ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΚΑΙ ΑΕΡΟΝΑΥΠΗΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΤΑΣΚΕΥΑΣΤΙΚΟΣ ΤΟΜΕΑΣ ΟΜΑΔΑ ΡΟΜΠΟΤΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Simulation, fabrication and programming of a low-cost quadruped robot

Νικολέτα Παναγιώτα Παπαδοπούλου

1059726

Παναγιώτης Κουστουμπάρδης, ΕΔΙΠ

Διπλωματική εργασία υποβληθείσα στο Τμήμα Μηχανολόγων Μηχανικών & Αεροναυπηγών του Πανεπιστημίου Πατρών

ΠΑΤΡΑ, [02/2024]

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΚΑΙ ΑΕΡΟΝΑΥΠΗΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΤΑΣΚΕΥΑΣΤΙΚΟΣ ΤΟΜΕΑΣ ΕΡΓΑΣΤΗΡΙΟ ΡΟΜΠΟΤΙΚΗΣ

Η παρούσα διπλωματική εργασία παρουσιάστηκε

από την

Νικολέτα Παναγιώτα Παπαδοπούλου

την 06/03/2024

ΠΕΡΙΛΗΨΗ

Προσομοίωση, κατασκευή και προγραμματισμός τετράποδου χαμηλού κόστους

Νικολέτα Παναγιώτα Παπαδοπούλου

Την τελευταία δεκαετία, έχουν σημειωθεί σημαντικές εξελίξεις στον τομέα των τετράποδων ρομπότ και η Open-Source κοινότητα έχει συμβάλει σημαντικά προς αυτή την κατεύθυνση. Εμπνευσμένη από το πρωτοποριακό έργο του Maurice Rahme στο Github, η παρούσα διπλωματική εργασία αφορά την κατασκευή και τον προγραμματισμό ενός τετράποδου ρομπότ χαμηλού κόστους, το OpenQuadruped/SpotMiniMini. Το SpotMiniMini διαθέτει ένα πλήρως τρισδιάστατα εκτυπωμένοπλαίσιο, 12 σερβοκινητήρες χωρίς ψήκτρες που κινούν τέσσερα πόδια 3 βαθμών ελευθερίας το καθένα. Ο ενσωματωμένος υπολογιστής του ρομπότ είναι ένα Raspberry Pi 4B που τρέχει ως λειτουργικό σύστημα το Ubuntu 20.04 LTS headless server. Το λογισμικό του τετράποδου είναι κατασκευασμένο μέσα σε περιβάλλον ROS Noetic. Αυτό το πλαίσιο δημιουργεί μια υποδομή επικοινωνίας που ενσωματώνει το RPi με έναν μικροελεγκτή Teensy 4.0 για τον χαμηλού επιπέδου έλεγχο των σερβομηχανισμών του ρομπότ. Το τελευταίο είναι εφικτό μέσω του σειριακού πρωτοκόλλου επικοινωνίας UART. Επιπλέον, το ρομπότ διαθέτει αισθητήρα γυροσκοπίου IMU που παρέχει δεδομένα για την ανίχνευση προσανατολισμού του και μπορεί να λαμβάνει εντολές μέσω ενός joystick από ένα ασύρματο gamepad που είναι συνδεδεμένο στο RPi. Μια μπαταρία λιθίου τροφοδοτεί την ειδικά σχεδιασμένη για το ρομπότ πλακέτα η οποία διανέμει σε όλα τα εξαρτήματα του ρομπότ ρεύμα. Για τη δημιουργία τροχιών των ποδιών του, το τετράποδο χρησιμοποιεί μια μέθοδο D^2- Randomized Gait Modulation με καμπύλες Bezier.

Λέξεις κλειδιά

Λογισμικό Ανοιχτού Κώδικα, Χαμηλού Κόστους, Ρομπότ, Τετράποδο, ROS

ABSTRACT

Simulation, fabrication and programming of a low-cost quadruped robot

Nikoleta Panagiota Papadopoulou

In the last decade, there have been major advancements in the area of quadruped robots and the Open-Source community has contributed tremendously towards this direction. Inspired by the groundbreaking work of Maurice Rahme on Github, this thesis presents a low-cost opensource quadruped robot development, the OpenQuadruped/SpotMiniMini. SpotMiniMini features a fully 3D printed frame, 12 brushless servo motors that drive four 3-DOF legs. The robot's onboard computer is a Raspeberry Pi 4B that runs as its operating system an Ubuntu 20.04 LTS headless server. The quadruped's software is built inside a ROS Noetic environment. This framework establishes a communication infrastructure that integrates the RPi with a Teensy 4.0 microcontroller for the low-level control of the robot's servos. The latter is possible through serial UART communication protocol. Moreover, the robot has an IMU gyroscope sensor that provides data for orientation sensing and can receive joystick commands from a wireless gamepad connected to the RPi. A lithium battery powers a PCB that distributes it to all the hardware components of the robot. Finally, the Quadruped uses a D2-Randomized Gait Modulation method with Bezier Curves for foot trajectories generation.

Key Words

Open-Source, Low-cost, Robot, Quadruped, ROS

LIST OF TABLES

LIST OF FIGURES

SYMBOLS & ABREVIATIONS

3D	Three Dimensional
CCPM	Cyclic/Collective Pitch Mixing
DOF	Degree of Freedom
GPU	Graphics Processing Unit
GUI	Graphical User Interface
L×W×H	Length × Width × Height
LLC	Low Level Control
PDB	Power Distribution Board
QDD actuator	Quasi Direct Drive actuator
RL	Reinforcement Learning
RPY	Roll, Pitch, and Yaw angles
ROS	Robotic Operating System
UBEC	Universal Battery Elimination Circuit

TABLE OF CONTENTS

1.0 INTRODUCTION

Scientists and Engineers have often gained inspiration from nature and the development of legged locomotion is no exception. The reason for that is that creatures ranging from insects and small mammals to humans are able to effortlessly achieve complex locomotion in unstructured environments with uneven terrains and yet current control methods are neither robust nor adaptable enough to deliver similar results in artificial systems. Hence, seeking inspiration from biology has become a crucial strategy in order to improve legged locomotion in robots. Nature's most imitated structure in robotics is arguably quadruped robots (or four-legged robots) and are often chosen over other legged robot configurations, like biped, hexapods, or even more exotic designs because of their stability, versatility, and real-world practicality. Some of today's most characteristic examples of quadruped robotics are to be found in the chapter *Literature Review* that follows.

The push for creating open-source platforms for quadrupedal robots is equally compelling, with researchers, hobbyists, and students from all over the world enabling the creation of frameworks towards low-cost, accessible quadrupedal locomotion. This approach of shared resources not only reduces costs regarding the development of quadruped robots, bridging the technological divide, but also serves an educational purpose, allowing hands-on learning and experience gaining to potential roboticists. As these platforms can introduce a degree of standardization, customization becomes simpler, and compatibility issues are reduced. Lastly, the

transparency inherent in today's robotic projects can promote trust as robots become more intertwined with our daily lives.

The inspiration behind this thesis, the development of the low-cost quadrupedal Spot mini mini [1] is drawn from one of the most prevalent and recognizable four-legged robots in the industry, Boston Dynamics's Spot robot [2]. Spot is a versatile quadruped that bears great resemblance to a dog's shape and movement. Equipped with multiple advanced sensors and sophisticated algorithms, it has the ability to map its environment, identify obstacles and move autonomously in uncertain terrains. The quadruped has a great number of capabilities and applications, ranging from inspection tasks in inaccessible environments [3], contribution to AI research in universities [4], uses in the entertainment industry to even utilization of Spot for security purposes and patrol [5]. Despite its numerous capabilities, Spot's price can exceed $75000. As a result, SpotMicroAI [6] was originally introduced by **Deok-yeon Kim** and released on Thingiverse, with a series of low cost updated versions of the Open-source quadruped following. The development of one of these versions, OpenQuadruped by Maurice Rahme and Adham Elarabawy is the focus of this thesis. However, the two collaborators used different approaches regarding the built of the robot. The first, incorporated a Raspberry Pi 4B as the robot's "brain" powering it with one lipo battery and was named OpenQuadruped/SpotMiniMini while the later chose a Jetson Nano and two lipo batteries for the quadruped. Apart from these differences regarding the computer board, their approaches hardware-wise is similar, both using a Teensy for the low-level control (LLC). The author of this thesis had previous experience with the RPi platform and therefore the OpenQuadruped/SpotMiniMini built was chosen and will be mentioned in the rest of this thesis as such.

However, by the time of development of this thesis the distribution of Ubuntu that the SpotMiniMini originally operated was reaching End of Standard Support and has now transitioned to 18.04 EOL (End of Life). This means that Canonical, the company behind Ubuntu is no longerproviding security updates, bug fixes, or official support for this particular version. So, with the goal of updating the robot's software and allowing future members of the open-source communityto utilize and build SpotMiniMini further, the whole project was transferred to Ubuntu 20.04 LTS(Focal Fossa). The updated and revisited code of the robot, as well as the bug fixes on the source code can be found on GitHub forked under the original project and constitute the further contribution of this thesis in the SpotMiniMini project.

2.0 LITERATURE REVIEW

With the study of biomechanics dating all the way back to the seventeenth century [7], legged locomotion is now considered one of the most popular research topics in bio-inspired robotics. In the scope of this literature review, several notable four-legged robotic platforms are going to be discussed. Inspired by the locomotion and agility of animals, their design provides many advantages in terms of stability and adaptability to various uneven terrains, as well as potential use in tasks where bipedal and wheeled robots ought to face challenges.[1]

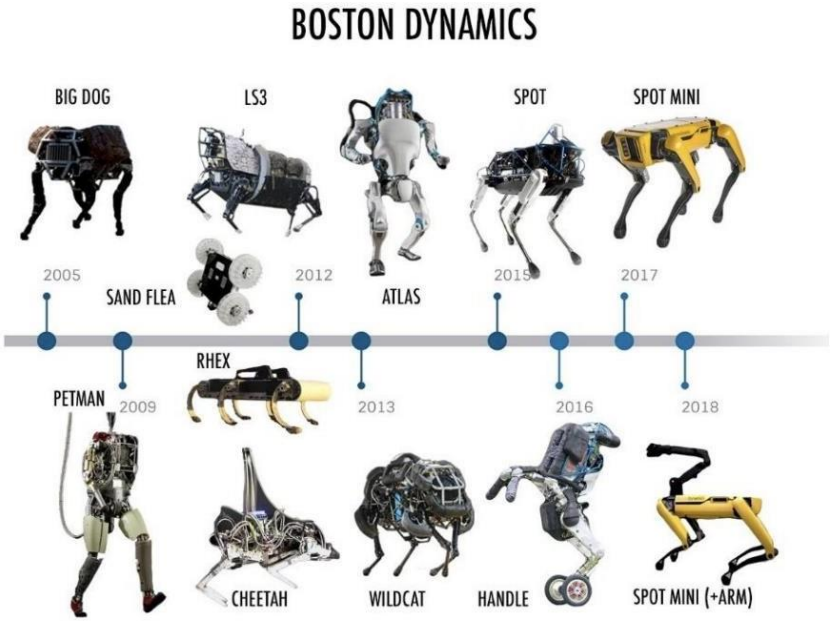


Figure 2-1 Boston Dynamics robotic achievements timeline [37]

Boston Dynamics, is a Massachusetts-based robotics firm renowned for creating advanced robots with exceptional mobility, agility, and dexterity. is considered to be one of the most

[1] In the chapter of *Literature Review*, a focused yet limited review pertinent is provided about contemporary quadrupedal robotic platforms. A comprehensive review regarding the most prominent four-legged robots, their actuators and leg architectures has been conducted separately and can be found in N. P. Papadopoulou, "Quadrupedal Bio-inspired Robots: Development & Characteristics", 2022 [36]

influential and important pioneers in the field of robotics and especially in quadrupedal locomotion for the past two decades as seen in the timeline of *Figure 2-1*.

Figure 2-2 Spot by Boston Dynamics [2]

Perhaps Boston Dynamic's most recognized by the public and latest quadrupedal robot Spot Mini as shown in *Figure 2-2* often referred as "Spot", presents a new approach to dynamic robot control [2]. The untethered dog-like robot can run up to 5.76 km/h with its dimensions being 1.1 m×0.5 m×0.84 m (L×W×H). Spot weighs only 30kg and has the capability of carrying 14kg of load. All the joints of the robot are electrically actuated and Spot, being powered by battery, is meant for sensing and inspection in remote or hazardous environments. It is self-charging, allowing it to autonomously perform routine or on-demand data collection without human interaction [8]. Its built-in dock detection and tablet interface allows it to return home to charge at the push of a button or call of a task, without requiring operator directions. The Spot has an inbuilt 3D vision system with simultaneous localization and mapping (SLAM), providing depth information, enabling the robot to navigate its surroundings, and avoid obstacles. Besides its basic capabilities, the bioinspired Spot leverages hardware to further improve safety, communications

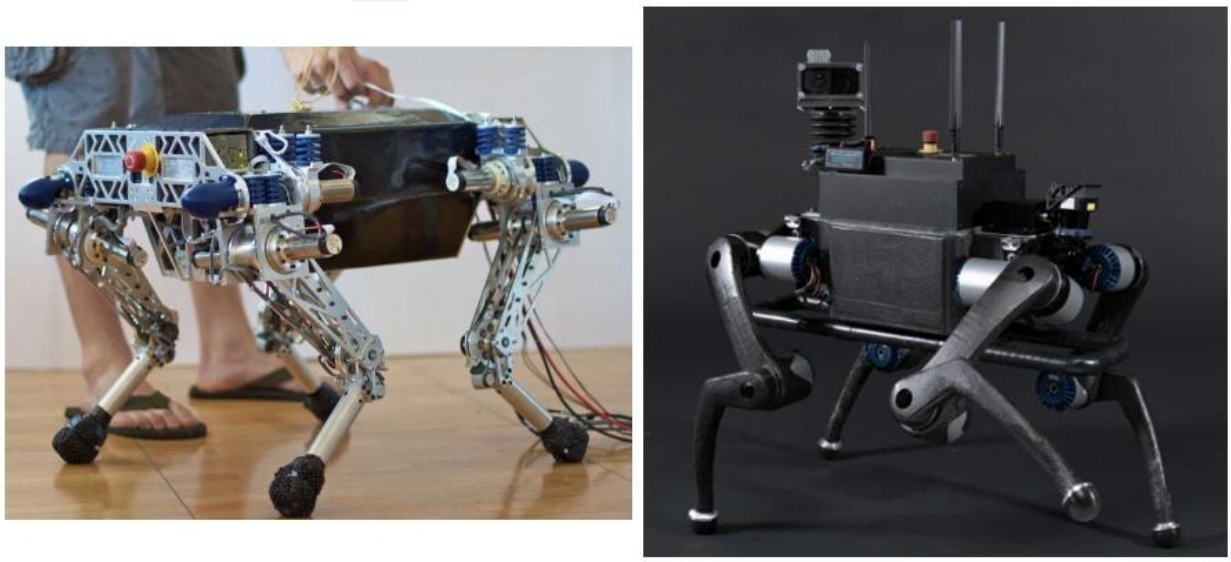
and behaviour on remote sites being able to climb, descend stairs, balance, and adjust to physical disturbance. The quadruped Spot becomes a most light-footed robot that can work in the office, home, and outdoor using its 5 DOF sensor arm. Interestingly enough, in 2019, Spot was utilized by Massachusetts State Police in two unnamed incidents for evaluating the robotic dog's capabilities in law enforcement applications, especially for remote inspection of potentially dangerous environments. Also, in 2021 Spot was reported being used by French army were part of a project by the École Militaire Interarmes school at a French army camp Saint-Cyr Coëtquidan for reconnaissance [9].

The quadruped made in the Swiss Federal Institute of Technology, is named StarlETH (Springy Tetrapod with Articulated Robotic Legs) [10], [11]. The robot is of lightweight construction in order to achieve fast locomotion, weighing around 23kg and reaching the velocity of 1 m/s. Star1ETH as shown in *2-3* has 3 DOF in each leg, a series of highly elastic actuators in order to replicate the way muscles and tendons behave. Moreover, the robot is allowed high fidelity joint torque control and protection from the impact loads, due to mechanical springs that separate the motors and overall gearbox from the joints. The kinematic data obtained by encoders mounted on the quadruped, alongside the IMU, allow estimation of position and maneuvering regardless of additional perception or an additional motion capture system. Lastly pressure sensors are added at the compliant ball feet, so as to provide feedback regarding the contact situation of the robot.

Robotic Systems Lab from the ETH institution also designed ANYmal [12], shown in Figure *2-3*. A multipurpose quadruped specifying in commercial and industrial operations that have the potential of being used for example, in applications of oil and gas sites inspection with additional features such as accurate localization, navigation and mapping. The robot's joint unit

deploys series elastic actuators with highly integrated electronics, sensors, and joint axle bearings for advanced interaction. Hence, the robotic legs of ANYmal do not require additional sensors, and joint axle bearings [13], [14].

MIT's most recent quadruped robot was introduced in 2019, under the name Mini Cheetah



2-3 StarlETH and ANYmal by ETH Zurich university [11], [12]

[15]. This quadruped shown in *Figure 2-4* is a lightweight, low-cost, and high performance partially open source quadruped robot. It weighs 9kg and can run up to 2.45 m/s while also having the ability to demonstrate highly dynamic behaviors including trot, trot-run, bounding, and pronking (spring into the air) and successfully land 360◦ backflips from standing. Similarly to ANYmal quadruped, Mini Cheetah utilizes SEAs (series-elastic actuators), with each actuator containing an electric motor, single stage planetary transmission, and power electronics. In order to achieve the low-cost "factor", Mini Cheetah bears motors originally designed for remote control

drones and airplanes instead of custom-built actuators like its predecessors. The four identical legs of the robot were made to achieve wide range of motion and minimize limb mass and inertia.

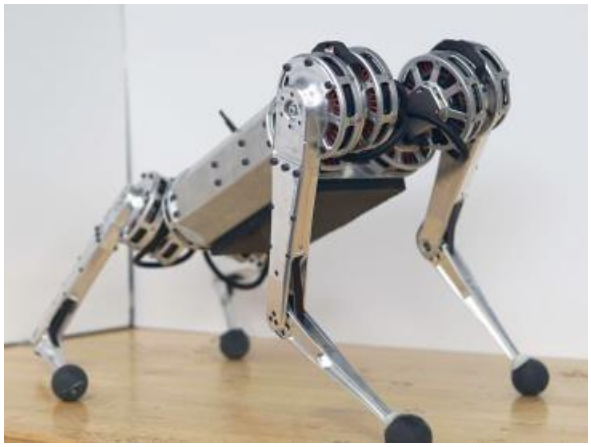


Figure 2-4 Mini Cheetah [15]

The TITAN-XIII is the last model of the TITAN series [16] and sprawling-type quadruped robot driven by tendons as shown in *Figure 2-5* .Its symmetric design gives it the ability to walk efficiently in any direction by trot gait with 1.38 m/s speed while being battery powered. The robot's legs bear the unique design of a wire-driven mechanism that includes a brushless direct current (DC) motor, a planetary gear set, two synthetic fiber cables, and input and output pulleys. In order to compensate for this design expansion, a coaxial tensioner shaft is installed as the input pulley to the wire-driven mechanism. By using the wire-driven mechanism to transmit power to each axis, the inertia is reduced since the motors can be placed at the base of the leg, with this particular architecture allowing for easier disassembly and therefore, easier maintainability. Furthermore, as a transmission and reduction mechanism to drive the hip yaw joint a timing belt mechanism is utilized, which is composed of a small timing pulley, a big pulley, and a timing belt.

Figure 2-5 Sprawling type quadruped TITAN-XIII [16]

The Minitaur [17] as shown in *Figure 2-6*, developed at the University of Pennsylvania in collaboration with Ghost Robotics, is a quadruped robot with symmetrically driven 5-bar linkages used as directly driven legs, meaning there are no gears or other mechanisms between the legs' drive motors and the legs themselves. This leads to an increased power efficiency as compared to motors with gears which may only have a maximum of 90% efficiency. This led to the Minitaur being able to bound at a speed of 1.5m/s and jump about 50 cm vertically.

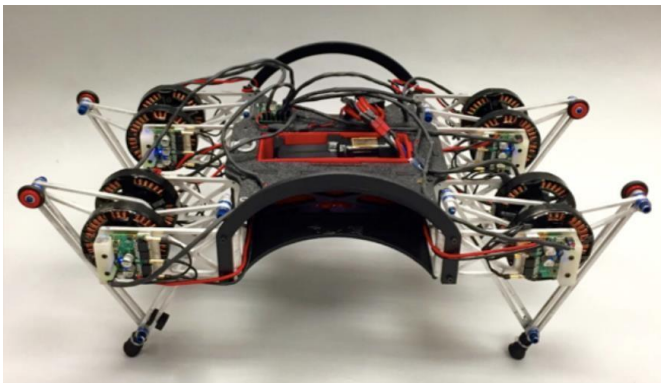


Figure 2-6 Minitaur by Ghost Robotics [17]

The contribution of the Open-source community to the evolution of low-cost, accessible quadruped robots has been extremely significant in the past few years. Along with the development of the Spot Micro Open quadruped family (*Introduction*), there have been noted othernoteworthy robotic platforms like the more recent PADWQ [18]. This dynamic quadruped robot presented in 2021, PADWQ (pronounced pa-dook) features 12 DOFs, driven by QDD actuators with high control bandwidth and can be seen *in Figure 2-7.* Additionally, the robot bears an onboard depth sensor and a GPU-equipped computer that allows for movement in uncertain terrains. The quadruped is made of off the shelf components and 3D printed plastic making it affordable to reproduce, yet still in a much higher price point of approximately $7700 comparing for example to Spot Mini Mini's $650.

Figure 2-7 The Open-source PADWQ robot [18]

The Department of Mechanical Engineering of Stanford University presented in 2019 Stanford Doggo [19] and in 2021 Stanford Pupper [20]. Stanford Doggo, shown in *Figure 2-8* is driven by quasi direct drive (QDD) actuation and the four-legged robot is capable of demonstrating characteristics of dynamic locomotion; steady velocity during running and height jumping as well

as maintaining vertical jumping agility. Despite its advanced features, it is open-source, and can be made using hand tools at a cost below $3000. In 2021, the department also introduced Stanford Pupper, shown in *Figure 2-8*, an open-source quadruped robot designed to make legged robotics research more affordable and aimed for standardization of the platform across institutions. This robot, made from readily available components or 3D printed parts, is easy to assemble in under 8 hours and costs less than $2000.

Figure 2-8 The Open source Standford Pupper and Stanford Doggo[19], [20]

3.0 DEVELOPMENT OF SPOT MINI MINI

3.1 PHYSICAL DESIGN & KINEMATICS

Spot Mini Mini's design is drawing inspiration from the model of SpotMicroAI [6] (seen in *Figure 3-1*-left) yet, the modifications performed on the original design suggest improvements regarding its weight distribution since the body is shortened by 40mm. Moreover, all SpotMiniMini's servos were moved to the hip of the robot making the lower part of the leg lighter by 65g (servo's weight). The quadruped's leg architecture is to be discussed further below. The modifications that took place resulted in a more efficient approach, requiring less torque. To enhance the structural integrity of the robotic leg, a support bridge was added connecting the said servo, upper part of the leg and the shoulder (seen in *Figure 3-2*). Finally, extra space was created for the electronics of the robot with the addition of an adapter plate (as seen in *Figure 3-3*).

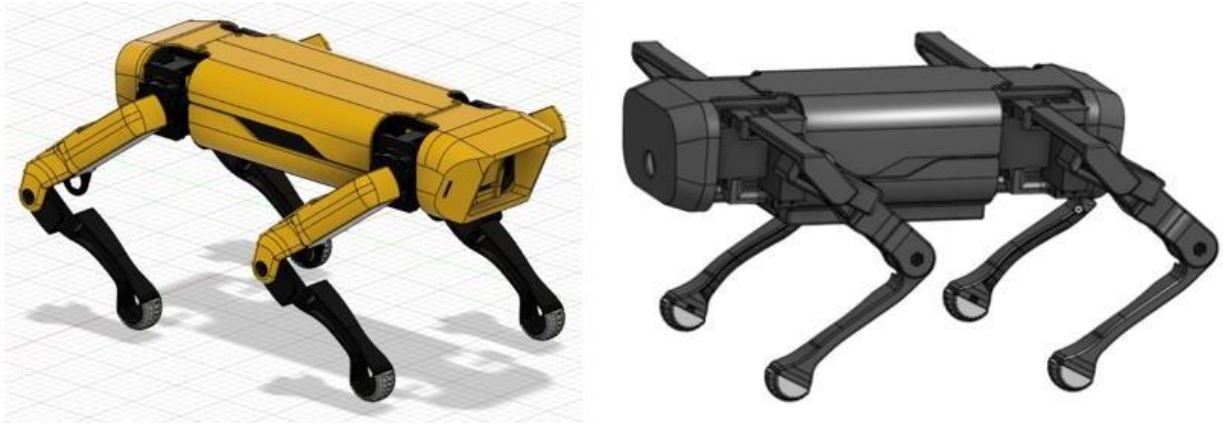


Figure 3-1 3D Visualization of SpotMicro [6] and Spot Mini Mini [1]

Figure 3-2 3D representation of Spot Mini Mini's leg structure

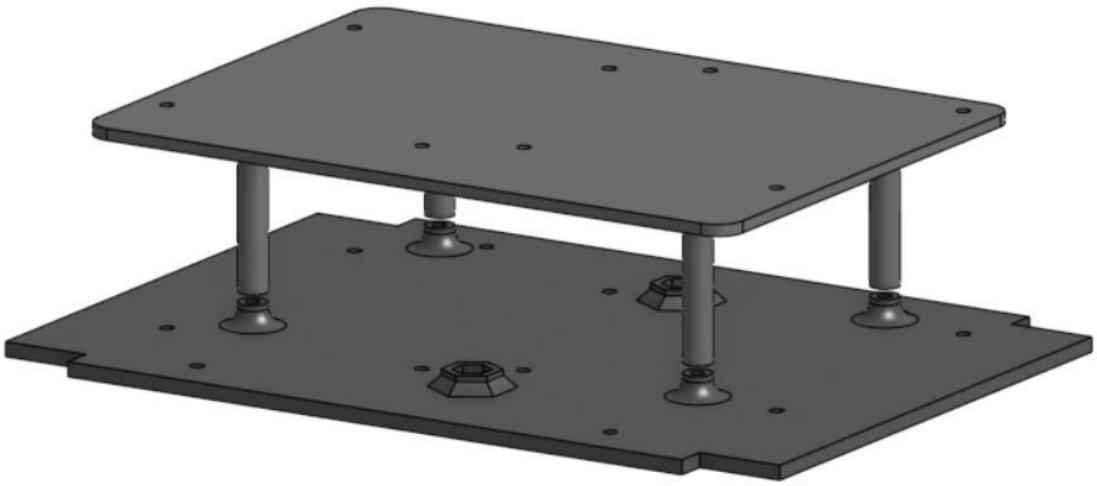


Figure 3-3 Adapter Plate (upper part)

3.1.1 3D Printing Study

The first step into the realization of Spot Mini Mini was to gather all the parts, and 3D print all its individual compartments. 3D printing is an additive layer manufacturing method to create 3- dimensional objects. For that purpose, the 3D printer Zortrax m200 was used, available at the Robotics' Laboratory.

The 3D printer Zortrax m200 provides comprehensive slicing and 3D printing management software named Z-SUITE. The standard triangle language (.stl) files were imported in Z-SUITE and in order to ensure the integrity and light-weightiness of the parts a filling of 60-80% was chosen. Zortrax m200 requires its unique compatible printing filaments for operation and Z-ABS and Z-ULTRAT were used. The filament Z-ABS is an affordable material often used in prototyping processes that was used to print the covers of the robot and the feet endings in orange color as well as the "shoulder bridges" in blue Z-ABS (as seen in *Figure 3-8*). For the rest of the robot's body, the ABS based material Z-ULTRAT in black color was chosen because of its resistance to high temperatures and impacts.

The Zortrax m200 printer demonstrated great performance during this almost 103 hour print process, requiring almost minimal maintenance. However, it is important to mention that with the aim of keeping the cost of the robot's build as low as possible, the filaments that are compliant with the 3d printer appear to be about two to three times more expensive than similar materials on the market. Both Z-ABS and Z-ULTRAT filaments were readily available at the university's laboratory and therefore, are not added to the final cost of the robot. The total weight of the robot's 3D printed parts adds up to 1309g, without calculating into this amount a few number

of test prints that took place in order to study and decide the ideal filament percentage for the parts and the total printing time is calculated to approximately 105 hours. A list of all 3D printed parts can be found in the APPENDIX A section *below*.

3.1.2 Kinematics Analysis

Fundamentally, an Inverse Kinematics (IK) model aims to convert some intuitive domain information, like xyz cartesian coordinate system, into directly useful values, like motor angles. The IK model of SpotMiniMini was derived from the article titled: "Inverse Kinematic Analysis of a Quadruped Robot". For the aim of finding the IK model, firstly the Denavit-Hartenberg (DH) method was used for the forward kinematic. The inverse kinematic equations obtained by the geometrical and mathematical methods were coded by the team in MATLAB and a program was obtained to calculate the legs joint angles corresponding to desired various orientations of robot and endpoints of legs [21]. The physical model of the quadruped robot is shown in *Figure 3-4* The

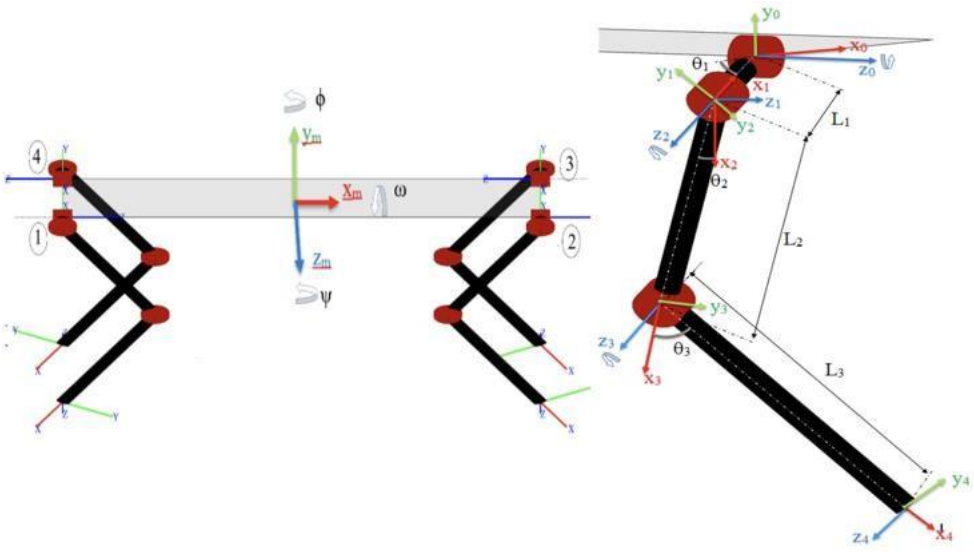


Figure 3-4 The physical model of the quadruped robot & coordinate frames of robot's leg [21]

parameters of robot are given in Table 1. The various configurations of the robot body are determined by the coordinates of its legs. Hence, the kinematic equation examines the rotational movements (φ, ψ, ω) relative to the center of the body's coordinate system (x_m, y_m, z_m) and the coordinate system of each leg endpoint (x_4, y_4, z_4).

Table 1 Parameters of Robot[21]

Physical Dimensions	The Length of Robot	L
	The Width of Robot	W
	The Length of Side Swing Joint	L1
	The Length of Hip Joint	L2
	The Length of Knee Joint	L3
Coordinate Systems	The Coordinate System of centre of Body	[x_m, y_m, z_m]
	The Main Coordinate System of Each Leg	[x_0, y_0, z_0]
	The Coordinate System of Side Swing Joint	[x_1, y_1, z_1]
	The Coordinate System of Hip Joint	[x_2, y_2, z_2]
	The Coordinate System of Knee Joint	[x_3, y_3, z_3]
	The Coordinate System of Endpoint of Leg	[x_4, y_4, z_4]
Variables	The Yaw Angle of Robot	ϕ
	The Pitch Angle of Robot	ψ
	The Roll Angle of Robot	ω
	The Angle of Side Swing Joint	θ_1
	The Angle of Hip Joint	θ_2
	The Angle of Knee Joint	θ_3

Initially, the transformation matrix (5) is obtained using the rotation matrices (1), (2), (3).

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) & 0 \\ 0 & \sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$R_y = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$R_z = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$R_{xyz} = R_x R_y R_z \quad (4)$$

$$T_M = R_{xyz} \times \begin{bmatrix} 1 & 0 & 0 & x_M \\ 0 & 1 & 0 & y_M \\ 0 & 0 & 1 & z_M \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The kinematic equation the centre of body's coordinate system (x_m, y_m, z_m) and The Main Coordinate System of each leg (x_0, y_0, z_0) is given by the transformation matrices given in (6), (7), (8), (9).

$$T_{rightback} = T_M \times \begin{bmatrix} cos(\pi/2) & 0 & \sin(\pi/2) & -L/2 \\ 0 & 1 & 0 & 0 \\ -\sin(\pi/2) & 0 & \cos(\pi/2) & W/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$T_{rightfront} = T_M \times \begin{bmatrix} cos(\pi/2) & 0 & \sin(\pi/2) & L/2 \\ 0 & 1 & 0 & 0 \\ -\sin(\pi/2) & 0 & \cos(\pi/2) & W/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$T_{leftfront} = T_M \times \begin{bmatrix} cos(-\pi/2) & 0 & \sin(-\pi/2) & L/2 \\ 0 & 1 & 0 & 0 \\ -\sin(-\pi/2) & 0 & \cos(-\pi/2) & -W/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$T_{leftback} = T_M \times \begin{bmatrix} cos(-\pi/2) & 0 & \sin(-\pi/2) & -L/2 \\ 0 & 1 & 0 & 0 \\ -\sin(-\pi/2) & 0 & \cos(-\pi/2) & W/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

In order to move the robot endpoint to the desired position, it is necessary to determine the rotational values of the joints with inverse kinematic analysis. The legs are in different orientations with each other but in the same structure, so it is sufficient to investigate the forward and inverse

kinematics analysis of a single leg. The coordinate systems and the angular positions of the right front leg joints can be seen in *Figure 3-4* . The Denavit-Hartenberg parameters for forward kinematics of the leg are given in Table 2.

Table 2 The Parameters of Denavit-Hartenberg [21]

Link	α_{i-1}	a_{i-1}	d_i	θ_i
0-1	0	L1	0	θ1
1-2	-π/2	0	0	-π/2
2-3	0	L2	0	θ2
3-4	0	L3	0	θ3

The transformation matrices are given in equations (10), (11), (12), (13) and the forward kinematic matrix is given in equation (14).

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & -L_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & -L_1\sin(\theta_1) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$T_1^2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$T_2^3 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & L_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & L_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$T_3^4 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & L_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & L_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$T_0^4 = T_0^1 T_1^2 T_2^3 T_3^4 = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \quad (14)$$

Table 3 The Elements of the Forward Kinematic Matrix [21]

$m_{11} = \cos(\theta_2)\cos(\theta_3)\sin(\theta_1) - \sin(\theta_1)\sin(\theta_2)\sin(\theta_3)$
$m_{12} = -\cos(\theta_2)\sin(\theta_1)\sin(\theta_3) - \cos(\theta_3)\sin(\theta_1)\sin(\theta_2)$
$m_{13} = -\cos(\theta_1)$
$m_{14} = L_2\cos(\theta_2)\sin(\theta_1) - L_1\cos(\theta_1) + L_2\cos(\theta_2)\cos(\theta_3)\sin(\theta_1) - L_3\sin(\theta_1)\sin(\theta_2)\sin(\theta_3)$
$m_{21} = \cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - \cos(\theta_1)\cos(\theta_2)\cos(\theta_3)$
$m_{22} = \cos(\theta_1)\cos(\theta_2)\sin(\theta_3) + \cos(\theta_1)\cos(\theta_3)\sin(\theta_2)$
$m_{23} = -\sin(\theta_1)$
$m_{24} = L_3\cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - L_2\cos(\theta_1)\cos(\theta_2) - L_3\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) - L_1\sin(\theta_1)$
$m_{31} = \cos(\theta_2)\sin(\theta_3) + \cos(\theta_3)\sin(\theta_2)$
$m_{32} = \cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3)$
$m_{33} = 0$
$m_{34} = L_2\sin(\theta_2) + L_3\cos(\theta_2)\sin(\theta_3) + L_3\cos(\theta_3)\sin(\theta_2)$
$m_{41} = 0$
$m_{42} = 0$
$m_{43} = 0$
$m_{44} = 1$

After obtained the transformation matrices and forward kinematic matrices required for the inverse kinematic solution of the Quadruped Robot, the inverse kinematic analysis isperformed using analytical methods and the equations expressing the angular position of the joints are found (15), (16), (17). There are nonlinear equations in the solution of inverse kinematics problems. For every mathematical expression computed, there may not be a physical solution. Also, there may be more than one solution for the legs endpoint to go to the desired position. For this reason, the legs of the robot (1 and 3) and the leg of the robot (2 and 4) have been realized in the same kinematic structure but in different configurations.

$$\theta_1 = \text{-atan2}(\text{-}y_4, x_4)\text{-atan2}(\sqrt{x_4^2 + y_4^2 - L_1^2}, \text{-}L_1) \quad (15)$$

$$\theta_2 = \text{atan2}(z_4, \sqrt{x_4^2 + y_4^2 - L_1^2})\ \text{-atan2}(L_3\sin(\theta_3), L_2 + L_3\cos(\theta_3)) \quad (16)$$

$$\theta_3 = atan2\left(-\sqrt{1-D^2}, D\right) (For\ legs\ 1\ and\ 3)$$

$$\theta_3 = atan2\left(\sqrt{1-D^2}, D\right) (For\ legs\ 2\ and\ 4) \quad (17)$$

$$D = (x_4^2 + y_4^2 - L_1^2 + z_4^2 - L_2^2 - L_3^2)/(2L_2L_3)$$

After deriving the Inverse Kinematics for each leg, the next step was to describe the IK for the body itself [22]. The approach used here considers a world frame *w*, which is the robot centroid's base position, and a body frame *b*, describing the robot's pose relative to the world frame. In addition, T_{ws}, which is a transform from the world frame to the robot's shoulder: this describes the base transform between the robot centroid and the shoulder. Finally, the inputs: T_{wb} which describes the desired transform from world to body (RPY or Roll-Pitch-Yaw and Translation), and T_{bf}, the desired foot position relative to the transformed body - this is useful for gait generation. The output of our process is T_{sf}, the transform between each shoulder and its respective foot required to achieve this motion - this is fed into the leg IK solver to retrieve joint angles. In the *Figure 3-5* bellow, an example of how the robot's elements, body frame, world frame and inputs "behave" in Roll pose.

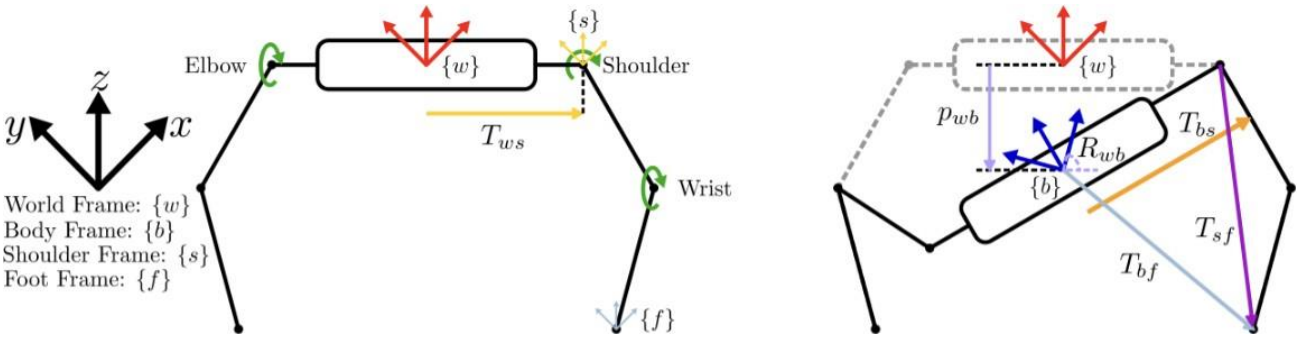


Figure 3-5Coordinate frames of robot's body in Roll pose[22]

3.2 HARDWARE

The collection of SpotMiniMini's parts was completed in a matter of a couple of months, considering that their sourcing came from either China, or vendors from Athens specialized in the robotics department. The *Table 4* that follows is the list of materials that were needed for the realization of this project, omitting the tools or consumables needed (like M2, M3, M5 bolts and matching nuts, loom sleeves, dupont lines, etc.). What follows is an explanation of how these compartments were utilized in building the quadruped and could also be considered a very brief and to the point assembly guide for the same purpose. Assembling the quadruped was an intuitive process yet a very tedious task that required good understanding of its design and function.

Table 4 List od Spot Mini Mini's hardware components

JX Ecoboost CLS6336HV (36KG)
5V UBEC
Custom PDB (pack of 5)
XT60 to T Plug
Gens ace 5500mAh 7.4V 2S1P 60C Lipo Battery-T Plug
Rocker Switch
Disc Servo Horns
Raspberry Pi 4B (4GB RAM)
SD Card (64GB)
BNO080 IMU sensor
Teensy 4.0
625 Ball Bearings (no flange)
6mm Wide 300mm Long Timing Belt

The powerful *JX Ecoboost CLS6336HV servos* are equipped with a brushless motor used for the actuation of the robot's legs and ensuring reliable performance. Twelve in total, these servos can rotate 180° and are considered the muscles that move the quadruped's legs with each leg bearing three of them. A servo located on the inner part of the body (shoulder joint), a servo located on the inner part of the upper leg structure (elbow joint) and a third servo located on the outer part of the upper leg. The latter one, through a 3d printed pulley that drives a durable driving belt (*6mm Wide 300mm Long Timing Belt*), is moving the lower part of the leg and can be considered the wrist joint. The placement of the joints can be easily observed in the 3D representation of the robot's leg design in *Figure 3-2* _bookmark20.The design approach that incorporates a driving belt – pulley structure, apart from lightening the leg, results in smoother motion. The *Disc Servo horns*, fastenedwith screws and securing nuts to the robot's connecting positions, are attached to the servo's shaft allowing this way the conversion of the servo's rotational movement into linear movement of its legs.

The Ball Bearings reduce friction where smooth movement is essential and enables two different movements. The first one by enabling the rotation of the hip joint component, seen in *Figure 3-6* (above), with the robot's body structure also seen in *Figure 3-6*(bellow). It is important to note that the hip joint component bears two servos: the one from the shoulder joint and the one from the elbow joint.

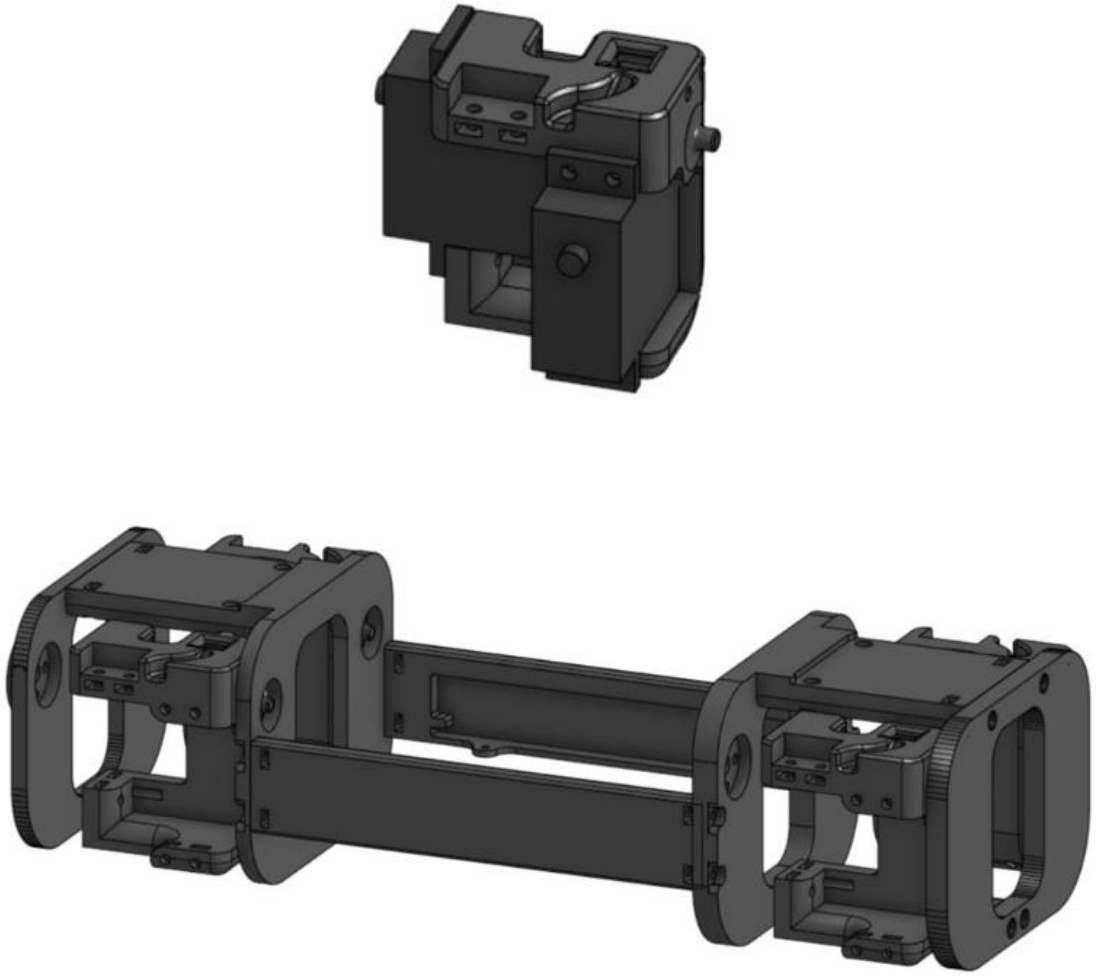


Figure 3-6 (Above) Hip Joint component of Spot Mini Mini (Bellow) Body structure and hip joint components of Spot Mini Mini

The second movement is regarding the movement between the shoulder bridge and the servo cover supporting the servo located on the outer part of the upper leg, as can be seen in Figure 3-7. The shoulder bridge serves the purpose of assisting the structural integrity of the robot legs and, connecting it securely with the hip joint component and therefore, the rest of the body. In the same photo, in lighter gray, can be seen a 3D printed idler in the middle of the upper leg structure.

This compartment is responsible for keeping the timing belt stretched enough so that the transmission of motion from the pulley to the lower leg can be possible.

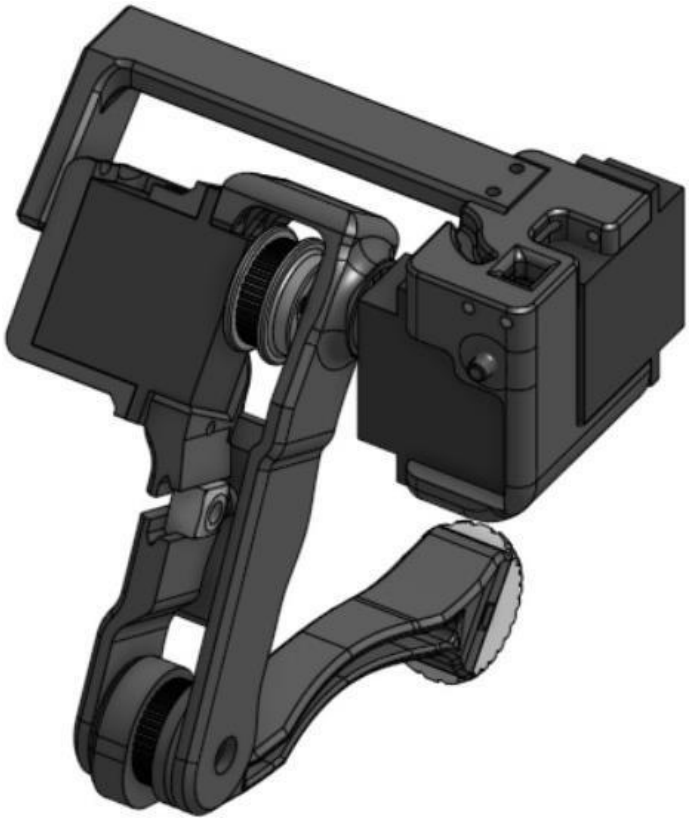


Figure 3-7 Spot Mini Mini leg and shoulder joint component

Variations and imprecisions in the process of the 3d printing and assembly, could have resulted in slight differences in the dimensions of the robotic components, including both the leg segments and joint interfaces of the robot's shoulders. These variations can affect the alignment and placement of the servo motors, leading to inconsistencies in the movement of the robot's servos. Moreover, the coupling of the servo motors with the servo horns cannot be considered perfect. Although the spline size of the servo horn corresponds to the one of the servo motor shaft, when connected together, it is inevitable that an angle occurs relative to the neutral position of the

servo coupling. To deal with these non-linearities it was imperative to give focus on the calibration of the motors. The process is discussed *below*.

After the main body of the robot and the legs were fastened together, what followed was the soldering and connecting of the electronics parts. More about this process is going to be discussed in the chapter below. Finally, the covers of the robot where added resulting in its final form.

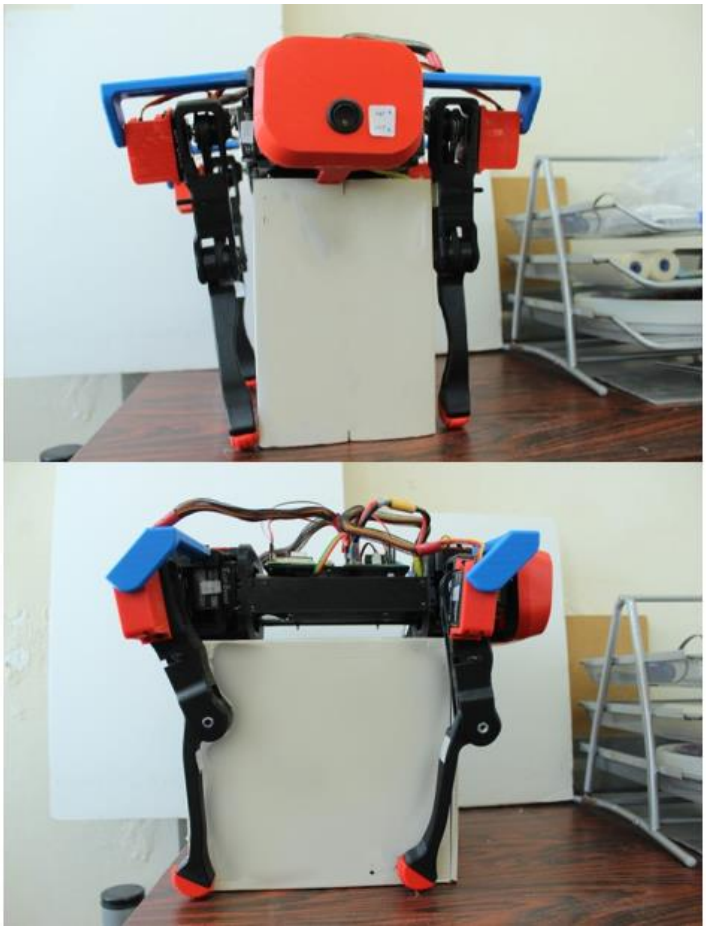


Figure 3-8 Photos of Spot Mini Mini

3.3 ELECTRONICS

A PDB (Power Distribution Board) was designed by Maurice Rahme and Adham Elarabawy, manufactured by the prototyping company JLCPCB [23]. By using a PDB in the build of the robot, what is accomplished is the connection of interfaces between the low-level control Teensy, the Raspberry Pi 4B, the IMU sensor, the robot's power that comes from a 5500mAh 7.4V LiPo battery. This circuit board is designed to interface the communication of the Raspberry Pi and the Teensy over Serial communication, so that the later can control the 12 servo motors and read data from analogue sensors that are connected to the board. In *Figure 3-9* can be seen a block diagram of the quadruped's hardware configuration.

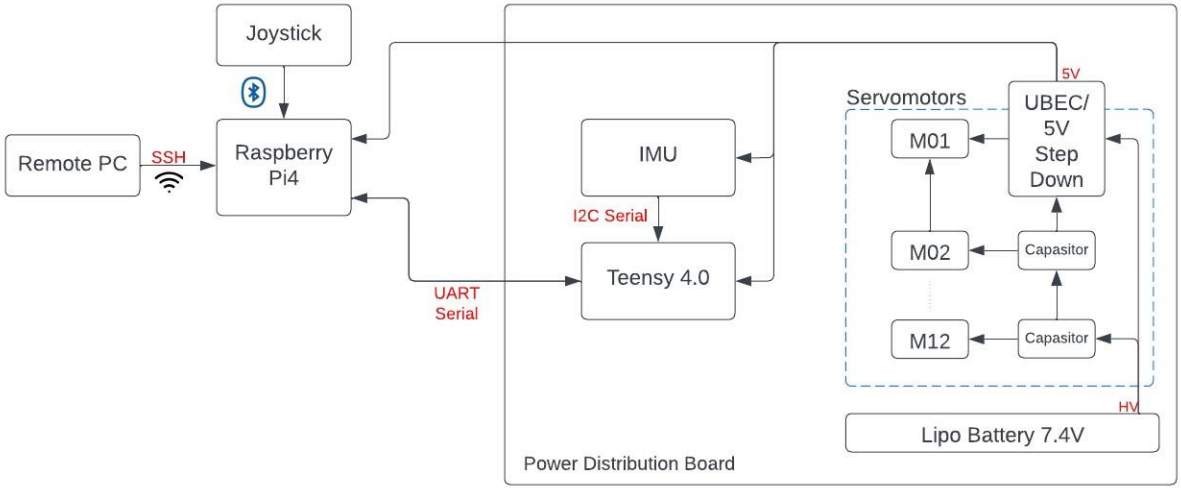


Figure 3-9 Spot Mini Mini's Block diagram of hardware configuration.

The PDB (as seen in *Figure 3-10*)of the project has a 1.5mm Track Width and could support current of up to 6A and an increase of 10 °C. The board, in order to enhance the heat dissipation, has been covered with copper grounding planes on both sides and for the same reason, the power lines have been designed to be parallel.

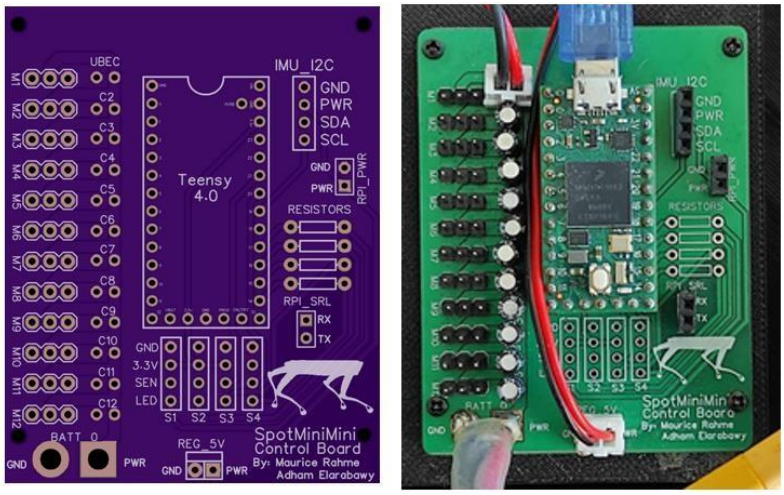


Figure 3-10 SpotMiniMini's PDB (simulation and photo)

Table 5 Servo Motor's Specifications [24]

Dead band	1µs 1520µs / 330hz
Motor	Brushless Motor
Operating Speed (6.0V) **Operating Speed (7.4V)**	0.13sec/60° 0.11sec/60°
Stall Torque (6.0V) **Stall Torque (7.4V)**	26.7kg.cm 33.7kgcm
Dimensions	40.5x20.5x40mm
Weight	65g
Connector Wire Length	JR 265 mm
Min and Max PWM	500 - 2500

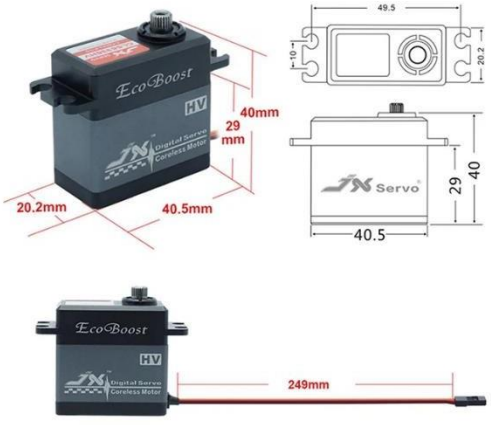


Figure 3-11 JX Ecoboost CLS6336HV (36KG)[26]

The servo motor (as seen in *Figure 3-11*)chosen for this build, JX Ecoboost CLS6336HV (36KG), is a high performance brushless servo motor designed with precision and efficiency in mind. The maximum stall torque of the servo, or the maximum torque the servo motor can exert when its output rotational speed is zero is 36 kg/cm [24]. The motor's specifications can be seen below in *Table 5*.

A servo motor draws substantial and sudden spikes of current especially during dynamic operations such as starting or changing direction. These sudden draws can cause transient voltage depressions in the power supply line. By connecting a shunt capacitor in series with a servo motor, it can supply the required energy to counterbalance this effect of energy dip, helping to ensure voltage stability and preserving the integrity of the power line. Of course, these voltage spikes can potentially prove damaging to sensitive electronics like motors. Moreover, capacitors can act as a filter by reducing the electrical noise generated by the motors. For the reasons mentioned and also considering that it was crucial to guarantee the longevity and protection of the twelve servo motors, being the most expensive component of the build, the capacitors chosen had high buffering capability, that of 470uF.

A Raspberry Pi 4 Model B (RPi 4B) functions as the brain of SpotMiniMini. The RPi 4B is an efficient and powerful minicomputer approximately equal to the size of a credit/debit card [25]. It has its own OS (Operating System) previously called Raspbian based on Linux. Raspberry Pi's specifications can be seen in *Table 6.*

Table 6 Raspberry Pi 4 Model B Specifications [27]

Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM	8GB LPDDR4-3200 SDRAM
Bluetooth	Bluetooth 5.0, BLE
Wi-Fi	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless
USB	2 USB 3.0 ports; 2 USB 2.0 ports
Ethernet	Gigabit Ethernet
HDMI	2 × micro-HDMI ports
Storage	MicroSD Card Slot
Power Supply	5.1V 3A USB type C Power
Dimensions	2 × micro-HDMI ports

For the low-level speed control of the servo motors, a Teensy 4.0 was chosen [26]. This microcontroller development board is popular among hobbyists and roboticists because of its small size, low cost, and powerful features. Moreover, Teensy has the ability to run at high clock speeds, and therefore, is suitable for real-time and performance-sensitive applications like the operation of SpotMiniMini's servo movements. Specifications seen in *Table 7.*

Table 7 Teensy 4.0 Specifications [28]

Ethernet	-none-
USB Host	2 SMT Pads
SDIO (4 bit data)	8 SMT Pads
PWM Pins	31
Analog Inputs	14
Serial Ports	7
Flash Memory	2 Mbyte
QSPI Memory	Program memory
Breadboard I/O	24
Bottom SMT Pads	16
SD Card Signals	0
Total I/O Pins	44

Finally, the only sensor used in SpotMiniMini is Adafruit's IMU BNO055. This sensor functions as a 9-DOF sensor that turns the sensor data from an accelerometer, gyroscope and magnetometer into stable three-axis orientation output [27]. Lastly, a UBEC is connected directly onto the PDB board functioning as a voltage regulator for the RPi at 5V. For the soldering the ATTEN 8502D solder station was used, with temperatures ranging from 200℃ to 480℃. The capacitors were the first components to be soldered onto the PDB board. In order to connect all of the other necessary components both pin headers and female headers where added onto the board, as well as a wired XT60 male connector that powers the board. In *Figure 3-12* can be seen the rocker switch of the robot, that is connected to the battery through a male T- plug connector.

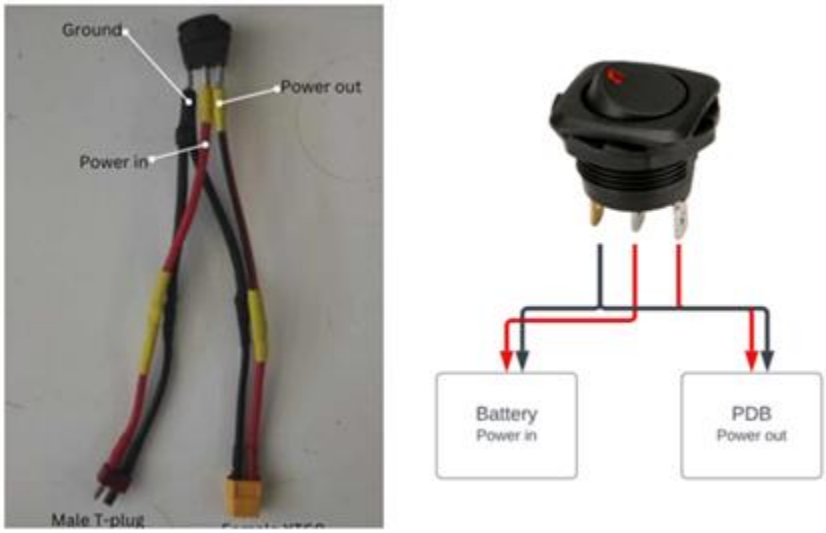


Figure 3-12 Rocker Switch

3.4 SOFTWARE

The quadruped is controlled with ROS (Robotic Operating System) integration. Its source code provides functionalities for calibrating the robot's servos, reading sensor data from the IMU, and commanding the robot's joints to move to desired positions. The robot can be controlled in real-time through a Bluetooth joystick, Logitech Gamepad F310 , using commands sent over ROS, and it can provide feedback about its state, including contact sensor readings and IMU data. In this current build, theonly sensor used is the IMU, for orientation sensing.

The software itself is developed within the ROS ecosystem and particularly in ROS Noetic that corresponds to the operating system of Ubuntu 20.04 LTS (Long Term Support). At the time this thesis was being developed, the original Ubuntu distribution used in the repository of SpotMiniMini was nearing the end of its standard support. It has since transitioned to 18.04 EOL (End of Life). To ensure the robot's software remains updated and accessible for future contributors in the open-source community, the entire project has been shifted to Ubuntu 20.04 LTS (Focal Fossa).

Ubuntu is a free platform and has a large and active community of users and developers, is based on the Debian Linux distribution and offers a very user friendly interface. However, for the build of the robot was required a “headless” configuration, meaning that there is no Graphical User Interface (GUI). The server image file for installing the OS Ubuntu 20.04 on Raspeberry Pi’s SD card was retrieved from Canonicals’ official website [28] and flashed to a 32 Gb microSD card with the help of the tool Raspeberry Pi Imager, formerly known as Raspbian [29]. This tool proved

very helpful since it allowed for the installation of Ubuntu, as well as the configuration of the RPi's Wi-Fi connection to the laboratory's network.

In order to remotely access the Raspberry Pi that is embedded to the robot, an SSH connection was established that connects to it from another machine. SSH or Secure Shell is a network communication protocol that enables two computers to communicate and share data, performing other secure network services over an insecure network [30]. For this goal, after configuring the Wi-Fi settings, SSH was activated by creating a specific file on the microSD card. Upon booting the Raspberry Pi, the system's IP address was identified by using a network scanning app, named Fing [31], and finally, establishing the SSH connection, allowing for the remote management of the Raspberry Pi.

ROS (Robot Operating System) is an open-source software development kit for robotics with a large footprint in the hobbyist robotics community. At its core, ROS provides a message-passing system, often called "middleware" or "plumbing". Communication is one of the first needs to arise when implementing a new robot application, or really any software system that will interact with hardware. There are several different ROS versions, and each one corresponds to specific operating systems. Originally, SpotMiniMini used the ROS Melodic Morenia [32]distribution that is supported by the Ubuntu 18.04 OS and was substituted by the ROS Noetic Ninjemys [33], corresponding version of Ubuntu 20.04. This update helped with resolving a number of python version conflicts that the original version faced.

The source code of the robot is uploaded from the RPi on the Teensy board using the PlatformIO ecosystem within the Visual Studio Code IDE. PlatformIO is an open-source

ecosystem for embedded development primarily used with microcontrollers and development boards that facilitates the uploading, compiling and management of the robot's code.

SpotMiniMini's source code employs different operating modes. The NOMINAL_PWM mode is used during the assembly of the robot and more specifically during the coupling of the servos with the servo horns. At this stage non-linearities in the assembly are expected and for that reason, the next step, calibrating the servos, was of great importance. The ROS calibration node is launched, and each servo is being given PWM commands to inspect its behavior. The goal is to find two known and measurable reference points (angles) for every servo and their corresponding PWM value. These values are recorded inside the code, as seen in this code snippet in *Figure 3-13* and the modes, as seen in *Figure 3-14* STRAIGHT_LEGS, LIEDOWN, PERPENDICULAR_LEGS allow for confirming that the calibration sequence for the servos is performed correctly. Finally, RUN mode is the default mode and SpotMiniMini is raising itself to a normal stance so that sensors and communications, including the joystick are ready.

```
// SERVOS: Pin, StandAngle, HomeAngle, Offset, LegType, JointType, min_pwm, max_pwm, min_pwm_angle, max_pwm_angle
//ROBOT POSE AT STARTUP

// Shoulders
double shoulder_liedown = 0.0;
FL_Shoulder.Initialize(2, 90 + shoulder_liedown, 90, -7.25, FL, Shoulder, 1185, 1860, 60, 120); // 0 | 90 mid - 0 out - 180 in
FR_Shoulder.Initialize(5, 90 - shoulder_liedown, 90, -5.5, FR, Shoulder, 1220, 1945, 60, 120);  // 3 | 90 mid - 180 out - 90 in
BL_Shoulder.Initialize(8, 90 + shoulder_liedown, 90, 5.75, BL, Shoulder, 1100, 1840, 60, 120);  // 6 | 90 mid - 0 out - 180 in
BR_Shoulder.Initialize(11, 90 - shoulder_liedown, 90, -4.0, BR, Shoulder, 1080, 1840, 60, 120); // 9 | 90 mid - 180 out - 90 in

//Elbows
double elbow_liedown = 25;
FL_Elbow.Initialize(3, elbow_liedown, 0, 0.0, FL, Elbow, 500, 2490, 180.0, 0.0);   // 1 | 0 in front & 180 in the back
FR_Elbow.Initialize(6, elbow_liedown, 0, 0.0, FR, Elbow, 500, 2500, 0.0, 180.0);   // 4 | 180 in front & 0 in the back
BL_Elbow.Initialize(9, elbow_liedown, 0, 0.0, BL, Elbow, 500, 2500, 180.0, 0.0);   // 7 | 0 in front & 180 in the back
BR_Elbow.Initialize(12, elbow_liedown, 0, 0.0, BR, Elbow, 500, 2500, 0.0, 180.0);  // 10| 180 in front & 0 in the back

//Wrists
double wrist_liedown = -160.0;
FL_Wrist.Initialize(4, wrist_liedown, 0, 0.0, FL, Wrist, 500, 1440, 0.0, -90.0);    // 2 | 0 straight - -90 forward
FR_Wrist.Initialize(7, wrist_liedown, 0, 0.0, FR, Wrist, 2150, 1310, 0.0, -90.0);   // 5 | 0 straight - -90 forward
BL_Wrist.Initialize(10, wrist_liedown, 0, 0.0, BL, Wrist, 500, 1480, 0.0, -90.0);   // 8 | 0 straight - -90 forward
BR_Wrist.Initialize(13, wrist_liedown, 0, 0.0, BR, Wrist, 2500, 1570, 0.0, -90.0);  // 11 | 0 straight - -90 forward
```

Figure 3-13 SpotMiniMini's calibration code

```
enum MODE {NOMINAL_PWM, STRAIGHT_LEGS, LIEDOWN, PERPENDICULAR_LEGS, RUN};
MODE spot_mode = RUN;
```

Figure 3-14 SpotMiniMini modes

The code uploaded to Teensy includes various libraries for controlling the servos, the IMU sensor and communication with ROS. In the low-level control of the robot, the main.cpp initializes the IMU and the servo motors, implements functions for updating them and sets the robot's stance according to the chosen operational mode from the 'MODE' enumeration. Moreover, it publishes data from the IMU to ROS topics that are used as the robot's control input.

In order to move the robot while in 'RUN' mode, a ROS launch file ('spot_real.launch') is called that launches various nodes for including a state machine node, a joystick node, a teleoperation node, a policy node, and a sensor interface node. The 'spot_real' node, calls the spot_real_interface.py file which subscribes to the ROS topics published by the Teensy. This file is responsible for the main operations of the robot's gait generation and control, discussed in the chapter *ROBOT GAIT*. Then, from the high level of the Raspberry Pi, data regarding the robot's servo motor angles are send through ROS again to the low-level Teensy and then translated to PWM commands for the servo motors to follow. This process is done recursively, with the IMU publishing data from the robot's with 50hz frequency. Bellow, in *Figure 3-15* can be seen a block diagram representation of the robot's code.

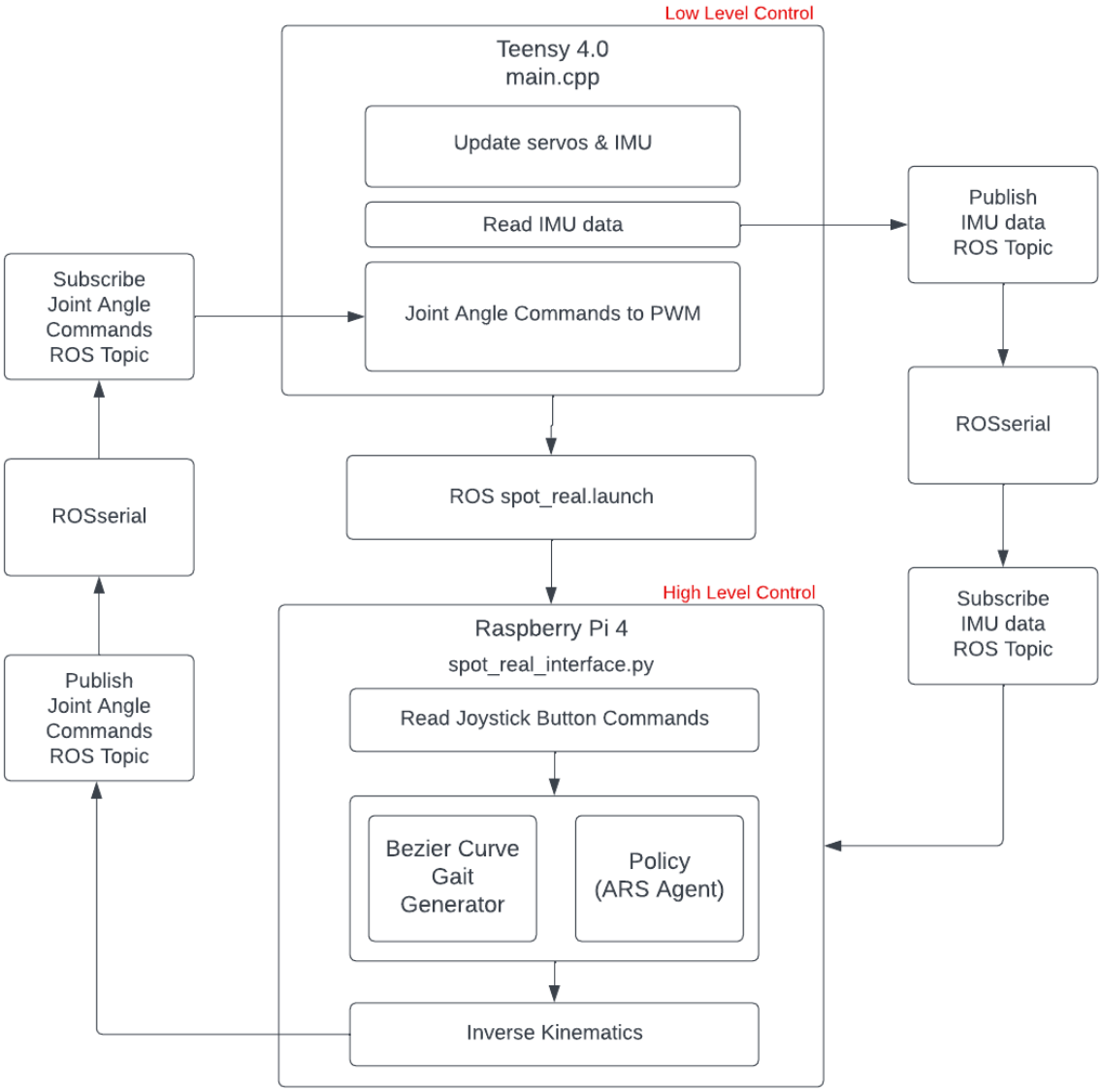


Figure 3-15 SpotMiniMini's flowchart

3.5 ROBOT GAIT

Spot Mini Mini uses a hybrid approach to legged locomotion that combines reinforcement learning (RL) and open-loop gaits. The method used, is named D2-GMBC [34] (Dynamics and Domain Randomized Gait Modulation with Bezier Curves) and is using 12-point Bezier curves in order to be able traverse unobserved and unmodeled rough terrain. The quadruped's gait modulation uses a learned policy in conjunction with an open-loop dynamics model for control. The model provides a baseline behavior while the policy uses sensor feedback to adapt the model and to improve control.

The locomotion task of the quadrupled is formulated as a Partially Observable Markov Decision Problem (POMDP). A POMDP is defined by a tuple (S, A, O, r(s,α), P(s'|α,s)), consisting of the state space S, the actions space A, the observation space O, a reward function r(s,a) and the transition probability P. The policy of the model, π as parameterized by the learnable parameters θ, is then tasked to find the best action for each state to maximize the expected reward. For each observation o∈O, the agent takes the action α ∈A based on the policy, formalized as α=πθ(o). Because the model is applied to the modulation task, the current state of the gait is included in this observation space.

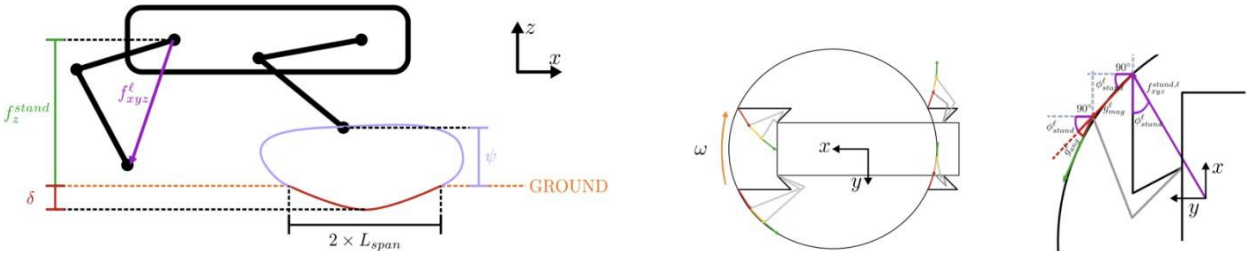


Figure 3-16 Schematic of foot placement based on Bezier Gait Generator[34]

The open-loop gait from D2-GMBC is used and computes a trajectory path for each foot using 12-point Bezier curves. Their model, based on the work of Hyun et al., [35] combines multiple 2D gaits into one 3D gait, which allows for flexible control of cyclical movements of the legs' extremities. As depicted in *Figure 3-16* ,the 2D curve of each foot contains two phases, the swing (purple), when the leg is lifted into the air, and the stance (red), when the foot remains in contact with the ground. For each foot l ∈ {FL, FR, BL, BR}3, these parameters are determined by the phase generator S'(t) ∈ [0,2), where the leg is in stance for S'(t)∈ [0,1), and swing for S'(t)∈ [1,2).

To adjust this open-loop gait a set of parameters is provided. The first two parameters are scalars that influence the trajectory of all legs. These are the clearance height ψ, which determines how high the leg lifts during the swing phase, and the virtual ground penetration δ, which is responsible for how high the respective shoulder of each leg is lifted during the stance phase. To adjust the position of the individual legs, a 3-dimensional residual position is defined by $\Delta f'_{xyz} \in R^3$, resulting in the total residual set Δf_{xyz}. The residual position adjusts the 3D foot position away from the current place in space as determined by the Bezier curve. The policy returns these parameters to adjust the underlying Bezier gait:

$$\Delta f_{xyz}, \psi, \delta = \pi_\theta (o_t) \quad (1)$$

The last input for the Bezier curve generator Γ, are the external motion commands $\zeta = [\rho, \bar{\omega}, L_{span}]$. From these inputs, L_{span} defines the stride length, ρ rotational angle relative to its momentum, and $\bar{\omega}$, the yaw velocity. During training the motion commands are fixed for L_{span} = 0.034 and ρ = 0, and an $\bar{\omega}$ is computed in real-time so that the direction of the robot is always

straight. The curve generator determines the foot positions by:

$$f^{tg}_{xyz} = \Gamma(S(t), \zeta, \psi, \delta) \quad (2)$$

After adding the residuals, the final position of all the foot endpoints is:

$$f_{xyz} = f^{tg}_{xyz} + \Delta f_{xyz} \quad (3)$$

After the foot positions have been determined, Inverse Kinematics (IK) is used to calculate the joint angles for each of the three actuators of a leg. The foot position from the gait generator Γ, is the position with respect to the shoulder corresponding to each leg. The IK then determines what angle each actuator of a leg should have to achieve this position, under the range constraints of the actuators.

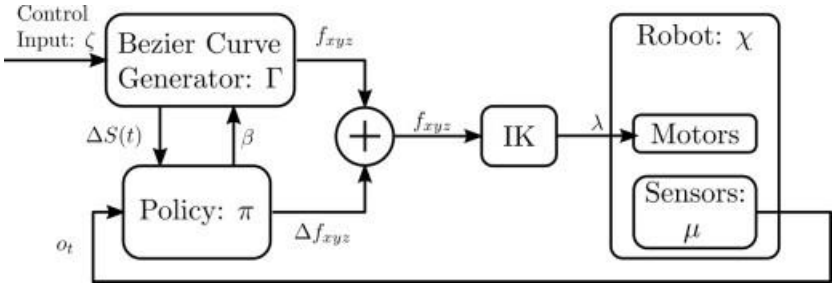


Figure 3-17 D²-GMBC System Diagram [34]

In order to construct the reward function for training of the quadruped, access to the full robot state is needed. Since full state information cannot be retrieved from the real robot, the policy is trained in simulation. In the pursuit of enhancing the sim-to-real efficacy of gait modulation policies, both Dynamics Randomization and Domain Randomization are employed via ARS (Augmented Random Search). Dynamics Randomization involves around the modification of key dynamic parameters within the simulated robotic model,such as the mass distribution across the

robot's legs and the friction between its feet and the ground. Domain Randomization refers to providing the simulated environment with diverse terrain characteristics by randomizing the terrain's geometry. Derived from [34] in *Figure 3-18* can be seen the gait modulation algorithms with (Algorithm 1) and without the ARS regarding the simulation (Algorithm 2).

Algorithm 1 Gait Modulation with Bezier Curves (GMBC)

Given: Policy π with parameter θ, (Bezier) Curve Generator Γ, External motion command ζ, robot sensor observations o_t, Leg phase $S(t)$

1: obtain gait modulation from π with learned parameter θ
2: $\Delta f_{xyz}, \beta = \pi(o_t, \theta)$
3: calculate (Bezier) gait foot placement
4: $f_{xyz} = \Gamma(S(t), \zeta, \beta)$
5: **return** $f_{xyz} + \Delta f_{xyz}$ to robot for IK joint control

Algorithm 2 RL Simulation training for D^2-Randomized GMBC (D^2-GMBC) using Augmented Random Search [5]

Initialize: policy parameters θ_0, D^2 distribution $\mathbb{P}$, reward function $\mathcal{R}$, GMBC (Algorithm 1), iteration number $k = 0$, construct ARS.

1: **while** training not converged **do**
2: $\sigma \sim \mathbb{P}$ sample D^2 parameters
3: ARS step of (1) with D^2-randomization + GMBC
4: $\theta_{k+1} \leftarrow \text{ARS}(\pi, \theta_k, \mathcal{R}, \sigma, k)$
5: $k \leftarrow k + 1$
6: **end while**
7: **return** θ_k

Figure 3-18 Algorithm 1 & Algorithm 2

4.0 SIMULATION

Part of the OpenQuadruped GitHub repository is the development of a simulator for SpotMiniMini, using Pybullet. The simulator serves as a safe environment for testing without risking damage to the physical robot and can also be used for reinforcement learning tasks. The training process involves using a Proportional controller for yaw correction and randomizing terrain, link masses, and foot frictions for policy robustness. The action space consists of 14 dimensions, and the results show successful navigation of terrain after training. The trained agent can also adapt to unseen commands, making integration with high-level systems straightforward. The setup requires dependencies such as ROS, Gazebo, Pytorch, and others, and joystick control is optional for teleoperation.

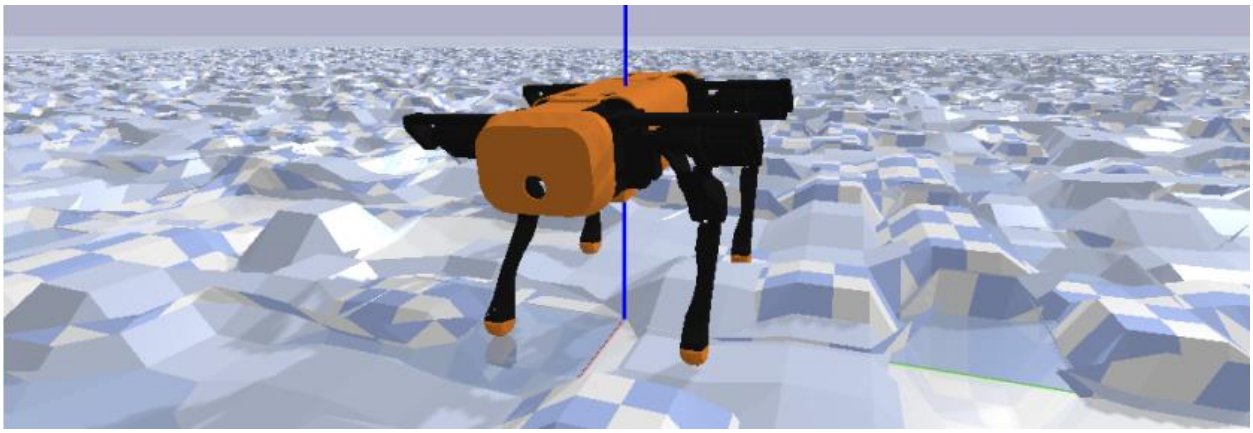


Figure 4-1 SpotMiniMini in simulated environment, domain randomized terrain [34]

5.0 SOFTWARE TOOLS

In the development of the quadruped robot described in this thesis, a range of software tools played integral roles in its creation and functionality. The core of the robot's control system was built upon a combination of Python and C++, with the robot's code sourced from the corresponding GitHub repository. Custom Serialized ROS messages using ROSSerial for ROS Noetic running on the Ubuntu 20.04 operating system, enable secure and efficient communication between the Teensy microcontroller and the Raspberry Pi. An SSH connection from an external PC to the Raspberry Pi was established, enabling remote access and control of the robot's operations. The Teensy's firmware was organized into multiple header files, addressing specific functionalities such as Servo Motor Operations, Inverse Kinematics and IMU data processing. To streamline the compilation process and enhance code management, PlatformIO was used.

In certain instances, ChatGPT, an AI language model, was utilized to gain deeper insights into complex concepts, providing valuable assistance in the development and understanding of critical aspects of the project. These software tools collectively contributed to the successful development and documentation of the quadruped robot, as detailed in this thesis.

6.0 CONCLUSIONS & FUTURE STEPS

Over the past decade, significant strides have been made in the realm of quadruped robots, with the Open-Source community playing a pivotal role in driving progress. In the scope of this thesis, the quadruped SpotMiniMini was simulated, fabricated and programmed based on the Open Source project of Maurice Rahme. Towards its realization the first step was gathering the parts of the robot, followed by the 3d printing of its legs and main body parts. After the soldering and connecting of the electronics was completed, the assembly ensued. A calibration sequence was performed to minimize the inconsistencies that result because of the coupling of the servo motors and during the assembly process. SpotMiniMini's software was updated from the original GitHub repository, a step necessary for enabling the implementation of the code of the robot and its control.

Although in the current build of SpotMiniMini the only analogue sensor used is an IMU, the board provides an interface for four more sensors that could for example be hall effect sensors, connected to the feet of each leg. The gait modulation used in the current build does not require them, but it could prove useful in the future to explore other control approaches that incorporates them. Another important improvement would be a redesign of the PDB in which, every capacitor is connected in parallel with a resistance so that greater measures of safety are achieved, ensuring sensitive and expensive hardware parts like the robot's servo motors. Another step for the robotic platform of SpotMiniMini could be a systematic integration of a LIDAR sensor for environmental mapping and obstacle detection. With enhanced perceptual capabilities, the quadruped could perform in unknown and complex environments with a higher degree of autonomy.

The development of the SpotMiniMini was suddenly halted due to its creator's non-disclosure agreement with their employer and therefore, the immigration of the project to the newer Ubuntu 20.04 version that supports ROS Noetic was an incredibly necessary step not only in the scope of this thesis but also regarding the continuation and further progress of the SpotMiniMini project by the open-source community. The GitHub fork repository that now has the updated code and the necessary bug fixes for the robot needs enhanced code readability and a more comprehensible documentation that will make the project easier to follow in the future.

7.0 REFERENCES

[1] "OpenQuadruped/spot_mini_mini: Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion." Accessed: Oct. 04, 2023. [Online]. Available: https://github.com/OpenQuadruped/spot_mini_mini

[2] "Spot® - The Agile Mobile Robot | Boston Dynamics." Accessed: Nov. 23, 2022. [Online]. Available: https://www.bostondynamics.com/products/spot

[3] E. M. Wetzel, J. Liu, T. Leathem, and A. Sattineni, "The Use of Boston Dynamics SPOT in Support of LiDAR Scanning on Active Construction Sites," Jul. 2022. doi: 10.22260/ISARC2022/0014.

[4] V. Vanniyakulasingam, "Autonomous mission configuration on Spot from Boston Dynamics," Apr. 2023.

[5] "Boston Dynamics' Robot Dog on Security Patrol at Boston Marathon." Accessed: Oct. 02, 2023. [Online]. Available: https://www.iotworldtoday.com/robotics/boston-dynamicsrobot-dog-on-security-patrol-at-2023-boston-marathon

[6] "Home - SpotMicroAI." Accessed: Oct. 04, 2023. [Online]. Available: https://spotmicroai.readthedocs.io/en/latest/

[7] N. Piolanti, S. Polloni, E. Bonicoli, M. Giuntoli, M. Scaglione, and P. Indelli, "Giovanni Alfonso Borelli: The Precursor of Medial Pivot Concept in Knee Biomechanics," Joints, vol. 06, no. 03, Sep. 2018, doi: 10.1055/s-0038-1675164.

[8] "All You Need To Know About Boston Dynamics' Spot," https://analyticsindiamag.com/all-you-need-to-know-about-boston-dynamics-spot/.

[9] "The French army is testing Boston Dynamics' robot dog Spot in combat scenarios," https://www.theverge.com/2021/4/7/22371590/boston-dynamics-spot-robot-military-exercises-french-army.

[10] M. Hutter, "StarlETH & Co.," ETH Zürich, vol. PhD Thesis, no. 21073, 2013, doi: 10.3929/ETHZ-A-009915229.

[11] M. Hutter, C. Gehring, M. Bloesch, M. Hoepflinger, and R. Siegwart, "Walking and Running with StarlETH," undefined, 2013, doi: 10.3929/ETHZ-A-010022793.

[12] "ANYmal – Robotic Systems Lab | ETH Zurich." Accessed: Nov. 08, 2021. [Online]. Available: https://rsl.ethz.ch/robots-media/anymal.html

[13] M. Hutter et al., "ANYmal - a highly mobile and dynamic quadrupedal robot," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Oct. 2016, pp. 38–44. doi: 10.1109/IROS.2016.7758092.

[14] M. Hutter et al., "ANYmal - toward legged robots for harsh environments," Advanced Robotics, vol. 31, no. 17, pp. 918–931, Sep. 2017, doi: 10.1080/01691864.2017.1378591.

[15] B. Katz, J. Di Carlo, and S. Kim, "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control," in 2019 International Conference on Robotics and Automation (ICRA), IEEE, May 2019, pp. 6295–6301. doi: 10.1109/ICRA.2019.8793865.

[16] S. Kitano, S. Hirose, A. Horigome, and G. Endo, "TITAN-XIII: sprawling-type quadruped robot with ability of fast and energy-efficient walking," ROBOMECH Journal, vol. 3, no. 1, Dec. 2016, doi: 10.1186/s40648-016-0047-1.

[17] Z. He et al., “Design Principles for a Family of Direct-Drive Legged Robots,” IEEE Robot Autom Lett, vol. 1, no. 2, pp. 2161–2168, 2016.

[18] J. Kim, T. Kang, D. Song, and S.-J. Yi, “Design and Control of a Open-Source, Low Cost, 3D Printed Dynamic Quadruped Robot,” Applied Sciences, vol. 11, no. 9, p. 3762, Apr. 2021, doi: 10.3390/app11093762.

[19] N. Kau, A. Schultz, N. Ferrante, and P. Slade, “Stanford Doggo: An Open-Source, QuasiDirect-Drive Quadruped,” in 2019 International Conference on Robotics and Automation (ICRA), IEEE, May 2019, pp. 6309–6315. doi: 10.1109/ICRA.2019.8794436.

[20] N. Kau, “Stanford Pupper: A Low-Cost Agile Quadruped Robot for Benchmarking and Education,” Oct. 2021.

[21] M. A. and B. V. and K. M. Şen, “Inverse Kinematic Analysis Of A Quadruped Robot,” International Journal of Scientific & Technology Research, vol. 6, Oct. 2017.

[22] “Maurice Rahme – Spot Mini Mini.” Accessed: Oct. 10, 2023. [Online]. Available: https://moribots.github.io/project/spot-mini-mini

[23] “PCB Prototype & PCB Fabrication Manufacturer - JLCPCB.” Accessed: Oct. 12, 2023. [Online]. Available: https://jlcpcb.com/

[24] “JX Ecoboost CLS6336HV 36KG Large Torque 180Degree CNC Digital Coreles – QWinOut.”Accessed: Oct. 13, 2023. [Online]. Available: https://qwinout.com/products/jxecoboost-cls6336hv-36kg-large-torque-180degree-cnc-digital-coreless-servo-for-rcmodels-helicopter-cars-truck-part

[25] “(PDF) A Review Paper on Raspberry Pi and its Applications.” Accessed: Oct. 13, 2023.

[Online].Available:https://www.researchgate.net/publication/348993428_A_Review_Paper_on_Raspberry_Pi_and_its_Applications

[26] "Teensy® 4.0." Accessed: Oct. 27, 2023. [Online]. Available: https://www.pjrc.com/store/teensy40.html

[27] "Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 : ID 2472 : $34.95 : Adafruit Industries, Unique & fun DIY electronics and kits." Accessed: Oct. 13, 2023. [Online]. Available: https://www.adafruit.com/product/2472

[28] "Enterprise Open Source and Linux | Ubuntu." Accessed: Oct. 26, 2023. [Online]. Available: https://ubuntu.com/

[29] "Raspberry Pi OS – Raspberry Pi." Accessed: Oct. 26, 2023. [Online]. Available: https://www.raspberrypi.com/software/

[30] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Connection Protocol," Jan. 2006, doi: 10.17487/RFC4254.

[31] "Fing App | Network toolkit and scanner | Fing." Accessed: Nov. 06, 2023. [Online]. Available: https://www.fing.com/products/fing-app

[32] "melodic - ROS Wiki." Accessed: Oct. 27, 2023. [Online]. Available: https://wiki.ros.org/melodic

[33] "noetic - ROS Wiki." Accessed: Jan. 10, 2024. [Online]. Available: https://wiki.ros.org/noetic

[34] M. Rahme, I. Abraham, M. L. Elwin, and T. D. Murphey, "Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion,"

Oct. 2020, Accessed: Nov. 09, 2023. [Online]. Available: https://arxiv.org/abs/2010.12070v1

[35] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah," http://dx.doi.org/10.1177/0278364914532150, vol. 33, no. 11, pp. 1417–1445, Aug. 2014, doi: 10.1177/0278364914532150.

[36] N. P. Papadopoulou, "Quadrupedal Bio-inspired Robots: Development & Characteristics", 2022.

[37] Photo sourced from an unofficial Boston Dynamics Reddit post: https://www.reddit.com/r/BostonDynamics/comments/fa67ts/boston_dynamics_robotics_timeline/

APPENDIX A

The table below contains a list of 3D printed components necessary for the build of SpotMiniMini.

.stl file name	Repetitions	Weight (g)	Hours (hr)
Body			
Adapter_Plate	1	32	5
Back_Inner_Shoulder	1	70	2.3
Back_Outer_Shoulder	1	32	2.45
Chassis_Left_Side	1	17	1.28
Chassis_Right_Side	1	17	1.28
Electronics_Plate	1	47	3.26
Front_Inner_Shoulder	1	70	2.3
Front_Outer_Shoulder	1	31	2.23
	Repetitions	Weight (g)	Hours
Calibrators-Extras			
Battery Mount	1	30	3
Hip_Calibrator	1	20	

L_Upper_Leg_Calibrator	1	10	
Lower_Leg_Calibrator	1	15	
	Repetitions	Weight (g)	Hours
Covers			
Bottom	1	110	9.9
Front	1	46	4.31
Rear	1	46	4.3
Top	1	90	8
	Rep	Weight (g)	Hours
Left & Right Legs			
Bridge	2	40	3
Left_Hip_Joint	2	62	3.3
Lower_Leg	2	62	5.2
Upper_Leg	2	90	12
Bridge	2	40	3
Right_Hip_Joint	2	62	3.3
Lower_Leg	2	55	5.2

Upper_Leg	2	90	12
Foot	4	21	2.14
Servo_Cover	4	70	3.43
Upper_Pulley	4	29	3.49
Idler_Mount	4	5	0.45
Total:		1309g	≈105 hrs

APPENDIX B

A basic guide for operating Spot Mini Mini is provided, regarding the basic steps:

Remotely connecting to the robot:

- Connect Spot Mini Mini to its battery.
- Turn the power switch on. Automatically, the robot will connect to the Wi-Fi network.
- From a remote PC connect to the same Wi-Fi network as the robot and open Visual Studio Code.
- Type **Ctrl+Shist+P** to access VS code's Command Palette. Choose the option "**+ Add new SSH host**".
- Type "ssh spotminimini@192.168.43.254". Press Enter to connect to Spot Mini Mini's remote server.
- Password: "spot"

Entering Spot Mini Mini's workspace:

- Open a terminal by typing **Ctrl+ ~**.
- The robot's workspace is named **robot_ws**. To access it type: **cd ~/robot_ws.**
- Essential for properly configuring your ROS workspace environment, enabling package execution, facilitating communication between ROS nodes, and accessing custom messages and services inside the ROS workspace: Type: **source devel/setup.bash**

- In the same directory build the workspace: **catkin_make**
- Navigate to the firmware directory where the platformio.ini file is located, for Platformio to upload the source code from the Raspberry Pi to the Teensy microcontroller:

  cd ~/robot_ws/src/spot_mini_mini/spot_real/Control/Teensy/SpotMiniMini/

  platformio run -t upload

Assembly and Motor Calibration:

- To access the main.cpp file that runs on the Teensy on VS code:

  cd ~/robot_ws/src/spot_mini_mini/spot_real/Control/Teensy/SpotMiniMini/src

  code main.cpp
- During assembly, the servo motors are powered and the NOMINAL_PWM mode is selected inside the source file's line: **MODE spot_mode = NOMINAL_PWM**

  Assembly is performed in a way that allows the robot's shoulder joint, elbow joint and wrist joint to be as close to 90°, 90° and 180° respectively, as possible. At this stage non-linearities can occur because of the imperfections in the coupling between the servo shafts and servo horns. To resolve this problem, in the same NOMINAL_PWM mode, a calibration sequence is going to be performed.
- In a terminal type: **roslaunch mini_ros spot_calibrate.launch**
- Open another terminal and do: **rosservice call /servo_calibrator** and to auto-complete the format, press **<TAB> <TAB>.**

  Then, for each joint 0-11, give a few different PWM commands ranging from 500 to 2500 PWM to inspect how every servo moves.

- For each servo, inspect the PWM value that moves it to two different and measurable positions. These reference points are going to be carefully decided based on real world evaluation (ex. 0 and 90°). In the Initialize() method of the main.cpp file, record the according PWM value and angle following the order: **[PWM0, PWM1, ANG0, ANG1].**
- The modes **STRAIGHT_LEGS, LIEDOWN and PERPENDICULAR_LEGS** are used in order to verify that the calibration of the robot is done correctly.

Moving Spot Mini Mini:

- Navigate to the main.cpp directory:

  cd ~/robot_ws/src/spot_mini_mini/spot_real/Control/Teensy/SpotMiniMini/src

  code main.cpp

- Change the robot's mode to 'RUN': **MODE spot_mode = RUN**
- Upload the code to the Teensy**:**

  cd ..

  platformio run -t upload

- Perform the standard ROS actions inside the workspace:

  cd ~/robot_ws/

  source devel/setup.bash

  catkin_make

  Finally by typing **roslaunch mini_ros spot_real.launch,** the serial connection between the Raspberry Pi and the Teensy will be established and Spot Mini Mini can be given commands from the joystick.