A Dynamic Robotic Quadruped

Senior Design Report

Written by Ben Dodson (bzd4) and Sumair Sawhney (sss323)

Under Advisement of Professor Douglas MacMartin

December 19, 2023

Table of Contents

Executive Summary

This section is intended to satisfy the requirements of the Senior Design Project format. The content of the design itself is contained within the Design Overview section.

1. Desired Functions

The goal of this project is to design and build a robotic quadruped capable of dynamic movement. Commonly referred to as ‘robot dogs’, robots in this category are able to perform basic locomotion, such as walking and running, as well as a variety of other tasks like jumping or hopping. Advanced models include multiple sensors that can collect environmental and conditional data. By the end of the semester, the aim was to have a quadruped that was capable of motion on a flat surface

2. Design Constraints

Many constraints were placed on this project. One of the greatest constraints was the manufacturability of the design. Due to the availability of certain machines and processes, it was very difficult to manufacture many of the parts. Another constraint was cost. Building complete robotic systems of any type can be extremely expensive due to extensive use of precision parts and high end components. Cost needed to be kept to a minimum in order to produce a feasible design. The last constraint was time as the semester provided a limited amount compared to the size of the undertaking.

3. Performance Objectives

The project is both a mechanical project, focusing on the design and construction of a mechanical system, and controls project, focusing on the software and electronics and motion planning for the robot. The goal for the mechanical aspect is to design and manufacture the structure of the quadruped to be able to not only support itself but also have enough strength to perform high torque maneuvers such as jumping. This would require around 20 Nm of torque in each of the leg actuators. The goal for the electrical aspect is to power the chosen motors, deliver the high instantaneous current needed for intensive maneuvers, and maintain safety throughout the system. The goal of the software aspect is to be able to control the quadruped’s movement at a fast rate while limited by the Raspberry Pi’s CPU performance. The Raspberry has a single core, 1.5 GHz processor which would be difficult to reach its limit but can be pushed with careless practices.

4. Alternative Designs

There are many forms of robotic quadrupeds. Different types of robotic quadruped were considered before selecting this design, such as delta-leg configuration or push rod legs. Additionally, a multitude of different designs were explored on a smaller scale at every step of the design process. Another design option considered was a two legged, two wheel quadruped as it would have reduced the time needed for manufacturing and reduced cost.

5. Analyses Methods

Analysis was done using several hand calculations, Ansys for mechanical parts, and significant research into the most prevalent designs and reasoning.

6. Industry or Society Standards

Some industry standards were used in the creation of this robot. Since robotics is still a maturing field, there are still relatively few industry standards for robotics, especially robotic quadrupeds. The industry standards used in this project are used internally, specifically within material properties and manufacturing techniques. Additionally, standards for controls for safety, organization, and execution were considered and followed when developing the electrical subsystem and developing the software stack.

7. Applied Coursework

MAE 2020 and MAE 3270 greatly influenced the mechanical design of this project. The concepts and skills from these two classes were used throughout the project.

MAE 2250 provided significant insight into design for manufacturing. Design for manufacturing was one of the most difficult challenges throughout this project.

MAE 3780 was valuable for its knowledge on electrical components and design. The knowledge gained from this portion of the course facilitated the construction of the entire electrical system. This also benefited in understanding controlling motors on autonomous robots.

MAE 4760 and MAE 4780 were very helpful for the robotic controls programming. The course introduced many robotic control methods that this design either implements or plans on implementing.

CS 1110, CS 2110, CS 3410, CS 3110 and others were very useful for programming the simulation and low level electrical control. These classes also provided useful experience in programming projects.

8. Design Evaluation

The mechanical design is almost entirely complete. There are only a few minor details that have not been entirely fleshed out; most of these details are awaiting further testing of existing components. The completed portions of the design have almost all been prototyped. The construction of the whole quadruped is still on-going, but the current testing and evaluation shows promising results. The actuator design has been shown to be successful and runs well. A single leg has been fully prototyped and is undergoing testing. The electrical system is fully functional. Finally, the software has been verified in simulation, and is undergoing additional updates.

9. Design Impact

Robotics can help save lives; Robots are able to perform complex tasks in dangerous environments that would otherwise endanger a person. Robotic quadrupeds are no exception to this. They have been increasingly used in workplaces and factories around the world for tasks such as monitoring, maintenance, and inspection.

Although this particular robotic quadruped design is not particularly superior to the existing quadruped designs, it attempts to make use of limited resources and capabilities in order to develop and build a fully functional robotic quadruped.

One noteworthy aspect of design impact is the concept of the universal actuator. Most robots from different companies use different actuators with different motors and gearboxes depending on their needs. The actuator designed for this quadruped has been designed specifically with the ability to quickly change gear ratio between different designs. This more universal design could theoretically allow for easier development and construction of robotic systems in the future.

10. Format

This project is inherently multi-disciplinary, so cannot take a singular format. The primary format for the mechanical design is the CAD design files. The manufacturing of this design also includes many CAM programs for CNC milling. The prototype of this design includes hundreds of mechanical parts, including a singular full leg assembly. The electrical system has taken the form of a fully developed and wired electrical system, capable of integrating the entire quadruped. Finally, the format of the software is the codebase that runs the simulation software and controls the simulated quadruped.

11. Roles

The project was split primarily between the mechanical and software aspects of the quadruped. Ben Dodson focused primarily on the mechanical portion of the project, while Sumair Sawhney concentrated on the software controls and simulation model. The electrical system was initially chosen by Ben based on mechanical requirements, but implemented and constructed by Sumair.

Design Overview

This section will overview the design of the quadruped, including the design decisions made, analysis of the design, and the resultant design.

Quadruped Morphology

The overarching goal of this project is to design and build a dynamic mid-sized robotic dog. The exact goals and desired functionality of this project have been loosely based around the existing mammal-type-quadrupedal robots (referred to henceforth simply as quadrupeds). An exploration of the robotic quadruped ecosystem is necessary in order to understand the goals outlined here.

Robotic quadrupeds can be broadly split into two categories: small, servo-based systems, and larger systems that use elastic actuators. The small servo-based designs are generally low cost and small enough to hold in one hand. These systems generally use hobby-grade servos for actuation. Hobby grade servos are not backdrivable under load which means that they cannot mimic biological quadrupedal movements well. Biological quadrupeds' joints are inherently backdrivable, that is, that the output of the system can drive the input to it. The ability to model this behavior is critical in the construction of a robotic system that can mimic the dynamic behavior and movement of a biological system.

Larger robotic quadrupeds can make use of elastic actuators, which are either backdrivable or mimic backdrivability through the use of sensors or compliant mechanisms. For the purposes of this project, I have only considered the larger robotic systems that can use elastic actuators and can more closely mimic actual quadrupedal movements.

A few notable examples of larger quadrupedal robotic systems are: Spot from Boston Dynamics, ANYmal from ANYbotics, the MIT Mini Cheetah, and the Go1 from Unitree. There are other systems on the market or used in research, but these are generally representative of the robotic quadruped ecosystem at the time of research. Spot and ANYmal occupy a larger weight class compared to the smaller Mini Cheetah and Go1. The Mini Cheetah and Go1 are approximately the same size and scale as a small dog at about 2 ft long[1], while Spot and

ANYmal are about 3.5 ft long[2]. The smaller robot dogs are much cheaper than the larger class and designed for dynamic movements, while the larger robots have a greater load capacity[3] and are designed for more commercial applications or intensive research.

Since the focus of this project is on dynamic movement, as opposed to load capacity or commercial readiness, the advantage of size was not a consideration in design selection. In addition to not needing the greater size of a large quadruped, it proved extremely difficult to research the components for these larger robots since the exact specifications of many industrial quadrupedal robots are not public information. Fortunately, some of the critical system characteristics for the Go1 and especially the MIT Mini Cheetah are publicly available. Thus, the majority of this project has been designed to fit within the weight class and structure of the MIT Mini Cheetah and the Unitree Go1. Using these two robots and others of similar size and weight can help more easily define specific system requirements without re-doing the years worth of research that have gone into the development of them.

Figure 1.1: MIT Mini Cheetah[4]

Figure 1.2: Unitree Go1[5]

The final overarching design decision for the quadruped is the layout of actuators for each leg. In three dimensions, there are six total degrees of freedom, three being spatial, and three rotational. In the case of robotic quadrupeds, the foot can be simplified to a ball, or simple rolling contact with the ground. Although this simplification removes the advantages of a realistic mammalian foot, such as the ability to increase stability and add contact area with the ground, it vastly simplifies the mechanical design. By treating the foot as a single ball with a single contact point, the three rotational degrees of freedom are no longer necessary for leg positioning. This leaves three spatial coordinates for consideration, which only requires three actuators. If the complexities of a foot had been included, the rotational dimensions would have necessitated additional actuators and mechanical complexities.

There are multiple methods of arranging the actuators in a leg in order to achieve these three spatial coordinates. The two most common types are the delta configuration and the series

articulated configuration. Both configurations serve a similar purpose. One actuator acts as an abductor, rotating the other two actuators. These other two actuators can then move to position the foot anywhere on the plane defined by the abductor. A lot of early quadruped research utilized the delta configuration because it allowed for a low moment of inertia while still providing the required range of motion. Recently, robotic quadruped design has focused more on series articulation, which is more similar to a biological quadruped.

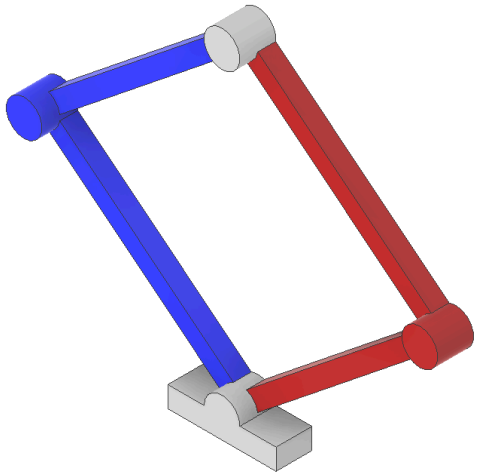


Figure 2.1: Delta Leg Configuration, Red and Blue Actuated Independently

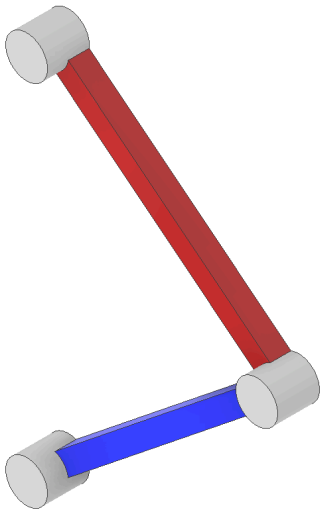


Figure 2.2: Series Leg Configuration, Red and Blue Actuated Independently

The design of series articulated quadrupeds is slightly more complicated due to the aforementioned issue of moment of inertia. However, as explained later in the Body and Leg Assembly section, these issues can be resolved fairly easily without any notable impact on functionality. One of the advantages of using series articulation is that it allows for a more of a biomimetic approach to control since biological quadrupeds basically use series articulation. Considering the approachability and control advantages of a series articulated leg configuration, this quadruped will make use of a series articulated leg.

Within reason, a robotic quadruped can rearrange the configuration of its actuators to change the linkage lengths, but the performance of the actuators themselves cannot be changed without the change of the actuator. For this reason, the single most critical part of quadrupedal design is the actuator assembly. Thus, the requirements on the actuators are the most strenuously researched and defined aspects of the overall design, and the focus of the following section.

Additional requirements mostly derive from the theoretical capabilities of the actuators, which define the loading on the remainder of the system. The rest of the robot is simply designed to make efficient use of the actuator capabilities at a low mass and low cost.

One of the critical high-level goals of this project is to develop a dynamic quadruped. The goal of dynamic actuation can best be thought of as the ability to respond rapidly to motion. In order to achieve this, one modern development in robotics has been the development and use of elastic actuators, “Intrinsically elastic joints have become increasingly popular over the last years. Commonly, they are considered to outperform rigid actuation in terms of peak dynamics, robustness, and energy efficiency”[6]. Elastic actuators can be juxtaposed with rigid actuators. Rigid actuators struggle to respond rapidly due to their low compliance and conformity, while elastic actuators can generally conform and adapt to their environment and loading conditions. Although dynamic movement and control is entirely possible using rigid actuators, they present many issues that elastic actuators can inherently avoid.

High rigidity or stiffness means susceptibility to high frequency noise or inputs, while elasticity allows for a mechanical filtering of noise and error. Compliant mechanisms are examples of elastic actuators. Compliance with the environment means that a mechanism can conform to the noise and high frequency inputs. Rigid, non-compliant mechanisms can achieve similar results, but only through the use of additional sensors, and control. Thus, in order to achieve dynamic control of the quadruped, it is easiest to use elastic actuators. A feature that is closely related to elastic actuation is impedance control. In the case of robotic design, impedance control can be thought of as a variable elastic actuator. It allows for the dynamic setting of elasticity. This is extremely useful for quadruped control since certain motions require different levels of elasticity. For example, a jumping motion requires a more rigid and rapid movement, necessitating a more rigid joint; while the motion of catching the quadruped when it is falling requires a lower rigidity in order to slow the fall. So, in addition to requiring elastic actuators, a goal for this quadruped is to also implement full impedance control.

In summary, the overarching goals for this project are to design and build a mid-sized dynamic quadrupedal robot. The quadruped will use series articulation for the legs, and make use of elastic actuators to allow for impedance control and dynamic movements. The majority of mechanical design focus will be on the development of a novel actuator design, while taking advantage of pre-existing research on robotic quadrupeds in order to expedite the development of the surrounding structures.

Actuators

The first area to explore in quadruped design will be the actuators. These are the core of the quadruped, and will define many of the required characteristics.

There are two main requirements for these actuators. The first requirement is the ability to use impedance control on each actuator. The second major requirement is on the torque output of the actuator. There is an additional requirement on the max rotational speed of the actuator, but as long as the speed is high enough to complete most dynamic movements, the speed is not a limiting factor. Mechanically, the actuator needs to be able to handle all loads imparted on it;

geometrically, it should be as small and lightweight as possible. One final important goal is to maximize the density of the actuator and make efficient use of space. Specific electrical requirements for the actuator are derived later.

Impedance control is fundamentally the ability to treat a joint, or any other movement, as a virtual mass-spring-damper. In the context of actuator design, this can be achieved either mechanically or electronically. Mechanically, a joint can have an adjustable spring and a damper implemented inside of it. Then the actuator could control the spring constant and the damping ratio of the output. This is very difficult to accomplish because it either relies on a physically adjustable spring and damper inside of a joint or setting the spring and damping constants without the ability to dynamically change them, and losing full impedance control.

Instead, a more effective way to accomplish this effect is to implement torque and positional feedback on the actuator. By using torque and positional feedback, the actuator can approximate a spring mass damper by using a programmed elasticity and a fast control loop (absolute minimum 1 kHz) to approximate the dynamics of a spring-mass-damper system.

Positional feedback can be done through the use of a variety of encoders or rotational sensors. A few options are: optical encoders, magnetic encoders, hall effect sensors, and potentiometers. The most attractive option for this actuator is the magnetic encoder. Magnetic encoders are relatively inexpensive absolute encoders with precise measurements and no contacting or wearing parts.

Torque feedback can be accomplished through two main avenues, a direct torque sensor or torque estimation. One method of accomplishing torque estimation is to use a brushless DC motor and a Field Oriented Controller (FOC). FOCs enable torque feedback through an estimation process based on the current passing through the motor. The relationship between motor current and torque is approximately linear (before saturation), which means that the torque estimation for any given motor is relatively accurate across most of the torque-speed curve.

Based on the need for torque feedback, I decided to use a brushless DC motor for the actuator. Assessing the need for a brushless motor controller and the desire for a magnetic encoder, there were only a few possible options for controlling the motor electrically. For these reasons, the Moteus R4.11 controller was selected for use with this actuator. The section on electrical selection and the MjBots ecosystem will discuss the selection of this particular controller in further details.

Backdrivability Requirement

Backdrivability is a necessity in order to achieve impedance control of the robot. Without a backdrivable actuator, the robot cannot respond to its environment elastically. Without backdrivability, the actuator is rigid and all of the work for converting a joint into a virtual mass-spring-damper system is useless. Thus, in addition to the aforementioned requirements on control, backdrivability is also a necessity. Backdrivability is quantifiably defined as the static

torque placed on the output that is required to move the input to an actuator. There are two ways for an actuator to exhibit backdrivability.

An actuator can be backdrivable mechanically through the use of low mechanical friction. As an example, biological joints are typically mechanically backdrivable. Mechanically backdrivable styles of joint allow elastic control of a joint without any torque sensors on the actuator output. The limitation of mechanically backdrivable systems is that they can be prone to wear, friction from dust and dirt, and can require frequent lubrication and maintenance.

Besides mechanical backdrivability, there is a way to mimic backdrivability through the use of torque sensors. This method senses the torque applied to the actuator output and then calculates the position that the actuator should move to based on a programmed joint elasticity. The advantage of this design is that it allows for the use of non-compliant mechanical systems that are not mechanically backdrivable. The disadvantage of this method is that it requires extremely rapid and accurate feedback between the torque sensor and the input motor. This method is limited by the update speed of the torque sensor and the controller; any dynamics that induce very high speeds, such as impact loads, will shock load the actuator if the electrical control is not fast enough. In addition to the problems posed by the implementation of this method, accurate torque sensors can be expensive and somewhat delicate. This method could be useful for backdrivable actuators that require low speeds and high torques. Higher torque actuators usually make use of higher gear ratios which decrease mechanical backdrivability, while lower speeds can make computing position based on elasticity easier.

Due to the challenges surrounding the use of a non-mechanically backdrivable actuator, I have decided to build around the concept of a mechanically backdrivable actuator.

Torque Requirement

It is relatively difficult to derive the exact torque needed for the actuator due to the following tautological requirement problem: a simulation of the quadruped is needed in order to find how much torque is required to make certain dynamic movements, while an output torque is needed in order to generate such a simulation. Although this workflow is possible, it would necessitate a simulation of a quadruped with an attempt at a walking gait before the design or construction of the quadruped itself. Once the simulation provides an estimate, design work must be done in order to determine if the simulated quadruped is even physically possible.

In the absence of any physical data or research, a productive workflow to determine a torque requirement could be to estimate the torque required in parallel with designing an actuator to test how much torque is actually required. The result of such a workflow would iterate on the quadruped until a simulation of the dynamics and the design capabilities align with one another.

However, due to the time constraints of this project, the extraordinary difficulties in implementing a consistent and reliable walking gait and control schema, the general uncertainty of simulating the physical dynamics, and the looming sim to real problem, a more reasonable

approach for an initial estimate is to use pre-existing torque figures for existing robotic quadrupeds of similar size. The MIT Mini Cheetah has an actuator torque output of 17 Nm[21] and the Go1 has 23.7 Nm of stall torque [22].

Both the MIT Mini Cheetah and the Go1 are extremely capable systems. They are both able to perform highly dynamic motions and each individually exceed the goals set out in this project. Due to this, the stall torque requirement of the actuator can reasonably utilize a similar stall torque requirement. Thus, the selected torque requirement was set at 20 Nm for these actuators. This requirement was verified from a statics perspective as well. Assuming 20 Nm stall and using a similar leg length to the Mini Cheetah and Go1, at around 9", the quadruped is capable of momentarily lifting up to 70 lbs off the ground. This is unrealistic due to the extremely high current draw that this would require, but provides good insight that the 20 Nm requirement is an acceptable target. This torque requirement is relatively high, but remains an approachable goal.

The first note about this requirement is that this is a stall torque requirement. The motors in each actuator cannot be run at full stall torque for more than momentary loads. This is due to the fact that the torque scales approximately linearly with the current until saturation. Near stall torque, additional current in an armature will fail to produce additional torque, so the motor saturates. This extra current is bad for a number of reasons. The primary concern with high currents is overheating of the motor, which can irreparably damage the motor by burning out the coils. Another concern is the overheating and current limits of the electrical control circuitry. Finally, there is a concern with high current draws from a limited battery. Battery cells have a max instantaneous current draw, which can limit the performance of the actuators if the actuators near this limit. The instantaneous current draw is definitely an issue when considering that a full quadruped has a total of 12 actuators running simultaneously and each motor could momentarily draw 50 Amps.

Reviewing all of the concerns about running at stall, and that the actuator stall should not be the actual running condition of the actuator, it is necessary to validate that this requirement is a valid design requirement at all. Although there is some concern regarding this as a requirement, it is a valid requirement due to the relatively linear nature of the torque vs current graph until saturation. If we consider that the stall torque figure is the max torque of the motor just before motor saturation, then the stall torque is at the very limit of the linear regime. So, even though the actuators should never actually run at full stall torque, the linear nature of the torque production means that the stall torque requirement should impose a good requirement for the actuator's continuous torque.

Motor Selection

Reviewing all the previous limitations and performance requirements, the design process has led to the need for a backdrivable actuator with a brushless DC motor and a torque output of

20 Nm. Upon conclusion of the requirements, the next phase was selection of the actual actuator components and actuator design. The two most critical components of the actuator are the motor and the gearbox. These two components will determine the torque output, speed, and backdrivability of the actuator.

The backdrivability requirement is the first requirement that will limit the motor and gearbox. In order to use a gearbox and achieve mechanical backdrivability, Quasi-Direct Drive (QDD) actuators (transmission ratios < 1:10)[7] should be used. The low gear ratio allows for any torques applied to the output to transfer to the input, allowing for a mechanically backdrivable actuator. The use of gear ratios higher than 1:10 risks losing backdrivability. Thus, the max gear ratio that I was willing to use was 1:10. Maximizing this gear ratio would be helpful since motors are usually able to run at relatively high RPM, but low torque. So, the gear ratio for these actuators was selected at the 1:10 soft-limit.

Given a 1:10 gear ratio, the motor needs a stall torque of at least 2 Nm. This is a lot of torque for an inexpensive brushless motor to produce. One of the most common sources for brushless DC motors is the drone industry. They produce a significant number of low-cost decent quality brushless DC motors. However, most of these motors are designed for high RPM and low torque due to the need to spin propellers and rotors. After a laborious search for an inexpensive motor candidate, a large agricultural drone motor was finally chosen due to its high torque output and large size.

The motor is sold inexpensively from a Chinese manufacturer as an 8308 motor. The disambiguation of this motor suggests a stator width of 83mm and height of 8mm. Despite the labeling, the motor actually appears to be more like a 9208 motor. According to several people on a public forum that utilized this motor, the stall torque of this motor was approximately 2 Nm. So, this motor should be ideal for the robot quadruped actuators.

In order to maximize the torque output of this motor, the lowest Kv rating was selected. The Kv of a motor determines its velocity based on input voltage. The lower the Kv, the slower the motor will spin, but this will increase the torque of the motor at a constant power draw. The purchased motors were listed as 90 Kv 8308 motors.

During my testing, the torque output was observed to be greater than 2 Nm. The stall torque output was recorded at a full 2.75 Nm for this motor. Although all design calculations were done considering 20 Nm to be the maximum torque, the additional torque is beneficial because it means that the motor will require less current than originally calculated. Any additional torque is a welcome addition, so there is no issue with these findings. Furthermore, if the torque is too high, the electrical system can limit the torque output of the motor by limiting the phase current and preventing any mechanical damage.

Figure 3: 8308 Brushless DC Motor[8]

The final part of actuator design is the gearbox. This is by far the most complicated mechanical design element on the entire quadruped. There are multiple ways to achieve a backdrivable actuator. A few examples are: planetary gears, belts or chains, a capstan drive. Notably, the selected gearbox design is not typically considered a highly backdrivable gearbox type. Conventional wisdom on QDD actuators states that planetary gears are one of the best ways of achieving a low reduction ratio while remaining backdrivable. Despite this, the chosen gearbox was a cycloidal drive. The precise reasoning behind this choice will be explained in later sections. The gearbox will be explored in this next section so that the choice can be understood.

The final constraint to consider is the size constraint. The size of the actuator should be as small as possible in order to minimize the size and weight of the legs. Achieving a high torque density both volumetrically and mass-wise is the defining flexible goal for these actuators. The flexibility in this goal means that there is no exact volume or mass that these actuators need to aim for; however, it is also true that minimizing these quantities is of significant importance.

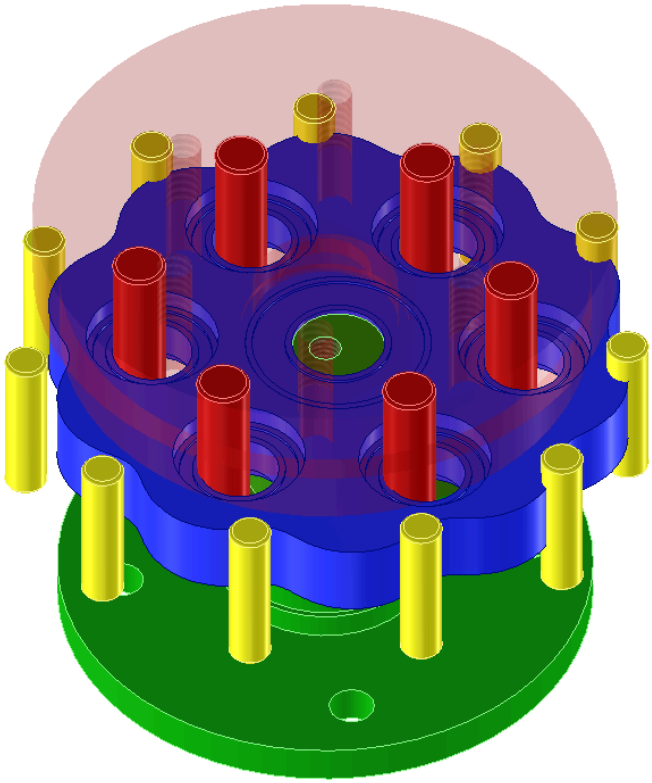
Based on the MIT Mini Cheetah, one of the best ways to achieve higher density is to make efficient use of the empty space inside of a motor. A motor consists of two parts: the stator and the rotor. On large enough motors, the stator is hollow, only needing to be constrained by a housing in order to prevent its rotation. This empty space allows for the possibility of adding a gearbox inside of the motor itself.

Cycloidal Gearbox

A cycloidal drive, or cycloidal gearbox, is a type of gearbox that provides a gear ratio between an input and output, generally increasing torque at the cost of rotational speed. Although the tradeoffs and intricacies of cycloidal gearbox design could itself consume the length of this report, this section will briefly overview the functioning principles of a cycloidal gearbox and then show the resultant design for brevity. Additionally, there are multiple configurations of cycloidal gearboxes, however, this overview will only cover the most common configuration.

A cycloidal drive can be considered as four parts: the eccentric input, the fixed pin ring, the cycloidal disk, and the output carrier. The eccentric input is the input to the gearbox. The input motor generally drives this shaft and it is upstream of the torque output. The cycloidal disk and fixed pin ring develop the gear ratio based on the ratio of fixed pins to lobes of the cycloidal disk. The output carrier, sometimes referred to as the output shaft, converts the eccentric motion of the cycloidal disk into axial-align rotation.

The focus of this work is not cycloidal gearboxes, so an analysis of the cycloidal gearbox will not dive into all of the specific choices for the parameters of the gearbox. The most important metrics are the gear ratio, the eccentricity, the fixed dowel pin diameter, and the fixed pin pattern diameter. These have been set to the following: gear ratio of 10:1, eccentricity of 1 mm, fixed dowel pin diameter of ⅛”, fixed pin diameter of 1.8”. The gear ratio was chosen at 10:1 for backdrivability. The eccentricity was set as high as possible while maintaining machinability in order to increase machining tolerances and lessen loads on the cycloidal disk itself. The fixed dowel pin diameter was set based on the cycloidal disk and machining capabilities, specifically tooling size. Finally, the fixed pin pattern diameter was maximized based on the geometric constraints of placing the gearbox inside of the motor. The resultant gearbox components are visible in the following figure.



Input Shaft: Green
Cycloidal Disk: Blue
Fixed Pin Ring: Yellow
Output: Red

Figure 4: Cycloidal Gearbox

The input shaft serves the purpose of converting the motor's rotation into the eccentric motion of the cycloidal disk. The input shaft connects to the rotor through four screws and converts the axial-aligned motion of the motor to eccentric motion with the eccentric portion of the shaft. This eccentric portion is where the cycloidal disk sits and allows for the proper function of the gearbox.

The input shaft is constrained by two bearings to prevent any unwanted moment loading. Additionally, this assembly ensures that the input shaft is tensioned against the inner race of the input bearing. This constrains the cycloidal disk assembly and prevents any vertical motion. The input shaft also features a magnet for a magnetic encoder.

The shape of the input shaft is mostly derived from machining needs. Due to the complications of multi-setup machining, it is easiest and most geometrically accurate to machine all critical features in a single setup.

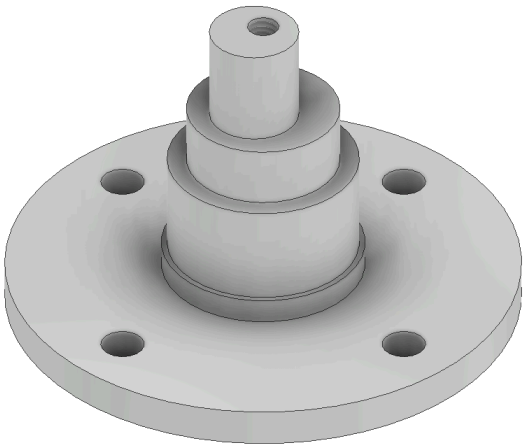


Figure 5.1: Input Shaft CAD

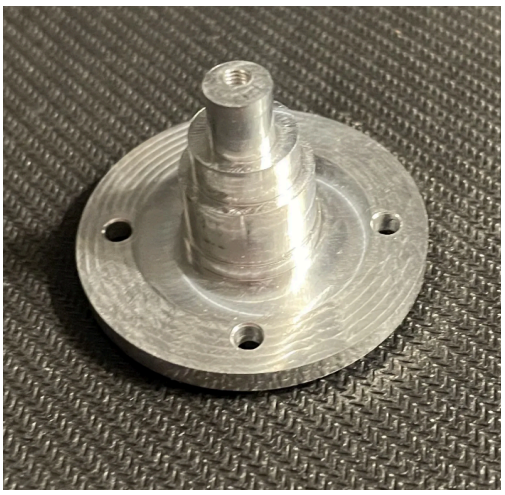


Figure 5.2: Input Shaft

Finite element analysis of the eccentric portion of the input shaft was conducted to verify that it would not fail during use. The setup uses the stall torque of the actuator. The results show that the input shaft should not fail under this loading condition. Moment loading was not considered an issue because two bearings are supporting the shaft on either side.

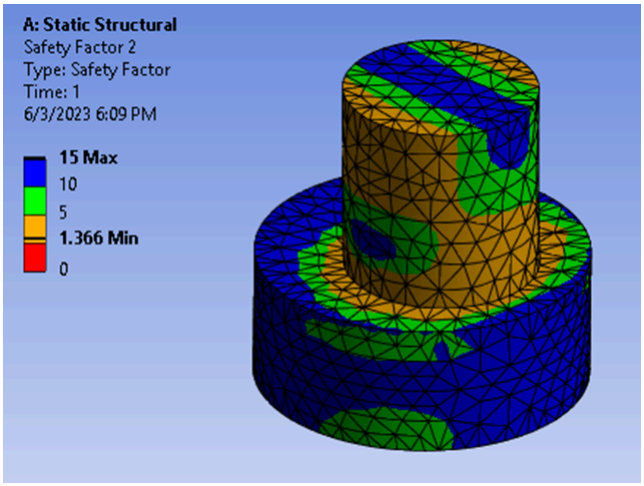


Figure 6: Input Shaft Analysis

The cycloidal disk is one of the most important parts of the entire actuator. Since the geometry of the disk is defined by the gearbox parameters, the primary concern during development was the material selection. One of the goals for this actuator is backdrivability. For a cycloidal gearbox, this means reducing the friction as much as possible. Cycloidal disks inherently deal with both sliding and rolling friction. The sliding friction is an especially important consideration because it limits the backdrivability of the actuator.

The material selection was dictated by a few key factors. The primary constraint was manufacturing. The cycloidal disk material needed to be a non-ferrous alloy or material that would also be strong enough. Many cycloidal disks are steel, but steel is a high hardness ferrous alloy that is not easily machined on the CNC available for this project.

The available non-ferrous metals are generally weaker, so material strength becomes a concern for the cycloidal disk. Material strength of the cycloidal disk can be broken down into two categories. The yield strength of the material and the hardness. Although these two properties are related, the yield strength more so determines if the overall disk will be damaged during use, while the hardness determines if the disk's surface will suffer damage when in repeated contact with the fixed pin ring. Multiple materials have sufficiently high yield strength to theoretically withstand the loading on the overall disk. However, there are very high loads applied to the edge of the disk that require a high surface hardness.

Anecdotally, the most commonly machined non-ferrous metal is aluminum. Aluminum alloys can have a decently high yield strength, but without hardening, the surface hardness is very low compared to the steel dowel pins. The steel dowel pins are 4140 steel with an expected hardness of 197 Brinell. Aluminum 6061-T6, one of the most commonly machined aluminum alloys, only has a hardness of 95 Brinell. If aluminum had been chosen for the disk, the steel dowel pins could have easily destroyed the disk. Aluminum has an additional problem that makes it entirely unsuitable for use as a cycloidal disk. Galling is an effect where materials like aluminum, titanium, and some stainless steels can chemically bond to similar metals. If aluminum had been chosen for the cycloidal disk, the difference in hardnesses would have caused the aluminum disk to wear and deposit onto the pins. This deposition could then gall with the disk and inflict further damage to the disk. In an extreme case, this could cause the disk to seize and become irreparably damaged; but, even in the absence of catastrophic failure, it is likely that this would significantly increase friction.

Generally, in gearboxes, gear hardnesses are either very closely matched or mismatched with sacrificial wear. In the common case of all steel gears, the similar hardness of all steel gears allows for little wear between the gears, but low friction. In the case of mismatched hardness, one gear is traditionally steel, while the other is usually a gear bronze. This can best be seen in the example of a steel worm gear and bronze wheel. This method actually does allow for wear between the gears. The high hardness of the worm gear will wear away the lower hardness bronze wheel. The bronze wheel will sacrificially wear, depositing some bronze into the surface of the steel. This wearing process will serve a dual purpose: it will grind the gears into a better mesh, and the deposition of the bronze into the steel will reduce the friction between the gears.

Figure 7: Hot Bronze Stock for Cycloidal Disk, I would recommend a metal-cutting bandsaw

The machining process and the need for specific material properties led to a relatively small pool of materials to choose from. One of the materials that presented itself as a solution to many of these problems was bronze. Specifically, bronze was an interesting material due to its good machinability and excellent material properties. Bronze does not suffer from galling, comes in numerous alloys of varying hardness, and is cost effective in low volumes.

Relatively high hardness bronze was selected due to the high contact loads that the cycloidal drive could theoretically impart to the disk. Any impact loading or rapid switching of actuator direction could impart dramatic peak loading to the disk. An in depth analysis of the surface loads was attempted, but the analysis was not sufficient to conclude anything beyond an expectation for high loads. The finite element analysis concerningly showed that small variations

in cycloidal disk geometry had high impacts on the expected loading. Due to machining inconsistencies, it was not possible to expect that analysis could properly predict the loading characteristics of the cycloidal disk. Although the term ‘high load’ is not quantitative enough to form accurate material expectations from, it shows that there is a need to use high hardness materials.

Despite the need for high hardness bronze, the hardness was not exactly matched to the dowel pins. If the hardness of the dowel pins was too similar to the bronze, the sacrificial wearing process would not have occurred as rapidly. This process allows for a reduction in friction as the cycloidal gearboxes are used more. Theoretically, this means that the gearboxes will get better over time and backdrivability will increase. Additionally, very high hardness bronzes can become exceedingly expensive. The selected C954 bronze is one of the higher hardness commercially available bronze selections at 179 Brinell hardness.

The cycloidal disk has an additional feature for maximizing backdrivability. On many cycloidal drives, the output pins are in constant sliding contact with the cycloidal disk. This is typically fine for cycloidal gearboxes that do not need to be backdrivable, but this gearbox is designed for backdrivability. The solution to this is to use bearings in the cycloidal disk to eliminate the sliding friction between the disk and the output pins.

In summary, the cycloidal disk makes use of several techniques in order to maximize backdrivability while still maintaining structural integrity and staying within the constraints of manufacturing capabilities. The 954 bronze has been selected specifically for this application. It makes use of material properties and extra ball bearings in order to create a highly backdrivable cycloidal drive. Finally, consistent lubrication is needed in order to maintain low friction and maximum backdrivability.

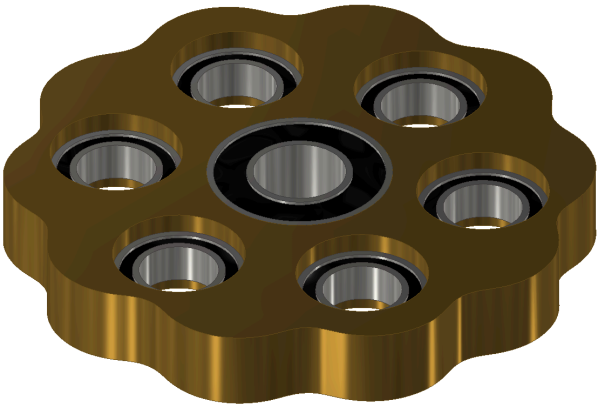


Figure 8.1: Cycloidal Disk Assembly CAD

Figure 8.2: Cycloidal Disk Assembly

The output carrier, or simply output, converts the eccentric motion of the cycloidal disk into axial-align rotation of the output. The output uses six output pins that fit into the six output

bearings on the cycloidal disk. The pins are retained with Loctite 638 retaining compound. The output also carries the output bearing and the constraining bearing. It also has an external mounting hole pattern for driving the quadruped's legs.

Figure 9.1: Output Carrier CAD

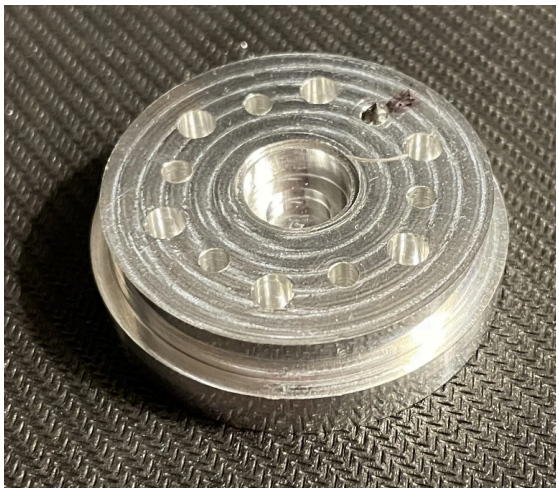


Figure 9.2: Output Carrier

The final machined part on the actuator is the housing. The actuator housing serves two main purposes. It primarily serves as the structural backbone of the actuator, mounting the motor's stator, offering mounting points, and protecting the internals. The housing also includes the geometry needed for the gearbox to be included inside the motor's stator. This includes locations for the input bearing, the output bearing, and the holes for the dowel pins.

The manufacturing of this part was by far one of the most tedious, loud, and difficult operations of the entire actuator. The most challenging part of manufacturing the housing is the multiple setups that it requires. The first setup machines the top face, including all of the critical gearbox-related geometry. The second setup flips the housing upside down to machine the rear. This operation must be concentric with the first setup in order to place the stator concentric with the input bearing. If the stator is not concentric with the input bearing, the rotor won't be either; subsequently, the motor will suffer significant performance losses. Each housing demanded multiple hours of machining time in order to complete both setups and verify that they were aligned correctly. Twelve housings were machined within 5 thousandths of an inch of concentricity.

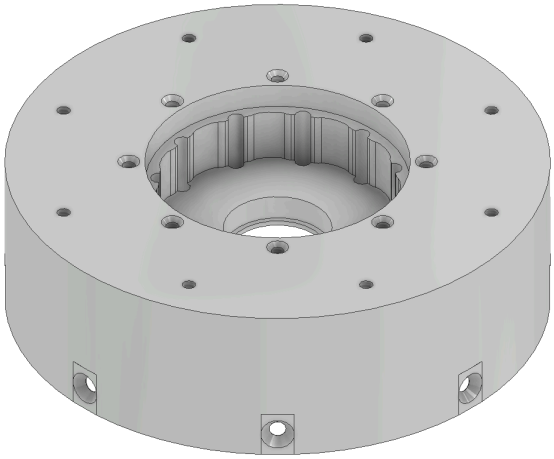


Figure 10.1: Actuator Housing CAD

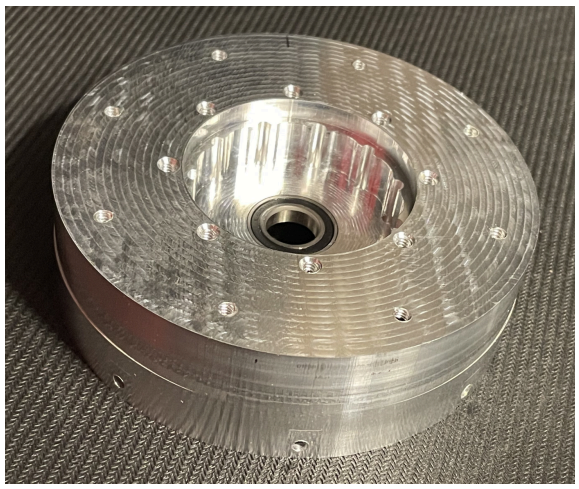


Figure 10.2: Actuator Housing

As a side note on structural analysis and verification, detailed structural analysis was only done on portions of the actuator that were considered high risk. These were mainly the gearbox parts, as all other parts had sufficient material so neglect detailed analysis. Some basic hand calculation analysis on these parts has been omitted because it is simple verification, and adds little to the content of this project. Although this method most likely left room for optimization in the mass of the actuator, the additional material was not detrimental to performance at all. In fact, all areas where mass could have been removed operate as a heat sink for the motor or motor controller.

This gearbox maximizes the space usage of this actuator. The space inside of the stator is almost entirely utilized in order to generate torque. The rotor is supported fully by the gearbox's bearings while simultaneously leaving the rotor top exposed for the integration of a magnetic encoder. The output carrier is fully supported by a crossed roller bearing and allows for easy attachment to the gearbox. The entire assembly consists of machinable parts, and allows for high enough tolerances that it can be capably machined using a hobby level CNC.

Bearing selection

Bearing selection and verification is an important part of the actuator design since bearings are needed in order to make sure that the actuator survives its loading and can rotate freely. This section will explore the specific bearing choices made in the development of this actuator.

The most critical bearing in the actuator is arguably the output bearing. It constrains the output from the cycloidal drive and accepts the loads placed upon it by the movement of the quadruped. Static loads like the weight or static belt tension are relatively low, while dynamic loads are the dominant factor in design consideration. Unfortunately, dynamic loads are

incredibly difficult to calculate by hand without knowing the exact software controller, field conditions, and many other factors ahead of time, so this design considers a high factor of safety necessary in order to prevent damage to the actuators. Thus, the worst case static loads have been used in order to define the loading conditions for the output bearing and a decent factor of safety has been required.

Most typical ball bearings are designed to handle radial loads. However, a quadruped's actuator can experience thrust and moment loading in addition to simple radial loads. Although a typical deep groove ball bearing can handle some thrust loading, even small moment loading could degrade the bearing rapidly. One possible solution to this is to assume that a stack of multiple radial bearings can accept a moment load as a force couple of radial loads. However, this solution relies on the assumption that machining tolerances are low enough that the loading can actually be distributed to these other bearings. This assumption was made in the design of the MIT Mini Cheetah[21]. The unitree Go1 also appears to take a similar approach to this since it is based on the MIT Mini Cheetah[20].

The reliance on low machining tolerances and very little play in ball bearings is not acceptable in this design for the following reasons: 1) the use of a cycloidal drive in place of a planetary means that the load pathing to the housing is less direct. 2) This robot was machined on a DIY hobby CNC with lower precision than an industrial grade CNC used for the production of the Mini Cheetah or the Go1. 3) A very small amount of radial play in the supporting bearings will prevent them from accepting the load and will instead put the main output bearing under a moment load, which could damage it. 4) There is no good way to test that the force-coupling assumption is valid due to the fact that it is extremely difficult to test the internal loads once the actuator is assembled. 5) variation in machining could lead to some actuators successfully accepting moment loads while others are prone to failure. The ratio of failed actuators to functional would only be known upon the completion and subsequent destruction of multiple actuators, which is not acceptable as this design is not being mass produced.

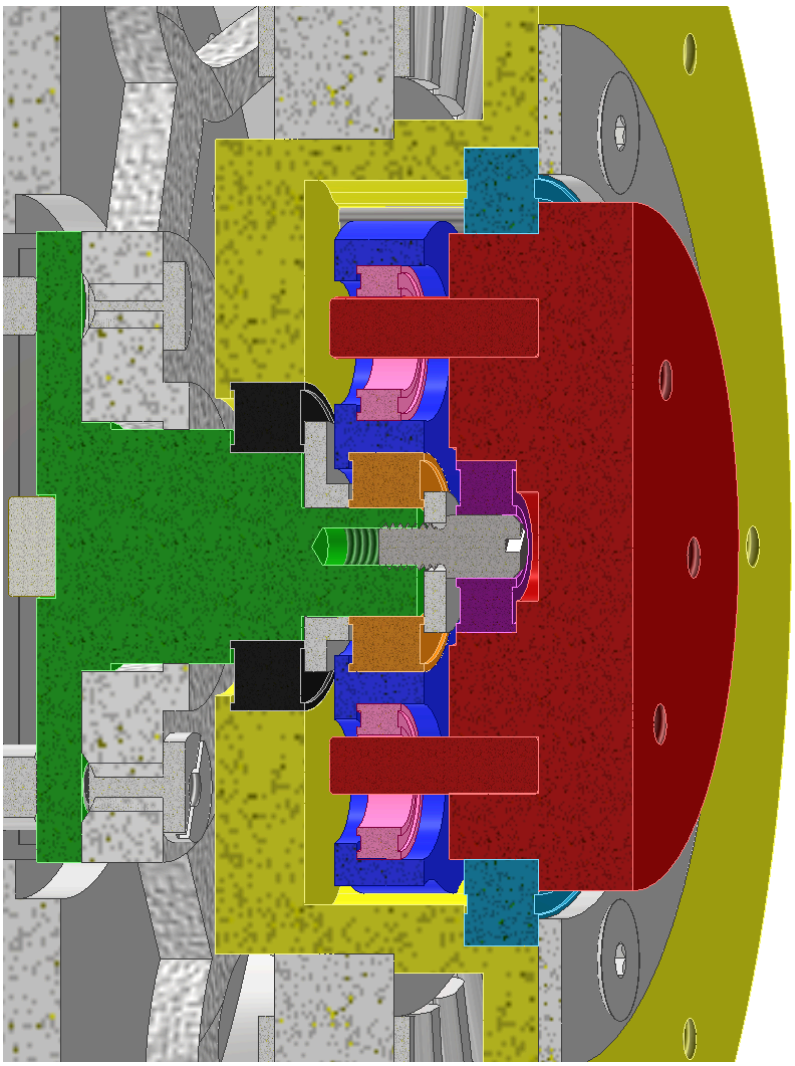
Due to these factors, a different method of accepting the moment loading is needed. The easiest way to accomplish this is to use a crossed roller bearing. A crossed roller bearing is able to accept radial, thrust, and moment loads in a single bearing. Presumably, the reason that the Mini Cheetah and the Go1 do not take a similar approach is due to the cost of real crossed roller bearings. However, knockoffs with similar properties to the originals are available from chinese manufacturers. This design selected a chinese copy of an IKO CRBT405 bearing. The loading characteristics of the original were obtained from Misumi[9], and suggest the bearing would survive all loading conditions. The analysis of this crossed roller bearing is based on Motion Control Tips[10] with 40 lbs max radial and axial loads, 20 Nm moment load and considering 100% shock loading factor.

Figure 11.1: CRBT405 Bearing[11]

Results	
Static life [%load]	0.499246
Dynamic Life [%load]	0.62905
Shock Life [%load]	0.998492

Table 11.2: Output Bearing Analysis Results

The resultant selection of this output bearing gives a very high factor of safety without any of the guesswork utilized on other designs as outlined above. This bearing should also effectively isolate the cycloidal drive from the loading conditions applied to the outside of the actuator since the loads will path directly from the output to the housing. This gives the additional benefit of greatly simplifying the loading conditions on the inside of the gearbox to only consider the effect of stall torque on the internal components.



Gearbox Parts
Input Shaft: Green
Cycloidal Disk: Blue
Housing: Yellow
Output: Red

Bearings
Output: Magenta
Constraining: Purple
Main Cycloidal: Orange
Input: Black
Cycloidal Output: Pink

Figure 12: Actuator Cross-section

Some underlying assumptions made during the bearing calcs for the internal bearings are as follows: The actuator is at stall torque in order to simulate the largest loading condition that the bearings will experience. There are no external loads applied to the internal bearings due to the use of the crossed roller bearing on the output. Statics approaches can be followed in order to determine loads, and a shock loading analysis of the bearings provides an estimate of the bearing lifespan.

The following static analysis was performed on the input shaft to determine the radial loads on the input bearing, cycloidal main bearing, and constraining bearing. It is relatively first-order, just to verify that these three bearings are not close to overloading:

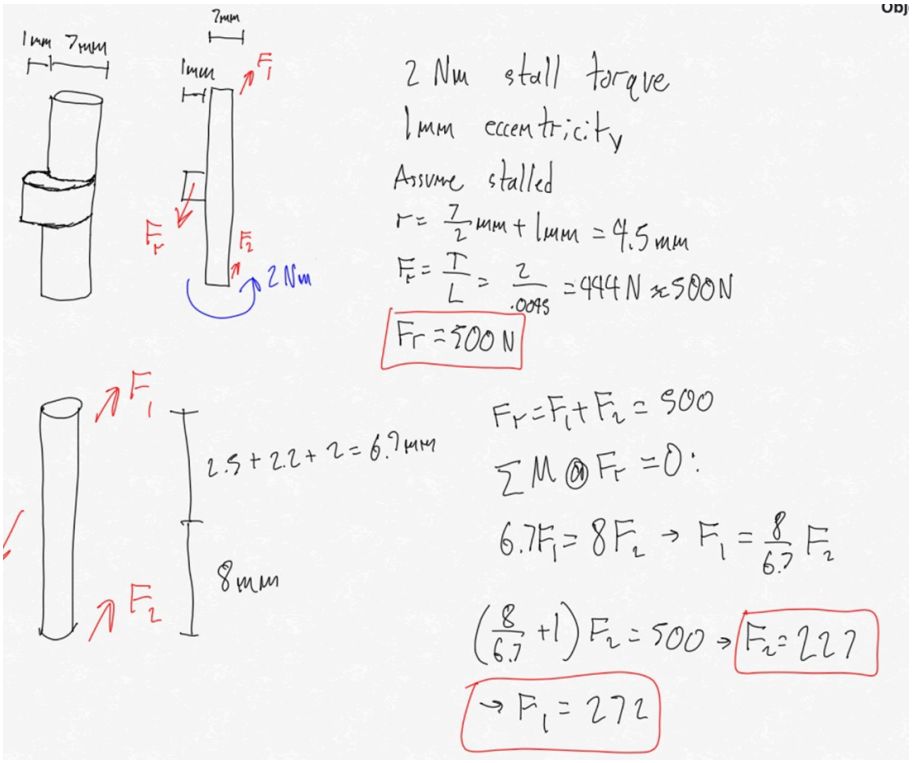


Figure 13: Statics Analysis of Input Shaft

The input bearing constrains the input to the cycloidal gearbox. It only experiences radial loads from the eccentric motion of the cycloidal disk. It has been sized so that it can be as flat as possible while still having a large enough internal diameter to accept the input shaft. It is also necessary to constrain the rotor concentrically to the stator.

Loading Condition:	Value	Obtained From
Radial [N]	227	Static Analysis
Axial [N]	0	None
Moment [Nm]	0	None
Shock Loading [%]	50	Guess

Bearing Parameters	Value
Type	Deep Groove
Static Rating [N]	1041
Dynamic Rating [N]	1915
Diameter [m]	0.021

Figure 14.1: Loading of Input, 6801 Bearing

Results	
Static life [%load]	0.21806
Dynamic Life [%load]	0.11854
Shock Life [%load]	0.17781

Figure 14.2: Input Bearing Analysis Results

The constraining bearing is necessary in order to prevent moment loading on the input bearing. It supports the input shaft from tilting which helps to constrain the input shaft to remain concentric under load.

Loading Condition:	Value	Obtained From
Radial [N]	272	Static Analysis
Axial [N]	0	None
Moment [Nm]	0	None
Shock Loading [%]	50	Guess

Bearing_Parameters	Value
Type	Deep Groove
Static Rating [N]	345
Dynamic Rating [N]	960
Diameter [m]	0.011

Figure 15.1: Loading of Constraining , 694 Bearing

Results	
Static life [%load]	0.78841
Dynamic Life [%load]	0.28333
Shock Life [%load]	0.425

Figure 15.2: Constraining Bearing Analysis Results

The cycloidal main bearing is eccentric with respect to the main axis of the motor and the input shaft. It allows the cycloidal disk to spin freely while accepting the radial loads from the cycloidal disk.

Loading Condition	Value	Obtained From
Radial [N]	500	See Below
Axial [N]	0	
Moment [Nm]	0	Assume all moment absorbed by top/bottom bearings
Shock Loading [%]	50	Guess

Bearing_Parameters	Value	
Type	Miniature Ball Bearing	
Static Rating [N]	800	* Not listed
Dynamic Rating [N]	1170	
Diameter [m]	0.014	

Figure 16.1: Loading of Cycloidal Main, 687 Bearing

Results	
Static life [%load]	0.625
Dynamic Life [%load]	0.42735
Shock Life [%load]	0.641026

Figure 16.2: Cycloidal Main Bearing Analysis Results

The cycloidal output bearings are designed to allow the cycloidal disk to transfer its eccentric rotational motion into axial-aligned concentric rotation with respect to the main axis of the actuator. Many cycloidal gearboxes do not include these bearings at all and simply have the output pins rubbing directly on the cycloidal disk itself. However, that introduces sliding friction to the system that reduces backdrivability. These bearings make the actuator more backdrivable so that it can be used better for the quadruped. It is difficult to analyze the load transfer of these bearings since the bearings will not all accept the loading equally. Ansys has been done to figure out the distribution of load across the six output bearings.

Setup: The OD of the cycloidal output bearings are fixed. The output pins are constrained to the output along the portion pressed into the gearbox output. The output has a 20Nm torque applied to it in order to simulate the actuator at stall torque.

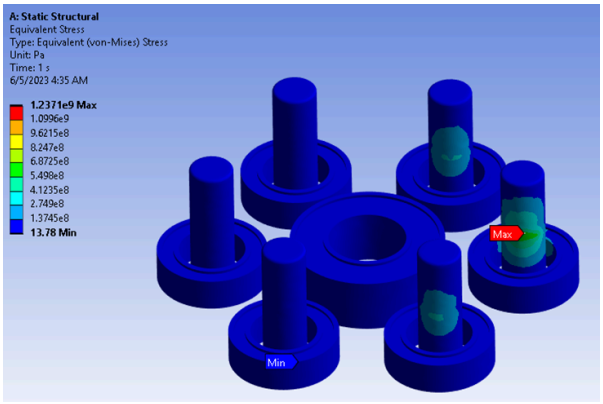


Figure 17.1: Ansys of Cycloidal Output

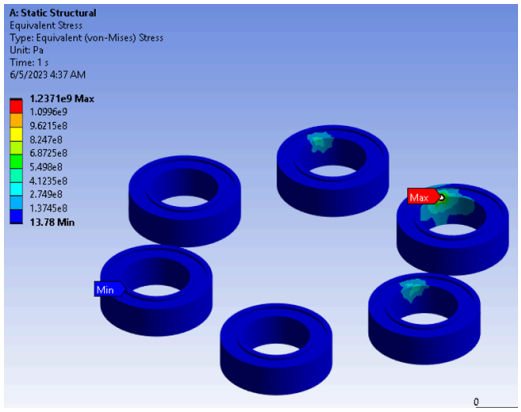


Figure 17.2: Ansys of Cycloidal Output showing only three bearings pathing load

It can be seen that only three of the output bearings are actually pathing any load, which makes sense since only half of the pins are in contact with the side of the bearing race with which they are moving towards. By looking at the force on each constraint, it was found the peak load for one bearing is about half of the total radial loading and that each other bearing accepts about a quarter of the radial load.

Bearing calculations will take the liberal assumption that the bearings are actually loaded equally. This assumption is from the fact that the Ansys assumes the bearings are completely rigid. In reality, the bearings have a small amount of radial play, which will allow the load to spread more evenly among the bearings that can accept loads. Since three of the six bearings are able to accept loads, it is reasonable to assume that the half of the cycloidal output bearings will evenly distribute the load among them. One note is that this orientation of the cycloidal disk is the most conservative since it distributes the loading across 3/6 bearings while other orientations will allow a 4th bearing to path loads.

Loading Condition:	Value	Obtained From
Radial [N]	476.1905	3 Pin Support
Axial [N]	0	Can't physically support axial loads
Moment [Nm]	0	Can't physically support moments
Shock Loading [%]	10	its always in contact, shock loads will just load the other sic

Bearing Parameters	Value
Type	Deep Groove
Static Rating [N]	218
Dynamic Rating [N]	495
Diameter [m]	0.01

Figure 18.1: Ansys of Cycloidal Output, MR106

Results	
Static life [%load]	2.18436
Dynamic Life [%load]	0.962001
Shock Life [%load]	1.058201

Figure 18.2: Cycloidal Output Bearing Analysis

These bearings are the only bearings in the assembly that show a factor of safety less than one. The factor of safety is only less than one for the static and shock loading cases. This is acceptable for this design because stall torque will be rarely used during operation, since the actuator should be run at continuous torque the majority of the time. Under typical use, the actuators should not be experiencing static loading since they will be used for dynamic

movements, which will load the bearings dynamically. Likewise, the shock loading is also an atypical loading since the control schema should not normally implement dramatic direction changes and shock load the actuator. Additionally, these bearings essentially serve to reduce friction. If they were to overload and fail, the result would be a decrease in backdrivability of the actuator. In the case of one bearing failing, it could be easily replaced. Furthermore, negative margins on factors of safety for bearings are acceptable because it can be calculated to simply reduce the bearing lifespan below the nominal lifespan. Overall, despite the immediate concern on these bearings, the actual effect of this concern is negligible for this design.

In summary, every bearing in the actuator has been selected and verified to a reasonable degree. Every bearing included in this actuator either has a factor of safety greater than one or a reasonable factor of safety and a valid line of reasoning for its acceptability.

Summary of Actuator Design

The culmination of this design process led to the final development of this actuator. It is a capable actuator, implementing a low gear ratio 10:1 cycloidal drive using a low cost brushless DC motor with a robust electrical control. It is capable of producing upwards of 20 Nm of torque at stall and thus far has exceeded the RPM needs of a quadrupedal robot. It makes no compromises on structural integrity, integrating a crossed roller bearing for load support, and sufficient factors of safety for all critical structural elements and bearings. Given these performance characteristics, the strongest feature of this actuator is actually the versatility and adaptability of this design.

There is a problem in robotic mechanism design. Almost every robot uses a different actuator with different communication protocols and different controls. This issue largely derives from the different needs of different machines. This makes the mechanical development of robotics far more difficult due to the lack of interchangeable parts. This actuator design presents itself as a solution to this issue.

The great advantage of this design is not its immediate function advantage over previous designs. When looking at this cycloidal gearbox compared to a planetary QDD, it is minimally backdrivable and could reduce overall performance of the robot. However, the real reason for exploring this design is the possibility of re-use. Planetary gearboxes in this size max out at around the 10:1 gear ratio. Any gear ratio larger than this begins to require gears too small or too large to be load bearing or manufacturable. Cycloidal drives do not have this issue. By simply changing the number of lobes/pins of the cycloidal drive, the gear ratio could be set upwards of 50:1. Although an actuator like this would almost certainly lose its backdrivability, the flexibility of this design would allow for very rapid development of different robotic designs.

This actuator was originally intended for use on a robotic quadruped; however, if they were required in the construction of a robotic arm with a high load capacity, the design could be easily modified to use a higher gear ratio and then manufactured. The only changes needed

would be the cycloidal housing, and the cycloidal disk. All other parts could be reused in their entirety. This opens the door to rapid prototyping of both low/medium load robotics with dynamic actuation, and high load applications. This project does not cover the additional use cases for this flexible design style at different gear ratios, but it would be easy and effective to implement for use in future projects.

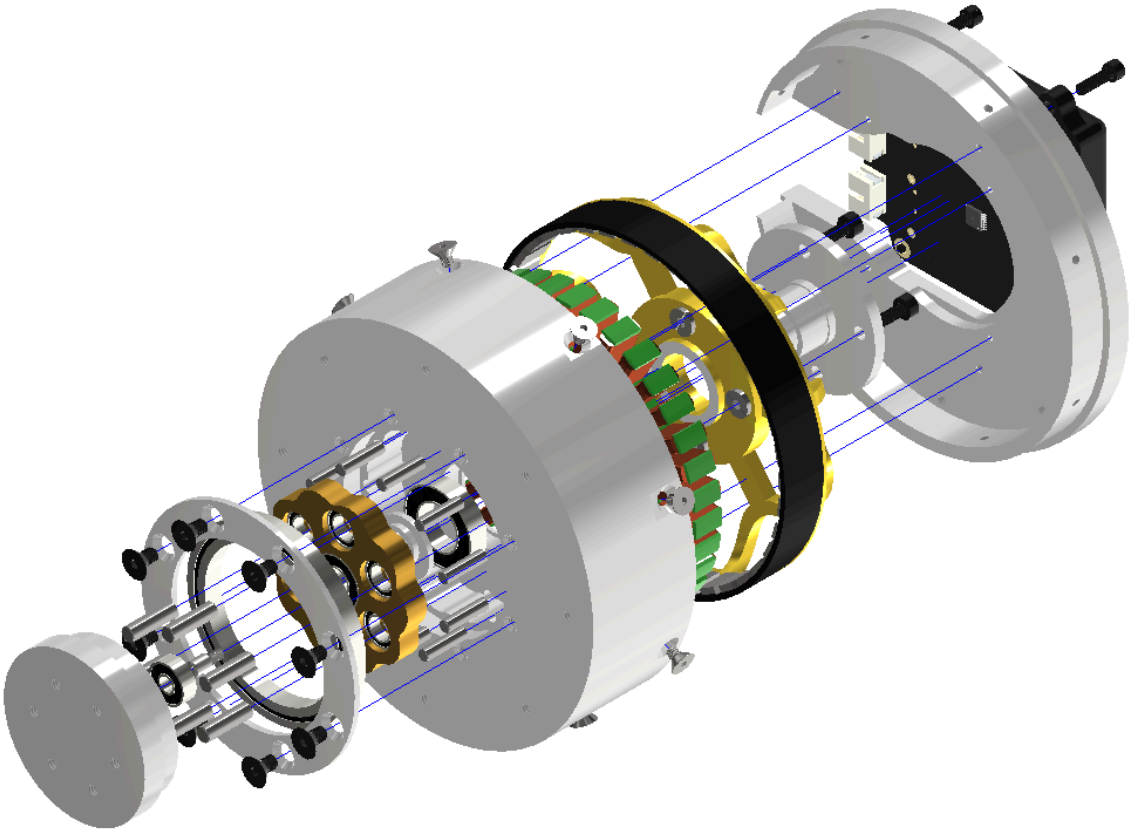


Figure 19.1: Actuator Front, Exploded View

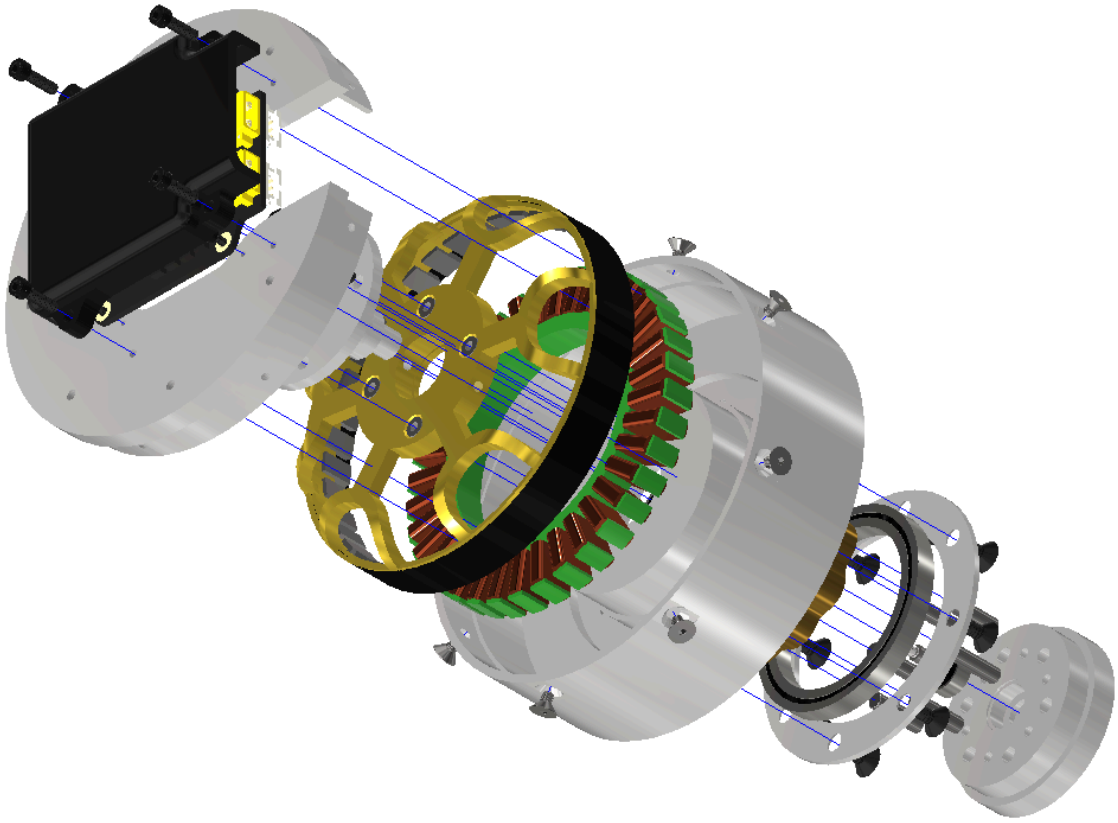


Figure 19.2: Actuator Rear, Exploded View

Figure 20: Component Sets of Three Actuators

Manufacturing

Manufacturing was a major hurdle throughout this project. It is quite easy to CAD an assembly or a part that is extremely difficult to actually manufacture. There were only a few manufacturing capabilities that were available for use during this project. The two major manufacturing capabilities were the use of 3D printing and the use of a DIY hobby CNC milling machine.

Special attention was paid to the capabilities of the DIY milling machine. It is a custom designed and built CNC milling machine. It has been designed as an extremely capable desktop machine with the ability to rapidly and accurately machine non ferrous metals. Despite very careful design and construction, the machine was still limited by the fact that it was built by hand using low cost, commercially available components. Some of the limitations of this machine were: Limited dimensional accuracy compared to an industrial machine; tolerances under a few thousandths of an inch were difficult to maintain; tramming inaccuracies; tooling was limited to ER-11 tooling, the main drawback of this was the tool size, 1.5Kw spindle, limited the material removal rate of the process, extended milling times; high speed machining only, 10k RPM+ milling only; realistically, the spindle was set towards between 15k RPM and 20k RPM in order to maximize the torque of the spindle; extremely difficult probing.

Despite the limitations of the CNC, it was the most crucial manufacturing method available during this project. Without the capacity to machine metals, high loads and low tolerances would simply not have been possible. Hobby 3D printers are still unable to achieve the tolerances required for a project like this, and the material properties of plastic 3D printing filaments are simply not usable given the loads considered during this project. Although ferrous metals and high hardness materials were unavailable during the manufacturing process, the ability to use aluminum and high hardness bronze made this project possible.

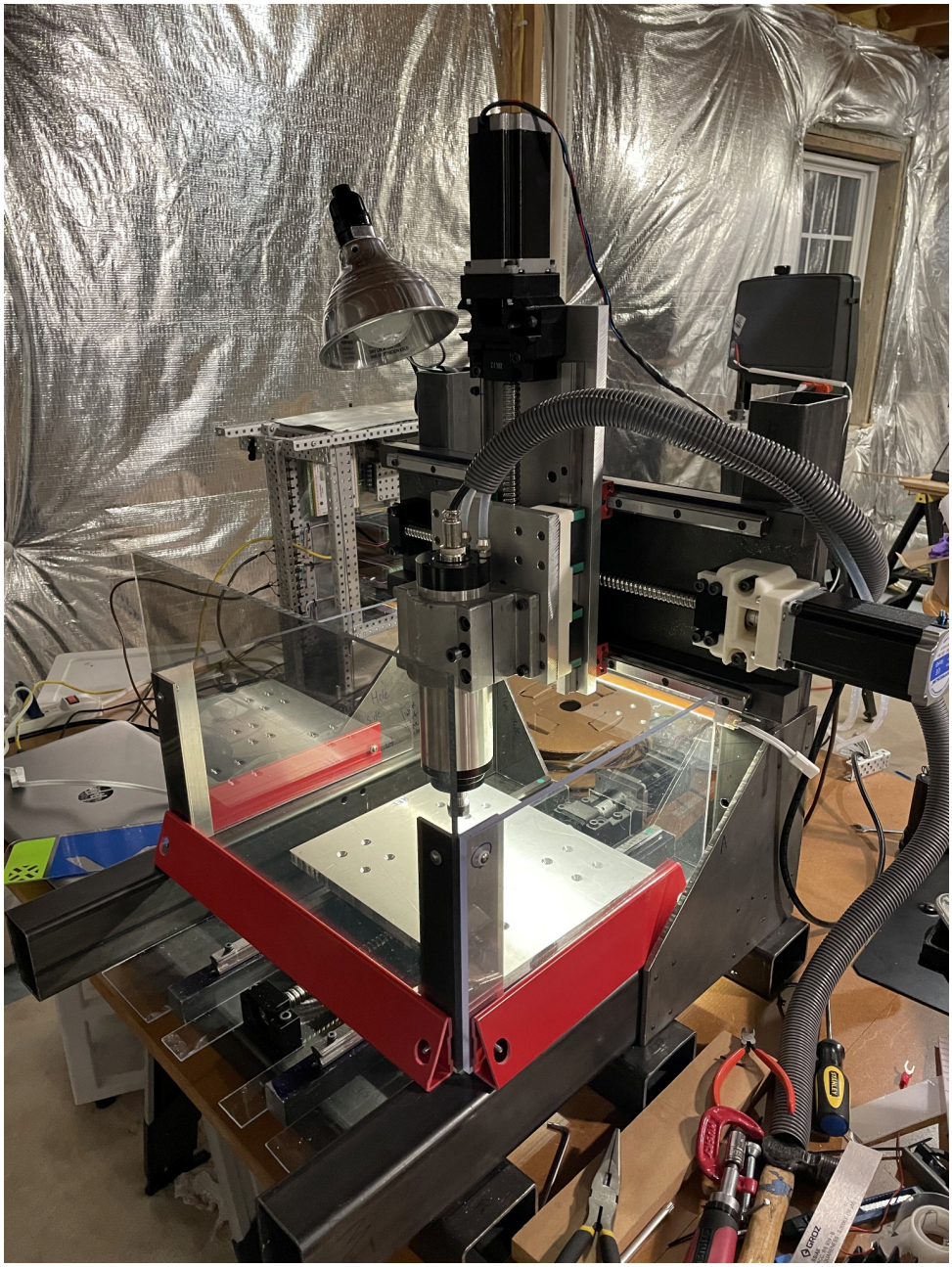


Figure 21: DIY Hobby CNC Mill

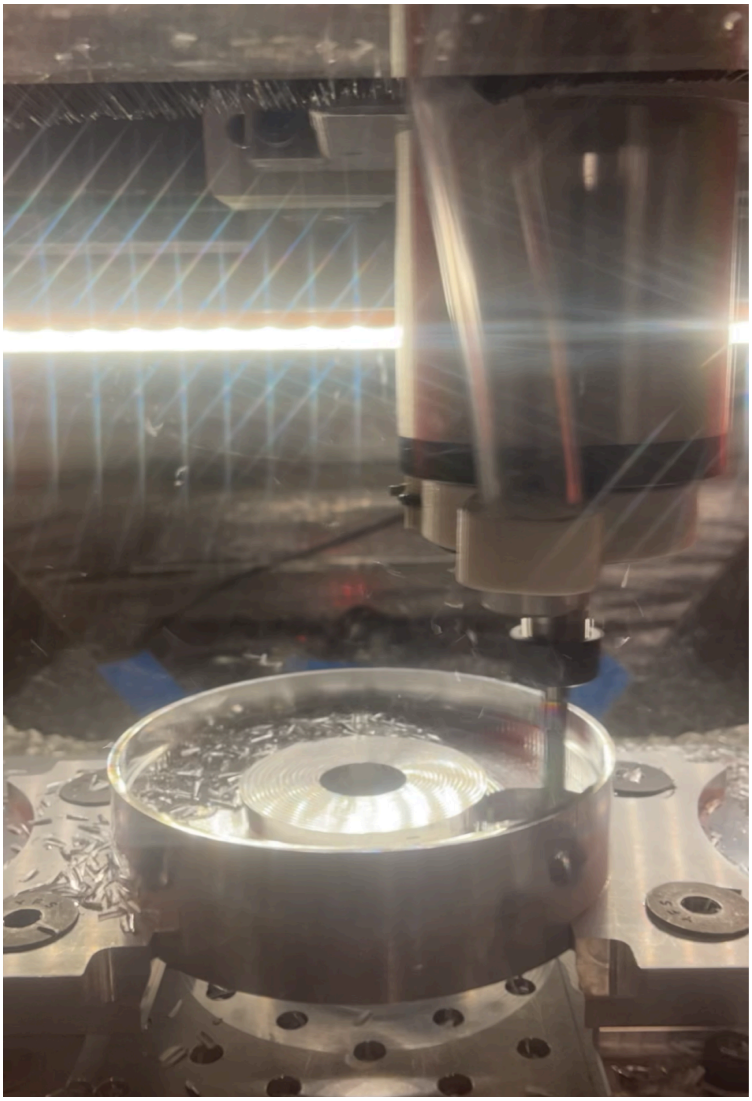


Figure 22: Machining of Actuator Housing

All of these aspects were considered in every single milled component on this robot. Design for manufacturing and assembly (DFMA) was a constant, and vital part of this project. Since it would be nearly impossible to list every single DFMA tradeoff and consideration made for this robot, I will focus on a few of the major considerations.

Probably the most critical consideration is the fact that the machining process is limited to high speed machining only. This process works perfectly fine in aluminum and softer materials, but is more difficult in very hard materials like steel. In materials like steel, the torque of a spindle begins to limit the speed at which material can be removed. For the spindle on this milling machine, the torque is so low that milling steel is extremely difficult. The effect of this is that it prevented the milling of steel gear sets. Most planetary gear sets are steel, which meant that this project could not effectively take advantage of planetary gears for the construction of an

actuator with a planetary gearing. The production of gears, especially ring gears, would have been prohibitively difficult, and plastic gears would not have been able to handle shock loads, so the result of this limitation was that the gearbox needed to use a different gearing. The end result is the development of a backdrivable cycloidal gearbox that could be used in place of the planetary gearing.

Another critical factor in the design of most parts in the actuator is the limitation of probing and the difficulties of doing multi-setup parts. This can be seen most easily in the design of the input shaft, which was designed such that all critical features could be milled in a single setup. In other cycloidal drives, the input shaft could not be milled in a single setup since the diameter of the input shaft does not strictly increase down the shaft. A design such as this would require multiple setups on the mill or the use of a lathe, which was not available.

3D printing has been utilized where possible in this design. The original leg design actually called for aluminum legs, but due to time constraints and the additional cost, the legs have been simplified to 3D prints. 3D printing allows for the quick and easy creation of complex geometry at a very low cost. It was attractive for use in the actuators themselves, but due to the relatively high loads in the actuators, plastic parts were not sufficient for this design. In addition to the limited strength of 3D printing, it is difficult to properly analyze FDM 3D prints due to their inherent anisotropic nature. Resin prints were possible, but most commercially available resins are brittle and would do poorly under shock loads. Meanwhile, engineering grade printers and resins are prohibitively expensive for use on this project.

3D printing has been used most actively in the legs where the parts are designed to be high strength with minimal material. Other parts have been 3D printed such as the actuator top and the Moteus cover to name a few.

Figure 23: Three Actuator Tops on a 3D Printer

Body and Leg Assembly

The main role of the quadruped body is the support and connection of the four legs. The body assembly consists of four separate leg assemblies. Typical mammalian bodies have different leg types for the front and rear legs. Although this design works well for animals, the manufacturing of two different types of legs for a robotic quadruped can be prohibitively expensive and time consuming. As such, this robot makes use of a universal leg design that is able to be used as both a front and rear leg. This way, the legs are interchangeable from a design and manufacturing standpoint. So in order to understand the body design, the first step is necessarily the specifics of the universal leg design.

Ideally, the legs themselves can be treated as near inertia-less. To achieve this, the torque of the actuator must be high enough that the leg inertia should not play a major role in determining the dynamics of leg movement. Instead of inertia dominating, the rotational speed

can be set exactly by the actuator itself. Thus, minimizing the moment of inertia for a leg is a priority in leg design. The actuator mass must be concentrated as high up the leg as possible. There are three actuators required for each leg: abductor, upper leg, and lower leg. Of these three, only the lower leg is a significant concern for the moment of inertia. Both the abductor and the upper leg can be placed high on the leg, essentially at the point of rotation and contribute very little to the moment of inertia. The lower leg actuator has to be dealt with in order to achieve the same reduction in inertia.

The key solution to this issue is offsetting the lower leg actuation. There are several methods of offsetting the lower leg movement. All methods involve a mechanical linkage that allows the actuator to be placed higher on the leg, so the movement of the lower leg can be controlled remotely. The three main options are: direct mechanical linkage in the form of a push/pull rod, ball screw or roller screw, and pulley or chain. There are of course a variety of other options to transmit power over a distance, but these three sets of options are the most widely used options that offer backdrivability in the knee joint.

The first option is a solid push/pull rod between the knee and the top of the leg. This method allows the lower leg to be driven by an actuator situated at the top of the leg through a four bar. The main disadvantage of this option is that the leg can only be driven in a single direction. This is similar to adding a kneecap to the leg since the lower leg can no longer be inverted over the knee joint. The other disadvantage is that this option requires the use of a heavy rod that can withstand impact loading from the leg. The conversion of the actuator's rotational motion into linear motion and then back to rotational motion at the knee joint also introduces multiple new failure modes and requires a massive increase in part count. Although a fully functional design was never constructed, weight estimates suggested that this design would be heavier than the final design choice.

The second option is the use of a ballscrew or a roller screw. This design places the lower leg actuator's main axis along the direction of the upper leg. Using the linear motion of a ball screw, the lower leg can be driven by the actuator. Although it isn't entirely clear due to the limited information available on commercial robotic quadrupeds, I believe this option is used on some of the larger robotic quadrupeds, such as Boston Dynamics' Spot. The disadvantages of this design are that it prevents inversion of the lower leg beyond the knee joint and that it adds the mass of heavy steel ball screws. Roller screws suffer the same issues, but are more backdrivable while costing significantly more. The great advantage of this design would be its weight bearing capacity. It can bear significantly more weight than other designs because of the size and capacity of the ball screws or roller screws.

The final option explored during the design phase is the use of belts or chains. This design is the simplest in nature because it just connects a pulley driven by an actuator at the top of the leg to the lower leg at the knee joint. One great advantage of this is that the lower leg can invert past the knee, there is no virtual kneecap to this design. Thus, it allows for much more creativity in programming and control since the mechanical design does not define the configuration of the knee directions when programming. The main disadvantage of this method

is the relatively low load bearing capacity of most standard belts and the weight of chains. Since there cannot be slippage between the lower leg and the actuator that is driving it, a timing belt is the only logical belt choice. However, most timing belts are relatively weak in tension since they aren't explicitly designed for heavy loads. The solution for this was to use high strength carbon fiber belts. These come with the added benefit of being extremely stiff, meaning that the belts add very little backlash between the actuator and the lower leg.

The next consideration with belts is the option of a gear ratio. Belts offer the ability to easily incorporate a gear ratio between the actuator and the knee joint. This is very important because the knee joint will need to be one of the strongest joints, meaning that it requires more torque than the others. When configured as a rear leg, the knee joint will be required to push off of the ground. While in use as a front leg, the knee joint needs to have the torque to pull the quadruped forward. The exact gear ratio is based off of the MIT mini cheetah, which selects a 1.55:1 gear ratio[21]. The selection of the gear ratio for this quadruped considered the max tension in the belt and selected appropriate pulleys based on the max tension in the belt. The selected ratio is 32:22 ~ 1.45:1, which enables the use of a 9mm carbon fiber belt (Gates 5MGT-600-09 Poly Chain GT Carbon Synchronous Belt) without overloading.

Thus far, the leg design includes the use of three actuators with the lower leg actuator being offset using a belt and pulley system. The final design elements of the leg are the structural components that bear the load on the leg itself. These were heavily inspired by the MIT mini cheetah, which, after an independent analysis, generally makes good choices in the overall design of the structural components. The two components in particular are the upper leg and lower leg. The upper leg is a clam shell design, two separate shells which screw together and leave a hollow cavity to allow for a belt to pass between the actuator and the knee joint. The second structural element is the lower leg, which is approximately an I-beam construction with attachments for rubber feet and the pulley.

The belt tensioner was one of the most critical, yet fruitlessly frustrating design aspects of the leg structure. The ideal belt tensioner does not apply loads to the clam shells directly. Instead, the forces would be almost entirely applied between the different sides of the belt. This would prevent shock loads from damaging the clam shell construction, instead wearing the belt, which is more easily replaced and more resilient to shock loading (due to the strength of the carbon fiber belts). Unfortunately, the space constraints of the clam shell design do not leave much room for a good tensioner design. The current tensioner design still applies loading to the sides of the clam shell, which is certainly an area for future improvement.

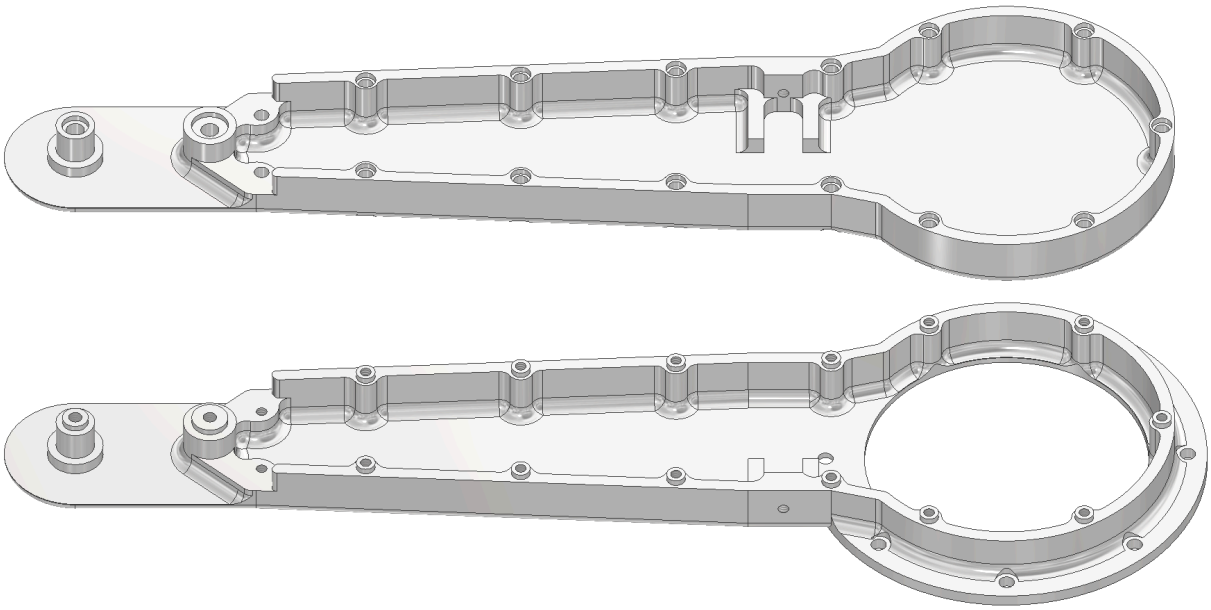


Figure 24: Lower Leg Clam Shell Assembly

Taken as a whole, this leg design is versatile design, with the ability to be used as both the front and rear legs. It minimizes part count, while making the very most of every part included. The moment of inertia has been minimized to allow for faster and easier leg control, while not compromising on strength or connection rigidity. The design is also lightweight and does not waste any mass. Future iterations can change the foot by simply connecting any new designs to the end. Future designs may also change the gear ratio of the knee by simply changing the pulleys out for different sizes.

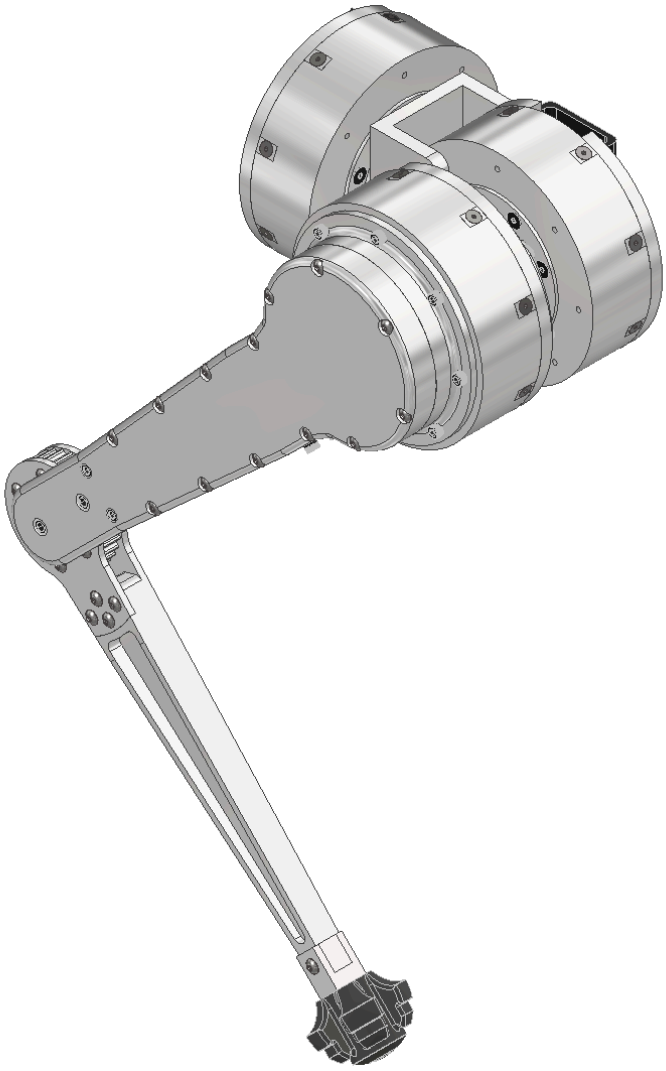


Figure 25: Full Quadruped Leg CAD

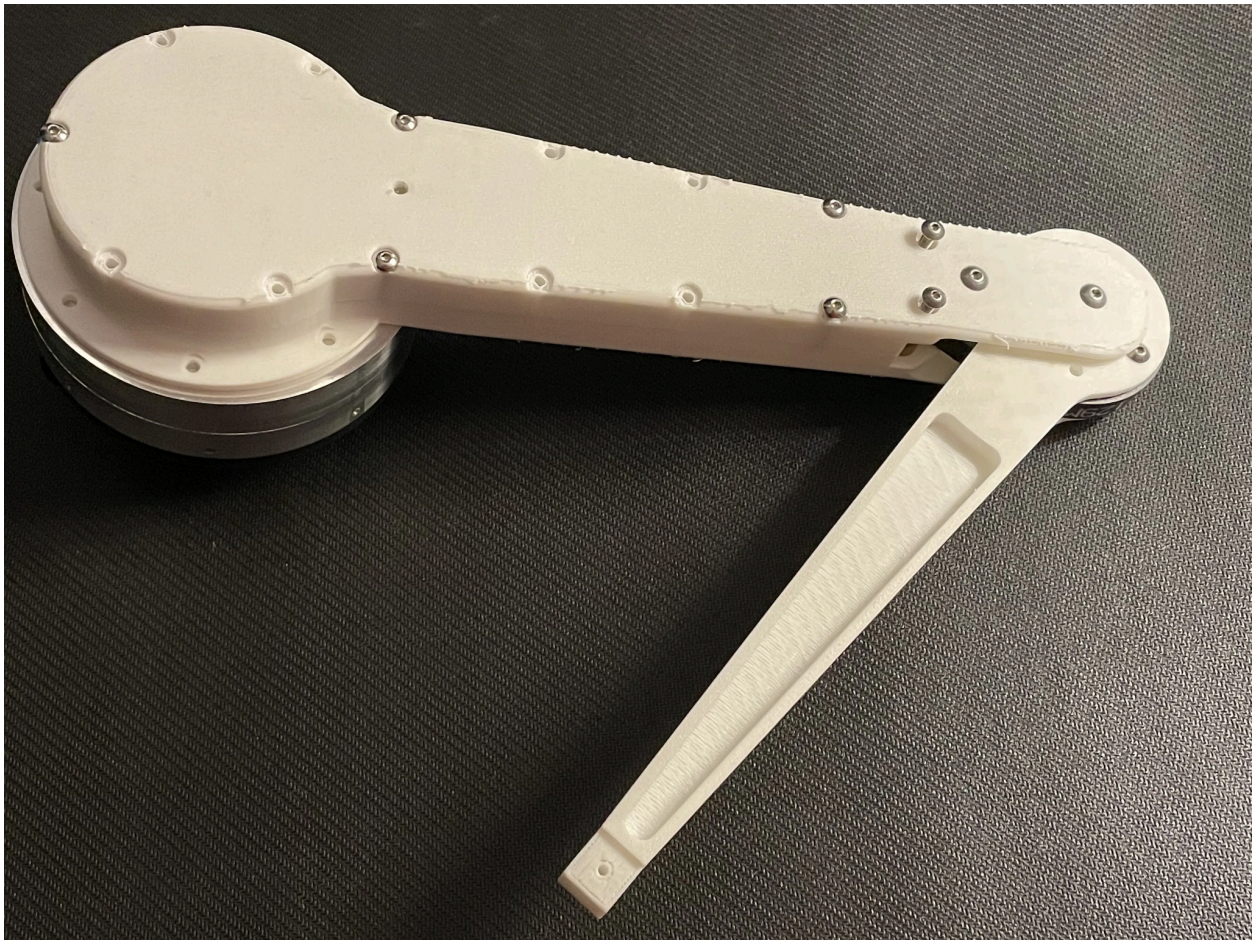


Figure 26: Full Quadruped Leg

The body of the quadruped is rather simple compared to the legs that it must support. It is a box-beam style design, employing bent sheet metal and 3D printed ribs to give lateral support and stability. Using this style allows for flexible and rapid design, but still gives a very rigid support structure. The legs are attached to the body by attaching the abductor actuators to the body assembly. The actuators can be supported on both sides which allows for a solid rigid connection of the four legs to the body assembly. The inside of the body assembly is left hollow to allow for electronics and the battery. The biggest issue with this design is maintaining the electronics inside. The current design requires that sheet metal panels be fully removed in order to diagnose wiring or access the electronics. Future designs would be wise to include a hinged doorway or other feature to make the electronics more easily accessible.

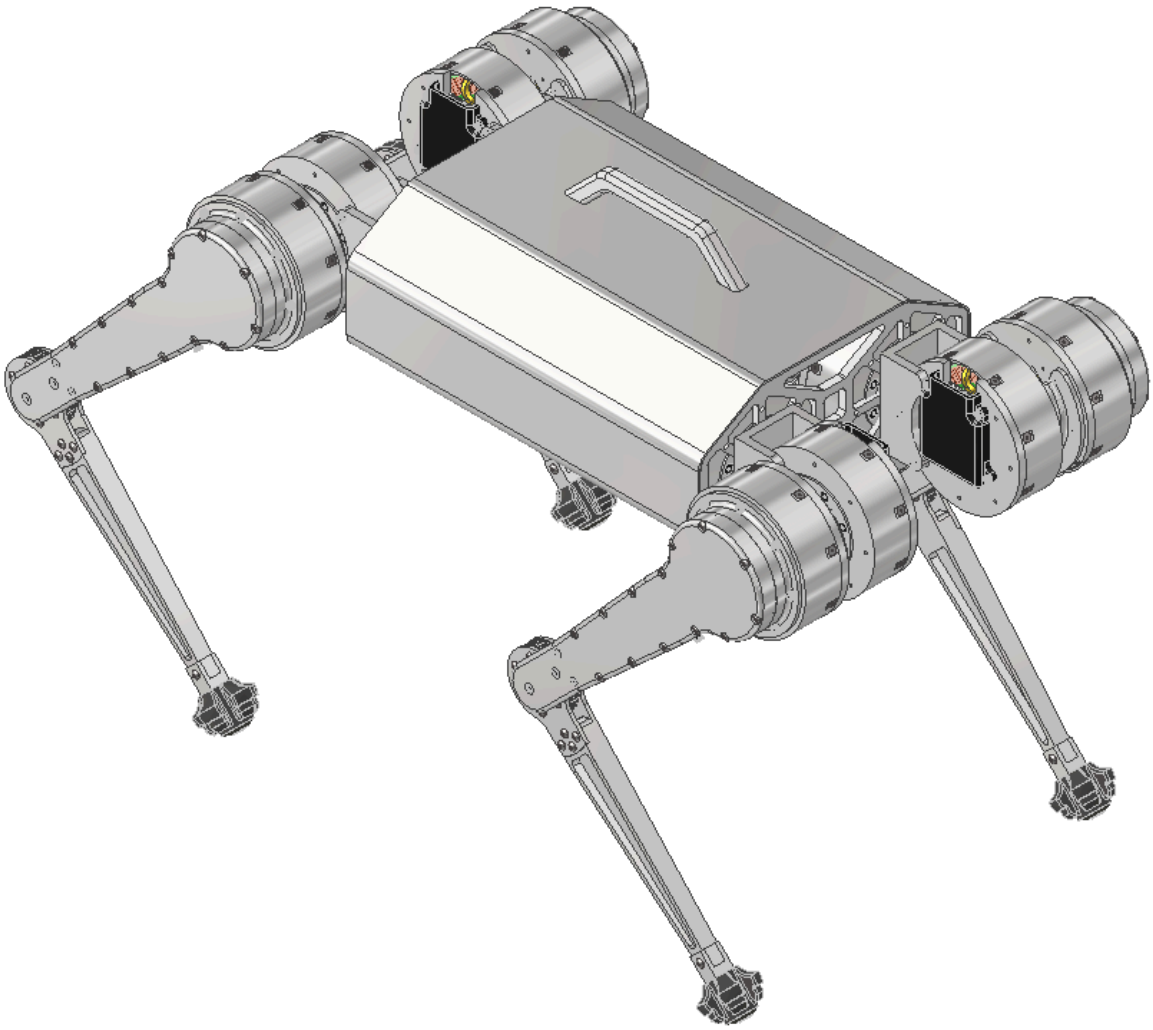
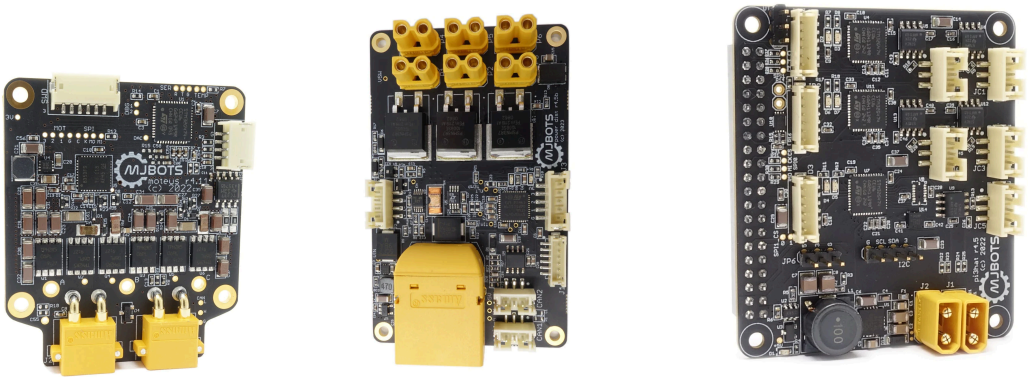


Figure 27: Full Body Assembly CAD

Electrical System

The main objective of the electrical system is to enable control of the brushless DC motors on each actuator. As described in the previous sections, the optimal controller will be a FOC with magnetic encoder support. The controller that best fit these requirements was the *Moteus r4.11 Controller* or referred here on as simply Moteus. The most important specifications which benefited this project were 3 phase control compatible with most standard brushless motors, FOC based torque control, 11 A continuous current limit without thermal management which allows for very high continuous torques, high speed CAN-FD communication (15-30kHz), and large voltage input range (10-44 V)[12].

Alternative controller options to Moteus were the ODrive controller, MIT Mini Cheetah controller, or several of hobby-grade, open-sourced FOCs. The open-sourced hobby controllers were generally underpowered, difficult to connect, and not as mature. Although the MIT Mini Cheetah controller is quite capable, there is no official supplier for these so fewer reliable copies were available. Thus, only the ODrive and Moteus provided industry grade controllers with well-known and consistent high performance. Between the ODrive and Moteus, the ODrive is more versatile and has had a longer development period. Although the Moteus itself lacks some of the versatility of the ODrive, it makes up for this with a great ecosystem of other electronics. The great advantage of the Moteus is the ecosystem of additional boards that enables the rapid development of robotic systems. MJBots also sells the *power dist r4.5b* Power Distribution Board (PDB) and the *pi3hat r4.5* (PiHiat), which facilitates communication between a Raspberry Pi and the Moteus controllers. The application of these boards will be discussed next.



Figures 28-30: MJBots Ecosystem Boards including r4.11 controller, power dist r4.5b, and pi3hat r4.5 (left to right respectively)[12,13,14]

For a robotic quadruped, each motor can be electrically treated as a resistor, capacitor, and inductor. Critically, the capacitive aspect of motors in addition to any supplementary capacitors can lead to potential issues when powering with batteries. When using battery powered systems on any capacitive load, the system can experience current inrush on start up. In the process of closing the circuit on a high capacitance system, the change of load on the battery is very large, which causes a high initial current until the capacitors are charged. This is also coupled by the fact that a motor draws the highest current at start up due to the lack of any back emf voltage at start up. A pre-charging circuit is used to protect against this. It limits the inrush current and prevents damage to more sensitive electronics. The MJBots PDB has an integrated pre-charging circuit to prevent damage to any Moteus boards or other capacitive loads connected to it. It allows for the power output from the battery to be split into six different lines for easy connection to different loads. This is ideal for a project like a quadruped due to the need for power on four separate legs and the main control board.

A robotic quadruped also requires a central computer for controlling the actual motion of the quadruped. The one drawback of using the PiHat is that it forces the use of a Raspberry Pi since the PiHat doesn't allow for easy adaptation to any other computer or main processor. The most high-level requirement for the central computer is to have enough processing power to control the actuators at a high rate. There are additional requirements on the voltage, power consumption, and ease of use, but these are all solved using a Raspberry Pi and PiHat since the PiHat will provide the pass-through power to the Raspberry Pi.

The one concern about the use of a Raspberry Pi is the specific types of processing power it has. The Raspberry Pi is primarily a CPU-based system, which would limit its ability to run GPU-based advanced machine learning models. Future work on this project could include the addition of a more GPU heavy processor such as a NVidia Jetson Nano or other board in the Jetson Series which have higher GPU processing capabilities. However, current work does not use more than a single-threaded process on the CPU, so the Raspberry Pi is more than sufficient for now.

The MJBots PiHat allows for CAN-FD communication between the Moteus controllers and a Raspberry Pi. The CAN protocol is an industry standard protocol used heavily in the auto industry for its robust communication and resistance to interference and noise. CAN uses a bus system which treats peripheral boards as members of a network rather than individual connections to each. This is achieved by daisy-chaining or linking communication terminals in a circular chain which reduces the number of ports on the PiHat but maintains the ability to communicate with all controllers. The FD variant is excellent because it allows for a Flexible Data (FD) rate. The flexibility of CAN-FD empowers the system to send more dense and frequent messages to and from the Raspberry Pi. This communication protocol is one major advantage above the other brushless controllers considered above. The higher frequency of messages allows for more actuators, more accurate control, more accurate sensing, and generally better usage.

The final element of the electrical system is the power supply. The first requirement on this system is the voltage of the battery. The MJBots boards all come with level-shifting circuits for their logic signals, so no specific voltage is required as an input as long as it is between 10 and 44 volts. Thus, the battery voltage will not affect the Moteus boards but will have a significant impact on the motors. The no-load speed of a motor is proportional to the voltage input, such that a higher voltage yields a higher base RPM. This would result in a greater current draw for the same torque when the motor is placed under load. Heat production in the coils of the motor is higher for a greater current than a greater voltage under the same load. Thus, a higher current would not only be less thermally efficient, but it could also potentially heat up the motors past a safe limit. This safe limit is defined by the motor manufacturer. Additionally, since these motors were originally designed for use on drones, keeping a larger margin between this safe limit is even more critical considering the lack of cooling in the quadruped setup compared to the design use. For the selected motors, there were two options for the running voltage: 22.2V (6s

LiPo Battery) and 44.4V (12s LiPo Battery). The 22.2V option was more efficient overall than the 44.4V counterpart, sacrificing a potential higher RPM.

There is a major concern of safety when working with high voltage batteries as well. Such batteries can generate much larger currents over the same load throughout the circuitry, increasing the danger of potential catastrophes from short circuits, power miscalculations, improper grounding, and other such errors. With the many manual solder joints and the student designed build, avoiding such risks is paramount as they could lead to potential a battery fire and the release of toxic fumes. The lack of protective circuitry in drone motors developed the need to select a battery that would include features such as fault protection or charging protection. The easiest and cheapest solution to this was a 24V cordless power tool battery and charger.

The last design requirement for this battery was ensuring that it was capable of high instantaneous current draws. A battery can be thought of having two different capacities, total capacity and instantaneous current draw capacity. The former is how much total power the battery can supply which is determined by all the cells while the latter is how much current the battery can quickly supply which is determined by the individual cells. The instantaneous current draw is critical for high current maneuvers such as jumping, a feature that will hopefully be implemented in future iterations. Selecting a battery with high quality, modern cells that can handle these high current demands was critical. Thus, the Kobalt 24V[19] was selected for this project as it has high power outputs for high, immediate current demands, has a long lasting 8 A-hr total capacity, and was found at an affordable price (it was on sale).

The final note on the battery setup is the flexibility of the system even after full deployment. An excellent feature of the MJBots boards is that the battery can be readily swapped for a different battery if necessary as the PDB has an input range of 10 to 54VDC. Most foreseeable future issues with the battery voltage, maximum instantaneous current, or total capacity can be remedied with a substitution of the battery. A change in battery voltage might necessitate the need for more active thermal management which is much easier to implement than an electrical stack overhaul.

The resultant electrical system uses a Raspberry Pi as a central computer, Moteus controllers to control the motors, and MJBots PDB for power distribution from the battery to actuators across the robot. The overall electrical schematic can be viewed in the figure below.

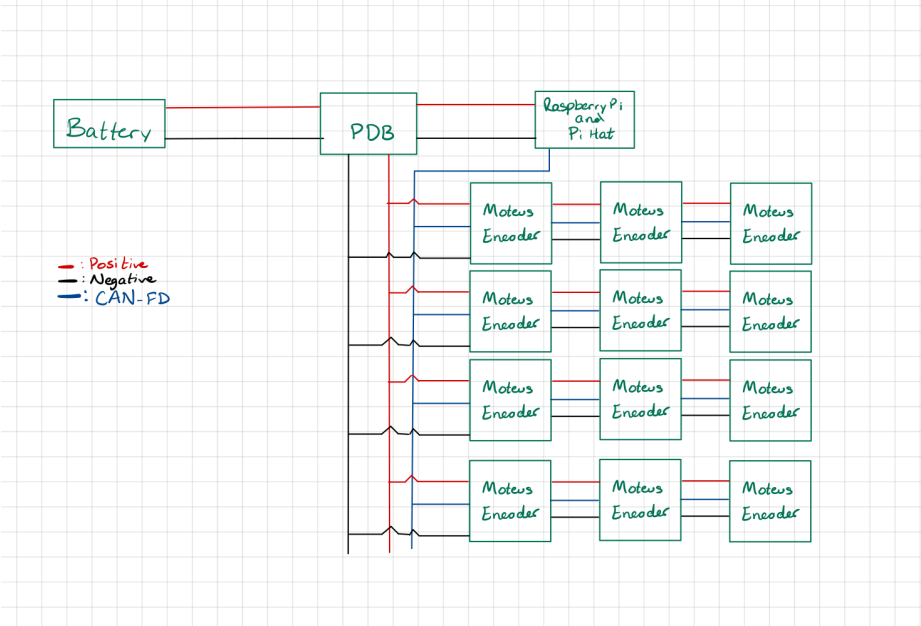


Figure 31: Overall Electrical Schematic

Electrical Assembly

Once the described parts were ordered, assembly was the next major step regarding the electrical systems. Most of the time was focused on manually soldering the connectors for power transmission, wires for CAN-FD messages, and between the Moteus encoders and the motors themselves. The first connection for the power transmission was the battery to PDB connection. Ensuring a strong joint on this connection especially was critical as it held the least regulated voltage in the entire electrical system and was responsible for meeting the required current draw for the motors. The most important metric for this consideration is surface area contact in regard to both within the wires themselves as well as the joints from wires to connection headers. If a current was traveling through a wire with reduced area, the wire would produce significantly more heat than if the area was increased. This would generate a fire risk or potentially lead to the wire burning itself apart.

The standardization for wire thickness is the American Wire Gauge (AWG), or simply gauge, where a lower gauge wire has a higher diameter which can handle high current loads and a higher gauge wire has a smaller diameter which results in a lower current limit. Wires can come in two forms: a single wire, or a solid wire, where the gauge represents the thickness of copper inside a plastic insulation coating or many high gauge wires wrapped together to create a lower gauge resulting in a stranded wire which is also wrapped in plastic insulation. Solid wire has better conductivity but has a higher risk of breaking due to its stiffness and a reduced surface area which means its heat dissipation is worse. It is also more challenging to ensure a proper

solder joint with solid wire due to this reduced surface area. For these reasons, stranded wires were chosen for the solder joints. Additional protection can be utilized such as twisted pairs and shielding to reduce electromagnetic interference (EMI) and noise. Twisted pair wires reduce the loop area created by the completed circuit, thus reducing the induced magnetic field and additional EMI that could be generated. Shielding similarly protects against EMI by blocking against the influence of external signals, provides more durability as it creates a stiffer wire, and provides additional insulation to the plastic coating.

For the battery to PDB connection, an old two-prong computer charger wire was used as it was an affordable way to provide a low gauge stranded wire that was also shielded. The wall plug was disassembled into the two separate prongs and inserted into a custom 3D printed adapter that would slide over the power tool battery and engage its locking mechanism to secure it in place. There was adequate spacing between the positive and negative terminals as well as the insulating plastic/air infill pattern that provided confidence in the safety of the connection to minimize the chance of a short circuit. This was attached to a XT-90 header which connected to the PDB. When soldering the wires here, it was important to ensure the solder flowed through both the wire and the connections. This is because the solder both mechanically joins the two pieces as well as electrically provides the necessary surface area contact for adequate current flow, especially under these high voltage conditions.

Slightly higher, 14 AWG wire and the smaller XT-30 header pins were used to connect the PDB to the PiHat as well as the to the Moteus controllers. There was difficulty when assembling the first wires as soldering was not one of my most developed skills. For these connections especially, the technique was more involved as the wire could not be wrapped around the header pins and the wire gauge was roughly the same size as the headers. Fortunately, solder quickly melts and joins with itself, almost acting as a sort of adhesive. I learned to use this to my advantage when creating these joints as I was able to complete the last few wires in the time it took to complete the first one.

Focusing on the CAN connectors, this involved soldering 3 pin headers onto either side of three 20 AWG wire. This higher gauge wire was selected as CAN-FD messages do not exceed roughly 3.5V throughout their communication, meaning a thinner wire could be used at a cheaper cost. However, there is a greater concern when using these wires as accidentally removing strands when stripping away the plastic encasing has a more significant impact on the overall surface area of the wire transmission. These joints were similar to the XT-30 and XT-90 power connectors in that a delicate hand was necessary to ensure robust connections but were easier after the initial practice and the nature of the header pins. Since both the CAN-FD and the power transmission wires could be daisy chained, a benefit of the MJBots ecosystem, this reduced the number of connections on the PDB and PiHat as well as the length of wires needed as all the actuators and boards would be centralized around the body, not proceeding down the leg. This allowed the cable lengths to be significantly reduced when considering the wire layouts of all the actuators leading back to the PDB and PiHat. A safe margin was kept in the wire lengths however as the body dimensions have not been finalized yet. Each of the joints described

were protected with heat shrink tubing to reduce the chance of short circuits due to accidental contact or arcs between high voltage wires.

The last solder joints were also some of the most important, connecting each of the stators of the Moteus controller boards. This required a distinct soldering technique over the connection wires as it required soldering three low gauge stranded wires, one for each phase of the stator, in a through hole connection on both sides of the assembled PCB. Being in close proximity to the other sensitive electronics with the potential large current draws of the motor, the criteria for robust connections was more stringent to avoid either the solder iron or stray wires from contacting the established electronics. A common failure when soldering to PCBs is having a cold solder joint. This is when the solder would be placed on the wire but not fully contacting the joint pad on the circuit board, reducing the true surface area achieved by the joint. These failures can be difficult to detect, especially upon first inspection, but after comparing to reference photos and repetition, it became easier to protect against. Additionally, when inserting these stranded wires through the PCB, it was important to neatly thread them so the most strands are in contact with the joint. Any stray wires that were not held by the solder were removed prior to heat shrinking to ensure proper current flow along the desired pathways. The last consideration was the angle that the wires were protruding from the PCB. The controller was to be placed right above the magnet and attached to a 3D printed part of the actuator assembly. To reduce the stress on the screws that will be in the softer plastic, the wires were reflowed after attachment to ensure that they were perpendicular to the board. This solder assembly required a slightly different learning curve than the previous joints but once a technique and rhythm was established, it became more rudimentary for the repeated joints.

Electrical Testing

After each wire set was completed, it was important to test the connection before placing a load under it to prevent any potential damage to any circuitry. A multimeter was utilized under continuity mode which alerts the user when any shorts circuits are detected. For the wires, the multimeter was placed in contact with the same terminal heads with the objective to hear the audible alert from the multimeter. This was useful not only for checking that the positive/negative terminals properly aligned on either side of the wire but also that the connection was strong enough. However, in contrast, when soldering the FOC controllers to the actuator stators, the objective was to ensure that no audible alert was heard across the test pins of the board such that no short had been created after the soldering process.

Once the first set of connections had been complete, a parallel effort was employed to test the capabilities of the electrical stack by using the Raspberry Pi and PiHat to control a single actuator, powered by the battery into the PDB. Initially, this involved testing the PiHat to controller subsystem with direct power from a verified USB wall adapter, a more reliable power source than the drill battery as it is commonly used to power the Raspberry Pi. After connecting

the Raspberry Pi to the computer, each successive connection was added after disconnecting to ensure an inrush of current is not formed while plugging into a live circuit, especially if accidentally done improperly. A test code was utilized to provide positional data to the Moteus encoder that would rotate the motor with no load, accounting for the gear ratio from the cycloidal gearbox. This test software is described in more detail in the software section. After it was confirmed that the motor could spin to the desired angles, from the inputs from the Raspberry Pi, the system was then tested with power from the battery to prove it could function independently. Similar to the testing described above, the battery was first connected to the PDB then the PiHat and encoder, each on their own power cycle to minimize any possible damage. Running the same test on the Raspberry Pi proved successful, demonstrating that the robotic quadruped could operate while mobile. This was in large part due to the robustness and user-friendliness when using the MJBots ecosystem, validating the rationale behind their selection.

Software System

Once the mechanical and electrical hardware designs were established, the next major objective was to create a software system that would enable the robotic quadruped to function and move. There are many ways that this goal could be achieved but given the limited time and the scope of the project during the semester, the target was to create a motion profile that the quadruped could use for forward motion. Since software testing was going to occur prior to the completion of the mechanical and electrical systems, it was thus necessary to implement a simulation software to demonstrate stability of the quadruped and its motion. This would also show the capabilities of the quadruped and provide a sandbox without risking damage to the assembled robot. The following dives into the process in implementing the software as well as difficulties encountered. Future steps are covered towards the end of the section.

Framework Selection

It was clear that Python was the easiest language as it has ample support and packages for this application and the MJBots ecosystem provides multiple Python packages, making it straightforward to control its encoders. However, the next choice would be the framework to structure the code and simulation. One option was to have a single thread Python script and use a physics-based simulation package such as MuJoCo, standing for Multi-Joint dynamics with Contact[18]. However, an industry standard framework arising from Stanford University is called ROS or Robotic Operating Software. This is a tool becoming more and more common in application as it not only utilizes multiple threads to increase speed of communication, but it is also designed with set libraries to make robotics development easy when working with sensors and actuators with the ability to work in both C/C++ and Python. Though this increases the required CPU power, it would not overwhelm the Raspberry Pi at this low demand application of walking. At the beginning stages of the software development, the ROS framework was chosen to meet these industry standards and create a greater learning opportunity in the design project.

ROS was originally designed for Linux based operating systems (OS) such as Ubuntu, the same basis for the operating system of the Raspberry Pi (PiOS) and has known compatibility with it. However, after initial setup and testing of ROS on a Linux computer, transferring such functionality to a Raspberry Pi proved to be more difficult than initially anticipated. The difficulty arises with the distinction between the Linux OS running on the Raspberry Pi compared to a standard Ubuntu system. On the Raspberry Pi, it is then required to make a virtual environment that contains an image of the proper Linux OS on the Raspberry Pi running PiOS to form the ROS framework. While it is possible to set up this virtual environment using Docker, an OS-level virtualization platform, which already holds ROS images or virtual environments containing ROS. it would significantly increase the effort to develop the software through this method as there were doubts with respect to robustness, the difficulty of testing, and the greater uncertainty of functionality which could be the potential source of issues when developing. For such reasons, base Python with MuJoCo simulation package was selected after this initial effort to meet the time requirement as well as ease the required set up to begin development, minimizing any potential issues.

Position and Motion Profile Calculations

Each leg is a system of three rotational joints linked together to create three degrees of freedom for movement in all three directions. The hip and knee joint create motion in the vertical plane perpendicular to the front of the body. Their combination is required in order to avoid movement only in a single radius and enable planar motion. The abductor rotates about an axis parallel to the front of the body, though having a reduced range of motion, creates the necessary range of 3D movement for the leg by rotating the plane of the hip and knee joint.

With these three joints and the geometry of the leg, it is straightforward to calculate the position of the foot, or the bottom of the last limb. A standardization was established to translate and rotate an origin reference frame to the last reference frame of a robotic system called the Denavit–Hartenberg parameters or DH parameters. Utilizing this convention breaks the quadruped's leg into a series of clear parameters that can be converted into rotation and translation matrices that are combined into a single DH Matrix which contains the resultant rotational and translational information. This can further derive matrices containing velocity, acceleration, momentum, and inertia information but this was not utilized in this project. Using the translational information only to keep the orientation of the final reference frame the same, it is thus possible to determine the position of the foot given any set of inputted actuator angles. This result is called the positional forward kinematics and can be extended to take angular velocity information to determine velocity forward kinematics. Its counterpart, inverse kinematics, is equally as important in which, given a set of desired coordinates and rotation, it would result in the joint angles that would move the final reference plane or foot to the desired location and orientation. Inverse kinematics is the more difficult result to determine as for higher degrees of freedom robotic arms as a desired orientation may result in multiple possible solutions. Fortunately, for the three degrees of freedom legs in which orientation is not of

concern, the inverse kinematics are mostly deterministic. Both are necessary depending on the objective of the motion at the time and for confirming the result of the other.

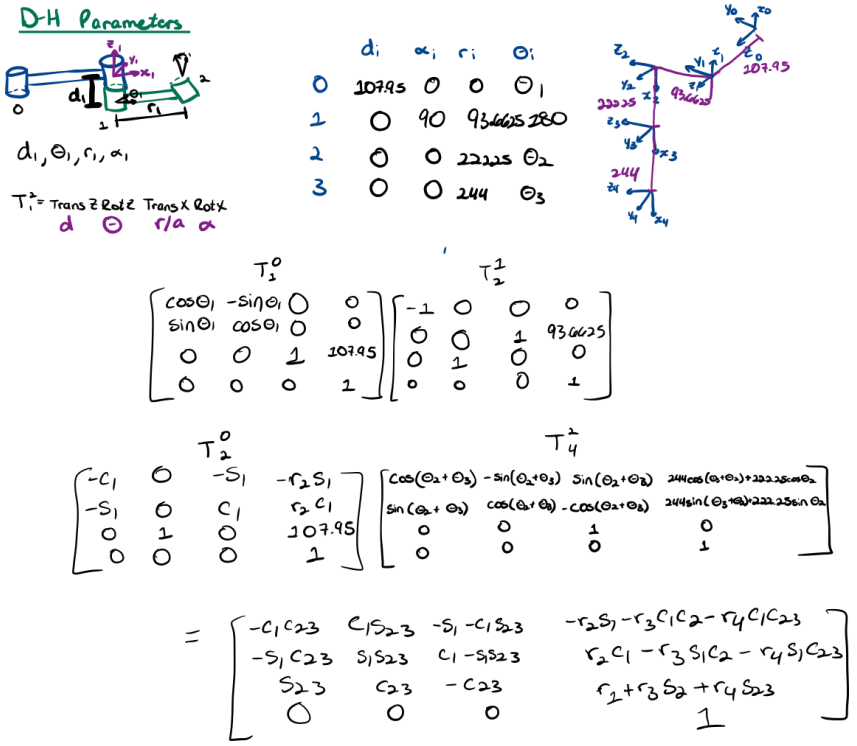


Figure 32: Image of DH Parameters and Forward Kinematics Derivation

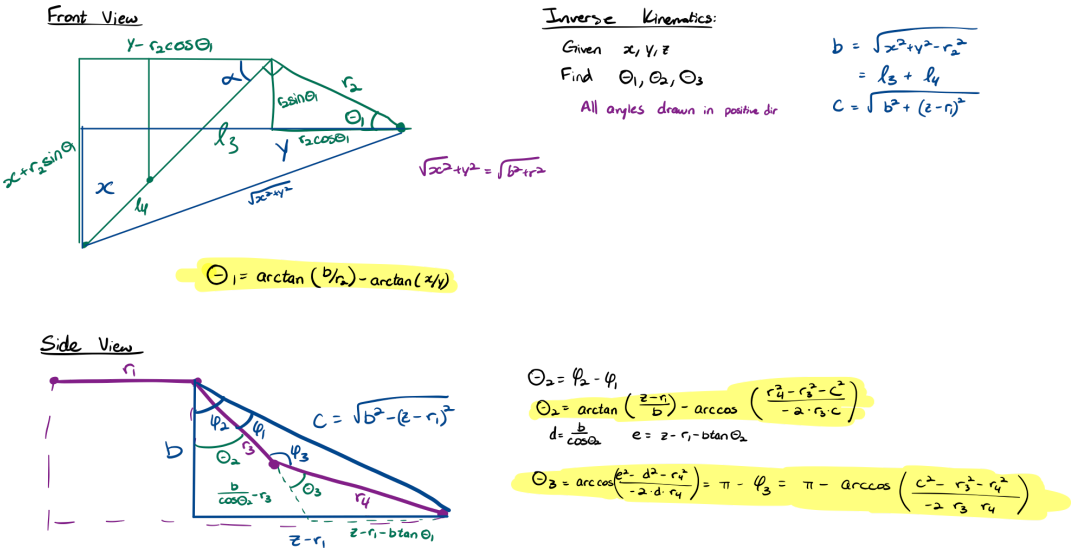


Figure 33: Image of Inverse Kinematics Derivation

After the calculation was determined for one leg, it can easily be applied for the other three legs. This only requires adjustment depending on the change in orientation of the origin

frame, which can be easily accounted for. Using the numpy library in Python, these resulting calculations can be implemented into Python as helper functions for the overall software system.

For any given change of position of a leg, each actuator likely needs to move a different distance than the others. In order to ensure that the desired location is reached by all joints at the same time, it is common to send distinct position and velocity at incremental time steps. This time step is usually in the milliseconds range whereas the encoder would handle the timesteps at frequencies beyond that. This is aptly given the name, Position Velocity and Time (PVT) motion. This motion control went through several iterations through development, described by the shape of the velocity profile over time. The first iteration for this profile was a rectangular profile where the velocity would increase as sharply as possible to cover the distance and then come to an abrupt stop. This would create a very abrupt motion pattern that could wear the limbs over time. The next iteration follows a trapezoidal velocity profile where instead acceleration follows a rectangular profile, with a period of no acceleration in the middle, then an inverse rectangular profile to slow down to a halt. This creates smoother motion as the actuators can ramp up to speed and then ramp down. One of the most common breakdowns of this profile is to have a ⅓,⅓,⅓ relationship where a third of the total time is spent on ramp up, constant velocity, and ramp down. For the quadruped, the ramp time was limited by the maximum acceleration which dictates the slope of the ramp. The next and often final iterations of this motion profile is called an S curve velocity profile where the trapezoidal profile falls to the acceleration and the jerk, the derivative of acceleration, follows a rectangular profile. This would result in an even smooth motion control but requires much more computational power that was not necessary for this application. Therefore, the trapezoidal velocity profile was selected. The following figure demonstrates the calculations for determining the time periods of the ramp up time, ramp down time, and constant velocity time given acceleration and the necessary distance to travel.

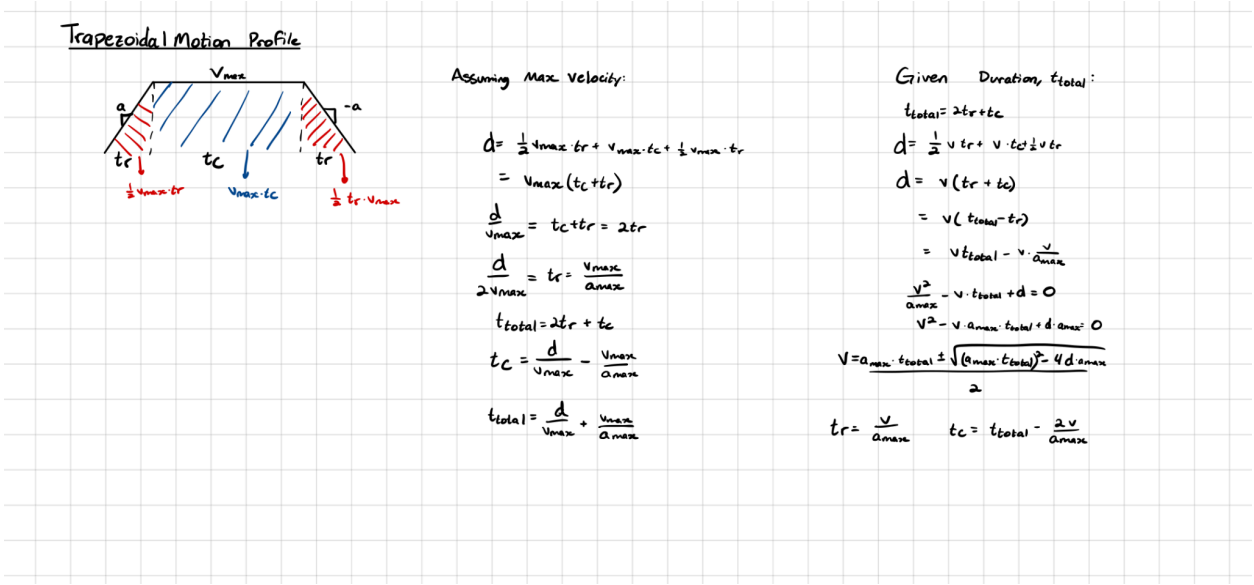
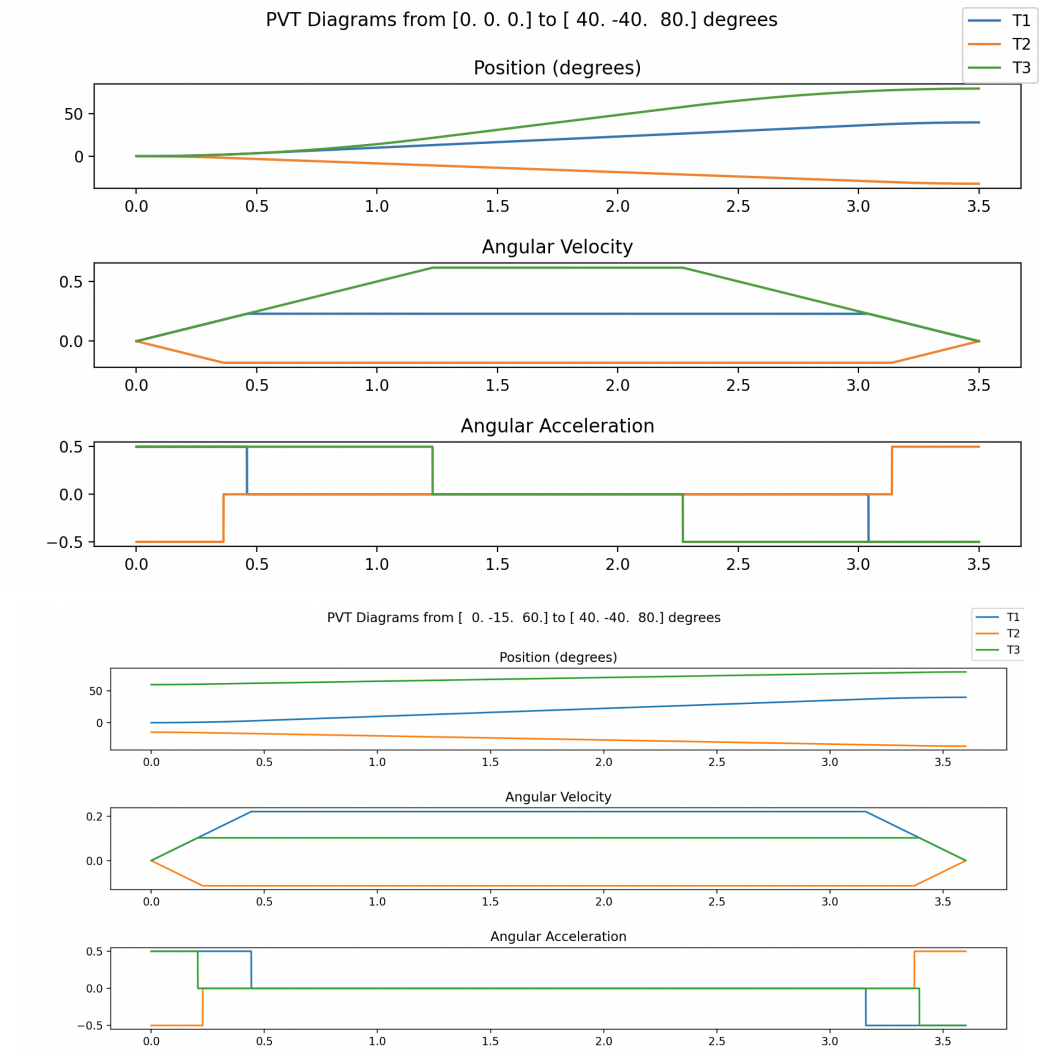


Figure 34: Derivation of Trapezoidal Motion Profile Equations

In terms of implementation for the software, the software relies on discrete calculations of position and velocity from the determined acceleration at each time step. Due to the nature of discrete calculations, some error is bound to occur more than continuous calculations. This error can be reduced using a shorter period but results in higher computations and memory needed. For each leg, the actuator that needs to travel the furthest distance will always move at maximum velocity of the system while the other legs will reduce their velocity accordingly. The ramp time is determined by the maximum velocity that the actuator can achieve such that it can ramp down for the same length. This creates the acceleration rectangular profile that can be used with the sample frequency to create position and velocity information.



Figures 35-36: Generated Test Plots for Trapezoidal PVT Motion for a Single Leg, plotted using the matplotlib.pyplot library

Gait Control and Simulation

Now that the quadruped can smoothly move any of its limbs from one position to another, the next step was to determine the gait pattern to move the quadruped forward. This is accomplished by manipulating the legs to move the quadruped's center of gravity forward. There are several gait patterns that can accomplish this, primarily differentiating on the number of legs that are moving at a given time. For a single leg motion, often called a crawl or creep, the center of gravity remains within the triangle formed by the other three stationary legs while the moving leg helps propel the body forward. This results in a statically stable movement as the three legs can help stabilize the body for most ranges of the moving leg. If two legs move, this creates a trot pattern where the center of gravity needs to be along the line between the two stationary legs. This produces a dynamically stable system which requires constant motion to resist falling over but can move faster than a crawl. This can further extend to a run where three legs are moving at a given time, but this creates an unstable system that requires higher precision in order to function without falling over.

In this application, only statically stable gaits were considered for their ease of use. After implementation on the assembled quadruped, dynamically stable gaits were going to be the next step after confidence was established in the initial performance. Two gaits were implemented, a crawl and a creep gait. The crawl gait moves one leg at a time while keeping the center of gravity within the triangle formed by the other three legs. This motion translates the center of gravity to the opposite corner of the moving leg to ensure the greatest stability as the leg moves away from the body. The crawl gait brings the center of gravity forward by the end of a cycle, but it leads to slightly more circular motion of the center of gravity. The creep gait functions slightly differently but also has only one leg in the air at a given time, keeping it statically stable. It instead slowly moves each grounded leg backwards while maintaining contact as the moving leg reaches forward. This motion profile requires more synchronization and movement but propels the center of gravity forward linearly. This also helps account for the change in center of gravity by bringing the grounded legs backwards while the moving leg moves forward. The creep gait does result in a slightly slower movement, however as the legs are sliding in contact with the ground. The motion timing of both gaits is shown below.

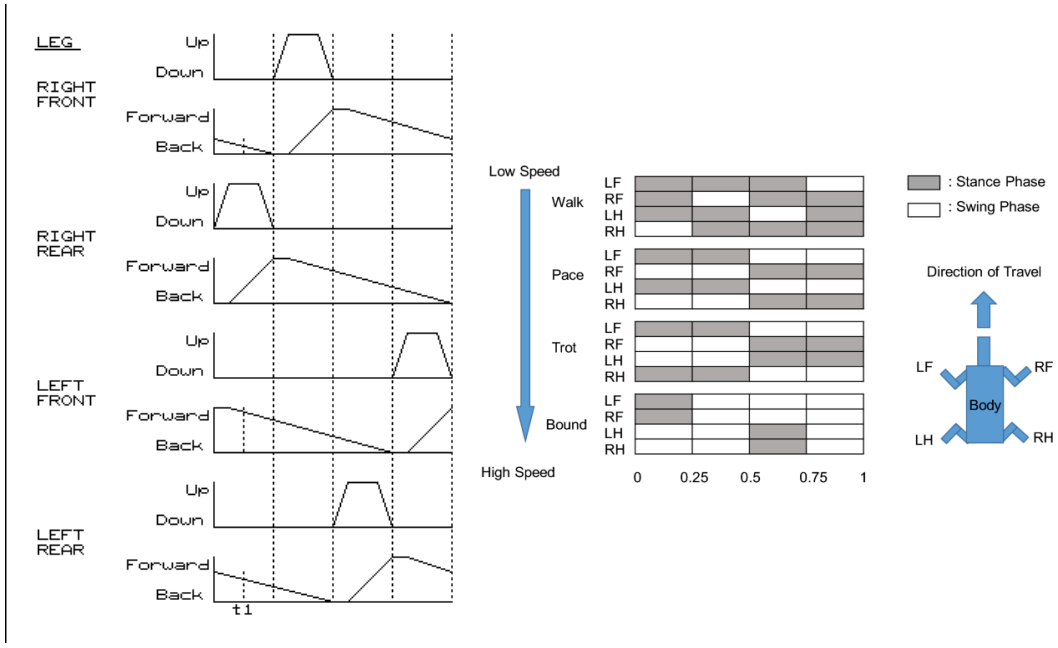
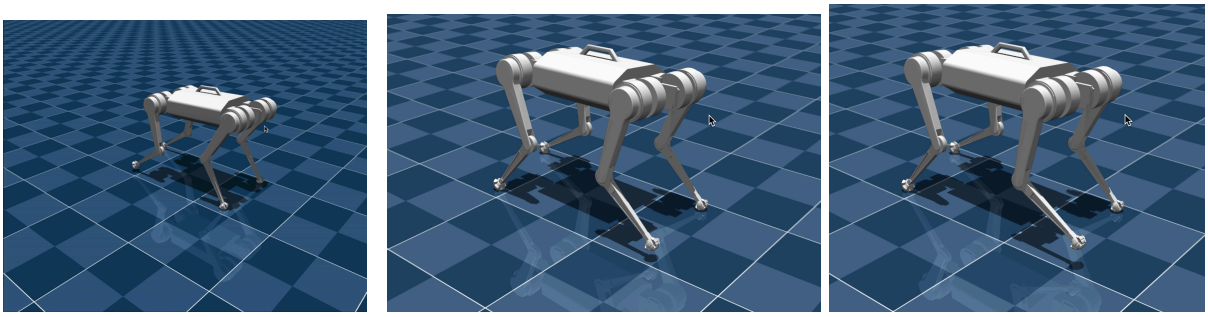


Figure 37-38: Image of Creep and Crawl Motion Timing Graphs Respectively[16,17]

Using the MuJoCo simulation software, it is easy to demonstrate the two gaits. MuJoCo can use the model of the robotic quadruped from the STL files and utilize them in an XML file. Within this XML file, the relationship between the joints and their rotational range can be defined. This can be placed into a virtual environment with the designated physics such as gravity. Though the weight specified is likely not accurate in the simulation, the relative weight of each limb to the body and to each other was maintained as well as possible. In this environment, the actuators are treated as positional servo motors as they can achieve an accurate position from an input. While this might seem inaccurate as the robotic quadruped utilizes brushless DC motors, the accuracy of the Moteus controller's positional feedback as well as the robustness of the Python package create a similar effect. After determining the steps that each gait requires and the position of each leg throughout the cycle, the simulation can run and demonstrate the forward motion. After the moving leg finishes in a step, it rotates the next leg, creating the motion cycle. One important consideration for the simulation is adding friction or no slip conditions between the foot and the ground. Though the CAD models do not include rubber feet, they would be added once the quadruped is finished. Therefore, the no slip condition with the ground needed to be manually added but it likely would not represent the actual friction coefficient of the finished product. Though the simulation is relatively slow, it can easily be sped up with a higher acceleration and velocity and refined accordingly.



Figures 39-41: Screenshots of MuJoCo Simulation Steps

Testing with Moteus

Implementation with the Moteus encoders requires some adjustments but is straightforward from the established code. For one, the Raspberry Pi must send all the information for every controller asynchronously, meaning concurrently with other processes, and this data needs to be continuously sent or the encoder will time out. Additionally, the current simulation is not capable of utilizing the velocity data from the PVT motion profile, a benefit of using the Moteus. The Moteus controller also contains a PID controller within them which can smoother the change of position. This can ease the sample rate on the PVT controller as well as create a higher accuracy on the final position. This PID was tuned for a single actuator and the same gains could be applied to the other actuators, but likely would change when fully assembled for the different joints. Like most PID controllers, the tuning began by increasing the proportional gain until the target position was reached within an acceptable settling time. However, the actuator had lots of chatter as it oscillated around the desired position at a high frequency. The proportional gain was slightly decreased and the derivative gain was then increased to add some damping and reduce oscillations. After which, the position error was low enough that an integral controller was not necessary. Under the current simulation, there is at most a 1 degree error with the final position. With the Moteus controller with the tuned PD controller, the error is reduced to around 0.1 degrees which is more than enough for our purposes. This implementation was not utilized as the final quadruped was not assembled, but it stands ready for that time.

Codebase Breakdown

This section aims to provide an overview of the developed code base that was described previously. There are two main groups of files, *sparky.xml* and *scene.xml* provide the information to create the models in the MuJoCo simulation while *sparky.py* is the primary file that runs the simulation and calculations. *sparky.xml* references a folder of STL files to form the quadruped

within the simulated scene. *sparky.py* is where most of the work was placed. There are multiple classes within this file that construct the objects needed for the breakdown of the quadruped. The Quad class is the highest level that encapsulates the rest. It uses the Leg class to create each of the leg objects, contains vital helper functions such as setting and getting the angles of all legs, and contains the functions for both creating the PVT diagrams as well as the step functions for both the creep gait and crawl gait. Currently, these gaits only work in the forward direction but extending this to multiple directions would be straightforward. The Leg class is utilized for defining the kinematics functions, more helper functions, and creating the motor control objects. The kinematic functions were placed within each leg to easily account for the offset needed depending on the orientation of each leg's origin reference frame. The leg object creates the Servo objects for each of the three actuators. This class is exclusively utilized for the MuJoCo simulation and enables the setting and reading of the angle and torque of the actuator. It directly links to the MuJoCo simulation but is not compatible with the Moteus controllers. For this purpose, the Moteus class is designed to allow the setting and reading of all Moteus controllers. This requires all Moteus data to be sent continuously on a single, separate thread to maintain their position. Currently this object is unused but its functionality from past testing with the assembled components as well as in preparation for the completed quadruped is within the code base. The simulation and objects are created within the main function area. There is also an option to avoid the simulation in order to test functionality. The current structure of the codebase is adequate but as functionality increases, separating the classes into their own files as well as a dedicated file for all helper functions would improve readability and better variable scope.

Further Improvements

For future iterations, there are many avenues which could be taken. It was already touched on that a dynamically stable or potentially unstable gait pattern would be examined next to move the quadruped faster. Additionally, dynamic control with inputs from a joystick would follow soon after. Soon after, capabilities to understand the torque feedback for the Moteus would allow the quadruped to determine if its legs are grounded when moving on uneven terrain, ramps, or stairs. After that, integration with more sensors would follow. The first would be an IMU which contains an accelerometer and gyroscope for live velocity, acceleration, and orientation data which could give more active feedback to maintain desired values. Furthermore, the addition of a LiDAR or ZED camera would allow the quadruped to see potential obstacles and recognize objectives in the surrounding environment. The last, most ambitious undertaking would be adding a form of AI/machine learning to provide intelligent feedback to the quadruped. This would far down the line however. The possibilities for growth, with enough time, are nearly limitless.

Conclusion

This project was extremely ambitious. It attempted to replicate the work of multiple decades of robotics research and development in the span of a couple months with very limited resources. The mechanical design is almost entirely complete by this point; however, manufacturing remains a high barrier due to the time and complexities involved with high-capacity machining. Due to these constraints, the quadruped will remain in development until sufficient time is available.

One of the biggest successes of this project was the development of a novel actuator. Currently, this is the only known design that attempts to place a cycloidal gearbox inside of a motor stator. Testing has proven this design to be robust and effective. It can produce at least 20 Nm at stall during testing and has so far had very good qualitative thermal performance, so far never overheating. This design could theoretically be expanded to achieve extraordinarily high torque densities. The potential applications of this actuator design are numerous, and it is hopeful that progress can continue.

The controls side remains an area of active development at the time of writing. Dynamic robotic control is extremely important, yet monstrously complicated for a system as involved as this one. Future exploration should include more advanced controllers, such as model predictive controllers or machine learning algorithms. The current hand-programmed walking gait is insufficient for field use unless conditions can remain ideal. The development of robotic quadrupeds remains an area of active research. Within the past 10 years alone, robotic quadrupeds have transitioned from almost exclusively the topic of research labs, well-funded companies, and defense contractors to the open market. The price of usable dynamic quadrupeds has dropped significantly with the introduction of systems like the MIT Mini Cheetah and Unitree Go1. Robots like these show great potential for the continued growth of the field. These machines have taken years to develop and mature. It is hopeful that this project will be able to continue and grow in the future.

Appendix

Actuator BOM

Component	Use	Link	Per Unit Cost	Quantity Used	Cost
6801-2RS	Bottom Bearing	https://www.an	1.523333333	1	1.523333
694-2Z	Cycloidal Top Bearing	https://www.an	0.8325	1	0.8325
687-ZZ	Cycloidal Disk Main Bearing	https://www.an	0.8325	1	0.8325
Actuator Top	Actuator Top	https://www.mi	12.145	1	12.145
BLDC	Control	https://mjbots.c	50	1	50
BLDC Cover	Cover BLDC	Printed	0	1	0
Brushless Motor	Motor	https://www.ali	58	1	58
CRBT405	Crossed Roller Bearing	Alibaba	28	1	28
Cycloidal Disk	Cycloidal Disk	https://alcobrar	12.4	1	12.4
Cycloidal Housing	Cycloidal Housing	https://www.mi	15.21666667	1	15.21667
Input Shaft	Input Shaft	https://www.mi	4.716666667	1	4.716667
Magnet	Encoder	(comes with BLD	0	1	0
McMaster-91251A108	BLDC Cover Fasteners	https://www.mc	0.22875	4	0.915
McMaster-91253A144	Bearing Retaining Fastener	https://www.mc	0.124479167	8	0.995833
McMaster-91263A375	Actuator Top Fasteners	https://www.mc	0.174791667	8	1.398333
McMaster-91290A013	BLDC Fasteners	https://www.mc	0.314166667	4	1.256667
McMaster-91290A113	Input Shaft Fastener	https://www.mc	0.192083333	4	0.768333
McMaster-91595A158	Cycloidal Output Pins	https://www.mc	0.381944444	6	2.291667
McMaster-94128A103	Top Pin	https://www.mc	1.715	1	1.715
McMaster-98381A471	Cycloidal Fixed Pins	https://www.mc	0.186818182	11	2.055
MR106-ZZ	Cycloidal Output Bearing	https://www.an	0.777222222	6	4.663333
Output	Output	https://www.mi	4.716666667	1	4.716667
Retaining Ring	Retaining Ring	Scrap	0	1	0
Spacer	Spacer	Scrap	0	1	0
Top Spacer	Top Spacer	Scrap	0	1	0

Leg BOM

Component	Use	Link	Per Unit Cost	Quantity Used	Cost
Actuator	Actuator		204.4425	3	613.3275
Abductor Connector	Abductor Connector	https://w	12.2875	1	12.2875
Neutral Pulley	Neutral Pulley	https://us	23.68	1	23.68
Driven Pulley	Driven Pulley	https://us	17.66	1	17.66
Belt	Belt	https://w	29.37	1	29.37
Hip to Knee Connector	Hip to Knee Connector	https://w	12.145	1	12.145
Knee Actuator Spacer	Knee Actuator Spacer	Scrap	0	1	0
Upper Leg Inner Clam Shell	Upper Leg Inner Clam S	https://w	19.65333333	1	19.65333
Upper Leg Outer Clam	Upper Leg Outer Clam	https://w	19.65333333	1	19.65333
Tensioning Block	Tensioning Block	Scrap	0	1	0
Lower Leg	Lower Leg	https://w	19.65333333	1	19.65333
Lower Leg Top Plate	Lower Leg Top Plate	Scrap	0	1	0
McMaster-93441A461	Joint Spacer	https://w	2.5125	1	2.5125
R8-2RS	Knee Bearing	https://w	0.99875	2	1.9975
McMaster-1688K3	Belt Guides	https://w	0.83	2	1.66
McMaster-98381A473	Tensioner	https://w	2.4225	1	2.4225
McMaster-6391K757	Tensioner	https://w	1.65	1	1.65
McMaster-92320A345	Belt Guides	https://w	1.75	2	3.5
McMaster-92949A148	Lower Leg Plate Fasten	https://w	0.121041667	4	0.484167
McMaster-91251A152	Belt Guides	https://w	1.5125	2	3.025
McMaster-96006A619	Knee Pivot and Support	https://w	1.2425	2	2.485
McMaster-92949A144	Knee Pulley Fasteners	https://w	0.112083333	12	1.345
McMaster-91251A151	Clam Shell Fasteners	https://w	0.225769231	13	2.935
McMaster-91251A152	Driven Pulley Fastener	https://w	0.504166667	6	3.025
McMaster-91251A148	General Fasteners	https://w	0.14025	20	2.805
McMaster-91864A019	General Fasteners	https://w	0.359	15	5.385

Sources:

1. https://dhkim0821.github.io/papers/2020_ICRA_Vision.pdf
2. https://support.bostondynamics.com/s/article/Robot-specifications
3. https://www.anybotics.com/anymal-technical-specifications.pdfx
4. https://news.mit.edu/2019/mit-mini-cheetah-first-four-legged-robot-to-backflip-0304
5. https://shop.unitree.com/products/unitreeyushutechnologydog-artificial-intelligence-companion-bionic-companion-intelligent-robot-go1-quadruped-robot-dog
6. https://ieeexplore.ieee.org/document/6386227
7. https://arxiv.org/abs/1904.03815
8. https://www.alibaba.com/product-detail/Eaglepower-8308-Brushless-Motor-kv90-130kv_1600096940369.html
9. https://us.misumi-ec.com/vona2/detail/221302297165/?HissuCode=CRBT405AC1
10. https://www.motioncontroltips.com/how-to-calculate-crossed-roller-bearing-life-with-various-load-types/
11. https://us.misumi-ec.com/vona2/detail/221302297165/?HissuCode=CRBT405AC1

12. https://mjbots.com/products/moteus-r4-11
13. https://mjbots.com/products/moteus-r4-11?pr_prod_strat=lmf&pr_rec_id=3b27a6118&pr_rec_pid=7358414749857&pr_ref_pid=7646323409057&pr_seq=uniform
14. https://mjbots.com/products/mjbots-power-dist-r4-5b
15. https://www.anchorbronze.com/single-post/2018/02/10/gear-bronze-the-effect-of-tin-content#:~:text=The%20selection%20of%20the%20bronze,than%20with%20steel%20to%20steel.
16. https://oscarliang.com/quadruped-robot-gait-study/,
17. https://robomechjournal.springeropen.com/articles/10.1186/s40648-020-00174-1
18. https://mujoco.readthedocs.io/en/stable/overview.html
19. https://www.lowes.com/pd/Kobalt-24V-8-0Ah-Battery/5001874651?cm_mmc=shp-_-c-_-prd-_-tol-_-ggl-_-PMAX_TOL_000_Priority_Items-_-5001874651-_-local-_-0-_-0&gad_source=1&gclid=CjwKCAiA1fqrBhA1EiwAMU5m_5PQugfGd_naISWN6iyc7VV5S1jvBDzoFQuQ3UKTZ2C_I_-TIN3EVhoC0ewQAvD_BwE&gclsrc=aw.ds
20. https://hackaday.com/2022/12/19/mini-cheetah-clone-teardown-by-none-other-than-original-designer/
21. https://pcb.mit.edu/lectures/lecture_x/1057343368-MIT.pdf
22. https://shop.unitree.com/products/go1-motor?_pos=4&_sid=808d18c2f&_ss=r