Design of an Underactuated Lower Body Exoskeleton Using a Pantograph

Evan Alexander Claessen

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

Alan T. Asbeck, Chair
Steve C. Southward
Alexander Leonessa

02/14/2022
Blacksburg, Virginia

Keywords: underactuated exoskeleton, lower body exoskeleton, gait

Design of an Underactuated Lower Body Exoskeleton Using a Pantograph

Evan Alexander Claessen

(ABSTRACT)

This paper presents the design of an underactuated lower body exoskeleton to assist with walking. It reduces the amount of bodyweight going through the user's leg by providing a supporting force to the user that is engaged and disengaged depending on the stage of the gait cycle the user is in. It is engaged when the leg is in stance, effectively pushing between the ball of the foot and the hips, and is disengaged during leg swing. This support force is provided by a linear actuator on each leg that consists of a compression spring, ball screw, and motor. It works by having the motor turn the ball screw, which moves a metal plate to either compress or decompress the spring. The actuator is designed to always be able to extend, to avoid limiting the user's motion. The spring is disengaged while the leg is in swing in order to reduce any impedance to the user's natural stride. The exoskeleton is also designed to minimize any range of motion limitations to reduce its restrictiveness. The exoskeleton was found to be able to provide 19 lbs (85 N) of support to the user per leg.

Design of an Underactuated Lower Body Exoskeleton Using a Pantograph

Evan Alexander Claessen

(GENERAL AUDIENCE ABSTRACT)

Exoskeletons are external devices worn to assist the user's natural movement or strength. This paper outlines the design of an exoskeleton that assists the user in walking by providing a supporting force on any leg that the user's weight is on. This effectively reduces the load on the user's legs, which could help reduce leg strain and fatigue. The exoskeleton releases this force when weight is removed from the leg to allow the user to easily swing their leg forward to step. The exoskeleton was designed to minimize limitations to the range of motion of the leg joints while walking, squatting, or sitting to ensure that the exoskeleton did not feel restricting or uncomfortable. Testing revealed that the exoskeleton was able to provide a supporting force of approximately 19 lbs (85 N) to the user per leg and met all the joint range of motion requirements to avoid restrictiveness.

Acknowledgments

First and foremost, I would like to thank my committee chairman, Dr. Alan Asbeck. I joined his lab over five years ago during my first semester of college and working in his lab has been vital to developing my practical engineering knowledge. His enthusiasm for assistive robotics has made working in his lab some of the most enjoyable and intellectually stimulating work I have ever done.

Thank you to my project partner, Zackory Biggers. His engineering skills and amiability have immeasurably helped me throughout the high points and low points of my graduate research.

I would like to thank Dr. Leonessa for serving on my advisory committee and for introducing me to Dr. Asbeck and his lab. Thank you to Dr. Southward for also serving on my advisory committee and for the useful practical knowledge that I learned in his graduate controls classes.

Thank you to my senior design team, whose initial work on this project was invaluable. I am glad to have been able to work with such a fun and skillful group of people. I would like to specifically thank Ryan Gottwald, whose advice, help, and friendship have aided me through my graduate program.

I would like to thank the undergraduate students, Christian, Forrest, and Madeline, who helped with the programming and manufacturing of the project.

Thank you to all my lab-mates, who have helped me throughout my undergraduate and graduate research in the lab.

Finally, I would like to thank my family and friends. They have provided me with the foundation that has supported me throughout my education. They never fail to provide me with encouragement or a relaxing break from work when I need it most.

Contents

List of Figures

List of Tables

Abbreviations

RoM	Range of Motion
CAD	Computer-Aided Design
LiPo	Lithium Polymer
I^2C	Inter-Integrated Circuit
RC	Resistor-Capacitor
DC	Direct Current
V	Voltage
A	Ampere
N	Newton
Nm	Newton-meter
lbs	Pounds
mm	Millimeter
ms	Milliseconds
Hz	Hertz

Chapter 1

Introduction

In recent years, there have been many exoskeletons designed for assistance with lower body motion. These exoskeletons can help those with lower body impairments and disabilities. Exoskeletons can help in a wide variety of ways from fully actuating the wearer's joints and balancing the body to simply providing a gravity/load compensating force while walking.

Fully actuated exoskeletons are beneficial to those with little to no remaining control over their lower body motor functions. There have been many fully actuated lower body exoskeletons. These include the EXO-H3 [1], BLEEX [2], and REX [3]. These exoskeletons have an actuation system for each joint, allowing the joints to be individually controlled. This is usually done by having a separate motor drive each joint. This provides a lot of control over the user's gait and allows for a large amount of assistance. However, this may feel restrictive to users that require less assistance overall, or assistance in only a specific joint.

Perhaps just as common as a fully actuated lower body exoskeleton are exoskeletons that support the hip and knee joints but not the ankle joint. The HAL-5 [4], LLEX [5], ReWalk [6], TWIICE ONE [7], Indego [8], and a hybrid exoskeleton made by Aguilar-Sierra et al. [9]. Most of these work with the same design principles as fully actuated exoskeletons, driving each joint with its own motor. The exoskeleton by Aguilar-Sierra et al. [9] differs slightly, actuating the joints with a combination of motors and pneumatics.

Another approach to lower body exoskeletons is to design them for a singular joint in the leg. Wiggin, Sawicki, and Collins [10] designed an ankle exoskeleton using springs to mimic the tendon system of the ankle. The springs were able to be disengaged using a clutch to allow free movement of the ankle during swing. This clutch was able to disengage the springs while being completely passive. MIT's AAFO [11] actively supports the ankle joint using a series elastic actuator. Dollar [12] designed a knee exoskeleton designed to store the user's energy from heel strike and transfer it to assist in toe lift. It achieves this by putting a spring in parallel with the user

at heel strike. The spring is then compressed at heel strike, which causes it to assist the user in extending the knee at toe lift. The RoboKnee [13] uses series elastic actuators to actively support the knee joint. The Keeogo [14] and Lockheed Martin's ONYX [15] are more traditional knee exoskeletons, both having a motor that directly actuates the knee to assist the user. Giovacchini et al. [16] developed a hip exoskeleton that assists the user in hip flexion and extension using series elastic actuators.

All the exoskeletons discussed thus far work by individually actuating the user's joints. Another approach is actuator designs that use a single actuator to assist the entire leg or multiple joints at once. This could be more beneficial to people with more minor lower body impairments or disabilities. For example, people who have joint weaknesses due to age or injury might have trouble walking long distances without significant pain or fatigue. A joint-based exoskeleton would help them reduce their fatigue but might be undesired due to its bulk, power consumption, and restrictiveness. For them, an underactuated exoskeleton that uses only one actuator to assist multiple joints may be more beneficial. This is because the reduction in actuators could potentially reduce bulk, weight, power consumption, complexity, and obtrusiveness. Despite this, these designs are much less explored than joint-based exoskeleton ones.

An example of an exoskeleton that does not follow a joint-based actuation is the BAExo [17]. It can be seen in Figure 1.1. It is a soft exoskeleton that uses serial elastic actuators to control the leg in a fashion similar to how muscles actuate joints. The stiffness of the artificial muscles is controlled in the stance phase of gait, while the motors are locked in swing, resulting in a passive spring system.

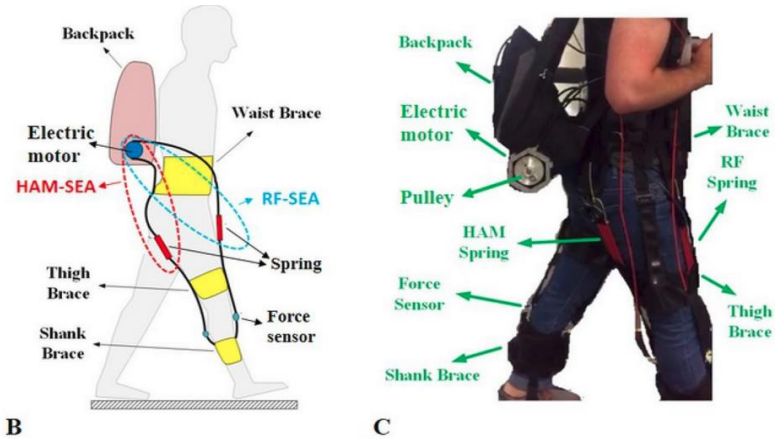

Figure 1.1: Soft underactuated exoskeleton by Firouzi et al. [17].

Another example of an underactuated exoskeleton that does not actuate individual joints is the PALExo [18]. For the PALExo, Wang et al. worked on improving the load capacity and comfortableness of lower body load-assistance exoskeletons. To do this, they used an underactuated design with two gas springs in parallel on each leg. This parallel structure allowed the exoskeleton to have a higher load capacity than a single gas spring design. The two gas springs were active and designed to be able to balance the torque imparted on the wearer's waist. The PALExo can be seen in Figure 1.2. While the underactuated design leads to a more comfortable and simple design, it does not follow the natural geometry of the leg, thus making certain motions more difficult.

Figure 1.2: The PALExo [18].

This paper outlines the design, implementation, and testing of a bodyweight supporting exoskeleton with only a single actuator per leg. This exoskeleton uses a spring to provide a support force during leg stance and can release the spring so the leg can freely swing. This should result in less restriction during leg swing allowing for a more natural gait than fully actuated designs. Our exoskeleton utilizes a pantograph design which causes the force to effectively push between the ball of the foot and the waist without needing an actuator directly between them. This allows for use of a lighter, more compact actuator. It also lets our design follow the geometry of the body, allowing for movements such as sitting and squatting.

Chapter 2

Overview/Motivation

2.1 Motivation

Joint weaknesses and injuries are extremely prevalent, especially among older individuals. Knee osteoarthritis affects “roughly 13% of women and 10% of men 60 years and older” [19]. Similarly, 12% of individuals over 50 have midfoot osteoarthritis [20]. For people with leg weaknesses and injuries such as these, a walking assisting exoskeleton could help reduce joint pain and fatigue and prevent further injury. Many leg exoskeletons to assist with walking individually actuate the user's limbs. While this is helpful to those with low mobility in the joints, it can feel restricting to users with full leg mobility. The exoskeleton outlined in this paper is designed to provide the user with a support force that reduces the effective weight supported by the user. This support can be engaged or disengaged depending on the stage of gait.

2.2 Biomechanics of the Leg

The stages of gait can be seen in Figure 2.1. In this paper, the initial contact stage of gait will also be referred to as heel strike, pre-swing as heel-lift, and initial swing as foot lift.

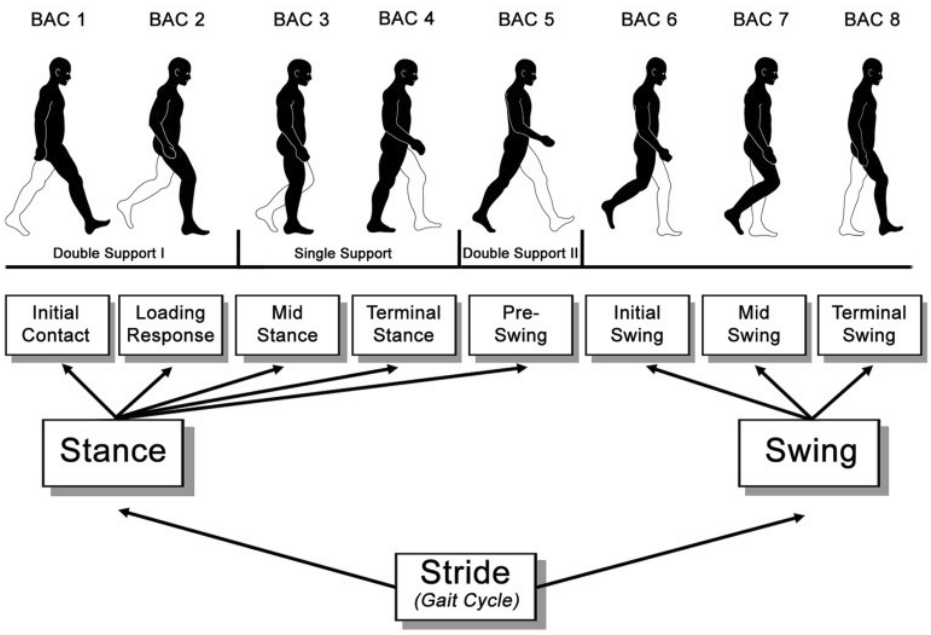

Figure 2.1: The stages of the gait cycle [21]

During the stance stage of gait, the exoskeleton will provide assistance to the user, which is removed at foot lift to allow for an unimpeded swing. The assistance during stance will not be directly actuating a specific joint and will instead be in the form of a support force between the user's hip and feet, reducing the amount of bodyweight necessary for the user to support.

Our exoskeleton design should minimize its restriction of the user's natural range of motion. The degrees of freedom in the hip and knee can be seen in Figure 2.2, and the relevant foot degrees of freedom can be seen in Figure 2.3. We would like the exoskeleton to not impede any of the degrees of freedom of the leg, including hip extension/flexion, hip abduction/adduction, hip internal/external rotation, knee extension/flexion, ankle dorsiflexion/plantarflexion, and ankle eversion/inversion. For hip extension/flexion we define 0° to be the leg straight down with flexion defined as positive motion and extension as negative. Similarly, for hip abduction/adduction, we define 0^o to be the leg straight down, with abduction being positive and adduction being negative. For the knee, we define the straight knee position to be 180°, with flexion being negative and extension positive. For hip rotation, a neutral leg is defined as 0° with external rotation being positive and internal rotation being negative. For the ankle, we define 90° to be the neutral position with the foot perpendicular to the lower leg, with plantarflexion being positive and dorsiflexion

being negative. For eversion and inversion of the ankle, the neutral position of the foot to your lower leg is defined as 0°, with inversion being negative and eversion being positive.

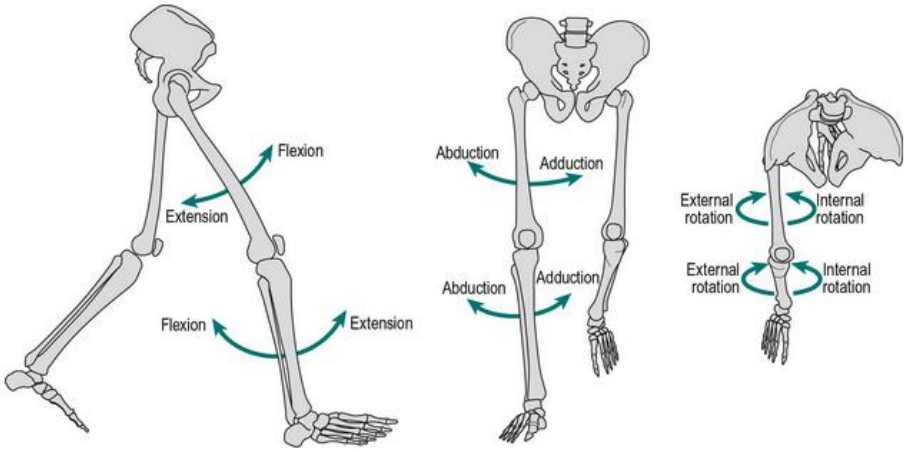

Figure 2.2: Rotational degrees of freedom in the hip and knee joints [22].

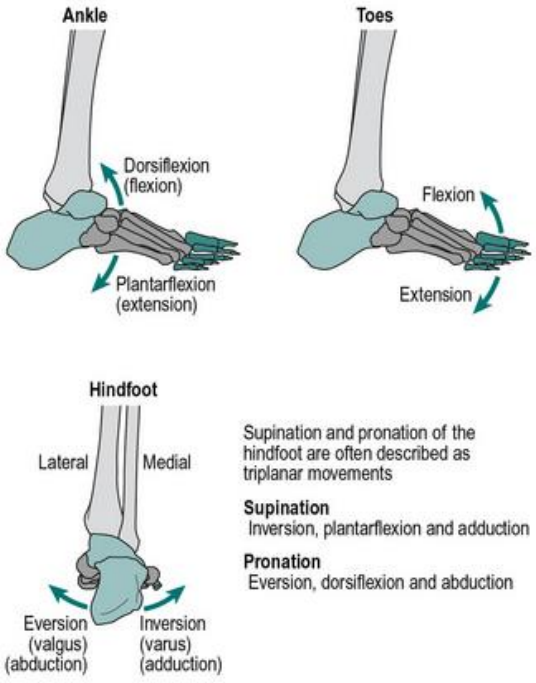

Figure 2.3: Degrees of freedom in the ankle and hindfoot [22].

To ensure our exoskeleton is comfortable and unobtrusive, several minimal ranges of motion (RoM) were required to be preserved. Table 2.1 shows the mean RoM of the right leg, converted to the angle definitions stated above, found by Roaas and Anderson [23] during a study on 30 to 40-year-old males. It also shows the RoM based on one standard deviation above the mean in all directions, to give a better idea of the ranges one might expect.

Table 2.1: Range of motion (degrees) of joints found by Roaas and Anderson [23]

Motion	Mean Range	Range 1 STD Greater
Hip Extension/Flexion	(-9.4) - 120.3	(-14.7) - 128.6
Hip Adduction/Abduction	(-30.5) - 38.8	(-37.8) - 45.8
Hip Internal/External Rotation	(-32.6) - 33.6	(-40.8) - 40.4
Knee Flexion/Extension	36.2 - 178.4	29.8 - 181.2
Ankle Dorsi/Plantarflexion	74.7 - 129.7	68.9 - 137.2
Ankle Inversion/Eversion	(-27.7) - (27.6)	(-34.6) - 32.2

2.3 Requirements

Based on the range of motion data, we decided on RoM requirements for the exoskeleton to not limit the user's natural RoM in normal activities such as walking, sitting, or squatting. We also define a minimum support force expected to significantly assist the user while using the exoskeleton. The requirements can be seen in Table 2.2.

Table 2.2: List of Requirements

Requirement	Target Minimum	Unit
Hip Extension/Flexion Range of Motion	(-15) - 120	Degrees
Hip Abduction/Adduction Range of Motion	(-10) - 10	Degrees
Hip Internal/External Rotation	(-15) - 15	Degrees
Knee Extension/Flexion Range of Motion	60 - 180	Degrees
Dorsi/Plantarflexion Range of Motion	80 - 130	Degrees
Ankle Eversion/Inversion Range of Motion	(-30) - 30	Degrees
Support Force	133.4 (30)	N (lbf)

Chapter 3

Exoskeleton Design

3.1 Design Overview

The final design of our exoskeleton consists of four main component sets: the frames, actuators, soft goods, electronics. The frame is the structure of the exoskeleton and is designed to mimic the shape of the human leg. The actuator is a spring-based linear actuator powered by a motor. The soft goods consist of a waist belt, thigh strap, calf strap, and overshoes. For electronics, our project uses batteries, circuit boards, sensors, and common electrical components. The overall exoskeleton can be seen in Figure 3.1. The entire exoskeleton has a mass of 14.3 kg (31.5 lbs).

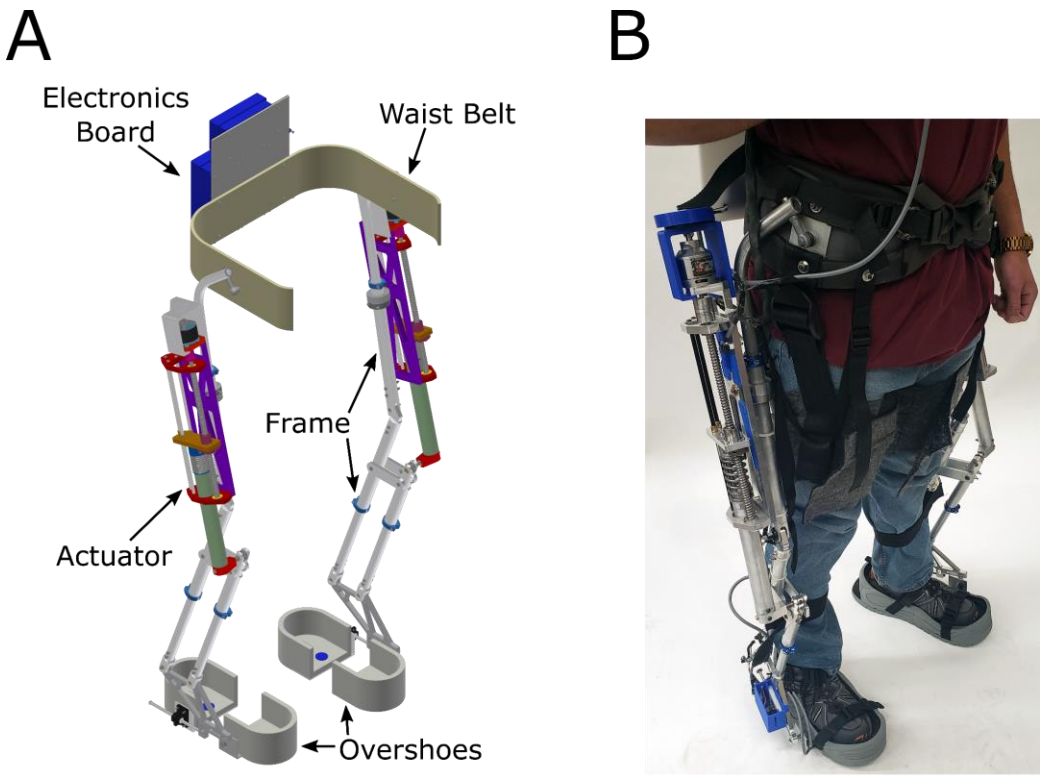

Figure 3.1: (A) The CAD model overall exoskeleton. (B) The final built exoskeleton.

3.2 Frame Design

3.2.1 Design of the Frame Structure

Our exoskeleton was designed to closely match the natural structure of the leg. The frame of the exoskeleton can be seen in Figure 3.2 below. The design has small tubes, seen in green in Figure 3.2, that sit inside larger-diameter tubes, seen in purple. The outer tubes are 1-inch in diameter, with a ¾-inch inner diameter. The inner tubes have a ¾-inch outer diameter and a ½-inch inner diameter. This allows the tubes to telescope, adding adjustability for different users' upper and lower leg lengths. After adjustment, the tubes can be locked in place by tightening clamps, seen in blue. We used the Promax FC-1 25.4mm fixed seat clamps. The mounts for the actuator are all between two adjustment points, within the outer tube area of the exoskeleton. This means that the mounting points for the actuator stay in the same positions relative to each other, independent of any adjustments of the upper or lower leg lengths. This keeps the relevant dimensions for the actuation of the exoskeleton constant, simplifying the control algorithm, as it does not have to factor in the adjustments made by the user when calculating the actuator length.

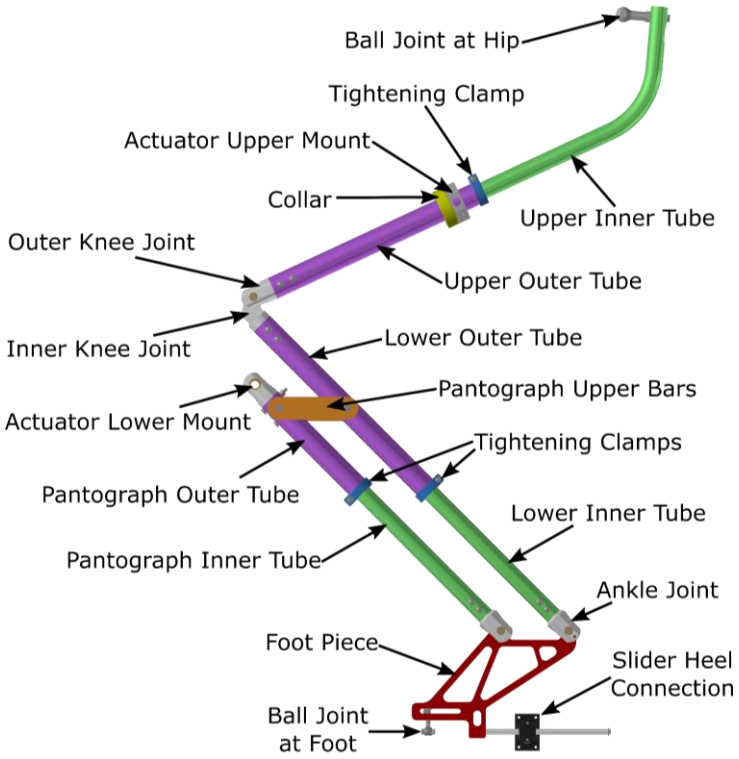

Figure 3.2: CAD model of the frame design.

One notable difference between the design and the actual structure of the human leg is the curved top piece of the frame. The exoskeleton attaches to the waist at the iliac crest, which is above the actual location of the hip joint. Because of this, a straight upper frame design does not perfectly follow the leg during motions such as squatting and sitting. The curve bar makes the exoskeleton follow the centerline of the upper leg more closely, making the knee angle better match that of the leg. This can be seen in Figure 3.3. This curve design also makes it so that the support force is applied to the waist belt by the frame pulling up on the ball joint, instead of pushing up.

Figure 3.3: (A) Straight bar design while standing. (B) Straight bar design while squatting. (C) Straight bar design while standing. (D) Straight bar design while squatting.

The pantograph is designed to allow for the actuator to push between the hip and ball of the foot without having to directly connect them together. It is preferred to have the point at which the actuator effectively pushes at the ball of the foot instead of the heel, as it allows the user to lift their heel at the end of stance while the exoskeleton still provides support. If the support was provided at the heel, we would have to remove support when the user attempts to lift their heel, which would result in no support when the user is pushing themselves forward right before lifting their foot. If we did not remove support at heel-lift, the user would need to fight the support force to lift their heel. This can be seen in Figure 3.4.

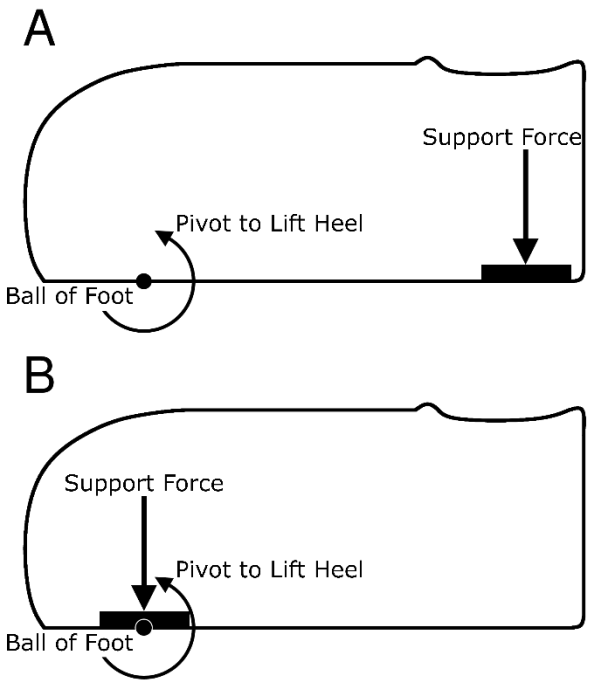

Figure 3.4: (A) Support provided at the heel of the force will work against the user pivoting their foot to lift their heel. (B) When the support is at the ball of the foot it goes through the pivot point of the foot lift, thus not interfering with heel-lift.

One option could have been to create an actuator that directly connected the hip to the ball of the foot, but that would have to be very large, and thus add a lot of weight to the exoskeleton. It would also protrude out behind the knee when squatting or sitting. This would make sitting in a chair difficult if not impossible. Another option would be to modify the frame to connect to the foot piece at the ball of the foot instead of at the heel, and then have an actuator between the two bars. This, however, would mean the frame would not follow the natural shape of your legs, as its ankle joint would offset from the actual ankle joint. This is made worse by the fact that the frame needs to allow the user to point their toes, and thus would be longer than the actual length of the legs. This would result in large protrusion from the legs as seen in Figure 3.5 below. To prevent these issues, we went with a pantograph design lower leg design.

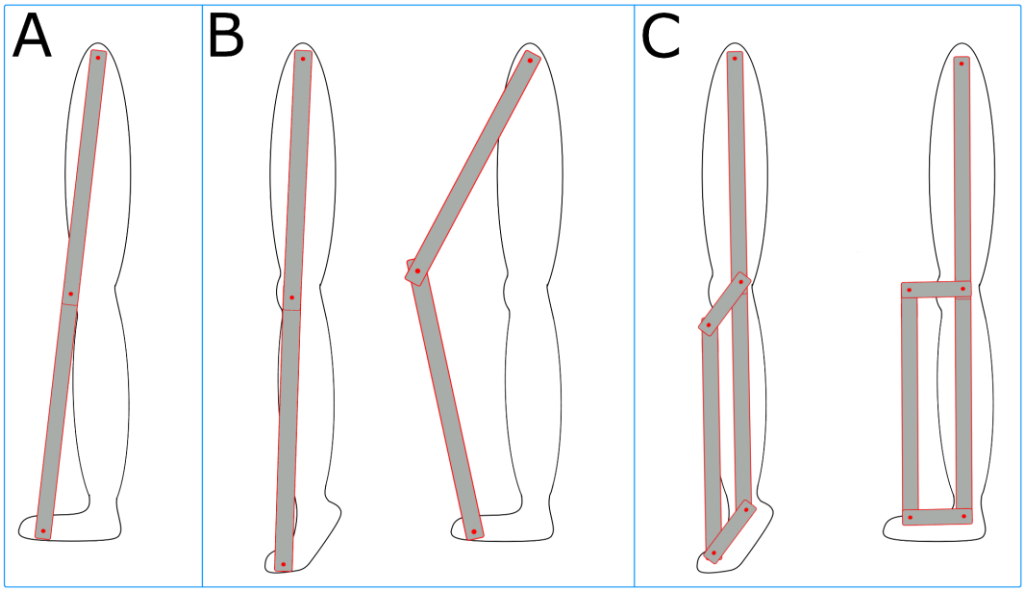

Figure 3.5: (A) A single joint design that connects directly from the hip to the ball of the foot at a normal standing position. This protrudes in front of the calf while standing and does not allow the foot to extend as the maximum length of the frame occurs when the foot is in a neutral position. (B) A single joint design that links the ball of the foot and hip with the maximum length of the frame occurring when the foot is extended. This, however, results in the exoskeleton protruding in front of the leg greatly when in a neutral standing position. (C) A pantograph design. This allows for the force to go between the hips and the ball of the foot by pushing on the outer pantograph bar. It also allows for pointing of the foot without having a large offset in a neutral standing position.

With the curve and pantograph frame design, the exoskeleton is able to closely follow the body in our expected motions. Figure 3.6 shows the exoskeleton while standing, stepping, slightly squatting, and sitting.

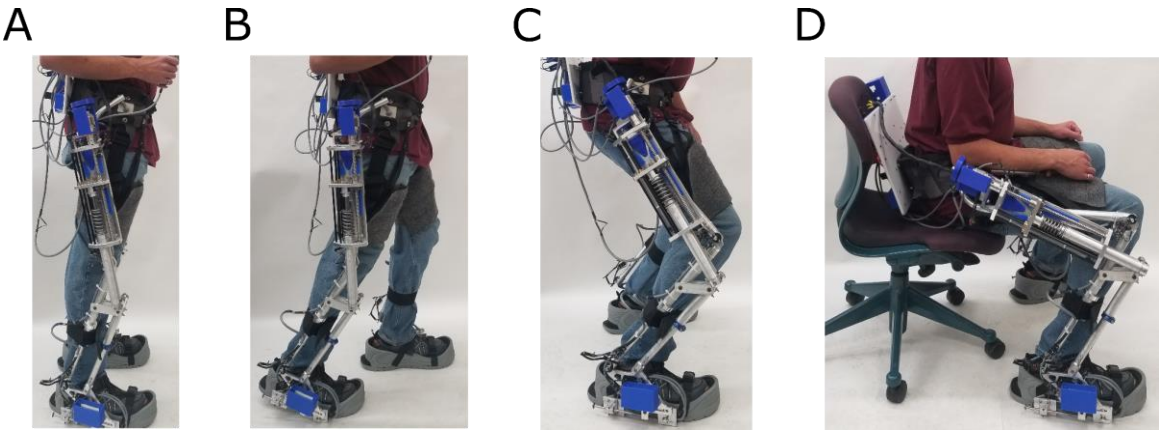

Figure 3.6:The exoskeleton while (A) standing, (B) stepping, (C) slightly squatting, and (D) sitting.

3.2.2 Joint Design

The frame design includes joints that align with the actual joints of the leg. There is a joint at the knee and the ankle and another joint in front of the ankle at the bottom of the front pantograph tubes. These joints are designed with an outer and inner joint piece. The outer piece has a groove through it that the inner piece is inserted into. This can be seen in the knee joint, shown below in Figure 3.7. For the foot joints, the inner joint is the foot piece. The outer joints have bushings in the hole for the joint pin, on the inside of the groove. This is to reduce friction in the joint and improve long-term durability. The bushings' McMaster-Carr part numbers are 7815K11. The joint pins have a ¼" shaft diameter.

Figure 3.7: The knee joint of the exoskeleton. The outer knee joint has a groove through it that the inner knee joint sits within. Bushings are inserted on the inside of the outer joint piece in the pinholes in order to reduce friction. The inner knee joint is also offset from the pinhole, meaning that the lower and upper tubes will not meet perfectly in line.

Another thing to note specifically about the knee joint is that the pinhole is offset in the radial direction from the cylindrical insert that goes into the lower leg tube. This means that the lower leg tube will be offset from the upper leg piece. The offset was added to allow for the user to reach more acute knee angles, to avoid impeding the user when sitting or squatting. This helps to meet our required knee RoM. This can be seen in Figure 3.8.

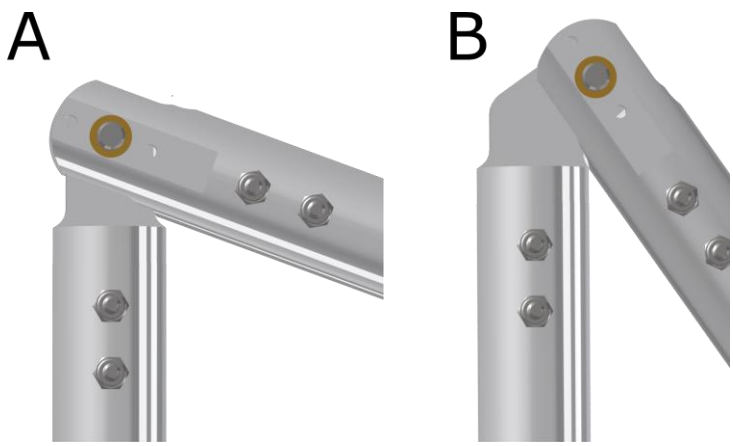

Figure 3.8: Minimum knee angle achievable with (A) a straight inner knee joint and (B) an offset inner knee joint. The straight joint was only able to reach a minimum knee angle of 70° while the offset joint was able to reach a minimum knee angle of 45°

The lower actuator mount follows the same design as the joints and attaches to the actuator via a shaft collar inside the joint's groove that is attached to the actuator's lower mount pin. The pin hole is larger than that of the knee and ankle joints, with a shaft diameter of ⅜". The McMaster-Carr part number for the lower actuator mount bushings is 7815K18 and the part number for the shaft collar is 6157K13. The upper actuator mount can be seen in Figure 3.9. The upper actuator pin is inserted into the mount and then is secured with set pins in the smaller holes. The mount is then bolted to a shaft collar that is used to mount it on the frame. The shaft collar McMaster-Carr part number is 6157K18. There are also two ball joints per leg, one at the hip and one at the foot. The one at the hip mimics the user's actual joint and should allow full RoM. The ball joint at the foot is to allow for foot eversion and inversion. There is also a steel bar connected to the back of the frame's foot piece that goes through a linear sleeve bearing. The sleeve bearing is held in a case that is mounted near the heel of the exoskeleton. This keeps the frame's foot piece orientated properly with the foot while still allowing for eversion and inversion of the foot.

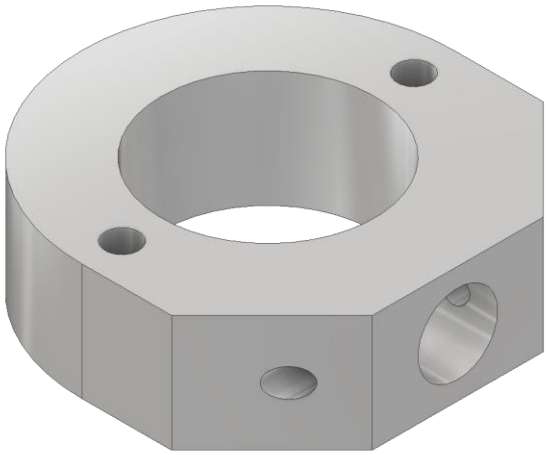

Figure 3.9: The upper actuator mount.

3.2.3 Required Actuator Force Calculations

An important property of the frame is how its geometry affects the actuation. The force from the actuator will not translate 1:1 to the support the user experiences while wearing it. Our goal is to provide a relatively constant support force to the user. So, to determine the required actuator force, we needed to find the equation for the actuator force given a desired support force. A diagram of the actuator and support forces on the frame can be seen in Figure 3.10. For the analysis, we will look at the forces when the frame is in static equilibrium. The actuator force is provided along the line between the upper and lower actuator mounting point. Likewise, the support force is provided along the line connecting the hip joint and the ball joint at the foot. The supporting force is assumed to be provided through the ball joint at the foot and not through the slider.

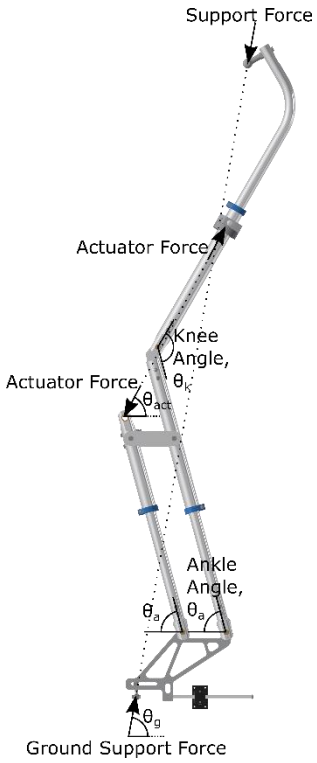

Figure 3.10: Diagram of the forces on the frame. The actuator force, F_{act}, is used to create the support force. The support force at the ground, F_g, is equal to the force at the hip, which provides support to the user. θ_a and θ_k are the ankle and knee angles respectively, representing the two inputs read by the encoders. θ_g, the angle of the ground support force, and θ_{act}, the angle of the actuator force, are found using the geometry of the leg.

To find the support force equation, we need to use the geometry of the leg to solve for the support force's angle and the actuator force's angle. The geometry used to find the angles can be seen in Figure 3.11. Using the geometry of the exoskeleton we can find the length in the x and y direction of the support line between the hip joint and the ball joint at the foot:

$$x_g = W_f + W_p - (L_p + L_k)cos(\theta_a) + W_k sin(\theta_a) + (L_u + L_c)cos(\theta_k - \theta_a) - W_c sin(\theta_k - \theta_a) \quad (3.1)$$

$$y_g = L_F + (L_p + L_k)sin(\theta_a) + W_k cos(\theta_a) + (L_u + L_c)sin(\theta_k - \theta_a) - W_c cos(\theta_k - \theta_a) \quad (3.2)$$

The ground support angle can then be found using:

$$\theta_g = tan^{-1}(\frac{y_g}{x_g}) \quad (3.3)$$

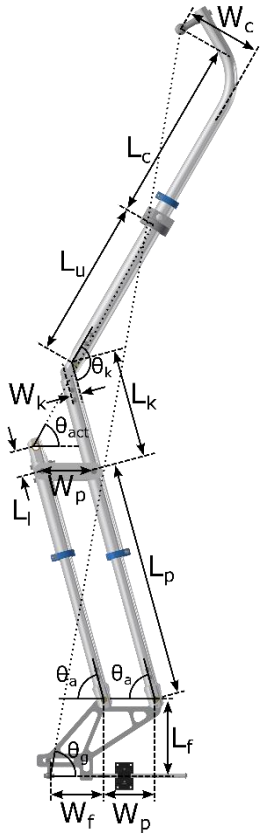

Figure 3.11: Geometry used to find the ground support force. W_f is the width of the foot piece protruding in front of the pantograph, W_p is the width of the pantograph, L_f is the height of the foot piece, L_p is the length of the pantograph bars, L_k is the length between the top of the pantograph and the knee joint pin, W_k is the width the knee joint pin is offset from the center of the lower tubes, L_u is the length between the knee joint and the upper actuator mount, L_c is the length between the hip joint and the actuator mount that is parallel to the straight part of the upper bar, and W_k is its perpendicular component.

To find the angle of the actuator force, the x and y components of the actuator length need to be found. The actuator length is the distance between the actuator's upper and lower mounting points. Figure 3.12 shows the geometry needed to find the actuator force angle. The x and y position of the upper actuator mount can be found with:

$$x_u = -L_k cos\theta_a + W_k sin\theta_a + L_u cos(\theta_k - \theta_a) \tag{3.4}$$

$$y_u = L_k sin\theta_a + W_k cos\theta + L_u sin(\theta_k - \theta_a) \tag{3.5}$$

Then the lower actuator mount position can be found using:

$$x_l = -L_l cos\theta_a - L_3 \tag{3.6}$$

$$y_l = L_l sin\theta_a \tag{3.7}$$

The lower actuator mount point's x and y position can be subtracted from the upper's x and y position to get the x and y components of the actuator length:

$$x_{act} = x_u - x_l \tag{3.8}$$

$$y_{act} = y_u - y_l \tag{3.9}$$

This gives:

$$x_{act} = (L_l - L_k)cos\theta_a + W_k sin\theta_a + L_u cos(\theta_k - \theta_a) + L_p \tag{3.10}$$

$$y_{act} = (L_k - L_l)sin\theta_a + W_k cos\theta_a + L_u sin(\theta_k - \theta_a) \tag{3.11}$$

Then the actuator force angle can then be found using:

$$\theta_{act} = tan^{-1}\left(\frac{y_{act}}{x_{act}}\right) \tag{3.12}$$

With these two angles found, we can calculate the required actuator force for a given ground support force (F_g). This calculation can be seen in Appendix A. The resulting equation is:

$$F_s = F_g \frac{(W_f + W_p)sin\theta_g - L_f cos\theta_g}{W_p sin\theta_{act}} \tag{3.13}$$

With this and the given lengths and joint angles, the required actuator force can be determined for any possible leg position.

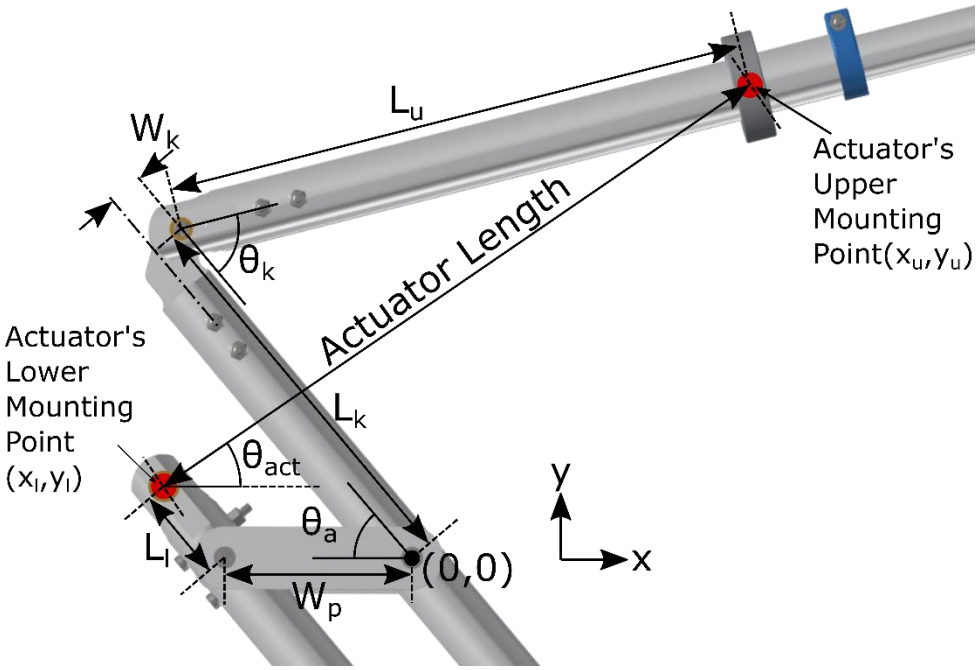

Figure 3.12: Geometry of the frame used to calculate the actuator length. The angle of the bar at the bottom has the same angle as the ankle angle because the lower leg pantograph is a parallelogram. W_p is the width of the pantograph, L_l is the length that the lower actuator mount protrudes above the pantograph, L_k is the length between the top of the pantograph and the knee joint pin, W_k is the width the knee joint pin is offset from the center of the lower tubes, and L_u is the length between the knee joint and the upper actuator mount.

To test to see what the expected actuator force to support force ratio will be, the geometry from the CAD model is used. It is important to note that the pantograph length, L_p, and the curve length, L_c, are adjustable, so lengths in the middle of our expected ranges of adjustment were used. The lengths can be seen in Table 3.1.

Table 3.1: Length values required to find the actuator length

Variable	Length (mm)
W_p	76.2
W_f	81.4
L_f	113.0
L_p	352.3
L_l	38.3
L_k	163.0
W_k	14.1
L_u	251.4
L_c	259.4
W_c	112.9

To test out the formula, the support ratio F_{act}/F_g was plotted for knee angles between 60 and 180 degrees and ankle angles between 70 and 140 degrees. These values roughly represented the expected range of motion of each joint. The plot of this ratio can be seen in Figure 3.13. As seen in the figure, for most of the range of the joint angles the plot of the support ratio is relatively flat near a value of 2, with a clear exception being the area of tangential behavior at larger ankle angles and low knee ankles. To increase the visibility of the overall shape, any point with a support ratio magnitude greater than 5 was not plotted. The tangential behavior happens due to the actuator becoming parallel to the foot piece thus no longer being able to generate a support force. An example of one of these positions can be seen in Figure 3.14. While this explains why there is tangential behavior in the first place, it does not explain why decreasing the knee angle makes the support force first head towards negative infinity instead of positive infinity. This is because the support force was found using the moment about the ankle, so at the point where the ball joint at the foot, the ankle joint, and the hip joint are perfectly aligned the support ratio is 0 as the ground support force is independent of the actuator force. Then after that point, when the support force creates a counterclockwise moment on the ankle, the equations we derived show that a negative actuator force is needed to maintain the support force. These results are probably due to our

assumptions breaking down at these knee and ankle angles. At these angles, the heel slider and/or calf strap are likely necessary to keep the system in static equilibrium. As seen in Figure 3.15, however, we are not expecting to have angle combinations that lie within these areas for our expected movements, such as walking or squatting, so the model is expected to be relatively accurate for the purpose of determining the support ratio during normal gait.

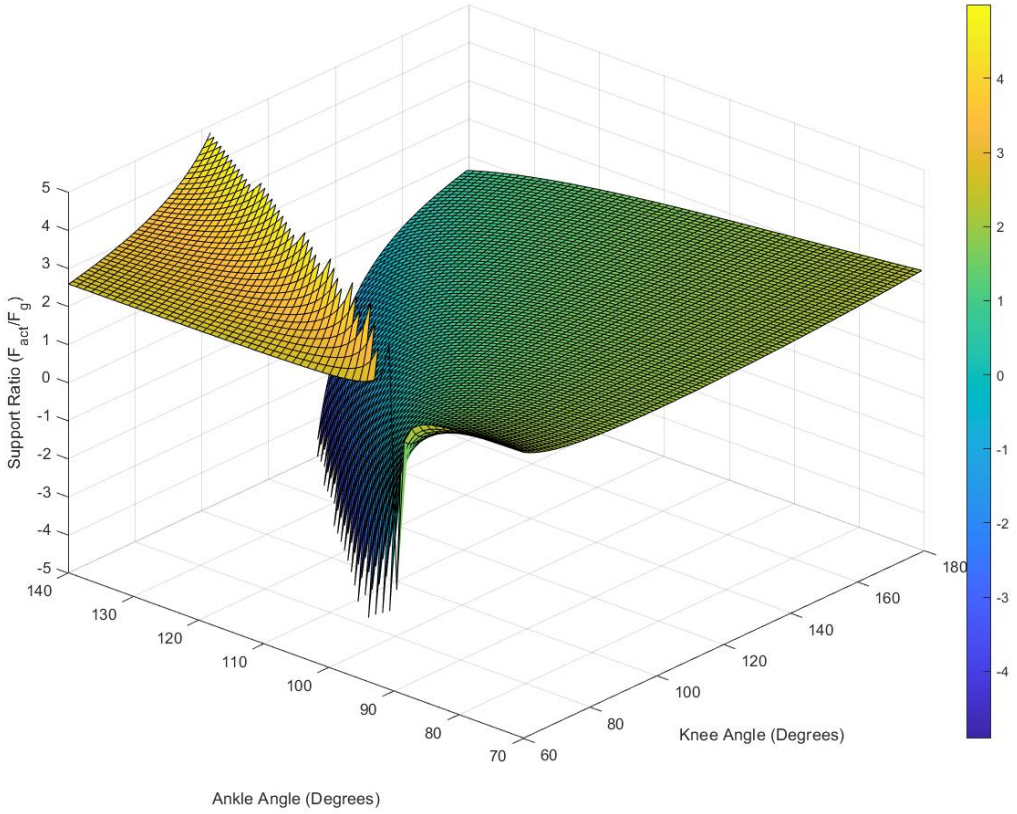

Figure 3.13: Support ratio for ankle angles between 70° and 140° and knee angles between 60° and 180°.

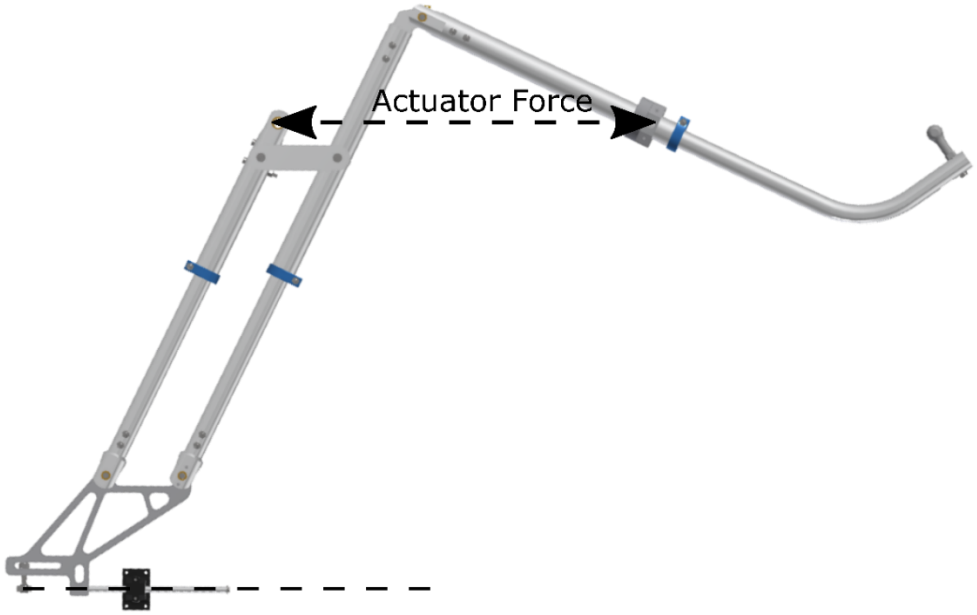

Figure 3.14: Example position where the actuator force becomes parallel to the bottom of the foot, thus making it impossible to generate a support force.

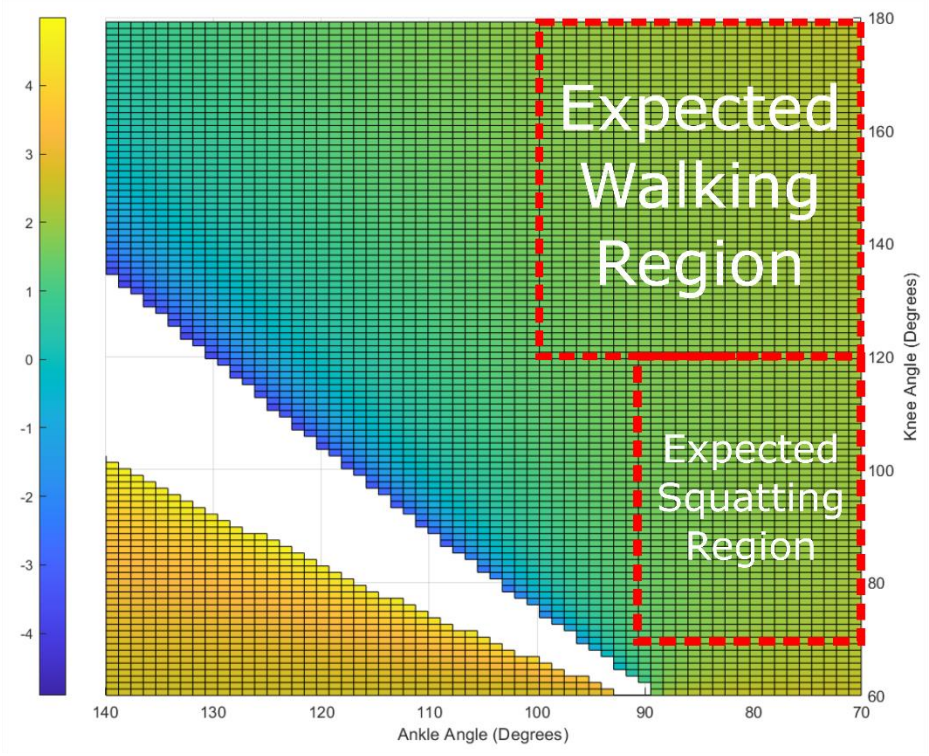

Figure 3.15: Top-down view of the support ratio plot with regions of expected movements plotted. Walking is at most expecting knee angles between 120 and 180 degrees and ankle angles between 70 and 100 degrees. Squatting is expected to have knee angles between 70 and 80 degrees and ankle angles of 70 to 90 degrees.

To further understand the expected support ratio, we looked at a few specific leg positions. A completely straight leg position, with an ankle angle of 90^o and a knee angle of 180^o, was tested along with a few other positions. This can be seen in Table 3.2.

Table 3.2: Force ratios for various angles.

Description	Ankle Angle	Knee Angle	Spring/Support Force Ratio
Standing Straight	90^o	180^o	2.05
Slight Bend	85^o	150^o	1.94
Medium Bend	75^o	120^o	1.97
Large Bend	70^o	90^o	2.02

From the table, we can see that the support ratio was about 2 pounds of actuator force for 1 pound of support. It also appears that the ratio does not deviate greatly, even for decently large bends. As we are primarily testing operation during a normal walk initially, large bends are not expected. Because of this, we decided to treat the support ratio as a constant, meaning that a constant actuator force is used during stance. The efficacy of this assumption will be investigated in the analysis section when looking at the actual leg angles during gait.

3.3 Actuator Design

Each leg has one linear actuator. This actuator is attached to the side of the frame at the lower and upper actuator mounts seen in Figure 3.1. A diagram of the actuator can be seen in Figure 3.16. The actuator is designed to provide the desired support force during stance state, and then remove all supporting force during swing. It is designed so it should only provide compressive force, never tensile force.

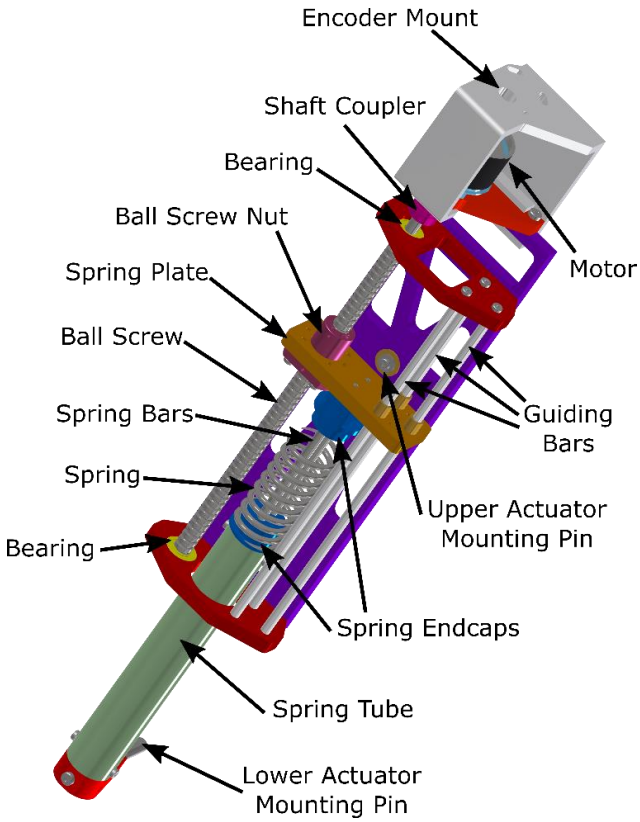

Figure 3.16: CAD model of the actuator.

3.3.1 Spring Plate

The actuator functions by using a ball screw to linearly move a plate, which will be referred to from now on as the spring plate. The ball screw used is the Befenybay Ball Screw SFU1204. It has a 12 mm diameter with a 4 mm pitch. This means the spring plate will move 4 mm for every full rotation of the motor. The spring plate has a max travel of approximately 190 mm from the top of the actuator to the bottom. The ball screw has angular contact bearings at its top and bottom. We used the 30/8ZZ angular contact double row bearings from www.vxb.com. The spring plate can be seen in orange in Figure 3.16. This plate can be moved towards or away from a compression spring, compression or decompressing it respectively. The springs (McMaster-Carr product number # 9657K551) have a spring constant of 84 lbf/in, approximately 14.71 N/mm. This means the desired force can be converted to a required spring compression using

$$Compression = \frac{F_{des}}{14.71\ N/mm} \tag{3.14}$$

where compression is measured in millimeters and desired force in Newtons. It has a max load of 127 lbs, approximately 565 N, and is 3” uncompressed and 1.5” compressed. A brushless outrunner motor directly drives the ball screw. The motor shaft is attached to the ball screw via a shaft coupler. The McMaster-Carr part number for the shaft coupler is 2463K4.

3.3.2 Guiding Bars and Friction Calculation

The spring plate is also guided by three ceramic-coated aluminum bars with a diameter of ¼”. These bars keep the spring plate moving linearly and counteract the moment on the spring plate by the force from the ball screw and the force from the spring. These bars go through bushings to reduce friction. Their McMaster-Carr part number is 7815K14. These bars are counteracting a relatively large moment, so the friction on the rods might significantly affect the actuator's functionality. Because of this, we decided to investigate these rods further. The diagram of the forces on the spring plate can be seen in Figure 3.17. Some assumptions were made in the force figure. Firstly, the problem was simplified to be two-dimensional. This means that the reaction forces from the two inner bars are represented by one set of forces, F_a. Also, it was assumed that the guiding bars do not fit perfectly inside the bushings on the spring plate, with there being some space between them. This, along with slack in the system, will cause the spring plate to be slightly rotated at a very small angle, thus making the guiding bars only contact the bushings at the bushing’s top and bottom points. The angle, however, was considered small enough that it could be assumed zero when calculating the moment caused by the spring and ball screw, and the moments caused by the rails. We also assumed that the bars guiding bars were counteracting the entire moment, meaning the forces from the internal spring bars are assumed to be zero. Finally, we assumed that the counter-moment forces on the bar were split so the force from the outer bar and the force from the inner bars were equal. For our first calculation, we considered the situation where the spring plate is being moved to compress the spring.

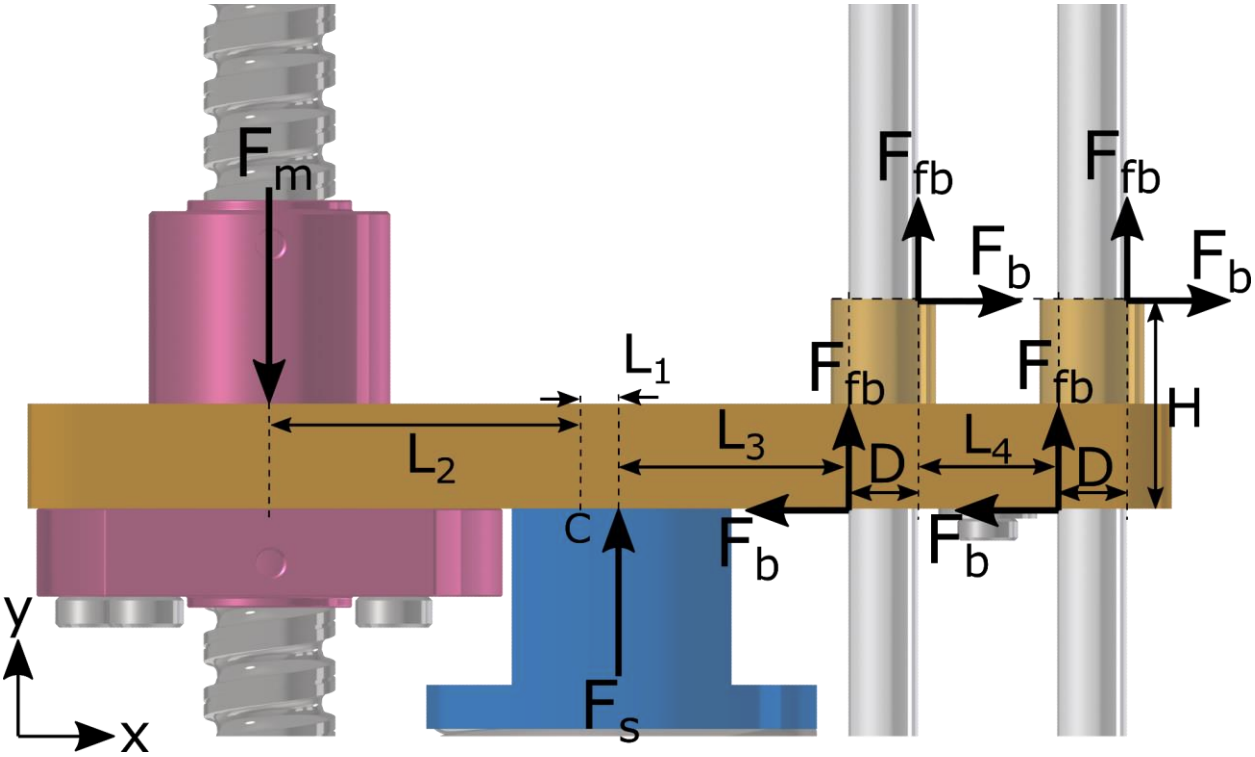

Figure 3.17: Diagram of forces on the spring plate when the plate is moving down to compress the spring. F_s is the spring force, F_m is the motor force, F_a and F_b are the forces on the guiding bars counteracting the moment caused by the motor and spring force, and F_{fa} and F_{fb} are the frictional forces caused by F_a and F_b respectively. C is the vertical line through the centroid of the spring plate

To investigate the effect of the friction on the guide rails, we looked at both the axial motor force, F_m, and the total frictional force, F_f. The full calculations to determine these forces can be seen in Appendix B. It is important to note that a positive acceleration is accelerating up, away from the spring, and a negative acceleration is accelerating down, towards the spring. The equation for the motor force is:

$$F_m = \frac{F_s(1+A) - m_{sp}a}{1-A} \tag{3.15}$$

where m_{sp} is the mass of all the components that move with the spring plate mass and A is:

$$A = \frac{2\mu}{H-\mu[2L_1+2L_3+2D+L_4]} \tag{3.16}$$

The variable μ is the coefficient of friction between the bushings and ceramic rods. The frictional force was found to be:

$$F_f = A(F_m L_2 + F_s L_1) \tag{3.17}$$

To find the motor torque we can use the ball screw torque equation [24]:

$$\tau_m = \frac{F_m P}{2\pi\eta} \tag{3.18}$$

τ_m is the motor torque, P is the ball screw lead in meters, and η is the ball screw efficiency. Once we had the equations, we could simulate the frictional force and motor torque. For our simulation, we tested a motor movement from 15 mm off the spring to 20 mm of compression, roughly estimating our expected compression movement at heel strike. We assumed the spring plate accelerated at 100 mm/s^2 and followed a pattern of constant acceleration then constant deceleration to reach its target position. The relevant dimensions and the spring plate mass were measured in the CAD model and can be seen in Table 3.3.

Table 3.3: Relevant dimensions for guiding bar friction calculations

Variable Name	Value	Unit
L_1	7.8	mm
L_2	24.9	mm
L_3	28.4	mm
L_4	13.5	mm
H	19.1	mm
D	6.4	mm
P	4	mm
m_{sp}	0.401	kg

The ball screw efficiency, η, was assumed to be 0.9. The coefficient of friction was tested at multiple values. It was found that for coefficients of frictions greater than 0.128, the frictional

moment overpowered the moment from the guiding bars, which implies that jamming would occur. We then assumed the coefficient of friction of 0.1. The plots of the forces and motor torque can be seen in Figure 3.18. As seen in the figure, the motor force required to move the spring plate peaks around 750 N at the end of the motion, where the friction force is about 450 N. The motor torque required to complete this motion was approximately 0.53 Nm. The motor we used has a maximum current of 60 A and a torque constant that is roughly 0.0191 Nm/A. This gives a max torque of 1.146 Nm. This is well above the required motor torque, showing that the actuator should be able to provide 20 mm of compression if the coefficient of friction is 0.1. However, the required motor torque grows rapidly at coefficients of friction above 0.1 and exceeds 1.146 Nm at coefficients of friction greater than 0.116. This can be seen in Figure 3.19. This means that rather slight increases in the coefficient of friction could cause serious issues with friction.

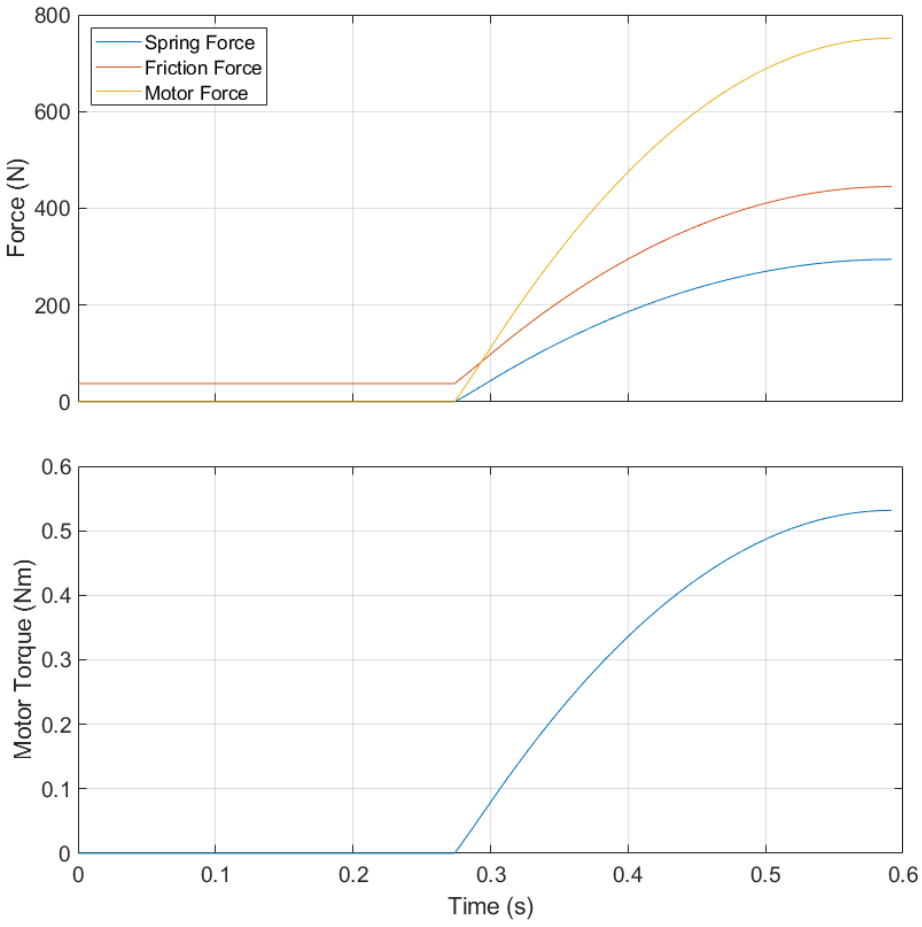

Figure 3.18: Spring, friction, motor force, and motor torque during the motion for μ = 0.1.

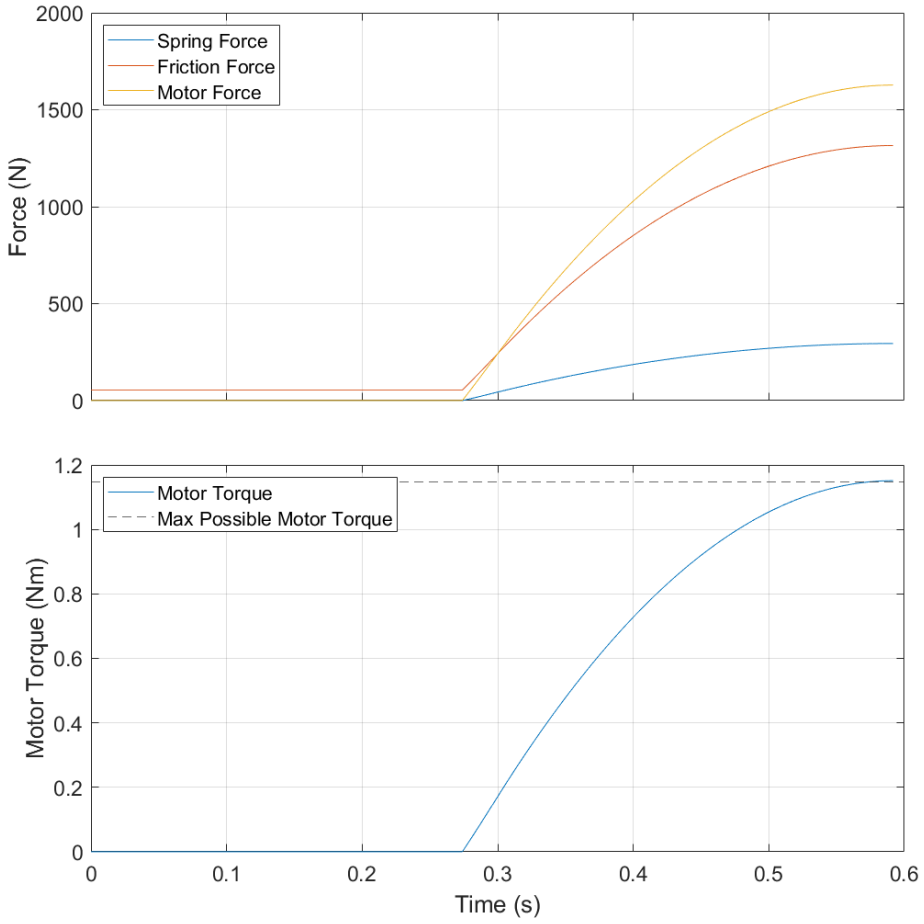

Figure 3.19: Spring, friction, motor force, and motor torque during the motion for $\mu = 0.117$.

3.3.3 Spring Tube

The spring is attached to a tube that extends down to the lower attachment point of the actuator. The tube has an endcap that inserts itself tightly in the bottom of the spring. Two steel rods, also ¼" thick, are used to keep the tube in line with the spring plate. These rods are inside the tube and spring. They are rigidly attached to the spring plate via bolts that thread into the top of the rods. They can slide through two holes in the endcap at the top of the tube. This endcap has bronze sleeve bearings inserted into it to decrease friction. Their McMaster-Carr part number is 6381K415. These rods also have washers attached at their other end, to prevent them from being pulled out of the endcap. A cutaway view of the tube endcap, the spring, and the adjustable rods

can be seen in Figure 3.20. Since these rods can slide within the endcap, the actuator can increase in size depending on the position of the legs without pulling on the spring. The choice in design to allow the spring plate to separate from the spring means that the spring will only ever be compressed, it will never be in tension. If the spring was in tension it would try to pull your leg up, which would impede the user's natural gait. This separation also allows the motor to back the spring plate off the spring during swing state. If the spring plate was rigidly attached to the top of the spring, the motor control system would have to actively adjust the position of the spring plate during swing to keep the spring perfectly uncompressed. This would require a very complicated control algorithm, which would still realistically be imperfect in predicting where your leg position will be, thus resulting in unwanted compression and tension during swing. Overall, the actuator design allows for the spring to be compressed during stance, and completely released during swing state.

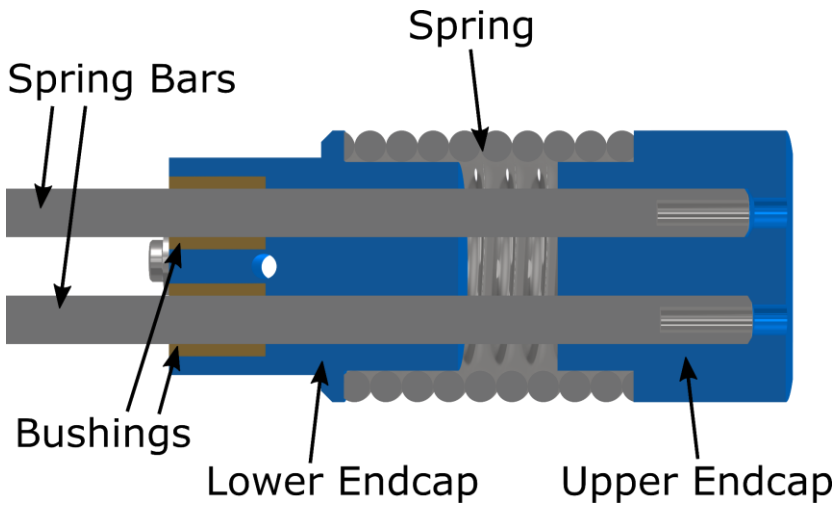

Figure 3.20: Cutaway view of the spring mechanism. The lower endcap is connected to the spring tube and the upper endcap is connected to the force tube.

This actuator design allows for the support force to be easily adjusted and completely removed depending on the current stage of the gait cycle. While the leg is in swing phase, the plate can be easily moved away from the spring, leaving it uncompressed and thus producing no spring force.

Then, when the leg is in stance, the plate can be moved down to compress the spring, thus providing the desired ground support force. If we desire this force to vary based on the gait percentage or current ground reaction force, we can easily vary the provided spring force.

3.4 Human Interface/Soft goods

To interface the exoskeleton with the user, thigh straps, calf straps, overshoes, and a waist belt are used. The main components of the human interface can be seen in Figure 3.21.

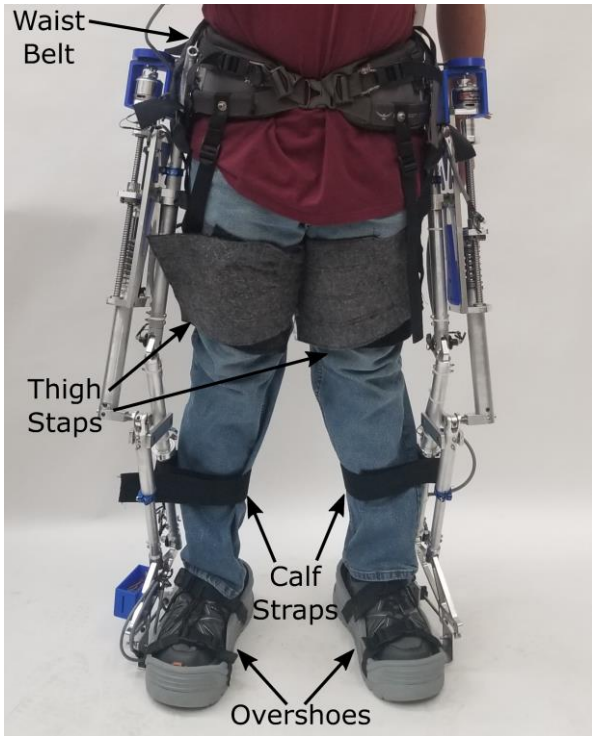

Figure 3.21: Main components of the human interface.

3.4.1 Upper Leg Human Interface

The upper leg soft goods, the waist belt and the thigh straps, are connected together via vertical straps. A diagram of the waist belt and thigh strap system can be seen in Figure 3.22. The thigh strap is not directly connected to the exoskeleton. The force from the exoskeleton is designed to be primarily transferred to the user through the thigh straps. This should provide a more comfortable experience for the user than if the force was being provided by the waist belt pushing against the waist. To achieve this, the straps between the waist belt and the thigh strap are adjusted to be taught, thus transferring the force upwards on the waist belt through the straps and to the thigh strap. A metal plate is bolted to two thick foam linings on the waist belt. The metal plate has a hole in it that the hip joint is attached to.

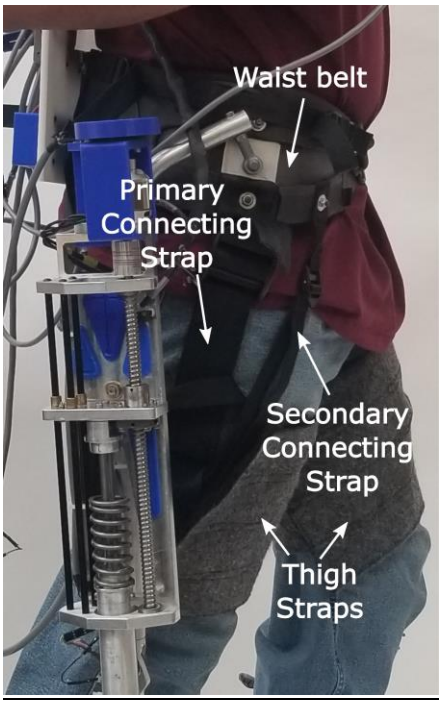

Figure 3.22: The upper leg soft goods. The waist belt is connected to the thigh strap via a primary and secondary strap. The primary strap transfers the force from the waist belt to the thigh strap. The secondary strap is to help ensure a good fit and goes through a loop in the thigh strap and connects at two points on the waist belt, one between the front and side of the belt and one between the front and back.

3.4.2 Lower Leg Human Interface

The lower leg's interfacing includes the calf straps and overshoes. The lower leg human interface can be seen in Figure 3.23. The calf strap is attached to the lower leg frame right above the adjustment point on the inner pantograph bar. The calf straps are kept from sliding up the exoskeleton by collars on the tube and prevented from sliding down by the clamps used to adjust the tube length. The straps keep the exoskeleton in line with the leg, making it follow the natural shape of the user's leg. The overshoes fit snugly over the user's shoes and are further secured by two adjustable straps with buckles on each shoe. One strap is near the toes of the user and one is near the ankle, like the straps of a pair of sandals. The overshoe has a metal plate bolted to the bottom of the shoe near the ball of the foot in a slot where material has been removed to fit it. This plate protrudes from the side of the overshoe and is bent at a 90-degree angle up. The part of the plate that is bent upwards has a hole in it that is used to attach the ball joint at the foot. This plate is where the supporting force is transferred to the ground. A similar plate is attached at the bottom of the foot near the back of the overshoe. This plate is connected to the linear sleeve bearing on the back of the foot piece.

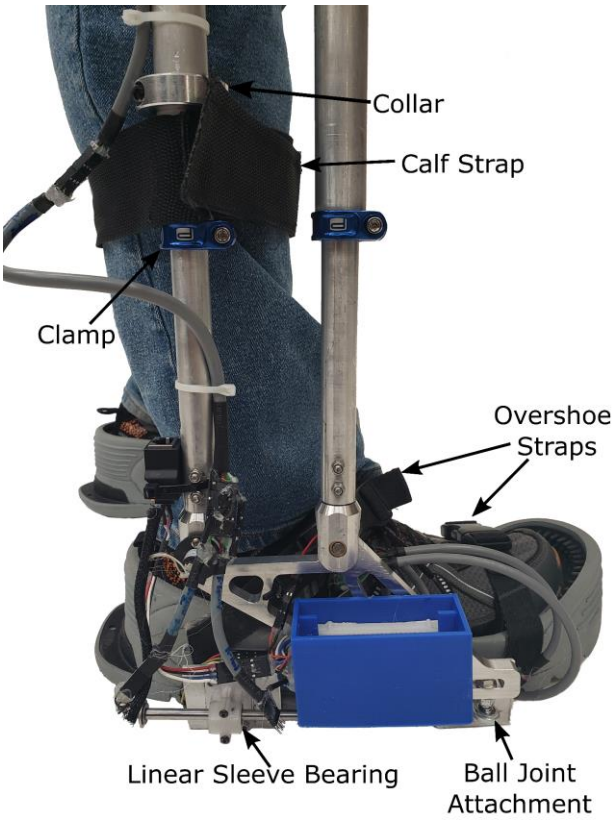

Figure 3.23: Lower leg human interface. The calf strap is attached directly to the exoskeleton and keeps it orientated well with the user's leg. The overshoe is strapped to the foot and is attached to the foot piece of the frame.

3.5 Electronics Design

For our electronic design, we used the Adafruit HUZZAH32 – ESP32 Feather microcontroller. We chose this processor for its small size, dual cores, fast 240 MHz clock speed, and its compatibility with Arduino IDE. We used the ODRIVE V3.6 motor controller. The ODRIVE was chosen for its peak current of 120 A and its easy programming via its python library. The full electrical diagram can be seen in Appendix C.

3.5.1 Foot Sensors

To determine the user's current motion in the gait cycle, we used two load cells and time of flight distance sensors per foot. They are embedded into the overshoe, with one of each sensor being at the heel of the foot and ball/toe region of the foot. The electrical diagram for the foot sensors can be seen in Figure 3.24. The overshoe sensor setup can be seen in Figure 3.25.

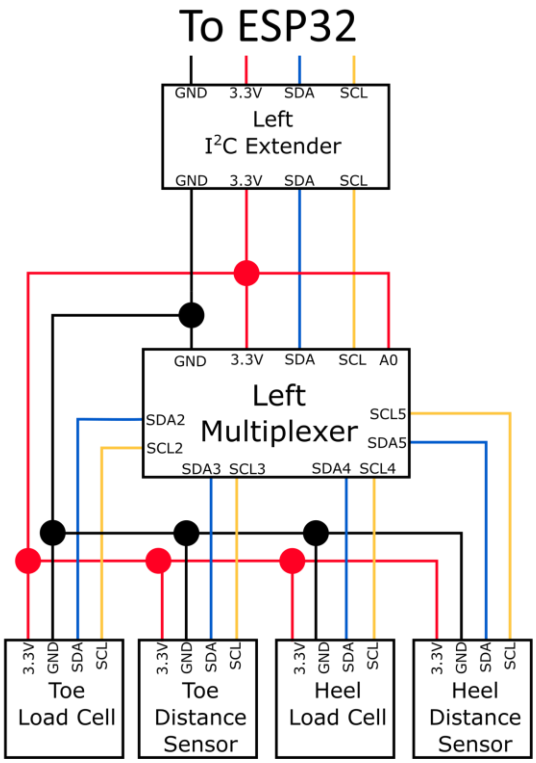

Figure 3.24: The left leg I^2C sensor setup. Each foot has two distance sensors and two load cells wired through a multiplexer. There is an I^2C buffer at the top of the wire path that extends the I^2C range and reduces errors due to motor noise. Between the buffer and the multiplexer, the I^2C lines are run through an ethernet cable. On the left multiplexer, port A0 is pulled to 3.3V. The pin is not pulled to 3.3V on the right multiplexer. This pin is used to change the multiplexer's address allowing us to differentiate between the left and right multiplexer when communicating with them. The SDA and SCL lines are pulled up via 680 Ω resistors.

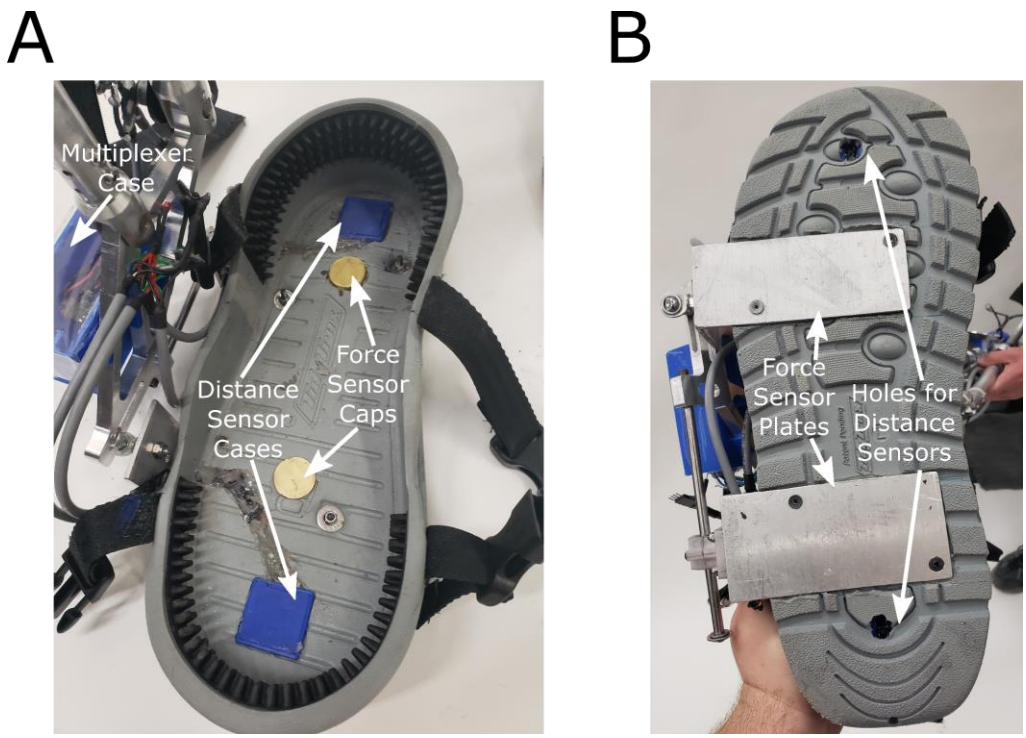

Figure 3.25: A. The inner overshoe. B. The bottom of the overshoe. The distance sensors are in cases with holes at the bottom so that they can see through the shoe and measure the distance to the ground. The Force sensors are underneath the force sensor caps and above the plates. The force sensors read the weight of the user going through the caps and to the ground through the plate.

We chose to use the FX29K0-100A-0100-L load cells. They can read up to 100 pounds of force and communicate via I^2C. We used the Adafruit VL6180X Time of Flight distance ranging sensor to measure distances with a range of approximately 5 mm to 100 mm. Like the load cells, they communicate via I^2C. Both of these sensors are run on 3.3V power. With the default parameters of the distance sensors, they can read at approximately 10 Hz. In order to increase the sample rate of the sensors, we had to change a few of these parameters. These changes can be seen in Table 3.4.

Table 3.4: Parameters changed on the VL6180x

Parameter	Default Value (ms)	Updated Value (ms)
Max Convergence Time	30	3
Integration Time	50	5
Average Sample Period	48	1
Intermeasurement Period	100	13

The end result of these changes was a faster sample rate with the trade-off being decreased accuracy and maximum distance. The distance we could accurately read was reduced to about 75 mm. However, we were able to up the sample rate to approximately 100 Hz. This trade-off was worth it for the increased sample rate, which improved the temporal resolution of our distance data.

We chose to have both types of sensors communicate via I^2C because it allowed us to read multiple sensors through the same wires. The distance sensors and load cells, however, do not have an easy way to change their addresses to differentiate between other sensors of the same type. If they were all run through the same wire, any request from the distance sensors would cause all the distance sensors to respond at once and be unable to be read. The same issue would occur with the force sensors. In order to prevent this, the foot sensors are run through a multiplexer. We used the TCA9548A I^2C multiplexer. To read the sensors we can send a command to the multiplexer to open its port leading to the sensor we want to communicate with, read the sensor, and then close the port. The multiplexer's address can also be easily changed by pulling specific pins on the board to the 3.3V supply voltage. This lets us use a multiplexer at each foot, thus letting us have only one set of SDA/SCL wires run down each leg for I^2C communication. There are also I^2C extenders between the ESP32 and the multiplexers, used to extend the range of the I^2C lines. We used the Adafruit LTC4311 I2C Extender. These were a late addition but were necessary, as they significantly increased the I^2C signal integrity, which was degraded due to electrical interference from the motors.

3.5.2 Joint Encoders

The knee and ankle angles are measured via absolute magnetic encoders. We used the MAE3 Absolute Magnetic Kit Encoder, part number: MAE3-A10-250-220-7-B. The electrical diagram for the joint in coders can be seen in Figure 3.26. These are primarily used to determine the position of the exoskeleton's frame, which can be used to calculate the current actuator length. This is then used when computing the desired plate position on the actuator to provide the needed force. The

encoders output an analog signal, which has a voltage proportional to the angle they are reading. Because the sensors run on 5V power, a simple voltage divider is used to step their output signal down to a safe range to be input into the ESP32. To calibrate the encoders, we set the exoskeleton's joints to angles that roughly represent a neutral standing pose, 180° at the knee and 90° at the ankle. We then adjusted the sensor until its output voltage matched the expected value for the given angle. Since the encoders are absolute, this zeroing does not need to be repeated every time the exoskeleton is restarted.

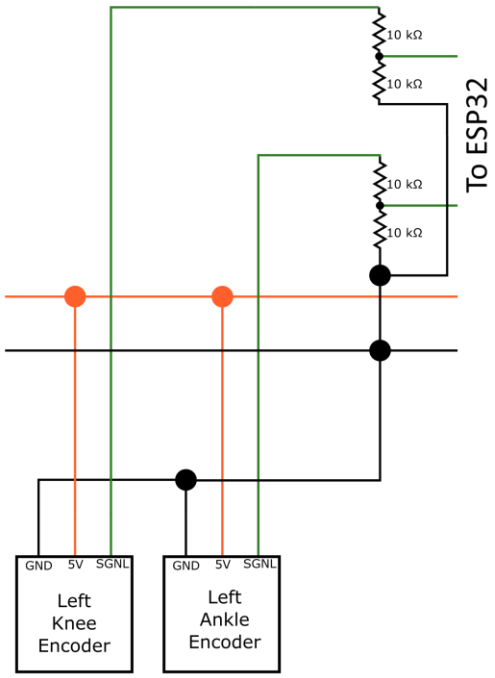

Figure 3.26: Encoder setup for one leg. The green wires at the top feed into the analog inputs of the ESP32. The black wires go to ground and the orange wire to the 5V rail.

Valefod LM2596 Buck Converters are used to step down the voltage from our batteries to provide power for the joint encoders and the ODRIVE. The encoders run on 5 volts, so their power is provided by a 9V battery that is stepped down to 5V. To provide power to the ODRIVE, two 14.8V LiPo batteries wired in series are used. We then step the output down to 25V. The buck converter setup can be seen in Figure 3.27.

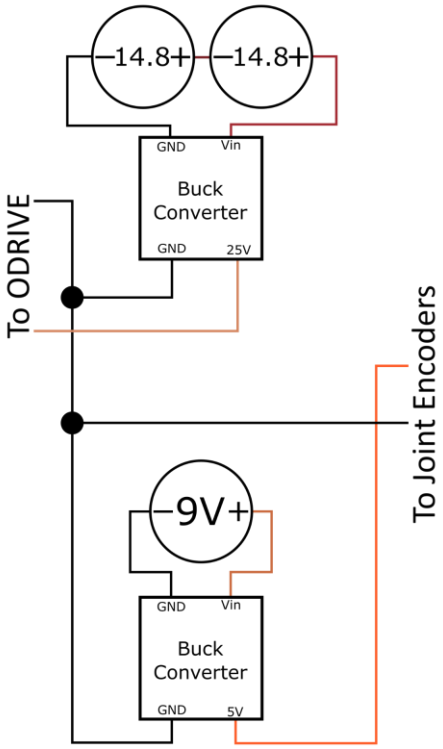

Figure 3.27: Electric diagram of the buck converters used to step down a 9V battery for the joint encoders and two 4s LiPo batteries for the ODRIVE.

3.5.3 Actuator Electronics

Each actuator is run via a brushless DC motor, the Turnigy AerODRIVE SK3 - 5045-500KV Brushless Outrunner Motor. They are each connected to an optical encoder, specifically the E3 Optical Kit Encoder (E3-1000-315-IE-D-D-B), which has its output fed back into the motor controller for closed-loop position control. The motor and motor encoder electrical diagram can be seen in Figure 3.28. The motor wires are looped through a ferrite ring to reduce noise. This helps reduce errors in the other sensors and in the motor encoder lines. Just like the I^2C extenders, this was a later change to help mitigate issues with electrical noise.

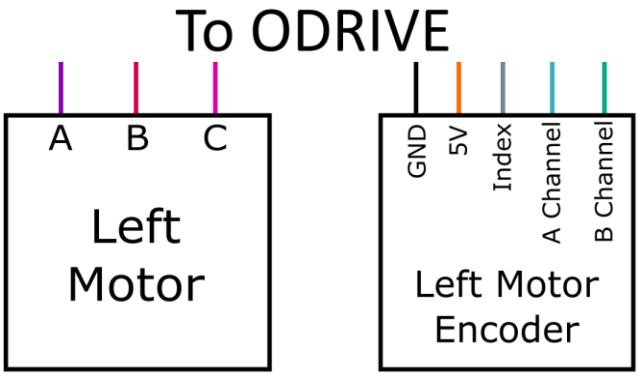

Figure 3.28: Electric diagram of the left motor and its encoder. Each wire goes directly into its relevant port in the ODRIVE.

There are also two limit switches per actuator, which are used to zero the motor and prevent the plate from colliding with the top or bottom of the actuator while walking. The part number for the limit switches is MS-105A01 (H). The limit switches are pulled up via the ESP32 and are run through an RC filter to debounce the switch and prevent noise from the motor accidentally setting off the limit switches and turning off the system. The limit switch electrical diagram can be seen in Figure 3.29.

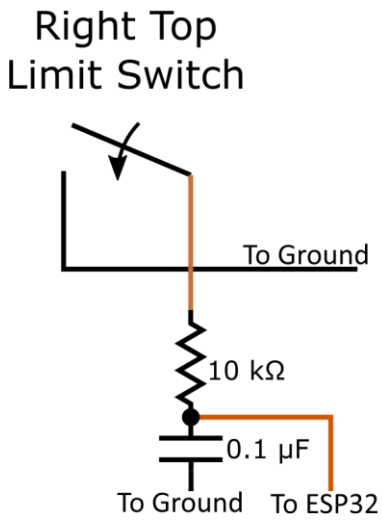

Figure 3.29: Electrical diagram for a limit switch. The input pin on the ESP32 is internally pulled up. The switch pulls down the pin when closed. An RC filter is used to debounce the switch and filter out motor noise.

3.5.4 Electronics Board

The final electronics board can be seen in Figure 3.30 below. Each board had a PLA 3D printed case made to mount it in. The ESP32 was mounted in its case on a breadboard that also had all of the simple circuitry on it. The cases were then mounted to a laser-cut acetal resin board. This board was mounted to the back of the waistband, as seen in Figure 3.31.

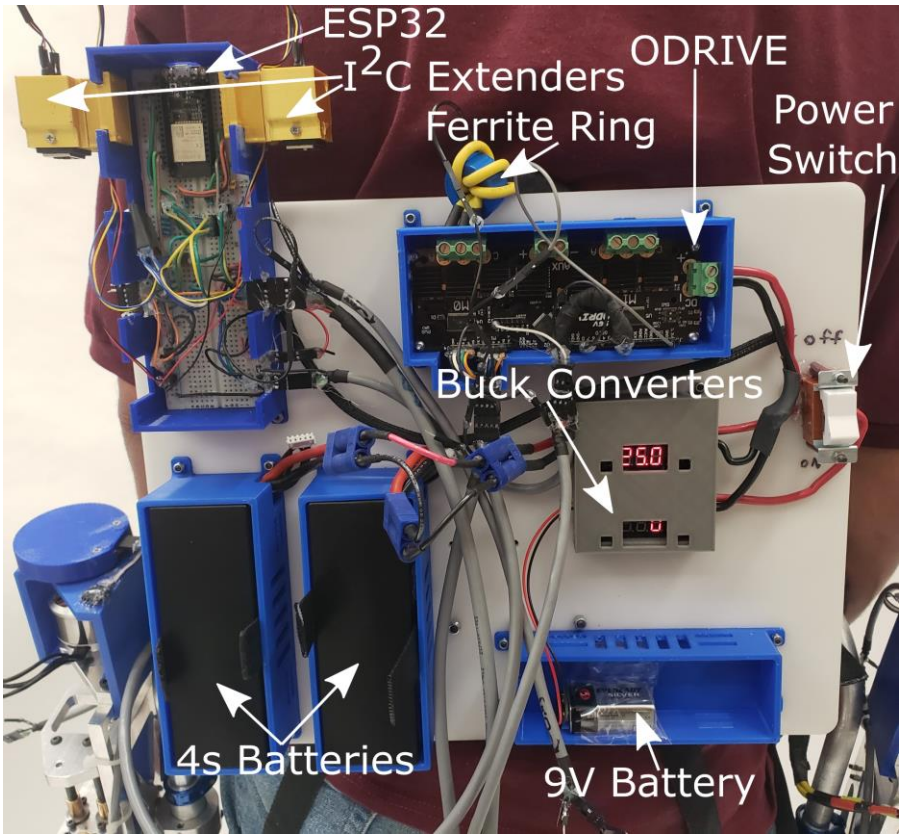

Figure 3.30: Final electronics board setup for one motor.

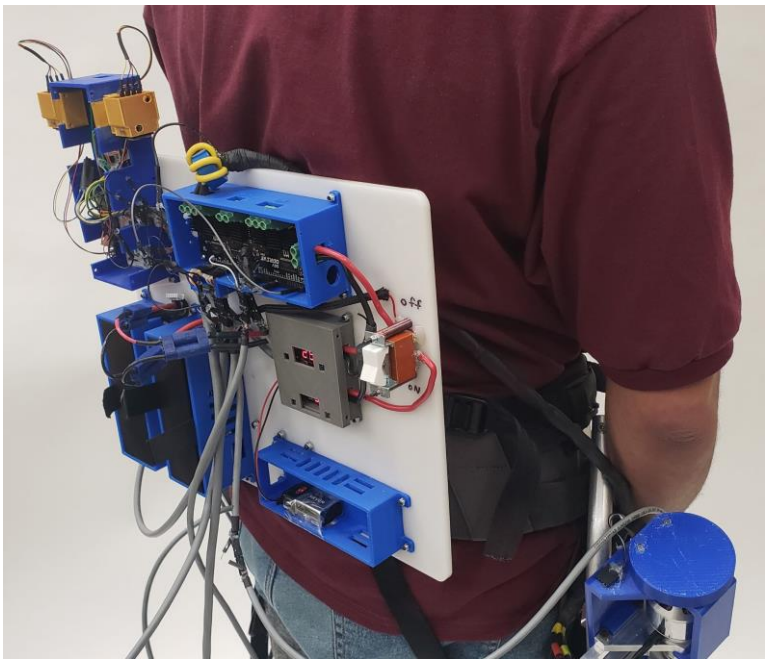

Figure 3.31: Electronics board mounted to the back of the waist belt.

3.6 Software

The software of the system controls the actuator based on the user's current stage of gait. It sends commands to the motor controller to move the spring plate off the spring in swing and to compress the spring in stance. Our main program is written in Arduino and runs on two cores. One core reads the foot sensors and uses a state machine to determine the current stage of gait, which is then used to update the desired compression of the spring. The other core reads the encoders to determine the positioning of the leg, uses the positioning of the leg to determine where the top of the spring is in the actuator, and then uses that knowledge along with the latest updated desired compression from the other core to tell the ODRIVE where to send the spring plate. The ODRIVE then handles the control of the motor to move the spring plate.

3.6.1 ODRIVE Software

For the motors, we used the ODRIVE to handle the low-level control of the motor position. This means that we simply sent desired motor position commands to the ODRIVE and it runs its built-

in closed-loop motor control to achieve our desired motor position. We can also override previous commands with new position commands, changing the ODRIVE's motor position goal even when it is in the process of a previous movement. To control the motor movement, we use the ODRIVE's trapezoidal position control scheme [25]. Figure 3.32 shows what trapezoidal position movement looks like. The acceleration and deceleration limits are in turns/sec^2 and can be adjusted separately from each other. The velocity limit is in turns/sec and can also be separately adjusted. The motor control parameters, the deceleration and acceleration limits, the velocity limit, current limit, and other minor parameters, for the ODRIVE were modified using the ODRIVE tool in python. The values that we used for these parameters will be discussed in Chapter 4.

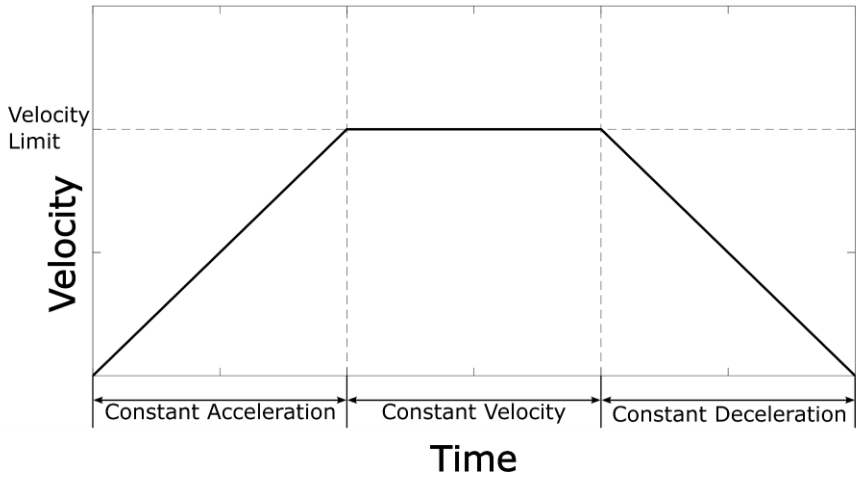

Figure 3.32: Trapezoidal trajectory used by the ODRIVE to control the motor. The velocity increases at a constant rate until it reaches the velocity limit. It stays at a constant velocity until it reaches the position where a constant deceleration would put it at the correct position once zero velocity is reached. It then decelerates.

3.6.2 Arduino Program Setup

The main program was made in Arduino. A diagram outlining the Arduino program can be seen in Figure 3.33. The setup for the program first starts the interrupt that controls when the sensors can be read. It sets up the force sensors to read at 200 Hz and the encoders to read at 100 Hz. Next,

the setup zeroes the motor. To do this, the program slowly moves the spring plate up, checking between each small movement to see if its corresponding top limit switch was pressed. When the corresponding limit switch is pressed, the program saves the current position and moves the spring plate off the limit switch. We can use this point to translate the desired motor position relative to the limit switch to a position relative to the ODRIVE's zero point, which corresponds to its position at startup. Finally, the setup takes three encoder points before the main control loop begins, which allows the filtered encoder values to converge to the measured values. After the setup, there is a different main loop running on each core of the ESP32. One handles reading the force sensors and then using the sensor data to determine the current stage of the gait cycle. The other core reads the joint encoders and sends commands to the ODRIVE.

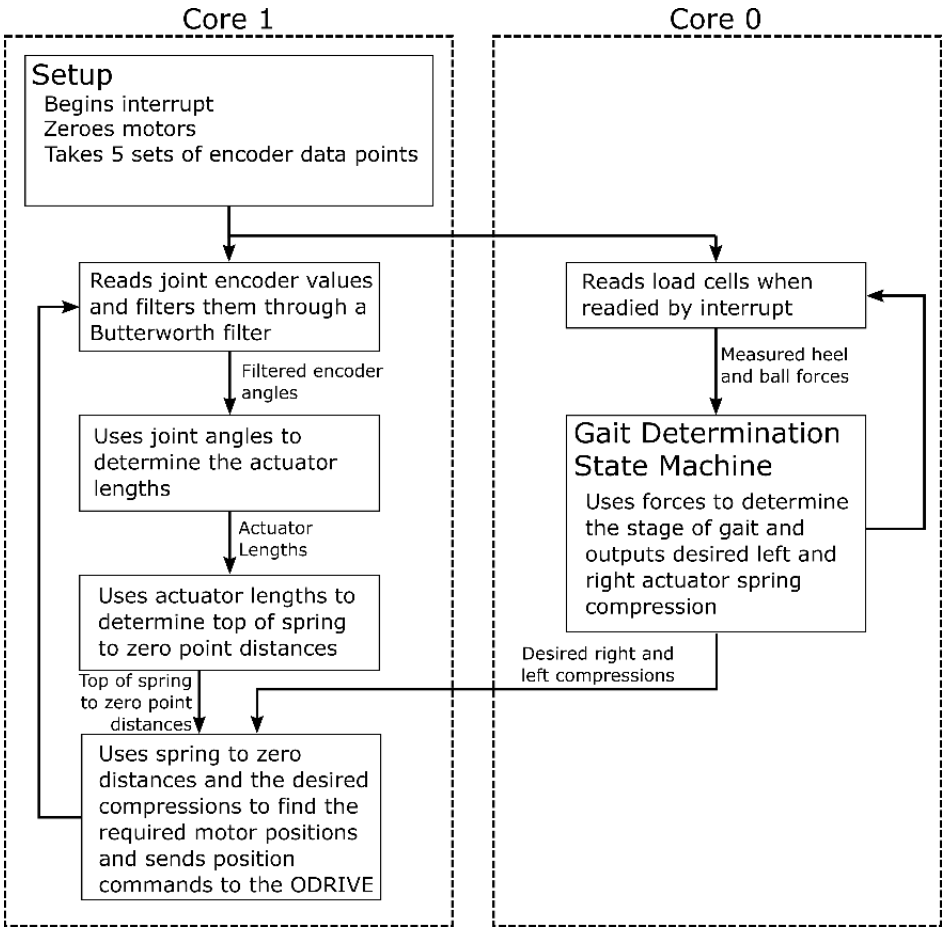

Figure 3.33: Diagram of the overall structure of the main Arduino program controlling the exoskeleton.

3.6.3 Gait Determination

The gait determination is split into two stages, swing and stance. As seen in Figure 3.33, the gait determination core first reads the values of the foot sensors when readied by an interrupt and then inputs them into the gait determination state machine. This state machine determines the desired compression of the spring. Its state machine diagram can be seen in Figure 3.34. While in swing, the desired compression is set as a negative value. This will be sent to the ODRIVE in the other core's loop. To detect heel strike, the program simply waits for the heel force to peak above a predetermined threshold that was chosen based on our preliminary walking tests. Once heel strike is detected, the program is switched to stance mode, where it outputs the desired spring compression The stance mode is split up into multiple stages to help define the gait. This allows for both the initial heel-lift and then the foot lift to be predicted. To detect the heel-lift, the program waits for the force measured by the heel force sensor to have a consistent negative slope while the ball force has a consistent positive slope. This indicates that the heel is lifting and the weight of the body is being transferred to the ball of the foot. This can be seen in the figure of the gait cycle, Figure 3.35. The program then waits for the heel force to stabilize close to zero pounds and the toe force to start consistently decreasing. It can also change to swing mode if the force on both the toe and heel drops below predefined thresholds at any state during stance, preventing missing the transition from stance to swing if the force data differs from its expected pattern.

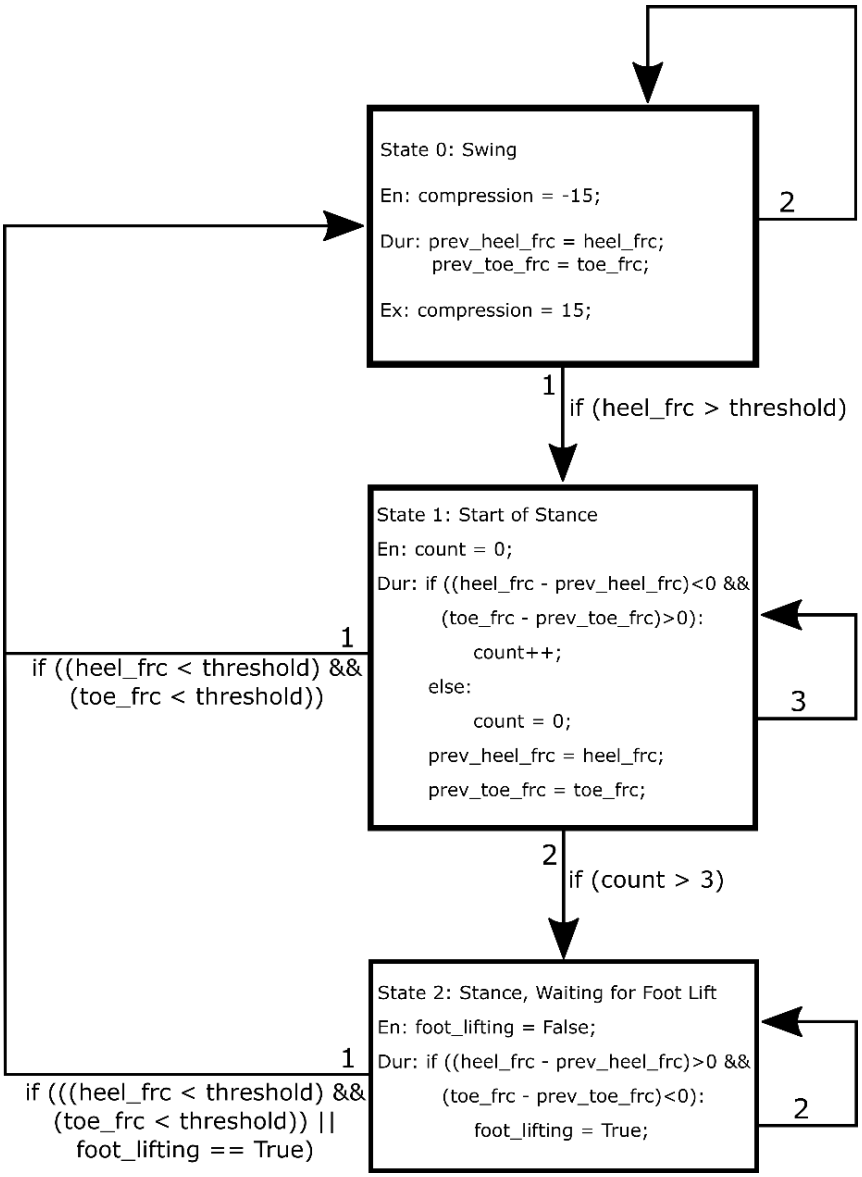

Figure 3.34. Gait determination state machine. The compression is the main output of the system and is sent to the other core to determine the desired motor position. State 0 represents the swing state while 1 and 2 represent parts of stance. En is the entrance code for each state and is executed when the state is entered from a different state. Ex is the exit code that occurs when leaving the state to go to another. Dur is the during code that is executed during each loop of the state. The lines going between the state have their exit condition in an if statement along the lines and are numbered starting with 1 in the order they are checked. The final numbered line from each state is a loop back to the same state, representing the state machine staying in the same state and repeating the during code if no exit condition is met.

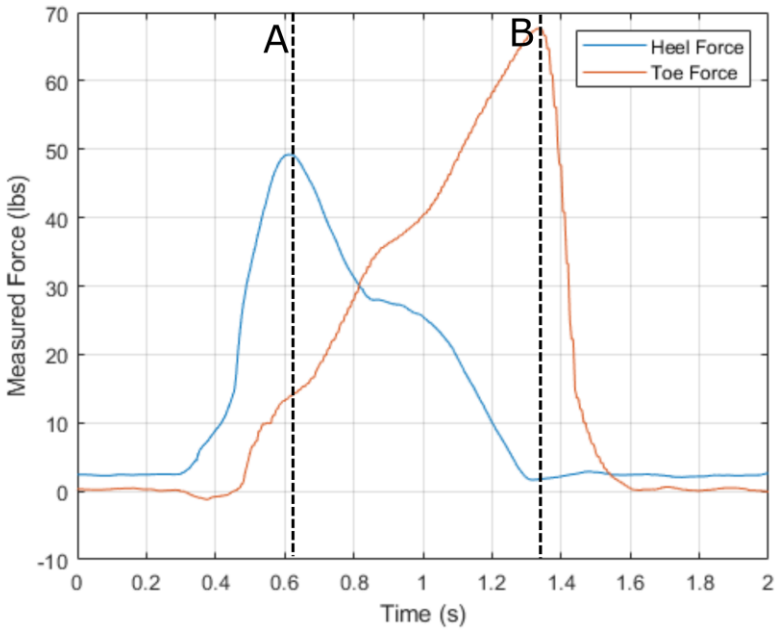

Figure 3.35: The foot lift detection waits for point A, where the heel force starts consistently decreasing and the toe force is consistently increasing. Then after point A is reached, the program waits for point B, where the heel force stabilizes around zero pounds with a slope of about zero and the toe force starts rapidly decreasing. At point B the actuator starts releasing the spring to allow the foot to lift.

3.6.4 Joint Encoders and Motor Commands

The second core reads the joint encoders and sends motor commands. The program first reads the raw analog input of the joint encoders and converts them to angles based on the ratio of the input voltage to the max input voltage of 3.3V. This raw reading, however, is very noisy, and if used directly to calculate the desired motor position results in turbulent movement in the actuator. To smoothen the output position and, thus, the output motor position, the encoder values are sent through a second-order Butterworth filter with a cutoff frequency of 8 Hz. Example encoder data before and after filtering can be seen in Figure 3.36. Because of the filtering, the encoder data needed to be read at a consistent rate. The foot sensors, especially the distance sensor, read at a varying, slower rate. This meant that in order to get the consistent read speed we required from the encoders, the encoder and motor functions were moved to their own core.

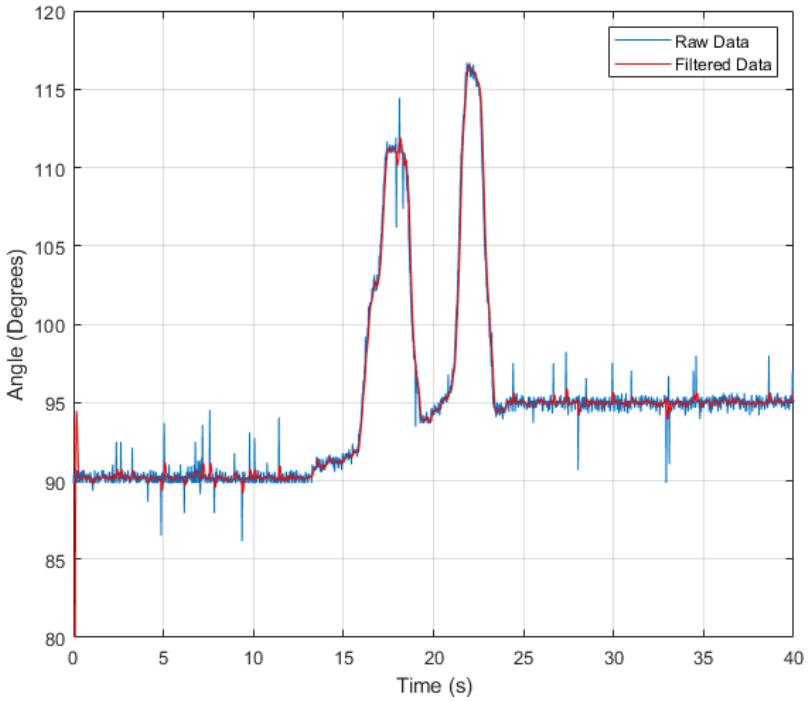

Figure 3.36: Example of angle data before and after being filtered.

The filtered encoder values are then used to find the actuator length. This can be found by using the x and y components of the actuator length found earlier in equations 3.10 and 3.11.

$$L_{actuator} = \sqrt{{x_{actuator}}^2 + {y_{actuator}}^2} \tag{3.19}$$

where θ_a is the ankle angle and θ_k is the knee angle. The required lengths to solve the equations can be seen in Table 3.5 below. These were measured on the actual exoskeleton, instead of being taken from the CAD.

Table 3.5: Lengths measured on the exoskeleton required to find the actuator length

Variable	**Length (mm)**
L_u	254.8
L_k	163.0
W_p	75.0
L_l	37.3
W_k	14.1

The actuator length is then used to calculate the input position required to move the spring plate to just touch the top of the uncompressed spring. This is simply the distance from the zero point, the top limit switch, to the top of the spring minus the thickness of the spring plate mechanism, since the zeroing occurs at the top of the spring plate and it contacts the spring at the bottom. The required lengths to find this can be seen in Figure 3.37 below. The equation used to calculate the length is:

$$Dist_{top\ of\ spring} = L_a + L_z - L_t - L_s \quad (3.20)$$

A description of the lengths and their measured values can be seen in Table 3.6.

Table 3.6: Lengths required to find the distance from the zero point to the top of the uncompressed spring.

Variable Name	Description	Measured Length (mm)
L_a	Actuator length found above	Variable
L_z	Length from the upper actuator mount to the zero point (upper limit switch)	61.1
L_t	Length of the spring-tube (From the lower actuator mount to the top of the uncompressed spring)	293.7
L_s	Length from the face of the endcap that compresses the spring to the top of the spring plate	29.6

The program can then use the calculated distance from the ODRIVE's zero point to the top of the uncompressed spring and the desired spring compression to calculate the necessary position of the spring plate. This is then converted to a number of motor rotations from the ODRIVE's zero point and is sent as a motor command. This position in units of motor rotations can be found using:

$$Required\ Motor\ Position = \frac{Dist_{top\ of\ spring} + Desired\ Compression}{Ball\ Screw\ Lead} + Setpoint \quad (3.21)$$

The setpoint is the position we measured when zeroing the spring plate. This is added to convert the input position to the motor from being relative to the limit switch to relative to the ODRIVE's zero point. This setpoint will always be negative, as the motor always starts below the limit switch.

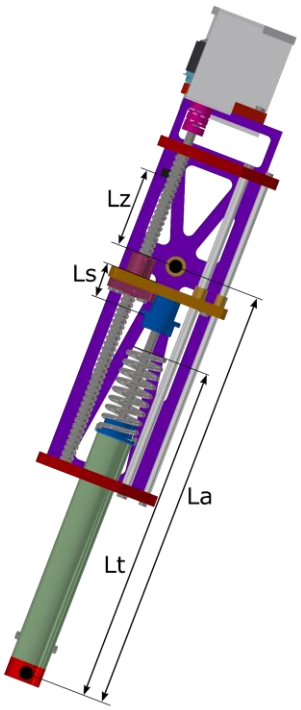

Figure 3.37: Actuator diagram of the required lengths to determine the top of the spring position. L_z is the length from the upper actuator mounting point to the spring plate's zero point, the upper limit switch. L_t is the length from the lower actuator mounting point to the top of the uncompressed spring. L_s is the spring plate's thickness, from the face of the endcap that contacts the top of the spring when compressing it to the top of the plate that pushes the limit switch.

These calculations all also work with a negative compression value, meaning that we can keep the spring plate a constant distance from the top of the spring by using a negative compression, allowing us to use the same equations to both provide compression in stance and maintain a constant distance in swing. There are also limits on the positions we allow the program to send a command to ODRIVE to go to. These are set up after the zeroing of the motor at the start of the program and keep the ODRIVE from trying to move the spring plate outside of the physical limits of the actuator. Thus, the physical limit switches are merely a safety backup, as a command that hits them should not be able to be sent under normal circumstances.

Chapter 4

Testing/Results

4.1 Walking Tests without Actuation

4.1.1 Preliminary Walking Tests without Frame

Our first set of testing was to collect walking data to better understand the characteristics of the gait cycle. Specifically, we recorded the forces measured by the load cell sensors at the foot. We also recorded distance data from time-of-flight distance sensors at the heel and ball of the foot. It is important to note that going into testing there were some differences in the electronics than in the final design, mainly that the I^2C extenders, RC filters on the limit switches, and ferrite rings were not yet added. For our first set of tests, we took walking data using just the foot pieces holding the sensors without the frame. While the frame is designed to follow the dimensions of the body, wearing the frame will still have effects on the user's gait, so these preliminary tests without the frame give a clearer picture of the force and distance for an unimpeded walk. This data was taken before the parameters of the distance sensor were tuned, so the temporal resolution is less than that of later tests, with the foot sensors reading at approximately 43 Hz. Example data can be seen in Figure 4.1. By looking at the force data, we can see a clear structure to the stance phase of the leg. The heel force relatively quickly peaks when the heel of the foot strikes the ground. Then the force is gradually transferred to the ball of the foot. The force at the ball of the foot then peaks as the user pushes themselves forward, which is then followed by a quick drop in force when they lift their leg. The distance sensors can give us insight into the movement of the foot in both stance and swing. For stance, the heel distance increases as the user lifts their heel when pushing themselves forward. This could be used to predict when the foot is about to fully lift off the ground. In swing, the heel distance stays high, until the heel strikes the ground. The ball of foot distance, however, peaks and then falls again right a little bit before heel strike, which could be used as a predictor.

Then it peaks again right as the heel strikes, before quickly falling as the front of the foot moves to the ground. Overall, the walking data gave a clear picture of how the forces and distances at the foot look during gait and gave many potential metrics we could use to predict heel strike and foot lift.

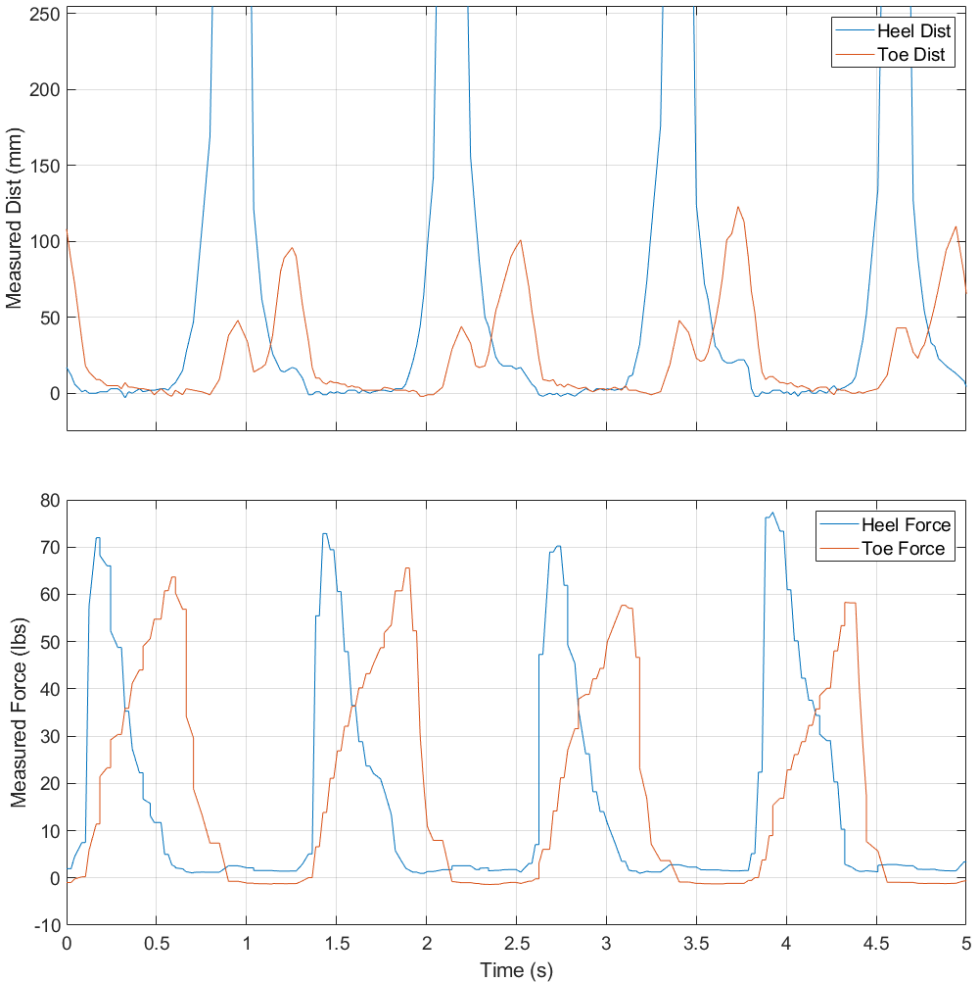

Figure 4.1: Force and distance data for walking at 3.2 kph (2 mph) without the frame on.

4.1.1 Walking Tests with Frame

We then performed walking tests with the frame on, but without running the actuator. At this point, the distance sensors had been tuned to increase their sample rate, meaning the distance sensors

were reading at 100 Hz and the force sensors at 200 Hz. This can also be seen in the increased temporal resolution and decreased max distance value. The results of this test can be seen in Figure 4.2. It is important to note that any distance above approximately 75 mm will read as 255, as that is the value the sensor returns when it does not converge on a value. While the frame was designed to minimize interference with the natural motion of the leg, it still changed the user's gait. The distance data appears to first peak then return back down, then the toe distance peaks and drops right before the force peaks. The initial force from heel strike also seems much lower. This gentler heel strike might mean the user is lifting their foot less than they do when not inhibited by the frame. While there are differences from before, the force follows generally the same pattern, thus the methods discussed above for determining foot lift should still work. Since the frame was attached, we were also able to collect data for the knee and ankle joint angles. These do not exactly match the actual angle of the user's leg, as the exoskeleton functioned better when slightly more bent than the user's actual leg. The ankle distance seemed to not change much in relation to the gait cycle. The knee joint angle seems to first decrease at the start of swing state then increase before quickly dropping at heel strike. It then gradually increases over the stance phase.

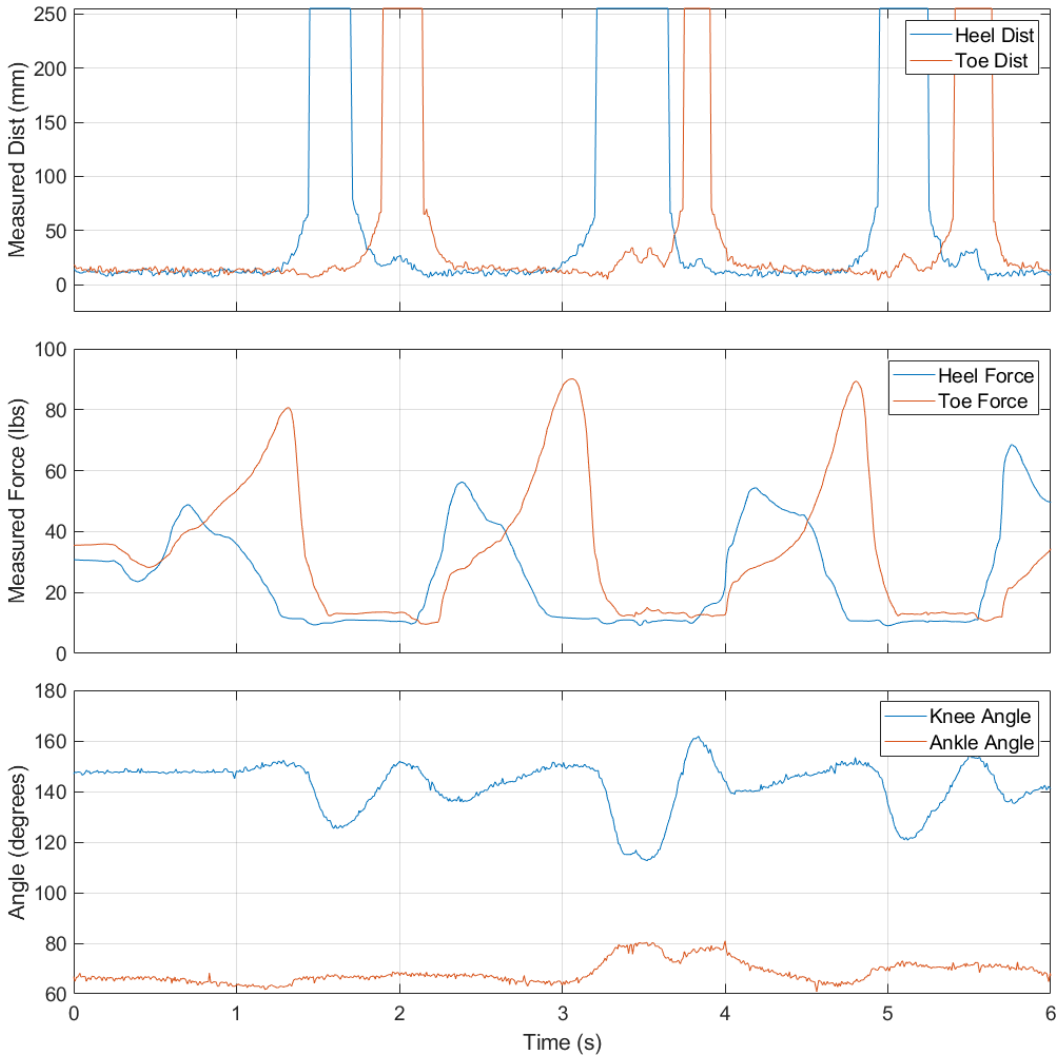

Figure 4.2: Walking data with the frame on but no actuation.

4.2 Range of Motion Tests

To check that the exoskeleton would not limit the user's mobility, we took measurements of the RoM of the exoskeleton. The exoskeleton was attached to a stand and each joint was moved to its physical limits. A goniometer was then used to measure the joint angle relative to the neutral position. Pictures of some of the exoskeleton's physical limits can be seen in Figure 4.3.

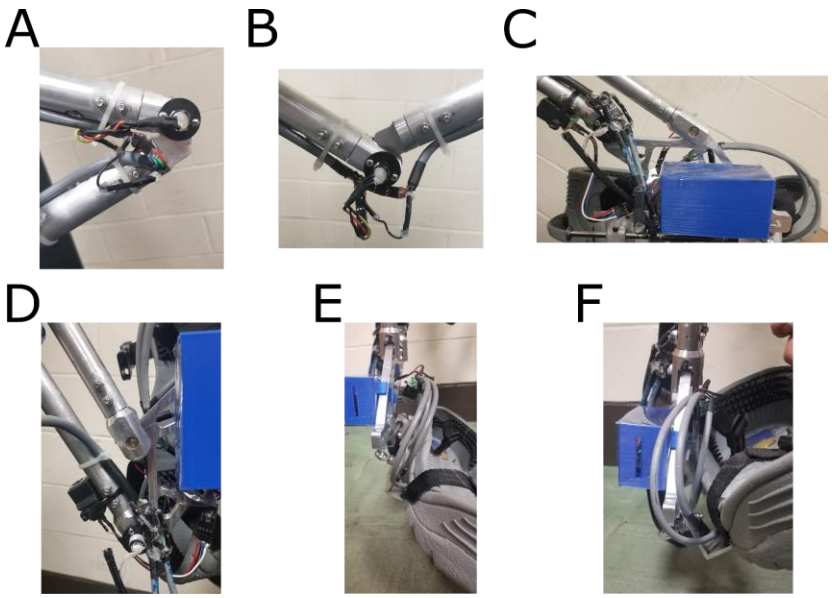

Figure 4.3: Examples of physical limits for joints. (A) Knee Flexion, (B) Knee Extension, (C) Ankle Plantarflexion, (D) Ankle Dorsiflexion, (E) Ankle Eversion, and (F) Ankle Inversion.

The results from the RoM testing can be seen in Table 4.1. As stated in section 1.2, 0° for hip motions is defined as a neutral leg in such that it would be in line with the upper body while standing. Extension is defined as negative and flexion as positive. Similarly, adduction and internal rotation are defined as negative and abduction and external rotation as positive. For the knee, a perfectly straight leg position is defined as 180°, with flexion being negative and extension being positive. For the ankle, a neutral standing position with the foot perpendicular to the calf is defined as 90°, with plantarflexion being positive and dorsiflexion being negative. For eversion and inversion of the ankle, the neutral standing position is defined as 0°, with eversion being positive and inversion being negative. These ranges of motion will likely not match perfectly with the end user's actual RoM, as the exoskeleton joints are not perfectly aligned with the user's hip and ankle joints in the frontal plane. Specifically, hip adduction/abduction, internal/external rotation, and ankle eversion/inversion might differ when wearing the exoskeleton.

Table 4.1: Range of motion of final exoskeleton.

Motion	Range of Motion (Degrees)
Hip Extension/Flexion Range of Motion	(-180) - 180
Hip Adduction/Abduction Range of Motion	(-17.5) - 14
Hip Internal/External Rotation	(-17.5) - 17.5
Knee Extension/Flexion Range of Motion	44 - 237.5
Ankle Dorsiflexion/Plantarflexion Range of Motion	60 - 148
Ankle Eversion/Inversion Range of Motion	(-31) - 37.5

The range of all the joint motions exceeded our requirements, meaning the structure of the exoskeleton should not limit joint mobility in normal movements such as walking, sitting, or squatting.

4.3 Benchtop Actuator Tests

4.3.1 Benchtop Test Overview

To determine the efficacy of the exoskeleton's actuation system, we conducted benchtop tests. These consisted of compressing the spring while the exoskeleton was mounted to a table to help us assess the ODRIVE controller and the actuation system before having a person wear it. A picture of the benchtop test setup can be seen in Figure 4.4. As seen in the figure, the exoskeleton was clamped at the foot-piece, at the top of the frame near the hip joint, and around the mid-calf. The clamping of the exoskeleton kept its positioning rigid. This was important to be able to compress the spring without having it move the frame. The electronics board was set on the table next to the exoskeleton. For most of the benchtop testing, we connected the ODRIVE to a laptop. This allowed us to send commands directly to the ODRIVE. This differed from how commands will work in the final implementation, where the ESP32 will send motor commands to the ODRIVE. We also hooked up a dead man's switch, which had to be held down for the motors to receive power from

the ODRIVE. The actuator was also covered with a plastic bin while it was operating, and safety glasses were worn during operation. The system was also designed to shut off if the spring plate was driven into any of the limit switches at the end of the actuator. For the analysis detailed in this section, the majority of testing was conducted on the right leg of the exoskeleton, but identical benchtop tests were conducted on the left leg.

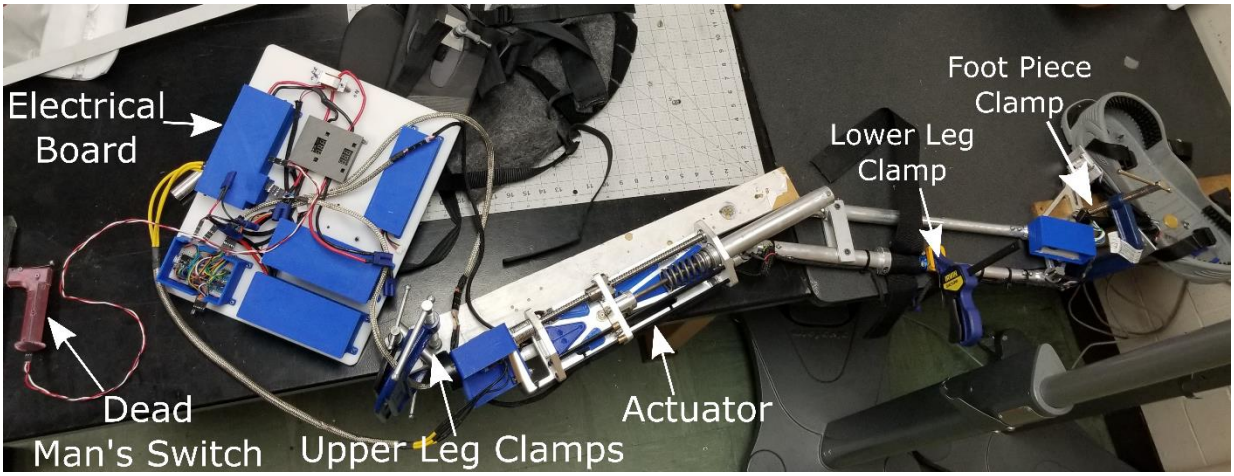

Figure 4.4: Benchtop setup to test the actuation system.

4.3.2 Slow Compression Tests

The first tests we ran with the right actuator were evaluating if we could move the spring plate while it was off of the spring. To do this we simply moved the spring plate up and down the actuator above the spring via motor commands. We discovered that while moving the spring plate that we could expect up to 10 A of current draw while not on the spring. We also were able to tune the ODRIVE's parameters. The acceleration and deceleration limits were tuned to 30 turns/s^2 and the velocity limit was tuned to 125 turns/s. These parameters were used for all the benchtop tests. The current limit was also set to 55 A. We then moved on to compressing the spring. We first measured the current as we slowly compressed the spring. The resulting plot can be seen in Figure 4.5. From the plot, we can see that the current drawn appears to increase by about 1 A per millimeter compressed, with a zero compression current of 3 A. To provide our desired support

force, we expected to need about 20 mm of compression, which based on this set of tests would only require approximately 23 A. This was promising as 23 A is well below the 55 A current limit.

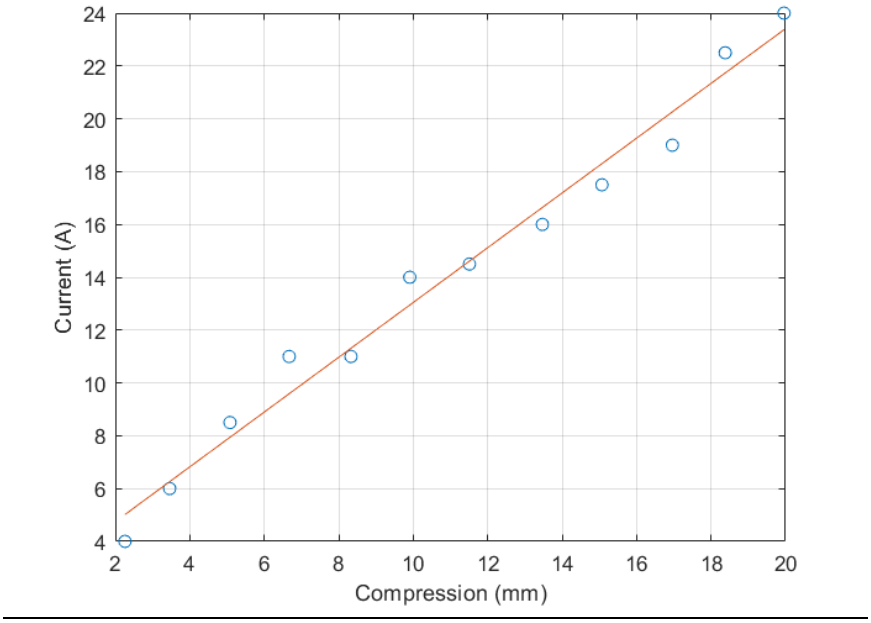

Figure 4.5: Current drawn from the ODRIVE as the compression of the spring increases. The spring was not released between compression steps for these tests. The current approximately increases by 1 A per mm compressed.

4.3.3 Quick Compression Tests

The slow compression tests, however, do not accurately reflect how the spring would be compressed while assisting in walking. In the actual implementation, the spring will be compressed in more of a ramming motion, with the spring plate going directly from off the spring to the desired compression. This differs from the compression testing where the spring force was slowly increased. Our next series of tests involved backing the spring plate off the spring by two turns, equivalent to 8 mm, and then sending it directly to the compression level we wanted to test. The results of the first set of these tests can be seen in Figure 4.6. The current at first appeared to increase by about 0.8 A for each mm increase, with an increased zero current of 4.6 A. However, upon manually measuring the compression of the spring, it was found that the actual spring plate

position differed from the commanded position. To investigate this difference, we calculated the set position minus the measured position. The results of these calculations can be seen in Figure 4.7. It appears that the error increases by approximately 1 millimeter for every 6 millimeters of compression specified. There was a zero offset of about -0.68 mm of error at zero compression. The zero offset is likely a measurement error, as the measured compression was manually found using calipers. We were also able to observe this position error via the position plots in ODRIVE's python tool. This error is likely due to the fact that the ODRIVE only uses a proportional gain for its position control loop [26]. To better understand how the current was being affected by the true compression, we corrected the set compression by assuming an error rate equal to approximately 1 mm of error per 6 mm of compression. The current due to the corrected compression can be seen in Figure 4.8. This resulted in the current increasing about 0.95 A per millimeter, very close to the first compression test, except now it had an increased zero offset of 4.62 A. This result seemed to roughly agree with the initial testing, giving us more confidence that 1 A per millimeter of compression would accurately reflect the current in all loading scenarios.

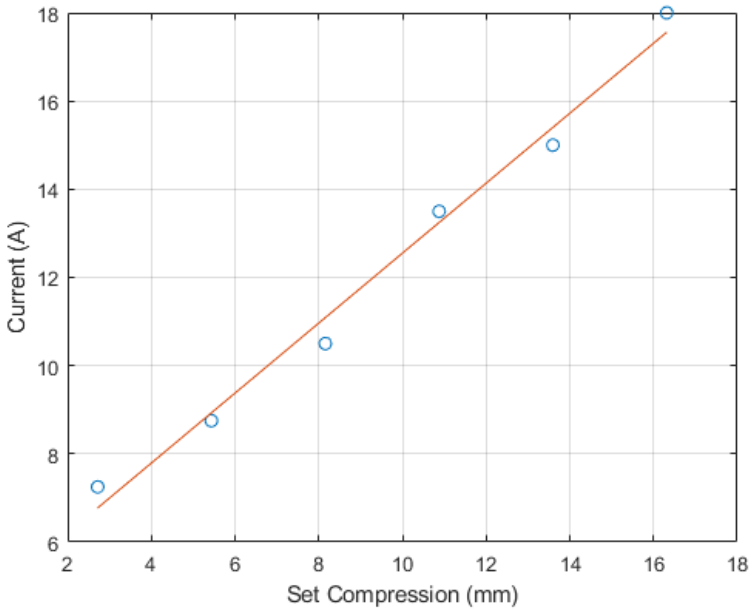

Figure 4.6: The current drawn by the ODRIVE over the compression value sent via command to the ODRIVE. The current increases by approximately 0.79 A per mm of compression, with an offset of about 4.62 A.

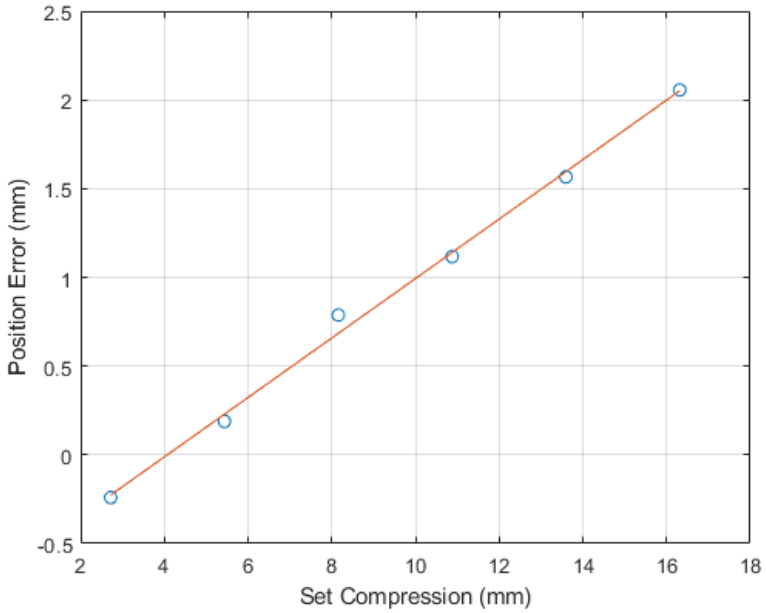

Figure 4.7: The position error, set compression minus measured compression, compared to the set compression. The error appears to grow at about 1 mm of error for every 6 mm of set compression. There is a zero offset of about -0.68 mm.

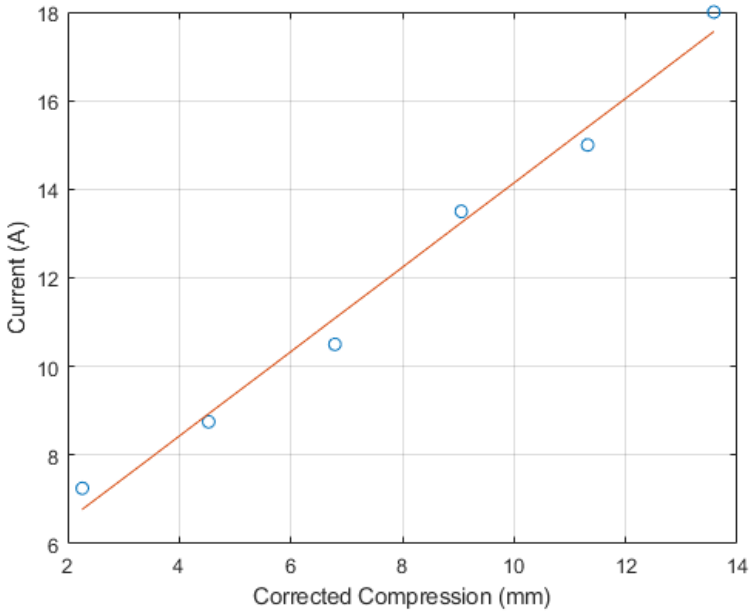

Figure 4.8: The current drawn from the ODRIVE over the measured compression. The current increases by approximately 0.95 A per mm of compression, with an offset of about 4.62 A.

To further assess the current, compression, and compression error, we decided to do further ramming tests of compression values that provide 200 N of force and above. These represent forces that we might expect to use in the actual implementation of the exoskeleton. We tested from 14 mm to 24 mm of compression which represents a range of approximately 200 to 350 N. The plots of current as a function of set compression and corrected position and the plot of position error can be seen in Figure 4.9 to Figure 4.11. The position error increases at a similar rate as before but appears to have a zero offset of about 2.5 mm. The zero offset could be due to a combination of measurement errors and flaws in the test setup. The most likely reason was that the clamps did not secure the frame well enough and the spring force from one of the compressions could have slightly moved the frame, making an offset between the measured and set compression. The set compression was also manually zeroed at the point where it just contacts the uncompressed spring. The caliper measurements could also be the cause of a small amount of the error in the zero offset. Error in the zero point could also have affected the offset. Since the error rate was similar to the previous error rate, it was assumed the zero offset was a testing error and the corrected compression was calculated with no offset. This resulted in the current drawn increasing at a rate of about 1.35 A per millimeter of measured compression, with a much greater zero offset of about 4.33 A. This was a slight increase from the previous test, but this would still allow for up to 20 mm of compression without exceeding the current limit.

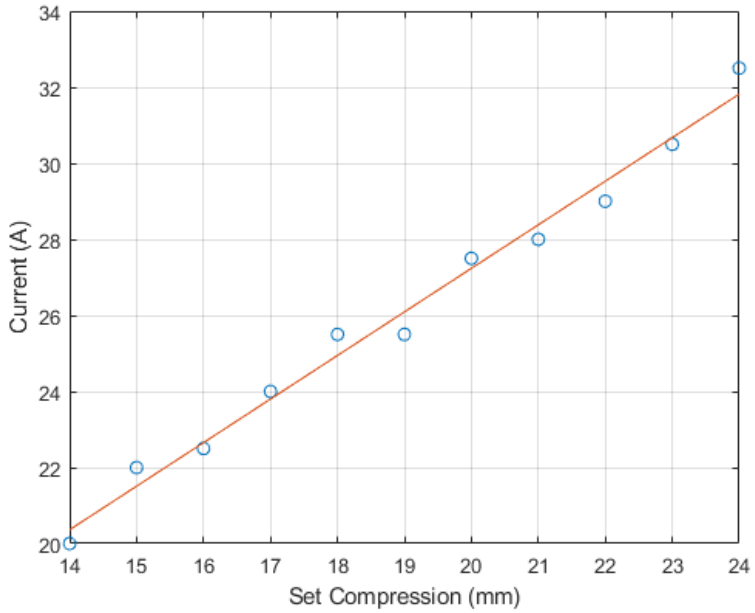

Figure 4.9: Current drawn from the ODRIVE over the set compression. The current increases by approximately 1.15 A per mm of set compression, with a zero offset of about 4.33 A

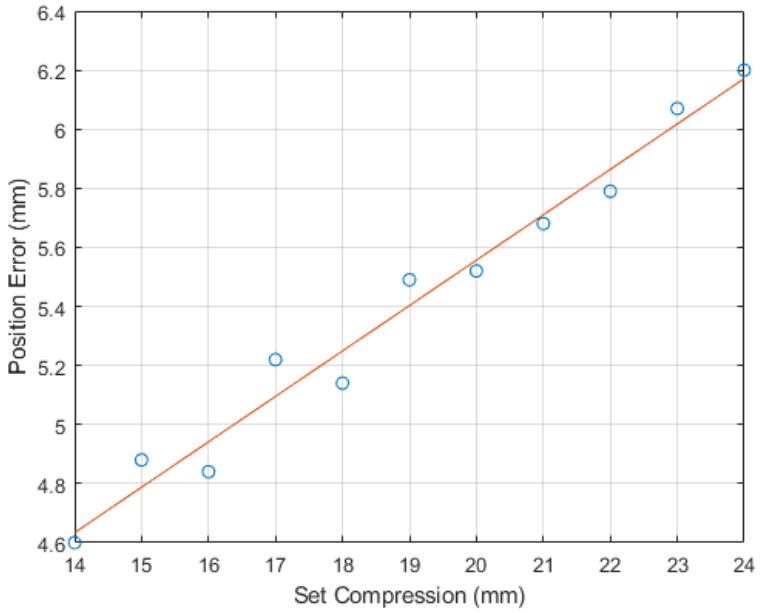

Figure 4.10: The position error, set compression minus measured compression, compared to the set compression. The error appears to grow at about 1 mm of error for every 6.5 mm of set compression. However, the zero offset appears to be about 2.48 mm.

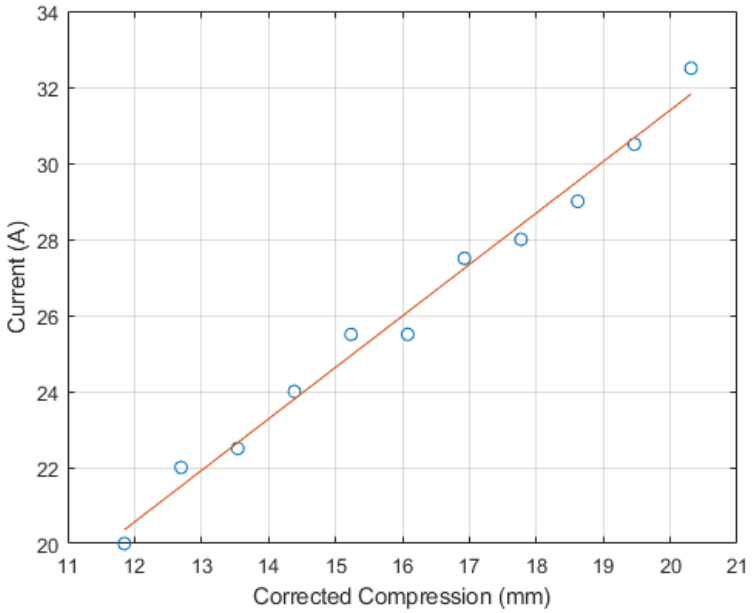

Figure 4.11: The current drawn from the ODRIVE over the measured compression. The current increases by approximately 1.35 A per mm of compression, with a zero offset of about 4.33 A.

During the benchtop testing, we also found an issue in addition to the difference between the position we set and the actual position. With the ODRIVE, we experienced an error where it would cut off motor power and throw an error when 35 A was exceeded on a single motor, regardless of the current limit set. The ODRIVE is supposed to be able to provide up to 60 A of continuous current per motor and we had set the limit to 55 A, but any current spike above 35 A would trigger a current error. This issue remained unresolved through the testing of the exoskeleton. During this stage of testing, this still let us get to our desired compression for our right leg actuator, but it limited the velocity and acceleration we could run the system at without risking an amperage spike that shuts the motors off. The left leg fared worse overall. It had significantly more frictions due to imperfections in the manufacturing and thus was much more likely to trigger the 35 A limit.

4.3.4 Testing of Overall Code

After testing the actuator by sending commands directly to the ODRIVE, we tested our overall code, thus testing the communication between the ODRIVE and the ESP32. While the code itself

worked as expected, we found another problem, noise from the motor. Motor noise affected a lot more of our setup than we had expected. The motor caused high voltage spikes on other signal lines due to electrical noise, particularly the limit switch lines and I^2C lines. Because of the motor noise, we added the low pass RC filters to the limit switches discussed in the electrical design section. We also decreased the pullup resistor values to increase the current through the I^2C lines. The system was rewired to use pre-shielded cables instead of the wires we had shielded ourselves. While these changes did decrease the noise on the wires, the motor noise remained an issue going into walking testing with actuation, especially with the I^2C lines, where the noise still caused a lot of misread data points. A plot of the raw I^2C data from the force sensors can be seen in Figure 4.12. The number of errors, seen in the spikes in the data, is considerable. The spikes down to -255 represent points where the force sensor's status bit indicated an error had occurred. Those are relatively easy to filter out and are automatically discarded by the program. The smaller spikes, on the other hand, are harder to programmatically remove. The distance sensors had more trouble operating with the noise, so they were not used in the final program for the walking with actuation tests.

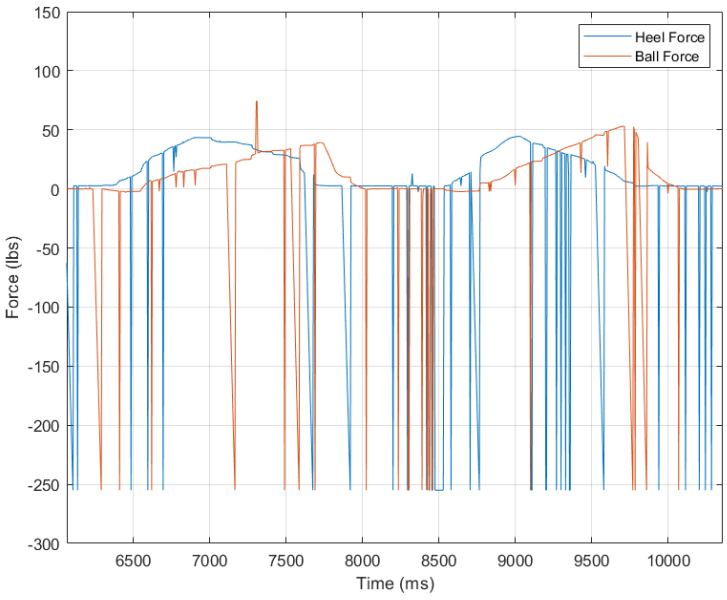

Figure 4.12: Raw force data read by the program. All spikes going down to -255 are automatically filtered out by our program, as those represent points where the force sensor's status bit indicated that an error had occurred. Smaller spikes needed to be detected manually.

The communication between the ODRIVE and the ESP32 mostly worked as expected, with the only exception being the incorrect motor position estimate reported from the ODRIVE. The ODRIVE's reported current position estimate of the motor appeared to be delayed. This delay was not observed when running the system, so it is assumed to be a communication delay from the ODRIVE. The delay can be seen in a sample of position data in Figure 4.13. We found the data better followed the desired position commands to the ODRIVE when shifted forward in time by 600 ms. This can be seen in Figure 4.14. We also found that the position reports from the ODRIVE were only updating at a rate of approximately 5.8 Hz. This means that most of the detail of the motion is lost. For the analysis in this paper, all the reported position values will be shifted forward by 600 milliseconds. This does, however, mean that we cannot determine the actual delay between sending a signal to the ODRIVE and the ODRIVE's execution of the command.

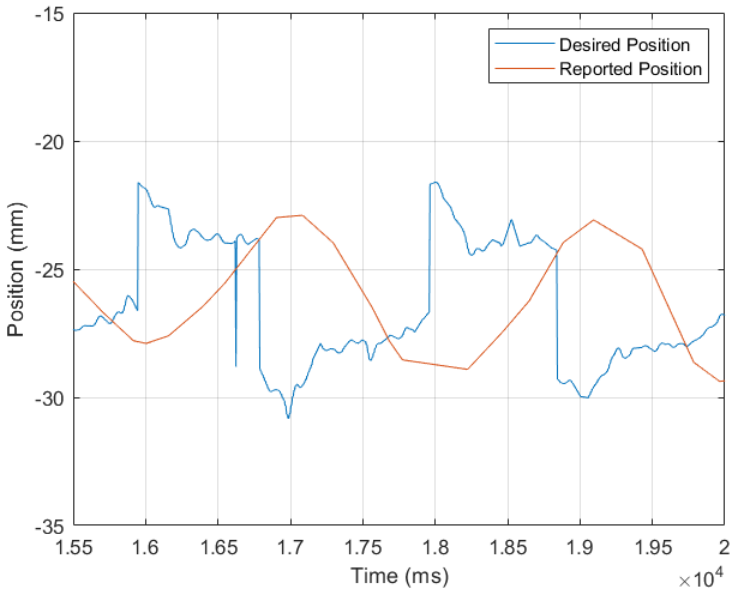

Figure 4.13: Desired position sent to the ODRIVE compared to the reported position from the ODRIVE. The positions are relative to the zero point of the actuator (the top limit switch) and converted from turns of the ball screw to millimeters of spring plate travel. The reported position from the ODRIVE only reads at 5 Hz and appears to be delayed.

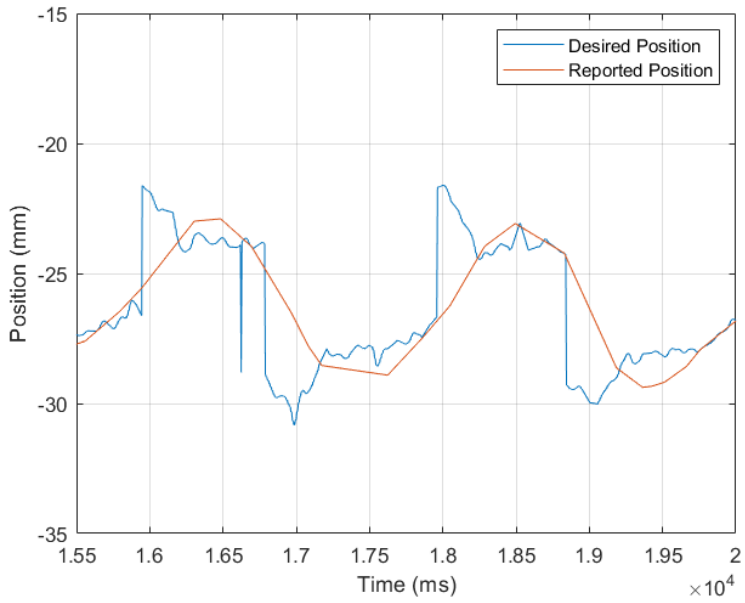

Figure 4.14: Reported position of the ODRIVE adjusted to remove the approximately 600 ms delay.

4.4 Walking Tests with Actuation

4.4.1 Initial Actuated Walking Test

The final stage of data collection was walking tests with actuation. For this set of tests, we only actuated the right leg. We conducted multiple walking tests with varying levels of ground support force. For this discussion, we will be focusing on one of the final datasets we took, which was with 15 mm of set compression. This compression had to be used, as we found that levels of compressions above this led to the current spikes while walking that exceeded 35 A. While this is less than our desired compression, it is still enough to test the efficacy of our design. This amount of compression corresponds to approximately 220.7 N (50 lbs) of spring force. However, based on the error rate we found of 1 millimeter of error per 6 millimeters of set compression, we can estimate the real compression was closer to 12.5 mm. This corresponds to about 183.9 N (41 lbs) of spring force. The data collected from this trial can be seen in Figure 4.15. As seen in the figure, despite filtering out any errors reported by the force sensors, there was still a lot of misreading due

to the motor noise. This level of error also effectively reduces the resolution of our motor data at various points where many errors occurred. The angle data looks as expected, with a clearer pattern being seen in the ankle than even in the non-actuated test. While the actual position data from the ODRIVE was undesirable, with its delay and low sample rate, we could still see that the motor was responding slower than desired. This meant that the motor providing and releasing support was slower than we would want for ideal assistance. Besides the speed of the motor, the program worked as expected and correctly switched between compressing and releasing the spring based on the stage of the user's gait.

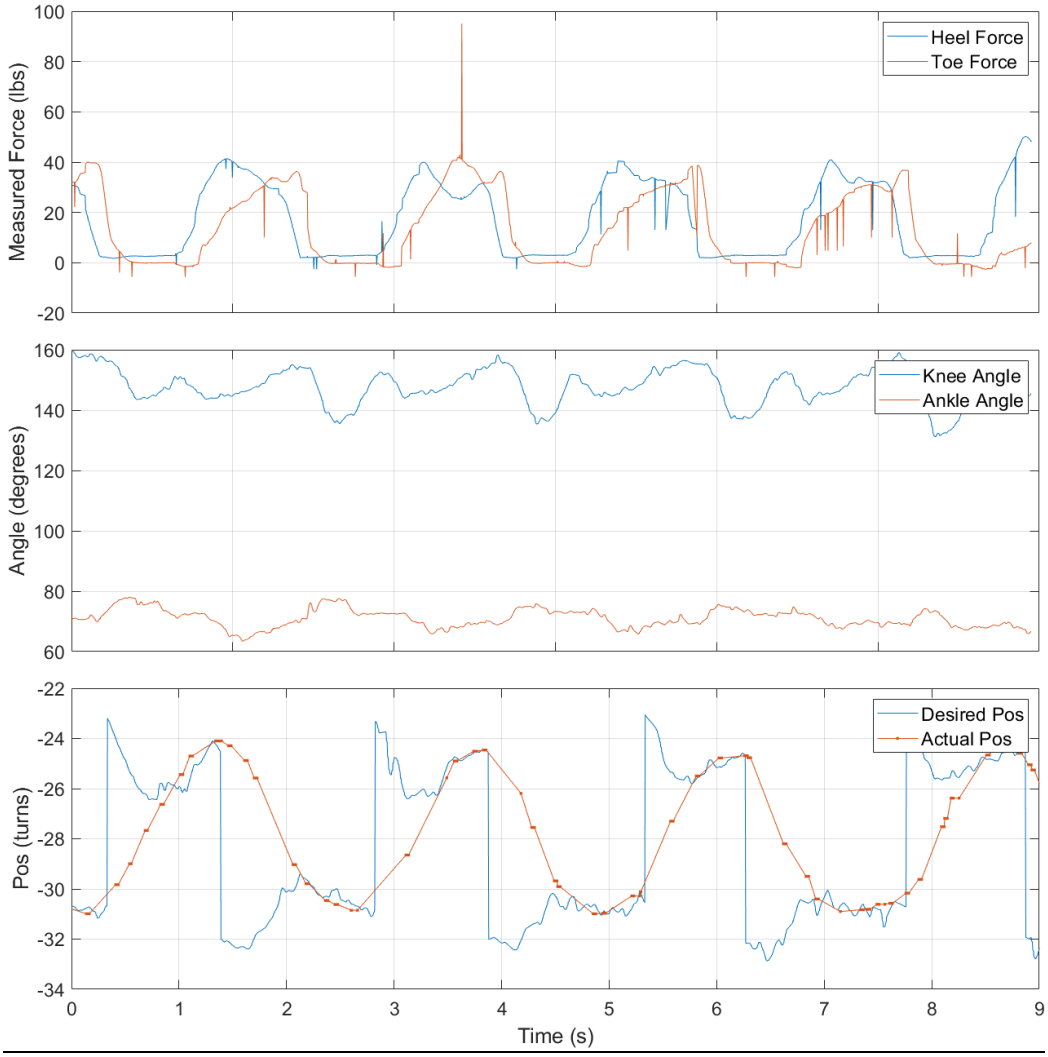

Figure 4.15: Walking test with 15 mm of set compression. As stated before, the actual position was offset by 600 ms. The data points are also shown on the actual position to show the low data rate we receive from the ODRIVE. The acceleration and deceleration limits for the motor were 30 turns/s2 and the velocity limit was 125 turns/s.

4.4.2 Support Force Assumption Analysis

Now that we had walking data with full actuation, the assumption that a constant actuator force well approximates the true necessary actuator force to provide a constant support force could be tested. First, we can estimate the support force using equations 3.1-3.13 from the support force calculations. We can use the average knee and ankle angles during stance from our data, approximately 149 and 69 degrees respectively. With these angles, we find that the expected support force is about 101.7 N (23 lbs) given the spring is providing 220.7 N (50 lbs) of force. If we take the more realistic expected support force of 183.9 N (41 lbs), we get approximately 84.7 N (19 lbs) of support force. For the sake of testing the assumption, however, we will use the 220.7 N of actuator force and 101.7 N of support force. Figure 4.16 shows the required actuator force to perfectly provide the 101.7 N of support force during stance compared to the constant actuator force we provided. The constant actuator force method we used seems to very closely match the ideal actuator force to provide a constant support force. The ideal force did change over the course of the stance, starting slightly lower and growing for about the first half of stance, and then usually staying relatively constant for the second half. However, the ideal weight only varied by at most about 14 N, which is small relative to the magnitude of the applied force. Because of this, we can conclude that a constant actuator force well approximates the ideal actuator force.

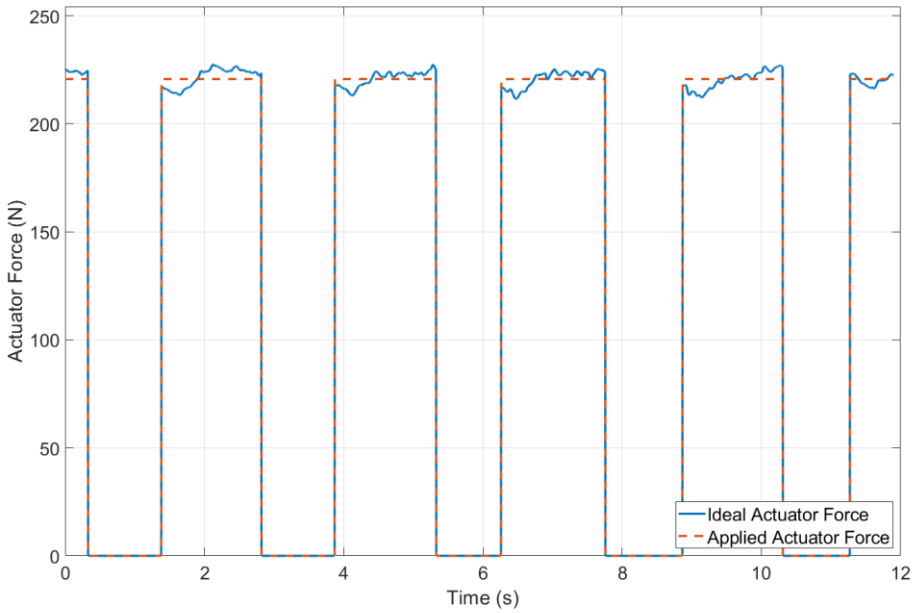

Figure 4.16: The ideal actuator force compared to the constant actuator force method we used. This constant force is equal to the force that would be generated by our set compression. With the measurement error when compressing the spring and with the slow motor reaction, the actual actuator force would differ greatly. We do not want to provide a supporting force during swing, which is why the actuator force drops to 0 in the plot.

The resulting support force from the constant actuator force can also be investigated. It can be seen in Figure 4.17. Just like with actuator force, the variations in the applied support force, about 6 N, are small compared to the magnitude of the force. The small level of variation means the support force from a constant actuator force does not vary greatly from the ideal constant support force.

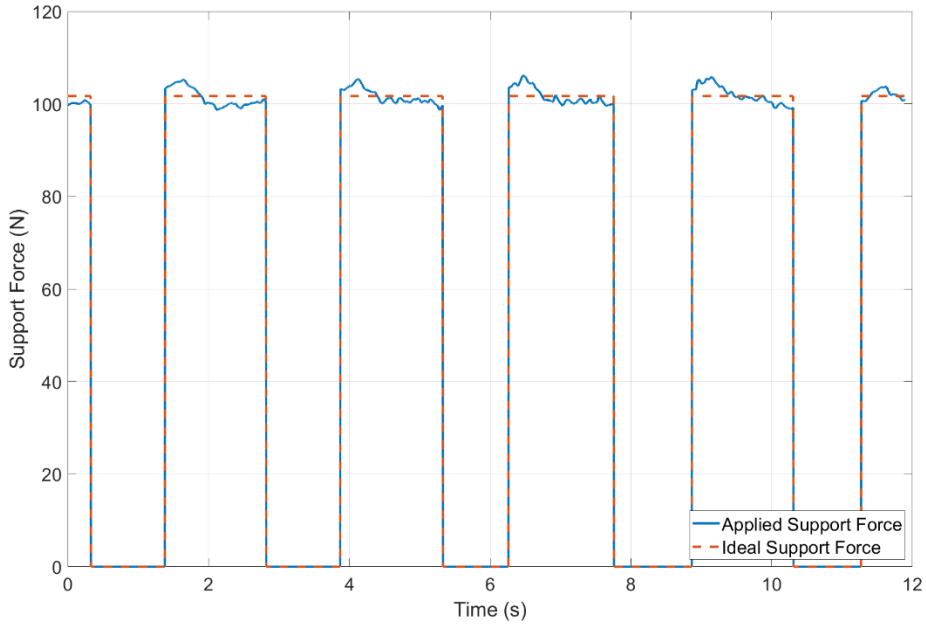

Figure 4.17: The support force applied by a constant actuator force as compared to the ideal constant support force. Just like in the actuator force plot, the support force goes to 0 while in swing state.

4.4.3 Actuated Walking Tests with Improved Sensor Reading and Motor Acceleration

While we had shown that the exoskeleton design was functional, the errors in the force sensor readings and the slow actuator speed were still major limitations. To fix that, we added the I^2C extenders to the I^2C lines and ferrite rings to the motor wires. We also better lubricated the actuator rods to reduce the friction in the system, thus reducing the current. With these changes, we were able to eliminate the unreported errors in the force data (the small spikes seen in Figure 4.12) and almost entirely eliminate the reported errors. A plot of the force data with the reported errors can be seen in Figure 4.18. Comparing Figure 4.18 to Figure 4.12 shows that the I^2C extenders resulted in a massive decrease in errors due to the motor noise.

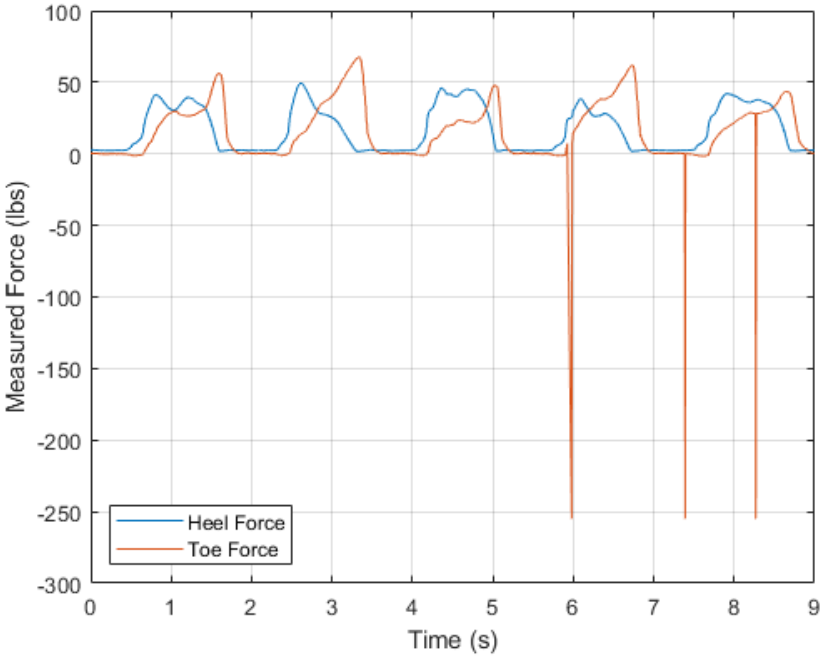

Figure 4.18: Raw force data read by the program after the I2C extenders were added. All spikes going down to -255 are errors that occurred when the status bit of the force sensor data reported that an error occurred.

We were also able to increase the motor acceleration and deceleration limits to 100 turns/s^2 from 30 turns/s^2. The velocity limit was also increased to 250 turns/s from 125 turns/s, but testing revealed that the acceleration was the limiting factor as the velocity never reached its limit. The new walking data from the series of tests with these improvements can be seen in Figure 4.19. With these changes, we were able to get consistent force data and were able to observe that the motor was reacting much quicker to heel strike and foot lift than before, applying the force at the correct time.

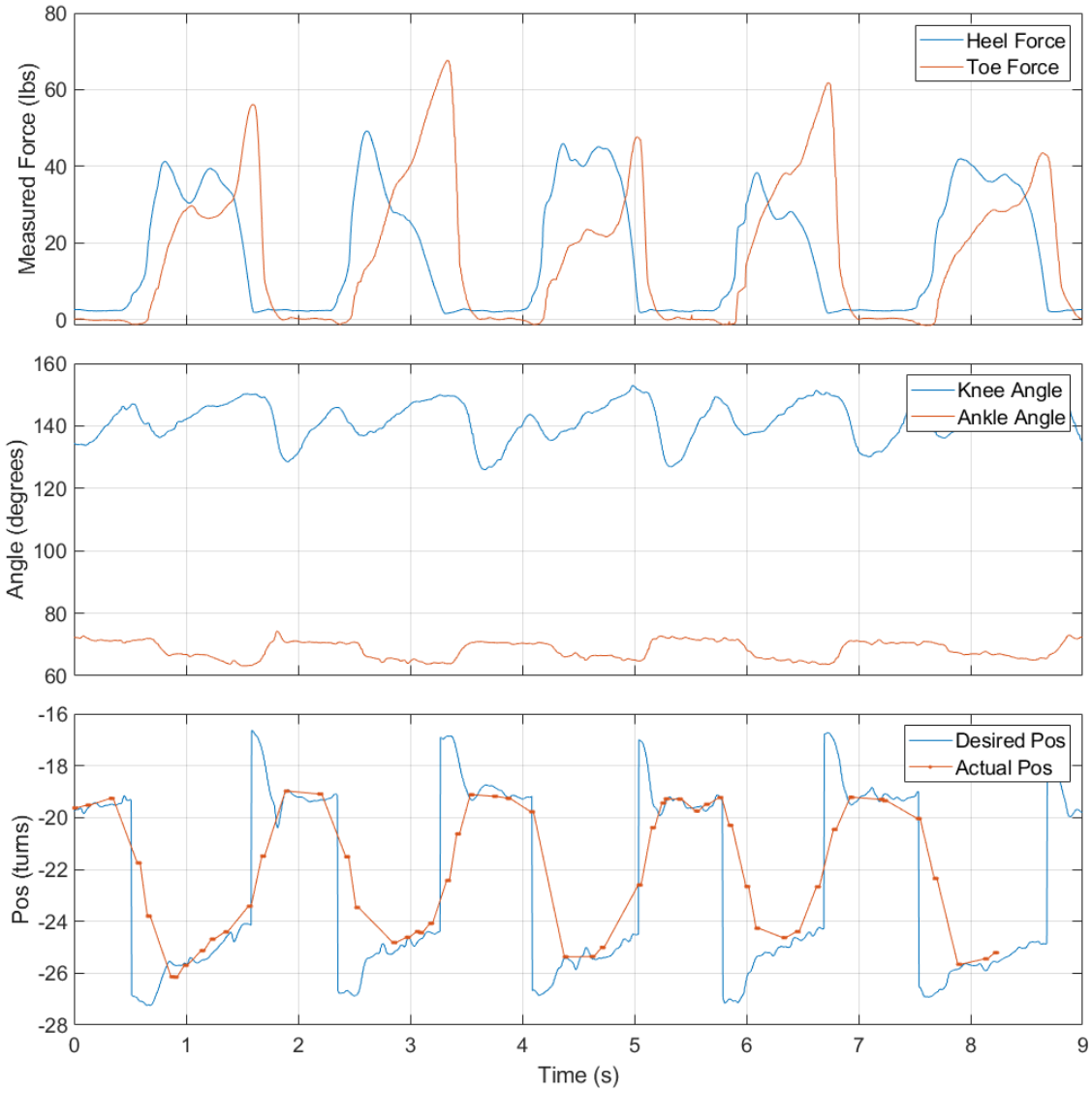

Figure 4.19: Walking data with 15 lbs of compression with the improved noise reduction in the I^2C lines and increased acceleration and deceleration values. The acceleration and deceleration limits for the motor were 100 turns/s^2 and the velocity limit was 250 turns/s. The actual position data has been offset by 600 ms to correct for the delay in reading the position from the motor.

Chapter 5

Conclusions and Future Work

5.2 Conclusion

A lower body underactuated exoskeleton was designed, manufactured, and tested. It was made to assist those with leg weaknesses that do not require a fully actuated exoskeleton. It assists the user by using a linear actuator to provide a supporting force on each leg while in stance. When a leg is in the swing stage of gait, the force from the linear actuator is disengaged, so as to not impede the user's natural movement. The linear actuator consists of a spring that can be compressed by a metal plate moved by a motor-driven ball screw. This means that the system can move the plate to engage the spring in the stance stage of gait and then completely back the plate off the spring in the swing stage. The exoskeleton's lower leg has a pantograph design that allows the actuator to effectively push between the ball of the foot and the heel. It was found that to provide a desired supporting force, an actuator force from the spring of approximately 2 to 2.2 times the support force is needed depending on the position of the legs. While the required actuator force for a constant support force varies over the course of the gait cycle, it was found to be well approximated as a constant. The exoskeleton was able to meet our range of motion requirements as to not limit the user's joint mobility. It was able to provide 84.7 N (19 lbs) of support force per leg in stance. While this showed that our design was functional, it was less than our desired support force of 133.4 N (30 lbs). This was mainly due to yet unresolved issues with the motor controller and friction in the actuator.

5.3 Future work

The most immediate future work will be directly reading the motor encoder values with the ESP32. This will allow us to get better data to quantitatively assess the motor control system's

response time when moving to compress or release the spring. Since testing the exoskeleton, we found out that our issues with exceeding 35 A of current are likely due to current limitations in our buck converters. So to fix the issue, we will need to change the electronics to remove the buck converter between the batteries and ODRIVE. If we still have current problems with the ODRIVE after implementing these changes, we might need to change to a power supply where we have more direct control of the motors. If we are able to run the motor at higher levels of current, we should be able to increase the provided support force and motor speed.

The I^2C noise issues have been mostly resolved in the force sensors, so we should be able to start testing programs that use the distance sensors to better define the user's current stage of the gait cycle. These could be used to predict when heel strike is about to occur, rather than having to wait for the heel force to increase. We could, for example, move the motor closer to the spring right before heel strike, improving its reaction time. We also might look into breaking up the stance stage into smaller stages of differing actuator forces to provide a support force more proportional to the current bodyweight going through the foot.

More major changes could be done if a second version of the exoskeleton is developed, mainly redesigning the actuator to make the motor and spring forces aligned. This would reduce the moment on the guiding bars and spring rods in the actuator, thus significantly reducing friction. This should prevent jamming and decrease the current drawn by the motor when operating, especially when moving to compress the spring. This would allow us to provide greater spring forces and reduce the motor's power consumption.

To fully test the efficacy of the exoskeleton, more rigorous testing needs to be done. To evaluate how much the exoskeleton is assisting the user, metabolic rate testing should be conducted. This would consist of measuring the metabolic rate of users while walking with and without the exoskeleton. If our exoskeleton is actually helping the user, we should see a reduction in the metabolic rate when using the exoskeleton.

Appendix A

Required Actuator Force Calculations

The free-body diagram of the frame with the support and actuator forces can be seen in Figure 3.10. To find the relationship between the ground and actuator forces, we start by looking at the foot piece. The free-body diagram can be seen in Figure A.1.

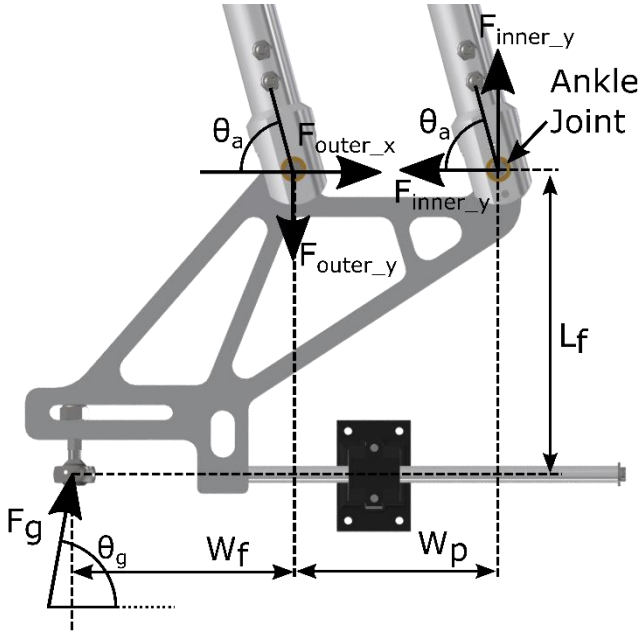

Figure A.1: Free-body diagram of the foot piece.

The outer pantograph frame is the bar that has the actuator lower mounting point, so as a first step we relate the ground support force to forces from the outer frame. To do this we start with taking the moment about the ankle joint:

$$\Sigma M = 0 = F_{outer_y} W_p + F_g cos\theta_g L_f - F_g sin\theta_g (W_p + W_f) \quad \text{(A.1)}$$

$$F_{outer_y} = F_g \frac{(W_p + W_f) sin\theta_g - L_f cos\theta_g}{W_p} \quad \text{(A.2)}$$

Now that we found the ground support force related to the y-direction force from the outer pantograph bar, we can move to the outer pantograph bar to relate it to the actuator force. The free-body diagram of the outer pantograph bar can be seen in Figure A.2: Free-body diagram of the outer pantograph.

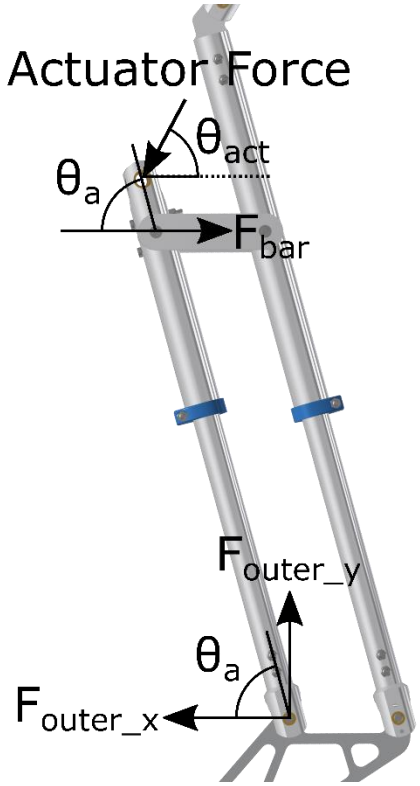

Figure A.2: Free-body diagram of the outer pantograph.

To relate F_{outer_y} and F_{act}, the sum of forces in the y-direction can be analyzed.

$$\Sigma F_x = F_{outer_y} - F_{act} sin\theta_{act} = 0 \tag{A.3}$$

$$F_{outer_y} = F_{act} sin\theta_{act} \tag{A.4}$$

Then the actuator force can be found in terms of the ground support force by combining Equations A.2 and A.4, resulting in the actuator force formula, A.6.

$$F_{act} sin\theta_{act} = F_g \frac{(W_p + W_f) sin\theta_g - L_f cos\theta_g}{W_p} \quad \text{(A.5)}$$

$$F_{act} = F_g \frac{(W_p + W_f) sin\theta_g - L_f cos\theta_g}{W_p sin\theta_{act}} \quad \text{(A.6)}$$

Appendix B

Motor Force and Friction on the Actuator Guide Rails

Based on the free-body diagram, the total frictional force, F_f, on the spring plate can be expressed as:

$$F_f = 4\mu F_b \tag{B.1}$$

The friction coefficient, μ, is the kinetic friction constant between the brass bushings and ceramic-coated aluminum rods. To find the frictional force, the moment about the center of the spring plate can be taken:

$$\Sigma M_C = 0 = F_m L_2 + F_s L_1 + \mu F_{fb}[L_1 + L_3] + \mu F_{fb}[L_1 + L_3 + D] + \mu F_{fb}[L_1 + L_3 + D + L_4] + \mu F_{fb}[L_1 + L_3 + D + L_4 + D] - 2F_b H \tag{B.2}$$

$$F_m L_2 + F_s L_1 + \mu F_{fb}[4L_1 + 4L_3 + 4D + 2L_4] - 2F_b H = 0 \tag{B.3}$$

$$F_b = \frac{F_m L_2 + F_s L_1}{2[H - \mu(2L_1 + 2L_3 + 2D + L_4)]} \tag{B.4}$$

Then we can combine equations B.1 and B.4 to get the frictional force equation:

$$F_f = 2\mu\left(\frac{F_m L_2 + F_s L_1}{H - \mu(2L_1 + 2L_3 + 2D + 1L_4)}\right) \tag{B.5}$$

To display the equation more simply we can set:

$$A = \frac{2\mu}{H - \mu[2L_1 + 2L_3 + 2D + L_4]} \tag{B.6}$$

Thus:

$$F_f = A(F_m L_2 + F_s L_1) \tag{B.7}$$

The sum of forces in the y-direction can then be investigated to find the motor force:

$$\Sigma F_y = m_s a = -F_m + F_s + F_f \tag{B.8}$$

$$\Sigma F_y = m_{sp} a = -F_m + F_s + A(F_m L_2 + F_s L_1) \tag{B.9}$$

It is important to note that a negative acceleration is defined as accelerating towards the spring, and a positive acceleration as away. To find the motor force, we can simply solve equation B.6 for F_m:

$$m_{sp} a = -F_m + AF_m + F_s + AF_s \tag{B.10}$$

$$F_m = \frac{F_s(1+A) - m_{sp} a}{1-A} \tag{B.11}$$

Appendix C

Full Electrical Diagram

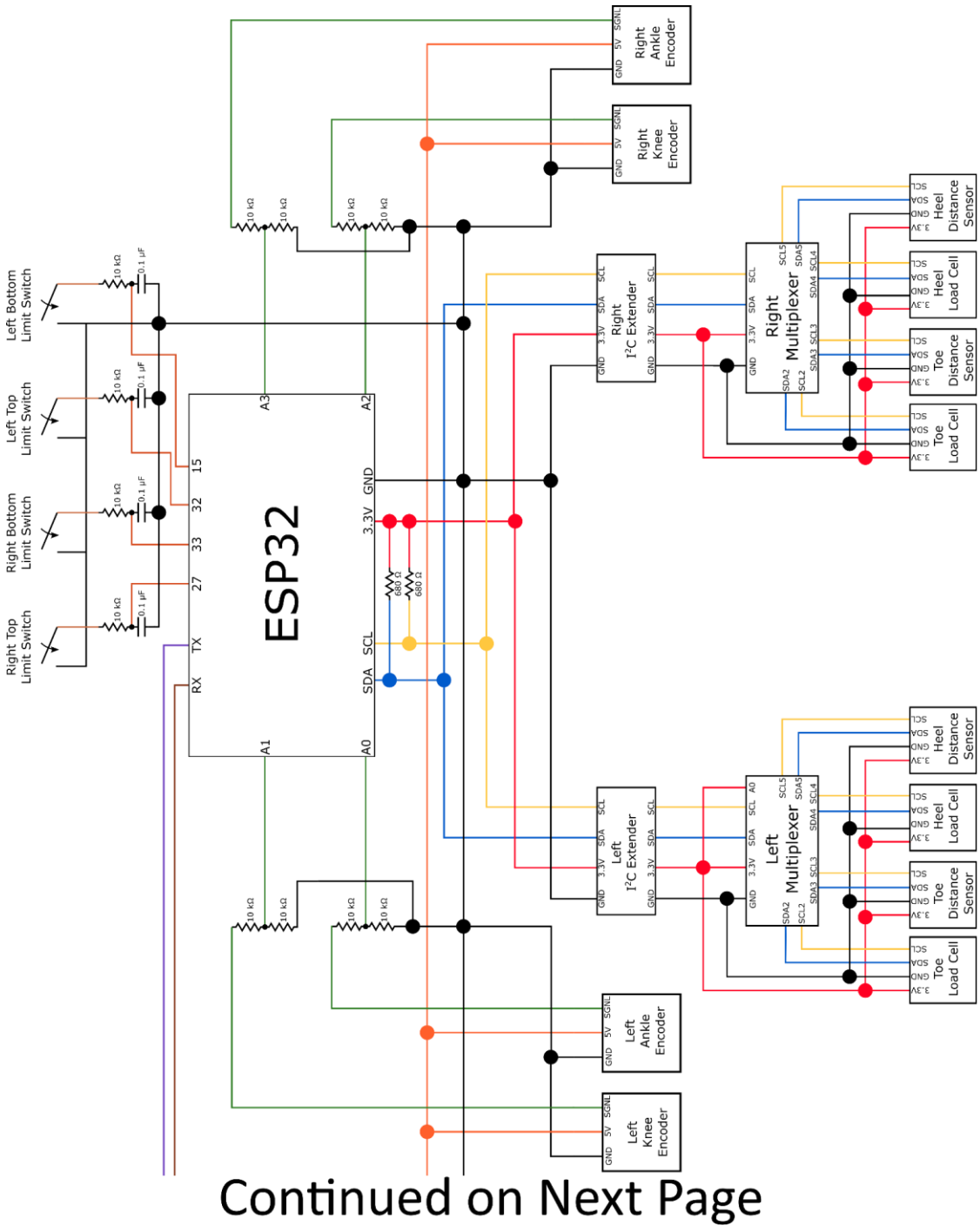

Figure C.1: Overall electrical diagram

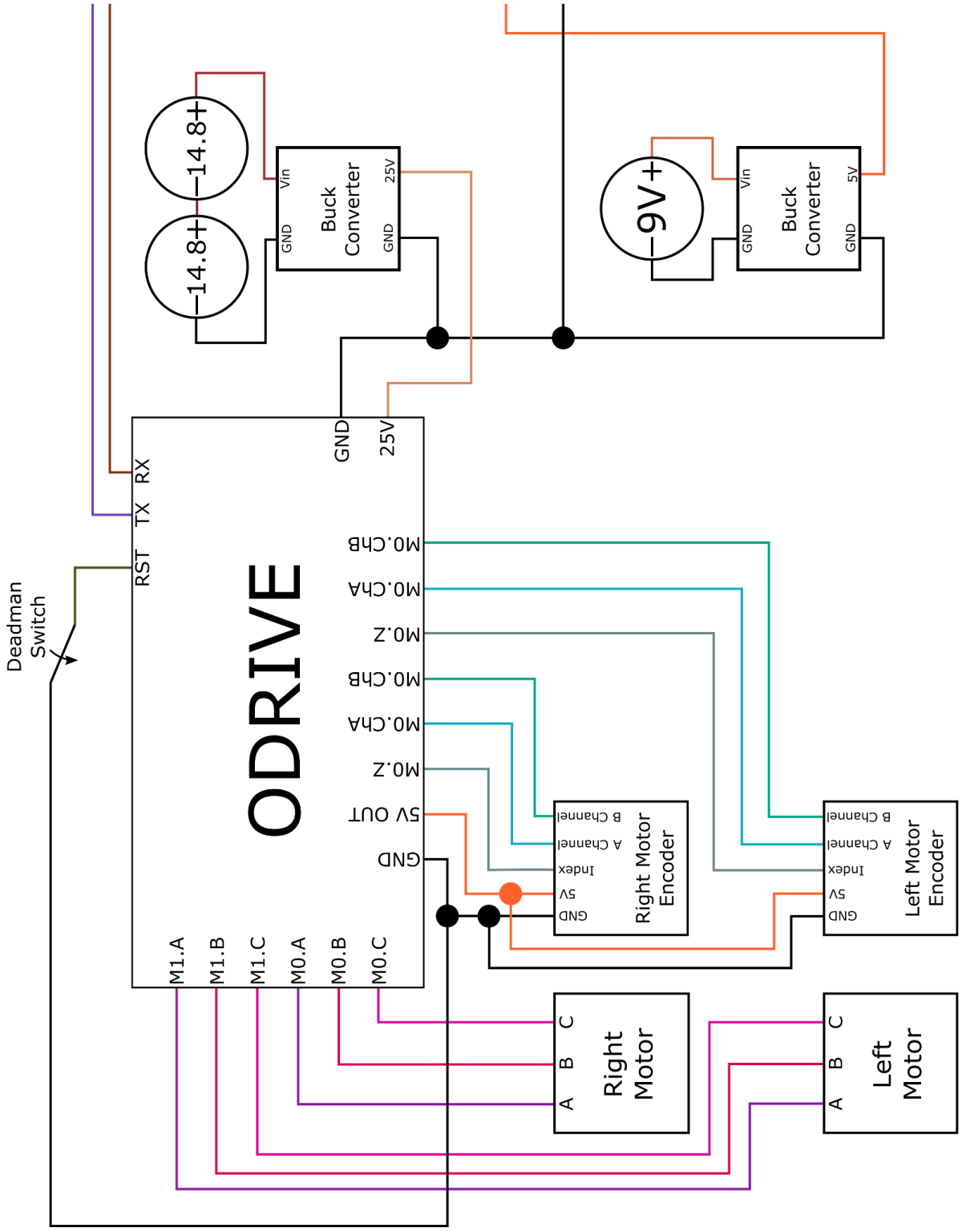

Figure C.2: Overall electrical diagram continued.

Bibliography

[1] "ExoH3," Exoskeleton Report, [Online]. Available: https://exoskeletonreport.com/product/exo-h3/. [Accessed 27 January 2022].

[2] S. Yang, "UCBerkelyNews," 03 March 2004. [Online]. Available: https://www.berkeley.edu/news/media/releases/2004/03/03_exo.shtml. [Accessed 27 January 2022].

[3] "Rex," Exoskeleton Report, [Online]. Available: https://exoskeletonreport.com/product/rex/. [Accessed 27 January 2022].

[4] Y. Sankai, "Leading Edge of Cybernics: Robot Suit HAL," in *SICE-ICASE International Joint Conference*, Busan, 2006.

[5] Y. Zefeng, N. Li, X. Long, H. Ren and X. Wu, "Bionic Mechanical Design and Stair Ascending/Descending Gait Planning of A Lower-limb Exoskeleton Robot," in *IEEE International Conference on Cyborg and Bionic Systems*, Shenzhen, 2018.

[6] "ReWalk," ReWalk Robotics, [Online]. Available: https://rewalk.com/rewalk-personal-3/. [Accessed 27 January 2022].

[7] "TWIICE," TWIICE, [Online]. Available: https://twiice.ch/exoskeletons/. [Accessed 27 January 2022].

[8] "Indego," Parker Hannifin Corporation, [Online]. Available: http://www.indego.com/indego/us/en/home. [Accessed 27 January 2022].

[9] H. Aguilar-Sierra, W. Yu, S. Sergio and R. López, "Design and control of hybrid actuation lower limb exoskeleton," *Advance in Mechanical Engineering*, vol. 7, 2015.

[10] M. B. Wiggin, G. S. Sawicki and S. H. Collins, "An exoskeleton using controlled energy storage and release to aid ankle propulsion," in *IEEE ... International Conference on Rehabilitation Robotics*, Zurich, 2011.

[11] J. A. Blaya and H. Herr, "Adaptive Control of a Variable-Impedance Ankle-Foot Orthosis to Assist Drop-Foot Gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 1, pp. 24-31, 2004.

[12] A. M. Dollar and H. Herr, "Design of a Quasi-Passive Knee Exoskeleton to Assist Running," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, 2008.

[13] J. E. Pratt, B. T. Krupp and C. J. Morse, "The RoboKnee: An Exoskeleton for Enhancing Strength and Endurance During," in *IEEE International Conference on Robotics & Automation*, New Orleans, 2004.

[14] "Keeogo," B-TEMIA INC, [Online]. Available: https://keeogo.com/?ref=steemhunt. [Accessed 27 January 2022].

[15] "ONYX," Lockheed Martin, [Online]. Available: https://www.lockheedmartin.com/en-us/products/exoskeleton-technologies/first-responder.html. [Accessed 27 January 2022].

[16] F. Giovacchini, F. Vannetti, M. Fantozzi, M. Cempini, M. Cortese, A. Parri, T. Yan, D. Lefeber and N. Vitiello, "A light-weight active orthosis for hip movement assistance," *Robotics and Autonomous Systems,* vol. 73, pp. 123-134, 2015.

[17] V. Firouzi, A. Davoodi, F. Bahrami and M. A. Sharbafi, "From a biological template model to gait assistance with an exosuit," *Bioinspiration & Biomimetics,* vol. 16, no. 6, p. 066024, 2021.

[18] T. Wang, Y. Zhu, T. Zheng, D. Sui, S. Zhao and J. Shao, "PALExo: A Parallel Actuated Lower Limb Exoskeleton for High-Load Carrying," *IEEE Access,* vol. 8, pp. 67250-67262, 2020.

[19] H. Hsu and R. M. Siwiec, Knee Osteoarthritis, Treasure Island (FL): StatPearls Publishing, 2021.

[20] M. J. Thomas, G. Peat, T. Rathod, M. Marshal, A. Moore, H. B. Menz and E. Roddy, "The epidemiology of symptomatic midfoot osteoarthritis in community-dwelling older adults: cross-sectional findings from the Clinical Assessment Study of the Foot," *Arthritis Research & Therapy,* vol. 17, no. 1, p. 178, 2015.

[21] T. Stöckel, R. Jacksteit, M. Behrens, R. Skripitz, R. Bader and A. Mau-Moeller, "The mental representation of the human gait in young and older adults," *Frontiers in Psychology,* vol. 6, 2015.

[22] "Basic sciences," Musculosketal Key, 26 December 2016. [Online]. Available: https://musculoskeletalkey.com/basic-sciences/. [Accessed 27 January 2022].

[23] A. Roaas and G. Andersson, "Normal Range of Motion of the Hip, Knee and Ankle Joints in Male Subjects, 30–40 Years of Age," *Acta Orthopaedica,* vol. 53, pp. 205-208, 1982.

[24] R. G. Budynas and J. K. Nisbett, "Screws, Fasteners, and the Design of Nonpermanent Joints," in *Shigley's Mechanical Engineering Design Ninth Edition*, New York, McGraw-Hill, 2011, p. 417.

[25] "Control Modes," ODRIVE Robotics, 2021. [Online]. Available: https://docs.odriverobotics.com/v/latest/control-modes.html#trajectory-control. [Accessed 22 February 2022].

[26] "Control Structure and Tuning," ODRIVE Robotics, 2021. [Online]. Available: https://docs.odriverobotics.com/v/latest/control.html. [Accessed 17 February 2022].