrap voltage supplying the MOSFET gate (blue)

It can be seen from the above figure that the bootstrap driving design accomplishes the generation of a MOSFET gate voltage that is higher than its source voltage. A significant cost of this design concept, however, is complexity: the number of components required to produce two MOSFET driving circuits per electrode controller module would be very high, and thus the monetary cost of production would also be high. To solve this problem, the 'do it yourself' design mantra was abandoned and a discrete MOSFET gate driver integrated circuit was selected as the objective gate driver. The IRS2101 was selected as a well-performing driver IC due to nanosecond-scale rise and fall times, and its application schematic is shown below in Figure 27.

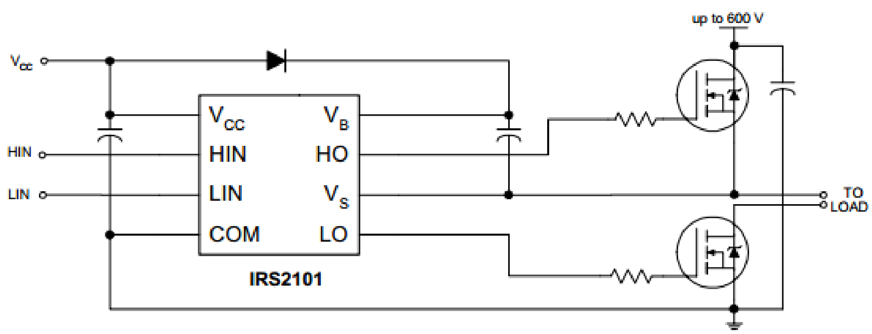


Figure 27 IRS2101 N-MOSFET half-bridge driver IC [10]

This driver IC is essentially the previous custom circuits packaged into one discrete IC; the bootstrap functionality is tuned by selecting the proper capacitor between the V_b and V_s lines. This tuning was performed per the bootstrap capacitance design equations documented by International Rectifier in application note AN-978 [17] and shown below.

$$C \geq \frac{2[2Q_g + \frac{I_{qbs(max)}}{f} + Q_{ls} + \frac{I_{Cbs(max)}}{f}]}{V_{cc} - V_f - V_{LS} - V_{Min}}$$

Qg = Gate charge of high-side FET

f = frequency of operation

ICbs (leak) = bootstrap capacitor leakage current

Iqbs (max) = Maximum VBS quiescent current

VCC = Logic section voltage source

Vf = Forward voltage drop across the bootstrap diode

VLS = Voltage drop across the low-side FET or load

VMin = Minimum voltage between VB and VS.

Qls = level shift charge required per cycle (typically 5 nC for 500 V/600 V MGDs and 20 nC for 1200 V MGDs)

The equation shown above optimizes the capacitance such that it is not significantly depleted, after being fully charged, once the gate of the MOSFET is switched from off to on. Using this equation, the bootstrap capacitance was chosen to be 22 [nC]. Resistors with 47 [Ω]impedance were chosen for gate drive, based on nominally acceptable resistances for protection against MOSFET gate overcurrent damage.

To drive two electrodes, satisfying the bidirectional stimulation design requirement, the electrotactile electrode controllers included two of these discrete gate driver circuits.

3.3.5 Electrode Design Modifications

Electrode design was completed in reference to the works of Sato [18] and Kajimoto [16]. Sato discussed the advantages of a dot matrix design of electrodes, and even suggests that force vectors can be represented if the electrodes can be individually stimulated in different spatial patterns and magnitude levels. Kajimoto documents the method of time division multiplexing of an electrode matrix, allowing for the use of one current regulation and driving system in combination with numerous electrode pairs, as shown below in Figure 28.

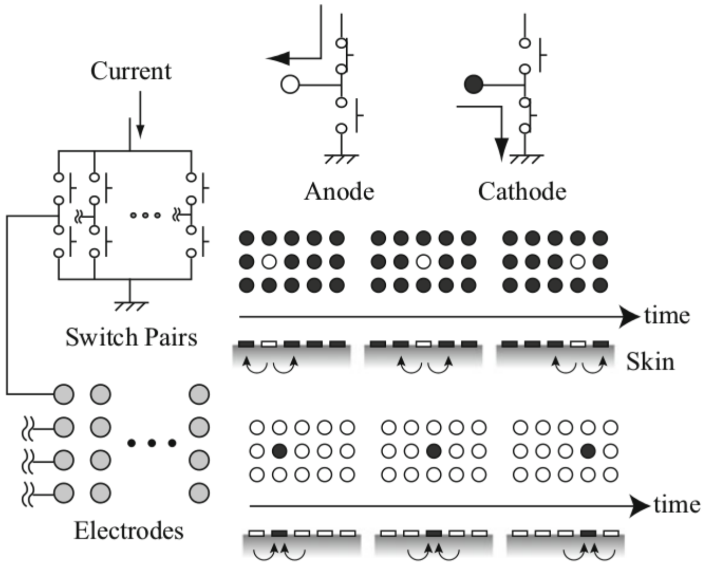


Figure 28 Electrode matrix time-division multiplexing

Sato et al [18] constructed a fingertip electrode built off this multiplexing concept, as shown below in Figure 29.

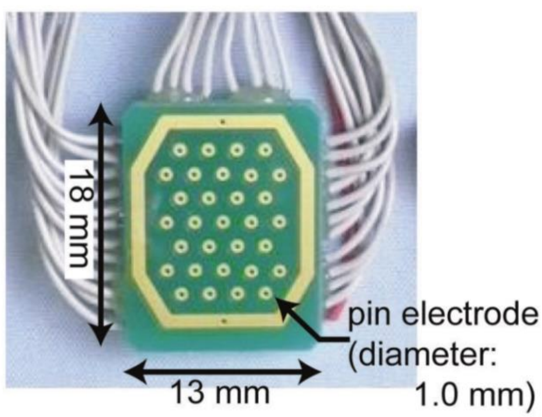


Figure 29 Gold-plated electrode matrix

Due to the lack of accurate models for the relationship between stimulus current density through skin tissue, area of contact of stimulating electrode, and voltage and current magnitude supplied to the electrodes, the design choice was made to design custom dot matrix electrodes with nominal diameters of 1mm. Following the design decision for reduced electrode and controller complexity, it was decided that the electrode designs would feature only 2 poles. This allowed for a single point stimulus with magnitude control, which reduces the required number of half-bridge drivers to two, making this system more affordable to produce. In consideration that a fingertip would contact the electrode from any direction, a concentric ring electrode design was chosen. Since these electrodes would be used for stimulation and not measurement, electrode material selection was flexible and tinned-copper electrodes were chosen for cost reduction.

The electrodes were manufactured with a standard PCB process, and LEDs were included in the design to allow for indication of stimulation direction. The designed electrodes are shown in Figure 30.

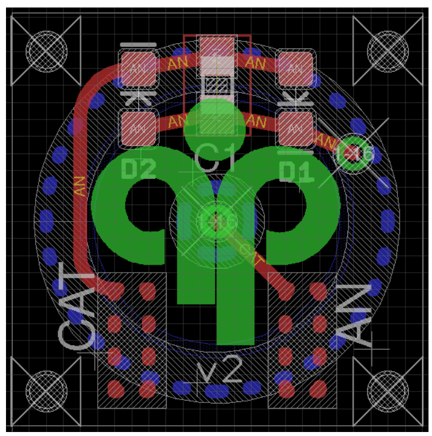


Figure 30 Concentric ring electrode design (rings in blue dotted lines)

The inner and outer electrodes were designed to be separated with a solder mask with a radius of 3mm to prevent electric current from arcing through the air between the electrodes, due to proximity. This 3mm distance was deemed to be appropriate per the IPC-2221B standard on PCB spacing requirements in high voltage conditions. This 3mm spacing was selected based on the IPC-2221B calculator shown below in Figure 31.

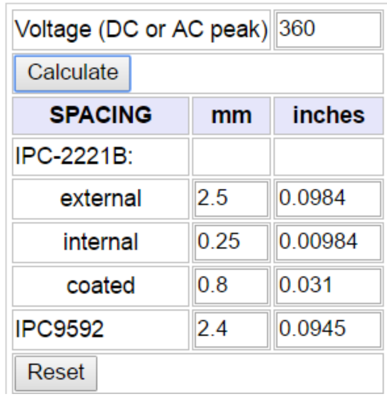


Figure 31 Minimum PCB trace spacing for arcing avoidance [19]

3.4 Manufacturing

This section discusses the manufacturing processes followed to fabricate the electrotactile system, which includes the PCB manufacturing for the electrotactile controllers and the electrodes, and fabricating the pods (mechanical enclosures) for the electrotactile controllers.

3.4.1 PCB Fabrication for Electrotactile Controllers and Electrodes

Table 7 below shows the PCB fabrication specifications that were sent to the fabrication house for the PCB fabrication of the final design for the electrotactile controller. The final board dimensions for the electrotactile controller are 55 [mm] x 85 [mm].

Table 7 PCB fabrication specifications for final design of the electrotactile controller

No.	PCB Specification	Description
1	Layers	2
2	Board Material	Normal FR-4 Board
3	Board Thickness	1.6 [mm]
4	Solder Mask	Red
5	Silkscreen	White
6	Surface Finish	HASL with lead
7	Finished Copper	1 [oz]

Figure 32 shows the final fabricated PCB for the electrotactile controller. It can be observed from Figure 32 that the electrotactile controller board has 3 ground mounting holes, 2 at the bottom and one on the top left corner. These are meant to provide mechanical connection to the encasing pods for the controller boards, but are also grounded so that external electromagnetic disturbances have minimal effect on the electrotactile controller. In addition, the electrotactile controller has 10 strain relief holes associated with the 10 wire pads respectively. By passing the wires to be connected to the wire pads through these holes first, the mechanical stress applied to the solder joint is relieved and the solder joint is less likely to break.

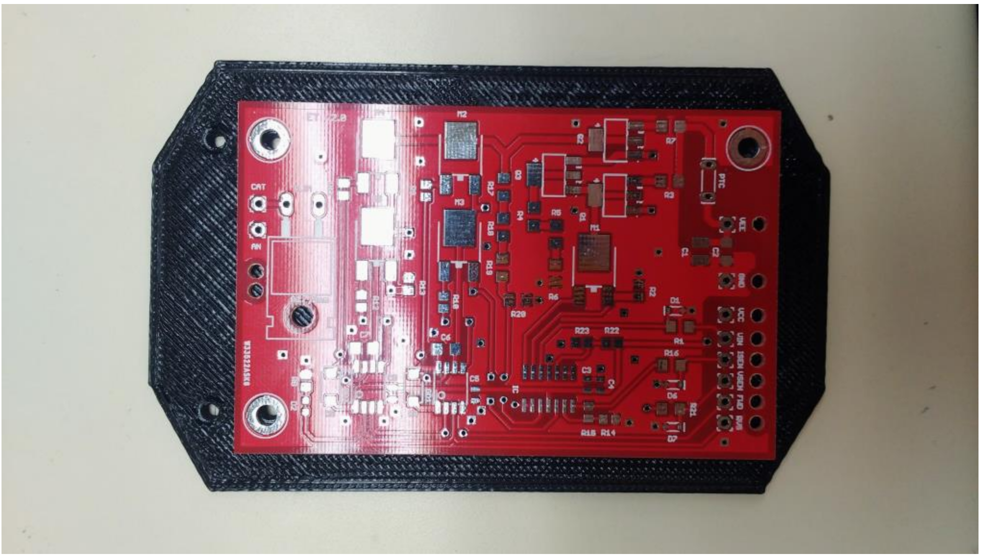


Figure 32 Fabricated final design of the electrotactile controller PCB

3.4.2 Electrode Manufacturing

The electrode PCB design was completed and assembled as shown below in Figure 33. Electrode dimensions were chosen to be 1.1 [cm] x 1.1 [cm] to accommodate placement on any segment of finger.

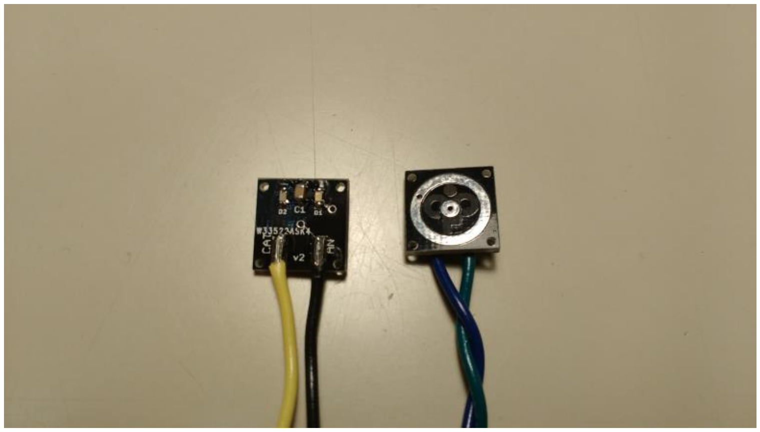


Figure 33 Complete and populated electrodes

Strain relief considerations were made in the design of the electrode PCBs; pads were created for flat coupling of anode/cathode wires, and wires were soldered to the PCBs such that insulation lay below the wire where it rested against the edge of the PCB. Holes in every corner allowed for coupling the electrodes to fabric material using thread. While the original design assumed that the electrodes would be fastened to a fabric glove in 16 different locations, to reduce the design complexity and therefore also cost, five different locations were chosen for electrode placement: one on every fingertip. This arrangement is shown below in Figure 34, where the electrodes have been hot-glued to Velcro loops that allow for adjustment around any location on a finger.

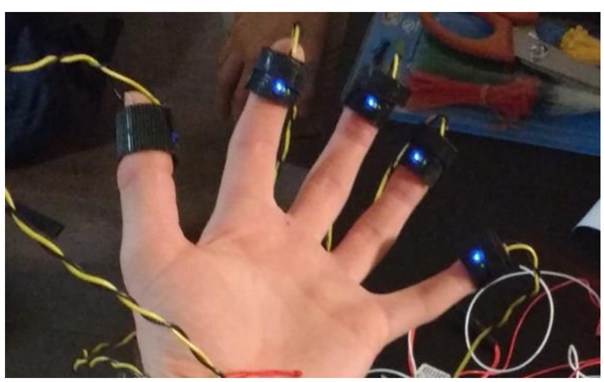


Figure 34 Electrode placements on fingertips

PCB print specifications were chosen as shown in Table 8 below. A lead-free finish was required due to the electrodes being in contact with a living human body.

Table 8 Electrode PCB settings

No.	PCB Specification	Description
1	Layers	2
2	Board Material	Normal FR-4 Board
3	Board Thickness	1.6 [mm]
4	Solder Mask	Black
5	Silkscreen	White
5	Surface Finish	Lead-free HASL
5	Finished Copper	1 [oz]

3.4.3 Pod Fabrication for Electrotactile Controllers

Manufacturing of the pods was achieved through 3D printing the CAD designs using white ABS and black PLA material with XYZ Davincii 2.0 printers. The pods are 3D printed using 1.75 [mm] spools. Later, the pods are painted white to match the white elastic bands. The resulting armbands is shown in Figure 35 below.

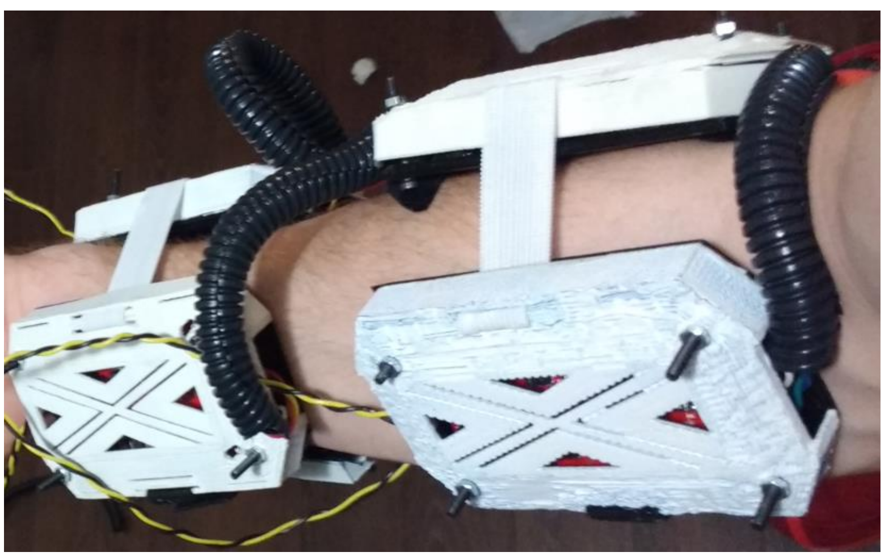


Figure 35 Electrotactile controller pods fully assembled and worn around a forearm

To avoid wire tangles, the team used zip ties and flexible PVC wire sheaths for the wires between the pods and the wires from the arm bands to the base station.

3.5 Commissioning

For the electrotactile system, the commissioned tasks include the PCB manufacturing for the electrotactile controllers and the electrodes, and the 3D printing of the pods for the electrotactile controllers. The required PCB fabrication was performed by PCBWay, a PCB manufacturer based in China. For manufacturing the pods, the XYZ Davincii 2.0 at the WatiMake Lab (University of Waterloo) was used to perform 3D printing. The process of manufacturing was split into different tasks and were assigned to different team members. Ben and Sunaal were responsible for the design and manufacturing of the electrotactile controllers. Si Te and Ben handled the design and fabrication of the electrodes, and they were also responsible for the final fabrication of the pods. Sunaal and Ben were also responsible for populating the electrotactile controllers, building the control module and power module circuits, and final testing of the populated electrotactile controllers and electrodes. In terms of the manufacturing timeline, the first PCB version of the electrotactile controller was fabricated and tested, followed by the fabrication of the electrodes. Drawing from the testing performed for the first version, several design changes were made and the second PCB version of the electrotactile controller was fabricated. Once all 5 required electrotactile controller boards were populated and tested successfully, the pods were fabricated. Lastly, the final power module and control module were built with a switch and fuse implemented in the power module.

3.6 Testing and Performance

The electrotactile system was validated in separate stages, and for each stage, each subsystem (regulator, switches) was tested for performance in comparison to simulations.

3.6.1 Simulations

The current regulator simulation schematic is shown below in Figure 36 and the simulated current output of the circuit (green) through R4, in comparison to the input current through R1, is shown below in Figure 37.

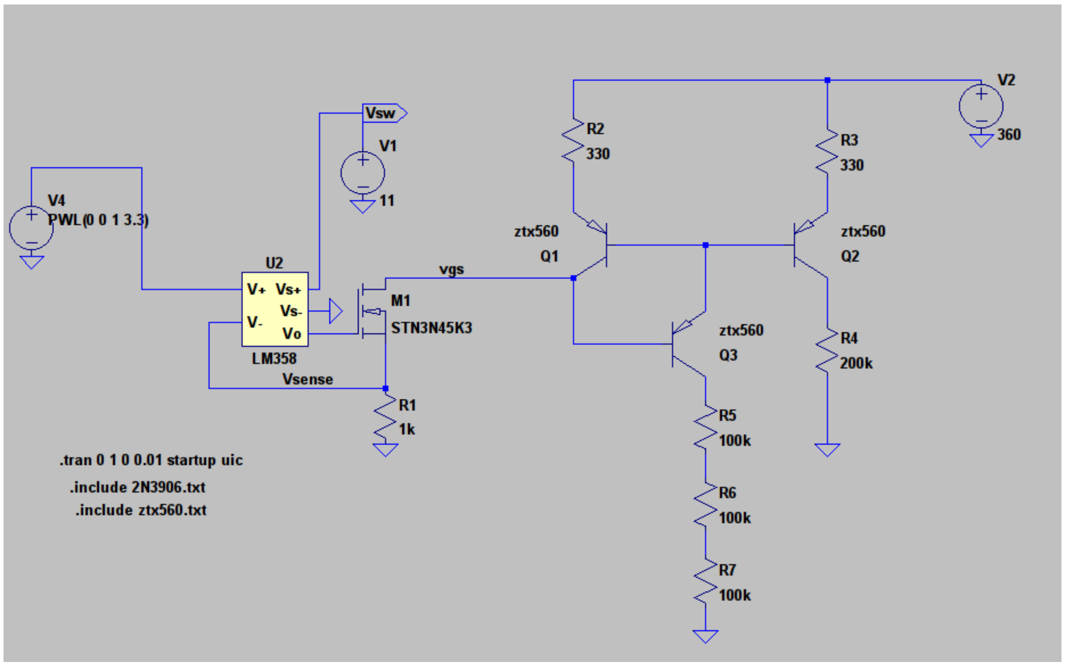


Figure 36 LTSpiceIV simulation of current regulator system

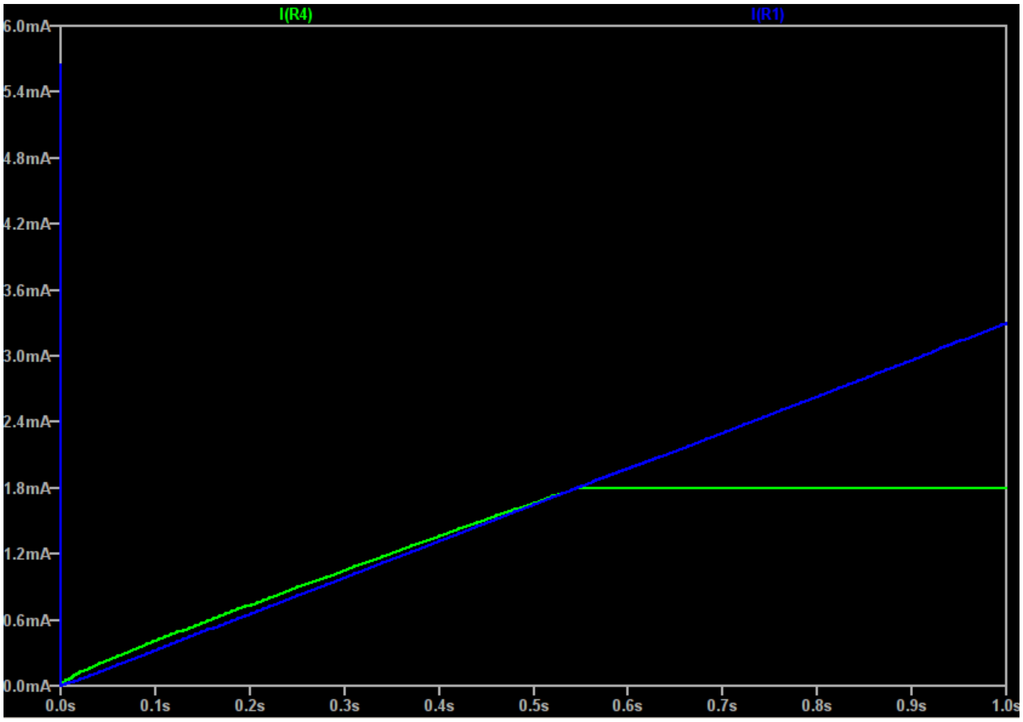


Figure 37 Output of current regulator simulation

It can be seen from the simulation above that the regulator is imposing a hard limit to the current magnitude that can pass through the load resistor R4. this hard limit is held at 1.8 [mA] for the 200[kΩ] resistor, supplied by a 360 [V] source.

The switching circuit was also simulated, with the MOSFET switching behaviour being evaluated for the representative IC MOSFET gate driver circuit in simulation. This circuit and the resulting switching waveform are shown below in Figure 38 and Figure 39.

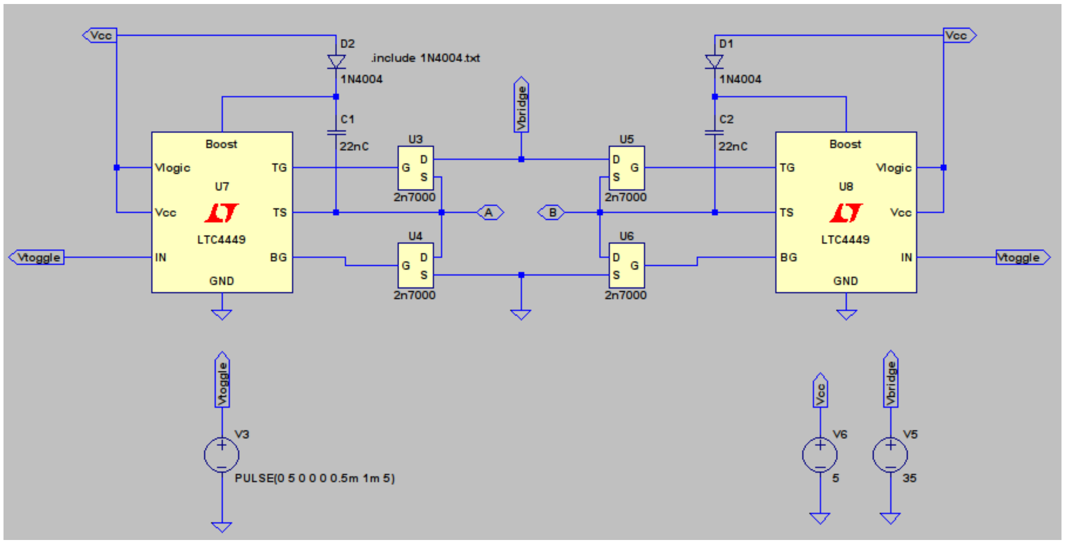


Figure 38 Representative IC driver switching circuit

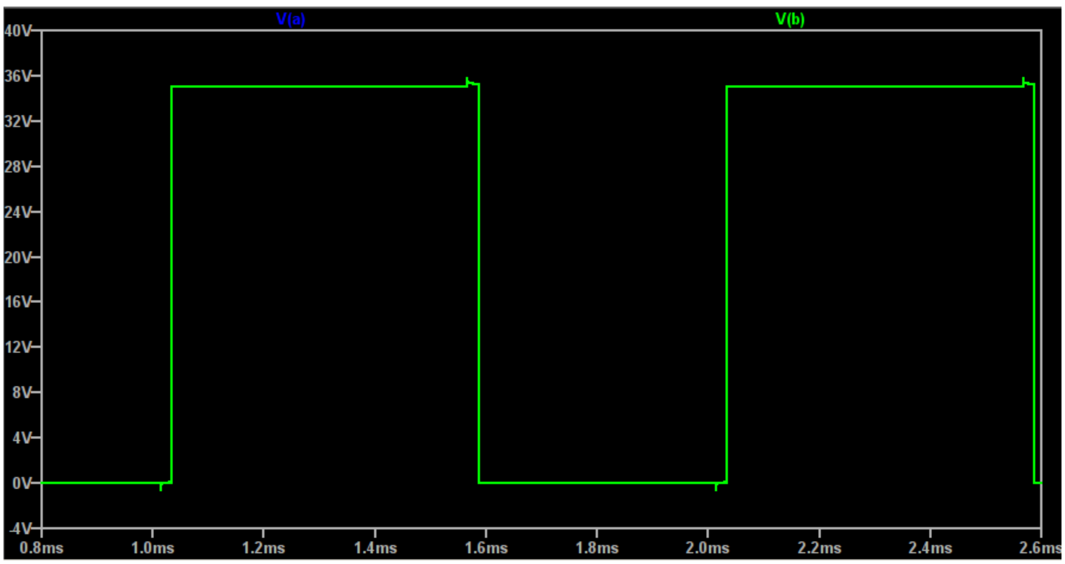


Figure 39 Output voltage at port A from representative switching circuit

From the simulations, it can be expected that the square wave stimulation waveforms supplying a resistive load will have no large distortions, with small switching transient effects at the rising and falling edges of the waveforms.

3.6.2 Functional Validation

The operation of the switching circuit was validated in operation in both directions and the output waveform across a purely resistive test load is shown below in Figure 40.

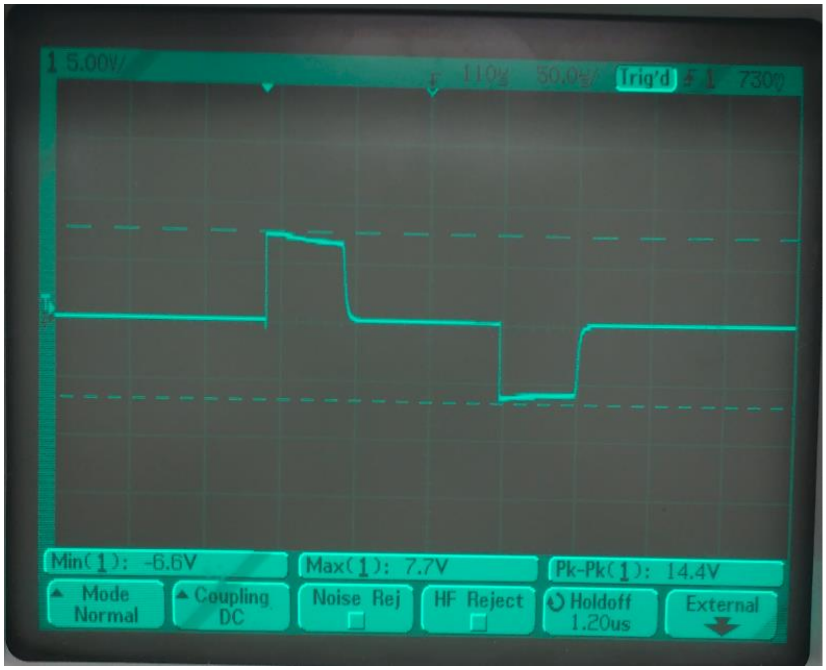


Figure 40 Square wave test waveform from breadboard circuit

It can be seen that the magnitude of the square wave decreases over time; this is due to discharging of the bootstrap capacitor in the driving circuit. This output waveform validates that the circuit produces relatively ideal square waves with low rise and fall times for a pulse width of $50\mu s$.

The current regulator was separately testing and proved to output consistent current magnitudes over a range of resistor values. After this validation step it was integrated with the switching circuit and the system operation was validated while in use with an electrode contacting the fingertip of a test subject. The resulting voltage across the skin of the subject as output from the electrotactile system is visualized below in Figure 41.

Figure 41 Complete system voltage output across human skin tissue

The waveform visualized above was designed to be biphasic with a balanced amount of current injected into the skin between both directions. It is observed that the area under the stimulation curves appear to be slightly off balance; this is due to the non-purely-resistive load effects of human skin. Even though the stimulation waveforms were balanced as they were output from the microcontroller, the capacitance of the skin caused the stimulation waveforms to be drawn out in both rise and fall times. It can also be observed that the ideal square wave edges are dulled; this is again due to the capacitive effect of skin tissue. Relative to the scaling on the oscilloscope output pictured here, the integrated system operation was validated. These results were also validated for the second iteration of PCB design for the electrotactile control modules. Following this validation stage, the system was passed on into a qualification assessment stage.

3.6.3 Functional Qualification

This section discusses the testing performed to investigate and validate the use of electrotactile technology to simulate tactile sensations by electrically stimulating mechanoreceptors located in the skin. For the scope of this project, the desired tactile sensation is limited to a single sensation of pressure through each electrode on the tips of the 5 fingers. It was quickly identified through testing on our own team members that the sensation stronger and more consistent at the tip of the finger than on the remaining areas on the palm surface of the hand. Hence, testing was limited to the tips of the fingers (skin area over the distal phalanx).

Drawing from testing on our team members, a current magnitude of 1.5 [mA] and a duty period or pulse width of 350 [us] is determined to be sufficient to create a tactile sensation that can be perceived and detected by each of our 5 team members when the stimulation is performed over the distal phalanx of the user. In addition, since the goal of the electrotactile system is to recreate a sensation of pressure in this project, the characteristic of the stimulation is cathodic in nature, since this type of stimulation is associated with pressure [18]. The anodic stimulation also exists but has a very low magnitude and a larger pulse width to ensure the net DC current passing through the skin is zero and the ionic balance in the skin is not significantly changed. Hence, since the anodic stimulation is performed only for charge balance in the skin, it does not stimulate receptors that are usually triggered by anodic stimulation since the magnitude of anodic stimulation is very low. In order to produce a simulated sensation of pressure, the frequency of the stimulation waveform is adjusted and subjective feedback from the user (one of our team members) is used to determine the role of stimulation frequency in simulation pressure sensation. Table 9 below shows data collected for 4 major frequency sample points, which highlight a trend of higher frequencies corresponding to a closer “pressure-like” sensation.

Table 9 Tactile sensation testing with respect to stimulation frequency

Stimulation Frequency [Hz]	Cathodic Width [us]	Cathodic Magnitude [mA]	Tactile Sensation Reported by User
20	350	1.5	Vibration sensation is reported where the user can distinguish the timing and intensity of the beats. No pressure sensation is reported.
40	350	1.5	Vibration sensation is reported again but the beats are faster and it is challenging for the user to distinguish the beats.
100	350	1.5	Vibration sensation and a slight pressure sensation is reported, but the user cannot distinguish the beats in time or intensity. Hence, it appears to be closer to a constant beat.
200	350	1.5	A pressure sensation is reported; however, the user can still sense some form of vibration. The user also reports that there is tickle-like sensation under the electrode during stimulation, and hence, the tactile sensation of pressure produced is not very natural.

As seen from Table 9 above, the testing results with the user show that a simulation of pressure can be achieved at higher frequencies (above 120 [Hz]). However, it should be noted that this sensation of simulated pressure is only limited to the distal phalanx of the 5 fingers and does not feel very natural. There were 3 major challenges with testing for tactile sensations using the electrotactile system in this project - skin condition, tactile perception variability and lack of quantitative feedback. Firstly, when the skin becomes sweaty due to constant use or the tight fit between the Velcro, electrode and skin, the amount of current passing through the skin from the electrode decreases. Hence, as the skin condition varies the actual stimulation waveform passing through the skin is also subject to change, since current will seek to take the shortest path and some of it will pass through the fluids, like sweat, present in the skin-electrode interface and hence, some receptors may not be triggered or triggered in a different manner than intended. As a result, the results can be inconsistent due to this constantly changing skin-electrode interface. The other main challenge with tactile testing is the perception of tactile sensation seems to vary frequently with the same user, as well as when comparing different users. This may be due to different thresholds of electrotactile stimulation, skin-electrode condition, stimulation location and the perception and processing of tactile sensations by the central nervous system of the user. Lastly, it is challenging to achieve quantitative feedback about tactile perception since the exact physiology of the receptors and neural processing of tactile sensation by the user is unknown. Hence, this testing phase relied heavily on qualitative or subjective feedback from the user, which can be subject to bias and errors. Hence, there exists a need to develop a relationship between certain measurable parameters, like frequency of stimulation, stimulation current magnitude or certain neural response parameters, and the tactile sensation perceived by the user.

4 Final Design of Software System

The final design and implementation details are explained in depth in this section. The software system includes code for virtual environment, interprocessor communication, and embedded platforms. Finally, the performance of the system is evaluated and described.

4.1 Final Design Details

The software system consists of a virtual reality game and embedded software code that communicates between the game and custom hardware. The virtual reality game is built and run on a Windows 10 gaming desktop with Intel Core i5-6600k CPU running at 3.5GHz and NVidia GTX 1080 GPU. The embedded software on the other hand, was running for Arduino and Teensy microcontrollers. To allow the user to play the game in full immersion, the team used Oculus VR headset as main display and Leap Motion IR sensor for hand tracking. The two hardware can be mechanically combined as shown in figure below.

Figure 42 Oculus and Leap Motion hardware [20]

4.1.1 Virtual Environment

To take the full advantages of a virtual reality environment for patient recovery, the team decided to make a virtual reality game that allows the patient to practice upper-limb artifacts. Unlike a step-by-step recovery tutorial, a game is much more open-ended and entertaining, and therefore increase the engagement of the patient to practice for longer periods of time. This gives the patients types of practices that are not possible in traditional physiotherapy. The game is built using Unity3D game engine for game scene design and all game application logic. The game characters and pillars are modelled in Blender and imported into the Unity IDE (Integrated Development Environment).

The virtual reality game is called Space Rockaway, as shown in figure below. It is a first-person rock defense game against polygon humanoids on a futuristic spaceship. The idea is to allow the users to exercise their upper limb and hand movements by repeatedly taking balls from an altar and throwing them at the humanoids to gain points. The humanoids crawl up from the ground and then run towards the player using Unity built-in AI navigation, and can be killed if the balls make contacts with them. The balls are regenerated on the altar whenever the previous one has been taken away.

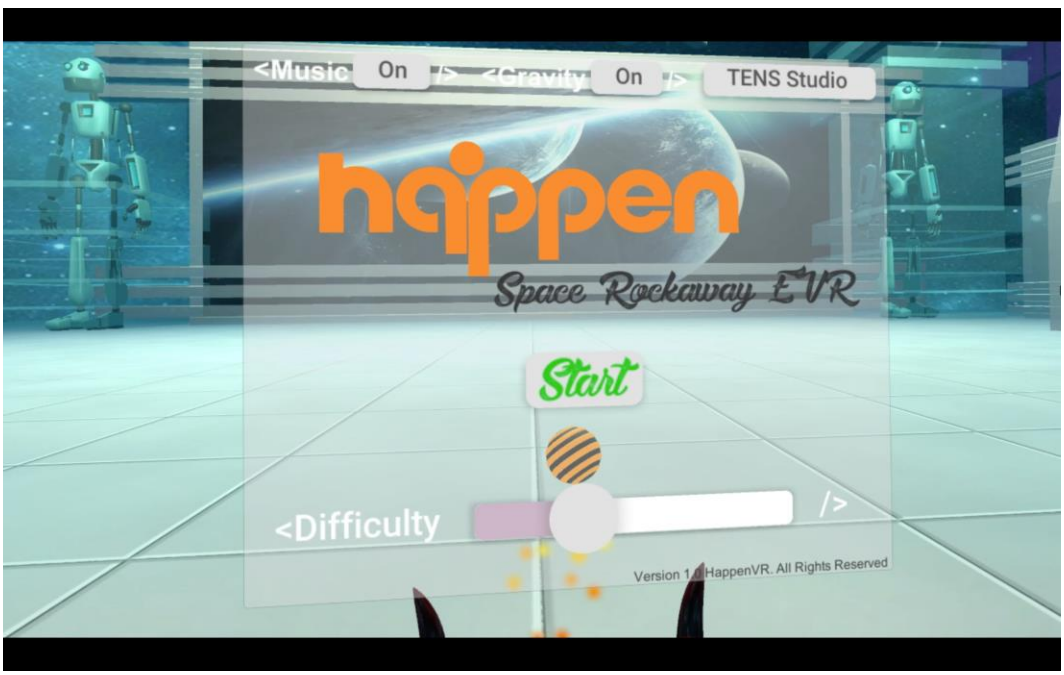


Figure 43 Happen Space Rockaway game main menu

There are several options within the game that can be adjusted in the main game menu, all of which responds to virtual touch thanks to the Leap Motion natural interaction API. The difficulty slider adjusts the speed of approaching humanoids. The gravity toggle enables or disables the gravity of balls. It was found that throwing is simplified by turning off the gravity.

The auxiliary scene in the VR game is called TENS Studio. The is an environment where the team can demonstrate the capabilities of the electrotactile and mechanical systems without the distractions of enemies approaching. TENS Studio can demonstrate the interaction with non-moving spheres of different sizes. A screenshot of the TENS Studio scene is shown in figure below.

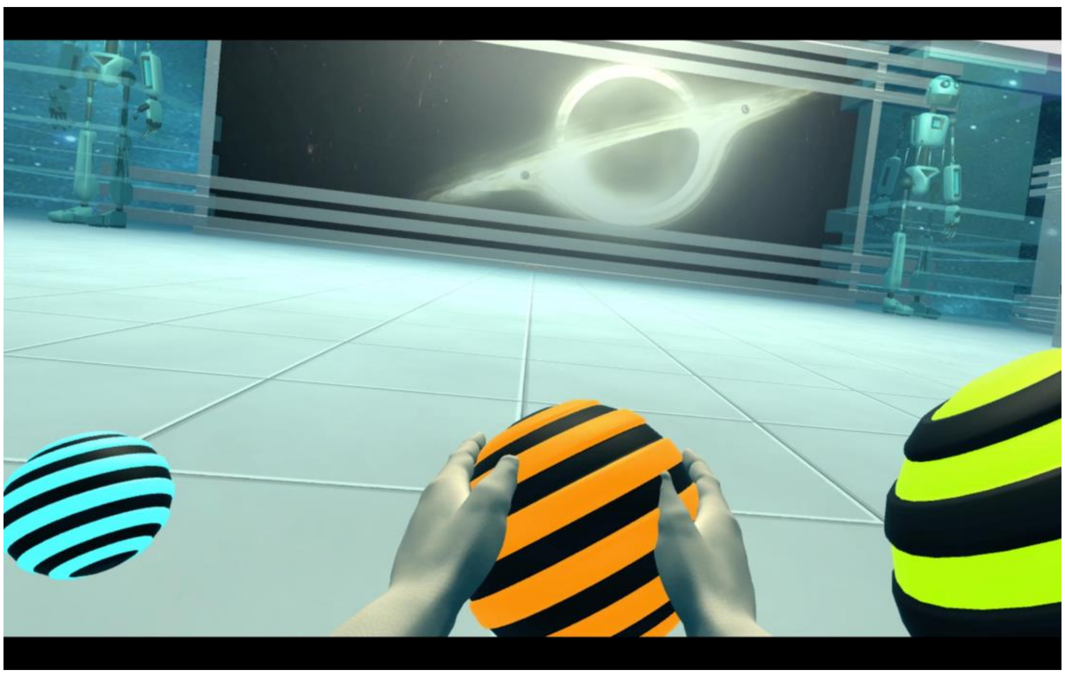


Figure 44 TENS studio environment

The key difference between Happen Space Rockaway and the few other Leap Motion enabled VR games is that Space Rockaway is connected with custom actuators and sensors described in previous sections. The game also fetches each bone position on the hand at any given frame, and uses that information to check if the fingers have collided with an object, along with other relevant details. These collision details add the sensation of touch for the VR hands, which is then replicated on the real user's hand as closely as the custom hardware allows.

4.1.2 Game Objects

The humanoids in the game are fully modelled and animated in the Blender 3D modeling program. Blender allows detailed bone structures to be specified for the humanoid body, which is constructed using armatures. This in turn allows animations to be added to the game character. Animation sequences that were created include crawling from the ground, crawling to running transition, running, jumping, and dying. The figure below shows Blender interface for designing the humanoid model.

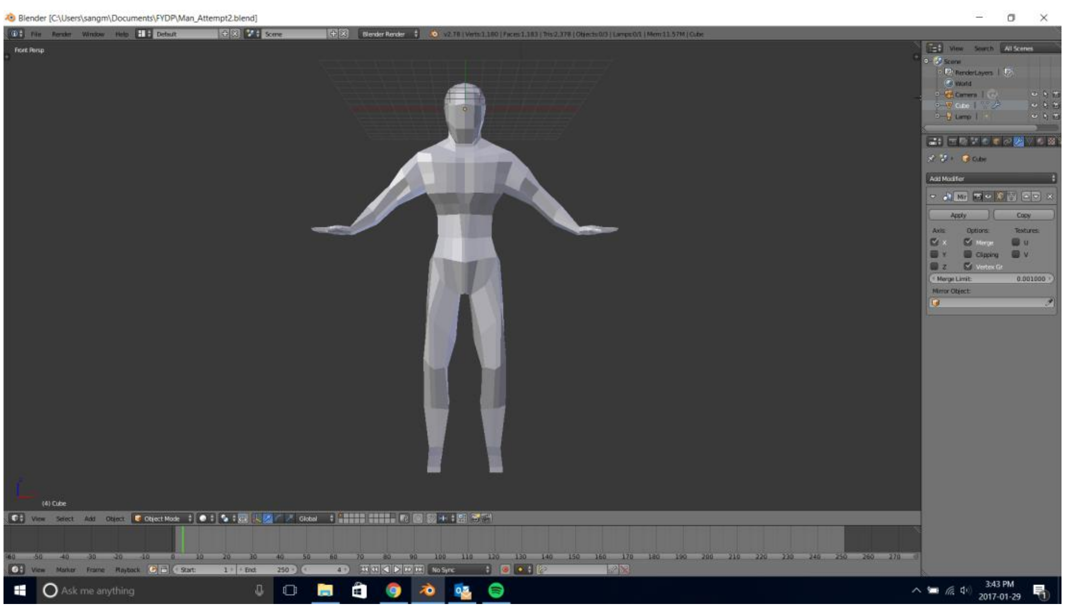


Figure 45 Blender interface for designing the humanoid model

After the model is imported into the Unity IDE, a C# class called SpawnZombie can retrieve the humanoid GameObject and generate any number of humanoids at the desired times. To make the game interesting, a randomized color is assigned to each new instance of the humanoid GameObject, a random location is picked from

10 possible spawning positions, and the speed is adjusted based on the difficulty set in the main game menu.

AnimationController is a game file that defines animation sequences for animatable GameObjects. When used in conjunction with the ZombieMovement class, the humanoid AnimationController is able to link discrete animations into one seamless animating character. However, animations do not imply movement. To allow the humanoid to move along with the running animation, a NavMesh Agent is added onto the humanoid GameObject. This allows the script to specify a goal location and move the humanoid with the specified speed. The Unity game engine takes care of generating the NavMesh map specifying where obstacles should be avoided, and automatically controls the humanoid to traverse through the VR terrain.

At the same time, a cylindrical collider component is added to each humanoid to detect collisions with balls. Adding a MeshCollider component in the shape of the humanoid is simple, but would require more computation per frame, causing a longer latency. A cylindrical Collider component is used to approximate the interaction since the humanoid shape is not as important. Each humanoid is instructed to play dying animation upon detection with a user projectile object. The humanoid is then deleted from the game after a few seconds after dying.

The hand model is another moving object in the game, but unlike the humanoid, it was provided by the Leap Motion API. A lot of the software needed to recognize the hand properly are pre-configured as part of the Orion SDK (Software Development Kit) [21]. The SDK transforms the Leap Motion sensor from the traditional desktop setup to a head mounted setup. Importing the hand model was simple and required few adjustments before it can interact with the ball objects.

The balls are the projectiles that the users grasp and throw. They are seemingly generated from the top of 3 altars with flame flying upwards. The ball GameObject contains 4 major components. It has a natural interaction object script provided by Leap Motion, which allows the ball perform physical interactions with the VR hand naturally. Without this script and using built-in physics instead, grabbing the ball would be almost impossible due to surface interferences. It has a custom InteractableCollision script that adds precise finger tracking abilities needed for controlling the motors and TENS pads. It has a trigger collider for detecting when the finger collides with the ball. Finally, it has a regular physics collider for the ball to physically interact with all other game objects except for the hand. The altar and flames are added around the ball later only for appearance.

Other GameObjects that are seen in the game environment are either custom made in Blender or downloaded from BlendSwap, a website that allow designer community to create and share Blender models free of charge [22].

4.1.3 Software to Hardware Interface

In order to give patients a full immersive rehabilitation environment. Processors from different systems need to communicate with one another to deliver in game interactions to the actuators, as well as relaying sensor data back to the processor. The overall communication interfacing diagram is shown in Figure 46.

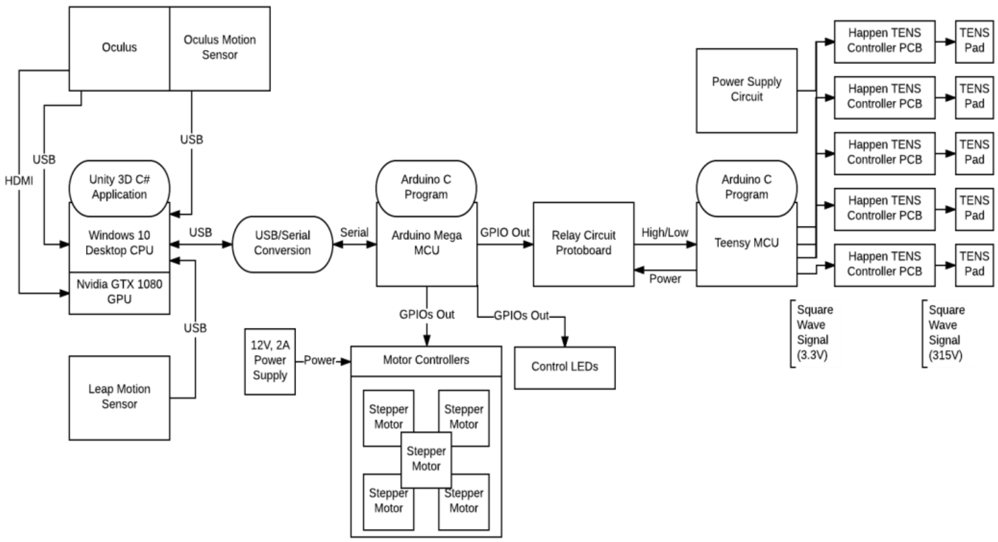


Figure 46 Software system interface diagram

In the nominal use case, the Space Rockaway game first recognizes a hand to object collision, and two functions are then summarily called in the Serial communication class: Motor_Write and TENS_Write. Behind the scenes, the communication class transforms the function calls and parameters into a serialized message in the structure of a byte array, which can then be readily sent to the main Arduino microcontroller. While the USB interface for the purchased VR equipment comes out of the box, serial interface with the Arduino microcontroller was implemented manually on both ends of the communication with the help of the Arduino Serial API and C# SerialPort API.

The central Arduino Mega receives the message and de-serialize it. If the message is to constrain the finger movement, it directly signals the appropriate motor controllers to enable stepper motors from a disabled state. If the message is enable TENS pads, the Arduino outputs a high signal from its GPIO (General Purpose Input Output) pin to the relay circuit as described in the electrotactile system section. Although the current hardware only supports signal being sent in one direction, extending the relay circuit to receive real-time current and voltage measurements from controller PCB to the Unity game would be possible with the current controller PCB design and software. The relay power isolation circuit needs to be revised in that case. Due to time limitations, the team decided not to perform software current and voltage monitoring on the finger tips. The hardware current limiting circuit works as intended.

The Teensy microcontroller receives the same message from the relay circuit— albeit electrically isolated for safety— and generates a corresponding square wave for the appropriate hand sensation. For the symposium, only one sensation configuration is specified for one TENS pad, but the software structure is written to be able to support unlimited types of sensations and number of pads.

The separate Teensy is used for signal generation for two reasons. Firstly, it allows for full electrical isolation between the wall power and the user's hand. Secondly, it will output a continuous signal at the highest resolution possible. The main Arduino has many time-consuming responsibilities already, such as serial communication, which would inevitably degrade the square wave signal and the resulting sensation of the user.

4.1.4 Embedded Software

On the central Arduino microcontroller, three GPIO pins are dedicated for enable, step, and direction on each stepper motor controller. Thus, the only connections used on the motor controllers are enable, Vcc, ground, step, and direction. The controller is Big Easy Driver from Sparkfun shown in Figure 47.

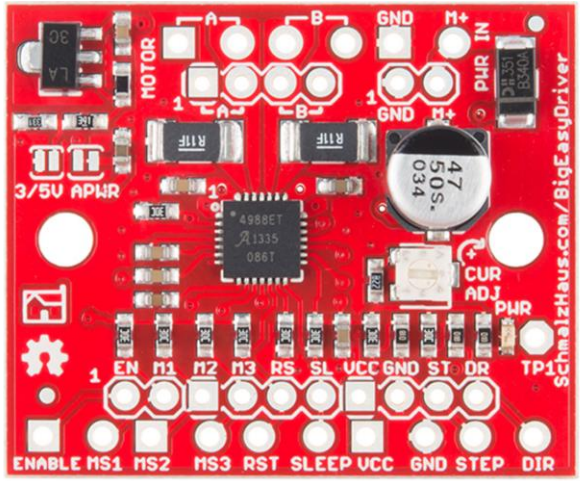


Figure 47 Big Easy Driver for stepper motor

There are setup and loop functions. Setup is responsible for setting up serial port and the pin mode on each of the pins. Loop is responsible for sequentially executing many tasks. In the beginning, it reads in new messages that have been accumulated on the Serial port buffer since the previous loop. This is done using the Serial.readBytes function. It then reads in the first byte representing the message type and interprets the message based on the communication protocol shown in Table 10 and Table 11.

Table 10 Communication protocol for TENS command message

Byte Length	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes
Representation	1 meaning TENS Command mode	Electrode Index	Finger Index	Finger Pad Index	Sensation	Intensity

Table 11 Communication protocol for motor command message

Byte Length	1 byte	1 byte	1 byte
Representation	2 meaning Motor Command mode	Finger Index	Enable State

After de-serializing the message, the loop then goes on to control the motors and TENS pads by enabling the motors, and the TENS pads accordingly. To accomplish this digitalWrite(motorDisablePin[i],LOW); and digitalWrite(relaypin,HIGH); respectively. The command to the motor controller is set to LOW because the datasheet specifies that an enable pin low command means enabling the motor.

For debugging purposes, we have added an additional physical switch in the base station that toggle allow the team members to toggle all motors and TENS pads on at the same time. This physical switch is connected to an Arduino GPIO, and the software commands the actuators according to this switch pin value.

Furthermore, the software controls 12 LEDs soldered onto the base station to indicate the state of all actuators. Five blue LEDs are used to indicate the enable state of the five stepper motors; five red LEDs are used to indicate the enable state of the TENS pads; two green LEDs are used to indicate system power and message transmission during bootup. The LEDs are shown in Figure 48.

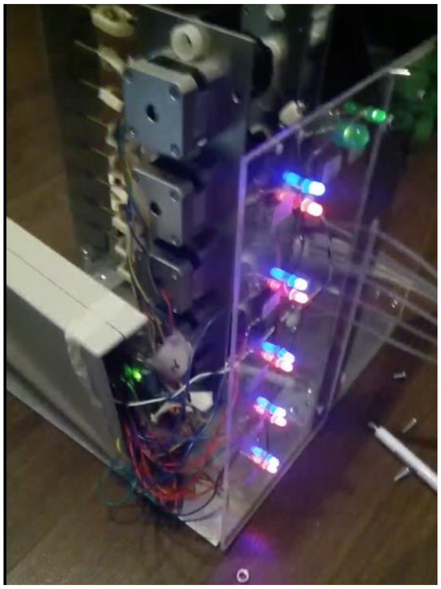


Figure 48 Control LEDs on casing

4.2 Modifications from Original Design

4.2.1 Virtual Environment Modifications

The original plan of the virtual reality software was to construct a few simple recovery tutorials, each targeting an area of recovery. This is similar to the exercises that the patient use in real physiotherapy with licensed professionals. However, such a recovery application would be tedious for both the user and the game developer, because every movement must follow strict rules under constraining guidance. Also, the tutorial program was imagined to mimic a nominal physiotherapy session in real life, which does not highlight the engaging aspect of a virtual reality software environment. For these reasons, it was decided that an open environment game is the better option.

The Space Rockaway game has also changed significantly since its original plans. The ball throwing self-defence idea was the same, but the initial game scene was intended to be as realistic as possible with physical baskets holding the balls to be thrown, as well as fully animated zombies that are to be modelled photorealistically. However, it was thought that the game would likely cause trauma for the large older generation user base. Zombies were then changed to rainbow-colored translucent polygon humanoids that help to convey a futuristic vibe. The old realistic environment with grass and a basket holding a limited number of balls was clearly unfit for the newly created character. The environment was recreated to not only to be the interior of a spaceship to give a sense of wonder, but also built to look spacious and clean.

The tools and platform has changed since the original prototype. A prototype was first built using the SceneKit 3D game engine on the macOS Sierra operating system. Despite of the low learning curve, the game engine soon had many issues that hindered the software progress. It has trouble executing concurrent animations, so no two objects can be controlled at a given time. Even serially executing animations one after another involves many nested callbacks, which makes the implementation unpleasant. It does not provide a graphical interface to manage the game environment, which means making a complicated environment is a time-consuming process that must be done in code. It also does not support Oculus or Leap Motion, both of which the team has decided to use for their features and ease of integration.

To address the limitation of the native game engine on macOS, Unity 3D game engine was used instead. Although the game engine can be installed on macOS, the Oculus VR headset requires a discrete GPU with high performances. For this reason, the team switched to a gaming desktop running Windows 10 with GTX 1080 GPU. The game engine provided many built-in game objects such as user facing text labels, buttons, progress bar, and particle effects, all of which can be seen in the Figure 49. These elements allowed the team to focus its energy on the game mechanics rather than supporting elements.

Figure 49 Space Rockaway game over menu

The second part of the original plan was to construct a virtual environment for the user to explore freely for leisure purposes. The underlying concept was to demonstrate the full electrotactile and force feedback capabilities of our device. However, this was difficult to implement because only spherical object can be used to provide tactile feedback. It was found that Leap Motion's integration with Unity is not yet complete, and the callback functions for when a finger intersects with the collider mesh of an object are significantly inaccurate. The reason is that the moving fingers are constantly being updated on the screen while its collider mesh can only be a static object. Therefore, the callback is not sensitive to the fine finger movements such as dipping just the index finger inside an object. To address the issue, a simple sphere collision script was written for only spherical objects. Other shapes are not supported due to the complexity of math involved with implementing an efficient real-time 3D object collision algorithm. The free environment idea eventually became the TENS Studio.

4.2.2 Hand Tracking Modifications

Hand tracking has changed from IMU sensor to Leap Motion for its accuracy, ease of implementation, and simplicity of use. The original plan of re-calibrating hand positional drift of the IMU in the beginning of each game is tedious, and would have tracking problems if the game were to be paused in the middle. Pausing the game is common considering it is the default behavior every time the user takes off the Oculus headset. Implementation would require significant efforts in calculating the precise finger positions based on raw IMU data. The Leap Motion on the other hand, is attached to the Oculus headset and utilizes the Oculus positional tracking for hand position. The Leap Motion API can be called to easily get the orientation and position of each finger segment in the global coordinate. In addition, the Leap Motion provides Interaction Engine, UI Input, and Hands API Modules that provides further support available exclusively for the Unity game engine. These include a pre-made model of the human hand and physics support for the hand interacting with objects. Figure 50 shows an in-game hand interaction.

Figure 50 In-game hand interaction with Leap Motion

4.3 APIs and Services

The Leap Motion does have its own finger tracking problems and a limited 150 degrees of field of view, but it always provides a reliable hand position when the hand is in front of the sensor. For this reason, the team decided Leap Motion is better than the IMU design that would have introduced drift problems.

During the development of the software systems, several services, APIs, and SDKs are used to make the final product possible. These include open sourced software: Arduino SDK, Arduino IDE, Teensyduino Arduino IDE add-on, and Sparkfun code snippets. The proprietary software used include: Xcode IDE, SceneKit, macOS SDK, Unity3D SDK, Unity IDE, Windows C# SDK, and Leap Motion Orion SDK.

The team used Github as software repository for code. Various game object models are downloaded from BlendSwap. These objects are as follows: light stripe ball, altar, cannon, humanoid robot, and space station 3D projection.

Several game object textures are found on the internet. These are: skin texture, game over image, window wallpaper in main game, window wallpaper in TENS studio, crate wood texture, floor tiles, and doors. Sound effects are downloaded from Freesound.org. The font “Boulevard Saint Denis” used in the game menu is created by Octoptype.

4.4 Testing and Performance

4.4.1 Virtual Environment

The Oculus headset was capable of displaying the VR environment without much latency. The refresh rate of the headset is 90 Hz with a 110 degree of view [23] on a capable GPU. The usual rendering poses no issues for the viewing experience. However, the code written for the game that are executed by the CPU causes glitches in the rendering from time to time that are noticeable only in the development mode. The built game, had no short stalling issues due to less CPU resources required in tracking errors and warnings.

However, playing the game for over 20 minutes often resulted in mild nausea and disorientation. Although the framerate is fast enough to be unnoticed, the effect of latency does contribute to nausea over a long period. In addition, the position of the Oculus is tracked by a stationary sensor, which has a very limited operational range of around 1 meter cubed. The position of the VR avatar immediately stalls as soon as the user goes out of range of the motion sensor, which contributes to nausea as well over time.

4.4.2 Hand Tracking

In short, hand movement tracking with Leap Motion allowed the general idea of the glove to be demonstrated adequately, but proved to be woefully inadequate to be used in any real-life scenarios.

The Leap Motion works out of the box and exhibits a high precision when all five fingers on a hand is visible in a well-lit environment. The default hand model provided by the Leap Motion Hands API is well made and realistic. Subtle skin texture was applied on the default model to make it more realistic, as can be seen in figure below.

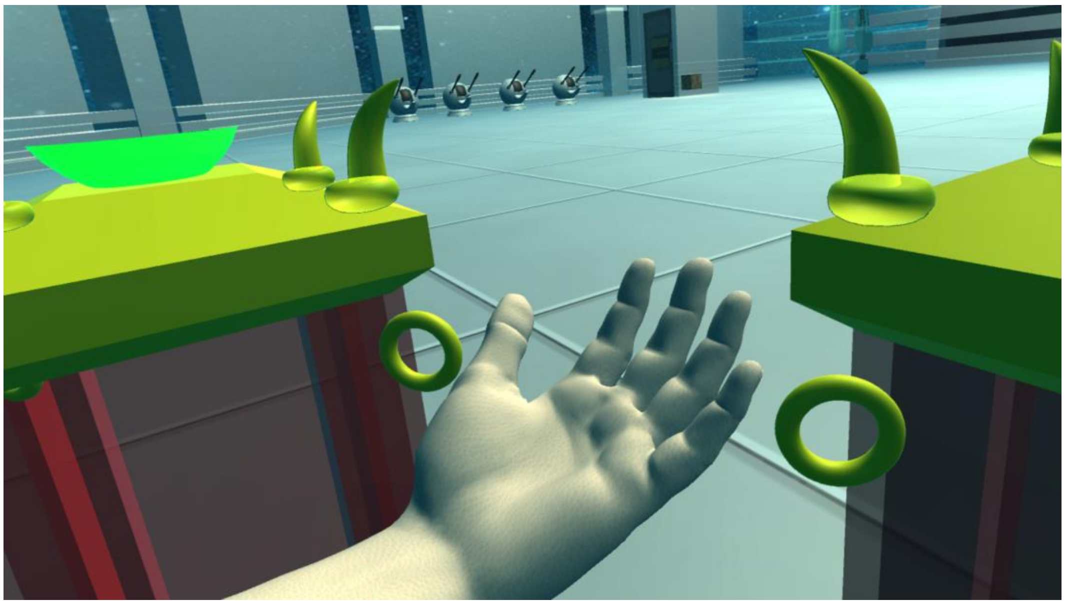


Figure 51 Close-up of in-game hand model

However, the advantages are overshadowed by various problems that were observed during the development and testing phases. The issues are not significant after getting use to the system, but raises future concerns for using the sensor.

The hand is always the same size configured at a fixed arm length. As a result, users see the wrong scale when using their VR hands. This will hinder the recovery progress as the users learn to move a slightly scaled version of their hands.

Hand gestures are also hard to be tracked by Leap Motion. Whenever a finger is hidden behind another finger or the palm, the Leap Motion software interpolates the position of that finger instead of relying on actual measurements. This can be exacerbated by a noisy background or a poorly-lit environment. The problem is particularly prominent when two hands are interlaced, where the sensor is not able to accurately output finger positions.

Similarly, the Leap Motion performs significantly worse or becomes non-functional if the hand is covered by any foreign objects. This was an issue when integrating the mechanical pull-wire system with software hand tracking. The Leap Motion was not able to recognize the hand when the custom pull wire glove was worn. The team has tried using different lighting, materials, and shapes to no avail. The Leap Motion software tracks hand using an internal hand model [24]. This means that any interference to the perceived shape of the hand will confuse the Leap Motion software hand tracking. The electrotactile system can be tracked because the Velcro loop only obscures the finger tips. However, the fingers positions are tracked significantly less accurately when compared to bare hands. Figure 52 shows an identifiable instance of mis-tracking when the thumb is obscured by the Velcro loop.

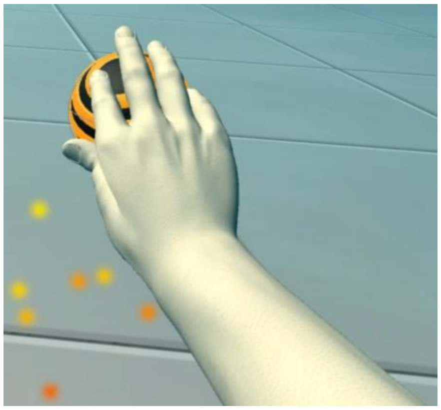


Figure 52 Leap Motion hand tracking when thumb is obscured

The field of view of the sensor is not adequate. While testing on users with little or no VR experience, the team found that people inadvertently move their hands outside of the sensor's scope. When throwing a ball, people also tend to swing their arm back first, which causes the sensor to lose track of the hand entirely. While the Leap Motion API does have hand position recovery strategy in place for object interaction for this specific issue, it rarely worked in real life.

The speed of Leap Motion data transfer is dependant on various factors. The company claims that a balanced tracking mode gives around 31ms of delay from sensing to rendering on the Oculus [25]. In practice, the latency was barely noticeable during testing, and did not undermine the VR experience.

5 Schedule

The Gantt chart shown in Figure 53 visualizes the intended and actual schedule for the Fourth-Year Design Project presented by the material of this report.

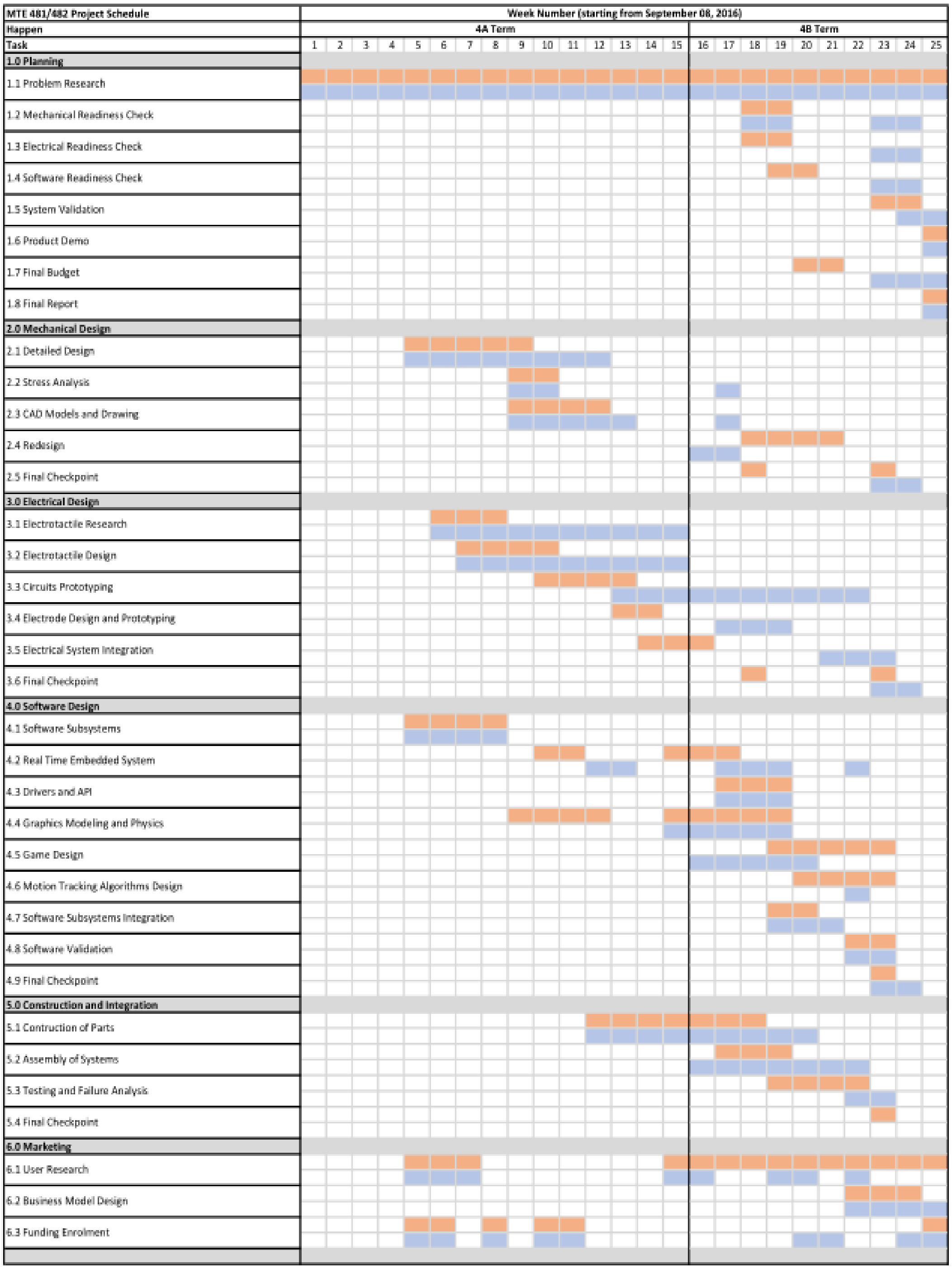


Figure 53 Gantt Chart of actual and intended schedule of tasks

Note that the orange represents the predicted schedule made in the initial assessment and the blue was the actual schedule. Noticeably there was significant differences between the two. Many of the discrepancies were from the delays while ordering parts and any unexpected difficulty while developing each system.

6 Budget

This section discusses the total budget that was spent on the project. The total budget was 2,983.90 CAD.

The predicted and actual monthly expenses are shown in Table 12.

Table 12 Budget decomposition by month

Month	Predicted Expenses (CAD)	Actual Expense (CAD)
September	78.15	155.94
October	503.89	676.3
November	499.05	461.8
December	333.33	478.3
January	333.33	667.33
February	333.33	544.23
Total	**2081.09**	**2983.9**

As shown in the table, there was a significant discrepancy between the predicted and actual expenses. This was for several reasons. The first is because during the initial calculation, some expenses were not accounted for or filed into the incorrect month. The second reason for the large discrepancy was due to the unexpected additional costs of the PCB printing and electrical components. Due to some development complications, some components were repurchased and additional components were needed. The last reason for the discrepancy is from the additional cost of 3D printing. When first printing the 3D printed components in the initial prototype, cheaper and lower quality options were available and acceptable. However, after redesigning certain components, a higher quality print was required, which added to the expenses.

The monthly predicted budget is shown in Figure 54. Note that blue represents the actual expenses and orange was the initially calculated and predicted budget.

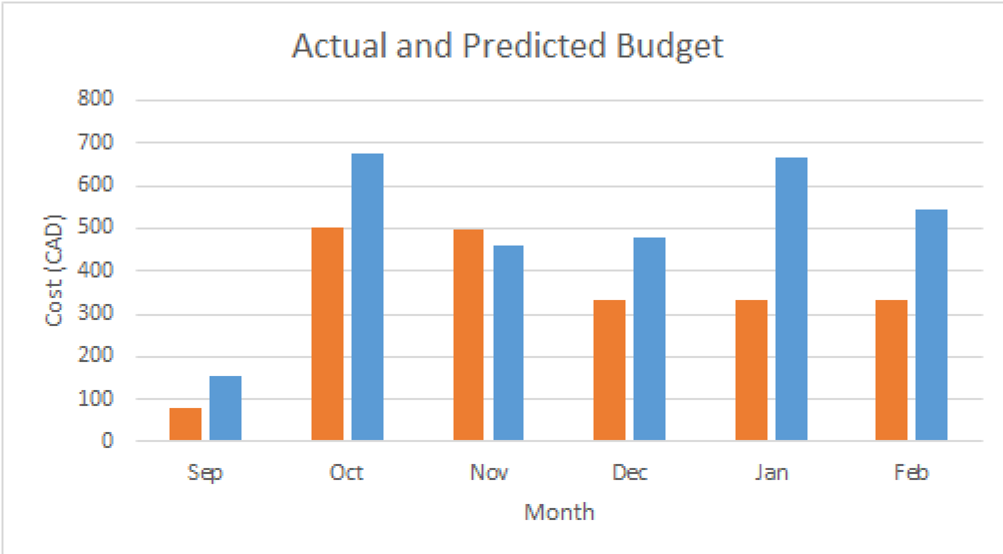


Figure 54 Actual and predicted monthly budget

7 Conclusion and Recommendations

This report entitled *"Virtual Reality Haptic Glove for Upper-limb Rehabilitation: Final Design and Implementation"* by group 18 communicates the detailed design details, modifications from original design, manufacturing, commissioning and testing and performance for the force feedback, tactile feedback, motion tracking and virtual reality systems, as well as the overall design review, project schedule and budget.

In terms of force feedback, the final design is a pull wire system which consists of a hand exoskeleton, a pull wire assembly and a pull wire lock mechanism. The cables that run through the hand exoskeleton are connected to stepper motors in the pull wire lock mechanism. When the user interacts with an object in virtual reality, the stepper motors are activated, which locks the cables due to the holding torque of the motors and provides force feedback to the user. The main modification in the force feedback design is that the final design does not include the use of encoders for motion tracking due to the poor accuracy. In terms of manufacturing, the hand exoskeleton was 3D printed while the pull wire assembly and mechanism was fabricated through machining and 3D printing. The final implemented design is tested by mounting the hand exoskeleton on a mannequin hand and observing the locking performance on the little finger for activation of the 3 different joints in the finger. Based on the results, it is observed that the pull wire force feedback system can sufficiently restrict motion for the middle and distal joints.

The final design for the electrotactile system consists of the electrotactile controller boards, the electrodes, the power module and the microcontroller module. The power module consists of an input battery supply as well as high voltage converter, while the microcontroller module contains the Teensy LC microcontroller. The final design involves the use of 5 electrotactile controllers and 5 electrodes, one controller for each electrode, where an electrode is placed on each of the five finger tips for tactile feedback. Powered by the power module, the electrotactile controller provides regulated current to the electrode based on the input commands from the microcontroller module. The final design of the electrodes is a concentric ring with an inner electrode diameter of 1 [mm], and can be used to provide anodic and cathodic stimulation. The final electrotactile stimulation pattern is cathodic in nature with an amplitude of 0-3.3 [mA], a duty cycle of 20-500 [us], a surface electrode-skin voltage of 350 [V], and a stimulation bandwidth of 0-200 [Hz]. Anodic stimulation is used at a very low magnitude and large pulse width to ensure zero net DC current through the skin. There were several modifications from the original design to provide electrical safety and improve performance, including the addition of a resettable fuse for electrical safety, a current limiting resistor to protect the user from electrical shock and selection of components with higher power ratings. The electrotactile controllers and electrodes were manufactured through PCBWay, a PCB manufacturer based in China, and all the boards in the final electrotactile system were populated and assembled by the team. Each electrotactile controller is enclosed in a 3D printed pod, and 6 pods are fastened on the forearm using flexible PVC wire sheaths, Velcro and elastic bands. The electrotactile system passed the functional validation stage, where different sections of the final design are tested and matched to simulation results. In terms of functional qualification, the final tactile sensation reported for the finger tip indicates a "pressure-like" sensation associated with higher frequencies and cathodic stimulation. However, the user reports a "tickle-like" sensation in addition to the "pressure-like" sensation, and hence the final tactile sensation does not feel very natural.

For motion tracking, the LeapMotion IR sensor is integrated with the Oculus rift in order to provide hand tracking through the Orion SDK provided by LeapMotion. The virtual environment is built using the Unity3D game engine, while the 3D models are generated through Blender 3D modelling software. The final virtual reality game, Spack Rockaway, involves interaction with balls that the user throws towards incoming characters to score points. Embedded software is written on an Arduino Mega, which communicates between Unity and the force feedback and tactile feedback systems. The main modification form the original design is the use of Unity3D game engine instead of the SceneKit 3D game engine because the former was determined to be a more powerful engine that can reduce development time. During final testing, it was determined that the VR game environment and LeapMotion tracking functioned as expected with minimal latency, with a refresh rate of 90 [Hz] and 110 [degrees] of view. However, mild nausea was reported among a few users during extended use of the system for over 30 minutes.

The team noticed a delay in schedule for systems integration and testing due to manufacturing delays as well as issues that required debugging in individual systems of the final design. The actual total costs for this project, from September 2016 to February 2017, is 2,983.90 CAD, while the budget predicted for the same period is 2,081.09 CAD. Hence, the team exceeded the budget by 902.81 CAD, which is accounted due to unexpected re-design costs and additional manufacturing costs for PCBs and better quality 3D printing.

There are several areas for improvement in this project. Firstly, it is recommended that the team designs a motion-tracking system in-house instead of LeapMotion. This is because LeapMotion provides very poor tracking when the user wears the hand exoskeleton, and hence, complete system integration is very challenging and impractical with the current state of LeapMotion. Secondly, the effect of stimulation parameters on tactile sensations requires more investigation. The current system involves the use of only one type of stimulation pattern with varying intensity, however, a combination of different patterns on different parts of the hand can be used to create more complex haptic interactions. Lastly, it is recommended that the system also provide motion metrics that can be used by a patient to understand recovery progress better in upper-limb rehabilitation.

8 Teamwork Effort

The teamwork is fluent and good overall. The team meshed well together and everyone had key contribution to the development of the project. Every team member was easy to contact and actively participated in the team meetings and completed their assigned task within the specified time frame. The team members were very vocal with respect their concerns and opinion. These elements made the teamwork environment stress-free and efficient. As a whole, the team members have experience in creating various medical devices, consumer electronics, mechanical tools, and user-facing software. In this project, each member of the team is specialized in either mechanical, electrical, or software. By separating the project into the three categories while maintaining collaboration, the team can implement and test the final design of each of the main systems, and integrate the final system together following successful independent system testing.

Si Te Feng helped to the software sections of this report. He has been mainly working on the virtual environment in Unity for the integration with Oculus, Leap Motion, and the microcontrollers. Si Te helped to create the website and was involved in pitching the idea at Velocity Fund Finals. He has also designed the controller PCB pods for the electrotactile system.

Ben Lambert worked on stroke rehabilitation research, electrotactile system research and design, marketing efforts, and systems integration and assembly. He contributed writing for the electrotactile sections of this report. He worked with Sunaal Philip Mathew primarily during the design and production of the electrotactile system, and enjoyed working with everyone simultaneously, as a team, when he had the chance.

Hernan Gomez was in charge of completing the force feedback system. The design tasks he completed included: create CAD models for the components of system, research and selection of component, manufacturing of parts, assembly of the system and mechanical testing. Additionally, he helped with the VR game design and helped testing the VR game. He wrote the force feedback section of this report.

Sunaal Philip Mathew worked on the final electrical design, fabrication and testing of the electrotactile controllers, the power module and the microcontroller module. In collaboration with Ben Lambert, he worked on the electrotactile system detailed final design section of this report. Once the final electrotactile system was built, he worked on performing user testing to investigate the simulation of tactile sensations. He was also responsible for populating and testing the electrotactile controller boards required for the final prototype.

Sang Min Shin worked on an alternative design to the pull wire soft exoskeleton that used a mechanical exoskeleton approach. He also assisted the mechanical effort by creating SolidWorks CAD models, as well as the software effort by writing all Serial communication related code and Blender models. Finally, he was responsible for the final report compilation and formatting.

9 Bibliography

[1] Ontario Stroke Network, ""Stroke Stats and Facts,"," Cardiac Care Network,, 2016. [Online]. Available: http://ontariostrokenetwork.ca/information-about-stroke/stroke-stats-and-facts/. [Accessed 1 April 2017].

[2] S. T. Feng, H. Gomez, B. Lambert, S. Mathew and S. M. Shin, "Final Design Report - Happen: Virtual Reality Haptic Glove for Stroke Therapy," Waterloo, 2016.

[3] Turnigy, "3300mAh 2S 30C Lipo Pack.," n.d.. [Online]. Available: https://hobbyking.com/en_us/turnigy-3300mah-2s-30c-lipo-pack.html. [Accessed 3 April 2017].

[4] Qianson, "Qianson DC-DC Boost Converter 8V-32V 12V to 45V-390V 110V/220V Adjustable High Voltage ZVS Capacitor Charging Power Supply Module: Home Audio & Theater," Amazon, n.d.. [Online]. Available: https://www.amazon.com/Qianson-Converter-45V-390V-Adjustable-Capacitor/dp/B01IVOWYSM. [Accessed 3 April 2017].

[5] SparkFun , "SparkFun Breadboard Power Supply 5V/3.3V," SparkFun, n.d.. [Online]. Available: https://www.sparkfun.com/products/114. [Accessed 3 April 2017].

[6] PRJC, "Teensy 3.2 & 3.1 - New Features," PRJC, n.d.. [Online]. Available: https://www.pjrc.com/teensy/teensy31.html. [Accessed 3 April 2017].

[7] Digikey, "NDD04N60ZT4G," Digikey, n.d.. [Online]. Available: http://www.digikey.ca/product-detail/en/on-semiconductor/NDD04N60ZT4G/NDD04N60ZT4GOSCT-ND/3462425. [Accessed 3 April 2017].

[8] Digikey, "FZT560TA," Digikey, n.d.. [Online]. Available: https://www.digikey.ca/product-detail/en/diodes-incorporated/FZT560TA/FZT560CT-ND/287080. [Accessed 3 April 2017].

[9] Digikey, "LM324DT," Digikey, n.d.. [Online]. Available: http://www.digikey.ca/product-detail/en/stmicroelectronics/LM324DT/497-1580-1-ND/592078. [Accessed 3 April 2017].

[10] Digikey, "IRS2101STRPBF.," Digikey, n.d.. [Online]. Available: https://www.digikey.ca/product-detail/en/infineon-technologies/IRS2101STRPBF/IRS2101STRPBFCT-ND/1189808. [Accessed 3 April 2017].

[11] Digikey, "B59774B115A70," Digikey, n.d.. [Online]. Available: http://www.digikey.ca/product-detail/en/epcos-tdk/B59774B115A70/495-3890-ND/652053. [Accessed 3 April 2017].

[12] Digikey, "MP925-50.0K-1%," Digikey, n.d.. [Online]. Available: https://www.digikey.ca/product-detail/en/caddock-electronics-inc/MP925-50.0K-1-/MP925-50.0KF-ND/1284429. [Accessed 3 April 2017].

[13] TUV Rheinland of North America, "Effects of Electrical Current in the Human Body. Product Safety and Quality.," TUV Rheinland of North America, n.d.. [Online]. Available: https://www.tuv.com/media/usa/aboutus_1/pressreleases/fieldevaluation/Effects_of_Electrical_Current_in_Human_Body.pdf. [Accessed 3 April 2017].

[14] Littelfuse, "Polyfuse PTC Selection Guide," Littelfuse, n.d.. [Online]. Available: http://www.littelfuse.com/~/media/electronics_technical/application_notes/resettable_ptcs/ec327-e_polyfuse_ptc.pdf. [Accessed 3 April 2017].

[15] Instructables, "Arduino Christmas Light Controller," Instructables, n.d.. [Online]. Available: http://www.instructables.com/id/Arduino-Christmas-Light-Controller/. [Accessed 3 April 2017].

[16] H. Kajimoto, S. Saga and K. Makashi, "Pervasive Haptics - Science, Design and Application," Springer, Japan, 2015.

[17] International Rectifier, "Application Note AN-978: HV Floating MOS-Gate Driver ICs," International Rectifier, El Segundo, n.d..

[18] K. Sato and S. Tachi, "Design of Electrotactile Stimulation to Represent Distribution of Force Vectors," The University of Tokyo, Kelo University, Tokyo, 2010.

[19] L. Rozenblat, "CALCULATING SPACING BETWEEN PCB TRACES FOR VARIOUS VOLTAGE LEVELS," Rozenblat, Lazar, 2009. [Online]. Available: http://www.smps.us/pcbtracespacing.html. [Accessed 3 April 2017].

[20] Leapmotion, "Leap Motion VR Mount + Oculus Rift CV1," Youtube, 5 April 2016. [Online]. Available: https://www.youtube.com/watch?v=OUdL3y-mrFM. [Accessed 3 April 2017].

[21] Orion Beta, "Orion Beta," Leap motion, n.d.. [Online]. Available: https://developer.leapmotion.com/orion. [Accessed 3 April 2017].

[22] BlendSwap, "BlendSwap," BlendSwap, n.d.. [Online]. Available: http://www.blendswap.com/. [Accessed 3 April 2017].

[23] A. Binstock, "Powering the Rift," Oculus, 15 May 2015. [Online]. Available: https://www3.oculus.com/en-us/blog/powering-the-rift/. [Accessed 3 April 2017].

[24] LeapMotion, "Hands," LeapMotion, n.d.. [Online]. Available: https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Hand.html. [Accessed 3 April 2017].

[25] R. Bedikian, "Understanding Latency: Part 2," Leap Motion, 13 July 2013. [Online]. Available: http://blog.leapmotion.com/understanding-latency-part-2/. [Accessed 3 April 2017].