HAPTIC FEEDBACK GLOVE

By

Noah Franceschini

Sohan Patel

Final Report for ECE 445, Senior Design, Fall 2022

TA: Jason Paximadas

6 December 2022

Project No. 3

Abstract

Our project was a user-worn device similar to a VR haptic feedback glove. It is designed to apply a variety of forces back to a user depending on the force exerted by a separate manipulator. We created the user-worn device as well as a rudimentary gripper. We were able to implement a finger tracking algorithm, and developed a method to measure force applied by the manipulator and apply it back to a user's hand. We were unable to get force feedback fully working due to unforeseen issues with electronics chosen for the project.

Contents

1. Introduction **1**

1.1 Subsystem Overview 1

2. Design **3**

2.1 Design Procedure 3

2.2 Design Details 3

3. Verification **7**

3.1 Glove 7

3.2 Manipulator 8

3.3 Microcontroller 9

3.4 Power 9

4. Costs **11**

4.1 Parts 11

4.2 Labor 12

5. Conclusion **14**

5.1 Accomplishments 14

5.2 Uncertainties 14

5.3 Ethical considerations 15

5.4 Future work 15

References **16**

Appendix A Requirement and Verification Table **17**

1. Introduction

From Zoom calls for work and instructional material, to YouTube videos where we can explore vast worlds through our screen, the way we interact with our world has become increasingly virtualized, but only in sight and sound. It is difficult to interact with objects through a screen such that if a material is not immediately in front of you then there is no good way to feel its physical properties. Our solution to this problem is the haptic feedback glove.

The haptic feedback system comes in two distinguishable parts: a robotic manipulator and a wearable glove. The robotic manipulator is a 3D printed robotic hand controlled via servos. The robotic fingers' positions mimic that of the user's hand in the glove and each finger reports back its current draw to tell the glove system how much force the servos are applying on whatever object the manipulator is holding. As for the glove, it applies forces read from the manipulator system back onto the user's hand as to recreate a sense of touch from the manipulator's held object, thus allowing users to interact with objects from a distance.

Throughout this report, we detail the intended design aspects of our project and how the system should cohesively mesh into the final product. Though, despite our best efforts, our end product did not hit the goals that we wanted it to. We were unable to get the force feedback part of the glove working properly; however, we were able to successfully implement finger tracking as well as successfully implement all of our components on our printed circuit board (PCB).

1.1 Subsystem Overview

There are four main components to our system: the glove subsystem and the manipulator subsystem, the microcontroller subsystem, and the power subsystem. The glove subsystem is a wearable device that is responsible for applying forces to the user's hand and keeping track of the user's finger position. The manipulator subsystem is a 3D-printed robotic hand that mimics the glove's finger positions and relays forces back to the glove subsystem. The microcontroller subsystem is the mediator of the glove and manipulator subsystem as it receives both the position and force information from both systems and tells each respective subsystem where it should be and what forces it should be applying. As for the power subsystem, that consists of a lithium-polymer (LiPO) battery that powers our entire PCB through a buck converter as well as the electronic speed controllers (ESC) that we use for our brushless motors.

To consider this project successful, we instituted these requirements that must be met by our design:

- Measure and apply forces back to user's hand to at least 10 lbs of force
- Have a latency of less than 1 second
- Support dynamically changing the force to match anything from slight resistance to fully stopping movement

Our reasoning for these requirements hinges on what we believed would create a quality experience for the user. We wanted to apply at least 10 lbs of force back to the user as that is about the pinch strength of most people [1], where pinch strength is the measure between what a person can hold between a finger and a thumb, and we wish to restrain the users movement of up to that amount of force. For the user to have a responsive experience with the glove, we figured that a full control loop should take less than a second. Lastly, we wanted our glove to provide a range of forces as to allow the user to have a better feel for what the manipulator was interacting with. The range of forces should allow the user to tell whether the manipulator is encountering a solid object or something softer like a stress ball.

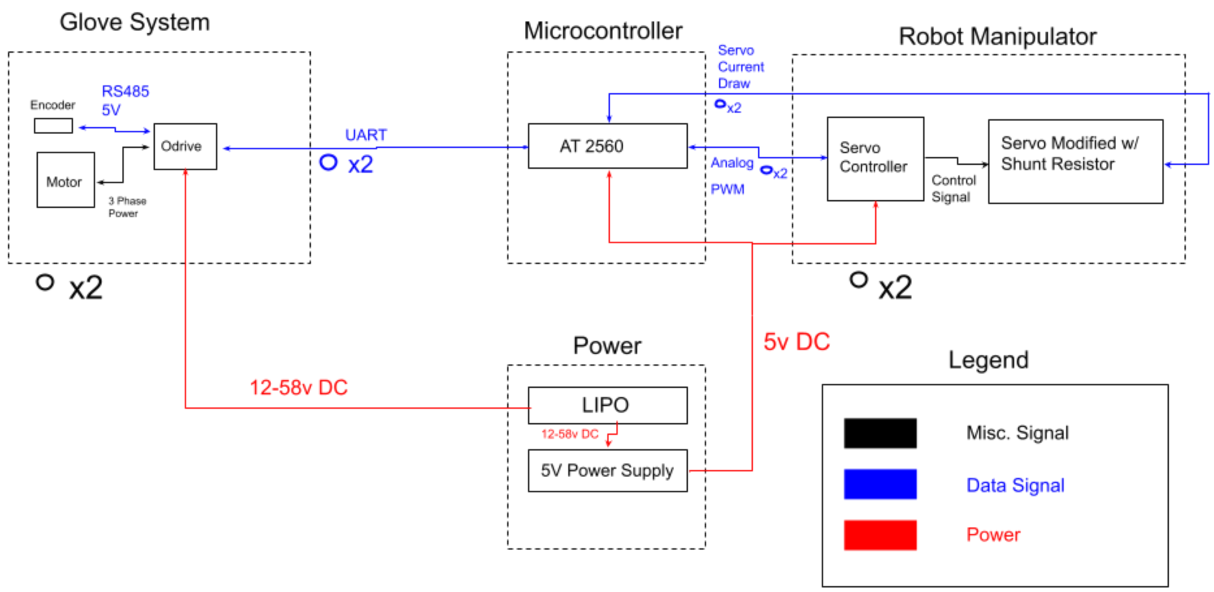

Figure 1: Project block diagram

2. Design

2.1 Design Procedure

2.1.1 Glove

We originally decided to use servos to apply forces to the user's hand, but they are not able to change forces, nor report back position without extensive modifications. We then looked into brushless motors and their controllers and found a few options: ESC, VESC [2] and ODrive [3]. ESCs, while cheap, can't support encoders and therefore can't measure motor position. VESCs do support encoders and are readily available, but low-cost designs use a chip in the DRV family from Texas Instruments that is in shortage at the time of this report, so we were unable to source them. The final option was ODrive, it is a robust system that has very precise control over torque and can be communicated with over the universal asynchronous receiver-transmitter protocol (UART) to send position information, the downside is that they are quite pricey and require a fair bit of configuration before use.

2.1.2 Manipulator

We decided on using simple hobby servos for this component. The alternatives would have been another controller with precise positioning like the ODrive, but we felt it would be overkill for our purposes.

2.1.3 Microcontroller

The microcontroller originally needed to support 5 UART devices, the only Atmel product we found that was readily available, well documented and inexpensive was the ATmega2560 [4]. The PCB also had a high side current sensing circuit that will be discussed in detail in a later section.

2.1.4 Power

The power system had a couple requirements. The first being 12-58V DC and 120A to the ODrives, the best candidate for this would be a LiPO battery, as they support the high discharge rate and a 6s battery has a voltage of 22.2V.

2.2 Design Details

2.2.1 Glove

The glove was relatively simple to design, as ODrives can do all that we need them too right from the factory. They are able to communicate with the AMT212B-V encoder over RS485 with some slight configuration. Using the ODrive and this encoder, we not only can drive the motor to an extremely precise position, we can also poll it via UART for that position. These UART messages are created using the ODrive Arduino library [3], which turned out to have bugs that we corrected. More specifically, the command for getting the encoder position was incorrect. It outputs this position as a positive or negative number in units of full rotations, so .5 would be a half turn clockwise. We controlled the force on the user's hand by setting the ODrive to a low current cutoff, just enough to take up slack in the strings attached to the user's fingers. We can then increase this cutoff via UART proportionally to the current consumed by the servos to pull back with increasing intensity on the wearer's hand. We found that 40A of current is enough to reach 10 lbs of force on the user's hand, with the correlation being roughly linear down to zero force. The motors and encoders were mounted to an arm sleeve, with strings attached to fingers of a glove.

2.2.2 Manipulator

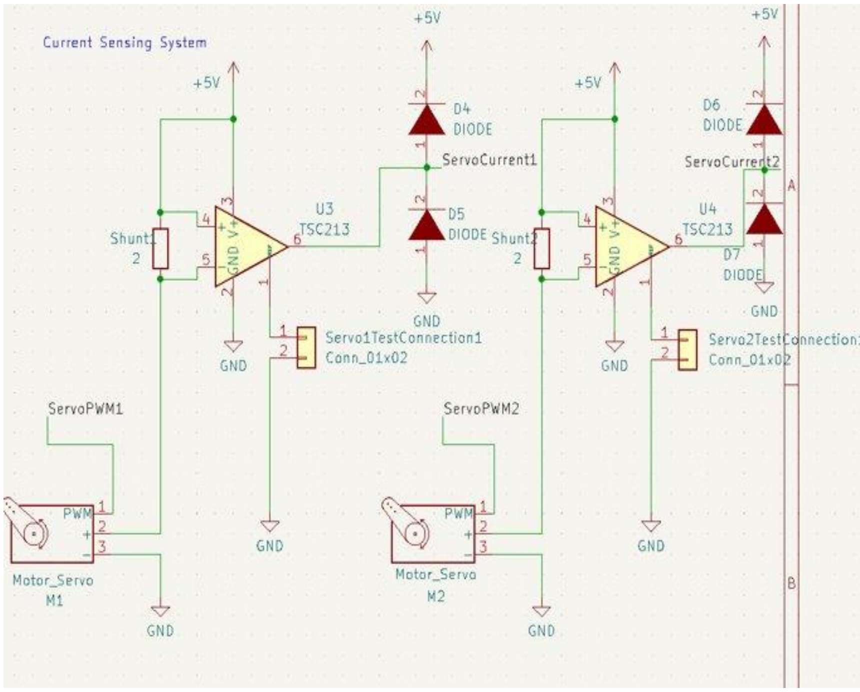

Figure 2: Current Sensing Circuit

The manipulator was one of the most important parts of our project, as it is where the entire idea of force feedback comes from. The manipulator was moved with servos that had high side shunt resistors. The voltage across these resistors is proportional to current via Ohm's Law. We can't drop the full 5V across this resistor however, as the servos pull 3A. This would lead to a 15W power dissipation which is quite high and it would be difficult to find resistors rated for that power. It would also mean we would have to consider other factors such as heat sinking or actively cooling the board. Because of this, we decided on a lower voltage drop of .1V. We chose a resistor of 30 mΩ. This allows us to drop just under .1V at 3A of current, at a much lower power dissipation. The issue with this is that the analog to digital converter (ADC) in the microcontroller unit (MCU) is only 10 bits, it can only read 1024 values. If we were to use the .1V directly, we would only be able to cover 20 discrete values. The concern we had was that this would lead to distinct steps that the user could feel as the force increased. In order to mitigate this, we decided to use an amplifier circuit to increase that .1V to the full 5V range, in order to take advantage of the full range of the ADC (Figure 2). We used a TSC213 [5] integrated circuit (IC), which has a gain of 50, which nicely landed us in the 5V range. There are some additional diodes in the schematic that are for protecting the MCU in the case that the voltage goes over 5V and could damage the ADC, but they were left out in the final PCB.

2.2.3 Microcontroller

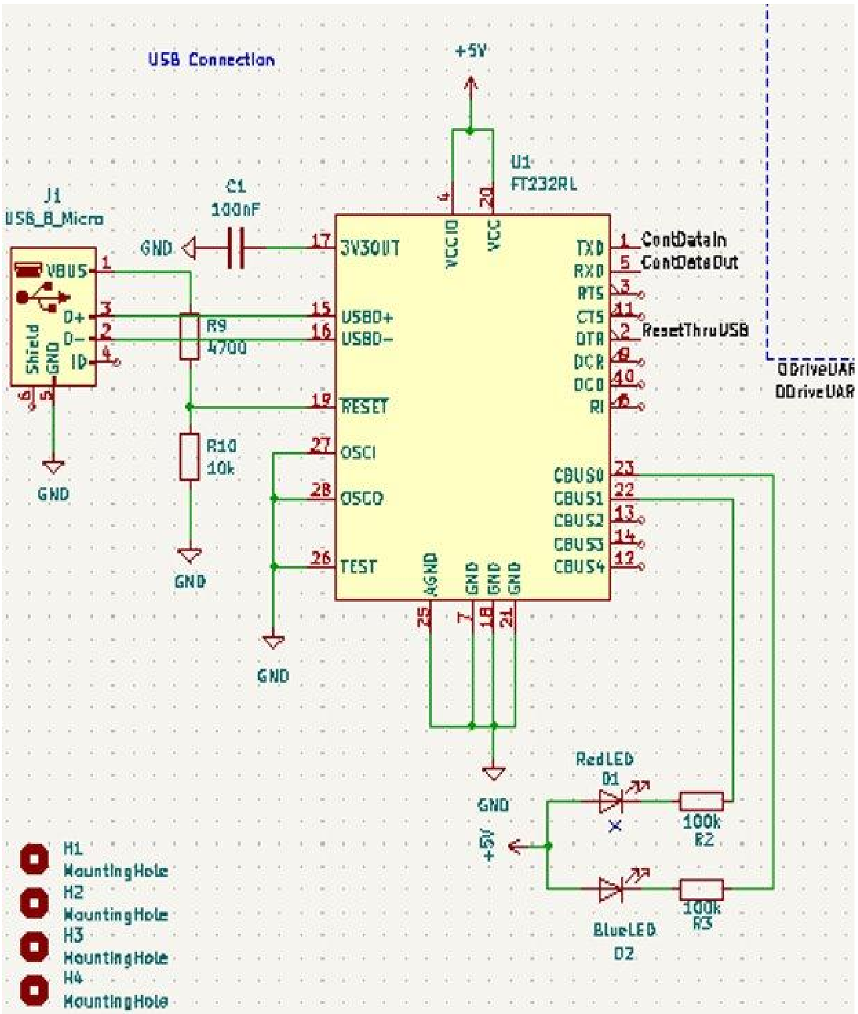

Figure 3: FTDI programming circuit

The microcontroller we chose was the Atmel ATMEGA2560. It was chosen because not only is it the same processor as an Arduino Mega, it is also capable of five UART connections at once. This was more important in our original design with five ODrives, but most arduino boards do not support more than one UART device. We wanted to use a MCU that was also on an Arduino board so we could flash an Arduino bootloader and program it via the Arduino integrated development environment (IDE). This would also allow us to stream serial information back to our computer for debugging. To do this we used a chip from FTDI that uses UART 0 on the MCU to flash programs (Figure 3). We also exposed the MCUs in-circuit serial programming (ICSP) pins and joint test action group (JTAG) pins in the case that the USB programming chip failed. We used the ICSP pins to flash the Arduino bootloader to the board and were able to program it over USB. However, we ended up not using the FTDI programmer in the final project because we found it easier to just flash over ICSP using an Arduino Uno.

2.2.4 Power

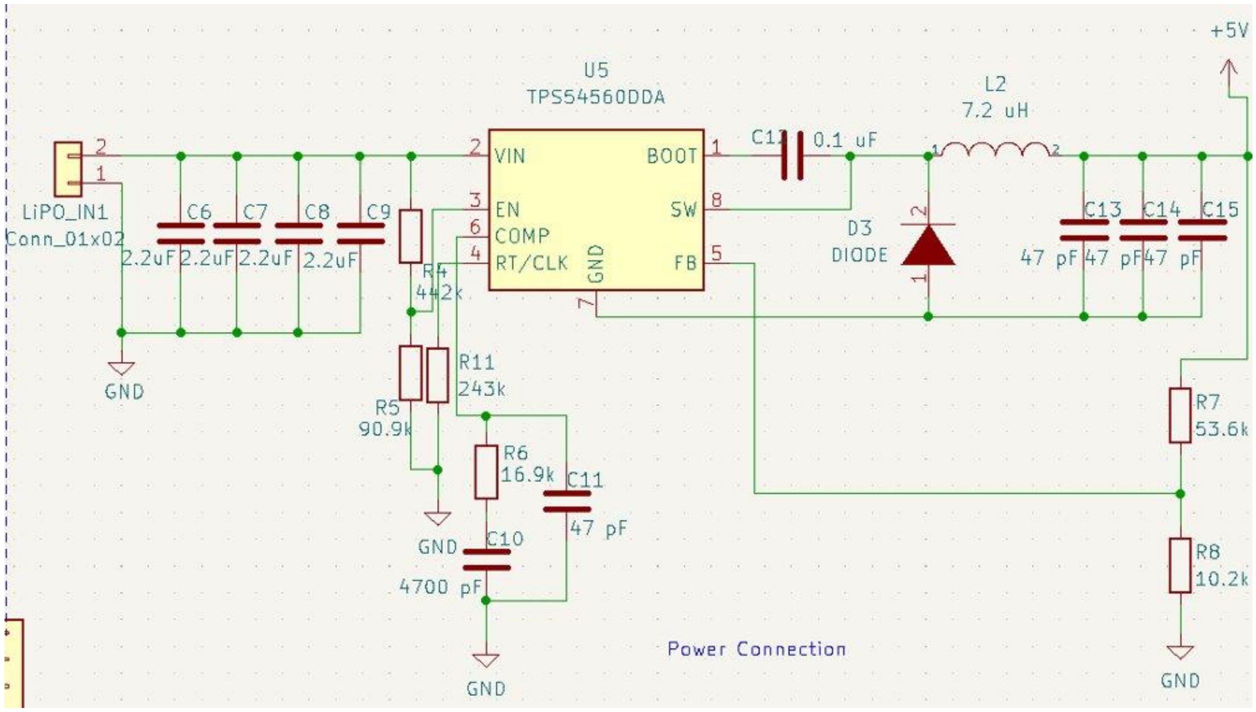

Figure 4: Buck Converter circuit

The power subsystem was quite simple. We chose a 6S, 5Ah, 45C LiPO to power the entire project. It can provide 22.2V at 5*45A, which is 225A, far more than the 120A we need. However the microcontroller only runs on 5V. We needed to step down the voltage and had 2 options, a linear regulator or buck converter. We ended up using a buck converter because a linear regulator would have to dissipate a large amount of power and would likely need to be cooled and would cause us to need additional design considerations, similarly to if we tried to dissipate 15W across a resistor. We chose a TPS54560DDA [6] chip from TI. It has an input voltage of up to 60V and can output 5V at 5A, which should be enough to run our entire project. The downside of it compared to a linear regulator is that it needs a fair amount of surrounding circuitry (Figure 4) in order to set the device's parameters. We also found that we should have added large output capacitors due to the transient loads produced by the servos turning on that ended up browning out the MCU.

3. Verification

3.1 Glove

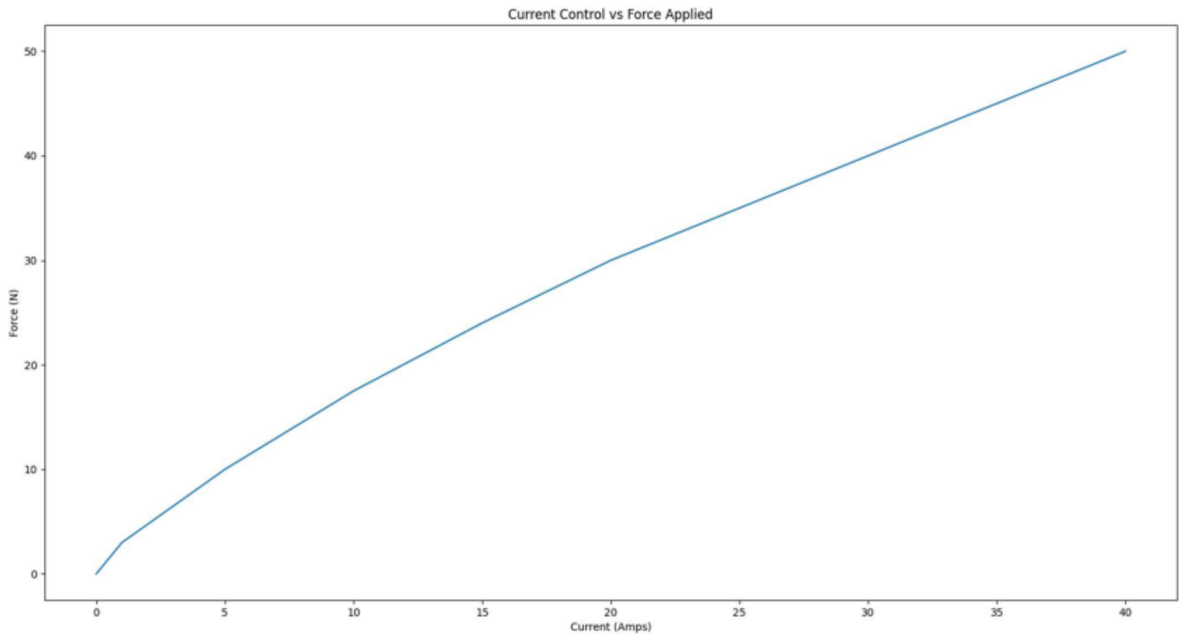

Figure 5: Current vs Force

The glove subsystem verifications were quite straightforward, to test UART we not only used an arduino mega to send commands, we also used an oscilloscope to view the waveforms. We were actually able to find an error in the ODrive Arduino library thanks to this. When we were receiving nothing in response when polling for encoder position, we decided to decode the UART signal and found that the ODrive was responding with random values. Upon diving deeper we found that the command being sent with the GetPosition() function was deprecated and meant for the old open-source ODrive 3.6, whereas we were using ODrive Pro. Figure 5 contains the forces we measured from the ODrive when applying various currents. Interestingly, this is somewhat linear which was not expected due to motor torque vs current not generally following a linear pattern. We were easily able to apply 10 lbs of force and could likely apply more, as the ODrive is capable of 60A sustained without external cooling. The final verification was if the glove could actually stop the user’s hand completely, we found that this was absolutely the case. When wearing the glove and attempting to close our hands it was easily able to keep them open with the 40A current value, which applied 50N of force.

3.2 Manipulator

Figure 6: Oscilloscope output of current draw

The manipulator was a bit more involved to debug, but was a similar process to the glove when testing servo communication via pulse width modulation (PWM). We measured the output of our PCB using an oscilloscope and found that it was, in fact, varying the PWM duty cycle in a way the servo could understand. We tested by sweeping it from 0 to 180 degrees and found that it did follow precisely. We were also able to get it to mimic finger positions quite easily by polling the ODrive encoder position and converting it to an angle between 0 and 180, then commanding the servo to that location. The most difficult part to verify was the voltage across our shunt resistor.

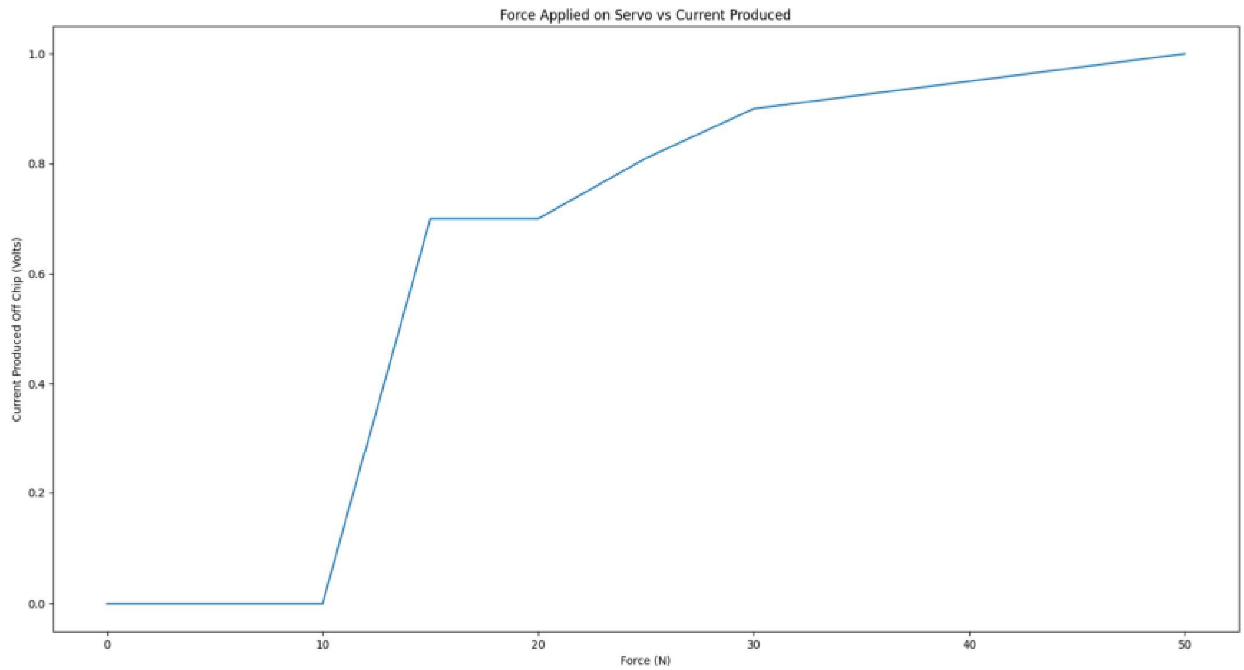

Figure 7: Servo force vs voltage

This graph (figure 7) is the measurement of the voltage across our shunt resistor against the force of the servo. This oddly shaped graph is due to unexpected behavior of the servo's controller. It would rapidly switch on and off, creating a waveform shown in figure 6. This caused our MCU to not sample the data correctly as it was more or less a floating signal between 0 and the voltage we wanted. Even when averaging it over time we were unable to get a usable signal, as the ADC was measuring mostly random values. Due to this we were unable to get force-feedback working.

3.3 Microcontroller

The microcontroller's verification went hand-in-hand with moving the servos and polling the ODrives. It had to facilitate the communication between glove and manipulator, as well as do so quickly. To test both of these on our custom board, we uploaded a program that simply mirrored ODrive position to the servo's and found that they would mimic each other's movement. To make sure the latency was low, we used an Arduino MEGA, which has the same MCU as our custom board and output the time the loop took to run to the serial monitor. The control loop ran in about 100ms, which updated the position 10 times in that loop. This means it took roughly 10ms to move the servos when the ODrives were moved, plenty fast for our use case.

3.4 Power

The power system was verified by measuring the output of our buck converter with a multimeter and confirming the voltage output was about 5V DC. We tested if changes to the input of the buck converter would affect the output in cases such as stopping a motor suddenly, where the voltage would increase slightly as current was sent back to the battery. We did this by rapidly changing the voltage input from 12 to 30V DC and found no change in the output. We did however, run into problems with transient loads generated by the servos moving. This would cause the board voltage to drop significantly, into the 3v range, which caused our ATMEGA 2650 to reset due to low power. This could have been solved by adding output capacitors to our design to help compensate for the sudden power demand.

4. Costs

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Quantity	Aggregated Cost ($)
ODrive Pro	ODrive Robotics	229.00	229.00	2	458.00
Servo 4-Pack	Deegoo	20.99	20.99	1	20.99
1kg PLA	Hatchbox	25.00	25.00	2	50.00
12g Wire	Haerkn	22.98	22.98	1	22.98
TPS54560DDA	Texas Instruments	6.86	6.86	6	41.16
47 pF Capacitor	KEMET	0.34	0.232	12	2.78
2.2 uF Capacitor	KEMET	0.80	0.466	12	5.59
4700 pF Capacitor	KEMET	0.31	0.31	4	1.24
0.1 uF Capacitor	KEMET	0.13	0.093	15	1.40
16.9 kΩ Resistor	Vishay / Dale	0.16	0.14	10	1.40
53.6 kΩ Resistor	Vishay / Dale	0.16	0.063	10	0.63
10.2 kΩ Resistor	Bourns	0.10	0.029	10	0.29
100 kΩ Resistor	Bourns	0.65	0.479	10	4.79
10 kΩ Resistor	Bourns	0.47	0.358	10	3.58
Schottky Diodes	Vishay General Semiconductor	0.54	0.438	12	5.25
7.3 uH Inductor	Wurth Elektronik	3.48	3.48	1	3.48
10 uH Inductor	KEMET	0.38	0.241	10	2.41
USB A Connector	GCT	0.87	0.87	1	0.87
8 MHz Crystal Oscillator	ECS	0.97	0.97	1	0.97
Solder Paste	Chip Quik	15.84	15.84	1	15.84
ATMEGA2560	Microchip Technology	17.79	17.79	2	35.58
Solder Wick	Chip Quik	1.99	1.99	2	3.98
Flux	Chip Quik	9.95	9.95	1	9.95
442 kΩ Resistor	Yageo	0.04	0.037	10	0.37
TSC213	STMicroelectronics	2.59	2.59	8	20.72
Red LEDs	Stanley Electric	0.40	0.40	6	2.40
7.2 uH Inductor	Wurth Electronics Inc.	3.57	3.57	3	10.71
90.9 kΩ Resistor	Stackpole Electronics Inc.	0.08	0.073	10	0.73
243 kΩ Resistor	Panasonic Electronic Components	0.10	0.088	10	0.88
Mini-USB Connector	Adam Tech	1.29	1.29	3	3.87

16 MHz Crystal Oscillator	Wurth Electronics	0.40	0.40	4	1.60
4.7 kΩ Resistor	Stackpole Electronics Inc	0.03	0.024	10	0.24
10 kΩ Resistor	Stackpole Electronics Inc	0.08	0.073	10	0.73
22 pF Capacitor	Vishay Vitramon	0.45	0.414	10	4.14
0.03 Ω Resistor	Ohmite	0.43	0.40	10	4.04
47 uF Capacitor	Samsung Electro-Mechanics America	0.36	0.311	10	3.11
PCBs	PCBWay	1.00	1.00	10	10.00
Rush Order PCBs	4PCB	33.00	33.00	3	99.00
Shipping and Handling	Digikey / Mouser	60.00	60.00	1	60.00
Part Subtotal					**915.70**
Labor Hours	Noah and Sohan	$48/hr		(10 Hours each week * 2 people * 9 weeks)	8,640.00
Total					**9,555.70**

4.2 Labor

Table 2 Schedule

Week	**Task**
10/2-10/8	Design PCB (Noah) Purchase electronic components (Sohan) Purchase PLA (Sohan)
10/9-10/15	Order PCB (Sohan) Receive PLA (Noah + Sohan) Begin Printing/Designing robotic hand (Noah + Sohan) Begin Designing Glove (Noah + Sohan)
10/16-10/22	Receive electronics (Noah + Sohan) Configure ODrive (Sohan) Test Servos (Noah) Test PCB Components on a breadboard (Noah) Test Power System (Sohan) Continue designing hand (Noah + Sohan) Continue designing glove (Noah + Sohan)

10/23-10/29	Solder PCB components into the PCB (Sohan) Assemble hand (Noah) Assemble Glove (Noah) Combine all subsystems together (Noah + Sohan) Begin implementing finger tracking algorithm (Noah + Sohan)
10/30-11/5	Finish implementing finger tracking algorithm (Noah + Sohan) Begin implementing force feedback (Noah + Sohan)
11/6-11/12	Continue implementing force feedback (Noah + Sohan)
11/13-11/19	Finish implementing force feedback (Noah + Sohan) DEMO TO TA
11/20-11/26	FALL BREAK Last minute fixes/tweaks (Noah + Sohan)
11/27-12/3	Final Demos (FINISH)

5. Conclusion

5.1 Accomplishments

Despite being unable to implement force-feedback with the glove, we still managed to meet most of our project's goals that we set out to accomplish. In terms of high-level requirements, we implemented the finger-tracking algorithm efficiently enough for the latency between hand movement and manipulator movement to be ~100ms. Although active force-feedback we could not accomplish, we were able to to test the motors with a variety of resistive forces and try them on the hand as shown in figure 5. Lastly, we managed to deliver upwards of 10 lbs of force as shown in figure 5 as well. In testing that amount of force was enough to completely arrest our hand movements.

Other small goals that we achieved were with finger tracking and the final PCB design. After we assembled the glove and manipulator system and after a bit of parameter tuning, we discovered that our finger tracking algorithm worked. When we would squeeze our fingers shut we found that the manipulator would track our fingers with ~5° of deflection. Another accomplishment was that our final PCB design proved to be successful as we could run all of the software off of the ATMEGA2560 on the PCB as well as all surrounding components such as the current sensing circuit.

Where these successes leave our project now is that we have a manipulator subsystem that accurately mimics the glove subsystem as well as has the potential to provide force feedback given that we could sample the current sensing circuit more accurately.

5.2 Uncertainties

Our failures come from unexpected difficulties with both the buck converter and the hobby servos that we used in our project. The buck converter in our design had an issue where when the servos were actuating the output voltage would sharply decrease from 5.3V to ~3.5V. This issue would cause the circuit to brown out and consequently cause our microcontroller to crash. After some investigation we believe that a large output capacitor would help mitigate the transient loads that the servos place upon the power system. Our reasoning for an output capacitor is that the servos draw more current than the buck converter can supply at its current switching frequency when the servos are holding their position. An output capacitor can act as a reservoir of charge for when the system is in a steady state, dissipating this charge when the servos require more current, keeping the 5V line from dropping below 5V as the buck converter attempts to switch switching frequencies to make up for the inadequate current supply.

Another issue was our choice of servos, though the problems did not manifest until trying to implement force feedback. When probing the voltage drop across the shunt resistor in the current sensing circuit, we noticed that the signal was relatively very noisy as seen in figure 7. What we believe is happening is that the servo controller rapidly turns the servo on and off to reach a desired position rather than a smooth increase in current to move the servo that other more expensive motor controllers do. The problem this controller causes is that it creates such a noisy signal that the sampling in our ATMEGA2560 is unable to correctly read the spike in the voltage drop and instead retrieves a value somewhere in between. This effect caused our force feedback algorithms in tests to be unresponsive at best and would frequently stutter at worst, neither of which felt like a proper force feedback mechanism. We attempted several solutions to remedy the spikes in voltage drop. We tried averaging samples over a short window to smooth out the force feedback. We tried using the maximum value over a short period to see if we could obtain the voltage spike seen in figure 7. We also tried different force coefficients to lessen the impact that large spikes would generate and none of those approaches proved fruitful. We believe that the best solution to this problem is better motor controllers such that when the manipulator motors are met with resistance they linearly increase in current draw rather than flickering on and off when met with resistance.

5.3 Ethical considerations

According to the IEEE Code of Ethics section 7.8 subsection II.9 [7], we have a responsibility to not harm others during the course of creating and also testing our project. The one concern that comes with a haptic-feedback glove is that the motors will apply torque to the user's hand and could cause bodily harm to the user. We have implemented a couple of risk mitigating factors into our design that should prevent any injury to the user. The first factor is a maximum current cutoff for the brushless motors using the ODrive. Once the motor current draw exceeds a predefined threshold the ODrives enter a fail state where the motors will no longer apply any forces to the user's hand until the program is reset. The cutoff is currently set to 50A which is about 10 lbs of force, any force exceeding that will shut down the motor such that we do not injure the user. Another feature we implemented was that the motors stop pulling on the user's hand once at the home position. The home position in this case is a hand in a resting open position. We implemented this safety feature so as to not allow the motors to pull back on the user's hand any further than what is comfortable so that we do not bend the fingers too far back and injure the user.

According to the IEEE Code of Ethics section 7.8 subsection I.5 [7], we must always strive to seek honest and critical feedback on our ideas. Hearing feedback can often be difficult, but we did keep an open mind to any suggestions made by the professors or our teaching assistant. In this case, our teaching assistant on numerous occasions made suggestions as to how we could improve our design. The first being how to restructure our PCB layout as to reduce noise across analog to digital components. Another suggestion was made as to how we could implement the current sensing circuit. In this design process we were happy to hear feedback and criticism so that we could produce the best project possible and in doing so we believe that feedback made our product better.

5.4 Future work

If we were to do this project again, there are several improvements that could be made. As suggested in the previous section, we would use better servos/controllers as the noise from the servos made force feedback nearly impossible to implement. We would also place a large output capacitor on our buck converter to prevent the servos from browning out our power supply, removing the need for a benchtop power supply. Another problem that we would take more heed with is assembling the PCB. Several times during the construction process we mismanaged wires and accidentally switched 5V and ground resulting in any active components on the PCB to be damaged or destroyed.

References

[1] T. Nilsen, M. Hermann, C. S. Eriksen, H. Dagfinrud, P. Mowinckel, and I. Kjeken, “Grip force and pinch grip in an adult population: Reference values and factors associated with Grip Force,” *Scandinavian Journal of Occupational Therapy*, vol. 19, no. 3, pp. 288–296, 2011.

[2] B. Vedder, “Vedderb/BLDC-hardware: Brushless DC Motor Controller from Benjamin Vedder,” *GitHub*, 03-Oct-2015. [Online] Available: https://github.com/vedderb/bldc-hardware.

[3] “Getting Started.” *Getting Started - ODrive Pro Documentation 0.6.3 Documentation*, ODrive Robotics, 2022, [Online] Available: https://docs.odriverobotics.com/v/latest/getting-started.html.

[4] “Atmega2560-16AUR: Digi-Key Electronics.” Digi, Digi-Key Electronics, 2022, [Online] Available: https://www.digikey.com/en/products/detail/microchip-technology/ATMEGA2560-16AUR/2357022 .

[5] “TSC211ICT STMicroelectronics: Mouser.” Mouser Electronics, Mouser Electronics Inc., 2022, [Online] Available: https://www.mouser.com/ProductDetail/STMicroelectronics/TSC211ICT?qs=iLbezkQI%252BsjsYAfcpKipFQ%3D%3D.

[6] “TPS54560,” Texas Instruments., [Online] Available: https://www.ti.com/product/TPS54560.

[7] “IEEE Code of Ethics.” IEEE, 2022, [Online] Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

Appendix A Requirement and Verification Table

Table 3 Glove Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Must send position information and receive force information	Using an Arduino MEGA, confirm test program functionality. Read encoder positions and send force data to board and manually confirm forces.	Y
Variable Resistance from 0 to 10lbs	Using ODrive Current control, adjust the target torque value and confirm motor resistance changes to match the input	Y
Completely Arrest hand movement and hold in place	Using ODrive position control, set a target position and try to move the glove from the set position, +/- 10° of deflection is acceptable	Y

Table 4 Manipulator Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Must be controlled via servos and have 180° of motion	Using an Arduino MEGA, control each servo independently and confirm that each one reaches 180° (+/- 5° on each side) even after the entire hand is assembled.	Y
Must report current consumed by servos via a shunt resistor and a high-side current sensing amplifier chip	Apply increasingly larger forces (between 0-50N) on the fingers of the manipulator. At each 5N increment we will record the output of the amplifier chip so that we know what kind of force to apply back to the glove at that voltage.	Y
Receive position commands via PWM and mimic finger positions of the glove subsystem	Ensure that actual finger placement versus manipulator finger placement is within 5° of each other.	Y

Table 5 Microcontroller Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Receive analog signal and have sufficient variation for ADC conversion	Target a variation of 0-5v +/- .1V for the input to the ADC	Y
Send and receive data	Using a test program, verify that the Odrives are being properly controlled by the MCU	Y
Send position data over PWM to Servo	Using a Oscilloscope, verify the correct waveform and frequency, then confirm the servo is moving in response to the signal.	Y
Process data and send commands quickly (Less than 1s)	Using the glove, change the position of the encoder and verify that the time for the manipulator to move is less than 1s	Y

Table 6 Power Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
5v power from buck converter to microcontroller	Using a multimeter, confirm voltage output is between digital high (5v-7v) within +/-.1v	Y
Lipo power system must be isolated from 5v power supply	Using a multimeter, monitor the voltage across the 5v power rail from the buck converter. We will rotate the brushless motors quickly and suddenly bring them to a halt. If the 5v	Y