Variable Impedance Actuators: Robotic Drumming Applications

A Major Qualifying Project
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfilment of the requirements for the
Degree of Bachelor of Science

By
Matthew Bisson
Nikhil Chintada
Jordan Grotz
Pheobe Yeung

Date: April 6th, 2021

Report Submitted to:
PROFESSOR SCOTT BARTON, ADVISOR
PROFESSOR MAQSOOD ALI MUGHAL, ADVISOR
PROFESSOR KENNETH STAFFORD, ADVISOR

maxon

WPI

Abstract

The goal of this project was to create a robotic drumming system which could play a variety of percussive instruments, produce basic and complex strikes, and demonstrate aural and visual expressivity. To accomplish this a robotic arm was designed to approximate a human arm, with a variable impedance actuator (VIA) utilized at the wrist. The VIA can be tuned to simulate the dynamics of a drum roll. Due to circumstances resulting from the COVID-19 pandemic, the team produced simulations to verify the design of the robot. Despite the arm being unable to fully reproduce the range of human playing ability, this simulation demonstrated the capability of the robotic arm to replicate, with moderate fidelity, the aural and visual characteristics of human players.

Acknowledgements

We would like to thank our project advisors Professor Barton, Professor Stafford and Professor Mughal, who have provided valuable insight, support and feedback throughout this project. We would like to thank Maxon Motors for their generous support of the project, and the Maxon EC90 Flat Motor #500266, along with the GP42C Gearbox #223089 and the MILE encoder #621795. In particular, we would like to thank Jason Vaccaro for his support and help in securing these parts and the sponsorship from Maxon. Finally, we would like to thank the following departments for their help and resources: Robotics Engineering, Mechanical Engineering and Electrical and Computer Engineering.

Table of Contents

List of Figures

List of Tables

Authorship

1.0 - Introduction

Musical robotics is a rapidly expanding field fueled by constant development in the capabilities of robotic systems. Since the 1970's, musical robots have been used as assistive devices or autonomous players (Kapur 2005). In such applications, the capabilities of musical robots are frequently compared to human players. Many assistive devices, autonomous players and even robots attempting to surpass human abilities still rely on human performance as one of the frameworks of quality and success. Research still continues on replicating and recreating the nuances of human performers in robots. In many simple percussive robotic instruments, strikes from an actuated drumstick do not capture the nuances of human playing. The motions of a human arm are intricate, with subtle movements of the wrist that allow for more complex techniques such as rolls. As such, actuators that can match a human player's capabilities for percussive output in dynamic range, temporal accuracy, dynamic accuracy and strikes per second are crucial to replicate the full range of human playing ability.

Though musical robots have primarily been designed with traditional actuators such as solenoids, pneumatics, and DC motors, in recent years researchers have begun exploring alternative actuators (Rooyen, Schloss, & Tzanetakis, 2017). Some researchers have pursued alternative actuators which can replicate attributes of human muscles such as variable stiffness (Kim, Garabini, Jaeheung Park, & Bicchi, Sep 2014). The field of prosthetics provides a number of possibilities towards this goal. One advancement in the last decade is the Variable Impedance Actuator (VIA). At this time, much of the research into VIA's have been focused on prosthetics and legged robots (Geeroms et al., 2018; Boaventura et al., 2012). VIAs seek to replicate the human body's ability to vary muscle stiffness in accordance to different tasks (Vanderbought et al., 2013).

Through research, we believe that VIAs could be implemented with great success in the field of music robotics. As this field has progressed in recent years, one of the recurring challenges has been the replication of the ability of the human percussionist. An actuator that can replicate the idiosyncrasies of human capability could help advance both autonomous robotic musicians and assistive musical devices. This project could further the use of a new actuator in the field of musical robotics and help create a foundation for further development and use of VIAs for undergraduate work at Worcester Polytechnic Institute (WPI).

2.0 - Background

2.1 - Metrics for Human-like Drum Performance

To replicate the performance of a human percussionist, there are a variety of metrics that must be understood. These metrics correspond to different aspects of percussion playing and can be used to both replicate performance and compare performances to each other. These metrics also help differentiate the attributes of the human performer and the idiosyncrasies of the instrument itself.

There are a variety of percussive instruments that vary in construction and physical properties. A human player must change their technique accordingly to play the different types of instruments. When analyzing percussion performances, we focused on the input of the human performer rather than the drum itself. While seemingly a simple instrument to play, there are a variety of complex metrics that must be understood to analyze a human percussionist.

The metrics we identified fall into five broad categories:

1. Movements of an arm while drumming
2. Dynamic and timbral control
3. Temporal and rhythmic control
4. Inaccuracy
5. Anticipatory gestures

These metrics and features help generate expressive performances that are more likely to engage and entertain audiences (Broughton, Stevens, 2009). An understanding of these metrics is pivotal to defining the requirements for a system that replicates human-like drumming. In this discussion, the musical concept of tempo will be referred to as a temporal value of onset intervals.

2.1.1 - Movements of an Arm While Drumming

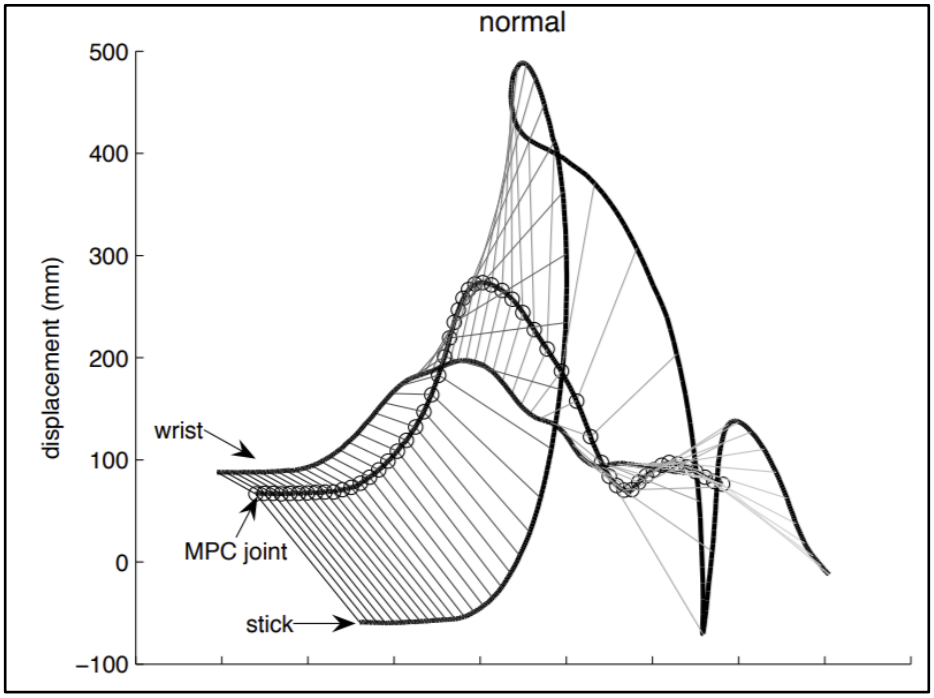


Figure 1: Trajectory of a wrist and drumstick at 25ms intervals (Dahl and Altenmuller, 2008)

The studies by Dahl, Altenmuller, Kawakami, use different analytical techniques to understand the complex control of a drumstick by a human percussionist while striking a drum (Dahl, 2011; Dahl and Altenmuller, 2008; Kawakami et al. 2008). In her 2008 and 2011 publications, Dahl provides motion analysis of human drum strikes. In each of these papers, she includes illustrations and listings of the trajectory and stick heights. This data illustrates the complex nature of the drum strike. The trajectory depends on several factors that skilled human percussionists are able to manipulate intuitively. First is the motion of the drumstick during a strike. As discussed by Dahl in her 2011 publication, an ideal drum strike impacts the drumhead then rebounds and swings back up. This motion path is detailed in Figure 1 above.

This image represents the three-dimensional displacement of the drumstick. Each circle and line connecting the paths represents 25ms intervals. Height is charted along the vertical axis. No quantity is represented on the horizontal axis. As each 25ms interval is shown on the line, the graphs are expanded in the horizontal direction for ease of visualization. This graph details the complex nature of a single strike.

When playing with short onset intervals, percussionists are able to utilize the rebound after the hit to prepare for the next. Another characteristic of the drum strike is the stick height. Dahl and Kawakami describe an interdependency between stick height, onset interval, and volume (Dahl and Altenmuller, 2008; Kawakami et al. 2008). In most cases, decreasing the onset interval will result in a decrease of stick height. Additionally, the stick height increases as the desired volume increases. Human percussionists are constantly adjusting these parameters as

they play. Kawakami and Dahl performed similar studies to support this conclusion; their research supports these conclusions and provides numerical and graphical data on human percussionists (Dahl and Altenmuller, 2008; Kawakami et al. 2008). In addition, these researchers confirm a correlation between stick height and volume. When instructed to play at a louder volume, the performer began their strike from a greater vertical distance from the drumhead. Figure 2 shows the change in stick velocity vs preparatory stick height. In this case, a higher stick velocity correlates to a higher perceived volume. The shapes of the markers represent different dynamic levels of volumes; the circles represent piano, the crosses are mezzo forte, and the squares are forte. These musical terms refer to quiet, moderately loud, and loud respectively. In contrast, as the performer decreased onset interval spacings in their performance, the maximum stick height decreased while maintaining a stable volume level. Performers created similar volumes from varying starting heights. This demonstrates the importance of movement for gesture and performance.

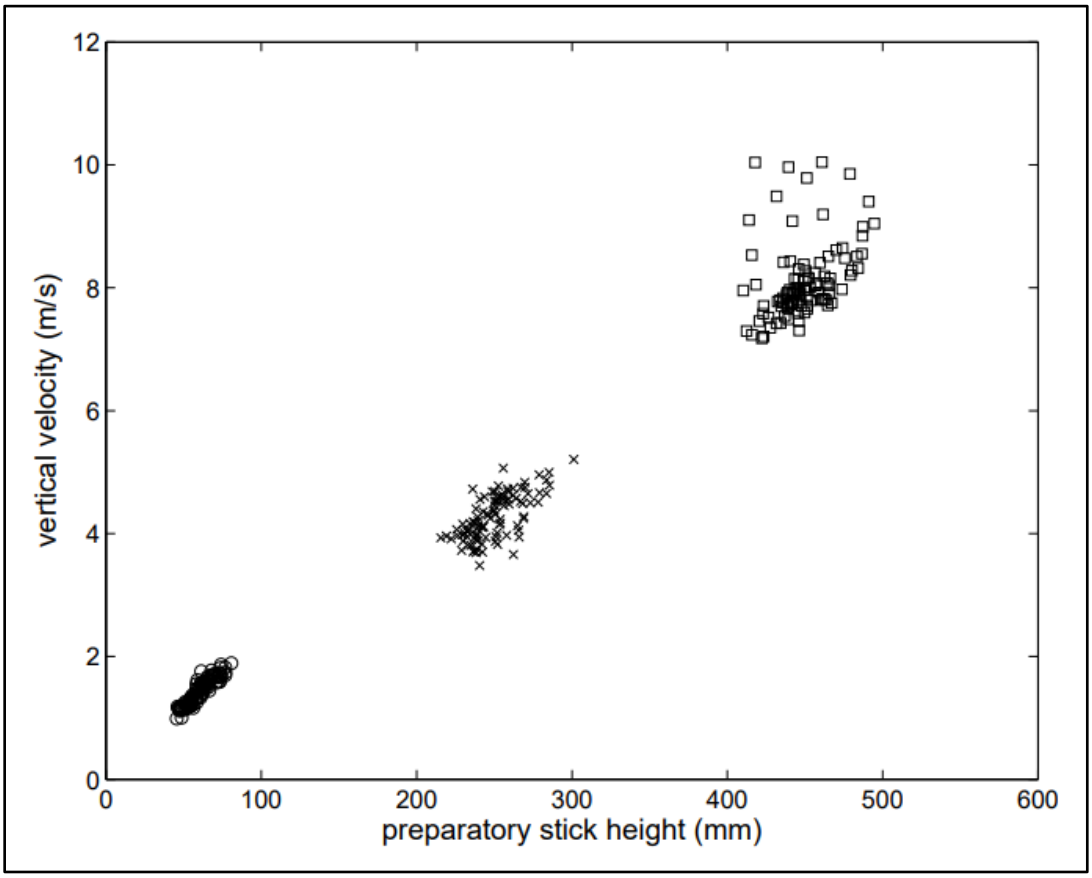


Figure 2: Vertical Velocity vs preparatory stick height (Dahl, 2011)

2.1.2 - Dynamic and Timbral Control

A percussionist intuitively controls the force of a drum hit to change the dynamic level. As the drumstick makes contact with the drum after the downswing of the stroke, the stick applies a certain force to the membrane of the drum. The most important characteristics of the

strike is the impulse. In this context, impulse is the change in momentum of the drumstick as it interacts with the drumhead. A study by Dahl in 2011 found a relationship between striking velocity, expected volume output, and incident forces on the drumstick. Human players were asked to play the full range of dynamic levels. The resulting drumstick tip velocities measured linearly increased from one to nine meters per second as the resulting volume increased (Dahl, 2011). When measured at the drum surface, the nominal applied force of a human player is between two and 100 newtons, and the approximate contact time has been shown to be between four to six milliseconds (Dahl, 2011, Wagner, 2006). As the drumstick makes contact, it depresses the membrane of the drum. Once the stick is removed, the membrane of the drum oscillates for a short period of time causing the air fluctuations creating sound waves.

The force applied to the drumhead is a result of the motions of a human arm controlling the drumstick interacting with the drumhead. The muscles in the upper arm and shoulder contract to cause a rotation through the elbow and wrist joint (Lamb, 2015).This distributes the forces throughout the arm and translates the force of the muscles in the upper and lower arm to the momentum of the stick.

In controlling the volume and timbre of the drum hit, human drummers often change the striking location. For a snare drum, strikes that are closest to the center of the drum will have the fullest and loudest sound, while strikes near the rim of the drumhead will be quiet and sharp (Tindale et al, 2004). The timbral changes are noticeable enough that an AI was trained to identify the region of the drum that was being played based on the produced audio (Tindale et al, 2004). In this way, musicians can have a fine degree of control over the sounds they are creating.

2.1.3 - Temporal and Rhythmic Control

Percussive performances can consist of many different rhythmic patterns at different tempos. As such, percussionists require a fine degree of temporal and rhythmic control. This control can be understood by analyzing the techniques of a performer. In applying these concepts of control to a robotic system, an understanding of these playing techniques is needed.

One complex type of drum hit seen in western music is the drum roll. In a drum roll, the drumstick makes multiple contacts with the drumhead with one downstroke from the musician. The human player dampens the oscillations of the drumstick so that each individual strike is consistent. A motion study by Robert Van Rooyen et al (2016) measured the motion of the drumstick tip for a double stroke roll. A camera was used to track the motion of the right and left drumstick tips. Due to the angle of the camera, the impacts of each stick appear at different heights. Figure 3 shows the graph of this motion.

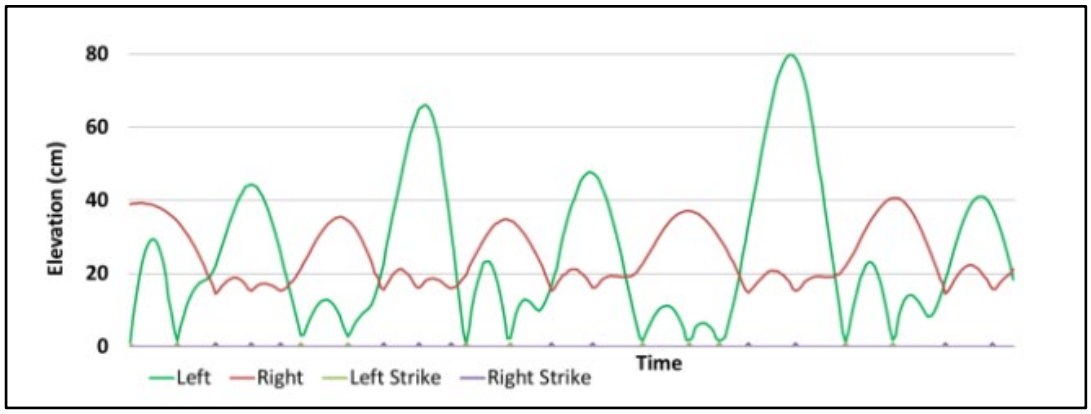


Figure 3: Motion capture of a double stroke roll. (Rooyen et al, 2016)

There are two main types of drum rolls: the double stroke roll and the buzz roll. In a double stroke roll, each downstroke results in two individual hits. A buzz roll refers to a roll in which each downstroke results in many small hits, creating a buzzing effect. These rolls have a wide range of applications and can be found in many different areas of music. In addition to rolls, there are many other types of hits and patterns. Each requires fine control and different amounts of damping on the drumstick to play accurately. Two such hits are the flam and drag. Played with two hands, a flam is a strike that is preempted by another strike. The hits occur with a very small degree of separation to properly play the flam. A drag is similar to a flam. However, on the initial preemptive downstroke, the stick rebound is damped to hit the drumhead multiple times. These types of hits are similar in nature but have different sound characteristics and different uses.

The world's fastest drummer can play at a frequency of 10Hz (100ms intervals) per hand (Fujii et al. 2009). It was thought that the max single-handed playing frequency was between five and seven hertz before the motions of the wrist became unstable due to fatigue. However, in studying the movements of the world's fastest drummer, reciprocal motion, associated with wrist compliance, was found to be more prevalent than co-contraction (which is associated with stiffness). While rapid playing is desirable it is important to note the inaccuracies present in human rhythmic performance. These minute inaccuracies are actually desirable as they play an important role in expressivity for the audience.

2.1.4 - Inaccuracy

A "perfect" performance is unobtainable as musicians perform with natural rhythmic and dynamic fluctuations (Broughton, Stevens, 2009). Research on a Ghanaian drummer playing to the beat of a metronome for five minutes showed that the player would fall ahead or behind of the metronome by about 10 to 20 milliseconds (Beck 2012). Furthermore, this imperfection almost seemed to follow a pattern, as the drummer tended to play ahead of the beat in anticipation. The science behind these deviations is still being researched. There are different clocks in the brain running on different scales, but it is unclear which neuronal network runs the

millisecond regime clock (Hennig 2012). This research is important to audio engineering because music listeners dislike music that lacks natural human error. Electronic music attempts to add a humanized feature, where randomized error is inserted into the music. However, this error is not appealing to the human ear.

Long range correlation error is more widely accepted by music listeners. This implies that people prefer listening to error that is natural and follows a more distinct pattern. With long range error, the musician will be ahead of the beat for several hits before falling behind for another extended range of time. There is a balance of pattern and errors that is appealing to the human ear because it is the way that people naturally play music. Finding that balance and replicating this through software is still being researched.

2.1.5 - Anticipatory gestures

Visual aspects of musical performances can influence the experience of the audience. These traits of nonverbal communication are demonstrated by conductors. The main function of the conductor is to lead an ensemble and coordinate the players. A wide range of gestures and expressions can be used to convey the desired playing style. Sharp and precise movements by the conductor can communicate a desired staccato playing style. Broader and grandiose movement by the conductor can communicate a desired bombastic and sweeping playing style. Conductors that used these visual motions were more successful as their ensembles were easily able to replicate the desired play style (Mayne 1992). This visual effect translates to the audience, where listeners can anticipate upcoming music style by watching the conductor's motions.

There is substantial evidence that visual motions of the musicians also affect the perception of a musical performance. If music was strictly an auditory experience, there would be little benefit in attending concerts and performances (Vines 2004). Whether conscious or unconscious, musicians tend to make movements to correlate with the emotion they are conveying to their listeners. There is speculation that people with more music experience rely less on visual cues for portrayed expressions (Voigt, 2016). Experienced listeners are able to pick out expressive cues and recognize the significance of different features without the aid of visuals. However, aurally static music with a dynamic performance was perceived as more expressive than a static performance of a well-played dynamic piece (Voigt, 2016). This shows that musical emotions are generally more affected by anticipatory gestures more than auditory quality.

2.2 - Replicating Human-like Motion with Robots

Actuators are essential to the creation and replication of human-like behavior in a mechanical model. Researchers are developing actuators that can emulate the mechanical advantages and properties of muscle within the field of robotics and biomimicry. One of the major advantages muscle systems have over traditional actuators is their adaptability in both damping and stiffness. Recently, non-traditional compliant actuators that vary in damping or

stiffness have become more common. One such emergent actuator is the Variable Impedance Actuator (VIA).

2.2.1 - Variable Impedance Actuators

VIAs utilize an equilibrium position, allowing them to deviate from initial set positions based on the mechanical properties of stiffness and damping. This is unlike traditional actuators that move to and hold in a specific position. While still an emerging technology, VIAs contain a broad range of designs being categorized based on how their stiffness and damping are achieved.

The term variable impedance actuators applies to any actuator system with an element that allows for damping or compliance of the system. VIAs can be described using three categories (Vanderborght et al, 2013):

- Active Impedance Control Actuators
- Inherent Compliance Actuators
- Inherent Damping Actuators

Active impedance control actuators use software to simulate the damping and compliance of the system. As a result of software control, these actuators are easy to tune for their particular applications, and they allow for complex control when necessary (Vanderborght et al, 2013). One implementation of this type of actuator is the Backdrivable Electromagnetic Actuator for Robotics (BEAR) from the RoMeLa Lab at UCLA (Zhu, Hooks, Hong, 2019). This actuator consists of a brushless DC (BLDC) motor, low gear planetary drive, and field-oriented control motor driver. The BEAR actuator is liquid cooled to compensate for the high energy consumption of the device. Active impedance-controlled actuators have higher than average energy consumption due to the simulation of energy storing elements. If designed to act like a spring, the actuator would need energy input into the system to oscillate in the same manner as a spring does with its stored elastic potential energy.

Inherent compliance actuators contain an element that allows for the natural compliance of the system. This natural compliance allows for an infinite bandwidth response, and energy is stored and released in the element. One common implementation of this type of actuator is the series elastic actuator (SEA) (Gomes et al, 2016). This application consists of a motor connected to an end effector in series through springs. The compliance of the system is proportional to the stiffness of the series spring, and the motor will not be back driven when external force is applied to the end effector. However, the compliance is typically static, and the tuning of the system requires physical manipulation once the element has been chosen.

Similar to inherent compliance, inherent damping actuators allow for natural damping of the system. These types of actuators are often implemented through the manipulation of fluid dynamics, friction, eddy currents, or electrorheological fluids (Vanderbought et al, 2013). The

damping elements cause stability issues when trying to control the system, requiring more attention for the design and implementation of these systems. The series dampening actuator is directly analogous to the series elastic actuator and can be implemented in the same manner. This is one simple implementation of inherent damping. When designing VIAs, the damping and compliance designs can be incorporated together to achieve different desired responses of the system.

Due to VIAs' ability to control their impedance and thus their stiffness and damping, they are often found in applications in which a robot must interact with a dynamic environment. Advancements in the variable stiffness actuator (VSA) subclass of VIAs allows for safer interactions between humans and robots (Tonietti et al, 2005). Control over the stiffness of the actuator as it moves can be optimized for safety and efficiency requirements. Tonietti states that the purposeful variation of stiffness is similar to motions observed in humans. Other use cases where VIAs are advantageous are shock absorbing, stiffness variation with constant load or constant position, cyclic movements, and explosive movements (Wolf et al 2015). Wolf states that these motions are useful for a variety of applications, such as human robot interaction, active prosthetics, running robots, and safety in robotic arms. The requirements and designs of VIAs for each of these applications differ but the underlying principles are the same.

2.2.1.1 - VIAs in Safety Mechanisms

Safety is held paramount when designing a robot for human robot interaction (Schiavi et al, 2008). Schiavi and a group from the University of Pias dedicated their time to developing a new, more practically implementable VSA for robotic application. Their actuator was able to improve torque capabilities and preserve safety when presented with an impact. Genesis Robotics is also working on a completely electromechanical VIA for SCARA robots and other industrial applications. One of their marketing slogans is "New Level of Speed and Safety" (Genesis Robotics, 2017). The company has a video of the actuator not breaking an egg that is placed in the actuator's path of motion.

2.1.1.2- VIAs in Legged Robots

The compliance of VIAs is advantageous in the field of legged robotics. Walking robots are often presented with dynamic environments that are advantageous to respond to in the force/torque domain. In addition, legged robots should be able to complete both accurate static motions, and dynamic motions like jumping. A VIA is well suited for these applications (Boaventura et al, 2012). The HyQ platform, described in Boaventura's paper, is a hydraulic VIA based quad-legged robot designed with the intent of replicating highly dynamic motion. The control of the hydraulics allows for virtual compliance, as it resists large impacts without the use of passive elements (Boaventura et al, 2012). The level of control allowed by VIAs can clearly replicate more organic and dynamic motion.

Directly driven leg actuators exhibit many of the required needs for legged robots. The Minitaur robot from University of Pennsylvania was created to showcase the benefits of semi directly driven robotic actuators (Kenneally et al, 2016). Direct Driven actuators allow for torque transparency, and the lack of a spring or damper removes the filter response due to their dynamics. Both of these allow for more accurate sensing of the force/torque readings on each joint. In addition, impulses can harm systems with large gear boxes, and there is greater mechanical efficiency because there is no loss due to the efficiency of the gears. However, this forces motors to operate in non-ideal conditions, causing heating (Kenneally et al, 2016). The fundamental designs of the Minitaur were improved upon in the Mini Cheetah from MIT, which continues to use proprioceptive actuators with torque control. The main improvements were weight reduction and better control (Katz et al 2019).

2.2.1.3 - VIAs in Prosthetic devices

The benefits of VIAs in legged robots creates a natural analog for their use in prosthetics. Everarts et al (2012) focuses on replacing passive elements in prosthetics to more accurately recreate the motion of the ankle while walking. In analyzing the power during the stance and swing sections of a gait, the ankle joint has a positive network, meaning that energy is added into the system. Passive elements cannot properly replicate this. In addition, Everarts speaks to how a Passive series elastic actuator can only be tuned to one application, while a VSAs can adapt to any user's gait with an appropriate controller. Geeroms et a.l (2018) approach the prosthetics design using mechanically adjustable compliance and controllable equilibrium position actuator (MACCEPA) in an ankle to test its capabilities. The focus of their research was to determine other VSA implementations that can reduce the energy consumption of the system. The MACCEPA's beneficial use in dynamic environments is shown in the design by Van Ham et al. (2007). The compliance and elasticity of the system can reduce power requirements; however the device is not energy efficient. Tuning the parameters of control leads to an efficiency increase of around 15%, however, their research shows that the energy consumption for dynamic control is still significant.

The systems that implement different types of VIAs showcase the actuator's effectiveness in a dynamic environment. This is similar to the dynamics needed for the variability of striking different percussion instruments during a performance. Research in the field of musical robotics shows the current methods implemented for reacting to these dynamic needs.

2.3 - Prior Art

There have been many developments in percussion based robotic systems within the field of musical robotic systems. One of the main research institutions pursuing work in percussion robotic players is GeorgiaTech, under the guidance of Professor Gil Weinberg. The first robot

developed by GeorgiaTech researchers in the area of percussion-based music is an autonomous marimba playing robot known as Shimon.

Figure 4: Picture of Shimon's new design with eight BLDC (Yang, Savery, Sankaranarayanan, Zahray, & Weinberg, 2020)

This robot was designed as a "robotic musician", with autonomously actuated two-segment arms each with a mallet mounted on them in order to play the marimba and reproduce human performance capabilities. The robot has been updated and improved since its development in 2011, with recent changes made to implement brushless direct current (BLDC) motors in an effort to increase Shimon's musical capabilities and fidelity to human players (Yang, Savery, Sankaranarayanan, Zahray, & Weinberg, 2020).

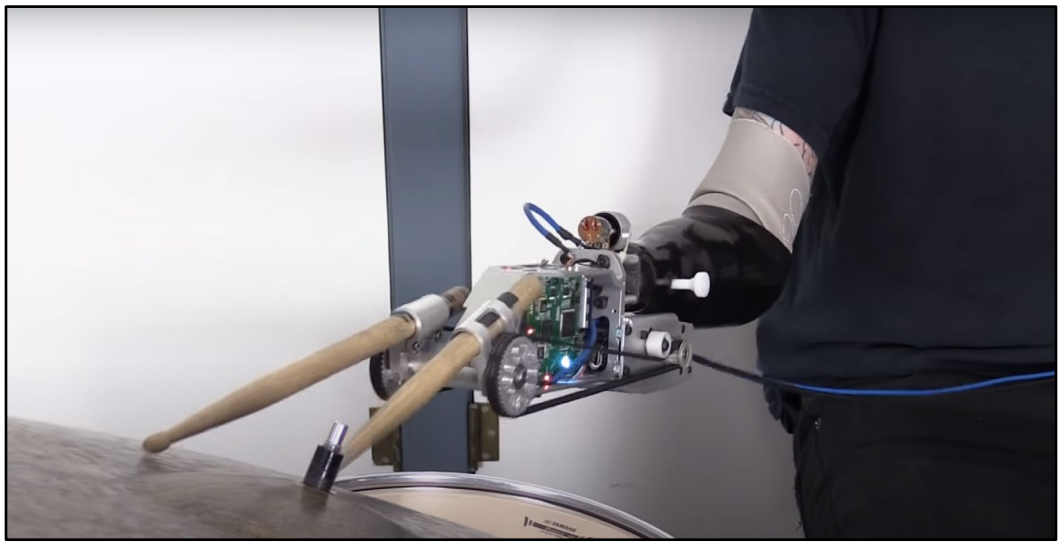


Figure 5: Drumming Prosthesis in use (Bretan, Gopinath, Mullins, & Weinberg, 2016)

Aside from Shimon, GeorgiaTech has also pursued work in the development of percussive assistive devices. The GeorgiaTech robotic drumming prosthesis was designed for an amputee player in order to enable him to play with both hands on a drum set. This device uses electromyography (EMG) in order to read signals sent from the user's muscles and use this input to create a physical response in the drumstick movement. The arm also has a second drumstick mounted on it. The purpose of this secondary drumstick is to enhance the playing ability of the user by adding an autonomous component which would play in response to the drumstick under the user's control. This device was mounted on the user's elbow, meaning it had only one degree of freedom (DOF), with a belt system used to control the drumsticks (Bretan, Gopinath, Mullins, & Weinberg, 2016). In addition to the robotic drumming prosthesis, GeorgiaTech researchers also developed the drumming third arm. This device is a shoulder mounted supernumerary robotic limb which is used as an additional arm for playing the drums. As this device is replicating an entire human player's arm it has four DOF. More recent work on the arm has been conducted in order to apply an input-shaping method in order to control vibrations through the arm for maximum user comfort (Khodambashi et al., 2018).

Figure 6: Drumming prototype used to test actuators (Rooyen, Schloss, & Tzanetakis, 2017)

Beyond the work at GeorgiaTech researchers such as Robert Van Rooyen from the University of Victoria have also worked on autonomous percussion robots. Van Rooyen and other researchers studied the use of voice coil actuators for percussion robots in place of standard motors such as solenoids. Van Rooyen's team found that voice coil actuators are a strong choice for use on musical robots as they have high acceleration, low latency, low hysteresis and precision positioning (Rooyen, Schloss, & Tzanetakis, 2017). A group of researchers across multiple institutions collaborated for the evaluation of solenoid-based designs, focusing on a robot called the MahaDeviBot designed by Ajay Kapur to play a variety of Indian folk music. The robot was designed with 12 arms in order to play a variety of instruments, with different arms designed for the different instruments. The solenoids had to be modified by the researchers

in order to reduce the sound they produce during actuation. These arms were tested for dynamic range and speed. It was found that when designing the arms with solenoids there was a tradeoff between speed and dynamic range. The faster it was designed to operate, the smaller its dynamic range was, with the arm with the most linear dynamic response also being the slowest (Kapur, Trimpin, Singer, & Suleman, 2007).

2.3.1 - Actuators Used in Percussive Robots

Within percussive robotic systems, solenoids are one of the most commonly used actuators (Long, Murphy, Kapur, & Carnegie, 2015). The rotary solenoid allows for a simple implementation of controlled mechanical rotation when voltage is applied to the coil. Similarly, linear solenoid actuators can be used to directly strike in a linear motion or attach to a mechanism which will translate their movement into rotary motion. Beyond solenoids, pneumatics and servos have also been used in robotic drumming applications. Due to the limitations of servos, they are much more common for small scale applications (Long, Murphy, Kapur, & Carnegie, 2015).

Solenoids are primarily used as they are simple to use, low cost, durability, versatility and produce very little noise during operation (Long, Murphy, Kapur, & Carnegie, 2015). However, rotary solenoids have a low stroke length, small dynamic range, and higher latency than many other actuators. Linear solenoids have a lower latency than rotary solenoids, however they also have a small dynamic range (Rooyen, Schloss, & Tzanetakis, 2017). In addition, linear solenoids can produce noticeable acoustic noise during operation (Long, Murphy, Kapur, & Carnegie, 2015). Pneumatics are the most powerful of the traditional set of actuators used for percussive robots. They are able to apply large forces, generating the highest dynamic output (Rooyen, Schloss, & Tzanetakis, 2017). However, they are expensive, and produce a lot of noise during operation due to the use of air compressors as well as their own linear actuation. Servos can be cheaper than solenoids and are easily obtained. They also have the ability to specifically set their angular position which can be advantageous for many applications. However, servos are sensitive to impact and strain making them less durable. They can also output noticeable noise during operation. Additionally, most servos have a limited range of motion and cannot move a full rotation, they typically also have a high latency (Long, Murphy, Kapur, & Carnegie, 2015).

2.3.1.2 - Alternative Actuators

Other alternatives as mentioned above include BLDC motors and voice coil actuators. BLDC motors using field-oriented control allow for smooth rotary operation and fast acceleration and deceleration. This also gives them a high dynamic range for musical applications. BLDC motors are more expensive in comparison to the previously mentioned actuators (Yang, Savery, Sankaranarayanan, Zahray, & Weinberg, 2020). Voice coil actuators offer numerous advantages, such as force control through voltage input to the coil, lack of backlash, and fast operation. These qualities allow for a large dynamic range. However, voice

coil actuators are expensive and have lower dynamic consistency than standard percussive actuators. (Rooyen, Schloss, & Tzanetakis, 2017).

One disadvantage all the prior actuators have is that they cannot mechanically replicate a human player's ability to perform a drum roll (Kim, Garabini, Jaeheung Park, & Bicchi, Sep 2014). Work by the researchers from the University of Victoria was done to create a PID feedback loop through PWM signals in order to create bounces on a voice coil actuator (Rooyen, Schloss, & Tzanetakis, 2017). A group of researchers in 2014 studied the use of a more recently developed actuator called a variable stiffness actuator. The researchers tested the actuator at different stiffness values in order to create a rolling stiffness at which the drumstick would bounce off the drumhead in order to create rolls.

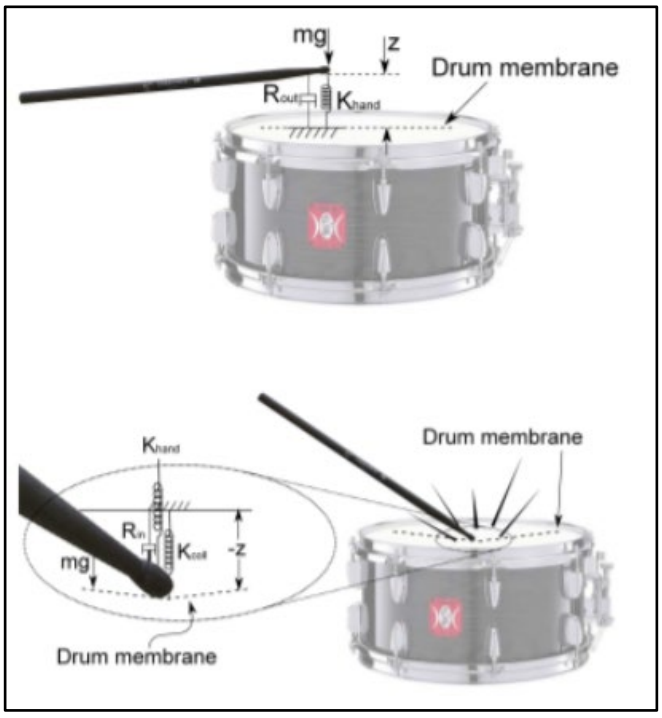


Figure 7: Diagrams demonstrating the mathematical model representing a drumstrike (Kim, Garabini, Jaeheung Park, & Bicchi, Sep 2014)

Using a mathematical model represented in Figure 7, Kim, Garabini, Jaeheung Park, & Bicchi represented the stiffness of the hand as K_{hand} which could be varied and multiplied with the deflection of the drum, z, to create the force needed for a rebound. They determined the appropriate stiffness for the actuator based on variables including the stick's height, length, damping value and the snare drum's properties. Their conclusion was that variable stiffness will allow percussion robots to create more complex motions such as rolls mechanically (Kim, Garabini, Jaeheung Park, & Bicchi, Sep 2014).

3.0 - Objectives and Design Requirements

Many current robotic percussion platforms do not accurately replicate both the complexities and nuance of human motion while preserving the ability to perform at similar levels as a human. In this project, we aim to build a robotic system that can mimic human-like drumming motions. It will accurately recreate the sound production and expressive motions of human performance. Our overarching goals are described in these three objectives:

1. Create a modular system that can play on a variety of percussive instruments.
2. Perform a selection of drumming techniques with one drumstick. The significant rudiments to replicate are a strike, double stroke roll, drag and flam.
3. Recreate an expressive performance both in terms of aural characteristics and visual motions.

To complete the stated objectives, we have outlined a series of design requirements that will allow us to ensure the success of our final design.

3.1 - Create a modular system to play on a variety of percussive instruments.

The robotic system will not be limited to playing on a single instrument, as the recreated motions should be able to be translated to playing across a variety of instruments. The system will perform on one instrument at a time. It should be easily moved and recalibrated to perform on a different instrument. To achieve this, we have designated requirements for end effect forces, strike range, and system feedback.

The robot should produce output velocities between 1-9 m/s for an impulse time of less than or equal to 6ms. This impulse time is the duration at which the drumstick is in contact with the drumhead. The resulting interactions may lie between two to 100 newtons, and the robotic system should be able to withstand these forces. These metrics were introduced in section 2.1.2. With these requirements, the robotic system should produce a wide dynamic range on any instrument while withstanding the incident forces. Finally, the dynamic accuracy will also be measured by playing the same note with required dynamic value (inputting the right force) and measuring standard deviation across each attempt to see dynamic accuracy.

The strike range of the arm will have a range of 18 cm in order to replicate human strike changes across a snare drum. This metric is described in section 2.1.2. The dynamics and timbre of a drum vary depending on where the drumhead is struck. While the number is designated for a snare drum, the range of motion will still allow for timbral changes on other percussion instruments.

The robotic system must have feedback to maintain proper performance. Given that the arm can play on a variety of instruments, a robust feedback mechanism will ensure that the arm has hit the percussive instrument even in the presence of unpredicted movements.

3.2 - Perform a selection of drumming rudiments with one drumstick.

The critical rudiments to replicate are a strike, double stroke roll, drag, and flam. A double stroke roll is a strike in which the stick rebounds and oscillates to create two hits. A drag is a double stroke roll with added third strike. A flam consists of two strokes in quick succession. The first stroke is a quieter grace note. Rudiments are sticking patterns that are typically done with two drumsticks. Attempting these rudiments with only one stick will require complex control. These simple rudiments provide a basis for the possibility of more involved and complex performances. These simple rudiments provide a basis for the possibility of more involved, and complex performances. A double stroke roll requires damping control of the stick's rebound, and a flam requires both quick changes in dynamics and movement. To ensure the success of performance, requirements for replication of damped oscillations, communication latency, and temporal accuracy have been defined.

The drumstick tip should follow the trajectory of a damped oscillation to properly play double stroke rolls at varying speeds. During a roll the motion of the drumstick is a damped oscillation due to the interaction of the energies of the strike, gravity, and the friction of the player's grip. (Rooyen et al, 2016; Kim et al, 2014). In order to replicate this motion with a robotic system, the stiffness of the actuated joint must be variable in order to create the correct stiffness for creating rolls on a particular percussive instrument.

A system communications latency less than 30ms (measured from when a signal is sent, to when the device starts to move) and an onset interval accuracy less than 50ms is required to ensure proper performance. A study on the effect of network latency of musical performances showed that performance quality was noticeably hindered with a latency of 86ms (Bartlette et al., 2006). In addition, the threshold of synchronicity for human perception was approximated to be 50ms (Rasch, 1988). Therefore, the maximum network latency should be 80ms so as to not hinder cooperative performance. Ideally the robot should be perceived as playing in time for a given synchronized pulse.

3.3 - Recreate a visually and aurally expressive performance.

Visual engagement coupled with aural characteristics are critical to the success of the system. The expressivity of a performance can be defined by the innately human idiosyncrasies of performance, but anticipatory gestures also help form a connection with the audience and help cue other musicians who the robot might be performing with. The trajectory of motion, temporal control and precision, and need for anticipatory gestures have been defined to accomplish this objective.

The robotic arm will be able to play at 100ms onset intervals (10 Hz) and have a standard deviation of less than 20ms for a set of hits. These metrics are based on the capabilities of the world's fastest drummer and the natural onset variance of human players (Fujii et al, 2009; Fujii et al, 2011; Hennig et al, 2012). A minimum onset interval of 100ms will allow our robot to play at any onset interval achievable by a human player. In regard to onset interval precision, if the robot is within the defined margin, the mechanical idiosyncrasies might naturally create the desired variance. A lower resulting standard deviation will allow for intentional modification of onset times to create naturally expressive performances. The timing requirements of the system have been outlined in Figure 8. This shows the relationship between communications latency and the onset interval accuracy and precision. The input signal will be preemptively sent to the arm to account for the movement time. This will limit the robot to playing predefined passages, as opposed to live composing in a digital audio workstation (DAW).

Lastly, the robot will be able to create anticipatory gestures that aid in cooperative playing with a human musician and engage audience members. Using the range of motion defined in the trajectory requirement, we will create pre-playing motion profiles. Through these design requirements, we believe the realization of our robotic arm will meet the objectives for the project goal.

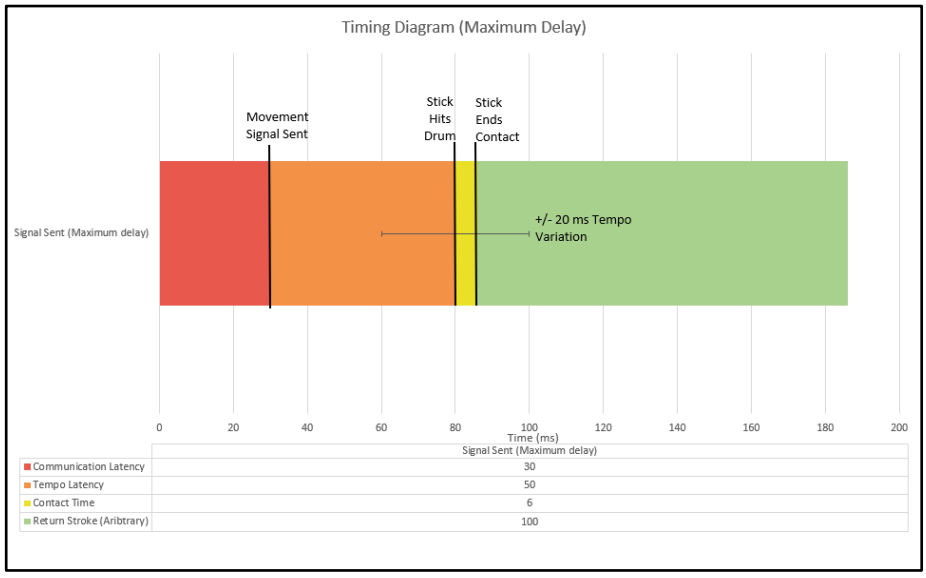


Figure 8: Latency Timing Diagram

4.0 - System Architecture Analysis

4.1 - Actuation Parameters

We developed several parameters based on our objectives and design requirements. These parameters were used to compare and discuss various designs as will be covered in this report. The parameters are organized into three sections: wrist, elbow and shoulder actuation.

4.1.1 - Wrist Actuation

To reproduce the complex motions of the wrist and perform across multiple instruments, the stiffness of the wrist actuator must vary to create a set stiffness for varying playing modes and instruments. It is critical that this variability is implemented at the wrist as the drumstick will mount directly to the wrist actuator. To create a system with variable stiffness, we are implementing a BLDC actuator with FOC to create an active impedance control system. We have concluded from our research that this is a core component. As such, it is necessarily included in all potential designs.

4.1.2 - Elbow Actuation

While the wrist is the core component in regard to the musical performance of the system, the elbow actuator is the primary driver of visual expressivity through the generation of anticipatory gestures. The elbow actuator will drive the entire forearm segment of the system, including the wrist actuator and end effector. Due to this, the torque generated from the wrist actuation will act as a load on the elbow actuator as well. In addition to the torque generated from the acceleration of the masses in the arm, the torque created from the wrist will be added in as well.

4.1.3 - Shoulder Actuation

With the wrist and elbow actuators driving the musical performance, expressivity and visual expressivity of the system the shoulder actuator will be primarily responsible for positioning the forearm segment of the robot and will not require the rapid accelerations that the wrist and elbow's functionality entails. The shoulder actuator will have to deal with the masses of the other components and the additional loads created by the torque of the elbow and wrist actuators.

4.2 - Design Options

With the actuation parameters defined for the system, our team developed a number of designs for the system. The major variation in each design is a result of different methods to handle the high torque requirement at the shoulder.

4.2.1 - Three Joint Actuator Design

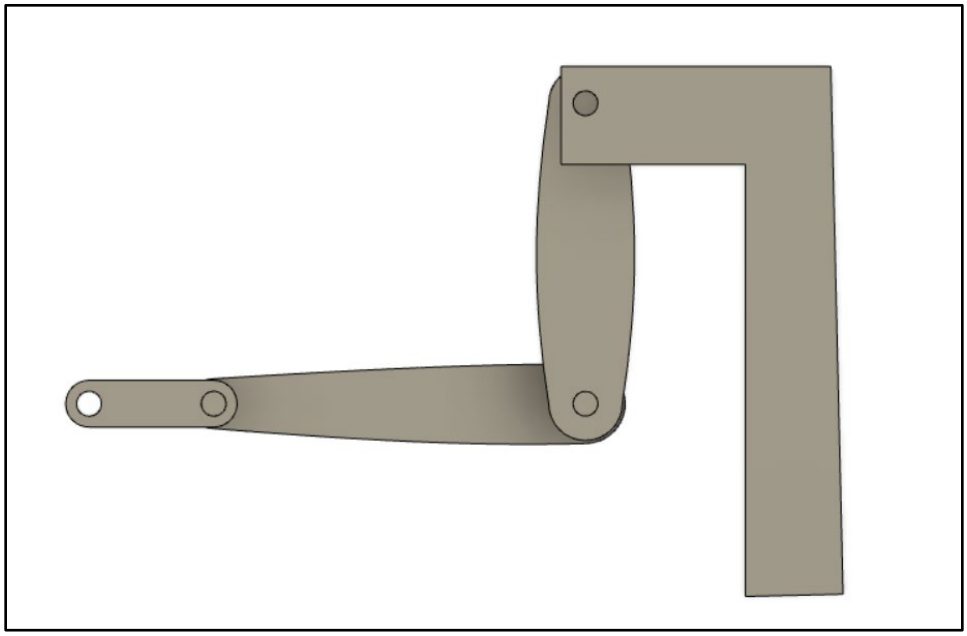


Figure 9: Three Joint Actuator Design Mockup

The first design proposed was the three joint actuator design, which utilizes three rotary actuators for the wrist, elbow and shoulder. This design is shown in Figure 9. This design is the most anthropomorphic in nature, with a link for the bicep and forearm and the end-effector attached to the BLDC actuator at the wrist. This design would allow for clear visual expressivity through the use of anticipatory gestures involving the elbow and potentially the bicep. It would also allow for greater modularity with the shoulder actuator allowing for adjustments in the height of the forearm segment relative to the percussion instrument. It would however require a high amount of torque at the shoulder actuator as the shoulder actuator would have to account for the loads at the wrist and elbow actuators directly. Additionally, horizontal displacement across a drumhead for changing timbre through playing location would require adjustments in vertical and horizontal displacement.

4.2.2 - Internal Linear Actuator Design

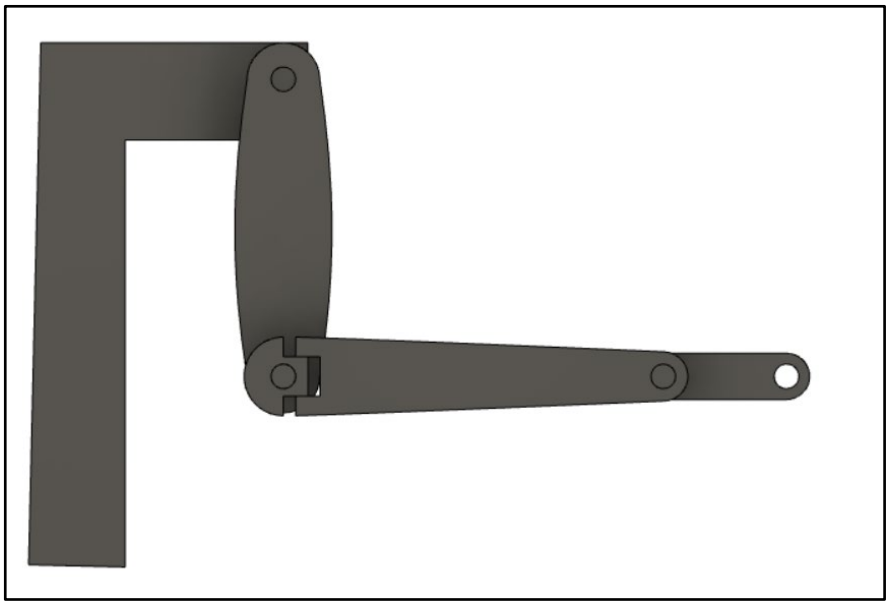


Figure 10: Internal Linear Actuator Design Mockup

Another design proposed is the internal linear actuator design. This design would utilize three rotary actuators like the three joint actuator design and add a fourth actuator within the forearm segment of the robot. This design is shown in Figure 10. Within the forearm of the robot would be a linear actuator which would actuate the front half of the forearm segment to create exclusively horizontal travel across the drumhead for change in timbre during performances. This design would be similarly expressive in comparison to the three joint actuator design; however, it would not be entirely anthropomorphic in design as the forearm segment could extend outward. This design would also create more load on the shoulder actuator. Additionally, the linear actuator would need to be mounted within the forearm since it would need to be compact while still having a stroke length of 18 cm (per our design requirements), and would necessitate an expensive actuator such as a voice coil actuator.

4.2.3 - External Linear Actuator Design

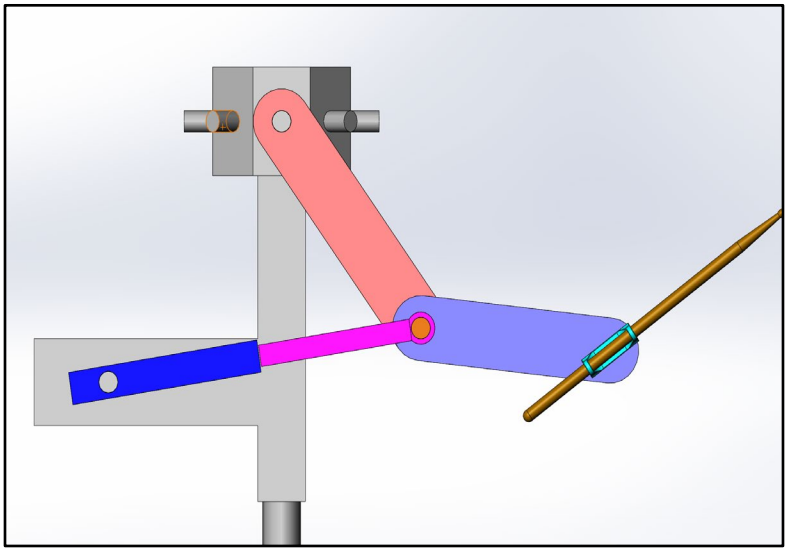


Figure 11: External Linear Actuator Design

The External Linear Actuator Design pictured in Figure 11 above, was created to reduce the complexity of the internal actuator design. It simply moves the linear actuator from internal to the arm to external and behind the elbow joint. This design would be simple to implement but suffers from several issues. The addition of an actuator behind the elbow significantly reduces the anthropomorphic nature of the system. In addition, at the desired stroke length, actuators of this type are very large, and would significantly increase the footprint of the system. The actuator also moves slowly. Finally, the motion of the actuator could be distracting to an audience and could reduce the impact of preparatory gestures that the system would create.

4.2.4 - Horizontal Slider-Crank Design

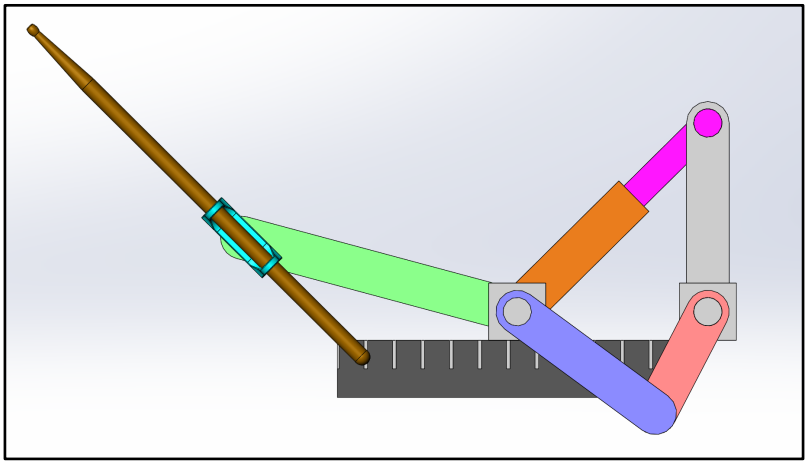


Figure 12: Horizontal Slider-Crank Design

To simplify the actuation of the shoulder, two designs were made that translated rotary motion into linear motion. Both of these designs utilized slider-cranks to make this translation. Figure 12 illustrates a simple horizontal slider-crank. The crank shown in red can make a full rotation about the pin joint. As it rotates, it pushes the blue coupler which changes the horizontal location of the elbow. The minimum displacement of the elbow occurs when the crank points to the right. The maximum occurs when it points left. As shown, the elbow and therefore the drumstick have a range of motion between zero and twenty centimeters. This can be easily changed by changing the length of the crank. This design successfully simplifies horizontal motion into rotational movement. It also allows for simple and precise locational control. However, this design necessitates a variable length shoulder link, shown in orange and pink. The addition of the extra links also reduces the anthropomorphic nature of the system.

4.2.5 - Angled Slider-Crank Design

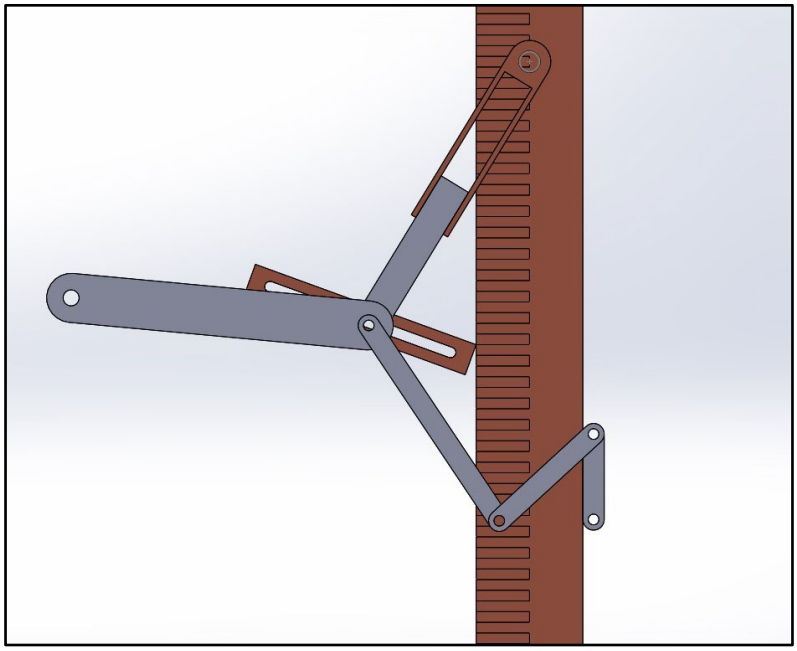


Figure 13: Angled Slider-Crank Design

The angled slider-crank, shown in Figure 13, incorporates many of the elements of the horizontal slider-crank design. However, the elbow joint is on an angled track, as opposed to a linear slider. This allows for both horizontal and vertical movement. However, it suffers from several of the same issues as the horizontal slider-crank. This design again requires a variable length shoulder link. In addition, it is significantly less anthropomorphic with the addition of the extra links when compared to the simpler systems shown in previous sections.

4.2.6 - Decision Matrix

After determining different design options for the system, our team created a decision matrix in order to analyze the advantages and disadvantages of the different options. To analyze these differences we distilled four key categories from our design objectives and requirements: how well will the design demonstrate visual and aural expressivity, how much will it cost, how effectively will it enable modular playing and how much power would it consume, which can be seen below in Figure 14.

These categories were compared both relative to each other and based on our requirements when ranking. Weights were applied to these categories based on their importance. Visual and auditory expressivity was ranked highest, as the performance of the robot is largely based on creating aural and visual expressivity similar to that of a human player. Cost was second highest, as it is our largest limiting factor, working on a budget of $1000 provided by our major departments. Third was modular playing, as determining which designs would help us achieve modularity was one of our design objectives. The modularity of the system was defined

by the ability to play percussive instruments of varying size in different locations. In the case of the horizontal slider-crank design, the horizontal bar limits the change in the height of the elbow. Fourth was power consumption, which while important was more of a byproduct of our other factors to be considered.

Each design was ranked in these four categories, with a range of 1-5 with 1 being the worst and 5 being the best in comparing the different designs. From calculating the total weighted score of each design after they had been ranked, we determined the design which seemed best suited to our objectives and requirements based on our analysis: the three joint actuator design.

	Modular Playing	Visual and Auditory Expressivity	Cost	Power Consumption	Total Score	Total Weighted Score
Three Joint Actuator	5	5	3	3	16	42
Horizontal Slider-Crank	1	2	5	4	12	28
Angled Slider-Crank	2	2	5	3	12	30
Internal Linear Actuator	5	4	1	2	12	31
External Linear Actuator	2	2	2	5	11	23
Weights:	2	4	3	1		

Figure 14: Decision Matrix

4.3 - Initial Design Criteria

4.3.1 - Static Analysis

We began with a static analysis of the system by examining the elbow and shoulder components at their maximum static load. The generalized free body diagram of the arm can be seen in Figure 15. In the normal use case, the elbow will operate between -45 and 45 degrees deviations from the horizontal axis. The maximum load on the elbow joint would exist when the elbow is parallel to the horizontal axis. The magnitude of the torque the motor needs to produce to keep the system in static equilibrium is plotted in Figure 16. The maximum static load is around 4.3 Nm. This is a baseline figure for the nominal torque the elbow actuator should be able to produce.

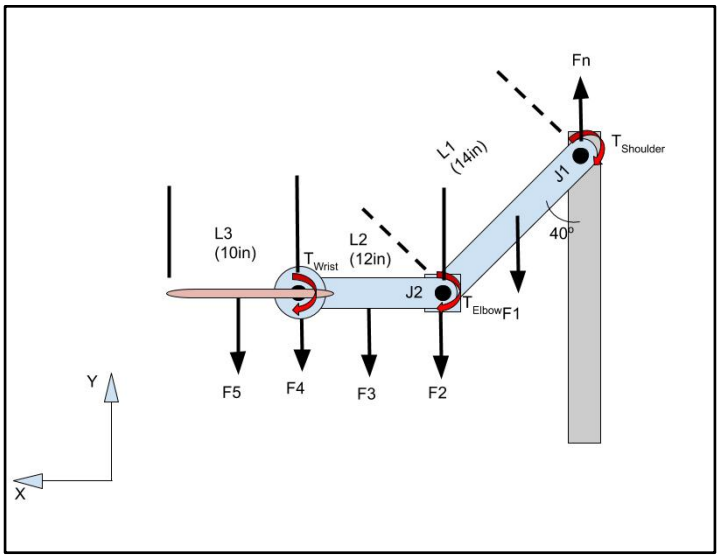


Figure 15: Free body diagram of drumming arm

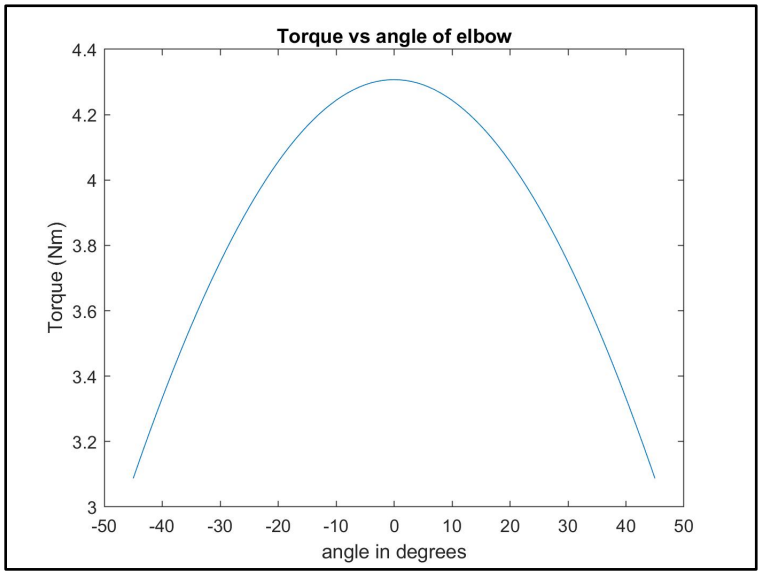


Figure 16: Static analysis of the elbow torques

The torque the shoulder actuator needs to produce for static equilibrium is shown in Figure 17. The plot depicts the torques when the shoulder actuator is parallel with the vertical axis, up to full extension. The elbow actuator is presumed to be fully extended at its max moment condition. As expected, the static torque requirements of the shoulder are higher than the elbow, with a max torque of 9.3 Nm. These values are a baseline figure for the nominal torque the elbow and shoulder actuators should be able to produce.

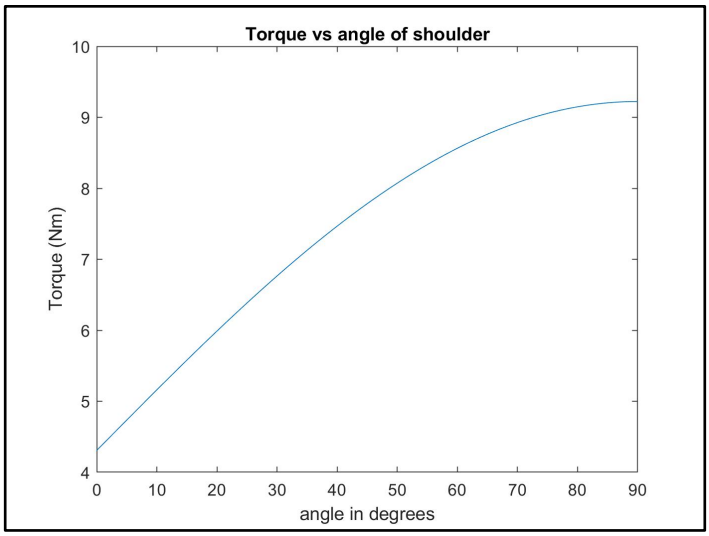


Figure 17: Static analysis of shoulder torques

4.3.2 - Dynamic Analysis

The dynamic loads of the system are important to understand so the oscillatory motions of the anticipatory gestures can be recreated. The shoulder's main purpose is to move the arm forward and back to play at different parts of the drumhead. Due to the application of the shoulder, the accelerations created by this movement would lead to trivial dynamic loading. Dynamic calculations were not performed due to this criterion. The dynamics of the elbow were analyzed in two different cases. The first case analyzed the instantaneous torque required to stop the elbow joint in 100 ms for different onset intervals. The onset intervals range from 100 to 1000 ms, representing the downstroke for tempos of 300 to 30 BPM. For each interval, the angular velocity of the arm is 45 degrees divided by the onset interval. Figure 18 shows that the maximum torque the motor needs to supply for this motion. The maximum required torque can be limited by choosing an interval greater than 100ms where the elbow will no longer contribute motion to the system, similar to the reduction of amplitude in a human player's motions when playing at faster intervals.

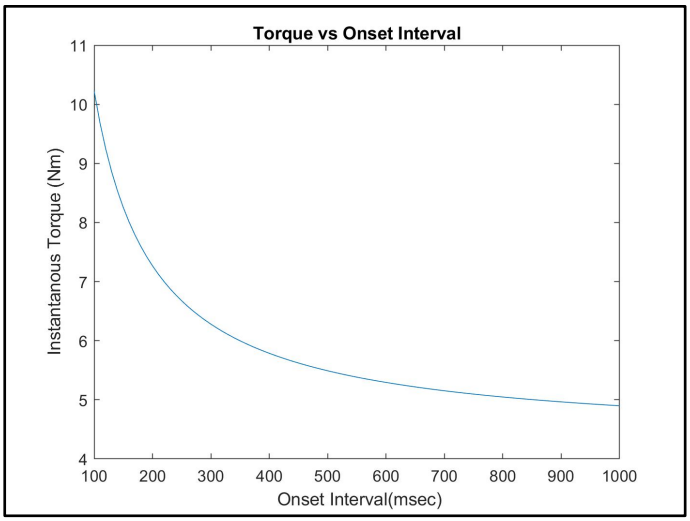


Figure 18: Dynamic torque of elbow vs onset interval

Figure 19 shows the relationship between the required stopping torque and the deceleration time. The maximum required torques can be drastically lowered by increasing the time of deceleration. By limiting the onset interval to a minimum of 200ms, with a stopping time of 100ms, a max torque of 7 Nm is required from the elbow motor.

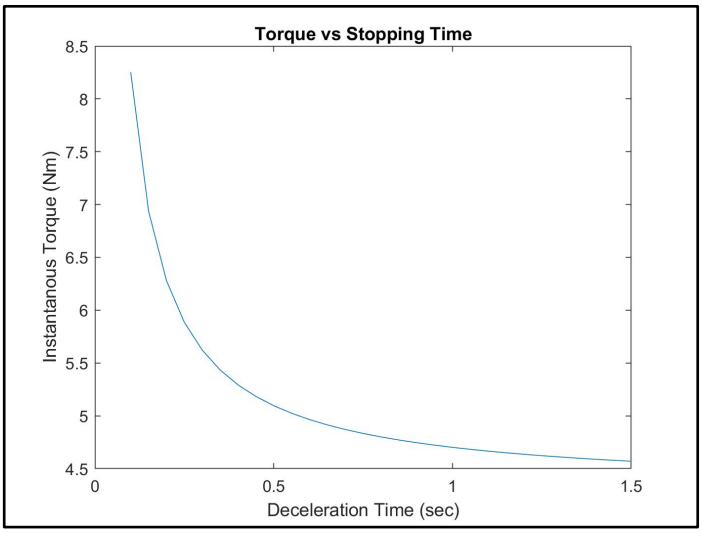


Figure 19: Dynamic torque of elbow vs stopping time

The wrist motor requirements were determined by analyzing the wrist motion at the max frequency of playing: 10 Hz or 100ms intervals. The required power of the motor was calculated by determining the torques required for a resultant force of 100 N tangential to the arc of the drumstick's motion. The 100 N was defined as the peak force measured from humans playing in Dahl 2011. This dynamic torque requirement is 26.6 Nm at stall conditions. The speed of operation was varied by changing the required angular displacement for each stroke. As the onset interval decreases, percussionists reduce their playing stick height. At 10 Hz, the world's fastest drummer appeared to have an angular displacement of approximately 15 degrees. To replicate

that, our motor would need an output power of 72 Watts. The relationship between power and angular displacement can be seen in Figure 20.

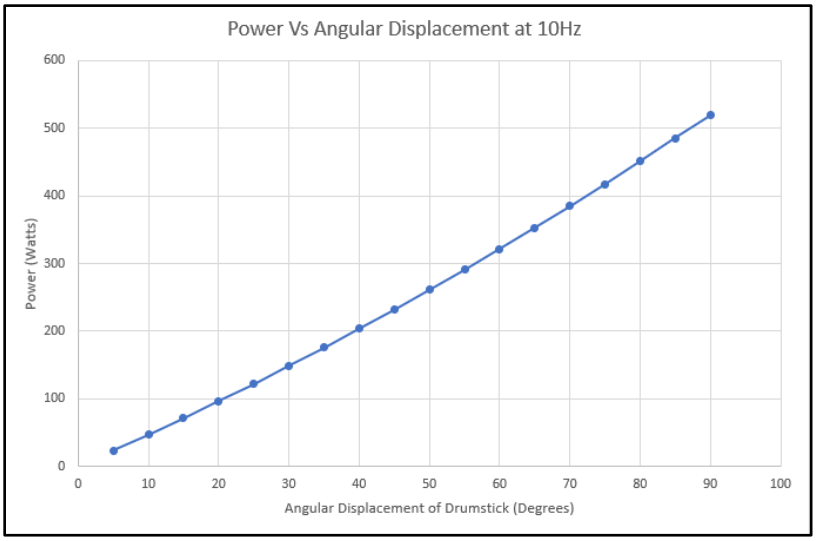


Figure 20: Power of the wrist motor vs angular displacement

4.3.3 - Power Requirements

The power requirement of the system was determined from the peak conditions of each actuator. As previously stated, the wrist joint will require 71.16W of power. Analysis of the static and dynamic loads shows that the elbow joint will require 25W and the shoulder will require 20W of power. This results in a total power of 116.16W.

4.3.4- Reducing Energy Consumption

Throughout the design process, it became clear that efforts to lessen the torque requirement for the elbow and shoulder actuators would be mechanically and fiscally beneficial. As such, the team began researching and developing ideas for reducing power consumption through lessening the torque loads. One proposed idea involved moving the elbow actuator higher up on the bicep segment of the arm instead of placing it at the elbow joint. The actuator would then be belt driven in order to create the necessary movement at the elbow joint while reducing the torque required in turn for the shoulder actuator by reducing the distance between the elbow actuator and shoulder actuator.

Beyond this, the use of springs was proposed as a method to reduce the load through the creation of gravity compensation. This would involve using a torsion spring, linear spring or constant force spring in such a system. Creating gravity compensation requires a system in which the torque created on the arm is always equal to the torque produced by the spring. To test this system, the team designed, and laser cut a wooden model at one half the scale as a prototype of a spring gravity compensated system. It can be seen in Figure 21.

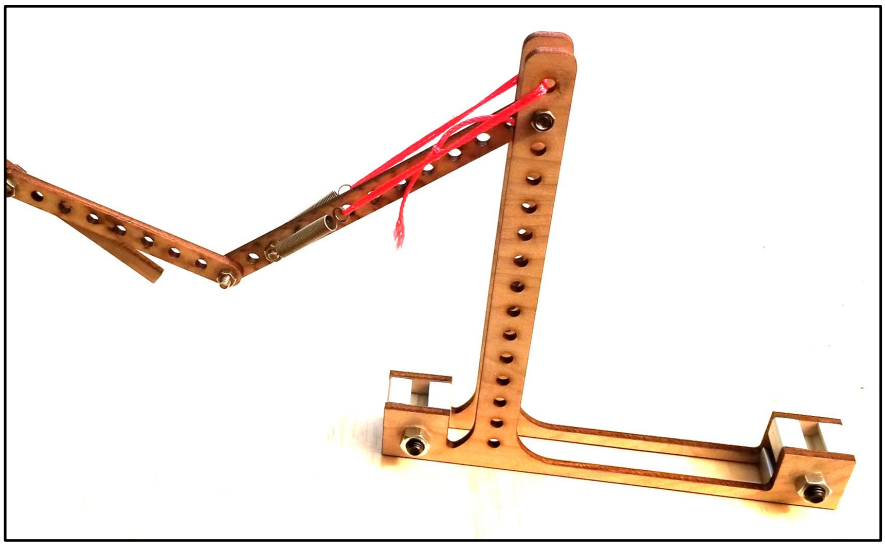


Figure 21: Prototype of Gravity Compensated Spring System

With confirmation from the prototype that such a system was viable, we analyzed the torque required with spring compensation, as well as the spring force created in response to the angle of the arm in the x and y directions.

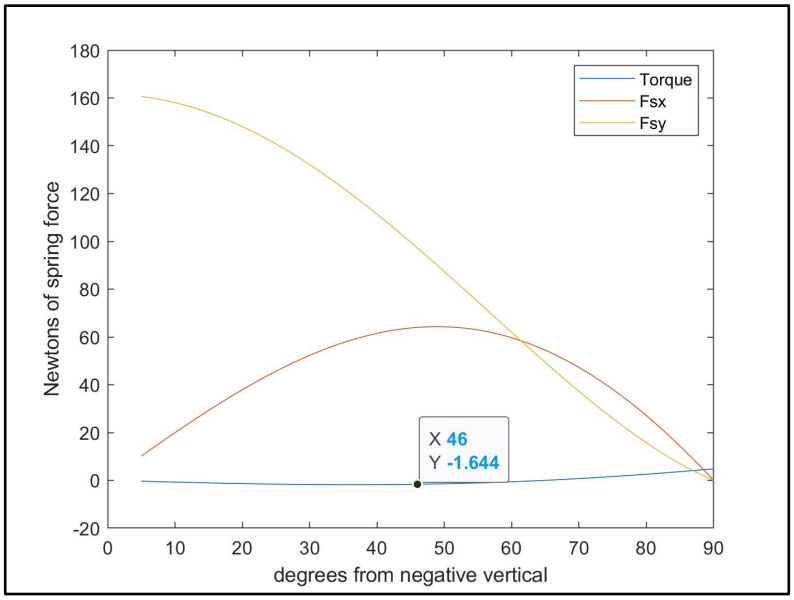


Figure 22: Required motor torque with spring compensation

In Figure 22, the team analyzed the spring force required for the spring in the x and y directions as well as the torque needed for the shoulder actuator in a system in which the shoulder actuator utilizes a spring for gravity compensation. This data supports the conclusion that the spring system can help reduce the load on the shoulder actuator, as indicated by the blue curve. It demonstrates that with this spring system, the torque is either nonexistent or small enough to be negligible. Interestingly, the torque curve suggests that there will be points where

the motor will oppose the spring to maintain static equilibrium. This is shown in Figure 19. In this case, the arm is at a position of 46 degrees and requires a torque of -1.644 Nm.

4.3.5 - Motor Selection

Based on the actuation parameters established through our objectives and design requirements, we first set out to determine the wrist joint actuator. We reached out to Maxon motors for a potential sponsorship based both on research of prior art and the use of Maxon actuators in previous student research at WPI. Maxon graciously agreed to provide our team with a motor at a discounted price. The Maxon motor is an EC Flat motor that provides 260W of power and has a stall torque of 14.6 Nm. The motor was used as the wrist joint and was implemented as an active impedance control system, with a 43:1 planetary gearbox and encoder.

Cost, torque, and power requirements, and manufacturer lead time were considered when finding motors for the system's elbow and shoulder actuators. As the process of acquiring a sponsorship with Maxon took over a month and the need for a longer lead time due to the custom request, we focused on motor vendors instead of specific manufacturers for the procurement of the elbow and shoulder actuators. A windshield wiper motor was selected for the elbow joint. The motor has an unloaded speed of 50RPM, rated torque of 5.9Nm and stall torque of 19.9Nm. This meets the requirement of having a motor able to provide eight Nm of torque. The motor is inexpensive and produces 50 Watts of power.

4.4 - Electrical Design

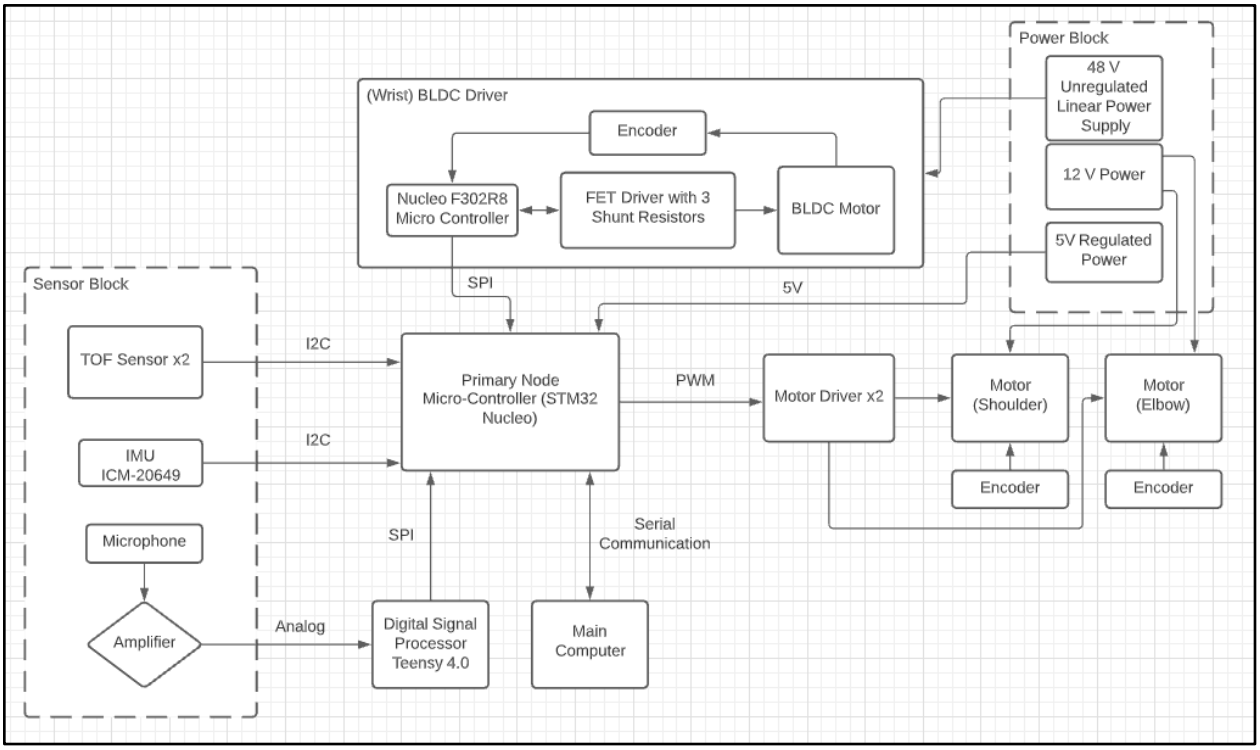


Figure 23: Electrical Architecture Block Diagram

The electrical architecture is shown in Figure 23. The main microcontroller will be a STM32 Nucleo board that interfaces with the various peripherals. Three motors will be controlled from the main microcontroller. The BLDC wrist motor controller will consist of a separate microcontroller, FET driver, BLDC motor and encoder. The FOC algorithm is resource intensive and will run entirely on its own microcontroller to simplify the main control loop. This microcontroller will interface with the primary microcontroller through SPI to send and receive setpoint and position information. The other two motors are DC brushed motors which are controlled with a standard PWM controlled driver. Each motor will have an encoder to track the angle of rotation. The sensor block contains the sensors used for calibration and hit detection. Our system comprises of two Time-of-Flight (ToF) sensors utilized for position data, an inertial measurement unit (IMU) for acceleration and movement data relative to the starting position of the arm, and a microphone for audio analysis of the strike. The amplified signal from the microphone will be processed by the Teensy 4.0 which has a FP unit for audio processing application and an Analog-Digital Converter (ADC) for converting the analog signal of the microphone into the digital domain. A main computer will connect through serial communication to send the commands to the microcontroller running the motor controllers. The system is powered by a 48V unregulated power supply to handle the high current peaks drawn by the BLDC motor. The power supply has a 5V regulated line to power the microcontroller and sensors. Lastly, the motors are powered by a separate 12V power supply.

4.4.1 - Motor Driver Choices

A driver capable of field-oriented control (FOC) is required to implement the VIA at the wrist. The necessary components are a high-side driver, power MOSFETs, three shunt resistors. The high side driver and power MOSFETs are required to properly handle the peak currents that will arise with the oscillating motion of the BLDC. The three shunt resistors are used for sensing current in the three phases of the BLDC, which is required for the proper operation of the FOC algorithm. In addition, the driver must be able to operate at 48V. Running the motor at a higher voltage reduces the current requirements. The parts were chosen from existing development kits to expedite testing. The chosen driver is TI's DRV8353. The development board for this chip operates between 9-95V, with a 15A continuous current limit. The board is designed for FOC with the inclusion of internal low-side current shunt amplifiers. The chip itself has both over current protection and thermal warning and shutdown. These safety features should ensure that the driver is operating safely.

4.4.2 - Sensor Choices

Aural feedback is a critical component for the arm to perform as intended. In order to capture aural information, a microphone was selected. The microphone is used for playing volume measurement and onset detection.

Sensors need to be implemented for hit detection and dynamic feedback. Several sensors were considered for dynamic feedback and locational awareness. A strain gauge can be used to determine when the drum is hit through deflection of the stick. However, initial testing using a culler strain gauge to form a quarter bridge did not yield consistent numbers. The strain gauge was attached to a drumstick and tested by hitting the drum and detecting the deflection through change in voltage. The results however were inconsistent, varying largely from a range of 0.02-2V. The circuit was decidedly impractical and other sensors were implemented for drum hit detection.

The arm will also have an IMU that will track the acceleration and orientation of the robot. The chosen IMU is the ICM 20649 chip. This unit will be able to handle the acceleration of the drumming motion, with a range of 30*g*. Initial tests show that the IMU can track when the stick hits the drum. Figure 24 below shows the values from the sensor attached to a human hand while playing a snare drum.

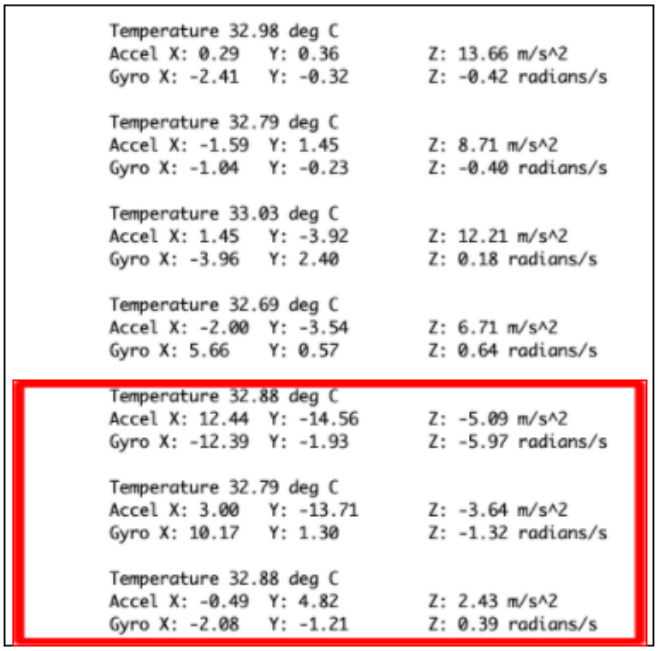


Figure 24: IMU data from testing

The change from negative to positive acceleration on the y-axis represents the change in direction of the stick, as shown in the section boxed in red. This data will be able to confirm contact between the drum and stick and may be used to determine the dynamic of the note.

To avoid hitting the drum accidentally, an obstacle detection sensor will be attached to different parts of the arm. This sensor will also show the distance between the robotic arm and drumhead and will be implemented into the algorithm for trajectory planning. An infrared sensor was obtained to identify the distance from the drum to the robot. Unfortunately, the parameters of the infrared sensor do not fulfill the requirements for the project. Basic testing with a Sharp IR sensor found that there was a noticeable margin of error. The range of this specific sensor is 10cm to 80cm. The reading would show 9cm until the sensor is about 12cm away from the object. From there, the value would be within 3cm of the actual distance. Unfortunately, this margin of error is too large. Based on research into other distance sensors, a ToF sensor was chosen, as they have a higher reading frequency and longer-range applications. ToF sensors send out an infrared laser and measure the time between emission and when the reflection of the laser is received to determine distance (“Time of Flight principle”, 2020).

4.5 - Software Design

4.5.1 - Calibration Process

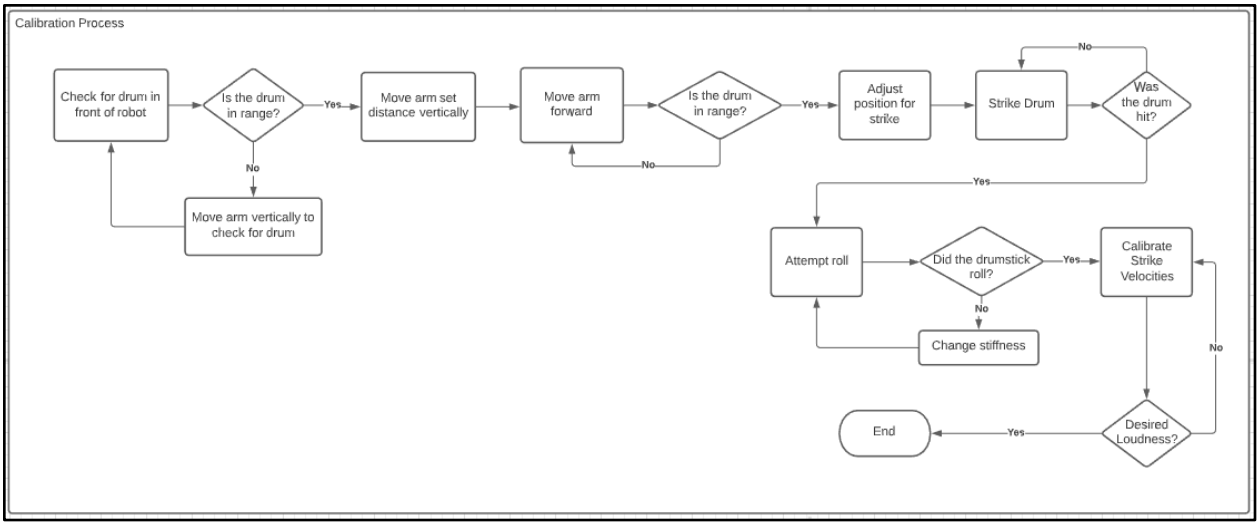


Figure 25: Calibration Process Flowchart

The software design planning began with the calibration process shown in Figure 25. Assuming the robot will start being placed close to the percussion instrument it will be playing, the arm will begin its calibration process by scanning vertically (by actuating the elbow joint) for the percussion instrument using a front mounted ToF sensor. Once the drum has been located the robot will adjust its position to a set distance above the drum to ensure it does not collide with the instrument as it moves forward in the next step. The robot will then move the arm forward slowly until an ToF sensor mounted under the wrist sees the drumhead. Once it has moved a set distance vertically and horizontally into a “striking position” the robot will strike the drum. It will then use the microphone and IMU to determine if there was a strike. If not, the position will be adjusted again, otherwise the robot will attempt a roll.

If a roll is not achieved, the robot will adjust the stiffness of the wrist actuator until the robot produces a roll. From this point the robot will calibrate strike velocities with analysis through audio input from the microphone to determine the dynamic levels on the instrument. At this point, the system will wait for manual input to determine the velocities for the desired dynamic levels in the system’s current environment. This is because dynamic levels vary both based on instrument and setting. For example, the perceived volume in a robotics lab will be different from the perceived volume in an auditorium. If the users do not approve of the dynamic level setting, the robot will try again until the users agree on the dynamic level settings, which will end the calibration process.

4.5.2 - Main Program

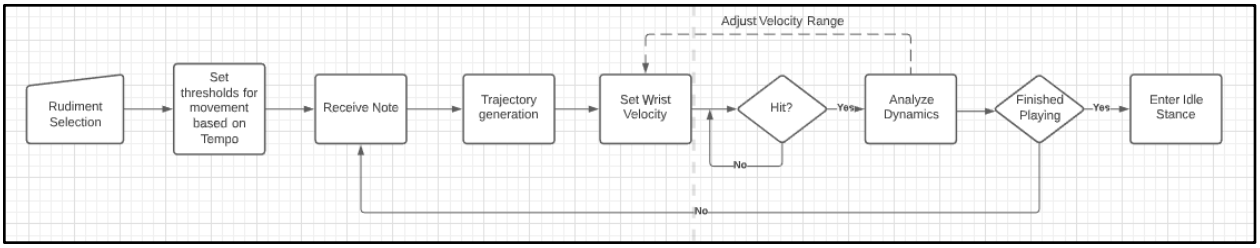


Figure 26: Flowchart for Main Program

After the calibration process, the robot will be in an idle state until users begin the main program, by selecting a rudiment for the robot to perform. The program will then set thresholds for the movement of the arm based on the temporal intervals of the beats in the piece. For example, if the chosen onset interval is a steady 300ms (3.33Hz), the elbow actuator will not be utilized, and the shoulder will only be used for timbral changes. If the desired rate is a steady 750ms onset interval, the elbow actuator will produce a low amplitude oscillation to act as a visually expressive gesture. Once these thresholds have been set, the first note of the rudiment will be processed.

Each note in a rudiment will go through the same process. First, the robot will generate a trajectory based on the note, rate of playing and dynamic level, this path will include movements to demonstrate visual expressivity. It will then set the wrist actuator velocity and perform a hit. If the hit was unsuccessful, the velocity of the wrist will be adjusted for the next notes. This process will continue as long as there are notes to receive, after which point the robot will go into an idle state. This program flow is shown above in Figure 26.

5.0 - Final Design

Three of the four members of our team were quarantined for 10 days near the end of the final term of the project. Given this impediment to the progress of the project, we decided to focus on simulation-based results to analyze the movement of the arm and wrist. The following design section focuses on the changes to the system and simulation design.

5.1 - Updated Requirements

Due to the sudden change in circumstances, the team had to review and revise the original requirements we set out to meet in order to ensure the system could still be evaluated at these requirements within the simulation. This meant that the requirements became less focused on demonstrating the robot performing as we had originally intended, and focused more on demonstrating the capabilities the arm had in order to allow the robot to perform musically. First, our design objectives were updated to the set of objectives described below:

1. Design a modular system that can play on a variety of percussive instruments.
2. Demonstrate ability to perform single and double strokes.
3. Demonstrate motions associated with visual expressivity and the potential for aural musical variety.

With updated design objectives established, we also updated our design requirements from the system, ensuring they could be evaluated through the simulation.

1. The robot should produce output velocities throughout the range of 1-9 m/s.
2. The contact should result in an impulse time of less than or equal to 6ms.
3. The system should have a strike range of 18 cm.
4. The temporal deviation of hits must be less than +/-50ms.
5. The system will have a maximum frequency of 10 Hz for onset hits.
6. As the onset interval decreases, the amplitude of oscillations will decrease with a target goal of an oscillatory motion having a period of 600ms (1.67Hz) with an angular displacement of +/- 25 degrees.
7. The drumstick tip should follow the trajectory of a damped oscillation to properly play double stroke rolls at varying speeds.
8. Discern a drum hit through sensors while filtering out external noise sources.

5.2 - Mechanical Design

In order to analyze and visualize all designs, several iterations of computer-based designs were created. Many parts and the first iteration of the arm were created using Autodesk Fusion360. These files were used to analyze, plan, and begin building the system. When the team pivoted away from a full physical build of the arm, the assembly was ported to SolidWorks. The transition was necessary as SolidWorks had the necessary toolbox to export the assembly into the Simulink arm simulation as is discussed in later sections of this report.

The mechanical design of the system was created to mimic human proportions while meeting the desired design criteria for a successful robotic percussion system. The system consists of three subsystems each of which is explored in greater depth later in the report. The first subsystem is the base. The base consists of two side frames that the arm assembly is mounted to. It also serves to mount the spring to the overall arm.

The next subsystem is the arm. The arm is broken up into the bicep and forearm links. The bicep is mounted to the base at the shoulder joint and the bicep connects to the forearm link at the elbow joint. The forearm joint moves with the elbow which is driven by the elbow motor that is contained in the bicep.

The final subsystem is the wrist system. The wrist contains the end effector and is composed of the motor, gearbox, stick holder, and several sensors. The wrist was designed to meet our design criteria while minimizing mass. The project was sponsored by Maxon Motors who generously provided one of their motors and gearboxes at a heavily discounted price.

As designed, the wrist subsystem will provide all of the motion necessary to meet the playing requirements. The wrist motor, gearbox, and other components were chosen to be able to meet these requirements with no assistance from the rest of the assembly. The bicep and forearm are responsible for the positioning and visual components of the design requirements. However, in the event that the arm system would be able to assist the wrist’s ability to play, this ability would be quantified and would allow the system to function in a greater capacity than designed.

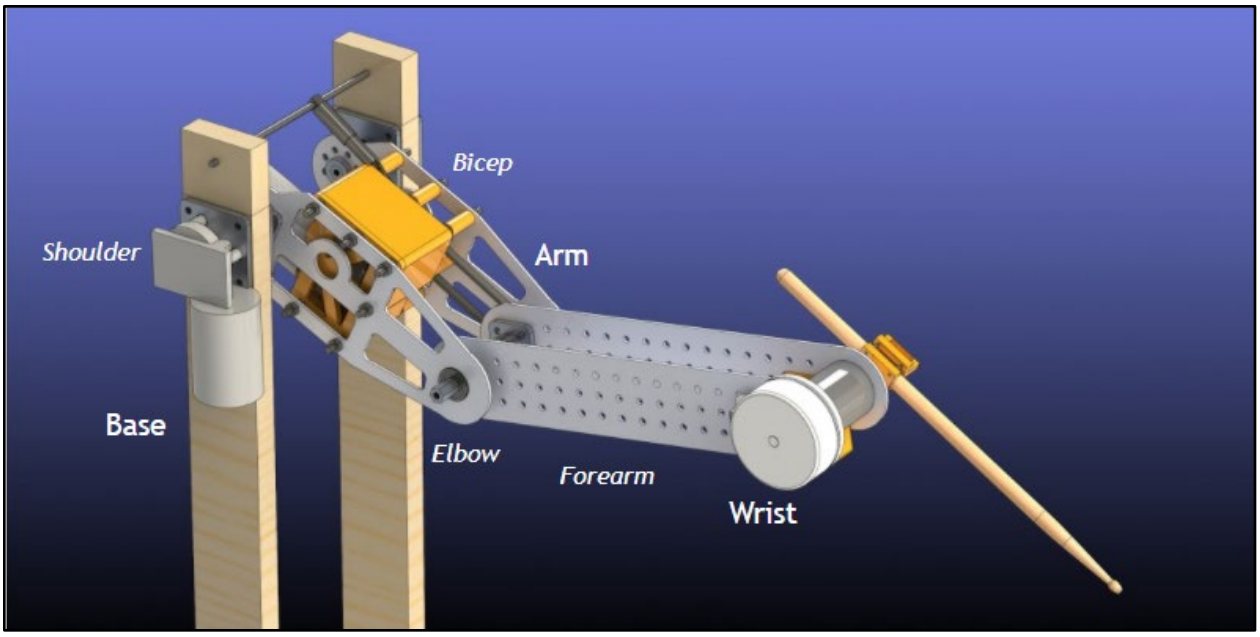


Figure 27: Finished CAD Render of the Final Arm Design

5.2.1 - Base Design

The base was designed to be simple in construction while providing the stability necessary for the arm to play at great speed and with the specified dynamic range. For simplicity and cost efficiency, the base was designed to be constructed out of simple wooden beams available from local hardware retailers. The bottom of each side of the base included struts for added stability. The top of each base section includes two custom machined mounting plates. One set of plates interface with the shoulder motor and provide a second point of contact from a bearing set into the plate. The other base simply provides two bearings that add stability to the arm.

Two axles were chosen over a single axle passing through the bicep's shoulder joint as it allowed the center of mass of the elbow motor to be in line with the rotational axis of the shoulder, as can be seen in Figure 27. This greatly reduced the torque applied to the system by the weight of the arm and its components. However, it did increase the complexity of the build as special care was needed to ensure the bearings, bearing mounts, and motor were all properly aligned.

5.2.2 - Arm Design

The bicep and forearm mechanical segment of the arm was designed to mimic the proportions of the human arm while incorporating space and design constraints for other critical components. The bicep and forearm lengths were chosen to match the lengths of an average human arm. From measurements, these lengths were determined to be 30.5 cm and 35.5 cm respectively.

In the final design, the bicep serves to contain the elbow actuator and move the arm about the shoulder joint. The link was designed to be high strength while minimizing mass. A reduction of mass decreases the power necessary to drive the shoulder joint. The bicep is held together by threaded rods, which hold the elbow motor mount in place. The motor mount can be seen in orange between the bicep arm plates in the above fixture. Additionally, the threaded rods include spacers that give room for the sprocket and belt system that actuate the elbow joint and forearm as shown below in Figure 28.

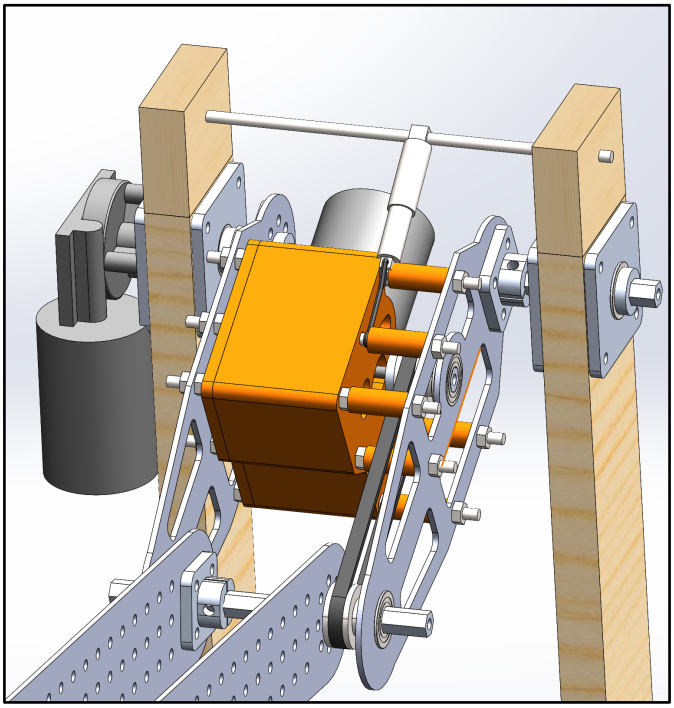


Figure 28: Bicep Design

The motor mount shown in orange was custom designed and 3D printed to meet the desired design requirements. It was created to match the mounting pattern of the elbow motor and includes slotted mounting holes so that the drive belt can be tensioned as necessary to tune the system. In addition, 3D printing the component allowed for the part to be high strength but low mass, contributing to our mass reduction efforts throughout the design.

The forearm actuates vertical displacement of the end effector and serves as the mount for the end effector or “wrist” section. The forearm was also designed to hold a number of electronic components within itself. These components include: the motor driver, wrist motor controller, ToF Sensors, IMU sensor, and all necessary wiring. These all must fit and be placed appropriately within the forearm to allow the arm to move through its full range of motion without impacting the electronic components.

For assembly, the arm plates of the system were outsourced to a third-party manufacturer. The company SendCutSend was employed to obtain high quality parts within the necessary

tolerance. The parts were plasma cut out of aluminum with a thickness of 0.125 inches to maintain strength and remain within the desired weight parameters. As the placement of the internal electronic components of the forearm were not finalized when the parts were ordered, the links were designed with ample mounting space for the components and standoffs for flexibility when building the final iteration.

Finally, the hex shaft was chosen for each of the axles in the assembly. The assembly makes use of ThunderHex from Vex. This was selected for its price, durability, and extruded hole for threaded connections. It also has the advantage of easy power transmission. Parts interfacing with the shaft simply needed the correct hex profile to transfer rotation from motors to the arm.

Once the mechanical design was complete, the mass calculations made clear that the weight of the system would be too great for the shoulder actuator to adequately perform. With this realization, a gravity compensation concept was introduced. To compensate for the mass of the system, a spring was connected from the top of the base struts and attached to the bicep at the threaded rod closest to the base, as seen in Figure 28. Given the height difference between the spring mount on the base and the shoulder joint as well as the spring constant, the shoulder motor experienced a reduced force, allowing it to move more freely. The equations for the calculations of the spring can be found in the MATLAB file attached in Appendix A.

5.2.3 - Wrist Design

The wrist component of the arm had the main purpose of holding the end effector, a drumstick, which would be driven by the VIA. Since our VIA was the Maxon EC90 flat motor with a gearbox that fit our torque specifications, we created separate mounts for the motor and gearbox which connected on the right link. The gearbox shaft would then connect directly to the drumstick.

To connect the two, we designed a drumstick holder. Additionally, this component was designed as a housing for the IMU. This would allow the IMU to be mounted close to, or on the axis of rotation. To ensure that the best design was chosen, this component went through many iterations. These varying designs can be seen in Figure 29 below.

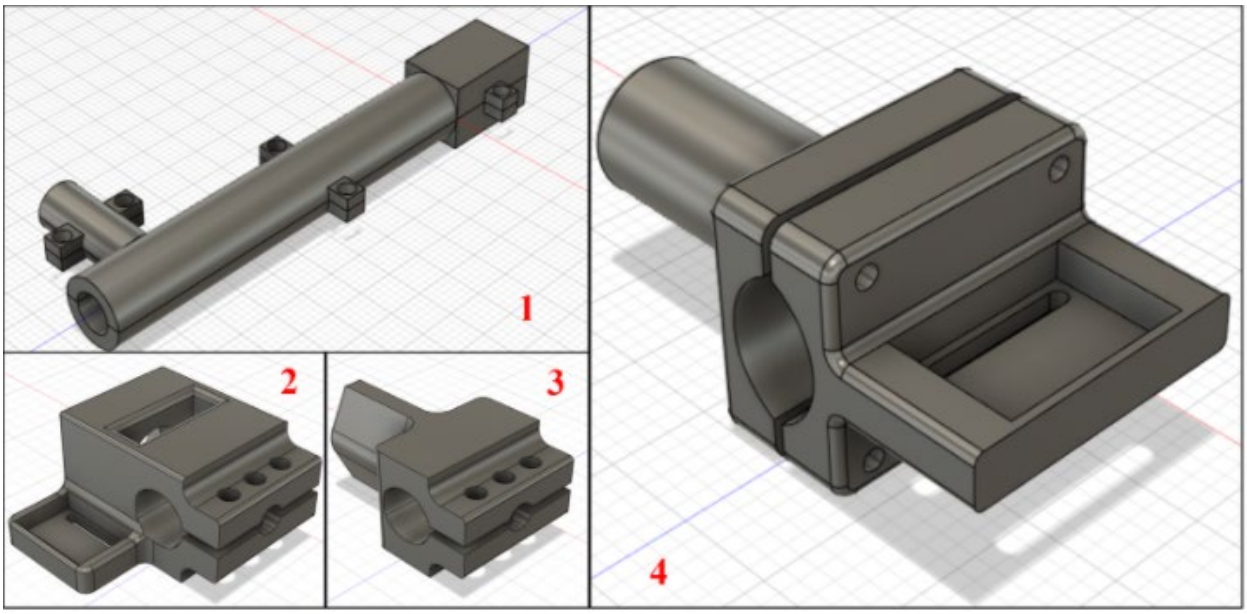


Figure 29: Drumstick Mount Design Iterations

The goal of this device was modularity, with the ability to play on a variety of percussive instruments. As shown in image one, the first iteration encapsulated the entire stick. However, it was abandoned as it could only hold a stick of a single diameter. Furthermore, this design would create unnecessary strain on the IMU wiring as the sensor would be mounted on the end of the stick. The holder in image two was created for testing the sensor array. The holder in image three was created as a prototype stick holder. This was used as a testing implement for drum hit detection and did not contain a space for the IMU. The fourth iteration was designed with an elliptical shape to accommodate multiple drumstick diameters. This design consisted of two halves that would be clamped together and secured with M3 bolts as seen in picture four of Figure 29. The right half of the drumstick holder was designed to fit onto the output shaft of the gearbox with its keyed ready slot. This segment would be machined in order to ensure structural integrity through torque application and high-speed movement. The left half of the drumstick holder was designed to also hold the IMU along the axis of rotation, where gyroscope readings will be most valuable given its function. This section of the drumstick holder would be 3D printed, as it would not require the same material strength as the output shaft.

The ToF sensor mount is placed between the two links under the wrist. There is a front facing mounting space for the first ToF sensor and a bottom mounting point for the second ToF sensor. The orientations of the sensor mounts were designed based on the needs of the robot during calibration and normal operation. The final design for this mount can be seen below in Figure 30.

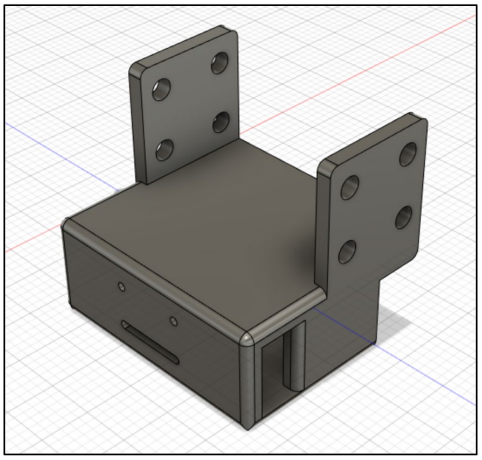


Figure 30: TOF Sensor Mount

5.2.4 - Physical Design

Prior to pivoting to simulation focused deliverables, significant progress was made in the physical implementation of the arm. In accordance with the original build plans, the bicep and several critical components had been created. The creation of the physical parts and building of subsystems led to a number of changes in some part designs.

Two components of the arm required significant iteration to include the desired functionality: the hex adaptors and motor mounts. Both components went through several rounds of design prior to the final manufacture. The first component is the hex adaptor. This component, shown in Figure 31 went through several iterations.

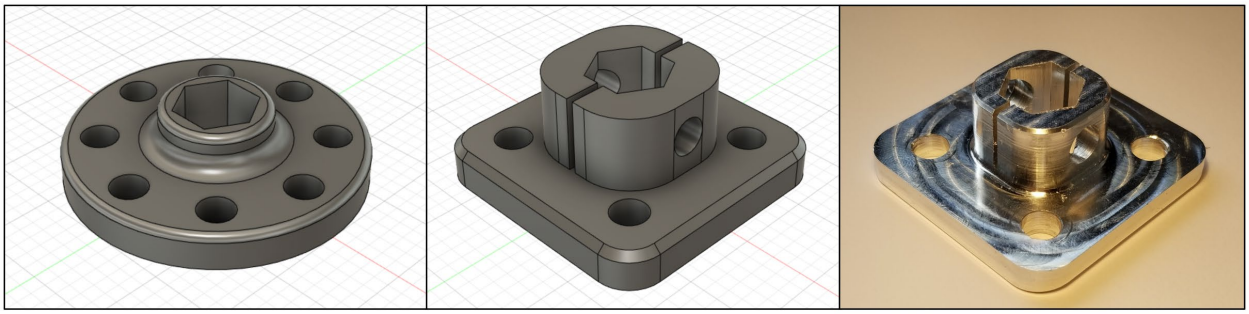


Figure 31: Hex Adaptor and Motor Mount CAD and Finished Piece

As shown above, the initial design process yielded the circular adaptor. It was intended to be manufactured through 3D printing. However, further mechanical analysis determined that a stronger solution was required. This led to the development of the hex adapter shown in the center of the image above. This adapter was machined and included the proper mounting holes to affix it to the arm assembly. In consulting with our advisory team, we realized that repeated loading over time would damage the hex profile, giving the shaft more play in the adapter. As such, the final iteration of the adaptor included a hole drilled perpendicular to the shaft direction

as well as a slot dividing the top portion of the divider. This in conjunction with a hole in the corresponding hex shaft allowed the adapter to clamp to the shaft, providing a simple and elegant solution to reduce the backlash and mechanical deformation on the adapters over repeated loading.

The other component that underwent significant modification was the motor mount block that was used to mount the elbow motor to the bicep plates. The initial, final, and physical implementation of the motor mount block can be seen below in Figure 32.

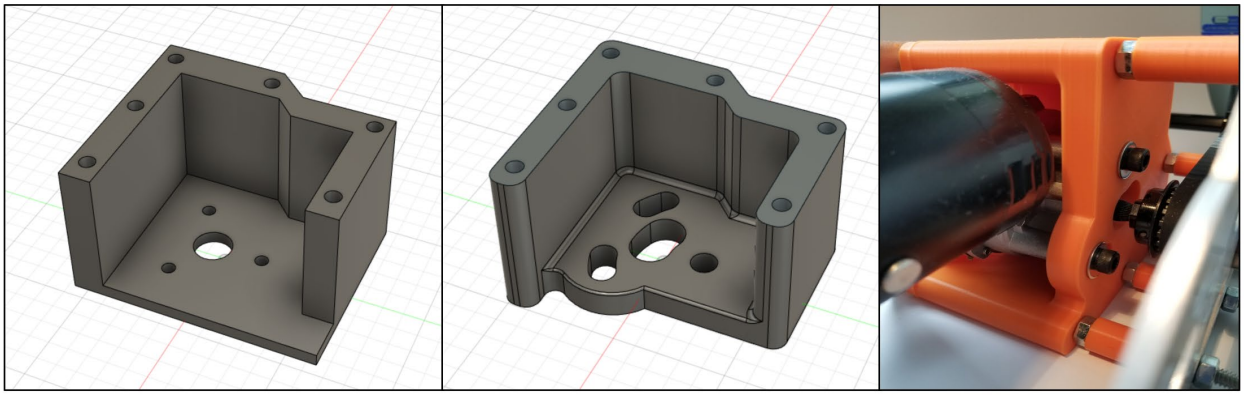


Figure 32: Elbow Motor Mount Design Iterations and Final Model

As illustrated above, the initial design was simple and merely conformed to the chosen motor and used the included mounting pattern to secure to the motor and the bicep. In the second revision, shown in the center, the shape was refined to provide clearance between the motor, and motor mounting block. After discussion with the advisory team, we also included slots for two of the three mounting holes and the center hole. As this motor attached directly to the drive sprocket of the elbow motor, the slots were included to allow for simple tensioning of the belt. By pivoting the motor back about the lower mounting hole, the output of the motor moves translationally back to provide more tension on the belt. This simplified the assembly by eliminating the need for a tensioning component. The final implementation can be seen at right in Figure 32.

5.3 - Electrical Design

5.3.1 - Calibration Sensor Block

Distance sensors were used to determine the arm's location relative to the drum. Given the team's aforementioned quarantine, the sensors were not mounted on the arm. However, testing was done to determine that the sensor block would achieve this task. Two ToF sensors were connected to an Arduino Mega. Servo motors were added to the circuit to simulate the arm motors and potentiometers were used as encoders. The potentiometer readings were then mapped to the servo values, so the potentiometers acted as knobs to control the motor positions.

In order to calibrate the system, two distance sensors were mounted at different positions. The first sensor is mounted to the front of the arm to prevent collision with the drum. While the sensor detects an object in front, the arm will move up. When the sensor reading is out of range, it indicates that the arm is clear of the drum. The location of the sensors is then recorded as a set point. From there, the second ToF sensor is enabled and runs until a distance of 80 millimeters or less is detected. Once this case is true, the positions of the motors are recorded as the edge detection set point. After the two points are set, the system is aware that the arm will not run into the drum and has successfully found the edge. The microphone and IMU sensors are then enabled to begin the hit detection algorithm. A diagram of the calibration algorithm can be found in Figure 33.

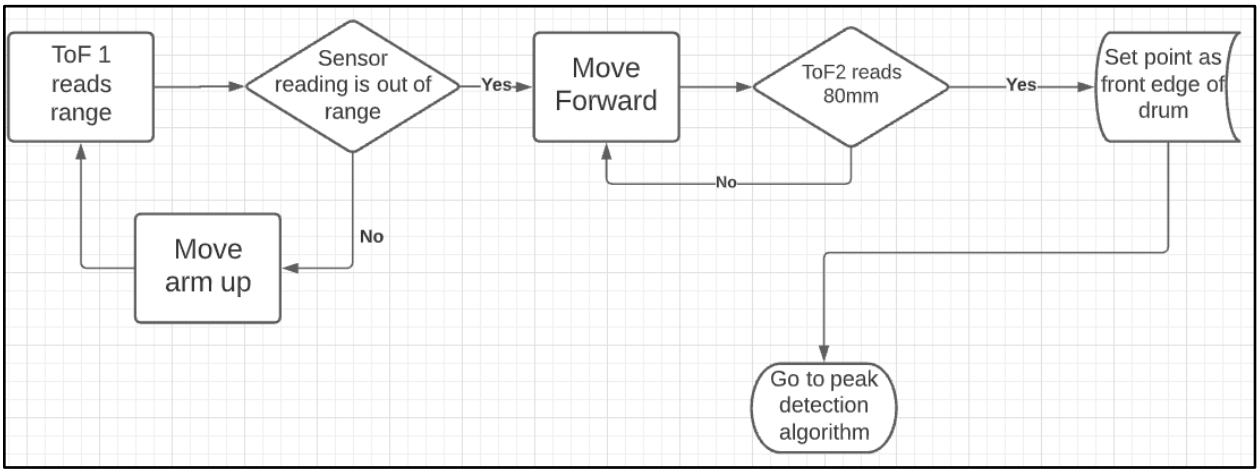


Figure 33: Diagram for the calibration procedure for the arm

5.3.2 - Hit Detection Algorithm

To sense if our robotic arm has properly made a hit on the drum we need sensor feedback. When using only an IMU to detect changes in acceleration, we run the risk of noisy data and detecting accelerations due to other motions of the arm. If just a microphone is used, external noise and other performers might cause false positive hit detections. Combining the sensor data can correlate the change in acceleration with the noise produced to ensure more accurate hit detection.

We used a real time peak detection algorithm that uses z-scores to determine whether a new data point lies outside a defined multiple of the aggregate standard deviation of the data. For this application, the data being used is the analog output of a microphone amplifier, and the angular acceleration readings from the IMU. A benefit of this method is the influence factor, allowing for mini peaks due to noise to be averaged into the data, so only larger peaks are detected. The averaging of the small peaks has the possibility to obscure important data. To account for this, the influence factor was set as a low factor, so that frequency peaks due to noise would be filtered, while infrequent peaks could still be detected. This method should be viable for both the audio and IMU data. The peak detection algorithm was implemented in Arduino and

ran on an Arduino Mega. A hit was detected only if a peak was detected on the IMU and Microphone data. The diagram for the algorithm is shown in Figure 34.

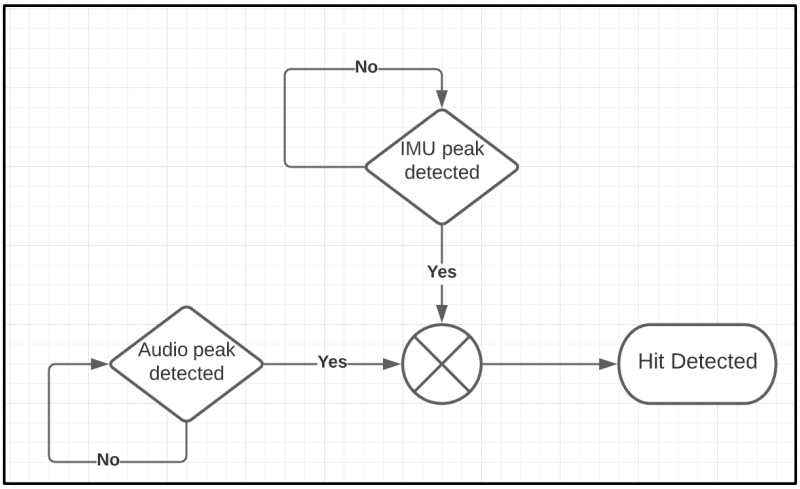


Figure 34: Diagram for the hit detection algorithm

5.4 - Simulation

The team chose to use Matlab's Simulink software in order to simulate the model of the robot as the software provided a robust framework in which we could demonstrate control of the system in a manner similar to our initial planned design, as well as measure and simulate the system's torque, position and velocity. Simulation work was divided up into work focused on dynamic control of the shoulder and forearm and dynamic control of the VIA at the wrist. The simulation work focused on the control of the shoulder and forearm sought to demonstrate the entire system in motion as well as its capacity to meet the motion requirements outlined in our design requirements. Meanwhile the dynamic control of the wrist motor was simulated in order to test the various requirements associated with our VIA.

5.4.1 - Dynamic Control of Shoulder and Forearm Links

To first test the dynamic control of the shoulder and forearm a simple model was created in Simulink using the built-in block objects. Appropriate masses were assigned, and the blocks were shaped to the approximate lengths and widths of the forearm, bicep and base. These were then tied to a world axis orientation. Gravity was then simulated in the y-axis as a basic test of the simulation software.

In order to demonstrate dynamic control of the arm, the digital model of the arm was imported into Simulink. The model was moved from Fusion360 into SolidWorks in order to utilize the XML export necessary for importing assemblies into Simulink. The design files were imported and connected to solid part objects in Simulink and connected through a series of transformations and joints, such as rotational and cylindrical joints. Ultimately all the core components of the arm were imported into Simulink and modeled by the software with the right

connections and orientation. The model was established with the appropriate masses for all the individual components and oriented along the Y-axis. A simulated gravity force was placed on this axis as well. The results of this effort can be seen below in Figure 35.

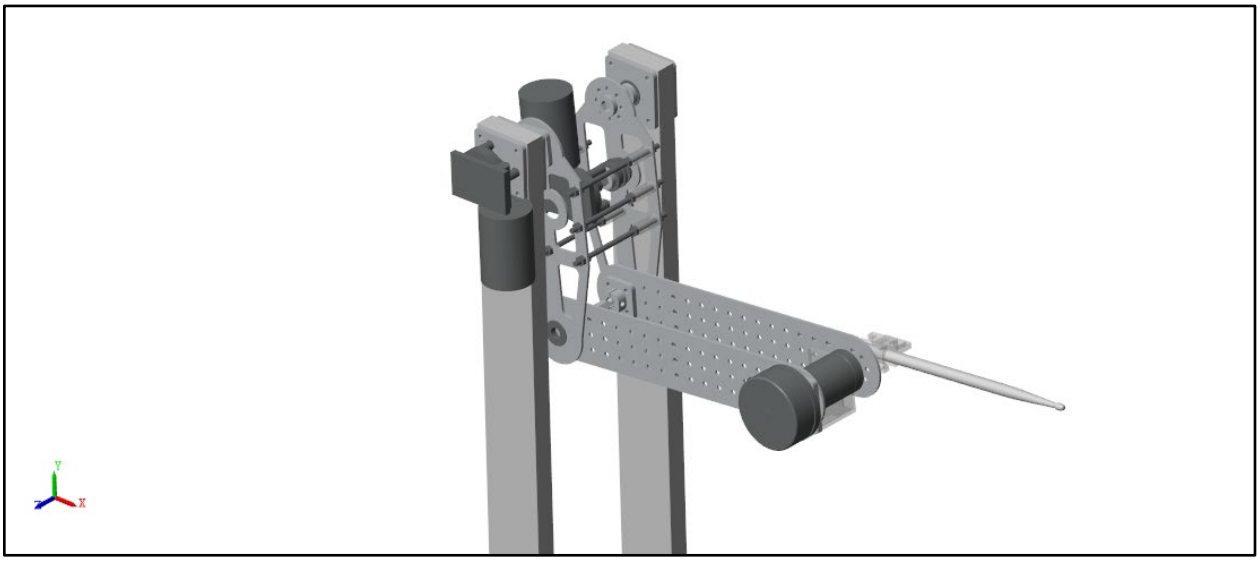


Figure 35: Simulink visualization of the arm model

In order to create a more accurate simulation, a model of the wiper motors we chose for the arm was created in Simulink. The parameters of the motor (such as stall torque, no-load speed and armature inductance) were included as a motor block within the simulation. The rotor inertia parameter of the simulation was set as the moment of inertia of a steel cylinder the same size as the motor casing and rotated through the center. This inertia was added to the rotor of the DC brushed motor model for better representation of the motor's response to a sinusoidal angular trajectory. In addition, mechanical breakaway friction was added to the motor to represent the initial torque needed for the motor to start spinning. The modeled control of the motor is the same as the physical electronics.

Each of these wiper motors for the elbow and shoulder joints were modeled as individual subsystems with a position input and output. The position input acted as a trajectory input for each motor. This would then be processed by a PID controller which was tuned through experimentally determined values. The PID controller would then compute an output signal that saturates at +/-12, representing the voltage that should be applied to the motor. The sign of the output determines the direction of rotation, and the magnitude is used to calculate a duty cycle for the PWM signal sent to the motor. This connects to an H-bridge which then interfaces with the DC motor block. The dead zone of the motor is compensated for by creating a lower limit for the duty cycle to ensure the voltage signals sent to the motor cause a rotation. The DC motor model only has position control, with feedback through an absolute rotary encoder. This model then produces a signal output which can be routed directly to the appropriate joint in the model in order to provide actuation. Each driven joint would be given a positional input as well as an

output for the torque this trajectory would generate. The diagram for this system can be seen in Appendix B.

For the shoulder actuator, the arm was actuated to an angular position of 90° to start, and would hold this position for a second before moving back down to a position of 0°. This actuation was determined in order to demonstrate the movement of the bicep within its normal playing range. Additionally, since the bicep segment of the arm acts as a form of positional control for the height of the end effector during play, as well as creating a horizontal displacement, this type of movement between angular positions is what the bicep segment would be limited to.

The elbow actuator then had a trajectory applied after this movement of the bicep, with a sinusoidal input applied for the position of the forearm. This actuation was determined in order to demonstrate the forearm during play at moderate tempos (such as 600ms onset intervals, or quarter notes at 100 BPM), during which the forearm would oscillate with an angular displacement in order to simulate the visual expressivity of a human drummer, with decreased oscillations as the rate of the playing increases.

With these trajectories applied to the elbow and shoulder actuators, the simulation could then model this behavior, visually and with measurements on appropriate data points. The Simulink file had numerous measurement points added, with scope blocks used to generate graphs of the data generated by the simulation. Measurements of the position of the shoulder and elbow motors compared to their set positions were used to determine the accuracy of the motor's position control, while measurements of the torque feedback generated by the joint was used to evaluate the torque generated by the motion of the arm at the shoulder and elbow to better understand the loads on these actuators. Finally, measurements of the PID, current and velocity within the motor subsystems were used to evaluate motor performance.

5.4.2 - Impedance Control of the Wrist

Proper control of the wrist actuator is achieved through the implementation of the FOC algorithm on an embedded system. This type of control is simulated through MATLAB's Simulink toolboxes. The architecture for both systems are identical, therefore the simulation shows the feasibility of achieving our results given the same control methodology is implemented.

The architecture for the wrist simulation contains two parts:

1.) Physical modeling of the stick to surface contact.

2.) Electrical modeling of FOC control of the motor.

For the physical model, the stick is modeled as a cylinder of the same length, diameter, and weight of the average drumstick. The stick is placed to rotate around the same pivot point as

defined in the mechanical design of the full system. The sphere at the end of the stick is a contact force which interacts with the plane beneath the stick. The simulation models a normal force based on the intersection of the sphere and the plane. The contact force has two parameters for simulating a surface. The first is the spring constant, which represents the amount of force the surface can provide given a displacement. The second parameter is the damping coefficient. This allows for the loss of energy in the interaction between surfaces to be simulated. The last major component of the physical model is the rotary joint. This joint is configured to have a positional input, which generates an output of the position, velocity, and the torques acting on the point of rotation. The position and velocity outputs are mapped to graphs for observational purposes, while the torque is fed back into the system. The torques encompass the moments due to gravity, and the moments created due to the contact force and the length of the stick to the point of rotation. The calculated torque is inverted and fed back into the electrical model of the motor. The physical model is found in Appendix C. The resulting 3D representation is shown in Figure 36. The intersection of the sphere and surface represent the displacement of the surface with an impact.

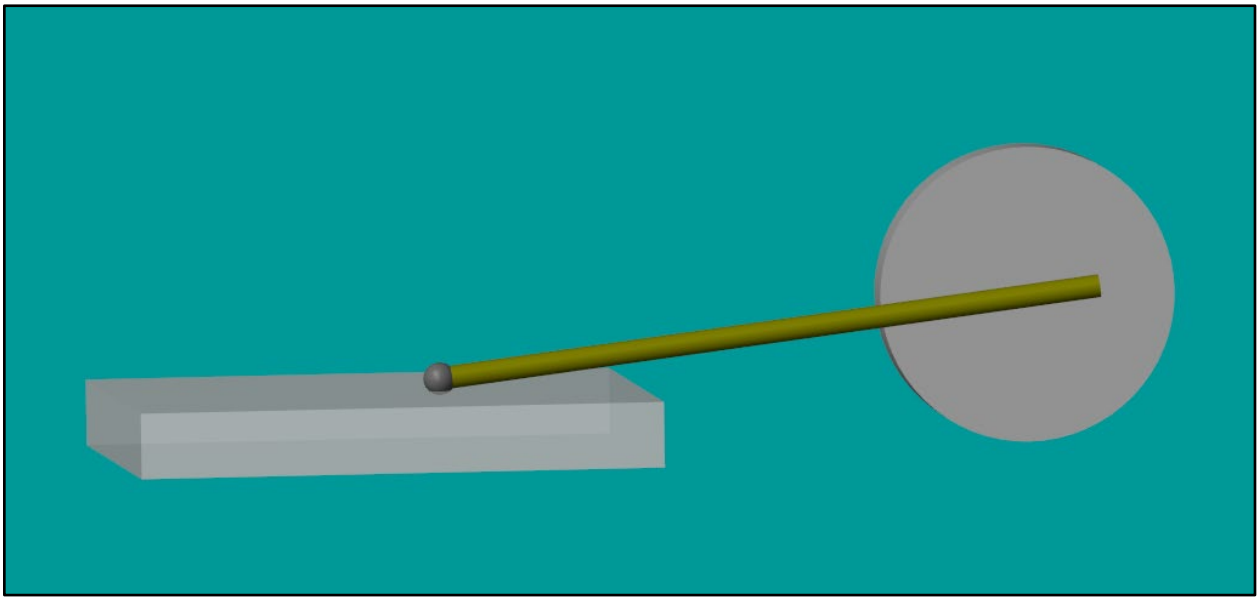


Figure 36: 3D primitive model of stick and drumhead interaction.

The electrical model determines the signals sent to the motor to properly commutate. The change in position of the motor shaft is connected to the physical model. The forces of gravity, interaction with a surface, and inertia of the stick are fed back into the electrical model as a load of the shaft of the motor. This architecture is achieved through the use of the Simscape Toolbox in Simulink as it allows for the coupling of electrical and mechanical systems. The electrical model of the motor contains the control algorithm, required electrical driver, and the model of the motor itself. The diagram for the electrical model can be found in Appendix C. The motor is chosen as a PMSM with the parameters defined by the Maxon motor datasheet. These parameters include the phase resistance and inductance, number of pole pairs, and magnetic flux

linkage. The rotor of the motor is attached to a rotational encoder, and the gearbox is modeled as an inertial load and a division of the output position. The three phases of the motor are connected to three half bridge drivers. The drivers are modeled as MOSFETs with a threshold voltage of one volt. Each of the three drivers is controlled through a pair of two-level PWM signals. The output on the control algorithm is the proper PWM waveforms. The simulation does not properly replicate the effective runtime of the algorithm on an embedded system. The algorithm features a number of floating-point calculations which, while deterministic in length, may take an extended time to run. The simulation was run at a rate of 1000 Hz. This rate was chosen as it matched the control frequency that was the baseline value in the FOC motor control software available from STMicro. A faster control algorithm might not be necessary, but we cannot properly replicate how unexpected operational noise might affect the entire system. The simulation also does not account for the thermal characteristics of the board or the motor. Impedance control causes higher currents to be drawn when the motor is back driven, however, we do not believe this element has to be properly simulated to showcase the efficacy of the design.

The FOC algorithm uses feedback from current sensing in the phases to control the torque of the motor. This creates a stable and controllable system, enabling external controllers to use it for position, velocity, or impedance control. On the hardware, the current sensing is handled with shunt resistors, but in Simulink, a separate current sensing block is used. The current, along with the electrical position of the rotor are fed into the Clarke Transform. The Clarke Transform takes the time domain three phase current signals and converts it into a two-phase signal in a frame that rotates with the rotor. This signal is still in an AC form, which is not easily controllable. The next step is the Park Transform which converts the two sinusoidal signals from the output of the Clarke Transform into two orthogonal DC current signals in the q and d direction. The decomposition of these two values is dependent on the electrical position of the motor, which can be easily calculated from equation 1.

$$\theta_e = \theta_m * P/2 \tag{1}$$

In this equation, $\theta_e$represents the electrical angle, θ_m represents the angle of the motor, P is the number of poles in the motor, and one electrical rotation is the change rotation needed to rotate through one pair of poles. The program requires floating point operations and a sine/cosine lookup table for proper computation of the Park, Clarke, and electrical angle calculations. The result of these steps is a DC value that can be controlled through a PI controller.

Reference values for the q and d direction currents are calculated with the equation 2.

$$T_e = \frac{3}{2}p(\lambda_{pm}i_q + (L_d - L_q)i_d i_q) \tag{2}$$

The electrical torque the motor produces is directly proportional to the currents in the d and q axis. L_d and L_q represent the inductance in each respective axis, and i_d and $i_q$are the currents in the q and d axis respectively. λ_{pm} is the permanent magnet flux linkage (or torque constant) of

the motor. For maximum torque efficiency the i_d should be zero, as current in this axis corresponds to flux weakening. The PI controller is used to maintain the constant current in the q axis as other external disturbances are introduced to the system. The output of the PI controllers can then be input into an inverse Clarke transform to reproduce the new sinusoidal signal in the rotating frame. The output of the final block needs to properly drive the electronics. Running the inverse Park transform would output three AC currents, but we need to produce PWM signals. This is achieved through space vector pulse width modulation, which outputs three pairs of pwm signals with duty cycles corresponding to the value of the AC voltage that should drive the motor at that instance of time. With this configuration, a reference torque can be set, decomposed into the corresponding q and d axis components, and controlled. Torque control can limit the output of the motor when the drumstick comes in contact with the drumhead.

Outer PID control loops can be added for position, velocity and impedance control. The encoder on the motor allows for easy determination of the values of position and velocity needed for the error measurements in the control loop. On our motor, the encoder is located before the gear box, therefore all setpoints will have to be scaled properly due to the gearboxes effect on the actual position and velocity of the output shaft. For velocity control, the angular velocity will be fed back for the error calculation. With the quadrature encoder on the motor, the velocity is calculated by measuring the time between the ticks of the encoder. The position of the motor is determined by the count of the quadrature encoder interface on the microcontroller. To reduce the run time of the simulation, velocity of the motor output was ideally measured. Impedance control can be implemented with positional feedback and a PD controller. The proportional term is directly related to the virtual spring constant, affecting the magnitude of torque the motor applies given the positional displacement. The derivative controller affects the damping of the system as it affects the change in position of the rotor. Together, these different controllers can be utilized to achieve different types of strokes.

6.0 - Evaluation

6.1 - Mechanical Design Results

While the model of the arm designed in CAD was utilized for the simulations, the team had been working on assembling the robot prior to the simulation shift. Though the robot was not fully completed physically, there were a number of things learned from the building of the arm. The bicep of the arm was fully constructed, with a belt system for the wiper motor to drive the elbow joint. Figure 37 shows the completed bicep portion of the arm.

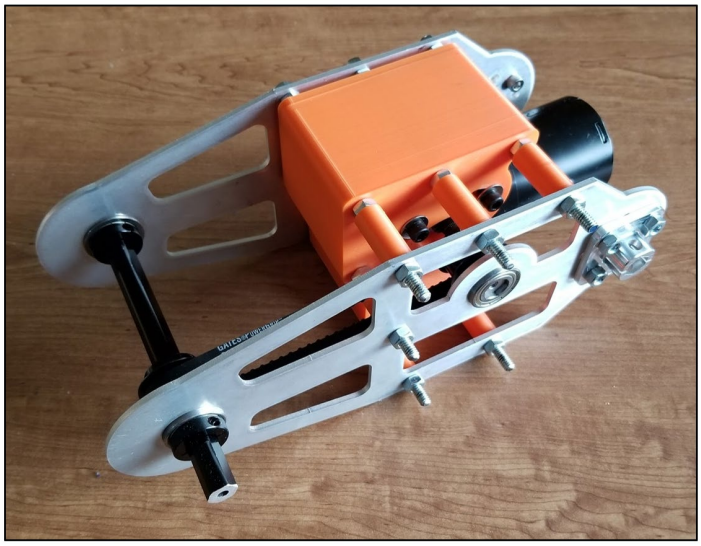


Figure 37: Bicep

With the manufactured forearm plates sent, the team also prototyped the arm with the forearm plates attached to the elbow joint and a weight close to the weight of the motor and gearbox (as the motor and gearbox had not yet arrived). In addition, the hex axles that were utilized for the robot were cut to the appropriate sizes. In order to account for the wiper motors' threaded output shafts, the axles connected to the wiper motors were threaded as well. However, this became an issue as it would allow for the motor output shaft to be unscrewed when higher torques were applied and a strain was placed on the shaft.

The base was also cut from wood stock and assembled. From the original design, two base supports were created that would provide the necessary support and stability for operation of the arm. The vertical struts would be mounted to a bottom panel that would provide the necessary stability and rigidity for the system. However, the construction of the base was found to not be perfectly sturdy due to the quality of the build. The base can be seen below in Figure 38.

Figure 38: The Completed Base for the Assembly

6.2 - Sensor Evaluation

The hit detection system was tested for five metrics: resilience to external noise, latency between strike and detection, runtime, hit detection accuracy, and volume prediction. The testing setup used the drumstick holder and IMU, with the microphone placed approximately 30 cm away from the snare drum head. To test the resilience to external noise, sound was made without moving the system. To test the IMU, the system was tested without sound. In either case, no hit was recorded. This demonstrated the success of the hit detection algorithm and verifies the necessity of a two-sensor approach to hit detection. The latency of the algorithm was tested with an oscilloscope to find the delay between the audio signal being output from the microphone, and the microcontroller toggling a digital pin high with detection. The runtime was tested by printing the internal clock counter at the end of each main loop. The accuracy was determined by monitoring the number of hits detected for a series of 10 hits at changing tempos. As this was not a machine-controlled test, the onset interval of each series of hits was determined during the testing time. For volume prediction, a relationship between the IMU, microphone, and the produced sound pressure level (SPL) was classified. A SPL sensor was placed 30 cm from the edge of the snare drum. The rotation of the bearing was actuated by hand by changing the height at which the drumstick fell from. For each hit, the microphone reading, IMU reading, and SPL sensor reading were recorded.

6.2.1 - Calibration Results

To test the calibration system, servos and potentiometers were used in place of the arm motors and magnetic rotary encoders. The potentiometer values were mapped to the servo values so that the potentiometers would act as knobs for the motor position. From there, the ToF sensors were enabled. The testing began with an object in front of the first ToF sensor. Once the sensor

reading is out of range, the servo positions were recorded as the first set point. On the actual arm, the system would move up until the distance reading was out of range. After this set point, the second ToF sensor was run until a distance of 80mm was detected. The motor positions were saved as the edge location. When the position was moved, the microcontroller would print a statement indicating that the servo was not in the recorded position. When the motor was returned to the location where the ToF sensor readings met the conditions, there would be a separate statement indicating that the motors were in a position where they would not hit the drum or had found the edge.

This system was able to successfully detect a drum and record arm positions where the robot would be clear of the drum. The ToF sensors were able to measure the distances with an accuracy of +/- two millimeters. Though not fully implemented on the final product, these tests show that the distance sensors provided reliable data in which motor positioning could be recorded. Furthermore, the algorithm is able to compare the saved set points with the motors' current state. Overall, the test proves that the robot would be able to sense a drum and save a position where it would not be at risk of knocking over the instrument. It also shows that the robot would be able to look for the drum's edge and return to this position if needed. With these sensors for positioning, we can have confidence that the system would be able to position itself well, relative to the drum.

6.2.2 - Hit Detection Results

The sensing system was tested at three different onset intervals. Ten strikes were played at each onset interval. The sensor array would output if any of the hits were detected. The results of one hit are seen in Figure 39. One cursor was placed at the beginning of the audio signal from the microphone and the other was placed at the point where the digital signal was pulled high. The latency was measured to be 19ms. This result was obtained without printing to the serial monitor, which would decrease the speed of the program. We tested a total of 10 times, with an average delay of 20ms. The duration of the pulse is due to a manual delay for easier detection on the oscilloscope.

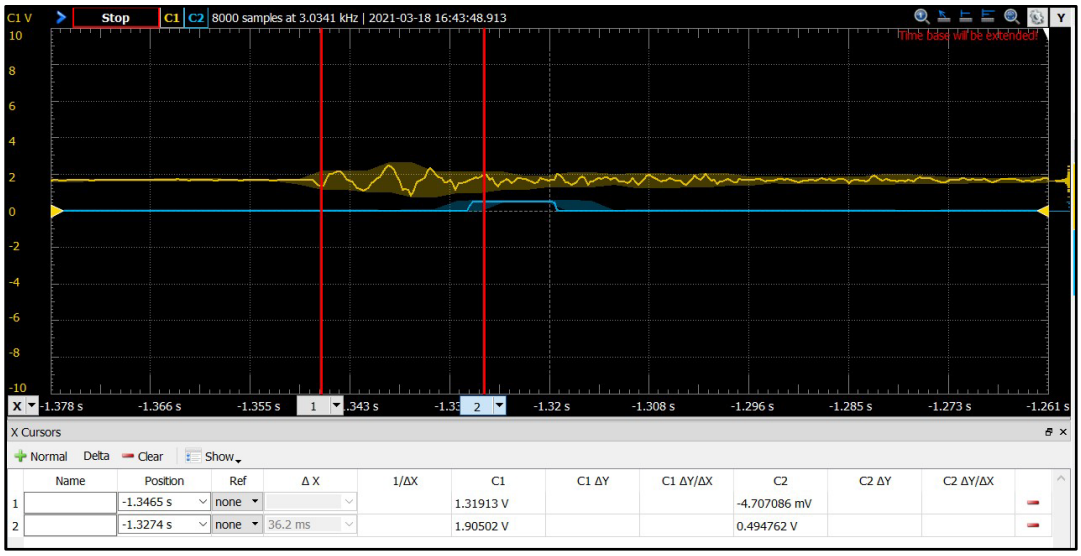


Figure 39: Latency from audio to detection

The average runtime for the algorithm was 24ms. This records data at a rate of 41.6 Hz. This rate cannot accurately represent the oscillations of an audio signal, as the algorithm must run at twice the frequency of the signal. However, we observed that the algorithm can detect the sudden change in the microphone output data due to a strike. One solution to speed up the run time is to have a board with a dedicated digital signal processing unit with faster floating-point operations. This algorithm was not tested on the Teensy microcontroller as specified in the electrical architecture diagram. The Teensy microcontroller was unresponsive, possibly due to a manufacturing error. Given time constraints, another Teensy microcontroller was not acquired for testing.

The hit detection was not perfectly accurate. We ran three trials of ten hits. The algorithm had an accuracy of 100% at 800ms intervals, 80% at 600ms intervals, and 70% at 300ms intervals. The limitations of the algorithm are the run time, which limits the rate at which hits can be detected, and the influence factor. For a drum roll, the multiple bounces are controlled internally through the FOC control and would not have to be detected for proper operation. However, as the rate of playing increases there will be added error due to the algorithm's influence factor which will then increase the threshold for a detected peak.

The results of the volume correlation test are shown in Figure 40 and 41. The microphone’s gain is minimized to prevent background noise interference and output saturation. However, the stimulating pressure waves incited the voltage output limit of the electret condenser microphone before the maximum output volume of the snare drum was reached. This is shown by the abundance of values close to 700 at the higher dBa readings. The microphone data is also inconsistent due to the operational frequency. For each hit there is no guarantee the audio is sampled at the first peak of the produced oscillation. This does not greatly affect the detection of the hit, but it affects the accuracy for which we are able to calculate the produced volume. A delay in sampling the signal results in a lower measured magnitude, as the amplitude

of the oscillations decreases after the initial stimulus. A trend can be fit to the IMU data as there is a noticeable increase in the magnitude of acceleration as the produced sound increases in volume. Faster hardware would be able to run the algorithm quicker to ensure better operating conditions. The peak detection algorithm code can be found in Appendix E.

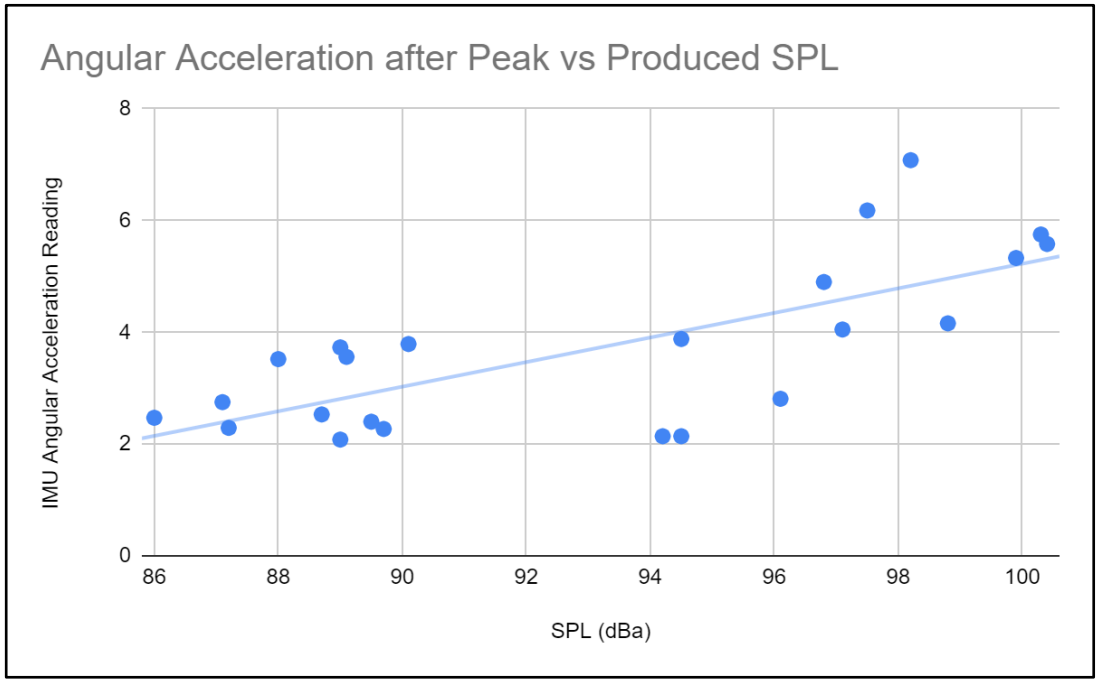


Figure 40: IMU vs Produced Sound Graph

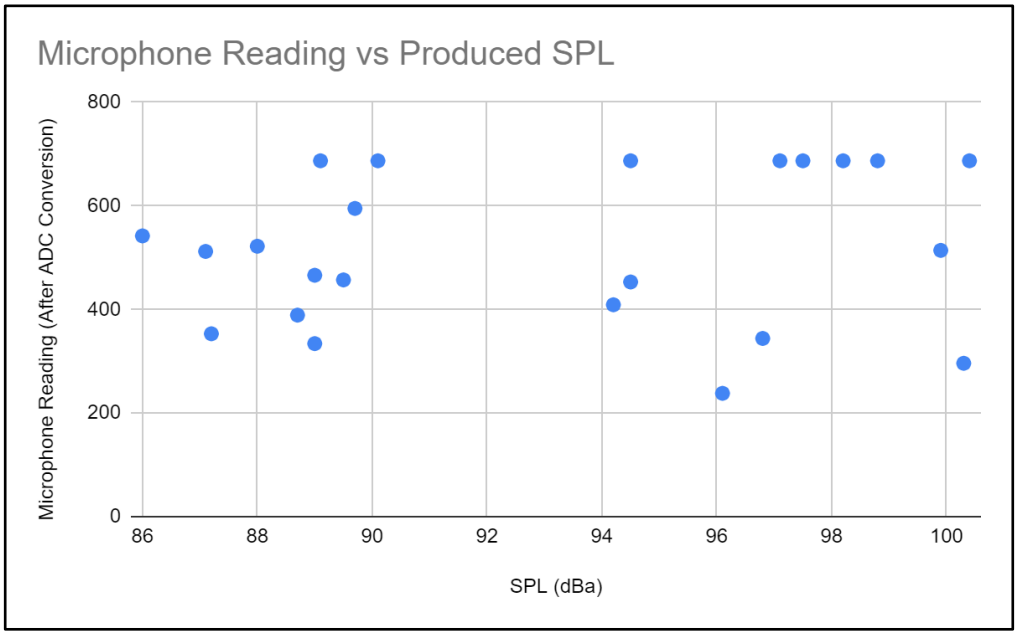


Figure 41: Microphone Reading vs Produced Sound Graph

6.3 - Simulation Evaluation

6.3.1 - Dynamic Control of Shoulder and Forearm Links Results

Based on our requirements, we first tested the oscillation of the forearm with an angular displacement of +/- 25° while playing with a period of 600ms, or a tempo of 100 BPM. Then, we tested this oscillation with a period of 400ms (a tempo of 150 BPM), 300ms (a tempo of 200 BPM) and 1200ms (a tempo of 50 BPM) to demonstrate how the motor's performance and positional control changes with increased tempos. These were input as trajectories to the motors and fed to the joints.

Once the trajectories were applied to the shoulder and forearm, graphs of the position data were collected. This demonstrates the accuracy of the position control of the motor in relation to frequency to demonstrate the arm's capacity to play at a period of 600 ms (the desired tempo of 100 BPM) with oscillations of +/- 25°. A base test was run with the forearm moving at this onset interval and specified oscillation, with the bicep being driven for the first second of the simulation, moving from an angular position of 0-90°. From this trajectory the following position graphs were produced.

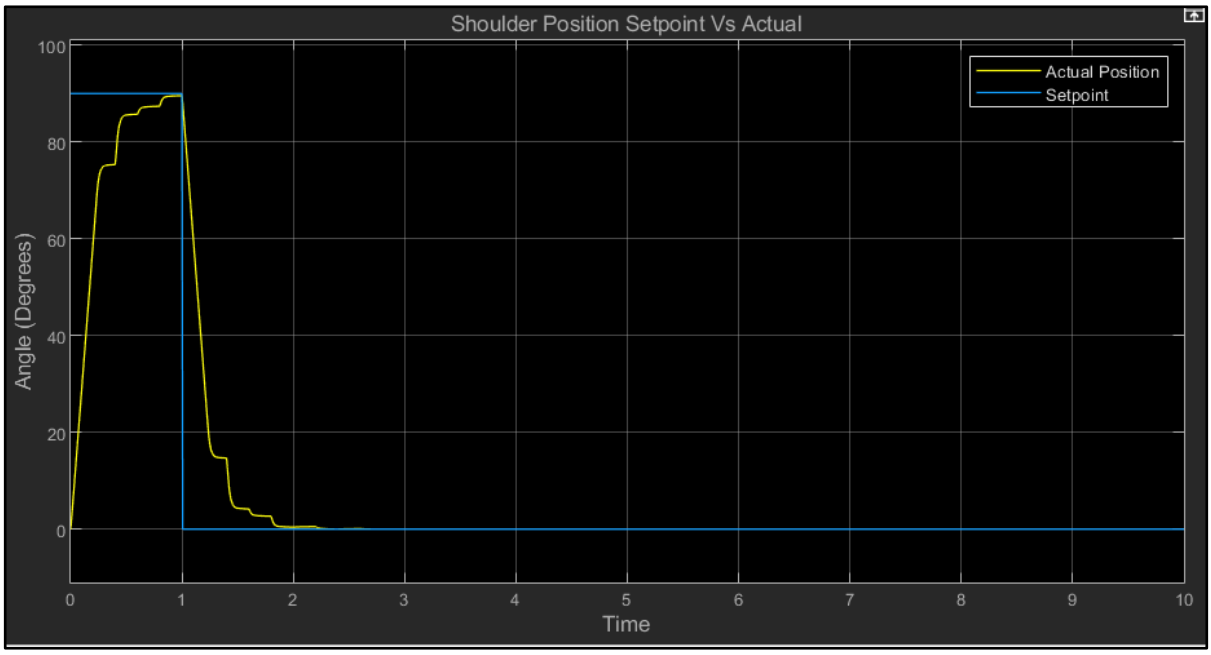


Figure 42: Shoulder Position Planned vs Actual Trajectory

As can be seen in Figure 42, the shoulder was actuated, starting from the base position with the motion from 0° to 90° being input as a square wave with a pulse of one second. During this period, it takes the bicep a second to reach the desired position, and another second to return to the base position of 0°, demonstrating the speed of the bicep within its playing range of 0° to 90°. Though this does not represent an ideal trajectory during playing, it was utilized to demonstrate the range the shoulder would be actuated during play. This trajectory would remain

the same for the shoulder for all the other testing of the forearm, as to evaluate the trajectories, our requirements were focused on the movement of the elbow actuator.

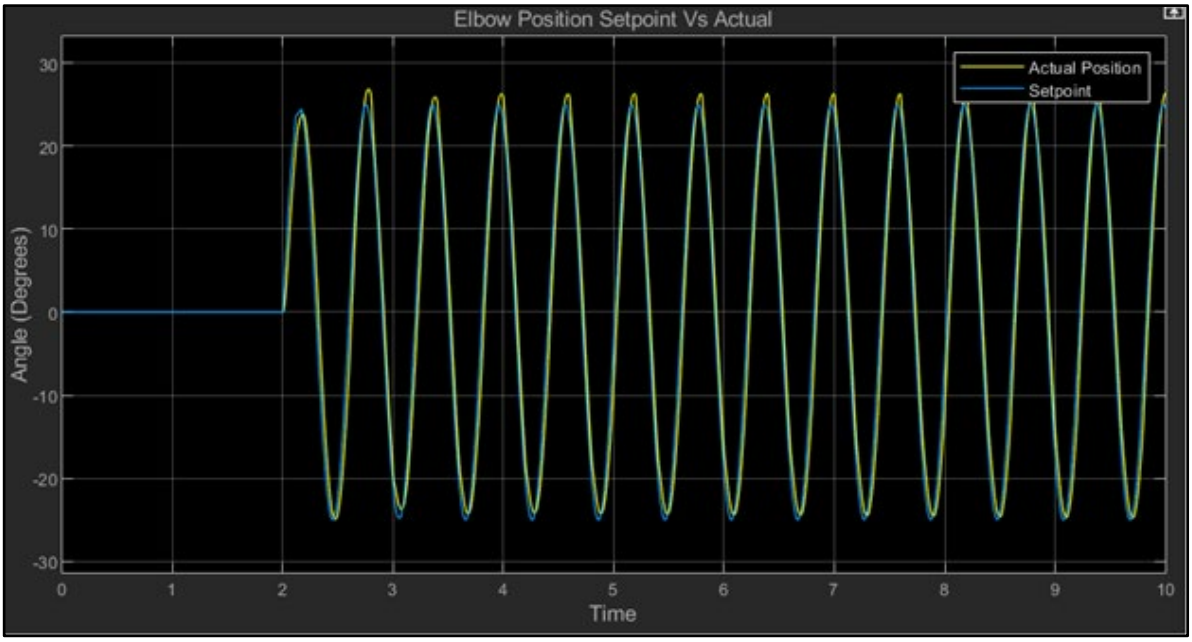


Figure 43: Elbow Position at Period of 600ms (100 BPM)

The elbow actuator was first tested at the tempo of a period of 600ms (100 BPM), as seen above in Figure 43. Here, we can see that the actual position of the arm and the set points provided by the trajectory almost entirely align throughout play. This demonstrated that at this standard playing tempo, the arm could play at our desired speed and with the desired oscillation range without major deviation.

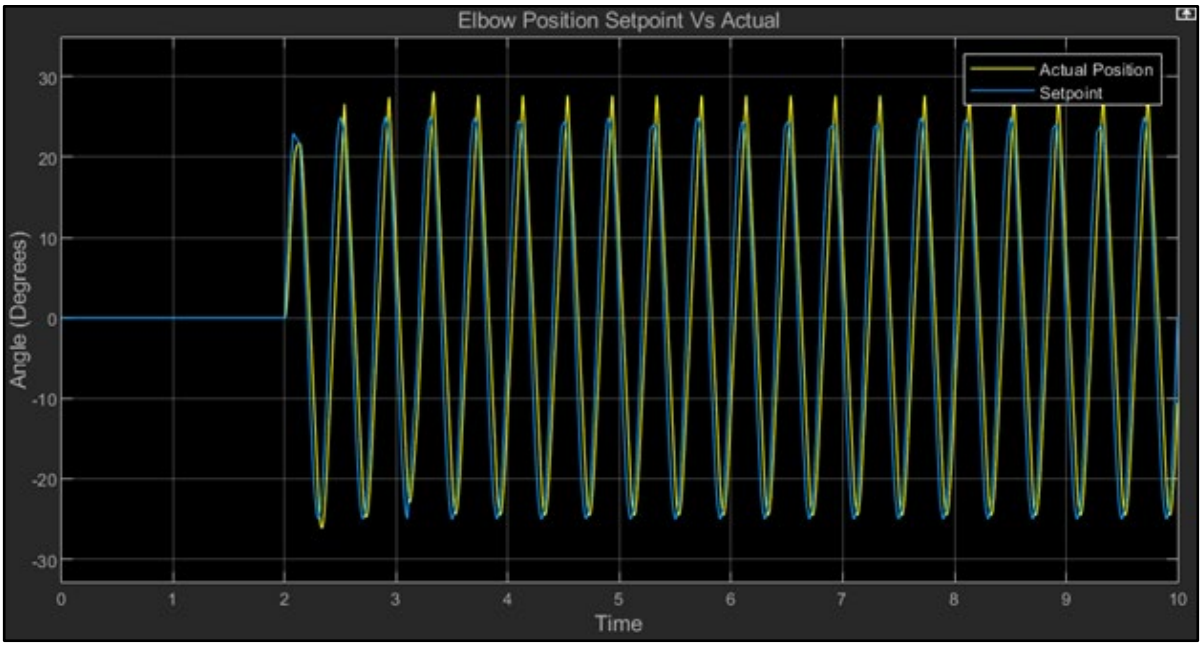


Figure 44: Elbow Position at Period of 400ms (150 BPM)

Following the base testing at standard tempo, two tests were done using elevated tempos. The first was a test at a period of 400ms (150 BPM) as shown in Figure 44. Here, we can see that the discrepancy between the actual elbow position and setpoint values begins to increase marginally, with each movement toward +25° surpassing the desired angular position by between two and four degrees each time, as well as a slight shift of phase.

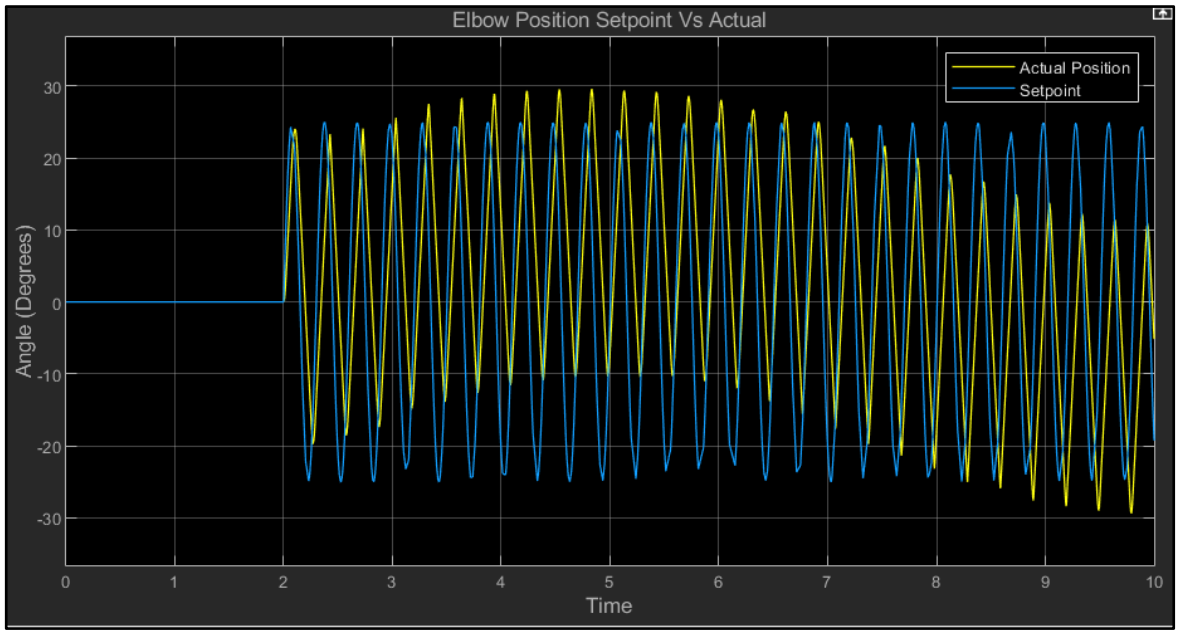


Figure 45: Elbow Position at a Period of 300ms (200 BPM)

This behavior only becomes worse as the frequency increases, with an additional test at a period of 300ms revealing major deviations in the actual and desired positions of the arm. There

is a larger oscillatory pattern to the movement of the forearm, with an undesired phase shift as well. Here, the inability for major oscillations at high frequencies is shown by the poor performance of the elbow actuator at this tempo. To further verify this behavior, a test was done at a rate slower than our standard of 600ms intervals.

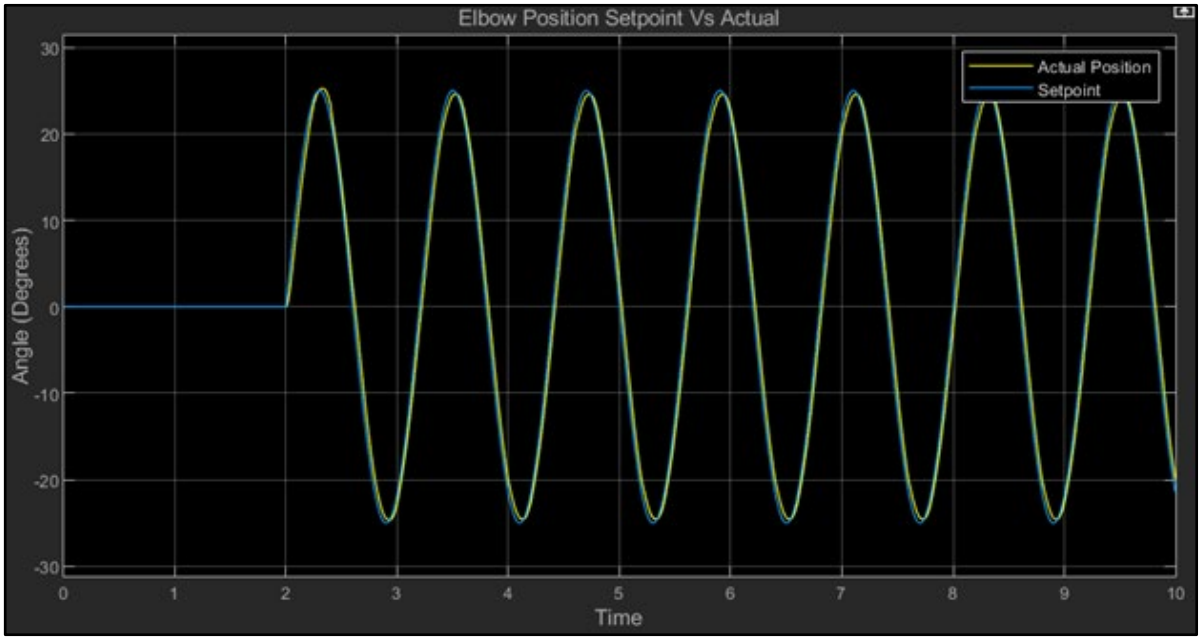


Figure 46: Elbow Position at a period of 1200ms (50 BPM)

The final test of the forearm was at a period of 1200ms (50 BPM) which demonstrated an exact match between the actual and desired position during play, better than the results of the forearm movement at the standard tempo of 100 BPM. This can be seen above in Figure 46. This demonstrates the arm's ability to play and perform standard visual gestures such as oscillations of the elbow and anticipatory gestures at slow and moderate tempos. However, the arm would reach undesired playing behavior once a period of 400ms (150 BPM) is crossed. The loads on the system and inertial properties create play that is out of phase, and positionally inaccurate by wide margins, as demonstrated in Figure 45.

6.3.2 - Wrist Actuator Testing Results

To test the effectiveness of each controller for certain stroke types, we conducted four tests: frequency response, impedance-controlled oscillation, torque-controlled hits, and velocity rise time analysis. The frequency response testing was performed in both the position and velocity domain. The strike of a drum hit should have its max velocity at the moment of impact, so a sine wave trajectory would not make sense. Following the diagram from Figure 1, the trajectory within close proximity of the strike is an approximate triangular shape. Perfect replication of this trajectory would be impossible because the velocity would have to reverse instantaneously. The motor will respond according to its physical limits. The angular displacement of the stick was measured using a triangle wave trajectory with an amplitude of +/-

45° at frequencies of 10, 5, 2.5, and 1.33 Hz. The reference trajectories should result in an impact of the drumstick and surface. However, the stick will ‘air drum’ if the displacement is not high enough. A similar test was conducted controlling the velocity tracking. The reference trajectory was square wave (representing the derivative of the triangle wave) reproduced at the same frequencies as described previously. The results of this test will show the ideal rate at which the motor can operate and how different control types affect the trajectory tracking of the wrist system.

The impedance-controlled oscillation test focused on parameter tuning and time response of a drumroll-like oscillation due when an impedance controller is used for the wrist. This is conducted by sending a setpoint to the motor that is slightly below the surface of the drumhead. When contact is made, the forces of impact should affect the system given the control constants are set properly. The trajectory of motion for the stick, torques, and oscillation frequency are measured to determine the efficacy of this design for replicating flams, drags, and other drum roll like gestures.

Actuating the wrist using only torque control has the benefit of stopping the stick when a maximum torque threshold is reached while interacting with a contact surface. This test demonstrates whether the motor can be actuated with simple torque control to recreate a hit. The motor should respond at the rate of its control loop, but it would need the external hit detection algorithm to create a signal to send the next set point. The motor control was tested at four points across the range of its nominal torque output. The velocity of the stick and response time of the motor were measured to determine the rate of playing with this control methodology.

The final test involves classifying the time needed for the motor to accelerate to the angular velocities of 35, 90, 146, 201, 257, and 313 rpm. These are six linearly spaced points within the range of values described in the design requirements. This test will involve no contact, and the rise time will be defined as the time between the motor to go from 10% to 90% of the desired speed. This is a standard definition for rise time in control systems. The rise times will determine the max dynamics range that can be achieved at different playing rates. These tests were performed first on the Maxon motor originally chosen for the project.

The results from using the original motor specification that were determined at the beginning of the project were undesirable. The motor was unable to produce the requisite velocities to reproduce the range of human torque output, with its gearbox significantly slowing down the performance of the motor. With the gearbox of 43:1 a speed of ~13,000 rpm would have been needed to achieve a speed of nine m/s. The gearbox also reduced torque transparency; a higher incident torque on the rotor is needed for the motor to sense that it has reached an obstacle. This led to a long contact time of 175ms which represents an extended time of the motor trying to drive the stick tip into the drumhead.

Given the focus on simulations, the motor requirements were reanalyzed to choose a motor that better met our requirements and would produce more meaningful results. The motor parameters are taken from the Maxon EC-60 608150. This motor is constructed and controlled in the same ways as our other motor. The motor is lighter, faster, but able to produce less torque. The gearbox on the motor was chosen to be 12:1. This allows for a greater torque transparency, producing better impedance control performance.

Table 1 contains the data taken from the frequency response testing. With each frequency having the same setpoint, the amplitude of oscillations rapidly decreases. The phase decreases as frequency increases due to the lower accelerations needed to change directions. Figure 47 and 48 showcase the difference between the velocity and position setpoints. Due to the orientation of the frame, the collision between the stick and the surface occurs at a positive position. For this setup, those hits occur at approximately eight degrees. The trajectory following is erratic due to the unexpected hit and integral gain on the PID controller. The velocity-controlled trajectories follow a more consistent trajectory likely due to the motor not decelerating as it reaches its positional setpoint. These results show that velocity control is better suited for rapid onsets. As the frequency increased, the motor oscillations still occurred, but the drumstick did not come close to hitting the surface. Additional control parameters would need to be in place to ensure the arm was located in a position that allowed the stick to make contact. All produced graphs not highlighted in this section can be found in Appendix D.

Table 1: Frequency Response Testing Results

	Position Controlled		Velocity Controlled	
Frequency (onset)	Amplitude (degrees)	Phase (ms)	Amplitude (degrees)	Phase (ms)
1.33 Hz (750ms)	15	-55	7	48
2.5 Hz (400ms)	7	100	5	37
5 Hz (200ms)	2.4	76	1	33
10 Hz (100ms)	1	27	0.15	24

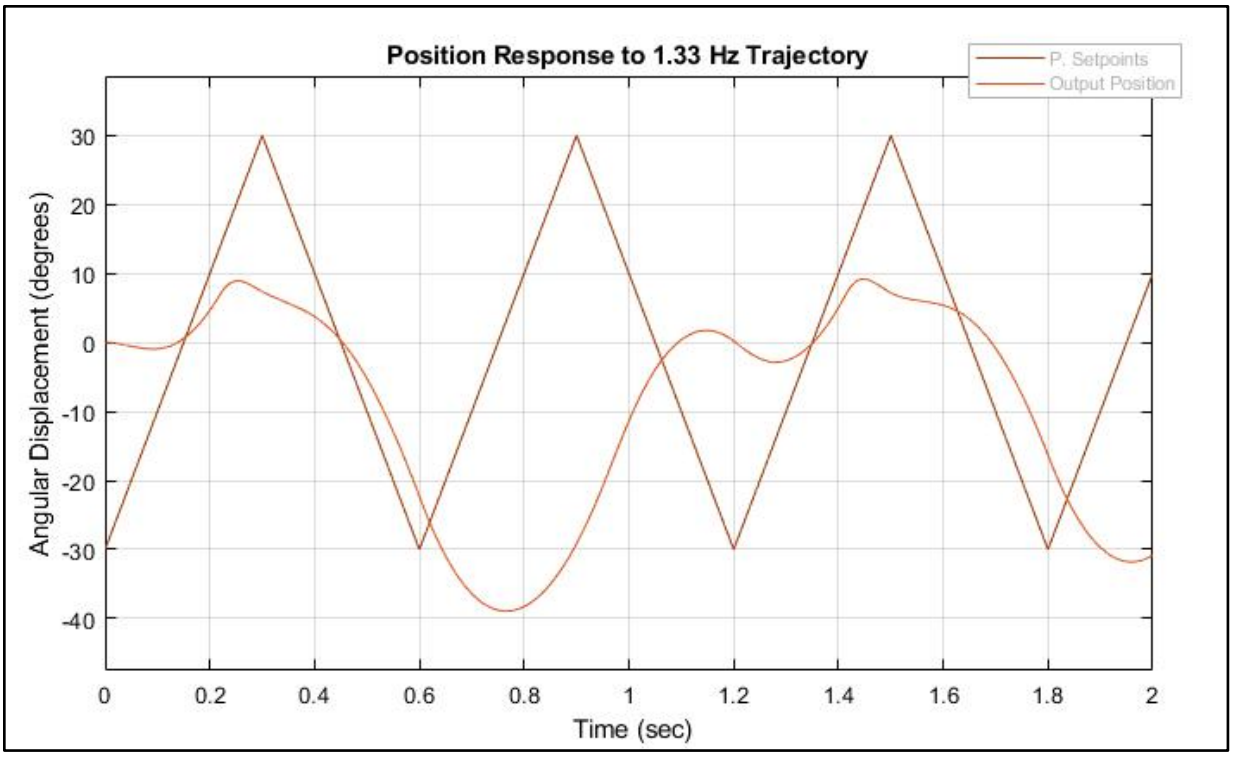


Figure 47: 1.33 Hz Frequency Response to Positional Trajectory

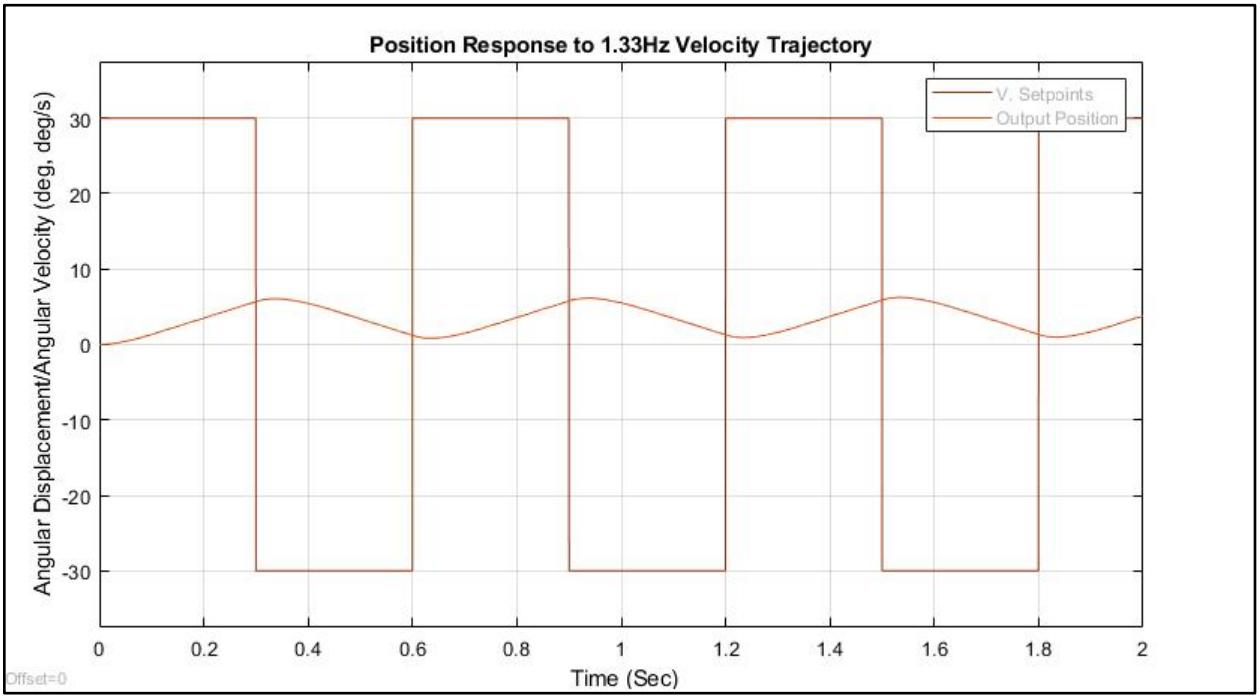


Figure 48: 1.33 Hz Frequency Response to Velocity Trajectory

Figure 49 and 50 shows the results of two different impedance-controlled hits of the surface. Between each hit, the proportional and derivative controller are increased by a magnitude of ten. Each trajectory consisted of a single period of a square wave representing an upstroke and downstroke. The scale for the angular displacement field was changed to be the arc length of the drumstick's motion. Once again, the hit occurs at a positive displacement. With the impedance control, the damped oscillation trajectory is easily achieved. However, the time scale of the events is not on the same scale as a human controlled drum roll. The average time for each impulse was 75.5 ms. Each concurrent hit has a longer impulse due to the reduced velocity of the stick. The minimum impulse was 67ms and the maximum was 88ms. The delay between the start of the simulation and the first hit is due to the small proportional gain. A complex motion is able to be achieved with a simple setpoint, allowing for more flexibility for altering the response.

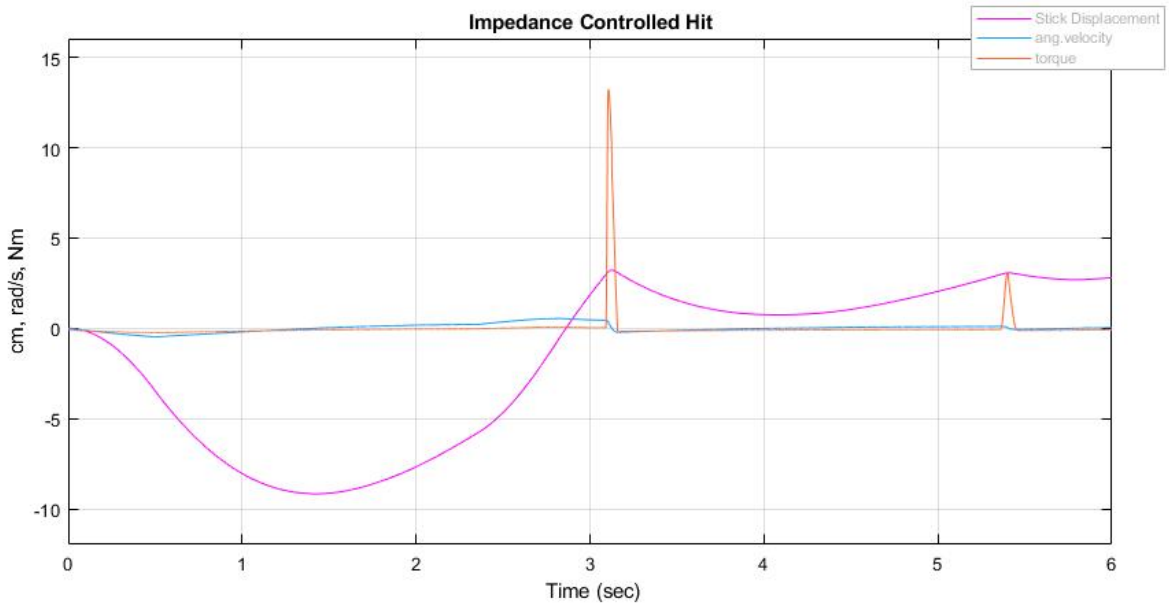


Figure 49: Impedance Controlled hit with nominal stiffness and damping

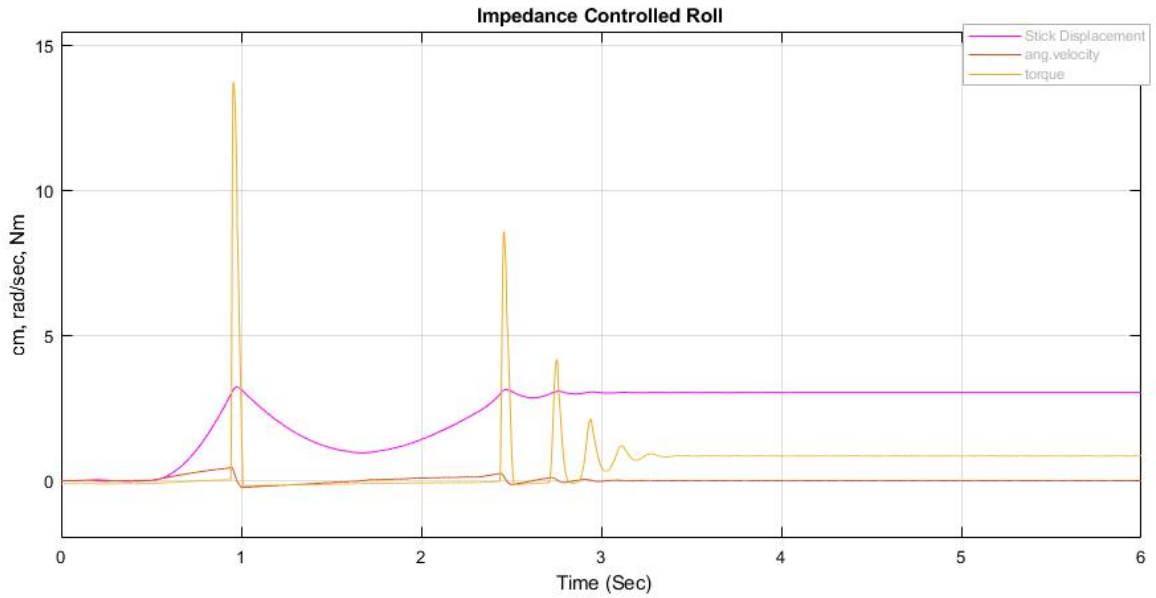


Figure 50: Impedance Controlled hit with higher stiffness and damping

Figure 51 shows the result of a strictly torque-controlled hit using a reference torque as an input. The input torque was set from points between 0.1-3 Nm. This covers the range between the nominal torque of the motor and past its stall condition. At the lower ranges, 0.1-0.4 Nm, there was no noticeable difference in performance, and those graphs were excluded. The greatest difference between each graph is the amount of torque required for the motor to be back driven. These results are different than what was predicted, as the motor was thought to just slow down as a torque was applied. This method of control is undesirable, as there is no direct control of the damping factor.

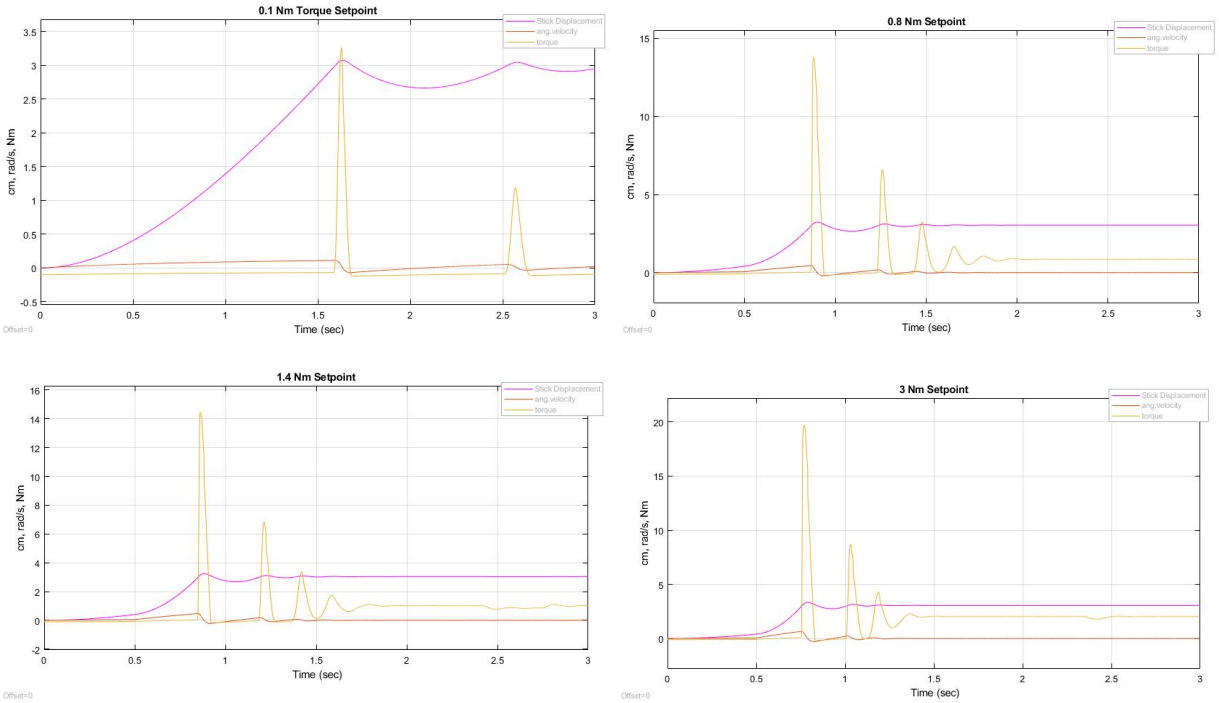


Figure 51: Position, Angular Velocity, and Torque applied to the shaft for Torque Controlled Setpoints

Table 2 shows the rise times for the different required velocity setpoints. Figure 52 shows the velocity response for the first run of the test. The chosen motor is unable to accelerate to all the required speeds. The simulations ran for five seconds, and the motor was unable to reach the setpoints for 5.8, 7.4, and 9 m/s. This is likely due to the load on the motor being greater than the nominal value at the required speed. In addition, it is possible that the simulation parameters for the q and d axis currents are not accurate, given that they are not directly specified in the data sheet. The values in the simulation were based on the values for the motor we physically had.

Table 2: Velocity Rise Times

Angular Velocity (RPM)	Associated Linear Velocity (m/s)	Rise Time (ms)
35	1	336
90	2.6	589
146	4.2	2730
201	5.8	> 5000
257	7.4	> 5000
313	9	> 5000

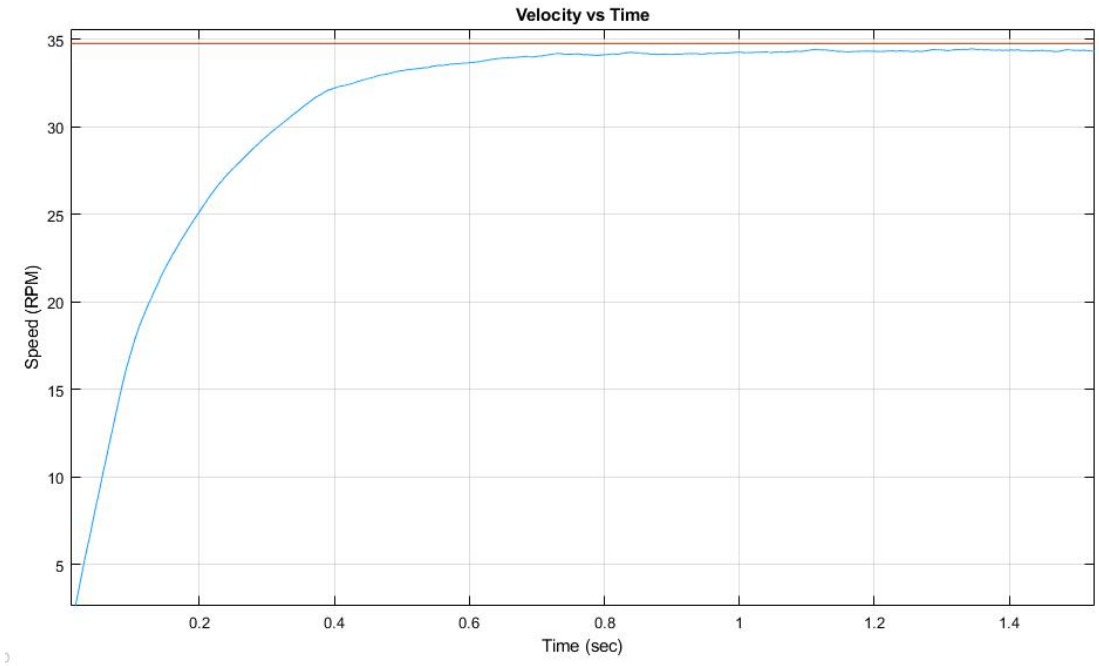


Figure 52: Velocity vs Time graph for 35 RPM setpoint.

7.0 - Discussion

To evaluate the system, we compared the data from our simulation and calculations to the objectives discussed in section 5.1. While our testing and result show that we did not satisfy every design requirement, we did meet our larger objectives. In this section, we will discuss each requirement in detail.

7.1 - Requirements 1 & 2: Strike Parameters

The actuator initially chosen for the wrist was not suited for the desired application. Our initial analysis for motor requirements focused on replicating the torque of the human wrist. As our understanding of the problem evolved, we realized that the output velocity should have been the focus of the motor choice. The gear box on the motor allowed for a high torque output, but reduced torque transparency to the motor. The gearbox was not removable, and so the motor was less flexibly integrated that we had originally planned. With the need for simulated results, we took the opportunity to choose a similar, but better suited motor for the design requirements. With this motor in simulation, we were able to more closely model the type of interactions we wanted to achieve.

The wrist actuator was unable to achieve the full velocity range, (1-9 m/s) of a human player. The motor was able to achieve velocities in the lower range up to linear velocity at the stick tip of 4.2 m/s. Past that point, the motor was unable to accelerate higher. In addition, the longer velocity rise times result in a large angular displacement that would likely cause a collision with the drum before reaching the target velocity. Without the physical testing, we cannot verify ourselves the volume these velocities map to. However, the range of 1-4.2 m/s according to (Dahl, 2011) creates an approximate range of 10-42 N, demonstrating a smaller dynamic range.

The requirement for a six millisecond impulse time was based on motion studies of the stick position while playing. The human ‘hand to stick’ interface allows for the forces of the drumhead to instantly affect the stick. The mechanical representation of this system, which included the motor, gearbox, and electrical control, greatly slowed down the response time of these forces. In the simulation, the impulse time was approximately 75ms. This is an order of magnitude greater than the minimum impulse time of a human player. However, the impulse time for the motor was a natural response of a simple control scheme, that can be improved through the choice of BLDC motors that are more easily back driven. The current impulse time still allows for drum playing, and it is a step forward in a system that replicates both complex and simple human-like drum motions.

7.2 - Requirement 3: Horizontal Strike Range

As demonstrated by the movement range of the digital model, the robot could meet the strike range of 18 cm, with an actuation of the shoulder allowing for a horizontal range of 18.3 cm in which the robot can play by keeping the stick at an angle of less than 90 degrees. This range would allow for the robot to be able to move between the middle of the drum and the edge closest to the robot during play. This provides desired timbral changes through the differing placement of the drumstick. The horizontal strike range is illustrated in Figure 53.

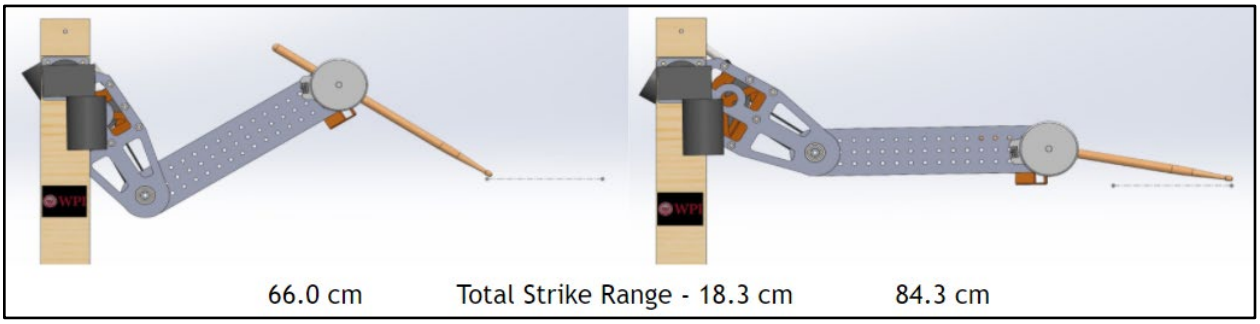


Figure 53: Strike Range of Drumming Arm

7.3 - Requirement 4 & 5: Onset Interval Characteristics

To match the upper limit of human abilities for drum playing, the system required a max onset interval of 100ms (10Hz), with a temporal deviation of less than 50ms to avoid a noticeable decline of synchronization. The frequency testing of the motor showed that the motor was able to produce a marginal response at that frequency. The stick moved with a one-degree amplitude movement, which relates to a 12 degree amplitude motion without a gear box. In reality, frictional factors within the gearbox might prevent this type of motion. The larger issue with the frequency response testing is that the oscillatory movement does not make contact with the drum. This could be resolved by setting a resting point closer to the drum for the faster oscillations. Additionally, instead of holding the stick slighting above the drum, the stick could rest on the drum before playing. These methods would require predetermined motion planning to ensure optimal positioning for a given playing frequency.

The phase of the motion tracking varied with the onset interval. At higher rates, the motor and stick have less time to accelerate, making it easier for the motor to generate the required torque to reverse the motor. Figures 47 and 48 show the 1.33 Hz response, and the rotation of the stick does not follow the expected oscillatory motion. This is due to the effect of a hit on the trajectory following. From this analysis, it is evident that at longer onset intervals, each hit should be treated as an individual event, where the stick returns to a known height. For shorter onset intervals, the trajectory needs to be altered because it would be more difficult to properly sense each hit.

7.4 - Requirement 6: Expressive Motions Relating to Onset Interval

Our simulations met our goal of a decrease displacement as onset interval decreased. The system can play at our set goal of oscillatory motion of the forearm with a displacement of +/- 25° at a period of 600ms. Further tests demonstrate the need for reduced oscillations of the forearm at higher tempos. This was an important consideration, as it gave us a range at which the forearm could be utilized to enhance the visual and musical expressivity of the performance. As corroborated by our research on human players, as the rate of playing increases, the arm must reduce its oscillations, and instead produce the requisite sound from the movement of the wrist.

7.5 - Requirement 7: Damped Oscillation

The wrist actuator was found to be able to produce the damped oscillations necessary to play a double stroke roll. This was a vital component of the evaluation, as it demonstrated that the system had the capability to produce this complex motion through FOC. This demonstrates the value of the active impedance control implementation of a VIA, as it allows for the reproduction of complex motions such as drum rolls, which standard actuators are unable to perform, opening up a wide range of possibilities for the robotic drummer to play with increased fidelity to human performance. The simulation does not account for oscillations on every type of surface, but it stands as a proof of concept and a starting point for research to find the BLDC motor best suited for this application.

7.6 - Requirement 8: Sensing

Finally, the testing of the sensors in conjunction with drum hits demonstrated moderately reliable behavior in classifying drum hits. The system demonstrated that for low intervals of 830ms the robot detected all ten hits. At a more rapid interval of 600ms, this accuracy fell to a recorded 8 of 10 hits. Finally, at a high interval of 300ms the system recorded 7 out of 10 hits. In addition, the individual components were also tested.

7.7 - Design Objectives

Though our system did not meet all our design requirements, we feel it did still meet the design objectives for our robot. Our first objective was to design a modular system which could play on a variety of instruments. The robot's design with its movable base and adjustable playing height through control of the shoulder and elbow actuators allows for the robot to play at a variety of heights and distances from different drums. Additionally, its modular drumstick holder which can be adjusted for drumsticks of different sizes and types allows for the robot to play on a variety of instruments, achieving our first design objective.

The robot also met our second design objective of being able to play both single and double stroke rolls. This was a crucial component of our design as it differentiates our system from many of the other percussive robots that have been implemented before, as it is able to

produce the oscillatory motion necessary to play a drum roll. These two components, of a strike and roll act as the basic building blocks of percussive performance. This was one of the core functions of the VIA at the wrist actuator, and the robot's ability to perform both types of hits gives it the potential for complex musical performance.

Our final object was to demonstrate motions associated with visual expressivity and the potential for aural musical variety. The anthropomorphic inspired design of the arm, along with demonstrated dynamic control of the shoulder, elbow and wrist actuators shows that our robot has the capability to achieve both aural and visual expressivity. Through actuation of the shoulder and elbow motors, anticipatory gestures as well as standard human playing motions such as oscillations of the forearm can be reproduced while playing. This, along with the strike range of 18.3 cm, and the ability to change the velocity of the strikes of the wrist actuator and play at speeds of up to 10 Hz, demonstrates the robot's ability to change dynamics while playing along with its timbre.

8.0 - Conclusion

Ultimately, despite being unable to physically complete the robot due to a number of setbacks, the robot's design and simulation showcases the potential capabilities of the arm to be a robotic percussion player with higher fidelity to both the aural and visual characteristics of human players, compared to many previous percussive robots. While our efforts fell short of having the robot be able to replicate the wide range of human playing, the system has been designed and implemented in a way that makes it a modular player that can produce strikes and rolls and demonstrate aural and visual expressivity through anticipatory gestures, mimicking human motion and dynamic and timbral changes.

Additionally, the novel use of a VIA in a musical robotics system demonstrated effective results, with our wrist motor allowing for the replication of the damped oscillations necessary to produce a roll, with FOC allowing for rolls to be created across many different surfaces. The use of VIAs to reproduce more complex motion seems to be a viable and effective strategy for robotic percussion players. This could open up many different avenues in the continued expansion of musical robotics. The active impedance control implementation of a VIA used for this project demonstrates a wider capability for BLDC motors with FOC as an appropriate actuator for replicating human motion and performance.

The field of musical robotics is constantly expanding, and the integration of technologies from other burgeoning fields of robotics such as prosthetics can help the design of robots based around human performance become more accurate in their recreations of human capabilities. The work done in this project has helped lay a foundation for continued work and expanded research into the use of VIAs for replicating human capabilities at WPI as well as demonstrated its potential in the field of musical robotics.

9.0 - Future Considerations

9.1 Future Mechanical Considerations

Through completion of the arm design, implementation through digital assembly, and torque analysis performed through Simulink, a number of areas for improvement were identified. The most critical is the wrist actuator. Testing showed that the Maxon motor we purchased for use as the VIA had too low of a speed output as a consequence of the fixed gearbox. This error was a result of two considerations. The first being was our focus on the range of stick force outputs of a human player (2-100 N) for our motor. This coupled with our initial goal of reaching speeds that the fastest human player could reach (10 Hz onset interval) caused us to gravitate to the Maxon motor for use in the project. The second was our initial understanding that the gearbox would be removable. When the motor and gearbox arrived, they were manufactured as one part. It became clear through our analysis of the motor that the motor output with the gearbox attached could not meet the updated goals we had set for the velocity of the end effector during play.

As such, we believe a different brushless DC motor, implemented with FOC, could better act as the wrist component of our robot. After revisiting at our mechanical requirements with a better understanding of the Maxon motor, we believe an output speed of 34.76-312.55 rpm given a stick radius of 27.5 cm would be necessary for play. Additionally, a static torque of 78 mNm and a dynamic torque of 3.5 Nm (which was determined as the torque needed to accelerate the maximum speed at an onset interval of 10 Hz). Upon reviewing the available Maxon motors, a good replacement for the current wrist actuator would be the Maxon EC-60 608150, which has a nominal speed of 4510 rpm, nominal torque of 146 mNm and stall torque of 1.26 Nm. Implementation with a Maxon gearbox of 15:1 should more closely meet the mechanical requirements for the system. Additionally, the wrist actuator should be moved back, to the elbow joint and actuated through a belt (similar to the current system for the elbow actuator). It became apparent when built that the extra weight of the gearbox as well as the length of the forearm segment increased the torque requirements for the system unnecessarily.

The current Maxon motor would still be useful to the system, as it would provide a strong alternative to the current elbow actuator. The maxon motor could provide the requisite torque and velocity for play at the elbow and allow for greater musical expressivity and performance with a VIA at the elbow, to compliment and assist the output of the VIA at the wrist.

There were a number of other mechanical considerations that arose during the assembly of the arm. Another issue was the output of the elbow and shoulder actuators. Both motors have a simple threaded output shaft. This was a hindrance, as it meant the axles for the system had to be threaded for these two actuators. This became an issue as high torques were applied to these joints. Under these conditions, the threaded shaft tended to loosen over time. One potential solution utilizes adhesives within the axle threads to bind the shaft to the motor output. This

solution is not ideal as it secures those parts in a manner which makes modifications or changes to the system difficult and ungainly. As such, replacing these actuators would be advised. The current wrist motor could be made the elbow motor as described above, and a new motor would likely need to be chosen for the shoulder.

Another consideration for future development is the mass of the system. In the current model, the lengths of the bicep and forearm were determined by measuring the human arm. As development proceeded, it became apparent that the applied torque to the shoulder motor was a significant amount. Our solution to this issue was to implement gravity compensation in the form of a spring mounted above the bicep. However, even with this solution implemented, it may be advantageous to reduce the length of the forearm to reduce the mass of the system and the subsequent load the shoulder motor must actuate.

Finally, the current design contains a split axle that secures the arm between the two base struts. This design was implemented as it was reasoned that this would allow for an orientation of the elbow actuator that would be most beneficial to the arm's center of mass. However, given the changes suggested, as well as seeing this design realized, we would likely change it so that there is a single axle at the top of the arm upon which the arm could rotate freely, as this would be more structurally secure.

9.2 Future Electrical Considerations

From our analysis of the design, we have identified three main areas of improvement for better operation of the drumming arm in future iterations. These improvements are higher fidelity sensors, more robust 12V power supply, CAN (Controller Area Network) communication between distributed controllers. The majority of the sensors purchased for this project were acquired from Adafruit for their affordable price point. However, the IMU provided noisy results, and the microphone and amplifier combination had an output different from the Teensy audio library's standard values. Given the results of the hit detection algorithm, it is worthwhile to invest in an interface between a standard microphone and amplifier and the chosen microcontroller. In addition, we recommend switching the potentiometers to magnetic rotation encoders. The potentiometers are absolute encoders; however, they have a limited range of motion. For the motion of the arm, neither joint's motion exceeds the rotational limits of the potentiometer, but mounting the potentiometer properly was difficult. The magnetic encoder detects the rotation of a magnetic field. This system can be implemented by placing a magnet on the point of rotation and mounting the sensor above that position. This should allow for the same quality of positional data and simplify the process for making any adjustments to the construction of the arm.

The power supply for the 12V wiper motors was chosen for high rated current and price. However, further inspection on the quality of product from that distributor shows the power is not likely to last long. The oscillatory motion of the wiper motors requires high current spikes

when the motor is momentarily stalled. A power supply similar to our choice of a linear, unregulated power supply for the BLDC would be a stronger option. Another option, and an interesting power distribution problem, would be the implementation of a battery-based system that automatically recharges. The batteries can handle high current spikes and are widely available.

The system in total would benefit from further architecture of a distributed system. The dedicated system for DSP with the microphone and IMU reduces computational strain. CAN is a fast and reliable communications protocol that requires only two wires of a single bus. With this wiring, there will only need to be one data line to connect every module. The protocol also lets any module send messages to any number of other modules at the same time, allowing for better synchronization. The STMicro Nucleo boards are equipped with interfaces for the hardware needed for this type of protocol. Adapting this communications method will allow for more modules to easily be added if needed, while still allowing for large bandwidth for information.

9.3 Future Software Considerations

The reliability of hit detection can be improved with the implementation of a more consistent and robust peak detection algorithm. One major improvement would be the analysis of a median value, rather than the mean. This change would focus on a more central tendency than the general average. Given that the range of numbers can vary based on the microphone gain, finding this general trend will result in a better algorithm. On a similar note, using a robust measure of scale would increase the accuracy of the peak detection software. Rather than analyzing the standard deviation, it would be more consistent to use a median absolute deviation. In general, focusing on the median would produce better results. Furthermore, streamlining of code would increase efficiency.

The complexity of the system does not allow for simple real time MIDI control. Once the full system is built, a software interface could be created that converts an input phrase of music into the necessary trajectory points for the arm to follow. A machine learning algorithm could analyze percussion performances to learn the players motions and translate them into trajectories for the arm. This would create a more autonomous system that does not rely on hand programmed motions.

The calibration process can be improved. Ideally, the sensors should be able to identify the center of the drum. This would involve recording the first motor position when an edge is found and calculating the distance traveled until the other edge of the drum is found. This process could also be done with camera vision, though implementation of this would require heavy processing power and the mounting of a camera system on the arm. This would make the calibration more autonomous, for the current system requires the center of the drum to be placed in front of the arm. With a vision system, the robotic system can have constant calibration tracking.

References

Bartlette, C., Headlam, D., Bocko, M., & Velikic, G. (2006). Effect of network latency on interactive musical performance. Music Perception: An Interdisciplinary Journal, 24(1), 49-62. doi:10.1525/mp.2006.24.1.49

B. Katz, J. D. Carlo, & S. Kim. (2019). Mini cheetah: A platform for pushing the limits of dynamic quadruped control. Paper presented at the 2019 International Conference on Robotics and Automation (ICRA), 6295-6301. doi:10.1109/ICRA.2019.8793865

Bretan, M., Gopinath, D., Mullins, P., & Weinberg, G. (2016). A robotic prosthesis for an amputee drummer. arXiv Preprint arXiv:1612.04391,

Broughton, M., & Stevens, C. (2009). Music, movement and marimba: An investigation of the role of movement and gesture in communicating musical expression to an audience. Psychology of Music, 37(2), 137-153. doi:10.1177/0305735608094511

C. Everarts, B. Dehez, & R. Ronsse. (2012). Variable stiffness actuator applied to an active ankle prosthesis: Principle, energy-efficiency, and control. Paper presented at the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 323-328. doi:10.1109/IROS.2012.6385789

Dahl, S. (2011). Striking movements: A survey of motion analysis of percussionists. Acoustical Science and Technology, 32(5), 168-173.

Dahl, S., & Altenmüller, E. (2008). Motor control in drumming: Influence of movement pattern on contact force and sound characteristics. The Journal of the Acoustical Society of America, 123(5), 3122. doi:10.1121/1.2933043

Fujii, S., Hirashima, M., Kudo, K., Ohtsuki, T., Nakamura, Y., & Oda, S. (2011). Synchronization error of drum kit playing with a metronome at different tempi by professional drummers. Music Perception: An Interdisciplinary Journal, 28(5), 491-503.

Fujii, S., Kudo, K., Ohtsuki, T., & Oda, S. (2009). Tapping performance and underlying wrist muscle activity of non-drummers, drummers, and the world's fastest drummer. Neuroscience Letters, 459(2), 69-73. doi:https://doi.org/10.1016/j.neulet.2009.04.055

Geeroms, J., Flynn, L., Jimenez-Fabian, R., Vanderborght, B., & Lefeber, D. (2018). Energetic analysis and optimization of a MACCEPA actuator in an ankle prosthesis: Energetic evaluation of the CYBERLEGs alpha-prosthesis variable stiffness actuator during a realistic load cycle. Autonomous Robots, 42(1), 147-158. doi:10.1007/s10514-017-9641-1

Genesis Robotics. (2017, March 24th). *New Level of Speed & Safety* [Video]. Youtube. https://www.youtube.com/watch?v=_K4xtUiHz7Y

G. Kenneally, A. De, & D. E. Koditschek. (2016). Design principles for a family of direct-drive legged robots doi:10.1109/LRA.2016.2528294

Gomes Leal Junior, A., Milanezi de Andrade, R., & Filho, A. B. (2016). Series elastic actuator: Design, analysis and comparison. IntechOpen, Retrieved from http://dx.doi.org/10.5772/63573

G. Tonietti, R. Schiavi, & A. Bicchi. (2005). Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. Paper presented at the Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 526-531. doi:10.1109/ROBOT.2005.1570172

Hennig, H., Fleischmann, R., Fredebohm, A., Hagmayer, Y., Nagler, J., Witt, A., . . . Geisel, T. (2012). The nature and perception of fluctuations in human musical rhythms. Aps, 2012, J54. 003.

Kapur, A. (2005). A History of Robotic Musical Instruments. ICMC.

Kapur, A., Trimpin, E., Singer, A. & Suleman, G. (2007). A comparison of solenoid-based strategies for robotic drumming. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2007.188

Kawakami, H., Mito, Y., Watanuma, R., & Marumo, M. (2008). Analysis of drum player's motion . Acoustics'08 Paris, , 10065-10068. Retrieved from https://pdfs.semanticscholar.org/61cf/861ec36058681692be7eba13dda543086131.pdf?_ga=2.152593904.898432225.1599944004-1174519571.1599522421

Khodambashi, R., Weinberg, G., Singhose, W., Rishmawi, S., Murali, V., & Kim, E. (2018). User oriented assessment of vibration suppression by command shaping in a supernumerary wearable robotic arm Retrieved from https://www.openaire.eu/search/publication?articleId=od_____18::59c127d978db5fe874fb6b92cf02e701

Kim, Y. G., Garabini, M., Jaeheung Park, & Bicchi, A. (Sep 2014). Drum stroke variation using variable stiffness actuators. Paper presented at the 3892-3897. doi:10.1109/IROS.2014.6943109 Retrieved from https://ieeexplore.ieee.org/document/6943109

Lamb, J. (2015). Anatomy of drumming: Move better, feel better, play better CreateSpace.

Long, J., Murphy, J. W., Kapur, A., & Carnegie, D. A. (2015). A comparative evaluation of percussion mechanisms for musical robotics applications. Paper presented at the 2015 6th International Conference on Automation, Robotics and Applications (ICARA), 173-178. doi:10.1109/ICARA.2015.7081143

Mayne, R. G. (1992). An investigation of the use of facial expression in conjunction with musical conducting gestures and their interpretation by instrumental performers (Doctoral dissertation, The Ohio State University). Retrieved from ProQuest Dissertations and Theses database. (UMI No. 9238229)

M. Sidler, M. Bisson, J. Grotz, S. Barton. (2020). Parthenope: A Robotic Musical Siren. Proceedings of the International Conference on New Interfaces for Musical Expression, Birmingham City University, pp. 297–300.

Rasch, R. A. (1988). Timing and synchronization in ensemble performance.

Van Rooyen, R., Schloss, A., & Tzanetakis, G. (2016). Snare Drum Performance Motion Analysis. New Interfaces for Musical Expression.

Rooyen, R. V., Schloss, A., & Tzanetakis, G. (2017). Voice coil actuators for percussion robotics. Paper presented at the doi:10.5281/zenodo.1176148 Retrieved from https://search.datacite.org/works/10.5281/zenodo.1176148

R. Schiavi, G. Grioli, S. Sen, & A. Bicchi. (2008). VSA-II: A novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. Paper presented at the 2008 IEEE International Conference on Robotics and Automation, 2171-2176. doi:10.1109/ROBOT.2008.4543528

Song, Y., Dixon, S., & Pierce, M. (2012). Evaluation of Musical Features for Emotion Classification. Conference: International Society for Music Information Retrieval Conference. Retrieved from https://www.researchgate.net/publication/277715954_Evaluation_of_Musical_Features_for_Emotion_Classification.

S. Wolf, G. Grioli, O. Eiberger, W. Friedl, M. Grebenstein, H. Höppner, . . . A. Albu-Schäffer. (2016). Variable stiffness actuators: Review on design and components doi:10.1109/TMECH.2015.2501019

T. Boaventura, C. Semini, J. Buchli, M. Frigerio, M. Focchi, & D. G. Caldwell. (2012). Dynamic torque control of a hydraulic quadruped robot. Paper presented at the 2012 IEEE International Conference on Robotics and Automation, 1889-1894. doi:10.1109/ICRA.2012.6224628

Time-of-Flight principle: Technologies and advantages. (2020, November 27). Retrieved from https://www.terabee.com/time-of-flight-principle/

T. Zhu, J. Hooks, & D. Hong. (2019). Design, modeling, and analysis of a liquid cooled proprioceptive actuator for legged robots. Paper presented at the 2019 IEEE/ASME International

Conference on Advanced Intelligent Mechatronics (AIM), 36-43. doi:10.1109/AIM.2019.8868596

Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., & Lefeber, D. (2007). MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. Robotics and Autonomous Systems, 55(10), 761-768. doi:https://doi.org/10.1016/j.robot.2007.03.001

Van Rooyen, R., Schloss, A., & Tzanetakis, G. (2016). Snare drum performance motion analysis. New Instruments for Musical Expression,

Vanderborght, B., Albu-Schaeffer, A., Bicchi, A., Burdet, E., Caldwell, D. G., Carloni, R., . . . Wolf, S. (2013). Variable impedance actuators: A review. Robotics and Autonomous Systems, 61(12), 1601-1614. doi:https://doi.org/10.1016/j.robot.2013.06.009

Vines, B. W., Wanderley, M. M., Krumhansl, C. L., Nuzzo, R. L., & Levitin, D. J. (2004). Performance Gestures of Musicians: What Structural and Emotional Information Do They Convey? Gesture-Based Communication in Human-Computer Interaction Lecture Notes in Computer Science, 468-478. doi:10.1007/978-3-540-24598-8_43

Voigt, T. (2016). Hearing What You See: A Case for the Use of Ancillary Gesture in Individual Percussion Performance.

Vonnegut, K. (2006). Player piano. New York: Dial Press.

Wagner, A. (2006). Analysis of drumbeats – interaction between drummer, drumstick and instrument

Yang, N., Savery, R., Sankaranarayanan, R., Zahray, L., & Weinberg, G. (2020). Mechatronics-driven musical expressivity for robotic percussionists. Retrieved from https://arxiv.org/abs/2007.14850

Appendix A: Matlab File for Spring Calculations

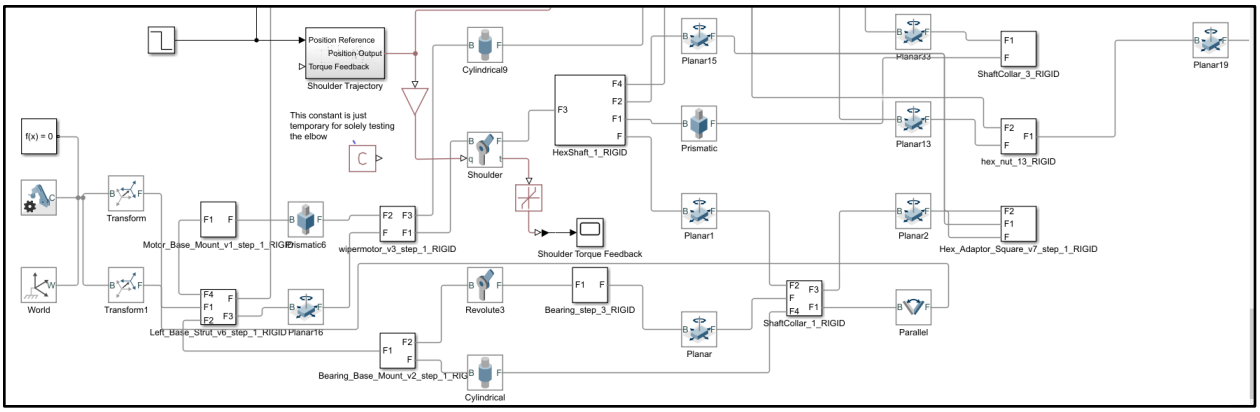
```
%% Torque on Shoulder Joint with Spring Compensation

clear all

%motor torque given spring compensation

H_off = 0.05; %m

S_off = 0.20; %m

angles = [0:1:90];

theta1 = [0:1:90]*(pi/180);

L = 0.3; %m

Fg = 19; %N

Fa =  22.1; %N



Sp = sqrt(S_off^2 + H_off^2 - 2*H_off*S_off.*cos(pi-theta1)); %length of Sp for different angle

theta2 = asin(H_off./Sp .* sin(pi-theta1)); %offset of spring from arm angle

Ty = sin(1.57-theta1+theta2); %spring y transform

Tx = cos(1.57-theta1+theta2); %spring x transform


%spring force calculation

Sp_nominal = Sp(end); %unstretched spring length

k = 1500; %spring constant N/m

Fs = k*(Sp-Sp_nominal); %Force of spring



Torque = (Fg*(L/2).*sin(theta1)+Fa*L.*sin(theta1))+(Fs.*S_off.*Tx.*cos(theta1))-(Fs.*S_off.*(Ty.*sin(theta1)));

Fsx = Fs.*Tx;

Fsy = Fs.*Ty;
```

```
plot(angles,Torque)
hold on
plot(angles,Fsx)
plot(angles,Fsy)
plot(angles,Fs)
%plot(angles,theta2*180/pi)
xlabel('degrees from negative vertical');
ylabel('Newtons of spring force');
%legend('Torque','Fsx','Fsy','theta2');
legend('Torque','Fsx','Fsy','Fs');
hold off

k * 0.255/39.37

figure;
plot(Fs)



%% Torque on Elbow Joint with Spring Compensation
clear all
Fw = 17.25; %N
Fe = 5; %N
h2off = 0.02; %m
L = 0.355; %m
L2off = 0.145; %m
k = 400;

anglesdeg = [0:90];
```

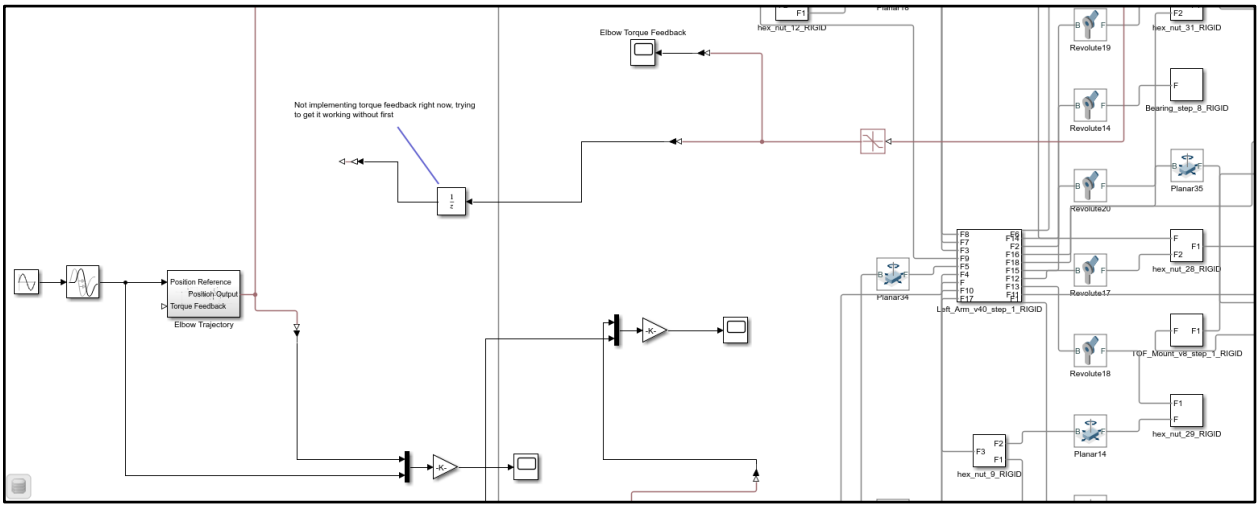
```
angles = [0:90]*pi/180;


Spring_length = sqrt((h2off-(L2off.*sin(angles))).^2+(L2off.*cos(angles).^2));
Nominal_length = Spring_length(37);


Spring_angles = atan2((L2off.*sin(angles)-h2off),L2off.*cos(angles));
Fsx = (Spring_length-Nominal_length).*k.*cos(Spring_angles);
Fsy = (Spring_length-Nominal_length).*k.*sin(Spring_angles);


Torque = -(Fe*(L/2).*cos(angles))-(Fw*L*cos(angles))+(Fsx.*L2off.*cos(angles))+(Fsy.*L2off.*sin(angles));
Torque_no_s = -(Fe*(L/2).*cos(angles))-(Fw*L*cos(angles));


hold on
plot(anglesdeg,Torque_no_s);
plot(anglesdeg,Torque);
xlabel("angles degree");
ylabel("Sum of Moments");
legend('No Comp.', 'With Spring');
hold off


k * 0.255/39.37


Fs = sqrt(Fsx.^2+Fsy.^2);
figure;
plot(Fs)
```

Appendix B: Drumming Arm Simulation Schematics

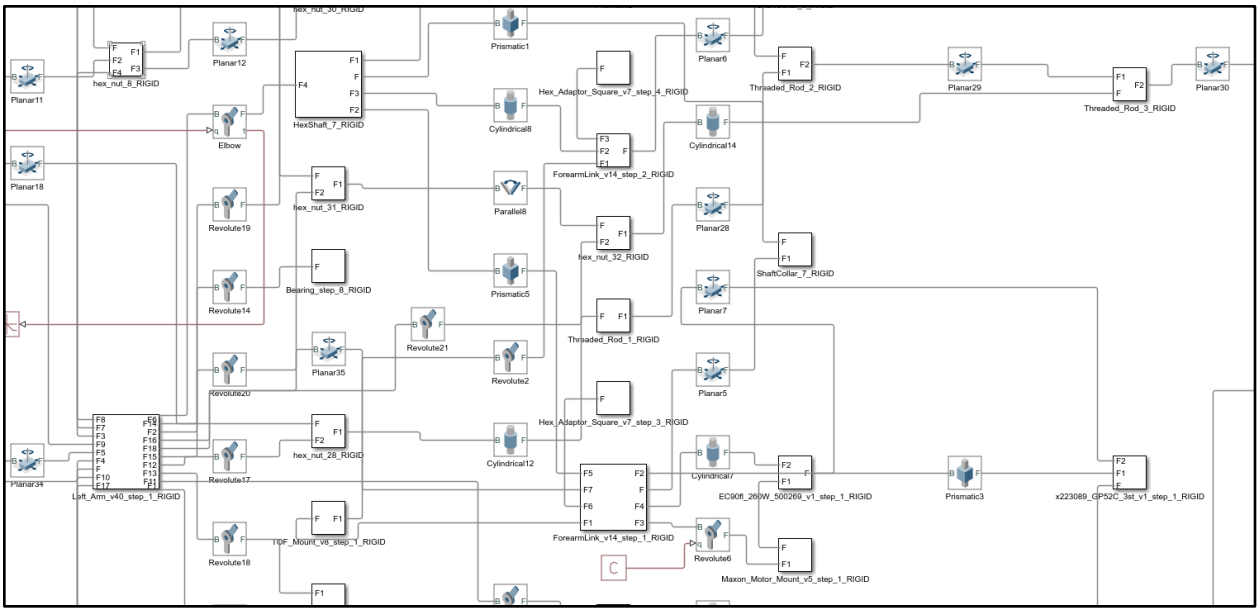
Physical Modelling for the Entire Arm



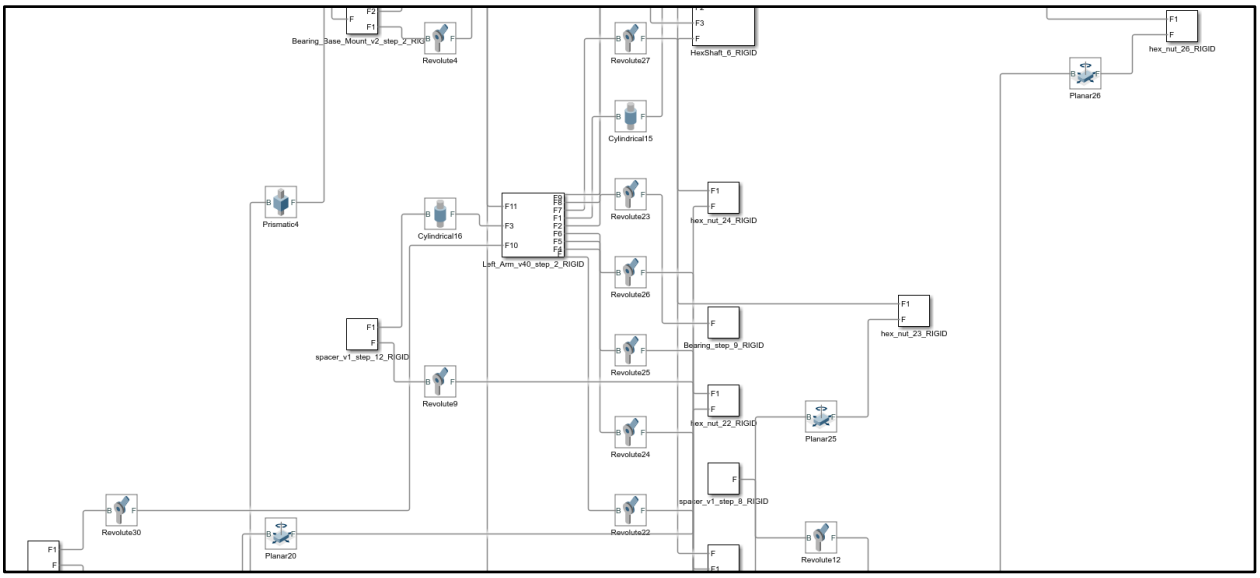
World Frame Reference, Solver, Shoulder Actuation Integration



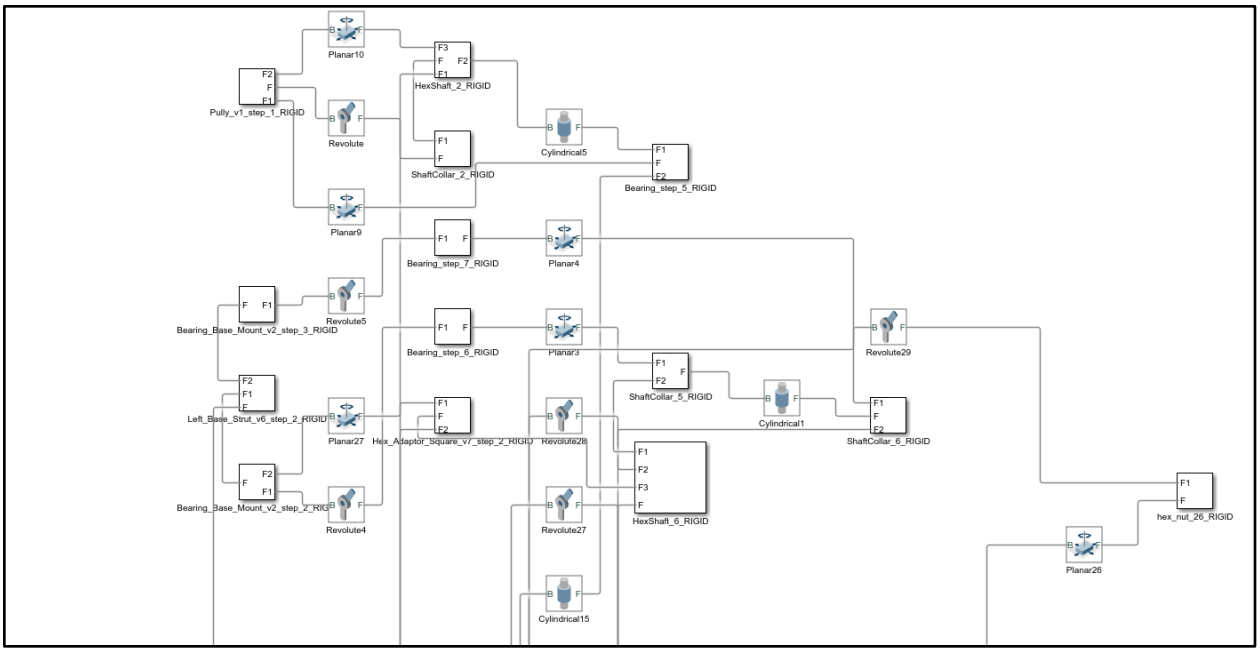
Elbow Actuation Integration Pt 1



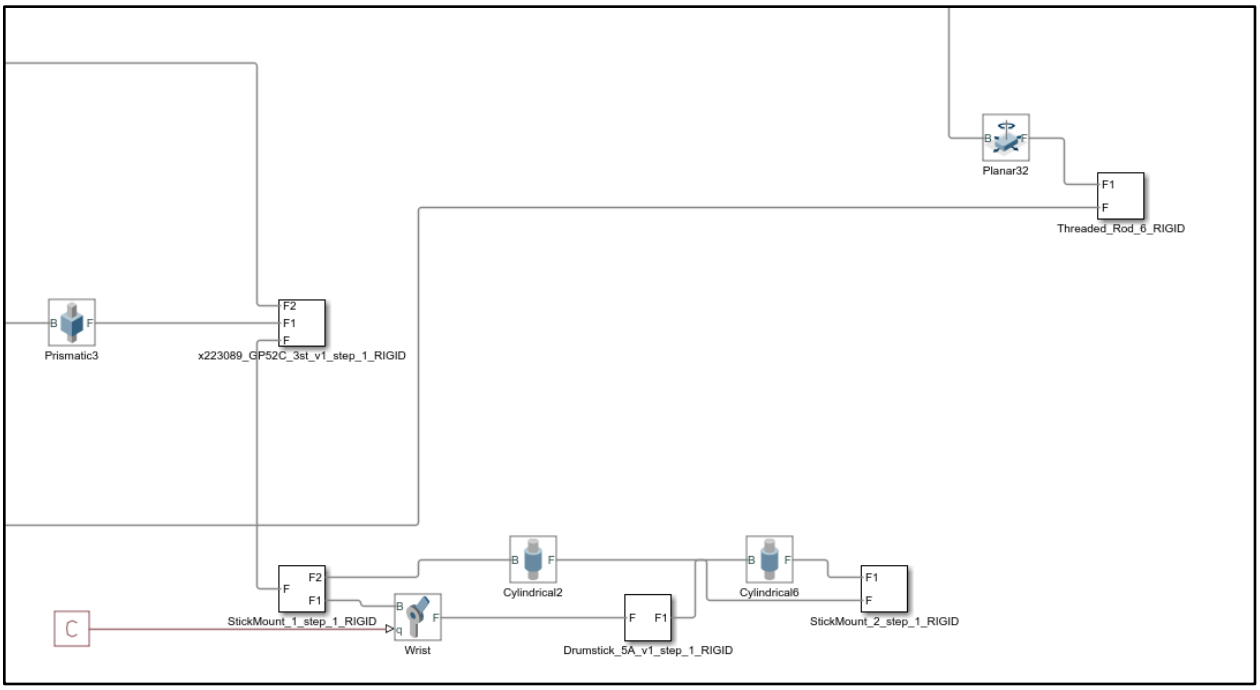
Forearm and Bicep Assembly



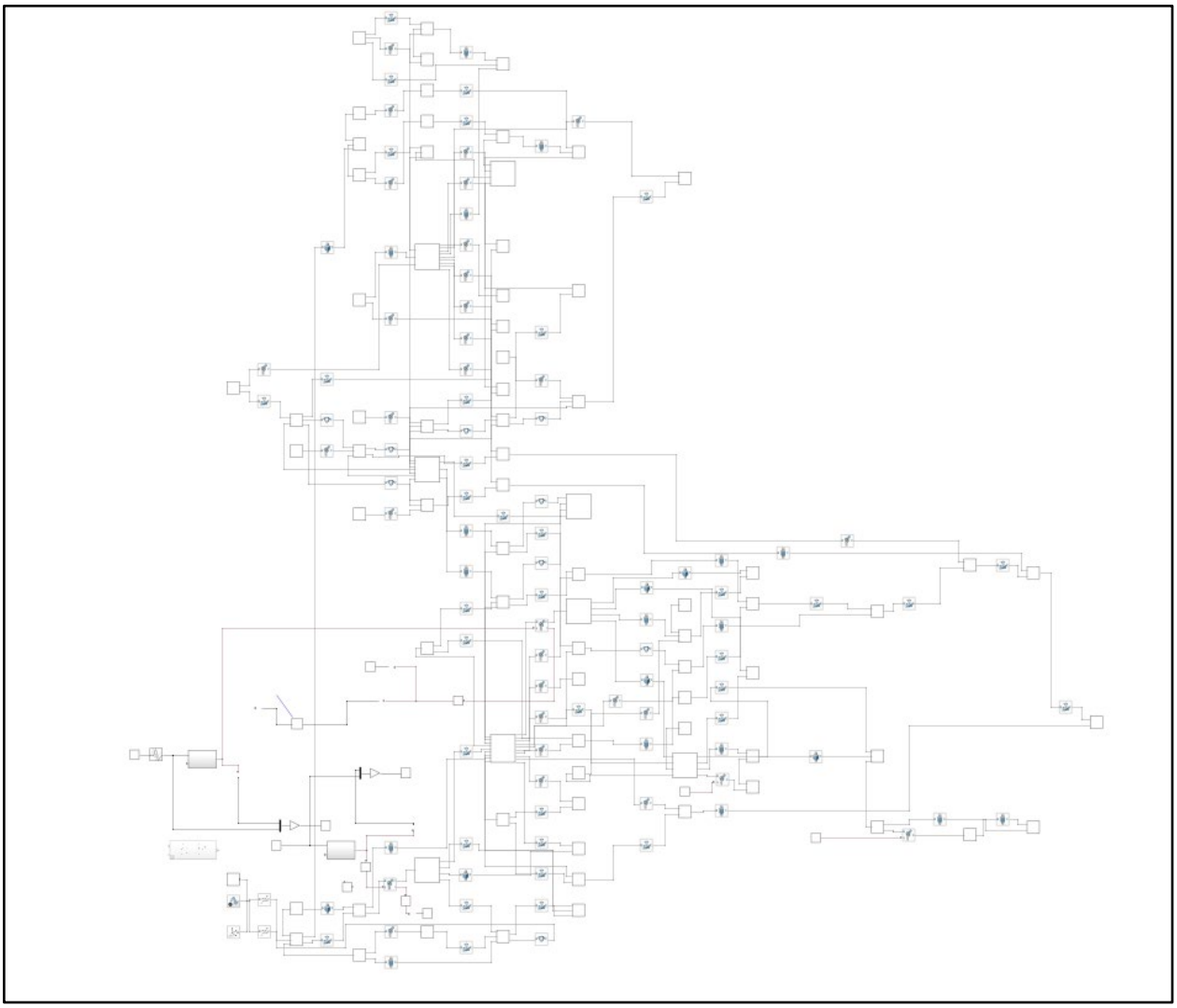
Left Half of Bicep Assembly



Base frame and Elbow Pulley System

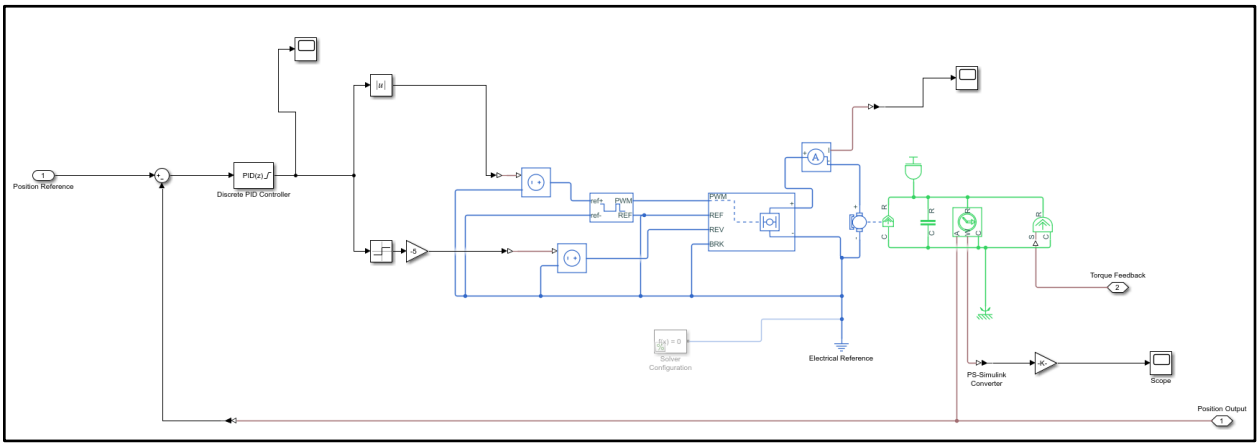


Wrist Subassembly



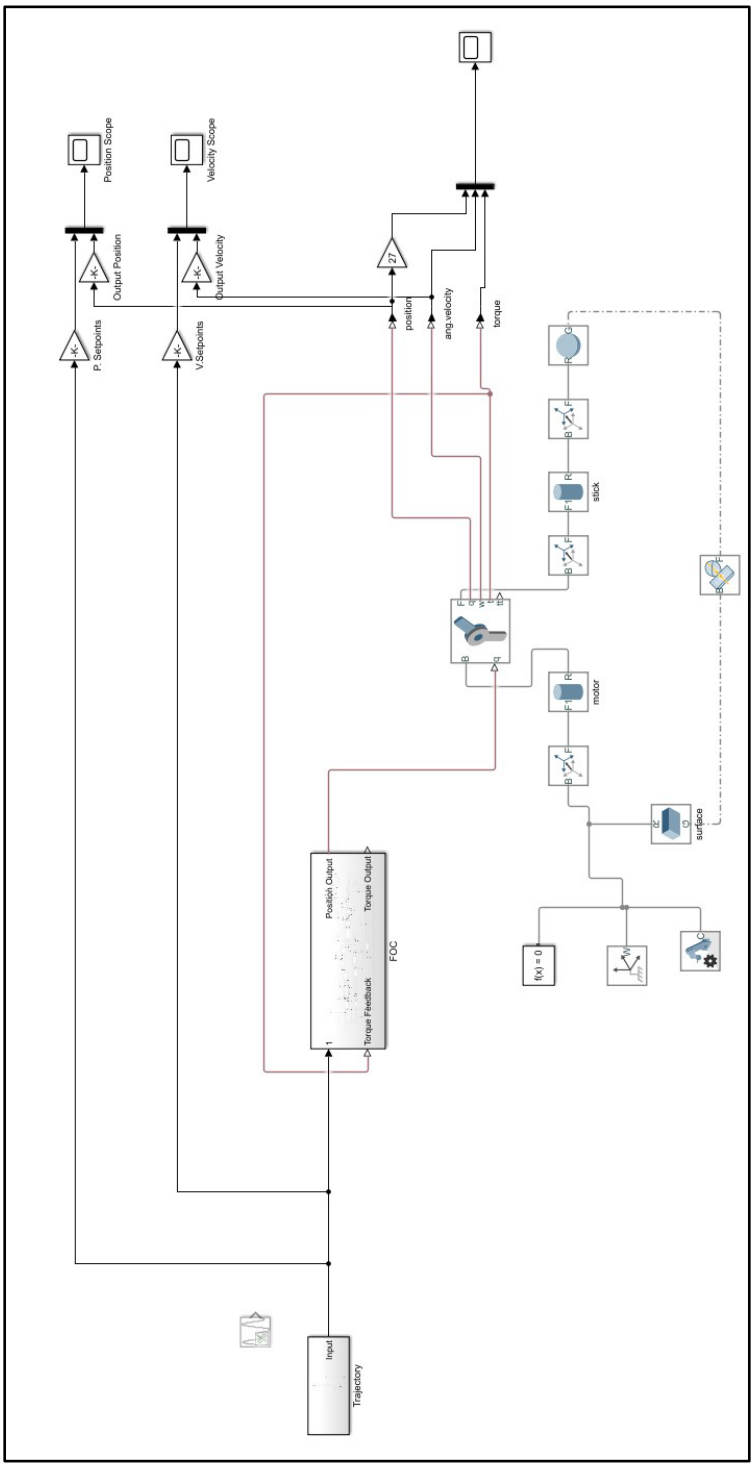
Entire Arm Assembly Overview

Physical Modelling of Wiper Motor

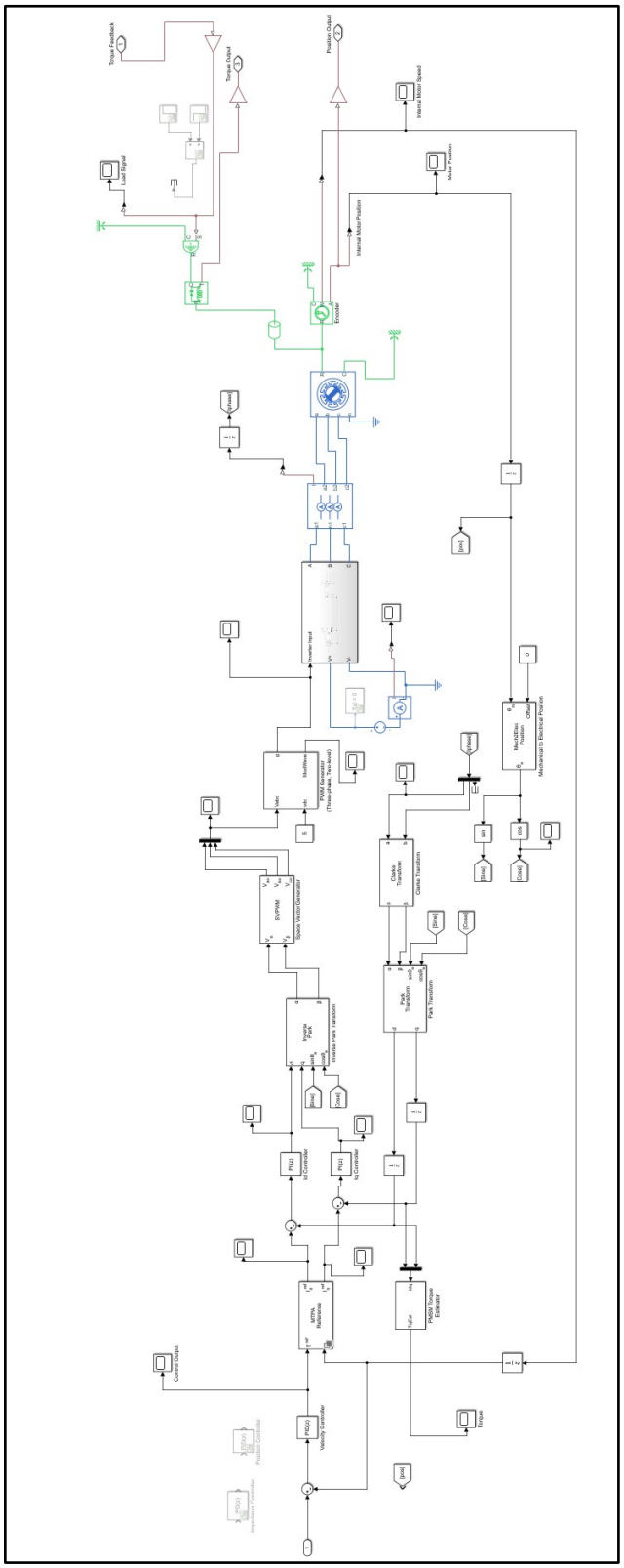


Appendix C: Wrist Actuator Simulation Schematics

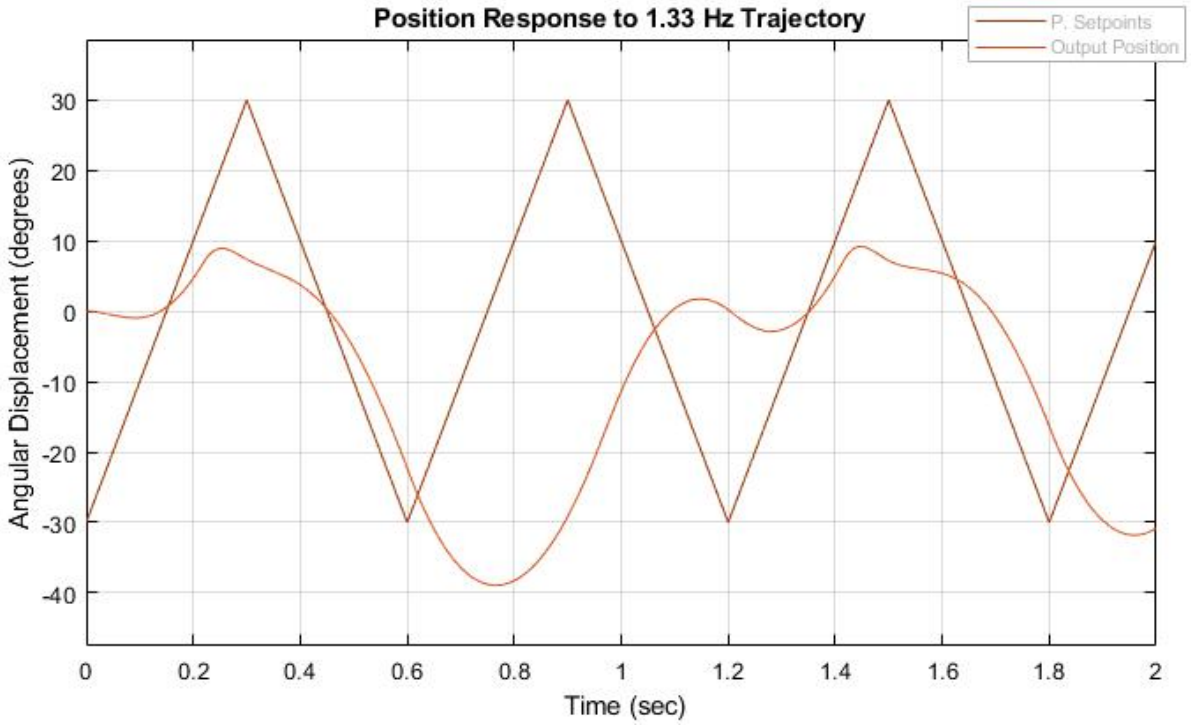
Physical Modelling for Wrist Actuator

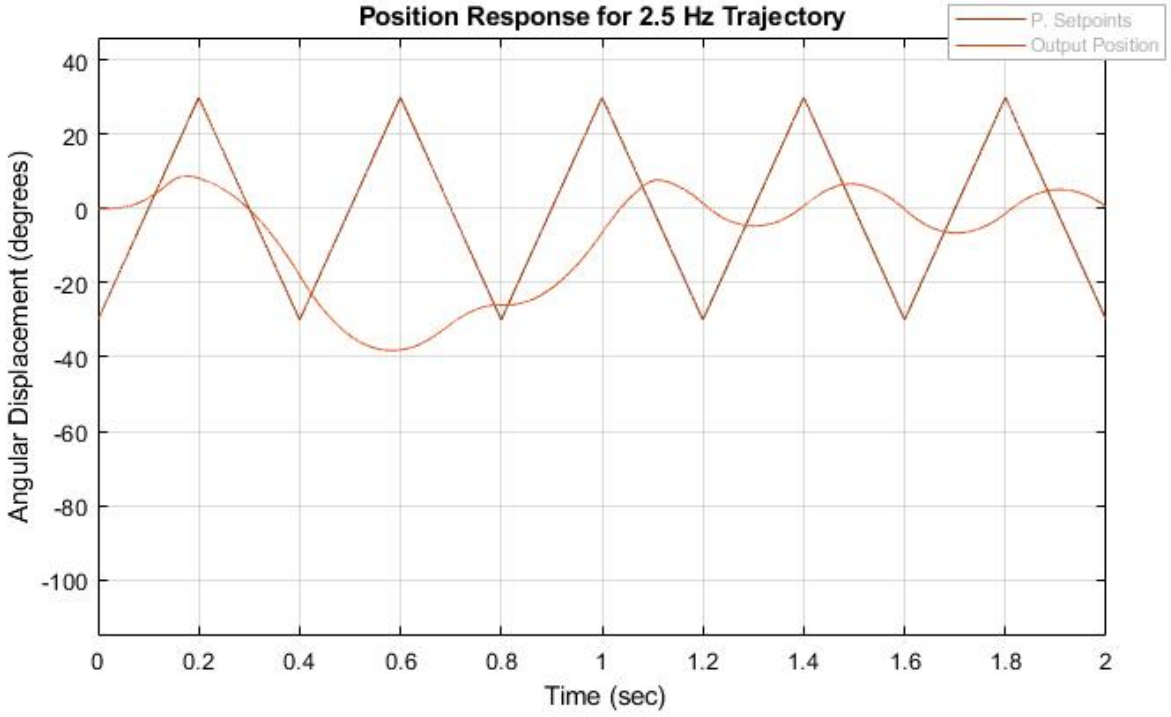


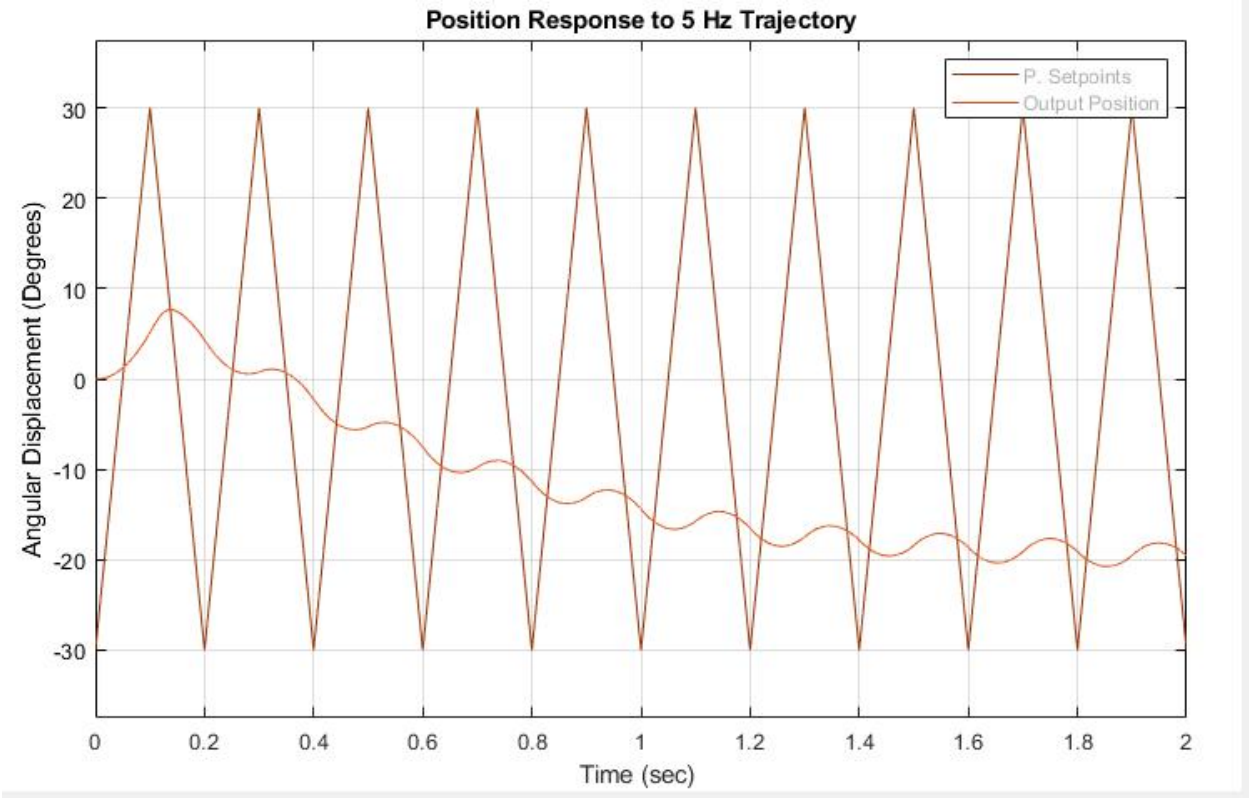
Electrical Modeling for BLDC Control

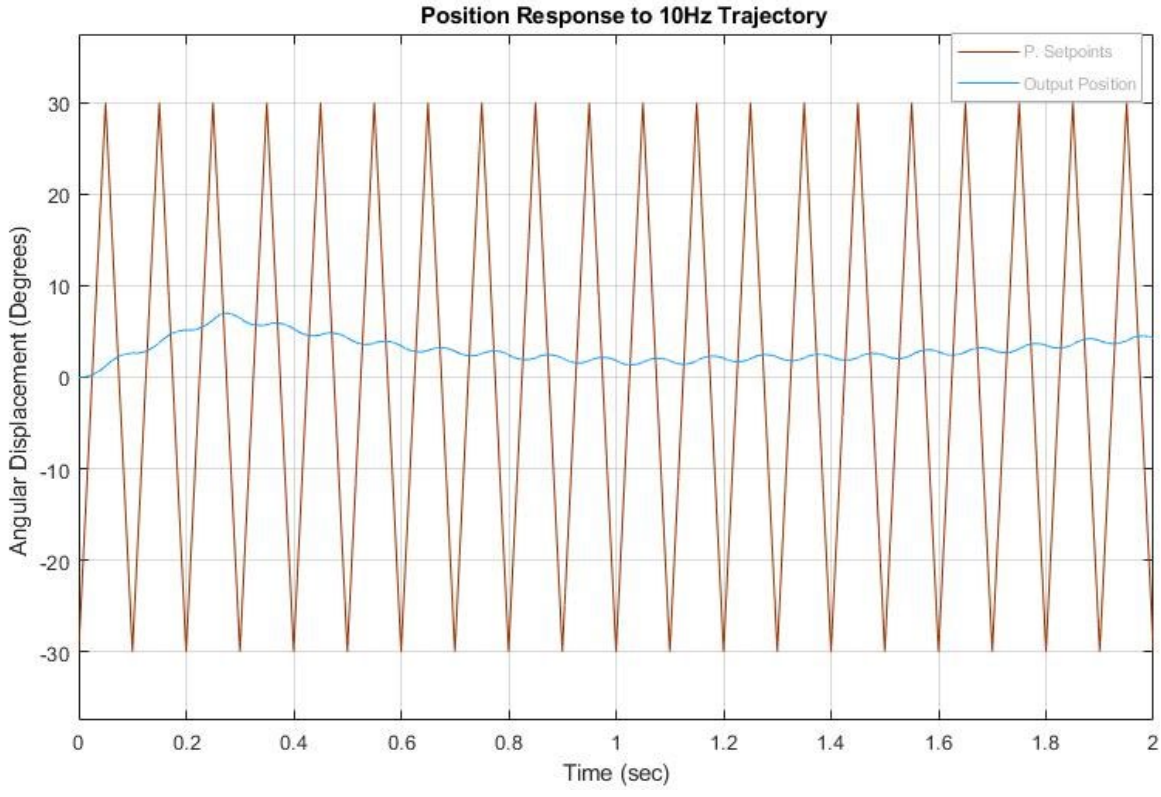


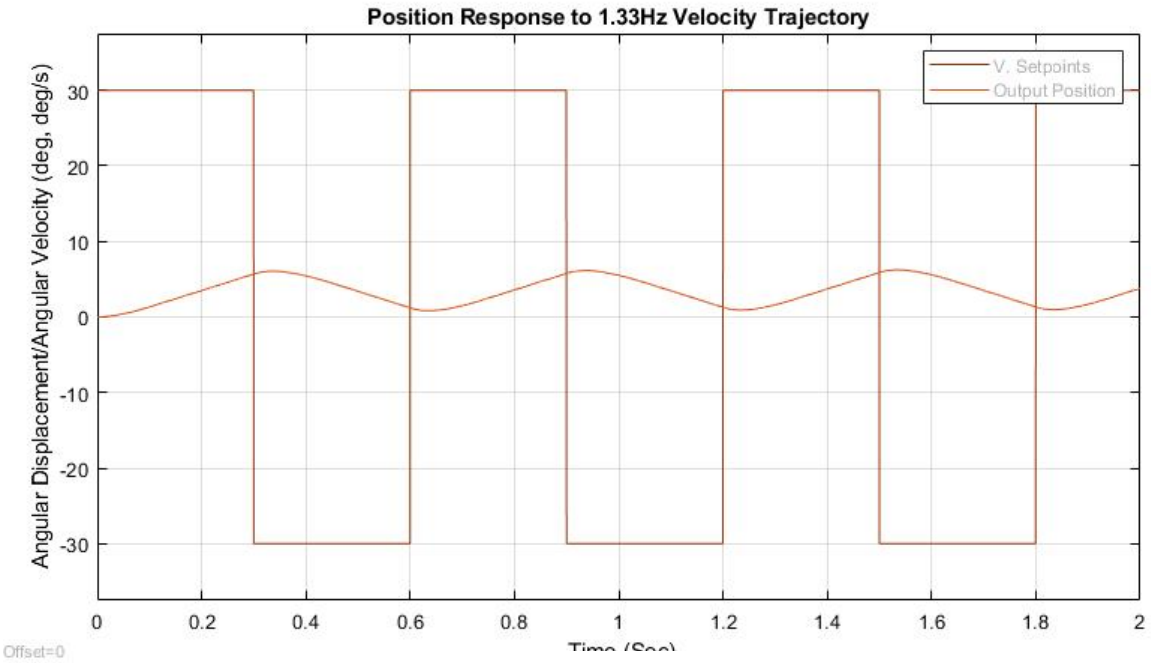
Appendix D: Frequency Response Graphs

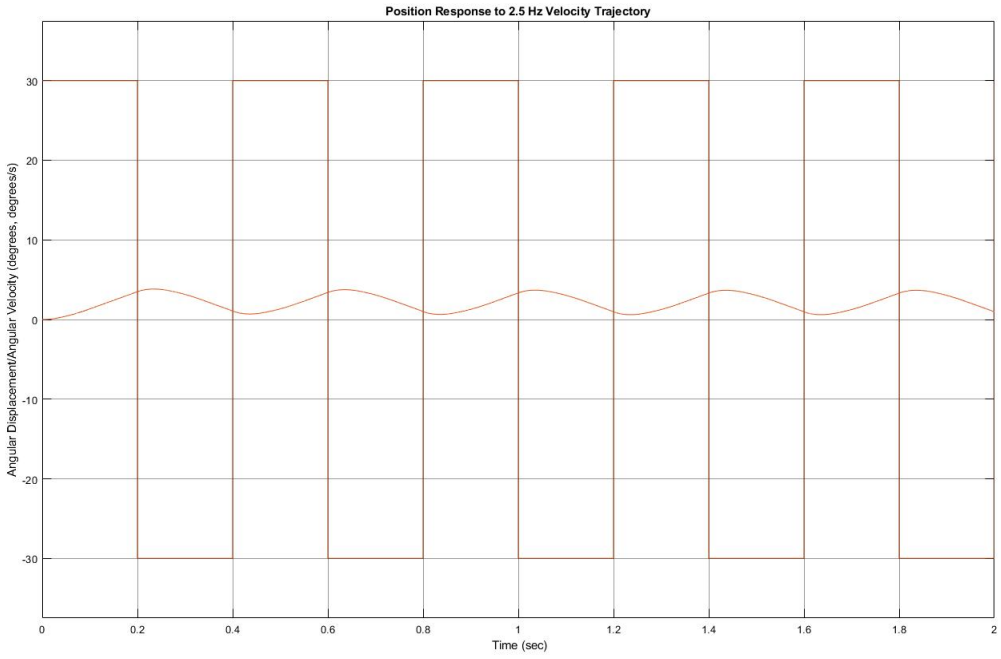


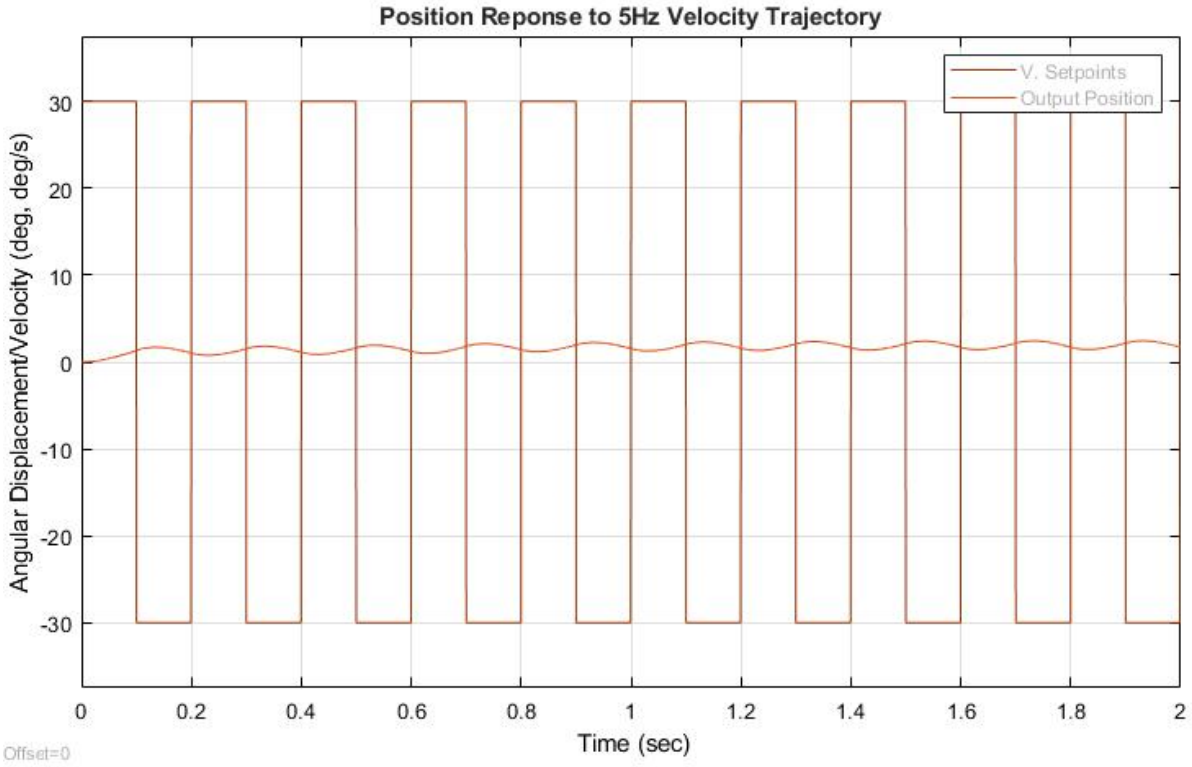


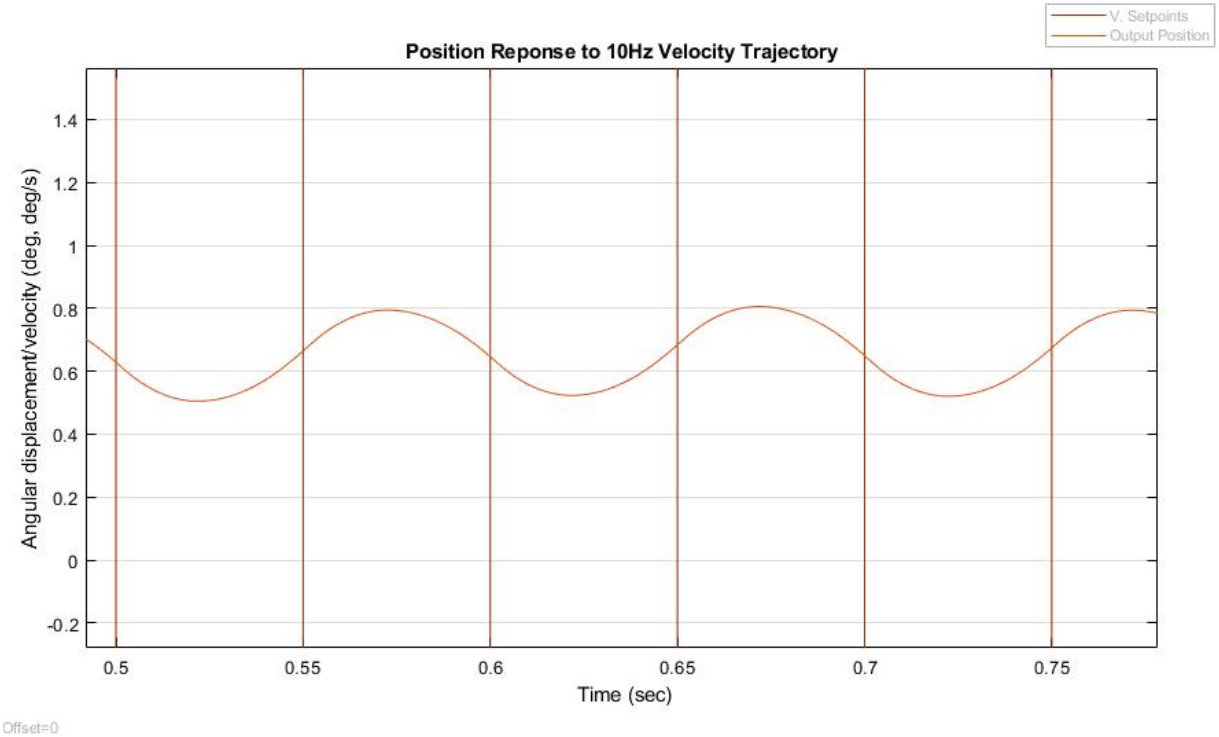

Position Response to 5 Hz Trajectory
P. Setpoints
Output Position
Angular Displacement (Degrees)
Time (sec)
30
20
10
0
-10
-20
-30
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Position Response to 10Hz Trajectory
P. Setpoints
Output Position
Angular Displacement (Degrees)
Time (sec)
30
20
10
0
-10
-20
-30
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Position Response to 1.33Hz Velocity Trajectory
V. Setpoints
Output Position
Angular Displacement/Angular Velocity (deg, deg/s)
30
20
10
0
-10
-20
-30
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Offset=0


Position Response to 2.5 Hz Velocity Trajectory
Angular Displacement/Angular Velocity (degrees, degrees/s)
30
20
10
0
-10
-20
-30
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (sec)


Position Reponse to 5Hz Velocity Trajectory
V. Setpoints
Output Position
Angular Displacement/Velocity (deg, deg/s)
30
20
10
0
-10
-20
-30
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (sec)
Offset=0


V. Setpoints
Output Position
Position Reponse to 10Hz Velocity Trajectory
Angular displacement/velocity (deg, deg/s)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0.5
0.55
0.6
0.65
0.7
0.75
Time (sec)
Offset=0

Appendix E: Peak Detection Algorithm Code

```
#include <PeakDetection.h> // import lib

#include <Adafruit_ICM20649.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

PeakDetection peakDetection; // create PeakDetection object
PeakDetection gyroPeak;

float rawGyro, minGyro, maxGyro, peakGyro;
float analogMic;

//Set up IMU
Adafruit_ICM20649 icm;
uint16_t measurement_delay_us = 65535; // Delay between measurements for testing
// For SPI mode, we need a CS pin
#define ICM_CS 10
// For software-SPI mode we need SCK/MOSI/MISO pins
#define ICM_SCK 13
#define ICM_MISO 12
#define ICM_MOSI 11

void setup() {
  Serial.begin(9600); // set the data rate for the Serial communication
  pinMode(A0, INPUT); // analog pin used to connect the sensor
  peakDetection.begin(48, 3, 0.3); // sets the lag, threshold and influence
  gyroPeak.begin(48, 10, 0.8);
  pinMode(13, OUTPUT);
  pinMode(5, OUTPUT);

  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("Adafruit ICM20649 test!");

  // Try to initialize!
  if (!icm.begin_I2C()) {
    Serial.println("Failed to find ICM20649 chip");
    while (1) {
      delay(10);
    }
}
//Sets up the IMU correctly
  Serial.println("ICM20649 Found!");
  Serial.print("Accelerometer range set to: ");
  switch (icm.getAccelRange()) {
  case ICM20649_ACCEL_RANGE_4_G:
```

```
    Serial.println("+-4G");
    break;
  case ICM20649_ACCEL_RANGE_8_G:
    Serial.println("+-8G");
    break;
  case ICM20649_ACCEL_RANGE_16_G:
    Serial.println("+-16G");
    break;
  case ICM20649_ACCEL_RANGE_30_G:
    Serial.println("+-30G");
    break;
  }

  Serial.print("Gyro range set to: ");
  switch (icm.getGyroRange()) {
  case ICM20649_GYRO_RANGE_500_DPS:
    Serial.println("500 degrees/s");
    break;
  case ICM20649_GYRO_RANGE_1000_DPS:
    Serial.println("1000 degrees/s");
    break;
  case ICM20649_GYRO_RANGE_2000_DPS:
    Serial.println("2000 degrees/s");
    break;
  case ICM20649_GYRO_RANGE_4000_DPS:
    Serial.println("4000 degrees/s");
    break;
  }

  uint16_t accel_divisor = icm.getAccelRateDivisor();
  float accel_rate = 1125 / (1.0 + accel_divisor);

  Serial.print("Accelerometer data rate divisor set to: ");
  Serial.println(accel_divisor);
  Serial.print("Accelerometer data rate (Hz) is approximately: ");
  Serial.println(accel_rate);

  uint8_t gyro_divisor = icm.getGyroRateDivisor();
  float gyro_rate = 3300 / (1.0 + gyro_divisor);

  Serial.print("Gyro data rate divisor set to: ");
  Serial.println(gyro_divisor);
  Serial.print("Gyro data rate (Hz) is approximately: ");
  Serial.println(gyro_rate);
  Serial.println();
}

void loop() {
  sensors_event_t accel;
```

```
  sensors_event_t gyro;
  icm.getEvent(&accel, &gyro);
    analogMic = analogRead(A0);
    double data = (double)analogRead(A0)/512-1; // reads the value of the sensor and converts to a range
between -1 and 1
    peakDetection.add(data); // adds a new data point
    int peak = peakDetection.getPeak(); // returns 0, 1 or -1
    double filtered = peakDetection.getFilt(); // moving average

    rawGyro = gyro.gyro.y;
    minGyro = 0;
    maxGyro = 0;

    if (rawGyro < minGyro) {
      minGyro = rawGyro;
    }
    else if (rawGyro > maxGyro) {
      maxGyro = rawGyro;
    }

    peakGyro = maxGyro - minGyro;

   //checks for and condition for gyroscope and audio
    if (peak == 1 && peakGyro > 2 ) {
    digitalWrite(5, HIGH);   // for oscilloscope measurement
    delay(10);
    digitalWrite(5, LOW); // for oscilloscope measurement
 //Print results after hit is detected
    Serial.print("Microphone Peak:");
    Serial.println(analogMic);
    Serial.print("Gyroscope Value:");
    Serial.println(rawGyro);

  }
}
```