Letter of Transmittal

Mr. James Johnson
The University of Texas at San Antonio
One UTSA Circle, San Antonio, TX 78249

SUBJECT: Final Report for Brobotics Inc. “Lil’Bro: An inexpensive open-source quadrupedal robot built for the Robotics and Motion Laboratory”. College of Engineering, The University of Texas at San Antonio.

Dear Mr. Johnson,

Enclosed is the final report for Team 31, Brobotics Inc., including topics of discussion within the document. Topics include the purpose of the project, the design solution, testing plan, and conclusions. For any questions or concerns, please contact the Team Lead of Brobotics Inc. Steven Farra at zvo618@my.utsa.edu.

Sincerely,

Steven Farra
Brobotics Inc. Team Lead

Pranav Bhounsule
Project Sponsor

Brobotics Inc. Team: Steven Farra, John Carroll, Mario Navarro, Emiliano Rodriguez.

Lil'Bro's
Final Report

Prepared by:

Steven Farra, Emiliano Rodriguez, John Carroll, Mario Navarro

Prepared for:

ME 4813 Senior Design II

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Robotics and automation have gained attention from many industries in recent years. One area of particular interest to industries such as the military, space exploration, and animatronics, is legged robotics. Researchers at the Robotics and Motion Laboratory (RAM Lab) at the University of Texas at San Antonio are interested in agile locomotion of legged machines, however, do not have a suitable robot for such usage. The robot's source code is encrypted and its components are only sold by its manufacturer and are, therefore, relatively expensive. Brobotics Inc. has designed, built, and tested a robot, known as Lil'Bro, for the RAM Lab to solve this problem. Lil'Bro is an open source four-legged robot that provides users with all of its source code and part designs. All of its legs are fabricated through 3D printing, so researchers are able to recreate the legs, or explore new designs, by printing the parts at the lab. Lil'Bro's cost is about 25% of that of the currently used robot. The researchers set specifications for Lil'Bro, which include a speed of 0.2 m/s, a carrying capacity of 25% of its total weight, and weighing less than 23 kg. These specifications were tested for within the RAM Lab, wherein tests revealed that the robot does meet all of the client's specifications. Future applications to this robot include adding control for more agile locomotion as well as increased dynamic stabilization.

1. Introduction and Background

1.1. Purpose

In recent years, legged robotics research has been of interest to industries such as the military, space exploration, and the entertainment industry. Most research explores different methods of establishing stabilization, agility, and efficiency, and most recently using deep learning algorithms to optimize for any of the three parameters. Researchers at the Robotics and Motion Laboratory are interested in contributing to this avenue of research, however, lack the tools to do so.

1.2. Objectives

The objective of this project is to design, build, and test an open source quadrupedal, or four-legged, robot for UTSA's Robotics and Motion Laboratory (RAM Lab). The robot, otherwise known as Lil'Bro, will be used by researchers at the RAM Lab for ongoing agile gait locomotion research. The manufacturer of the robot currently used in the lab does not provide the robot's source code with the purchase of the robot. The robot is also expensive and hard to maintain, which is why the Brobotics Inc. team aims to minimize the cost of and maximize accessibility to the robot's design. An online Github repository for the project will be created to allow current and future users to easily access the robot's source code. The team intends to further increase accessibility to users by designing the fabricated parts of the robot to be 3D printed, which has not been done by any other providers of similar-sized robots.

1.3.Specifications

The following are some key specifications for Lil'Bro:

I. Linear walking speed of 0.2 m/s
II. Carrying capacity of 25% of the robot's total weight
III. Weigh less than 23 kg
IV. Profile dimensions within one cubic meter

V. Open Source and licensed under GPL3

2. Concept Design

The design process began by establishing constraints set by the sponsor. This includes using a particular set of motor drivers, motors, and encoders. Incorporating all of these components into a design was necessary and was, therefore, the guide for the design. With the knowledge of the type of motors to be used, choosing a leg design was the next step. For this, the team looked to pre-existing robots as well as nature for inspiration, observing leg configurations of legged creatures. Two main configuration were realized, the inward knee joint configuration and the outward knee joint configuration.

2.1. Inward knee joint configuration

The inward knee joint configuration can be found in most mammals. As the name implies, the knee joints of all legs orientate the limbs as shown in Figure 1. The knee joints along a lateral half of the robot are closer to each other than the corresponding feet.

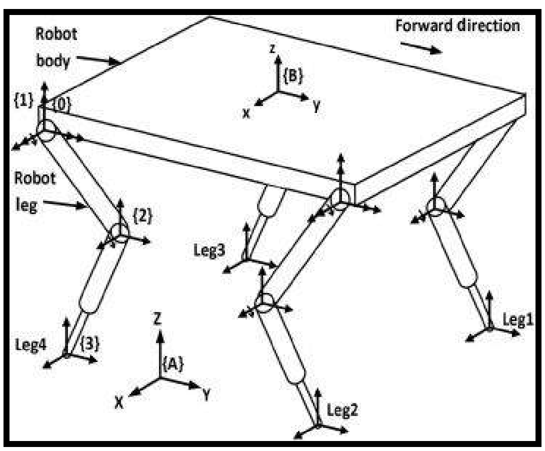


Figure 1. Inward knee joint configuration.

Due to its common presence in nature, this configuration is known to be optimal for a range of behavior. Agility is the main advantage this leg configuration provides, allowing its owner the ability to use dynamic gaits such as bounding and galloping.

2.2. Outward knee joint configuration

The outward knee joint configuration, on the other hand, offers different advantages and is found elsewhere. This configuration is mostly found in arachnids, with knee joints being orientated away from the body. Figure 2 shows the design inspired from this configuration.

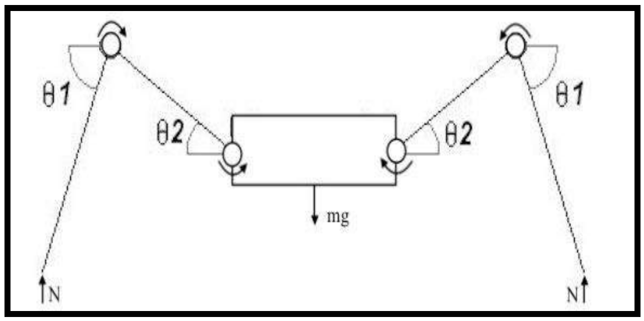
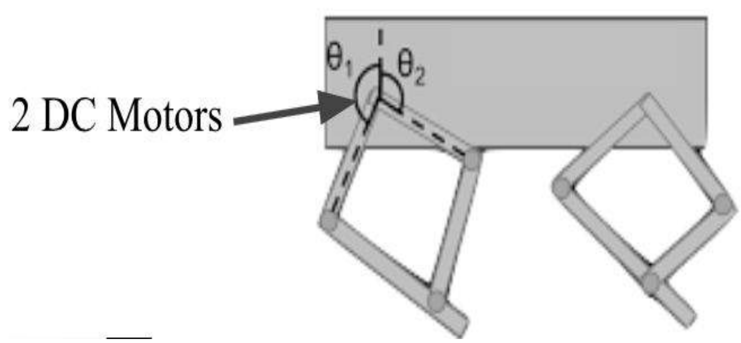


Figure 2. Outward knee joint configuration.

This leg configuration offers significant carrying capacity, as seen in ants, based on weight. In turn, this translates to robots, wherein a robot with this exhibits notable lifting capabilities. This also allows the leg bearer to be more power efficient at lower loading.

2.3. Hybrid knee joint configuration

While both leg configurations offered appealing advantages, they were not optimal. In an attempt to combine the advantages of both, the hybrid knee joint configuration was used. Coined as the symmetric five bar leg, researchers have investigated this configuration that isn't present in nature for its possibility of combining the advantages of the two previously mentioned leg configurations. The figure on the right shows this leg configuration. As indicated, two actuators are placed at the hips of the robot, as opposed to the presence of one actuator at the hip and another actuator at the knee. While it might not be as agile as a robot with



inward knees, this type of robot maintains some agility and carrying capacity. To investigate whether its proposed advantages hold true, a design tradeoff study was conducted.

2.4. Selection criteria

The selection criteria was based on a design tradeoff study conducted on the three designs proposed above. This study uses estimates of desired parameters that are expected to vary between the three designs to determine the optimal choice. The parameters include payload capacity, cost, operation time, and customer preference. Cost was estimated based on the leg mechanism necessary for each leg design, operation time was based on assuming the usage of a certain battery while applying a load at the foot and monitoring the reaction torque at the joints. When presented with the three designs, the project's sponsor expressed interest in a particular leg configuration, so it was taken into account. Table 2.1 shows the tradeoff matrix for the three designs with weighted values for each parameter.

Table 2.1. Tradeoff matrix.

	Payload Capacity Wt. 0.35		Cost Wt. 0.3		Operation Time Wt. 0.25		Customer Preference Wt. 0.1		Total
	U	W	U	W	U	W	U	W	W
Concept 1	0.3056	0.12224	0.0975	0.02925	0.62963	0.18889	0.8	0.08	0.42038
Concept 2	0.7863	0.31452	0.0175	0.00525	0.7619	0.22857	0.4	0.04	0.58834
Concept 3	0.8899	0.35596	0.1525	0.04575	0.83333	0.25	0.6	0.06	0.71171

Based on the shown results, concept 3, or the hybrid knee joint configuration, appeared to be most suitable for the parameters inspected in this study. With this fact in mind, the team decided to move forward with this design.

3. Prototype Design

The final design of Lil'Bro can be seen below in Figure 3.1.

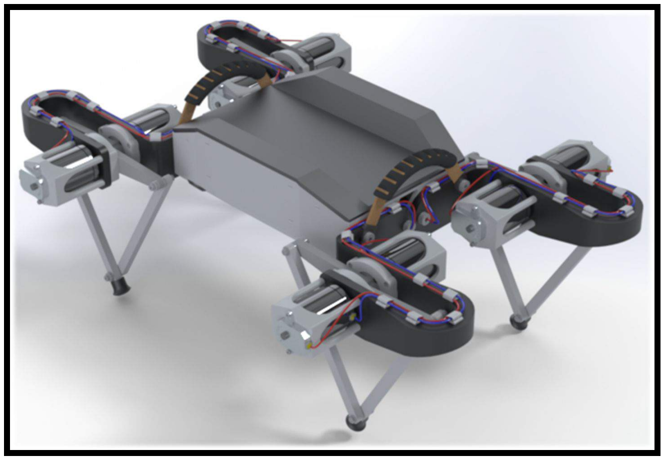


Figure 3.1. Lil'bro final design.

3.1. Key Features

Some key features of the robot include 8 Brushless DC motors that can be controlled through position, velocity, and current control by 4 motor drivers. This allows users to set any of the three control modes to each of the motors and pass commands to them independently. Each leg is actuated by two motors at the hip, and both motors connect to the same driver for ease of control. A gearbox is mounted on the motor that transmits power between the motor and the leg, while providing a step down.

Another feature includes wireless user-robot interaction. This is done through a Dualshock 3 handheld controller that connects to the robot via bluetooth. Users have the ability to pass 6 different analog inputs and 15 digital inputs independently for a variety of commands. The controller can also be charged by the robot itself through its single board computer, the Raspberry Pi.

A 22.2 V Lithium Polymer battery powers the robot, carrying a capacity of 4000mAH and constant discharge rate of 75 A. This battery is rechargeable and can be charged using fast charging.

3.2. Component breakdown with dimensions

3.2.1. Overall Assembly

Lil'bro's overall assembly consists of 5 main subassemblies, which are two left leg subassemblies, two right leg subassemblies, and a body subassembly. The each leg subassembly also includes a gearbox subassembly for each motor, so eight gearboxes overall. Appendix D includes drawings of the overall assembly and the five subassemblies.

3.2.2. Leg Subassembly

As mentioned above, there are four leg subassemblies. Each leg subassembly contains 76 parts including fasteners. All parts mount onto the leg mount, shown below in Figure 4.

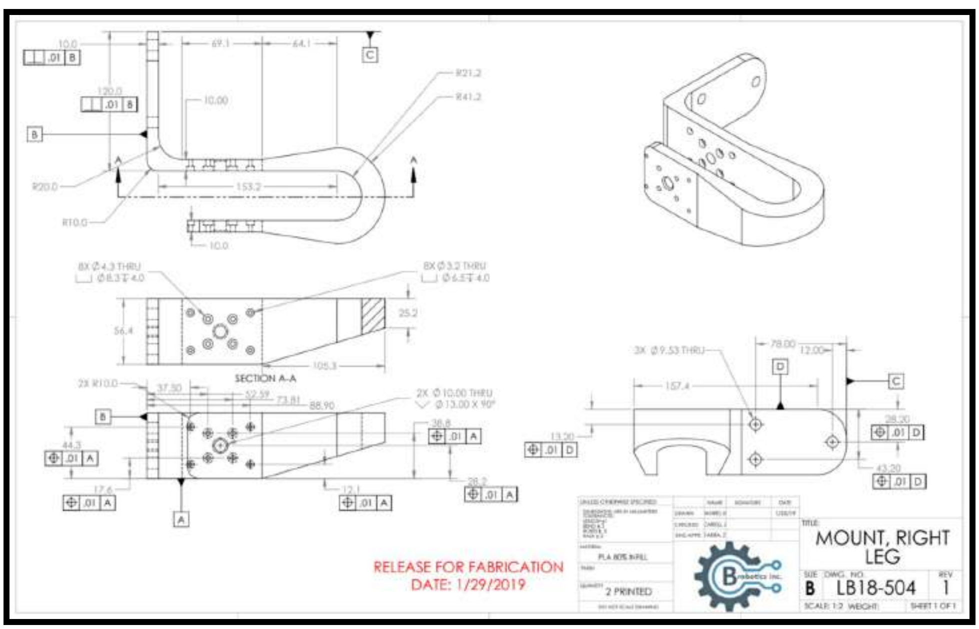


Figure 3.2. Leg mount part drawing.

This part is the most important part in the subassembly, due to its size and function. The motors mount directly onto the mount, and it in turn mounts onto the body in the overall assembly.

3.2.3. Body Subassembly

The body subassembly contains most of the electronics that drive the system including the single board computer and the four motor drivers, as well as the battery. Figure 3.3 shows the body shell, which is the component that houses the previously mentioned components, and the body subassembly consists of 122 parts. Serving the same function as the mount but for the body subassembly, the body shell is the component that determines the size of the subassembly. Therefore, the dimensions of the subassembly are driven by the dimensions of the body shell. Other noteworthy components that haven't been mentioned are included in the subassembly are the inertial measurement unit (IMU) and a variable voltage regulator. The IMU is used for the localization of the body and the voltage regulator steps the battery's voltage down to a suitable voltage for the single board computer.

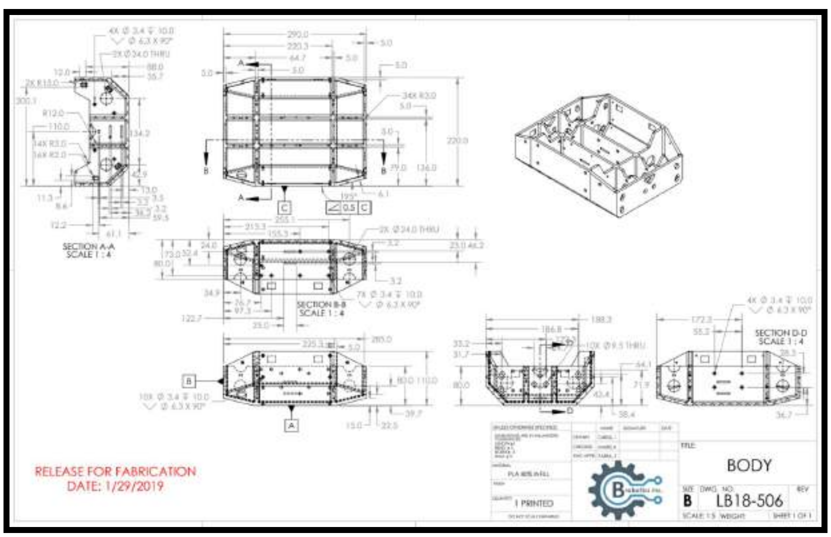


Figure 3.3. Body part drawing.

3.2.4. Gearbox Subassembly

A modular gearbox sub-assembly was designed to provide the individual upper leg sections the torque required for stable operations. The sponsor supplied motors provided a maximum output torque of 2.1 Newton meters. During tests the manufacture specified output was determined to be insufficient, and caused the motors to operate at a temperature that could damage other components on the assembly. Due to the limited amount of space available on the assembly, and the importance of keeping the weight down, a planetary gearbox design was used. The planetary gearbox configuration would apply the input from the motor, to the sun gear, which in turn drives the three planet gears, as seen in Figure 3.4 below.

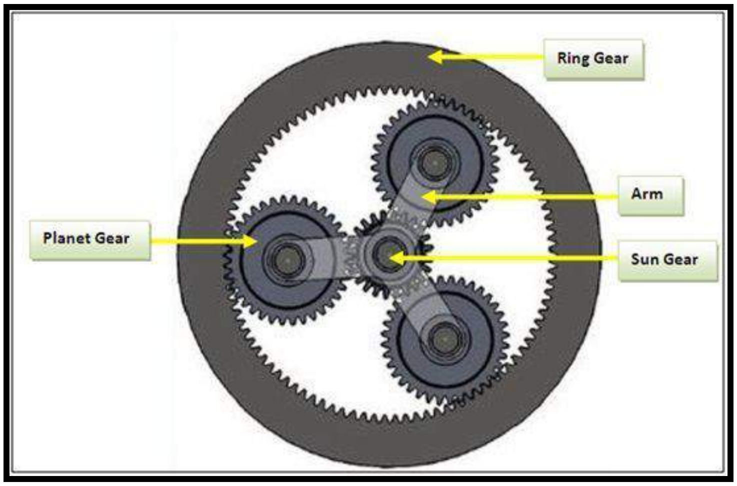


Figure 3.4. Planetary gearbox components.

The upper leg is designed to act as the arm, which is the output. The gears operate within the ring gear, which acts as a housing for the assembly, as seen in Figure 3.5 below.

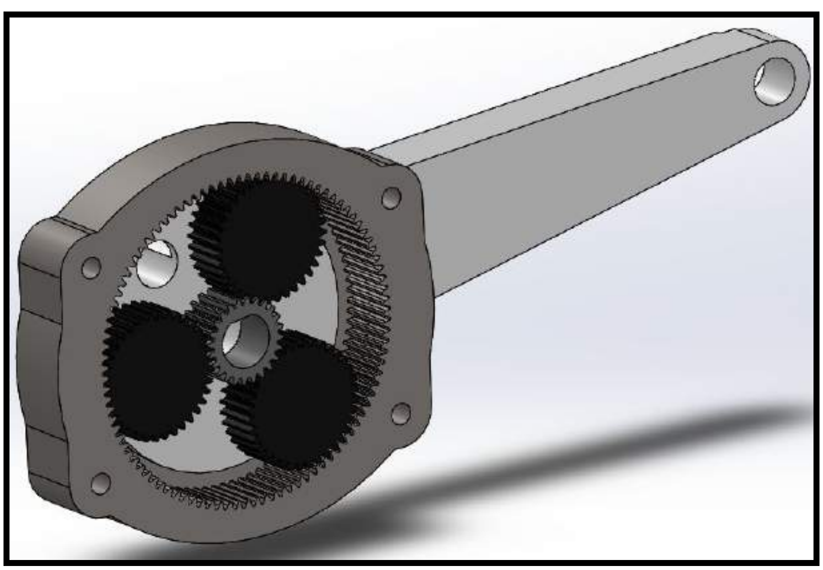


Figure 3.5. Gearbox design, with upper leg as arm.

The gearbox provides a step down ratio of 5:1 from the motor to the upper leg section. A diametrical pitch value of 0.5 was used for the design, with a pressure angle of 20 degrees. All the components for the gearbox are fabricated of polylactide excluding the sun gear. Due to the large 8mm d-bore in the sun gear, carbon steel sun gears were used to prevent failure. The entire gearbox assembly is comprised of six parts, and is designed to be mounted to the leg mounts, using the M3 fasteners that secure the motor housings.

3.3. Performance expectations

3.3.1. Linear Walking Speed

A determined linear walking speed of 0.2 meters per second was set as a project performance expectation. To ensure the expectation was met, a simple method was performed using the equation below,

$$Speed = Distance/Time$$

The method used a set distance and a recorded time. The expectation would be met when the distance covered, divided by the elapsed time to cover the distance resulted in an average of equal to or less than 0.2 meters per second over eight attempts. The chosen method of testing the performance expectation was designed to minimize the margin of error, however a few factors could affect the final results. If the unit did not walk in a straight path the calculated speed would be incorrect, the speed value would also be incorrect if the recorded time was not started or stopped correctly. To increase accuracy of the tests performed, tests where the unit did not walk in a straight path were discounted, and an average of eight attempts was used.

3.3.2. Carrying Capacity

A determined additional weight carrying capacity of, 25% of the units own weight was set as a project performance expectation. To ensure the expectation was met, a simple method was performed using the equation below,

$$Carrying\ Capacity\ =\ (Total\ weight\ of\ assembly) * 0.25$$

The method used a recorded value of the units weight to determine the additional weight to be carried. The expectation would be met when the unit could support the additional weight, while standing still, for eight seconds, on ten separate attempts. The chosen method of testing the performance expectation was designed to minimize the margin of error, by performing it multiple times to prove the performance expectation could be met.

The integration of a gearbox in the design of the assembly, provided the legs with enough torque to support 25% of the units weight, without any visible fluctuations in elevation, over all the attempts. In theory the 5:1 gear ratio increased the torque ratings of the motors by a factor of five, in an ideal situation without friction.

3.3.3. Estimated Life

Estimated life spans for the critical components included in the system were determined to provide the sponsor. For the power source, a lithium polymer battery, the manufacture rated life span is 400 to 600 charge cycles. With the average current used in the units operation, an estimated runtime over the life span of the battery resulted in 500-1500 hours, depending on use. The estimated life span of the plastic printed parts are well over 60 years, however due to the biodegradable nature of the material, if the parts are not kept in a clean, climate controlled environment, the components could begin to degrade in as quick as a year. The new development of the drivers, encoders, and motors prevent accurate reliability ratings, however estimated life span on the Raspberry Pi3 is known to vary based on the operating temperature; providing an ideal operating environment, the Raspberry Pi3 is estimated to be able to last 50 years. Due to the lack of redundancies in the system, the overall reliability of the system is low, this is due to the necessary 60+ components, that are needed for operation. Reliability of the system can be determined using the equation below,

$$\textit{Reliability of system} = \prod(\textit{Components reliability})$$

The components seen to have the lowest reliability were the drivers and the gearbox. Multiple driver failures were experienced during the fabrication process, some failures were a result of mishandling, others from unknown reasons. The gearbox failures were a result of the material they were fabricated from, these failures resulted in design changes.

3.4. Product Safety and Failure Modes

To prevent the unit from damaging itself or persons in the vicinity, certain measures were installed into the design to anticipate failure. One method of failure that was considered was possibility of the unit walking into objects, or even people. If the actuating legs were to collide with a rigid object, it could possibly result in parts failures. If the unit were to collide with a person it could cause injuries. To prevent the collision of actuating legs, the leg mounts were designed as guard for the legs; this way if the unit were to collide with an object while walking, it would not damage the components or the object it collided with.

Another anticipated failure is the possibility of a users loss of control. If the user lost control of the unit, it could continue moving and possibly damaging itself or people. To prevent this, the software will include safeguards to prevent the unit from moving when the controller connection is lost. A external power switch was also installed into the design, this would allow the cutoff of power to the motors without having to open the assembly. Carry handles were also included in the design, these would allow a single person to elevate the unit and prevent undesired motion. The carry handles can be seen in Figure 3.6 below.

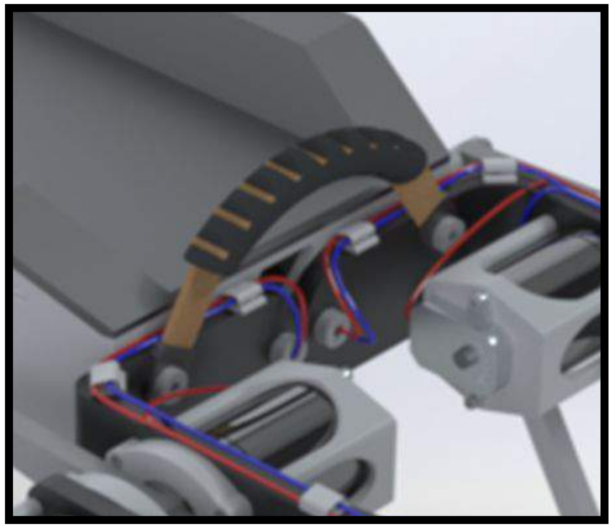


Figure 3.6. Illustration of carry handles (Used to lift unit)

Another failure that could cause the unit to damage itself is, operation attempts when the battery charge level is to low for proper operation. For operation on a low battery charge, parameters were tested to ensure the unit would not attempt to operate itself unless sufficient charge in the battery was present.

3.5. Design Refinements and Optimization

The most notable design refinement undergone by the robot is the addition of the gearbox subassembly. The need for it was revealed during some exploratory testing with legs, which were directly driven by the motors. The robot exhibited instability when subjected to a disturbance in its direction of motion. It also required a high current to operate, which in turn caused the motors to overheat. Initial attempts were made to cool the motors through adding a cooling fan to each motor, the provided forced convection did not suffice. The gearbox has already been elaborated on in a previous section, however, with the addition of the gearbox the motor housing design also changed. This was in an effort to remove forced convection and rely solely on free/natural convection by exposing more of the motor's surface to ambient air.

The body was also refined from its original design. The original body, while being spacious, was a plain box. It included components that were determined to be fragile, and complex to create. The design was optimized to used space more efficiently, by having just enough room for the components needed. It also utilized a simple to create structure that is stronger than the original, while also using less material. Bores were designed into the structure, for mounting components, instead of the original extrusion tabs. This prevented the need for creating threads in plastic parts, which also prevented the potential stripped threads that come with plastic threads.

The most important improvement made, was changing the sun gear from a plastic material to a metal. The sun gear on the gearbox requires a d-bore to mount to the rotor shaft on the motor, the geometry of the d-bore creates two critical stress concentration points. The use of plastic sun gears resulted in many failed tests with weight applied.

3.6. Physical Principles

Just as any walking creature, the robot utilizes Newton's Third Law to walk. This is done by applying a downward force through pushing the legs into the ground. This causes the ground to apply an equation reaction force in the opposite direction, propelling the robot forward with enough magnitude. When the robot remains static, the reaction force applied by the ground is equal to the weight of the robot. To move the robot, however, the magnitude of the ground's normal force has to be greater than the weight of the robot. A reasonable estimation for the required magnitude of the normal force is 1.5 times the weight at the hip. Figure 3.7 shows a free body diagram of the asymmetric five bar leg of the robot. The fifth link is the imaginary link l that extends from the hip to the foot. The necessary torque from the motors were calculated using the jacobian of the legs and the estimated force required by the robot to either remain static or move.

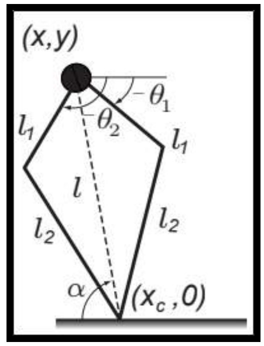


Figure 3.7. Asymmetric five bar leg diagram.

Motor speed reduction is carried out by a gearbox for each leg. The principle of power transmission while changing the properties of the transmitted power can be seen in the following relationship,

$$P = \tau \cdot \omega$$

where τ is the torque and ω is the angular speed. Given a constant power input between two gears, the torque and angular speed of the gear pair can be manipulated based on the sizes of the gears. In Lil'bro's case, the motor's power input is first transmitted to a driving gear that is smaller in diameter and number of teeth than the following gears. This causes the driving gear to rotate at a higher speed than that of the following gears. Assuming equal power between the gears, this causes the torque induced by the driven gears to increase proportionally.

4. Prototype Fabrication

The final fabrication of Lil'Bro can be seen below in figure 4.1.

Figure 4.1. Fabricated Lil'Bro

4.1. Fabrication Methods

The 3D printing method that team decided to utilize was fused deposition modeling (FDM). FDM printers use a thermoplastic filament which is then heated to its melting point and extruded layer by layer creating a three dimensional object. The team's sponsor also provided the use of his Robotics and Motion Laboratory (RAM) which contained 4 FDM printers. The easy access to these printers was an important reason for why the team opted for FDM printing.

4.1.1 3D Printing

As mentioned previously the RAM lab contains 4 FDM printers that were utilized by Brobotics Inc. to fabricate the important components of Lil'Bro. The printers used were the Ultimaker 3 extended and the 2 plus, and a Makerbot Z18. The fourth printer was not used in order to allow one printer to be available for other students in the lab. A considerable amount of troubleshooting and calibration were conducted on each printer in order to get a quality print. Calibration included adjusting the temperature of the nozzle which was increased from 200 to 215 degrees Celsius, adjusting the z axis on the ultimaker printers, reducing the layer height from .2 mm to .1mm giving the print a more refined appearance. The infill was modified from originally having a value of 20% to 80% on certain components. The pattern for which the infill will be printed in is very important, it was researched that cubic pattern was a much stronger infill pattern than the linear and diamond pattern. The nozzle size on both Ultimaker printers were replaced from a .4 mm to a .25 mm nozzle increasing the quality of the print. These are the main configurations that Brobotics Inc recommend users to try in order to increase the quality of their print.

4.1.2. Drawings of Fabricated Parts

Brobotics Inc. understood that 3D printing was not always the most accurate when it came to tolerances and exactness. After an extensive amount of research and multiple prints of the upper leg the team concluded that the tolerances for each part would contain a tolerance of .01 mm. Before printing the legs, a decision had to be made on which design was the optimal choice for the project that the team had created. A leg design that utilizes both an outward and inward knee configuration ultimately allowed for relatively lightweight legs to be used. The lightweight legs allow for quick position transition, while using minimal power to do so. Every designed component has had a drawing file/CAD created. Each file has the exact dimensions with its respective tolerances and properly spaced out allowing for anyone to easily understand the drawing and are able to print a part of their own.

4.1.3. Assembling Fabricated Parts into Subassemblies

4.1.3.1. Leg Subassembly

The leg subassembly is composed of two upper links, each connected to a lower link. These legs are fastened together with shoulder bolts that are placed inside sleeve bearings. Thrust contact bearings are placed between each upper leg and its corresponding lower leg with a washer on each side. A breakdown of the components can be seen in the exploded view included in figure 4.2.

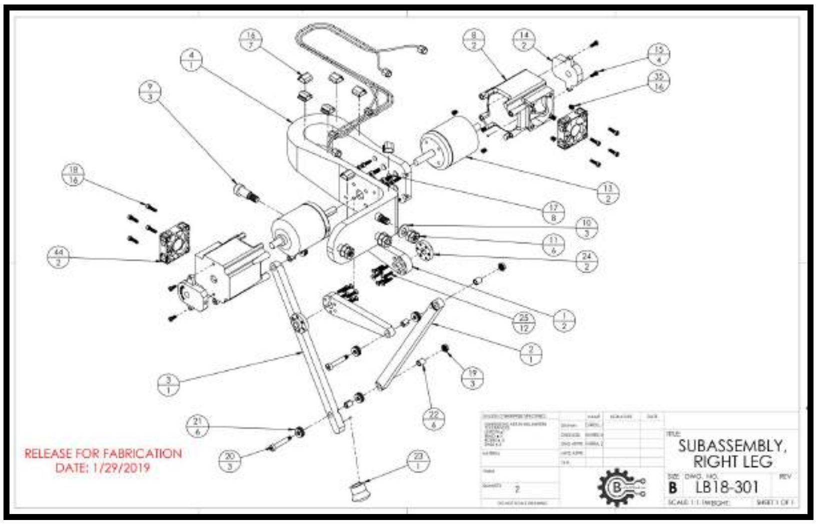


Figure 4.2. Leg subassembly exploded view.

4.1.3.2. Gearbox Subassembly

Because of the limited spacing, a planetary gearbox was implemented. The gearbox subassembly is made up of 5 gears; 1 sun gear, 3 planetary gears, and 1 ring gear. The sun gear is mounted on the motor shaft directly, while the planetary gears orbit the sun gear with the ring gear keeping them in bound.

4.1.3.3. Body Subassembly

The body sub assembly has several mounting holes and includes many components. All the electronics are housed in the body; the motor drivers, raspberry pi, imu, voltage regulator are all properly mounted in a designated spot. All wiring is navigated through the bores in each compartment of the body and are connected according to the wiring schematic in appendix D.

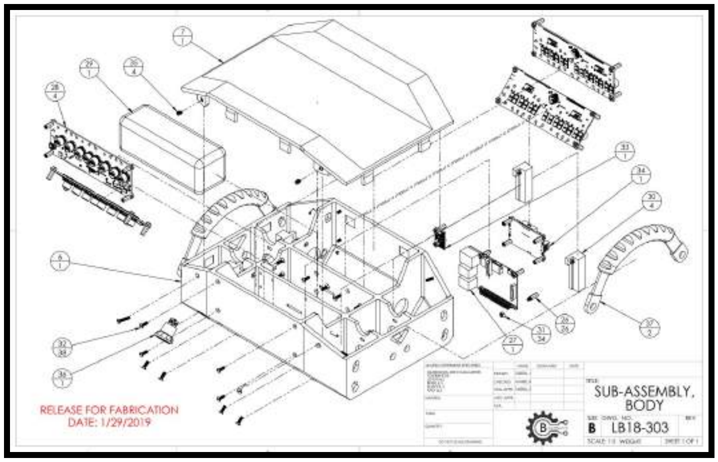


Figure 4.3. Body sub assembly exploded view.

4.1.4. Full Robot Assembly

The full assembly of the robot must be done in a sequential manner. The legs can be assembled and attached to the motors and mounts, which can then be attached to the body. Then, the body sub assembly can be conducted, fastening all wires and electronic components and directing the wires to their respective location. Finally, the cover can be latched on top to complete the assembly.

4.1.5. Software Development

Python 3 is the core language used on the software package of the quadruped robot. Multiple libraries are utilized in addition to some newly developed classes for the operation of the robot.

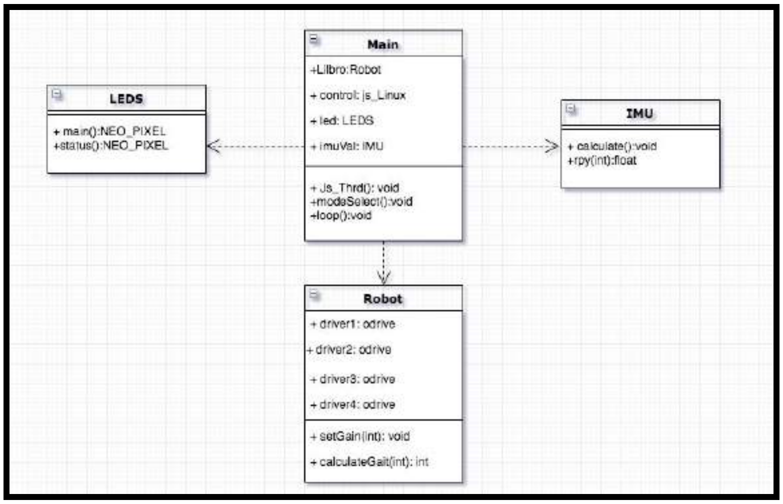


Figure 4.4. Software UML Class Diagram

There are 3 implemented classes which are instantiated in the main script. The main script utilizes these classes to read/write the necessary contents in order for the robot to work, and is what is executed when the robot is turned on. The robotic software will include common tasks such as feedback control loops, and will serve as the instructions for what tasks the robot shall execute.

4.2. Bill of Materials

The vital components when assembly Lil'Bro are as follows the raspberry pi 3 b+, the four Odrive motor drivers, the eight Odrive motors, the eight encoders, the voltage regulator, the 3D filament, and the battery. Each of these parts met the requirements specified by the team and can be seen in appendix C the test report under the subsection for visual tests. In Figure X you can clearly see the amount spent on material acquisition alone. When calculating the budget with the sponsor the cost of the necessary materials were discussed and calculated to be roughly around $2,000. Throughout the semester certain components had to be reordered, increasing the expenditure to $2,235.38 which remains vastly less than the currently used robot in the lab which has a value of $10,000. Overall there is a significant amount of savings, which is great for the team and the sponsor and can be seen in the figure 4.5 below.

Brobotics Inc. Purchased Bill of Materials						
ITEM NO.	QTY.	PART NUMBER	DESCRIPTION	MATERIAL	VENDOR/NOTES	PRICE
1	12	509	FASTENER, LEG MOUNT	PURCHASE	93897A542 MCMASTER-CARR	$8.45 /25
2	12	510	WASHER, LOCK	PURCHASE	3573A16 MCMASTER-CARR	$8.21 /100
3	24	511	NUT, HEX	PURCHASE	96537A109 MCMASTER-CARR	$8.63 /50
4	4	512	FASTENER, COVER	PURCHASE	96817A274 MCMASTER-CARR	$6.69 /25
5	8	513	MOTOR, DC	PURCHASE	D5065-270KV, ODRIVE	$69.00 EA
6	8	514	ENCODER, CPR	PURCHASE	CVLAMT102-V, ODRIVE	$39.00 EA
7	16	515	FASTENER, ENCODER	PURCHASE	90666A104, MCMASTER-CARR	$8.40 /100
8	28	516	HOLDER, CABLE	PURCHASE	7565K31, MCMASTER-CARR	$5.74 /25
9	32	517	FASTENER, MOTOR	PURCHASE	93705A824, MCMASTER-CARR	$7.25 /25
10	48	518	FASTENER, MOTOR HOUSING	PURCHASE	91292A114, MCMASTER-CARR	$4.72 /100
11	12	519	LOCKNUT, LEG-JOINT	PURCHASE	93795A205, MCMASTER-CARR	$10.94 /100
12	12	520	FASTENER, LEG-JOINT	PURCHASE	90278A375, MCMASTER-CARR	$3.79 EA
13	24	521	BEARING, THRUST	PURCHASE	6655K51, MCMASTER-CARR	$2.78 EA
14	24	522	BEARING, SLEEVE	PURCHASE	6679K33, MCMASTER-CARR	$2.31 EA
15	4	523	FOOT, SLIP-ON	PURCHASE	2517T71, MCMASTER-CARR	$6.64 /25
16	8	524	HUB, MOTOR	PURCHASE	2693, POLOLU	$6.99 /2
17	48	525	FASTENER, HUB	PURCHASE	92929A246, MCMASTER-CARR	$8.44 /100
18	26	526	STANDOFF, HEX	PURCHASE	SPC-NLB3/KIT1-1, AMAZON	$9.99 /60
19	1	527	COMPUTER, RASPBERRYPI	PURCHASE	3775, ADAFRUIT INDUSTRIES	$35.00 EA
20	4	528	DRIVER, MOTOR	PURCHASE	ODRIVE ROBOTICS	$129.00 EA
21	1	529	BATTERY, DC	PURCHASE	9067000073-0, HOBBY KING	$103.30 EA
22	4	530	BRAKE RESISTOR	-	INCLUDED W/ ITEM NO. 28	-
23	34	531	NUT, STANDOFF	-	INCLUDED W/ ITEM NO. 26	-
24	38	532	FASTENER, STANDOFF	PURCHASE	93840A126, MCMASTER-CARR	$5.55 /100
25	1	533	INERTIAL MEASURE. UNIT	PURCHASE	GY-521MPU6050, AMAZON	$5.79 EA
26	1	534	VOLTAGE REGULATOR	PURCHASE	SAINSMARTLM2596, AMAZON	$8.99 EA
27	36	535	INSERT	PURCHASE	94510A240, MCMASTER-CARR	$7.20 /50
28	1	536	SWITCH, POWER	PURCHASE	UR19B-2NOGNC, AMAZON	$10.78 EA
29	2	537	HANDLE, CARRYING	PURCHASE	ASIN: B072XHBKPJ, AMAZON	$9.99 /2
30	4	538	STRAP, BATTERY	PURCHASE	ASIN: B01MY257JD, AMAZON	$12.97 /10
31	12	539	ZIP-TIES	PURCHASE	ASIN: B07CTH2D8T, AMAZON	$3.99 /100
32	9	540	CONNECTORS, WIRE	PURCHASE	B07CWF8CRX, AMAZON	$13.99 /20
33	1	541	CONTROLLER, PS3	PURCHASE	ASIN: B07DGHY13L, AMAZON	$14.99 EA
34	8	542	FILAMENT, 3D PRINTER	PURCHASE	MP05778, MAKERBOT	$42.99 EA
35	1	543	WIRE, ELECTRICAL	PURCHASE	BNTECHGOTD-18, AMAZON	$18.98 /30FT
						$13.95 EA

APPROVAL DATE:	3-29-2019
PART COUNT:	512
TOTAL COST:	$2,238.38

TITLE: PURCHASED BILL OF MATERIALS
SIZE A DWG. NO. LB18-001 REV
SCALE: 1:1 WEIGHT: SHEET 1 OF 1

SOLIDWORKS Educational Product. For Instructional Use Only.

Figure 4.5. Bill of Materials

5. Prototype Tests

A variety of tests were conducted on Lil'Bro's in order to validate its design specifications. There are four design specifications that required testing, while the rest of the performance specifications were validated by visual observation. The specifications, or features, validated through testing are, mass, displacement volume, linear velocity, and additional weight bearing capacity.

5.1. Test Plan Summary

A variety of tests were planned out to be conducted on Lil'Bro to validate its design specifications. These test were carried out in the Robotics and Motion Laboratory under a control environment. There are four specifications that require testing, while the rest of the specifications can be validated by visual observation. Each test was conducted multiple times for repeatability and validation of the design specifications. Once the test plan was established, each test was conducted as planned.

5.1.1. Design Specifications

The tests performed will signify the validation of the required design specifications. The robot is to meet the following specifications:

- The robot shall be able to reach a speed of 0.2 m/s
- The robot shall be able to have a payload capacity of 25%
- The robot shall weigh less than 23 kg
- The robot shall be open source, with the source code licensed under GPL3

5.1.2. Performance Specifications

The performance specifications can be validated by visual observation. They are as follows:

- The robot can be controlled through a handheld device.
- The robot can move without external physical support.
- The robot can collect data in operation
- The robot's processor is capable of processing at 16 MHz.
- The robot's leg links can be interchanged between assemblies

5.2. Test Facility and Instrumentation

All testing was performed in the Robotics and Motion Laboratory located in room 2.216 of the Biotechnology Sciences and Engineering building, at the University of Texas at San Antonio's main campus. The robot was fabricated and assembled in this laboratory, and software development and testing have been carried out there as well. The laboratory is equipped with the necessary structural and power support devices to operate and test the unit in a safe manner. The laboratory provides a climate controlled environment, and a secure workplace that allowed for the storage of testing equipment and prevent any tampering.

5.2.1. Linear Walking Speed

The linear walking speed of the robot was calculated by setting distance parameters; with spaced out tags 20 cm apart each, for 100 cm, and timing when the robot finished the constrained distance. The test would be passed if the robot could finish the path in under 5 seconds, resulting in a speed of 0.2 m/s.

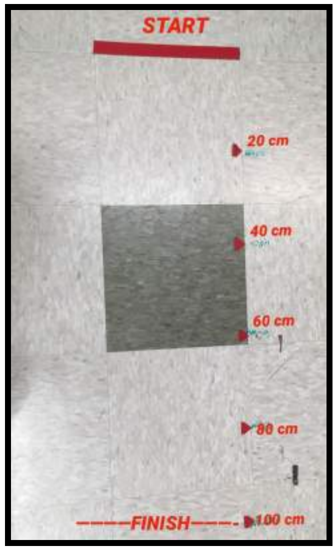


Figure 5.1. Linear Walking Speed Test Track

The test track can be seen in figure 5.1 above. After performing this test for a total of 10 trials, the average linear walking speed surpassed the hypothesized expectations and the robot qualified for meeting the linear walking speed specification.

5.2.2. Weight Carrying Capacity

The carrying capacity test was developed by assigning the robot the ability to carry 25% of its own mass. The development of the robot was composed of three phases; a monoped, biped, and quadruped robot. The biped phased allowed further development of the carrying capacity test; resulting in an unstable system which could only carry itself by a very small margin. This resulted in the addition of a 5:1 ratio gearbox to the leg configurations, supplying the required torque for this test.

The entire mass of the robot was calculated and the value acquired was 9.52 kg or 21 lbs, this mass was then multiplied by 25 percent which was calculated to be 2.384 kg or 5.25 lbs. This test was conducted by taking a container, and filling it with sand; small amounts of sand were then removed until the mass of the container and the sand were equal to the of 25 percent of the main assembly's total mass.

After installing the gearbox, the robot held this additional weight for at least eight seconds. This process was repeated for five iterations.

Figure 5.2. Weight Carrying Test being Conducted

5.2.3. Displacement Volume

The primary goal of this test is to validate that the dimensions of the robot are within a cubic meter box. The volume of the robot and the displacement of its parts were determined in this test. In order to verify that the robot's volume remains within the required specification of a cubic meter, a number of tests were conducted within the testing facility. With the use of an accurate measuring tape, a series of measurements were taken in order to verify that the volume of the robot remains within the bounds of a cubic meter. Different leg configurations were executed, tested and measured (L x W x H) as follows; the standby leg configuration (normal standing position), legs vertically stretched out (highest standing position), and legs extended out (forward and backwards of maximum limits during operation). All positions mentioned were static when measured, and set by a controlled program that verifies that the robot remains under a cubic meter. In all three positions the volume of the robot never once exceeded the set parameter of a cubic meter.

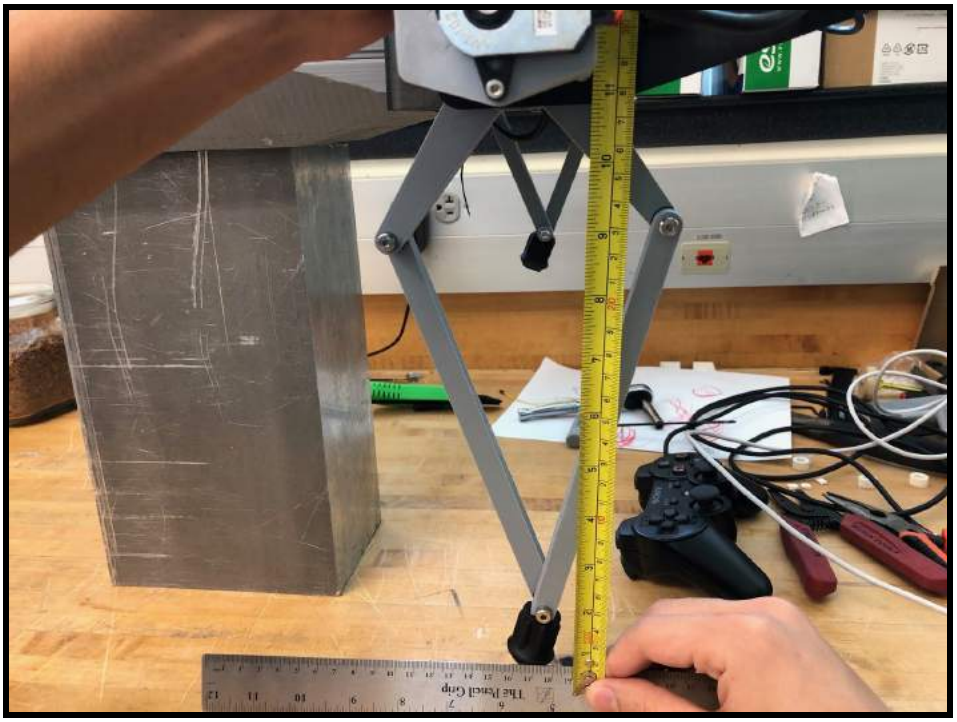


Figure 5.3 High Standing Leg Position.

5.2.4. Total Weight

Due to the weight capacity of the scale available in the testing facility, the sum of the individual sub-assemblies were used to test for this value. To prepare for this test, each of the leg subassemblies were removed from the body subassembly, and any loose wires were secured in such a way that they didn't interfere with the measurement. All the components that were used in the final operation of the robot were securely mounted in their respective sub-assembly before any measurements were made, this includes all wiring, wire fasteners, and battery. Each sub-assemblies mass were measured five separate times, and values measured and calculated. This test was performed in a controlled environment, where the impact from the ambient conditions did not affect the results.

5.2.5. Visual Inspection Tests

For the test in which the robot can be controlled via a handheld device, the operator must be able to control the robot through the use of a handheld device. This test was conducted by connecting a Dual Shock 3 controller to a raspberry pi 3 B+. This connection was established through a bluetooth connection.

Figure 5.4. Robot Control via Dual Shock 3 controller

Another visual test consisted of the robot being able to stand and operate without any form of assistance or support from a boom or other weight bearing supports. This allows for the robot to be operated and be tested in an uncontrolled environment.

Figure 5.5. Robot Operation Without Tether Support

The final visual test consisted of four forms of data collection; The raspberry pi must be capable of collecting and storing data on the motors position, motor speed, motor current draw, and the center of mass position in the system's memory. This data could then be viewed for performance monitoring as well as being utilized to configure the robot's source code. The data is saved into the "Data" folder and each data type is written into a text file. The data collected can be seen in figure below.

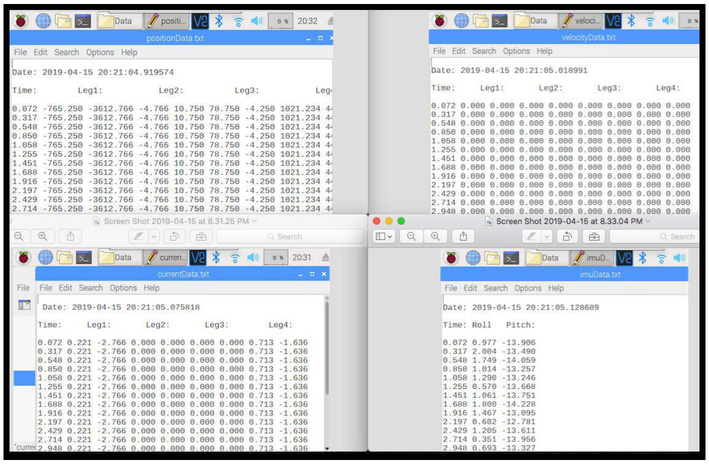


Figure 5.6. Data Collection For Each Data Type

5.3. Test Results & Testing Matrix

The test results yield the overall design quality by meeting required design specifications. The testing matrix contains the test results in order for analysis and validation of tests.

Table 5.1. Post-Testing Compliance Matrix.

Item No.	Feature/Specification	Specification Ref. in Appendix A	Feature/Specification Met (Y/N)
1	The robot can be controlled through a handheld device.	Section A.1.1	Y
2	The robot can move without external physical support.	Section A.1.2	Y
3	The robot moves by walking.	Section A.1.3	Y
4	The robot can support an additional 25% of its weight while standing.	Section A.1.4	Y
5	The robot can collect data in operation.	Section A.1.5	Y
6	The robot's processor is capable of processing at 16 MHz.	Section A.1.6	Y
7	The robot can move at a speed of at least	Section A.1.7	Y

	0.2 m/s.		
8	The robot's dimensions are within a cubic meter box.	Section A.2.1	Y
9	The robot weighs less than 23 kg.	Section A.2.2	Y
10	The robot's software is licensed under GPL3.	Section A.2.3	Y
11	The robot's leg links can be interchanged between assemblies.	Section A.2.4	Y

6. Project Management

From the beginning of SD1 a detailed Gantt chart was created using the syllabus for both SD1 and SD2. This was done in order to keep the team on schedule with the project delivery as well to assure the sponsor that the project will be handed over on the set date. The Gantt Chart was then modified at the beginning of SD2 since the original syllabus it was based off of was for the fall not the spring, a few changes were made such as the addition of new deliverables and the change of dates for those deliverables. The tasks for each deliverable was separated into two categories, programing/software and fabrication/design/mechanical assembly. The team was then split into their respective category whether it is software/programming or fabrication, design and mechanical assembly.

6.1 Personnel

The programming and software tasks were conducted by both Steven and Emiliano due to their vast amount of experience/knowledge in this field. The fabrication, design, and mechanical assembly was lead by John and assisted by Mario. As for the paperwork and the presentations that work was evenly separated between the group members. The financial and project management side was conducted by both Mario and Steven.

6.2. Overall Schedule

6.2.1. Gantt Chart

Initially when SD1 began the work was primarily paperwork and research, these tasks were separated evenly between the team. Towards mid semester the team decided to get ahead on the software and programming, the team decided that Steven and Emiliano were going to be in charge of the software and programming of Lil'Bro. On the other hand John had begun designing the robot while Mario began researching on the fabrication of these necessary parts. At the end of SD1 the team had a 3D printed mount and legs as well as the ability to control the motors operating the legs. Throughout SD2 the team already had an understanding what tasks pertained to each group member as well as which sections of the report they had to write. John had begun assisting in the fabrication of parts in order to speed up the process and have more parts to test on. Mario was still fabricating as well as keeping track of the teams tasks and time spent on each task. Steven and Emiliano were working together on software development in order to achieve the set specifications of Lil'Bro.

6.3. Percent Completions of Tasks

The tasks conducted throughout SD1 were 100% completed at the beginning of SD2. A few redesigns and revisions were done to the fabrication package over the christmas break. As for SD2 there are three major deliverables still to complete. The deliverables are as follows the tech symposium, progress report 4 and the final binder. The deliverables before these three are 100% completed, this includes the poster for tech symposium, the test report, test plan, operations manual, fabrication package, thematic outline and in class presentations. Figure 6.1 displays the recently completed deliverables.

13	Draft Poster & Final Review (Due 4/9)					
13.1	Draft Poster development	Mon 4/8/19	Wed 4/10/19	3	100%	3
13.2	Draft Final Review Development	Mon 4/8/19	Wed 4/10/19	3	100%	3
13.3	Walking Algorithm Development	Mon 4/8/19	Wed 4/10/19	3	100%	3
14	Test Report (Due 4/16)					
14.1	Walking algorithm Development	Wed 4/10/19	Sun 4/14/19	5	100%	3
14.2	Weight Test	Thu 4/11/19	Thu 4/11/19	1	100%	1
14.3	Dimension/Volume Test	Thu 4/11/19	Thu 4/11/19	1	100%	1
14.4	Carrying Capacity Test	Fri 4/12/19	Fri 4/12/19	1	100%	1
14.5	Walking Test	Fri 4/12/19	Mon 4/15/19	4	100%	2
14.6	Processing and Data collection test	Sat 4/13/19	Sat 4/13/19	1	100%	0
14.7	GPL3 License Verification	Sat 4/13/19	Sat 4/13/19	1	100%	0
14.8	Trouble shooting the Robot	Sat 4/13/19	Mon 4/15/19	3	100%	1
14.9	Development of Test Report	Thu 4/11/19	Mon 4/15/19	5	100%	3
14.10	Review of Test Report	Tue 4/16/19	Tue 4/16/19	1	100%	1
15	Final Project Poster (4/18)					
15.1	Review of Draft Final Poster	Wed 4/17/19	Wed 4/17/19	1	100%	1
15.2	Development of Final Poster	Thu 4/18/19	Thu 4/18/19	1	100%	1
15.3	Completion of Final Poster	Fri 4/19/19	Fri 4/19/19	1	100%	1
16	Draft Final Report (Due 4/19)					
16.1	Concept Design	Mon 4/15/19	Mon 4/15/19	1	100%	1
16.2	Prototype design and Fabrication	Mon 4/15/19	Mon 4/15/19	1	100%	1
16.3	Software	Mon 4/15/19	Mon 4/15/19	1	100%	1
16.4	Earned Management	Mon 4/15/19	Mon 4/15/19	1	100%	1
16.5	Development of Final report	Mon 4/15/19	Fri 4/19/19	5	100%	5
16.6	Revising final report draft	Fri 4/19/19	Fri 4/19/19	1	100%	1

Figure 6.1. Gantt Chart

6.4. Personnel Assignments

At the formation of the team, the team lead Steven had asked each member what their strengths and weaknesses were. From that information he spoke with the team and assigned each member with a section of the overall project. Steven would lead the software design with Emiliano assisting him and creating the walking algorithm for the robot which was the most crucial specification for the team. John lead the design and mechanical assembly of the robot with Mario assisting him. Mario lead the fabrication as well as the project management which includes the financial standpoint of the team and the assurance of keeping the project on track.

7. Financial Performance

The earned value provides a method for measuring the project's performance. It compares the amount of work that was planned versus what was accomplished to determine if cost and schedule performance is as planned. Earned Value is separated into three different elements, Budgeted Cost of Work Scheduled (BCWS), Budgeted Cost of Work Performed (BCWP), and the Actual Cost of Work Scheduled (ACWS). Once calculating the three key elements of an earned value management, they can be used to calculate the Cost Performance Indicator (CPI) and the Schedule Performance Indicator (SPI). The Cost Performance indicator was calculated using the Budgeted Cost of Work Performed over the Actual Cost of Work Performed while the

Schedule Performance Indicator was calculated using the Budgeted Cost of Work Performed over the Budgeted Cost of Work Scheduled. The CPI and SPI assists the team in determining whether or not the team is using both the time and resources allotted to them appropriately.

7.1. Overall Planned Cost & Time vs. Actual Cost & Time

As mentioned previously the BCWS is the budgeted cost of work performed throughout both SD1 and SD2. This is the the hourly wage of an engineering which was given to be $100, this would then be multiplied by the amount of originally scheduled time for each task at the end of every week. On the other hand ACWP is the actual cost of work performed by the team. This was calculated with the same hourly wage of $100 except in this case it was multiplied by the actual amount of time it took the team to complete the task during the week. Figure 7.1 represents the BCWP compared to the ACWP.

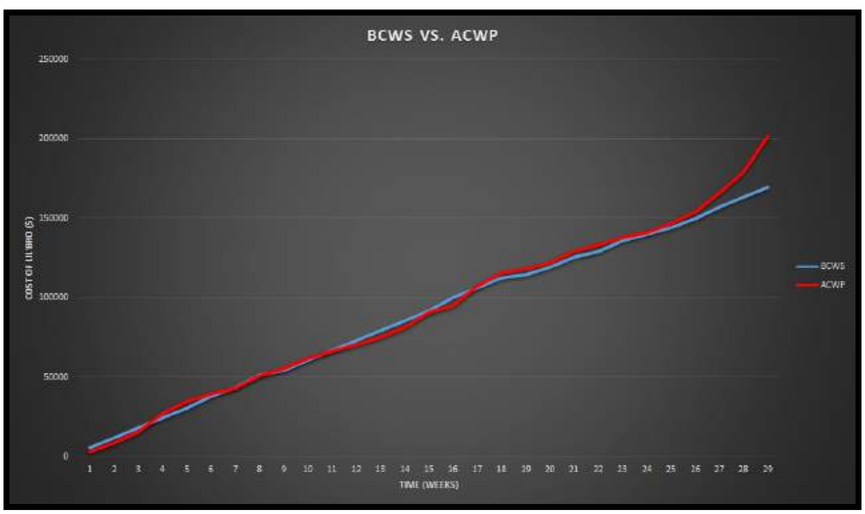


Figure 7.1. BCWP vs ACWP

7.2. Planned Labor Cost per task vs. Actual Labor Cost per task

As mentioned previously the BCWP is the budgeted cost of work performed throughout both SD1 and SD2. This is the Budget at Completion (BAC), which was calculated to be $200,000, on top of this value the team added an additional two thousand dollars due to material cost as well as being the actual budget of the project. The BAC was then multiplied by the percentage of work to be completed by the end of that week. On the other hand ACWP is the actual cost of work performed by the team. This was calculated with the same hourly wage of $100 except in this case it was multiplied by the actual amount of time it took the team to complete the task during the week. Figure 7.2 represents the BCWP compared to the ACWP.

Figure 7.2. BCWP vs ACWP

7.3. Planned Materials cost vs. Actual Materials Cost

The original planned material cost for this project was discussed with the professor, and it was concluded to be two thousand dollars. Over the course of both SD1 and SD2 there were certain parts that had to be reordered as well as the robots components having to be reprinted due to errors with the printer or the tolerances were to great. With every issue the actual final material cost was calculated to be $2,235.38. So the team was over the budget of $2,000 by $235.38. This was not an issue since the overall project is still greatly less in price than the robot currently used in the lab which has a value of $10,000.

8. Conclusion

Brobotics Inc. has successfully designed, built, and tested an open source quadrupedal robot for the RAM Lab. The test results suggest that all of the set specifications of Lil'bro are met. This indicates that the robot is research ready once handoff is complete. Future work using lil'bro includes implementing more agile locomotion through dynamic gaits, as well as dynamic stabilization.

9. References

[1] Bhounsule, Pranav A. Pusey, Jason. Moussouni, Chelsea. "A comparative study of leg geometry for energy-efficient locomotion".

[2] Cross, Rod. "Standing, walking, running, and jumping on a force plate". Physics Department, University of Sydney, Sydney, New South Wales 2006, Australia.

10. Appendices

Appendix A: Operations Manual

Nomenclature

Symbols/Acronyms	Definition
l_1	**The upper leg link length**
l_2	**The lower leg link length**
l	**The length of the fifth imaginary leg link**
θ_1	**The angle between the horizontal and rightmost upper leg link**
θ_2	**The angle between the horizontal and the leftmost upper leg link**
m	**Meter in length, width and height**
kg	**Kilogram**
$N \times m$	**Newton meter**
AWG	**American Wire Gauge**
$GPIO$	**General Purpose Input/Output,**
IMU	**Inertial Measurement Unit**
SCL	**The serial clock line**
SDA	**The serial data line**
USB	**Universal serial bus**
$HDMI$	**High-Definition Multimedia Interface**

1. Product Description

1.1 Product Background

The objective of this project is to design, build, and test an open source quadrupedal, or four-legged, robot for UTSA's Robotics and Motion Laboratory (RAM Lab). The robot, otherwise known as Lil'Bro, will be used by researchers at the RAM Lab for ongoing agile gait locomotion research. The manufacturer of the robot currently used in the lab does not provide the robot's source code with the purchase of the robot. The robot is also expensive and hard to maintain, which is why the Brobotics Inc. team aims to minimize the cost of and maximize accessibility to the robot's design. An online Github repository for the project will be created to allow current and future users to easily access the robot's source code. The team intends to further increase accessibility to users by designing the fabricated parts of the robot to be 3D printed, which has not been done by any other providers of similar-sized robots.

1.2 Document Scope

This document was prepared based on guidelines set forth by course ME 4813 Senior Design II at the University of Texas at San Antonio. For any questions regarding the subject material in this document, refer to Section 5.2 for contact information for the Brobotics Inc. team.

1.3 Product Overview

The overall design of Lil'Bro can be seen in Figure 1.1. below.

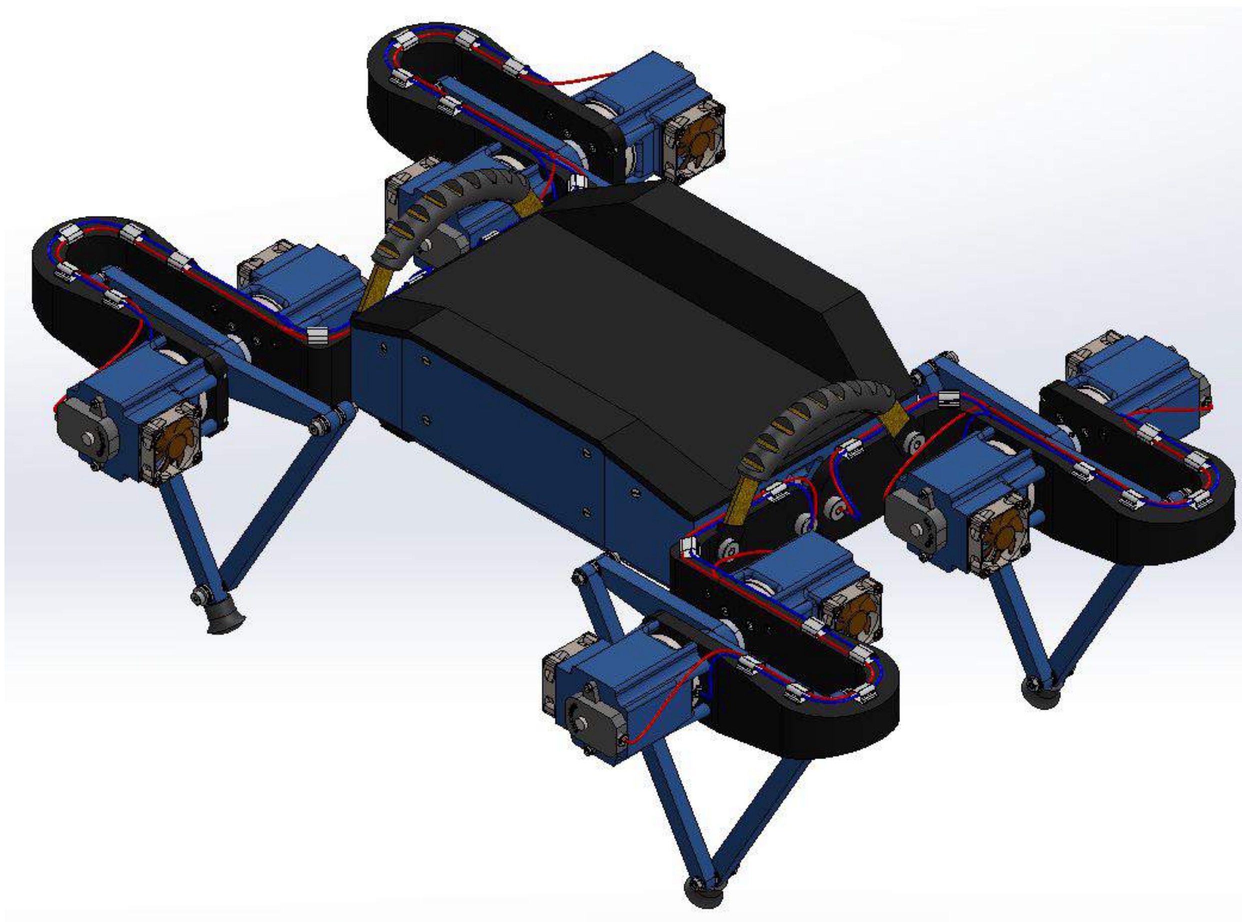


Figure 1.1.. Lil'Bro Complete Assembly.

Lil'Bro will can walk via user interface from a remote controller. The unit's estimated dimensions and mass are shown below in Table 1.1 below.

Table 1.1. Estimated profile dimensions and mass of unit.

Length	0.69 m
Width	0.57 m
Height	0.22 ± 0.08 m
Mass	9.5 kg

There are a total of five sub-assemblies, two left leg sub-assemblies, two right leg sub-assemblies, and a body sub-assembly. These assemblies can be seen in the figures below.

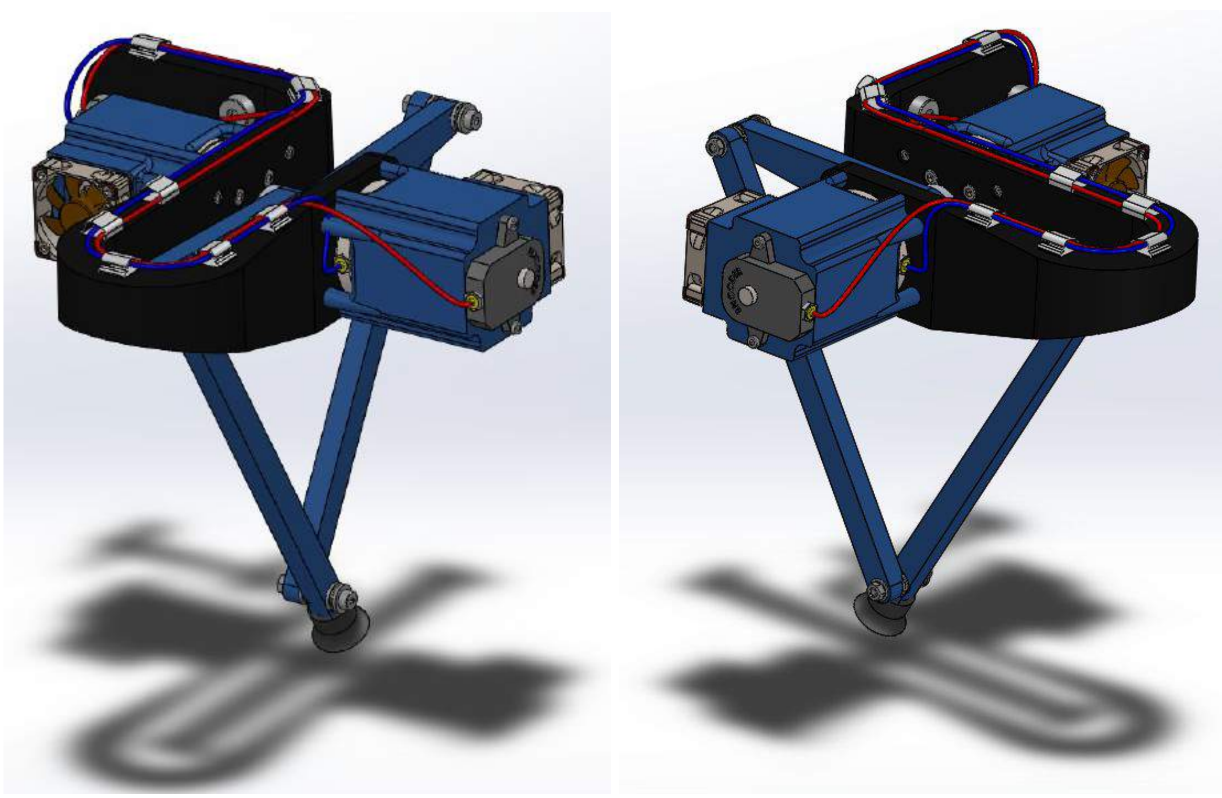


Figure 1.2. Right and Left leg sub-assemblies.

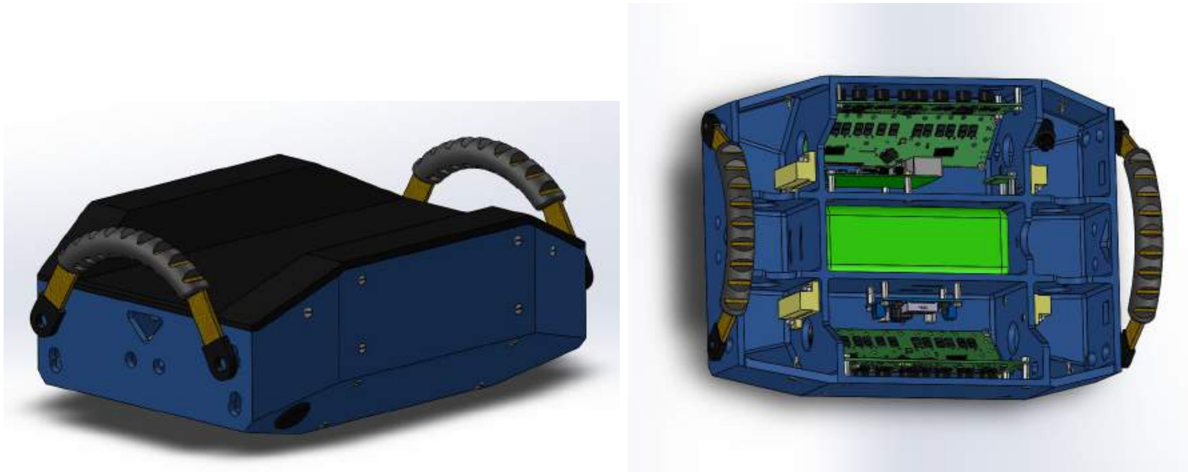


Figure 1.3. Body sub-assembly (External, and internal views).

The four leg sub-assemblies will attach to the body sub-assembly, which will house the necessary components to control and power the legs, these components are listed in section 2.2 of this manual. The leg sub-assemblies are designed for easy removal from the body sub-assembly, this allows for easy assembly of each leg while isolated from the main assembly. The leg sub-assemblies are connected to the body with three nylon shoulder screws, these screws are also used to attach two carrying handles to the assembly, for lifting and transporting the unit.

2. Theory of Operations

2.1 Leg Mechanism and Locomotion

Lil'Bro is propelled by four leg sub-assemblies attached to a body sub-assembly. Each leg is powered independently, and the position and motion relative to the other legs will be controlled via user interface.

Each leg is powered by two motors, that can provide a potential 2.14 Newton meters of torque each. These motors will be attached to the units body using a fabricated motor mount, seen in Figure 2.1 below.

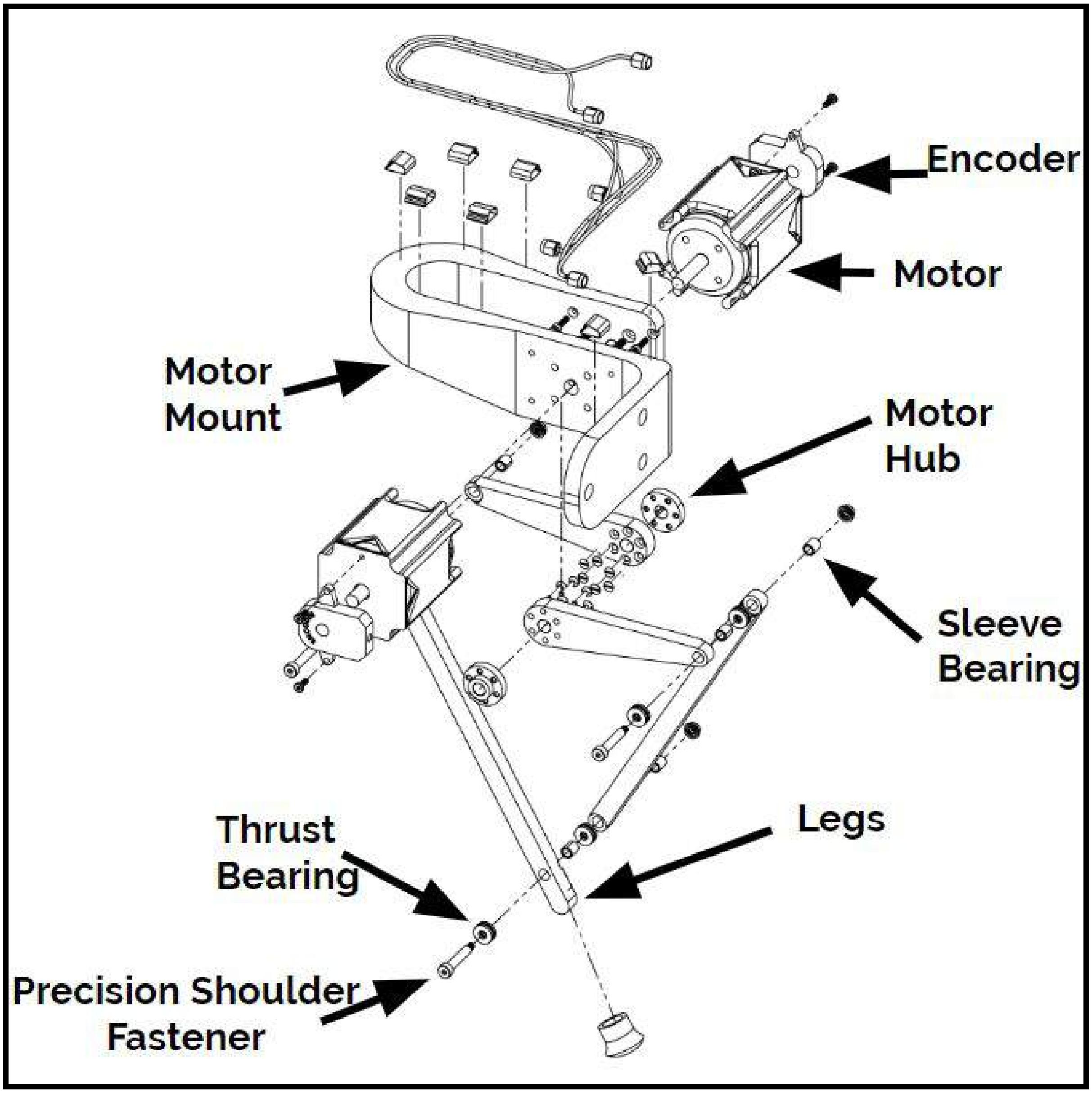


Figure 2.1. Parts included in a leg sub-assembly.

The shafts from the motors will be used to support the legs directly, this is done by using a hub to connect the upper leg sections to each motor shaft. The hubs will also transfer the torque from the motors rotors to the upper leg sections. The upper leg sections of the leg sub-assembly are connected to the lower leg sections using a shoulder screw that runs through sleeve bearings that are pressed into the leg sections; the lower leg sections will be connected using the shoulder screw as well. The shoulder screws will allow the legs to rotate freely at their locations, seen in Figure 2.2. To securely fasten the leg sections to each other without impeding the free rotation, thrust bearings are placed on the shoulder screw between the leg sections. Each leg is comprised of four leg sections, two upper leg section and two lower leg sections. One lower leg section is longer than the other, this will provide a surface area free of moving parts that will act as a foot for the unit. The upper and lower sections attached to each motor are offset from those on the other motor, this allows for a large range of motion without interference. The design of the leg will theoretically allow for the fabrication of leg sections that are low mass, and will not require designated actuators for the knee joints to control the position of the foot.

The position of the foot relative to the fixed position of the motors will be controlled by applying torque from the motors to the upper leg sections, as seen in the figure below.

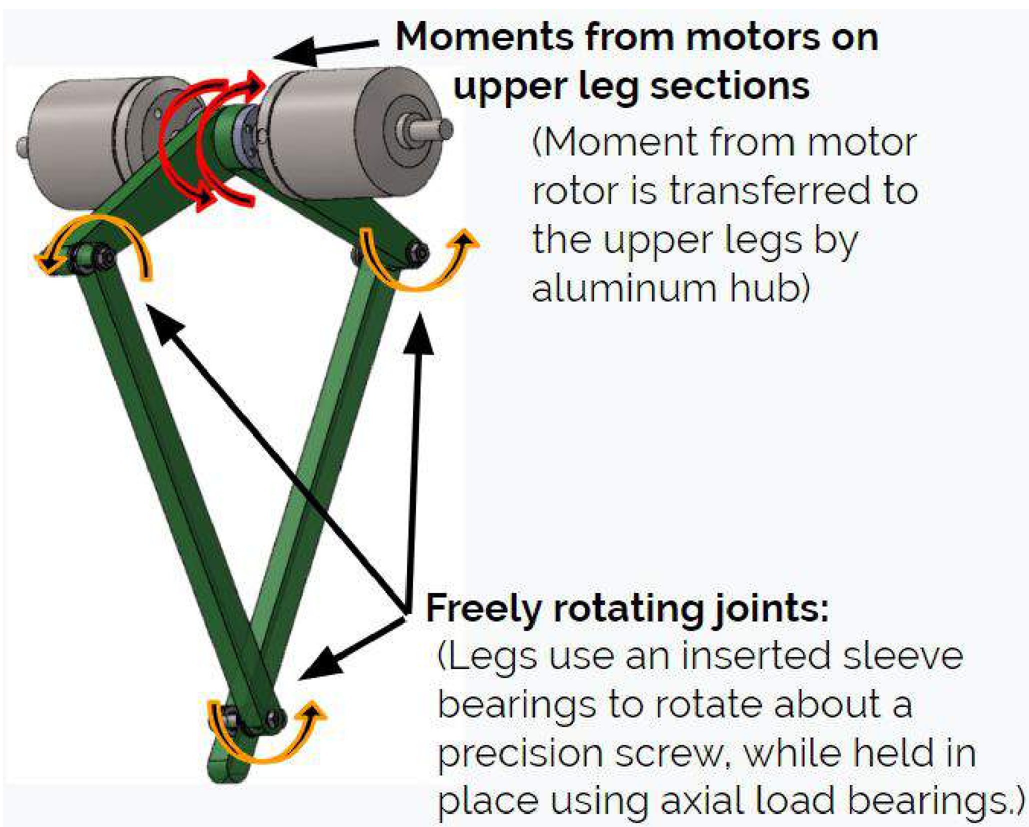


Figure 2.2. Mechanical operation of leg sections.

Controlling the angle of each upper leg section relative to a fixed datum will theoretically allow for the accurate placement of the foot. When the unit is in the standing position and not walking, the weight of the unit will create a moment force on the motor rotors, this moment will be matched by an equivalent torque output from the motors. This will theoretically allow the unit to stand at a fixed height and position.

The unit will walk by controlling the angle of each upper leg section relative to a fixed datum. These angles can be seen in the figure below.

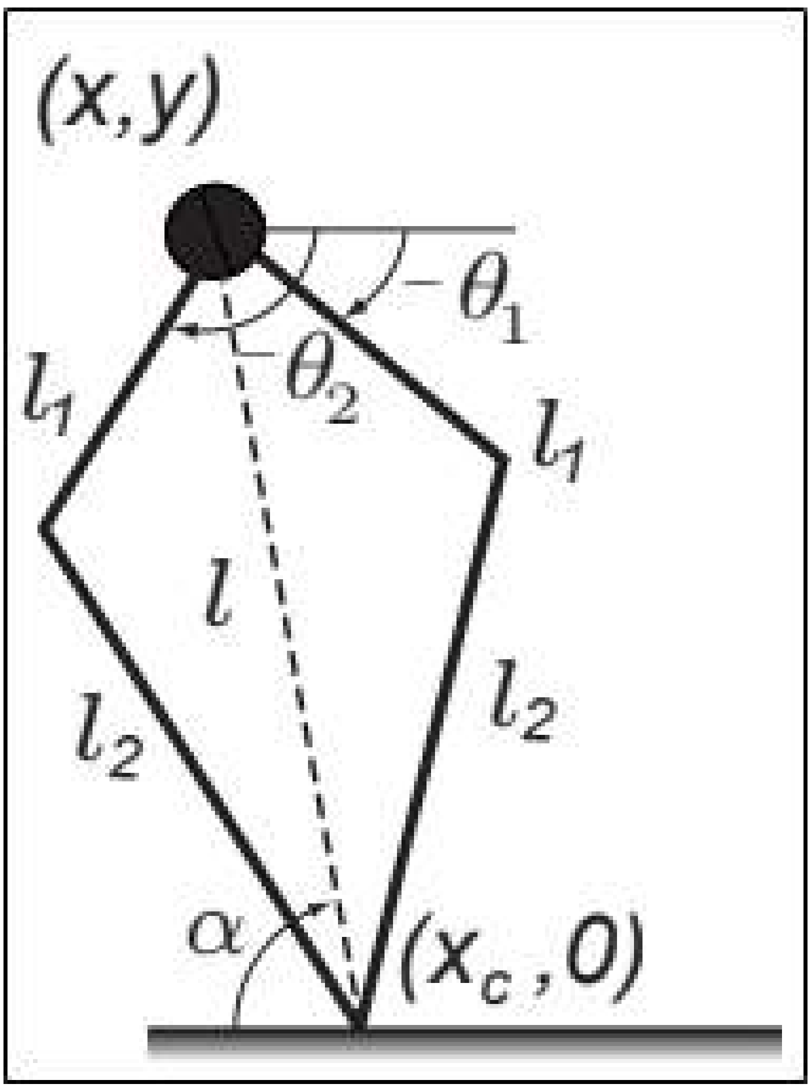


Figure 2.3. Free body diagram of leg.

Using this method of controlling the foot position and speed, the height of a leg can be decreased by increasing the angle of one upper leg section while decreasing the angle of the other. Similarly, the entire leg can be moved forward by decreasing the angle of both upper leg sections simultaneously. These methods of leg motion will be used to make each leg move and theoretically allow the unit to walk.

A paper by Bhounsule, Pusey, and Moussouni[1] analyzes the symmetric five bar leg configuration, alongside two other configurations, however, they are not of interest for this robot. Figure 2.3 shows a schematic of the symmetric five bar leg configuration.

A walking gait will comprise of three legs propelling the unit in a forward or reverse direction, while the fourth leg elevates and cycles from a predetermined rear position to a predetermined forward position. The relatively lightweight legs will theoretically allow the motors to elevate and transition the legs back to the predetermined forward position in a short period of time, keeping the weight of the unit evenly distributed.

2.2 Electrical System

Lil'Bro is composed of the following electrical components:

Table 2.1. The Component's functionality in Lil'Bro.

Component	**Function**
4 x ODrive Motor Drivers	● Provides position, velocity, and current control to two motors. ● Receives position and velocity feedback from the encoder attached to the motors ● Interfaces with Python
8 x 3-Phase Brushless DC Motors	● The motors are attached to the upper legs of the robot. The alternating current through the 3 phases of contact

	produce rotation about the shaft causing the legs to move. ● Includes a built-in thermistor which sends an analog signal of motor temperature readings to the driver.
4 x Brake Resistors	● Used for braking energy dissipation
8 x Noctua NF-A4x10mm 5V Fans	● Will receive a digital input from the Driver once motors reach a certain temperature and will begin to cool the motors down.
8 x Encoders	● Will provide feedback to the drivers on position and velocity of the motors via analog signals.
1 x 22.2 V Battery	● Provides 22.2 Volts and 4000mAH, which powers the entire robot.
1 x Power Switch	● Initializes/Shuts off the robot.
1 x Voltage Regulator	● Serves as a step down from 22.2 Volts of input to 5 Volts of output.
1 x Raspberry Pi 3 Model B	● Considered as the brain of Lil'Bro, hosts the main Python script that communicates with the 4 drivers.
1 x Inertial Measurement Unit (IMU)	● Contains a 3 axis gyroscope and 3 axis accelerometer. ● For the gyro, the 3 axes are a pitch, a roll, and a yaw. ● For the accelerometer they are x, y, and z axes. ● All six axes are analog inputs that are returned to the Raspberry Pi.

As mentioned in the previous section, each leg assembly will include the following components: 2 ODrive motors, 2 Noctua fans, 2 Encoders, 1 brake resistor, and 1 ODrive board. The ODrive boards and brake resistors are not located on the leg itself, but each leg is connected to two motors, a driver, and brake resistor. The ODrive board of each leg assembly will then connect to the Raspberry Pi via USB. Figure 2.4, shown below, is a circuit diagram of the entire system.

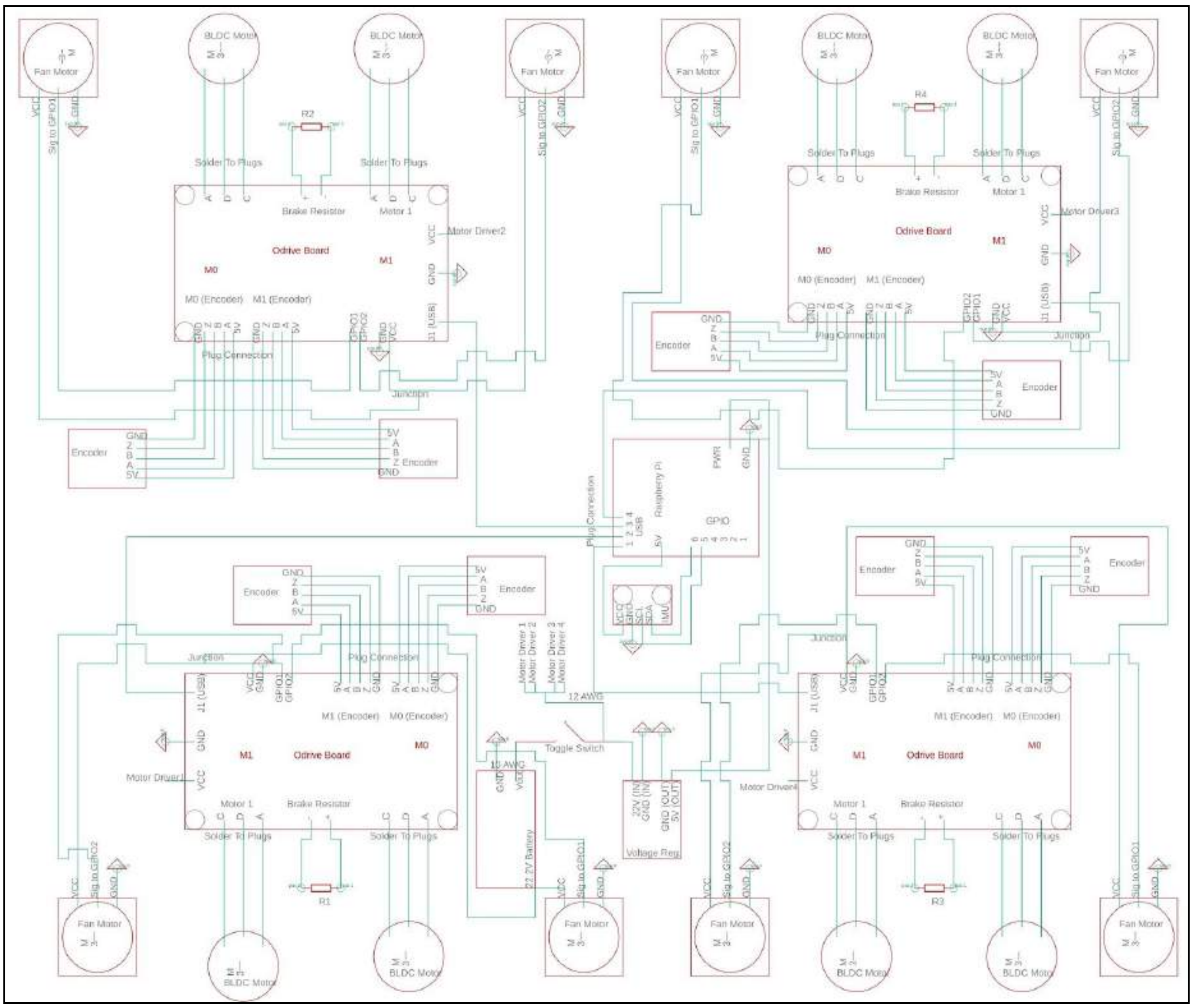


Figure 2.4. Circuit Diagram of Electrical Components.

The following procedure will describe how the wires for one leg assembly will be connected, this process can be repeated for the remaining three leg assemblies. Figure 2.5 shows the primary connections between the battery, power switch, voltage regulator and the 4 ODrive boards.

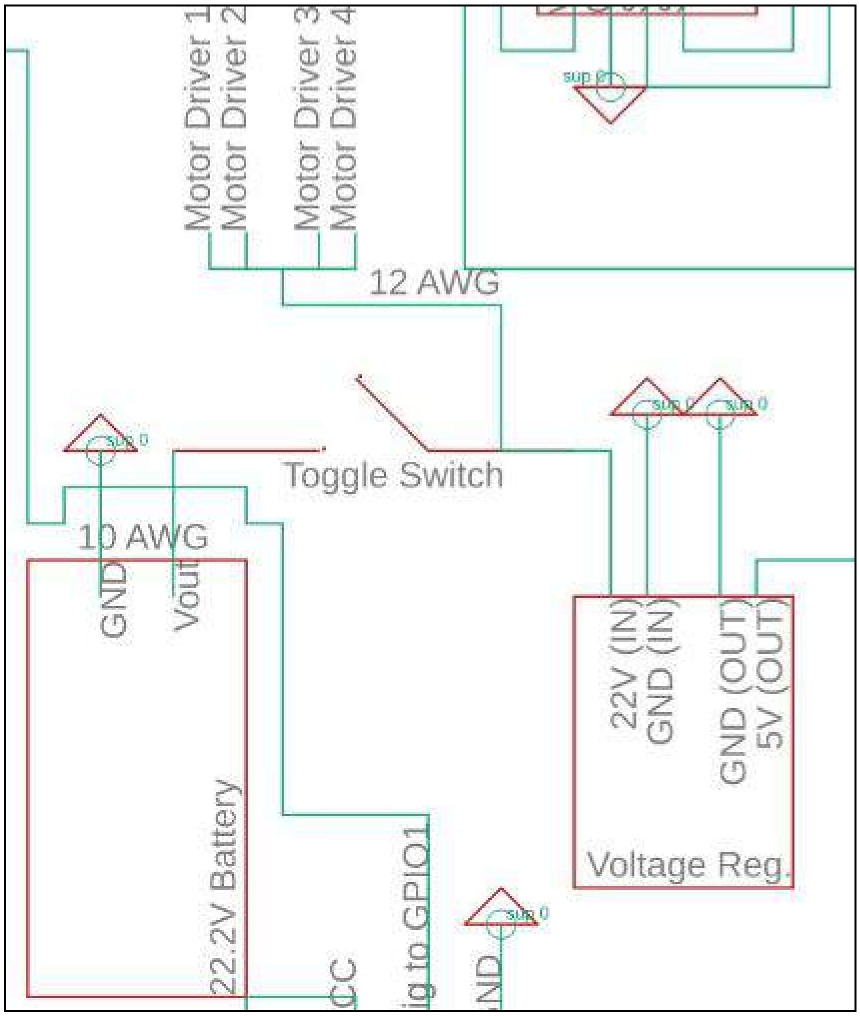


Figure 2.5. Battery, power switch, voltage regulator and ODrive board wiring.

The battery's two leads connect to the batteries positive and negative terminals, or voltage out and ground. The voltage out will then connect to the power switch. The other end of the power switch will be connected through 12 American Wire Gauge (AWG) wire which splits into two different connections. The first part of the wire provides power to the voltage regulator, and the voltage regulator then provides 5 V to the Raspberry Pi. Each connection is grounded separately.

The switch also controls the flow of current to the four drivers by connecting to the 10 AWG wire that carries current from the battery to the four drivers. Once current flows from the power line through the 5 wire junction to each motor driver, it is redistributed by the ODrive board to the leg assembly through the following connections, shown in Figure 2.6.

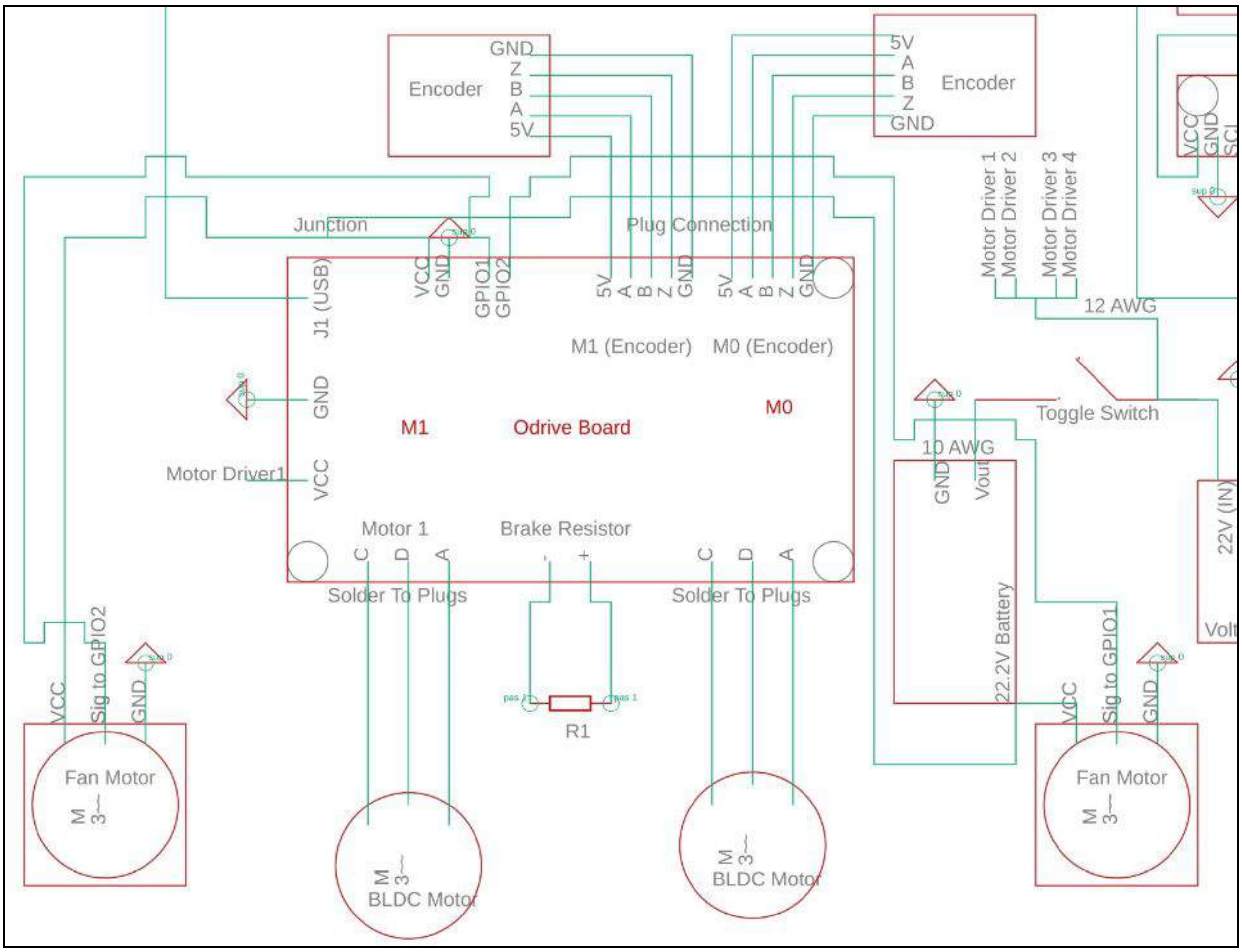


Figure 2.6. ODrive to leg assembly connections.

The ODrives connect to the motors through the plugs labeled ADC which are on the left and right sides of the brake resistor. The three connections represent the three phases of each motors and there is no particular manner in which they should be connected to the motor driver. The brake resistor uses an 18 AWG to connect to the driver. The signal input for the fans will be connected to GPIO pins 1 and 2 using a male to male 20 AWG jumper wires. The power for both fans will come from the connection of the 3.3 volt output of the driver. The encoders will be connected to the driver using the pins VCC, A, B, Z, and GND. This will be accomplished by a male wire of 20 AWG which connects to the female input of the driver. Lastly, the connection between the raspberry pi and the driver is shown below in Figure 2.7.

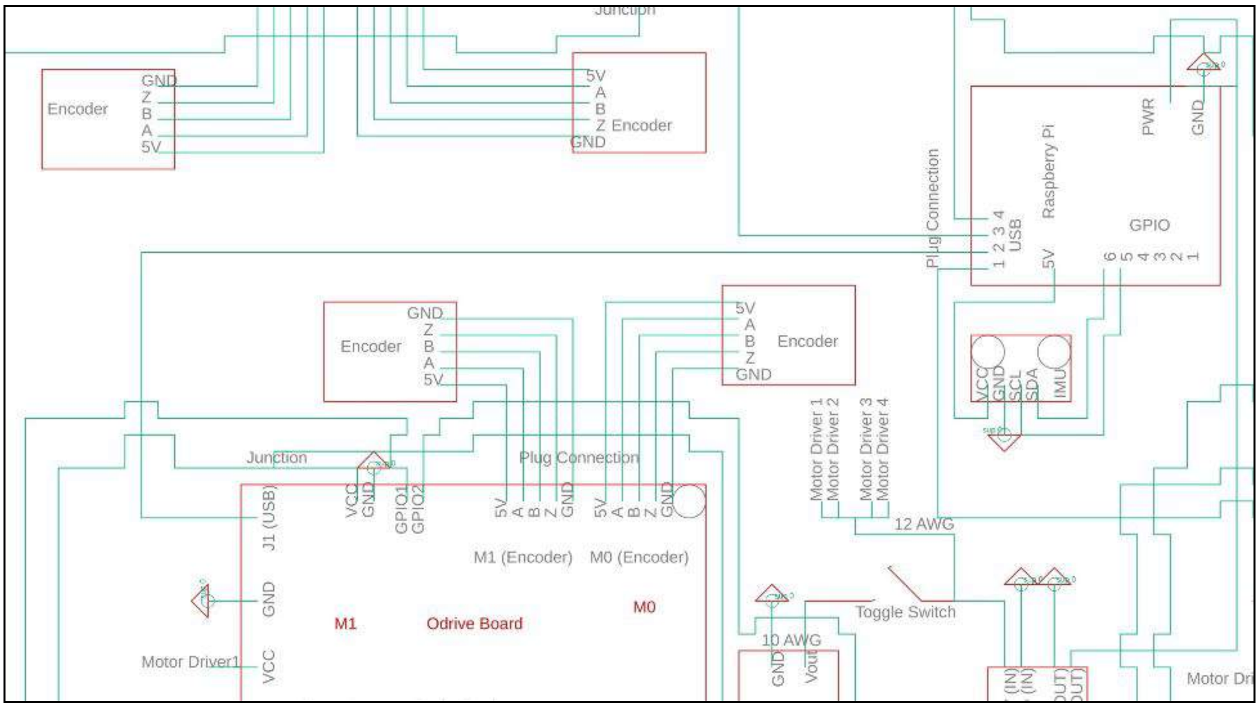


Figure 2.7. Raspberry Pi and IMU connection.

The Raspberry Pi is connected to each driver via USB. This allows the raspberry pi to communicate with the drivers at a higher baud, and still communicate with other components. An inertial measurement unit (IMU) is also connected to the pi. The IMU is supplied 5 volts from and grounded by the Raspberry Pi, and GPIO 5 & 6 will serve as the analog signal inputs from the SCL and SDA pins to the drivers. The signals will be that of the gyroscope and the accelerometer readings.

2.3 Software

The python programming language is implemented on Lil'Bro for operation; This programming language offers an extensive framework for agile robotic software development. The Raspberry Pi is capable of hosting a Python script for the operation of the robot. Multithreading is a process wherein a program executes two threads simultaneously as opposed to the conventional sequential execution. This allows the threads to share computing and processing resources with one another, resulting in more efficient use of resources and faster runtime. It is used in the software of Lil'Bro to receive inputs from the handheld controller (controller thread) while executing the walking algorithm (main thread). The operation of the robot can be summarized in the following block diagrams, where Figure 2.8 shows the main thread and Figure 2.9 shows the secondary controller thread.

The host script begins by turning the robot on, and the input and output interface is initialized. This establishes a connection between the Raspberry Pi, ODrive boards, and the Dualshock controller. The script is then split into the two threads, receiving inputs from the controller thread while executing commands in the main thread.

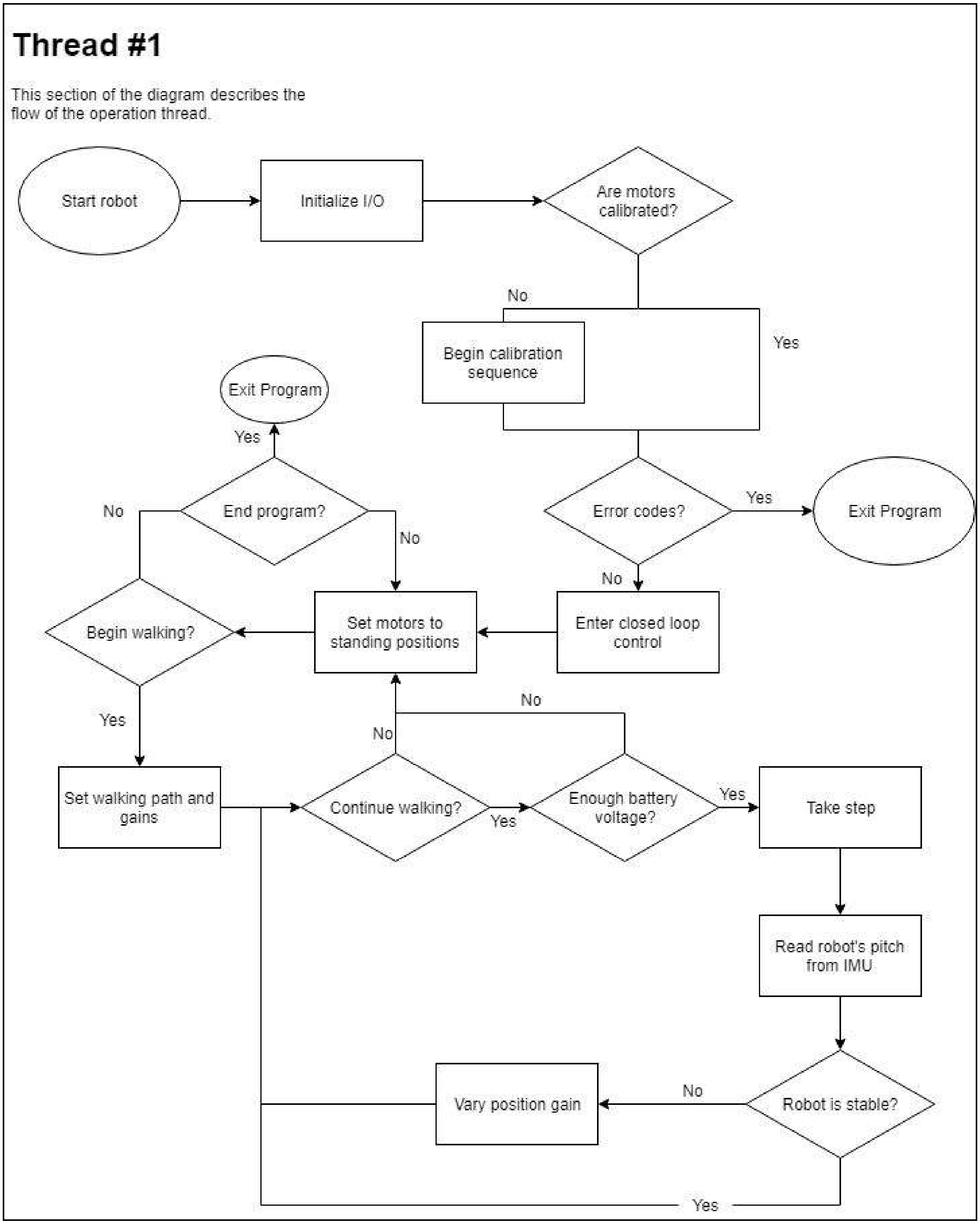


Figure 2.8. Block diagram of the main thread.

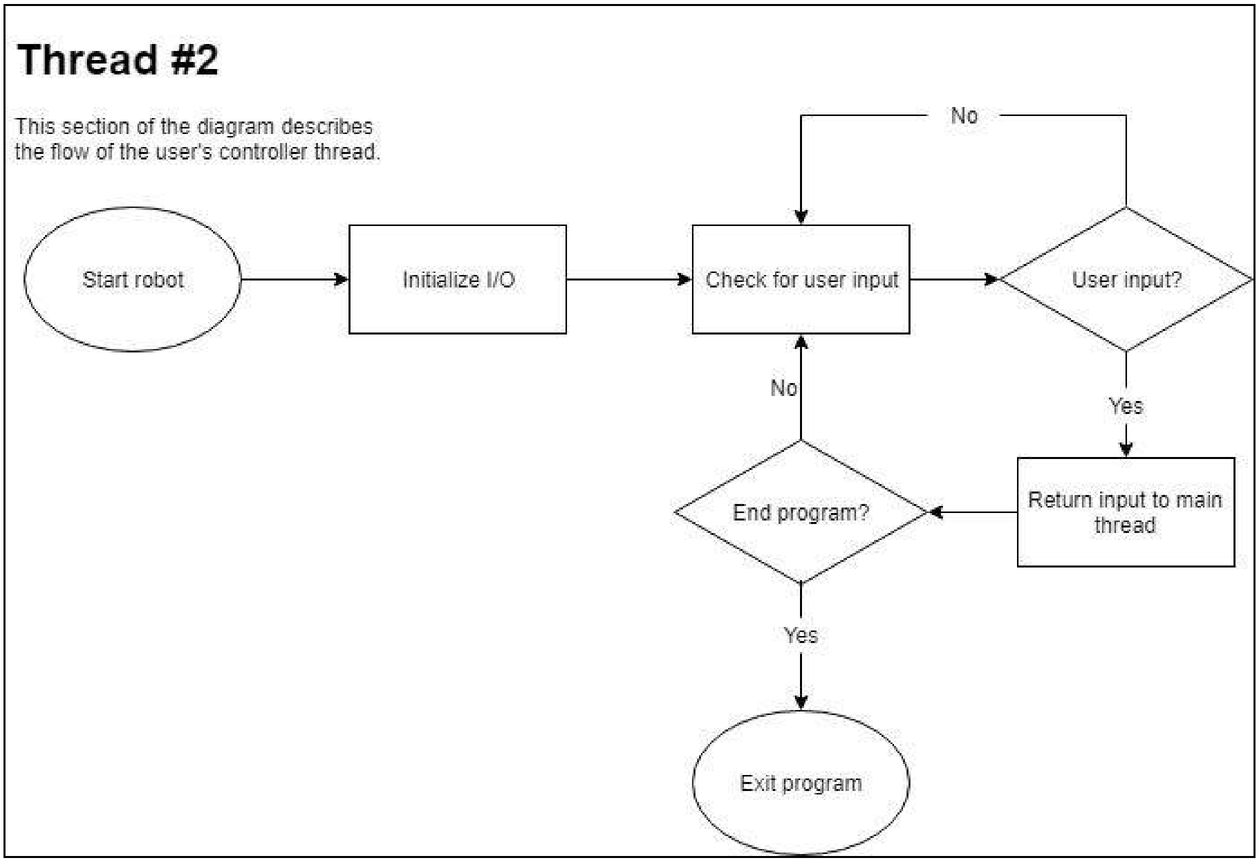


Figure 2.9. Block diagram of the input controller thread.

The main thread executes the calibration sequence once the command is received from the controller thread. For more information regarding the calibration sequence, refer to docs.odriverobotics.com/commands.

Following calibration, the program checks for any error codes that may have been generated. If error codes are not detected, the program enters closed loop control mode when signaled from the user. If an error code is present, the motor(s) that triggered the error no longer respond(s) to commands from the host. The program is terminated for this reason, and Section 4, Troubleshooting, discusses how to clear the present error codes.

Once in closed loop control, all motors are set to operate in position control, which is a closed loop control mode in which motors are set to positions through their encoders. Motors will attempt to remain at the set position, responding to disturbances by applying current in the opposite direction of the disturbance. Standing is effectively achieved through this control mode because the motors are set to the desired positions which result in the robot standing and independently remaining at those positions. For information regarding the closed loop control mode the ODrive motor drivers use, visit docs.odriverobotics.com/control.

The robot then enters a standby mode, wherein standing is maintained until a user's input is detected. The expected input is for activating the walking algorithm, otherwise, the robot will remain in place. If the input indicates the execution of walking, the walking path is then set according to the position controller's proportional gain. Before taking a step, the program checks if the user has entered an interruption command, which stops the execution of the walking algorithm and returns the motors to standby mode.

If the user wishes to continue with the execution of the walking algorithm, the remaining battery voltage is checked. If it is above the allowable voltage of operation, the walking algorithm executes. If not, the robot returns to standby mode. Instructions on charging the battery are included in Section 3, operating and setup instructions.

With the path and gains set, the robot takes a step. The robot evaluates its stability once it has recovered through its inertial measurement unit (IMU). The IMU's pitch reading is evaluated over the course of landing and the subsequent stabilization. Furthermore, the change in the pitch of the robot from zero after the completion of a step. If the pitch variation is greater than a preset threshold, undesired vibrations may be experienced by the robot that are due to lack of sufficient motor stiffness. This can be attributed to the need for increasing the proportional gain of the position controller. Once each of the four legs takes a step, a single cycle is complete. The robot evaluates the performance of the cycle using the measured pitch, examining the points within the cycle that caused the highest variation in the pitch. The controller(s) of the leg(s) causing the variation at the points of interest is addressed by the script, increasing the proportional gain(s) accordingly.

The script loops until interrupted by the user or by the battery voltage condition. Once interrupted by the user, the robot can be set into walking mode again through the appropriate input. Otherwise, the robot will remain in standby. While in standby, the user is able to place the motor drivers in Idle mode, which removes all torque stiffness from the motors and allows them to freely rotate.

3. Operating and Setup Instructions

3.1 Setup Instructions

The initial setup instructions are shown below and are as follows:

1. Set the robot's body position to the starting (laying down) position as shown in the figure below.

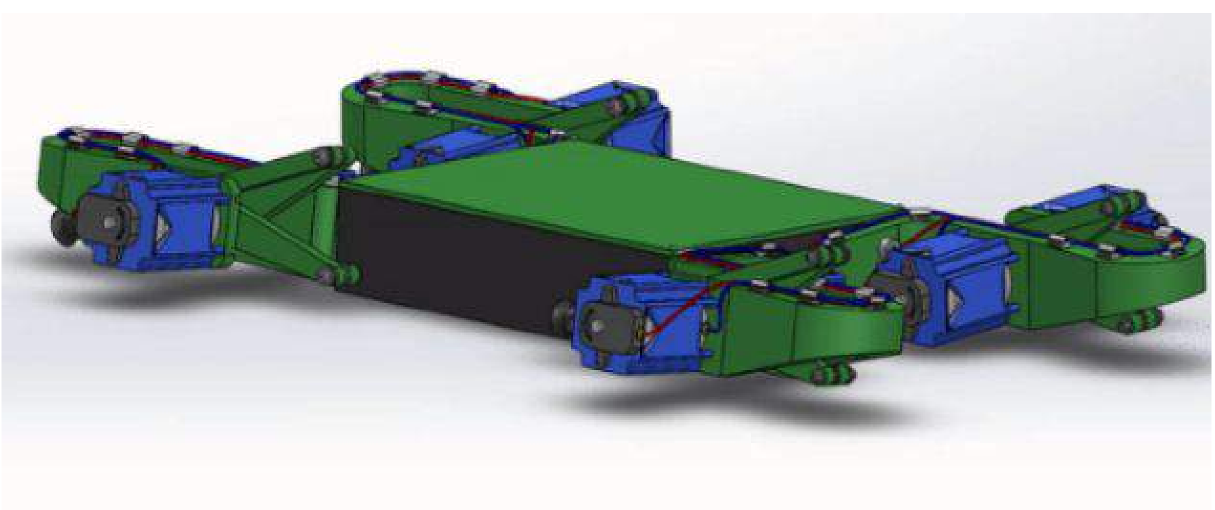


Figure 3.1. Starting position of the robot.

The leg configuration shown above is important for the starting position as the motors need to calibrate and this will provide the correct configuration for calibration (see section 4.2 for calibration instructions).

2. Press the pushbutton switch located on the body to turn on all power to the system.

Figure 3.2. Pushbutton power switch.

Once the button is on (indicated by the LED on button) and the robot boots up (~30 seconds). The user is notified by a green LED on the body, signifying that the system is ready for normal operation and the initial setup process is complete The user can continue to Operation Instructions below.

Note: if green LED is not displayed on body, the initial setup instructions were not met and the user needs to restart setup instructions, see section 4.1 for general troubleshooting instructions.

After the setup instructions are complete and the system is ready, perform the following steps for the operation of the robot.

The operation instructions are as follows:

1. Make sure the robot is still in the (correct) starting position (laying down) as shown in the figure below. It is important to keep this starting position as motors will calibrate if the system is on for the first time. No user input is needed at this step.

Figure 3.3. Starting position of the robot.

The calibration sequence rotates the motors ONE FULL rotation clockwise and ONE FULL rotation counter clockwise (otherwise check section 4.1 general troubleshooting instructions for encoders). After the calibration sequence is completed (make sure calibration offset is correct for OUTER and INNER motors, see section 4.2 motor calibration offset configuration for verification and further instructions).

Figure 3.4. Initial standing position of the robot.

2. If calibration sequence is correctly done, press the start button on the controller for the robot to begin its initial standing position as shown in the figure above. Successful calibration is indicated by the motors fully rotating in both directions. If calibration was not successful, refer to docs.odriverobotics.com/troubleshooting for help.

   This initial standing position is maintained throughout the program if the controller receives no input from the user. This is the base position that the robot will continue to fall under most of the time in order to maintain balance and readiness for movement execution based upon user input.

Figure 3.5. Forward walking of robot with controller input shown.

3. The robot can walk forward with the use of the controller by simply pressing upward on the left analog joystick. This analog input allows the robot to walk forward at varying speeds; the higher the user pushes the joystick, the faster the robot walks forward.

It is important to note that the robot will execute its forward walking gait when the joystick is pushed upward, but when the joystick is idle (not being pushed) then the robot will always go back to its initial standing position (as shown in step 2). Also, directional walking is possible with the use of the right analog joystick on the controller by shifting it left or right depending on desired turn by the user.

Figure 3.6. Backwards walking of robot with controller input shown.

4. The robot can walk backwards with the use of the controller by simply pressing downward on the left analog joystick. This analog input allows the robot to walk backwards at varying speeds; the lower the user pulls the joystick, the faster the robot walks backwards.

Similar to forward walking, it is important to note that the robot will execute its backward walking gait when the joystick is pulled downward, but when the joystick is idle (not being pushed) then the robot will always go back to its initial standing position (as shown in step 2). Similarly, directional walking is possible with the use of the right analog joystick on the controller by shifting it left or right depending on desired turn by the user.

3.2 Charging Instructions

The battery used for Lil'Bro is the **Turnigy Graphene Panther 4000mAH 6S 75C.**

Figure 3.7. Battery pack.

Note: It is important that the battery does **NOT** contain any defects. Inspect battery anytime before operating the robot and confirm that the battery is charged more than 50% of its capacity in order to prevent damage to the robot.

Charging the battery: The **HiTEC multi charger X1 Touch** is the recommended charger to use. But other conventional chargers can also be used provided that it can support the **Turnigy Graphene Panther 4000mAH 6S 75C.**

Figure 3.8. HiTEC Multi Charger.

For charging instructions, read the manual provided in the following link:

https://hitecrcd.com/files/X1_touch_manual.pdf

4. Troubleshooting

4.1 General Troubleshooting

Table 4.1. Solutions to possible errors in operation of Lil'Bro.

Symptom	**Check**
Nothing in the robot turns on	• Check if switch is pressed properly • Check if the battery is plugged in • Inspect battery connection for damage • Charge battery • Test battery
Robot is on, but motors drivers are not	• Check if the power connectors from battery to the drivers are plugged in • Inspect motor driver connectors for damage • Measure voltage potential to motor drivers • Contact manufacture
Robot is on, but Raspberry Pi is not	• Check wire harness from battery for loose connections • Inspect the voltage regulator (display should be on) • Measure voltage potential from regulator • If voltage out is correct, troubleshoot

	Raspberry Pi • If voltage out is incorrect, contact regulator manufacture
All components receive power but robot does not respond to commands	• Connect to Raspberry Pi and check if script is running • Check USB connections between Raspberry Pi and motor drivers • Check if DS controller is on
Robot runs script but some or all motors don't begin calibration sequence	• Check motor drivers for error codes • Check the connections from drivers to motors • Measure current to motors • Check the connections from drivers to encoders • Check if motor hubs are fastened to motor shafts
Motor(s) do not finish calibrating	• Check error codes on motor drivers • Inspect encoder connections for damage
Motors calibrate but a position offset is present	• Refer to instructions below on how to correct offset in section 4.2
Motors spin but the legs do not move	• Check if motor hubs are fastened to motor shafts
Robot stands but legs are not stable	• Increase the proportional position gain through DS controller

	• Check if motor hubs are fully fastened to motor shafts
One of the legs moves slower than the rest	• Set all drivers to “IDLE” mode and check the ease of movement of all legs by hand • If one leg requires a greater force to move, loosen the fastening tightness of the motors to the mount for that leg
Knee joint(s) of leg(s) does/do not rotate like intended	• Set all drivers to “IDLE” mode and apply lubricant on bolt and in bearings • Check contact surface between thrust bearings and legs
Robot is too noisy	• Increase fastening tightness of motor fasteners • Check for leg components rubbing

4.2 Motor Calibration Offset Configuration

The first time you configure a new robot, you must provide the motor calibration offsets; otherwise the positional values of the motors will be inaccurate and operation will fail to execute correctly.

NOTE: It is important to verify that the motor's position setpoint of 0 (after calibration) are in the correct position (~10 o'clock for outer motors and 12 o'clock for inner motors, see Figures 4.2 and 4.6 below, otherwise continue to follow the calibration offset instructions below.

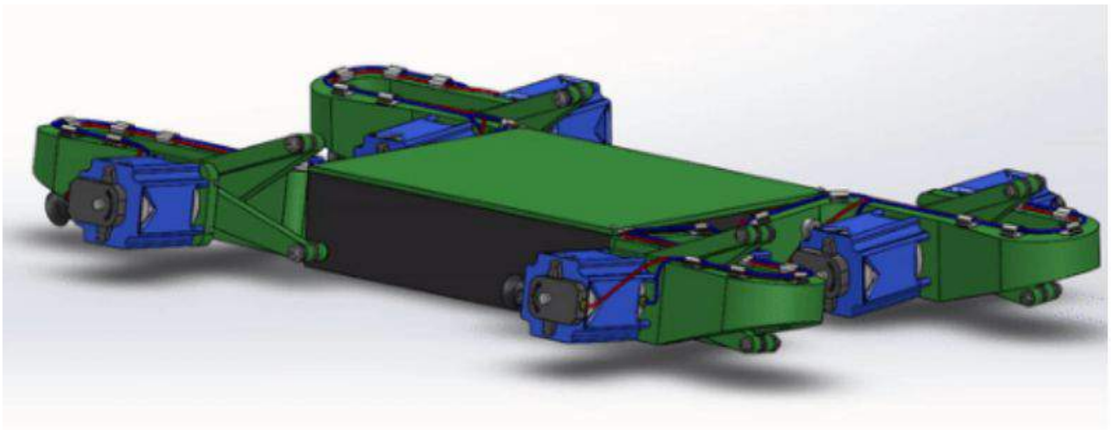


Figure 4.1. Pre-calibration starting position.

The leg configuration (pre-calibration starting position) shown above is important for the calibration sequence, this configuration sets the motor hubs in the correct position for tightening the set screws, and any other configuration will result in failure of operation.

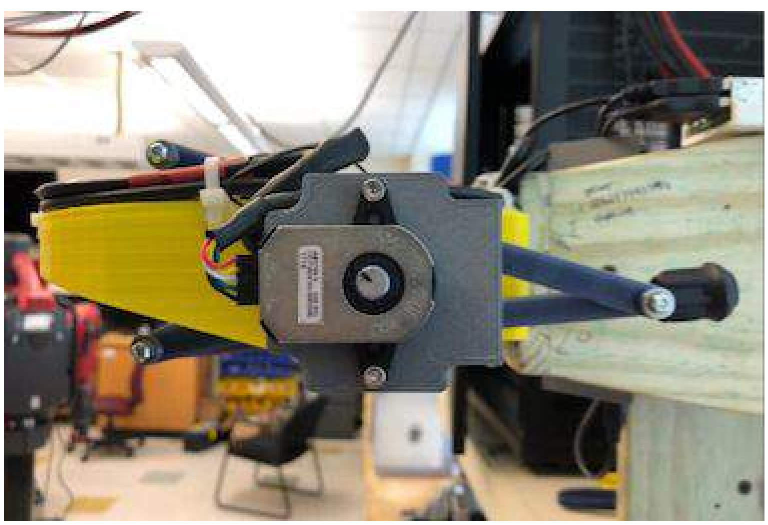


Figure 4.2. Correct shaft position of the outer motor after calibration.

The figure above demonstrates the correct position of the outer motor shaft after calibration. The tick mark indicates the flat side of the motor shaft for tightening the set screws of the motor hub. If this position (~10 O'clock) of the shaft is achieved after calibrating the "OUTER" motors and (12 O'clock) for "INNER" motors, then there is no need to offset the calibration and can continue with normal operation of the robot (Skip the steps below).

1. Connect to robot interface (via ssh or HDMI) and open up a new terminal window.

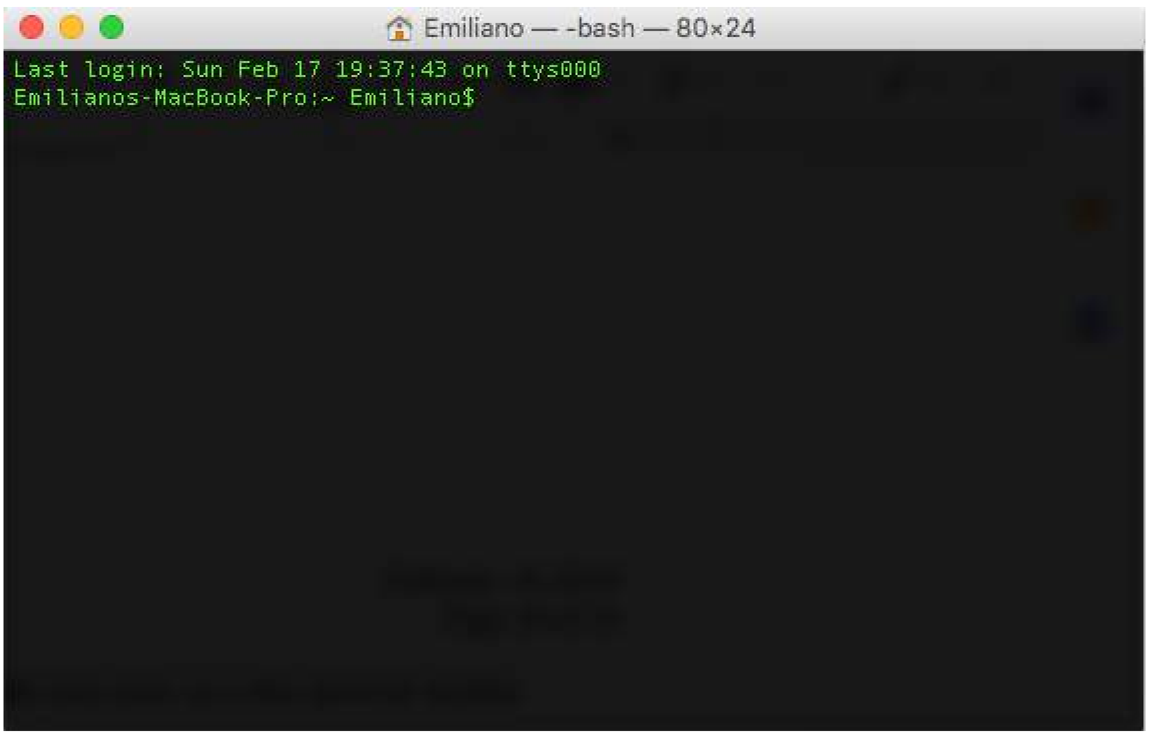


Figure 4.3. New Terminal Window.

2. Make sure odrive motor drivers are connected via USB, type odrivetool and press ENTER.

You can read more about odrivetool here:

https://docs.odriverobotics.com/#start-odrivetool

Figure 4.4. Odrive tool interface.

3. Once the ODrive is connected, type the following command and press ENTER

   `odrv0.axis0.requested_state = AXIS_STATE_FULL_CALIBRATION_SEQUENCE`

Figure 4.5. Odrive tool interface calibration sequence.

NOTE: All the following commands should be typed and executed for all 8 motors.

- *odrvX is the motor driver name where X is an integer value assigned depending on how many boards are connected [0-n]. Where n = board number.*

- *axisX is the motor name where X is an integer value assigned by the connection point on the odrive (can either be 0 or 1).*

The command shown above executes the calibration sequence for the motors. If motors are in the correct position after calibration (see figure 4.6), there is no need to perform the following steps and can continue operation as normal.

4. Once full calibration sequence is done and motors are calibrated, manually move the motors (by hand) to correct tick mark position as indicated below.

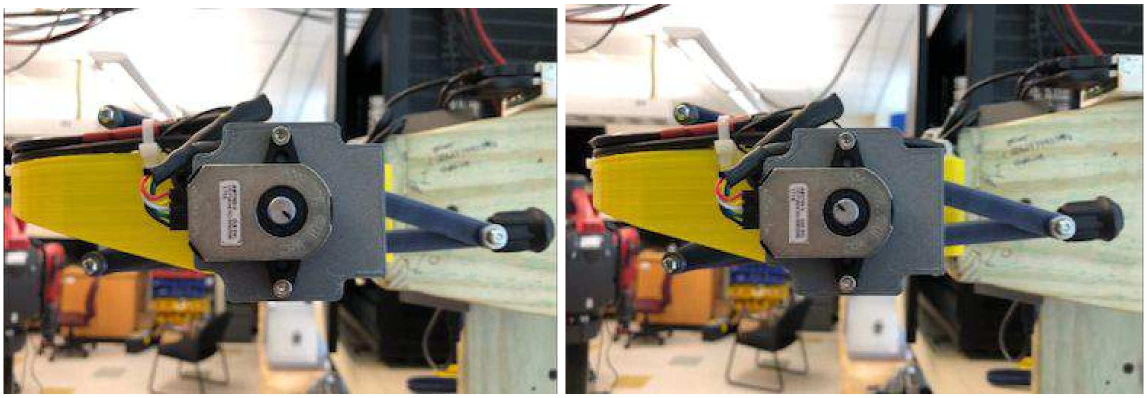


Figure 4.6. Left picture: wrong position of motor shaft after calibration, Right picture: correct position after manually moving motor.

The left figure shown above shows the wrong position after the motor has been calibrated, move the motor (manually by hand) to the correct position shown by the picture on the right above (**Note:** this is for OUTER motors, INNER motor's correct tick position is 12 O'clock).

5. After motor shaft is in the correct position from the previous step. type the following command and press ENTER, this will set the new positional set point of 0.

```
odrv0.axis0.encoder.pos_cpr = 0
```

Figure 4.7. Odrive tool interface assigning new starting position.

The command above sets the current position of the motor to zero after calibration. This step is important as this new position will be recognized as zero after calibration of motors.

6. Save the new configuration by typing the following command and pressing ENTER.

```
odrv0.save_configuration()
```

Figure 4.8. Odrive tool interface saving configuration.

The command above saves the motor driver configuration, this stores the new calibrated positions in memory and can be accessed after reboot or shutdown of motors.

7. Reboot the system by typing the following command and pressing ENTER.

   `odrv0.reboot()`

Figure 4.9. Odrive tool interface rebooting.

Once the command above is executed, the motor driver will disappear from odrivetool and reboot the motor driver. Once rebooted, it will automatically reconnect the driver and recalibration will be necessary in order to use motors again.

8. Once the ODrive restarts and reconnects, re-calibrate the motors by typing the following command and pressing ENTER.

   `odrv0.axis0.requested_state = AXIS_STATE_FULL_CALIBRATION_SEQUENCE`

Figure 4.10. Odrive tool interface recalibration.

9. By visual inspection, make sure the calibration offset is now in the right position (see figureX below, ~10 o'clock for outer and 12 o'clock for inner motors). If correct, tighten

the set screws to the motor hubs and continue operation as normal. Otherwise repeat steps 3-9.

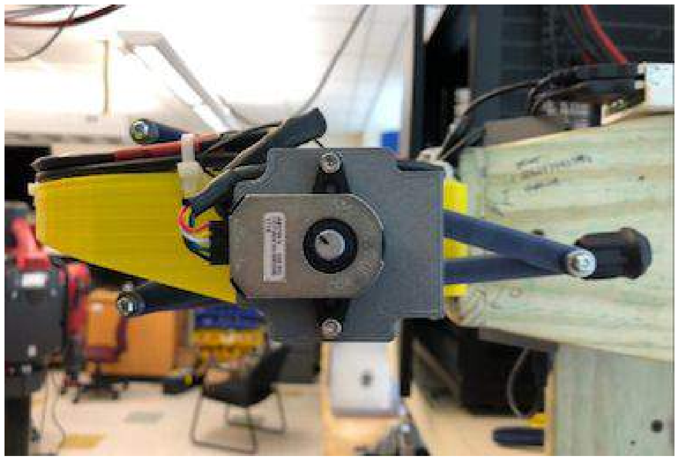


Figure 4.11. Correct shaft position of the outer motor after calibration.

5. References and Contact Information

5.1 References

[1] Bhounsule, Pranav A. Pusey, Jason. Moussouni, Chelsea. “A comparative study of leg geometry for energy-efficient locomotion”.

[2] https://odriverobotics.com/

[3] https://hitecrcd.com/files/X1_touch_manual.pdf

5.2 Contact Information

Appendix B: Test Plan

1. INTRODUCTION

The objective of this project is to design, build, and test an open source quadrupedal, or four-legged, robot for UTSA's Robotics and Motion Laboratory (RAM Lab). The robot, otherwise known as Lil'Bro, will be used by researchers at the RAM Lab for ongoing agile gait locomotion research. The manufacturer of the robot currently used in the lab does not provide the robot's source code with the purchase of the robot. The robot is also expensive and hard to maintain, which is why the Brobotics Inc. team aims to minimize the cost of and maximize accessibility to the robot's design. An online Github repository for the project will be created to allow current and future users to easily access the robot's source code. The team intends to further increase accessibility to users by designing the fabricated parts of the robot to be 3D printed, which has not been done by any other providers of similar-sized robots. It is important to prove that the robot the team builds delivers upon the features it was promised to include. In this document, a test plan for the specifications of the built unit is laid forth, detailing how each of the specifications is to be validated.

2. SCOPE

The necessary materials, tools, personnel, and facilities will be provided by the Robotics and Motion Laboratory at the University of Texas at San Antonio to conduct the testings described herein.

3. FEATURES TO BE TESTED

3.1. FEATURES TO BE EVALUATED

A variety of tests will be conducted on Lil'Bro to validate its design specifications, which are available in Appendix A. There are four specifications that require testing, while the rest of the specifications can be validated by visual observation. The specifications, or features, to be validated through testing are mass, displacement volume, linear velocity, and additional weight bearing capacity. Since the robot is operated via software, details regarding the used architecture and its features will be included.

3.2. COMPLIANCE MATRIX

Table 3.1. Compliance matrix for Lil'Bro.

Item No.	Feature/Specification	Specification Ref. in Appendix A	Testing or Verification Procedure
1	The robot can be controlled through a handheld device.	Section A.1.1	Visual confirmation of final assembly in operation.
2	The robot can move without external physical support.	Section A.1.2	Visual confirmation through testing for item no. 7.
3	The robot moves by walking.	Section A.1.3	Visual confirmation through testing for item no. 7.
4	The robot can support an additional 25% of its weight while standing.	Section A.1.4	Measurement of weight bearing capacity.
5	The robot can collect data in operation.	Section A.1.5	Visual confirmation of collected data through access to the robot's files.
6	The robot's processor is capable of processing at 16 MHz.	Section A.1.6	Confirmation through the manufacturer's specifications.
7	The robot can move at a speed of at least 0.2 m/s.	Section A.1.7	Performance testing with speed measurements.
8	The robot's dimensions are within a cubic meter box.	Section A.2.1	Visual and measurement of physical features.
9	The robot weighs less than 23 kg.	Section A.2.2	Measurement of weight.
10	The robot's software is licensed under GPL3.	Section A.2.3	The software is accessible through an online repository.
11	The robot's leg links can be interchanged between	Section A.2.4	Confirmation through physical assembly and

	assemblies.		technical drawings.

3.3. SOFTWARE

3.3.1. Extensibility

The embedded software framework in the robot will possess extensibility support. This extensible system will allow researchers, future operators and programmers to add on more features and applications without affecting the internal structure and dataflow of the original program's behavior.

Extensibility imposes fewer constraints on developers and will provide cleaner dependencies during development, assisting in usability, maintainability, and extensibility of the system.

3.3.2. Maintainability

The maintainability of the software implies how brittle the code is when subjected to changes. It is important for the software to be flexible and testable for the overall maintainability of the source code. Future developer's will need to easily understand the base code in order to maintain it. To achieve maintainability, it is necessary to document the code properly, and remove any excess code that will cause confusion and make it difficult to understand or modify.

Refactoring the source code is the process of restructuring the existing code without changing its external behavior. By refactoring the code, it will separate key concepts in the program and make it easier for future developers to find broken pieces in the software.

4. TEST FACILITY

4.1. CONFIGURATION

All testing will be performed in the Robotics and Motion Laboratory located in room 2.216 of the Biotechnology Sciences and Engineering building, at the University of Texas at San Antonio's main campus. The robot is currently being fabricated and assembled in this laboratory, and software development and testing have been carried out there as well. The laboratory is equipped with the necessary structural and power support devices to operate and test the unit in a safe manner. The laboratory provides a climate controlled environment, and a secure workplace that will allow for the storage of testing equipment and prevent any tampering.

Figure 4.1. Laboratory testing area.

In the event that more space is needed for Test 3, or the linear velocity walking test, the team has access to UTSA's Recreation Center which can accommodate a much larger area for movement.

Table 4.1. Measured Values.

Symbol	Measured Quantity	Instrument Type	Target	Accuracy	Units
m_1	Mass of front right leg sub-assembly	Scale		±0.001	Kg
m_2	Mass of front left leg sub-assembly	Scale		±0.001	Kg

m_3	Mass of back right leg sub-assembly	Scale		±0.001	kg
m_4	Mass of back left leg sub-assembly	Scale		±0.001	kg
m_5	Mass of body sub-assembly	Scale		±0.001	kg
x	Distance traveled in 15 steps	Measuring Tape		±0.01	m
t	Time spent taking 15 steps	Stopwatch		±0.01	s
L	Length of robot	Measuring Tape		±0.01	m
W	Width of robot	Measuring Tape		±0.01	m
H	Height of robot	Measuring Tape		±0.01	m
t_2	Time spent standing with additional weight	Stopwatch		±0.01	s
Z	25% of the robots weight	Scale		±0.001	lbs
W_1	Weight of the robot	Scale		±0.001	lbs
TW_1	Total weight of robot with the additional 25%	Scale		±0.001	lbs

Table 4.2. Calculated Values.

Symbol	Calculated Quantity	From Variables	Accuracy	Units
$\sum m_{1,2,3,4,5}$	Mass of robot	m_1, m_2, m_3, m_4, m_5	±0.005	kg
$\bar{v}$	Average velocity	x, t	±0.02	m/s
$V = (L \, x \, W \, x \, H)$	Volume of robot	L, W, H	±0.01	m^3

4.2. DATA ACQUISITION

All data will be hand recorded from values given by testing devices. All mass values will be determined by using an Etekcity EK-3252 scale, which has an accuracy rating of one gram, and can be seen in the figure below.

Figure 4.2. Scale. Used to determine mass in testing.

All length values will be measured using a Class II Black and Decker tape measure, which has an accuracy rating of 0.23mm per meter measured and can be seen in the figure below.

Figure 4.3. Tape measure used to determine length in testing

All time values will be measured by using the Apple Clock App, which has a stopwatch accuracy of a hundredth of a millisecond.

4.3. CALIBRATION

Due to the testing being performed, no calibration of testing tools will be needed.

5. TESTING

5.1. MASS TESTING

5.1.1. PROCEDURE

The mass of the entire robot assembly (LB18-101) is determined in this test. Due to the weight capacity of the scale available in the testing facility, the sum of the individual sub-assemblies will be used to test for this value. To prepare for this test each of the leg subassemblies will be removed from the body body subassembly, and any loose wires will be secured in such a way that they do not interfere with the measurement. All components that will be used in the final operation of the robot will be securely mounted in their respective sub-assembly before any measurements will be made, this includes all wiring, wire fasteners, and battery. Each sub-assemblies mass will be measured five separate times, and can be seen in Table 5.1

5.1.2. TEST CONDITIONS

This test will be performed in a controlled environment, where the impact from the ambient conditions will not affect the results.

5.1.3. TEST PARAMETERS

The summation of the measured values in Table 5.1 will result in the calculated value of the Robots Mass ($\sum m_{1,2,3,4,5}$), the target for this value in order to meet project specifications, is $\sum m_{1,2,3,4,5} < 23$Kg

5.1.4. TEST MATRIX

The test matrix shown in Table 5.1, list the measured values of this test. Each sub-assemblies mass value will be taken in five separate measurements to provide accuracy, these values will be recorded in Table 5.1. An average of the five measured values will also be recorded and used to determine the mass of the total assembly. Each measured value will be recorded only after allowing the scale to stop fluctuating.

Table 5.1. Mass Test Matrix

	m_1	m_2	m_3	m_4	m_5	$\sum m_{1,2,3,4,5}$
TEST 1						
TEST 2						
TEST 3						
TEST 4						
TEST 5						
AVERAGE						

SYMBOL	**NOMENCLATURE/NOTATION**
m_1	Mass of front right leg assembly
m_2	Mass of front left leg assembly
m_3	Mass of back right leg assembly
m_4	Mass of back left leg assembly
m_5	Mass of body assembly
$\sum m_{1,2,3,4,5}$	Robots mass or summation of $(m_1, m_2, m_3, m_4, m_5)$

5.2. VOLUME/DISPLACEMENT TESTING

5.2.1. PROCEDURE

The primary goal of this test is to validate that the dimensions of the robot are within a cubic meter box, item no. 8 in Table 1 (compliance matrix). Note that the volume of the robot and the displacement of its parts (simulation of operation) will be determined in this test.

In order to verify that the robots volume remains within the required specification of a cubic meter, a number of tests will be conducted within the testing facility. With the use of an accurate measuring tape, a series of measurements will be taken in order to verify that the volume of the robot remains within the bounds of a cubic meter. Different leg configurations will be executed, tested and measured (L x W x H) as follows; the standby leg configuration (normal standing position), legs vertically stretched out (highest standing position), and legs extended out (forward and backwards of maximum limits during operation). All positions mentioned will be static when measured, and set by a controlled program that verifies that the robot will remain under a cubic meter.

5.2.2. TEST CONDITIONS

This test will be performed in a controlled environment, where the impact from the ambient conditions will not affect the result. This will be a supervised test that will analyze the maximum limits on the robots positional coordinates. The robot will NOT be in motion when taking measurements to obtain accurate readings.

5.2.3. TEST PARAMETERS

Table 5.2 will include all the volume based on the displacement measurements. All measurements will have an accuracy of **X ± 0.01** mm and the volume (L x W x H) < 1 m^3 in order to meet the project's specification requirements.

5.2.4. TEST MATRIX

Table 5.2. Volume/Displacement Test Matrix

	L	*W*	*H*	*V* = (*L x W x H*)
TEST 1 (**Standby Position)**				
TEST 2 **(High Standing)**				
TEST 3 **(Extended Legs)**				

SYMBOL	NOMENCLATURE NOTATION
L	Length
W	Width

H	Height
$V = (L \times W \times H)$	Product of Robots Volume (L x W x H)

5.3. LINEAR VELOCITY TESTING

5.3.1. PROCEDURE

The primary goal of this test is to validate the linear speed, item no. 7 in Table 1 (compliance matrix), of the robot. The test will be conducted in such a way that items no. 2 and 3 from the compliance matrix, the independent operation and locomotion specifications, will also be validated.

To verify that the speed of the robot meets the required minimum of 0.2 m/s, the time the robot spends traveling 0.2 m will be measured. As specified by the specification, the robot is to take 15 steps while traveling at a minimum speed of 0.2 m/s. The first step of the test will be selecting a starting point for the robot, as well as, a point of reference on the robot. This point will be the point monitored by the test administrator throughout the test. It is recommended that the point selected is on a leg mount or the body of the robot because the legs themselves will be continuously moving in a nonlinear walking path. Once a starting point is selected, it will be marked on the ground. The robot is to walk up to the marked starting point, and once the point of reference on the robot crosses the starting point, a timer is initiated. The robot will then take 15 steps, monitoring the distance it travels throughout. The timer is stopped once the 15th step is successfully completed. Using the reference point on the robot, the distance is then measured from the marked starting point to the final position robot reached.

This test requires an experienced robot operator to be performed, who will be controlling the robot using a handheld device. In doing so, item no. 1 in Table 1 is validated. Measurements taken throughout the test will be taken by multiple testers to minimize bias, with 10 iterations of testing.

5.3.2. TEST CONDITIONS

The following conditions are to be maintained while the test is conducted:

- Flat ground, with level variations within 1°

- Testing facility's flooring is to remain dry
- Robot is to be operated by a single operator
- Multiple test conductors are to take distance and time measurements

5.3.3. TEST PARAMETERS

The measured parameters from this test are listed in Table 4.1, and they will be used to calculate the average velocity $\bar{v}$. Those parameters are:

- The distance traveled over 15 steps x
- The time t taken to travel distance x

5.3.4. TEST MATRIX

The matrix for this test is shown below in Table 5.3 and will be used to record measurements from the test, with a brief nomenclature included for the symbols used.

Table 5.3. Linear Velocity Test Matrix.

Iteration	x (m)	t (s)	$\bar{v}$ (m/s)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

SYMBOL	**NOMENCLATURE NOTATION**

x	Distance
t	Time
$\bar{v}$	Average Velocity

5.4. ADDITIONAL WEIGHT BEARING TESTING

5.4.1. PROCEDURE

The additional 25% weight bearing of the robot while standing will be determined in this test. The additional weight to be used will be a gallon of distilled water. This will be conducted by placing an empty gallon of distilled water onto the scale mentioned in figure 4.2 which is provided by the testing facility. Once the entire mass of the robot is calculated it will then be multiplied by 25%.This will be done using the equation $Z = W * .25$. Z will be the additional weight to be added. Now using the following equation $TW = W + Z$, this will give us the Overall total weight with the additional weight. The additional weight value will then be displayed by distilled water in the gallon. The empty gallon will first be weighed and zeroed out. Afterwards, water will be poured into the gallon until reaching the calculated 25% value. Once acquiring the correct additional weight we will then place it on top of the robot, which will be in the standing position. The robot will be holding this additional weight for eight seconds. This will be repeated for five iterations. The results of each iteration will be placed in table 5.4.

5.4.2. TEST CONDITIONS

This test will be performed in a controlled environment, where the impact from the ambient conditions will not affect the result. The robot will not be walking in this test. It will only be in the standing position.

5.4.3. TEST PARAMETERS

Table 5.4 will include the time and the total weight measurements of the robot. It will also include whether or not the robot withstood the eight seconds of standing with the additional 25% weight added onto it. If the robot lasts greater than or equal to 8 seconds then it will meet the specification of the project.

5.4.4. TEST MATRIX

Table 5.4. Additional Weight Bearing Test Matrix

	Z	*Total Weight*	*Stabilization Time t*	*Pass/Fail*
TEST 1				
TEST 2				
TEST 3				
TEST 4				
TEST 5				

SYMBOL	NOMENCLATURE NOTATION
TW_1	Total Weight (including the extra 25%)
t	Time
W_1	Weight of the robot
Z	Additional 25% weight

6. DATA ANALYSIS

6.1. ANALYSIS OF MASS

The total mass of the robot will be the summation of the subassembly masses, the equation for the total mass of the robot can be seen in the equation below.

$$\sum m_{1,2,3,4,5} = m_1 + m_2 + m_3 + m_4 + m_5$$

6.2. ANALYSIS OF VOLUME/DISPLACEMENT

Volume is the amount of space that an object occupies. The parameter of interest in this test is the volume that the robot takes up. In order to quantify the results, the following equation is used to find volume:

$$V = L\,x\,W\,x\,H$$

The measuring tape utilized on this test is accurate enough to yield a reliable reading to justify the space occupied by the robot. In the equation mentioned above, the volume V is found by multiplying the robot's length L, width W, and height H. The desired volume by the product of the length, width, and height is to be no more than a cubic meter, meeting the required specification for this test.

6.3. ANALYSIS OF LINEAR VELOCITY

Velocity, or speed, is the rate of change of position over time. Average velocity is the parameter of interest in the linear velocity test due to the nature of the test. The overall distance x traveled by the robot over a period of time t yields average velocity $\overline{v}$ in the following equation:

$$\overline{v} = \frac{x}{t}$$

Considering that 10 iterations of this test will be carried out, a mean of all calculated average velocities will be calculated to obtain a single definitive value. The mean will be calculated using the following equation:

$$\overline{a} = \frac{\sum_{i=1}^{n} a_i}{n}$$

In the context of this test, a represents average velocity and n is the number of iterations. The desired calculated average velocity would be a value greater than or equal to 0.2 m/s, signifying that the specification is met.

6.4. ANALYSIS OF ADDITIONAL WEIGHT BEARING

Weight is the mass of an object multiplied by the gravity of the earth. The additional 25% weight to be added to the robot while standing and seeing if it could support the weight is the parameter of interest. In order to calculate both the weight of the robot and the additional weight the following equation is applied:

$$W_1 = m * 9.8\frac{m}{s^2}$$

Now in the context of this equation W_1 can be either the weight of the robot or the weight of the additional weight to be added. The scale used to calculate the mass is accurate enough to yield a reliable value. The desired result is that of the robot managing to support an additional 25% of its weight while standing, signifying that the specification is met.

6.5. PRESENTATION OF RESULTS

Results for each of the tests can be seen in Table 6.1 below.

Table 6.1. Test Results.

Desired Test Ranges	Test Results
Mass ≤ 23 (kg)	
Volume ≤ 1 (m^3)	
Linear Velocity ≥ 0.2 (m/s)	
Additional Weight Bearing ≥ 25% of Robot Mass (kg)	

Table 6.2 shows the compliance matrix with an input field for whether or not a feature/specification was met and indicating it with a Y or N, for yes or no, respectively. Although only four tests included collecting data, visual observation of the remaining features suffices.

Table 6.2. Post-Testing Compliance Matrix.

Item No.	Feature/Specification	Specification Ref. in Appendix A	Feature/Specification Met (Y/N)
1	The robot can be controlled	Section A.1.1	

	through a handheld device.		
2	The robot can move without external physical support.	Section A.1.2	
3	The robot moves by walking.	Section A.1.3	
4	The robot can support an additional 25% of its weight while standing.	Section A.1.4	
5	The robot can collect data in operation.	Section A.1.5	
6	The robot's processor is capable of processing at 16 MHz.	Section A.1.6	
7	The robot can move at a speed of at least 0.2 m/s.	Section A.1.7	
8	The robot's dimensions are within a cubic meter box.	Section A.2.1	
9	The robot weighs less than 23 kg.	Section A.2.2	
10	The robot's software is licensed under GPL3.	Section A.2.3	
11	The robot's leg links can be interchanged between assemblies.	Section A.2.4	

6.6. UNCERTAINTY ESTIMATES

The mass will be determined by a scale where the error will have a nonlinear relation to the mass being tested. The increase in sensitivity with the increase of mass being tested, will not factored in due to the controlled climate where the test will take place. The mass values will be measured five separate times to reduce the factor of human error. Due to the mass tested, being on the upper end of the scales capacity, the relative uncertainty is estimated to be 0.05% of the total calculated mass value.

For the volume/displacement test, measurements are taken by human operators through measuring tapes, which may result in human error due to incorrect forms of measuring. In order to eliminate this uncertainty, the robots volume (for each test) will be measured 3 times until the volume is similar and within acceptable values, multiple test conductors will take these measurements to validate previous volumetric measurements.

For the linear velocity test, measurements are taken by human operators through measuring tapes and stopwatches, which subject measured quantities to human error. To account for this, multiple test conductors will be taking measurements to minimize bias from a single conductor.

For the additional weight bearing test, the stopwatch is human operated meaning that the stop time of 8 seconds may be exceeded or reduced depending on the reflexes of the operator. The additional weight is also being placed by a human operator. This placement may occur before or after the operator of the stopwatch starts the time. To account for this many practice runs will be conducted between both operators to ensure the best synchronization.

7. SCHEDULE

A detailed timeline for conducting the test over the course of 39 days is shown in the Gantt Chart on the next page. The project as shown can be completed by the end date of May 3, 2019. The main tasks and the amount of days required to accomplish them are:

1. Development of Test Plan (5 days)
2. Locate Test Facility (1 day)
3. Acquisition of Materials & Tools (3 days)
4. Functional Check of Robots & Calibrations (2 days)
5. Conduction of Tests (4 days)
6. Collection of Data & Data Analysis (11 days)
7. Test Report (12 days)

The schedule does include two deliverables. The final Test plan which is due 16 days before the in the initiation steps. The final test report is due 16 days after the tests have been conducted. It is believed that this test schedule complies with the original Gantt chart and initial contract. The project will be completed and handed over to the sponsor on said due date.

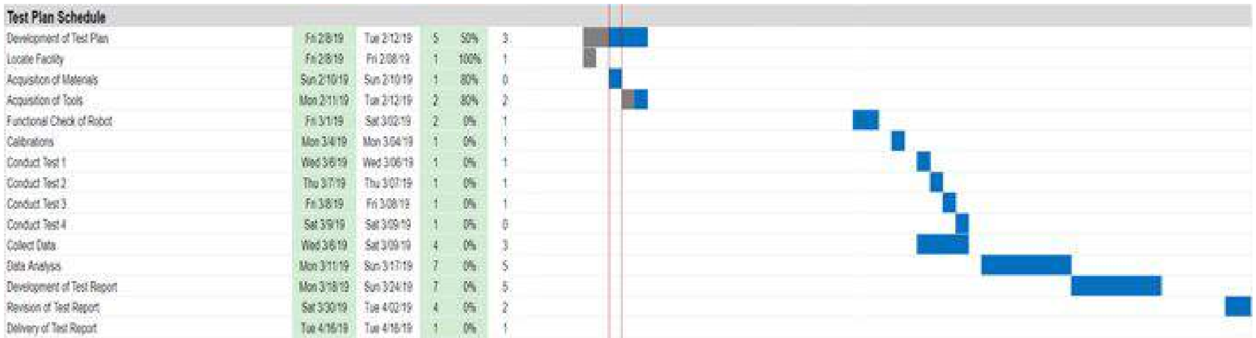


Figure 7.1. Test Schedule GANTT Chart

8. PROGRAM RISKS

The primary area of concern for this project pertains to how the tests outlined in this document are contingent upon the timely completion of the software development of the robot. Although making alterations to the software is expected throughout testing, a robust base architecture is necessary for any kind of operation. To do so requires time, which is currently limited, so the team will be making an effort to accelerate the development process by eliminating planned exploratory testing.

9. COMMUNICATION

Questions regarding the information in this test plan or any aspect of this project should be directed to the Team Lead at Brobotics Inc., Steven Farra (zvo618@my.utsa.edu).

Appendix C: Test Report

1. INTRODUCTION

1.1. Description and Purpose of Testing and Scope

The objective of this project is to design, build, and test an open source quadrupedal, or four-legged, robot for UTSA's Robotics and Motion Laboratory (RAM Lab). The robot, otherwise known as Lil'Bro, will be used by researchers at the RAM Lab for ongoing agile gait locomotion research. The manufacturer of the robot currently used in the lab does not provide the robot's source code with the purchase of the robot. The current robot is also expensive and hard to maintain; Lil'Bro's 3D printed parts can be easily replaced, and purchased parts arrive within a week of a placed order. Brobotics Inc. aims to minimize the cost of and maximize accessibility to the robot's design; the current robot provided by Ghost Robotics, is priced at ~$10,000, while Brobotics Inc.'s solution reduces price down to $2,000. An online Github repository for the project will be created to allow current and future users to easily access the robot's source code. The team intends to further increase accessibility to users by designing 100% of the fabricated parts of the robot to be 3D printed, which has not been done by any other providers of similar-sized robots. It is important to prove that the robot the team builds delivers upon the features it was promised to include. In this document, a test report for the specifications of the built unit is laid forth, detailing how each of the specifications is validated. The necessary materials, tools, personnel, and facilities will be provided by the Robotics and Motion Laboratory at the University of Texas at San Antonio to conduct the testings described herein.

1.2. Features Tested

A variety of tests were conducted on Lil'Bro to validate its design specifications, which are available in Appendix A. There are four specifications that required testing, while the rest of the specifications were validated by visual observation. The specifications, or features, validated through testing are, mass, displacement volume, linear velocity, and additional weight bearing capacity.

Table 1.1. Measured Value Nomenclatures

Symbol	Measured Quantity	Instrument Type	Accuracy	Units
m_1	Mass of front right leg sub-assembly	Scale	±0.001	Kg
m_2	Mass of front left leg sub-assembly	Scale	±0.001	Kg
m_3	Mass of back right leg sub-assembly	Scale	±0.001	kg
m_4	Mass of back left leg sub-assembly	Scale	±0.001	kg
m_5	Mass of body sub-assembly	Scale	±0.001	kg
x	Distance traveled in 15 steps	Measuring Tape	±0.01	m
t	Time spent taking 15 steps	Stopwatch	±0.01	s
L	Length of robot	Measuring Tape	±0.01	m
W	Width of robot	Measuring Tape	±0.01	m
H	Height of robot	Measuring Tape	±0.01	m
t_2	Time spent standing with additional weight	Stopwatch	±0.01	s
Z	25% of the robots weight	Scale	±0.001	lbs
W_1	Weight of the robot	Scale	±0.001	lbs
TW_1	Total weight of robot with the additional 25%	Scale	±0.001	lbs

Table 1.2. Calculated Value Nomenclatures

Symbol	Calculated Quantity	From Variables	Accuracy	Units
$\sum m_{1,2,3,4,5}$	Mass of robot	m_1, m_2, m_3, m_4, m_5	±0.005	kg
$\bar{v}$	Average velocity	x, t	±0.02	m/s
$V = (L\ x\ W\ x\ H)$	Volume of robot	L, W, H	±0.01	m^3

2. TEST EVALUATION CRITERIA

2.1. Compliance Matrix

Table 2.1. Compliance matrix for Lil'Bro.

Item No.	Feature/Specification	Specification Ref. in Appendix A	Testing or Verification Procedure
1	The robot can be controlled through a handheld device.	Section A.1.1	Visual confirmation of final assembly in operation.
2	The robot can move without external physical support.	Section A.1.2	Visual confirmation through testing for item no. 7.
3	The robot moves by walking.	Section A.1.3	Visual confirmation through testing for item no. 7.
4	The robot can support an additional 25% of its weight while standing.	Section A.1.4	Measurement of weight bearing capacity.
5	The robot can collect data in operation.	Section A.1.5	Visual confirmation of collected data through access to the robot's files.
6	The robot's processor is capable of processing at 16 MHz.	Section A.1.6	Confirmation through the manufacturer's specifications.
7	The robot can move at a speed of at least 0.2 m/s.	Section A.1.7	Performance testing with speed measurements.
8	The robot's dimensions are within a cubic meter box.	Section A.2.1	Visual and measurement of physical features.
9	The robot weighs less than 23 kg.	Section A.2.2	Measurement of weight.

10	The robot's software is licensed under GPL3.	Section A.2.3	The software is accessible through an online repository.
11	The robot's leg links can be interchanged between assemblies.	Section A.2.4	Confirmation through physical assembly and technical drawings.

2.2. Pretest Uncertainty Analysis

The mass was determined by a scale where the error will have a nonlinear relation to the mass being tested. The increase in sensitivity with the increase of mass being tested, did not factor in due to the controlled climate where the test will take place. The mass values were measured five separate times to reduce the factor of human error. Due to the mass tested, being on the upper end of the scales capacity, the relative uncertainty is estimated to be 0.05% of the total calculated mass value.

For the volume/displacement test, measurements are taken by human operators through measuring tapes, which may result in human error due to incorrect forms of measuring. In order to eliminate this uncertainty, the robots volume (for each test) were measured 3 times until the volume is similar and within acceptable values, multiple test conductors took these measurements to validate previous volumetric measurements.

For the linear velocity test, measurements are taken by human operators through measuring tapes and stopwatches, which subject measured quantities to human error. To account for this, multiple test conductors took measurements to minimize bias from a single conductor. Human reaction time is estimated to have a 0.25 second delay, on average, to visual stimuli, which will be taken into consideration.

For the additional weight bearing test, a stopwatch is used to determine the time the robot remains stable under loading. The additional weight is also being placed by a human operator. This placement may occur before or after the operator of the stopwatch starts the time. To account for this many practice runs were conducted between both operators to ensure the best synchronization.

3. TESTING FACILITY

3.1. Facility Configuration

All testing was performed in the Robotics and Motion Laboratory located in room 2.216 of the Biotechnology Sciences and Engineering building, at the University of Texas at San Antonio's main campus. The robot was fabricated and assembled in this laboratory, and software development and testing have been carried out there as well. The laboratory is equipped with the necessary structural and power support devices to operate and test the unit in a safe manner. The laboratory provides a climate controlled environment, and a secure workplace that allowed for the storage of testing equipment and prevent any tampering.

Figure 3.1. Laboratory testing area.

3.2. Instrumentation

All the data was hand recorded from values given by the following testing devices. All mass values were determined by using an Etekcity EK-3252 scale, which has an accuracy rating of one gram, and can be seen in the figure below.

Figure 3.2. Scale. Used to determine mass in testing.

All length values were measured using a Class II Black and Decker tape measure, which has an accuracy rating of 0.23mm per meter measured and can be seen in the figure below.

Figure 3.3. Tape measure used to determine length in testing

All time values were measured by using the Apple Clock App, which has a stopwatch accuracy of a hundredth of a millisecond.

4. TESTING

4.1. Mass Test

4.1.1. Procedure

The mass of the entire robot assembly (LB18-101) is determined in this test. Due to the weight capacity of the scale available in the testing facility, the sum of the individual sub-assemblies were used to test for this value. To prepare for this test, each of the leg subassemblies were removed from the body subassembly, and any loose wires were secured in such a way that they didn't interfere with the measurement. All the components that were used in the final operation of the robot were securely mounted in their respective sub-assembly before any measurements was made, this includes all wiring, wire fasteners, and battery. Each sub-assembly's mass was measured five separate times, and values measured and calculated can be seen in Table 4.1 and 4.2 below,

Table 4.1. Measured Values.

Symbol	Measured Quantity	Instrument Type	Accuracy	Units
m_1	Mass of front right leg sub-assembly	Scale	±0.001	Kg
m_2	Mass of front left leg sub-assembly	Scale	±0.001	Kg

m_3	Mass of back right leg sub-assembly	Scale	±0.001	kg
m_4	Mass of back left leg sub-assembly	Scale	±0.001	kg
m_5	Mass of body sub-assembly	Scale	±0.001	kg

Table 4.2. Calculated Values.

Symbol	Calculated Quantity	From Variables	Accuracy	Units
$\sum m_{1,2,3,4,5}$	Mass of robot	m_1, m_2, m_3, m_4, m_5	±0.005	kg

4.1.2. Conditions

This test was performed in a controlled environment, where the impact from the ambient conditions did not affect the results.

4.1.3. Instrumentation and Calibration

All mass values were determined by using an Etekcity EK-3252 scale, which has an accuracy rating of one gram. A calibration method was performed before each use, this includes wiping the scale down and setting a new zero.

4.2. Displacement/Volume Test

4.2.1. Procedure

The primary goal of this test is to validate that the dimensions of the robot are within a cubic meter box, item no. 8 in Table 1 (compliance matrix). Note that the volume of the robot and the displacement of its parts (simulation of operation) were determined in this test.

In order to verify that the robot's volume remains within the required specification of a cubic meter, a number of tests were conducted within the testing facility. With the use of an accurate measuring tape, a series of measurements were taken in order to verify that the volume of the robot remains within the bounds of a cubic meter. Different leg configurations were executed, tested and measured (L x W x H) as follows; the standby leg configuration (normal standing position), legs vertically stretched out (highest standing position), and legs extended out (forward and backwards of maximum limits during operation). All positions mentioned were static when measured, and set by a controlled program that verifies that the robot remains under a cubic meter.

The length of Lil'Bro was taken from the edge of a mount to the edge of the mount directly across from it. This measurement remained the same for each position due to the mounts being at the edge of the robot and the position of its legs never exceeding the mounts. This can be seen down below in figure 4.1.

Figure 4.1. Length of assembly

Due to the complexity of the design of Lil'Bro, the measurements of both the width and height of the robot were split into sections then summated, and this can be seen in tables 5.3, 5.4, and 5.5.

The method for measuring the width of the robot remained the same for all three positions. Height, on the other hand, was measured differently for all three positions.

As mentioned previously, width was measured in three sections, those sections include measuring from the shaft to the encoder, encoder to motor housing, and lastly from the outside motor housing of one leg to the exterior motor housing of the leg across. Figure 4.2 demonstrates the measurement from the shaft to the encoder. Figure 4.3 displays the measurement from the encoder to the motor housing, and lastly figure 4.4 shows the measurement between the exterior motor housings.

Figure 4.2. Measurement of Shaft of the Encoder

Figure 4.3. Measurement from Encoder to the Motor Housing

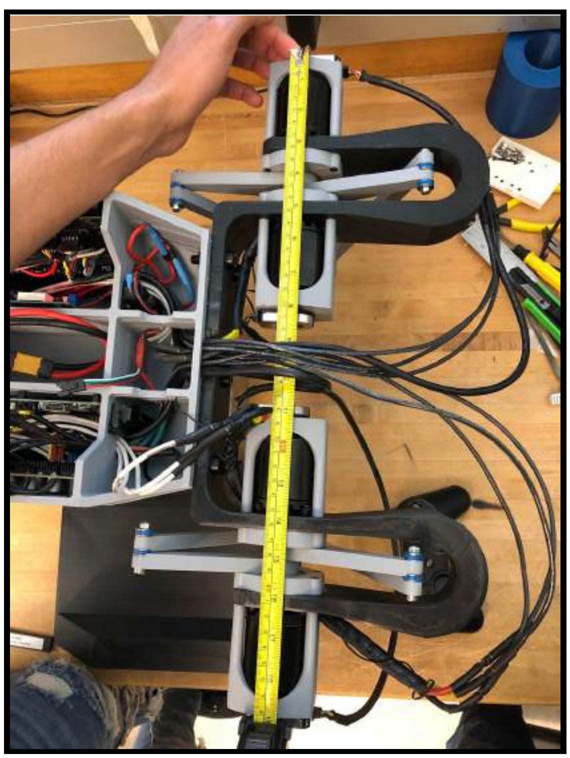


Figure 4.4. Measurement from motor housing to motor housing

The first position measured for height was the standby leg position. This position was split into three sections those sections include from the bottom of the foot to the bottom of the mount, from the bottom of the mount to the top of the body, and the thickness of the cover shown in figure 4.5. With each section measurement there was an estimated error and is shown in table 5.4.

Figure 4.5. Thickness of the Cover

The second position measured was the extended leg position. This test was split into three sections as well, but due to the location of the extended leg the measurement was taken from the bottom of the foot to the top of the mount. The measurement from the bottom of the foot to the bottom of the mount is known since it was measured in the previous position. The measurement reading from the bottom of the foot to the top of the mount was subtracted by the thickness of the mount. The last section is the thickness of the cover. Table 5.5 displays the measurement readings for each section with its respective estimated errors.

The third and last position measured was the high standing leg position. Similar to the other two tests, this test was also separated into three sections. The measurements were taken from the bottom of the foot to the bottom of the mount as can be seen in figure 4.6, from the bottom of the mount to the top of the body, and lastly the thickness of the cover which can be seen in figure 4.5. These measurements are displayed in table 5.6 with its respective errors.

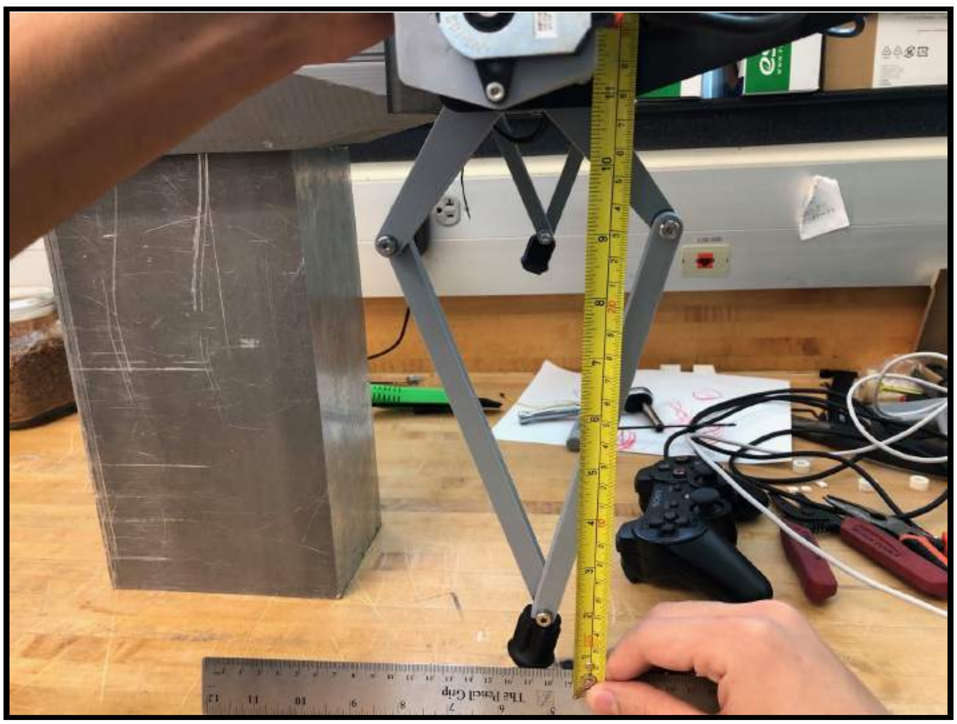


Figure 4.6. Measurement of High Standing Leg Position

4.2.2. Conditions

This test was performed in a controlled environment, where the impact from the ambient conditions did not affect the result. This was a supervised test that analyzed the maximum limits on the robot's positional coordinates. The robot was static when taking measurements to obtain accurate readings. The necessary tools utilized were a tape measure and a digital caliper. The resolution of the tape measure and digital caliper were 1mm and .1 mm respectively.

4.2.3. Instrumentation and Calibration

Due to the testing that was performed as well as the testing tools utilized, no calibration were needed.

4.3. Linear Velocity Test

4.3.1. Procedure

The primary goal of this test is to validate the linear speed, item no. 7 in Table 1 (compliance matrix), of the robot. The test was conducted in such a way that item no. 3 from the compliance matrix, the locomotion specification, was also validated.

To verify that the speed of the robot meets the required minimum of 0.2 m/s, the time the robot spends traveling 1 m was measured. As specified by the specification, the robot is to take at least 15 steps while traveling at a minimum speed of 0.2 m/s. The first step of the test was selecting a starting point for the robot, as well as, a point of reference on the robot. This point was the point monitored by the test administrator throughout the test. It is recommended that the point selected is on a leg mount or the body of the robot because the legs themselves were continuously moving in a nonlinear walking path. Once a starting point is selected, it was marked on the ground. The robot walked up to the marked starting point, and once the point of reference on the robot crossed the starting point, a timer was initiated. The robot then took at least 15 steps, monitoring the distance it traveled throughout. The timer was stopped once the 1m distance mark was successfully reached.

This test required an experienced robot operator to be performed, who controlled the robot using a handheld device. In doing so, item no. 1 in Table 1 was validated. Measurements taken throughout the test were taken by multiple testers to minimize bias, with 10 iterations of testing.

4.3.2. Conditions

The following conditions were to be maintained while the test was conducted:

- Flat ground, with level variations within 1°
- Testing facilities flooring is to remain dry
- Robot is to be operated by a single operator
- Multiple test conductors are to take distance and time measurements

4.3.3. Instrumentation and Calibration

The measured parameters from this test are listed in Table 5.7, and they were used to calculate the average velocity $\overline{v}$. Those parameters are:

- The distance traveled over at least 15 steps x
- The time t taken to travel distance x

The distance is measured by a measuring tape and time with a stopwatch, both of which do not need calibration.

4.4. Additional Weight Capacity Test

4.4.1. Procedure

An additional 25% weight carrying test on the robot, while it is standing, was determined in this test. The additional weight used was a measured amount of sand in a container. This test was conducted by taking a container, and filling it with sand; small amounts of sand were then removed until the mass of the container and the sand were equal to the of 25 percent of the main assembly's total mass. This is shown in figure 4.7. The scale mentioned in figure 3.2 was used for this test. The entire mass of the robot was calculated in the mass test in section 5.1. The value acquired was 9.52 kg or 21 lbs, this mass was then multiplied by 25%.This was done using the following equation,

$$Z = W * .25$$

Z was calculated to be 2.384 kg or 5.25 lbs. The following equation provides the total weight of the assembly and the additional 25%.

$$TW = W + Z$$

Figure 4.7. Additional 25% Weight.

Once acquiring the correct additional weight, it is placed atop the robot in the standing position which can be seen in figure 4.8 below.

Figure 4.8. Carrying Capacity Test (Using measured amount of sand as additional weight)

The robot held this additional weight for at least eight seconds. This process was repeated for five iterations. The results of each iteration are placed in table 5.4.

4.4.2. Conditions

This test was performed in a controlled environment, where the impact from the ambient conditions did not affect the result. The robot did not walk in this test. It was kept in the standing position.

4.4.3. Instrumentation and Calibration

Table 5.8 includes the time and the total weight measurements of the robot. It also includes whether or not the robot withstood the eight seconds of standing with the additional 25% weight added onto it. If the robot lasts greater than or equal to 8 seconds, the specification is met. Due to the test performed calibration of the scale was conducted in order to get an optimal reading.

4.5. Visual Confirmation Tests

The following tests are conducted through visual inspection. These tests do not require any instrumentation as well as any data tables/matrixes. There is no mathematical analysis required for these tests, due to the fact that these specification requirements can be confirmed through a visual inspection test.

4.5.1. Handheld Device Compatibility Test

For this test, the operator must be able to control the robot through the use of a handheld device. This test was conducted by connecting a Dual Shock 3 controller to a raspberry pi 3 B+. This connection was achieved through a bluetooth connection. Figure 4.7 below, demonstrates that this specification was met.

Figure 4.9. Handheld device connection

4.5.2. Independent Operation Test

For this test the robot must stand and operate without any form of assistance or support from a boom or other weight bearing supports. This allows for the robot to be operated and tested in an uncontrolled environment. A handheld tether was attached to the robot as a precautionary method to stop the robot and prevent it from any damage. Figure 4.8 below, displays the robot standing without any assistance or support meeting the set specification.

Figure 4.10. Standing Operation

4.5.3. Data Collection Test

The raspberry pi must be capable of collecting and storing data on the motors position, motor speed, motor current draw, and the center of mass position in the system's memory. This data could then be viewed for performance monitoring as well as being utilized to configure the robot's source code. The data is saved into the "Data" folder and each data type is written into a text file. The data collected can be seen in figure 4.10 below, which is a visual representation that the set specification was met.

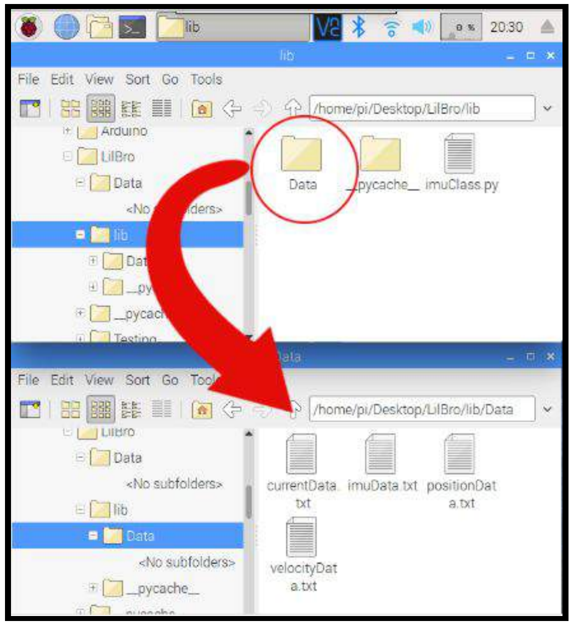


Figure 4.11. Data Collection Folder

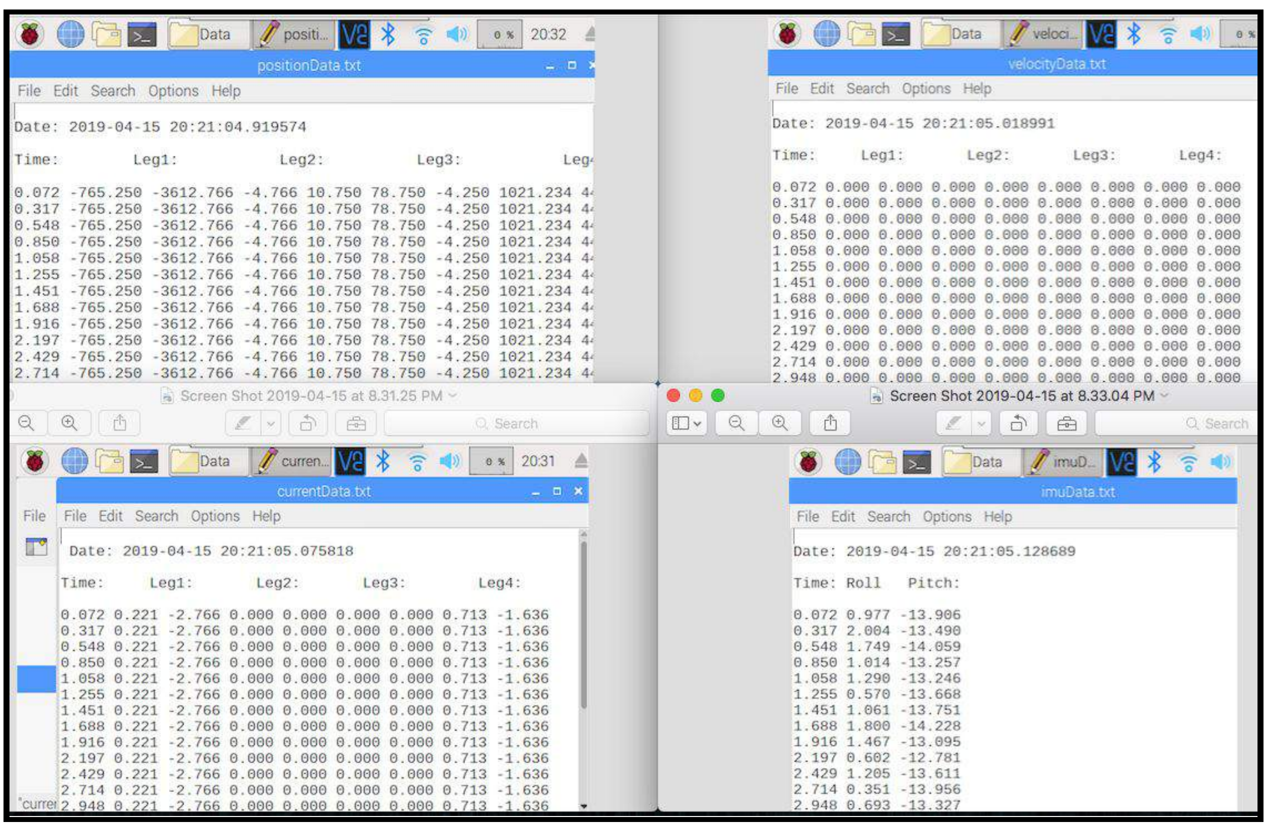


Figure 4.12. Data Files Position (top left), Velocity (top right), Current (bottom left), Imu(bottom right)

4.5.4. Processing Capabilities

The robot's computer, the raspberry pi 3 B+, is capable of processing data at a frequency of 16 MHz which is stated by its manufacturer. This was tested by conducting a benchmark test. A benchmark test is the process of load testing a component (raspberry pi 3 B+) to determine its performance characteristics. This type of test is repeatable in that the performance measurements captured will vary only a few percent each test run. This allows for changes to be made to the component in attempt to determine if there was a performance improvement or degradation. Figure 4.11 displays the maximum,minimum,current, and transition latency frequency (in kHz).

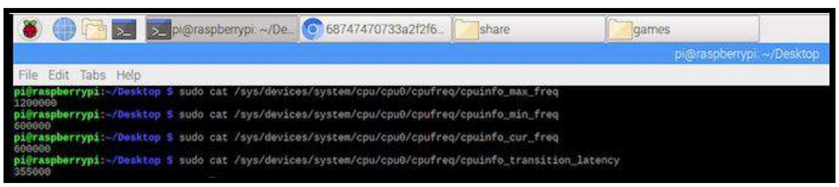


Figure 4.11. Frequency of Raspberry Pi 3 B+.

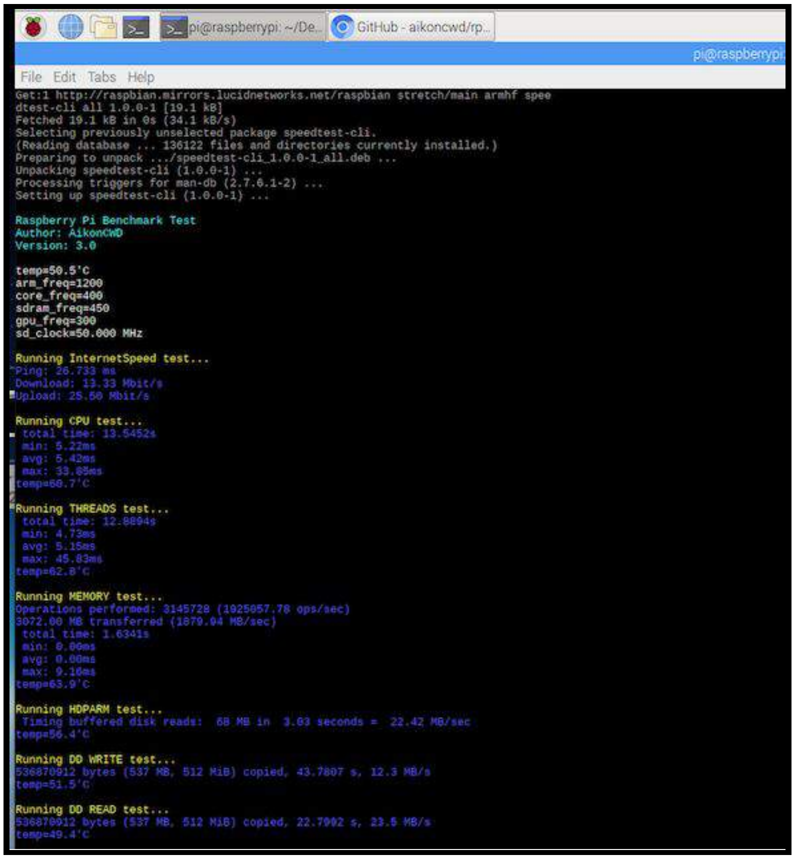


Figure 4.13. Benchmark Test for Raspberry Pi 3 B+.

From the benchmark tests displayed above, the maximum and minimum processing capabilities of the Raspberry Pi 3 b+ are 1.2 GHz and 600 MHz, respectively. This signifies that the 16 MHz required specification is comfortably met.

4.5.5. GPL3 License Verification

Attributable to the robot being open sourced, all technical drawings and their respective part files, as well as its source code must be licensed under General Public License 3(GPL3). The process of acquiring the GPL3 are as follows: copyright disclaimer was acquired from the sponsor/employer Dr. Bhounsule, as well as each file contains the proper copyright notice at the top of the file, an added COPYING file with a copy of the GPL as well as an added COPYING.LESSER file with a copy of the Lesser General Public License (LGPL). The figure

4.14 below displays a script file with the appropriate copyright information, this demonstrates that the set specification was met.

```
#!usr/bin/python
# ----------------------------------
# An Odrive class used to control lil'Bro; an open source quadrupedal robot for UTSA's Rob
otics and Motion Laboratory
#
#
# @authors Steven Farra, Emiliano Rodriguez
# @version 1.0
# @since 2019-03-11
#
#
#
#
# @section license License
#
# This file is a part of the LilBro library
#
# BroboticsInc LilBro is free software: you can redistribute it and/or modify it under the
# terms of the GNU General Public License as published by the Free Software Foundation,
# either version 3 of the License, or (at your option) any later version
#
# BroboticsInc LilBro is distributed in hope that the software will be useful, but
# WITHOUT WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE. See the GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along with LilBro.
# If not, see
# <http://www.gnu.org/licenses/>
#
# ----------------------------------

from __future__ import division
import time
import odrive
import math
import numpy as np
import globals

#AUTOMATIC  = 1
#MANUAL = 0
#DIRECT = 0
#REVERSE = 1
#P_ON_M = 0
#P_ON_E = 1

class robot():
    """This is the constructor for the robot class.

    In this constructor, the proportional gain, current limit,
    velocity limit, and drivers are initialized and delclared.

    @params self
"robot.py" 693L, 29679C written                               31,1          Top
```

Figure 4.14. Script File Copyright.

4.5.6. Interchangeability Test

Due to the design of the robot, each leg is identical, allowing them to fit into an assembly of the same type. The legs were constructed to be easily accessible as well as interchangeable in the event of damage occurring to any of the legs. Multiple legs were printed as spare parts in case the testing legs suffer any form of damage and are in need of replacement. Figure 4.15 below displays the spare legs as well as the interchangeability of the legs.

Figure 4.15. Leg Interchangeability

5. Results and Data Analysis

5.1. Data Tables

5.1.1. Mass Test Data

Table 5.1. Mass Test Matrix

	m_1	m_2	m_3	m_4	m_5	$\sum m_{1,2,3,4,5}$
TEST 1	1.532	1.544	1.536	1.537	3.371	9.520
TEST 2	1.532	1.544	1.536	1.538	3.371	9.521

TEST 3	1.533	1.544	1.535	1.538	3.370	9.521
TEST 4	1.532	1.545	1.535	1.538	3.371	9.521
TEST 5	1.531	1.544	1.536	1.538	3.371	9.520
AVERAGE	**1.532**	**1.5442**	**1.5356**	**1.5378**	**3.3708**	**9.5206**

SYMBOL	**NOMENCLATURE/NOTATION**
m_1	Mass of front right leg assembly
m_2	Mass of front left leg assembly
m_3	Mass of back right leg assembly
m_4	Mass of back left leg assembly
m_5	Mass of body assembly
$\sum m_{1,2,3,4,5}$	Robots mass or summation of (m_1, m_2, m_3, m_4, m_5)

5.1.2. Displacement/Volume Test Data

Table 5.2.. Volume/Displacement Test Matrix

	L	W	H	$V = (L \times W \times H)$
TEST 1 (Standby Position)	701 mm	507.6 mm	245.4 mm	0.0873 mm
TEST 2 (High Standing)	701 mm	507.6 mm	380.4 mm	.135 mm
TEST 3 (Extended Legs)	701 mm	507.6 mm	321.4 mm	.114 mm

SYMBOL	**NOMENCLATURE NOTATION**
L	Length
W	Width
H	Height
$V = (L \times W \times H)$	Product of Robots Volume (L x W x H)

Table 5.3. Width Error

Measurement Location	Reading	Error
Shaft to Encoder	2.1 mm	± .1 mm
Encoder to Motor Housing	9.1 mm	± .1 mm
Motor Housing to Motor Housing	484 mm	± 1 mm

Table 5.4. Standby Position Height Error

Measurement Location	Reading	Error
Bottom of the Foot to the Bottom of the Mount	140 mm	± 2 mm
Bottom of the Mount to Top of the Body	95 mm	± 3 mm
Thickness of the Cover	5.3 mm	± .1 mm

Table 5.5. Extended Legs Position Height Error

Measurement Location	Reading	Error
Bottom of the Foot to the Bottom of the Mount	140 mm	± 2 mm
Bottom of the Foot to the Top of the Mount	269 mm	± 3 mm
Bottom of the Mount to Top of the Body	95 mm	± 3 mm
Thickness of the Cover	5.3 mm	± .1 mm
Thickness of Mount	56 mm	± 1 mm

Table 5.6. High Standing Position Height Error

Measurement Location	Reading	Error
Bottom of the Foot to the Bottom of	275 mm	± 2 mm

the Mount		
Bottom of the Mount to Top of the Body	95 mm	± 3 mm
Thickness of the Cover	5.3 mm	± .1 mm

5.1.3. Linear Velocity Test Data

The matrix for this test is shown below in Table 5.3 and was used to record measurements from the test, with a brief nomenclature included for the symbols used.

Table 5.7. Linear Velocity Test Matrix.

Iteration	***x* (meters)**	***t* (seconds)**	**$\bar{v}$ (meters/second)**
1	1 m	4.87 s	0.2053 m/s
2	1 m	4.96 s	0.2016 m/s
3	1 m	4.93 s	0.2028 m/s
4	1 m	4.71 s	0.2123 m/s
5	1 m	4.73 s	0.2114 m/s
6	1 m	4.69 s	0.2132 m/s
7	1 m	5.02 s	0.1992 m/s
8	1 m	5.1 s	0.1961 m/s
9	1 m	4.75 s	0.2105 m/s
10	1 m	4.68 s	0.2137 m/s

SYMBOL	**NOMENCLATURE NOTATION**
x	Distance
t	Time
$\bar{v}$	Average Velocity

5.1.4. Additional Weight Bearing Capacity Test Data

Table 5.8. Additional Weight Bearing Test Matrix

	Z	**Total Weight**	**Time**	**Pass/Fail**
TEST 1	2.38 kg	11.9 kg	11 seconds	Pass
TEST 2	2.38 kg	11.9 kg	12 seconds	Pass
TEST 3	2.38 kg	11.9 kg	14 seconds	Pass
TEST 4	2.38 kg	11.9 kg	>> 8 seconds	Pass
TEST 5	2.38 kg	11.9 kg	>> 8 seconds	Pass

SYMBOL	**NOMENCLATURE NOTATION**
TW_1	Total Weight (including the extra 25%)
t	Time
W_1	Weight of the robot
Z	Additional 25% weight

5.2. Data Analysis

5.2.1. Mass Test Analysis

The total mass of the robot is the summation of the subassembly masses, the equation for the total mass of the robot can be seen in the equation below.

$$\sum m_{1,2,3,4,5} = m_1 + m_2 + m_3 + m_4 + m_5$$

After measuring the mass of each of the subassemblies, the summation of all masses were exceedingly under the 23 kg specification. Signifying that the mass test met its specification.

5.2.2. Displacement/Volume Test Analysis

Volume is the amount of space that an object occupies. The parameter of interest in this test is the volume that the robot takes up. In order to quantify the results, the following equation is used to find volume:

$$V = L \, x \, W \, x \, H$$

The measuring tape utilized on this test is accurate enough to yield a reliable reading to justify the space occupied by the robot. In the equation mentioned above, the volume V is found by multiplying the robot's length L, width W, and height H. The desired volume by the product of the length, width, and height is to be no more than a cubic meter, meeting the required specification for this test.

The complexity of the robots design created an issue when measuring the height and width. It was required to split the measurements into sections in order to acquire the overall measurement for height and width. With the sections created this allowed for error to propagate. An example of this can clearly be seen in table 5.5 which are the measurements taken for the height of the robot in the extended leg position. In this case this test was split into four sections due to the space and ability to measure the height in those exact locations. These locations were best suited to conduct measurements considering that they were easily accessible to the team. The first measurement taken was from the bottom of the foot to the top of the mount. The team then decided to subtract the thickness of the mount from that measurement. The reasoning behind this decision was due to the difficulty of measuring from the top of the mount to the top of the body.

There is not enough space available to insert the tape measure in the appropriate spot to measure from the top of the mount to the top of the body. Now the team had already taken the measurement from the bottom of the mount to the top of the body. In the end subtracting the mounts thickness from the initial measurement would then allow the team to use the measurement from the bottom of the mount to the top of body. The error for each measurement can be seen in table 5.4. The final measurements seen in table 5.3-5.6 include the errors for each section.

For the first position the standby position, the measurements for length, width, and height were concluded to be 701, 507.6, and 245.4 mm respectively. The volume was calculated by multiplying the length, width and height of this position. The volume was concluded to be 0.087 m^3 which is well under the 1 m^3 which states that this position meets the set specification.

The second position the extended leg position, the measurements for length, width, and height were concluded to be 701, 507.6, and 321.4 mm respectively. The height for this position was calculated by subtracting the thickness of the mount from the measurement from the bottom of the foot to the top of the foot as mentioned previously. The volume was calculated by multiplying the length, width and height of this position. The volume was concluded to be 0.114 m^3 which is well under the 1 m^3 which states that this position meets the set specification.

The final position the high standing leg position, the measurements for length, width, and height were concluded to be 701, 507.6, and 380.4 mm respectively. The height was calculated by adding the measurements of the three sections mentioned in table 5.6. The volume was calculated by multiplying the length, width and height of this position. The volume was concluded to be 0.135 mm. Concluding the measurements of all three positions, the robot met the specification of having a volume less than 1 m^3 .

The error obtain from each position was primarily due to human error. There were two team members conducting this test each had a different view of what the measurement tool was reading. Along with the different readings the fact that the measurements were split into sections allowed for the error to propagate since each section would have its own error and the summation of all errors were taken towards the end.

5.2.3. Linear Velocity Test Data

Velocity, or speed, is the rate of change of position over time. Average velocity is the parameter of interest in the linear velocity test due to the nature of the test. The overall distance *x* traveled by the robot over a period of time *t* yields average velocity $\overline{v}$ in the following equation:

$$\overline{v} = \frac{x}{t}$$

Considering that 10 iterations of this test that was carried out, a mean of all calculated average velocities were calculated to obtain a single definitive value. The mean was then calculated using the following equation:

$$\overline{a} = \frac{\sum_{i=1}^{n} a_i}{n}$$

In the context of this test, *a* represents average velocity and *n* is the number of iterations. The desired calculated average velocity would be a value greater than or equal to 0.2 m/s, signifying that the specification is met.

5.2.4. Additional Weight Bearing Capacity Test Data

Weight is the mass of an object multiplied by the gravitational force of the earth. With an additional 25% weight mounted to the robot while standing, the parameter of interest is seeing if the robot is able to support that additional weight. In order to calculate both the weight of the robot and the additional weight the following equation is applied:

$$W_1 = m * 9.8\frac{m}{s^2}$$

Now in the context of this equation W_1 can be either the weight of the robot or the weight of the additional weight to be added. The scale used to calculate the mass is accurate enough to yield a reliable value. The desired result is that of the robot managing to support an additional 25% of its weight while standing, signifying that the specification is met.

This test was run in a controlled environment within the Robotics and Motion Laboratory. As stated in section 4.4.2 the robot was not walking, it was in standby position. Once in standby position the additional 25% weight was placed on the top of the robot and was left there for about roughly eight seconds. Human error was present in calculating the exact additional weight as well as timing how long the robot lasted with the weight on top. For the additional weight it was calculated that the additional weight be 2.38 kg, when adding the condensed sand the team member was not able to acquire that value, instead the value acquired was 2.384 kg which is relatively close that the error will not have a noticeable impact on the test and robot. This led to having a tolerance of .005 mm on the additional weight. Human error was also displayed in the use of a stopwatch. This was induced to the fact that each team member will have different reflexes meaning one may stop the timer before or after the eight seconds. For all five iterations Lil'Bro was able to carry the additional weight for more than 8 seconds meeting the set specification.

Although there were minor human errors when conducting the carrying capacity test, the robot passed each iteration. This signifies that the robot has met the set specification of being able to support and additional 25% of its own weight.

6. Schedule

A detailed timeline of the tests conducted over the course of 14 days is shown in the Gantt Chart on the next page. The project as shown can be completed by the end date of May 3, 2019. The main tasks and the amount of days required to accomplish them are:

1. Development of Test Report (5 days)
2. Weight Test (1 day)

3. Dimension and Volume Test (1 day)
4. Carrying Capacity Test (1 day)
5. Walking Test (4 days)
6. Troubleshooting of the Robot (3 days)
7. Processing and Data Collection Test (1 day)

The schedule does include two deliverables. The final test report is due 16 days after the tests have been conducted. It is believed that this test schedule complies with the original Gantt chart and initial contract. The project will be completed and handed over to the sponsor on said due date.

Figure 6.1. Test Schedule GANTT Chart

7. Conclusions and Recommendations

Results for each of the tests can be seen in Table 6.1 below.

Table 7.1. Test Results.

Desired Test Ranges	Test Results
Mass ≤ 23 (kg)	9.52 kg
Volume ≤ 1 (m^3)	Yes
Linear Velocity ≥ 0.2 (m/s)	0.2066
Additional Weight Bearing ≥ 25% of Robot Mass (kg)	Yes

Table 6.2 shows the compliance matrix with an input field for whether or not a feature/specification was met and indicating it with a Y or N, for yes or no, respectively. Although only four tests included collecting data, visual observation of the remaining features suffices.

Table 7.2. Post-Testing Compliance Matrix.

Item No.	**Feature/Specification**	**Specification Ref. in Appendix A**	**Feature/Specification Met (Y/N)**
1	The robot can be controlled through a handheld device.	Section A.1.1	Y
2	The robot can move without external physical support.	Section A.1.2	Y
3	The robot moves by walking.	Section A.1.3	Y
4	The robot can support an additional 25% of its weight while standing.	Section A.1.4	Y
5	The robot can collect data in operation.	Section A.1.5	Y
6	The robot's processor is capable of processing at 16 MHz.	Section A.1.6	Y
7	The robot can move at a speed of at least 0.2 m/s.	Section A.1.7	Y
8	The robot's dimensions are within a cubic meter box.	Section A.2.1	Y
9	The robot weighs less than 23 kg.	Section A.2.2	Y
10	The robot's software is licensed under GPL3.	Section A.2.3	Y

11	The robot's leg links can be interchanged between assemblies.	Section A.2.4	Y

The primary area of concern for this project pertains to how the tests outlined in this document are contingent upon the timely completion of the software development of the robot. Although making alterations to the software is expected throughout testing, a robust base architecture is necessary for any kind of operation. To do so requires time, which is currently limited, so the team will be making an effort to accelerate the development process by eliminating planned exploratory testing.

8. Communications

Questions regarding the information in this test plan or any aspect of this project should be directed to the Team Lead at Brobotics Inc., Steven Farra (zvo618@my.utsa.edu).

Appendix D: Assembly Design Drawings

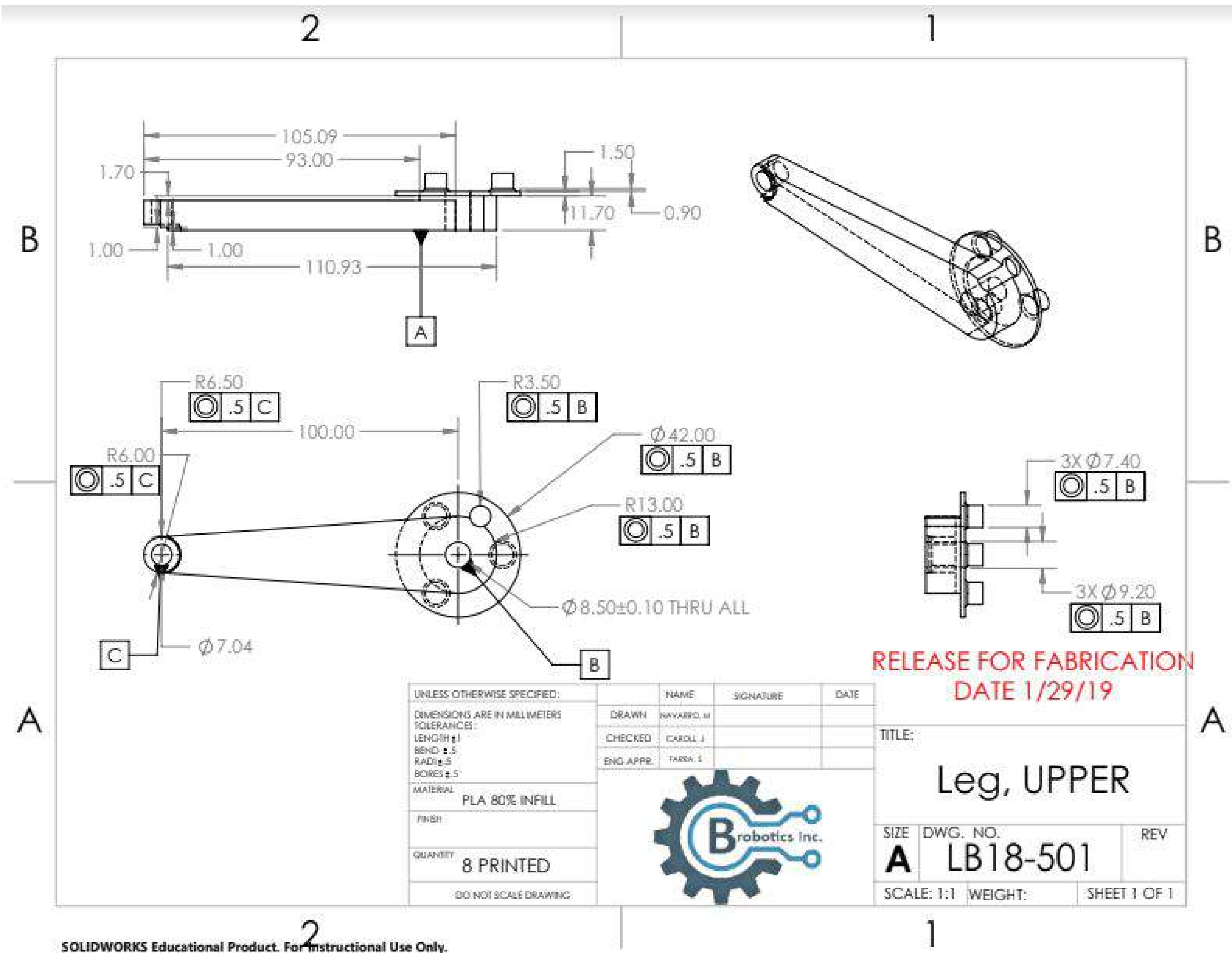


Figure 1. Upper Leg Drawing

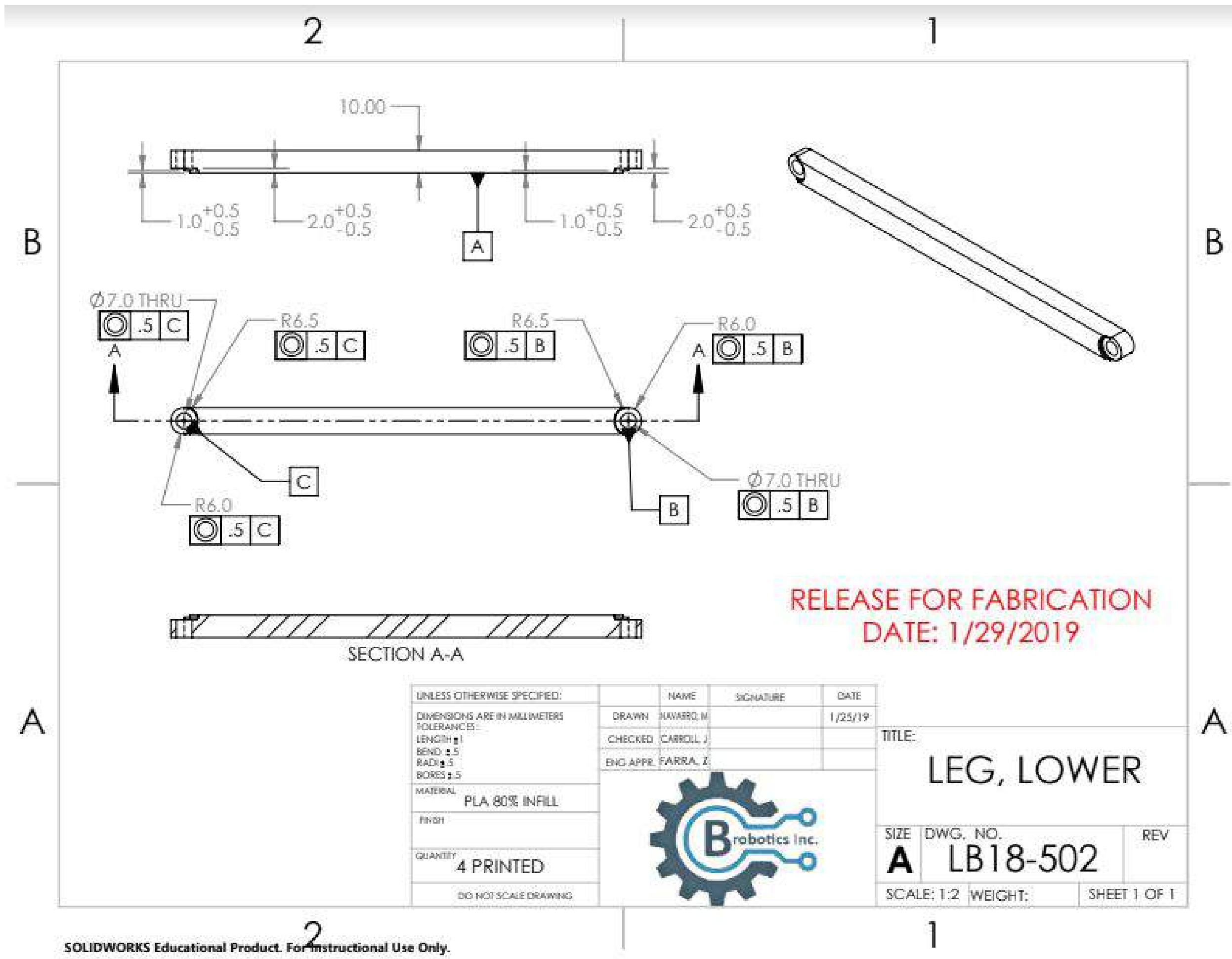


Figure 2. Lower Leg Drawing

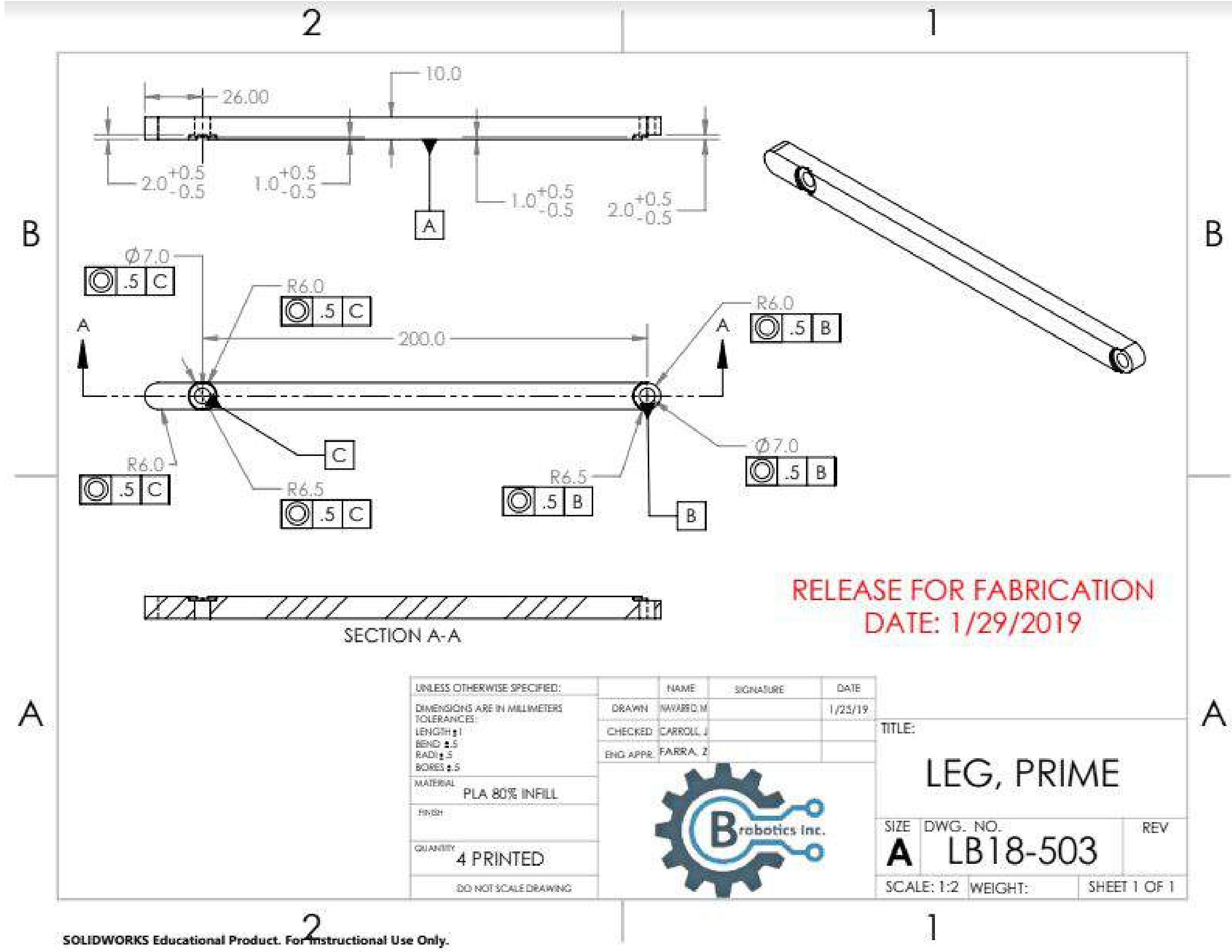


Figure 3. Prime Leg Drawing

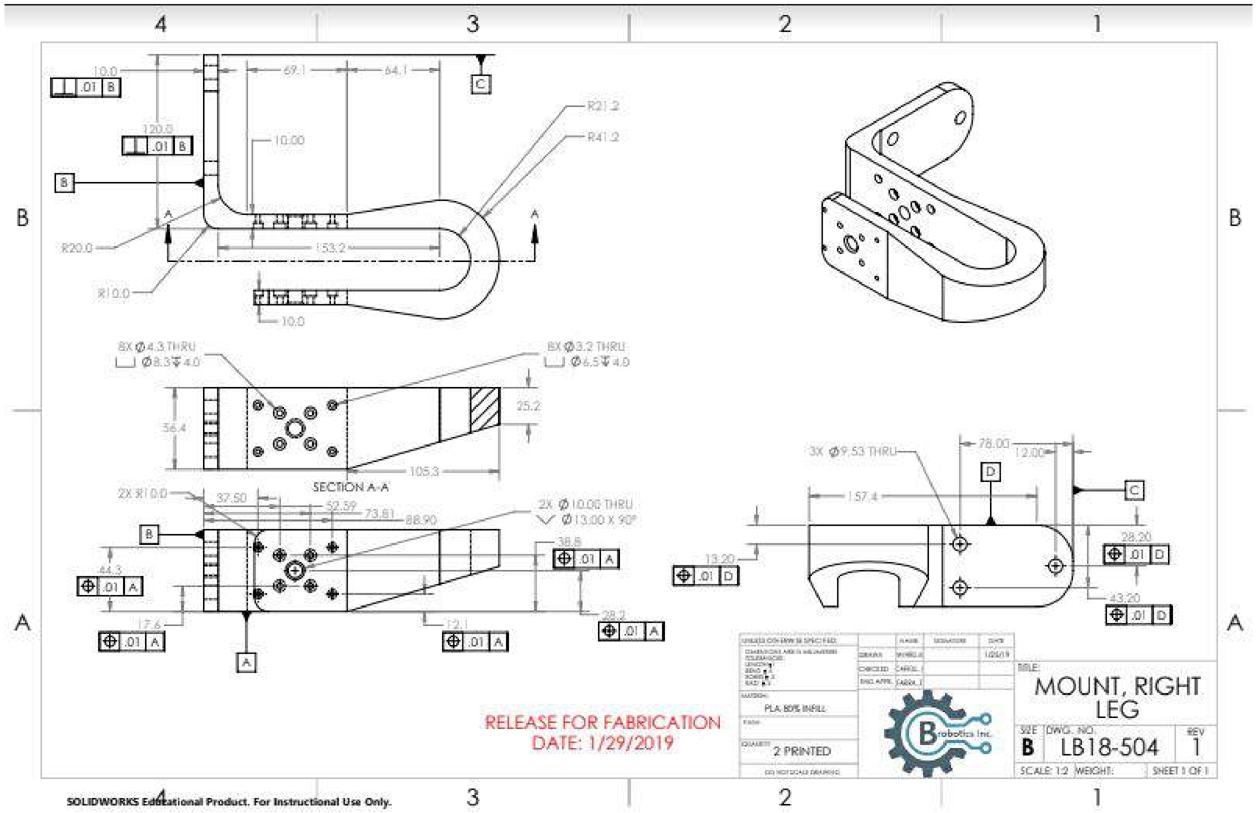


Figure 4. Right Mount Leg Drawing

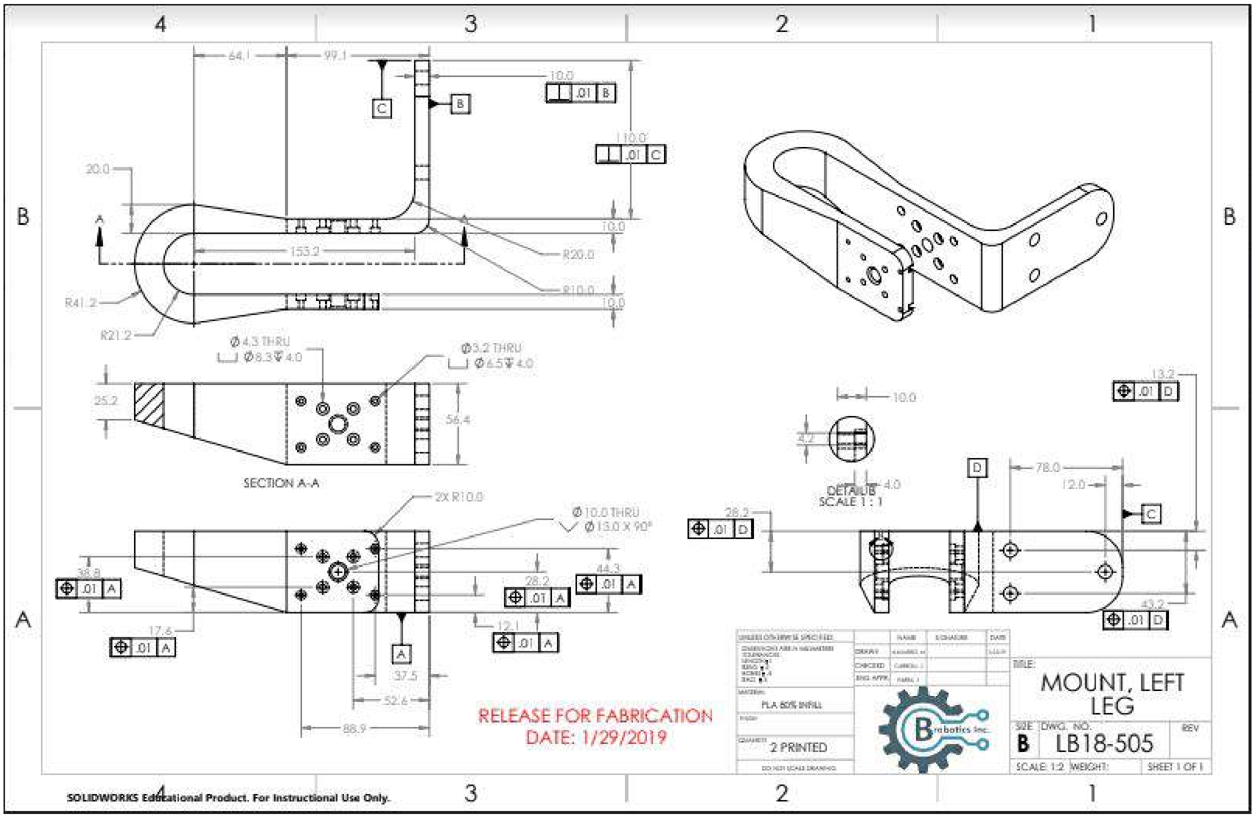


Figure 5. Left Mount Leg Drawing

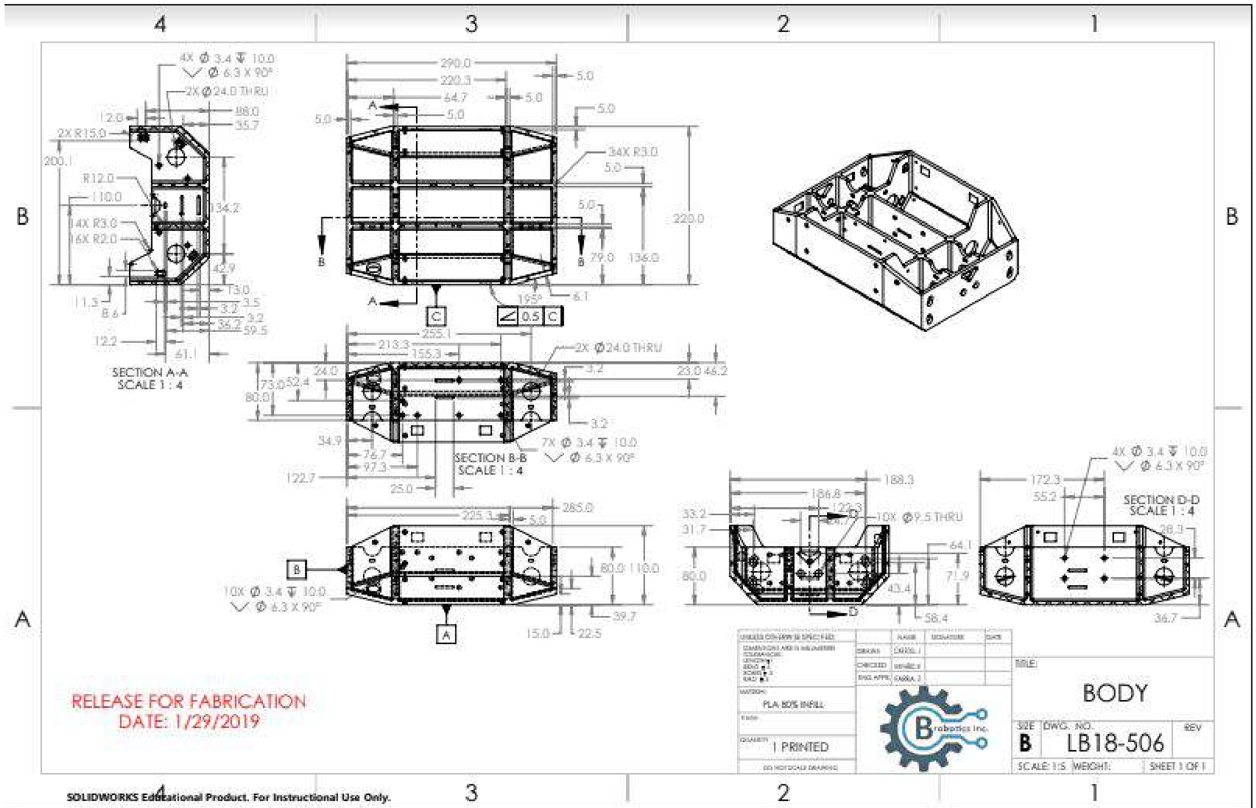


Figure 6. Body Drawing

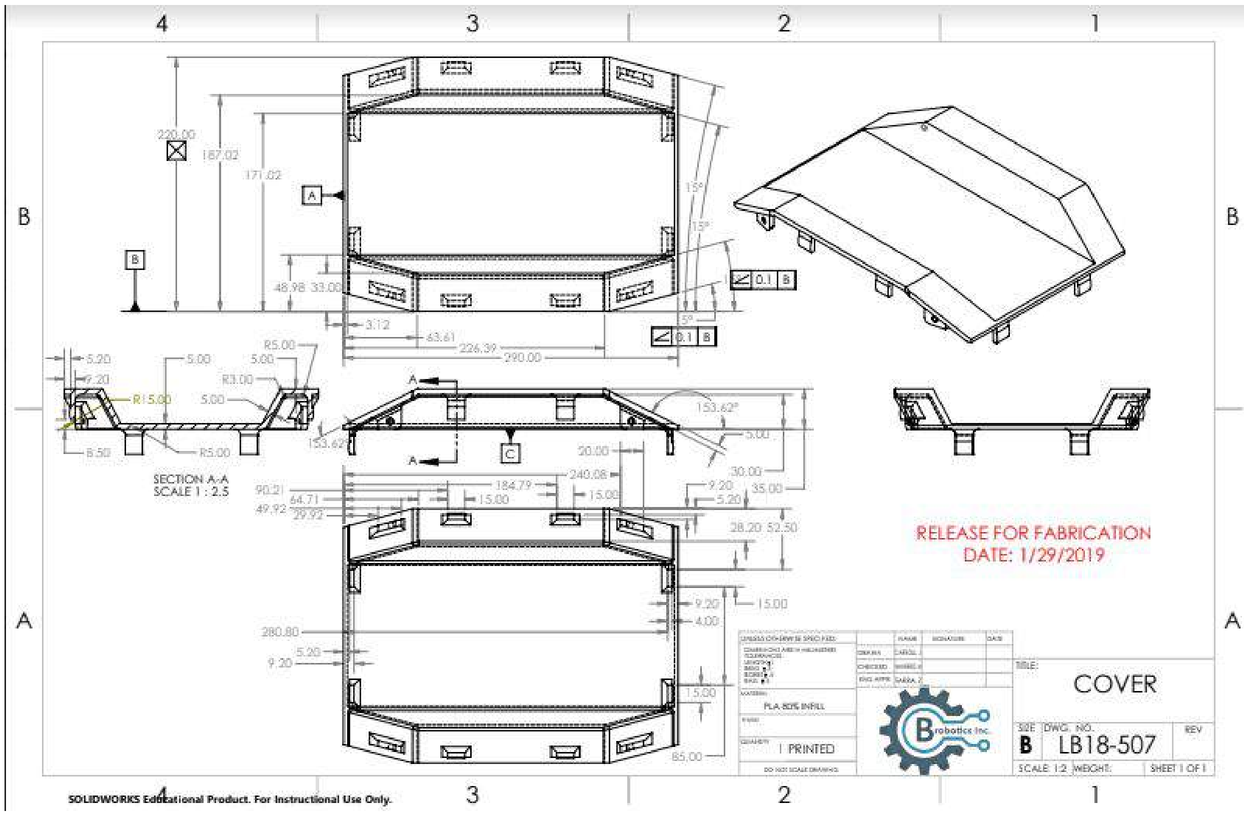


Figure 7. Cover Drawing

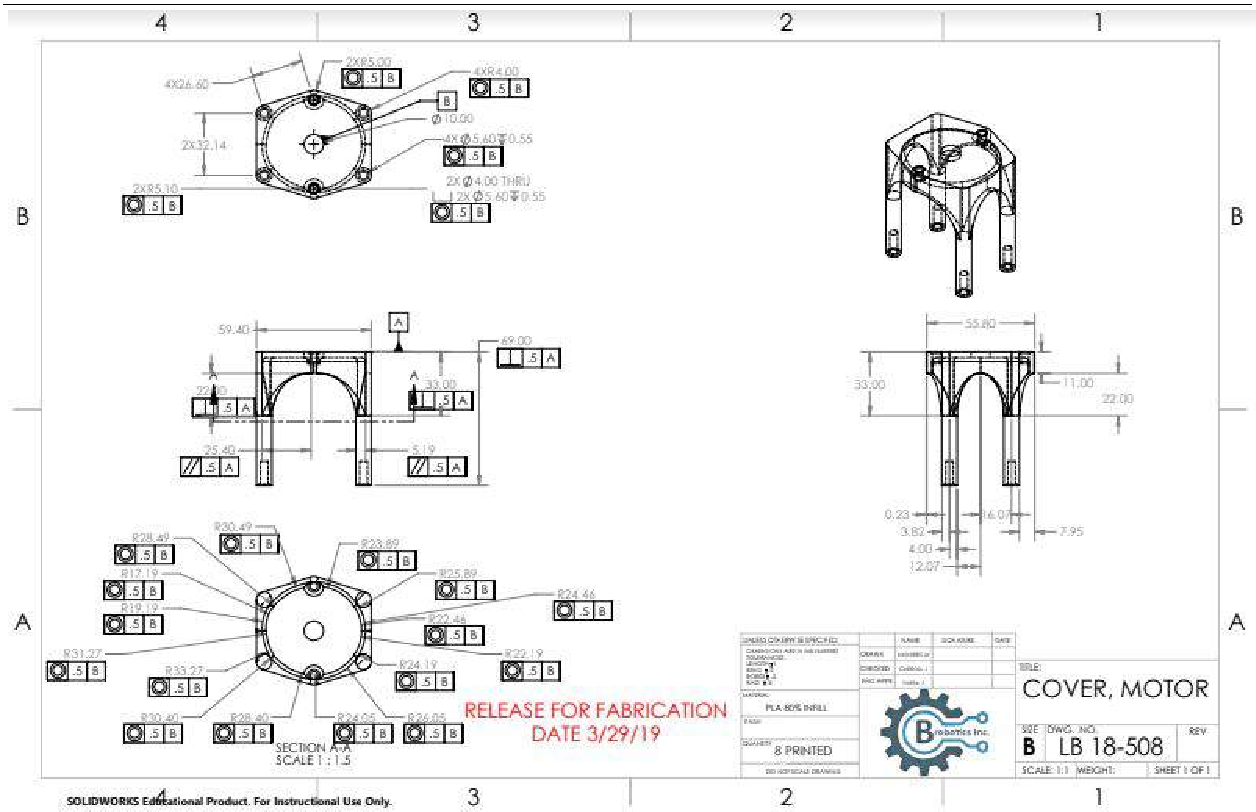


Figure 8. Motor Cover Drawing

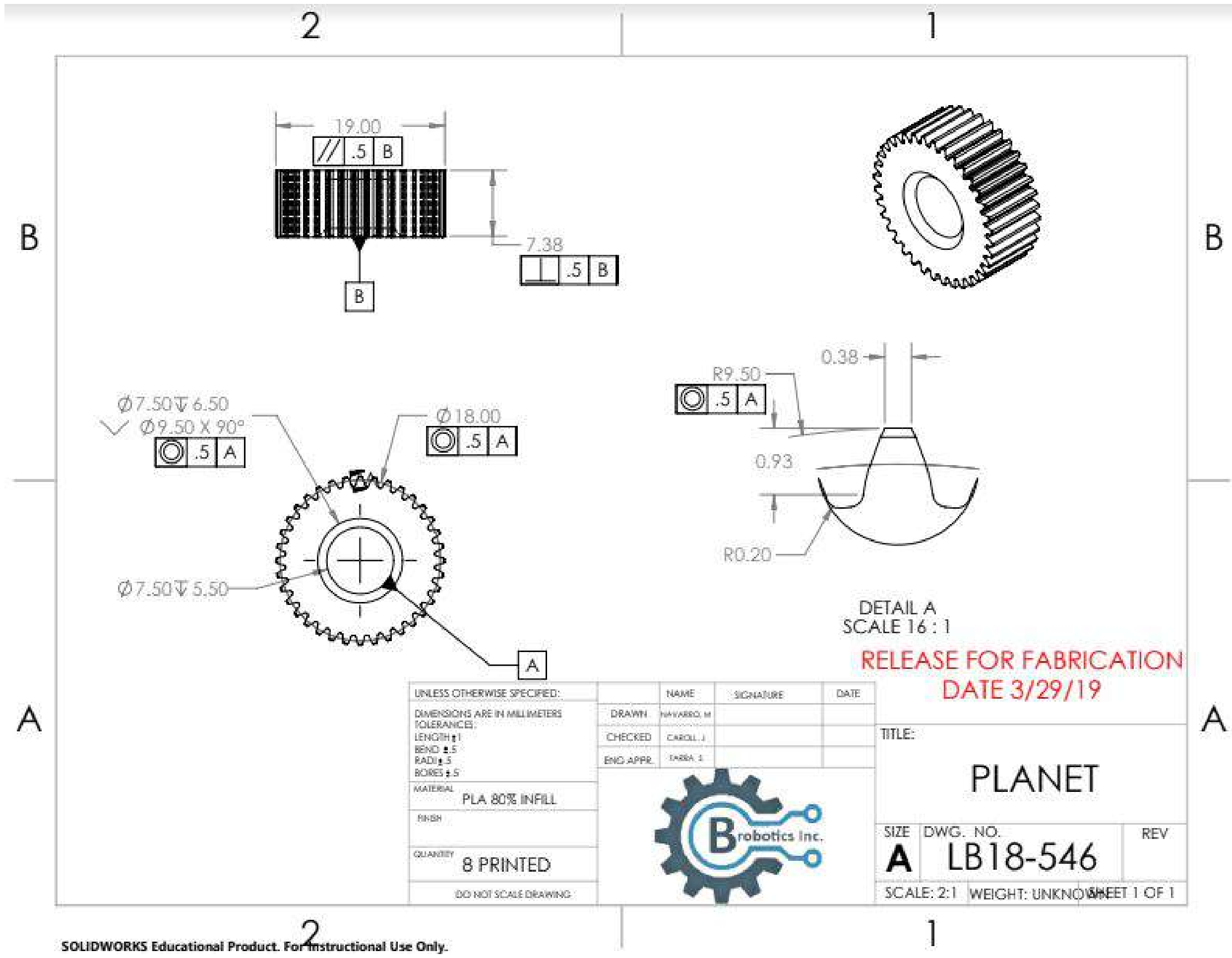


Figure 9. Planet Gear Drawing

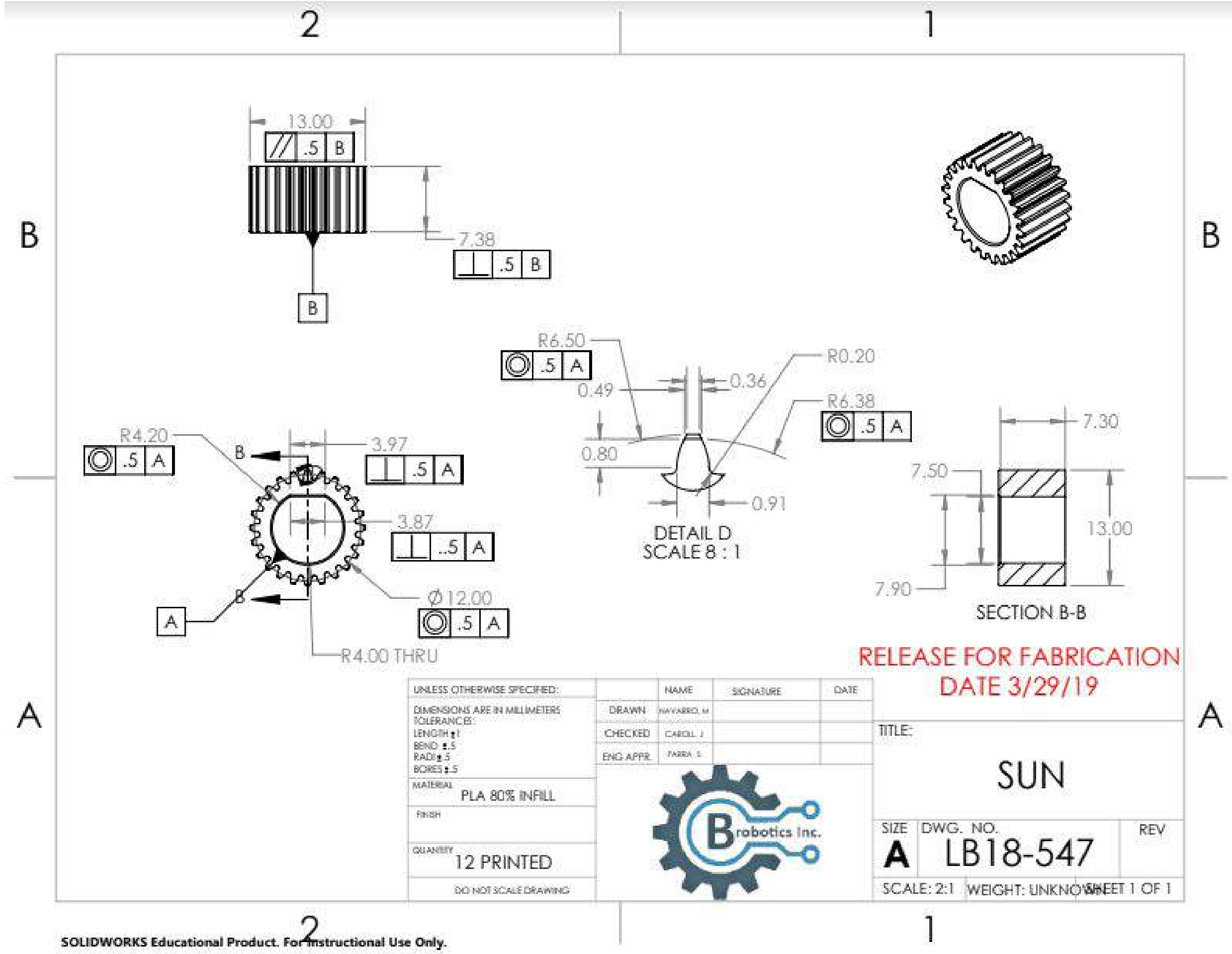


Figure 10. Sun Gear Drawing

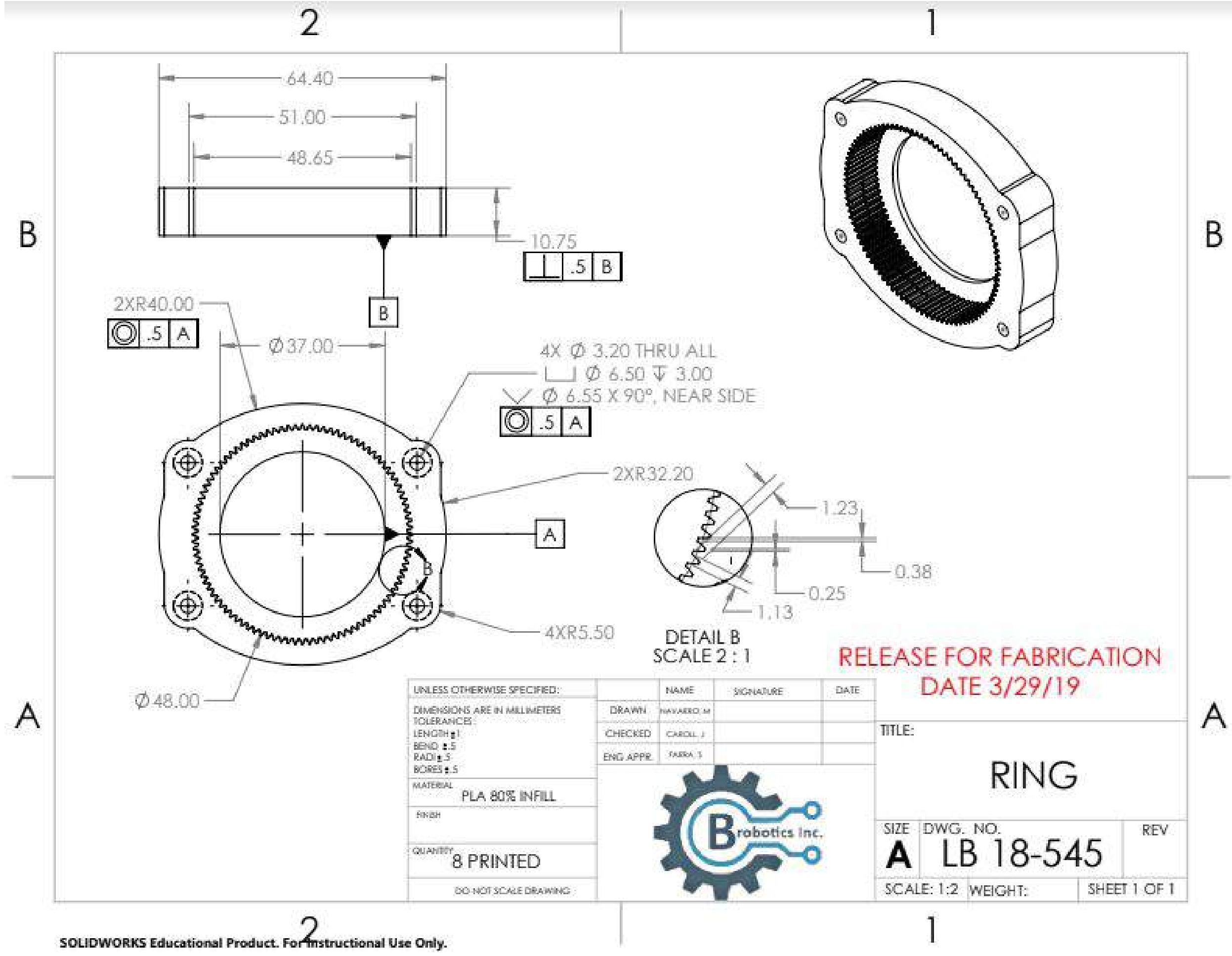
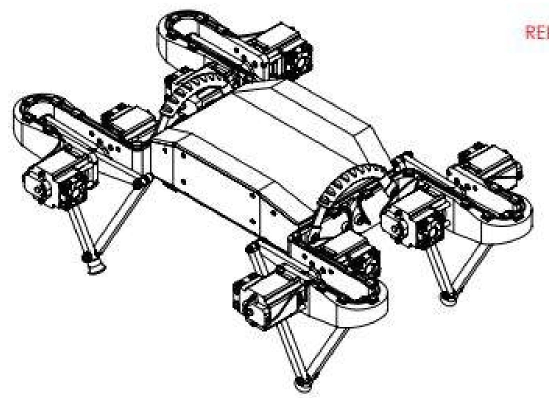


Figure 11. Ring Gear Box Drawing

GENERAL NOTES:



RELEASE FOR FABRICATION
DATE: 1/29/2019

REVISIONS				
ZONE	REV.	DESCRIPTION	DATE	APPROVED

SUB-ASSEMBLY CHART		
SUB NO.	QTY	DESCRIPTION
301	2	ASSEMBLY, RIGHT LEG
302	2	ASSEMBLY, LEFT LEG
303	1	ASSEMBLY, BODY

ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL	VENDOR OR NOTES
44	8	544	FAN, MOTOR	PURCHASE	NF-A4X10-FLX 5V, AMAZON
43	1	543	WIRE	PURCHASE	BNTECHGO10-1B, AMAZON
42	4	542	FILAMENT, 3D PRINTER	PURCHASE	MP05778, MAKERBOT
41	1	541	CONTROLLER, PS3	PURCHASE	ASIN: B07DQHY13L, AMAZON
40	9	540	CONNECTORS, WIRE	PURCHASE	B07CWFBCRX, AMAZON
39	12	539	ZIP-TIE	PURCHASE	ASIN: B07C1H2Q8L, AMAZON
38	4	538	STRAP, BATTERY	PURCHASE	ASIN: B01MY2S7JD, AMAZON
37	2	537	HANDLE, CARRY	PURCHASE	ASIN: B072XHBKPJ, AMAZON
36	1	536	SWITCH, POWER	PURCHASE	UR19B-2NO2NC, AMAZON
35	36	535	INSERT	PURCHASE	94510A240, MCMASTER-CARR
34	1	534	REGULATOR	PURCHASE	SAINSMARTLM2596, AMAZON
33	1	533	INERTIAL MEASURE UNIT	PURCHASE	GY-521MPU6050, AMAZON
32	38	532	FASTENER, STANDOFF	PURCHASE	93840A120, MCMASTER-CARR
31	34	531	NUT, STANDOFF	PURCHASE	INCLUDED W/ ITEM NO. 26
30	4	530	RESISTOR, BRAKE	PURCHASE	INCLUDED W/ ITEM NO. 28
29	1	529	BATTERY, DC	PURCHASE	9067000373-0, HOBBY KING
28	4	528	DRIVER, MOTOR	PURCHASE	ODRIVE3.5, ODRIVE ROBOTICS
27	1	527	COMPUTER, RASPBERRYPI	PURCHASE	3775, ADAFRUIT INDUSTRIES
26	26	526	STANDOFF, HEX	PURCHASE	SPC-NLBS/KIT-1, AMAZON
25	48	525	FASTENER, HUB	PURCHASE	93840A126, MCMASTER-CARR
24	8	524	HUB, MOTOR	PURCHASE	2693, POLOLU
23	4	523	FOOT, SLIP-ON	PURCHASE	2517171, MCMASTER-CARR
22	24	522	BEARING, SLEEVE	PURCHASE	6679K33, MCMASTER-CARR
21	24	521	BEARING, THRUST	PURCHASE	6655K51, MCMASTER-CARR
20	12	520	FASTENER, LEG-JOINT	PURCHASE	90278A375, MCMASTER-CARR
19	12	519	LOCKNUT, LEG-JOINT	PURCHASE	91270A310, MCMASTER-CARR
18	48	518	FASTENER, MOTOR HOUSING	PURCHASE	91292A114, MCMASTER-CARR
17	32	517	FASTENER, MOTOR	PURCHASE	93705A824, MCMASTER-CARR

ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL	VENDOR OR NOTES
16	28	516	HOLDER, CABLE	PURCHASE	7565K31, MCMASTER-CARR
15	16	515	FASTENER, ENCODER	PURCHASE	90666A104, MCMASTER-CARR
14	8	514	ENCODER, CPR	PURCHASE	CVI AMT102-V, ODRIVE
13	8	513	MOTOR, DC	PURCHASE	D5065-270KV, ODRIVE
12	4	512	FASTENER, COVER	PURCHASE	96817A274 MCMASTER-CARR
11	24	511	NUT, HEX	PURCHASE	96557A109 MCMASTER-CARR
10	12	510	WASHER, LOCK	PURCHASE	3573A16 MCMASTER-CARR
9	12	509	FASTENER, LEG MOUNT	PURCHASE	93897A542 MCMASTER-CARR
8	8	508	COVER, MOTOR	PLA	FABRICTAE
7	1	507	COVER	PLA	FABRICATE
6	1	506	BODY	PLA	FABRICATE
5	2	505	MOUNT, LEFT LEG	PLA	FABRICATE
4	2	504	MOUNT, RIGHT LEG	PLA	FABRICATE
3	4	503	LEG, PRIME	PLA	FABRICATE
2	4	502	LEG	PLA	FABRICATE
1	8	501	LEG, UPPER	PLA	FABRICATE

MATERIAL: PLA 80% INFILL

TITLE: QUADRUPED, LILBRO

SIZE B DWG. NO. LB18-101 REV A

SCALE: 1:4 WEIGHT: SHEET 1 OF 1

Figure 12. Lil'Bro Drawing

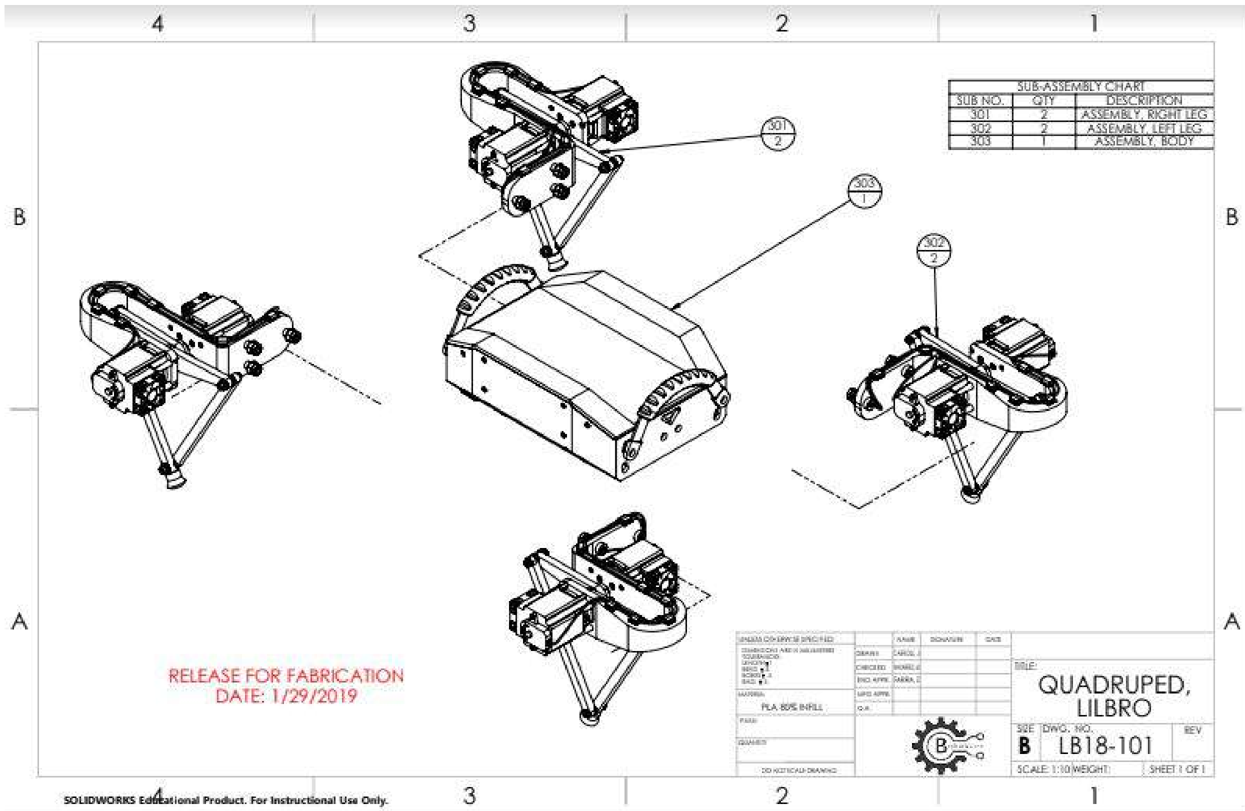


Figure 13. Exploded View Drawing