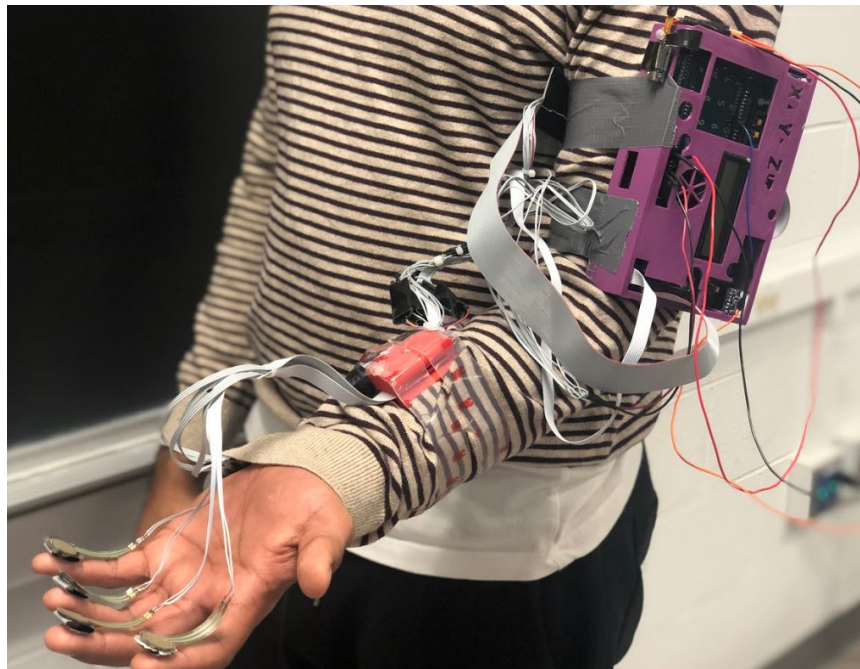University of Illinois Urbana Champaign

PHYS 398 DLP

Exploring an Inexpensive Sensor Package for Prosthetic Limbs

March 28, 2019

Vrain Ahuja
Nabarun Banerjee
Kunal Bhatia
Kyle Holben

Professor: George D Gollin

Teaching Assistants: Michael Habisohn, Maddie Horvat

Contents

Abstract

A feature missing from modern prosthetics is the recreation of a sense of touch or haptic feedback. Although many amputees are able to use prosthetics to perform normal daily functions, they lack the sensory feedback that would come from a natural limb. This sparks opportunity to research a sensor package that is affordable and integratable into existing prosthetics. The objective of our project is to develop this sensor package so that it can improve the usability and functionality of prosthetics. By basing the hardware and software development platform on an Arduino, a variety of sensors can be incorporated into the prosthetic. With a total cost of $335, we were successful in creating a sensor package that provided a sense of touch for amputees by using fingertip force dependant vibrations delivered by haptic drivers.

Introduction

A survey of existing prosthetics showed that cheaper prosthetics in the market made of 3D printed parts, linkages, and hinges allow the prosthetics to grab and hold objects. These linkages made use of some form of elastic material or mechanical advantage and manipulation to function[2]. These inexpensive prosthetics generally lack the advanced sensors and controls that come with more high end prosthetics. The average price of advanced prosthetics can range from $50,000 - $60,000 with some ranging as high as $100,000.

Our project aimed to study prosthetic sensor packages based on the Arduino platform that can be retrofitted onto existing 3D printed prosthetics to upgrade the functionality. These packages would ideally be inexpensive, durable, and lightweight. We hope the information gained from this study can be used to help amputees who do not have access to advanced prosthetics, such as those in developing countries.

The Hero Arm[3] (Figure 1), is an existing prosthetic that is affordable in developed countries. It utilizes several premium features and served as an inspiration for our project. A few of these features include muscle actuated controls, 180 degree wrist rotation, and a comfortable fit. The Hero Arm is able to maintain lower cost by using recycled materials and consumer grade electronic parts. Despite their efforts to create an affordable solution, the Hero Arm is priced at $6,000 and remains beyond the reach of the 1.4 million amputees in developing countries, where the average annual income ranges from $500 - $5000.

Figure 1: Hero Arm by Open Bionics

Our group identified that the most important feature lacking from both inexpensive and expensive prosthetics was touch based, or haptic, feedback. Haptic feedback devices (often called haptics) make the use of technology to simulate and reproduce sensations that would be felt by a user interacting directly with physical objects. Another function we wanted to implement was capacitive touch. We decided to monitor the general use case scenarios of prosthetics in order to analyze loads, temperatures, bumps, tilts, etc. experienced by an arm on a regular basis. These data will aid in understanding the parameters that prosthetics should be built for, effectively reducing production costs and improving usability.

Problem Statement

Out of every 1000 people, 1.5 are amputees[1] and 0.3 are born missing one or more limbs[19]. The current population of people missing limbs is estimated at about 10 million. Of this, 30% have upper limb amputations, putting their number at about 3 million worldwide. Within these 3 million, 2.4 million are estimated to be from developing countries. These amputations can be attributed to war, invasion aftermaths, political instability, and disease. The upper limb is defined as the region of the body extending below the deltoid and including the upper arm, forearm, wrist joint, and hand (see Figure 2). Of the upper limb amputations, 59% are below the elbow, 28% are above elbow including elbow disarticulation, and 13% are of the shoulder or wrist. This puts the number of below elbow amputations in developing countries at 1.4 million.
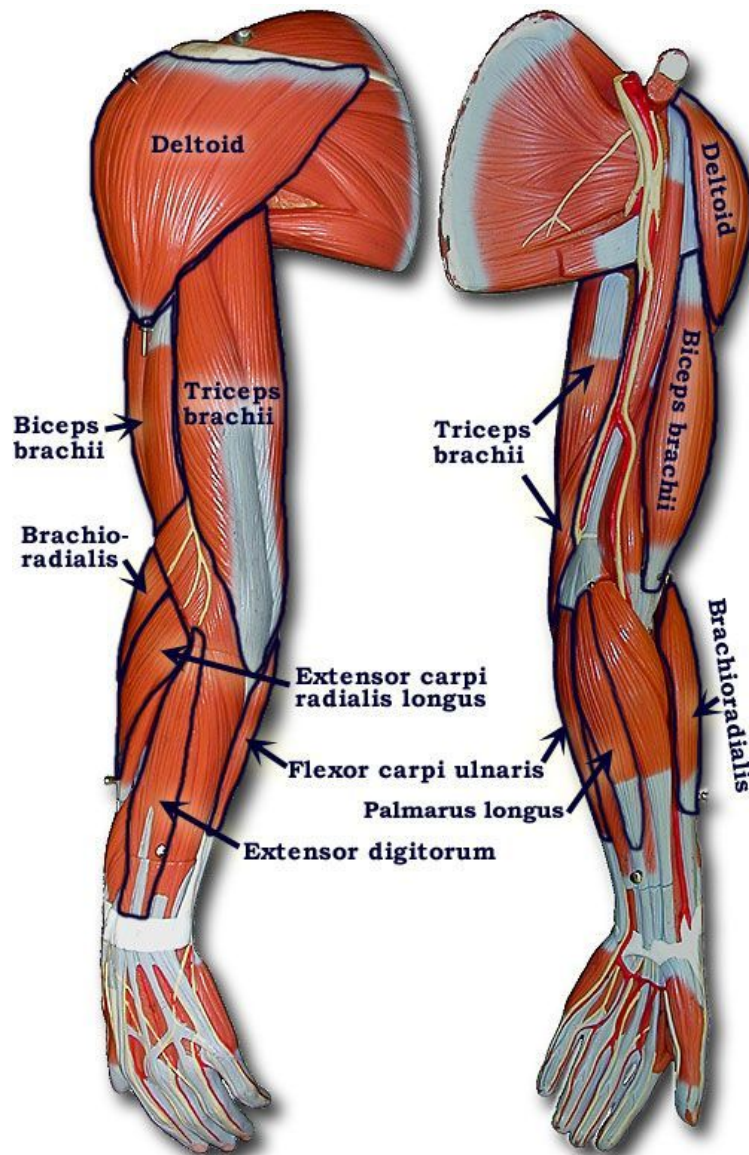
Figure 2: Anatomy of an Arm

       Prosthetics used in underdeveloped nations experience more rugged wear as compared to prosthetics in developed countries. Due to the absence of disability benefits, amputees are forced to continue work in order to sustain a livelihood. This results in the prosthetics seeing near maximum threshold loading on a regular basis. Because of the extreme use conditions, these prosthetics require frequent maintenance or repairs that can only be done by highly trained professionals. Unfortunately, technical expertise is limited and the deployment of these experts would drive up the cost of the prosthetic.

The point of our sensor package is to provide the relevant data needed for manufacturers to build low cost prosthetics. With clear data, over-engineering can be avoided, leading to a scalable and cost effective solution for the developing world.

Setup

Components:

The project was divided into two parts: real-time sensor implementation and offline data analysis.

The real-time sensor package contains all the sensors and components needed to provide the user with live feedback while using the prosthetic (Figure 3 & 4). Our choice of hardware components was dedicated to measuring the force at the users fingertips, indicators that alert users of fingertip contact, and actuators that provide haptic feedback. Detailed descriptions of individual hardware components are listed on page 6.

The offline data analysis portion of the project encompassed python coding and data modeling to visualize the general use case scenarios. Observations of this data led to our results and conclusion.
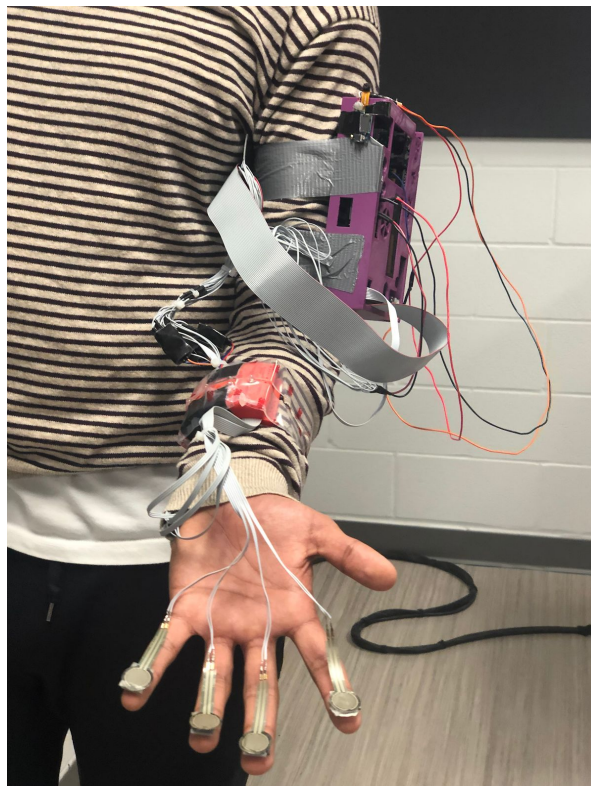

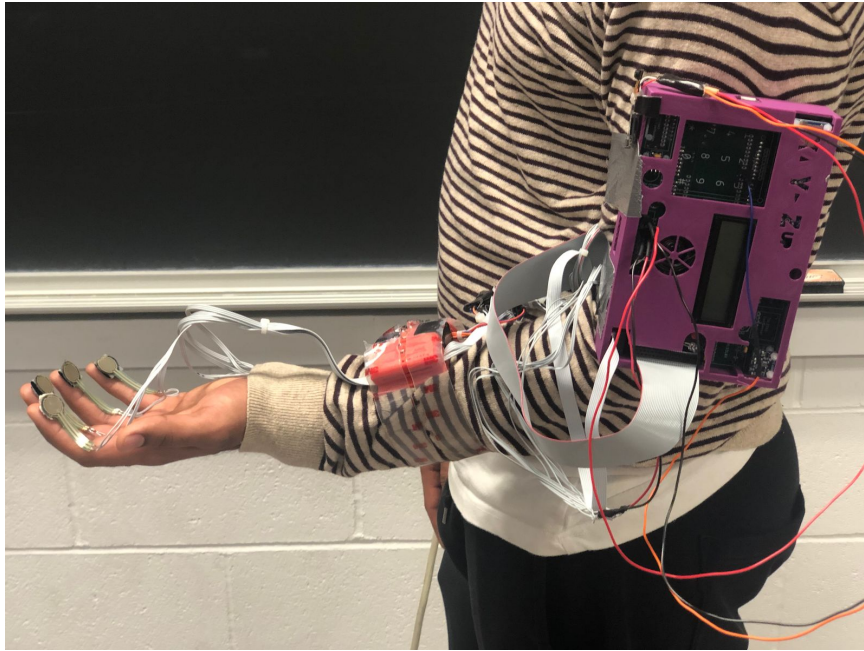Figure 3: Complete Sensor Package On User - Front View

Figure 4: Complete Sensor Package On User - Profile View

Parameters:

The parameters that our group decided to focus on for the sensor package were the following:

1. Force on each fingertip
   a. The force sensitive resistors were taped onto a latex glove on the pad of each fingertip.
2. Proximity of fingertips from object
   a. The capacitive touch sensor were taped in front of the force sensitive resistor.
3. Acceleration and angular velocity in the x, y, and z direction.
   a. The 9-axis accelerometer was placed on the Arduino board, which was strapped onto the left shoulder of the model.

Hardware:

The chosen electrical hardware is as follows. The devices are numbered according to the indicated objects in Figure 5 below. The technical specifications for each device is explained in the Appendix.
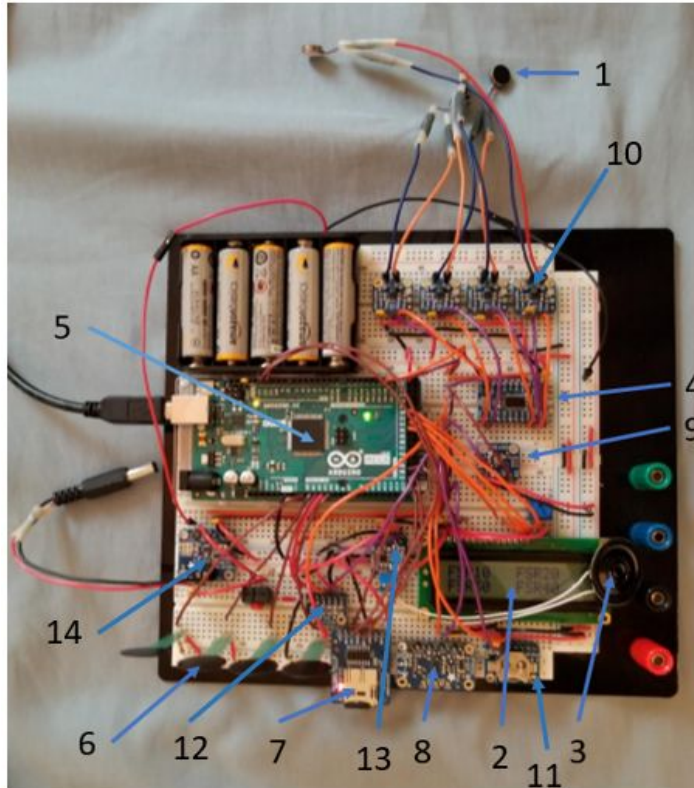
Figure 5: Sensor Package Wired on Breadboard

1. Vibrating Mini Motor Disk - A collection of four of these motors provide the haptic feedback to the users forearm. They are driven and controlled by the DRV2605L Haptic Feedback Controller Breakout

2. LCD[11] - The LCD is used to display the real time forces on each of the FSRs. It is used to monitor and debug the system in the sensor package. It also aids in calibration of the package.

3. Speaker - This is used to give the user auditory feedback when the prosthetic reaches its maximum load rating.

4. TCA9548A I2C Multiplexer[14] - Communication with each individual device would not be possible without the use of an I2C multiplexer.

5. Arduino Mega 2560 - The Arduino Mega is the heart of the whole project. It communicates with all the sensors and contains all the logic required for our package.

6. Force Sensitive Resistor (FSR)[5] - An array of four of these sensors are used to monitor the force on each finger tip.

7. Micro SD Card Breakout Board[15] - The micro SD breakout board is used for data acquisition in the sensor package. The SD card breakout first creates a text file named using the RTC time stamp.

8. LSM9DS1 Accelerometer + Gyro + Magnetometer 9-DOF Breakout[4] - This sensor is used to monitor the acceleration of the upper limb in the x, y and z direction and reports in m/s^2. The sensor also keeps track of the angular velocity with respect to the x, y and z axis and reports it in degrees/ second.

9. Adafruit BME680 - Temperature, Humidity, Pressure and Gas Sensor[7] - This sensor is used to measure temperature, humidity and pressure. This helps to better understand the conditions a prosthetic is used in.

10. DRV2605L Haptic Feedback Controller Breakout[9] - The forces on each finger tip is communicated to the user via haptic feedback. The intensity of the vibrations generated by the haptic feedback driver correspond to the magnitude of force on each finger tip.

11. DS3231 Precision RTC Breakout[8] - The Real Time Clock (RTC) is used to time various components on the sensor package. Most importantly, it is used to time the data acquisition program and help relate it to the pictures captured by the camera.

12. MCP4725 12-Bit DAC[12] - The Digital to Analog Converter (DAC) takes the digital signal generated by the Arduino and converts it into an analog signal to pass onto the amplifier.

13. PAM8302A Adafruit Mono 2.5W Class D Audio Amplifier[13] - The amplifier takes signals provided by the DAC and amplifies them. It then sends those signals to the speaker

14. INA219 High Side DC Current Sensor Breakout - 26V ±3.2A Max[21] - Allows us to control the current source to the Arduino between the USB port and the battery..

    Not Shown:

15. Spy Camera - The camera is used to capture photos when the force sensitive resistors trigger it. It is used to correlate loading conditions to the magnitude of force, acceleration, or angular velocity experienced by the prosthetic.

16. LED - The LEDs are used to provide visual feedback to the user based on capacitive touch sensors. They are the LED's built into the capacitive touch sensors, remapped and wired to be within the users field of vision.

17. Capacitive Touch Sensors[6] - A collection of four of these sensors determined proximity of the fingertips to an object. The change in capacitance is used to trigger an external LED to provide visual feedback to the user.

Item	Price	Quantity	Total
Arduino Mega 2560	$38.50	1	$38.50
LSM9DS1 Accelerometer	$14.95	1	$14.95
Force Sensitive Resistor	$7.00	4	$28.00
Capacitive Touch Sensor	$5.95	4	$23.76
Adafruit BME680	$22.50	1	$22.50
DS3231 Precision RTC	$13.95	1	$13.95
Mini Spy Camera	$12.50	1	$12.50
Adafruit DRV2605L Haptic Controller	$7.95	4	$31.8
Vibrating Mini Motor Disc	$1.95	4	$7.80
LED	$0.14	4	$0.56
LCD	$9.95	1	$9.95
MCP4725 12-Bit DAC	$4.95	1	$4.95
Adafruit Mono 2.5W Class D Audio Amplifier	$3.95	1	$3.95
Speaker	$1.95	1	$1.95
TCA9548A I2C Multiplexer	$6.95	1	$6.95
Micro SD Card Breakout Board	$7.50	1	$7.50
Printed Circuit Board Setup	$100	1	$100
Printed Circuit Board	$6	1	$6
TOTALS		**33**	**$335.57**

Table 1: Cost of Individual Hardware Components

Experimental Setup

The sensor package was preliminarily assembled on a breadboard to determine which sensors were to be used and how to implement the Arduino software accordingly (Figure 5). The breadboard functioned as a simple test bed to try various sensor configurations and coding methods before applying the setup to a Printed Circuit Board (PCB) (Figure 6).

Once the final experimental setup was chosen, it was mounted on a PCB and place in a 3D printed plastic case (Figure 6).The PCB housing consisted of a visible LCD for live monitoring of the forces on each fingertip, a speaker acting as an overload protection warning, and a 9 axis accelerometer mounted inside. Additionally mounted to the PCB was the BME 680, I2C multiplexer, LCD, DAC, amplifier, SD card reader, and the RTC.
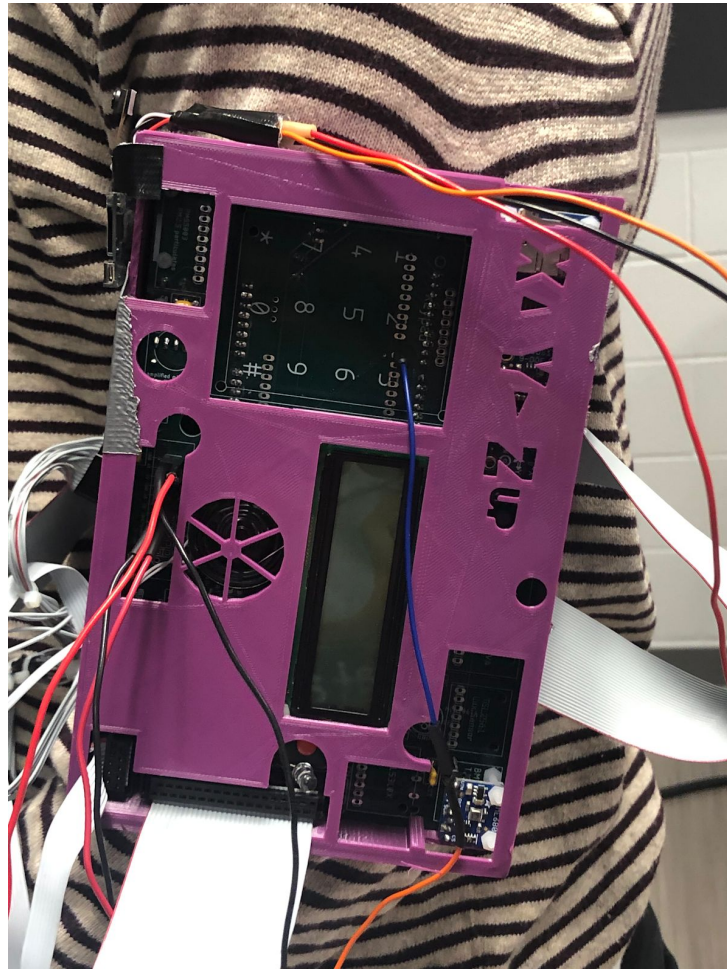

Figure 6: Sensors Mounted on PCB in 3D Printed Housing

The real time sensor package that would be mounted on the fingertips of a prosthetic consisted of a two fold system. The first layer, mounted directly to the fingertips, were four force sensitive resistors (FSRs) as shown in Figure 7. The second layer, mounted underneath the FSRs consisted of capacitive touch sensors as shown in Figure 8. The force sensitive resistors were wired to the Arduino which interprets the incoming data and controls the haptic feedback drivers accordingly.

Figure 7: FSR Only on User's Hand

Figure 8: Capacitive Touch Sensor On Top of FSR

The haptic feedback drivers were mounted to the users forearm using a special, separate enclosure. This enclosure also has cutouts for LEDs that provide the user with visual cues from the capacitive touch sensors as shown in Figure 9.

Figure 9: LEDs in 3D Printed Housing on User's Forearm

The PCB was attached to the wearer's left arm on top of the shoulder with both velcro and duct tape to ensure the box would not have extraneous movement. We then attached the spy camera to the right side of the box pointing downwards at the hand as shown in Figure 10. The contraption containing the LED lights was taped to the bottom side of the left forearm using double sided tape, where the wearer would receive vibrating responses as feedback from touch. Since our tests were conducted by users with limbs, a latex glove on their left hand was used in lieu of a prosthetic. The force sensitive resistors were super glued to the fingertips of the glove and the capacitive touch sensors were attached underneath.
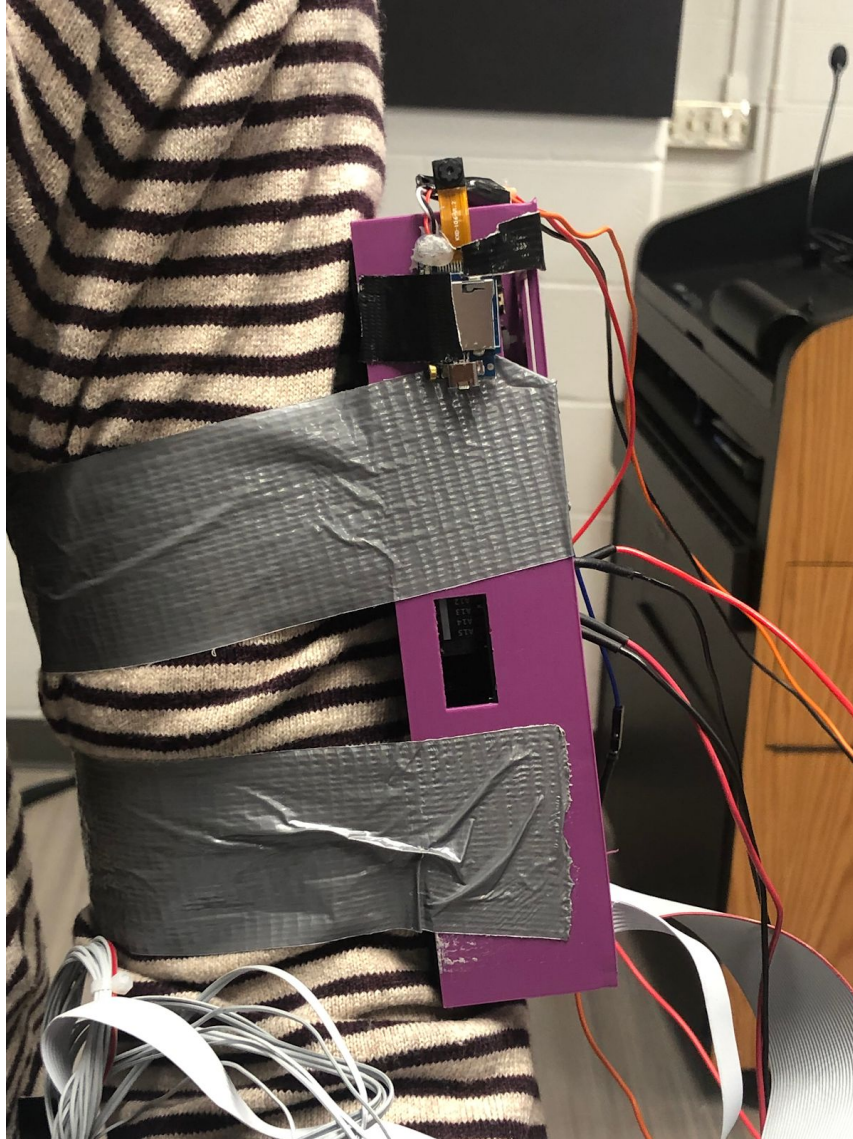
Figure 10: Spy Camera Attached to PCB Housing

Data Acquisition

Our data acquisition was broken up into 5 different prosthetic use cases. In each use case, the measured data was resistance in ohms, accelerometer in m/s^2 and angular velocity in degrees/second . Our python code converted the values from ohms to newtons by dividing by ten, a ratio provided from the manufacture.

Case 1) **Typing and touching on a laptop and smartphone:** Our goal was to capture data that would allow a prosthetic user access to jobs that are conducted while sitting on a chair upright. These activities lasted for a duration of 500 seconds. The typing was conducted on an 2017 Apple MacBook Pro. The iPhone activity is broken down to lifting the phone itself, and swiping, or touching, left or right on a popular dating app (Tinder).

Case 2) **Gripping, lifting, and moving a rigid body:** Our goal is to analyze this motion as it is something that is quite common for an everyday amputee. For example, opening a hinged door or roll-up shop entrance (a common activity for business owners in underdeveloped countries). Gripping the bottle required complete wrap around from the users hand and required significant control from my hand, and we believe this activity is a good match up to some of the activities mentioned earlier.

The rigid body we used was a 24oz stainless steel water bottle with a circumfrence of 8.8 inches. The activities conducted involved lifting the bottle to drink water and maintaining a firm grip of the bottle while the we screwed the top back in. These activities totaled 70 seconds.

Case 3) **Unlocking a bicycle lock and bicycling:** The activities included unlocking the cycle lock, wrapping the lock around the base of the cycle, holding the lock while mounting it, gripping onto the handlebars, and operating the brakes for the duration of the bicycle ride. These activities totaled 70 seconds.

Our study was conducted on a standard flat bar bicycle and a chain lock.

Case 4) **Soft bodies, was conducted with a trash bag:** The wearer lifted the empty trash bag from a table and held the bag open while a second person incrementally added objects in the bag. The objects were 2 water bottles, a textbook, and a metal rod. These activities lasted 80 seconds.

Case 5) **Tapping a standard wooden table:** The wearer repeatedly tapped a table 60 times in a row as fast as he could uninterrupted.

Each time the Arduino was powered, a new .txt text file was created. The file contained measurements of both acceleration and gyroscope in the x, y, and z directions as well as force, temperature, humidity, time, and an approximate photo count. Measurements were recorded by the Arduino every 150 milliseconds and stored on the Sandisk 8gb memory card.

In order to analyze the data recorded by the Arduino, we wrote a python script that reads the text file line by line. Using the strip function within Python, we created arrays of different data sets generate graphs.

The code calculates roll, pitch, and yaw using accelerometer and magnetometer data. We use a high-pass filter to reduce noise in Yaw. The angles help us to calculate the orientation of the device placed on the shoulder of the user. To filter the noisy yaw data we used a high pass filter with a cutoff frequency of half of our sample rate. This data allowed us to monitor the degrees of freedom required in most use cases and to study the correlation between the angular velocity and angles traversed by the user's shoulder.

In order to better identify the data, we used the camera to correlate activities with different force sensor readings. The camera was triggered every time the force sensors registered a change in resistance at any one of the fingers. This created a unique timestamp and increased the count for us to match up the graph readings with visual cues. Photos were taken at a frequency of 3 Hz. The photos however were not usable due to the angle of the camera.

The arduino was reading data at roughly 6.67 Hz, showing the real time force reading on the LCD and writing it to the SanDisk 8 GB memory card. Throughout our code, we have defined FSR1 as the index finger, FSR2 as the middle finger, FSR3 as the ring finger, and FSR4 as the pinky finger.

Observations

Case 1 - Typing

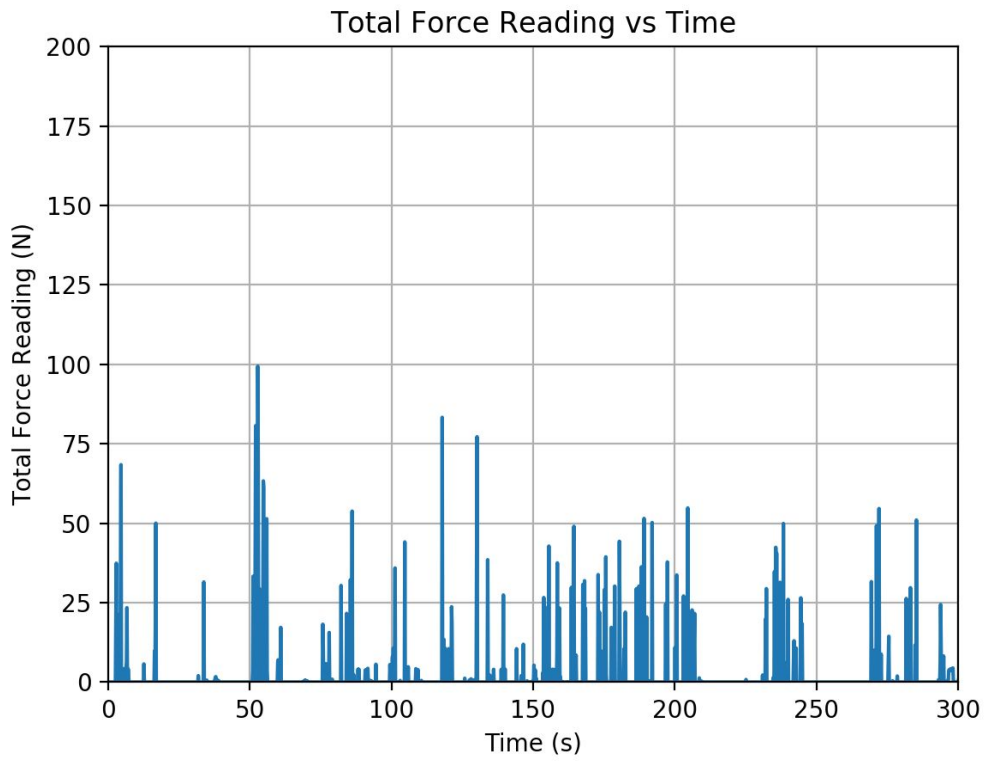Total force refers to the sum of the force experienced by each individual FSR.

Figure 11: Total Force Experienced on all Fingers While Typing on Keyboard
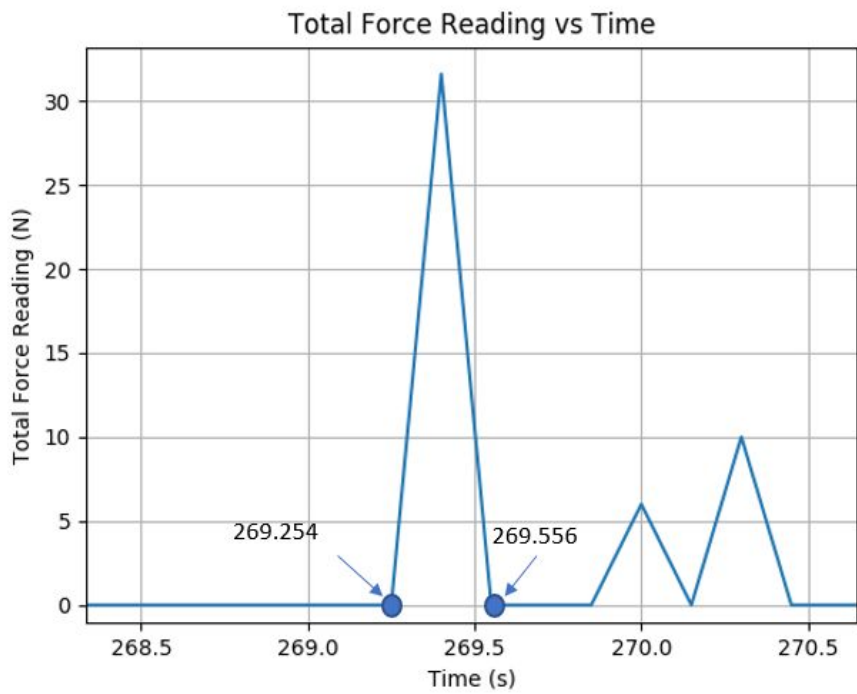

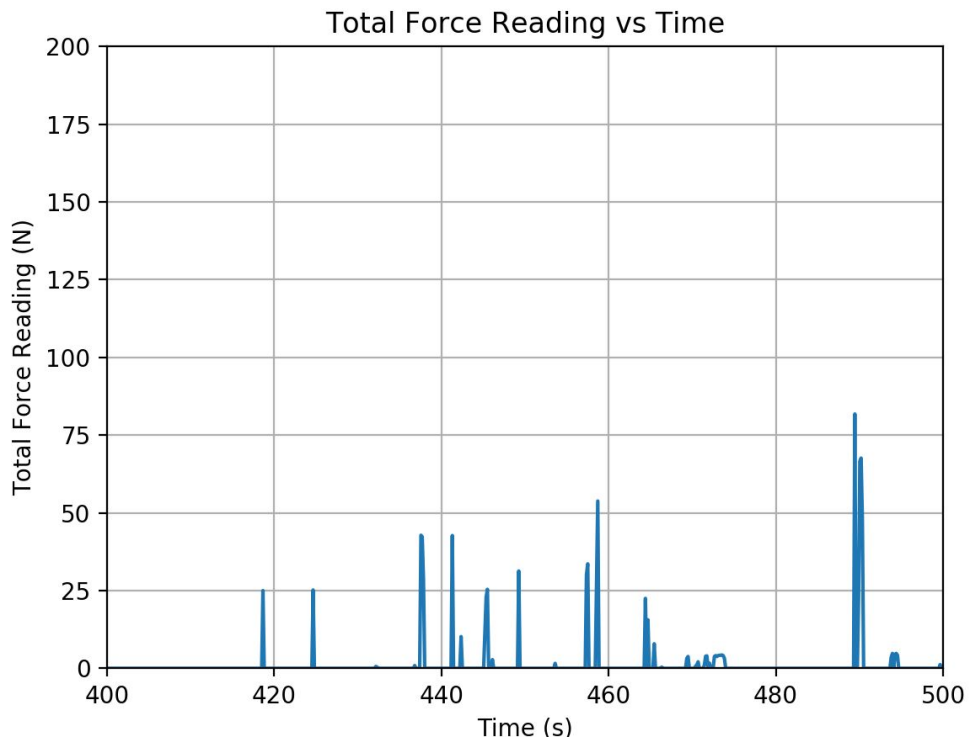Figure 12: Total Force Reading for Single Tap on Keyboard - Zoomed In

Figure 13: Total Force Experienced on all Fingers While Typing on Phone
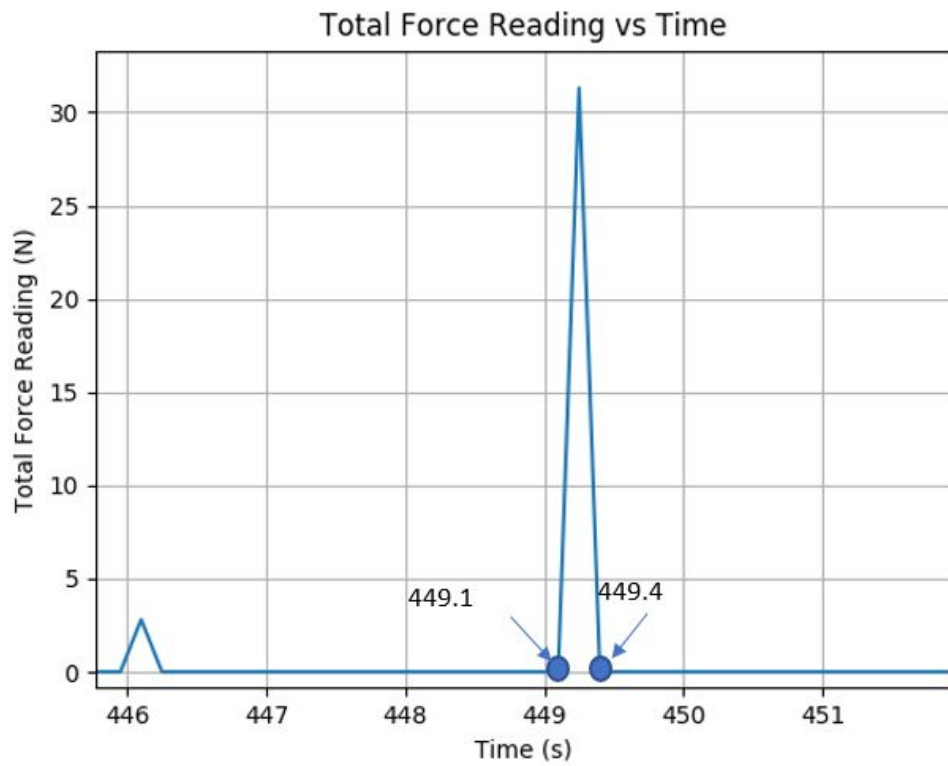

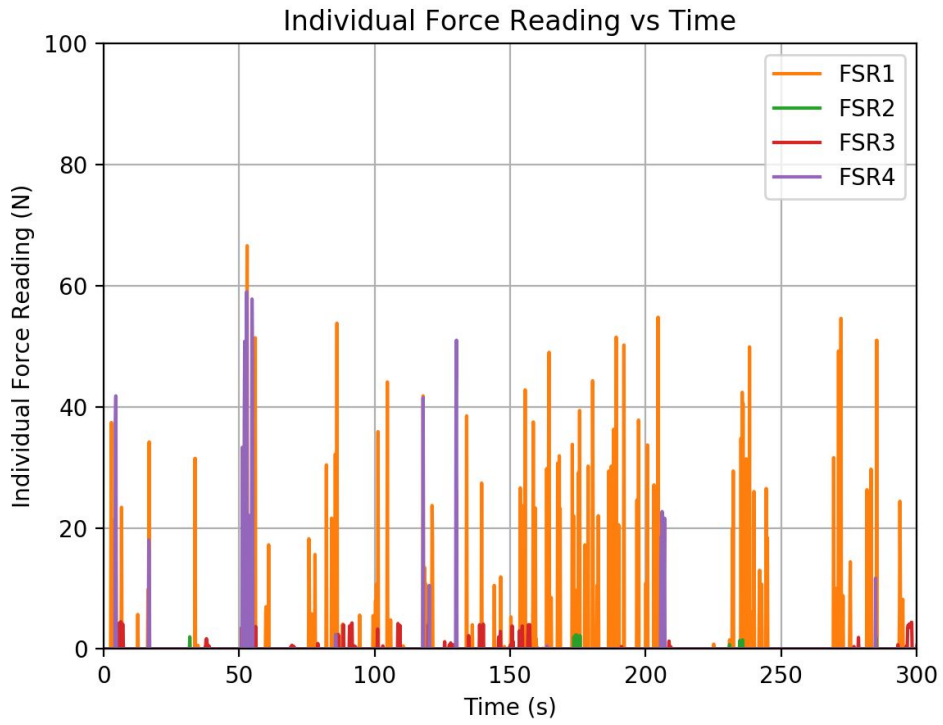Figure 14: Total Force Reading for Single Tap on Phone - Zoomed In

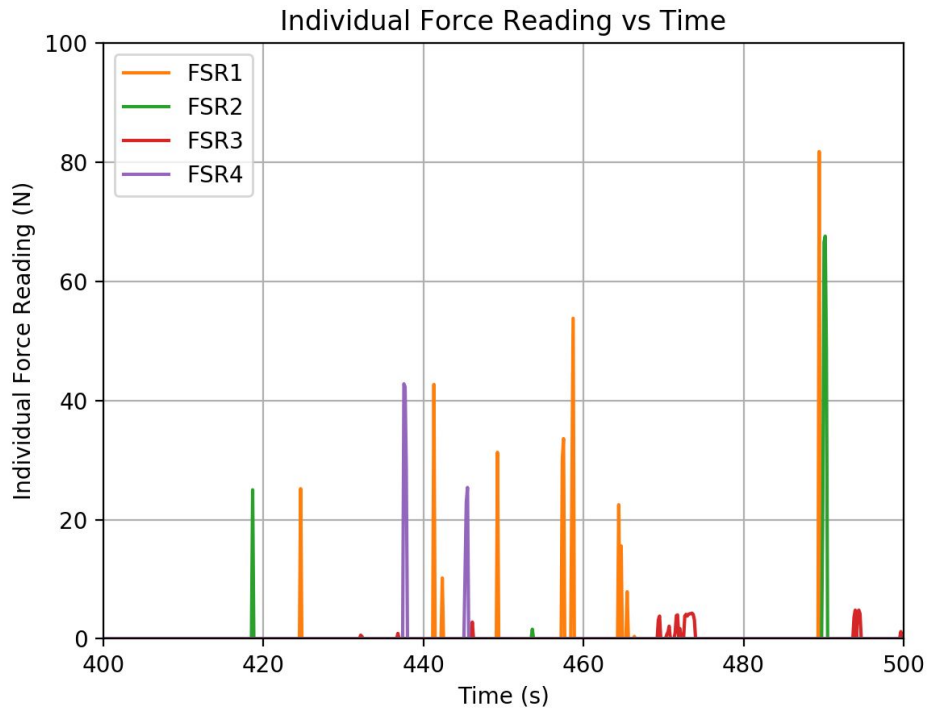Figure 15: Individual Finger's Force Readings While Typing on Keyboard

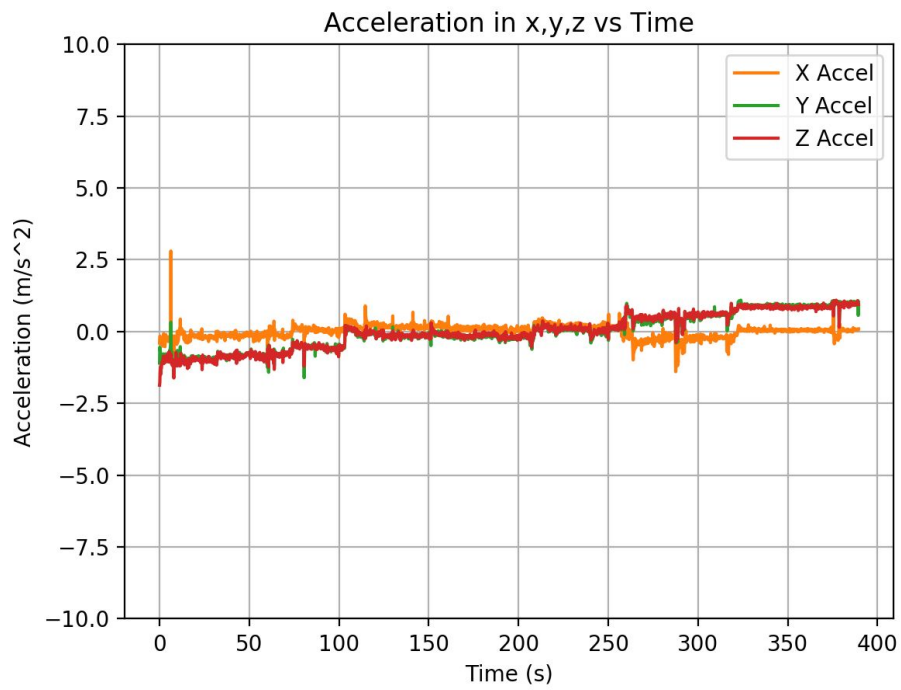Figure 16: Individual Finger's Force Readings While Typing on Phone

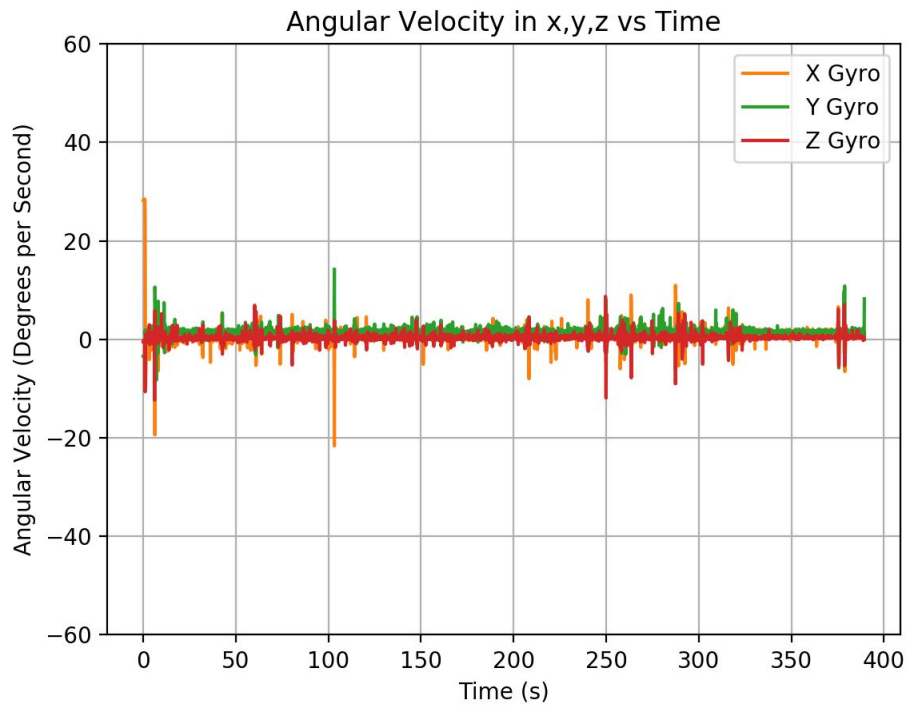Figure 17: Accelerations of Shoulder While Typing on Phone and Keyboard

Figure 18: Angular Velocities of Shoulder While Typing on Phone and Keyboard
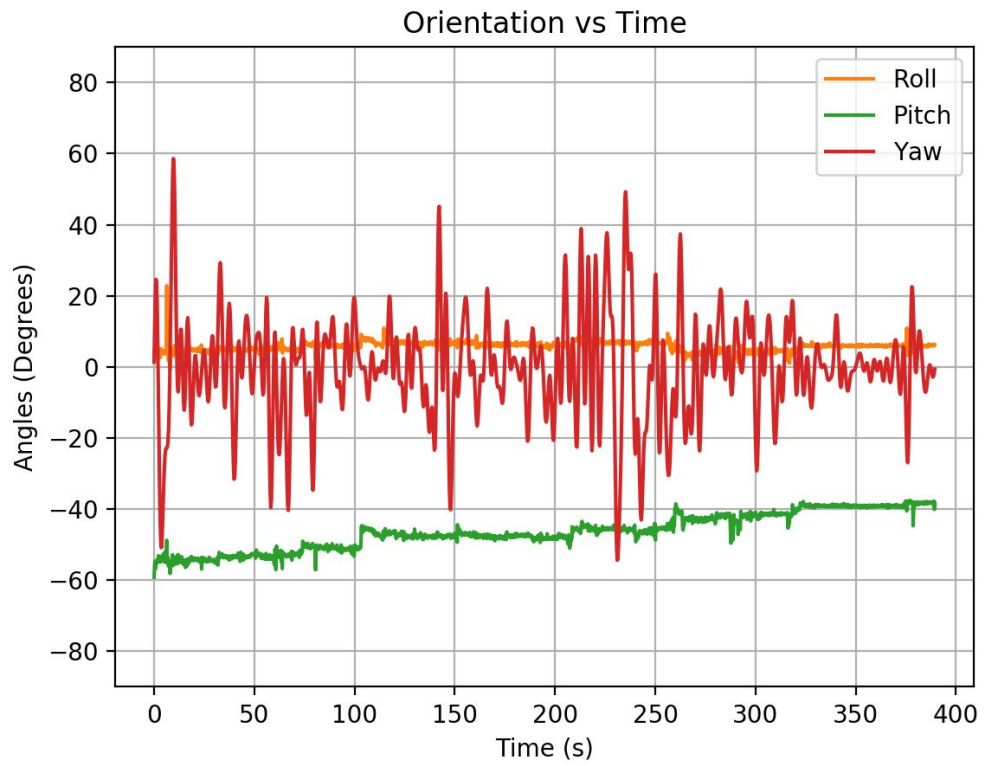
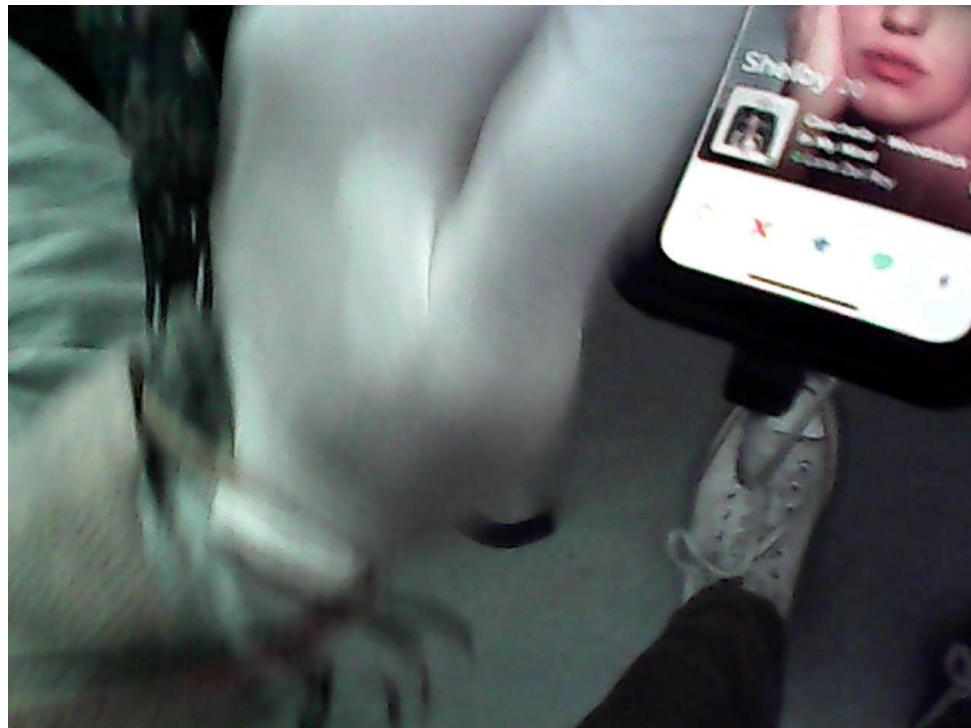Figure 19: Orientation of Shoulder While Typing on Phone and Keyboard

Figure 20: Spy Camera While Typing on Phone

In the case of typing on both the keyboard and the screen, we notice that the force of the taps vary from finger to finger. FSR1 or the index finger is triggered the most and FSR2 or the middle finger is triggered the least as see in Figure 15 and Figure 16. We also see that the average time between two keystrokes from the same finger is about 0.6 seconds for both cases of typing. The average value for the total force on the fingertips was about 14.83 N. By plotting individual taps while typing on a keyboard and typing on a phone, we also observe that both key presses take the same amount of time, about 0.3 seconds as shown in Figure 12 and Figure 14.

As typing is a stationary task, we notice no significant trends in the angular velocities, as seen in Figure 18, or in accelerations in the x, y and z directions, as seen in Figure 17. In Figure 20, the activity of FSR2 swiping on an iPhone corresponds to about 70N. An interesting observation from this study is that typing on a keyboard required significantly more force than typing on a phone screen. This can be seen in Figure 11 and Figure 13 from the difference in average forces, which corresponds to typing on a keyboard and a touch screen phone respectively. The total FSR reading for typing on a keyboard is higher than typing on phone as we use all 4 fingers as compared to using only one finger while typing on the phone. There was a single significant spike in angular velocity that corresponds to the user supporting themselves on their arms and sat upright on their chair. The orientation data was noisy but as expected in an activity such as typing, where rotation about the shoulder socket was common, the variations in the yaw was the greatest as shown in Figure 19.
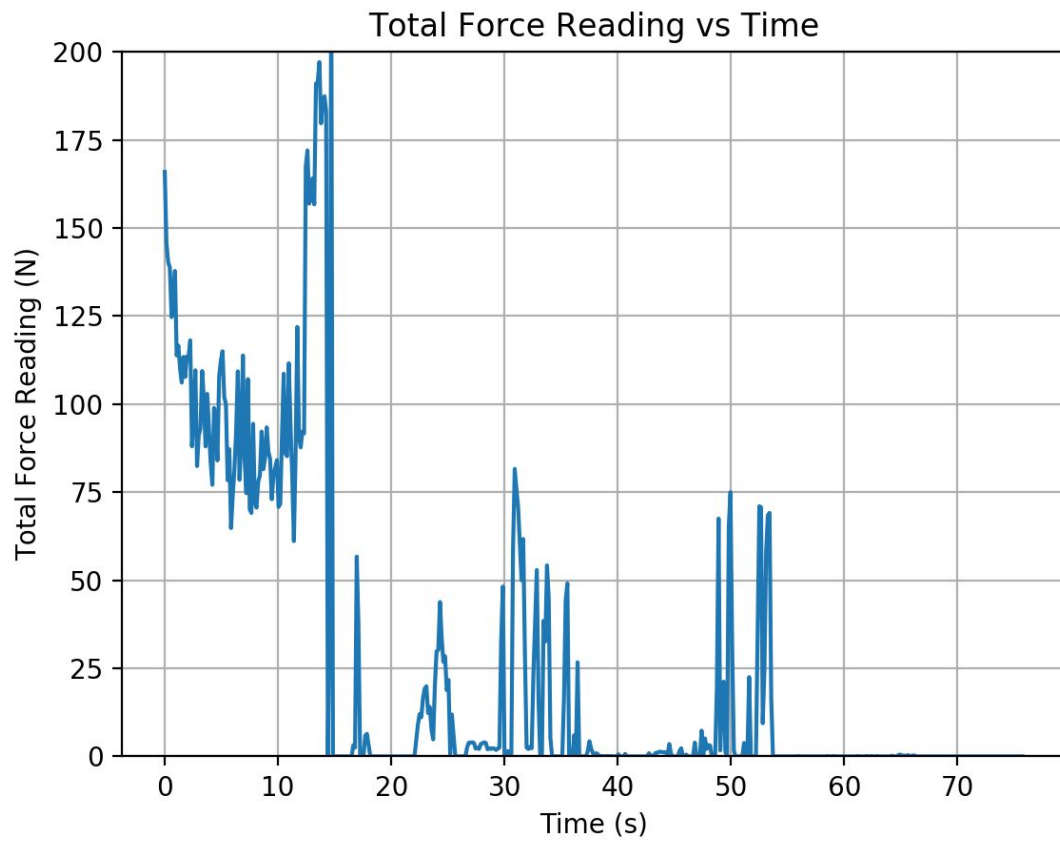
Case 2 - Rigid Body

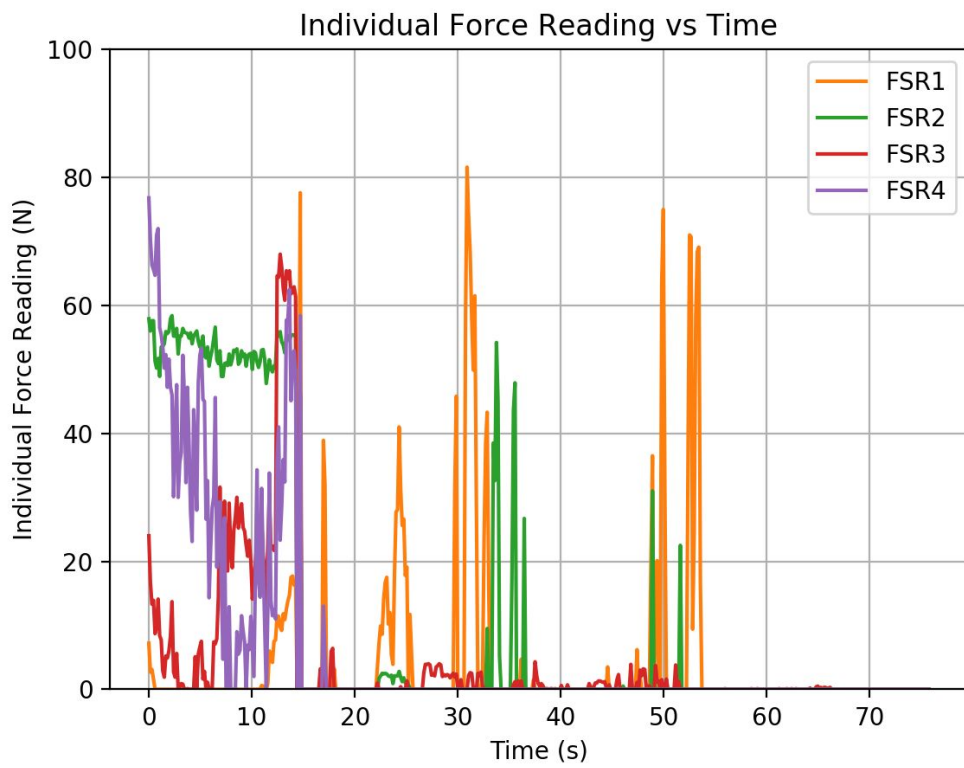Figure 21: Total Force Experienced on all Fingers While Picking up Rigid Body

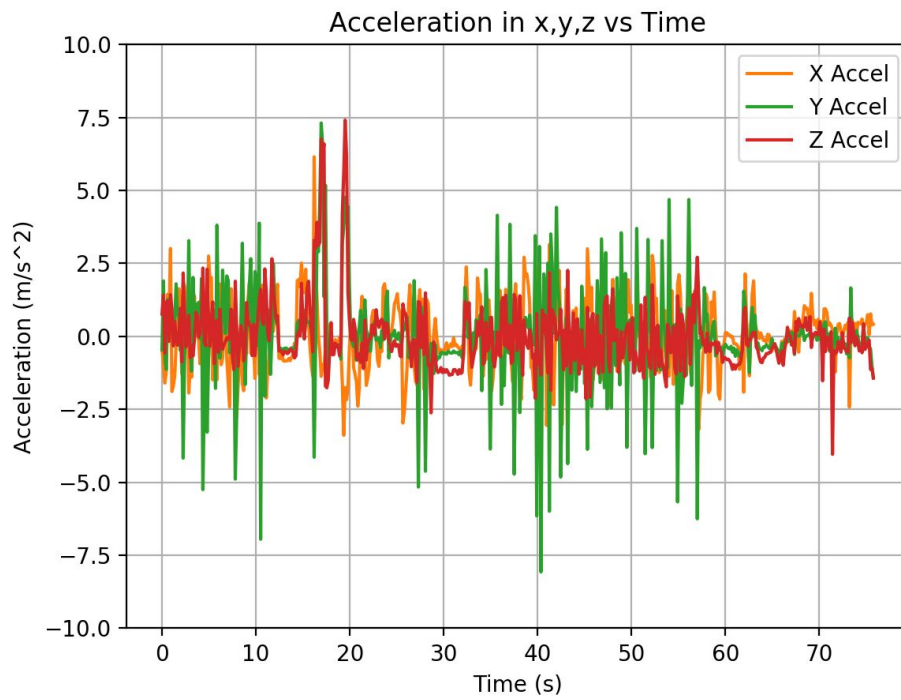Figure 22: Individual Finger's Force Readings While Picking up Rigid Body

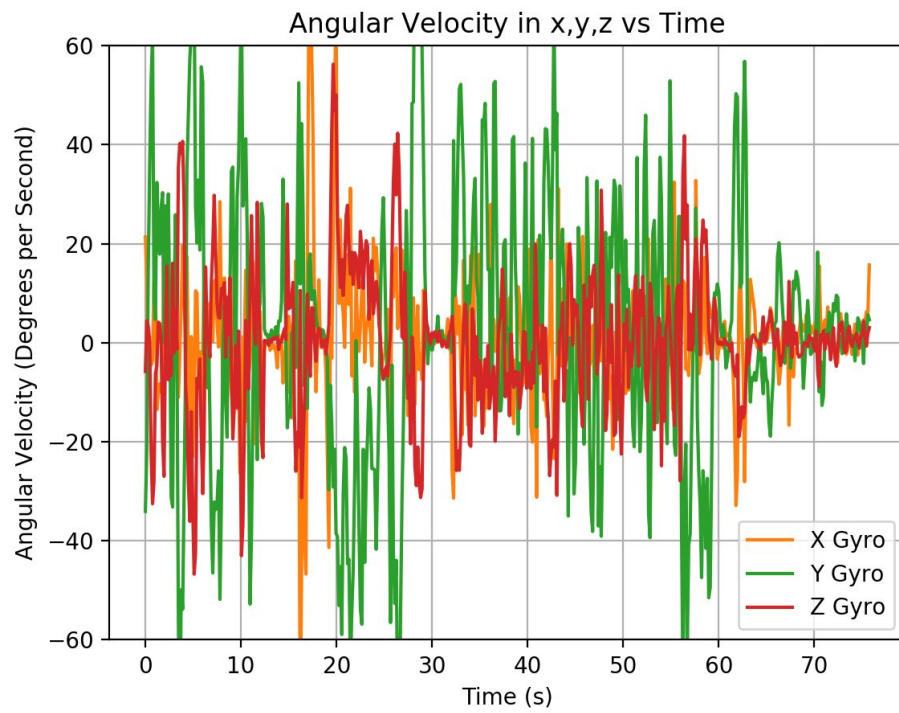Figure 23: Accelerations of Shoulder While Picking up Rigid Body

Figure 24: Angular Velocity of Shoulder While Picking up Rigid Body
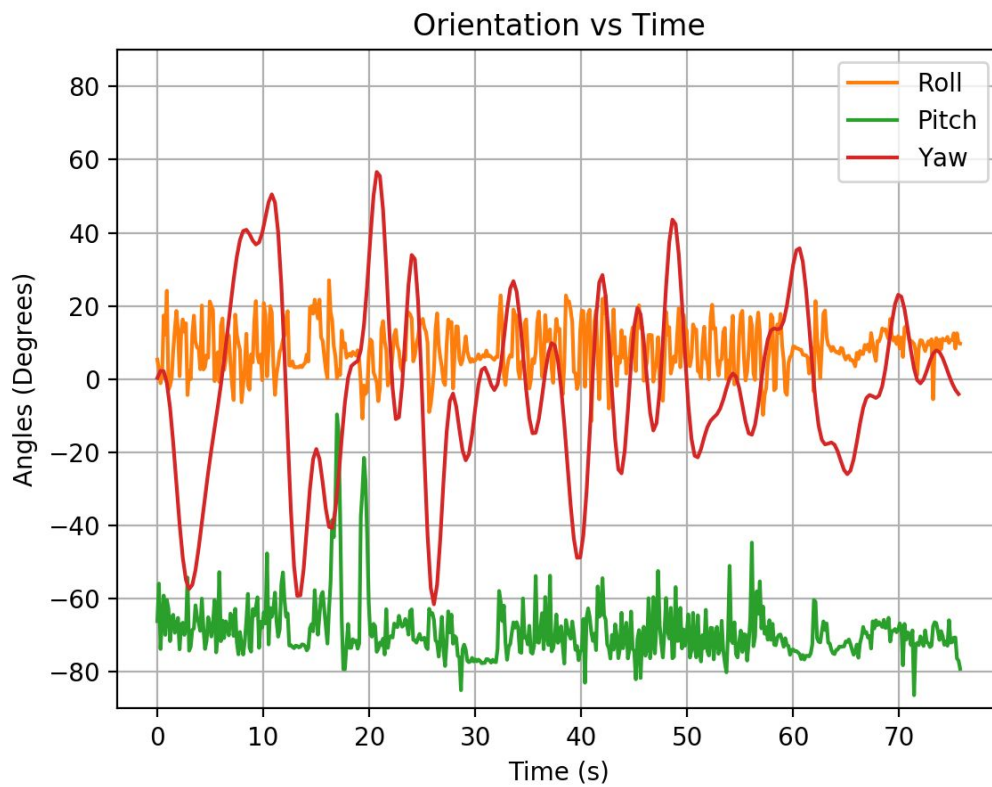

Figure 25: Orientation of Shoulder While Picking up Rigid Body

Figure 26: Spy Camera While Picking up Rigid Body

Case 2 consisted of picking up and handling rigid body objects such as a book, water bottle, closing a door, etc. The first thing that was apparent was that the loads were not equally balanced between all fingers and in most cases there was significant biasing of one finger over others as is clearly visible from Figure 22. There was care taken to use the fingers and hand naturally while manipulating rigid bodies. The user for the tests ensured that the grip was not biased towards using only finger tips while manipulating the objects. The average force for each finger is: FSR1 - 4.25 N, FSR2 - 11.03 N, FSR3 - 4.21 N, FSR4 - 6.10 N. The trends in the total force reading can be seen in Figure 21. The average for total force is 25.59 N.

From Figure 22 we see that the most used finger for picking up rigid bodies is the second, or middle, finger. In fact, the middle finger gets approximately twice the force per second than any other finger and almost half of average total force. From Figure 24, we see that the Y gyro has the most fluctuation, indicating that the arm/hand rotates more often compared than it lifts or extends. This trend is also shown in in the orientation plots with yaw dominating as shown in Figure 25. As the user moved about the room while performing these tasks there are slight jerks which show up in Figure 23. Figure 26 shows the spy camera triggered while picking up a water bottle.
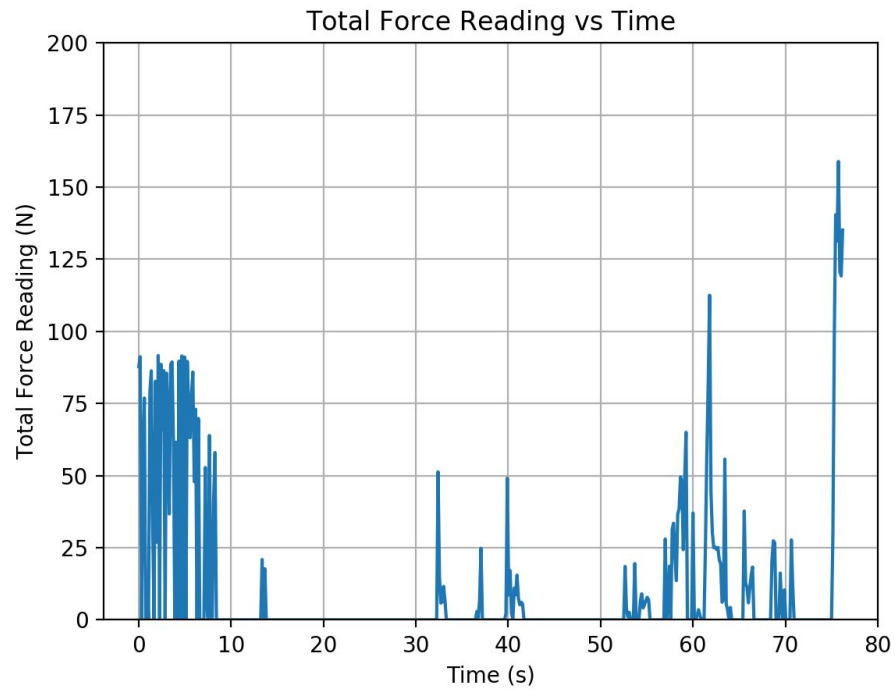
Case 3 - Biking

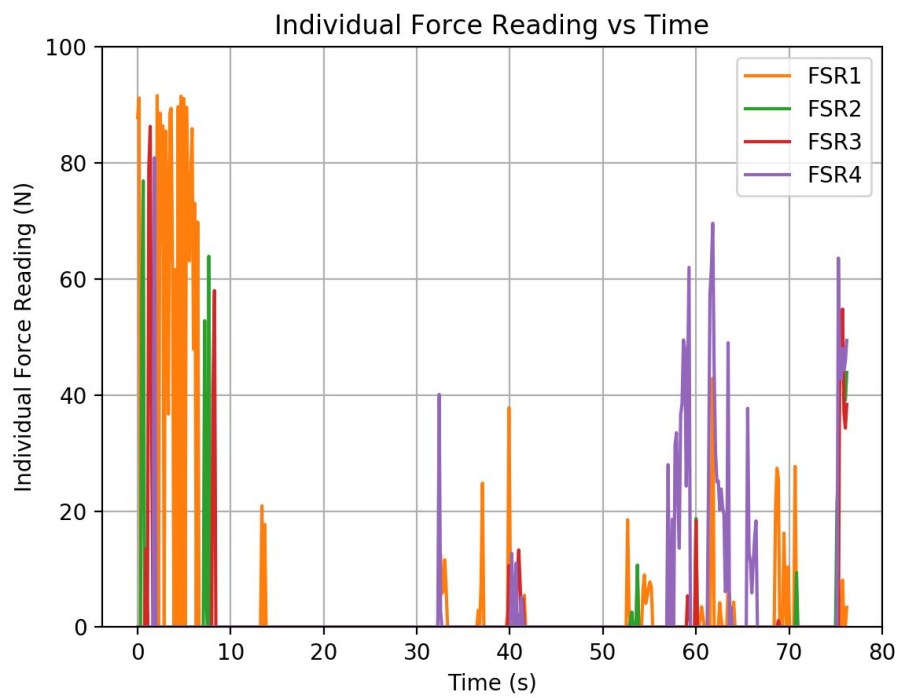Figure 27: Total Force Experienced on all Fingers While Biking

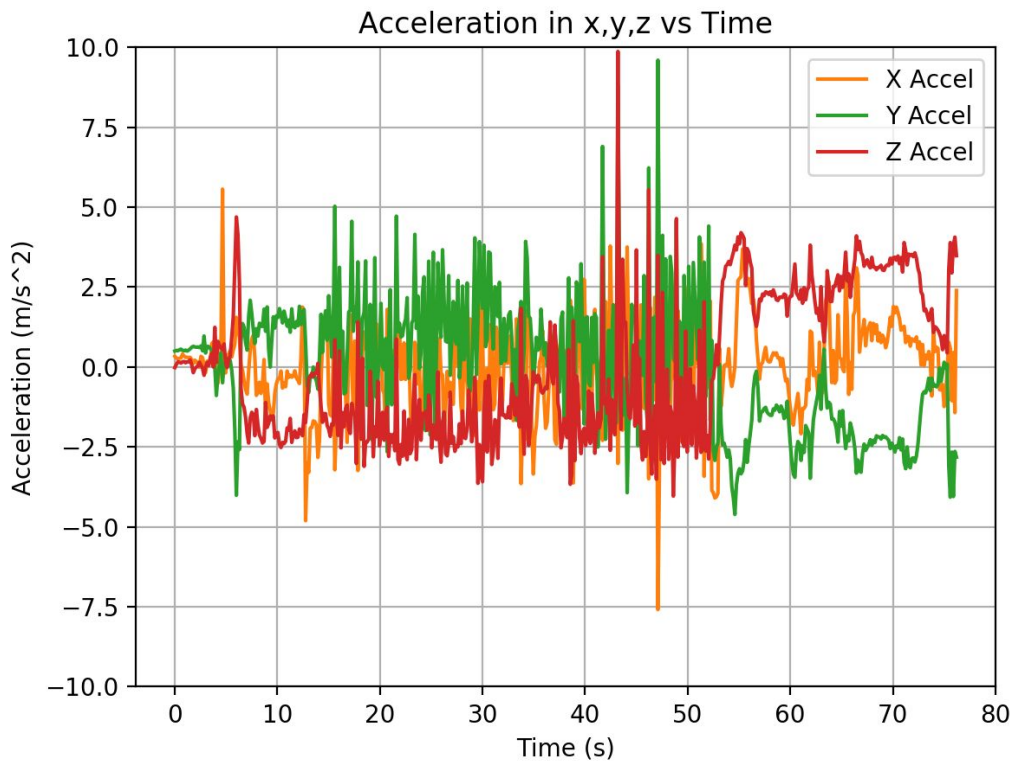Figure 28: Individual Finger's Force Readings While Biking

Figure 29: Accelerations of Shoulder While Biking
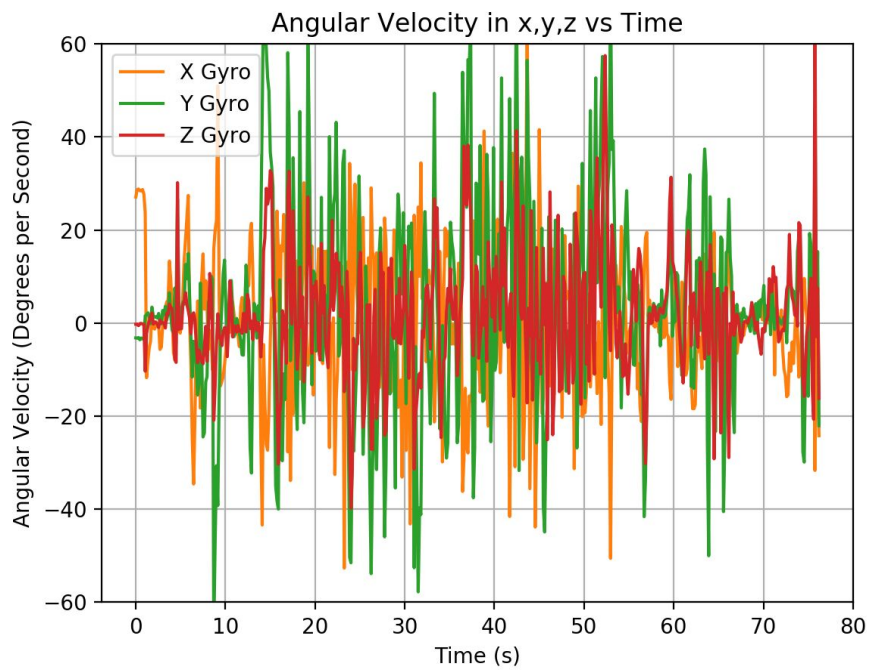
Figure 30: Angular Velocity of Shoulder While Biking
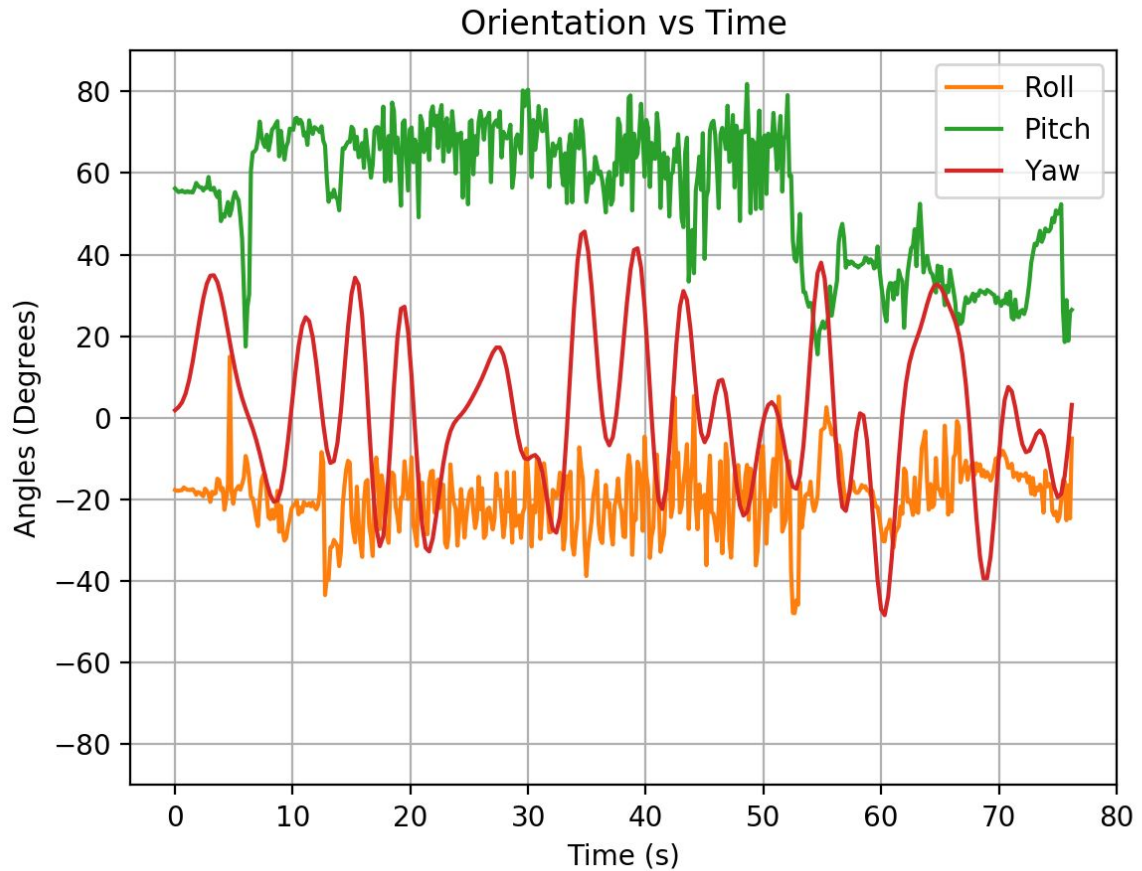
Figure 31: Orientation of Shoulder While Biking

The biking case was the best case for dynamic loading. The user put the sensors through a wide variety of loading conditions. The accelerometer picked up slight bumps while handling the bike, but the accelerations were less than 1g as shown in Figure 29. Also, the angular velocity was not the highest amongst all cases, but varied the most as shown in Figure 30 . It had peaks ~60 degrees per second magnitude in often. Figure 28 also points out that moving the bike initially was very heavy on the index finger and the total force for navigating the bike was much lower as shown in Figure 27. Orientation varied the most in magnitude in yaw, as shown in Figure 31, as was expected from biking, but pitch and roll were also significant during biking pointing towards complex arm manipulations required for biking.

Case 4 - Soft Bodies

Figure 32: Total Force Experienced on all Fingers While Picking up Soft Body

Figure 33: Individual Finger's Force Readings While Picking up Soft Body

Figure 34: Accelerations of Shoulder While Picking up Soft Body
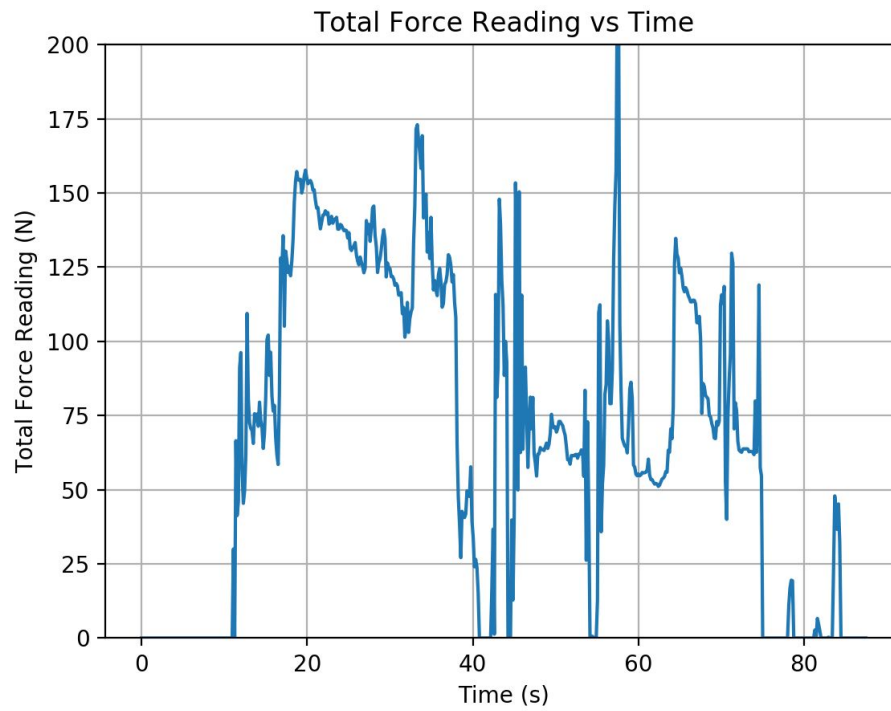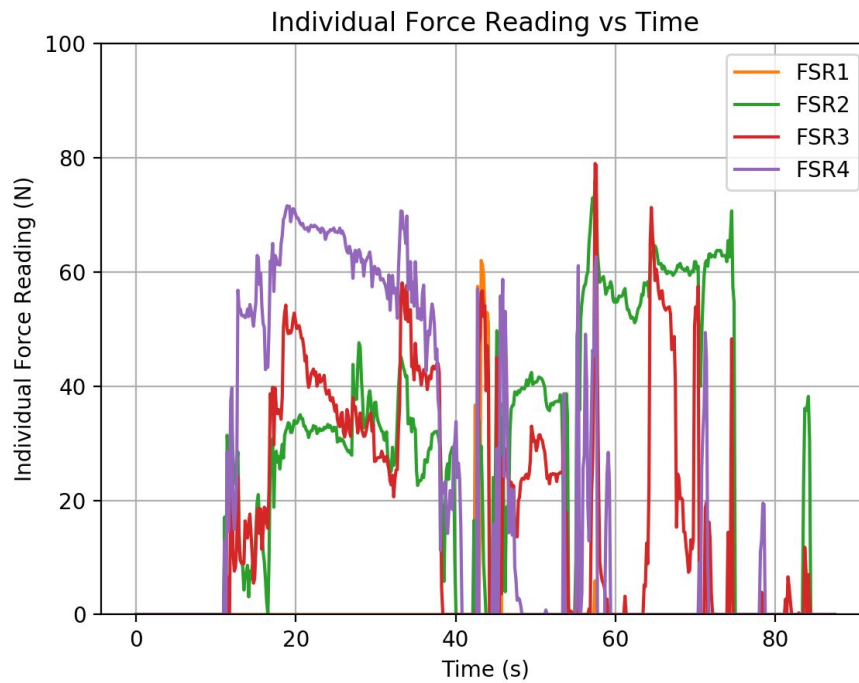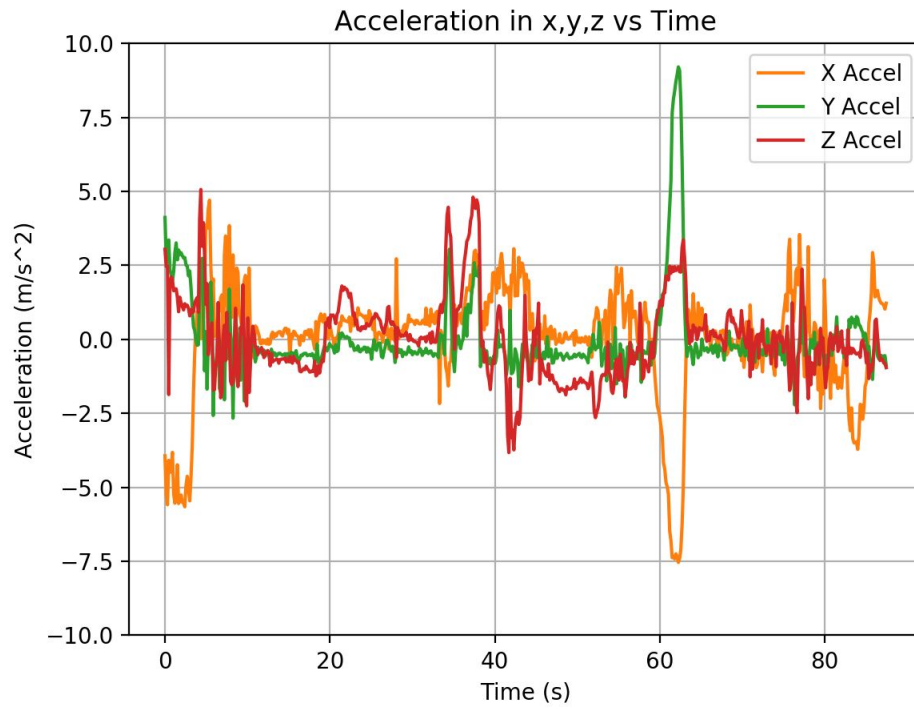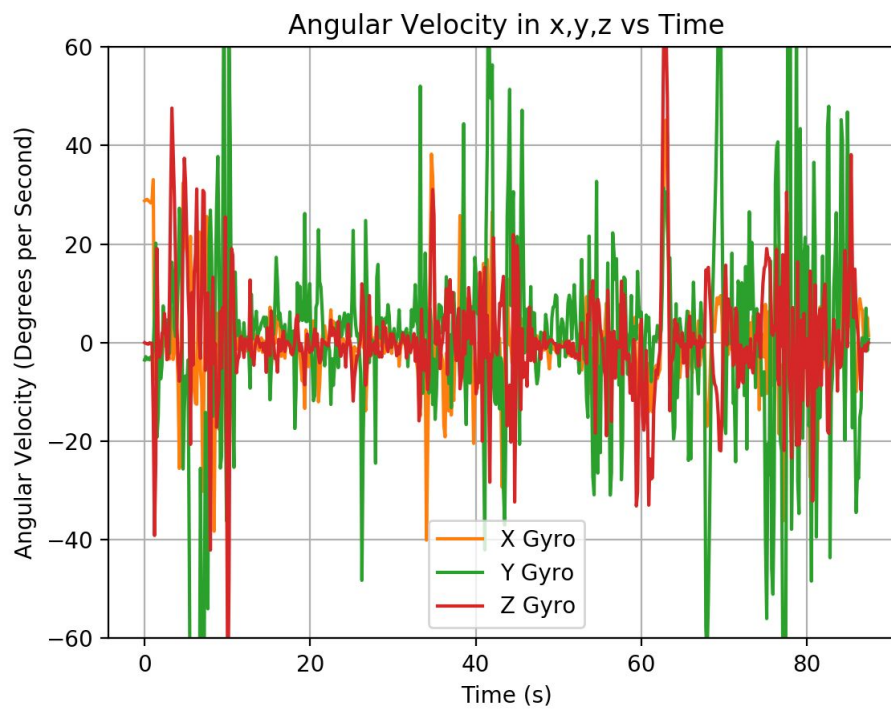

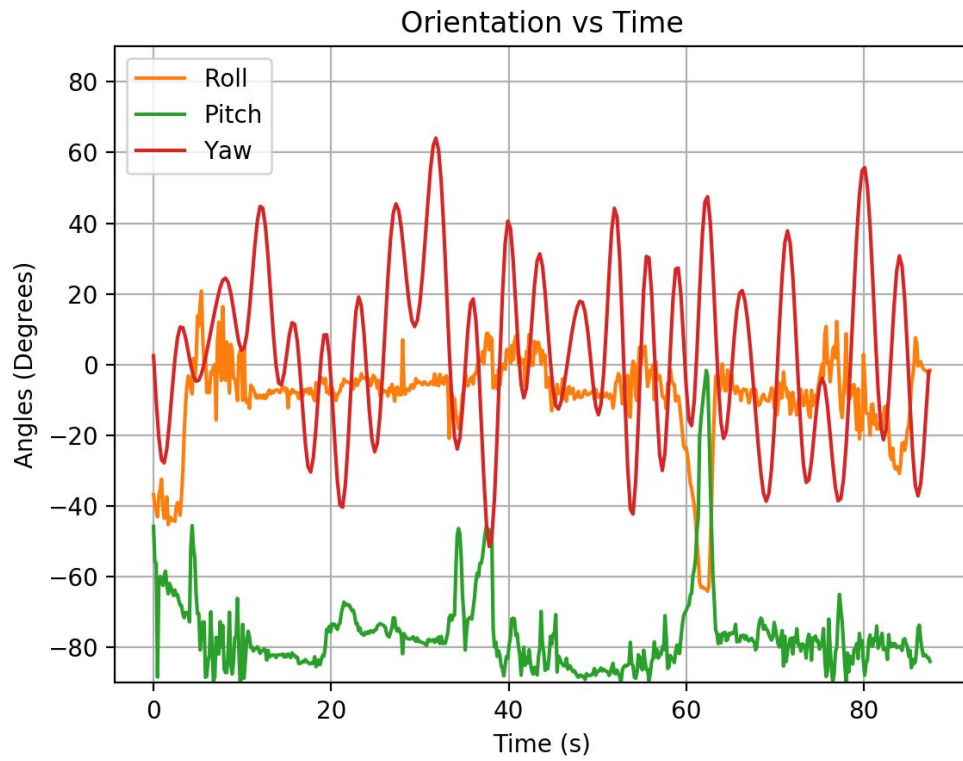Figure 35: Angular Velocity of Shoulder While Picking up Soft Body

Figure 36: Orientation of Shoulder While Picking up Soft Body
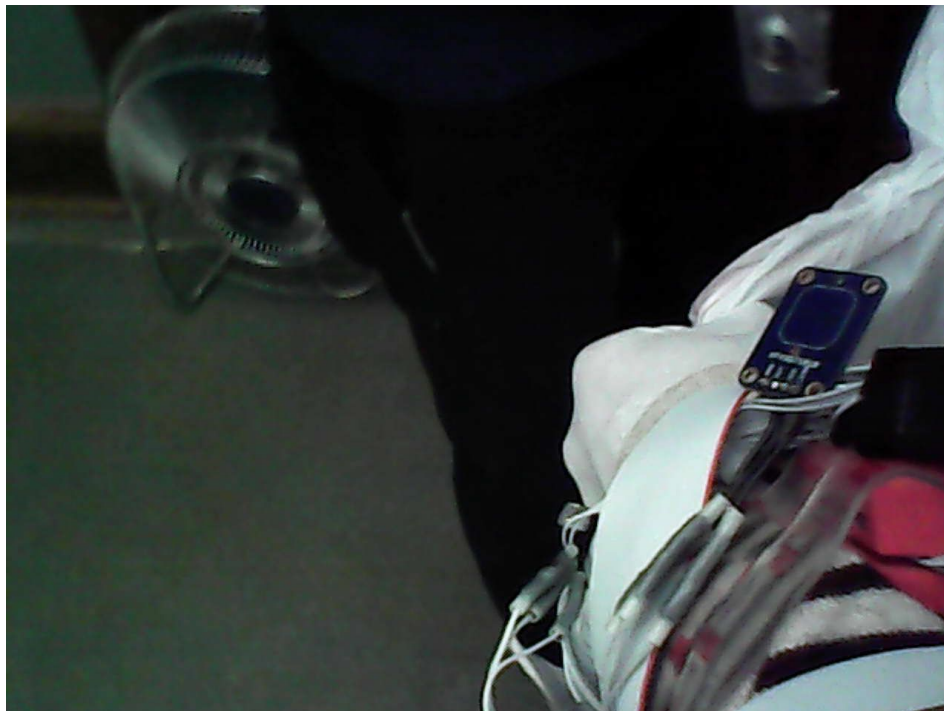
Figure 37: Spy Camera While Picking up Soft Body

In the case of picking up soft bodies we see the greatest load was in the pinky finger represented by FSR4 as shown in Figure 33. The total force, as shown in Figure 32, was also very similar to the force on FSR4 supporting the observation that the pinky was supporting the majority of the load. This depends on how the bag was gripped and, in the test case, the little finger took the majority of the weight of the objects placed in the bag. The loads on every finger were consistent throughout the test, but the magnitude of force varied from finger to finger significantly. The index finger, FSR1 did the least to support the soft body. During the impulse, when the weights were dropped inside the bag, the accelerometer register an acceleration of close to 10 m/s^2 as shown in Figure 34. The impulse also resulted in angular velocities in the region of 150 degrees per second around the z axis as shown in Figure 35. Orientation again showed the greatest variation in yaw as can be seen in Figure 36. The spy camera captured an image of the user supporting the soft body with all 4 fingers, Figure 37, which corresponds to lower individual force on fingers but a larger total force as seen in Figure 34 and Figure 33.

Case 5 - Tapping
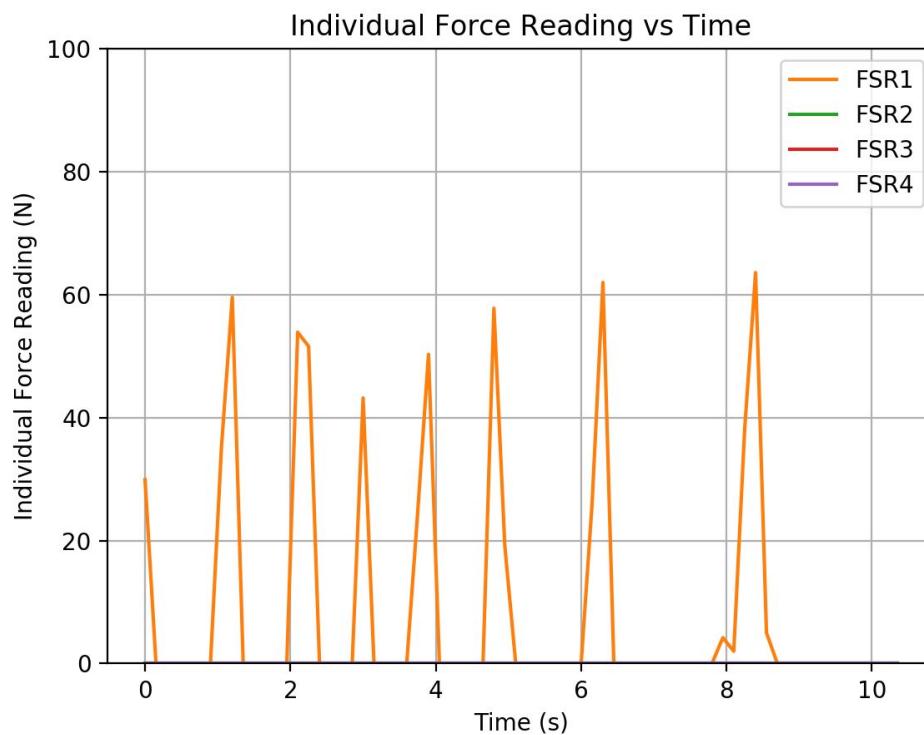
Figure 38: Individual Finger's Force Readings While Tapping

The case of tapping was considered to test the response time of the prosthetic and to see how quickly it could report the changes of resistivity in the force sensors. We conducted this case exclusively using the index finger, or FSR1, and thus why only that recording is showing in figure 38. The FSR readings were much lower than the number of taps as seen in Figure 38. There were only 20 registered taps for a test case where the user tapped 60 times, this is a case undersampling and will be explained in

analysis section. There were no significant accelerations or angular velocities in the x, y or z direction for tapping.

Capacitive Touch

Our initial decision to include capacitive touch was to provide the user a real-time visual response to alert them of their proximity to an object. This would be the first response for approaching an object, as it would light up before the vibration would be activated. The capacitive sensors are linked to the LED's which sit on the forearm of the users, which are in the direct line of sight.

While using the capacitive touch sensor, we noticed that the LED would often trigger before the sensor made contact with any objects. Because of the nature of capacitance, we hypothesised that the proximity in which the LED was triggered might depend on the type of material in which it is approaching.

To test this, we used one insulating surface and one conducting surface, namely a laminate desk and a aluminum laptop. We measured the distance between the surfaces and the sensor the moment the LED lit up. As a result, the sensor approached 5 mm from the conducting surface and 1 mm from the insulator before the LED went off.

There were a few stipulations with our prototype that restricted our data acquisition. We found that the presence of a capacitive sensor on the bottom of the FSR was a huge impairment to the quality of data. The presence of a chip on the base of a finger severely restricted the user from effectively mimicking everyday tasks such as typing. Although the capacitive touch sensors do serve a important role, they should not be an interference in the users ability to perform tasks. Additionally, the contraption housing the vibrating motors was too small. This forced the vibrators to be in close proximity inhibiting the user from accurately differentiating which finger was being triggered, and oftentimes going unnoticed entirely. Ideally, the user would have distinct locations on their arm that corresponds to the specific vibration sensors and an alternative, less cumbersome capacitive sensor.

The camera did not provide quality images. The point of the camera was to correlate our data with what was physically being done by the wearer. Unfortunately, we could not keep the camera in a position such that it always captured the actions or data measuring. For the most part, the hand and sensor package was out of frame.

Analysis

The projects primary scope was to study methods to provide feedback to prosthetic users. The ability to quickly read the force on the fingertips and relay that to the haptic feedback drivers and finally the user via vibrating motors was the primary focus. In that respect, our prototype functioned effectively. Our test users were able to discern the haptic feedback from the right two fingers from the left two fingers. However, the user was unable to differentiate between the the two central fingers. The user was also effectively able to distinguish between the difference in the forces at the fingertips via the different intensities and frequencies of vibration of the haptic feedback driver. The overload function built into the prototype functioned as well, triggering a clicking noise via the speaker when the prosthetic was overloaded. The haptic feedback was almost instantaneous with the user being unable to determine any lag between the touch and feedback. The real time functioning of prototype package satisfied the requirements of an inexpensive sensor package that could be retrofitted onto existing prosthetics.

Our data illustrated a few key points regarding the functionality of our sensor package and features required by a prosthetic to mimic the operations of a human limb.

Firstly, the accelerations recorded in the system peaked at 10 m/s^2. This shows that under impulse loading the accelerations felt on the prosthetic are similar to that of freefall. Additionally, the maximum total force observed in our loading conditions was approximately 51.7lbs or 230N. For the rigid body case, this is only the force on the fingertips, which does not account for force on finger joints or palm because the natural way to grip an object is with the entire hand and not just the fingertips. For the soft body case however, the the majority of the weight was directly on the fingertips. This force is significant when added to the weight of the prosthetic itself because of how much stress and torque is experienced by actuators, pivot points, and connection points. Therefore, a prosthetic would need to be durable enough to resist forces and torques accordingly. Based on the average weight and size of a prosthetic hand (about 1 lbs, 6 in)[18], this equates to approximately 235.8N and 36Nm.

Additionally, we were able to gather data for amputations below the shoulder using the gyroscope sensor. Orientation of our box was calibrated and referenced as shown in Figure 39 and Figure 40. Where Figure 39 was the reference and Figure 40 was the orientation of our box with respect to to the reference orientation. We can see from the orientation plots that the yaw orientation has the most drastic changes over time. Therefore, prosthetics connected at the shoulder will experience the most wear in arm yaw rather than roll or pitch and thus need to be built accordingly.
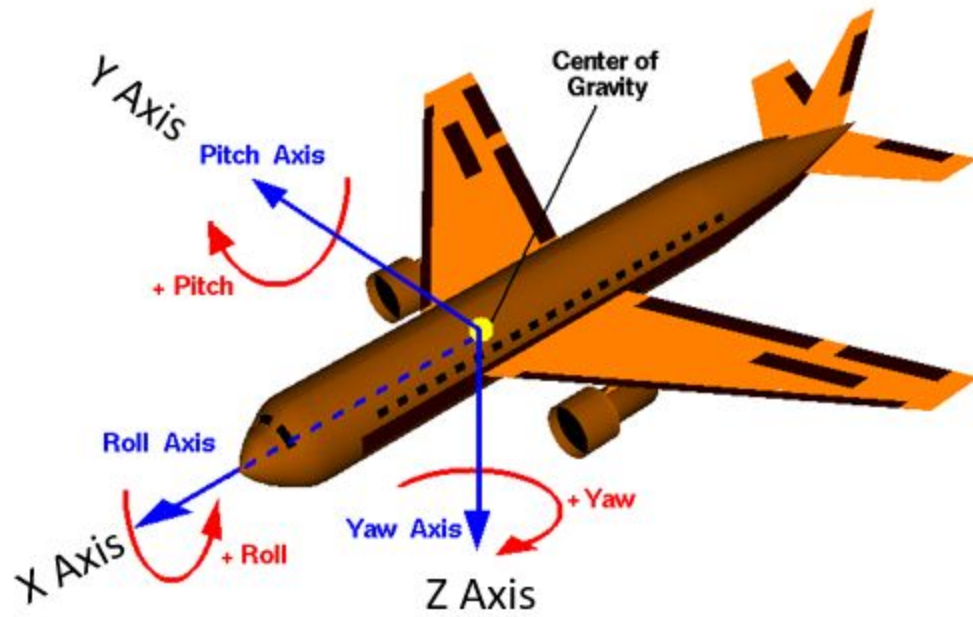
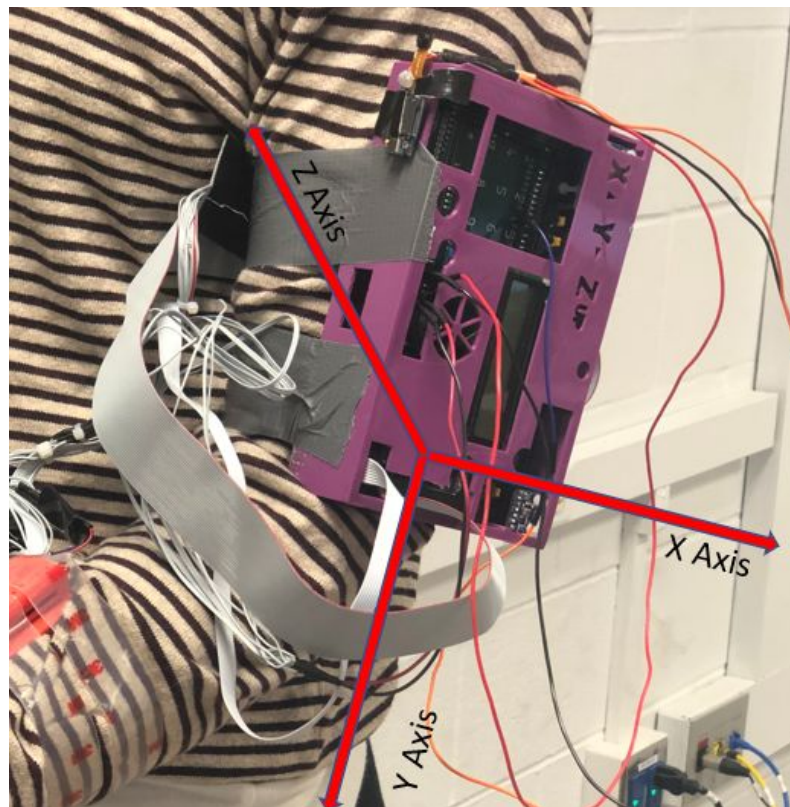Figure 39: Angular orientation in Terms of Roll, Pitch and Yaw

Figure 40: Orientation of Sensor Package in Cartesian Basis

The orientation data extrapolated from the accelerometer, using formula (i),(ii) and (iii), showed that most the user moved their shoulder was -60° to +60° in yaw and -20° to +20° in roll and pitch. This data, along with our angular velocity data, can be used to implement and select motors and actuators required in prosthetic manufacturing. This would in turn help drive the cost down by minimizing over-engineering these joints. Current advanced prosthetics provide large angular freedom thereby giving users unnecessary functions and charging a premium for features that may or may not ever be used. The actual extent of angular freedom required in most cases ranged from -60° to +60° as shown by the range of yaw values from our tests.

$$\tan(\theta) = \frac{a_y}{a_z} \quad ...(i)$$

$$\tan(\phi) = -\frac{a_x}{\sqrt{(a_y^2 + a_z^2)}} \quad ...(ii)$$

$$\psi = \tan^{-1}\left(\frac{m_y}{m_x}\right) \quad ...(iii)$$

Where a_x is x-axis accelerometer reading
a_y is y-axis accelerometer reading
a_z is z-axis accelerometer reading
m_x is x-axis magnetometer reading
m_y is y-axis magnetometer reading
θ is angle with respect to x-axis
ϕ is angle with respect to y-axis
ψ is angle with respect to z-axis

The component of *g* from all acceleration readings was removed by applying the following rotation matrix and assuming that G = {0,0,-g}:

$$R(\alpha,\beta,\gamma) = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma) =$$
$$\begin{pmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{pmatrix}.$$

Where α = roll angle
β = pitch angle
γ = yaw angle

Thus, the transformed gravity vector is:
$$G'=G*R(\alpha,\beta,\gamma)=\langle -g\sin(\beta), g\cos(\beta)\sin(\gamma), g\cos(\beta)\cos(\gamma)\rangle...(iv)$$

And the corrected acceleration vector is given by the following equation, where A is the original acceleration vector:

$$\vec{A}'=\vec{A}-\langle -g\sin(\beta),g\cos(\beta)\sin(\gamma),g\cos(\beta)\cos(\gamma)\rangle...(v)$$

This orientation data is also useful for below the shoulder prosthetics as it gives the exact dynamics and kinematics of the residual limb. This allows for precise modeling of an average users requirements. In many cases, especially typing, we noticed a lot of yaw and fast x and y axis gyroscope readings suggesting that effective prosthetics must have a fast actuation in these directions to help amputees perform everyday tasks. Another important aspect to consider is the stability of the prosthetic on the users residual limb; the current fastening mechanisms of prosthetics rely on a custom cup and straps. Many prosthetic users regularly complain of bad fitment and prosthetics coming off. The orientation data confirmed this because of the wide range of motion and accelerations experienced by the deltoid region.

Next, the tapping test proved that the sampling rate of our force sensors is too slow. Counting 60 taps resulted in only 20 peaks which means that the data has to be sampled at least 3 times faster in order to register all the taps. This is due to two factors. When tapping consistently at a fast pace, it's difficult to apply enough force to the FSR to trigger any feedback. Luckily, our capacitive touch sensors made up for the force inconsistencies. That being said, there was not a significant change in number of peaks when more force was applied to each tap. Another factor contributing to the sampling rate was the duration of contact between the FSR and the surface. By doubling the time duration of each tap, and consequently halving the speed of the taps, we were able to increase the amount of taps recorded from 33.3% to 62%. Unfortunately, this indicates that the FSRs we are using cannot interpret a change in resistivity at speeds similar to typing.

Suggestions

Consideration 1

Many studies have been done with respect to haptic feedback and its uses. One study in particular, done by the University of Warwick[16], assessed subjective responses to haptic feedback in automotive touchscreens. Their experiment consisted of a simulated driving scenario while interacting with a touch screen interface. The interface had multiple configurations for feedback including haptic, visual, and audio, or any combination of the three. The study's results indicated a preference for multi-modal feedback rather than only visual. The highest response was to have all three feedback types which had twice as many ratings as only two types, visual and audio were the most prefered.

In our project, we make use of both haptic and visual feedback with drivers and LEDs mounted on the forearm to relay force on a user's fingertips and proximity, respectively. Additionally, we also use an audio signal when the sum of the forces on each finger exceeds the total FSR capacity. The combination of the feedback systems worked well for our users but could use some improvements. The capacitive touch, for example, was often too sensitive and triggered the LED before making contact, especially when approaching a conductive surface. The breakout board used for capacitive touch in our project was larger than the tip of the user's finger and hence using it while doing activities such as typing was not possible. The capacitive touch sensors were discarded in favor of getting better data from the FSRs. Possibly looking into capacitive touch sensors that are soft and flexible rather than the ones our team used is an avenue that can be explored by future studies of this nature.

Consideration 2

Another study, done by the Institute of Electrical and Electronics Engineers[17], or IEEE, was extremely similar to our project except that they were experimenting with lower-limb prosthesis and forces thereon. In their experiment, they used force sensitive resistors on 4 primary contact points of the foot. These sensors sent data back to a system controller which in turn drove four corresponding pneumatically controlled balloon actuators. The balloon actuators, enclosed in a band, surround the middle of the thigh. The results were as follows:

> "Six normal subjects wearing the actuator cuff were able to differentiate inflation patterns, directional stimuli and discriminate between three force levels with 99.0%, 94.8%, and 94.4% accuracy, respectively."

Although our project was more focused on relaying a sense of touch rather than pure force, it is interesting to see how well the subjects were able to distinguish differing force levels. Our feedback system worked fairly well when notifying the user of varying intensities of forces on the fingers as a whole but lacked success when differentiating the force on each individual finger. Our user was able to differentiate the pinky finger feedback from the index finger feedback but struggle to differentiate between the feedbacks of the middle and ring finger. This is due to a small separation of the vibrators on the users forearm. If we were able to use balloon actuators as a form of haptic feedback, we may be able to solve the issue of differentiating touch from individual fingers. Additionally, it would be more effective to represent force on an individual finger. Instead of translating the sensation of force with vibrations, we could simply relocate the force from the finger to the arm. It would also be beneficial to surround the forearm with haptic drivers and strategically place them instead of placing them all in the same relative location as it would help distinguish which finger is making contact. For example, placing one haptic driver on each nerve tract (Figure 41) would greatly improve differentiation of individual fingers, but it would limit feedback to only 3 fingers.
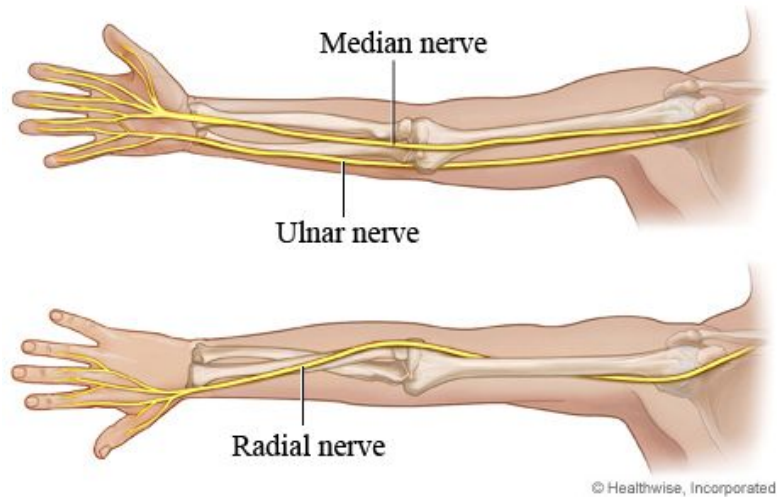
Figure 41: Nerves in Arm

Conclusion

In conclusion, we were successful in creating a sensor package that is affordable and able to monitor usage and load bearing limits for a prosthetic hand. Considering an average cost of a prosthetic hand of about $20,000, our sensor package, totaling only $335.57, would be relatively cheap to implement. However, further study can be done to improve feedback as a translation for touch sensitivity. We were able to measure forces and indicate contact for individual finger, but only able to give general feedback when touching or squeezing objects. To better this package, we could attach an external capacitor to the sensor that would effectively fit seamlessly under the FSR. Furthermore, we could spread out and strategically place each haptic feedback driver. With these slight modifications we could greatly improve the amount of measurable data, the quality of the data, and the perception of touch sensitive feedback. To continue these studies, we would want to research and experiment with actual implementation of our sensor package onto an existing prosthetic.

References

1. Maurice LeBlanc, MSME, CP "Give Hope - Give a Hand" - The LN-4 Prosthetic Hand
   https://web.stanford.edu/class/engr110/2011/LeBlanc-03a.pdf
2. The Most Common 3D Printed Prosthetics,
   https://all3dp.com/2/the-most-common-3d-printed-prosthetics/
3. Hero Arm, https://openbionics.com/hero-arm/
4. LSM9DS1 Accelerometer + Gyro + Magnetometer 9-DOF Breakout,
   https://learn.adafruit.com/adafruit-lsm9ds1-accelerometer-plus-gyro-plus-magnetometer-9-dof-breakout/overview
5. Force Sensitive Resistor, https://learn.adafruit.com/force-sensitive-resistor-fsr
6. Capacitive Touch Sensors, https://learn.adafruit.com/adafruit-capacitive-touch-sensor-breakouts
7. Adafruit BME680 - Temperature, Humidity, Pressure and Gas Sensor,
   https://www.adafruit.com/product/3660
8. DS3231 Precision RTC Breakout, https://www.adafruit.com/product/3013
9. Adafruit DRV2605L Haptic Controller Breakout, https://www.adafruit.com/product/2305
10. Vibrating Mini Motor Disc, https://www.adafruit.com/product/1201
11. LCD, https://www.adafruit.com/product/2130
12. MCP4725 12-Bit DACl, https://www.adafruit.com/product/935
13. PAM8302A: Adafruit Mono 2.5W Class D Audio Amplifier,
    https://www.adafruit.com/product/2130
14. TCA9548A I2C Multiplexer, https://www.adafruit.com/product/2717
15. Micro SD Card Breakout Board, https://www.adafruit.com/product/254
16. Pitts, Matther J, et al. "Assessing Subjective Response to Haptic Feedback in Automotive Touchscreens ." ACM Press, AUTOMOTIVEUI '09 Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications, September 21-22, 2009, University of Duisburg-Essen, Germany, 2009, pp. 11–18, dl.acm.org/citation.cfm?id=1620512.
17. Fan, R.e., et al. "A Haptic Feedback System for Lower-Limb Prostheses." IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 16, no. 3, 2008, pp. 270–277., doi:10.1109/tnsre.2008.920075.
18. 2016 Project Elijah Final Report Pdf,
    https://web.wpi.edu/Pubs/E-project/Available/E-project-042612-145912/unrestricted/MQP_PaulV_Complete_Final_3.pdf
19. Congenital Limb Deficiencies and Acquired Amputations in Childhood,
    https://www.amputee-coalition.org/resources/amputations-in-childhood/
20. How Much Does a Prosthetic Arm Cost?, https://health.costhelper.com/prosthetic-arms.html
21. INA219 High Side DC Current Sensor Breakout - 26V ±3.2A Max,
    https://www.adafruit.com/product/904

Appendix

Detailed Hardware List

1. **Arduino Mega** - Uses a 5V operating voltage and a 16Mhz clock.

2. **LSM9DS1 Accelerometer + Gyro + Magnetometer 9-DOF Breakout** - This sensor communicates with the Arduino using the I2C protocol. It has ±2/±4/±8/±16 g ranges.

3. **Force Sensitive Resistor (FSR)** - Has a Devise Rise Time of 2msec. Force sensitivity range is from 10kg to 100g. Resistance ranges from 100k Ohm to 200 Ohm. Uses less than 1 mA of current. The active area of the sensor region is 12.7 mm and a nominal thickness of .55mm.

4. **Capacitive Touch Sensors** - Dimensions are 20mm x 29.35mm. This sensor weighs 2.03g. The delay between bursts in Fast mode is approximately 1 ms.

5. **Adafruit BME680** - This sensor communicates with the Arduino over the I2C protocol. It measures humidity with ±3% accuracy, barometric pressure with ±1 hPa absolute accuracy, and temperature with ±1.0°C accuracy.

6. **DS3231 Precision RTC Breakout** - It communicates with the Arduino using the I2C protocol. Has an accuracy of ±3.5ppm from -40°C to +85°C and a Digital Temp Sensor Output accuracy of ±3°C. It also features a backup-battery for continuous timekeeping.

7. **Mini Spy Camera** - It takes photos at a resolution of 1280x720 in JPEG format and video at 640x480 in AVI format. It has a standby current of 80 mA and and operating current of 110 mA. Camera dimensions 6.2mm x 6.2mm x 4.4mm. The PCB dimensions are 28.5mm x 17mm x 4.2mm.

8. **Adafruit DRV2605L Haptic Controller Breakout** - This sensor communicates with the Arduino over the I2C protocol. It has a rise time of 35ms and a brake time of 10ms

9. **Vibrating Mini Motor Disc** - This motor is 10mm in diameter and 2.7mm thick. With a 5V current it draws 100mA and with a 4V current it draws 80mA. It can vibrate up to 11,000 RPM at 5V.

10. **LCD** - 16 characters wide with 2 rows for text. It uses and electroluminescent backlight. Utilizes 64 x 8-bit character generator RAM and 90 x 8-bit display RAM. It corresponds to high speed

MPU bus interface at 2 MHz.

11. **MCP4725 12-Bit DAC** - This communicates using the I2C protocol at standard (100 kbps), fast (400 kbps), and high-speed (3.4 Mbps) modes.. Contains on-board non-volatile memory (EEPROM). It has a settling time of 6 μs.

12. **PAM8302A Adafruit Mono 2.5W Class D Audio Amplifier** - This breakout board has an output power of 2.5W at 4Ω. It features 50dB PSRR at 1KHz, a fixed 24dB gain, onboard trim potentiometer for adjusting input volume and thermal and short-circuit/over-current protection.

13. **TCA9548A I2C Multiplexer** - Allows up to 8 devices on the I2C bus, in any combination. Uses up to 400 kHz clock frequency.

14. **Micro SD Card Breakout Board** - Communicates over digital pins and has 256Kbytes of flash storage and 4Kbytes of EEPROM storage. Formats is FAT16 and FAT32.

15. **INA219 High Side DC Current Sensor Breakout** - It has 0.1 ohm 1% 2W current sense resistor. Up to +26V target voltage, up to ±3.2A current measurement, with ±0.8mA resolution and a size of 0.9" x 0.8".