Bio-Inspired Interactive Dino
A Major Qualifying Project

Submitted by:
Samantha Grillo
Kenny Rhodes
Kalani Picho
Alec Kneedler

Advised by:
Professor Marko Popovic
Professor Gregory Fischer
Professor George Heineman

This project aimed to create an interactive robot by modeling the head and neck of a *Deinonychus*. The neck is modeled as a continuum arm with six actuated degrees of freedom controlled with Hydro Muscles, elastic linear actuators which replicate biological muscle motion. Core components drew inspiration from anatomical structures to enhance lifelike behavior. The robot is capable of neck, and jaw, and eye movement. Future work will focus on developing sensor-based behavior.

Executive Summary

Researchers are still exploring the concept of human robot interaction. Robots have potential to entertain, interact, and educate the public with their existence. Modern animatronics typically have a set sequence of poses and motions programmed into them and their bodies are typically rigid and therefore not appropriate for safe, direct interaction with humans. The field of robotics can bring in the aspect of sensing, and reacting to animatronics, while soft robotic models allow compliant and fluid structures to be fabricated. The goal of this project was to create a bio-inspired interactive dinosaur.

To understand the physical structure and believed motions of the *Deinonychus,* the team researched paleontological findings, read about osteology and muscular research of the dinosaur and its analogs, and conversed with paleontologists and other experts in relevant fields. The team also performed research on Popovic Lab's Hydro Muscles & CRFC valves, Arduino networking techniques, and kinematic modeling of continuum arms. The focus of the project was to approximate the head and neck movement of the *Deinonychus* as determined by the range of vertebral flexion found in two analogs, with minimum and maximum flexion determined by the *Allosaurus* and the common bird. A goal of organic movement and bio-inspiration heavily drove the design aspects of this project.

This project aims to create the impression of a living *Deinonychus* via synthetic skin and audio based on current scientific knowledge and the public perception of the *Deinonychus*. The robot should be capable of two distinct interactive modes. The first mode is a kiosk mode in which the robot performs ambient actions and sounds until a spectator is detected. The robot then reacts by following the face of the spectator, creating the illusion that the robot is "watching" the spectator. The second mode is a grab mode during which the robot will locate and grab onto a ~ 3-inch diameter object in the robot's workspace.

To design and implement the integrated robotic system for the *Deinonychus*, the team had to consider the physical, electrical, and software aspects of the system. The team designed custom 3D printed vertebrae based on skeletal research as well as the desired range of motion. The team developed 9 different vertebra prototypes before selecting the final design. The final vertebrae design features four processes to enable deflection in two directions and fit together with ball-and-socket joints. The spine was constructed of ten vertebrae and split into three sections, each with different flexion capabilities. The top three vertebrae make up Segment A, the middle four vertebrae make up Segment B, and the bottom three vertebrae make up Segment C. SpiderWire tendons were attached to the processes of the top vertebra in each section, and Bowden cables were used to translate the motion of the tendons more efficiently.

The vertebral column was actuated using Hydro Muscles, hydraulically actuated muscle analogs developed at Popovic Labs. The team chose Hydro Muscle specifications based on the estimated force needed to actuate the spine via the tendons. The team then designed a hydraulic system capable of actuating the muscles. The system included custom-made CRFC valves to control the Hydro Muscles and a bypass pump with the necessary pressure and flow to provide for estimated system operating needs. The Hydro Muscles were housed in muscle boxes located next to the vertebral column to reduce the weight of the spine. Each muscle box was capable of

housing two Hydro Muscles in an antagonistic pair and included slider mounts for the potentiometers used to measure the change in muscle length.

The team created the robot skull based on a modified laser scan of a Bones Unlimited replica of a *Deinonychus* skull from CarneyLabs.org. A hinge was added to the jaw, and hardware was added for movable eyes. The eyes were actuated via a single servo turning a central gear. The jaw was actuated using two servos, with one servo mounted on each side of the jaw. The skull was mounted to the top of the vertebral column via an actuatable joint that enabled the skull to be rotated axially using a servo mounted on the top vertebra of the spine.

The team developed and assembled an electric system capable of supporting the servos and sensors necessary to actuate and control the robot. The electrical system bridges the gap between the robot's software and hardware development. The primary goals during development were to incorporate sensors to measure muscle length and neck position, to provide PWM control for the hydraulics valves, and to power the entire system independent of lab power supplies.

The team implemented an expandable network of Arduino Nano Every's to coordinate Hydro Muscle movement simultaneously between multiple vertebral sections. This network served to distribute the load of controlling each pair of Hydro Muscles. The network was expanded to provide methods for controlling groups of muscles to bend vertebrae sections into the desired vertebral column shape and report the robot pose in physical space.

The team conducted multiple experiments on separate aspects of the robot before integrating the components into one cohesive robotic system. The team tested various Hydro Muscle tubing dimensions before selecting the final parameters for the muscles. The final muscle design was chosen based on the desired change in tendon length and estimated force required to actuate the tendons. The team also tested the actuation and range of the eye and jaw mechanisms to ensure their functionality before mounting the head on the vertebral column. A simple testing code was used to move the servos on each system, and the range of motion was observed. The team tested the correlation between the deflection of a vertebral section and the movement of the potentiometers used to measure the change in muscle length. This was used to map the muscle system to the deflection of the vertebral column. Once the vertebral column was fully assembled and attached to the Hydro Muscles via tendons, each section was actuated, and the deflection of the spine was observed. Simple commands were used to fully deflect each actuated section to its two extremes, and the resulting spinal pose was observed.

The team integrated the hydraulic system to actuate the physical robotic system to a limited degree. The electronic system was utilized to power and connect the subassemblies, and a control system was developed to coordinate and control the robot and its microcontrollers. The skull assembly was fully functional and included a neck joint, eyes, and jaw. The spinal column, while actuatable, was not as nimble or quick as desired. The neck itself was composed of 3D printed vertebrae that were selected after several prototypes were fabricated. The chosen design was physically capable of the deflection maximums desired, however, problems created by friction and the axial rotation allowed by the ball and socket joint resulted in limited range of motion when actuated by the Hydro Muscles. These vertebrae were grouped into three sections, and each was actuated via SpiderWire cable contained by Bowden tubes. These cables experienced little friction within the PTFE Bowden tubing, but the tubing gradually deformed under the force of the cables. This actuation obtained most of the desired deflection and was achieved via antagonistic pairs of Hydro Muscles grouped together in custom muscle boxes. The muscle boxes linearized the

change in muscle length, however, their large size made them difficult to fabricate via 3D printing. Each muscle's length was measured by a linear potentiometer and controlled by a single CRFC valve. System power and networking functioned as desired, and the camera, IMUs, and microphone were installed for future integration. The control system provided a platform to control the robot's suite of Hydro Muscles and test robot motion via command line. The control system also established the foundations of kinematic control via modified DH parameters of the robot. Kinematic control was not ultimately obtained in the given time frame, but the team plans to incorporate a higher level of control in the future. When implemented, the kinematics will assist in implementing a variety of more complex autonomous behaviors, such as facial tracking and object manipulation.

Though the robot was capable of being actuated by the Hydro Muscles, the team encountered various issues with the physical system. Throughout the design and manufacturing process, the team encountered delays in shipping, manufacturing, and reduced access to physical lab space due to COVID19. When integrating the separate systems of the robot, the team experienced many problems due to friction, the weight of their robot head, and other design flaws. The spine had excessive friction between the ball and socket joints of its vertebra, which increased the force needed to actuate the neck. This friction increased the amount of force required to actuate each vertebral section, resulting in the selected Hydro Muscles not being strong enough to fully flex the spine. The head also added significant weight to the end of the spine, further increasing the amount of force the Hydro Muscles needed to exert to hold the neck steady and actuate the system. The neck had to be tied to the ceiling to allow the Hydro Muscles to actuate the neck without having to fight the force of gravity.

Based on these obstacles, the team generated some suggestions to guide future project work. The team believes that the vertebrae should be replaced with a modified version of vertebra prototype 4 to reduce friction in the spine and remove the need for Bowden tubes to direct the tendons. The skull should also be lightened to allow for the Hydro Muscles to actuate the spine, and Hydro Muscle designs capable of providing a greater force output should be considered. When 3D printing, higher tolerances should be used to allow for easier assembly. These improved mechanical subassemblies, along with integrated IMUs and camera, will allow software to be developed that uses the continuum arm kinematic model.

The electrical system worked as expected, but the control system was limited by the issues faced with the mechanics of the robot.

The muscle controller API successfully enabled communication between Arduino Nanos powered by the electrical system. The API uses an expandable messaging format built on top of I2C to command changes in Hydro Muscle length. Minor changes to this format could greatly increase its potential number of supported messages, but the current controller is effective at commanding muscle length deltas. The length of each muscle is maintained on secondary Arduino Nanos running a closed loop control PID implementation. This system was found to be effective for muscles under minimal load, but restrictions to muscle expansion limited the maximum length change.

Hydro Muscle lengths, alongside physical vertebrae dimensions, were used to calculate the parameters needed to calculate the pose of the robot in terms of Euclidean coordinates via a kinematic model. However, limitations in the actuation of the robot and time constraints prevented the inverse kinematics model from being used to command the pose of the robot. Behavior of

the robot is currently controlled via command line. Inputs to the command line allow for changes to muscle lengths and reporting of robot state. This interface is accessible via Serial monitor and allows an operator to interact with the robot. The interface is capable of being expanded to accommodate additional robot functionality and behaviors.

The team plans to continue to work on the *Deinonychus* robot during the summer of 2021. This work will take into consideration the suggested solutions made in the Discussion section and will focus on reaching several goals. Future work will prioritize the improvement of the actuation of the spine and the integration of sensor-driven interactive modes. These modes will be like the interactivity modes described in the original project goals. Additional software will be developed to integrate the IMUs and camera, while the kinematic model of a continuum arm will be used to allow the dinosaur to follow a subject. The team hopes that Perry the *Deinonychus* robot may prove to be a promising prototype for a future interactive science museum display.

Acknowledgements

Table of Contents

Table of Figures

Table of Tables

Table of Drawings

Table of Listings

1 Introduction

Researchers in the field of robotics engineering are constantly striving to improve the way in which humans interact with robots. Modern animatronics typically have a set sequence of poses and motions programmed into them and their bodies are typically rigid and therefore not appropriate for safe, direct interaction with humans. The field of robotics can bring in the aspect of sensing and reacting to animatronics while soft robotic models allow compliant and organic-like structures to be fabricated. The goal of this project was to create a bio-inspired interactive dinosaur. The team developed Perry the *Deinonychus* during the summer of 2020 through May of 2021.

As a child, one of the team members, Sam, wanted to build a robotic dinosaur. Every year her town would have a Christmas parade where people and organizations from the local area would create floats and enter their unique vehicles for a holiday celebration. The parade would contain dancers, vintage cars, and handmade floats. Sam attended this parade every year and when she became old enough to drive, she got an off-road vehicle and imagined driving her car in the parade with its own prop dinosaur. As Sam grew up and found out she was interested in engineering, and particularly robotics engineering, she realized that a robotics degree along with her creative hobbies would provide her with the ability to create a robotic dinosaur.

Worcester Polytechnic Institute offers a Social Implications of Robotics course that focuses heavily on human robot interaction (HRI). In this course, students are faced with different theoretical situations where humans rely on robots to drive them to work, administer them medicine, or take care of children. The class discusses ethics and proposed situations in many fields that use robotics. This class exposed Sam to HRI, the concepts implications, and the affects a robot can have on an individual. Robots have potential to entertain, interact, and educate the public with their existence. The team hopes that their dinosaur robot could end up as an interactive exhibit in a museum one day. Additionally, with the advancements made in fluid robotic systems in the past two decades, the team saw possibility in replicating organic movements with their dinosaur. Continuum arms, such as the one discussed in The Elephant Trunk's manipulator have been studied for their fluid shapes and properties for over twenty years [1]. With this structure for the dinosaur neck, powered by the smooth flow of a hydraulic system, the team believed that they could create an installation that would allow someone to feel as though they were interacting with a member of a species that went extinct over 100 million years ago [2].

Humans have a long history of creating static and moving figures for entertainment. Before electronic figures were implemented and plastic was readily available, much of America relied on taxidermy for museum displays. Then, in the early 1960s, Walt Disney and his engineers began creating the first animatronics. In 1963 he invented the Tiki Birds, which were the first animatronics to appear in Disneyland. That same year, the Abraham Lincoln animatronic became the first human animatronic figure. In 1964, the first animatronic to be used in a film appeared in Mary Poppins and in 1993 the animatronic *Tyrannosaurus rex* became the largest animatronic to ever be created for film. Other famous animatronics include E.T., the shark from Jaws, and The Terminator [3]. Animatronics, specifically those used in Jurassic Park, were a large inspiration for the project. Disney's most advanced animatronic to date is the Na'vi Shaman, which has 81 degrees of freedom, 42 of which are in the head. This animatronic took several years to develop and is fully capable of fluid movements [4].

Humanity creates animatronics to imitate life or represent a character for entertainment and typically lack the sensors required for them to be labelled as a robot. A robot can sense and perceive and affect its environment. Animatronics are given a set sequence of motions and do not act upon their environment. While the project was inspired by the creativity and purpose of animatronics, it needed to be capable of more.

The team chose the *Deinonychus* because they believed that the most well-known dinosaurs are the ones displayed in the Jurassic Park Movie Franchise. The team needed a dinosaur that was large enough to be assembled easily and small enough that the project was feasible. The team also wanted to be able to demonstrate flexibility with their model, which removed stiff-necked dinosaurs, such as the *Triceratops,* from the possible options. The team assumed that working with a well-known species would be beneficial for interacting with the public. One of the most popular dinosaurs displayed in the Jurassic Park Movies are the *Velociraptors.* These dinosaurs the franchise calls *Velociraptors* do not match any one species, so the team decided to find a real therapod analog for these movie-famous creatures. These size specifications and movie inspiration lead the team to the *Deinonychus.*

The team researched similar projects and found a collection of relevant work. Information regarding the continuum arm was found in The Elephant Trunks Manipulator paper as well as the fluid elastomer manipulator paper [1,5]. The Kronos project had similar interactive goals but did not include the fluid structure that the team was looking to implement in their dinosaur [6,7].

Finally, the feasibility of the project was assessed. Originally, Sam had proposed the development of a full dinosaur, but due to the time constraints and the limited team members, the team decided to focus only on the actuation of the head in neck. This design would allow for all the actuation to occur externally in a hidden room while foliage could be placed under the neck to make the robot appear to be in a jungle and disguise the fact that the robot had no body.

The team reached out to various paleontologists and other contacts during the beginning stages of the project. Many of these communications resulted in receiving sources but no direct interactions, though there were a few contacts that played a more prevalent part in the team's project.

Martin Christainsen, Collections Management Specialist: One of the team members met Martin at the Worcester Polytechnic Institute career fair in 2020. Martin is the Collections Management Specialist of the Worcester, MA, Ecotarium. The Ecotarium contains many exhibits about science and displays several dinosaur skeletons. Martin gave the team advice on the structures of bird necks, described anatomical features, and shared the Ecotarium's *Coelophysis* skeleton with the team.

The team also met virtually with Patrick Cupp. Patrick had experience making skin for animatronics and the team was able to ask him questions regarding different methods of creating synthetic skin. The team wanted an inexpensive, lightweight but detailed solution. It was advised that the typical latex skins can be very heavy and break down over time. Additionally, the standard animatronic latex skin requires a sculpted design that is then casted and molded. Due to the time and budget restraint of the project, Patrick recommended a fabric solution. Fabric skins are frequently implemented in larger scale performances where performers are puppeteering a large robot. Based on this interview with Patrick, the team decided that the best solution was to use a large sheet of fabric, use liquid latex to give it a skin texture, and airbrush for a final dinosaur-like appearance.

2 Background & Goals

The team began by conducting relevant research. The research topics included *Deinonychus* head and neck anatomy, general skeletal structures, Popovic Lab's Hydro Muscles & CRFC valves, and Arduino networking techniques. Some related work was reviewed, which included the Kronos animatronic and the Elephant Trunk Manipulator [1,6,7].

2.1 Understanding *Deinonychus* Anatomy & Movement

To understand the physical structure and believed motions of the *Deinonychus,* the team researched paleontological findings, read about osteology and muscular research of the dinosaur and its analogs, and conversed with paleontologists and other experts in relevant fields.

2.1.1 The *Deinonychus*

Deinonychus antirrhopus was a theropod[1] dinosaur whose name translates to "terrible claw". It was ten feet long, five feet tall, and believed to be around 175lbs. It was alive around 100 million years ago, placing it in the late cretaceous period, and it had large forward-facing eyes, suggesting it may have hunted at night. It had a flexible, curved neck and a strong jaw that was presumed to be able to bite through bone [2].

The *Deinonychus* had backwards curving and serrated teeth, three fingers on each hand, and four toed feet. On each foot, this second toe has the five inch 'terrible claw' which sat upward while it walked on the other three toes. Its fossils were found in Montana, Utah, and Wyoming, and it had a large cranial area, suggesting this animal was very smart. The physical appearance around colors, and scales versus feathers will need to be further discussed with the group, though fossils have been found that show theropods had feathers, as can be seen in the figure below [2].

---

[1]A carnivorous dinosaur of a group whose members are typically bipedal and range from small and delicately built to very large.
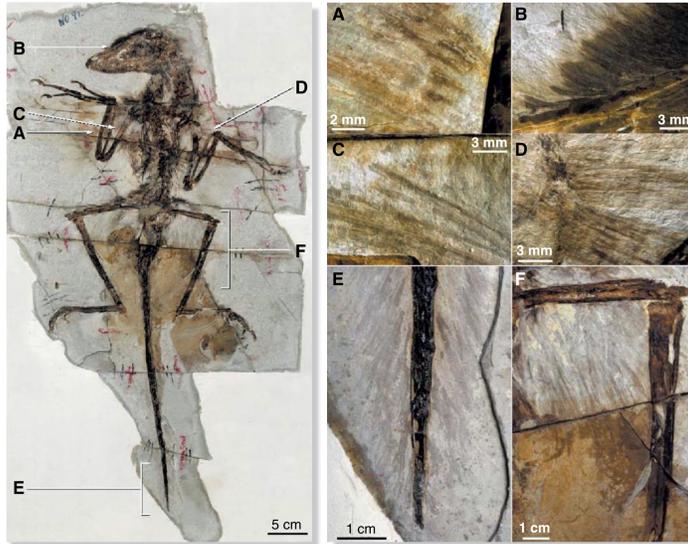
Figure 1: Feather-like structures on a small theropod fossil [8]

The team is most familiar with the dinosaur noises from the Jurassic Park movie franchise. The noises are produced by what the characters call *Velociraptors*, which more closely resemble *Deinonychus*. Dr. Julia Clark of University of Texas found the oldest avian voice box ever discovered, which suggested dinosaurs made closed-mouth booming noises much like that of an ostrich, rather than the 'reptilian scream' featured in modern media [9].

2.1.2 Skeletal Research

Birds are believed to be closest living relative to dinosaurs, and specifically that of theropods [10]. A phylogenetic tree below displays how closely related the *Deinonychus* is to the modern-day bird. This relation encouraged the team to study both dinosaur and bird osteology.
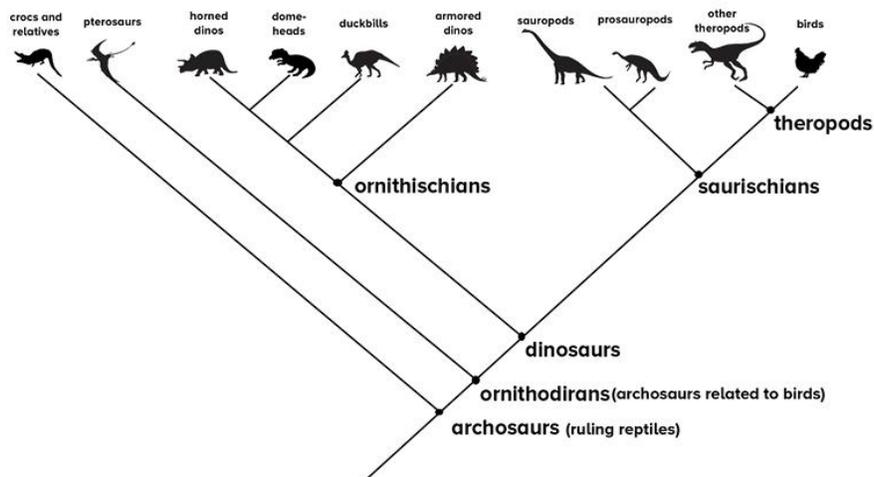
Figure 2: Phylogenetic tree by University of California Museum of Paleontology [10]

2.1.2.1 Osteology of the Head

John Ostrom estimated that the ratio of the skull length to the presacral vertebral column was unusually high, at about 0.35 - 0.40. Therefore, the length of the skull of the Deinonychus is estimated to be a little longer than a third of its spine (excluding the fused sacral vertebra at the end). Ostrom also speculated that the jaw mobility was limited. The specimens found in 1969 by John Ostrom contained a skull that was ~12.5" long and only 6" wide [11].

Greatest length of skull	320 mm
Greatest width of skull	150 mm
Greatest height of skull	115 mm
Maxillary tooth row length	130 mm
Upper tooth row length	160 mm
Orbit height	75 mm
Orbit length	?50 mm
Lateral temporal fenestra height	80 mm
Lateral temporal fenestra length	?35 mm
Principal antorbital fenestra height	60 mm
Principal antorbital fenestra length	?80 mm
Lower jaw length	310 mm
Dentary tooth row length	140 mm
Maximum lower jaw depth	?50 mm

Table 1: Estimated Deinonychus skull dimensions of specimens found by Ostrom [11]

2.1.2.2 Osteology of the Neck

The team researched ostrich neck capabilities, as it is a popular bird to study neck movements of, due to its length and size [12]. The *Deinonychus* was believed to have a much thicker neck than an ostrich due to the large neck muscle requirements for being a carnivore. The image below shows the thick muscles that were likely on a *Tyrannosaurus rex* neck, which was a larger therapod carnivore than the *Deinonychus*.
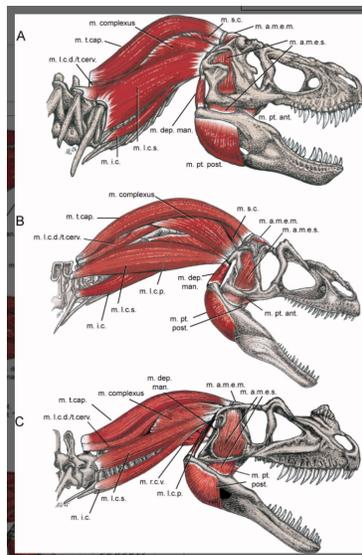
Figure 3: Neck muscles on the T. rex [13]

The *Deinonychus* had short cervical bones that act as hard stops with regards to the curvature capable by each vertebra. These cervical bones are believed to also affect the distance between muscles and bones and therefore helps dictate the thickness of the neck. Therefore, the dinosaur would not have been able to move its neck to the same extremes as the ostrich. A figure of a section of vertebra on the *Tyrannosaurus rex*, can be seen below:
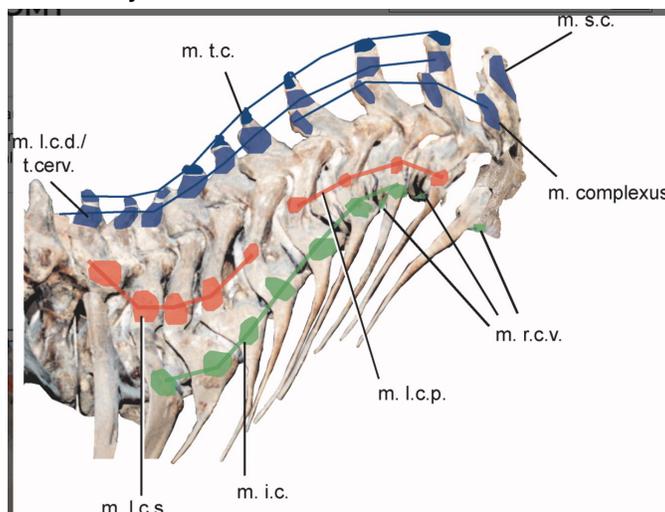
Figure 4: T. Rex vertebra [13]

Based on the partial skeleton he studied, John Ostrom estimated that Deinonychus had 21 presacral vertebrae, including eight cervical vertebrae [11]. A diagram of the cervical vertebrae placement as speculated by Ostrom can be seen below:
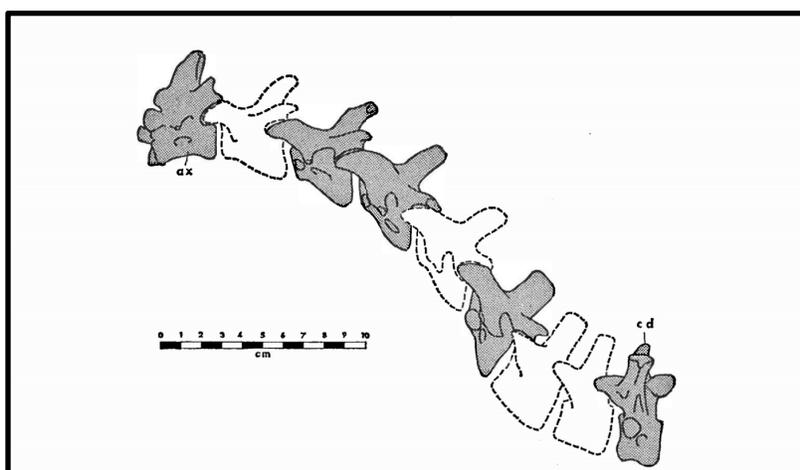
Figure 5: Diagram of Deinonychus cervical vertebrae placement [11]

Vertebrae, in general, have a complex shape. Vertebrae have a central hole for the spinal column to run through, multiple attachment points for other structures such as muscles, and change in shape as they progress up the spine. Each vertebra fits into the complex socket of the last and contains processes or prongs. A sketch from John Ostrom's findings during the initial discovery of the *Deinonychus* can be seen in the image below.
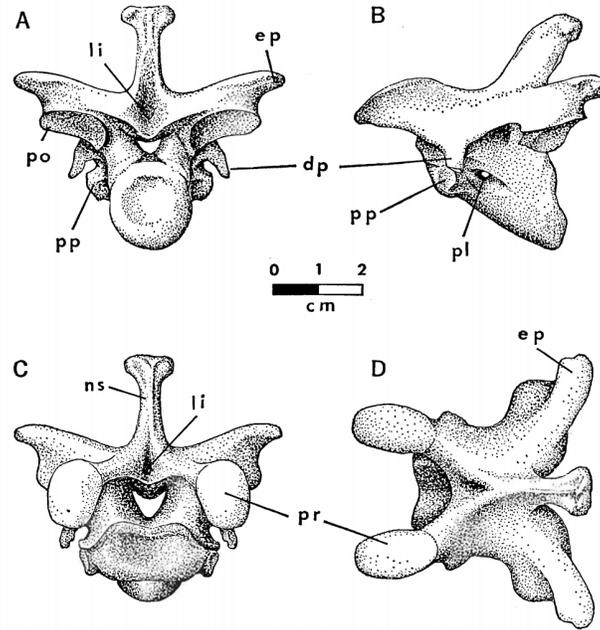
FIG. 29. Fourth (?) cervical vertebra of *Deinonychus antirrhopus*, YPM 5204, in posterior (A), left lateral (B), anterior (C) and dorsal (D) views. Notice the strong angling of the centrum and the curved zygapophyseal facets. Abbreviations: dp—diapophysis; ep—epipophysis; li—scars of interspinous ligament; ns—neural spine; pl—pleurocoel; po—postzygapophysis; pp—parapophysis; pr—prezygapophysis.

Figure 6: Deinonychus vertebrae [11]

2.1.2.3 Flexion Research

Another similar analog to the *Deinonychus* was the *Allosaurus*. *Allosaurus* was smaller than the *Tyrannosaurus rex* but larger than a *Deinonychus*. Based on the mechanical stops of the *Allosaurus* vertebra, it is estimated that the Allosaurus could move its neck in the following ways. Side to side motion is called lateral flexion, upwards motion is dorsiflexion, and downwards motion is ventral flexion. The neck was the most flexible with lateral flexion, completing a 63-degree bend from either side of the centerline. The *Allosaurus* was moderately flexible in its dorsal flexion movements with 35 degrees of upward flexion. Finally, the next was the stiffest when it came to ventral flexion, with only 15 degrees of downward motion [14].
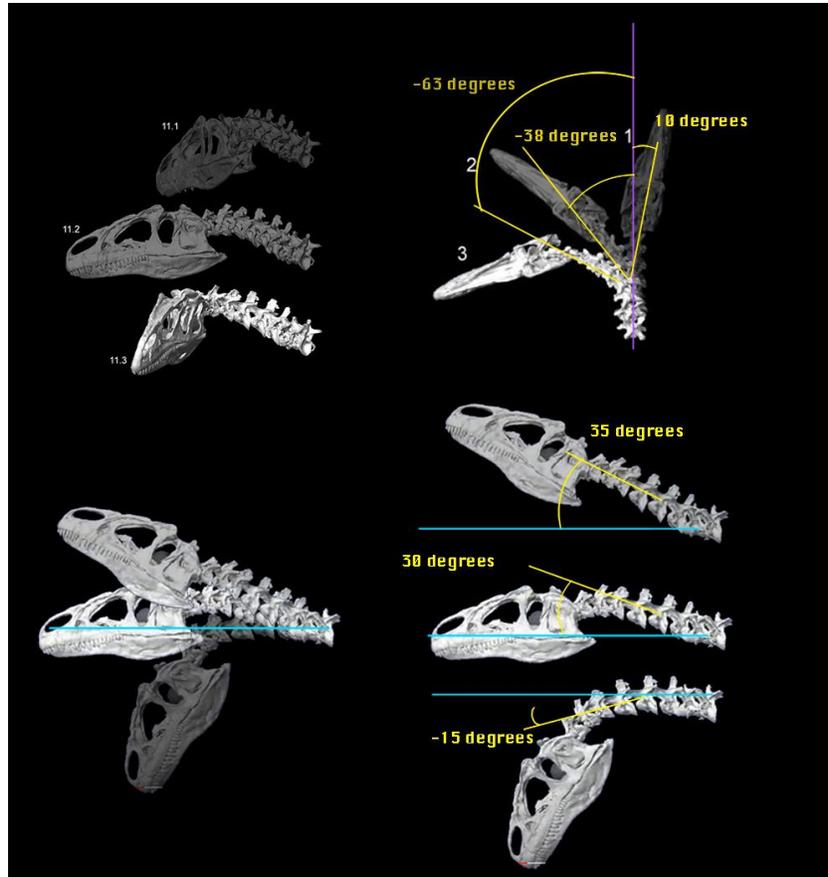
Figure 7: Modified figures of Allosaurus Neck Flexion from [14]

2.2 Actuation: Hydraulic Muscles & Valves

The team decided to focus on developing a robot capable of smooth, controlled motions in order to create a robot that gives the illusion of life. The team decided to utilize artificial hydraulic muscles that mimic the motion of biological muscles. While researching the biological structure of theropods it was noted that there was some muscle redundancy, supposedly to decrease muscle fatigue for an individual. Since muscles in nature are redundant and it was decided that the muscles could be paired into antagonistic pairs. This section will give a brief overview of the artificial muscles the team chose to use, as well as a variable control valve used to control them.

2.2.1 Hydro Muscles

Hydro Muscles are hydraulically actuated muscle analogs developed at WPI's Popovic Labs [15]. They were designed to mimic the expansion and contraction of biological muscles. Each Hydro Muscle is composed of an elastic tubing (such as latex rubber tubing) sheathed in fabric that limits the radial expansion of the muscle while allowing lengthwise elongation and contraction. One end of the tube is blocked by a stopper, while the other has a single nozzle that allows water to flow in and out of the muscle. Hydro Muscles expand when pressurized, resulting in a pushing force, and contract when pressure is released, resulting in a pulling force. Because the fabric sheathing limits radial expansion, less effort is wasted on expanding the tubing radially,

focusing the change in pressure on creating the desired pulling or pushing force of the muscle. Readers who are interested in the equations that define Hydro Muscles force output and performance should reference the original work, "Hydro Muscle –A Novel Soft Fluidic Actuator" [15].

Relaxed
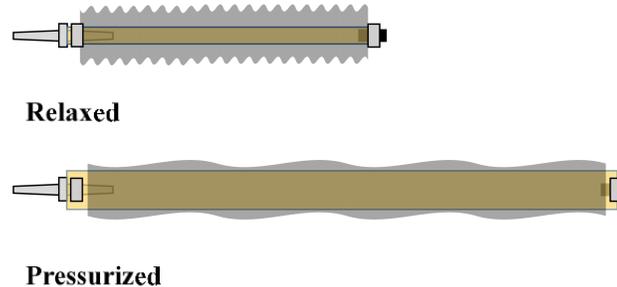
Pressurized

Figure 8: Diagram of Hydro Muscle shown in relaxed and pressurized positions [15]

Researchers developed Hydro Muscles specifically for use in biologically inspired and wearable applications, such as joint exo-musculatures and bio-inspired quadrupeds [15]. In practice, Hydro Muscle operation is limited by the type of valve used to control fluid flow into and out of the inner tubing. It is recommended that the muscles are controlled using a valve that offers continuous pressure or flow control rather than a binary open/closed design [15].

2.2.2 CRFC Valves

The Compact Robotic Flow Control (CRFC) Valve, or "Smiley Valve" is a variable flow valve developed at Popovic Labs specifically to work with Hydro Muscles. The CRFC valve features two separate tubes to direct fluid flow and a servo that is used to open and close a choke point on each of the tubes. The valves feature three main operating states: both tubes closed, left tube closed with right tube open, and left tube open with right tube closed. In either of the "open" states, the amount of flow through the open tube is variably controlled by the angle of the servo horn; the further the horn is tilted towards the open tube, the more flow that is allowed through the tube. CRFC Valves can be used with both air and water and are capable of handling pressures of over 100 PSI. Readers who are interested in the equations that define CRFC valve deflection and flow rate should reference the original work, "Development of Bioinspired Exosuit Actuated with Hydro Muscles and Novel Compact Robotic Flow Control Valve" [16].

Figure 9: CRFC valves, left closed & right open (left) and both closed (right)

When used with Hydro Muscles, one CRFC valve is used to control the flow into and out of each muscle. Both valve tubes are connected at one end to the nozzle of the Hydro Muscle using a Y connector, and the other ends of the tubes are connected to the pressurized fluid input and output reservoir, respectively.
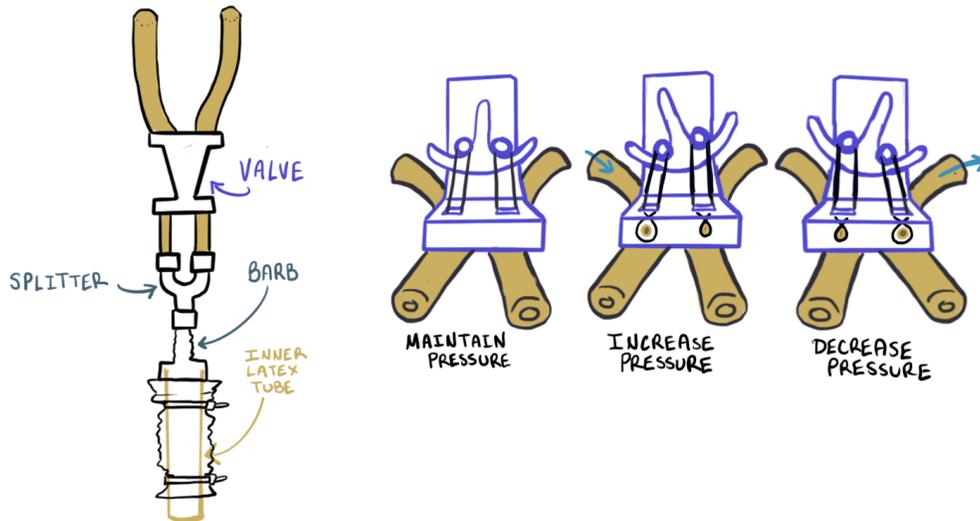
Figure 10: CRFC valve with Hydro Muscle (Left) and CRFC valve flow diagram (Right)

2.3 Arduino Networking

To maintain control over a robot, the control system may need to interface with a human or computer operator, constantly maintain a control loop for a number of actuators and calculate a representation of the physical state of the robot. A system will also need to have plenty of I/O support to be able to connect with many servos and potentiometers. Some of these requirements are conflicting in a time limited system such as robots. The calculations of the pose of a robot can involve many terms and take some time, while control loops need to be constantly maintained to frequently sample the real-world state of the system. To meet these needs, a computational

solution either must be extremely fast and robust or must be distributed. By distributing the load, a system can be expanded more easily. The Arduino Nano Every Board supports multiple standards of fast serial communication including I2C, Serial, and SPI. These methods vary in terms of physical hardware, latency, and ease of implementation and may be convenient for a given use case.

2.3.1 I2C

The Arduino framework exposes access to the I2C standard using the wire.h library. This library provides a means to create simple serial connections between multiple devices. The library has simple wiring requirements that the Arduino Nano Every must use. Pin A5 is used as (Serial Clock) SCL, pin A4 is used as (Serial Data) SDA, and the Arduinos must share a common ground. The network establishes a primary/secondary relationship between one primary Arduino and many secondary Arduinos. The primary Arduino controls the serial clock speed, which coordinates messaging. The clock speed is set to 100000 Hz by default and the Arduino Avery can be set up to 400000 Hz. The I2C standard supports 7-bit addresses [17]. On the Arduino the first 8 bytes are reserved allowing for up to 120 devices to be connected. Messages over I2C are limited 32 bytes, which is the maximum for each message sent [18]. Each message starts with an 8-bit address sequence, where the first 7 bits are the address, and the last bit corresponds to read/write. After this sequence, the primary Arduino passes control to the secondary Arduino to receive ACK/NACK on the address sequence. Assuming ACK is received by the primary Arduino, the Arduino specified by the read/write bit then sends the message in sets of 8 bytes of data, up to the 32-byte buffer [17]. This completes the transfer of data over I2C.

2.3.2 Serial

Arduino provides serial communication over USB using the Serial library. This library allows the Arduino to establish a connection to another device with a predetermined baud rate (bits / second). Both the Arduino and the computer may read and write over serial. The size of the serial buffer is 64 bytes. Data is sent sequentially over serial and may incur a long time delay, so printing to serial often is not recommended during time sensitive operations as it may block. Serial is used commonly with Arduino for debugging purposes in the Arduino IDE serial monitor and the python library pyserial is available for interacting with Arduino via python script.

2.4 Denavit–Hartenberg Parameters

One approach to creating a computational model of a robot's physical position is through Denavit–Hartenberg parameters. The Denavit–Hartenberg parameters, or DH parameters, is a standard way of creating a model of a robot with rigid links so that the forward and inverse kinematic models can be developed. In this section one can learn about the standard process for calculating a robot's DH parameters. The team used a modified version of the DH parameters since the standard process only works for rigid links.

The standard DH parameter process begins by identifying the types of joints present on a robot. Typically, the joints fall into two categories: prismatic (linear) or revolute (rotational). The direction of positive rotation or motion is defined. Next, a Z axis is defined at each joint along the

respective joint's positive axis of rotation or translation. The X axis of each coordinate frame is assigned so that it is along the common normal. The common normal is defined as the direction that is perpendicular to both the current joint's Z axis and the next joint's Z axis. Some special cases for applying the X axis include: If adjacent Z axes are parallel, then the X axis is applied to the plane between them, and normally along the link armature. If the adjacent Z axes intersect, the X axis is perpendicular to the plane between the two Z axes. Finally, the Y axes are applied according to the right-hand rule.

Now that the model and its coordinate axes are set, the DH table is created. There are different conventions for the organization of the table, but a table with the first row being, θ, d, a, α is recommended, as seen in the figure below. The number of rows depends on the number of joints there are in the robot being modelled. In the example table below, there are three rows, so the table is for a three degree of freedom robot. Each row corresponds to a coordinate frame created previously. The table is then filled in by the following process. It is recommended that the table be filled out row by row by completely defining all four parameters for a coordinate frame before moving onto the next frame.

θ: Theta is first evaluated. θ is the angle about the current coordinate frame's Z to line up the current X axis with the next joints X axis. θ is written in the table with an asterisk (*) to represent that it is a variable when the current joint is a revolute joint. If the current joint is a prismatic joint, then θ is zero.

d: Next the d parameter is evaluated. d is the translation about the current coordinate frame's Z axis to reach the next coordinate frame. This value is a constant for revolute joints and a variable with an asterisk for prismatic joints. The unit is a measurement of length.

a: a is then applied. a is the translation along the intermediate X axis (after possibly rotating by θ) to reach the next axis. This value is typically a constant for both revolute and prismatic joints. The unit is a measurement of length.

α: α is the last parameter to be applied. α is the rotation about the intermediate X to line up the current coordinate frame's Z with the next coordinate frame's Z. This value is typically a constant in radians for both revolute and prismatic joints.

The resultant DH table's values are then entered into standard transformation matrices and the forward kinematics matrix can be generated. Readers interested in a more in depth process can find many examples and processes explained online. The modified process for DH parameters based on the Elephant Trunk's Manipulator paper can be found in the following section.

θ	d	a	α	θ
θ1*	3	1	π/2	θ1*
θ2*	4	1	0	θ2*
0	4	1	0	0

Table 2: Exemplary DH table

2.5 Related Work

The team explored a few relevant topics and papers before designing their robot. These topics included Hydro Muscle exosuits, animatronics, interactive creature projects, and a paper that detailed the mathematics of controlling and modeling a robotic continuum arm.

2.5.1 Bioinspired Exosuit

Hydro Muscles were used in the development of a bioinspired exosuit created at Popovic Labs. The exosuit was meant to be used for therapy and assistance of human leg motion, specifically during the walk cycle. The system was actuated by six Hydro Muscles on each leg; the Hydro Muscle location and function were chosen so that they would act as analogs for biological leg muscles. The Hydro Muscles used in the exosuit were actuated using compressed air supplied at 100 PSI. The exosuit also utilized SpiderWire tendons attached to the Hydro Muscles to create the desired motion. The tendons were attached to eyelets located at approximate muscle origin locations. On/off solenoid valves were used in the pneumatic system to control air flow to and from the Hydro Muscles [16].

The exosuit was implemented to actuate a biomimetic skeleton through a six-state walking gait on a treadmill moving at constant speed. The movements of the skeleton were recorded, and the resulting postures were found to be comparable to biological walking postures. The skeleton actuated by the exosuit was able to stand and walk on its own with only light tethering. Additionally, the creators of the exosuit found that the binary nature of the solenoid valves limited the smoothness of the Hydro Muscle operation and resulting exosuit motions. This inspired the development of a cost-effective valve capable of providing a finer level of fluid control, leading to the creation of the CRFC valve [16].

2.5.2 Kronos

Brian Burns from Georgia Southern University created the Kronos dragon. The dragon animatronic laid on a table and would lift its head to follow someone if they entered the room. The tail moved in one repetitive motion, the body was stationary, and the neck moved so that the nose of the dragon was always pointed at the user. It would occasionally move its wing and growl. The video shows a camera external to the animatronic following the user around the room. The skin was rough and looked like polymer clay or foam [6]. During the project, an updated video was posted revealing the beginning stages of the Kronos III Dragon project. The new robot had a highly detailed 3D printed head, neck, and left foot and a metal frame underneath the robot to house electronics. The animatronic's head was able to be controlled with a video game controller and appeared to be actuated with motors & servos. At the time of the video the animatronic was unfinished [7].

2.5.3 Jurassic Park *T. rex.*

The team needed to research both mechanical and artistic methods to create the dinosaur robot, as a skin was required for the final robot to appear realistic. The team researched how the dinosaur models were developed for the Jurassic Park movie franchise. The original Jurassic

Park movie mostly used animatronics, and only used CGI for some dinosaurs due to budget constraints. The *Tyrannosaurus rex* was initially sculpted completely out of clay at 1/5th size. The creators scaled the model by projecting an image of the small sculpture at full size and used the projection as a template to cut the wooden pieces for the full-sized figure. The foam and wooden slices were then attached to the metal frame of the full-sized model. The wood was used as support for a layer of fiberglass that was then covered with clay. Gauge sticks were used to help differentiate between sections that should have thicker or thinner skin. Artists then spent three months sculpting dinosaur skin texture into the clay. The resulting dinosaur would go on to hold the record for the largest animatronic featured in a film to date [3,19].

2.5.4 Elephant Trunk Manipulator Kinematics

Hyper-redundant continuum robots have become more common in recent decades and can be implemented as surgical probes, as well as search and rescue. While more traditional arms consist of electro-mechanically actuated joints connected via rigid members, continuum robots have members that actuate themselves by inducing a constant curvature along its length. A model for a continuum robot was described in 2002 at Clemson University South Carolina and used to control a robotic Elephants trunk [1]. This trunk was composed of vertebral disks connected with springs. Sets of disks were divided into subsections and actuated by cables attached to servo motors. This robot had many degrees of freedom, but the computational complexity of kinematically modeling the robot's position was reduced by using the geometry of a continuous curve to develop a modified Denavit-Hartenberg parameter table. By controlling the length of the cables, the robot was able to take shapes to grasp different objects and navigate a workspace.

The relationship between a spatial curve's deflection and the spatial vector between its start and end point can be determined geometrically and is necessary to generate a modified DH parameter table. The calculation can be simplified by first considering a planar curve as shown in figure 11.
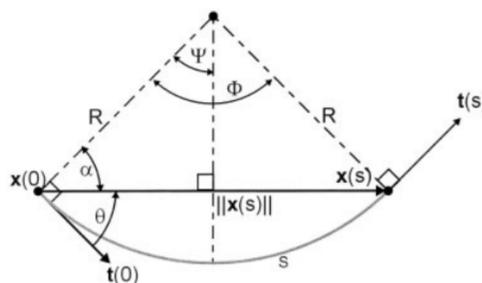
Figure 11: Geometric representation of a planar curve [1]

The relationship between the deflection (ks) and the vector $\|x(s)\|$ can be found via geometric relationships as shown below in Table 3.

Φ = s / R = ks
Ψ = Φ/2 = ks / 2
P=Ψ= ks / 2
$\|x(s)\|$ = (2 / k) * sin(**P**)
$\|x(s)\|$ = (2 / **P**) * sin(**P**)

Table 3: Geometric Relationships to solve planar curve position vector [1]

This curve will provide the necessary position vector relationship between the start of a curve and any point along its length s. Modified DH frames for a planar robot section forward kinematics can be created assuming it maintains constant curvature. The geometric relationship can be used in this way to create DH frames for a planar robot in Figure 12.
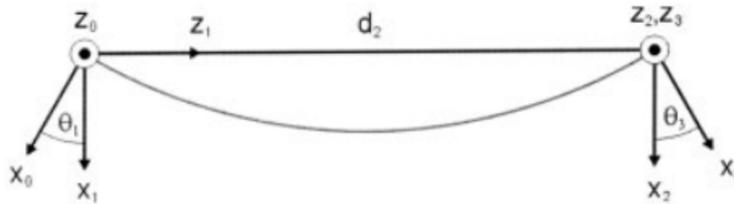
Figure 12: DH frames for a planar robot [1]

These frames can be simply extended to describe a spatial robot section by adding an additional rotation (P_1) to bring the curve out of plane and into 3D space.
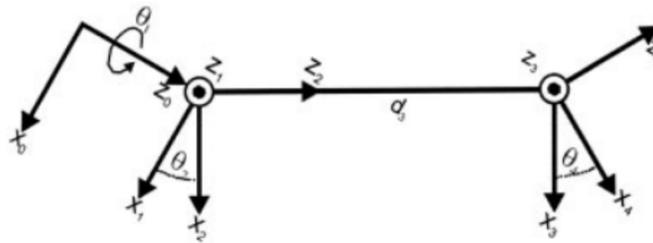
Figure 13:DH frames for a spatial robot [1]

The D-H frames for a spatial robot can be described with the following D-H Table.

link	θ	d	a	α
1	*	0	0	90°
2	*	0	0	$-90°$
3	0	*	0	90°
4	*	0	0	$-90°$

Table 4: DH frames for a spatial robot [1]

Finally, the homogeneous transformation matrix for a continuum robot section assuming ($P_2 = P_4$), the section experiences curvature k, arc length l, and initial rotation angle P_1 [1].

$$A_0^4 = \begin{bmatrix} \cos(\phi)\cos(\kappa l) & -\sin(\phi) & -\cos(\phi)\sin(\kappa l) & -\frac{1}{\kappa}\cos(\phi) + \frac{1}{\kappa}\cos(\phi)\cos(\kappa l) \\ \sin(\phi)\sin(\kappa l) & \cos(\phi) & -\sin(\phi)\sin(\kappa d) & -\frac{1}{\kappa}\sin(\phi) + \frac{1}{\kappa}\sin(\phi)\cos(\kappa l) \\ \sin(\kappa l) & 0 & \cos(\kappa l) & \frac{1}{\kappa}\sin(\kappa l) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Table 5: Homogeneous transform matrix for a spatial robot [1]

2.5.5 Spatial Fluidic Elastomer

The Design, kinematics, and control of a soft spatial fluidic elastomer manipulator paper by Andrew D. Marchese and Daniela Rus demonstrates a robotic manipulation system with a soft continuum arm with many segments. This paper was written in 2016 and the authors explicitly state that their robot had many novel qualities. These novel design elements were that the robot was fabricated in 100% soft silicone rubber, that the robot was powered by four elastomer actuators on each of the three segments, and that the system was "designed with a modular morphology suitable for automation". The authors also stated that their repeatable fabrication methods and deformation and motion models were novel. Finally, the team stated that their evaluations for the simulations and physical experiments for an autonomous three-dimensional fluidic elastomer manipulator were novel. This paper had a high degree of novelty and contained relevant information regarding the modeling and usage of a continuum arm [5].

2.6 Project Goals

The team decided on goals before beginning the design process of the project. These goals are summarized as follows. A more in-depth explanation of the team's goals can be found in Appendix A.

The focus of the project was approximating the head and neck movement of the *Deinonychus* as determined by the range of vertebral flexion found in two analogs, with minimum and maximum flexion determined by the *Allosaurus* and the common bird.

This project aims to create the impression of a living *Deinonychus* via synthetic skin and audio based on current scientific knowledge and the public perception of the *Deinonychus*. The robot should also feature two distinct interactive modes. The first is a kiosk mode which performs ambient actions and sounds. When a spectator is detected, the robot reacts to visitor movements by following the face of the spectator, creating the illusion that the robot is "watching" the spectator, and emitting audio. The second is a grab mode during which the robot will locate and grab onto a ~ 3-inch diameter object in the robot's workspace.

2.6.1 Mechanical Goals

The team wanted the robot to contain a vertebral column actuated by a system of Hydro Muscles. The vertebral column should be capable of specified ranges of motion described in detail in Appendix A. The head and neck should be able to achieve operating speeds up to 15 in/sec and maintain spatial accuracy of +/- 1 inch from the target end effector (head) location and rotational accuracy of +/- 10 degrees from the target angle. The head should include a mobile jaw capable of grasping a ~3-inch diameter ball and have eyes capable of simple random or repetitive movement.

2.6.2 Electrical Goals

The team wanted to implement a multiple-sensor suite to collect position and environmental information to enable autonomous functionality. This suite should include a microphone, camera(s), inertial measurement units, a speaker, and any other sensors need to complete the general goals. The electrical system should support one or more microcontrollers to control key sensory activities and contain a robust control system for coordinating artificial muscles, servos and support the system power needs. The final electrical schematic should be transferred to custom printed circuit boards.

2.6.3 Software Goals

The low-level control system should contain an API for controlling Hydro Muscles and provide control over the length and force of each Hydro Muscle, measurements about the Hydro Muscles such as their maximum pull length, the current pull length, and the current pull force for each muscle via a sensor suite. Muscles will be able to be grouped into antagonistic pairs which will restrict how they can move relative to each other. The mid-level control system should contain an API for controlling and coordinating neck sub-assemblies and the head functional group will consist of an articulable jaw, eyes, and sensors to record position. The API will provide controls for changing the state of the subassemblies from one to another over a given time interval as well as for obtaining the current state for the subassemblies. To appear responsive, the robot should be able to respond to higher level commands within half a second. The high-level control system should convert camera data into actions for the *Deinonychus* to execute. These actions will fall under two modes that are described in the interactivity goals.

2.6.4 Interactivity Goals

Two distinct interactive modes should be featured by the robot. The first mode is the kiosk mode where the robot performs different actions and sounds depending on the distance between the robot and the user. The second mode is the grab mode in which the robot should detect an object and move to grasp the object within its jaws. The robot should feature synthetic skin based on current scientific knowledge and the public perception of the *Deinonychus*. The robot's movements should incorporate the methods of overlapping actions, follow through, and ambient Motion [20].

3 Methodology

The process of design, redesign, fabrication, and part assembly played a large role in this project. The team designed and fully fabricated the vertebra, spine, muscle boxes, skull, hydraulic system, and networking systems. The kinematics control system was implemented from the Elephant's Trunk Manipulator paper [1].

3.1 Vertebrae Development

The team's vertebrae design saw nine iterations, each one built to correct the problems with its predecessor. The team began each prototype design in SolidWorks, then 3D printed the prototypes to test out in the lab. Various observations were made to compare the different prototypes. The team decided that vertebrae prototype nine had the most potential and it was easily modified with one dimension to adapt to the required flexion needed between certain points in the spine.

3.1.1 Vertebrae Requirements

The team recognized that it would be difficult to determine accurate ranges of motion of the head and neck of *Deinonychus* without the ability to study a muscled specimen. To overcome this complication, the team decided to study similar bird species as a general reference of joint capabilities and movements. Depending on the species, birds have from 9-25 cervical vertebrae, with 14-15 being most common. Avian vertebrae are typically saddle-shaped, which enable dorsal/ventral flexion and lateral flexion but resist axial rotation [21].

A study of the range of motion of turkey necks was performed by Kambic, Biewener, and Pierce in 2017. The study aimed to better quantify the range of motion of between different pairs of the cervical vertebrae, encompassing vertebrae C3-C10. The research study identified three distinct sections of vertebrae with similar motion characteristics [21]:

1. Vertebrae C3-C5
   a. High dorsal/ventral flexion, favoring dorsal flexion
   b. High lateral flexion
   c. High axial rotation
2. Vertebrae C5-C7
   a. High dorsal/ventral flexion, slightly favoring dorsal flexion
   b. Fair amount of lateral flexion
   c. Little to no axial rotation
3. Vertebrae C8-C10
   a. High dorsal/ventral flexion, favoring ventral flexion
   b. High lateral flexion
   c. Little to no axial rotation

The specific ranges of motion found between each pair of vertebrae in each direction are summarized in the following graph:
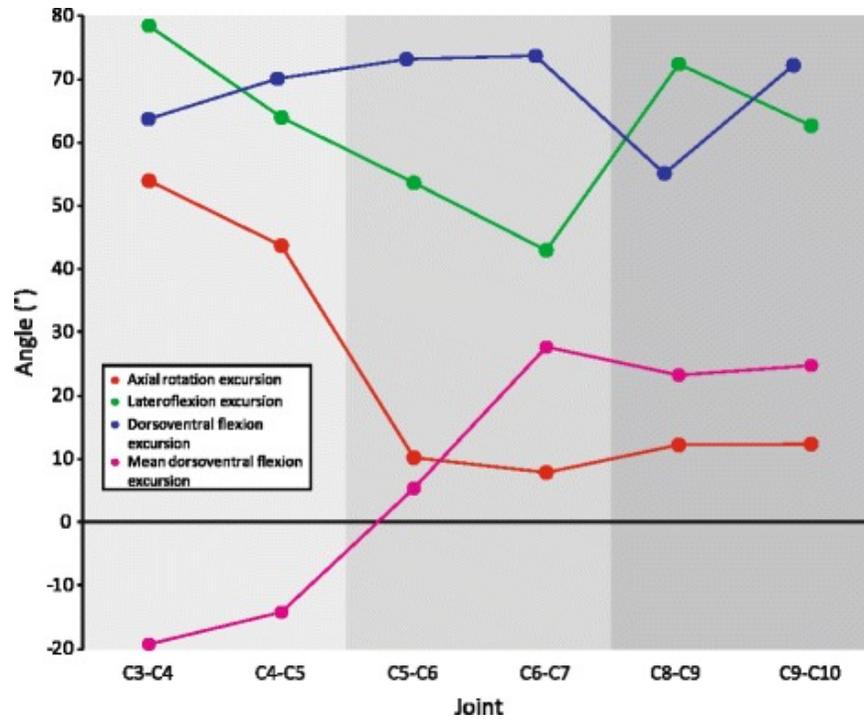
Figure 14: Graph of turkey cervical vertebrae ranges of motion [21]

These findings indicate that turkey and similar avian necks exhibit a high amount of mobility near the skull joint, with somewhat stiffer vertebrae closer to the body. The team plans to use these findings as a starting point for developing vertebrae structure and mobility for the robotic *Deinonychus*.

3.1.2 Prototypes

When discussing vertebrae prototypes, a letter convention will be used to differentiate the location of each vertebrae in the partial vertebral column. This convention states that the top vertebra will be called "A", the one attached below that will be referred to as "B," and so on.

The team initially began their design and assessment of vertebra models by utilizing an on-hand set of coyote vertebrae. The team used the lower three thoracic vertebrae of the spine to perform rough initial movement exploration and tendon-based actuation. The team strung fishing line through ¼" Bowden tubing with an 1/8" inner diameter. Fishing line was tied to one vertebra, and the Bowden tube was independently tied to the vertebra below. The attachment of the Bowden tube was done by making a hole perpendicular to the tube through both sides and looping string through that hole and tying it to the bone. This created a system where the top of the uppermost vertebra had the fishing line tied to it, which was then guided down the length of this first vertebra. The following vertebra had the top of a 5" piece of Bowden tube tied to it in which the first vertebra's fishing line was thread through until it came out of the base of the Bowden tube. This setup was mirrored on the opposite side to simulate analog antagonistic tendon pairs. When pulled, the top vertebra would swivel on the ball and socket joint between it and the lower one. Only limited movement was achieved, and this was attributed to a small moment arm for the string attachment. The distance between the center of the vertebra and where

the string was attached was less than an inch, requiring the actuator to use greater forces for limited movement.

3.1.2.1 Vertebra Design 1
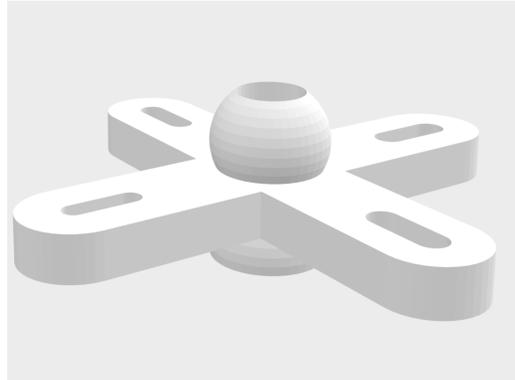
Figure 15: SolidWorks model of vertebra prototype 1

Vertebra design 1 was 3D printed in PLA on September 1st of 2020. The design goal was to create something that was simple to manufacture and could be actuated with 2 rotational degrees of freedom (DOF). The design consisted of a ball and socket joint connection between vertebra, a central channel for a spinal cord analog, and four prongs in the shape of a plus sign. Each prong had a small slot in which fishing line could be fed through or tied to. The team initially 3D printed three of these vertebrae at about 3'' wide and fed a length of Bowden tubing through the central channel to give the structure some flexible rigidity. The team tested three methods to connect and control the antagonistic pairs of the vertebra arms. In Method 1, the fishing line was tied to the hole in one arm of the top vertebra (A) and fed through the holes in the arms on the same side of the two vertebrae below (B and C). The same tying method was completed on the opposite arm. Pulling on these strings resulted in minor motion as the angle of flexion was limited by the vertebra arms hitting those of the adjacent vertebra. Method 2 had the string tied to the holes of opposite arms of vertebrae (A & B) and freely threaded through the hole in the arms of vertebrae (C). This produced a more limited motion because the top vertebra could only bend so far before the knot tied to the vertebra below it limited the motion. Method 3 used the same set up as Method 1, but each vertebra was tied to the one below it with a ~ ¾" strip of fishing line to create mechanical stops. This method greatly reduced the flexibility of the motion. The ratio of vertebra height vs length, or the apparent 'flatness' of this design, was also observed to restrict movement, as the long arms (as compared to the short distance between ball and socket joints) created sudden mechanical stops, shortening the flexibility of the three-vertebra structure.

3.1.2.2 Vertebra Design 2
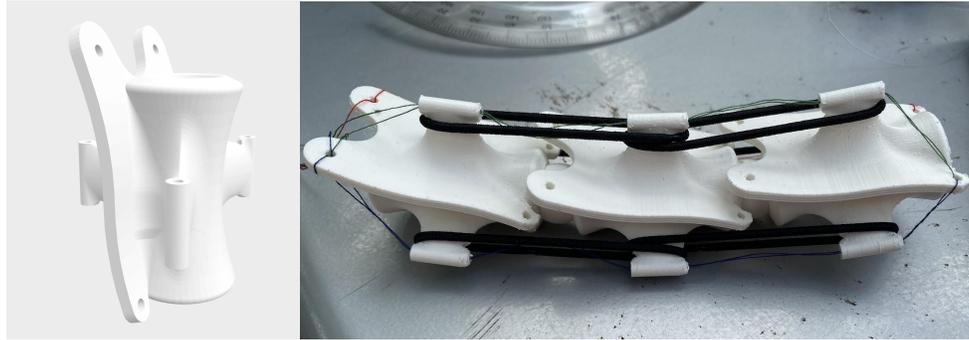
Figure 16: SolidWorks model of vertebra 2 (left), Three assembled vertebra 2 (right)

Vertebra design 2 was inspired by the organic shapes found in actual vertebrae and attempted to include curved mechanical stops built into the actual vertebra rather than being done with string. The team created design 2 to use elastics to hold a series of the vertebrae together rather than incorporating a central channel.

The vertebrae were 3D printed with PLA and a high layer count to increase surface smoothness and therefore reduce friction between the 3D printed parts. Between the vertebrae shapes were 3D printed balls to fill the ball and socket joint and allow pivoting between vertebrae. Strings were also utilized to control vertebrae movement, like the fishing wire in design 1. The team determined that the vertebrae were able to move too far laterally and the subsequent vertebra design 3 was designed with more intentional and limiting mechanical stops.

3.1.2.3 Vertebra Design 3
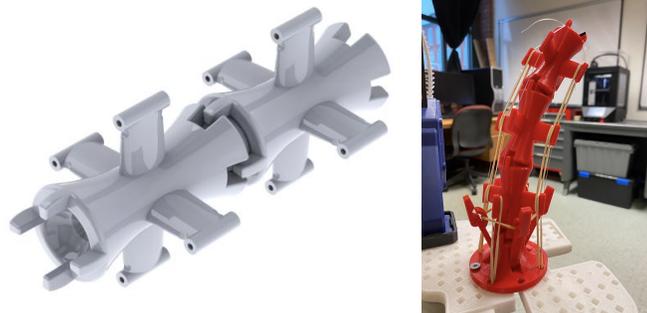
Figure 17: SolidWorks model of vertebra 3 (left), Three assembled vertebra 3 (right)

Vertebra design 3 combined the organic shape of design 2 with the mechanical functionality of design 1. Vertebra of design 3 had a similar length-to-height ratio of real vertebra and that of design 2 but contained 1-inch arms which allowed for the favorable moment arm seen in design 1. Additionally, the team utilized elastics between arms to translate forces between vertebra and create a more distributed movement profile. The elastics also created a 'standing position' of the tower when no actuation was present. The downside of including the passive elastics was that the system was now 'fighting itself' as the elastics increased friction and created forces that opposed actuation in the opposite direction.

Between each vertebra, a 3D printed sphere created a ball-and-socket joint while prevalent mechanical stops prevented each respective vertebra from rotating too far in any direction. When strung up with a thin, elastic, rubber tubing running through the spinal column, elastics between arms, and fishing line through the arm holes, the team was able to puppeteer a three-vertebra spine with favorable results. Observations from this actuation include an evenly distributed curvature and smooth movement when the fishing line was pulled. The three vertebrae strung together were able to produce 52 degrees of ventral, dorsal, and lateral flexion and were not capable of axial rotation.

3.1.2.4 Vertebra Design 4

Figure 18: SolidWorks model of Vertebra prototype 4

With each of the first three designs, there was the issue of fitting Bowden tubes through the center of each vertebra because of the space limitations of fitting tubing within a ball and socket joint. Vertebrae design 4 replaced the ball and socket joint by utilizing a joint shape like an interlocking chain joined by a ring at the ends. Vertebra design 4 removed the need for Bowden tubes by containing channels through the length of each vertebra, which were functionally the same as if the design used tubing, but without the weight and space taken up by rubber tubes. Design 4 was capable of 68 degrees of ventral, dorsal, and lateral flexion and was not capable of axial rotation. The draw length between neutral stance to full actuation was 3.25 inches which was much longer than found in previous designs.

Figure 19: Three assembled vertebra prototype 4 (right)

When design 4 was rigged for manual testing, the team found that the movement was disjointed, where one vertebra would 'clunk' to a position, and then the next and so on, rather than producing a fluid motion. Additionally, it was found that the order in which the strings were pulled affected the subsequent shape of the vertebra. For example, if string 1 was pulled before string 2, this would result in a different shape than if string 2 was pulled before string 1. This feature would make automated control more complex.

3.1.2.5 Modular Vertebra Designs 5 & 6
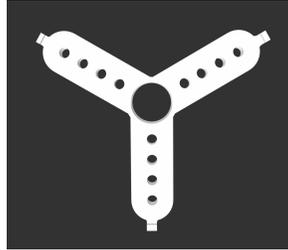
Figure 20: SolidWorks model of modular vertebra prongs

Design five and 6 made use of a modular vertebrae set the team designed and printed to be able to easily create new vertebra designs without having to wait for a 3D printer or use up resources. The modular design used a central column, end caps to represent different joint types, and arm options. The central column had a row of holes where screws could be threaded through to attach arm or end cap options. The team made two arm options: One with 3 processes and one with 4 processes, allowing the team to compare actuation between two vertebrae that were structurally the same except for the number of processes. Additionally, each arm design had holes along its individual process so that the string for actuation could be threaded through at different distances from the central channel. This design addition allowed the team to test the balance between force and torque. The longer the moment arm, i.e., the longer the distance from the central channel of the vertebra to the tie-on point, the less force would be needed to actuate the vertebra to its full extent. Having a moment arm longer than a few inches would prove excessive for this project, though, because the longer the moment arm, the longer the draw length to reach full actuation, and therefore the longer the Hydro Muscle needed. The arms also had little hooks on the tips of each process to allow for the attachment of rubber bands. The end caps allowed for different joint type options, yet the team only printed ball in socket caps for the modular design. The modular vertebra design, therefore, allowed the team to test a variety of variables: Joint type, elastics vs no elastics, 3 processes vs 4 processes, where the arms were attached in relation to the height of the central column, and force vs torque.
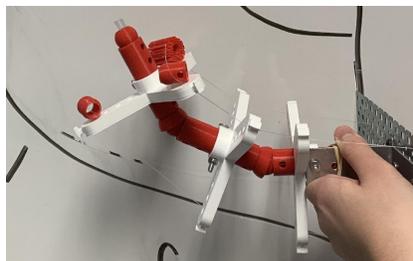
Figure 21: Three assembled vertebra prototype 5

Vertebrae design 5 used the 3-process arm with the strings through the farthest hole from the central column. The team oriented the vertebra so that two processes were on the top, and only one process was on the bottom, to mimic the orientation of actual vertebrae. The team observed that the drawing force for this design versus previous designs appeared lighter. The draw length was 2.25 inches from neutral if pulling dorsally (with the two top processes) or 1. inch if pulling ventrally. The team determined that although less processes meant less Bowden tubes and therefore less weight, the control with only three processes would be more complex because there was no longer one process for each cardinal direction.
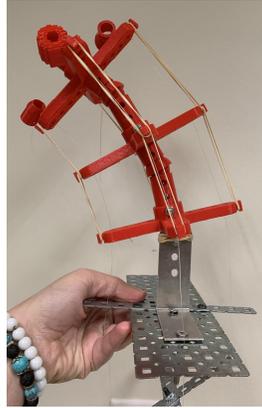
Figure 22:Three assembled vertebra prototype 6

Vertebrae design 6 used the same modular assembly as Design 5 but with a 4-process arm. The team tied the strings through the farthest hole from the central column. By observation, the team found that the force to draw to full actuation was like that found in vertebra design 3. The team decided a vertebra with four processes was the optimal choice to maintain the structure of antagonistic pairs and allow tracking in four directions.

3.1.2.6 Vertebra Designs 7, 8, and 9
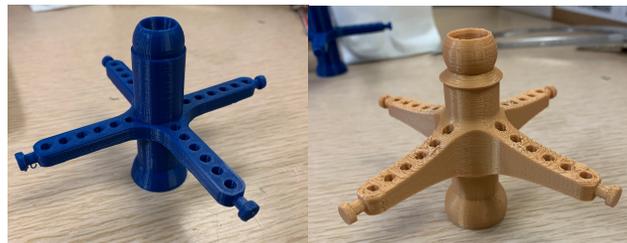
Figure 23: Vertebra prototype design 7 (left), Vertebra prototype design 8 (right)

The next two designs were iterations of Design 6, with the intention of thickening the processes to allow for more strength. Design 7, which was printed in blue, had processes that would bend if a team member leaned on the arm, and the design also lacked a reliable mechanical stop to control the degree of flexion between two vertebrae. Design 8, printed in gold, used thicker processes, fillets to strengthen sharp corners, and added a disc under the ball to limit the movement of the vertebra above.

3.1.3 Final Vertebrae Design

The final vertebra design was a further strengthened and shortened version of design 8, with larger space for elastics and a more prominent mechanical stop. The final vertebra design was used as a base model for a single modification specific to determining flexion angles between adjacent vertebrae. This modification required one dimension in the SolidWorks model to be changed to dictate the desired flexion. These deflection requirements are discussed more in the following section. This model was also used to create versions with an inertial measurement unit mount for specific vertebra.

Figure 24: SolidWorks rendering of vertebra 9 with IMU mount.

3.2 Spine Development:

The spine is constructed of ten of the ninth, and final vertebra design. To split up the spine into sections for clearer modeling, the team divided the structure into groups A, B, and C, where segment A was the top three vertebrae, segment B was the middle four vertebrae, and segment C were the bottom three vertebrae. The team modified the angle within the socket of the final vertebrae design to create desired angles of flexion between each of the ten vertebrae. Boden tubes and spider wire were attached for actuation.

3.2.1 Spinal Flexion

The diagram to the right outlines the calculations to determine the angles between each vertebra. The team decided that the socket angle would be 17.5 degrees on every vertebra and then the angle on the disc that acted as a mechanical stop would change. The socket angle is referred to as "Socket" and labelled in green in the diagram. The disc's angle is labelled in purple

and referred to as "Ball" in the diagram. The desired angle is labelled "Angle" in black. It was found that Ball - Socket = Angle. Before 3D printing the spine, the team created a model in SolidWorks with mechanical stops to make sure the physical assembly of the vertebrae with the angles chosen would produce the desired "S" shape. This model can be seen in Figure 25. Once the team was satisfied with the spinal shape, the final vertebrae were 3D printed. The team printed the vertebra so that the layer lines were parallel to the central column of the vertebra, reducing the likelihood that a prong would break due to the force of the Spiderwire. When assembling the spine, the team strung a ¼ inch flexible tube through the center to help support the structure. This tube was flexible enough that the resultant friction was considered negligible.
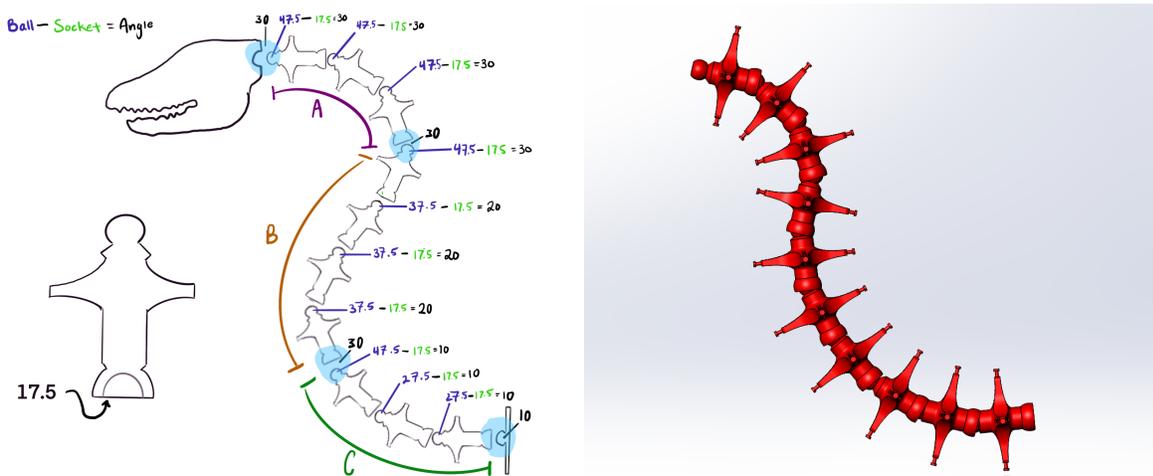
Figure 25: Flexion calculations (right), Vertebra flexion diagram in SolidWorks(right)

3.2.2 Tendons & Bowden Tubing

Pulling the Spiderwire tendons actuate the robot. These tendons are threaded through Bowden tubing. The team decided to use the outermost hole on each process of the vertebra, to make use of the longest moment arm and achieve the most bend with the least amount of force. The tendons are attached to the outermost hole of the process on the top vertebra of a section. The Spiderwire tendon is then fed through the same hole of the other vertebra of the same section and beneath the first. Once the wire is fed through the hole of the vertebra of the top of the following section, the wire is fed through Bowden tube. The tube is then loosely wound along the neck and fed through the vertebrae baseplate on the base so that it can guide the tendons to the Hydro Muscles. The Bowden tube allows only the section in which the wire is tied to be actuated, effectively leaving the sections below it unchanged. This method makes control much easier.

3.2.3 Base Design

The base was made from pieces of heavy Trex decking boards and constructed so that there is a vertical face in which the vertebrae mounting plate was attached to with screws. This vertebra mounting plate was constructed to be the same ball-in-socket joint used in the main vertebrae but attached to a large plate that could be attached to the wood with screws. This plate

was designed to allow for a stable mounting point for the rest of the spine. A rendering of this baseplate in SolidWorks can be found in Figure 26.
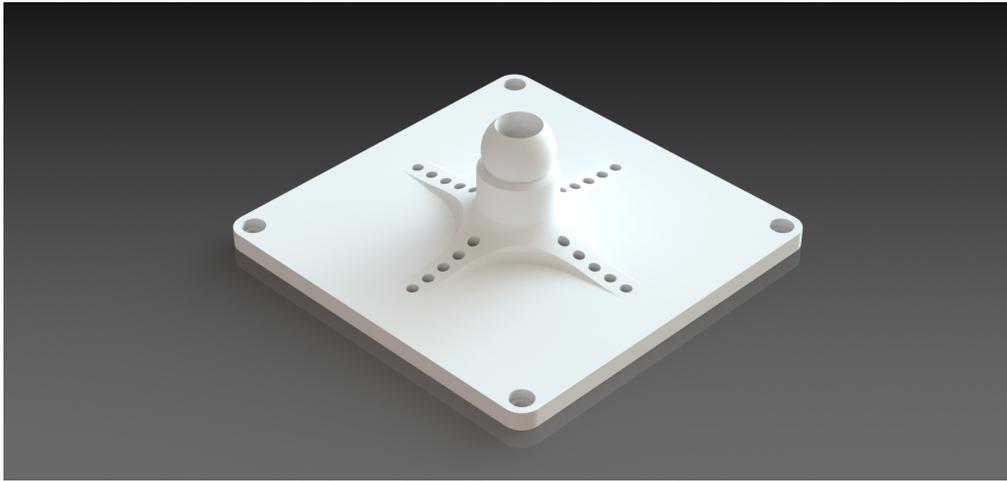
Figure 26: Baseplate, SolidWorks rendering

The base itself is hollow, as to allow for tubing to be fed through the mounting plate and out the back to connect to the muscle boxes. The bottom piece of the base extends out past the main structure to allow for the base to be fixed to the table and counteract the weight of the head and neck.

3.3 Muscle Box Development

Once it was established that the Hydro Muscles would not be mounted on the neck superstructure directly, the team began work on an external housing for the actuators and their supporting electronics. The design evolved as testing continued with the Hydro Muscles and the needs became more apparent. This section will detail the development of the muscle box subsystem and explain how the final design was decided upon.

The initial design requirements of the system were based on the hardware of the electrical system, the intended control scheme, and the preliminary muscle size. One Arduino Nano Every can support four Hydro Muscles along with their associated sensors. The muscle diameters were not expected to exceed 1.5" with their lengths ranging from 8" to 12". The accessibility of the muscles and electronics was also taken into consideration during the design process along with the ease of manufacture and assembly. Finally, while not a core requirement, the boxes were all designed to encase the Hydro Muscles without overly obscuring them from view. The design value matrix is shown below in Table 6. If a later system design change has altered a design's efficacy, the new value is included after a colon.

Criteria	Value	Alpha	Beta	Charlie	Delta
Supports 1 neck segment	6	6:4	6:4	6:4	6
Fits Muscles	10	5	10:5	10	10:5
Fits Electronics	7	0	4	7	7
Ease of Manufacture	6	5	2	3	4
Ease of Assembly	5	5	1	4	4
Component Accessibility	6	6	4	3	5
Safety	8	5	6	8	7
Stability	7	7	7	7	7
Aesthetics	5	4	5	5	4
Cost	5	5	4	1	4
Total	65	48:41	49:42	54:52	58:53

Table 6: Muscle box design matrix

3.3.1 Design Alpha

Initial plans for neck actuation called for a set of four Hydro Muscles for each neck section. To accommodate this design, the first iteration of the muscle box featured mounting points and sliders for four muscles. As shown in drawing 1 below, muscle box alpha was built to be modular, with recessed portions on the top for alignment. This box was the smallest by far, with all components capable of fitting on a single print bed.

For several reasons, the team never manufactured this design. There were no mounting points for the linear displacement sensors which would be required for PID control of the muscles. Additionally, the box proved to be too short to contain muscles long enough to actuate the robot. There were other concerns about the friction caused by the sliders moving through the 3D printed tracks. Several components of this design were incorporated into the later models including the prismatic joint slider system, four-muscle box, and reinforced front and back panels.

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	200-01	Muscle Box Base Alpha	1
2	200-02	Muscle Box Slider Alpha	4
3	200-03	Muscle Box Cap Alpha	1
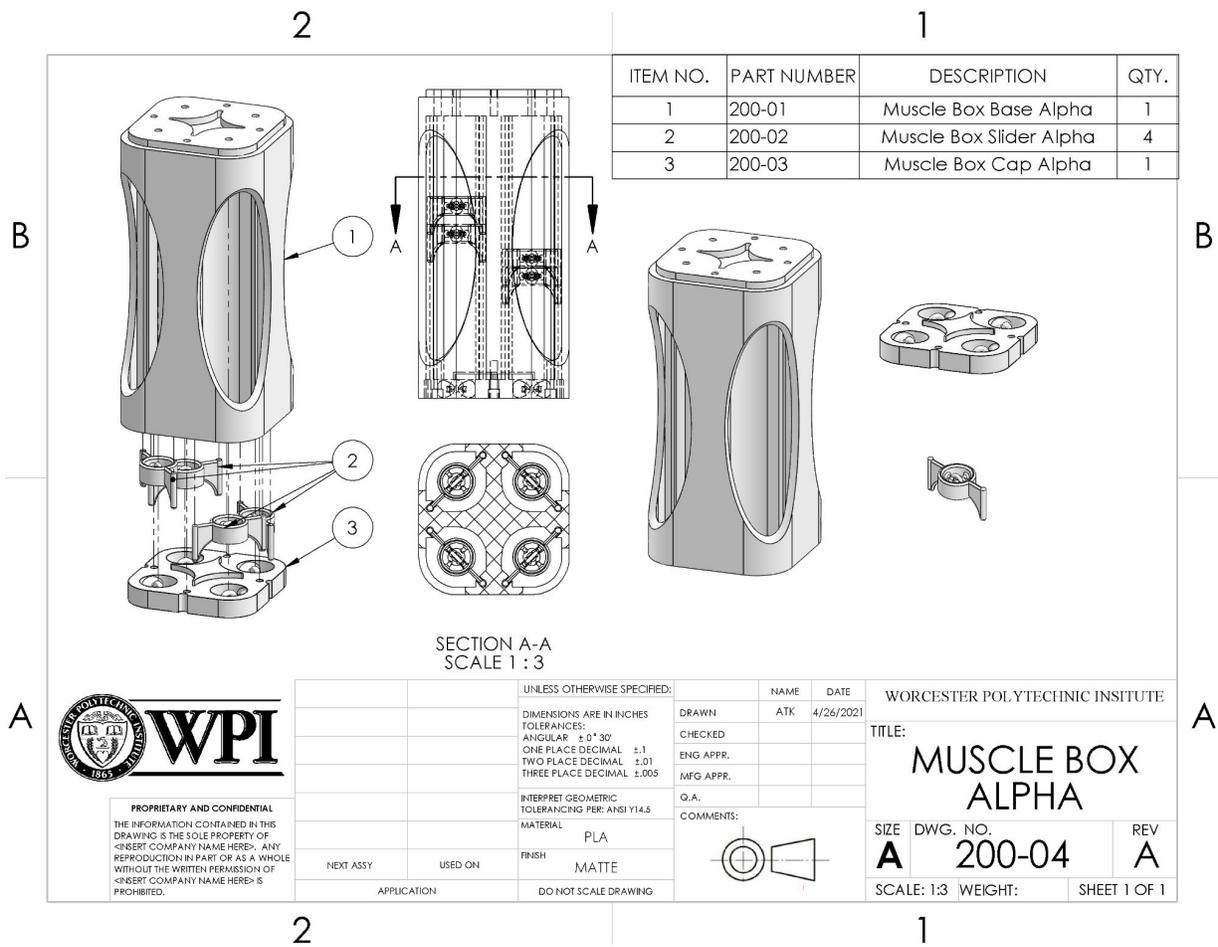
Drawing 1: Muscle box Alpha drawing with bill of materials

3.3.2 Design Beta

The second design addressed the issues discovered while evaluating the first model. Constructed primarily of clear PLA, Beta was almost twice the length of its predecessor with the housing split halfway down due to manufacturing constraints. The muscles were to be attached to a group of wheeled trolleys sealed in the sensor portion of the case. The wheels would press down on four contact potentiometers glued in troughs with embedded cable management routing the wires to a recessed portion on the top meant for the PCB and power converter.

This design was abandoned due to the complexity of the system and likely issues which could arise during printing and assembly. Furthermore, the team concluded that linear slide potentiometers would be easier to install and less expensive than the contact potentiometers used in this design.

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	201-01	Muscle Box Beta	1
2	201-02	Muscle Box Roller Beta	4
3	201-03	Muscle Box Beta Back	1
4	201-04	Muscle Box Beta End Cap	1
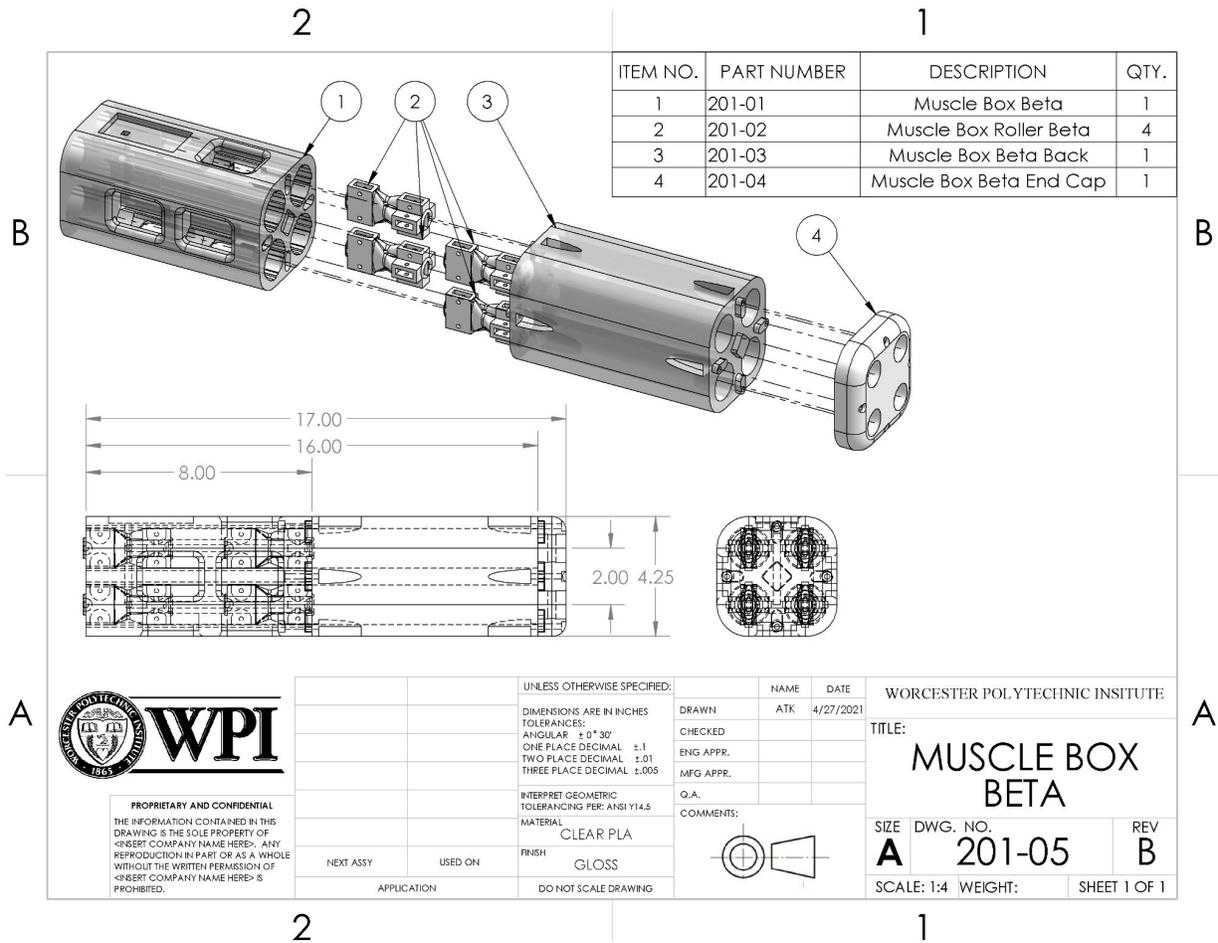
Drawing 2: Muscle box Beta drawing with bill of materials

3.3.3 Design Charlie

Design Charlie incorporated an adjustable length and slide potentiometers. This iteration had separate attachment points for the PCB and the power converter complete with holes for mounting hardware. To reduce printing times, the printed superstructure was split in two and connected with a series of 13 aluminum shafts and four polycarbonate panels. The muscles were intended to attach to a series of four sliders running along eight of these shafts. This was intended to capitalize on the low friction coefficient between PLA and aluminum alloys. The sliders also included arms designed to fit over the levers of the slide potentiometers, which were housed in a five-piece assembly mounted on the central shaft.

The size of this muscle box dwarfed its previous iterations, with the width almost doubling when compared to the Beta model. Further issues arose when the cost of aluminum rods and polycarbonate was estimated to be $100 per box. The decision to reduce the number of muscles per neck segment also strained the feasibility of the box, which would need to house an additional electrical subsystem to support the new control scheme. One of these muscle boxes was printed and used for preliminary testing. Ultimately, this model was discarded when the mount for the sensor module was found to be 45 degrees off its intended position, causing an alignment error.

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	202-01	Muscle Box Final	1
2	202-02	Muscle Box Pane	3
3	202-03	Muscle Box Rod	4
4	202-04	Muscle Box Rod Small	9
5	202-05	Muscle Box Slider Final	4
6	202-06	Potentiometer quad	4
7	202-07	Potentiometer cap	1
8	202-08	Muscle Box Top Final	1
9	202-09	Muscle Box Short Pane	1
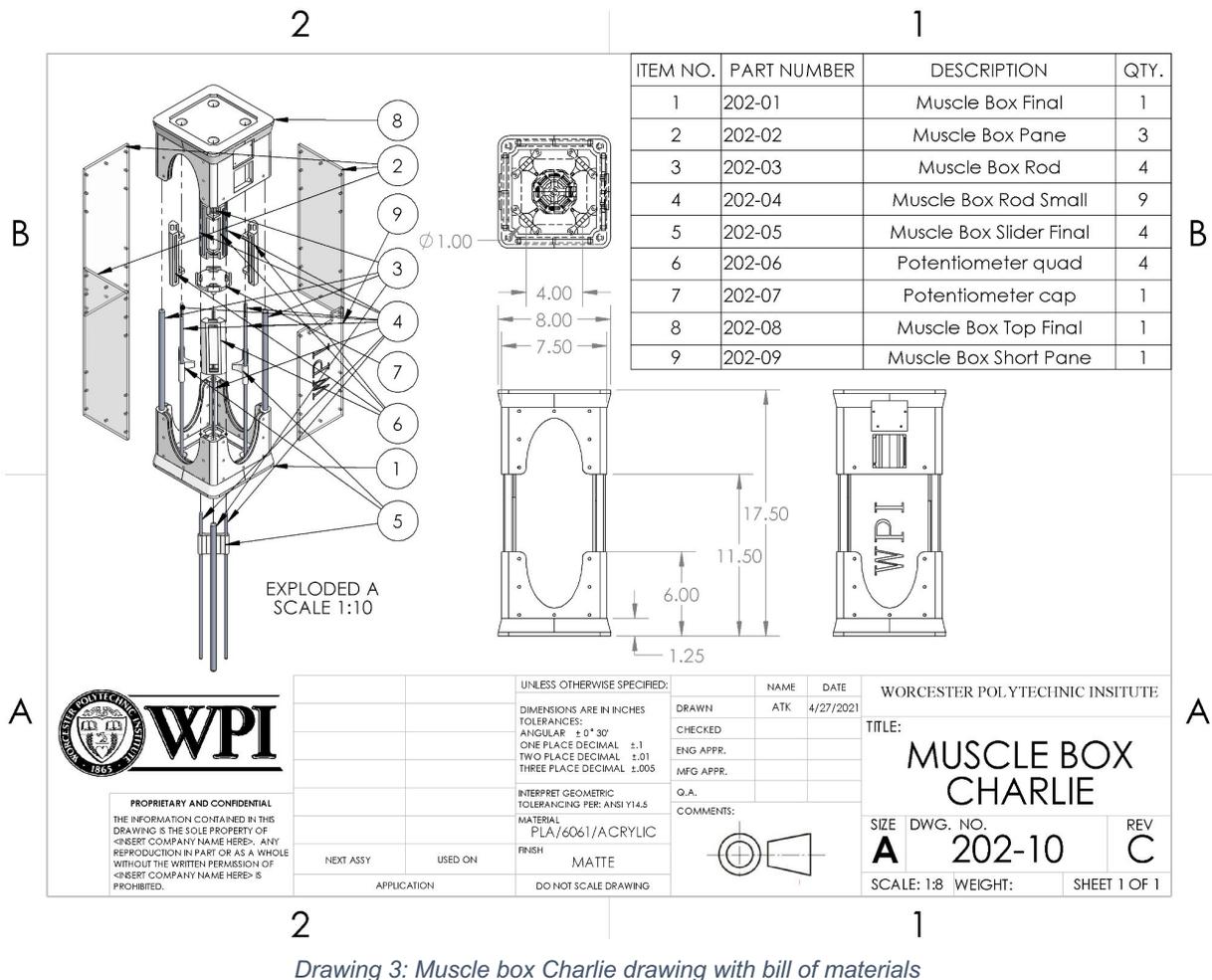
Drawing 3: Muscle box Charlie drawing with bill of materials

3.3.4 Design Delta

The current design borrows heavily from the innovations tested in Charlie, while taking advantage of its reduced muscle count. Muscle box Delta features two 3D printed end caps, one for the electronics and another housing the sensors. Connecting them are a collection of wooden reinforcement rods and aluminum guidance rods of various lengths. These rods serve a dual purpose - ensuring correct alignment and stability of the system if the box is extended to accommodate longer muscles and keep things from interfering with or being injured by the actuation of the muscles.

Six Hydro Muscles currently power the robot and are housed within three model Delta boxes. While the assemblies are functioning as intended, there is room for improvement. The boxes have an excess of space increasing print times and material cost. Future development could focus on reducing these housing units and increasing the accessibility of the sliders. There have also been minor tolerancing issues encountered during assembly.

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	203-01	2 Muscle Box 2b	1
2	203-02	2 Muscle Box 1	1
3	203-03	4in rod	2
4	203-04	10in rod	2
5	203-05	14in rod	2
6	203-06	15.5in rod	4
7	203-07	Muscle Box Slider 2	2
8	203-08	Potentiometer backer	1
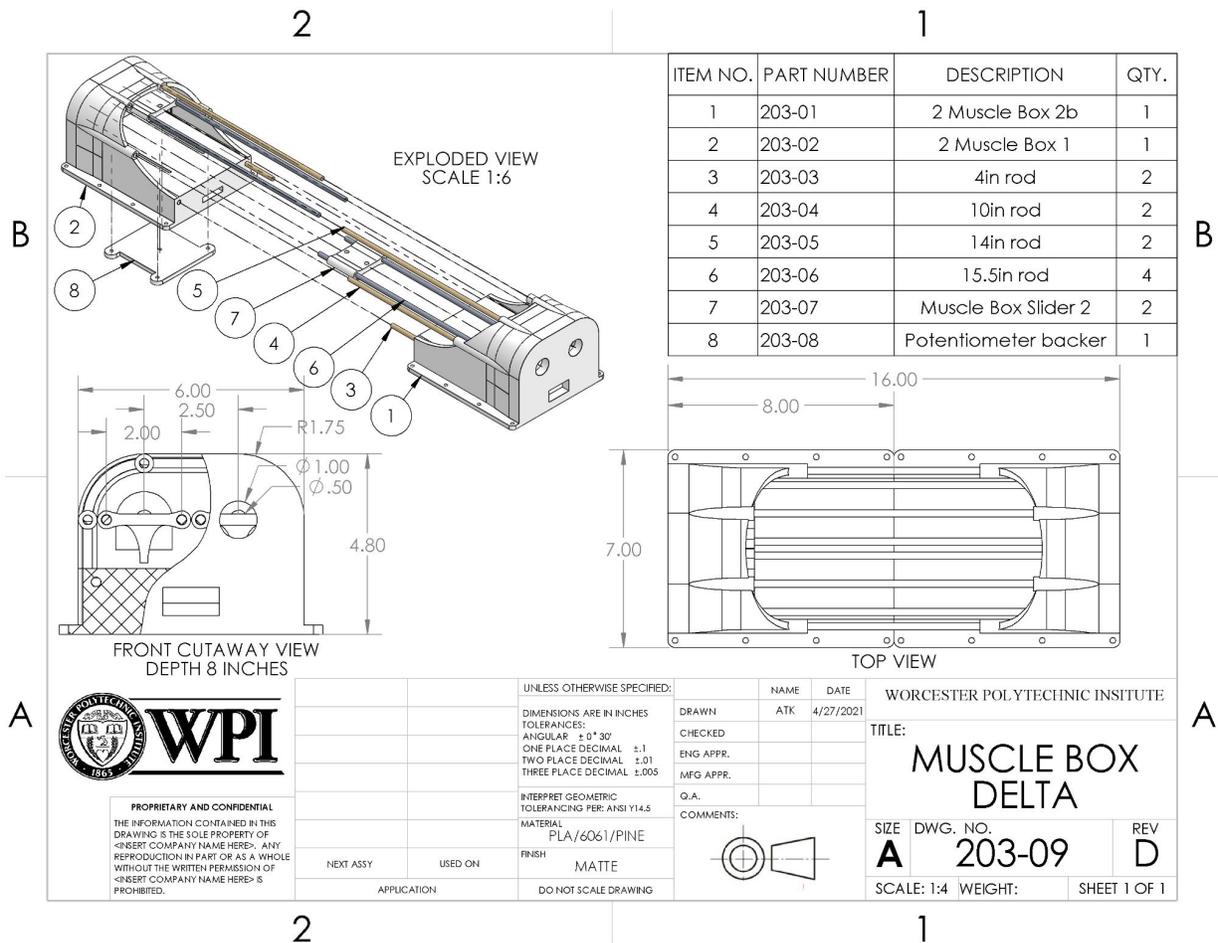
Drawing 4: Muscle box Delta drawing with bill of materials

3.4 Skull Development

The skull-based end effector was an integral part of the team's movement and interactivity goals. The skull was expected to house mechanisms for controlling the movement of the eyes and jaw in addition to a rotational joint connecting to the last vertebra of the neck. The following section details the evolution of the skull assembly and the justifications for design decisions.

3.4.1 Skull Sourcing and Hinge Design

One of the team's primary goals was to replicate the *Deinonychus* as closely as possible. To that end, the starting point for the skull end effector was based on a 3D scan of a fossilized example. The model was sourced from Ryan Carney of Carneylabs.org but was heavily modified before being imported into SolidWorks. The scan was asymmetrical, detailed, and contained a plethora of internal structures which would have impaired its usability. To prepare it for SolidWorks the model was decimated, mirrored, and extraneous features were removed using ZBrush. The simplified mesh was then imported into SolidWorks and scaled to 12 inches. The base skull in SolidWorks is shown below in Figure 27.

Figure 27: Decimated skull after being imported into SolidWorks.

The first design objective was to separate the jaw from the skull. In addition to easing the manufacturing process, this step was crucial in enabling controlled jaw movement. Figure 28 below shows the skull assembly after the addition of the jaw hinges and supporting strut.

Figure 28: Skull model after the addition of the jaw hinge

3.4.2 Eye Development and Print Size Reduction

The mechanical and electrical stretch goals for the project included coordinated eye movement. Once the structure of the skull was finalized, the focus of development shifted to adding mechanically linked eyes. The team decided that the eyes did not have to focus on the same point as modern descendants of the *Deinonychus* exhibit independent control over their eyes. This simplified the system requirements significantly, as did restricting the movement to one axis of rotation. Figure 29 below contains the eye sub assembly.

Figure 29: The eye assembly alone and mounted in the skull.

The eyes and their shared pinion gear feature custom designed M2 teeth and can pivot 135 degrees. The pinion gear has a recessed key which fits the horn of the servo selected to actuate the eyes. The eyes themselves are 1.5 inches in diameter with holes in the top and bottom to secure them to the skull superstructure and the base plate. The top of the skull was modified to accommodate the subassembly. Along with structural reinforcements, the team added a servo housing complete with wire routing. Support struts for mounting the base plate and the eyes were also added. Figure 29 above shows the assembly with the new additions.

Next, the team divided the major skull components into smaller sections for printing. Initially, the skull and jaw were both 12 inches long, making them difficult to manufacture on most 3D printers. To resolve this issue both pieces were vertically cut eight inches from the back. The team then added sliding joints with retaining bolts for assembly. See Figure 30 below for more details.

Figure 30: The skull after being divided for printing.

3.4.3 Adding Jaw and Vertebra Actuation

The last stage of development focused on actuating the jaw and connecting the skull to the top vertebra. Custom gears were designed to fill both needs. To control the jaw, two mirrored servo mounts were added with the center distance from the motor shaft to the axis of rotation of the jaw joints set to 0.75 inches. This gear set was developed using Imperial measurements to simplify its integration into the existing hinge parts. The team extended the inner portions of the lower jaw joint before cutting out the space around the 1/32 diametral pitch teeth. The outer diameter of this section was also expanded to keep the heads of the teeth flush with the rest of the joint. An accompanying set of 16 tooth spur gears were also designed to fit the driving servos. To stabilize these gears, a crescent shaped retaining wall was added to the skull which would prevent them from being pushed away from the integrated joint gears.

The top vertebra had to be mounted to the back of the skull via a revolute joint. To accomplish this within the project's timeframe, the team decided to use a simple gear system with the skull and top vertebra sharing a central shaft for alignment. M1 gears were selected for this project as the servo mounted pinion gear was too small to accommodate larger modules. The high torque ratio was selected to reduce the stress on the smaller teeth while concurrently enabling finer positional accuracy. Figure 31 below shows the completed skull assembly. For additional documentation of the gears please refer to Drawing 204-14 Sheet 3 in appendix B.

Figure 31: Fully assembled skull with top vertebra attached.

3.4.4 Skull Assembly and Troubleshooting

Only one copy to the skull was printed due to time constraints. Most of the features came out well, but the lower halves of the jaw servo mounting brackets were lost. The tolerancing on the printed components was gradually increased over the course of the project which unfortunately caused the mounting holes on the surviving support struts to be too high for the gears to mesh properly. To fix this, soft steel supports were fabricated to push the servos into proper alignment. This modification was successful, and the jaw was able to function as intended afterwards.

Low tolerancing in the eye subassembly design resulted in significant friction when turning the eyes. This was reduced by filing down the eye pins.

Initially, the minimal support along the axis of rotation of the skull-vertebra revolute joint resulted in its unintended function as a cylindrical joint leading to the sporadic loss of the end effector. This issue was mitigated by adding elastic supports to stabilize the joint.

Figure 32 below shows an exploded view of the final assembly. For a detailed bill of materials and mechanical documentation please refer to Drawing 204-14 in appendix B. Additional drawings and documentation are available upon request.
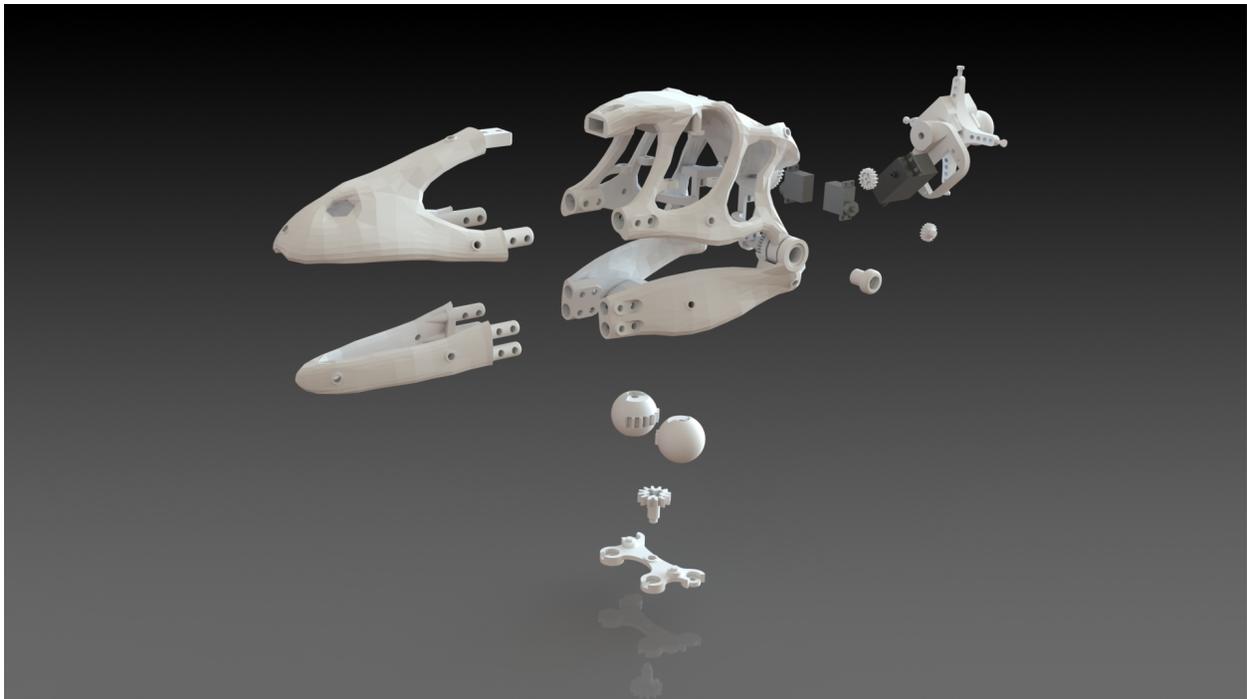
Figure 32: Exploded view of the skull assembly

3.5 Electrical Development

The electrical system bridges the gap between the robot's software and hardware development. The primary goals during development were to incorporate sensors to measure muscle length and neck position, to provide PWM control for the hydraulics valves, and to power the entire system independent of lab power supplies. This section will break down the development decisions made regarding component selection, circuit design and assembly procedures.

As discussed in future sections, early plans for the hydraulic system called for each neck section to be actuated by four Hydro Muscles and include one inertial measurement unit (IMU) for position monitoring. Each muscle was to be independently monitored and controlled. The parallelism exhibited in this design made it an ideal candidate for modular puppet subsystems operating under the direction of a primary microcontroller. The team decided to utilize a suite of Arduino Nano Everys to control the puppet subsystems, which made them the base for the system's electrical design.

3.5.1 Sensor Selection and Power System Design

Once the modular control scheme had been established, the electrical system began to take shape. Each muscle box would house one custom printed circuit board (PCB), which would monitor and control four Hydro Muscles and one IMU using an Arduino Nano Every. With a basic layout prepared, the team shifted its focus to sensor requirements and acquisition, starting with the IMUs.

The IMUs play an important role in positional monitoring and by extension kinematic controlled movement. If the robot were to function autonomously for extended periods of time as intended, these chips would have to have low in run biases to prevent increasing offsets during continuous usage. The robot was not intended to move exceptionally fast, so the precision of the gyroscope and accelerometer took precedence over the maximum measurable rotation and acceleration. After researching several options for precision IMUs on the market, the team reached out to Analog Devices Incorporated (ADI) about four specific avionics grade products. The company generously provided five sensors in total, including three ADIS16475s.

The ADIS16475 product family features accurate sensors with factory calibration. Additionally, the parts the team received were already mounted on breakout boards, making them easy to connect. The A section of the vertebral column features an ADIS16475-2, while the B and C sections each contain an ADIS16475-1. These models are separated by the sensitivity and dynamic ranges of their gyroscopes. For the lower neck sections, the 16475-1 has a dynamic range of +/- 125 degrees per second with an in-run stability of 2 degrees per hour. The 16475-2 has a +/- 500-degree dynamic range with an in-run stability of 2.5 degrees per hour. The accelerometers in both can measure +/- 8 g of acceleration with an in-run bias of only 3.6 ug per hour. The ADIS 16475-2 was selected for the top vertebra as this section would have the most movement.

To measure the muscles' positions the team considered force sensors and linear displacement sensors. The latter was chosen to reduce system complexity. The first potentiometers the team purchased were two-lead contact-based potentiometers. Muscle box beta featured recessed tracks to mount these sensors and a wheeled trolley system to apply

pressure at a specific point relative to the end of the Hydro Muscle. These sensors were less resistive than the later models tested. To maintain an output range of 4.5V, the circuit would have drawn up to 2A. To prevent sensor current changes from impacting the servos or board power, the potentiometers would have been supplied by a unity gain buffer circuit. In addition to power issues, the contact potentiometers exhibited nonlinear behavior and were problematic to mount. As a result, the buffer was scrapped to reduce complexity and the subsequent change in sensors further reduced power concerns. Later designs included three-lead slide potentiometers which were easier to install, less expensive, and offered greater resistivity and length options which helped reduce power consumption.

Once the core system components had been selected, the team's focus shifted to connecting and powering them. The Arduino Nano Every is a 5V board which requires an input voltage between 7V and 15V when not supplied by a micro-USB. The ADIS is a 3.3V board which requires a power supply between 3V and 3.6V. A 5V to 3.3V logic converter was added to communicate between the microcontroller and the IMU.

To capitalize on the full range of the Arduino's analog to digital converter (ADC), the analog output reference voltage of the potentiometers should range between 0V and 5V. Therefore, it should be supplied with 5V. The final potentiometers chosen had a resistive range between 1kΩ and 100kΩ, over a 4in stroke length. Each potentiometer draws 50μA, for a total of 1mW of potentiometer power consumption per muscle box. Finally, the board was designed to support four servo motors. Approximating their stall current at 1A each, they could draw a maximum of 20 Watts combined.

With the power requirements of the subsystem components established, the power system began to take shape. The first stage would need to convert 120V AC to 12V DC power. This would supply the Arduinos. Next, a series of voltage regulators would further reduce the power to 5V for the potentiometers and servos. One of these regulators would be paired with each board. Finally, a low power 5V to 3V voltage regulator would power the level converter and IMU.

3.5.2 PCB Design and Manufacture

One of the electrical goals for this project was to transfer the electronics to PCBs. This process began once the design of the peripheral electrical systems had been finalized. Each PCB was intended to house one Arduino Nano Every, one 5V - 3.3V level stepper, one 5V to 3.3V voltage converter, and four resistors for the potentiometer circuits (not used in the final build). Additionally, there were attachment pins added for the 12V and 5V power supplies, the IMU interface, four potentiometers, and four servos (PWM controlled). Any unused pins on the Arduino were also given pins on the board for later use. The 12V to 5V regulators were large, so to save space on the board they were attached to the muscle boxes separately.

Figure 33 below shows an annotated view of the PCB. The two-layered board was designed using Fritzing, an open-source circuit design platform, and manufactured using their first party fab in England. Inclement weather in Texas delayed the arrival of the boards by two weeks which, when combined with the scheduled lead time, resulted in a month long pause on electrical work. Initially, three boards were ordered as a test batch. Due to time constraints, the excellent performance of the test boards, and the scale reduction of the project, these parts were never replaced.
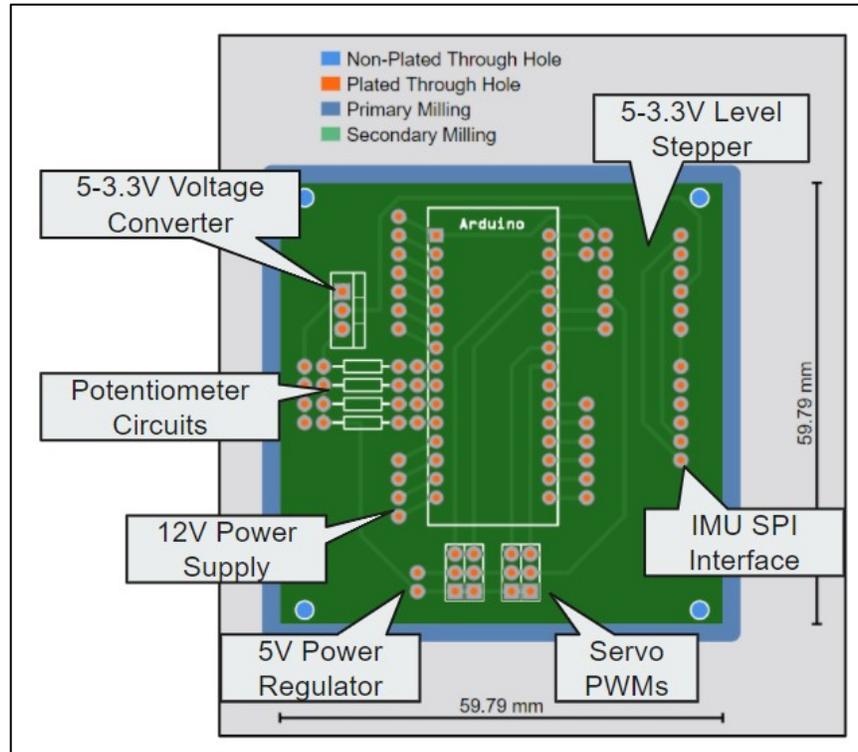
Figure 33: Labeled schematic of the PCB.

3.5.3 Additional Electrical Components

While development of the electrical system was heavily focused on the peripheral subsystems responsible for controlling and monitoring the neck, the robot also includes a camera, a speaker, and a microphone, all of which are connected to the Raspberry Pi via USB. While the camera and microphone are not currently used by the software, future work aims to utilize them for autonomous behaviors. The system could also be improved by providing an independent power supply for the hydraulic pump which currently requires a lab power supply.

3.6 Hydraulic System Development

In conjunction with vertebral column development, the team designed a hydraulic system that would provide the main actuation for the physical robotic system. The main components of the hydraulic system are the Hydro Muscles, which provide the pulling force for the vertebral tendons, and the CRFC valves, which control the movement of the Hydro Muscles. The team studied the design principles and methodology from the original Hydro Muscle and CRFC research and applied them to fit the specific needs of the *Deinonychus* robot. The fabrication process that the team used to create the Hydro Muscles and CRFC valves used in the system can be found in Appendix C. The following sections will discuss the calculations used to determine Hydro Muscle performance metrics and develop a hydraulic system to provide for muscle actuation needs.

3.6.1 Determination of Hydro Muscles Specifications

The team began the hydraulic system fabrication with the Hydro Muscles, as the specifications of the muscles depended on the physical needs of the system. In order to effectively actuate the vertebral column, the Hydro Muscles needed to provide both the desired flexion in each vertebral section and adequate pulling force to move each section.

The team began by determining the desired flexion and pulling force for each muscle to produce. Vertebral section flexion is determined by the change in length of the tendons actuating that section. The required change in tendon length was determined experimentally by deflecting a single vertebral section to its maximum deflection in each direction and measuring the overall change in tendon length. The change in length for Section B was used, as this section includes four vertebrae and would thus require a greater change in tendon length than the other shorter sections. Using this method, the team found that about three inches of tendon length change is required to fully deflect Section B from one directional extreme to the other.

To determine the pulling force required from each Hydro Muscle, the team estimated the pulling force required to actuate Section C of the neck. Because this section is located at the base of the vertebral column and bears the weight of the rest of the robot, the team estimated that this section would require the largest amount of force to actuate. The required force was determined by assembling Section C and attaching weights to the top vertebrae to mimic the weight of the other sections of the head. This section was rigged with tendons, and the force required to deflect the section by pulling on each tendon was measured using a force gage attached to the tendon being pulled. Using this method, the team found that it took a maximum of about three pounds of force to actuate Section C in each direction under the simulated weight of the system.

Having determined the desired performance metrics of each Hydro Muscle, the team then determined the design specifications required to produce the necessary force and change in tendon length. The team considered four main Hydro Muscle characteristics:

1) **Sheathing Type** - the characteristics of the fabric sheathing around the Hydro Muscle tubing. This tubing must restrict expansion in the radial direction of the muscle while allowing elongation and contraction. This can be achieved either by using non-stretchable fabric and bunching up the sheathing to allow slack, or by using one-way stretch fabric that allows the tube to increase in length as much as the stretch factor of the fabric will allow without allowing the muscle to increase in diameter.

2) **Muscle Length** - the relaxed length of the muscle tubing and surrounding sheathing. The starting muscle length determines the maximum change in muscle length and consequently the change in tendon length. The maximum change in length is dependent on both the tube length and the maximum sheathing length, which determines how much the muscle can elongate. In the context of the broader system, the muscle length determines how much vertebral flexion the muscles will produce.

3) **Tube Inner Diameter** - the inner diameter of the flexible tubing in the muscle. This determines the output force of the muscle and affects the pressure required to elongate the muscle. In the context of the broader system, this determines the pressure that must be supplied by the pump to elongate the muscles and the force the muscles exert on the vertebral tendons.

4) **Water-to-Tube Ratio** - the ratio of the cross-sectional area of water in the muscle to the cross-sectional area of tubing. This is dependent on the inner diameter of the tube and the thickness of the muscle tubing, which relates to the pressure required to elongate the muscle. A higher water-to-tube ratio indicates that less change in pressure is required to produce a change in muscle length. In the context of the broader system, this ratio determines the amount of muscle elongation achieved at a specific system pressure.

The team evaluated different tubing lengths and radial dimensions to make a muscle with the desired force and length change. To provide the desired change in tendon length, the team decided on an 8-inch muscle sheathed in fabric capable of stretching 1.5 times its original length in the lateral direction. The fabric holds its shape radially, preventing the muscle from expanding radially, while allowing the muscle to stretch from 8 inches to a maximum length of 12 inches. This results in a change in length of four inches, which provides the desired three inches of tendon movement while allowing a one-inch buffer for the muscle to become pressurized at the beginning of its stroke.

After selecting the desired length, the team tested tubes with various water-to-tube ratios to find a muscle tube that gave the desired force while being capable of being pressurized by an off-the-shelf pump. These tests are discussed in more detail in the Experiments section. Based on this testing, an 8-inch muscle with a tubing ID of ⅜" and OD of ½" was originally chosen for use to actuate the vertebral sections. However, after implementing the 8-inch muscles in the robotic system, the team needed to add pre tensioning mechanisms to ensure that the muscles constantly held the vertebral column under tension via the attached tendons. This addition of pretensioners resulted in a need for a longer stroke length, and the original one-inch buffer proved to be insufficient. To provide the necessary stroke length, the team switched to using 10-inch muscles with the same tube specifications. With the 1.5x stretch fabric, this would theoretically result in a stroke length of 5 inches, increasing the change in tendon movement by one inch.

3.6.2 Hydraulic System Design

Once the muscle specifications were chosen, the team next needed to design a hydraulic system capable of supporting muscle operation given their desired force and speed performance. The pressure and water flow to the muscles is provided by the supporting hydraulic system, which consists of valves to control muscle flow and a pump that supplies water to the muscles via flexible tubing.

3.6.2.1 Hydraulic System Requirements

The desired output from the hydraulic system is primarily determined by the number of Hydro Muscles needed to actuate the robot. The physical structure of the robot consists of a vertebral column with three actuated sections, as discussed in Section 3.2, and a mobile head, as discussed in Section 3.4. The team went through several iterations of muscle layouts to move the spine and head. The team initially estimated that each vertebral section would require four separate muscles to actuate deflection via each of the four vertebral processes of each section's top vertebrae, as well as two muscles to actuate axial rotation of the head in both directions. This design required a total of 14 muscles, and after considering the amount of upkeep and hydraulic capacity required to sustain a system this size, the team decided to develop a simpler actuation scheme that required fewer Hydro Muscles. The team then developed a system of actuation that makes use of passive elastic elements to implement antagonistic muscle pairs with one muscle instead of two. This design change reduced the number of muscles needed from 14 to 8, as each degree of freedom only required two muscles to actuate. Finally, the team decided to implement the axial rotation of the head using a motor mounted to the top vertebrae rather than two muscles, resulting in a final actuation system that makes use of six Hydro Muscles to actuate the three degrees of freedom in antagonistic pairs.

One of the main considerations when designing the hydraulic system was the selection of a pump to provide water flow and pressure to the Hydro Muscles. The muscle operational needs were determined based on the volume of the muscles and the desired elongation speed. The volume of each muscle is based on the tube inner diameter and muscle length, and the total volume of the muscles is determined from the total number of muscles in the system. To achieve a realistic movement speed and reach the system movement goals, the team decided that the muscles should be capable of extending from a resting filled position to their maximum extension in no more than one second. The chosen pump must be capable of providing a water flow rate to the muscles that is fast enough to achieve this movement as well as a water pressure that is high enough to extend the muscles.

When choosing a pump, the team also had to consider the design of the rest of the hydraulic system and the hydraulic losses inherent to this system. The water from the pump would be directed through flexible tubing, and the flow to the muscles would be controlled by CRFC valves. The team needed to consider the water flow rate through the system as well as the pressure loss through the system, both of which the pump would need to be capable of overcoming. To account for this, the team modeled the water flow through the system, working backwards from the desired pressure and flow rate at the muscles to the corresponding output pressure and flow rate from the pump.

3.6.2.2 Hydraulic System Pressure and Flow Rate Calculations

To calculate the required pressure and flow rate capabilities of the pump, the losses and flow throughout the system were modeled. When modeling the system, the team assumed ideal conditions and neglected minor losses. The team based the hydraulic system calculations on the equations found in *Fluid Mechanics* [22].

The team began by calculating the volume and flow rate required to fill and elongate the Hydro Muscles. Assuming the muscle has previously been filled with water in its relaxed state, the change in muscle volume from the relaxed muscle position to full elongation can be found by:

$$\Delta Volume = \pi \left(\frac{D_I}{2}\right)^2 \Delta L \qquad (1)$$

Where D_I is the inner diameter of the muscle tubing, and ΔL is the total change in muscle length. This value can be multiplied by half the number of muscles to estimate the maximum volume output from the pump required in the highest demand scenario, i.e., if all sections were fully deflected from one extreme to the other simultaneously. After selecting a desired muscle filling speed, this volume can be used to calculate the required volumetric flow rate into a single muscle:

$$Q = \frac{\Delta Volume}{Speed} \qquad (2)$$

Where *Speed* is the desired elongation speed of the muscles from resting to full extension. Along with the desired muscle pressure, the volumetric flow rate into a muscle can be used to determine the output requirements for the pump. The number of muscles being actuated and the pressure losses throughout the system must be considered before estimating the pump requirements. This was done by considering the losses through each segment of pipe and additively estimating the total losses throughout the system.

For a given tube section, the pressure loss can be approximated by:

$$P_{Loss} = f\left(\frac{L}{D}\right)\left(\frac{\rho V_{avg}^2}{2}\right) \qquad (3)$$

To calculate the pressure loss, the values for average velocity and friction factor must be calculated. The average velocity can be found by:

$$V_{avg} = \frac{Q}{A} \qquad (4)$$

where A is the cross-sectional area of the flow through the tube:

$$A = \pi \left(\frac{D_I}{2}\right)^2 \qquad (5)$$

The friction factor (f) of the tube section must also be calculated. To find f, the Reynolds Number must first be used to characterize the flow through the tube:

$$Re = \frac{\rho V_{avg}^2 D_I}{\mu} \qquad (6)$$

Where ρ is the density of water at 20° Celsius and μ is the viscosity of water at 20° Celsius. Based on the Reynolds Number, the flow through the tube is characterized by:

Reynold's Number	Flow Type
Re < 2,300	Laminar
2,300 < Re < 10,000	Transitional
Re > 10,000	Turbulent

Table 7: Characterization of Flow Type

The friction factor for laminar flow can be calculated using equation 7 below.

$$f = \frac{64}{Re} \qquad (7)$$

For transitional & turbulent flow, the friction factor can be calculated using the Haaland Equation:

$$\frac{1}{\sqrt{f}} = -1.8log(\frac{6.9}{Re} + (\frac{\varepsilon/D_I}{3.7})^{1.11}) \qquad (8)$$

Where ε is the tube roughness. Once these values have been calculated, the values can be plugged into the pressure loss equation to estimate the pressure loss across each tubing section. For tubing sections connected in series, the pressure loss along the flow path is cumulative, while the volumetric flow rate remains constant (White, 2015). Because the flow is being split from the pump to be distributed to each muscle, the branching flow at each Y connector must also be considered. The volumetric flow rate into the Y connector is equal to the sum of the volumetric flow rates leaving the Y connector (White, 2015). In addition, the pressure drop across the parallel segments after the Y connector is equivalent; this means that the pressure losses can be calculated for one branch and applied to other identical branches (White, 2015).

The team used the equations and process described above to model the hydraulic system and select a pump with the flow and pressure output needed to effectively actuate the chosen Hydro Muscles. The calculations were performed to support a system of eight muscles to accommodate the expansion of the system to accurate more degrees of freedom in the future. The calculations were originally performed for 8-inch muscle design operating at a pressure of 50 PSI and a fill time of one second, which was used for pump selection. However, the calculations were updated when the team switched to 10-inch muscles to confirm that the pump still provided the required specifications. For more detail, please see Appendix D for the Excel spreadsheets used to perform the pressure and flow rate calculations. The final calculated values are as follows:

System Requirements: Six 8-inch Muscles	
Flow Rate (L/min)	1.737
Pressure (PSI)	50.25
System Requirements: Six 10-inch Muscles	
Flow Rate (L/min)	2.171
Pressure (PSI)	50.36

Table 8: Hydro Muscle Requirements

Based on these results, the team selected the available FATBOY series 5538-2E1-94A pump. This bypass pump is capable of providing a flow rate of 27 L/min and a maximum pressure of 60 PSI, both of which are sufficient for providing for the muscle needs calculated from the hydraulic system.

3.6.2.3 Hydraulic System Diagram & Components

After determining the required pump specifications, the team had all the information necessary to design and implement a hydraulic system capable of fulfilling the actuation needs of the vertebral column. A diagram of the final hydraulic system is shown below, followed by a description of each of the main components.
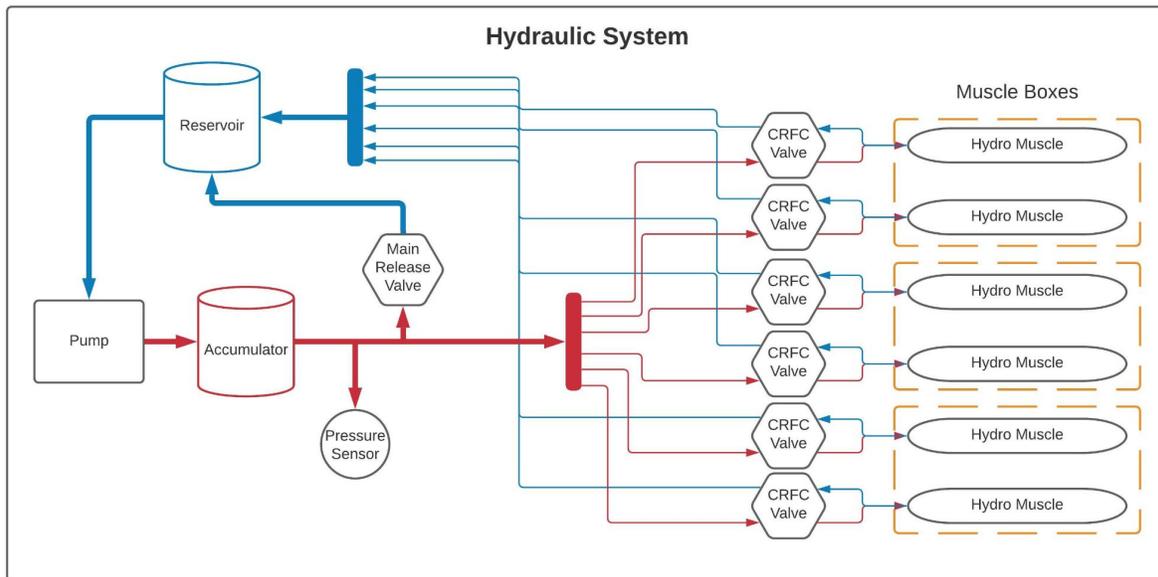
Figure 34: Diagram of final hydraulic system

1. **Pump** - The hydraulic system modeling showed that the team would need a pump capable of providing a flow rate of at least 2.171 L/min and pressures of at least 50.36 PSI. Because the flow needs of the hydraulic system would fluctuate based on the number of muscles being actuated at a given time, the team decided to use a bypass pump. This would allow the pump to run continuously regardless of the water draw of the system. As mentioned above, the team chose to use a bypass pump that was readily available at Popovic Labs: the FATBOY series 5538-2E1-94A.
2. **Accumulator** - To reduce fluctuation in the pressure and flow to the hydraulic system, the team decided to use an accumulator placed between the pump and the rest of the system. This system includes a SHURflo Accumulator Model #181-201. The team encountered difficulties measuring and setting the accumulator precharge but ultimately implemented the accumulator with a pressure setting of about 25 PSI.
3. **Pressure Sensor** - The team utilized a digital pressure sensor to monitor the pressure in the system.
4. **Main Release Valve** - For safety, the hydraulic system includes a manual release valve between the accumulator and the Hydro Muscles. This valve remains closed during

regular system operation but can be opened to relieve system pressure in the event of a system failure.

5. **CRFC Valves** (6)
6. **Hydro Muscles** (6)
7. **Reservoir** - The pump draws water from an open reservoir bucket at atmospheric pressure. The water discharged from the muscles is directed back into the reservoir.

The components in the schematic were obtained and connected using flexible tubing of varying sizes. Barbed connections secured with super glue and epoxy were used to connect tubing, as well as metal clamps to prevent leaking. To split the flow from the pump output to the valves and muscles, barbed Y-connectors were connected using similar methods.

3.7 Control System Development

To coordinate movement simultaneously between multiple vertebral sections, each with several Hydro Muscles, the team implemented a network of Arduino Nano Every's. This network served to distribute the load of controlling Hydro Muscles. The network was expanded to provide methods for controlling groups of muscles to bend vertebrae sections into the desired vertebral column shape and report the robot pose in physical space.

3.7.1 Muscle Control

The implementation of a control method is crucial in the use of Hydro Muscles as a method of actuation. The method chosen required responsive and dynamic control over the length of the Hydro Muscle to smoothly and accurately puppet the robot.

When designing a control system, an open or closed loop method may be used. In an open loop system, commands are sent to change the position of the actuator with no further reading of the real-world result. In the case of Hydro Muscles, this control system may be implemented via dead-reckoning or opening a valve for an experimentally predetermined amount of time to obtain a desired change in muscle length. This method is simple, however it is unsuitable for the team's application of hydraulic muscles as the method cannot respond to changes in the state of the system, such as a change in system pressure or force applied to a Hydro Muscles. The muscles also experience an accelerating change in length over time when exposed to constant pressure, which makes pre-determining an amount of time to open a valve troublesome if not impossible.
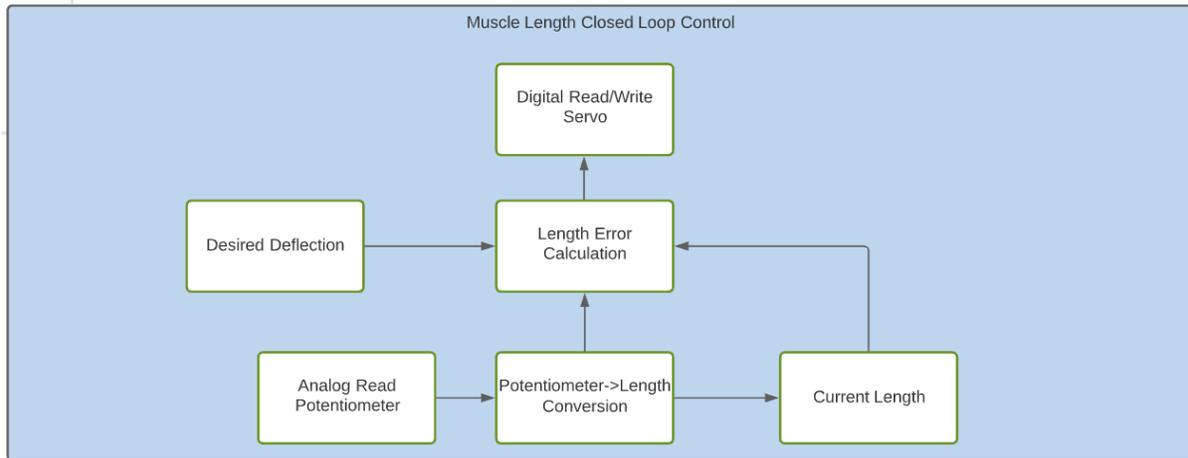
Figure 35: Close loop control diagram based on length.

Instead, a closed loop control method was used. In a closed loop control scheme, once a command is sent to change the position of an actuator, the real-world effect is measured via some sensor. This measure can then be used to scale the command sent to the actuator to correct for any discrepancy in desired position. A commonly used control scheme that had previously been evaluated for Hydro Muscles is a PID controller [23]. A PID controller takes measurements of an actuators position and uses them to calculate the proportional error, the past state of the error, and the rate of change of the error to scale commands to the actuator.

3.7.1.1 Linear Potentiometer Calibration
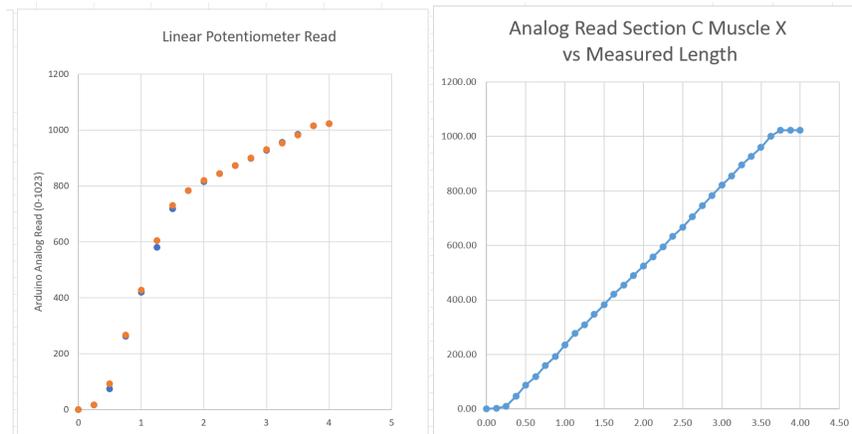
Figure 36: Potentiometer linearization data

To implement closed loop control, the length of Hydro Muscles needed to be measured. There are several methods that have been tested however the team decided to use linear potentiometers [A. Barth, J. Sorrells, S. Ueda, and J. Wu, "Hydro Muscle Control System", 2015]. The output from these sensors was experimentally measured and compared to the change in length of the Hydro Muscle to evaluate sensor output for linearity. This method is important for obtaining a reliable measurement of length. Once this method was determined, a function was then written to convert the ADC value into a length measurement in inches.

3.7.1.2 CRFC Valve Calibration

Actual control over the length of the Hydro Muscles is achieved by controlling the CRFC valve position. These valves are hand built and have a servo that deflects left to empty a muscle of water and right to fill a muscle. The amount filling or emptying is proportional to the valve's deflection to either side and may be different from valve to valve based on a variety of factors. For this reason, there are some parameters that can be experimentally measured to create a constantly behaving valve. These parameters include a minimum amount of deflection for creating any change in length, a maximum amount of deflection achievable by the servo due to torque constraints, and a maximum amount of deflection past which no gain in flow is achieved. To protect the valves from overheating and to avoid dead-band errors in control, the first two of these parameters were determined and used to constrain available CRFC valve positions.

3.7.1.3 Hydro Muscle Safety

The system has no limit switches to protect Hydro Muscles from over-extension. To still provide protection for the Hydro Muscles, a parameter for the maximum length reading of the muscle was chosen that was less than the maximum length reading of the linear potentiometer. This artificial maximum, if surpassed, triggers the muscles to begin emptying regardless of desired position.

3.7.2 Networking Vertebral Sections

The control of many Hydro Muscles posed a challenge for the team. The exact number of muscles was not immediately clear from the outset of the project but was expected to be large, so the method needed to be expandable. These muscles needed to be grouped by section and each needed their own control loop and I/O. Any method chosen would need to be able to accommodate all these requirements at any muscle scale as well as maintain all other aspects of control of the robot such as positional models. To distribute the computational and I/O load, a custom network of Arduino Nano Everys was built using I2C.
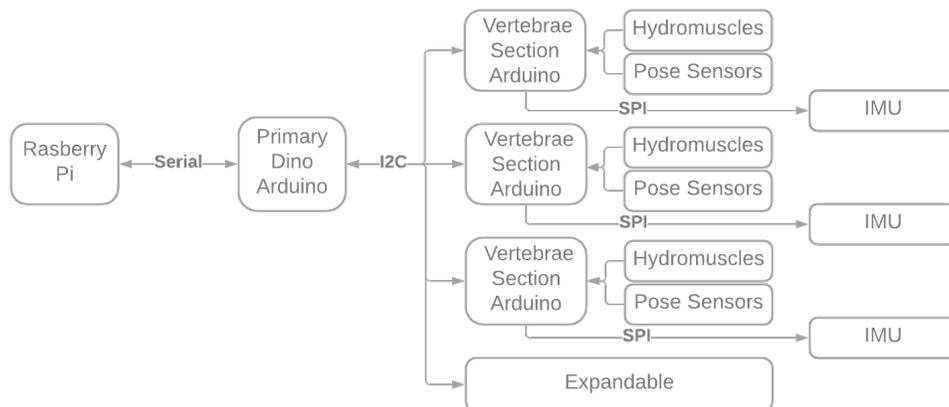
Figure 37: High level robot networking

The network consisted of one primary Arduino and three secondary Arduinos. The primary Arduino controls the network and is the main interface between any user and the robot. The primary takes commands and coordinates robot movement via further commands to secondary Arduinos which are responsible for obeying requests for muscle movement or data. The primary is also responsible for control over head functions such as eye and jaw movements, as well as measuring system pressure. The network may handle up to 128 secondary Arduino Nanos which are responsible for implementing Hydro Muscle control loops. Each secondary Arduino Nano corresponds to a vertebral section, and as such each can calculate pose parameters for that section based on muscle lengths. Though currently not in use, the IMU's are connected to the system via Serial Peripheral Interface.

To facilitate the network, a messaging format was defined on top of the I2C communication protocol which was chosen for its simplicity and speed. Only primary can initiate data transfer in I2C, but the network needed to provide 2-way transfer of information and be fast as messaging is blocking and the robot needs to be spending as little time as possible communicating. Each message can be up to 32 bytes long due to the available buffer size.

Messages sent over I2C are in the form of a char array, with the address not included as it is separately used to send data. To enable communication, the first four bytes of the message are used as a header to identify the message. This header is identical regardless of the message being sent from primary to secondary or visa-versa, but each treats the header differently.
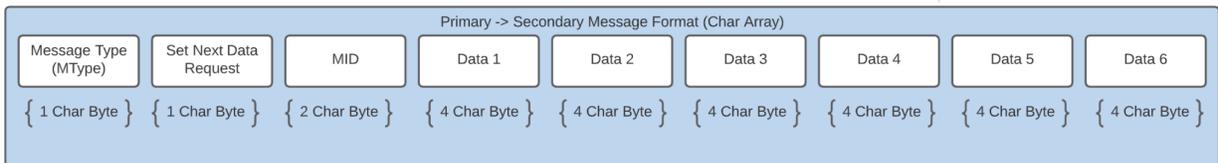
Figure 38: I2C message format: Primary to Secondary

When a message is sent from the primary to the secondary, the first byte of the message header is used as an unsigned char to identify the type of message (mType) to the vertebral section. This will affect the response the section should have to the message and a glossary of MType values can be found in Appendix F. The second byte of the header is used as an unsigned char to identify that the primary would like to receive data and the type of data to be received. This byte is referred to as the SetNextGet byte.

The MType of 1 is special and signifies that the primary will be requesting data, of the type specified in the next byte, and that there is no data accompanying the message. This byte can be set without a MType of 1, to preemptively prime the secondary to respond with specified data. A glossary of SetNextGet values can be found in Appendix F. The next 2 bytes of the header are used as an integer as a message ID. This is used to validate responses from sections and keep track of messages for debugging purposes. Once the secondary receives a message it parses it for its MType and SetNextGet type. Depending on these values message data is read and the local values specified by the MType are modified accordingly.

Secondary -> Primary Message Format (Char Array)								
MType of Request	Warning Flag	MID of Request	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6
{ 1 Char Byte }	{ 1 Char Byte }	{ 2 Char Byte }	{ 4 Char Byte }	{ 4 Char Byte }	{ 4 Char Byte }	{ 4 Char Byte }	{ 4 Char Byte }	{ 4 Char Byte }
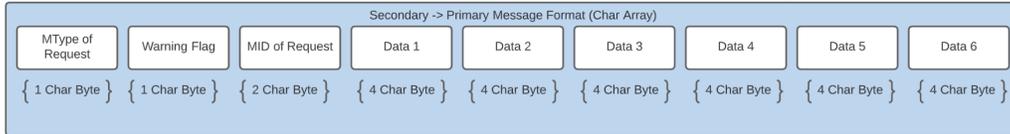
Figure 39: I2C message format: Secondary to Primary

When a message is sent from the secondary back to the primary, data is being requested and the first four bytes are delegated similarly. The first byte refers to the MType of the requesting message, the second byte is a warning flag that is used to alert the primary that the secondary has data that the primary may want to obtain. This is also used to alert the primary of networking failures, such as requesting data that does not exist. A glossary of warning values can be found in Appendix F.

Regardless of sender, the next 24 bytes are broken into 4-byte increments to store upto 6 floats of data per message. This design was originally chosen to accommodate six position measurements (XYZ, Roll Pitch Yaw). The format of the data varies for each MType and can be found in Appendix F. There remains an additional 4 bytes of space in the protocol that could be used for any purpose for future work.

3.7.2.1 Data Type Conversion

As the data must be in the form of a char array the data can be simply converted by using a union struct.

```
union floatToBytes {
  char floatBuff[4];
  float floatVal;
};
```

Listing 1: Union struct conversion from float to byte array.

This technique is a faster method than a proper conversion and only works because the Arduinos share data type implementations, the converted char pointer does not need to be operated on as a float, and the float buffer size takes up the same amount of space in memory (four bytes). One value of the struct is the datatype wanted to send and the other is a char pointer. To convert between float and char pointer the struct can be assigned a float value, and then the char representation in memory can be obtained to be sent.

3.7.2.2 Dino Controller API

To control individual Hydro Muscles and sections, an API was implemented that used the messaging format built on top of I2C to provide simple functions for interacting with vertebral sections. This API provides easy to use methods for configuring control of individual Hydro Muscles on a per vertebrae section basis as well as commanding individual Hydro Muscle position. The API also allows for obtaining vertebrae section connection status and vertebrae section configuration parameters such as the number of vertebrae in the section and the length of section.

To provide write methods, such as commanding a change in muscle position. A single wire.h message is sent with wirehelper. The header value types MType and SetNextGet are set

to values pre-defined and found in Appendix E. Then the muscle length data is formatted to be char and concatenated with the header and sent to the secondary. To provide read methods, such as reading the length of the muscles in a section, two messages must be sent. The first message must have a SetNextGet value corresponding to muscle positions to prime the vertebrae section to respond with the correct data. Then a request for data can be sent to the section and the section responds with the MType requested, any warning flag, and the data requested. This API can be expanded by creating new MTypes to provide new commands and new SetNextGet types to provide new read commands.

3.7.3 Kinematic Implementation

To calculate the pose of the robot in 3D space, a relationship between the Hydro Muscles and the vertebrae sections position must be established. Given the similarities between the team's robot and the robot described by Elephant's trunk manipulator paper, the team found it appropriate to use the modified Denavit-Hartenberg forward kinematic approach they described.

To do so the team's robot was divided into three sections, each actuated by a pair of orthogonally placed tendons actuated by Hydro Muscles. Each section is assumed to have no axial rotation (rotation along the length of the section), and to approximate constant curvature. To have constant curvature the neck must not buckle. If these assumptions are met the straight-line position vector between the start of the neck selection and the end can be calculated as though it were a spatial curve.

$\Delta L = l_1 - l_2$
$\pi = \gamma + 2\alpha + \beta$
$\alpha = \arctan(w/h)$
$\beta = \arctan(\sqrt{4r^2 - (2r^2 - l_2^2)^2}/(2r^2 - l_2^2))$
$\gamma = \pi - 2\arctan(w/h) - \arctan(\sqrt{4r^2 - (2r^2 - l_2^2)^2}/(2r^2 - l_2^2))$
$K = 2n\gamma / s$
$\theta = n\gamma = ks / 2$
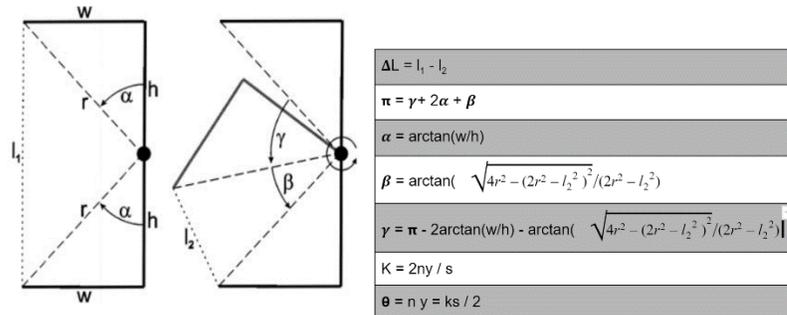
Figure 40: Geometry of two vertebrae deflection with equations [1]

The length of each Hydro Muscle is used to approximate a deflection of the neck in the orientation aligned with the muscle via the geometric process outline in Figure 40. The approximation assumes the change in tendon length is the same as the change in muscle length, and that the change in length causes each vertebra to bend the same amount. With these assumptions and the geometry of the team's vertebrae, each section can calculate a pair of perpendicular deflections. These can be converted to the rotation and deflection parameters used in the modified DH approach [1].

Each vertebral section Arduino supports read and write commands to section deflection and rotation via the I2C network. Write commands for a deflection and rotation get converted into perpendicular deflections which can be obtained by changing the muscle length. These deflections may be used in place of length in the closed loop control, albeit with different tuning parameters, to achieve control over the section. Commands to read the current pose of the vertebrae section will retrieve the current deflection and rotation, which must be calculated from the perpendicular muscle deflections.

$$A_0^4 = \begin{bmatrix} \cos(\phi)\cos(\kappa l) & -\sin(\phi) & -\cos(\phi)\sin(\kappa l) & -\dfrac{1}{\kappa}\cos(\phi) + \dfrac{1}{\kappa}\cos(\phi)\cos(\kappa l) \\[2em] \sin(\phi)\sin(\kappa l) & \cos(\phi) & -\sin(\phi)\sin(\kappa l) & -\dfrac{1}{\kappa}\sin(\phi) + \dfrac{1}{\kappa}\sin(\phi)\cos(\kappa l) \\[2em] \sin(\kappa l) & 0 & \cos(\kappa l) & \dfrac{1}{\kappa}\sin(\kappa l) \\[2em] 0 & 0 & 0 & 1 \end{bmatrix}.$$

Figure 41: Spatial Robot homogeneous transform matrix [1]

Once obtained, the deflection (k), rotation(Φ), and length (l) of a section can be fed into the homogeneous transformation matrix to calculate the pose of a single section. Multiple matrices may then be multiplied together to obtain the pose of the robot's end effector.

Inverse kinematics, or the process of calculating the deflection and rotation in each section required to achieve a desired end effector position, was not implemented due to time constraints and complexity. However, the robot may still be described using the kinematic model as is, and it could be extended to implement a numerical method of solving the inverse kinematic problem.

As of now, the inverse kinematic model was not completed and the

3.7.4 Dino Command Line Interface

The *Deinonychus* will not be a standalone robot but will interact with other programs or humans to receive commands to move. As such, an interface between the robot and the world needed to be developed. Arduino supports easy communication between an Arduino and another computer over USB via the Serial library, so this technology was used to build a command line interface (DinoCLI). The purpose of this DinoCLI was to allow the team to debug robot control functions, modify control configurations, obtain experimental data, and allow the robot to be commanded via kinematic model parameters. This CLI also needed to be non-blocking as to allow the primary Arduino measure pressure and maintain its model of robot position while it is not receiving instructions.

During operation, the DinoCLI constantly checks for user input via a call to *Serial.isAvailable()* which returns the number of characters available to be read from the serial buffer. If no input is available, the DinoCLI does nothing and allows other functions and measurements to occur. This allows the CLI to be non-blocking. If there is input available, the input is read until it sees a newline character with a customizable maximum line length of 128 characters. The line is the split on space characters to parse arguments from the string. If the arguments match those for a predefined command, then that commands behavior is executed, whether it is returning a value or commanding a change in pose. These argument strings consist of one or more keywords, and each keyword may have one or more parameters. The command *listvert* has no parameters and returns a list of connected secondary Arduinos. In contrast the command *setvert mpos address deltaMuscleLengthX deltaMuscleLengthY* has multiple keywords (*setvert, mpos)* and parameters (*address*, *deltaMuscleLengthX deltaMuscleLengthY*) and commands a change in the lengths of muscles in a section. Additional command strings are enumerated in Appendix E. To allow the DinoCLI to support both human debugging and access

to other computer programs, it was built with output in English and json formats. Switching between these two formats can be accomplished with the command *climode mode* where the mode is either debug or json.

The commands available currently provide functions for controlling muscle lengths and obtaining section deflection and rotation via the *getvert pose* command. However, the DinoCLI was never expanded to include functions to control neck sections via deflection and rotation, or via XYZ coordinates, because of complications with the kinematic model that would drive them.

4 Experiments

The team conducted several experiments on the separate subsystems of their robot. These subsystems included the Hydro Muscles and the dinosaur head with its eyes and jaw. Experiments were run with the control code to verify deflection calculations and the entire system was tested together in the neck flexion tests.

4.1 Hydro Muscle Force

The team tested various Hydro Muscle tubing dimensions before selecting the final parameters for the muscles. The different Hydro Muscle characteristics discussed in the Methodology section were chosen and evaluated to produce favorable results. The team used an initial testing pump to evaluate the muscles, a GOODPUMPs model with a maximum pressure output and flow rate of 130 PSI and 6 L/min, and a force gage to measure the contracting force of the muscles. To test the pulling force of each Hydro Muscle, the team secured the front end of the muscle and tied a fishing wire tendon to the end cap. This tendon was connected to a force gage. For each test, the muscle was pressurized to its maximum length, which was measured and recorded. The force gage was then secured with the tendon pulled taught, and the muscle was depressurized to measure the pulling force.

The team first tested an 8 inch long Hydro Muscle with a latex rubber tube with a ¾ inch inner diameter and a 1-inch outer diameter. The Hydro Muscles used for this project are sheathed in 1.5x one-way stretch fabric. The team had originally decided that a change in length of at least 3 inches was needed to actuate each vertebral section. As such, the team chose an 8-inch muscle design that should be able to reach a 12-inch extension, providing 4 inches of length change. This muscle had a wall thickness (outer diameter minus inner diameter) of ¼ inches, giving a water-to-tube ratio of about 1.29. When tested, this muscle reached a maximum length of 11.5 inches (3.5-inch extension) and produced a pulling force of 17.8lbs. This force greatly exceeded the estimated 3lb of force needed, so the team decided to consider Hydro Muscles with a smaller inner tube diameter. Using a smaller Hydro Muscle would reduce the pressure and flow volume required from the system pump.

The team then tested an 8-inch Hydro Muscle with a ¼ inch inner diameter and a ⅝ inch outer diameter. This muscle had a wall thickness of 3/16 inch and a water-to-tube ratio of 0.19, much smaller than the previous design. The team repeated the same test process but found that the test pump was unable to pressurize the muscle due to the small cross-sectional area of water in the muscle in comparison to the tube thickness.

Based on these results, the team decided to test a muscle with a similarly small outer diameter and a thinner wall thickness. The team chose an 8-inch muscle with an outer diameter of ⅜ inches and an inner diameter of ½ inch. This resulted in a wall thickness of 1/16 of an inch and a water-to-tube ratio of 1.29, the same as the original muscle tested. The team performed the length and force gauge test and found that this muscle yielded a pulling force of about five pounds, which was more reasonable to achieve the team's desired force of three pounds. The team also confirmed that this muscle could reach the desired 12-inch maximum elongation (4-

inch extension) when sheathed in the 1.5x one-way stretch fabric. After this positive result, the team decided to select these muscle parameters for the final system.

4.2 Eyes & Jaw Control

The team initially assembled and actuated the skull independently of the spine. The team began by inserting the eye subassembly. This process was done by attaching a plastic servo horn to a micro servo, then inserting the single servo that actuates both eyes into the slot in the skull. The servo was centered using an Arduino Uno and code on the Arduino website. The gear was fitted to the servo horn and the eyes were inserted and their plate was placed to hold them in. The team began by sending the servo to random locations to test the range of the subassembly. The team found that the minimum servo position was 70 and the maximum was 111. These values were determined based on the limitations of the gearing between the eyes and by observing where the pupil of the eyes ended up at certain servo positions. Once these values were found, the team wrote simple random motion code to run if the head received power. The code can be seen below:

```
int HeadController::randomEyeMotion(int positionChangeFrequency){
    currentMillis = millis();

    //test whether the sample period has passed
    if (currentMillis - sampleMillis >= positionChangeFrequency){
        int newPos = random(70,111);
        setEyeServoRot(newPos);

        sampleMillis = currentMillis;
    }
}
```

Listing 2: Random eye position code

Next, the team tested the range that the dinosaur jaw could obtain. The team began by inserting the servos, which required additional supports due to parts of the 3D printed baseplate being damaged. The servos were centered using the same Arduino centering code used on the eye subassembly. Once the team was able to receive consistent meshing between the jaw and the servos, the team attached the jaw, and sent the jaw servos a series of positions to understand its behavior when receiving small changes in orientation vs large and what the maximum and minimum values for the jaw servos should be. To find the maximum, the team observed the dinosaur head and decided, anatomically, what appeared to be a reasonable angle to which the dinosaur should be able to open its mouth. This was the 130-servo position, For the minimum jaw angle, the team sent the jaw servo positions under 90 until the jaw closed. To make sure the mouth was able to hold itself shut against the weight of the jaw, the jaw servos needed to be sent to a 'closed' position in which the servos could barely not reach, as to make sure the servos were always powered. The value of 70 was used to hold a 3-inch ball to maintain the power of the servo.

4.3 Deflection Calculations

To test the calculation of deflection in a vertebrae section, a muscle box's sliders were connected to a single vertebrae section and an elastic band was used to maintain tension in the section. This deflection caused the sliders to move, and vice versa. The neck was positioned such that there was zero deflection in the section and the lengths of the muscles were recorded. The Arduino serial monitor was used to output the values being measured by the Arduino for muscle length, perpendicular deflections, and the single deflection and rotation needed to fill a sections homogeneous transform matrix. The value of these perpendicular deflections was validated with real world measures of deflection captured using a protractor. A graph of measured deflection versus actual deflection was then created with deflection measurements taken every 5 degrees and plotted.
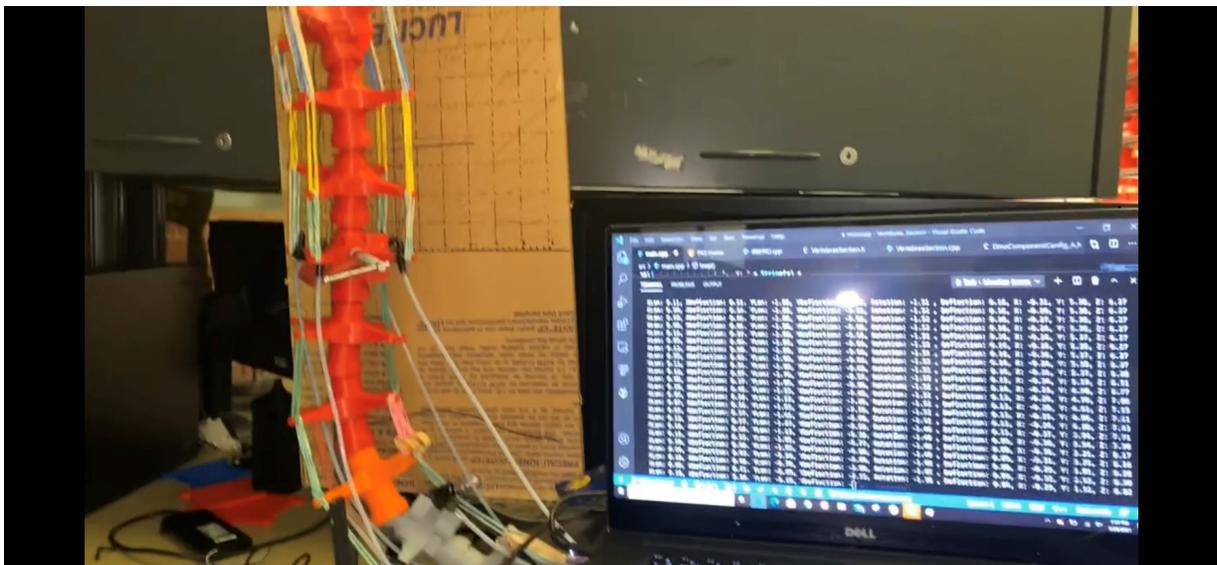
Figure 42: Deflection test rig

4.4 Neck Flexion Tests

The team created a simple to use system to perform flexion experiments on the dinosaur. The Arduino Nano Every was uploaded with the main system code, which allowed anyone with a micro-USB cable and the Arduino IDE to plug their computer into the system and control the dinosaur with the serial monitor. The team could then type commands into the field to send the robot to desired positions. The list of commands, parameters, and effects are shown in the table in Appendix E.

The team began by testing the system without the head, as the head was heavy, and the team wanted to see how movement was without its weight. The team also set up the Spiderwire tendons so that the antagonistic pair to each muscle was a large rubber band. This method was tested in hopes that more sections could be actuated by reducing the number of muscles needed per an actuated degree of freedom.

This test was done by using the *setvert* command, which only implemented one section of movement. This command worked as expected. To make sure the experiment was only

working with maximum and minimum lengths of muscles, the team used -200 to send minimum muscle length and +200 to send maximum muscle length, as the code automatically rounded to the minimum or maximum desired change when given very large values. This was a successful method to only work with the values that would produce maximum and minimum muscle lengths and therefore maximum flexion. Once the team made sure the control worked, they attached the head. The team observed that flexion was not as far as they wished, so they removed elastics as the antagonistic pairs and decided to pair muscles together. This change would allow the team to only actuate 3 degrees of freedom since now two muscles were needed per actuated joint, but more flexion could now be achieved per an actuated section.

Based on observations, the team decided that side-to-side motion in segment C, side-to-side motion in segment A, and up-down motion in segment A produced the most visual movement. The team then tied segment A's left-right tendons to muscle box at register 16, segment A's up-down tendons to muscle box at register 17, and segment C's left-right tendons to muscle box at register 18. The team then implemented the *twosection* CLI code command and tested actuating two sections simultaneously. This command produced favorable movement, so the team then implemented the *threesection* CLI command which successfully ran three sections simultaneously.

The team then wrote the testSequence command which sends muscle lengths using the *twosection* CLI command while also running the random eye position code, the jaw movement code, and the chosen audio clips from zapslat.com. This testSequence command was used to test the robot for the rest of the duration of the project. If flexion seemed reduced, the team would tighten the tendon wires in the respective section and run the sequence again, observing if the flexion changed favorably. All variables that were believed to affect the flexion obtained by the robot were then tested with the testSequence code. For example, one of the variables modified was where the fishing line that suspended the robot was tied to.

5 Results

The team successfully integrated the hydraulic system to actuate the physical robotic system, an electronic system to power and connect the subassemblies, and a control system to coordinate and actuate the robot and its microcontrollers. System power and networking functioned as desired, and the camera, IMUs, and microphone were installed for future integration. The robot was successfully able to actuate its neck, eyes, and jaw, but when integrating the individual subsystems, the team encountered some unintended results.

5.1 Spine and Head Functionality

During initial testing, the ninth vertebra prototype appeared effective for creating the desired spinal flexion. However, the final vertebrae design, as well as the implementation of the vertebral column, ultimately limited the functionality of the spine.

5.1.1 Vertebra and Vertebral Column

Once the vertebral column was assembled, the observed that the actual angle on the 3D printed vertebra was within 5 degrees of the desired angle value, which was precise enough for the team's application. However, physical interactions between vertebrae, as well as the structure of the spine itself, limited the mobility of the vertebrae. The team encountered friction between the 3D printed ball joints on the vertebrae. In addition, the vertebral column was prone to kinking due to the friction between the joints as well as the pulling force of the elastics added to smooth out the vertebral motion. The team reduced the number of elastics on each section, allowing the system to move more freely, but the tension on the tendon lines continued to cause kinking at extreme deflection angles.

The tendons and Bowden tubes were effective at translating muscle movement to deflect the vertebral column. The team initially encountered issues with the Bowden tube attachments slipping forward through the top vertebrae of each section. Small stoppers were added to the top of each Bowden cable, which successfully secured the Bowden cables and increased their effectiveness. However, the team found that the Bowden tubes distorted over time, bunching up at their attachment points due to the high tension on the tendon lines.

In the current robot configuration, section A of the neck can be actuated in the horizontal plane and the vertical plane, and section C can be actuated along the horizontal plane. Each actuated degree of freedom can be controlled and actuated independently or simultaneously with others. While the spine was capable of deflection in the desired directions (dorsal, ventral, and lateral), the team found the deflection in each direction to be smaller than expected based on the individual flexion capable at each vertebral joint. The spine could achieve the desired 15 degrees of dorsiflexion but was only capable of 38 degrees of ventral flexion compared to the desired 45 degrees. Further, the robot was not capable of the full 40 degrees of lateral flexion in each direction; the assembly could reach 30 degrees of flexion to the right and 2 degrees of flexion to the left. The robot also moved at a slower speed than expected. While the robot could reach the

desired movement speed of 15 degrees per second during certain motions, most of the deflections happened at slower speeds.

5.1.2 Skull Assembly

The skull assembly was able to provide the desired eye and jaw motions. Once installed, the eye assembly could provide coordinated eye motions via the shared gearing attached to the eye servo. The team initially encountered friction between the 3D printed parts of the eye assembly but sanding down these parts resulted in smooth eye motion. The eyes were programmed to move to random rotational positions, resulting in the desired simple random eye motion.

The jaw assembly was also capable of providing the desired range of opening and closing motion. The jaw could grasp a 3-inch diameter object placed in the mouth of the skull and could release the object once the jaw was opened again. The team initially had trouble mounting the two servos that accurate the jaw, as the high number of supports used in the skull printing process resulted in the unintentional removal of portions of the originally designed mounts. The team added metal braces to support the servos, resulting in less servo shifting and a tighter mesh between the gears on the servos and the gears on the lower jaw. This method of securing the servos successfully enabled the jaw to actuate and produce the desired motion.

The gearing and actuated joint attaching the head to the top vertebrae provided sufficient axial rotation at the head joint. The rotational freedom at this joint initially resulted in the skull being able to slide along the joint axle and un-mesh the gears. To prevent this, the team attach elastics to hold the skull more securely onto the joint while still allowing the head to rotate along the axle. The elastics were effective at keeping the skull secure and ensuring the gearing remained meshed. This joint would greatly benefit from mechanical stops and a more robust shared axis.

5.1.3 Weight and Stability

The team encountered issues with the overall weight and stability of the integrated neck and head assembly. Without the head attached, the vertebral column was capable of actuating freely under its own weight and could remain upright without assistance. However, the weight of the head pulled the spinal column downward and prevented the robot from maintaining a steady position. The team added a higher number of elastics to compensate for the added weight and was able to achieve a steady system, but the muscles were unable to overcome the increased resistance provided by these extra elastics and they were removed. To support the increased weight of the robot structure more effectively, the team attached a thin fishing wire tied to elastics to the ceiling. The elastics attached to the wire allowed the spine to flex dorsally and ventrally while still aiding hold the system upright. This wire successfully enabled the system to remain at a steady position and actuate via the Hydro Muscles.

5.2 Electrical Functionality

When purchasing the PCBs, the team decided not to order a soldering template to reduce the cost. This choice resulted in a longer and more difficult assembly process. During quality assurance testing, all soldered joints were examined for shorts and open circuits. Of the 300 joints tested, no shorts were detected between neighboring pins and only one plated through hole on the board was damaged. This was the result of attempts to straighten an improperly mounted set of DuPont pins. This broken contact prevented one of the four PWM pin sets on the third board from functioning. Fortunately, there was 100% redundancy in the PWM and potentiometer pins due to the reduction in muscles per neck segment.

Once the system was fully assembled, the voltages were tested at the output of the 5V and 3.3V voltage regulators. It was also measured at the inputs of the Arduino and servos while the muscles were being actuated. All test voltages were within 0.05V of their anticipated values. While the IMUs have not been used yet, the connections from the IMU breakout board pins to the logic converter were secure.

At this point the robot is powered primarily by a 12V 30A power supply. The Raspberry Pi is powered separately, as is the hydraulic pump.

5.3 Muscle Box Functionality

Muscle box delta was used in the final design of the system. This model was effective in linearizing the muscles movement, monitoring their length, and housing the associated electronics. After implementing the system, though, some issues became apparent. The structural overhang made accessing the Hydro Muscles difficult once the system was under tension. The adjustable support rods were prohibitively complicated to replace, resulting in the team's decision to cut pass-through holes for the muscles. Low tolerancing resulted in issues with the fit of sliders and electrical components. The sliders rubbed against each other, reducing the Hydro Muscles' ability to lengthen easily. In some cases, this would cause the Hydro Muscles to buckle when the sliders got stuck. The potentiometers and PCBs required sanding to fit into their intended positions in the box. Finally, the box was larger than necessary, resulting in increased print times and cost. Overall, this design effectively housed the Hydro Muscles and their supporting electronics but would benefit from minor revisions to increase accessibility and reduce bulk.

5.4 Hydraulic System Functionality

Overall, the hydraulic system was able to provide water flow and pressure to the muscles as expected. The pump was capable enough to extend and contract the Hydro Muscles, and the bypass pump was able to run continuously to support the system. The team initially encountered various leaks throughout the system tubing that were fixed with super glue gel and Epoxy. The Hydro Muscles were also found to leak at the end caps and nozzles after continuous use, and a layer of Epoxy was added to the end seams to mitigate this leaking. Once the leaks were fixed, the system was able to maintain pressure and flow to the muscles.

Despite being able to expand and contract as anticipated, the muscles were ultimately unable to provide the necessary force and speed to actuate the system as desired. The team found that the 8-inch muscles were not long enough to provide the desired flexion after adding the pretensioners to the muscle box. The team then increased the muscles to 10 inches long but were still unable to produce the desired flexion. The muscles also did not provide adequate pulling force to fully deflect the vertebral sections being actuated. The Hydro Muscles struggled to pull the tendons, especially the tendon deflecting section A in the dorsal direction, as this motion acted against the weight of the head. In addition, the team found that the muscles were slower to fill and drain than expected. The Hydro Muscles took on average about 2-3 seconds to fill rather than the desired one second, and the draining was even slower. The team found that the drainage tubes running from the CRFC valve to the reservoir were longer than necessary and shortened these tubes to reduce the travel path of the drained water. This helped the muscles to drain slightly faster, but the Hydro Muscle motion remained slower than desired.

5.5 Control Functionality

The control scheme of the robot was evaluated in multiple ways to ensure functionality. This includes validation testing of the network connecting Arduinos, tuning of the closed loop control scheme they implement for Hydro Muscles, and eventually modeling the deflection each muscle causes. The ability to perform these functions is necessary for the robot to be commanded to move in different ways.

5.5.1 I2C Networking Speeds

To provide an effective alternative to operating all the control loops on one Arduino, the I2C vertebrae section controller network needed to be fast. In order to measure the speed of the network, the values of 6 data points were modified via the network (in one message) and the roundtrip time to write the change and confirm the update to these values was measured.

Message	Speed
Write Value	2206 microseconds
Read Value	4349 microseconds
Write and confirm	6452 microseconds

Table 9: I2C networking speed experiment data

5.5.2 Muscle Length Change Data

The popular closed loop control method of using a PID controller requires experimental tuning of the system. This tuning results in three gain values that can be used to control the position of a Hydro Muscle. Their values can improve or impart noise in the control of the Hydro Muscle. After tuning a muscle, the results of commanding an unloaded muscle to expand 2 inches is shown in Figure 43 below.
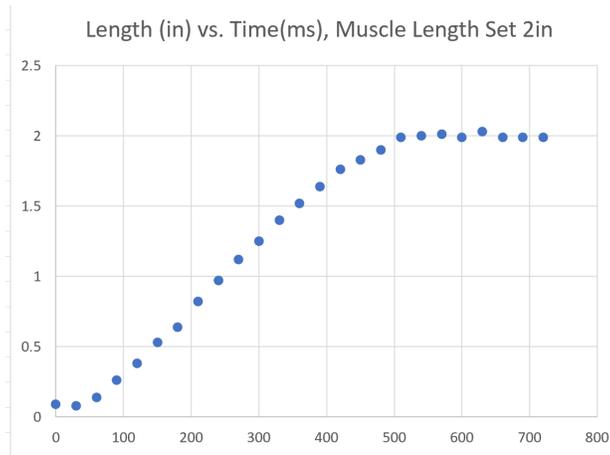
Figure 43: Hydro Muscle 2in extension experiment data

The muscle reached its set position in just over half a second. The PID values used for the control were kP = 1.2, kI = 0, and kI = 0.

5.5.3 Deflection Calculation Measurements
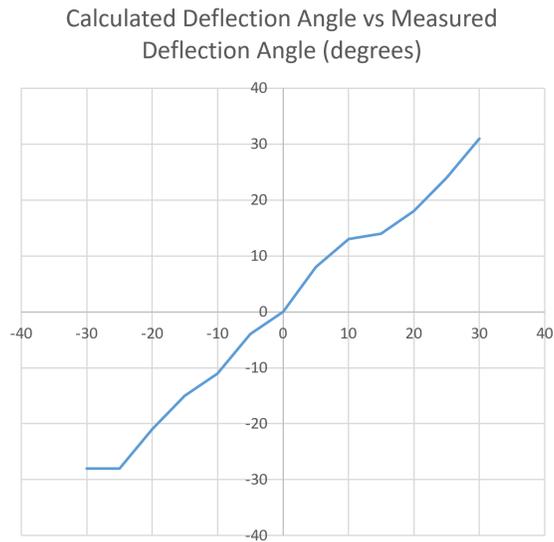
Figure 44: Deflection data along the Y vertebrae axis

While the graph shows a linear relationship between the measured and calculated deflections, which validates the method of calculation, the deflection measures did seem to be biased towards a negative deflection. Adjusting the zero deflection points of the muscles seemed to correct this bias.

5.6 Goal Results

The following section assesses the project goals with respect to each discipline. A copy of the official goals document can be found in Appendix A.

5.6.1 Mechanical Goals

The mechanical goals can be grouped into three subsections. Flexion goals pertain to the degree of actuatable flexibility in the robot's neck structure, measured in degrees. These goals were developed based on the Allosaurus flexion studied by the team. Operational accuracy & performance goals include operating speeds and spatial accuracy with respect to 3d position and rotation. Finally, physical ability goals include the robot's physical ability to move a certain mechanism.

5.6.1.1 Flexion Goals

The team was only partially successful in achieving the desired flexion goals.

At least 15 degrees of dorsiflexion and 45 degrees of ventral flexion:
The robot was successfully able to reach 15 degrees of dorsiflexion. The robot was only able to reach 38 degrees of dorsiflexion. This was able to be done by moving the fishing line suspending the robot from the ceiling to the second to last vertebra.
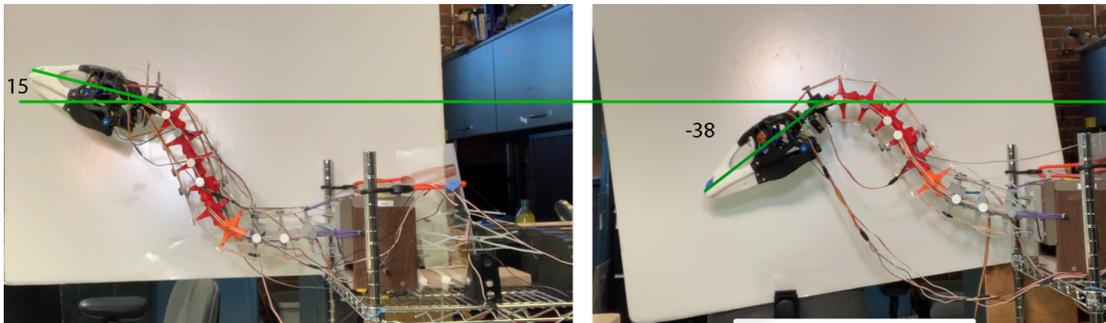
Figure 45: 15 degrees dorsiflexion (Left) -38 degrees ventral flexion (Right)

At least 40 degrees of lateral flexion in each direction:
The robot was unsuccessful in reaching a total of 80 degrees of lateral flexion of movement. The robot was able to turn 30 degrees to its right and only 2 degrees to its left.
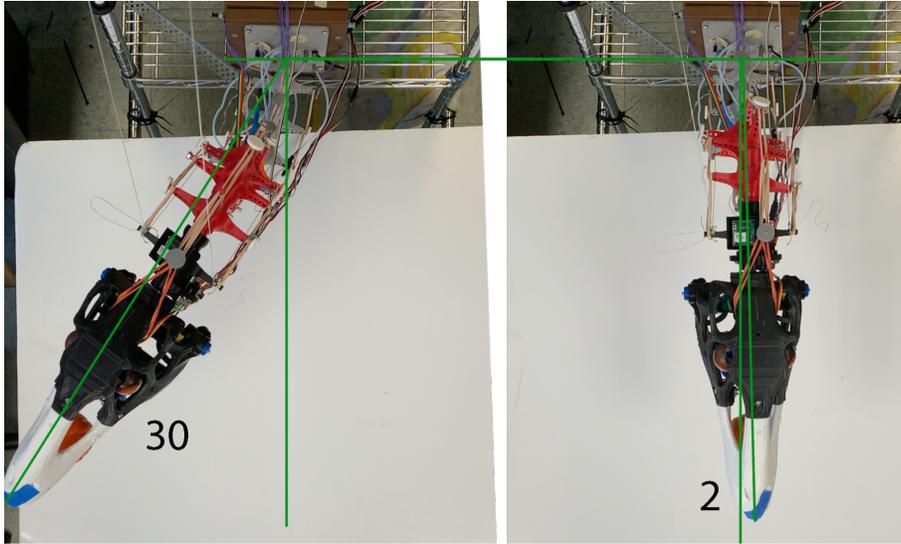
Figure 46: 30 degrees lateral flexion (Left), 2 degrees lateral flexion (Right)

5.6.1.2 Operational Accuracy & Performance Goals

Operating speeds up to 15 in/sec

The team was able to observe an instance in which the robot traveled 15in/sec, but this speed was only reached for a short duration. Overall, the robot was slower than 15 in/sec in most instances. Figure 47 below shows the captured instance of the robot moving 15 in/sec. The math follows:

2 degrees to - 15 degrees = 17 degrees = Theta = 0.297 radians

Time = 3 seconds

Radius = r = 12.3''

S = r *theta = 3.65 radians

Circumference = 2*pi*r =77.3

Arc Length = Circumference*(S/2*pi)

Arc Length = 44.9"

Speed = 44.9"/3 seconds

Speed = 15"/ second
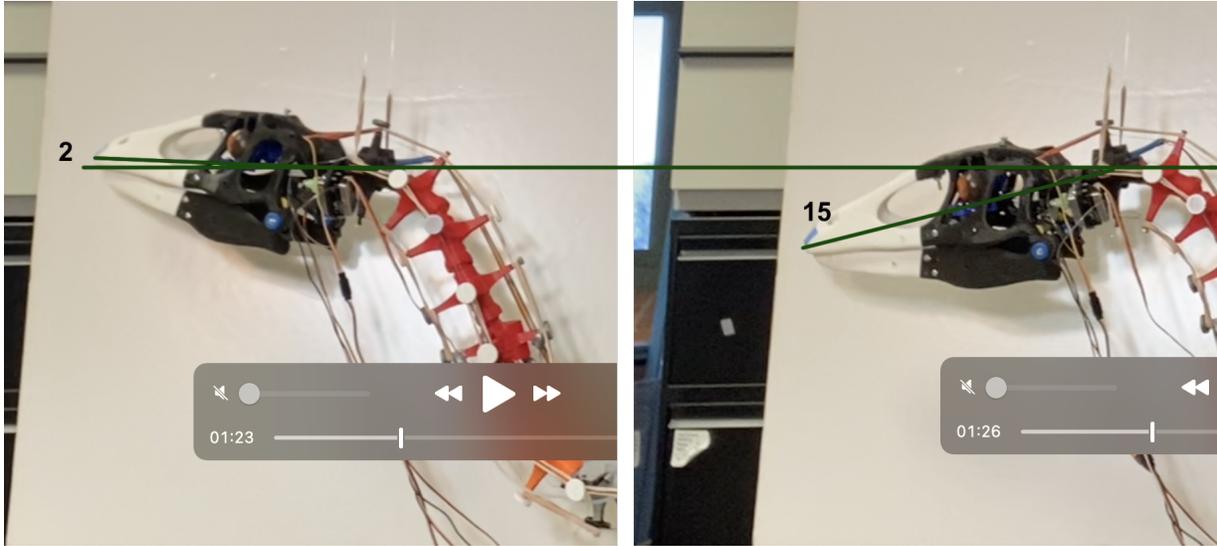
Figure 47: Timestamped change in flexion for speed calculations

Spatial accuracy of +/- 1 inch from the target end effector (head) location & Rotational accuracy of +/- 10 degrees from the target angle

The team was able to calculate a forward kinematic model for the neck based on previous research done with continuum arm robots. The team was unable to completely calculate an inverse kinematic model for the neck due to the complex calculations of the continuum arm kinematic model. Due to time constraints, the team was unable to verify the positioning based on the forward kinematic model.

5.6.1.3 Physical Ability Goals

Develop a *Deinonychus* head that is capable of three degrees of rotation at the joint where the head meets the neck.

The robot's motion was able to complete three degrees of freedom at the neck. The wording of this goal was discussed with the professors mid-project and it was agreed that the dinosaur's neck motions effectively create the lateral and up-down degrees of freedom, and the motor at the top of the neck creates the axial rotation.
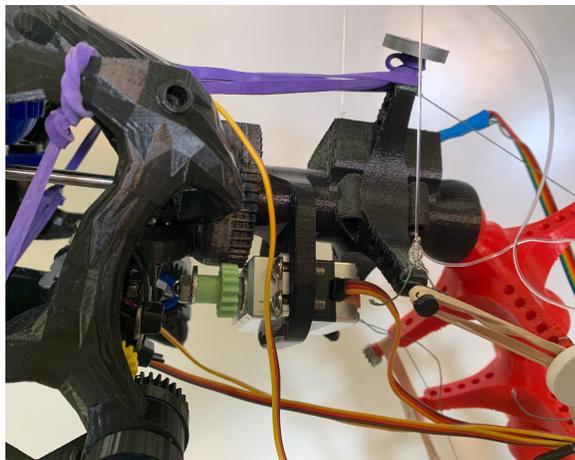
Figure 48: Motor embedded in the top vertebra of the neck

A mobile jaw assembly that is capable of grasping a stationary object with a ~3-inch diameter located in the robot workspace.

The dinosaur was able to hold a 3-inch diameter stress ball in its jaws, therefore this goal was met.

Figure 49: Dinosaur grasping a 3-inch stress-ball in its jaws.

Eyes with simple random or repetitive movement commonly seen in modern animatronics.

The eyes were successfully able to perform simple random motion. If the robot was powered, the code described in the Experiments section ran so that the dinosaur eyes moved to random positions.

Reach Goal: Integrate complex eye movements which can include coordination between eyes, movement to follow objects, blinking, etc.

The team used a single gear to actuate both eyes, allowing the two eyes to be coordinated relative to the other. The eyes are not able to follow objects or blink, therefore this reach goal was partially met.

5.6.2 Electrical Goals

The team met all electrical goals, including the reach goal of enabling coordinated eye movement.

Implement a multiple-sensor suite to collect position and environmental information to enable autonomous functionality.

This goal required the inclusion of at least one microphone, camera, and IMU. Currently, the peripheral Arduinos receive input from six linear potentiometers and three IMUs to measure muscle position and robot pose. The Raspberry Pi is also connected to a USB camera, speaker, and microphone. Though these items were not implemented, they are installed, operational, and available for the further work planned for this project.

Enable software goals by supporting one or more microcontrollers to control key sensory activities.

The robot is currently controlled by four Arduino Nano Every microcontrollers and one Raspberry Pi computer. Three of the microcontrollers are incorporated into the peripheral PCBs. The Raspberry Pi and remaining Arduino are powered separately, but still connected to the larger electrical system.

Develop a robust control system for coordinating artificial muscles & motors, and support system power needs.

The hydraulic system and peripheral electronics are powered by a three-stage system regulating the 120V 60HZ AC input down to 12V, 5V, and 3.3V DC outputs. The regulators chosen were able to handle 150% of the maximum system power consumption. This margin of safety has almost doubled due to the subsequent reduction in Hydro Muscles.

Incorporate a speaker to allow the robot to emit audio.

The team purchased a speaker on Amazon that was able to be plugged into a Raspberry Pi with a USB port. The speaker is activated in the team's movement test sequence and continuously emits jungle ambient audio and dinosaur roars when the robot opens its mouth. These audio clips are courtesy of zapsplat.com.

Transfer final electrical schematic(s) to custom printed circuit boards.

Each muscle box contains a custom PCB designed to house one puppet Arduino with its supporting sensors, servos, and power supply.

5.6.3 Software Goals

The team completely met one of the three software goals, partially met one goal, and did not meet the highest-level goal relating to coordinating neck movement to meet interactivity goals. While progress was made towards furthering all goals, the higher-level goals were unable to be met largely due to hardware dilemmas addressed earlier.

Low-Level Arm Control: An API for controlling Hydro Muscles by controlling muscle length and force of each Hydro Muscle,

To manage a suite of an arbitrary number of Hydro Muscles an API was built on top of I2C. This API provided methods for concurrently controlling the length of multiple Hydro Muscles as well as updating their controller configuration. Each Hydro Muscle is directly controlled by a closed loop PID controller with inputs from a calibrated linear potentiometer and outputs to a handmade CRFC valve. While the team did not implement PID control for the force of the Hydro Muscles, this measure was not needed to complete other goals as the robot was not expected to do load bearing work.

Mid-Level Control: An API for controlling for changing the state of neck sub sections in terms of XYZ position will be created. These sections include neck functional groups and the head functional group.

To control the pose of neck vertebrae subsections, a kinematic model of the robot was partially implemented based on the implementation found in the elephant trunk manipulator paper. The I2C network was expanded to provide methods for obtaining kinematic parameters from vertebrae sections. These parameters were calculated based on the change in muscle length. While these parameters could be obtained from vertebral sections, and used in the homogeneous transform matrix, they could not be used to command change in pose in terms of XYZ position. In addition, while the parameters could be calculated, the weight of the neck and vertebrae imparted axial rotation to the vertebrae and prevented continuous curvature in some instances. This prevented accurate control or actuation of the neck sub-assemblies.

The head sub assembly, composed of the neck, eyes, and jaw, are fully functionally and can actuate each element within half a second.

High Level Control: This API will convert camera data into actions for the *Deinonychus* to

The actions of the *Deinonychus* robot were able to be commanded via a python program and coordinated with a speaker to allow the robot to roar. The team also configured a raspberry pi to be able to obtain information from the robot's serial cli and camera. However, camera data was not used to direct robot behavior due to time constraints and hardware issues.

Reach Goal: Incorporate audio localization.

The team was not able to integrate audio localization to achieve this reach goal.

5.6.4 Interactivity Goals

Kiosk Mode: Ambient actions and sounds

The robot can execute a test sequence in which the spine, eyes, and jaw moves, and audio is emitted, suggesting ambient motion.

Kiosk Mode: Specific actions based on sensed range of interacting subject

The robot is not capable of sensing an interacting subject; therefore, the range-based behaviors were not met.

Grab Mode: Detects an object placed in its workspace and moves to grasp the object with its jaws.

The robot is not capable of sensing a ball, but the robot can grasp a 3-inch diameter object in its jaws.

Lifelike synthetic skin based on current scientific knowledge and the public perception of the *Deinonychus.*

The dinosaur has a full synthetic skin made of fabric, liquid latex, and paper towels and was painted with an airbrush. The teeth were 3D printed in white PLA and the eyes were painted with primer, acrylic paint, and then sealed with gloss spray.

Figure 50: Final Skin for the Deinonychus (top) & Painted Glossy Eyes (bottom)

Movements will give the impression of life by incorporating the following movement techniques: Overlapping actions, Follow through, Ambient Motion (S. Snibbe, M. Scheeff, and K. Rahardja, "A layered architecture)

During the robot's test sequence, the system simultaneously actuates three degrees of freedom in its neck, continuously generates new eye positions, and opens its mouth. Due to these simultaneous actions, the overlapping action method was achieved. The dinosaur has fluid motions due to the nature of the hydraulic muscles. Because of these fluid movements, the follow through method was achieved. In the robot's test sequence, different timings can be set between motions, allowing for the ambient motion method to be completed. Readers interested in further explanations of these types of motion should refer to (S. Snibbe, M. Scheeff, and K. Rahardja, "A layered architecture)

Reach Goal: The robot will feature life-like anticipation in its movements.

The robot does not perform any form of anticipation, such as looking at its new goal location. Therefore, this reach goal was not met.

6 Discussion

Though the robot was capable of being actuated by the Hydro Muscles, the team encountered many problems. During the process of developing the project, the team encountered delays in shipping, manufacturing, and reduced access to the lab due to COVID19. The team will be continuing to work on the robot to produce a more effective system that achieves the original design goals to a greater extent.

6.1 Spine & Head Actuation

The final vertebra design chosen was not optimal and requires further redesign. The vertebrae that were used on this robot (Design 9) do not move as fluidly as expected and introduce problems when actuated. The Bowden tubes worked as expected but distorted over time. The Spiderwire tendons worked sufficiently. Using rubber bands as passive elastic actuators negatively affected actuation. The spine could not hold its own weight and had to be tied to the ceiling to actuate the robot. The main issues the team experienced were low actuation and slow actuation.

The skull's full actuation was achieved. The eyes are coordinated with respect to each other, and they are capable of simple random motion if the system has power. The jaw can open and close. The team encountered a few issues with the assembly of the head and the orientation of the jaw servos. The team believes the main actuation problems were due to the following:

Friction & Gravity

During final system integration, pulling the wire to manually actuate the spine resulted in a noticeable amount of resistance. This resistance is caused by many elements. The most dominant cause appears to be friction between the ball and cup surfaces. 3D printed surfaces are not smooth enough to fluidly rub surfaces. In addition, the Bowden tubes distorted after continuous use. After repeated tests, the Bowden tube in some spaces curled as it tried to force itself through the hole it was adjacent to. This curling causes friction and therefore a loss of energy, which negatively affects actuation.

The robot had to be tied to the ceiling to counteract the force of gravity and hold the spine up when the head was attached. The head was the greatest source of weight. The team tried to increase the amount of passive elastic elements (rubber bands) to add rigidity to the spine but found that if there were enough rubber bands to hold the neck up, they counteracted the Hydro Muscles and limited actuation.

Hydraulic Choices & Leaks

Based on the team's original estimate, the maximum pulling force required by each Hydro Muscle was about three pounds (considering dorsal flexion of Section C). The team chose a final Hydro Muscle design capable of fulling 5 pounds to account for friction and efficiency losses in the system. Despite this over-estimate, the team found that the muscles were not strong enough

to keep the vertebral column in place and actuate each section without assistance from a ceiling-mounted tether to support the weight of the structure.

The team also encountered issues that restricted the muscle extension and therefore reduced the flexion of the neck. The extension of the Hydro Muscles is limited by the maximum extension of the muscle sheathing. The Hydro Muscles were sheathed in one-way-stretch fabric that allows the Hydro Muscle to increase by 1.5 times in length without increasing in diameter. However, this sheathing limits the ability of the muscles to extend to a great enough extent to cause the desired deflection. In addition, energy is lost if the Hydro Muscle increases in diameter, and force output of the muscle is reduced. The sheathing used on the Hydro Muscles was somewhat loose around the tube, allowing radial expansion to occur and taking away from the muscles' pulling force capabilities.

The team originally estimated that a change in tendon length of about three inches was required to fully deflect each section of the vertebral column. As such, the team chose to use Hydro Muscles capable of providing a change in tendon position of four inches to provide an extra inch of extension. However, in practice the deflection provided by the muscles was unable to fully deflect the actuated sections. This problem may be due in part to the length change of the muscles as well as the inability of the muscles to pull with the necessary force to deflect the sections. These factors both contributed to the lack of full deflection in the spine.

Water is an incompressible fluid. The rate at which the muscles pressurize depends on the capacity of the pump being used and flow rate of water into the muscles. The tubing that goes through the valves and therefore is the exit and entry point to the Hydro Muscle has a smaller than ⅛ inner diameter, so the amount of water that can leave or enter the tube is restricted. The team noticed the muscles would shrink slower than they extended. The muscles are actively pressurized by the pump when extending but rely on passive draining into the reservoir when contracting. The slow passive drainage from the muscles resulted in relatively slow contraction of the muscles when compared to their elongation. Due to the setup of Hydro Muscles in antagonistic pairs, this translated into slow system movements.

The team found that the Hydro Muscles tended to leak at the end caps and front nozzles after continuous use in the system. The muscles were under continuous tension from the tendons, and over time the tension began to pull the caps and nozzles out of the muscle tubing. To prevent this from happening, the team added a thin layer of epoxy along the muscle tubing seams at both ends of the muscle. This was found to effectively increase the life cycle of each muscle; however, a few muscles persisted to have slow leaks despite the additional epoxy.

The team initially encountered issues with periodic leaks in the hydraulic system tubing. After the initial assembly of the system, some of the tubing connectors were found to leak when the system was pressurized. The team hypothesized that this may have been due to certain barb connectors not being inserted far enough into the tubing, as the rigidity of the tubing made it difficult to insert the barbed connectors. To fix the leaking, the team added Epoxy to connections with minimal leaking and reattached connections with more serious leaks. The team also used metal clamps to secure the barbs in larger tube connections. These changes were effective at eliminating leaking in the hydraulic system.

Design & Fabrication Flaws

The team 3D printed most of their parts. 3D printing is convenient, but the fabrication method has some flaws. Because the skull's pieces were 3D printed and they were irregular shapes, printer supports needed to be used. This need for supports resulted in a large block of extra filament to be produced during the printing process. It was difficult to differentiate small details from the supports when removing the extra material and part of the jaw motor mounting plates were damaged when the supports were removed, making it difficult to line the servos up in a way that allowed them to properly mesh with the jaw gears. Additionally, the eye motion was restricted due to friction between the 3D printed piece that they rotated on and their outer surface. To mitigate this, the team sanded down the peg. Also, 3D printed gearing, though convenient for prototyping, tends to be unreliable for long term use. Elastic deformation of the gears may have led to some irregular actuation of the dinosaur jaw.

Minimal supports were designed along the axis of rotation of the skull-vertebra revolute joint which resulted in its unintended function as a cylindrical joint leading to the sporadic loss of the end effector. This issue was mitigated by adding elastic supports to stabilize the joint, but a more robust solution should be explored.

Suggested Solutions

The team believes a change in vertebra design will be beneficial for the progress of this project. The friction between 3D printed surfaces was much greater than the team anticipated. A design with U shaped notches like how a metal chain connects that contain low-friction bearings could greatly reduce the friction between vertebrae. A modified version of vertebra prototype four could be suitable. This design also has a large central channel which could contain the wire for actuation and remove the need for any Bowden tubes. Additionally, or alternatively, the length of the vertebrae prongs could be extended. Force = Torque/moment Arm therefore if the prongs (moment arms in this case) were longer, the required force to actuate the neck could be reduced. A downside of a longer moment arm is an increased change in muscle length to reach full flexion.

The team may consider reducing the length of the actuated sections. The team originally planned to have section A to have three vertebrae, but in a test to see if actuation would change, a vertebra from section B was moved to section A, making section A four vertebra long and section B three vertebra long. Post project, the team observed that with vertebra design 9, the length needed to fully actuate three vertebra with one set of rubber bands between them was three inches, while four inches were needed to actuate four vertebrae, meaning that by making section A four vertebrae long the resultant flexion was actually reduced. The actuated sections should be the shorter sections so that more flexion is produced with the same change in length of the Hydro Muscle.

The team noticed that the lateral flexion was uneven. The robot could turn much farther to its right than to its left. The team believes that this is because the right wire was tied much tighter than the left wire. When tying antagonistic pairs, a system should be developed to allow the dinosaur to have even flexion on either side of its center position.

For the hydraulics system, the team may consider modifying the connecting tube lengths. The valves empty through 2-foot long ⅛-inch inner diameter tubes. The team believes that by

trimming these tubes and leading all the small tubes into a single 1-inch inner diameter tube could help increase the rate at which the muscles emptied.

The team could also fabricate a lighter head and remove the neck servo to reduce the weight on the end of the neck. The team 3D printed the *Deinonychus* skull in PLA with 10% infill to reduce the weight of the piece while keeping it structural. If the head were made from foam, it would reduce the weight even more, requiring less force from the Hydro Muscles to actuate the spine. The head is also the element farthest from the base, and therefore creates a great amount of torque, pulling the spine down with its large mass. The servo at the top of the spine could also be removed as it does not add much additional movement and it adds weight on the end of the continuum arm. Mechanical stops should be added to the new skull to prevent it from sliding off the spine or, if the top servo is removed, a more robust mechanical attachment should be developed. If the team decides to 3D print for the next skull prototype, increased tolerancing in the SolidWorks model should be done to help reduce friction in the eyes and aluminum gears may allow for a more durable and lightweight solution for the jaw than 3D printed gears can.

6.2 Muscle Boxes

While the final design of the muscle boxes met all their core design requirements, complications arose during assembly and testing which should be addressed in future revisions. Initial plans for muscle box Delta featured a laser cut wooden base. This part was intended to stabilize the system without restricting access to the electronics inside. It was not fabricated due to time constraints, and in its absence the boxes were mounted directly onto the wireframe table which supported the rest of the system. In its current configuration the table restricts access to the box's electronics which are mounted on the underside of the boxes. This has made reprogramming the Arduinos and testing the PCBs difficult. The next iteration of the muscle box should move the electronics to a more accessible location or include a superstructure which allows access to the underside of the boxes once they are mounted.

Over the course of the project the size of the muscle boxes increased dramatically. The original design housed four muscles inside a box with a four-inch square base. By comparison, Delta has a five inch by seven-inch base and houses half as many muscles. During assembly, the amount of extra space and the thickness of the walls were remarkably excessive. The bulk of this model led to long print times (over two days to fully print one box) and forced the team to print the boxes with lower resolution and without supports. This exasperated existing tolerancing issues in the design and made revising the boxes impossible given the project's timeframe. The next iteration of the muscle boxes should refine the existing system and remove excess material.

The recessed portions of the box containing the PCB and potentiometers were 0.05 inches larger in all dimensions than the parts themselves. This tolerance was based on previous parts printed at the WPI Makerspace on Ultimaker 3s. The muscle boxes were printed on a different machine as the Makerspace was unable to accommodate parts with such a long print time. During assembly, the team discovered issues fitting the electrical components into the boxes caused by dimensional deviance of the electrical components and insufficient tolerancing of the printed boxes. These issues were resolved by filing down the edges of the PCBs, sliders, and potentiometers. Future work should increase the tolerancing of the muscle box.

Four aluminum guidance rods and six wooden support/safety rods connect the upper and lower sections of the model delta muscle box. In theory these were supposed to be interchangeable to allow the box to accommodate larger Hydro Muscles. In practice they were too complicated to change out. The team found it simpler to 3D print extenders to pre-tension the longer muscles rather than buy new rods. This resulted in the team feeding the muscles through the hole in the back of the box that was supposed to house the inlet tube, not a Hydro Muscle. The muscle's sheath required a ½ inch of allowance on the outside edge of the stitch to help reduce unraveling of the one-way stretch fabric. This need for extra fabric rendered the hole in the back of the muscle box in which the muscle was fed water, too small, as the muscle was not able to pass through the whole cleanly. The team bored out these holes to be larger to allow for more space. If adaptability remains important for future iterations of this project, the current design will need to be simplified.

6.3 Control

The I2C network is functioning as expected. The network provides fast and reliable communication between the primary and secondary Arduino Nano sections to enable the muscle controller network. This network is expandable and currently contains implementations for scanning the network and controlling the lengths of muscles in the network. In order to improve the I2C network, some changes to the message format could be made. As of now, the maximum number of commands is limited to 254 commands and 254 requests for data as the message type and request for data types is represented by an unsigned char with 255 total values and some values are used as defaults. Changing the representation to an integer may simplify the secondary Arduino network implementation and greatly increase the maximum number of possible messages.

While the networks connecting the Arduinos in the system are effective, controlling the vertebrae sections using the kinematic model is currently not fully implemented. While portions of the forward kinematic model can be used to measure some aspects of neck pose, the kinematic model remains largely untested and cannot be used to obtain a given end effector position. The fundamental lengths of muscles can be measured using a linear potentiometer and the deflection along the axis being pulled by the muscles can be estimated based on these measures, along with configuration data about the dimensions of neck vertebrae. Perpendicular deflections of each neck section can then be converted into the rotation (phi) and deflection (k) used in the homogeneous transformation matrix for the neck section and reported by each secondary Arduino over the network. This deflection calculation was tested experimentally on a single section while the section was under no load and seemed to be effective in this ideal case as the assumptions of zero axial deflection and bends without kinking were true. The command to obtain a given deflection and rotation is also supported by the I2C network, but the section cannot obtain the position requested. It is unable to do so because the weight of the skull and neck imparted axial deflection which breaks an assumption of the model. In addition, the previously mentioned difficulties with the neck prevented the muscles from obtaining the desired lengths while experiencing the weight of the neck. To be able to achieve a given end effector position, or obtain a more complex neck shape, the forward kinematics need to be validated to be accurate under load.

Achieving complex behavior such as scanning the room, tracking faces, or obtaining a given pose has not been implemented due to the difficulties in achieving basic motion. The robot could be made to reach these goals using several techniques given the necessary improvements to muscle actuation. Simple facial tracking could be achieved using a relative control scheme based on camera data. This method would work by having the robot deflect a section to keep a face centered in a camera's field of view. Determining the inverse kinematic model allows for the robot to calculate the muscle lengths required to have the robot obtain a XYZ position for the end effector as well as for the top of each section. This capability would allow the robot to move between XYZ waypoints to create different poses, and track faces via changes in XYZ position of the head relative to the position of the face in a camera's field of view.

Behavior of the robot is currently controlled using the robots command line interface. This interface allows a user or program to interact with the robot and is functional and expandable. Interactions supported include changes in muscle pose and configuration, changes in head section pose like eye position, and changes in data about the status of these subsystems. The command line interface does not currently support changes to robot subsection or end effector pose in terms of XYZ coordinates. These functions can be added as the kinematic model is improved.

6.4 Interaction & Integrated System Operation

The team faced their greatest problems during system integration. Ambient motions were completed during the experiments through the robot's test sequence, but the robot is unable to sense a subject to interact with. Integrating the Raspberry Pi camera will allow the team to reach their interactivity goals.

In the early stages of the project, the spine was actuatable by hand and the skull's servos were able to actuate the eyes and jaw. Connecting the skull to the spine greatly increased the weight the neck needed to support. The hydraulic system was able to power the 6 muscles and the computer network was able to control the muscles, but when the spine's tendons were attached to the muscles, the head's weight and the vertebrae's friction became apparent. The team suspended the dinosaur's neck from the ceiling by tying a fishing line to the top vertebra. This solution removed the need to fight gravity and the muscles were then able to actuate the system. Though the resultant flexion was not as great as the team wished, the spine can be improved with the range of solutions outlined earlier and the team will reduce the head weight so the team can eventually remove the suspension wire.

Together, removing this wire and increasing flexion will allow the team to implement the forward and inverse kinematic models for the robot. These models will allow the team to give the program muscle lengths and produce a (x, y, z) position for the robot end effector, or give the program a desired (x, y, z) position and have the muscles move to the desired length to produce this position, which is an important goal for the team's summer work. The latter model is more important for the project, as this mathematics will allow the camera to measure a location and send the robot to a desired pose. This pose can then be verified by implementing the inertial measurement units on each section, as the inverse kinematics of a continuum arm is implemented by compounding the position of the individual segments.

The power and networking worked as planned but the mechanical aspects of the robot must be improved and many of the elements of the electrical system must be utilized to allow the programming aspect of the project to reach its full potential.

6.5 Conclusions & Future Work

The team fully designed, fabricated, powered, and programmed the robotic *Deinonychus* between the summer of 2020 and May of 2021. The team began by researching the anatomy of the dinosaur and connected with paleontologists and other relevant experts. Then the team developed nine different vertebra prototypes, assessing weight, size, flexion, and other parameters with each iteration. The spine was assembled, and the hydraulics system was prototyped and assembled. After battling months of leaky valves and tubes, the final hydraulics system was integrated with the muscle box and the networking/power system. The head was designed, manufactured, and assembled and attached to the spine. The spine was suspended to negate gravity and the robot was successfully actuated by the Hydro Muscles.

The team plans to continue to work on their *Deinonychus* robot over the summer of 2021. This future work will take into consideration the suggestions made in the Discussion section of this paper and will focus on reaching a variety of goals. The priority of this future work will be to improve the actuation of the spine and integrate sensors for interactive modes. These modes will be like the interactivity modes described in the original project goals. Additional software will be developed to integrate the IMUs and camera, while the kinematic model of a continuum arm will be used to allow the dinosaur to follow a subject.

The team was inspired by animatronics and robots that are often seen in public places such as theme parks and movies. The field of robotics is increasing in potential every year. While robots are typically restrained to factories, they also have potential to interact with humans. Human robot interaction is an aspect of robotics that is still being explored, and the team believes interactive robots have potential to entertain, amaze, and educate. The team wanted to create a robot that allowed an individual to feel as though they were interacting with a member of a species that has been extinct for over one-hundred million years. The team hopes that one day Perry the *Deinonychus* may prove to be a prototype of an interactive science museum display.

7 Bibliography

[1] M. W. Hannan and I. D. Walker, "Kinematics and the Implementation of an Elephant's Trunk Manipulator and Other Continuum Style Robots," *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003.

[2] F. Passantino, *deinonychus*. YouTube, 2017.

[3] "The History of Animatronics," *Stan Winston School of Character Arts*, 23-Jul-2015. [Online]. Available: https://www.stanwinstonschool.com/blog/animatronics-world-review.

[4] *Na'vi Shaman, Disney's Highly Expressive Humanoid Alien Robot*. YouTube, 2019.

[5] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, 2015.

[6] *Kronos II – Animatronic Dragon – Person Tracking*. YouTube, 2015.

[7] *Kronos III – Animatronic Dragon – Progress Update 1*. YouTube, 2020.

[8] K. Bostwick, "Feathers and Plumages," *Handb. Bird Biol.*, pp. 101–147, 2016.

[9] National Science Foundation, "*Discovery of fossil "voice box" of Antarctic bird suggests dinosaurs couldn't sing"*, 2016, Accessed on Sept. 21 2020 [Video File]. Available:https://www.youtube.com/watch?v=NtnXsaXt-8E&feature=emb_logo

[10] M. Kosko, "Can You Figure Out How Dinosaurs Walked?," *Science Friday*, 06-Aug-2019. [Online]. Available: https://www.sciencefriday.com/educational-resources/how-do-you-figure-out-how-dinosaurs-walked/. [Accessed: 15-Sep-2020].

[11] J. H. Ostrom, "Osteology of Deinonychus antirrhopus, an Unusual Theropod from the Lower Cretaceous of Montana," *Osteol. Deinonychus antirrhopus, an Unusual Theropod from Low. Cretac. Mont.*, no. July, 2019, doi: 10.2307/j.ctvqc6gzx.

[12] M. J. Cobley, E. J. Rayfield, and P. M. Barrett, "Inter-Vertebral Flexibility of the Ostrich Neck: Implications for Estimating Sauropod Neck Flexibility," *PloS One*, vol. 8, no. 8, 2013, doi: 10.1371/journal.pone.0072187.

[13] E. Snively and A. P. Russell, "Functional variation of neck muscles and their relation to feeding style in Tyrannosauridae and other large Theropod dinosaurs," *Anat. Rec.*, vol. 290, no. 8, pp. 934–957, 2007, doi: 10.1002/ar.20563.

Bibliography

[14] E. Snively, J. R. Cotton, R. Ridgely, and L. M. Witmer, "Allosaurus feeding Multibody dynamics model of head and neck function in Allosaurus (Dinosauria , Theropoda) Allosaurus Musculoskeletal Anatomy Multibody Dynamics of Head and Neck Motion Feeding Apparatus Dynamics of Allosaurus : Goals and Hypotheses Bo," pp. 2006–2007, 2013, doi: 10.1098/rsbl.2012.0056.Bates.

[15] S. Sridar, C. Majeika, P. Schaffer, M. Bowers, et.al, Hydro Muscle – A Novel Soft Fluidic Actuator

[16] J. D'Agostino, E. Clarrissimeaux, S. Moffat, and J. Florez-Castillo, Development of Bioinspired Exosuit Actuated with Hydro Muscles and Novel Compact Robotic Flow Control Valve

[17] "Wire," *Arduino*. [Online]. Available: https://www.arduino.cc/en/reference/wire. [Accessed: 15-Apr-2021].

[18] Dejan, "How I2C Communication Works & How To Use It with Arduino," *HowToMechatronics*, 05-Feb-2021. [Online]. Available: https://howtomechatronics.com/tutorials/   pprox./how-i2c-communication-works-and-how-to-use-it-with-arduino/. [Accessed: Apr-2021].

[19] *How They Made Jurassic Park's T-Rex – Sculpting a Full-Size Dinosaur*. YouTube, 2012.

[20] S. Snibbe, M. Scheeff, and K. Rahardja, "A layered architecture for lifelike robotic motion," Proceedings, 1999.

[21] R. E. Kambic, A. A. Biewener, and S. E. Pierce, "Experimental determination of three-dimensional cervical joint mobility in the avian neck," Frontiers in Zoology, vol. 14, no. 1, 2017.

[22] F. Kreith, *Fluid mechanics*. Boca Raton, FL: CRC Press, 2000.

[23] A. Barth, J. Sorrells, S. Ueda, and J. Wu, "Hydro Muscle Control System", 2015

8 Appendices

Appendix A: Official Goals Document

By May 2021 Samantha Grillo, Alec Kneedler, Kalani Picho & Kenneth Rhodes will have completed a Major Qualifying Project worthy of credit for Robotics Engineering, Mechanical Engineering, Electrical Engineering, and Computer Science Majors at WPI. To this end, the team will design and implement a robotic *Deinonychus* head and neck platform capable of the following:

Movement: The project has a focus on approximating the head and neck movement of the *Deinonychus* as determined by the range of vertebral flexion found in two analogs, with minimum and maximum flexion determined by the *Allosaurus* and the turkey respectively. The vertebrae will be capable of up to three kinds of movement: dorsal/ventral flexion, lateral flexion, and axial rotation. The resulting target movement angles can be found in the Mechanical Objectives section.

Interaction: This project aims to create the impression of a living *Deinonychus* via synthetic skin and audio based on current scientific knowledge and the public perception of the *Deinonychus*. The robot will also feature two distinct interactive modes:

1. A kiosk mode that performs ambient actions and sounds. When a spectator is detected, the robot reacts to visitor movements by following the face of the spectator, creating the illusion that the robot is "watching" the spectator, and emitting audio.
2. A grab mode during which the robot will locate and grab onto a ~ 3 inch diameter object in the robot's workspace[2].

Both of these modes will be associated with a final demo deliverable demonstrating each mode's functionality.

Listed Project Goals:

Mechanical Objectives

General Goals

❏ Achieve a fully-actuated vertebral column consisting of fabricated vertebrae analogs capable of up to three kinds of movement: dorsal/ventral flexion, lateral flexion, and axial rotation. The vertebral column will be actuated by a system of Hydro Muscles. The vertebral column will be capable of the following ranges of motion with respect to the robot's starting position:
  - ❏ At least 15 degrees of dorsiflexion and 45 degrees of ventral flexion (Figure 1)
  - ❏ At least 40 degrees of lateral flexion in each direction (Figure 1)
    Please see Figure 1 for an example of measured *Allosaurus* neck dorsal/ventral and lateral flexion capabilities.

---

[2] As of 9/23/20 the robot workspace is expected to be a four foot radius.

- ❏ Achieve head and neck motion capable of:
  - ❏ Operating speeds up to 15 in/sec
  - ❏ Spatial accuracy of +/- 1 inch from the target end effector (head) location
  - ❏ Rotational accuracy of +/- 10 degrees from the target angle
- ❏ Develop a *Deinonychus* head that is capable of three degrees of rotation at the joint where the head meets the neck. The head will include:
  - ❏ A mobile jaw assembly that is capable of grasping a stationary object with a ~3 inch diameter located in the robot workspace.
  - ❏ Eyes with simple random or repetitive movement commonly seen in modern animatronics.

Reach Goals

- ❏ Integrate complex eye movements which can include coordination between eyes, movement to follow objects, blinking, etc.

Electrical Objectives

General Goals

- ❏ Implement a multiple-sensor suite to collect position and environmental information to enable autonomous functionality.
  - ❏ Include a microphone
  - ❏ Include camera(s)
  - ❏ Include Inertial Measurement Unit(s)
  - ❏ Any unlisted sensors deemed necessary for completing general goals
- ❏ Enable software goals by supporting one or more microcontrollers to control key sensory activities.
- ❏ Develop a robust control system for coordinating artificial muscles & motors, and support system power needs.
- ❏ Incorporate a speaker to allow the robot to emit audio.
- ❏ Transfer final electrical schematic(s) to custom printed circuit boards.

Reach Goals

- ❏ Include servo(s) in the head to achieve coordinated eye movement.

Software Objectives

General Goals

- ❏ Low Level Arm Control: An API for controlling Hydro Muscles will be created. This API will provide control over the length and force of each hydro-muscle, and be able to provide measurements about the Hydro Muscles such as their maximum pull length, the current pull length, and the current pull force for each muscle via a sensor suite. Muscles will be able to be grouped into antagonistic pairs which will restrict how they can move relative to each other.
- ❏ Mid Level Control: An API for controlling and coordinating neck sub assemblies will be created. These sub assemblies will correspond to the neck vertebrae functional groups and the head functional group. The neck functional groups will consist of vertebrae, antagonistic Hydro Muscle pairs, and sensors to record orientation and position such as an IMU and cameras. The head functional group will consist of an articulable jaw, eyes, and sensors to record position. The API will provide controls for changing the state of the subassemblies from one to another over a given time interval as well as for obtaining the current state for the sub assemblies. This will be accomplished using the low level API and should create motion that appears

natural. In order to appear responsive the robot should be able to respond to higher level commands within half a second. State information includes measurements about the XYZ position, orientation, position of jaw, position of eyes.

❏ High Level Control: This API will convert camera data into actions for the Deinonychus to take. These actions will fall under two modes that are described in the interactivity goals. There will be a kiosk mode, where the dino searches for people to look at and an acquire target position mode where the dino will and try to grasp a target that comes within its workspace. This will involve recognizing faces/objects, calculating their distance/location, and coordinating a response to the input via the Mid level control.

Reach Goals

❏ Incorporate audio localization.

Interactivity Objectives

General Goals

❏ The robot will feature two distinct interactive modes:

  ❏ Kiosk Mode:

    ❏ The robot performs ambient actions and sounds.
    ❏ Specific Actions (measured from safety zone)
        ❏ Long Range (10 ft or max detection range): Curious sounds? Robot looks at the visitor with one side of the head.
        ❏ Mid Range (3-10 ft): When a spectator is detected within mid range the robot moves to follow the face of the spectator, creating the illusion that the robot is "watching" the spectator. If there are multiple spectators in this range it will change its focus between them to keep track of them.
        ❏ Close Range (0-3 ft): The robot recoils and faux pounces at the visitor with an associated "roar".

  ❏ Grab Mode: the robot detects an object of ~1 inches in diameter placed within its workspace (pprox.. 4ft semicircle around the robot), and moves to grasp the object within its jaws.

❏ The robot will feature synthetic skin based on current scientific knowledge and the public perception of the *Deinonychus.*

❏ The robot movements will give the impression of life [1] by incorporating the following movement techniques:

  ❏ Overlapping actions
  ❏ Follow through
  ❏ Ambient Motion

Please see Figure 4 for an in-depth description of these movement techniques.

Reach Goals

❏ The robot will feature life-like anticipation in its movements.

Figures
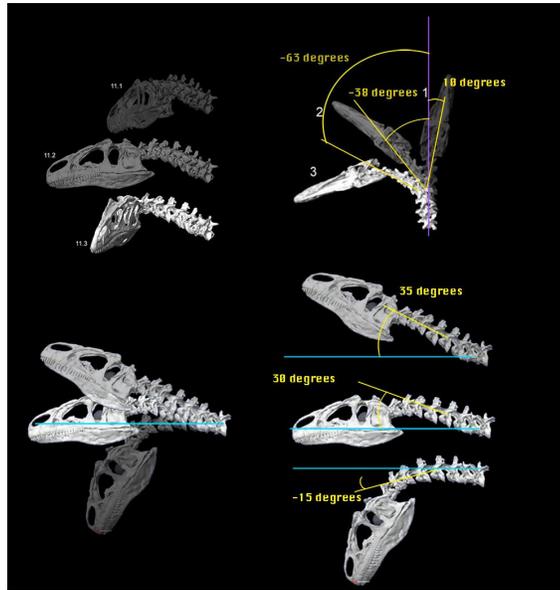
Figure A- 1: Allosaurus neck assessment, angles of fourth vertebra in three poses. Figure based on images from [14]
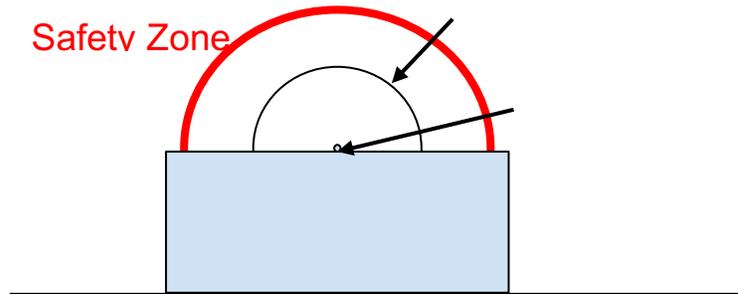
Figure A-2: Overhead of the robot workspace, estimated to be 4ft, and safety zone, estimated to be 7ft
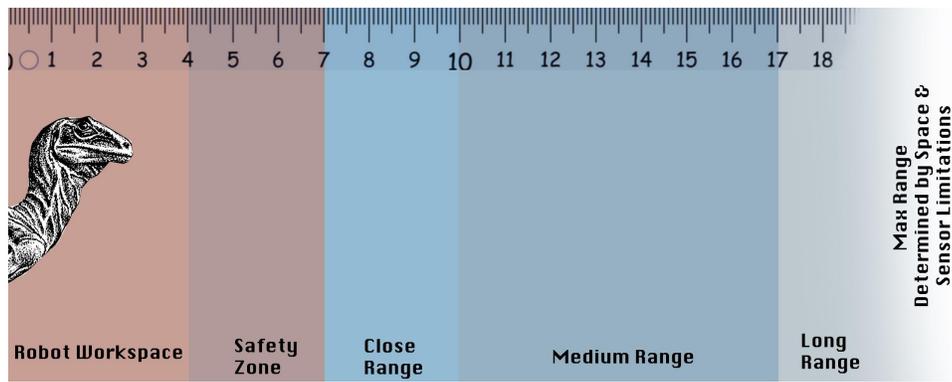
Figure A-3: Interaction ranges, with estimated robot workspace = 4ft, modified image by

Overlapping Action. Actions overlap in space and time. For example, while a person walks towards a door, he is reaching for the doorknob at the same time. This term also refers to the time lag between body parts – for example, a character's eyes will follow the point he is tracking much more quickly than his head moves.

Follow Through. Actions do not come to an immediate halt but continue on after the end-goal of the motion is achieved, as in a baseball swing.

Anticipation. Before entering into an action, a character will anticipate such action, giving the impression of thought or preparation.

Arcs. Objects follow curved paths in the real world, due to body geometry and basic physics.

Ambient Motion. Real creatures never come to a complete stop – there is continual movement and rhythm such as nodding, blinking, bobbing and shifting of weight.

Ease In/Ease Out. Motions begin and end with a ramp up to speed and a slow down to a halt. It is impossible for real creatures to make immediate and precise changes in position or velocity.

Figure A-4: Elements of Organic Motion, Base on Animation Principles [20]

Appendix B: Detailed Component Drawings
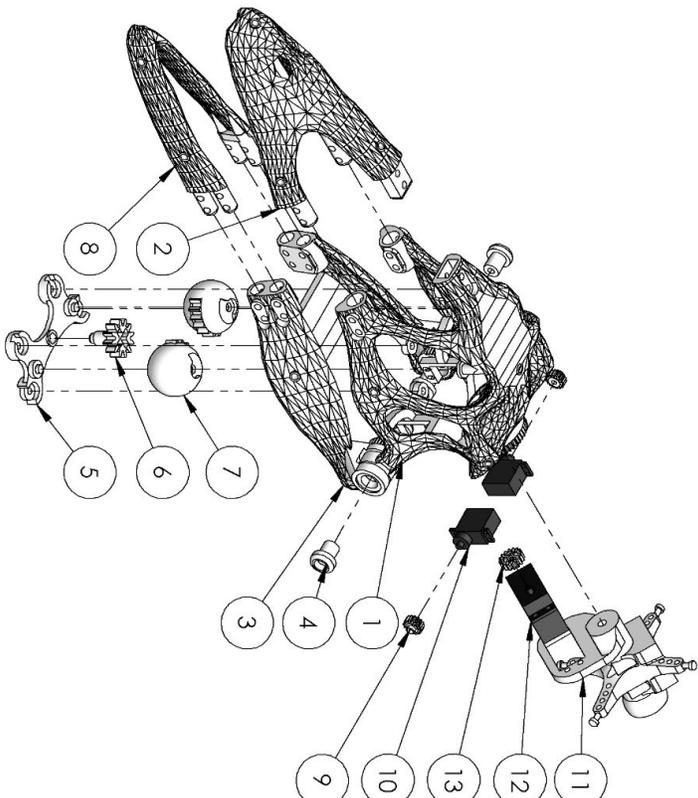
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	204-01	Upper Jaw Back	1
2	204-02	Upper Jaw Front	1
3	204-03	Lower Jaw	1
4	204-04	Skull Hinge	4
5	204-05	Lower Eye Support	1
6	204-06	Pinion Gear Alpha	1
7	204-07	Eye alpha	2
8	204-08	Lower Jaw Front	1
9	204-09	16 tooth 32nd	2
10	204-10	Servo_Model	2
11	204-11	topIMU	1
12	204-12	neck servo	1
13	204-13	16 tooth M1	1

WORCESTER POLYTECHNIC INSITITUE

TITLE: **ACTUATED SKULL SUBASSEMBLY**

SIZE **A** | DWG. NO. **204-14** | REV **D**

SCALE: 1:4 | WEIGHT: | SHEET 1 OF 3

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ± 0° 30'
ONE PLACE DECIMAL ±.1
TWO PLACE DECIMAL ±.01
THREE PLACE DECIMAL ±.005

INTERPRET GEOMETRIC TOLERANCING PER: ANSI Y14.5

MATERIAL PLA
FINISH MATTE
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATK	4/28/2021
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

NEXT ASSY | USED ON
APPLICATION

WORCESTER POLYTECHNIC INSITUTE

TITLE:
ACTUATED SKULL SUBASSEMBLY

SIZE A   DWG. NO. 204-14   REV D

SCALE: 1:2   WEIGHT: SHEET 2 OF 3

	NAME	DATE
DRAWN	ATK	4/28/2021
CHECKED		
ENG. APPR.		
MFG. APPR.		
Q.A.		
COMMENTS:		

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ± 0° 30'
ONE PLACE DECIMAL ±.1
TWO PLACE DECIMAL ±.01
THREE PLACE DECIMAL ±.005

INTERPRET GEOMETRIC
TOLERANCING PER: ANSI Y14.5

MATERIAL
PLA

FINISH
MATTE

DO NOT SCALE DRAWING

NEXT ASSY   USED ON

APPLICATION

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

WPI

6.18
5.78
4.50
6.55

6.53
3.83

Ø.20
R.50
Ø.75

CUSTOM GEARS ARE AS FOLLOWS

D 12:29.8 TOOTH M2
E 16:32 TOOTH 1/32
F 16:48 TOOTH M1

DETAIL E
SCALE 2 : 3

DETAIL D
SCALE 2 : 3

DETAIL F
SCALE 2 : 3

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS
DRAWING IS THE SOLE PROPERTY OF
<INSERT COMPANY NAME HERE>. ANY
REPRODUCTION IN PART OR AS A WHOLE
WITHOUT THE WRITTEN PERMISSION OF
<INSERT COMPANY NAME HERE> IS
PROHIBITED.

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ± 0° 30'
ONE PLACE DECIMAL ±.1
TWO PLACE DECIMAL ±.01
THREE PLACE DECIMAL ±.005

INTERPRET GEOMETRIC
TOLERANCING PER: ANSI Y14.5

MATERIAL
PLA

FINISH
MATTE

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATK	4/28/2021
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

NEXT ASSY | USED ON

APPLICATION

WORCESTER POLYTECHNIC INSTITUTE

TITLE:
ACTUATED SKULL
SUBASSEMBLY

SIZE **A** | DWG. NO. 204-14 | REV D

SCALE: 1:4 | WEIGHT: | SHEET 3 OF 3

Appendix C: Fabrication Instructions

Construction of the CRFC Valves:

The team found that the following process worked for them. To construct a smiley valve one needs the following:

Item	Quantity
Spiderwire	2, 8inch pieces, and 4, 4inch pieces
Oblong, 0.25 beads	ct. 2
3d printer filament	< 10 grams
⅛ inch ID Surgical Tubing	4 inches
Parachute Fabric	2, 2x2inch squares
Thick, braided thread	15 inches
Micro Servo	1
Servo Horn & Matching Screw	1
⅛ inch Diameter Screws	2
⅛ inch to ⅛ inch plastic barbs	4
⅛ ID plastic flexible tubing	4, 2" pieces
3d Printed Smile	1
3d Printed Servo Base[3]	1
Tools: Pliers, thick sewing needle, superglue, aquatic purposes epoxy (such as J-B Weld), screwdriver, vise	

Prepping the Tubes

Cut two, two-inch lengths of surgical tubing. Place one ⅛ th inch barb into either end so that they are snug within the tubing. Take a four-inch piece of spiderwire and tie the wire around the end of the tube, where the barb begins to thin out. Tie the spiderwire tight enough that you can see the indentation from the barb ending. Tie the other side the same way. Super glue over both strings and let the glue dry.

---

[3] It is important that these bases are printed so that their layer lines are parallel to the piece that holds the servo. If the layer lines are parallel to the surface it sits on, the force from the knots tied will cause the piece to snap in half.

Cut two, two-inch by two-inch squares of the parachute fabric. Thread the thick, braided thread through strong hand-stitching needles. Normal needles will break from the force of being pulled through the parachute fabric. Wrap the parachute fabric around the surgical tubing in a 'U' shape so the length of the surgical tube lays in the middle. Fold the parachute fabric in half, and starting at one end of the tube, and as close to the tube as possible without piercing it, push the needle through both layers of the parachute fabric and pull the thread a until there is about one inch of the thread left from the other side. Tie a standard double knot with the needle and thread and pull the knot tight. Proceed to complete a standard in and out stitch along the length of the tube. It is important that this stitch is done close to the side of the tube but without piercing it, so the tubing cannot increase diameter as it is pressurized. Once you have reached the end of the parachute fabric, tie another double knot with the thread and pull it tight.

Next, take a four-inch piece of spiderwire and tie the wire around the end of the tube, roughly where the first spirewire was tied. Tie the wire tightly and glue, as was done before the parachute fabric was added. Repeat on the other side.

Assembling the Base

First, place one small bead on each of the two, 8 inch pieces of spiderwire. Complete a bowline knot in the center of the wire piece so that the bead ends up on a loop that is about ¾inch long. See the diagram below: Take the 3D printed servo base and place a micro servo inside. Screw the servo in with the matching mounting screws. Center the servo with your code of choice. Align the servo horn so it is centered on the micro servo and attach it with the servo horn screw. Attach the 3D printed smile to the servo horn with two appropriately sized screws. Loop the beads onto either side of the smile and feed the two strings from each bead through the holes in the base. Next, cross the strings and insert a surgical tube assembly. It is important that the strings are crossed before the tube is placed in the base. This crossed string is what causes the surgical tube to be crushed and prevent flow, effectively closing the valve. Do not tie the strings yet.

Finishing the Valve

Tying the valves is a two person job. Once both tubes are inserted, flip the valve over and place the assembly up-side-down in a vice. Align the servo so that the horn is straight. Make sure the beads are on the smile, as forgetting this step will result in knots that are too tight. Start with the two pieces of wire that belong to the same tube. One person will need to tie a regular knot and pull it tight with pliers so that the other person can then use a small, flat object, such as an allen wrench, to press down on the knot. As the knot is being pressed down, the other member should tie another knot. That same team member should then pull on the strings with pliers so that the second teammate can then release their pressure on the first knot and then press down the second knot. Complete this process until there are three knots on both tubes. Once the knots are tied, epoxy over each barb-to-tube connection is fixed to the base. This step is to help prevent leaks.

Hydro Muscle Manufacturing Process

The following steps describe the manufacturing process the team followed to make the Hydro Muscles used in the project. These specific muscles use latex tubing with an ID of ⅜" and an OD of ½," but the process can be applied to muscles of other sizes by substituting the chosen ID and OD of the muscle tubing. To construct a single muscle, the following materials are needed:

Item	Quantity
Latex Rubber Tubing (⅜" ID, ½" OD)	Cut to desired length of muscle
Plastic Barbed Connector, ⅜" to ¼" Tube ID	1
Flexible Plastic Tubing (¼" ID, ⅜" OD)	~2 inches
Plastic Barbed Connector, ¼" to ⅛" Tube ID	1
Flexible Plastic Tubing (⅛" ID, ¼" OD)	~1 inch
Plastic Barbed Y Connector, ⅛" Tube ID	1
Plastic Barbed Stopper, ⅜" Tube ID	1
Metal Eyelet Screw, ½" Diameter	1
One-Way Stretch Fabric (1.5x Stretch)	Rectangular Piece (~6" width and ½" longer than length of muscle tubing)
Thin Sewing Thread	Enough to complete stretch stitch along twice the length of the muscle
Worm-Drive Hose Clamp	2
Tools: Dremel, scissors, measuring tape, sewing machine, superglue gel, aquatic purposes epoxy (such as J-B Weld), flathead screwdriver	

Assemble Inner Tubing

The first step is to create the barbed connector that will attach the front of the muscle to the CRFC valve. Glue the ⅜" to ¼" barb to one end of the ¼" ID tubing piece; this is done by placing a ring of super glue around the last ridge on the ¼" side of the barb and pushing the ¼" ID tubing onto the barb until secured. Allow the glue to dry for a few minutes, then use a similar process to glue the ¼" to ⅛" barb to the other side of the ¼" tubing. The resulting component will be a piece of ¼" ID tubing with a ⅜" barb protruding from one end and a ⅛" barb protruding from the other end. Next, glue the ⅛" ID tubing to the ⅛" barb on the ¼" ID piece of tubing assembled in the previous step. Finally, glue the ⅛" Y connector to the other end of the ⅛" ID tube so that

the two ends of the Y are sticking out of the tube. The resulting connector piece should look similar to the one shown below:
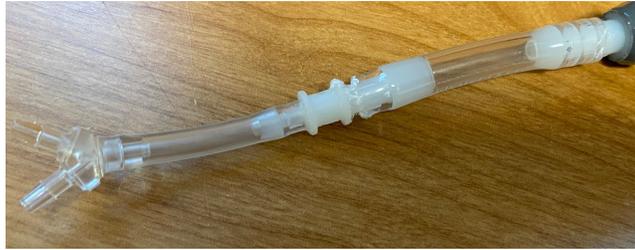
Figure C-1: Front barb connector for Hydro Muscle

Next, assemble the hooked plug that will attach the end of the muscle to the tendon. Use the Dremel to make a small indent in the center of the top of the plastic stopper. Screw the eyelet into the indent until secure, and add a small dab of Epoxy around the seam where the eyelet meets the top of the stopper. Allow the Epoxy to set as directed on the label, and continue.

Finally, attach the stopper and connector to the muscle tubing. For each piece, place a small ring of super glue around the upper ridge of the barb and push into the muscle tubing as far as possible. Add a thin layer of Epoxy around the seams at either end of the muscle tubing, and allow to set as directed. At the end of this step, the muscle tubing should have a connector at one end and a hooked stopper at the other.

Create Muscle Sheath

The next step is to prepare the muscle sheathing. Begin by cutting the one-way stretch fabric to the desired dimensions as specified in the materials list, making sure that the stretchable direction is oriented lengthwise along the piece of fabric. Fold the piece of fabric lengthwise, and use a sewing machine to sew a stretch stitch (or equivalent) lengthwise along the open end of the folded fabric. Next, insert the muscle tubing into the folded fabric, and use the tubing to mark where the second line of stitches should be placed. Sew another stretch stitch lengthwise along this line, and check the fit of the sheath on the muscle tubing. The sheath should be snug enough to prevent radial expansion but loose enough to allow the sheath to be pulled over the muscle. The sheath should look similar to the one below:

Figure C-2: Sewed Muscle Sheath

If needed, repeat this process until a proper fit is achieved. Once the correct fit is achieved, cut the folded end of the fabric and trim away any extra fabric.

Assemble Final Muscle

Finally, assemble the completed muscle. Slide the muscle sheath onto the muscle tubing so that the sheath is centered along the tube and the entire tube is covered. Slide one of the hose clamps onto the plugged end of the muscle and tighten the clamp so that it rests on the flat portion of the barbed connector on the plug (inserted into the muscle). Repeat the process for the barbed connector on the front of the muscle. The Hydro Muscle should now be fully assembled and ready to be incorporated into the hydraulic system.

Appendix D: Pressure Loss Diagram & Spreadsheet

The calculated pressure loss throughout the system is found by starting at the muscles and working backwards. Because the flow splits evenly at Y connectors, the pressure loss is followed along one flow path to a single Hydro Muscle, and the resulting flow at the end of path becomes cumulative for the entire system. The calculations were performed by solving at intermediate points as shown in the following diagram:

Figure D-2: Pressure loss intermediate steps, one muscle
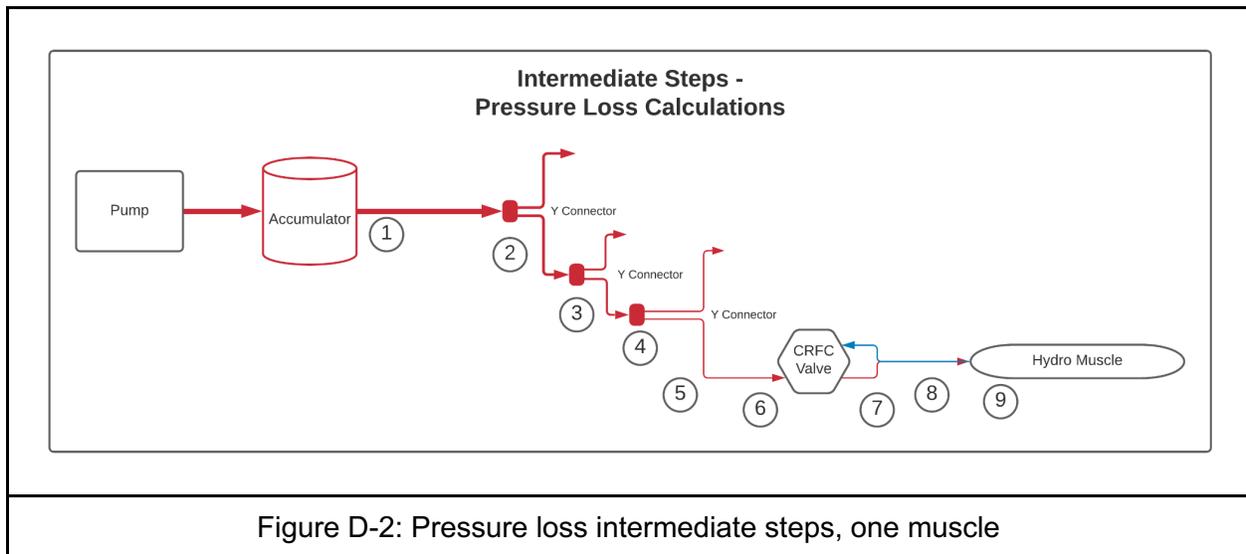
The following spreadsheets show the calculations for the pressure and flow throughout the system, making use of the equations discussed in Section 3.6.2.2. The calculations for an 8-muscle system of the original 8 inch muscles is shown first, followed by identical calculations for a similar system using 10 inch muscles.

Hydraulic System Pressure Loss Calculations - 8 Inch Muscles

gray = user input

Hydro Muscles

Inner Diameter (in)	0.375
Outer Diameter (in)	0.5
Change in Length (in)	4
Volume Change, 1 Muscle (in3)	0.4417734375
Speed to Fill One Muscle (s)	1
Volumetric Flow Rate (in3/s):	0.4417734375 (one muscle)
Volumetric Flow Rate (m3/s):	0.0000072393392 (one muscle)
Total # Muscles	8
Muscle Operating Pressure P9 (PSI)	50
Muscle Operating Pressure P9 (Pa)	344750

Segment 9-8: 1/4in Hard Tubing (1 Muscle) [Muscle to Soft Tubing]

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
velocity V9 (m/s)	0.2285990414
Tube Length (in)	2
Tube Length (cm)	5.08
Tube Length (m)	0.0508

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.2286	1444.367760	Laminar	0.000472441	0.04431005	9.243609	0.009243609	0.00134062

Resulting Pressure Drop (PSI, Pa):	0.00134062	9.243609
Pressure to Soft Tube P8 (Pa)		344759.243609

Segment 8-7: 1/8in Soft Tubing [1/4in Hard Tubing to Valve] (1 Muscle)

Tube ID (in)	0.125 (1/8) inch
Tube ID (cm)	0.3175
Tube ID (m)	0.003175
Tube Length (in)	3
Tube Length (cm)	7.62
Tube Length (m)	0.0762

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.3175	0.003175	0.00000792	0.9144	2888.73520	Transitional	0.000944882	0.04553233	455.932441	0.455932441	0.06612508

Resulting Pressure Drop (PSI, Pa):	0.06612508	455.932441
Pressure to Valve P7 (Pa)		345215.176051

Segment 7-6: Valve

losses assumed to be minor

Pressure to Soft Tubing P6 (Pa)	345215.176051

Segment 6-5: 1/8 in Soft Tubing [Valve to 1/4in Hard Tubing] (1 Muscle)

Tube ID (in)	0.125 (1/8) inch
Tube ID (cm)	0.3175
Tube ID (m)	0.003175
Tube Length (in)	2
Tube Length (cm)	5.08
Tube Length (m)	0.0508

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.3175	0.003175	0.00000792	0.9144	2888.73520	Transitional	0.000944882	0.04553233	303.954961	0.303954961	0.04408339

Resulting Pressure Drop (PSI, Pa):	0.04408339	303.954961
Pressure to Hard Tubing P5 (Pa)	345519.131011	

Segment 5-4: 1/4in Hard Tubing [Soft Tubing to Y] (1 Muscle)

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
Tube Length (in)	1
Tube Length (cm)	2.54
Tube Length (m)	0.0254

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.2286	1444.36760	Laminar	0.000472441	0.04431005	4.621805	0.004621805	0.00067031

Resulting Pressure Drop (PSI, Pa):	0.00067031	4.621805
Pressure to Hard Tubing P4 (Pa)	345523.752816	

Segment 4: Y Connector, 1/4 to 1/4

losses assumed to be minor

Volumetric flow into Muscle (m3/s)	0.0000072393392
# Conenctor Branches	2
Total Volumetric Flow into Y (m3/s)	0.00001447867847
Pressure to Hard Tubing P4 (Pa)	345523.752816

Segment 4-3: 1/4in Hard Tubing [Y to Y] (1 Muscle)

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
Tube Length (in)	1.5
Tube Length (cm)	3.81
Tube Length (m)	0.0381

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.4572	2888.73520	Transitional	0.000472441	0.04519407	28.284081	0.028284081	0.00410211

Resulting Pressure Drop (PSI, Pa):	0.00410211	28.284081

Pressure to Hard Tubing P3 (Pa)	345552.036897

Segment 3: Y Connector, 1/4 to 1/4

lossses assumed to be minor

Volumetric flow out of single branch Y (m3/s)	0.000014478677847
# Conenctor Branches	2
Total Volumetric Flow into Y (m3/s)	0.00002895735694
Pressure to Hard Tubing P3 (Pa)	345552.036897

Segment 3-2: 1/4in Hard Tubing (Y to Y) (2 muscles)

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
Tube Length (in)	12
Tube Length (cm)	30.48
Tube Length (m)	0.3048

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.9144	5777.47041	Transitional	0.000472441	0.03655035	731.984871	0.731984871	0.10616169

Resulting Pressure Drop (PSI, Pa):	0.10616169	731.984871

Segment 2: Y Connector, 1/4 to Pump Tubing

Pressure to Pump Tubing P2 (Pa)	346284.021768
Total Volumetric Flow into Y (m3/s)	0.00005791471388
# Conenctor Branches	2
Single Volumetric out of single Y branch (m3/s)	0.00002895735694

lossses assumed to be minor

Segment 2-1: Pump Tubing [Hard Tubing to Accumulator]

Pressure to Pump Tubing P2 (Pa)	346284.021768
Total Volumetric Flow into Y (m3/s)	0.00005791471388
# Conenctor Branches	2
Tube ID (in)	0.5
Tube ID (cm)	1.27
Tube ID (m)	0.0127
Tube Length (in)	24
Tube Length (cm)	60.96
Tube Length (m)	0.6096

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
1.27	0.0127	0.00012667	0.4572	5777.47041	Transitional	0.000236220	0.03632421	181.864008	0.181864008	0.02637622

Resulting Pressure Drop (PSI, Pa):	0.02637622	181.864008

Resulting Pump Requirements

Pressure from Pump P1 (Pa)	346465.885776
Pressure from Pump (PSI)	**50.24885943**
Total Volumetric Flow (m3/s)	0.00005791471388
Volumetric Flow, Fill 1/2 of Muscles (m3/s)	0.00002895735694

Max Volumetric Flow from Pump (L/min)	1.737441416	
Universal Constants		
density of water (kg/m3):	998	(at 20C)
viscosity of water (Ns/m2):	0.001003	(at 20C)
pipe roughness (cm):	0.0003	

Hydraulic System Pressure Loss Calculations - 10 Inch Muscles

gray = user input

Hydro Muscles

Inner Diameter (in)	0.375
Outer Diameter (in)	0.5
Change in Length (in)	5
Volume Change, 1 Muscle (in3)	0.5522167969
Speed to Fill One Muscle (s)	1
Volumetric Flow Rate (in3/s):	0.5522167969 (one muscle)
Volumetric Flow Rate (m3/s):	0.000009049174 0 (one muscle)
Total # Muscles	8
Muscle Operating Pressure P9 (PSI)	50
Muscle Operating Pressure P9 (Pa)	344750

Segment 9-8: 1/4in Hard Tubing (1 Muscle) [Muscle to Soft Tubing]

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
velocity V9 (m/s)	0.2857488017
Tube Length (in)	2
Tube Length (cm)	5.08
Tube Length (m)	0.0508

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (kPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.2857	1805.45950	Laminar	0.000472441	0.03544804	11.554512	0.011554512	0.0016757578

Resulting Pressure Drop (PSI; Pa): 0.0016757578   11.554512

Pressure to Soft Tube P8 (Pa): 344761.554512

Segment 8-7: 1/8in Soft Tubing [1/4in Hard Tubing to Valve] (1 Muscle)

Tube ID (in)	0.125 (1/8) inch
Tube ID (cm)	0.3175
Tube ID (m)	0.003175
Tube Length (in)	3
Tube Length (cm)	7.62
Tube Length (m)	0.0762

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (kPa)	Change in Pressure (PSI)
0.3175	0.003175	0.00000792	1.1430	3610.91901	Transitional	0.000944882	0.04246304	664.372506	0.664372506	0.09635569

Resulting Pressure Drop (PSI; Pa): 0.09635569   664.372506

Pressure to Valve P7 (Pa): 345425.927018

Segment 7-6: Valve

losses assumed to be minor

Pressure to Soft Tubing P6 (Pa): 345425.927018

Segment 6-5: 1/8 in Soft Tubing [Valve to 1/4in Hard Tubing] (1 Muscle)

Calculate Pressure Drop:

Tube ID (in)	0.125 (1/8) inch
Tube ID (cm)	0.3175
Tube ID (m)	0.003175
Tube Length (in)	2
Tube Length (cm)	5.08
Tube Length (m)	0.0508

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.3175	0.003175	0.00000792	1.1430	3610.91901	Transitional	0.000944882	0.04246304	442.915004	0.442915004	0.06423713

Resulting Pressure Drop (PSI, Pa): 0.06423713 442.915004

Pressure to Hard Tubing P5 (Pa): 345868.842022

Segment 5-4: 1/4in Hard Tubing [Soft Tubing to Y] (1 Muscle)

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
Tube Length (in)	1
Tube Length (cm)	2.54
Tube Length (m)	0.0254

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.2857	1805.45950	Laminar	0.000472441	0.03544804	5.777256	0.005777256	0.00083789

Resulting Pressure Drop (PSI, Pa): 0.00083789 5.777256

Pressure to Hard Tubing P4 (Pa): 345874.619278

Segment 4: Y Connector, 1/4 to 1/4

losses assumed to be minor

Volumetric flow into Muscle (m3/s)	0.0000090491740
# Conenctor Branches	2
Total Volumetric Flow into Y (m3/s)	0.0000180983480809
Pressure to Hard Tubing P4 (Pa)	345874.619278

Segment 4-3: 1/4in Hard Tubing [Y to Y] (1 Muscle)

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
Tube Length (in)	1.5
Tube Length (cm)	3.81
Tube Length (m)	0.0381

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	0.5715	3610.91901	Transitional	0.000472441	0.04208553	41.154126	0.041154126	0.00596869

Resulting Pressure Drop (PSI, Pa): 0.00596869 41.154126

Pressure to Hard Tubing P3 (Pa): 345915.773404

Segment 3: Y Connector, 1/4 to 1/4

lossses assumed to be minor

Volumetric flow out of single branch Y (m3/s)	0.0000180985834809
# Conenctor Branches	2
Total Volumetric Flow into Y (m3/s)	0.0000361966969618
Pressure to Hard Tubing P3 (Pa)	345915.773404

Segment 3-2: 1/4in Hard Tubing (Y to Y) (2 muscles)

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
0.635	0.00635	0.00003167	1.1430	7221.83801	Transitional	0.000472441	0.03432390	1074.056733	1.074056733	0.15577328

Tube ID (in)	0.25 (1/4) inch
Tube ID (cm)	0.635
Tube ID (m)	0.00635
Tube Length (in)	12
Tube Length (cm)	30.48
Tube Length (m)	0.3048

Resulting Pressure Drop (PSI, Pa): 0.15577328 | 1074.056733

Segment 2: Y Connector, 1/4 to Pump Tubing

lossses assumed to be minor

Single Volumetric Flow out of single Y branch (m3/s)	0.0000361966969618
# Conenctor Branches	2
Total Volumetric Flow into Y (m3/s)	0.0000723939339235
Pressure to Pump Tubing P2 (Pa)	346989.830137

Resulting Pressure Drop: 346989.830137

Segment 2-1: Pump Tubing [Hard Tubing to Accumulator]

Calculate Pressure Drop:

Tube ID (cm)	Tube ID (m)	Pipe Flow Area (m2)	Velocity (m/s)	Re	Flow Type	E/D	Friction Factor	Change in Pressure (Pa)	Change in Pressure (KPa)	Change in Pressure (PSI)
1.27	0.0127	0.00012667	0.5715	7221.83801	Transitional	0.000236220	0.03406870	266.517785	0.266517785	0.03865378

Tube ID (in)	0.5
Tube ID (cm)	1.27
Tube ID (m)	0.0127
Tube Length (in)	24
Tube Length (cm)	60.96
Tube Length (m)	0.6096

Resulting Pressure Drop (PSI, Pa): 0.03865378 | 266.517785

Resulting Pump Requirements

Pressure from Pump P1 (Pa)	347256.347922
Pressure from Pump (PSI)	**50.36350224**
Total Volumetric Flow (m3/s)	0.0000723939339235
Volumetric Flow, Fill 1/2 of Muscles (m3/s)	0.0000361966969618

Appendices

Max Volumetric Flow from Pump (L/min)	2.171801771	
Universal Constants		
density of water (kg/m3):	998	(at 20C)
viscosity of water (Ns/m2):	0.001003	(at 20C)
pipe roughness (cm):	0.0003	

Appendix E: Serial Monitor CLI

Command	Purpose	Args
listvert	Responds with a list of connected vertebrae boxes	none
pingvert	Responds with a boolean representing if the address given is connected	
setvert pidconf address muscle kp ki kd	Updates the muscles at the address to use the given PID config, but does not save these changes after power off.	address: any address 0-128 muscle: x/y kp,ki,kd: numbers (1.2, 0, 0)
setvert mpos address deltaXLen deltaYLen	Updates the muscles to change their length from their current position.	address: any address 0-128 deltaXLen: change in length from current position of section muscle x. deltaYLen: change in length from current position of section muscle y.
setvert pose address deltaDeflection deltaRotation	Currently has no effect, will change the setpoint for the section in terms of deflection and rotation	address: any address 0-128 deltaXLen: change in deflection from current. deltaYLen: change in rotation from current.
Getvert mpos address	Get current muscle lengths at the given address	
Graphmpos address deltaXLen deltaYLen	In the serial plotter (opened with cmd+shift+L), this command will function the same at setvert mpos but will output muscle length of x and y and plot them for 5 seconds.	address: any address 0-128 deltaXLen: change in length from current position of section muscle x. deltaYLen: change in length from current position of section muscle y.
twosection		address1: any address 0-128 deltaXLen1: change in length from current position of section muscle x1. deltaYLen1: change in length from current position of section muscle y1.

		address2: any address 0-128 deltaXLen2: change in length from current position of section muscle x2. deltaYLen2: change in length from current position of section muscle y2.
threesection		address1: any address 0-128 deltaXLen1: change in length from current position of section muscle x1. deltaYLen1: change in length from current position of section muscle y1. address2: any address 0-128 deltaXLen2: change in length from current position of section muscle x2. deltaYLen2: change in length from current position of section muscle y2. address3: any address 0-128 deltaXLen3: change in length from current position of section muscle x3. deltaYLen3: change in length from current position of section muscle y3.

Appendix F: MTypes and SetNextGetTypes

unsigned char	MType	Expected Behavior	Implemented
0	notSet	No return data is set.	yes
1	setRequestMtype	This mType means the message only contains a header request for data in its setNextGet field and contains no data past the header	yes
3	commandPoseDelta	This mType means the message contains 6 floats of data corresponding to an amount to change the XYZRPY pose of the vertebral section. The data is formatted [float X, float Y, float Z, float R, float P, float Y]	yes
5	commandMuscleXPIDVals	This mType means the message contains 3 floats of data corresponding to new PID settings for the X muscle of a vertebral section. The data is formatted [float kp, float ki, float kd, 0, 0, 0]	yes
7	commandMuscleXYDeltaLen	This mType means the message contains 3 floats of data corresponding to new a change in length of the muscles X and Y. If the muscle is requested a value outside of its range it should just go to the max. The data is formatted [float deltaXMLength, float deltaYMLength, 0, 0, 0, 0]	yes
9	commandMuscleYPIDVals	This mType means the message contains 3 floats of data corresponding to new PID settings for the Y muscle of a vertebral section. The data is formatted [float kp, float ki, float kd, 0, 0, 0]	yes

unsigned char	SetNextGet Type	Expected Behavior	Implemented
0	notSet	returns {-1, -1, -1, -1, -1, -1}. Warning flag is set to 2.	yes
2	getCurrentPose	This setNextGet field is used to prime the vertebral section to respond to a request for data with XYZ RPY pose data. The data should be formatted as [float X, float Y, float Z, float R, float P, float Y]	yes
4	getCurrentPoseDelta	This setNextGet field is used to prime the vertebral section to respond to a request for data with the currently being acted upon change in position. The data should be formatted as	yes

		[float dX, float dY, float dZ, float dR, float dP, float dY]	
6	getMuscleXPIDConfig	This setNextGet field is used to prime the vertebral section to respond to a request for data with the configuration of the muscle X PID gains. The data should be formatted as [float kp, float ki, float kd, 0, 0, 0]	yes
8	getMuscleXYLen	This setNextGet field is used to prime the vertebral section to respond to a request for data with the current lengths of the muscles X and Y in a Vertebral section. The data should be formatted as [float muscleXLength, float muscleYLength, 0, 0, 0, 0]	yes
10	getMuscleYPIDConfig	This setNextGet field is used to prime the vertebral section to respond to a request for data with the configuration of the muscle X PID gains. The data should be formatted as [float kp, float ki, float kd, 0, 0, 0]	yes
14	getVertConfig	This setNextGet field is used to prime the vertebral section to respond to requests for data with configuration data for the section. The data should be formatted as [float NumberOfVert, float LengthOfProngs, float HeightOfVertebrae, 0 , 0 , 0]	yes

warningFlag values

unsigned char	Meaning
0	No error
1	Illegal Mtype
2	Unset SNG during a request
3	Unknown SNG during a request