

IMPERIAL

**ENGINEERING A COST-EFFECTIVE,
AUTONOMOUS QUADRUPED**

PAVING NEW PATHS IN AUTOMATED DISASTER RESPONSE

Author

G. FENOCCHI
CID: 01860614

Industrial Supervisor
J. McNAMARA

Internal Supervisor
MRS E. PEREA

Second Marker
DR E. STOTT

A Thesis submitted in fulfilment of requirements for the degree of
Master of Engineering in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering
Imperial College London
2024

Abstract

The development of a cost-effective, autonomous quadruped robot for navigating disaster zones, assisting in rescue operations, and entering environments that are unsafe for humans aims to survey and collect data. The objective was to engineer a system that is reliable, efficient, and capable of operating under challenging conditions, an important aspect for support in natural disasters.

The design involved 3D CAD, with an emphasis on integrating additive manufactured components and commercial-grade motors to balance cost, durability, and performance effectively. A key feature of the proposed robot is its modular design, which is essential for enabling quick onsite repairs through modular field-replaceable units (FRUs). Additionally, the robot integrates WatsonX from IBM to collect data from potential casualties, enhancing its ability to assess and respond to their needs.

This document outlines the research, design, and testing phases, including the challenges anticipated, the solutions implemented, and the final milestones achieved. The development and testing ensured the robot's effectiveness and reliability in various scenarios, such as rescue operations and hazardous environment surveys.

Declaration of Originality

I affirm that I have submitted, or will submit, an electronic copy of my final year project report to the provided EEE link.

I affirm that I have submitted, or will submit, an identical electronic copy of my final year project to the provided Blackboard module for Plagiarism checking.

I affirm that I have provided explicit references for all the material in my Final Report that is not authored by me, but is represented as my own work.

I have minimally used ChatGPT, as an aid in the preparation of my report, principally in assisting to prepare tables in LaTeX and in the wording of a few complex sentences. All technical content and references comes from my original text.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgments

I would like to extend my deepest gratitude to those who supported me throughout the development of this project. Firstly, I am immensely grateful to my industrial supervisor, Prof. J. McNamara (IBM), for his guidance, expertise, and encouragement. This project would not have been possible without his support. His insights and feedback were greatly appreciated. I also wish to express my sincere thanks to my internal supervisor, Mrs. E. Perea, for her continuous support, mentorship, and constructive critiques. Her academic guidance and support significantly enriched this work. Additionally, I would like to acknowledge Dr. E. Stott, the second marker, for his time and effort in reviewing this thesis. Finally, I appreciate the support and understanding of my family and friends, whose encouragement kept me motivated throughout this journey. Thank you all for your contributions to the success of this project.

Contents

Abstract	i
Declaration of Originality	iii
Copyright Declaration	v
Acknowledgments	vii
List of Acronyms	xv
List of Figures	xvii
1 Introduction	1
1.1 Problem Statement and Specification	1
1.2 Key Questions and Problems Addressed	2
1.3 Structure of the Report	2
2 Background	3
2.1 Introduction to Disaster Response Robotics	3
2.2 Advancements in Quadruped Robotics	5
2.3 Economic and Humanitarian Impact of Natural Disasters	6
2.4 Challenges in Disaster Response	8
2.5 Innovative Solutions and Future Directions	8
2.6 Review of Existing Solutions	9
2.6.1 Boston Dynamics' BigDog	9
2.6.2 Boston Dynamic's Spot	10
2.6.3 Unitree's GO2	11
2.6.4 AT&T	12
2.6.5 Ghost Robotics	13
2.6.6 Deep Robotics	13
2.6.7 Other Solutions	14
2.6.8 Proposed Solution: REX	14

3 Requirements Capture	17
3.1 Addressing Current Limitations	17
3.1.1 Proposed Advancements	17
3.2 Requirements	18
3.2.1 Design and Build	19
3.2.2 Autonomous Navigation	20
3.2.3 Communication Interface	21
3.2.4 Facial Recognition Technology	21
3.2.5 Web Application Development	21
4 Initial System Design & Analysis	23
4.1 High-Level Design Overview	24
4.1.1 Introduction	24
4.1.2 System Architecture	24
4.2 Chassis Design	25
4.2.1 Materials and Construction	25
4.2.2 Motor Assembly	28
4.3 Leg Design	29
4.3.1 Exploration of Leg Mechanisms	29
4.3.2 Proposed Leg Mechanism	32
4.3.3 Motor Configuration	34
4.3.4 PWM Servo Control	34
4.3.5 Design Choices	34
4.4 Front-End Sensor Integration	36
4.4.1 9-axis IMU (Inertial Measurement Unit)	36
4.4.2 Environmental Temperature Sensor	36
4.4.3 OLED Display	36
4.4.4 Depth Camera (OAK-D LITE)	37
4.4.5 Dual Microphone Array	37
4.4.6 Mounting and Protection	37
4.5 Rear-Section	38
4.5.1 Power Control Switches	38
4.5.2 USB-C Quick Charge Module	38
4.5.3 Safety Switch	39

4.5.4	Mounting and Accessibility	39
4.5.5	Magnetic Rear Cover	39
4.6	Side and Top Sections	39
4.7	Battery Pack Design	41
4.7.1	Rationale of Li-Ion Cells	41
4.7.2	Power Calculations	41
4.7.3	Charging Mechanism	42
4.7.4	Design Considerations	43
4.8	Considerations for Productionisation	44
4.9	Electrical Design	45
4.9.1	Power Distribution Overview	45
4.9.2	Battery Charging Circuit	45
4.9.3	Motor Power Circuit	47
4.10	Software Design	50
4.10.1	Design Principles	51
4.10.2	Software Architecture	51
4.10.3	Navigation Subsystem	51
4.10.4	Vision Subsystem	52
4.10.5	Control Subsystem	52
4.10.6	Communication Subsystem	52
5	Hardware Implementation	55
5.1	Introduction	55
5.2	Chassis and Main Assemblies	56
5.2.1	Chassis Construction	56
5.2.2	Motor Mounting	58
5.2.3	Leg Mechanism Construction	60
5.2.4	Front-End Sensor Assembly	62
5.2.5	Rear Panel Assembly	70
5.3	Power and Drive System Implementation	72
5.3.1	Battery Pack Assembly and Charging	72
5.3.2	Power Delivery and Motor Control	74
5.4	Bill of Materials (BOM) and Expenditure	80

6 Software Implementation	81
6.1 Introduction	82
6.2 Calculation of Kinematics	82
6.2.1 Forward Kinematics (FK)	82
6.2.2 Inverse Kinematics (IK)	82
6.3 Software Simulation with PyBullet	86
6.3.1 Background on PyBullet	86
6.3.2 Gait Cycle Generation and Walking Simulation	87
6.3.3 Geometry Transformations	88
6.3.4 URDF (Unified Robot Description Format)	90
6.3.5 Conclusion	93
6.4 Path Planning Experimentation	93
6.5 Software Setup and Configuration	94
6.5.1 Initial Setup	95
6.5.2 LIDAR Setup	96
6.5.3 Cartographer Setup	97
6.5.4 Oak-D Lite Depth Camera Setup	98
6.5.5 Nav2 Setup	99
6.6 Control System for REX	100
6.6.1 Servo Calibration	100
6.6.2 Calibration Program	101
6.6.3 Controller Service	101
6.6.4 Testing and Debugging	102
6.7 Coordinate Acquisition from Depth Camera	102
6.7.1 Detection and Depth Calculation	102
6.7.2 Angle and Position Calculation	104
6.7.3 Coordinate Transformation	104
6.7.4 Final Coordinates in the Robot Frame	105
6.7.5 Tracking and Data Publishing	105
6.8 Autonomous Navigation Implementation	106
6.8.1 Initialisation and Parameter Loading	106
6.8.2 Map and Odometry Processing	106
6.8.3 Frontier Identification and Grouping	107

6.8.4	Path Planning Using A* Algorithm	107
6.8.5	Path Smoothing with B-Splines	107
6.8.6	Pure Pursuit Path Tracking	107
6.8.7	Local Obstacle Avoidance	107
6.8.8	Exploration Loop	107
6.8.9	Wavefront Exploration Simulation	108
6.9	WatsonX Assistant	109
6.9.1	Wake Word Detection	110
6.9.2	Speech-to-Text Conversion	110
6.9.3	Interaction with Watson Assistant	110
6.9.4	Text-to-Speech Conversion	110
6.9.5	System Integration	111
6.10	Website Dashboard	111
6.10.1	Overview	111
6.10.2	Key Components	113
6.10.3	Functionality and User Experience	114
6.10.4	Conclusion	114
6.11	Final Overview	114
7	Project Planning & Evaluation Strategies	117
7.1	Project Timeline and Tasks	117
7.1.1	Key Milestones	120
7.2	Evaluation of Project Success	120
7.3	Safety Consideration	120
7.3.1	Lab Access and Safety Protocols	120
7.3.2	Risk Assessment	120
7.4	Ethical Considerations	121
7.4.1	Safety and Reliability	121
7.4.2	Privacy and Data Protection	121
7.4.3	Dual-Use Considerations	121
7.4.4	Accountability in Autonomous Actions	121
7.4.5	Socio-Economic Impacts	122
7.4.6	Human-Robot Interaction	122
7.5	Environmental Considerations	122

Conclusions	127
7.6 Conclusions and Evaluations	127
7.7 Performance Evaluation	128
7.7.1 Outdoor Navigation	128
7.7.2 Human Interaction	128
7.7.3 Overall Performance	128
7.8 Future Work	129
7.9 Design Files	130
A Appendix	131
Bibliography	133

List of Acronyms

AI Artificial Intelligence

AM Additive Manufacturing

AMCL Adaptive Monte Carlo Localisation

AR Augmented Reality

ASA Acrylonitrile Styrene Acrylate

BMS Battery Management System

CAD Computer-Aided Design

COTS Commercial Off-The-Shelf

CV Computer Vision

EMI Electromagnetic Interference

FEA Finite Element Analysis

FRU Field-Replaceable Unit

GPS Global Positioning System

HMI Human-Machine Interface

IBM International Business Machines Corporation

IMU Inertial Measurement Unit

LIDAR Light Detection and Ranging

ML Machine Learning

NLU Natural Language Understanding

PM Project Management

PLA Polylactic Acid

PLC Programmable Logic Controller

ROS Robot Operating System

RPAS Remotely Piloted Aircraft Systems

SLAM Simultaneous Localization and Mapping

UAS Unmanned Aircraft Systems

UGV Unmanned Ground Vehicles

UAV Unmanned Aerial Vehicle

UIM User Interface Management

UW Underwater

VR Virtual Reality

List of Figures

2.1	PackBot, tracked robotic vehicle [Image: i-robot]	4
2.2	PackBots exploring Fukushima Plant in 2011 [Source: CNN]	4
2.3	Collossus, Paris 2019. [Photo: Aurelien Meunier/Getty Images]	5
2.4	Snakebot [Photo: Carnegie Mellon, 2021]	5
2.5	Ghost Robotics' Vison 60 Quaruped robots, aiding Japan's earthquake relief operation, January 2024 [6]	6
2.6	Collapsed residential buildings in Hatay, Turkey. [Photo: David Lombeida/Bloomberg]	7
2.7	Sea of Japan Earthquake, 01 Jan 2024, [Photo: Haji Mada]	7
2.8	Boston Dynamics' BigDog	10
2.9	Boston Dynamics's Spot	10
2.10	Unitree's GO2	11
2.11	AT&T's Allen Beadel with Payload [Photo: AT&T]	12
2.12	Ghost Vision 60 [Photo: Ghost Robotics]	13
2.13	EX30 in a simulated rescue mission [Photo: Deep Robotics]	13
2.14	REX - a Cost-Effective & Autonomous Quadruped Robot	15
2.15	Size Comparison of Current Quadrupeds and REX	16
2.16	REX and Spot, side-by-side	16
3.1	3D Printing of REX's Legs	20
4.1	Software and Hardware Subsystems	24
4.2	Chassis Assembly - CAD Model	25
4.3	Exaggerated Displacement of a Carbon-Fibre Tube under a 300N Load	26
4.4	Chassis Cross-Section - CAD Model	27
4.5	FEA of Chassis Brace Under 30N Load	28
4.6	Side by side comparison of CAD models.	28
4.7	Chassis & Motor Assembly - CAD Model	29
4.8	Belt-Mechanism for the MIT Cheetah Leg	30
4.9	Patent of Boston Dynamics Leg Mechanism	30
4.10	Watt Six-Bar Leg Mechanism from Rahman et al. [27]	31

4.11	Full Leg Assembly - CAD Model	32
4.12	Diagram of Leg Linkages	33
4.13	Design inclusion of ball bearings	35
4.14	Front Sensor Arrangement - CAD Model	36
4.15	Front Face Cover - CAD Model	37
4.16	Rear-Section - CAD Model	38
4.17	Rear-Section with and without cover	39
4.18	Quadruped with Top Panels and LIDAR Sensor	40
4.19	Side Panel with Speaker	40
4.20	Comparison of Energy Density in Battery Cells [30]	41
4.21	Battery Pack Design with Integrated Battery Management System	43
4.22	Custom, 3S/2P, Battery Pack with Integrated BMS: Nominal voltage: 11.1 V Nominal capacity: 7000 mAh	43
4.23	Battery Pack Centre Placement in Main Assembly	44
4.24	Electrical Subsystem	45
4.25	Cycle Life of Different Charge Rates in Lithium-Ion Cells [34]	46
4.26	Schematic Diagram for the Charging Circuit	47
4.27	Synchronous Buck Converter Topology	47
4.28	Simple Buck Converter Topology	48
4.29	Schematic of the 300 W, Synchronous Buck Converter	48
4.30	Schematic Diagram of the Motor Power Circuit	49
4.31	CFD & Thermal Analysis of Buck Converter (left is static, right with CFD overlay)	50
5.1	Full Construction of the Quadruped	56
5.2	Prototypes of minimal chassis assembly and modular coxa motor frame.	56
5.3	Brass-Inserts and Side Components of Chassis	57
5.4	Carbon-Fibre Tubes and Base Panels	58
5.5	Prototypes of carbon-fibre tube end-braces and clamps with different tolerances.	58
5.6	Nylon Motor Frame with Motors Fitted	59
5.7	Assembled Chassis with Motors	59
5.8	Constructed and Mounted Leg	60
5.9	Tibia Break at Linkage Joint	61
5.10	Components for the Femur and Rotational Joint	61
5.11	Stress Analysis of Redesigned Shoulder Joint (left old, right new)	62

5.12 Stronger Tibia Joint (Double-Sided)	62
5.13 Front Face and Sensor Arrangement	63
5.14 Previous Iteration with Intel RealSense D435i	64
5.15 Lidar with Flip Mount	65
5.16 Prototypes of Front Face Covers and Sensor Mounts	66
5.17 Earth's Magnetic Field Intensity at Imperial College [UK Geomagnetic Survey] . .	67
5.18 Two magnets in the centre of the model, extending to a radius of 200mm	67
5.19 Magnetic field intensity plot from two magnets, section through X-Z plane, outer radius: 200 mm	68
5.20 Model of the main DC bus conductors	69
5.21 Magnetic field intensity plot as a result of the current in the DC bus conductors .	69
5.22 Rear Panel of the Quadruped	70
5.23 Internal Panel	71
5.24 Multiple Battery Packs Assembled	72
5.25 Spot Welded Battery Pack	73
5.26 Components used in the charging circuit - [Fig.B: HY Semiconductor REV. 6, 30- Dec-2014]	74
5.27 Buck Converter - 20A/300W	75
5.28 Schematic of the main power-delivery (including charger circuit)	75
5.29 Measurement of inrush current and ripple when energising buck converter and drives	75
5.30 Left and Top: unfiltered noise from buck converter and motor-drives interfering with controller I ² C bus. Lower Right: after filter modifications added.	76
5.31 Power Distribution Board Fused with Adafruit PWM Controller	76
5.32 Enter Caption	77
5.33 Thermal measurements taken using a FLIR Camera, after 15 minutes of continuous operation. A: Overall top view; B: Rear Section with DC-DC converter; C: Rear-Leg Motors	79
6.1 Calculating Kinematics for θ_1	83
6.2 Calculating Kinematics for θ_2 & θ_3	85
6.3 Simulated Environment using PyBullet	87
6.4 Simulated Model with Roll Orientation	89
6.5 Gait Cycle - 0 Degrees	91
6.6 Gait Cycle - 45 Degrees	92
6.7 Python Path Planning Program - A* Algorithm	94

6.8	Comparison of Cost Intensities	94
6.9	2D Lidar Mapping	96
6.10	Example of a Local Map Generated by Cartographer with Localisation	97
6.11	Oak-D Lite Camera	98
6.12	Demo Example using DepthAI's GUI [39]	99
6.13	Controls for the Quadruped - Xbox Controller	102
6.14	MobileNetV2 Tracking with Bounding Box	103
6.15	Transformation and Tracking Workflow for Depth Camera Data	103
6.16	Determining Coordinates of a Point from a Scene	104
6.17	ROS2 Tracking Data	106
6.18	Example: Detection of Frontier Goals	108
6.19	Path of Wavefront Exploration - Simulation	108
6.20	Assistant Control Flow	109
6.21	Example Prompts - Recognising Medical Emeergencies	111
6.22	Website Dashboard - Case View	112
6.23	Website Dashboard - Deployments View	112
6.24	Website Dashboard - Control View	113
6.25	Software System Architecture Diagram	115
7.1	Project Gantt Chart	119
7.2	Schematics of various form factors for Li-ion cells, including: A) cylindrical, B) prismatic, and C) pouch. [Source: Sandia National Laboratory]	124
7.3	Example of Liithium-Ion battery recycling process [Source: Elcan Industries] . . .	124

1

Introduction

Contents

1.1 Problem Statement and Specification	1
1.2 Key Questions and Problems Addressed	2
1.3 Structure of the Report	2

1.1 Problem Statement and Specification

The primary objective of this project has been to develop an autonomous quadruped robot designed to enhance the safety and efficiency of disaster response operations. Traditional search and rescue missions often involve significant risks due to unstable and hazardous conditions typical of disaster sites. The proposed quadruped robot aims to address these challenges by offering superior agility, stability, and affordability, making it a practical solution for navigating difficult environments and performing critical tasks such as search and rescue, emergency aid delivery and survivor detection.

Disaster response operations encounter significant challenges worldwide. The increasing need for humanitarian assistance and protection underscores the urgency for effective disaster response solutions. Natural disasters result in substantial economic losses each year, with poorer economic countries disproportionately affected.

A key emphasis in this project is placed on the design principles of low cost, modularity, and ease of repair. These principles are vital for ensuring that the robot is not only effective but also accessible and maintainable in resource-constrained settings. By focusing on affordability, the project aimed to make advanced robotics technology available to a wider range of users. Modularity ensures that components can be easily replaced or upgraded, enhancing the robot's longevity and adaptability. Ease of repair is important for operational endurance, especially in the challenging conditions typical of disaster environments.

1.2 Key Questions and Problems Addressed

This project sought to answer the following key questions:

- How can we design the robot to be cost-effective and accessible, especially for resource-constrained settings?
- How can we develop an efficient, autonomous quadruped robot capable of navigating diverse terrains typical of disaster environments?
- What are the most effective methods for integrating advanced navigation, machine learning, and human interaction capabilities into the robot.
- What are the specific benefits and limitations of existing quadruped robots, and how can the proposed design overcome these limitations?
- How can we ensure the robot is modular and easy to repair, facilitating quick maintenance and component replacement in the field?
- How can we streamline rescue response by integrating IBM's WatsonAI to communicate with victims, assess their condition, and provide crucial information to first responders?

1.3 Structure of the Report

This report is structured to provide a comprehensive overview of the development process, from initial concept to final evaluation. The chapters are organised as follows:

- **Background:** An overview of the current state of disaster response technology, focusing on the role of robotics.
- **Requirements Capture:** Detailed description of the functional and non-functional requirements for the autonomous quadruped robot.
- **Analysis and Design:** Examination of the system design, including architectural diagrams and design rationales.
- **Implementation:** Detailed account of the software and hardware development processes, highlighting key components and challenges.
- **Testing:** Description of the test plans and methodologies used to verify the robot's performance, along with the results.
- **Results:** Analysis of the robot's performance based on testing data, including qualitative and quantitative assessments.
- **Evaluation:** Critical evaluation of the project, comparing the outcomes with the initial objectives and discussing any deviations or discoveries.
- **Conclusions and Further Work:** Summary of the project's achievements, limitations, and suggestions for future improvements.

2

Background

Contents

2.1	Introduction to Disaster Response Robotics	3
2.2	Advancements in Quadruped Robotics	5
2.3	Economic and Humanitarian Impact of Natural Disasters	6
2.4	Challenges in Disaster Response	8
2.5	Innovative Solutions and Future Directions	8
2.6	Review of Existing Solutions	9
2.6.1	Boston Dynamics' BigDog	9
2.6.2	Boston Dynamic's Spot	10
2.6.3	Unitree's GO2	11
2.6.4	AT&T	12
2.6.5	Ghost Robotics	13
2.6.6	Deep Robotics	13
2.6.7	Other Solutions	14
2.6.8	Proposed Solution: REX	14

2.1 Introduction to Disaster Response Robotics

The dynamic field of disaster response is increasingly relying on robotics to tackle environments where human intervention is risky or impossible. The aftermath of natural disasters often presents highly unpredictable and hazardous conditions, making traditional response methods sometimes inadequate.

Robotic vehicles, using wheels or caterpillar tracks have been around for decades but have limited capabilities in the face of most disaster situations where the terrain becomes obstructed and difficult to traverse, and the distances become far.

During rescue operations, following the 9/11 incident in New York, 2001, robotic vehicles were trialled for the very first time in a real-world disaster recovery situation. These tracked vehicles, named "PackBots", the size of a shoe-box and designed to be carried in a back-pack, were deployed to search buildings for survivors and to assess damage [1].



Figure 2.1: PackBot, tracked robotic vehicle [Image: i-robot]

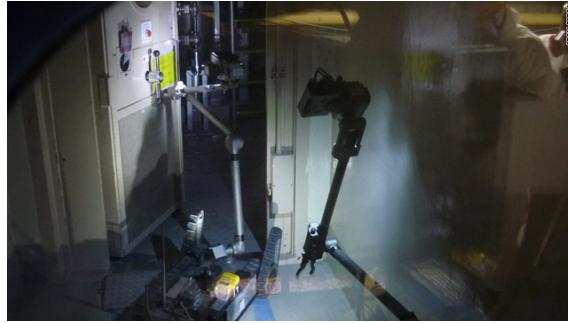


Figure 2.2: PackBots exploring Fukushima Plant in 2011 [Source: CNN]

The same robots were used again 10 years later in the Japanese Fukushima nuclear plant disaster but, during this 24 year time-span, they have not really established themselves in this type of disaster management and recovery role. This is perhaps due to limited autonomy. Although many thousands of PackBots have been built and deployed, mainly in military, bomb-disposal and reconnaissance environments, they are essentially remote-controlled vehicles (via cable or radio) with little or no autonomous capabilities.

In 2016, an EU project, named TRADR [2], successfully deployed two unmanned ground vehicles (UGVs) and three unmanned aerial vehicles (UAVs) to provide reconnaissance of the region following the severe earthquake in the town of Amatrice, in Italy. This exercise was an interesting example because the remotely-operated UAVs and UGVs had no autonomous capabilities and therefore the imagery and data from UAVs had to be used to assist the UGVs to enter in to partially-collapsed buildings. It demonstrates that enabling a robotic vehicle with autonomous capability can potentially reduce the amount of resources required.

In 2019, a Collossus robot, manufactured by Shark Robotics, assisted firefighters in the Notre Dame fire in Paris. However, the robot was not deployed in the upper levels for fear that its weight could damage the remaining historic parts of the cathedral [3]. Whilst this robot is another example of a remotely-guided vehicle (a larger version of the PackBot), it demonstrates that in some instances there may be a need for light-weight units.

Tracked robots are not the only kind that have been used in rescue and recovery situations. The Snakebot, developed by Carnegie Mellon University's Biorobotics Lab is a modular device capable of underwater operation, with the ability to pass through small crevices and openings.

An early version of this device was used to assist the search for survivors in the Mexico earth-



Figure 2.3: Colossus, Paris 2019. [Photo: Aurelien Meunier/Getty Images]

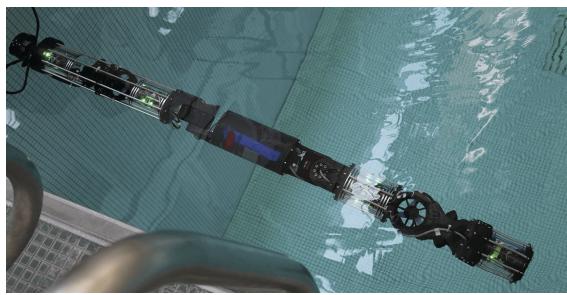


Figure 2.4: Snakebot [Photo: Carnegie Mellon, 2021]

quake in 2017 [4].

Nowadays, autonomous quadruped robots stand out for their exceptional ability to traverse diverse terrains, enabled by advances in control-systems and AI technology, offering new possibilities in search and rescue operations, emergency aid delivery, and survivor detection. The push towards creating affordable and efficient quadruped robots is driven by the urgent need to bridge a big technology gap in disaster relief, aiming to enhance operational safety and efficiency in crisis situations.

Figure 2.5 shows an autonomous quadruped robot assisting in Japan’s earthquake disaster at the start of this year.

The projected market global market for so-called "Inspection Robots" is estimated to reach (GBP) £11 bn by 2030 [5] and, together with AI, is an area of growing activity.

2.2 Advancements in Quadruped Robotics

The development of autonomous quadruped robots has been significantly shaped by the pioneering efforts of companies such as Boston Dynamics and Unitree. Boston Dynamics has set benchmarks with models like Spot, demonstrating unparalleled mobility and adaptability in complex scenarios. Conversely, Unitree has focused on standardising the technology, making it more affordable and thus expanding its accessibility to a wider audience. These companies' contributions have advanced the field technically and demonstrated the varied applications of quadruped robots, from commercial uses to critical emergency response roles.



Figure 2.5: Ghost Robotics' Vison 60 Quaruped robots, aiding Japan's earthquake relief operation, January 2024 [6]

Ideas investigated 20 years ago by scholars Alvarez & Hunt (2005) [7] and Casper & Murphy (2003) [8] provide valuable insights into the psychological and practical aspects of integrating robotics into disaster response. Their work looks into the resilience required by human responders and the complexities of human-robot collaboration in search and rescue missions, highlighting the multifaceted benefits and challenges of deploying autonomous robots in disaster scenarios. These studies underscore the importance of synergising human and robotic efforts to maximise the effectiveness of disaster management and response initiatives.

2.3 Economic and Humanitarian Impact of Natural Disasters

Natural disasters have a profound impact both economically and humanitarily. While high-income countries often face greater absolute dollar values in losses, the economic impact on low-income and lower-middle-income countries is disproportionately higher when considered as a share of their GDP. On a global scale, direct economic loss averages 0.37% of the GDP of countries reporting via the Sendai Framework Monitor from 2015 to 2022 [9]. The human toll is also significant, with millions of people affected each year.

A recent notable example of such a disaster is the 2021 earthquake in Haiti. This earthquake resulted in an estimated economic loss of (GBP) £1.3 bn, which is equivalent to 9.6% of Haiti's GDP [10]. The extensive damage highlights the severe economic strain that such disasters can place on already vulnerable economies, making recovery and rebuilding efforts incredibly challenging. Additionally, the earthquake led to significant loss of life and displacement, further exacerbating the humanitarian crisis in the region.

More recently, in February 2023, the earthquake in Turkey caused extensive devastation and loss of life, with an economic cost estimated at (GBP) £82 bn [11]. The picture below provides an understanding of the terrain that could be encountered in such disasters.

The earthquake in Japan on 1st January 2024 killed over 236 people and has been estimated to cost GBP £5 bn [12]. In this instance robot quadrupeds were deployed by the Japan Self-Defense Forces.



Figure 2.6: Collapsed residential buildings in Hatay, Turkey. [Photo: David Lombeida/Bloomberg]



Figure 2.7: Sea of Japan Earthquake, 01 Jan 2024, [Photo: Haji Mada]

2.4 Challenges in Disaster Response

Disaster response teams face numerous challenges, including navigating hazardous environments, limited accessibility, and the need for rapid decision-making under pressure. Traditional methods often fall short in addressing these challenges, leading to delays and increased risks for both victims and responders.

Autonomous quadruped robots, with their ability to traverse difficult terrains and perform complex tasks autonomously, offer a promising solution to these challenges. However, the high cost and complexity of many of these robots have been significant barriers to their widespread adoption.

2.5 Innovative Solutions and Future Directions

The drive behind the development of an autonomous quadruped robot is primarily motivated by the urgent need to enhance the safety and effectiveness of disaster response operations. This is supported by the significant attention the topic has received in recent global research, as well as the numerous ongoing R&D programs in Europe, notably under the Horizon 2023 initiative [13]. Traditional search and rescue missions, often conducted by humans, are fraught with challenges in the unstable and dangerous conditions typical of disaster sites. Autonomous quadruped robots, characterised by their agility and stability, promise a significant improvement in navigating such difficult environments.

Further evidence comes from scholarly works such as "Disaster Rescue via Multi-Robot Collaboration" by Yutaka Watanobe et al. [14], and "Robotics Applications in Natural Hazards" by Minhao Wu [15]. Both papers underscore the transformative impact of robotics in disaster response, with Watanobe et al. focusing on the collaborative aspect of multi-robot systems for rescue operations, and Wu examining the broad applications of robotics in managing natural hazards.

To address the high costs and complexity, this project focused on the design principles of low cost, modularity, and ease of repair. By utilising advanced manufacturing techniques such as Additive Manufacturing (AM) and integrating commercial off-the-shelf (COTS) components, the project aimed to develop a quadruped robot that is not only effective but also accessible and maintainable in resource-constrained settings.

Furthermore, the integration of IBM WatsonX AI enhances the robot's capabilities by allowing it to interact with victims, assess their condition, and determine the type of medical attention they may need. This capability is vital in providing first responders with critical information quickly and accurately, thus streamlining rescue operations and improving overall efficiency.

In conclusion, advancements in quadruped robotics and the strategic integration of AI technologies hold great potential in transforming disaster response operations. By focusing on affordability, modularity, and ease of repair, this project has attempted to demonstrate the possibility of creating advanced robotics technology accessible and practical for real-world applications, ultimately enhancing the safety and effectiveness of disaster response efforts. The adopted approach lowers

both acquisition and maintenance costs, making advanced robotics more accessible and adaptable for disaster response.

2.6 Review of Existing Solutions

In order to understand the current state of the art, a review was undertaken of existing, commercially-available quadruped robots. For the last two decades, enterprises have developed various quadruped robots with different energy supplies and actuation systems. Initially, quadruped robots were hydraulically-actuated, and over time, they evolved into electrically-driven robots. This section of the report evaluates various quadruped robots, focusing on their design, capabilities, and operational contexts. By analysing the design and capabilities of models in the industry, such as those developed by Boston Dynamics [16] and Unitree [17], their strengths and limitations in real-world scenarios were considered.

This review looked at how these quadrupeds navigate challenging terrains, and how they could perform search and rescue tasks, and their adaptability in disaster environments. The insights gained directly informed the development of the *REX* quadruped robot, ensuring it addresses the specific needs and challenges identified in current solutions. Below, in table 2.1, is a comparative table illustrating the evolution and diversity of current quadruped design attributes.

Robot	Year	Payload	Est. Cost	Modular	Weight	Dimensions (m)
BigDog [18]	2005	✓	£100,000	✗	110 kg	1.00 x 0.70 x 0.95
MIT Cheetah 3 [19]	2018	✓	£50,000	✗	45 kg	0.67 x 0.75 x 0.60
Sony Aibo	1999	✗	£2,500	✗	1.6 kg	0.18 x 0.29 x 0.30
AT&T Robot	2023	✓	£100,000	✗ + AI	30 kg	0.90 x 0.65 x 0.40
Deep Robotics X30	2021	✓	£70,000	✗ + AI	56 kg	1.00 x 0.80 x 0.60
Spot [16]	2017	✓	£60,000	✓ + AI	25 kg	0.80 x 0.70 x 0.34
Unitree Go2 [17]	2022	✓	£3,400	✓ + AI	18 kg	0.80 x 0.50 x 0.30
Vision 60	2022	✓	£120,000	✓ + AI	32 kg	0.87 x 0.60 x 0.36
REX (proposed)	2024	✗	£1,000	✓ + AI	2.9 kg	0.36 x 0.20 x 0.20

Table 2.1: Comparison of Quadruped Robots

2.6.1 Boston Dynamics' BigDog

The BigDog robot, developed by Boston Dynamics, marked a significant milestone in the field of quadruped robotics. Introduced in 2005, BigDog was designed to support soldiers in the field by carrying heavy loads over rough terrains. This robot paved the way for subsequent advancements in quadruped technology.

BigDog's design features a combination of hydraulic actuators, which power its high-strength legs, with a small internal-combustion engine to drive the hydraulic pump. Each leg has four main joints: the hip, upper leg, knee, and lower leg, providing the robot with remarkable stability and mobility. BigDog has the ability to carry up to 154 kg of payload, providing utility in military and logistics applications.

However, BigDog's reliance on a combustion engine presents challenges, particularly in terms of noise and operational duration. The engine's noise can be a drawback in stealth operations and,



Figure 2.8: Boston Dynamics' BigDog

of-course, the robot's operational time is limited by its fuel capacity. Additionally, hydraulic systems, whilst good at providing high specific forces, require significant maintenance, add weight and require space, which may restrict BigDog's use to particular applications where its unique capabilities outweigh these considerations. These drawbacks, along with its high price tag of approximately (GBP) £100,000, limit its practicality for widespread use in disaster recovery response.

In summary, BigDog represents a milestone in the evolution of quadruped robotics. Its design and capabilities have set the stage for the development of more advanced robots, capable of performing complex tasks in a variety of challenging environments.

2.6.2 Boston Dynamic's Spot



Figure 2.9: Boston Dynamics's Spot

Quadruped robotics has seen remarkable advancements in recent years, with Boston Dynamics' Spot robot standing out as a pioneering example in this domain. Spot represents a significant improvement over its predecessor, BigDog, particularly in terms of agility and electronic control.

While Spot sacrifices some load-carrying capability compared to BigDog, its enhanced agility and adaptability make it more suitable for a wider range of applications.

A notable example of Spot's utility in dangerous applications is its deployment in the Chernobyl Exclusion Zone, where it was equipped with radiation sensors to safely assess hazardous areas that have been inaccessible to humans for decades. Costing around (GBP) £60,000, its size makes it sufficiently compact for various terrains and applications, including disaster response. However, its acquisition cost remains high, coupled with its reliance on manual control or programming for navigation and tasks, which may limit its accessibility to certain organisations, particularly those with budget constraints or limited technical resources. This factor could make Spot less feasible for widespread adoption in diverse operational scenarios, including those where autonomous, cost-effective robotics solutions are needed.

The adaptation of the Spot robot for autonomous operation, as detailed in the paper by Bouman et al. [20], nevertheless underscores a significant leap in quadruped robotics, demonstrating their potential in autonomous exploration of extreme environments. This work highlights how the integration of advanced autonomy architecture can extend the capabilities of quadruped robots like Spot, enabling them to undertake large-scale, long-duration missions in challenging terrains, a critical advancement for disaster response applications.

2.6.3 Unitree's GO2



Figure 2.10: Unitree's GO2

Unitree's quadruped robots, such as the GO2, are positioned as more affordable alternatives in the quadruped robot market. The GO2, particularly noted for its cost-effectiveness, is designed for versatility in research and development settings, offering capabilities like AI-driven human detection and navigation in complex terrain. This makes the GO2 a significant improvement over earlier quadrupeds by offering advanced functionalities at a lower price point, though it does come with trade-offs in terms of robustness and specialised applications.

A common challenge for quadruped robots like those from Unitree includes balancing advanced functionalities with cost and practical usability in specific fields such as disaster response. While the GO2 offers impressive technological advancements, its application in specialised scenarios like search and rescue operations in disaster-stricken areas might not be as refined as robots specifically

designed for those purposes. Their robot can be purchased at a base price of (GBP) £3,400, with optional sensors at additional cost. However, despite having a relatively low base-price, the cost of individual parts is exceptionally high. In hazardous environments, quadrupeds are prone to damage and will require ongoing maintenance.

The cost of maintenance in the robot can be seen as a significant drawback, especially from a budgetary perspective. At (GBP) £400 for a single motor, summing to (GBP) £4,800 for all 12, the expenses for maintenance or repair could become substantial. This high replacement cost could potentially limit the robot's affordability and accessibility, particularly for smaller organisations or in scenarios where budget constraints are a primary concern. Such a cost structure may also impact the long-term sustainability of using the robot, as frequent or multiple replacements could lead to prohibitively high operational costs over time.

In light research on the "Adaptive Force-based Control for Legged Robots" [21], it is evident that advancements in control systems can significantly enhance the performance of more affordable quadruped robots like those from Unitree, potentially addressing some of the challenges related to terrain adaptability and load-bearing. This could make robots like the GO2 more viable in diverse operational scenarios, including disaster response, by improving their functionality while maintaining cost-effectiveness.

2.6.4 AT&T



Figure 2.11: AT&T's Allen Beadle with Payload [Photo: AT&T]

This quadruped robot is a relative new-comer, having been launched a little over a year ago, in March 2023. It is a collaborative design with Ghost Robotics (resemblance to Vision 60) intended for varied applications, including military and, specifically, first-aid response in disaster recovery. It is approximately the size of a Labrador dog and one of its notable features is that it appears to be, to some extent, water-proof and capable of wading in water. It operates on AT&T's FirstNet

communications network, which was put in place in the USA, following the 9/11 incident. Much of the robot is built around open-source technologies, but it also employs AT&T's patented Geocast system that enables it to operate 'beyond line of site' [22].

2.6.5 Ghost Robotics

The company was founded in 2015 with the aim of "making a better robot than Boston Dynamics". It specialises in quadruped robots, having delivered around 450 units since then. Its newest design is the Vision 60, hardened for severe environments, with a cost price of (GBP) £120,000 [23], which appears to have found markets in homeland security, military, and a limited number of commercial applications. On 11 November 2022, Boston Dynamics filed a lawsuit against the company for alleged infringement of several key patents.



Figure 2.12: Ghost Vision 60 [Photo: Ghost Robotics]

2.6.6 Deep Robotics

A Chinese company has developed a number of quadruped robots, the Lite, the X20 and the larger, X30 (56kg), for a variety of applications. Appearing like a Boston-Dynamics look-alike, the X30 is marketed as capable of being deployed in rescue missions. The company appears notable not



Figure 2.13: EX30 in a simulated rescue mission [Photo: Deep Robotics]

only for the capability of its robotic quadruped platforms, but also for the wide variety of sensors and detection equipment that it offers as on-board options. The company also makes its developed motor and joint technology available to third parties [24].

2.6.7 Other Solutions

Beyond Spot, Unitree's GO2 and the others, mentioned above, there are several other quadruped robots in the market, each with unique features and capabilities. However, many of these robots are not ideally suited for rescue and search operations due to various limitations.

MIT Cheetah: The MIT Cheetah series, including Cheetah 3, is known for its high-speed capabilities and agility. These robots are designed for advanced locomotion research and can perform dynamic movements such as running and jumping. However, their focus on speed and agility comes at the expense of durability and load-bearing capacity. The MIT Cheetah robots are not optimised for carrying heavy loads or operating autonomously in unpredictable and challenging terrains, which are important requirements for effective rescue operations.

Sony Aibo: Originally launched in 1999, Sony's Aibo is a small, consumer-oriented robot designed primarily for entertainment and companionship. While Aibo showcases impressive robotics technology in a compact form, it lacks the robustness required for search and rescue missions. Its lightweight build and focus on social interaction features make it unsuitable for navigating hazardous environments.

These robots, while technologically advanced and suitable for their intended applications, do not meet the specific needs of rescue and search operations. Effective disaster response requires robots that can navigate through rough terrain and operate autonomously for extended periods.

2.6.8 Proposed Solution: REX

This project's quadruped robot, named REX (Rescue & Exploration), was designed specifically to address many of the shortcomings of existing solutions by building upon the foundational principle of providing assistance in disaster response scenarios. REX has attempted to combine the strengths of current technologies while overcoming their limitations, ensuring it is highly agile and stable, capable of navigating challenging terrains with ease. Its design is centred around the purpose of delivering aid and support in critical situations.

REX has been designed with a focus on modularity and ease of maintenance, which allows for efficient repairs and reconfiguration in diverse operational environments. The open-source nature of REX's underpinnings encourages community-driven enhancements and customisation, potentially accelerating technological advancements and fostering widespread adoption. Furthermore, REX aims to provide a more cost-effective solution compared to existing alternatives, thereby increasing its accessibility to a broader spectrum of organisations.

Additionally, REX integrates IBM WatsonX, a powerful AI platform, to interact with individuals and improve situational awareness and response. This integration allows REX to collect data and relay vital information to off-site operators, to assist in emergency response.



Figure 2.14: REX - a Cost-Effective & Autonomous Quadruped Robot

In summary, REX was developed to provide a robust, agile, and cost-effective solution for disaster response operations, with a design built explicitly around the goal of assisting others. It leverages advanced AI technologies like IBM WatsonX to deliver superior performance in hazardous situations, addressing the needs identified in current quadruped robot technologies. The budget for this project had been agreed with its sponsor, IBM, to be limited to (GBP) £1,200. The design was therefore developed to meet this target figure.

Size

As seen in Figure 2.15, REX is the smallest robot in the group. Evidently, while its diminutive size may be an advantage in certain scenarios, more likely than not, it will struggle to navigate through areas where there may be a lot of rubble and debris from fallen buildings (see Figure 2.6). This was a consideration of the design process, which ideally would have seen the development of a larger robot. However, due to this project's budgetary and manufacturing constraints, it was necessary to make some compromises. Nevertheless, this robot remains an effective demonstrator of the key enabling technologies contained within its small envelope. The technology within REX could be easily re-located into a larger chassis, with larger battery-pack and more powerful motors. There would also be room to accommodate additional sensors and features.

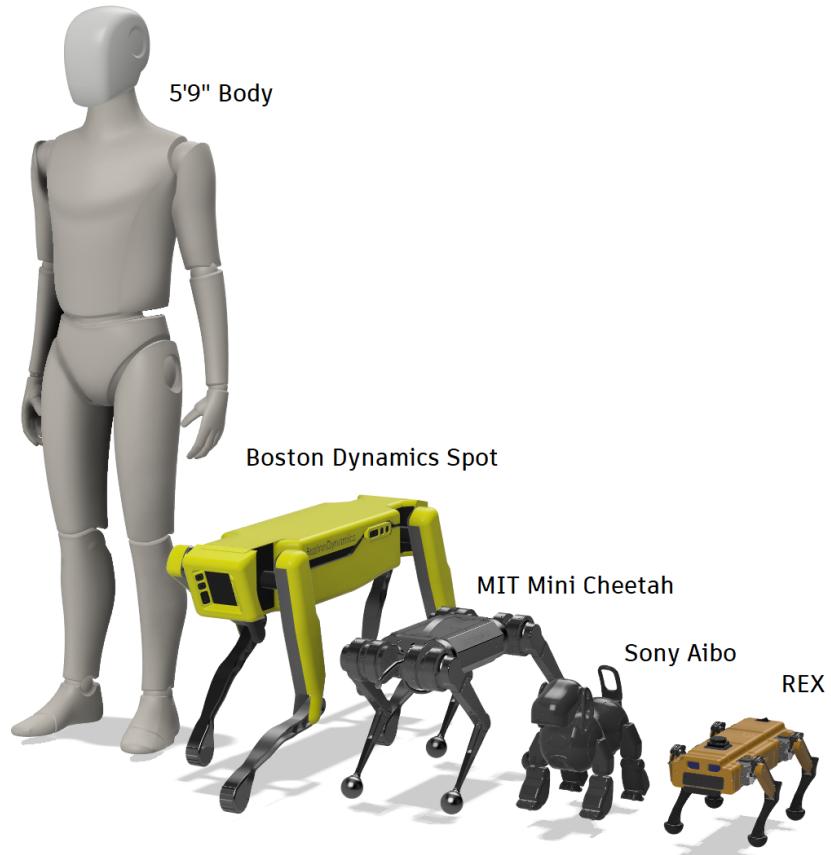


Figure 2.15: Size Comparison of Current Quadrupeds and REX



Figure 2.16: REX and Spot, side-by-side

3

Requirements Capture

Contents

3.1 Addressing Current Limitations	17
3.1.1 Proposed Advancements	17
3.2 Requirements	18
3.2.1 Design and Build	19
3.2.2 Autonomous Navigation	20
3.2.3 Communication Interface	21
3.2.4 Facial Recognition Technology	21
3.2.5 Web Application Development	21

3.1 Addressing Current Limitations

Current quadruped robots, despite their advanced mobility, often lack fully integrated AI, limiting their autonomy in unpredictable disaster scenarios. Most also come with high acquisition and maintenance costs, restricting widespread deployment. Furthermore, their interaction capabilities with humans and other robots are generally limited, which is a critical aspect in collaborative search and rescue missions.

3.1.1 Proposed Advancements

This project proposed a number of key features and improvements, aimed at addressing current limitations of other quadruped robots.

Integrated AI and Machine Learning

Incorporating deep learning and machine learning algorithms for enhanced autonomy enables real-time decision-making and adaptive navigation in complex environments without human intervention.

Autonomous Operation

Equipped with sophisticated sensors and navigation systems, it autonomously maps disaster zones, avoids obstacles, and locates survivors, supported by advanced algorithms for efficient navigation.

Interactive Capabilities

With natural language processing and signal-based communication, the quadruped has improved interaction with victims and human responders, optimising collaborative operations.

Affordability

The design focuses on cost-effectiveness by employing novel and economical manufacturing techniques, such as Additive Manufacturing (AM), and utilising COTS components. This approach aims to lower both acquisition and maintenance costs, making advanced robotics more accessible and adaptable for disaster response.

3.2 Requirements

Requirement	Description
Integrated AI	Incorporate deep learning and machine learning for enhanced autonomy
Autonomous Operation	Equip with sensors and navigation systems for mapping, obstacle avoidance, and survivor location
Interactive Capabilities	Implement natural language processing for improved interaction with victims and responders
Affordability (less than £5,000)	Use cost-effective materials and manufacturing techniques (AM, COTS components)
Durable Materials	Design chassis with carbon-fibre-composite for strength and lightweight
Swappable Battery System	Enable easy battery replacement to extend operational efficiency
Compact Design	Maintain a footprint of 360x200mm for mobility and cost-effectiveness
Heat-Resistant Components	Use heat-resistant thermoplastic for motor enclosures
Navigation Algorithms	Implement OAK-D Lite depth camera, LD19 LIDAR, and SLAM for dynamic mapping
Terrain Adaptability	Integrate IMU sensors for stability on uneven surfaces
Communication Interface	Integrate IBM WatsonX Assistant for interactive communication
Facial Recognition	Use advanced algorithms for locating and identifying missing persons
Web Application	Develop application for real-time data display, sensor readings, and manual control

Table 3.1: Requirements Table for Quadruped Robot

A comprehensive requirements capture process was conducted to ensure that the quadruped robot met all necessary functionalities for rescue and exploration missions. Key requirements identified included robust autonomous navigation, effective obstacle recognition and avoidance, and reliable communication interfaces. These requirements shaped the design and development phases, ensuring that the robot could operate effectively in disaster zones and provide valuable assistance in search and rescue operations.

Challenges and Solutions

Common challenges in requirements capture include unclear expectations, evolving requirements, and conflicting interests. In this project, these challenges were addressed by:

- **Regular Communication:** Maintaining continuous communication with my industrial supervisor to ensure alignment of expectations and provide progress updates.
- **Iterative Development:** Allowing for incremental updates and adjustments to requirements as the project progressed.

3.2.1 Design and Build

Materials and Components

In designing the quadruped robot, the choice of materials and components was driven by the need for durability, cost-effectiveness, and ease of maintenance. However, the eight month timescale for this project also had to be considered and this favoured a certain manufacturing approach. Additive Manufacturing, also known as 3D printing, was selected as the primary method due to its benefits for rapid prototyping and efficient design testing. AM allows for the creation of complex parts in one piece, which might not be feasible with other manufacturing techniques. Additionally, this method supports eco-friendly practices by minimising waste, using only the necessary material for each part.

On-demand production was another requirement, ensuring that parts could be created as needed, thereby reducing the need for extensive inventory and facilitating easy and timely replacements or upgrades. The flexibility offered by 3D printing was significant for a more adaptable and scalable manufacturing approach.

To ensure the robot's structural integrity, the chassis utilises parts in carbon-fibre-composite, a material chosen for their exceptional strength-to-weight ratio. This material enhances the robot's durability without compromising agility, which is vital for manoeuvring in varied terrains.

For the electronic components, the decision was to use off-the-shelf commercial electronics and a single-board Raspberry Pi 4B computer. The latter, is a powerful computer which can be purchased for less than (GBP) £80 and is an example of a cost-effective solution. Its wide availability helps ensure longevity in the repair or replacement process, making the robot more accessible and maintainable.



Figure 3.1: 3D Printing of REX's Legs

Design Considerations

The robot's design had to incorporate a swappable battery system to enhance operational efficiency, particularly during extended missions in disaster zones. This feature was essential to ensure minimal downtime and facilitate easy battery replacement in the field. Ease of maintenance was also a critical requirement, leading to the design of field-replaceable components that are easily accessible.

To achieve ease of mobility and nimbleness, the robot's footprint was kept compact at 360x200mm, which also helped in keeping the cost down. The motor enclosure required a heat-resistant thermoplastic to withstand the heat generated during operations, ensuring the longevity of the motors. The legs, facing diverse terrains and pressures, were constructed from a tough material to prioritise durability and strength. These material choices were essential to the robot's functionality, ensuring reliability and endurance in demanding conditions. Additionally, AM can process metals and ceramics, offering a straightforward upgrade path for future performance enhancements in harsh environments.

3.2.2 Autonomous Navigation

Navigation Algorithms

For the robot's autonomous navigation capability, an OAK-D Lite depth camera and LD19 LIDAR was paired with Simultaneous Localisation and Mapping (SLAM) technology. This setup enables the robot to dynamically map its surroundings, necessary for navigating complex disaster environments. Algorithms for obstacle recognition and avoidance were also required, tuned to identify various obstacles and hazards typical in disaster zones, allowing the robot to navigate safely and

effectively. The combination of SLAM with machine learning ensures a high degree of adaptability and situational awareness, essential for autonomous operations in unpredictable settings.

Terrain Adaptability

To handle diverse terrains, the robot required Inertial Measurement Unit (IMU) sensors to maintain balance on uneven surfaces. These sensors enable the robot to dynamically adjust its posture, ensuring stability during movement. Inertial measurement was crucial for each leg to adapt to varying ground levels, providing real-time adjustments for optimal stability and mobility. The integration of IMUs and Time of Flight sensors was necessary for effective navigation across a range of terrains, from flat surfaces to more complex, uneven landscapes encountered in disaster areas.

3.2.3 Communication Interface

Interaction Capabilities

The robot needed to autonomously identify and approach people, inquiring if they required assistance. This required the integration of IBM WatsonX Assistant for interactive communication, posing questions and responding appropriately based on the individual's answers. This advanced conversational interface was essential for gathering vital information, such as state of consciousness or medical needs. Moreover, the interactions needed to be accessible via a web-UI, allowing remote human operators to monitor interactions and view the location of each person on a map.

3.2.4 Facial Recognition Technology

Enhancing the robot's search and identification capabilities in disaster zones necessitated advanced facial recognition technologies. This was crucial for locating missing persons, efficiently updating their status, and assisting emergency response teams. Reliable and accurate recognition algorithms were required to function effectively even in challenging conditions common in disaster environments.

3.2.5 Web Application Development

Web Application Functionality

A comprehensive web application was required to display real-time data from the robot, including sensor readings and various status indicators. The application needed to show positional data of each human encounter and have a dedicated section for identified individuals. This section would display information recognised by the facial recognition system, including their medical condition and severity. Depending on the severity, operators could sort through each entry, and the marker on the map would change colour based on urgency. This multi-faceted design ensured users could

access all critical data, facilitating informed decision-making during disaster response operations. Additionally, the application needed to allow users to take manual control of the quadruped robot.

4

Initial System Design & Analysis

Contents

4.1 High-Level Design Overview	24
4.1.1 Introduction	24
4.1.2 System Architecture	24
4.2 Chassis Design	25
4.2.1 Materials and Construction	25
4.2.2 Motor Assembly	28
4.3 Leg Design	29
4.3.1 Exploration of Leg Mechanisms	29
4.3.2 Proposed Leg Mechanism	32
4.3.3 Motor Configuration	34
4.3.4 PWM Servo Control	34
4.3.5 Design Choices	34
4.4 Front-End Sensor Integration	36
4.4.1 9-axis IMU (Inertial Measurement Unit)	36
4.4.2 Environmental Temperature Sensor	36
4.4.3 OLED Display	36
4.4.4 Depth Camera (OAK-D LITE)	37
4.4.5 Dual Microphone Array	37
4.4.6 Mounting and Protection	37
4.5 Rear-Section	38
4.5.1 Power Control Switches	38
4.5.2 USB-C Quick Charge Module	38
4.5.3 Safety Switch	39
4.5.4 Mounting and Accessibility	39
4.5.5 Magnetic Rear Cover	39
4.6 Side and Top Sections	39
4.7 Battery Pack Design	41
4.7.1 Rationale of Li-Ion Cells	41
4.7.2 Power Calculations	41
4.7.3 Charging Mechanism	42
4.7.4 Design Considerations	43
4.8 Considerations for Productionisation	44
4.9 Electrical Design	45
4.9.1 Power Distribution Overview	45
4.9.2 Battery Charging Circuit	45
4.9.3 Motor Power Circuit	47

4.10 Software Design	50
4.10.1 Design Principles	51
4.10.2 Software Architecture	51
4.10.3 Navigation Subsystem	51
4.10.4 Vision Subsystem	52
4.10.5 Control Subsystem	52
4.10.6 Communication Subsystem	52

4.1 High-Level Design Overview

4.1.1 Introduction

This section provides a high-level overview of the autonomous quadruped robot's system design, summarising the key architectural components and design principles. Building on the requirements captured in the previous chapter, this section aims to give a coherent picture of the overall system and its intended functionality.

4.1.2 System Architecture

The system architecture integrates multiple subsystems to achieve the robot's operational goals. Figure presents a block diagram of the overall architecture, highlighting the major components and their interactions.

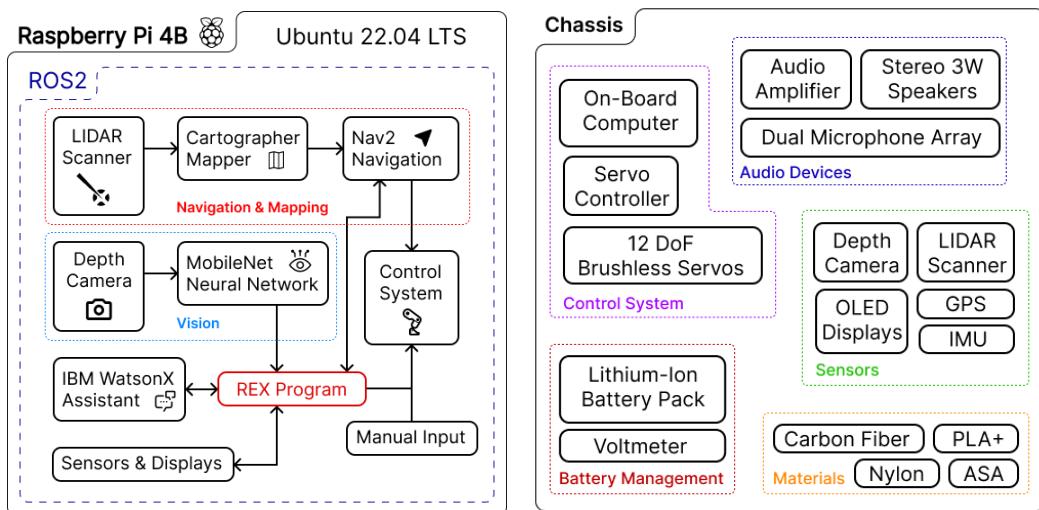


Figure 4.1: Software and Hardware Subsystems

The architecture includes:

- **Hardware Subsystem:** This subsystem includes the physical components of the robot such as the chassis, control electronics, audio devices, sensors, battery management, and the

materials used for construction. It is designed to provide a robust and durable platform for all operational needs.

- **Power Subsystem:** This includes the power management systems that ensure efficient power distribution for charging the battery, powering motors, and supplying energy to the on-board computer and sensors, enabling sustained operations.
- **Software Subsystem:** Encompasses navigation algorithms (SLAM), neural networks for person detection, communication interfaces for the assistant, control algorithms, and ROS2 integration. These components facilitate autonomous navigation, real-time decision-making, and effective interaction with the environment and users.

4.2 Chassis Design

The chassis is an important component of the quadruped robot, providing the structural framework that supports all other subsystems. The design considerations for the chassis focused on rigidity, mass, and integration capabilities.

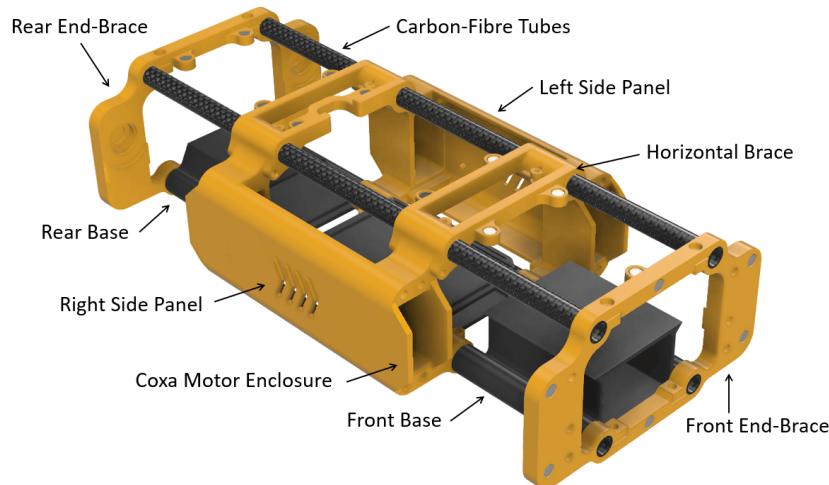


Figure 4.2: Chassis Assembly - CAD Model

4.2.1 Materials and Construction

The chassis was designed with carbon-fibre-composite tubes, as the main structural members. The material was chosen for its high strength-to-weight ratio , as shown in table 4.1, to help ensure the robot can withstand the physical stresses encountered in diverse and rugged terrains, while remaining sufficiently lightweight to maintain agility and reduce energy consumption. The hollow nature of the tubes helps to increase the bending stiffness, as well as reduce mass.

A basic stress analysis was carried out on the carbon-fibre tube design to determine maximum bending under an extreme load of 30kg on a single tube. The results are shown in 4.3, which demonstrates a maximum displacement on 0.50 mm under load.

Property	Value
Wall Thickness	1 mm
Outside Diameter (OD)	10 mm
Inside Diameter (ID)	8 mm
Weight per metre	45.6 g
Colour	Black
Construction	Woven Finish Roll Wrapped
Resin Matrix	Epoxy
Fibre Orientation	0, 90
Tensile Strength (Lengthways)	650 MPa
Tensile Modulus (Lengthways)	64 GPa
Tg Onset (DMA)	120 °C

Table 4.1: Carbon Fibre Material Properties

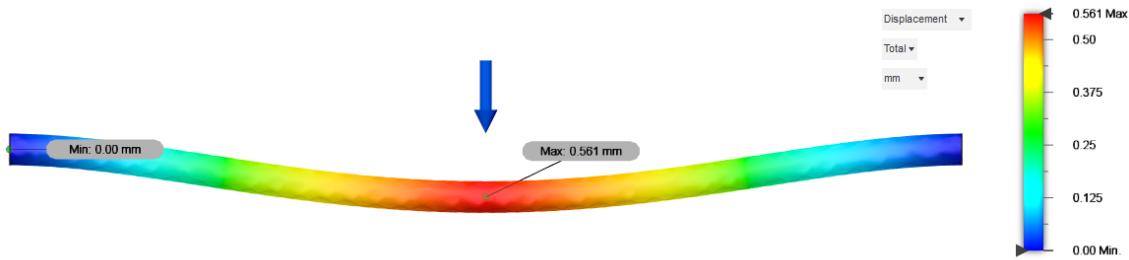


Figure 4.3: Exaggerated Displacement of a Carbon-Fibre Tube under a 300N Load

To complement the carbon fibre tubes, 3D-printed PLA+ components were used to build the supporting structure, including side panels, and mounting panels. PLA+ (Polylactic Acid Plus) is an enhanced version of standard PLA, a biodegradable thermoplastic derived from renewable resources like corn starch or sugarcane. PLA+ has been modified to improve its strength, flexibility, and durability, making it more suitable for demanding applications compared to regular PLA.

The use of PLA+ for the 3D printed components ensured high tensile strength and flexibility (table 4.2), which are critical for parts subjected to mechanical stress and strain. Its good elongation at break allows components to deform slightly under load without breaking, adding to the overall durability of the chassis. Additionally, the high flexural modulus of PLA+ indicates excellent stiffness, providing necessary rigidity for structural components.

By combining carbon-fibre-composite tubes and 3D printed PLA+ parts, the chassis design balances light weight with high strength, helping to ensure the robot's performance and longevity in the proposed challenging environments.

Property	Value
Density	1.23 g/cm ³
Tensile Strength	60 MPa
Elongation at Break	20%
Bending Strength	74 MPa
Flexural Modulus	1973 MPa

Table 4.2: PLA+ Material Properties

The chassis was designed with a modular structure, allowing for easy assembly, maintenance,

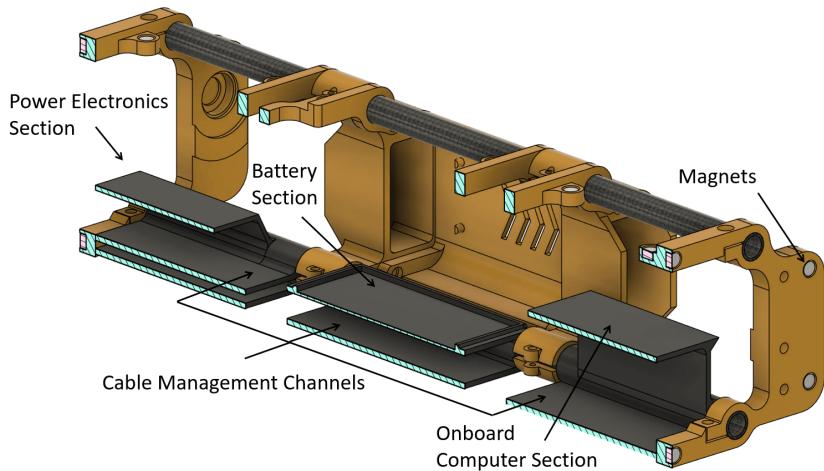


Figure 4.4: Chassis Cross-Section - CAD Model

and upgrades. This modularity also facilitated rapid prototyping and testing of different configurations. The rectangular, monocoque chassis was designed around four carbon-fibre tubes, one at each external extremity, leaving an open, internal space for components. The spatial separation of the tubes helps to provide a good resistance to flex and twist. Various components, including motor mounts, side panels, and mounting panels, slide into the carbon fiber frame, permitting flexible positioning and easy integration. Most of the screw fasteners used were M3, ensuring standardisation and ease of assembly. Specific mounting points, made possible with threaded brass inserts, were integrated into the chassis for secure attachment of sensors, control systems, and other components. Additionally, dedicated channels for cable management ensured that power and data cables were securely and separately routed and protected from damage during operation. The cross section of the chassis, shown in figure 4.4, highlights these channels.

The figure below (Figure 4.5) is an FEA of the chassis' brace. A vertical load of 30N had been applied to each mounting point, which is approximately 4 times the expected load. The analysis indicated that overall the design is structurally sound. As expected, the thinnest sections demonstrate the highest stress levels, particularly in the direction of the load, with the lower, inner, corners of the carbon-fibre mounting holes demonstrating the highest stress of 1.7 MPa. There can sometimes be porosity issues in 3D printing that prevent the material reaching its quoted maximum tensile stress, which in this case is 60 MPa. However, this analysis showed there is more than an adequate margin in this design.

To further enhance maintenance and repair capability, 6mm x 3mm neodymium magnets were added to the top, front, and rear ends of the chassis. These magnets allow the corresponding covers to securely attach magnetically, enabling users to have tool-free access to internal components. This design feature simplifies the process of performing maintenance and repairs, helping to ensure that the quadruped can be kept in optimal working condition with minimal effort. Unfortunately, this early decision with magnets did subsequently cause some difficulties with the implementation of a navigational magnetometer. This issue is described in the next section, on Hardware Implementation.

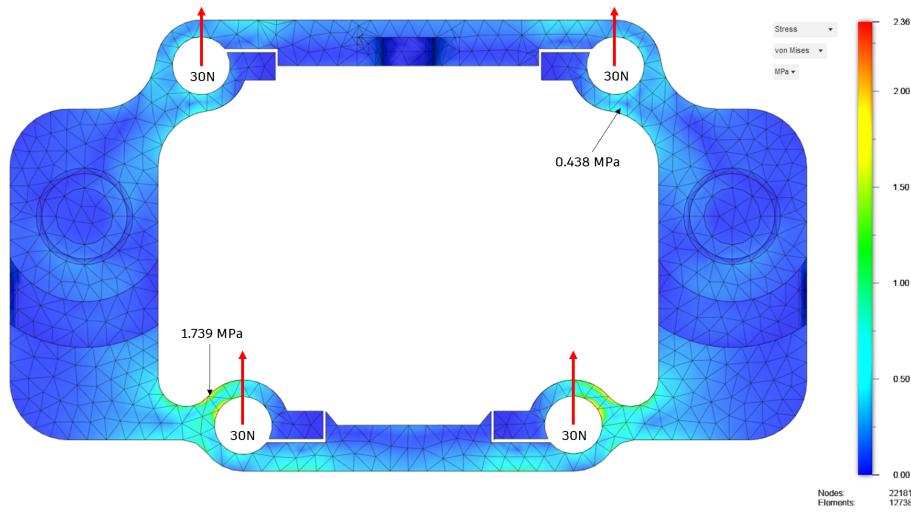
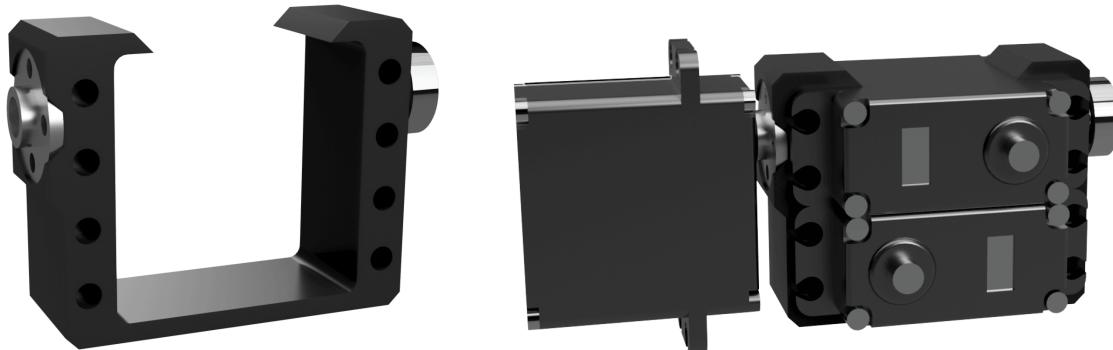


Figure 4.5: FEA of Chassis Brace Under 30N Load

4.2.2 Motor Assembly

After careful consideration, nylon (Polyamide 6) was selected as the material for the motor bracket due to its relatively high melting point of 220°C, with a glass transition temperature of 45°C. This characteristic is particularly important for these parts since the motors can generate significant thermal losses and operate at elevated temperatures. The nylon material ensured that the casing could withstand this heat without deforming, thus maintaining the integrity and performance of the motors over time. Furthermore, nylon was suitable for the motor brackets because of its toughness and durability, which help it withstand mechanical stress and vibrations. Additionally, its good vibration damping properties allow it to absorb and dissipate motor vibrations, reducing their transmission to other parts. Note that the analysis omitted the fasteners which clamp onto the carbon-fibre tubes, thus simulating a far worse case scenario.



(a) Nylon Motor Bracket - CAD Model

(b) Motor Layout - CAD Model

Figure 4.6: Side by side comparison of CAD models.

Each motor bracket was designed to hold the two main servo motors: one for controlling the femur and another for controlling the tibia. A third servo, equipped with an aluminum servo arm disc, is also secured to the bracket. This servo is responsible for actuating the quadruped's coxa joint. Additionally, a ball bearing on the opposite side of the bracket supports and ensures smooth

rotation.

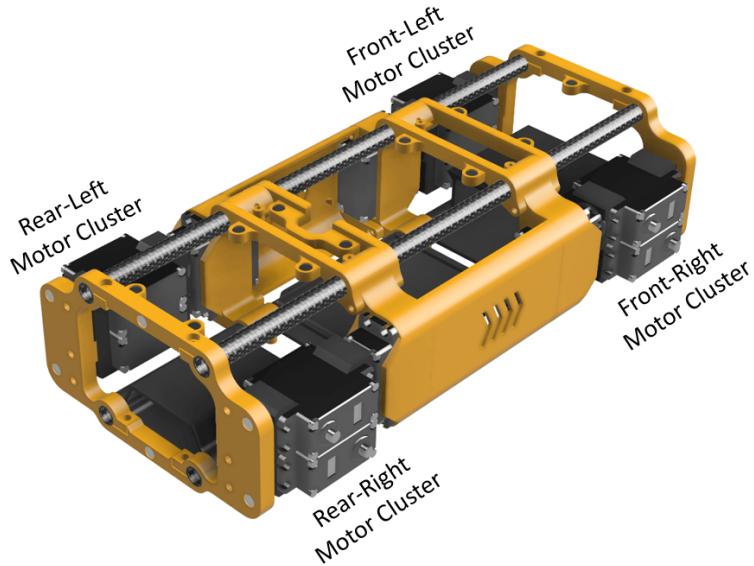


Figure 4.7: Chassis & Motor Assembly - CAD Model

The chassis, depicted in Figure 4.7, clearly shows the arrangement of the 12 servo motors, with three motors per leg. These motors are securely held within the chassis, helping to ensure stability and precise control of each leg's movement.

4.3 Leg Design

4.3.1 Exploration of Leg Mechanisms

The design of leg mechanisms in quadruped robots greatly impacts their mobility, efficiency, and adaptability to different terrains and tasks. To develop an optimal leg mechanism, it was important to research and evaluate the most common types of leg mechanisms, examining their advantages and disadvantages. This research provided valuable insights which guided the decision-making process in designing the most suitable legs for the quadruped robot.

Belt-Driven Leg Mechanism

Belt-driven leg mechanisms have become a notable advancement in powering quadruped robots, providing a mix of efficiency and flexibility. A study by Tanikawa et al. [25] showcases the effective use and benefits of belt and pulley systems in robot movement. This method uses the inherent efficiency and dynamic capabilities of belts to enable intricate leg movements, establishing a standard for future robotic mobility designs. By reducing the mass located on the joints, this approach improves the robot's dynamics, leading to more agile and responsive movements.

Belt-driven systems employ one or more belts, typically made from reinforced rubber or synthetic materials, which loop around pulleys attached to the motors and the leg joints. The rotation

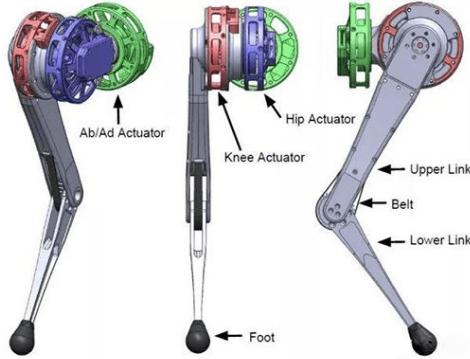


Figure 4.8: Belt-Mechanism for the MIT Cheetah Leg

of the motor pulley translates into linear or rotational movement of the leg joints via the belt, enabling the robot to perform walking, running, or climbing motions.

These mechanisms can accommodate a variety of leg configurations, from straightforward single-joint movements to complex multi-jointed articulations. Toothed belts are especially effective in ensuring efficient power transmission with minimal slippage when correctly tensioned, contributing to the robot's overall energy conservation, a vital aspect for battery-operated models. However, maintaining the correct belt tension is critical, as over-tensioning can lead to premature wear, while under-tensioning might result in slippage and loss of control precision. Belts can be sensitive to environmental factors such as dust, moisture, and temperature fluctuations, which can affect the belt's material properties and performance.

Linear Actuator Leg Mechanism

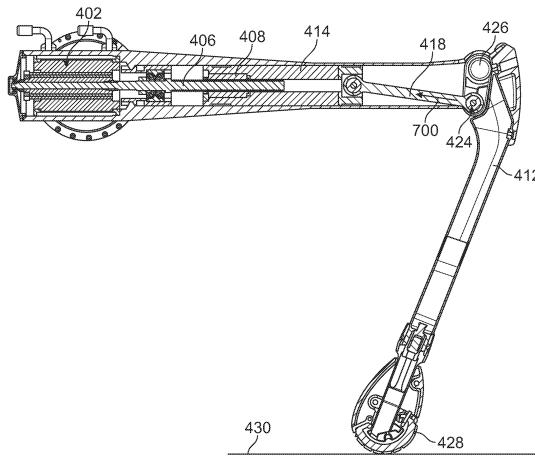


Figure 4.9: Patent of Boston Dynamics Leg Mechanism
[26]

Linear actuator legs represent a significant advancement in the design and functionality of quadruped robots, exemplified by the renowned Spot robot developed by Boston Dynamics. In

this type of design, the legs incorporate linear actuators to extend and retract, enabling precise control over the robot's movement and posture.

Linear actuator legs consist of telescoping mechanisms that allow for extension and retraction along a straight line. Driven by electric, hydraulic, or pneumatic actuators, these legs enable precise positioning with the help of sensors, simplifying navigation across complicated terrains and the execution of particular tasks. Linear actuators used in the Spot robot are capable of handling significant loads, making them well-suited for carrying heavy payloads—a key requirement for numerous commercial and industrial applications.

However, linear actuators must be sized to handle the full force required, which can sometimes lead to a heavier assembly. While they offer several advantages, such as high control precision and efficiency, they also introduce mechanical complexity, potentially affecting maintenance needs and the reliability of the overall system.

The Watt Six-Bar Leg Mechanism

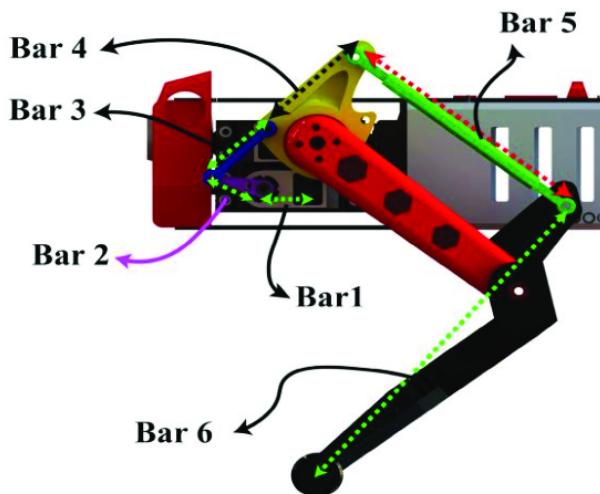


Figure 4.10: Watt Six-Bar Leg Mechanism from Rahman et al. [27]

Watt's Linkage, a type of mechanical linkage originally invented by James Watt for steam engines, has found applications in various fields, including robotics. In the context of quadruped robots, an adaptation known as the Watt six-bar mechanism provides an innovative approach to leg design, offering a simple, but effective design for locomotion.

The design depicted in figure 4.10 exemplifies a low-cost yet effective approach to quadruped locomotion, utilising the Watt six-bar mechanism. This design strategy, detailed in the work of Rahman et al. [27], demonstrates the potential for creating accessible and reliable robotic systems suitable for a wide range of applications. The authors' methodology underscores the importance of integrating robust mechanical concepts with cost-effective materials and fabrication techniques, setting a precedent for future developments in the field of affordable robotic quadrupeds.

Leg Mechanism Conclusion

Having evaluated the various leg designs for quadruped robots, it became evident that each mechanism presented a unique set of advantages tailored to specific operational requirements and design constraints. Rigid legs offer simplicity and durability, compliant mechanisms excel in energy efficiency and adaptability, articulated designs provide exceptional mobility and load distribution, and belt-driven systems appear notable for their smooth operation and maintenance ease. However, when considering the implementation of these designs on a smaller scale, as in the development of a compact quadruped robot, the choice of the most suitable leg mechanism became critically dependent on the spatial constraints and functional demands of the robot.

4.3.2 Proposed Leg Mechanism

The Watt's Linkage mechanism, particularly in its advanced form as the Watt six-bar design, emerged as the most appropriate choice for a small-scale quadruped robot. This preference was grounded in the geometric and spatial efficiency of the Watt's Linkage, which allows for a compact yet highly functional leg design without the need for internal actuators or belt systems that might not be feasible or practical to implement. With its compact design, precise control, and efficient use of space, it stood out as the most suitable leg mechanism for this application.

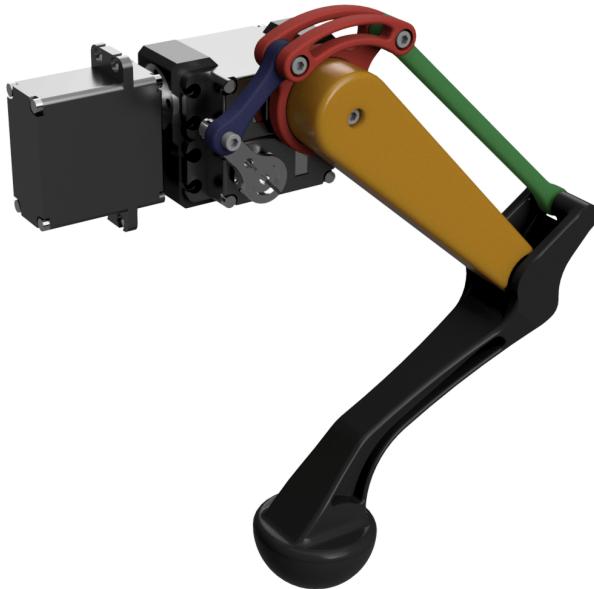


Figure 4.11: Full Leg Assembly - CAD Model

A decision was therefore taken to incorporate a Watt six-bar linkage system, which provides a naturalistic range of motion and stability. Figure 4.12 illustrates the specific configuration of the six-bar linkage design. Extensive use of CAD modelling was made to optimise the motion and geometry of each joint. As mentioned above, the robot was designed with three servos per leg, responsible for three distinct types of movement that mimic the joints in a biological leg:

- **Hip Servo (Coxa):** This servo controls the side-to-side movement of the leg, allowing the robot to widen or narrow its stance. It acts similarly to a hip joint in animals, providing the

lateral motion that is the first degree of freedom. In the quadruped robot, this is crucial for balance, maneuvering, and initiating walking gait patterns.

- **Upper Leg Servo (Femur):** Positioned at what would be the thigh, this servo controls the forward and backward swinging of the leg. This movement is akin to the flexion and extension observed at an animal's hip, enabling the robot to step forward or backward. It is responsible for propelling the robot and is critical for the second degree of freedom in the leg's movement, directly affecting stride length and walking speed.
- **Lower Leg Servo (Tibia):** This servo operates the bending and extending motion of the lower leg, similar to the knee joint in a biological leg. It allows the robot's foot to lift off the ground and then be placed down for the next step. This servo is essential for adjusting the leg's height and angle to navigate different terrains and is key to the third degree of freedom, contributing to the stability and adaptability of the robot's gait.

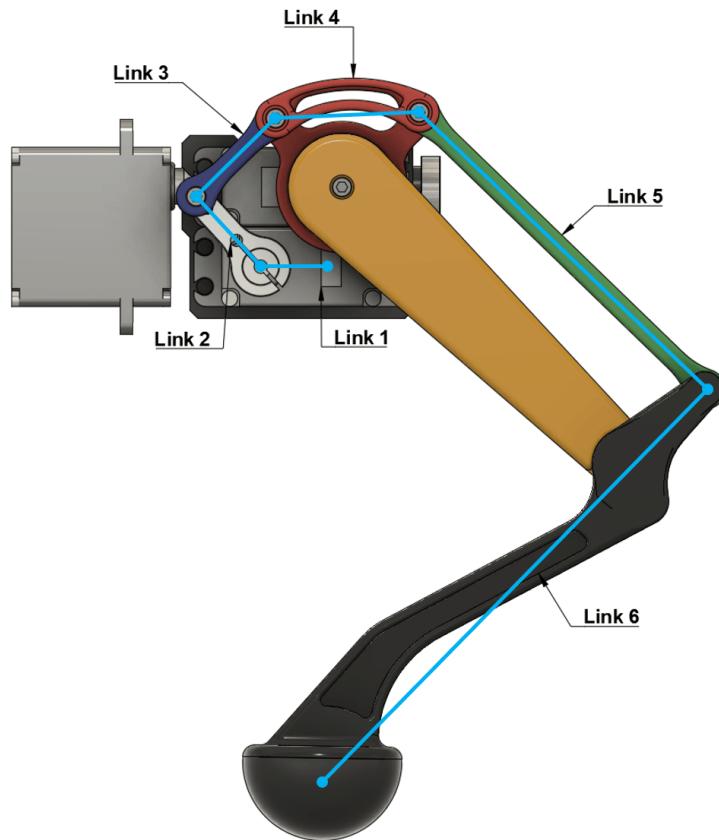


Figure 4.12: Diagram of Leg Linkages

The lower servo motor's rotational movement is transmitted through a series of links and pivots that work in together to control the leg's motion. The first link (labeled 2), directly coupled to the lower servo motor, initiates the movement, which is then modulated by the interconnected bars of the linkage. The coupler (labeled 3), connects to the rotational link (labeled 4) at the femur with a bearing, ensuring the leg moves in the desired trajectory. This is where the precise control over the leg's path is achieved, as the coupler dictates the arc of movement and extent of the leg's sweep.

Link 5, the longest bar in the assembly, extends to the upper femur, translating the rotational motion into the leg's stride. The final link in the system (labeled 6) forms the tibia section of the leg, featuring a curvature that has been optimised to handle the forces exerted during locomotion while maintaining a lightweight profile. Overall, static and dynamic joints are used throughout.

4.3.3 Motor Configuration

Each leg is outfitted with three high-torque motors to ensure the robot has the strength to navigate different terrains and execute complex maneuvers. These motors were selected to operate efficiently, offering a balance between power consumption and response capabilities. This setup guarantees the necessary force and precision for a range of movements, from fine adjustments to powerful strides.

Specification	Detail
Model	BLS-HV45MG
Material	Metal Gear, Aluminum Case
Motor Type	Brushless Servo Motor
Torque	45kg · cm
Dead Band Width	2 μ sec
Control System	PWM Control
Operation Voltage Range	6.0V - 8.4V
Current	4000mA @ 8.4V
Operation Temperature Range	-20°C/60°C
Limit Angle	180°±1
Pulse Travel	500→2500 μ sec
Dimensions	40x20x48.6mm
Weight	75g (without servo horn)

Table 4.3: Key Specifications of BLS-HV45MG Servo

4.3.4 PWM Servo Control

Motor control is achieved through a brushless-DC, motor controller with a Pulse Width Modulation (PWM) signal input, permitting precise control over motor positions with high efficiency. In these controllers, PWM signals are not used to directly vary motor current /torque, but instead the PWM duty cycle is directly interpreted as a position demand to adjust the servo motors' shaft rotation angle. An integral rotary position encoder on the motor provides closed-loop feedback for the precise positioning of each leg. This control method is not only effective for reaching desired positions but also for power management, as the motor controller is also able to modulate power to the servos based on the required movement or holding position, thereby optimising energy usage.

4.3.5 Design Choices

To ensure a smooth and fluid motion at the joint where the femur and tibia meet, two 10mm ball-bearing were incorporated (seen in Figure 4.13a). These ball-bearing, assist in reducing friction and wear at the joint, enabling the leg to articulate smoothly without significant resistance.

Furthermore, a larger 27mm bearing (Figure 4.13b) was embedded within the rotational linkage. This component is necessary for the linkage system that connects the tibia to the robot's servo motor, allowing the entire leg assembly to rotate smoothly. This design configuration enables the replication of the seamless motion observed in natural quadrupeds, ensuring that each step taken by the robot is both stable and efficient.

In addition, the material chosen for the tibia was acrylonitrile styrene acrylate (ASA), due to its strength and temperature resistant properties [28]. ASA was preferable to acrylonitrile butadiene styrene (ABS) because it offers similar mechanical properties but with enhanced UV resistance and improved printability. ASA is less prone to warping and provides a smoother surface finish, making it more suitable for precise and durable parts in the robot's design.

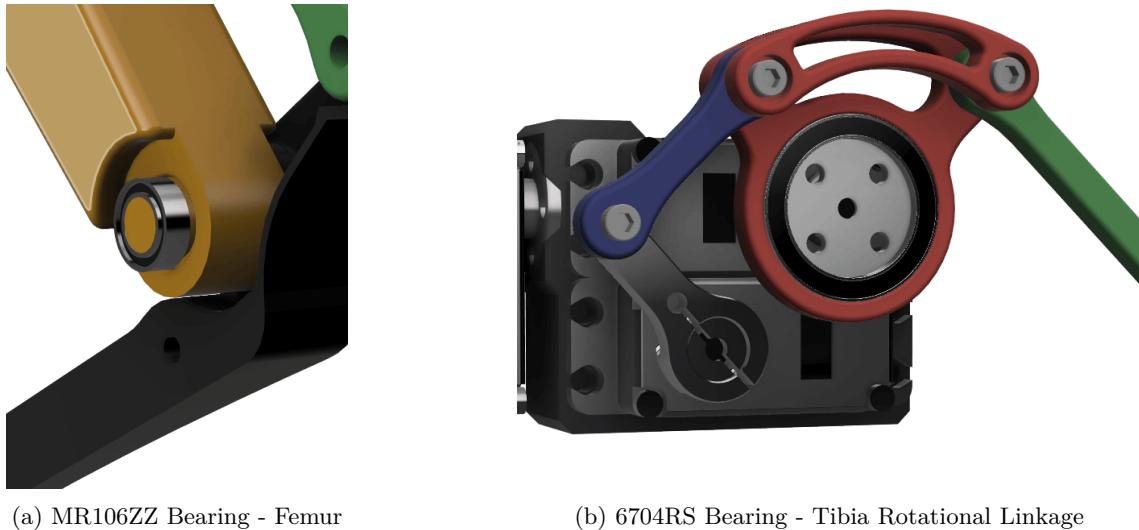


Figure 4.13: Design inclusion of ball bearings

The foot of each leg was produced from a dissected squash-ball to provide a rubbery-friction-grip with some compliance. This choice was motivated by several key properties of squash balls that make them particularly suited for this application. Firstly, the rubber material provides excellent grip, which is essential for stability and traction on various surfaces. Secondly, the inherent elasticity of the squash ball allows for a degree of shock absorption. This feature is beneficial in reducing the impact on the leg and the robot as a whole when walking or running, mimicking the natural cushioning found in biological feet. Lastly, the durability and resistance to wear of squash ball material ensure that the foot can withstand repeated use without significant degradation, making it a practical and long-lasting solution for the robot's design.

In summary, the design of the quadruped leg, with its carefully selected components and materials, reflects what is believed to be a comprehensive approach to replicating natural movement and stability.

4.4 Front-End Sensor Integration

The head of the quadruped robot is a key component, housing all the primary sensors that ensure its efficient operation and interaction with the environment. This section provides an overview of the sensor types and their integration within the robot's head, emphasising modularity and cost-effectiveness.

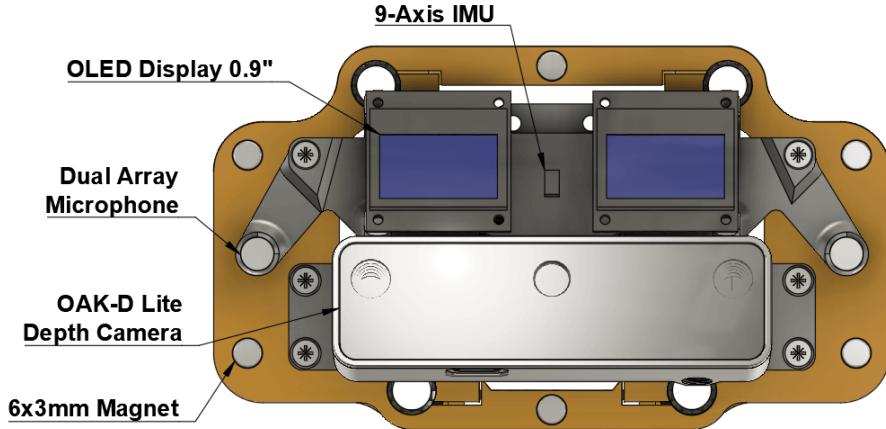


Figure 4.14: Front Sensor Arrangement - CAD Model

4.4.1 9-axis IMU (Inertial Measurement Unit)

The 9-axis IMU function is aid with the robot's orientation and stabilisation. This sensor combines an accelerometer, gyroscope, and magnetometer, providing comprehensive data on the robot's movement and position. By continuously monitoring its spatial orientation, the IMU helps maintain balance, particularly important for a quadruped robot navigating uneven terrain.

4.4.2 Environmental Temperature Sensor

An environmental temperature sensor is included to monitor the ambient temperature. This sensor not only permits the robot's operation to adapt to varying thermal conditions but also ensures the internal components are protected from extreme temperatures. It adds an extra layer of environmental awareness, enabling the robot to operate safely and effectively in diverse settings.

4.4.3 OLED Display

The head incorporates two 0.9" OLED displays. These displays serve a dual purpose:

- As an Information Display: They can show real-time data, status updates, or notifications, making the robot's operation transparent and user-friendly.

- As an Aesthetic Feature: By statically displaying an 'eye' image, these screens contribute to a friendlier and more approachable appearance. This is particularly important for robots intended to assist or interact with people, as a welcoming design can significantly enhance user acceptance and comfort.

4.4.4 Depth Camera (OAK-D LITE)

The front of the robot's head is equipped with a depth camera, specifically the OAK-D LITE. This camera performs depth calculations and is capable of detecting objects or people in the robot's path. It enhances the robot's ability to navigate and interact with its environment, ensuring precise and safe operations.

4.4.5 Dual Microphone Array

A dual microphone array is mounted on the front, enabling the robot to capture and process audio from its surroundings. This feature enables interactive applications, where the robot needs to respond to verbal commands or identify sound sources.

4.4.6 Mounting and Protection

All these components are securely mounted on a core bracket attached to the robot's chassis using M3 screws. This ensures stability of the sensors during operation, whilst also enabling ease-of-maintenance. Additionally, a magnetically attachable faceplate covers the front of the head, providing protection to the sensors. The magnetic attachment not only offers easy access for maintenance but also allows for quick module exchanges, enhancing the modularity of the design.



Figure 4.15: Front Face Cover - CAD Model

4.5 Rear-Section

The rear section of the quadruped robot was designed for power control and management to ensure safe operation. It features several key elements aimed at providing easy control and monitoring of the robot's power systems.

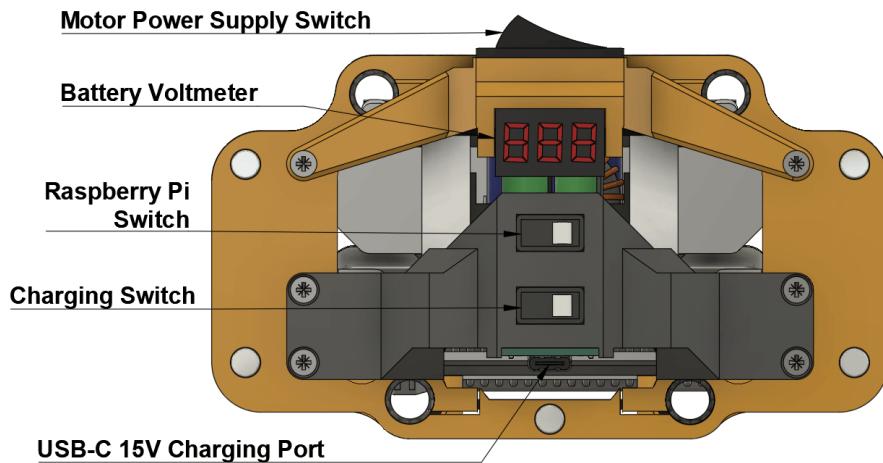


Figure 4.16: Rear-Section - CAD Model

4.5.1 Power Control Switches

There are two small slide switches located on the rear-facing panel:

- **Upper Slide Switch:** Controls power to the onboard Raspberry Pi, allowing for easy shutdown and startup.
- **Lower Slide Switch:** Controls the charging circuitry and activates the digital voltmeter, which monitors the battery level.

Each slide switch is equipped with coloured dots to indicate the on and off positions: white for off and red for on. This visual aid simplifies operation and helps prevent errors in handling the power controls.

4.5.2 USB-C Quick Charge Module

The rear-end panel includes a USB-C quick charge input jack capable of accepting 3A at 15V. This provides a rapid charging system for the robot's battery, meaning that no special charger is required, minimising downtime and enhancing ease of use.

4.5.3 Safety Switch

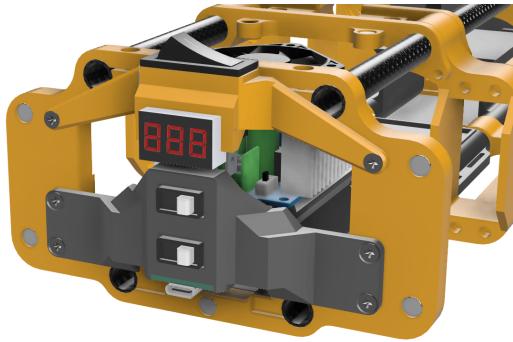
A vertically mounted top switch is included to isolate power to the motors. This switch provides a convenient method to disable the robot's movement, in the case of emergencies or when maintenance is required. The robot's controller and sensors, however, remain enabled.

4.5.4 Mounting and Accessibility

All components in the rear section are fixed with screws, facilitating easy disassembly and maintenance. This design consideration ensures that the components can be quickly accessed and serviced without requiring specialised tools or complex procedures.

4.5.5 Magnetic Rear Cover

The rear section is protected by a cover that magnetically attaches to the chassis. This cover also features coloured dots to indicate the on and off positions of the slide switches, providing a clear and user-friendly interface for managing the robot's power systems.



(a) Rear Section



(b) Rear-Section Panel Cover

Figure 4.17: Rear-Section with and without cover

By integrating these elements, the rear-section of the quadruped robot combines functionality, safety, and ease of maintenance, supporting the overall robustness and reliability of the system.

4.6 Side and Top Sections

The design further incorporates a modular approach for the top and side sections to enhance accessibility and functionality. The top of the robot is divided into three sections, each fitted with a cover panel. This division allows for easy opening and quick access to the internal components for maintenance and upgrades without requiring tools. This magnetic attachment ensures that the covers remain securely in place during operation but can be easily removed when needed.



Figure 4.18: Quadruped with Top Panels and LIDAR Sensor

One of the top panels features a cut-out slot specifically designed to accommodate a LIDAR sensor. This slot ensures the sensor itself is securely housed within the robot while maintaining full functionality and unobstructed view for environmental scanning.

The side panels of the robot have been designed with additional features. These panels include speaker grills that house two 4-ohm, 3W speakers. The integration of these speakers enables the robot to produce audio, which can be used for communication, alerts, or interactive applications.

Due to these side panels housing the speaker components, they have been designed to pivot downward for access, rather than to be totally removable, seen in Figure 4.19. Nevertheless, they still permit more than adequate viewing of and access to the robot's internal structures and electronics.



Figure 4.19: Side Panel with Speaker

The combination of these design features ensured that the quadruped robot is both user-friendly

and easy to maintain. The modular section and panel design enhances the robot's versatility and supports efficient operations by facilitating quick and easy access to the main components.

4.7 Battery Pack Design

4.7.1 Rationale of Li-Ion Cells

In the process of designing the battery pack for the quadruped robot, various battery types were investigated to identify the most suitable option. Lithium-Ion batteries, particularly the *LG MJ1 18650* cells, were chosen due to their highest energy density per volume (as seen in Figure 4.20), making them the best choice for applications where both energy capacity and weight are critical factors. These particular cells are widely recognised for their high capacity and reliability [29].

Each *LG MJ1 18650* cell operates at a nominal voltage of 3.7V with a capacity of 3500mAh. The battery pack consists of six of these cells, arranged in a 3-series/2-parallel configuration. This arrangement was selected to achieve a balance between voltage and capacity, tailored to the selected motor drives, whilst remaining relatively compact to fit within spatial constraints of the quadruped chassis.

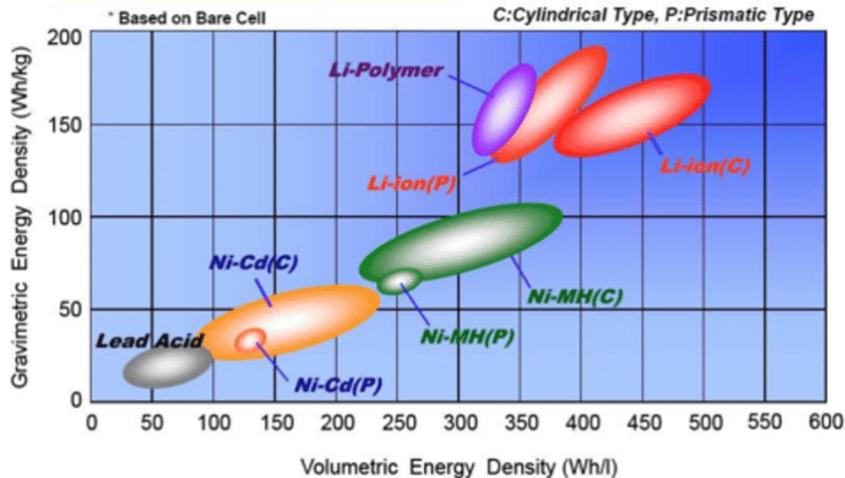


Figure 4.20: Comparison of Energy Density in Battery Cells [30]

Off-the-shelf battery solutions were evaluated; however, they failed to meet the specific size and power requirements for the quadruped. Thus, a custom battery pack was designed, prioritising compactness and power output. The 3S/2P configuration was chosen to provide an optimal voltage and capacity for the robot's systems, particularly the motors and their drives, ensuring sufficient power delivery to all systems without occupying too much space.

4.7.2 Power Calculations

The total output voltage of the battery pack in a series configuration (3S) is calculated by summing the voltage of each cell:

$$V_{\text{total}} = N_{\text{series}} \times V_{\text{cell}} \quad (4.1)$$

where V_{total} is the total output voltage of the battery pack, where N_{series} is the number of cells in series, and V_{cell} is the voltage of each cell. For this battery pack:

$$V_{\text{total}} = 3 \times 3.7V = 11.1V \quad (4.2)$$

The total capacity of the battery pack in a parallel configuration (2P) is calculated by summing the capacity of the cells in parallel:

$$C_{\text{total}} = N_{\text{parallel}} \times C_{\text{cell}} \quad (4.3)$$

where C_{total} is the total capacity of the battery pack, N_{parallel} is the number of cells in parallel, and C_{cell} is the capacity of each cell. For this battery pack:

$$C_{\text{total}} = 2 \times 3500mAh = 7000mAh \text{ or } 7Ah \quad (4.4)$$

The total energy stored in the battery pack can be calculated using the formula:

$$E_{\text{total}} = V_{\text{total}} \times C_{\text{total}} \quad (4.5)$$

where E_{total} is the total energy of the battery pack. Substituting the values we have:

$$E_{\text{total}} = 11.1V \times 7Ah = 77.7Wh \quad (4.6)$$

This calculation indicates the total energy available in the battery pack for powering the quadruped. Assuming the servo motors require an total mean of 10A at 8V (under load), the battery life can be estimated to be approximately 0.97 hours (assuming low losses in the buck-converter). When idle, the current draw may be 25 percent of that, resulting in an estimated battery life of approximately 4.00 hours. The operating time is in-line with many of the other robots that were reviewed.

In practise, whilst the nominal cell voltage is 3.7V, it will vary between 3.2V (discharged) and 4.2V (fully-charged); therefore 3.7V is actually a mean voltage.

4.7.3 Charging Mechanism

The battery pack can be charged using a USB-C 15V, 3A DC charger, with the charging voltage regulated to 12.6V to match the battery pack's requirements ($3 \times 4.2V$). This ensures a safe and efficient charging process, tailored to the lithium-ion cells' charging characteristics. A DC buck converter with constant current (CC) and constant voltage (CV) capabilities is used to step down



Figure 4.21: Battery Pack Design with Integrated Battery Management System

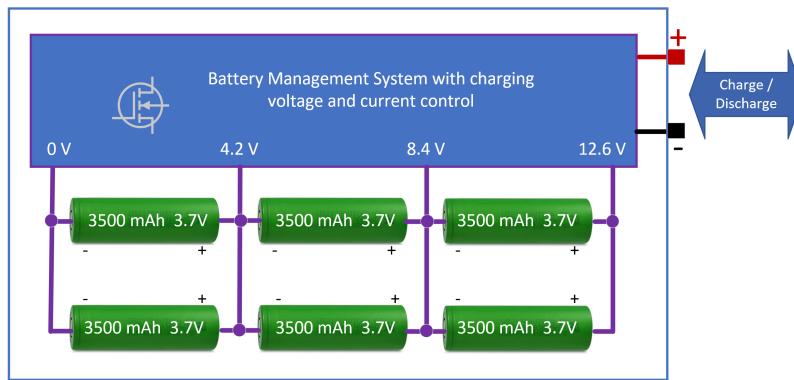


Figure 4.22: Custom, 3S/2P, Battery Pack with Integrated BMS: Nominal voltage: 11.1 V Nominal capacity: 7000 mAh

the voltage appropriately. Additionally, a Battery Management System (BMS), with a 40A current capacity was designed into the battery pack itself (Figure 4.21) to prolong battery life and ensure safe operation. This BMS supports the 10A discharge rate *per cell*, safeguarding against over-current conditions and improving the pack's overall safety and longevity. Furthermore, the BMS handles balanced charging and provides short circuit protection, ensuring each cell is charged evenly and that the battery pack is protected from potential damage from over-current. The arrangement of the cells with the BMS is shown in Figure 4.22.

4.7.4 Design Considerations

The spacers between the cells were designed to be produced from ASA filament, chosen for its thermal resistance. ASA's ability to withstand high temperatures ensured that the spacers remained structurally stable, preventing deformation that could compromise the battery pack's integrity. The spacers were also designed with grooves to guide the nickel strips, ensuring precise and secure placement.

Additionally, the cells were interconnected using nickel strips, which were affixed using a spot welder. This method ensures a robust and reliable connection, essential for both electrical conductivity and safety. Spot welding minimised the risk of heat damage to the cells, a common concern

with other types of soldering.

The battery pack was specifically sized to fit exactly into the chassis of the quadruped. It was designed to be placed in the center of the chassis; since this is a heavy component, it helps to distribute the weight evenly, contributing to the overall stability and balance of the quadruped.

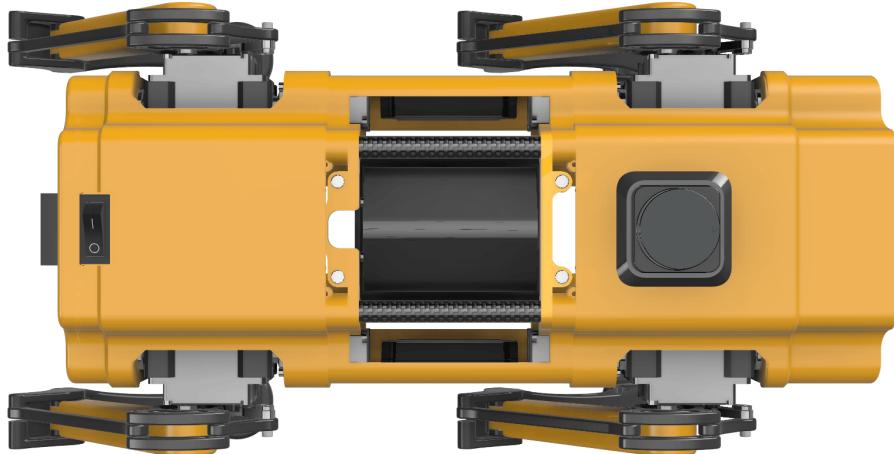


Figure 4.23: Battery Pack Centre Placement in Main Assembly

4.8 Considerations for Productionisation

Whilst AM has been selected to provide an efficient manufacturing method for the development and prototyping stages, it is likely that productionisation of the design will involve a review of many of the selected materials. For example, the use of carbon-fibre composites may be carried over to a wider range of parts, particularly the limb components that require a combination of strength with low mass to minimise inertia. As an alternative for lower-cost, glass-fibre-reinforced thermoset plastics could be used with molds for each part, if the production quantities justify the tooling cost. An intermediate route could be the use of machined-alloy parts, anodised for corrosion resistance.

Any productionisation would likely follow an extensive trial period of the present prototype to evaluate its strengths and weaknesses as well as determining other required design revisions or improvements. Indeed, it may be that AM techniques can continue to be used, particularly for low volumes. Whilst home/lab based 3D printing is generally limited to the use of a few varieties of thermoplastics, industrial, AM, machines can work with almost any materials using processes such as Laser Powder Bed Fusion, which produces solid parts from a variety of powdered metal alloys. In fact, recent work [31][32][33] has demonstrated the production of electric machine components, including, soft magnetic parts, rare-earth magnets and copper coils using this process. (Copper is a very difficult material to process with AM due to its high reflectivity and high conductivity.)

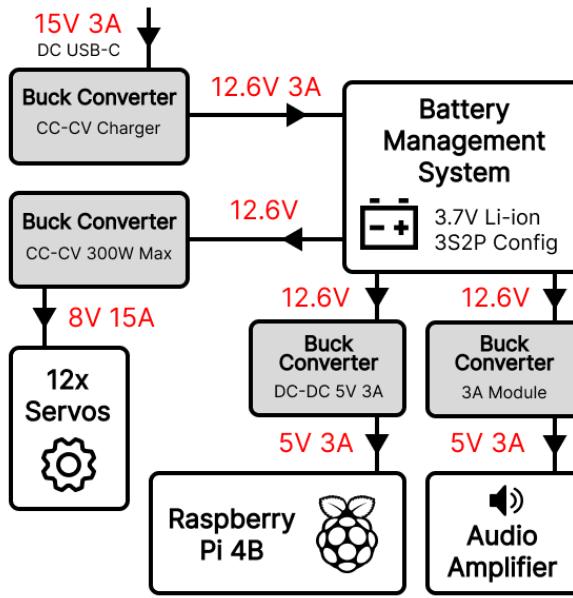


Figure 4.24: Electrical Subsystem

4.9 Electrical Design

4.9.1 Power Distribution Overview

This section describes the design and functionality of the power distribution system for the autonomous quadruped robot. It includes a detailed explanation of how power is managed and distributed to various components, such as the motors and control systems.

The system diagram, Figure 4.24, illustrates the complete power distribution network within the quadruped robot, highlighting the flow of electrical energy from the battery pack to all critical components. This includes the battery, motors, sensors, audio amplifiers and on-board computer.

4.9.2 Battery Charging Circuit

The battery charging circuit is a key component of the quadruped robot, ensuring the battery pack is charged safely and efficiently. The system was designed to take a 15V, 3A input from a DC power supply, which is fed using a USB-C connector. This input power is then regulated and controlled to charge the battery pack appropriately.

The charging process begins with the input power being fed into a dual-mode, constant-current (CC) and constant-voltage (CV) buck converter. This converter regulates the charging voltage to 12.9V and the charging current to 3A. The regulated voltage and current are calculated to match the requirements of the designed 3S/2P battery configuration. Given that each cell in the battery pack has a capacity of 3500mAh, the charging rate can be calculated as follows:

$$C_{\text{rate}} = \frac{I_{\text{charge}}}{N_{\text{parallel}} \times C_{\text{cell}}} \quad (4.7)$$

where $I_{\text{charge}} = 3A$, $N_{\text{parallel}} = 2$, and $C_{\text{cell}} = 3500mAh$.

$$C_{\text{rate}} = \frac{3A}{2 \times 3.5A} \approx 0.43C \quad (4.8)$$

This 3A charging rate ensures that the battery pack is charged at approximately 0.43C, which is within the safe charging limits for the selected lithium-ion cells, promoting battery longevity and safety. Charging at a lower C rate extends the battery's lifecycle, as seen in Figure 4.25, leading to a longer overall lifespan and less frequent replacements. This practice reduces waste and promotes a more eco-friendly approach by minimising the environmental impact associated with both battery disposal and manufacturing. This information is supported by findings from a paper discussing methodologies of increasing electric vehicle range [34], in which Li-ion cells are exclusively used.

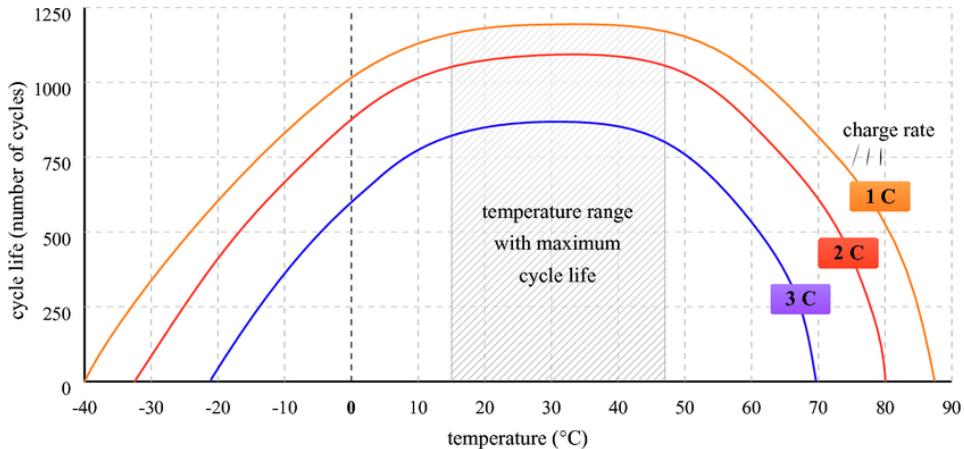


Figure 4.25: Cycle Life of Different Charge Rates in Lithium-Ion Cells [34]

After the buck converter, the charging current passes through a Schottky diode, which reduces the charging voltage by a further 0.3V to the required 12.6V. The purpose of this diode is actually to prevent any reverse current from the battery flowing back to the charging circuit, which could potentially damage the charging components or cause a safety issue. The circuit arrangement of the buck-converter, as discussed later in relation to that powering the motor drives, would permit a reverse-current flow, without this additional diode.

Following the Schottky diode, a switch is placed in the circuit. This switch serves two functions: it allows for the disconnection of the charging circuit, and it enables the voltmeter connected across the battery. By toggling this switch, the user can safely cut off the charging process and also deactivate the voltmeter when it is not needed.

The design of the battery charging circuit, together with the battery's built-in BMS, ensures a safe, efficient, and user-friendly charging process for the quadruped robot's battery pack.

The overall arrangement is depicted in Figure 4.26, illustrating the flow of power from the DC

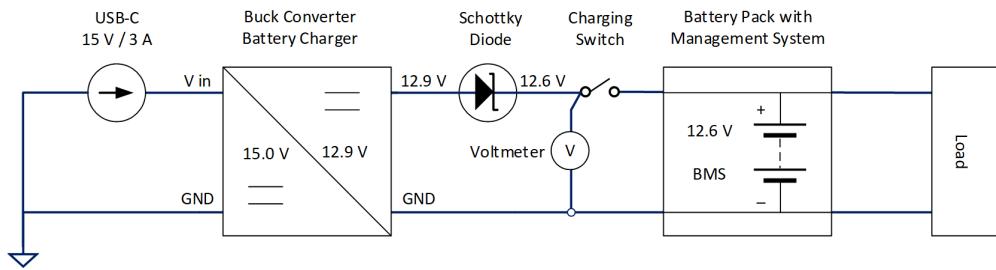


Figure 4.26: Schematic Diagram for the Charging Circuit

supply input, through the buck converter, Schottky diode and switch, and ultimately reaching the battery pack, with its integrated BMS. The careful design considerations and components used in this battery sub-system are critical to maintaining the performance and safety of the quadruped robot.

4.9.3 Motor Power Circuit

Selection of DC-DC Converter

The motors and their drives require an operating voltage of 8V and consequently it is necessary to step down the battery voltage.

A synchronous buck converter (as shown in Figure 4.27) was selected for its higher efficiency, compared to the simple version (Figure 4.28)

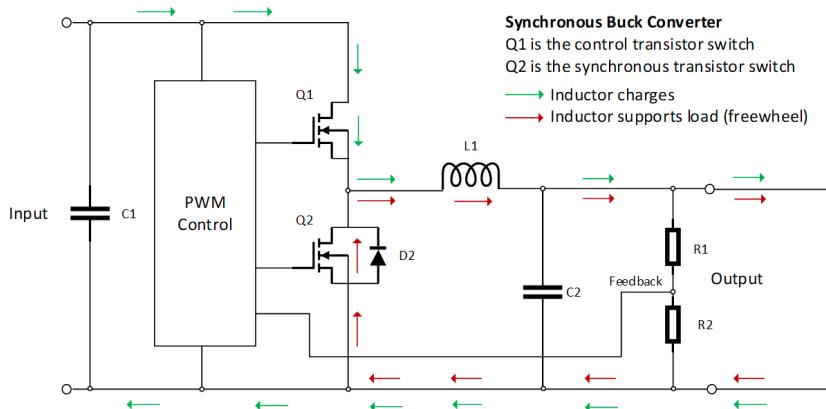


Figure 4.27: Synchronous Buck Converter Topology

The addition of a second switch, Q_2 , allows the efficiency of the buck converter to be improved through synchronous rectification. In a simple buck converter arrangement (Figure 4.28), the diode (D_2) would start to conduct once its forward voltage rises above its junction voltage. This voltage drop, however, reduces the efficiency of the converter. In the synchronous converter, Q_2 can be turned-on during the freewheeling cycle, bypassing the diode and its voltage drop. Q_2 , of course, is selected to have a very low $R_{DS(on)}$ resistance.

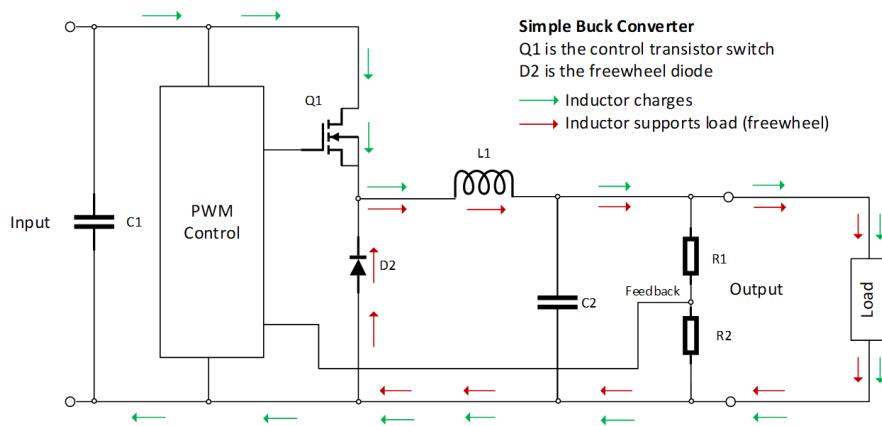


Figure 4.28: Simple Buck Converter Topology

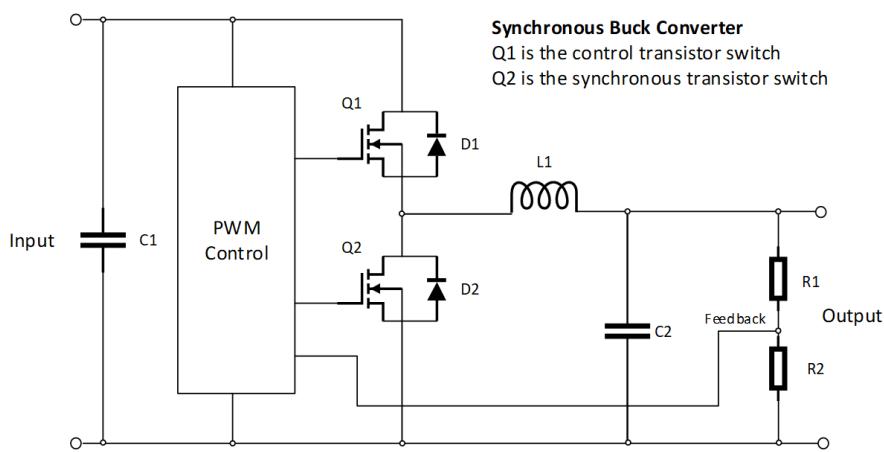


Figure 4.29: Schematic of the 300 W, Synchronous Buck Converter

There was very limited choice in selecting a synchronous buck converter, with the required power capacity, from readily available COTS products. The selected model is rated for 300W and can operate in either constant-current (CC) or constant-voltage (CV) mode with a claimed efficiency of up to 96%, using two 75V/80A power MOSFETs. All MOSFETs intrinsically contain a so-called ‘body-diode’ in parallel with the channel of the device, shown in Figure 4.29 as D1 and D2. Therefore, a separate anti-parallel, freewheeling, diode is not actually required [35]. This means, however, that through the body-diode , D1, there is a possibility, in certain circumstances, for power to reverse from output to input (as the case in the battery-charging buck-converter).

Motor Power Circuit Design

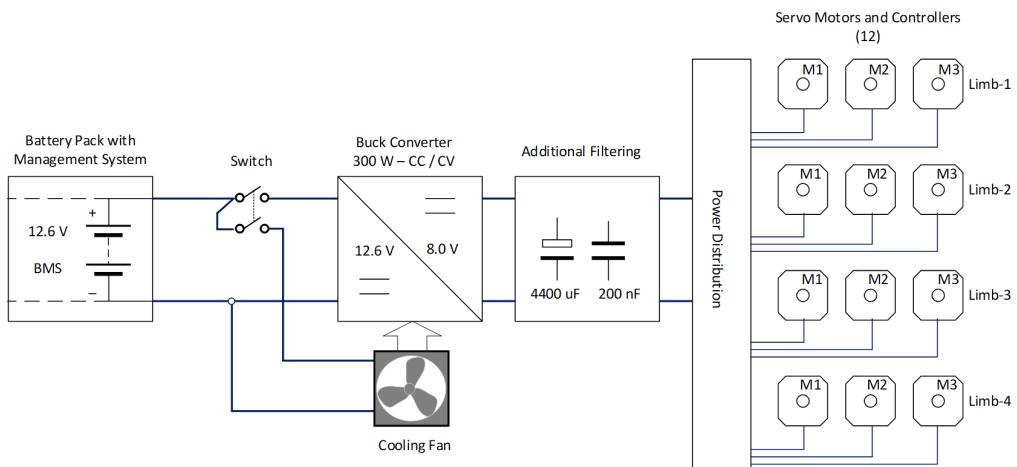


Figure 4.30: Schematic Diagram of the Motor Power Circuit

The servo controllers and 12 motors are the largest power consumers on the robot. It is therefore important to efficiently manage and distribute power from the battery to the servo motors, ensuring reliable and stable operation under various loads. The circuit schematic, in Figure 4.30, shows the connections between the 3 main elements: the battery-pack, the DC-DC, buck converter, and the motor drives.

The input of the buck-converter operates over the battery’s voltage range of 9.6V - 12.6 V and steps this voltage down to 8V, the motors’ nominal operating voltage, while featuring a current limiter that can be adjusted as needed for development. The selected buck converter is equipped with an integrated enable/disable switch, allowing it to be turned on or off internally. Nevertheless, a separate power switch was included in the design and this also powers the 40mm cooling fan for this converter.

It was decided to mount a cooling fan above the two heatsinks of the buck converter, with airflow in the direction of the fins. The fan was deemed necessary for cooling the power MOSFETs, ensuring they remain at optimal operating temperatures even under high load conditions. The manufacturer’s datasheet gives an efficiency figure of 86% to 96% for the converter, depending on load, meaning an estimated worse case loss of 20W. A full thermal analysis had initially not been carried out, basing the initial design on simple calculations. However, a combined 3D thermal/CFD finite-element analysis was eventually carried out for the worst-case condition and with the fan

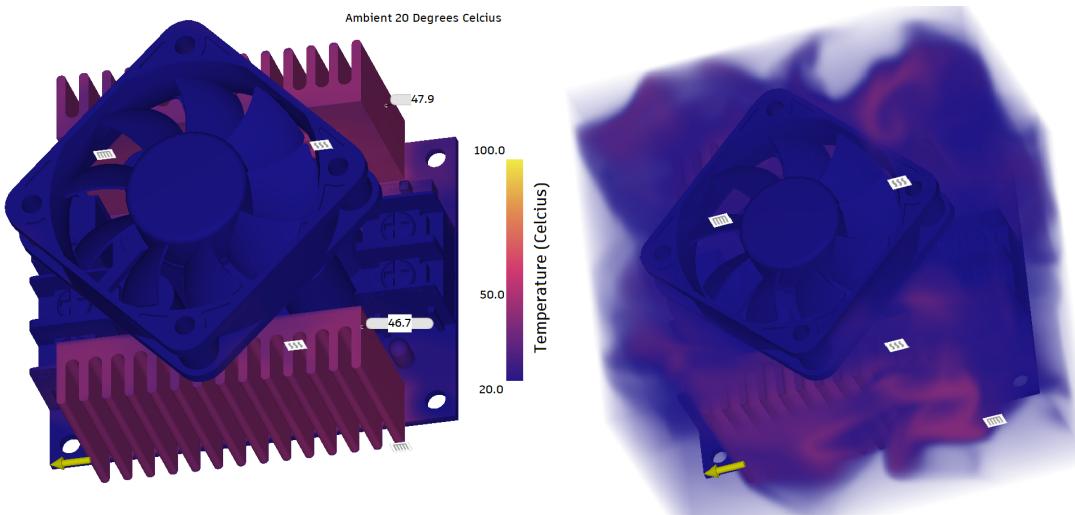


Figure 4.31: CFD & Thermal Analysis of Buck Converter (left is static, right with CFD overlay)

airflow at 5 CFM. This analysis confirmed the validity of the design, showing heatsink temperatures of around 50°C (ambient 20°C).

In practice, the maximum current draw from the motors and drives is expected to be 15A, and at 8V, this equates to a load power of 120 Watts, implying adequate headroom from the 300W rated capacity of the converter. Arguably, to reduce power consumption, the fan speed should probably be modulated according to measured heatsink temperature and this would be a future consideration. The diagram (Figure 4.31) shows a side-by-side comparison of the thermal and airflow analyses, based on the above operating parameters, illustrating the effectiveness of the cooling strategy.

From the buck converter, the regulated power is fed into a custom distribution board. This board is responsible for distributing the 8V supply to each of the servo motors. To maintain stable power delivery and handle transient current surges, the design included two large 2200 μ F, 16V capacitors. These audio-grade, low ESR, capacitors act as local power reservoirs, smoothing out current fluctuations and providing a reserve of energy during transient peak demands.

Additionally, the circuit includes a couple of 100nF, de-coupling, capacitors which were added following initial testing, to filter out high-frequency noise. These capacitors helped to address electrical noise that was finding its way to the on-board computer and, potentially, to other sensitive components.

The combination of the buck converter, cooling mechanisms, custom distribution board, and filtering help ensure that the design of the motor power circuit would be robust, efficient, and capable of handling the dynamic power loads of the quadruped robot.

4.10 Software Design

The software design of the autonomous quadruped robot was fundamental to its functionality, enabling its operational capabilities, autonomy, and interactive features. This section offers a

comprehensive overview of the software architecture, the guiding design principles, and the essential components involved.

4.10.1 Design Principles

The software design was guided by several fundamental principles aimed at ensuring reliability, scalability, and efficiency:

- **Modularity:** The software architecture was divided into distinct modules, each responsible for specific functionalities. This modularity enhances maintainability and allowed for independent updates and testing of each module.
- **Scalability:** The design supported scalability, enabling the integration of additional sensors and modules without significant modifications to the core system.
- **Real-time Processing:** The software was designed to handle real-time data processing, essential for navigation, obstacle detection, and decision-making.
- **Interoperability:** Leveraging ROS2 (Robot Operating System 2) ensures interoperability between different software components and facilitated integration with various hardware platforms.

4.10.2 Software Architecture

The software architecture integrated several subsystems that worked cohesively to achieve the robot's operational objectives. The primary subsystems included navigation, vision, control, and communication. This architecture relied on several key components:

- **Raspberry Pi 4B:** The Raspberry Pi 4B (8GB model) is a powerful single-board computer that serves as the central computing unit for the robot. It provides sufficient processing power and interfaces for the various necessary sensors and peripherals.
- **Ubuntu 22.04 LTS:** Ubuntu 22.04 Long-Term Support (LTS) was chosen as the operating system due to its stability, security, and support for a wide range of software packages developed for robotic applications.
- **ROS2 Humble:** ROS2 Humble Hawksbill was selected for its long-term support and advanced features, including improved real-time performance (compared to previous versions), enhanced security, and better support for complex robotic systems.

4.10.3 Navigation Subsystem

The navigation subsystem is responsible for mapping the environment, planning paths, and executing navigation commands. Key components included:

- **LIDAR Scanner and Cartographer Mapper:** The LIDAR scanner collects environmental data, which is processed by the ROS2 package, Cartographer, to create detailed maps. This data is essential for understanding the robot's surroundings and planning safe paths.
- **Nav2 Navigation Stack:** Built on ROS2, the Nav2 stack handles path planning and obstacle avoidance. It utilises the maps created by the Cartographer Mapper to generate safe and efficient routes for the robot. The stack integrates various algorithms and plugins to handle tasks such as global and local path planning, recovery behaviors, and environmental perception, ensuring the robot can navigate complex environments autonomously.

4.10.4 Vision Subsystem

The vision subsystem enables the robot to perceive and understand its environment. Its design consisted of:

- **Depth Camera:** Captures depth information, providing a 3D view of the surroundings.
- **MobileNet Neural Network:** Processes the images from the depth camera to detect objects and people. This neural network was optimised for real-time processing, ensuring both accurate and quick detections.
- **Human Localisation:** Upon detections from the neural network, depth information from the image is used to approximate the relative position of discovered entities.

4.10.5 Control Subsystem

The control subsystem integrates inputs from various sensors and modules to manage the robot's movements and actions. It was designed to include:

- **Central Control System:** Received data from the navigation and vision subsystems is evaluated to make real-time decisions to control the robot's movements.
- **REX Program:** Acts as the decision-making hub, processing inputs and executing commands based on predefined algorithms and real-time data.
- **Movement Process:** Handles the locomotion of the quadruped, including inverse kinematics, gait patterns, and cycles.

4.10.6 Communication Subsystem

Effective communication was deemed vital for interactive functionalities and remote control. The communication subsystem includes:

- **IBM WatsonX Assistant:** Provides a natural language interface for voice commands and conversation, enabling users to interact with the robot seamlessly.

- **ElevenLabs Text-to-Speech (TTS):** Deliveres high-quality voice responses, enhancing user interaction with natural and expressive speech synthesis.
- **Onboard Microphone and Speakers:** Facilitates voice input and output, allowing the robot to 'listen' to user commands and respond audibly.
- **Web Application:** Enables remote monitoring and control of the robot, providing a user-friendly interface for interaction and management.

5

Hardware Implementation

Contents

5.1	Introduction	55
5.2	Chassis and Main Assemblies	56
5.2.1	Chassis Construction	56
5.2.2	Motor Mounting	58
5.2.3	Leg Mechanism Construction	60
5.2.4	Front-End Sensor Assembly	62
5.2.5	Rear Panel Assembly	70
5.3	Power and Drive System Implementation	72
5.3.1	Battery Pack Assembly and Charging	72
5.3.2	Power Delivery and Motor Control	74
5.4	Bill of Materials (BOM) and Expenditure	80

5.1 Introduction

The implementation phase of this project involved translating the concepts, CAD designs and plans into a functional autonomous quadruped robot. This chapter provides a detailed account of the processes, techniques, and methodologies employed to build the robot, from the assembly of the hardware components to the integration of the software systems.

The primary objective of this chapter is to document the step-by-step construction and configuration of the robot, ensuring that each component and subsystem is described in full. This covers the selection of materials, manufacturing methods, assembly procedures, and the installation of various sensors and control systems.

Key aspects covered in this chapter include:

- **Hardware Implementation:** Detailed account of the construction of the robot's chassis, the design and assembly of its legs, and the integration of sensors and actuators.
- **Electrical Implementation:** Description of the power distribution system, battery management, motor control circuitry, and safety features.

- **Testing and Debugging:** Discussion of the various testing methodologies applied to ensure the robot's functionality and reliability, along with the challenges encountered and solutions devised.

5.2 Chassis and Main Assemblies



Figure 5.1: Full Construction of the Quadruped

5.2.1 Chassis Construction



Figure 5.2: Prototypes of minimal chassis assembly and modular coxa motor frame.

Materials Used

The chassis was constructed, according to plan, using a combination of carbon-fibre-composite tubes and 3D-printed PLA+ components. Carbon-fibre-composite had been chosen for its high strength-to-weight ratio, providing a robust yet lightweight structure. PLA+ was selected for its durability and flexibility, making it suitable for various structural components.

Item	Details	Quantity
Carbon Fiber Tubes	10x8x300mm	4
3D Filament	Yellow PLA+	1 kg
3D Filament	Black PLA+	1 kg
M3/M2.5 Screws	Hex 6mm	24
Threaded Inserts	M3/M2.5 Size	46

Table 5.1: Materials for Chassis

Manufacturing Techniques

Additive Manufacturing (3D printing) was used for many of the robot's structural components, enabling rapid prototyping and efficient design testing. The carbon-fibre-composite tubes, originally 500mm in length, were cut to 300mm using a wet saw with a diamond blade. These tubes were then assembled with braces to form the main structural chassis of the robot. The screw fittings, within the end-braces, clamp onto the carbon-fibre tubes, providing a firm hold that prevents slippage or movement while the robot is in action. This clamping method not only reinforces the integrity of the structure but also simplifies the process of part replacement and repair. Should a component need to be replaced, the operation can be effected quickly without the potential mess or waiting time associated with adhesives.



(a) Heated Brass-Inserts



(b) Chassis Components

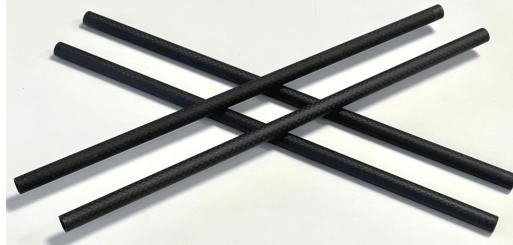
Figure 5.3: Brass-Inserts and Side Components of Chassis

Assembly Process

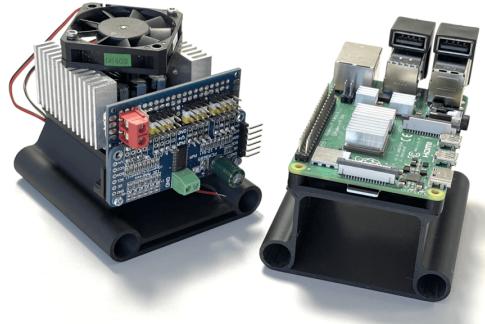
The assembly of the chassis involved several steps:

1. **Frame Construction:** The carbon-fibre-composite tubes were cut to the required lengths and assembled using braces and compression fittings to form the main frame.
2. **Component Mounting:** PLA+ components, including side panels and mounting brackets, were 3D printed and attached to the frame using standard M3 screws and brass-threaded insert fittings, as seen in Figure 5.3a.

3. Cable Management: Dedicated channels for cable management were integrated into the design to ensure clean and secure routing of power and data cables.



(a) 300mm Carbon-Fibre Tubes



(b) Front and Rear Chassis Base

Figure 5.4: Carbon-Fibre Tubes and Base Panels

Prototyping

The end-braces of the chassis underwent several design iterations, each improving upon the previous. The first design established the basic fit, with no clamping mechanism. The second iteration improved on this, ensuring a better fit with the carbon-fibre tubes, through the use of a compression fitting. By the third iteration, extra mounting screw holes were added for future features. Finally, the fourth iteration included magnets for a magnetic top cover panel that was planned to be added later. These changes were focused on enhancing the structural integrity, ease of assembly, and versatility for future upgrades. The design progression can be seen in Figure 5.5a.



(a) Prototypes of End-Braces



(b) Prototypes of Different Clamping Tolerances

Figure 5.5: Prototypes of carbon-fibre tube end-braces and clamps with different tolerances.

5.2.2 Motor Mounting

Nylon was selected as the material for the motor bracket due to its high temperature resistance, ensuring minimal deformation while under load. However, working with nylon presented challenges. Its hygroscopic properties lead to issues during the 3D printing process, affecting both the strength

Item	Details	Quantity
Nylon Filament	500g spool	1
Flange Ball Bearing	F696ZZ	4
Threaded Inserts	M3 Size	40
Screws	M3 6mm Hex	36
Servo Motors	45kg · cm BLS-HV45MG	12
Aluminium Servo Disk Horns	20mm	4

Table 5.2: Materials for Motor Bracket and Assembly

and finish of the printed parts. The nylon filament was fully saturated with ambient humidity, so it was left to dry at 60°C for 24 hours before printing. This preparatory step was essential to ensure that the print quality met the standards required for the precision components of the robot.



Figure 5.6: Nylon Motor Frame with Motors Fitted

One end of the casing is firmly screwed into the hip motors, helping to anchor them securely in place. The opposite end features a compression fitting designed to accommodate a flanged bearing. This bearing fits into the end-brace of the chassis, enabling the motors to rotate smoothly and without resistance. Figure 5.7 shows the assembled chassis with motors.



Figure 5.7: Assembled Chassis with Motors

Item	Details	Quantity
M3 Screws	6mm	16
M3 Screws	12mm	4
M3 Screws	10mm	16
Threaded Inserts	M3 Size	36
ASA Filament	Black	1 kg
PLA+ Filament	Yellow	1 kg
Bearings	MR106ZZ	8
Bearings	6704RS	4
Aluminium Servo Discs	20mm	4
Aluminium Servo Arms	Standard	4
Squash Balls	Cut in half	2

Table 5.3: Materials for the Leg Assembly

5.2.3 Leg Mechanism Construction

The implementation phase for the leg mechanism involved several revisions to address initial design challenges. One significant issue was a breakage at the tibia linkage joint, as shown in Figure 5.9. To resolve this, the material was changed from PLA+ to ASA, providing greater strength and temperature resistance. The design was further reinforced, and 3D print settings were optimised with higher infill and additional wall layers to enhance durability.



Figure 5.8: Constructed and Mounted Leg

Design Adjustments and Material Changes

To address the challenges faced with the initial material, the tibia was redesigned using ASA, known for its superior mechanical properties. This material change, combined with increased infill and additional wall layers during the 3D printing process, significantly enhanced the strength and durability of the part.

The assembly of the leg mechanisms generally proceeded smoothly. An earlier prototype, shown in Figure 5.10a, was used for testing the fitment of bearings, ensuring proper alignment and

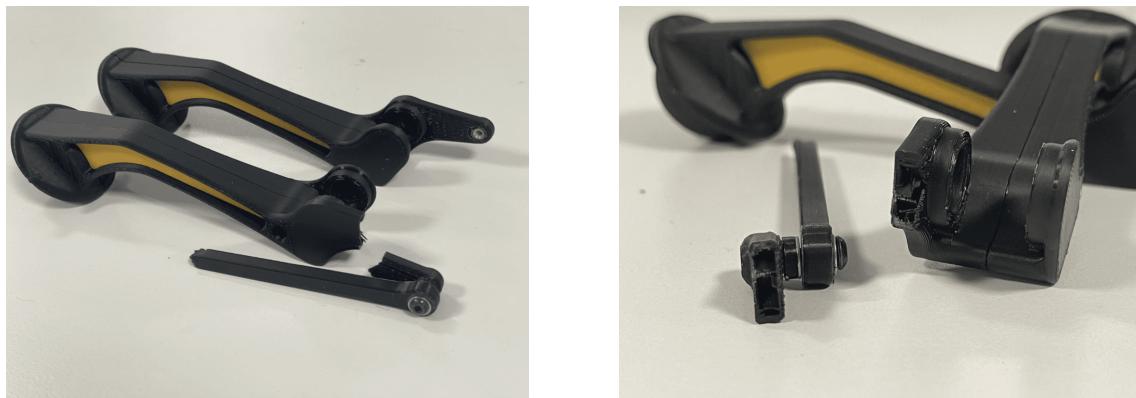
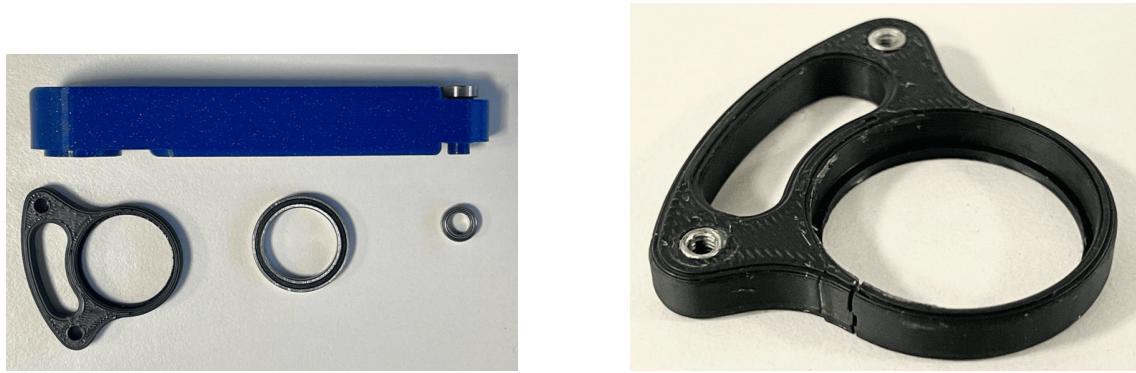


Figure 5.9: Tibia Break at Linkage Joint

functionality. Additionally, Loctite 243 thread-locker, was applied to the linkage screws, providing added security in preventing them from loosening during operation.



(a) Early Femur Prototype

(b) Old Shoulder Design Featuring a Fracture

Figure 5.10: Components for the Femur and Rotational Joint

During testing, the rotational shoulder joint broke due to unintentional motor movements, revealing a weakness in the design. To address this, the annulus region of the part was redesigned with a 1mm increase in wall-thickness. Figure 5.10b shows a crack in the plastic, which loosened all the linkages and significantly impacted the walking gait. After redesigning the part, an FEA was conducted to compare the stress levels on the new design against the old one. The model applied a rotational force of 30N centred on the bearing centre with the rod location point being fixed. The analysis confirmed that the stress in the region of breakage had been reduced from 0.022 MPa in the old design to 0.006 MPa in the new one, an improvement of over 3-fold.

Final Integration and Testing

The revised leg mechanism for the tibia, now utilising ASA plastic and optimised for durability, was integrated into the overall assembly (Figure 5.12). Extensive testing was conducted to ensure that the modifications addressed the initial breakage issues and that the leg mechanisms operated smoothly under various conditions. This testing phase was crucial for validating the design changes and ensuring the robot's overall reliability and performance. Testing included:

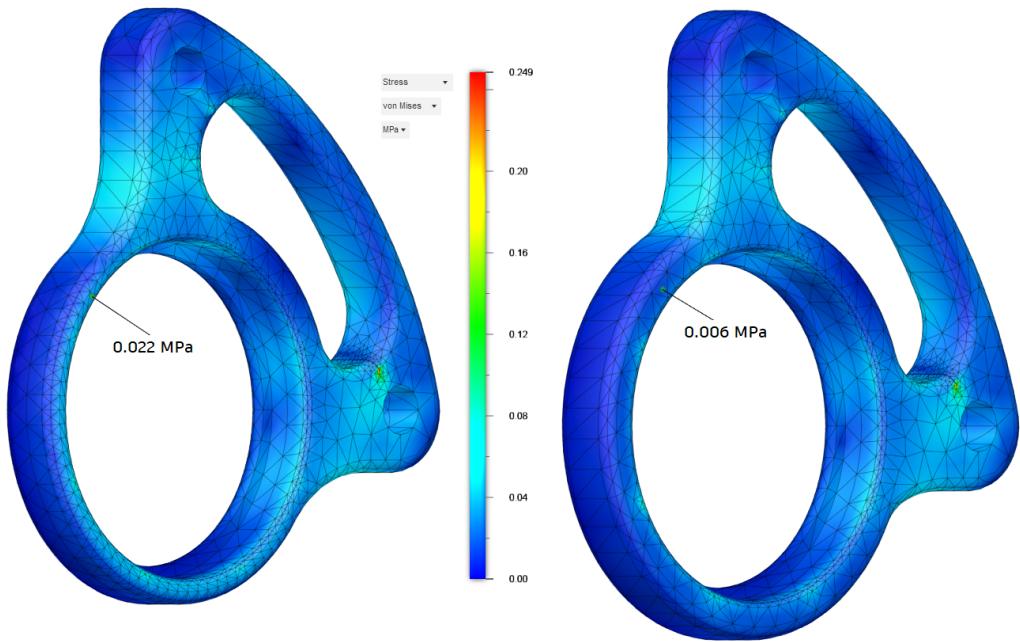


Figure 5.11: Stress Analysis of Redesigned Shoulder Joint (left old, right new)

1. Controlled impacts to simulate accidental bumps and falls, verifying that the structure could absorb shocks without compromising functionality.
2. Incremental loads up to 3kg to simulate various walking scenarios and ensure that the tibia could withstand the stresses without deformation or failure.



Figure 5.12: Stronger Tibia Joint (Double-Sided)

5.2.4 Front-End Sensor Assembly

The building of front-end sensor assembly of the quadruped robot primarily involved the integration of various sensors and components within the head, focusing on secure mounting and reliable

Item	Details	Quantity
M3 Screws	6mm	8
M2.5 Screws	6mm	8
M2.5 Threaded Inserts	Standard	8
PLA+ Filament	Black	1 kg
PLA+ Filament	Yellow	1 kg
OAK-D Lite	Depth Camera	1
LD19 D300	LIDAR Scanner	1
Stereo Microphone Array	USB Module	1
9-Axis IMU	I ² C Interface	1
OLED Displays	0.9", I ² C Interface	2
Temperature Sensor	I ² C Interface	1
USB C Cable	20cm	1
Neodymium Magnets	6x3mm	12

Table 5.4: Materials for Sensor and Component Assembly

functionality. This section details the assembly process, along with the modifications made to accommodate changes in component specifications.



Figure 5.13: Front Face and Sensor Arrangement

Assembly Overview

The assembly process for the front-end sensors included mounting the sensors, ensuring proper alignment and connectivity, while securing the components within the head of the robot. The primary sensors include a 9-axis IMU (inertial measurement unit), an environmental temperature sensor, two OLED displays, a depth camera, and a dual microphone array. This can be seen on the right-side image in Figure 5.13.

Depth Camera Installation and Redesign

Initially, the robot was designed to use an Intel RealSense D435i depth camera, as seen in Figure 5.14. However, a late decision was made to switch to the OAK-D LITE camera, primarily due to API / driver software incompatibility of the Intel camera, which took a significant amount of time and effort to try to resolve (eventually unsatisfactory). This change necessitated a redesign

of the front face and mounts to accommodate the new camera model. The OAK-D LITE was securely mounted, ensuring a clear field of view and stable operation.



Figure 5.14: Previous Iteration with Intel RealSense D435i

OLED Displays and I²C Address Conflict Resolution

The head incorporates two 0.9" OLED displays, which serve both informational and aesthetic purposes. These displays operate using the I²C communication protocol, which allows multiple devices to communicate over the same bus using unique addresses. However, both displays were configured to the same I²C address by default, leading to a conflict.

To resolve this, it was necessary to solder a configuration-resistor to change the address of one of the displays, thereby ensuring that each display had a unique address. This modification allowed both OLED displays to operate simultaneously without any communication issues. The displays were then securely fitted within the head.

LD19 D300 LIDAR Integration

An LD19 D300 LIDAR sensor was mounted on the top of the robot, near the head. This sensor is needed for advanced environmental scanning and object detection. To facilitate easy access to the robot's internals, particularly near the on-board computer, a custom bracket was designed for the LIDAR. This bracket allows the sensor to be securely screwed in place while also enabling it to be flipped up and out of the way when necessary. This can be seen in Figure 5.15.

Other Sensor Integrations

The environmental temperature sensor was mounted to monitor ambient conditions, ensuring the robot can adapt to varying thermal environments and protect internal components from extreme temperatures.

The 9-axis IMU, essential for orientation and stabilisation, was placed in the centre of the head. This sensor combines an accelerometer, gyroscope, and magnetometer, providing comprehensive data on the robot's movement and position.



Figure 5.15: Lidar with Flip Mount

The dual microphone array was also mounted on the front of the head, positioned to capture audio effectively from the surroundings. This setup is used for interactive applications, enabling the robot to respond to verbal commands and identify sound sources accurately.

Cable Management and Connectivity

A short 90-degree USB-C cable was used to connect the depth camera to the Raspberry Pi. This cable fits snugly under the camera and is neatly routed to avoid any interference with other components. Power bus wiring was routed on the opposite side to minimise cross-talk interference.

Final Assembly and Cover Installation

Once all sensors and components were securely mounted and connected, the core bracket was attached to the robot's chassis using M3 screws. This secured the entire assembly.

A magnetically attachable faceplate was then added to cover the front of the head, providing protection to the sensors while allowing easy access for maintenance and quick module exchanges.

The assembly of the front-end sensors was completed successfully, with all components securely mounted and properly connected. The modifications made during the assembly process, such as changing the camera model and resolving the I²C address conflict, were the only issues encountered at this stage. Extensive testing was subsequently conducted to ensure the functionality and reliability of the sensor integration.

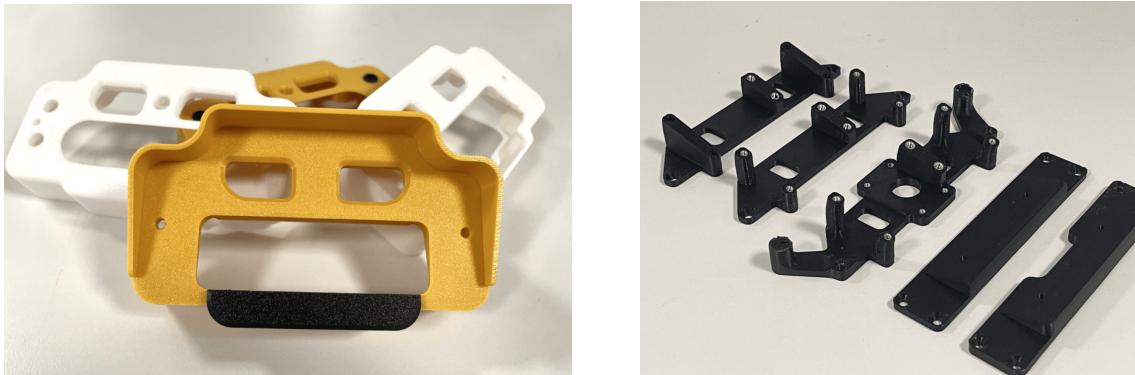


Figure 5.16: Prototypes of Front Face Covers and Sensor Mounts

Issue discovered on further testing

All the components worked as expected apart from an unfortunate issue arising from an earlier design decision. The small, Neodymium, button magnets, used to attach the outer body panels to the chassis were found to have sufficient magnetic leakage to affect the readings of the magnetometer built in to the IMU unit. This prompted a review of not only the button magnets but also of other sources of possible magnetic interference such as the motors and DC power-bus conductors.

Magnetometer Sensitivity

A magnetometer was part of a design decision to improve the navigational accuracy of the quadruped. A GPS-based system remains an option, but due to its differential nature, the coordinates of the direction of motion could then only be established after the robot had travelled a finite distance.

The earth's magnetic field varies according to latitude and altitude and, as an example, at the location of Imperial College, it has been measured by the National Geomagnetic Service as having a vertical component of 45 μT downwards with a horizontal component of 19.5 μT and overall magnitude of 50 μT .

For the magnetometer to provide accurate measurements, it should be free of external magnetic interference at levels in the order of, or greater than, 5 μT (an order of magnitude less than the earth's magnetic field).

An analysis was conducted to see whether the required operating conditions for the magnetometer may be achievable.

Magnetic Latches

A magnetic analysis was carried out to determine the effects of magnetic leakage from the button-magnets adopted to secure the access covers on the quadruped. In this case, a 3D model with a spherical air-boundary was used. A pair of magnets representing one on the quadruped's chassis and the other on the detachable panel have been modelled.

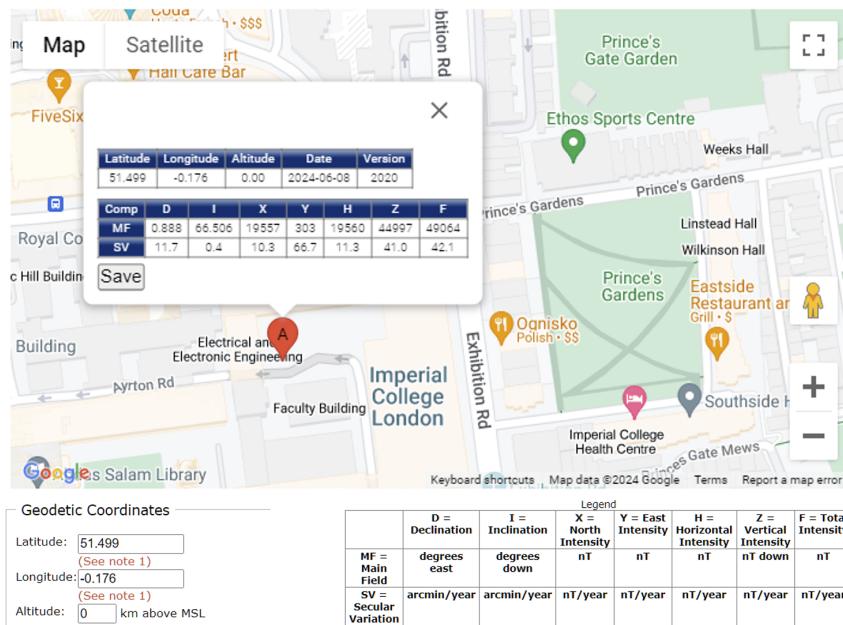


Figure 5.17: Earth's Magnetic Field Intensity at Imperial College [UK Geomagnetic Survey]

Table 5.5: Parameters for Magnetic FEA Model

Analysis Type	3D
Magnet size	2 magnets: D 6mm, L 3mm
Magnet Grade	NdFeB, N30 (30 MGOe)
Gap between magnets	0.1 mm (in case of imperfect contact)
Model size	Radius 200 mm
magnet element size	0.1 mm max
Air-space element size	5 mm max
Number of elements	2,166,842

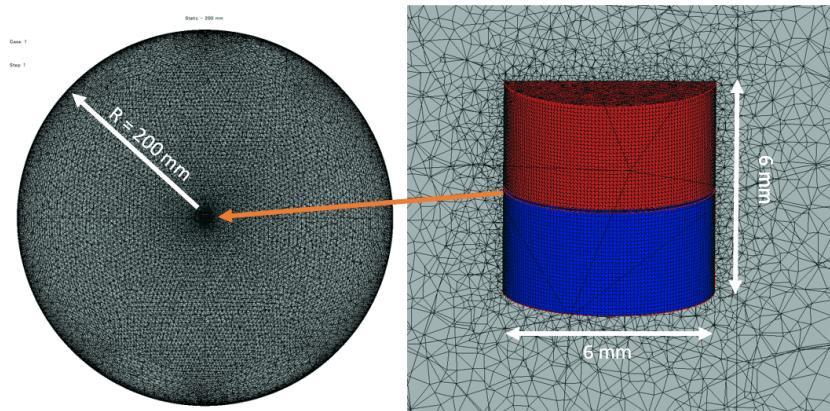


Figure 5.18: Two magnets in the centre of the model, extending to a radius of 200mm

The results of this analysis, shown in Figure 5.19, are displayed such that the magnitude of the earth's magnetic field coincides with the green coloured contour section. Therefore, to be sufficiently distant from the effects of magnet's leakage flux, the magnetometer, should be at least 150 mm away from them.

However, if on the various locations of the removable panels, the magnet pairs could be arranged with, say, all S or N poles facing inwards, a central region of theoretically null magnetic flux density could conceivably be created in which to place the magnetometer.

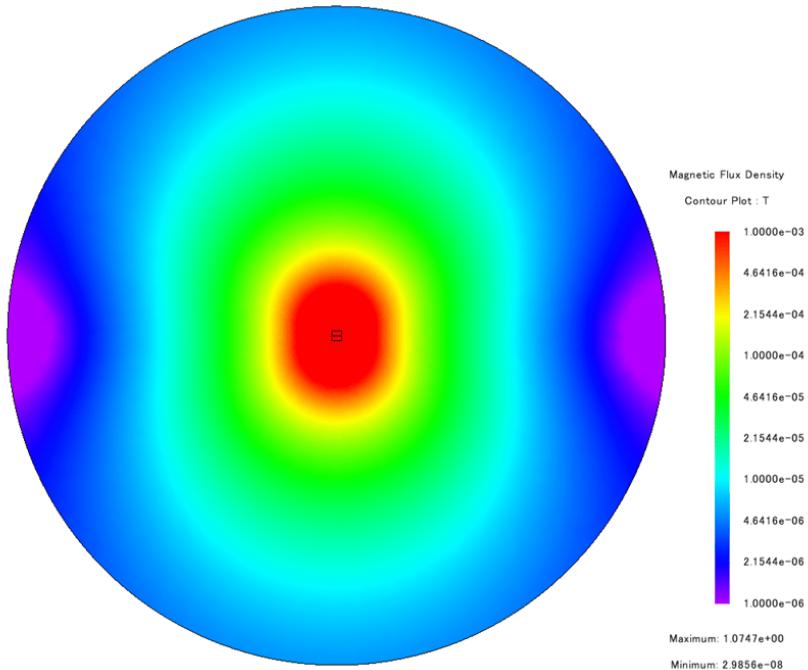


Figure 5.19: Magnetic field intensity plot from two magnets, section through X-Z plane, outer radius: 200 mm

Nevertheless, the most appropriate solution is probably to abandon the magnetic latches for an alternative, mechanical fixing method.

Modelling magnetic field due to main DC bus conductors The main battery current is carried by two conductors of 1.5mm^2 cross-section, spaced 20 mm apart laterally and carrying 15 Amperes each, in opposite directions. This two-conductor arrangement was similarly analysed using a finite element method, with the parameters shown in the table, below. The boundary of the model air-region was placed as far as practicable (500 mm) from the conductors it its centre.

Table 5.6: Parameters for DC Bus FEA Model

Analysis Type	2D
conductors	Two: 1.5 mm^2 , 20 mm apart
Material	copper
Model size	1000 mm x 1000 mm
Conductor element size	0.1 mm max
Air-space element size	5 mm max
Number of elements	1,841,956

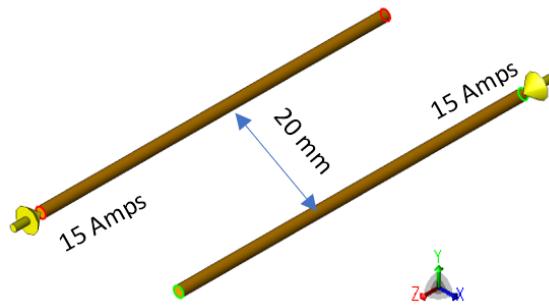


Figure 5.20: Model of the main DC bus conductors

The results of this analysis are depicted in Figure 5.21. This shows the magnetic field strength radiating from each conductor. The coloured contours are on a logarithmic scale with the limits selected to place the green band in the region of the earth's magnetic field strength (white arrow on scale). This region is centred on a radius of approximately 40 mm, with the magnetic field becoming sufficiently low at a distance of 100 mm or more from the conductor.

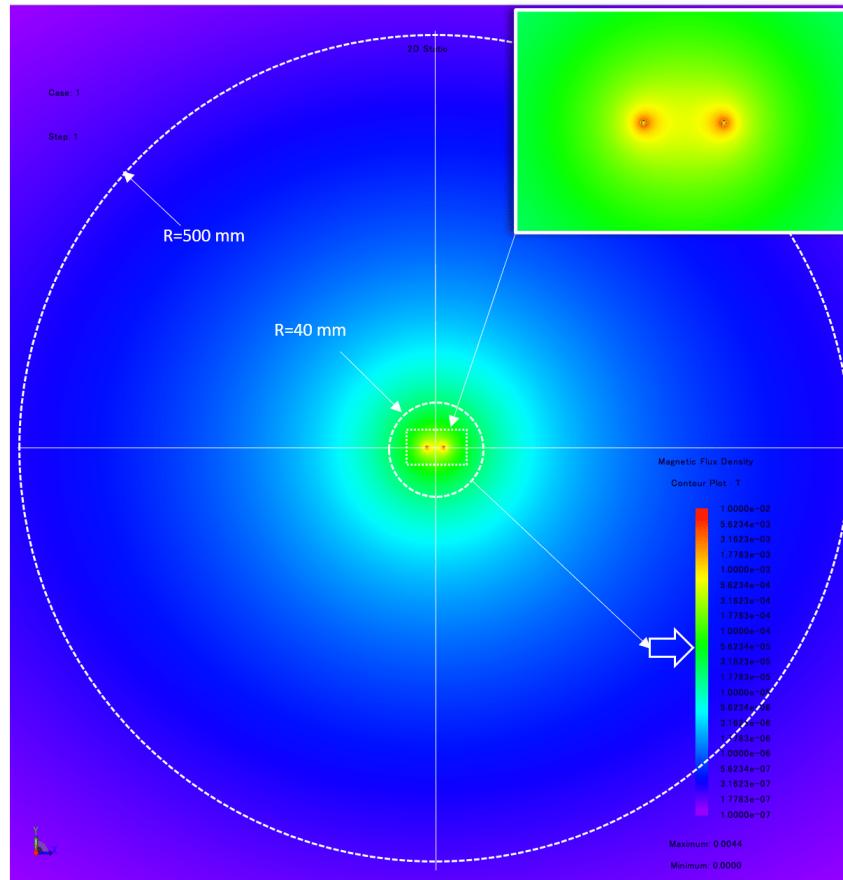


Figure 5.21: Magnetic field intensity plot as a result of the current in the DC bus conductors

Magnetically shielding the bus conductors may be a solution but, due to their location along the chassis, care would be needed not to also shield the sensor area from the earth's magnetic field.

A more practical solution to this may be to adopt a co-axial bus-bar arrangement, where the

magnetic field cancel out (a method adapted for undersea DC cables).

Motor Magnetic Field

There is likely to be significant magnetic leakage from the electrical steel lamination forming the motor's stator back-iron. Given that the motor uses permanent magnets, a simplified motor arrangement without windings could be modelled to obtain an estimate of potential magnetic flux leakage from the motor casing. Nevertheless, it would first be necessary to determine the motor construction and preferably also the grade of magnet used. This would likely be a destructive approach and was therefore avoided at this stage, since the above two other sources of magnetic interference would also need to be resolved first.

In any case, because of their nature and high operating current, it is likely that magnetic interference from the drive motors would need to be addressed. High-permeability materials (mu-metals) would probably be the best solution for shielding the outer-casing of each motor.

5.2.5 Rear Panel Assembly

Item	Details	Quantity
M3 Screws	6mm	6
Slide Switches	Single Pole	2
Switch	Dual Pole	1
Voltmeter	7 Segment Display	1
USB-C Module	15V Quick Charge	1
PLA+ Filament	Yellow	1 kg
PLA+ Filament	Black	1 kg
Molex Connector	Two pin set	2
Neodymium Magnets	6x3mm	10

Table 5.7: Materials for Switches and Connectors



Figure 5.22: Rear Panel of the Quadruped

Control Switches and Voltmeter Integration

The rear panel was built according to design, with two slide switches and a voltmeter, each serving distinct functions. One switch controls the power to the onboard computer, and the other switch manages the charging circuit. The voltmeter provides real-time monitoring of the battery voltage, allowing for immediate assessment of the battery's status.

Mounting and Connectivity

The entire rear panel assembly was designed for easy removal and maintenance. This is facilitated by the use of Molex connectors, which allow the rear panel to be disconnected from the main circuitry of the robot. These connectors ensure that the panel can be swiftly detached without the need to de-solder any electrical connections.

Rear Panel Cover

The rear panel is protected by a cover that features cutouts for the switches and voltmeter, ensuring they remain accessible while being shielded from external damage. This cover was attached using magnets, which not only secure the cover in place but also allow for quick and easy removal when access to the internal components is needed.

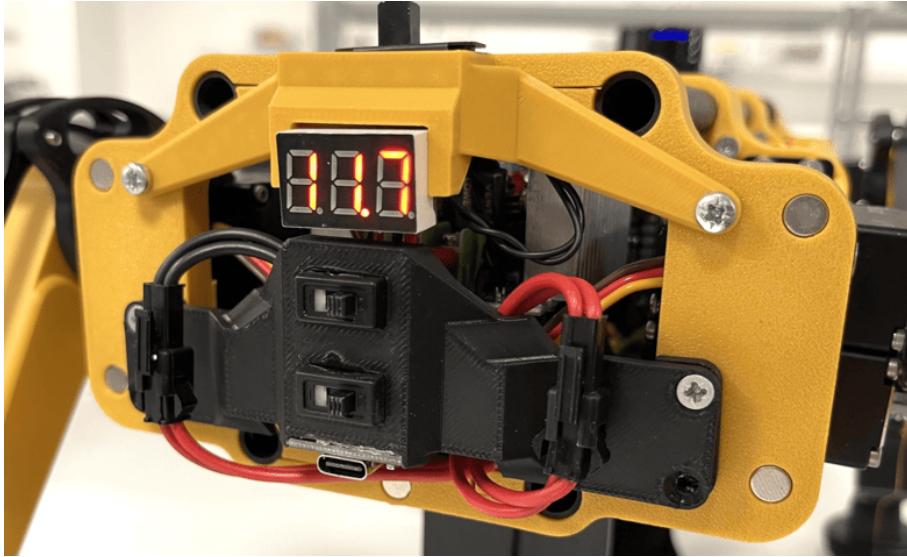


Figure 5.23: Internal Panel

The rear panel assembly was designed with a focus on modularity and user-friendly maintenance. The integration of slide switches and a voltmeter enhances the control and monitoring capabilities of the robot, while the use of Molex connectors and a magnetically attached cover ensures that the panel can be easily managed during operation and servicing.

5.3 Power and Drive System Implementation

This section covers the implementation of the power distribution system, battery management, motor control circuits, and safety mechanisms. The primary objective of this section is to provide a comprehensive overview of the electrical systems and their integration into the robot.

Key aspects covered in this section include:

- **Power Distribution System:** Details of the power supply arrangement, voltage regulation, and distribution of electrical power to various subsystems.
- **Battery Management:** Description of the configuration and assembly of the battery system, including charging circuits, monitoring, and safety features to ensure reliable power supply.
- **Motor Control Circuits:** Outline of the design and implementation of motor drivers and control circuits for precise movement and operation of the robot.
- **Safety Mechanisms:** Discussion of the safety features integrated into the electrical system to protect against overcurrent, overheating and other potential hazards.

5.3.1 Battery Pack Assembly and Charging

The battery pack is a major component of the robot's power system, designed to provide a stable and reliable power supply. This subsection details the assembly process, material selection, and integration of the battery pack.

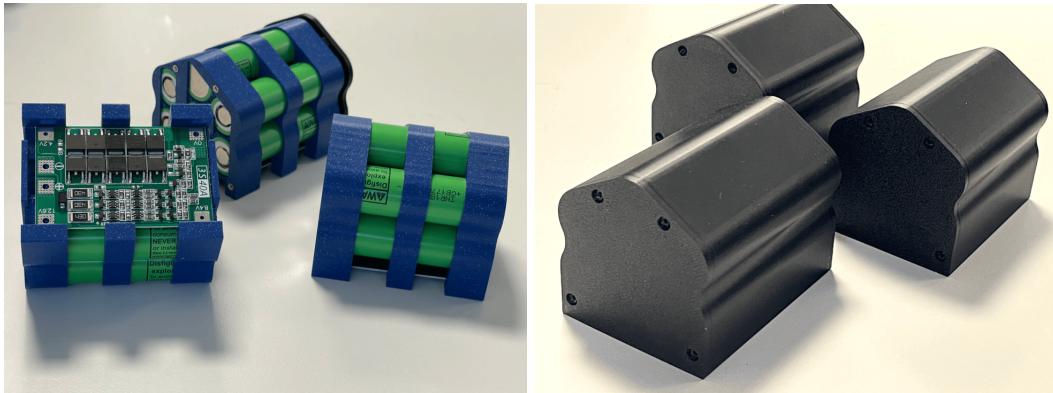


Figure 5.24: Multiple Battery Packs Assembled

Assembly

The spacers between the cells were fabricated from ASA filament, selected for its excellent thermal resistance. ASA's ability to withstand high temperatures ensures that the spacers remain structurally stable, preventing deformation that could compromise the battery pack's integrity.

Multiple test prints were conducted to ensure a perfect fitment of the cells in the outer casing, guaranteeing that nothing could be loose and liable to vibration.

The cells were interconnected using nickel strips, which were affixed using a spot welder, as seen in Figure 5.25. This method provides a robust and reliable connection, essential for both electrical conductivity and safety. Spot welding minimises the risk of heat damage to the cells, a common concern with other types of soldering. The nickel material also provides a certain degree of corrosion resistance in case of operating in humid environments.

After spot welding the nickel strips, solder was used to connect the strips to the Battery Management System (BMS) board. This ensured that the BMS could effectively monitor and manage the individual cells, providing protections such as overcharge, over-discharge, and short-circuit protection.

For the main battery cable, 1.5mm copper single core wire was used, selected for its high rated DC current capacity of 15A. An XT-60 connector was attached to the end of the cable for safe and reliable connections. The XT-60 connector is known for its secure connection, making it a suitable choice for the robot's power system.

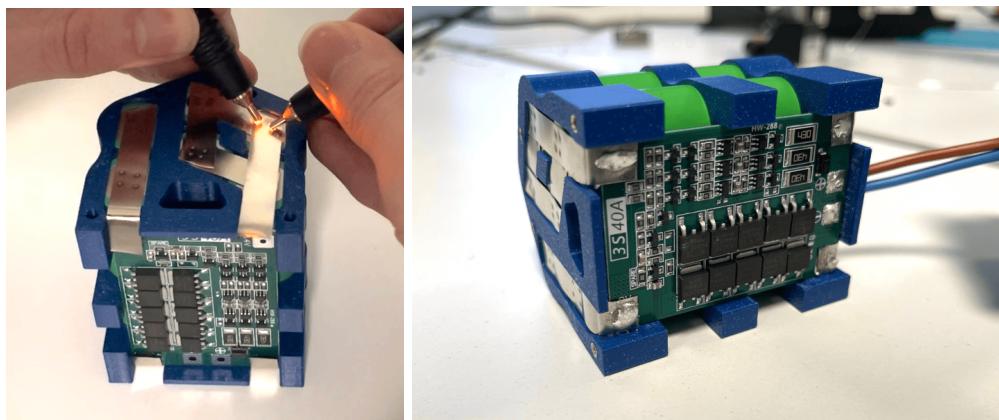


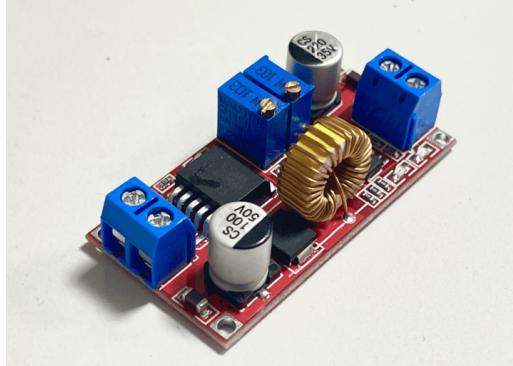
Figure 5.25: Spot Welded Battery Pack

To minimise downtime, three battery packs were produced, reducing the time spent charging between operations, as shown in Figure 5.24. Care was taken during the assembly process to ensure reliability, particularly with regard to the spot-welded connections and handling of the partly-charged batteries.

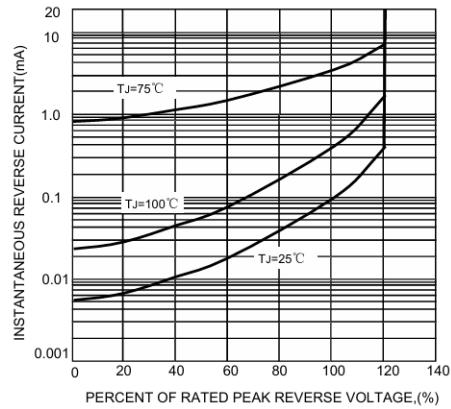
Charging Circuit

To charge the battery pack, a 5A rated, CC/CV buck converter meeting the design requirements (Figure 5.26a) was sourced and configured to charge the battery at 12.6V and 3A. During the initial test-phase, it was discovered that there was an issue with reverse current from the battery causing the charging circuit to partially energise, even when the battery was not being charged. A design modification was required to address this problem, which involved the addition of a blocking diode and re-calibration of the converter's output voltage to compensate for the voltage drop across the diode. As shown previously in Figure 4.26, an SR560 Schottky diode was added between the

output of the charger and the battery. The Schottky diode was chosen for its low voltage drop and minimal reverse current leakage, effectively preventing reverse current from flowing back into the charging circuit and ensuring reliable and efficient charging of the battery pack.



(a) Charging Circuit Buck Converter



(b) SR560 Diode Reverse Current Graph

Figure 5.26: Components used in the charging circuit - [Fig.B: HY Semiconductor REV. 6, 30-Dec-2014]

The graph shown in Figure 5.26b indicates the reverse current characteristics of the Schottky diode at different temperatures and percentages of the rated peak reverse voltage. Charging at 12.6V corresponds to about 20% of the SR560's rated voltage. From the graph, at 20% of the rated peak reverse voltage, the reverse current draw is approximately 0.005mA, which is quite minimal and acceptable for this application.

5.3.2 Power Delivery and Motor Control

Delivering power to the motors and controlling them is an important aspect of the quadruped robot's design. This subsection details the components and configuration used to manage power distribution and motor control effectively.

Power Delivery

The battery pack was connected to the 300W buck converter, which steps-down the voltage to 8V as required by the motor drives. During initial testing, it was found that a significant amount of noise (see Figure 5.30) from both the buck converter and the motor drives connected to it, was present on the power lines, potentially interfering with the Raspberry Pi's I²C communication.

Figure 5.29, below shows voltage and current measurements taken on the main DC bus, when energising the buck converter and drives. An inrush current with a 40A peak was expected, due to the capacitor charging but the continuous ripple affected the bus voltage and was carried through to the controller, evident on the I²C data bus, shown in Figure 5.30.

A custom power-distribution board was therefore developed to address this issues. This board was placed between the 8V output of the buck-converter and the array of motor drives. The power

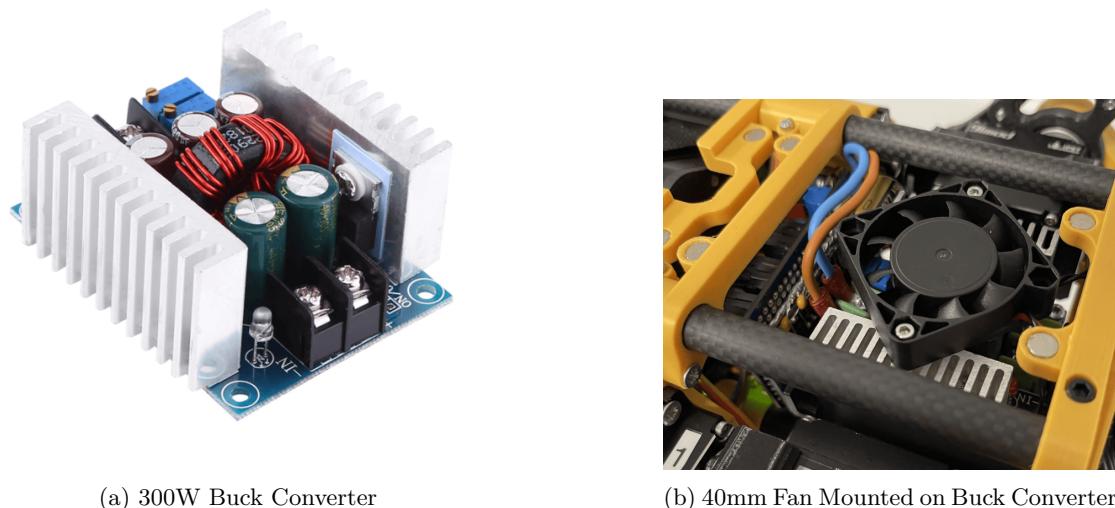


Figure 5.27: Buck Converter - 20A/300W

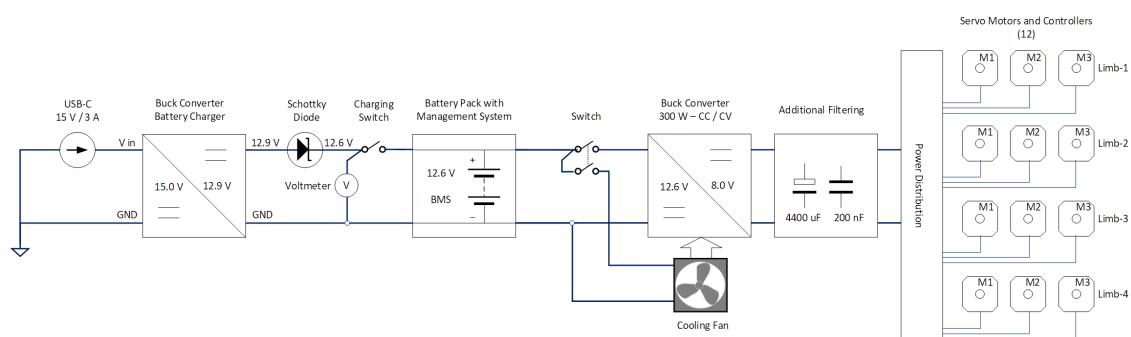


Figure 5.28: Schematic of the main power-delivery (including charger circuit)

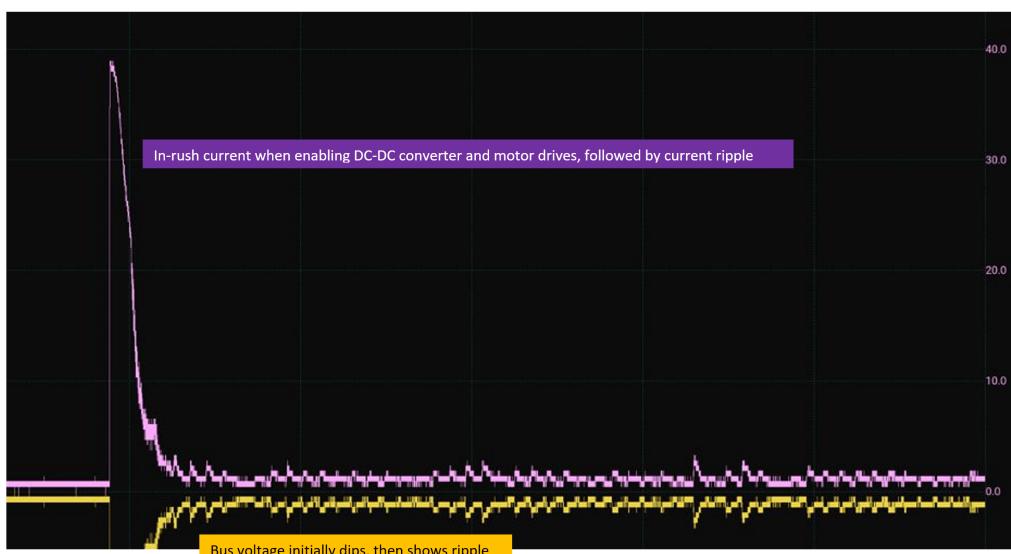


Figure 5.29: Measurement of inrush current and ripple when energising buck converter and drives

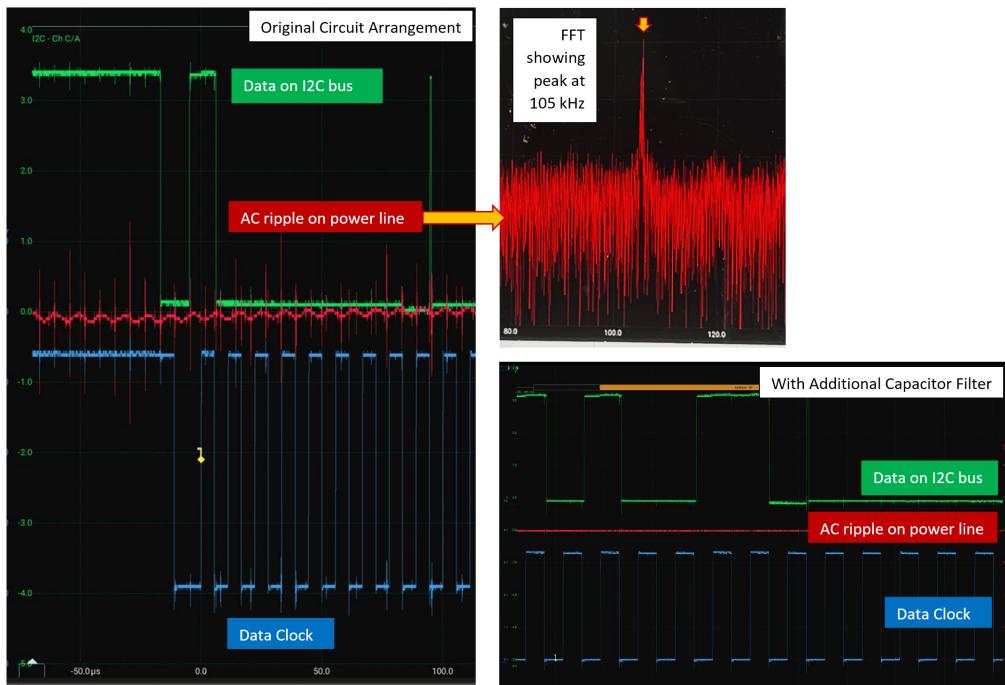
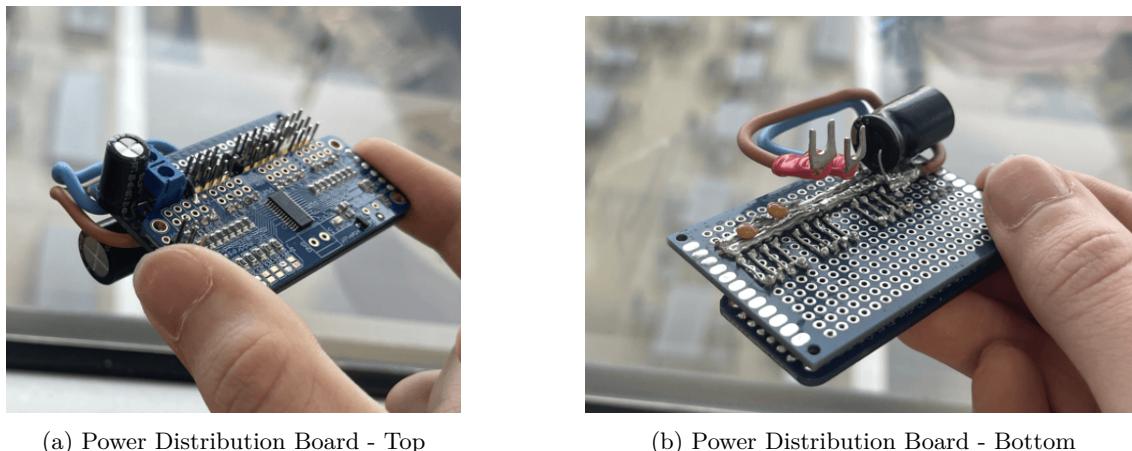


Figure 5.30: Left and Top: unfiltered noise from buck converter and motor-drives interfering with controller I²C bus. Lower Right: after filter modifications added.

distribution board includes two 2200 μ F capacitors to serve as power reservoirs, providing a steady supply of power during transient, high current draws by the motors. Additionally, two 100nF decoupling capacitors were included for filtering high-frequency noise that could affect the control circuits, ensuring smooth and stable operation. This can be seen in Figure 5.31.

Motor Control



(a) Power Distribution Board - Top

(b) Power Distribution Board - Bottom

Figure 5.31: Power Distribution Board Fused with Adafruit PWM Controller

The output of the power distribution board was connected to the PWM, motor position-controller, specifically a 16-channel, Adafruit PWM/Servo driver, of which 12 channels were used. This controller is responsible for generating the PWM signals required to control the motion of

each servo motor accurately, according to the demand encoded in each PWM signal. Each channel of the PWM was individually configured and calibrated, allowing precise control over each motor's movement and ensuring coordinated locomotion of the quadruped robot.

The 300W buck converter together with a modified power-distribution board to include capacitive filtering, was able to provide a stable, 8V power supply for the motor circuits, with minimal electrical interference to other circuits. The integration of the 16-channel PWM controller enabled precise motor control, which was essential for the robot's smooth and coordinated movement. These components worked together to create a reliable and efficient power and control system for the quadruped robot.

Integration of Raspberry Pi and Audio

REX's internal layout, as shown in Figure 5.32, highlights several key components that contribute to its functionality. The two speakers were mounted on the side panels with hot glue and mounting holes for alignment. Originally an I²C audio module was used, but there were driver problems with the operating system, leading to instability issues. Therefore, I opted for a separate audio amplifier and used an aux cable to connect to it.

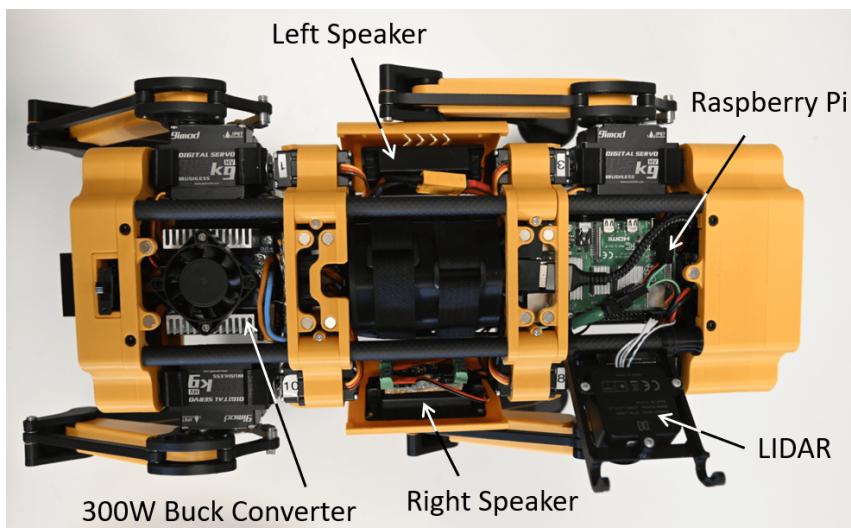


Figure 5.32: Enter Caption

Wider Implications and Electromagnetic Compatibility

When designing electronic equipment, one must pay particular attention to the various standards and practices to ensure electromagnetic compatibility. The equipment should not emit excessive interference, either conducted or radiated and, equally, should be sufficiently immune to the same from other sources. There are various national standards at different levels – commercial, military, aerospace and so on, but most of the basics are captured under the IEC 61000 series with certain tests specified in the CISPR 16 series.

In so far as the design of this quadruped is concerned, the following (see Table 5.8) were identified as possible sources of electromagnetic interference (EMI):

Table 5.8: Possible sources of EMI

Source	Type	Possible effect	Mitigation	Notes
DC-DC Con-verter	Conducted, 105kHz ripple	Interference with 100kHz I ² C bus	Separate power and control cir-cuits; filter with decoupling capacitors	
Motor Drive	Conducted PWM ripple	Interference with control cir-cuits/systems	Separate power and control cir-cuits; filter with decoupling capacitors	
Motor Drive	Radiated RF har-monics	Interference with radio comms	Best-practise de-sign	Not tested
Conductors	Radiated mag-netic fields	Interference with magnetometer	Use of co-axial cables for main power-bus; twist-ing of conductors; placement of conductors	Design revision required
Servo Motors	Radiated mag-netic fields	Interference with magnetometer	Use of hi-mu, magnetic shield-ing, such as Permalloy ®	Future work
Panel retention magnets	Radiated mag-netic fields	Interference with magnetometer	Replace with al-ternative fixings	Future work

Components operating at high currents and switching at high frequency are the most likely to create conducted interference. A certain amount of radiated, radio-frequency, interference may also be produced but EMC test receivers and antennas are required to make these measurements in a controlled environment.

Thermal Measurements

As part of the testing and validation process, an extensive thermal scanning of the robot and its components was carried out under operating conditions. A selection of the obtained images is shown in Figure 5.33. The two images labelled 'A' show the thermal map from the top of the robot, with the battery pack in the centre, reading 28.6 °C. The two images labelled 'B' show the rear section of the robot and particularly, the main DC-DC converter, operating at 48.4 °C. Interestingly, this figure correlates well with the previous thermal FEA modelling of this component, which indicated 50 °C . The two images labelled 'C' show the rear motors of the robot and, at 56.9 °C, these are clearly operating at the highest temperatures of any of the components.

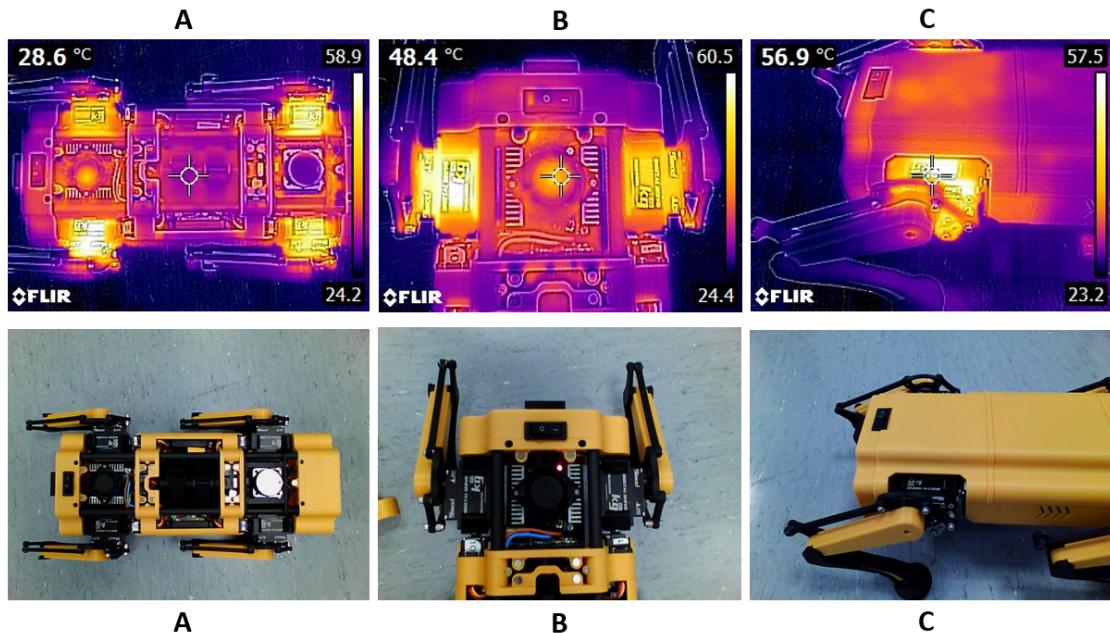


Figure 5.33: Thermal measurements taken using a FLIR Camera, after 15 minutes of continuous operation. A: Overall top view; B: Rear Section with DC-DC converter; C: Rear-Leg Motors

5.4 Bill of Materials (BOM) and Expenditure

The full bill of materials can be found in Appendix A, Table A.2. This verifies the overall cost of the project was within the budget of £1,200, which had been agreed with the project sponsor IBM. The total expenditure, including raw materials for manufacturing, amounted to £964.67, which is considerably lower than that of any other quadrupeds reviewed in this report. As indicated in the BOM, all components were commercially available, eliminating the need for designing custom circuits or sensors.

In addition to the BOM, other parts were purchased for trials and experimentation, but eventually not used in the robot. The total cost of these parts amounts to £185.40. This can be found in Table A.1.

6

Software Implementation

Contents

6.1	Introduction	82
6.2	Calculation of Kinematics	82
6.2.1	Forward Kinematics (FK)	82
6.2.2	Inverse Kinematics (IK)	82
6.3	Software Simulation with PyBullet	86
6.3.1	Background on PyBullet	86
6.3.2	Gait Cycle Generation and Walking Simulation	87
6.3.3	Geometry Transformations	88
6.3.4	URDF (Unified Robot Description Format)	90
6.3.5	Conclusion	93
6.4	Path Planning Experimentation	93
6.5	Software Setup and Configuration	94
6.5.1	Initial Setup	95
6.5.2	LIDAR Setup	96
6.5.3	Cartographer Setup	97
6.5.4	Oak-D Lite Depth Camera Setup	98
6.5.5	Nav2 Setup	99
6.6	Control System for REX	100
6.6.1	Servo Calibration	100
6.6.2	Calibration Program	101
6.6.3	Controller Service	101
6.6.4	Testing and Debugging	102
6.7	Coordinate Acquisition from Depth Camera	102
6.7.1	Detection and Depth Calculation	102
6.7.2	Angle and Position Calculation	104
6.7.3	Coordinate Transformation	104
6.7.4	Final Coordinates in the Robot Frame	105
6.7.5	Tracking and Data Publishing	105
6.8	Autonomous Navigation Implementation	106
6.8.1	Initialisation and Parameter Loading	106
6.8.2	Map and Odometry Processing	106
6.8.3	Frontier Identification and Grouping	107
6.8.4	Path Planning Using A* Algorithm	107
6.8.5	Path Smoothing with B-Splines	107
6.8.6	Pure Pursuit Path Tracking	107
6.8.7	Local Obstacle Avoidance	107

6.8.8 Exploration Loop	107
6.8.9 Wavefront Exploration Simulation	108
6.9 WatsonX Assistant	109
6.9.1 Wake Word Detection	110
6.9.2 Speech-to-Text Conversion	110
6.9.3 Interaction with Watson Assistant	110
6.9.4 Text-to-Speech Conversion	110
6.9.5 System Integration	111
6.10 Website Dashboard	111
6.10.1 Overview	111
6.10.2 Key Components	113
6.10.3 Functionality and User Experience	114
6.10.4 Conclusion	114
6.11 Final Overview	114

6.1 Introduction

The software implementation of the quadruped robot involved developing the code necessary to control the robot's movements, process sensor data, and enable autonomous operation. This chapter covers the development and integration of the software systems, starting with the calculations for inverse kinematics, which are required for simulating and controlling the quadruped's movements. The complete source code for this project is available at the following repository: <https://github.com/SpaceBod/QuadREX>

6.2 Calculation of Kinematics

6.2.1 Forward Kinematics (FK)

For a quadruped robot, forward kinematics involves calculating how each leg of the robot will move based on the angles of the joints (hip, femur, tibia, etc.). This is done by defining a series of coordinate frames at each joint and using transformation matrices to describe the position and orientation of one frame relative to the previous one. By sequentially multiplying these matrices, it is possible to compute the position and orientation of the end effector in the base frame's coordinate system.

6.2.2 Inverse Kinematics (IK)

Inverse kinematics [36] determines the joint parameters that provide a desired position of the end effector. Essentially, it is the reverse problem of forward kinematics. IK is more complex because there could be multiple joint configurations for a single end effector location, making the solutions not unique. This is known as kinematic redundancy.

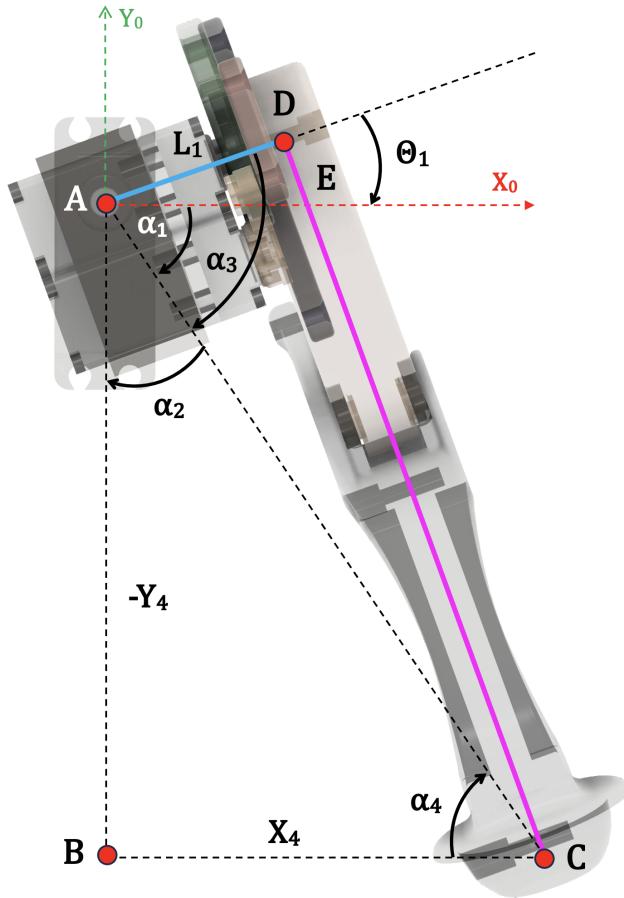


Figure 6.1: Calculating Kinematics for θ_1

Calculating θ_1

Based on figure 6.1, the angle θ_1 can be determined through a series of geometric relationships within the robot leg structure. The following equations present the step-by-step calculation:

$$\theta_1 = \alpha_3 - \alpha_1. \quad (6.1)$$

In $\triangle ADC$, the angle α_3 is given by:

$$\alpha_3 = \arctan \left(\frac{\sqrt{x_4^2 + y_4^2 - L_1^2}}{L_1} \right). \quad (6.2)$$

For $\triangle ABC$, considering the right angle at B , we have:

$$\alpha_1 + \alpha_2 = 90^\circ. \quad (6.3)$$

This implies that:

$$\alpha_1 = 90^\circ - \alpha_2. \quad (6.4)$$

Furthermore, for $\triangle ABC$, sum of angles in a triangle gives us:

$$\alpha_2 + \alpha_4 + \angle B = 180^\circ, \quad (6.5)$$

where $\angle B$ is a right angle. Therefore,

$$\alpha_2 = 90^\circ - \alpha_4. \quad (6.6)$$

Substituting the value of α_2 into the earlier equation yields:

$$\alpha_1 = 90^\circ - (90^\circ - \alpha_4), \quad (6.7)$$

which simplifies to:

$$\alpha_1 = \alpha_4. \quad (6.8)$$

The angle α_4 in $\triangle ABC$ can be expressed as:

$$\alpha_4 = \arctan\left(-\frac{y_4}{x_4}\right). \quad (6.9)$$

Now, substituting the values of α_1 and α_3 into Equation 6.1, we get:

$$\theta_1 = \alpha_3 - \alpha_1, \quad (6.10)$$

which results in:

$$\theta_1 = \arctan\left(\frac{\sqrt{x_4^2 + y_4^2 - L_1^2}}{L_1}\right) - \arctan\left(-\frac{y_4}{x_4}\right). \quad (6.11)$$

Calculating θ_2

From $\triangle ABFE$,

$$\alpha_1 + \alpha_2 + \theta_2 = 90^\circ \quad (6.12)$$

$$\therefore \theta_2 = 90^\circ - \alpha_1 - \alpha_2. \quad (6.13)$$

From $\triangle ABC$,

$$\alpha_2 = \arctan\left(\frac{-Z_4}{\sqrt{x_4^2 + y_4^2 - L_1^2}}\right). \quad (6.14)$$

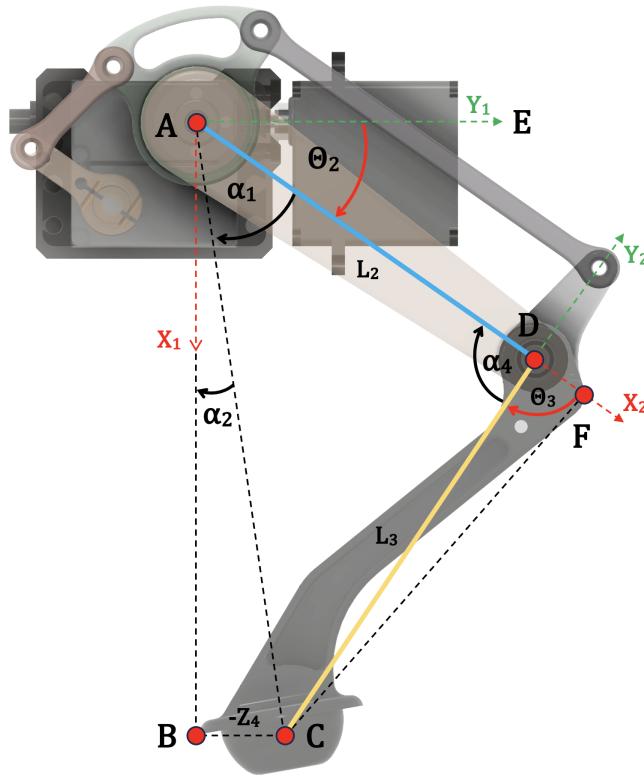
From $\triangle ACF$,

$$\alpha_1 = \arctan\left(\frac{CF}{AF}\right). \quad (6.15)$$

Furthermore, from $\triangle CDF$,

$$\sin \theta_3 = \frac{CF}{CD'}, \quad (6.16)$$

$$CF = CD' \sin \theta_3, \quad (6.17)$$

Figure 6.2: Calculating Kinematics for θ_2 & θ_3

$$CF = L_3 \sin \theta_3 \quad [CD' = L_3], \quad (6.18)$$

$$\cos \theta_3 = \frac{DF}{CD'}, \quad (6.19)$$

$$DF = L_3 \cos \theta_3 \quad [CD' = L_3], \quad (6.20)$$

$$AF = AD + DF, \quad (6.21)$$

$$AF = L_2 + L_3 \cos \theta_3 \quad [AD = L_2]. \quad (6.22)$$

Similarly,

$$\therefore \alpha_1 = \arctan \left(\frac{L_3 \sin \theta_3}{L_2 + L_3 \cos \theta_3} \right). \quad (6.23)$$

Now, substituting the values of α_1 and α_2 into Equation (6.13),

$$\theta_2 = 90^\circ - \arctan \left(\frac{L_3 \sin \theta_3}{L_2 + L_3 \cos \theta_3} \right) - \arctan \left(\frac{-Z_4}{\sqrt{x_4^2 + y_4^2 - L_1^2}} \right), \quad (6.24)$$

Calculating θ_3

Finally, we solve for θ_3 using Figure 6.2:

$$\theta_3 = 180^\circ - \alpha_4. \quad (6.25)$$

In $\triangle ACD$, by the Law of Cosines we get:

$$AC^2 = AD^2 + CD^2 - 2 \cdot AD \cdot CD \cdot \cos(\alpha_4) \quad (6.26)$$

which leads to the angle α_4 :

$$\alpha_4 = \arccos \left(\frac{AD^2 + CD^2 - AC^2}{2 \cdot AD \cdot CD} \right). \quad (6.27)$$

In $\triangle ABC$, the Law of Cosines gives us:

$$AC^2 = AB^2 + BC^2, \quad (6.28)$$

and by substituting the known values we find:

$$AC^2 = \left(\sqrt{x_4^2 + y_4^2 + L_1^2} \right)^2 + (-Z_4)^2, \quad (6.29)$$

which simplifies to:

$$AC^2 = x_4^2 + y_4^2 + L_1^2 + Z_4^2. \quad (6.30)$$

Now, substituting the value of AC^2 into Equation (6.27), we obtain α_4 :

$$\alpha_4 = \arccos \left(\frac{L_2^2 + L_3^2 - x_4^2 - y_4^2 - L_1^2 - Z_4^2}{2L_2L_3} \right), \quad (6.31)$$

where $AD = L_2$ and $CD = L_3$. Thus, we can find θ_3 :

$$\theta_3 = 180^\circ - \arccos \left(\frac{L_2^2 + L_3^2 - x_4^2 - y_4^2 - L_1^2 - Z_4^2}{2L_2L_3} \right). \quad (6.32)$$

In summary, by deriving these angular values (θ_1 , θ_2 , and θ_3), with the corresponding lengths and joint constraints, the robot can be programmed to execute movements with high fidelity. The equations developed herein serve as the backbone for the algorithmic control that synchronises the robot's limb movements with the intended locomotive patterns.

6.3 Software Simulation with PyBullet

To simulate the quadruped robot and its walking gaits, PyBullet was used as the primary simulation tool. This section provides an overview of PyBullet, its advantages, and the process of using it to simulate the quadruped's movements.

6.3.1 Background on PyBullet

PyBullet is an open-source physics engine and simulation environment that supports real-time physics simulations for robotics and machine learning. It is built on top of the Bullet Physics

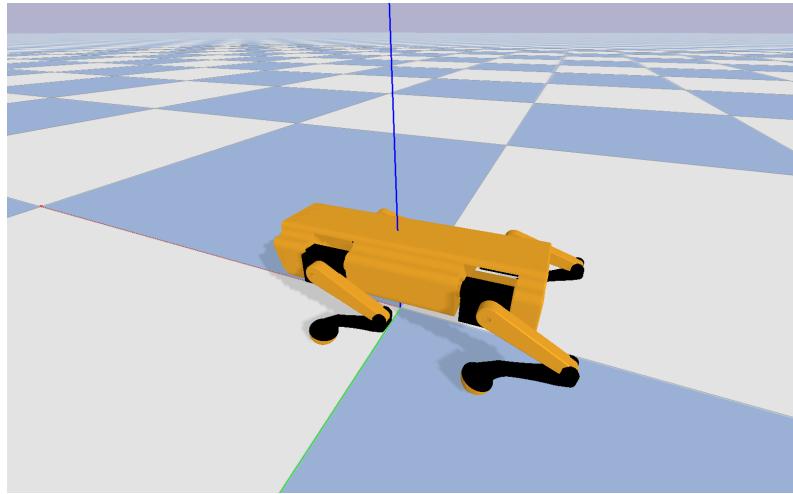


Figure 6.3: Simulated Environment using PyBullet

Library and provides a simple Python API, making it accessible and easy to use for developers. It offers several key features that make it a good choice for simulating robotics systems:

- Accurate Physics Simulation: PyBullet provides accurate and realistic physics simulations, including rigid body dynamics, collision detection, and soft body (eg squash-ball parts on feet) dynamics.
- Real-Time Simulation: It allows for real-time simulation and visualization, enabling developers to interact with and control the simulation as it runs.

6.3.2 Gait Cycle Generation and Walking Simulation

The generation of the gait cycle and the use of inverse kinematics were central steps in achieving realistic walking behavior for the quadruped robot. This section explains how the gait cycle was generated and how inverse kinematics were utilised to control the robot's movements.

The gait cycle of the quadruped robot was generated using a gait planner, specifically designed to create coordinated leg movements that mimic natural walking gaits. A Python class, *TrotGait*, was implemented to handle the generation of the gait cycle.

The key components of the gait cycle generation included:

- Stance Phase Calculation: The stance phase was calculated based on the current phase (ϕ_{St}), velocity (V), and trajectory angle. The stance phase determines the movement of each leg when it is in contact with the ground, providing propulsion and stability.
- Swing Phase Calculation: The swing phase was calculated using a Bezier curve, which provides a smooth transition of the leg through the air. This phase ensures that the legs move to the next position without colliding with the ground or other obstacles.

- Step Trajectory Calculation: The step trajectory for each leg was computed by combining the stance and swing phases. This trajectory determines the precise path that each leg would follow during a single step cycle.
- Loop Function: The loop function integrates the step trajectories for all four legs, ensuring synchronised movement and coordination. The function updates the positions of the feet relative to the body, maintaining balance and smooth motion.

Utilising Inverse Kinematics for Walking

The inverse kinematics calculations were used to convert the desired foot positions generated by the gait planner into the corresponding joint angles required to achieve those positions. This process ensured that the quadruped's legs move accurately and smoothly. The steps for using inverse kinematics in the walking simulation included:

- Definition of Joint Angles: The joint angles for each leg were calculated based on the desired foot positions provided by the gait planner. The solve method in the *RobotKinematics* class was used for this purpose.
- Setting the Joint Angles: The calculated joint angles were then applied to the robot's joints using PyBullet's in-built function, which controlled the position of each joint.
- Simulation of Movement: The robot's movements were simulated in the PyBullet environment, allowing for real-time observation and adjustment of the walking gait.

Generating Points with Linspace

In the gait cycle generation, linspace was used to generate evenly spaced points over a specified interval. This was particularly useful for creating smooth transitions and movements within the gait cycle. The linspace function generates a specified number of points between a start and an end value, which are then used to define the positions of the feet during the swing and stance phases. These time points are then used to compute the Bezier curve, which defines the trajectory of the foot during the swing phase.

6.3.3 Geometry Transformations

Geometry transformations were used to rotate and translate the coordinates of the feet, allowing for adjustments in pitch, yaw, roll, and translations along the x, y, and z axes. These transformations were essential for simulating different walking angles and adjusting the robot's orientation.

The transformations involved converting the coordinates of the feet from the body frame to the world frame, and vice versa. This allowed for the application of rotations and translations to achieve the desired foot positions. By using these geometry transformations, the gait planner is able to adjust the foot positions for various orientations and translations:

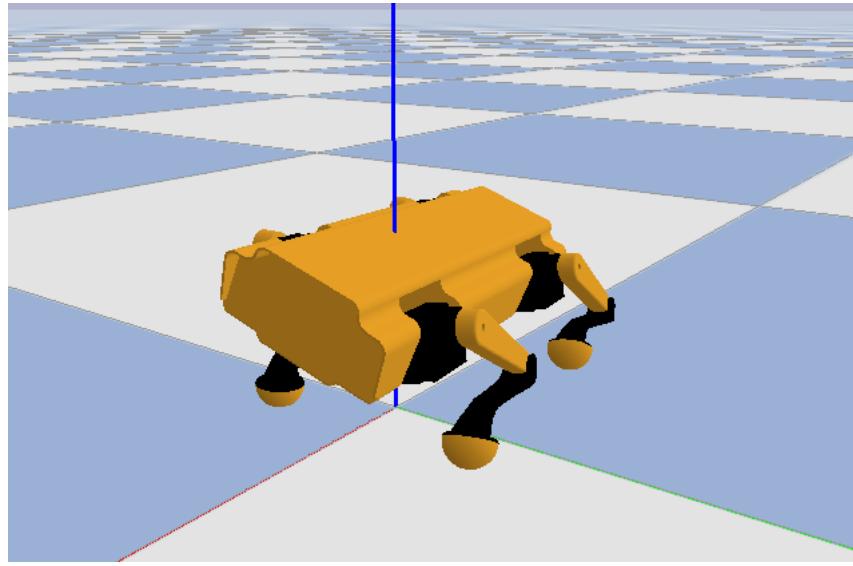


Figure 6.4: Simulated Model with Roll Orientation

- Pitch, Yaw, Roll: These rotations are applied to change the orientation of the robot. For example, a pitch rotation would tilt the robot forward or backward, while a yaw rotation would turn it left or right.
- Translations: These movements are applied to move the robot along the x, y, and z axes. For example, translating along the x-axis would move the robot forward or backward.

Rotation Matrices

The rotation matrices for roll (R_x), pitch (R_y), and yaw (R_z) are given by:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.33)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.34)$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.35)$$

Translation Matrix

The translation matrix for a translation by (x_0, y_0, z_0) is:

$$T(x_0, y_0, z_0) = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.36)$$

Combined Transformation Matrix

The combined transformation matrix M is obtained by multiplying the rotation and translation matrices:

$$M = R_{xyz}(r, p, y) \cdot T(x_0, y_0, z_0) \quad (6.37)$$

where

$$R_{xyz}(r, p, y) = R_x(r) \cdot R_y(p) \cdot R_z(y) \quad (6.38)$$

To apply this transformation to a vector $\mathbf{v} = [v_x, v_y, v_z, 1]^T$:

$$\mathbf{v}' = M \cdot \mathbf{v} \quad (6.39)$$

Figures 6.5 and 6.6 illustrate the gait cycles at different angles:

6.3.4 URDF (Unified Robot Description Format)

The Unified Robot Description Format (URDF) is an XML-based format used to describe the physical and visual properties of a robot model. It is widely used in robotics applications, particularly within the Robot Operating System (ROS) and simulation environments.

Key Components

URDF files consist of several key components:

- **Links:** Represent the rigid bodies or parts of the robot.
 - *Inertial:* Defines mass and inertia properties.
 - *Visual:* Describes the appearance using meshes or primitive shapes.
 - *Collision:* Specifies the collision geometry for simulations.

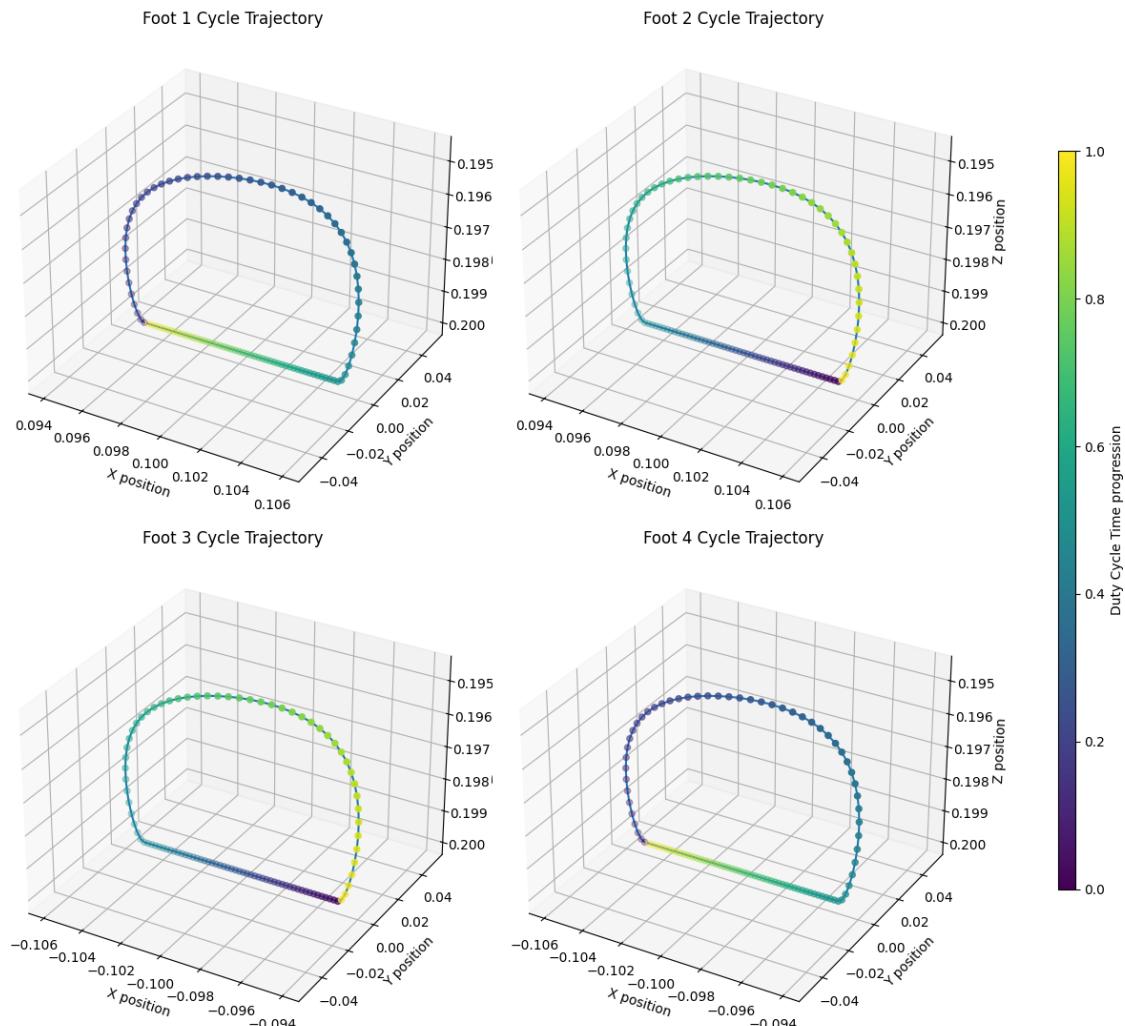


Figure 6.5: Gait Cycle - 0 Degrees

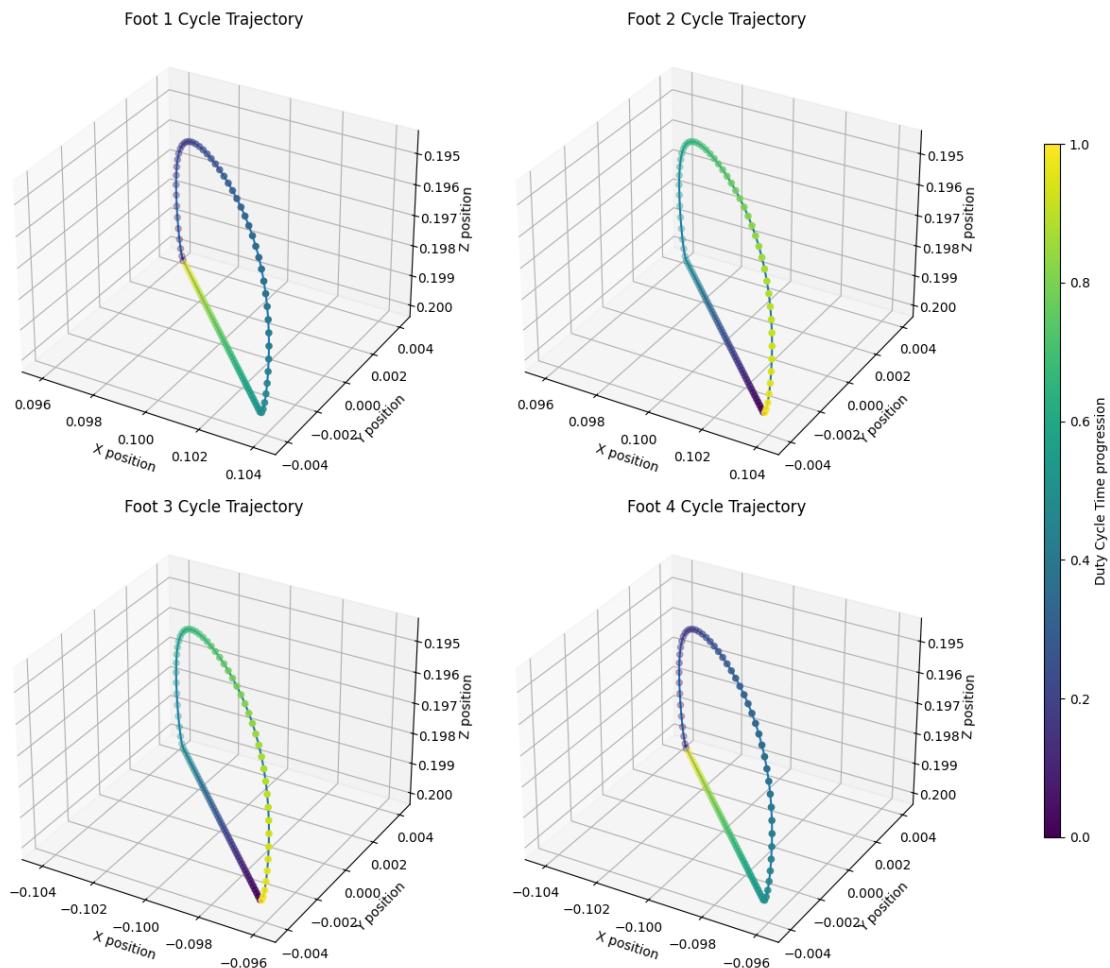


Figure 6.6: Gait Cycle - 45 Degrees

- **Joints:** Define the connections between links and their relative movements.
 - *Types:* Fixed, Revolute, Prismatic, Continuous, Floating, Planar.
- **Transmission:** Describes the relationship between actuators and joints.
- **Sensors:** Specifies the placement and types of sensors, such as cameras, LIDAR, and IMUs.

A URDF file was created to simulate REX in PyBullet by simplifying the geometry of the design to reduce computational load during simulation. Using Fusion360, a simplified model was exported and the visual and collision properties of the robot's component were accurately represented using 3D STL files.

6.3.5 Conclusion

The simulation of the quadruped robot using PyBullet served as an essential basis for implementing the real-world robot. By generating and refining the gait cycles in the simulation, it was possible to develop and test the control algorithms comprehensively before applying them to the physical robot. This approach ensured that the robot's movements were smooth, coordinated, and efficient.

6.4 Path Planning Experimentation

To determine the most effective path planning algorithm for the quadruped robot, a series of experiments were conducted, involving various algorithms. A Python program was developed, specifically designed to generate random maps populated with obstacles of varying quantities and sizes. This allowed for a comprehensive evaluation of each algorithm's performance under different conditions.

The program allowed edit the obstacle parameters to be edited, including their quantity and size, providing a flexible testing environment. Additionally, it was possible to adjust the intensity of the cost map, which was helpful for evaluating how different algorithms handle the computational load and complexity of path planning (Figure 6.8).

The primary objective was to identify the most efficient and reliable algorithm for navigating through complex environments. The algorithms tested included Breadth-First Search (BFS), Depth-First Search (DFS), Dijkstra's algorithm, and A*. Each algorithm was assessed based on its ability to find the shortest path, computation time, and overall reliability.

After extensive testing, it became evident that the A* algorithm outperformed the others. A* was consistently able to find the shortest path quickly, with lowest cost, even in highly obstructed maps. Its heuristic-based approach, which combines the benefits of Dijkstra's algorithm and greedy best-first search, allowed it to navigate through the maps with very good precision and speed.

The A* algorithm uses a heuristic function, commonly the Euclidean or Manhattan distance, to estimate the cost to a goal. It maintains two lists: an open list of nodes to be evaluated and a closed list of nodes that have already been evaluated. The algorithm prioritises nodes with the

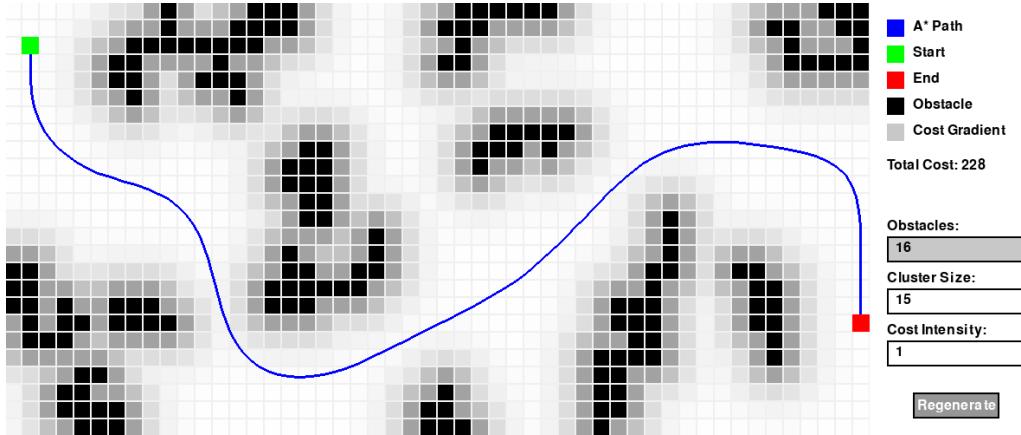


Figure 6.7: Python Path Planning Program - A* Algorithm

lowest total estimated cost, calculated as the sum of the actual cost from the start node to the current node and the heuristic cost from the current node to the goal.

The search process involves expanding the node with the lowest total cost, evaluating its neighbours, and updating their costs and paths accordingly. This process continues until the goal node is reached. The final path is then reconstructed by tracing back from the goal node to the start node using the recorded parent nodes.

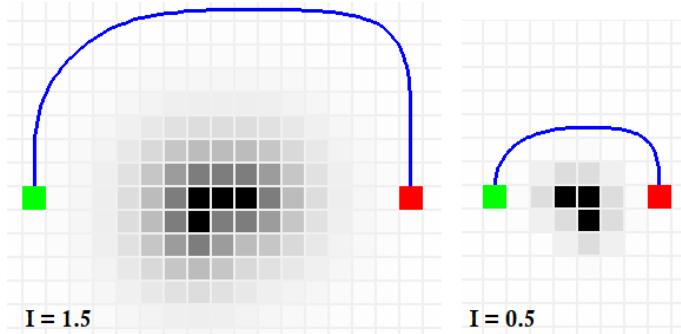


Figure 6.8: Comparison of Cost Intensities

The results of these experiments led to the conclusion that the A* algorithm was the best choice for the quadruped robot's path planning needs. Its superior performance in diverse and challenging environments made it the best candidate for implementation in the robot's navigation system. While there are more advanced algorithms based on A*, such as Theta* [37], Jump Point Search [38] and others that could potentially increase efficiency further, the performance of the standard A* algorithm was found to be more than satisfactory.

6.5 Software Setup and Configuration

This section details the steps taken to set up and prepare the Raspberry Pi 4 with Ubuntu 22.04 LTS and ROS2 Humble, ensuring a robust environment for the development and implementation of the quadruped robot's software.

6.5.1 Initial Setup

Installing Ubuntu 22.04 LTS

The first step in preparing the Raspberry Pi 4 involved installing Ubuntu 22.04 LTS. This process began by downloading the official Ubuntu image from the Ubuntu website due to its long-term support and compatibility with ROS2 Humble.

Setting Up ROS2 Humble

With Ubuntu 22.04 LTS installed, the next step was to set up ROS2 Humble. The system was first updated and upgraded to ensure all packages were current. Locale settings were configured to prevent any language-related issues. Subsequently, ROS2 repositories were added to the system by installing necessary software properties and updating the package list. The full desktop version of ROS2 Humble was then installed, providing a comprehensive development environment. The environment setup involved sourcing the ROS2 setup file to ensure ROS2 commands and tools were accessible. Finally, additional tools and dependencies required for building ROS2 packages were installed, including development tools and ROS2-specific dependencies. This preparation ensured the Raspberry Pi 4 was ready for robust development and deployment of the quadruped robot's software systems, facilitating efficient development and testing processes.

Establishing SSH Connection

To interact with the Raspberry Pi 4, especially in a headless configuration (without a display), an SSH (Secure Shell) connection was established. This method allowed for secure and remote access to the Raspberry Pi.

Automating IP Address Upload

A useful component of this setup was the implementation of a Python script that automated the upload of the Raspberry Pi's IP address to *ThingSpeak*, a public API for fetching and storing data. This script was configured to execute upon system boot, ensuring that the current IP address was always available for remote access. The Python script utilised network interfaces to determine the Raspberry Pi's current IP address. Upon retrieval, it used the *ThingSpeak API* to upload this data to a dedicated channel. The script was set to run at startup using *systemd* services. A service file defined the execution parameters and dependencies, guaranteeing that the script ran with each system startup.

6.5.2 LIDAR Setup

Hardware Connections

The LD19 LIDAR package was important for processing laser readings and converting them into LaserScan /scan messages, forming the foundation of the robot's navigation stack. To integrate the LD19 LIDAR sensor with the Raspberry Pi, the input port had to be switched from the USB interface to direct serial communication via the Pi's GPIO pins. This required the following connections:

- **5V & Ground:** Power supply for the LIDAR sensor.
- **Rx:** Receive data pin for serial communication.
- **PWM:** Pulse Width Modulation for controlling the rotation frequency.

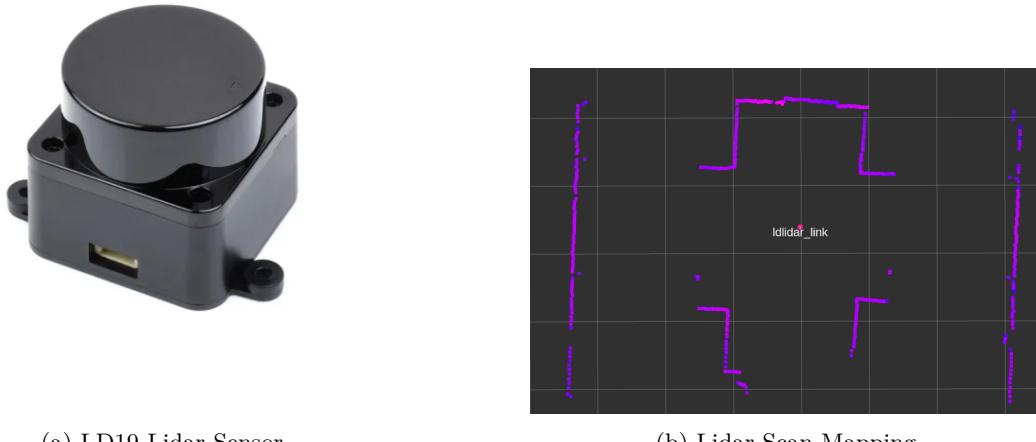


Figure 6.9: 2D Lidar Mapping

Software Configuration

The sensor was configured to use the /dev/Serial0 interface, and appropriate access privileges were granted for ROS to interact with it. The following commands were used to run the LIDAR node and manage its lifecycle:

```
ros2 run ldlidar_node ldlidar_node
ros2 lifecycle set /lidar_node configure
ros2 lifecycle set /lidar_node activate
```

Lifecycle Management

ROS2 lifecycle management was used to provide better control over the states of a node. By using lifecycle nodes, it is possible to manage the state of the node, allowing it to transition through different states such as configuring, activating, and deactivating. This enhances reliability and flexibility, enabling smooth startup, shutdown, and configuration of nodes without requiring a full system reboot. A launch file was created to handle the activation and start of this node. This ensured that the LD19 LIDAR sensor was consistently and reliably integrated with the Raspberry Pi, providing essential LaserScan data for the robot's navigation capabilities.

6.5.3 Cartographer Setup

Introduction to Cartographer

Cartographer is a ROS package developed by Google for real-time simultaneous localisation and mapping (SLAM). It is designed to create detailed 2D and 3D maps of an environment by processing data from various sensors. In this project, Cartographer was configured to use the LIDAR scans from the LD19 package, enabling the quadruped robot to perform accurate mapping and navigation.

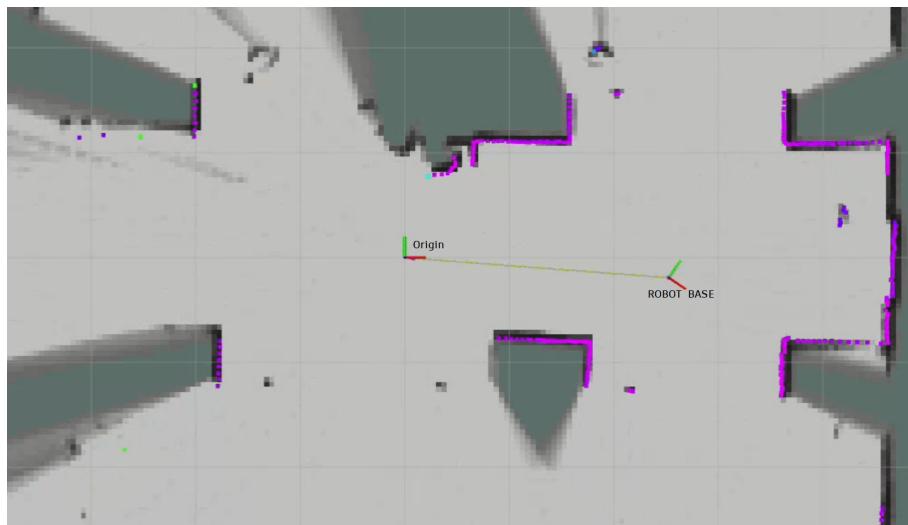


Figure 6.10: Example of a Local Map Generated by Cartographer with Localisation

Configuration Adjustments & Localisation

Due to the nature of the quadruped robot, it lacks encoders, like traditional wheel-based robots, to measure odometry directly. This meant it was harder to localise the robot in its environment. Cartographer was able to overcome this limitation by using scan data to estimate the robot's position through a process called Adaptive Monte Carlo Localisation (AMCL). This method allows the robot to localise itself within a map by matching LIDAR scan data to the existing map, thus compensating for the lack of direct odometry measurements.

The Cartographer configuration files were modified to accept LIDAR data from the LD19 package. This involved setting the correct topic names and adjusting parameters to match the specifications of the LD19 LIDAR sensor. A new launch file was created to handle this.

6.5.4 Oak-D Lite Depth Camera Setup

Installing DepthAI ROS2 Library

The DepthAI ROS2 library was used to interface with the Oak-D Lite camera. This library, which facilitates the integration of Oak-D Lite cameras with ROS2, was installed by building and compiling it from source. The installation process was straightforward, involving cloning the repository, checking out the appropriate branch, and building the package using *colcon* (a command-line tool for building software packages). Furthermore, the DepthAI SDK was installed separately to test functionality of the camera. By running the demo application, I was able to load a trained neural network to the camera and visualise the outputs, as seen in Figure 6.12.



Figure 6.11: Oak-D Lite Camera

Hardware Requirements

To ensure the Oak-D Lite depth camera operated correctly, it was essential to use a USB 3.0 cable instead of a USB 2.0 cable. The higher data transfer rate of USB 3.0 is necessary for handling the depth data generated by the camera. Using a USB 2.0 cable could result in connectivity issues and reduced performance.

Software Configuration

The DepthAI ROS2 library was configured to accept depth data from the Oak-D Lite camera. This involved setting the correct topic names and adjusting parameters to match the specifications of the depth camera. A new launch file was created to handle the activation and start of the depth camera node, ensuring it could start streaming depth data reliably.

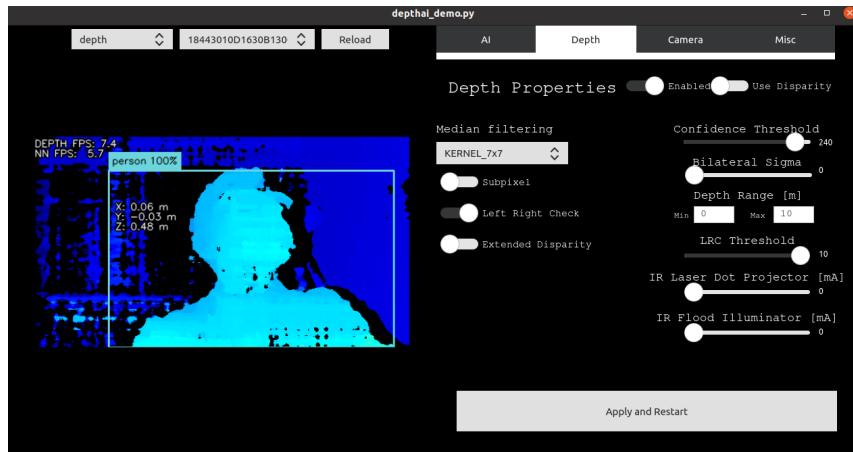


Figure 6.12: Demo Example using DepthAI’s GUI [39]

6.5.5 Nav2 Setup

Introduction to Nav2

Nav2, also known as Navigation2, is a ROS2 package designed for mobile robot navigation. It provides a framework for autonomous navigation, including path planning, control, and recovery behaviours. In this project, Nav2 was configured to work alongside Cartographer to enable the quadruped robot to navigate its environment effectively.

Nav2 Components

Nav2 utilises the maps generated by Cartographer and the LIDAR scan data from the LD19 package to perform path planning and obstacle avoidance. The integration between Nav2 and Cartographer allows the robot to create a map of its surroundings using Cartographer’s SLAM capabilities and then use this map for navigation tasks.

The full Nav2 stack includes several key components:

- **AMCL:** Uses the map generated by Cartographer and the LIDAR scan data to estimate the robot’s position. However, this was disabled, for I pre-configured Cartographer to perform this action instead.
- **Planner Server:** Generates a path from the robot’s current position to the goal position.
- **Controller Server:** Controls the robot’s movements along the planned path while avoiding obstacles.
- **Recoveries Server:** Handles recovery behaviours when the robot encounters issues during navigation.

Configuration and Tuning

The Nav2 setup involved several key configurations. First, the robot's physical parameters, such as dimensions, sensor placements, and kinematics, were defined in the URDF file. Sensor integration was ensured by correctly integrating the LIDAR sensor data from the LD19 package, making it accessible to Nav2. The map server was configured to load and manage the maps generated by Cartographer.

Path Planner

The Planner Server was configured with appropriate path planning algorithms, costmaps, and global/local planners to generate feasible paths. The default path planner used in Nav2 is the Smac Planner, which includes implementations of the 2D-A* algorithm. This algorithm operates on a 2D grid representation of the environment where each cell represents a potential position of the robot. The 2D-A* algorithm finds the shortest path between two points by considering both the actual cost to reach a node and a heuristic estimate of the cost to reach the goal from that node. This combination helps guide the search process efficiently. Given that my previous experimentation identified A* as a reliable algorithm, I was satisfied with leaving this as the default path planning method in Nav2.

Controller and Recoveries Servers

Additionally, the Controller Server parameters, including velocity limits, control frequency, and obstacle avoidance settings, were defined to ensure smooth and safe navigation. The Recoveries Server was configured with strategies for different types of failures, such as clearing obstacles or re-planning paths. ROS2 launch files were created and configured to initialise and run the Nav2 stack along with Cartographer. Extensive tuning of parameters, such as inflation radius, path tolerance, and recovery behaviours, was performed to optimise the robot's performance in its specific environment.

6.6 Control System for REX

Transition from Simulation to Hardware

The control system code for the quadruped robot, REX, was developed by building upon the existing code from the prior PyBullet simulation. This provided a strong foundation for implementing the real-world control algorithms necessary for REX's movements and behaviours.

6.6.1 Servo Calibration

The primary focus was on accurately controlling the servos that drive the robot's legs. Given that the simulation environment in PyBullet allowed for precise control, transitioning to real hardware

required additional effort, particularly in calibrating the motors. This was due to variances in the servos where the same command might result in different actual angles (e.g., 90 degrees on one motor did not precisely match the actual angle produced on another).

6.6.2 Calibration Program

A dedicated calibration program was therefore written to permit users to cycle through and fine-tune offsets for each motor in real-time, ensuring commanded angles matched physical angles accurately. Once calibrated, the program exports offset error to a JSON file, which the main control code reads during runtime to apply necessary adjustments, ensuring consistent and accurate movements.

6.6.3 Controller Service

A key component of the control system was the creation of a controller service that interacts with the control program for REX. This service listens to the `cmd_vel` topic in ROS2, translating the velocity commands into movements for the robot. The `cmd_vel` topic is a standard ROS2 topic used for controlling robot velocity. It typically includes linear and angular velocity components, which dictate the speed and direction of the robot's movement:

- **linear.x**: Linear velocity in the x-direction (forward/backward).
- **linear.y**: Linear velocity in the y-direction (sideways movement).
- **linear.z**: Linear velocity in the z-direction (up/down height).
- **angular.x**: Angular velocity around the x-axis (N/A).
- **angular.y**: Angular velocity around the y-axis (N/A).
- **angular.z**: Angular velocity around the z-axis (Rotation).

To accommodate additional control features, such as adjusting the robot's stationary orientation, I introduced a new topic: `orientation`. This topic allows for the control of non-dynamic aspects of the robot's orientation, specifically yaw, pitch, and roll. This complements the `cmd_vel` commands, which handle dynamic orientation during movement. The combination of these control mechanisms ensures comprehensive and flexible management of REX's behaviours and movements, both when in motion and stationary.

The main parameters for the robot's movement, which needed to be tuned, included:

- **Stride Length**: Determines the length of each step.
- **Walking Angle**: Specifies the direction of walking.
- **Rotation**: Allows the robot to rotate on the spot.
- **Gait Cycle Period**: Defines the duration of one complete gait cycle.

The implementation of this controller service was essential for integrating Nav2, as it relies on the `cmd_vel` topic to command the robot's movements. By effectively translating these commands, the service ensures that REX can follow planned paths and navigate its environment accurately.

6.6.4 Testing and Debugging

For debugging the project, another controller service was created that permitted the use of a game controller to send manual movement commands to the robot. This service not only facilitated troubleshooting and fine-tuning of the robot's movements but also provided an additional mode of operation. Users can manually control the quadruped robot by disabling the autonomous navigation, offering a flexible alternative for various applications. The controls can be seen in Figure 6.13.

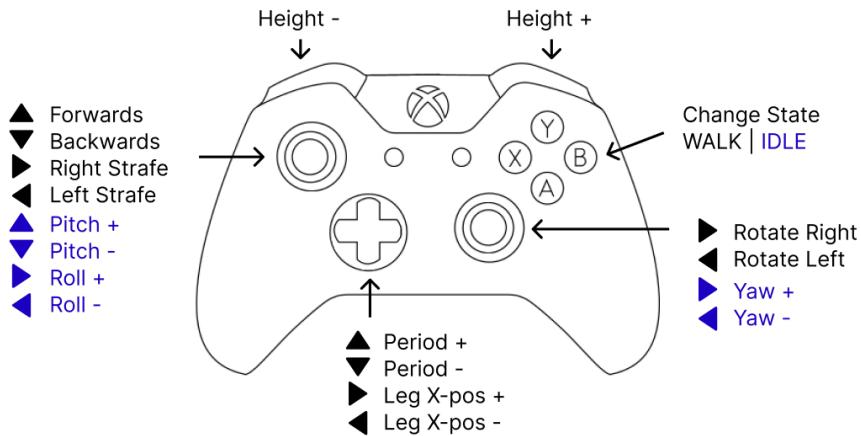


Figure 6.13: Controls for the Quadruped - Xbox Controller

6.7 Coordinate Acquisition from Depth Camera

To accurately locate people within the environment, the Oak-D Lite depth camera used and combined with MobileNetV2, a lightweight convolutional neural network. This setup enabled efficient detection and localisation of individuals based on depth information. The process involved several steps, including neural network processing, depth calculation, coordinate transformation, and tracking.

6.7.1 Detection and Depth Calculation

MobileNetV2 was used to identify objects in the camera's field of view. By limiting the detected classes to people, I ensured that the system focused solely on relevant targets. However, the framework was designed to be flexible, making it straightforward to extend detection to other objects or animals.

Once a person had been detected, the depth within the bounding box around the detected individual was calculated. The Oak-D Lite camera, which is capable of processing neural networks

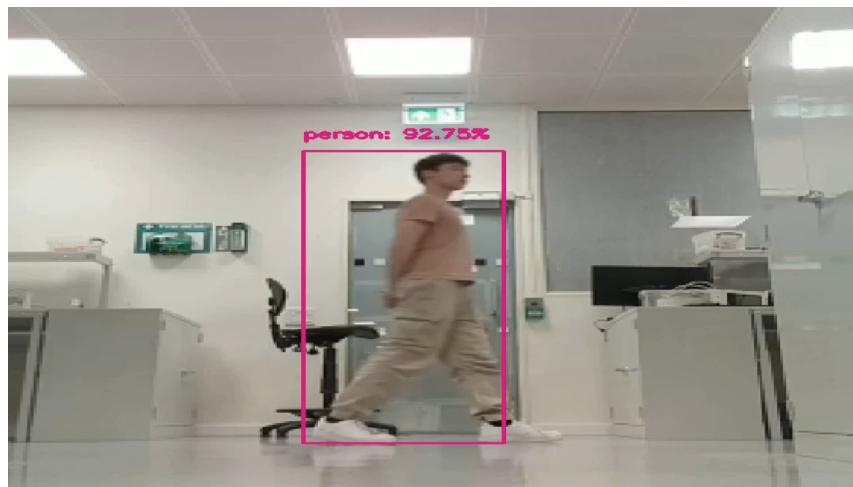


Figure 6.14: MobileNetV2 Tracking with Bounding Box

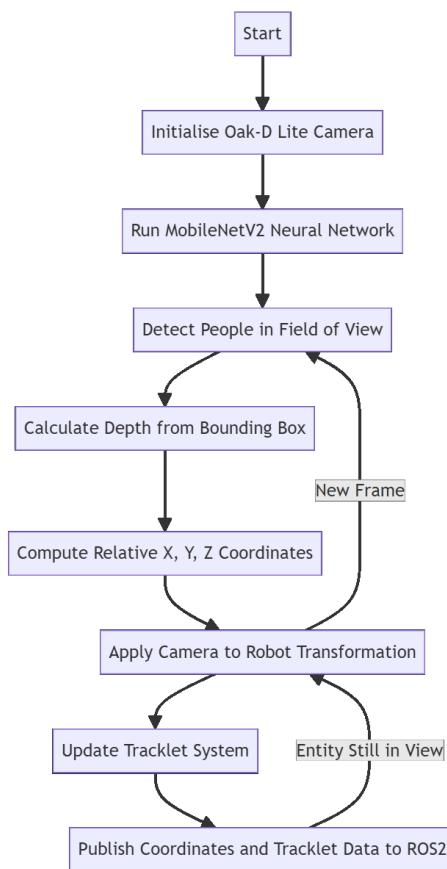


Figure 6.15: Transformation and Tracking Workflow for Depth Camera Data

onboard, handled this task efficiently, utilising up to 54% of its onboard CPU at a steady 16 FPS. This offloaded significant computational demands from the Raspberry Pi, freeing up resources for other tasks.

6.7.2 Angle and Position Calculation

Using the intrinsic properties of the camera, such as the focal length and principal point (optical center), the coordinates of the detected entity relative to the camera were determined.

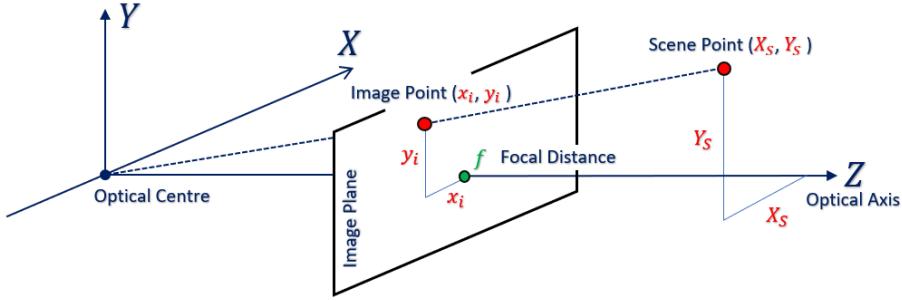


Figure 6.16: Determining Coordinates of a Point from a Scene

First, I derived the image coordinates (u, v) (centre of the bounding box) from the scene point (X_s, Y_s, Z_s) using the following projection equations:

$$\begin{aligned} u &= f \frac{X_s}{Z_s} + c_x \\ v &= f \frac{Y_s}{Z_s} + c_y \end{aligned}$$

where f is the focal distance of the camera, and c_x and c_y are the optical centers.

Given the depth Z , the scene coordinates (X_s, Y_s, Z_s) were calculated from the image coordinates (u, v) by rearranging the projection equations:

$$\begin{aligned} X_s &= Z \cdot \frac{u - c_x}{f_x} \\ Y_s &= Z \cdot \frac{v - c_y}{f_y} \end{aligned}$$

Here, f_x and f_y are the focal lengths of the camera in the x and y directions.

6.7.3 Coordinate Transformation

To transform the person's coordinates from the camera frame to the robot's coordinate frame, a transformation matrix was used. This matrix accounts for the robot's position and orientation in

the environment.

Given the camera coordinates $(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})$, the robot's translation vector $(X_{\text{trans}}, Y_{\text{trans}}, Z_{\text{trans}})$, and the rotation angle θ about the x-axis, the transformation is defined as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} + \begin{bmatrix} X_{\text{trans}} \\ Y_{\text{trans}} \\ Z_{\text{trans}} \end{bmatrix}$$

where $\begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix}$ are the coordinates in the camera frame.

6.7.4 Final Coordinates in the Robot Frame

Combining the above steps, the coordinates of the detected entity in the robot's frame (X_p, Y_p, Z_p) were obtained by first calculating the camera coordinates from the image coordinates and then transforming these camera coordinates into the robot's frame using the transformation matrix T :

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

Here, T includes both rotation and translation components as described above.

$$T = \begin{bmatrix} 1 & 0 & 0 & X_{\text{trans}} \\ 0 & \cos \theta & \sin \theta & Y_{\text{trans}} \\ 0 & -\sin \theta & \cos \theta & Z_{\text{trans}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This ensures that the detected entity's coordinates are accurately transformed from the image frame to the camera frame and finally to the robot's frame.

6.7.5 Tracking and Data Publishing

To maintain accurate location data as the robot moves, a tracklet system was implemented. This system tracks detected objects until they leave the camera's field of vision. Each new entity is assigned a unique tracklet ID and a "time alive" parameter representing the duration for which it has been tracked.

The tracklet system works by assigning a new ID to each detected person and updating their position and time alive as long as they remain in the field of view. If a person leaves the field of

view and then reappears, a new tracklet ID would be assigned. This method ensured continuous and accurate tracking.

This data, including the coordinates and tracklet information, was published to a ROS2 topic. A custom parser read this information (Figure 6.17), applied the necessary transformations, and sent the coordinate destination to the NAV2 stack. The quadruped robot then navigated to the specified location. Upon arrival, the assistant module would take over to provide further interaction or assistance.

```
rex@rex-desktop:~/ros2_ws$ ros2 run rex tracklets_parser
[INFO] [1717793833.259990134] [tracklets_parser]: Initializing TrackletsParser node...
[INFO] [1717793833.809962949] [tracklets_parser]: Subscription to /color/tracklets created.
[INFO] [1717793835.244769855] [tracklets_parser]: Position - x: -0.46, y: -0.00, z: 1.89 - Tracking(17): No
[INFO] [1717793835.495780911] [tracklets_parser]: Position - x: -0.56, y: -0.00, z: 2.27 - Tracking(17): No
[INFO] [1717793835.829292559] [tracklets_parser]: Position - x: -0.74, y: 0.02, z: 2.84 - Tracking(17): Yes
[INFO] [1717793835.859587114] [tracklets_parser]: Position - x: -0.75, y: 0.05, z: 2.84 - Tracking(17): Yes
[INFO] [1717793836.044961355] [tracklets_parser]: Position - x: -0.77, y: 0.02, z: 2.84 - Tracking(17): Yes
[INFO] [1717793836.177741503] [tracklets_parser]: Position - x: -0.84, y: -0.01, z: 3.09 - Tracking(17): Yes
[INFO] [1717793836.537431280] [tracklets_parser]: Position - x: -1.01, y: -0.06, z: 3.40 - Tracking(17): Yes
[INFO] [1717793836.603196503] [tracklets_parser]: Position - x: -1.20, y: -0.09, z: 3.78 - Tracking(17): Yes
```

Figure 6.17: ROS2 Tracking Data

The tracklet system ensures that the quadruped maintains accurate positional information, allowing for effective navigation and interaction within dynamic environments. By leveraging the onboard processing capabilities of the Oak-D Lite and the computational resources of the Raspberry Pi, the system was able to demonstrate efficient and reliable performance in test scenarios.

6.8 Autonomous Navigation Implementation

Autonomous navigation for the quadruped was achieved using a ROS2 package that facilitates frontier-based exploration, based on methodologies presented in a thesis on autonomous exploration [40]. Frontier-based exploration is a strategy in robotics where the robot identifies and navigates towards the boundaries between known and unknown areas to systematically expand its mapped environment. The core functionality revolves around processing occupancy grid maps, performing path planning using the A* algorithm, and executing the navigation commands from NAV2 to explore unknown areas.

6.8.1 Initialisation and Parameter Loading

The navigation node begins by subscribing to essential topics such as the occupancy grid map, odometry, and laser scan data. Parameters for the exploration algorithm, such as lookahead distance, speed, expansion size, and target error, are loaded from a configuration file.

6.8.2 Map and Odometry Processing

The occupancy grid map provides information about the environment, including obstacles and free spaces. The robot's current position and orientation are obtained from the odometry data, which is essential for accurate navigation and path planning. The map data is processed to expand obstacles by a specified size to ensure safe navigation margins.

6.8.3 Frontier Identification and Grouping

Frontiers are identified by examining the occupancy grid map for cells that are adjacent to unknown areas. These frontier cells are then grouped into clusters based on their proximity to each other. The groups are ranked based on their size, and the top groups are selected for further processing.

6.8.4 Path Planning Using A* Algorithm

The A* algorithm is used for path planning between the robot's current position and the centroid of the selected frontier groups. A heuristic function, in this case the Euclidean distance, guides the search to find the shortest path to the target. If no path is found, the algorithm selects the closest feasible point to the target as the new goal.

6.8.5 Path Smoothing with B-Splines

To ensure smooth navigation, the planned path is refined using B-spline interpolation. This technique generates a smooth curve that approximates the waypoints along the path, resulting in more fluid and natural movements for the robot.

6.8.6 Pure Pursuit Path Tracking

The pure pursuit algorithm is employed for path tracking. This method calculates the desired steering angle and velocity based on the current position and heading of the robot relative to the next waypoint on the path. The robot continuously adjusts its movement to follow the planned path accurately.

6.8.7 Local Obstacle Avoidance

Local obstacle avoidance is integrated into the navigation system to handle dynamic obstacles detected by the laser scanner. If an obstacle is detected within a specified radius, the robot temporarily adjusts its path to avoid the collision and resumes its original path once the obstacle is cleared.

6.8.8 Exploration Loop

The exploration process operates in a loop where the robot continuously seeks new frontiers, plans paths, and navigates towards them. Upon reaching a target frontier, the exploration function is triggered again to find the next frontier, ensuring continuous and comprehensive exploration of the environment.

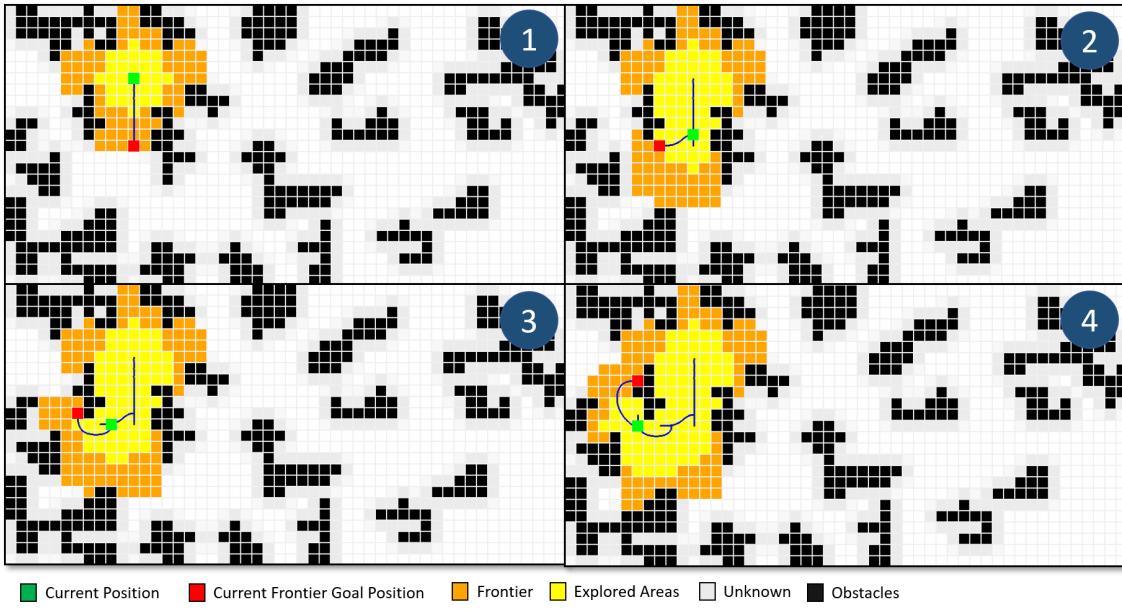


Figure 6.18: Example: Detection of Frontier Goals

6.8.9 Wavefront Exploration Simulation

The previous program, which simulated path planning, was modified into simulating wavefront exploration. The updated Python program simulates a wavefront and selects a frontier based on factors like closeness and cluster size. The simulation worked effectively, demonstrating the path a robot would take during exploration. Below is an image of the program (Figure 6.19) and various stages of the exploration process can be seen in Figure 6.18.

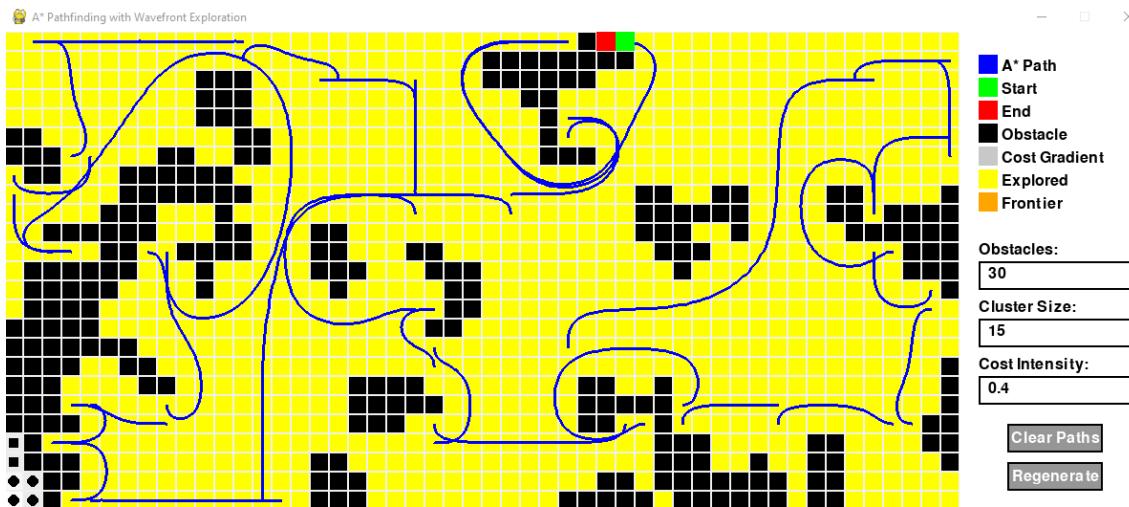


Figure 6.19: Path of Wavefront Exploration - Simulation

The simulation was particularly useful for quickly testing and tuning parameters, allowing the algorithm to be efficiently refined. To ensure that the simulation's exploration was realistic, especially when determining the visibility of cells in the presence of obstacles, the Bresenham's line algorithm was used to simulate raycasting. This algorithm helps to check if there is a clear

line of sight between two points on the grid. The Bresenham's line algorithm calculates the set of points that form a straight line between the start and end coordinates, considering integer-based grid coordinates.

During the wavefront exploration, the *bresenham_line* function is used to trace the path of a "ray" from the robot's current position to each cell within its detection radius. If any point along this ray encounters an obstacle, the target cell is considered not visible. This ensures that the robot's field of view accurately respects the presence of obstacles, preventing the detection of cells that are hidden behind walls or corners.

By incorporating this visibility check, the wavefront exploration simulation can more accurately simulate how a robot would explore an environment, taking into account realistic line-of-sight limitations. This improvement enhances the realism of the simulation and allows for more effective testing and development of exploration strategies.

6.9 WatsonX Assistant

The WatsonX Assistant was integrated into the quadruped robot as the conversational and diagnostic backbone of the system, designed to assist individuals in need of medical help. The system included several key components: wake word detection, speech-to-text, text-to-speech, and interaction with IBM's Watson Assistant. This section details the integration and functionality of these components, highlighting their roles and interactions within the overall system.

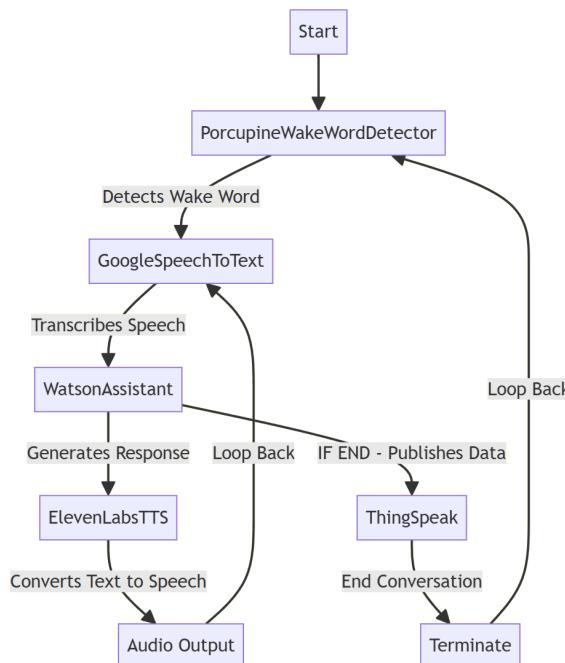


Figure 6.20: Assistant Control Flow

6.9.1 Wake Word Detection

A wake-word detector was incorporated into the system and this is managed by the `PorcupineWakeWordDetector` class within the `wakeWordDetection.py` script. Porcupine is a wake word detection utility, developed by *Picovoice*, which continuously listens for a specific keyword that activates the robot's listening mode.

- **Initialisation:** The detector is first initialised with specific keyword paths and sensitivity settings. Two key words (or phrases) were implemented for testing: "Hey, Rex..." & "Help me!"
- **Listening Mode:** It listens for the wake word and upon detection, plays a notification sound to indicate that it is ready to receive commands.

6.9.2 Speech-to-Text Conversion

Once the wake word is detected, the `GoogleSpeechToText` class (also within `wakeWordDetection.py`) takes over to convert spoken commands from the user into text. The system used Google's Speech-to-Text API to transcribe spoken words into text. This transcribed text served as the input for further processing.

6.9.3 Interaction with Watson Assistant

The core interaction with the user is handled by the `WatsonAssistant` class in `watson.py`. This class leveraged IBM's Watson Assistant to process the transcribed text and generate appropriate responses.

- **Session Management:** The assistant manages conversation sessions, ensuring context was maintained throughout the interaction.
- **Data Processing:** It checks for specific data points related to medical conditions, which are then published to a ThingSpeak channel if all required data is present.
- **Response Generation:** Watson Assistant processes the input text and generates a suitable text response based on its training and predefined actions. Example training data can be found in Figure 6.21.

6.9.4 Text-to-Speech Conversion

The generated response from Watson Assistant is then converted to audio using the `ElevenLabsTTS` class defined in `text2speech.py`. ElevenLabs is a text-to-speech platform that uses artificial intelligence to generate high-quality, natural-sounding speech from text input. This was found to be exceptionally useful, as it enables a friendlier and more natural interactions with end-users.

I'm having chest pain and sweating	I'm experiencing severe chest pain	I need help my chest hurts
I have a stabbing pain in my chest	I have discomfort in my chest	Chest pain help
I'm feeling a heaviness in my chest	There's a burning feeling in my chest	I'm experiencing chest pain
I have chest pain and shortness of breath	I'm having a squeezing sensation in my chest	I have pain in my chest
My chest pain is spreading to my arm	My chest pain won't go away	My chest hurts

Figure 6.21: Example Prompts - Recognising Medical Emeergencies

To reduce latency when waiting for audio files to generate, a caching mechanism was implemented. Each audio file that is generated is stored and hashed using the text as the key. This hash is used to quickly retrieve the audio file from the cache if the same text needs to be converted again. By reusing previously generated audio files, this approach eliminates the need to generate a new audio file each time, thereby significantly reducing the waiting time from about 3 seconds to virtually instantaneous retrieval. This method proved to be extremely effective, since there are a fixed number of phrases the assistant can provide.

6.9.5 System Integration

The overall system was designed to work in a cohesive manner, allowing the quadruped robot to effectively assist and diagnose individuals, providing a valuable tool in scenarios where immediate medical assistance might be required. The entire code for the assistant was encapsulated within a ROS package, enabling integration with the rest of the existing ROS framework.

6.10 Website Dashboard

The website dashboard is an important component of the REX system, providing an interface for managing rescue operations, deploying units, and monitoring real-time data through an intuitive web application. The dashboard was designed to offer navigation and control over various functionalities, ensuring efficient and effective rescue and exploration missions.

6.10.1 Overview

The dashboard consists of three main sections: Cases, Deployments, and Control. Each section serves a specific purpose and is accessible through the navigation menu at the top of the page.

Cases Section

The Cases section provides an overview of all recorded rescue cases. It includes a detailed table listing the time of the incident, the name of the person involved, age, type of incident, specific

details, and severity. The table, seen in Figure 6.22 is interactive, allowing users to click on individual cases, and see their location on a map.

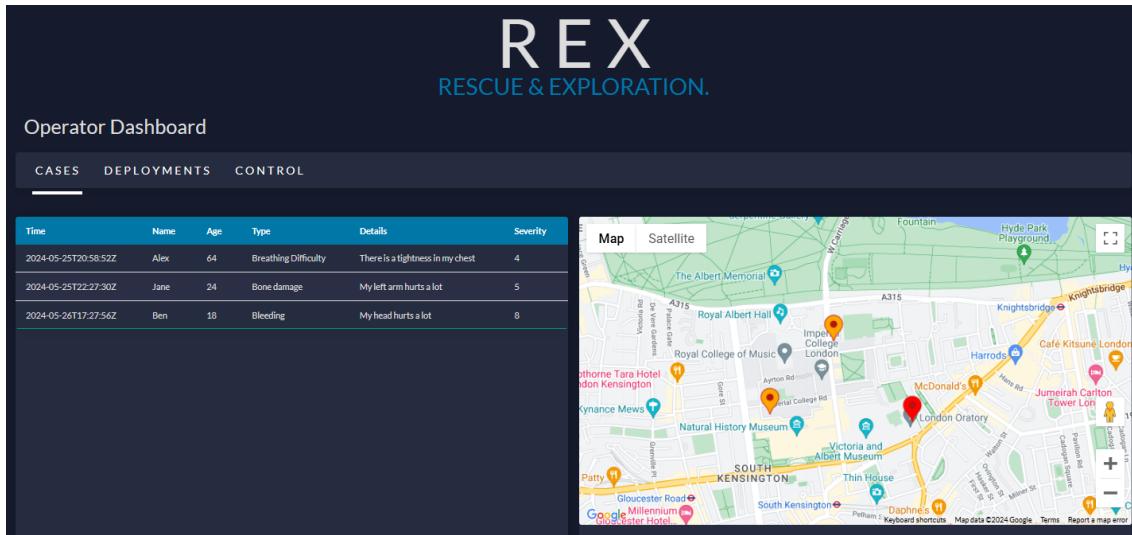


Figure 6.22: Website Dashboard - Case View

Deployments Section

The Deployments section manages the deployment of rescue units. It features a table displaying the ID, name, location (coordinates), temperature, time, and IP address of each unit. Additionally, a map on the right side helps to visualise the real-time location of each unit, enhancing situational awareness and coordination.

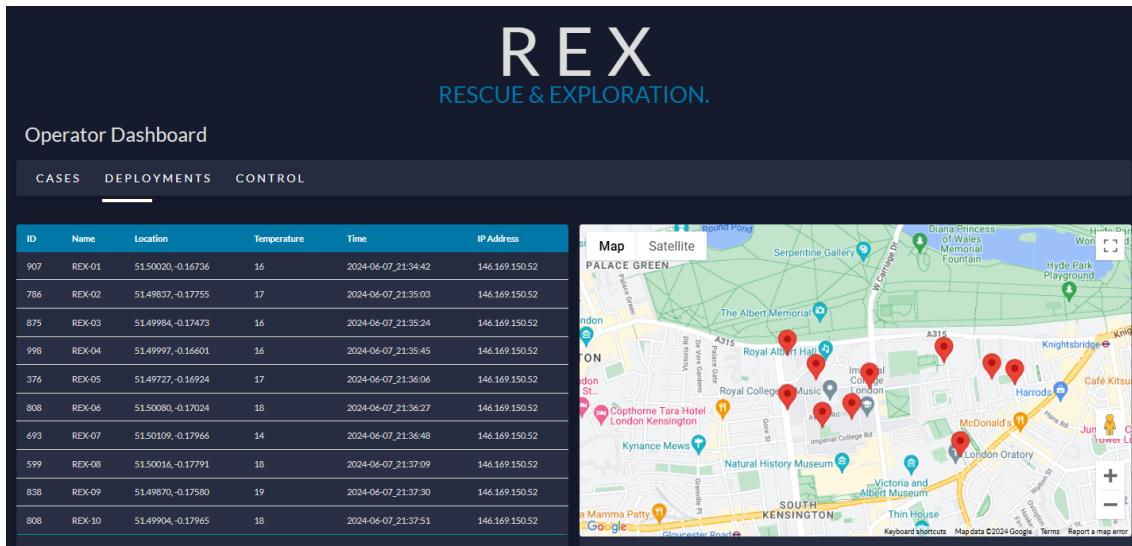


Figure 6.23: Website Dashboard - Deployments View

Control Section

The Control section offers functionalities for managing the deployed units. It includes a table with the ID, name, location, and IP address of each unit, along with a control interface for operating the units. Users are able to toggle between autonomous and manual modes and use directional controls to move the units. A live video feed section capability is included, however, this has not yet been fully integrated with the system.

ID	Name	Location	IP Address
136	REX-01	51.49904, -0.17668	146.169.150.52
885	REX-02	51.49823, -0.15847	146.169.150.52
385	REX-03	51.49710, -0.16727	146.169.150.52
952	REX-04	51.50049, -0.17014	146.169.150.52
897	REX-05	51.50137, -0.16750	146.169.150.52
925	REX-06	51.49914, -0.16650	146.169.150.52
605	REX-07	51.49864, -0.17769	146.169.150.52
546	REX-08	51.49904, -0.17870	146.169.150.52
434	REX-09	51.50087, -0.16627	146.169.150.52
688	REX-10	51.49718, -0.17800	146.169.150.52

Figure 6.24: Website Dashboard - Control View

6.10.2 Key Components

The dashboard was built using several key components, each designed to handle specific tasks and data visualisation:

GoogleMap Component (GoogleMap.js)

The GoogleMap component integrates Google Maps to visualise data entries. It uses markers with color-coded pins based on case severity and moves the map centre and zoom level to highlight selected entries.

RobotMap Component (RobotMap.js)

Similar to GoogleMap, the RobotMap component focuses on robot deployments. It highlights selected markers with specific styles and provides an interactive map interface for monitoring robot locations.

ThingSpeakTable Component (ThingSpeakTable.js)

This component fetches data from the ThingSpeak API and displays it in an interactive table. It includes functionalities for sorting and selecting rows, allowing users to interact with and manage data entries effectively.

Control Component (Control.js)

The Control component handles data fetching and displays deployment information in a table format. It includes a control component, which provides directional control buttons (up, down, left, right) and includes a section for displaying a live video feed from the robots.

Dashboard Component (Dashboard.js)

This is the main component of the operator dashboard, managing state and rendering different tabs based on user interaction. It integrates all other components and handles the overall layout and functionality of the dashboard.

6.10.3 Functionality and User Experience

The website dashboard was designed for ease of use. The interactive tables and maps provide real-time data visualisation and control, allowing operators to manage rescue operations effectively. The clear and intuitive interface ensures that users can quickly access the information and controls they need, enhancing the overall user experience.

6.10.4 Conclusion

The website dashboard for the REX quadruped provides a powerful and user-friendly interface for managing rescue operations and robot deployments. Its integration of real-time data visualisation, interactive controls, and comprehensive information management makes it a useful tool for rescue and exploration missions. The use of modern web technologies and APIs ensures that the dashboard is robust, reliable, and capable of handling complex operational requirements.

6.11 Final Overview

To provide a clear overview of the software system architecture and its interactions, the following diagram (Figure 6.25) illustrates the various states and transitions within the system. This diagram encapsulates the flow of operations from initialisation to the autonomous navigation, human interaction, and remote monitoring functionalities. It also highlights key components such as SLAM, obstacle avoidance, and frontier exploration within the autonomous navigation module, as well as the web application interface for remote access and control.

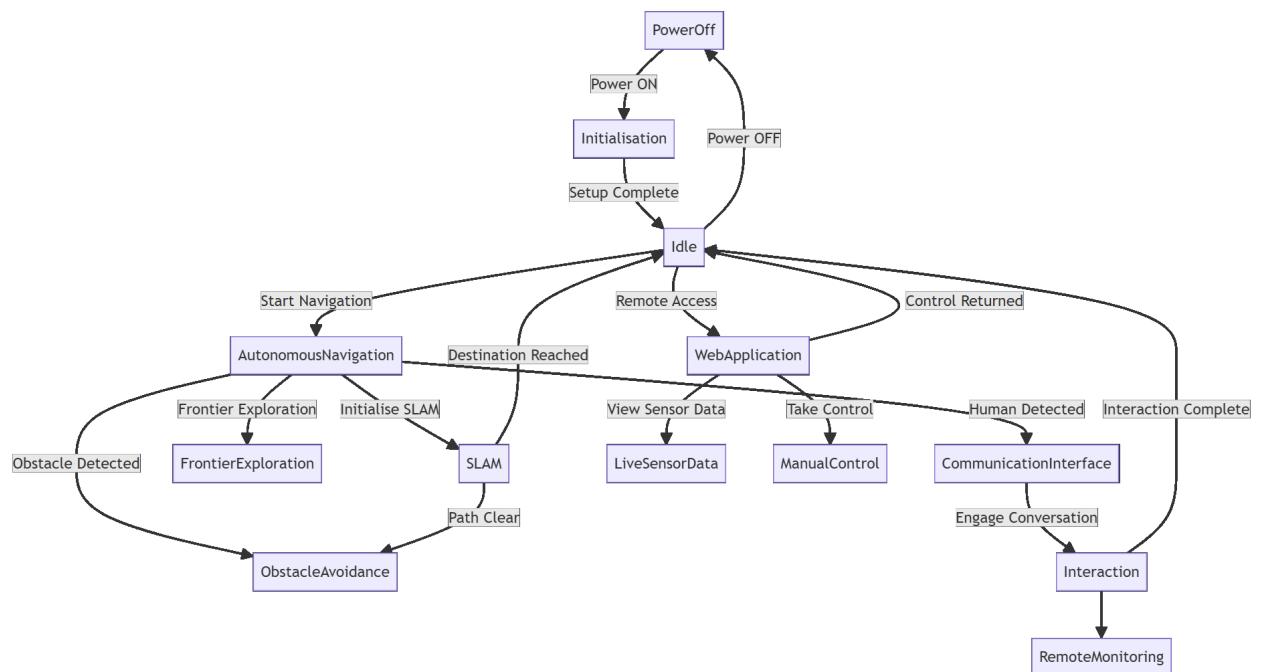


Figure 6.25: Software System Architecture Diagram

7

Project Planning & Evaluation Strategies

Contents

7.1	Project Timeline and Tasks	117
7.1.1	Key Milestones	120
7.2	Evaluation of Project Success	120
7.3	Safety Consideration	120
7.3.1	Lab Access and Safety Protocols	120
7.3.2	Risk Assessment	120
7.4	Ethical Considerations	121
7.4.1	Safety and Reliability	121
7.4.2	Privacy and Data Protection	121
7.4.3	Dual-Use Considerations	121
7.4.4	Accountability in Autonomous Actions	121
7.4.5	Socio-Economic Impacts	122
7.4.6	Human-Robot Interaction	122
7.5	Environmental Considerations	122

7.1 Project Timeline and Tasks

The project time-line is shown in the Gantt chart in Figure 7.1. Due to the resources being limited to one person, its nature is rather linear for each task, although many tasks did overlap.

The work was broken-up in to tasks under the main headings shown below, each with a number of sub-tasks, detailed below.:

Project Management: The chart formed the basis for the project management and it was reviewed roughly every 2 weeks. Due to the nature of development work, some tasks continued further to the right than was originally planned.

Background Research: This task looked at previous applications of robotics to disaster management and recovery situations and explored their limitations and required improvements.

It also looked at the economic impact of recent disasters to make an economic case for low-cost solutions.

Requirements definition: A list of requirements was compiled as a design basis for engineering the quadruped.

Mechanical Design: This task focused on developing the mechanical requirements, particularly trying to identify and implement the most suitable limb kinematic arrangement. It included the detailed chassis design and choice of materials, as well as detailed design and simulation of the limb kinematics. Following a design-review, the manufacturing CAD files were prepared for the manufacturing stage to follow.

Electrical Design: This task partly overlapped the Mechanical Concepts, particularly as the motor selection went hand-in-hand with the limb design. The overall electrical concept was developed and refined into a specification for the selection of the required components. These included those for the power-system (comprising of battery, charger, main DC-DC converter and drives for the motors), as well as the those for the primary controller system and its associated sensors.

Purchasing Components: This task involved the production of a Bill of Materials (A.2) and the staggered purchasing of the required components, while reviewing the cost against the budget.

Manufacturing: This task centred around the 3D printing of the parts required for the chassis and limb assemblies of the robot. Some initial trials were necessary to establish the best machine settings (hatch distance and speed) for each of the materials to be used. The task continued throughout the most of the project as during testing, a number of components was found to require design modification.

Assembly: This task involved the mechanical assembly of the chassis and its motor and limb sub-assemblies. This was followed by assembly of the battery pack and the integration of main power components (charger, buck-converter and drives). The battery pack itself had first to be assembled with its power management system. Finally, the sensor and main controller was added.

Initial Testing: In this task, the main components were tested using manual control of the motors and limbs. It was during this stage that some short-comings were identified, principally in a couple of the limb components and in the LIDAR compatibility.

Software Development: Once the robot had been assembled with its supporting electronics, the software development stage could begin. This task was partly in parallel with the initial testing, which had identified some issues potentially affecting the software. The key capability features of the robot were developed in this task, including the various algorithms required for autonomy and the integration of AI.

Final Testing and Refinement: The robot was put through a number of real and simulated test scenarios to test and refine the performance of the hardware and software.

Reporting: This task involved compiling information and preparing the interim and final reports

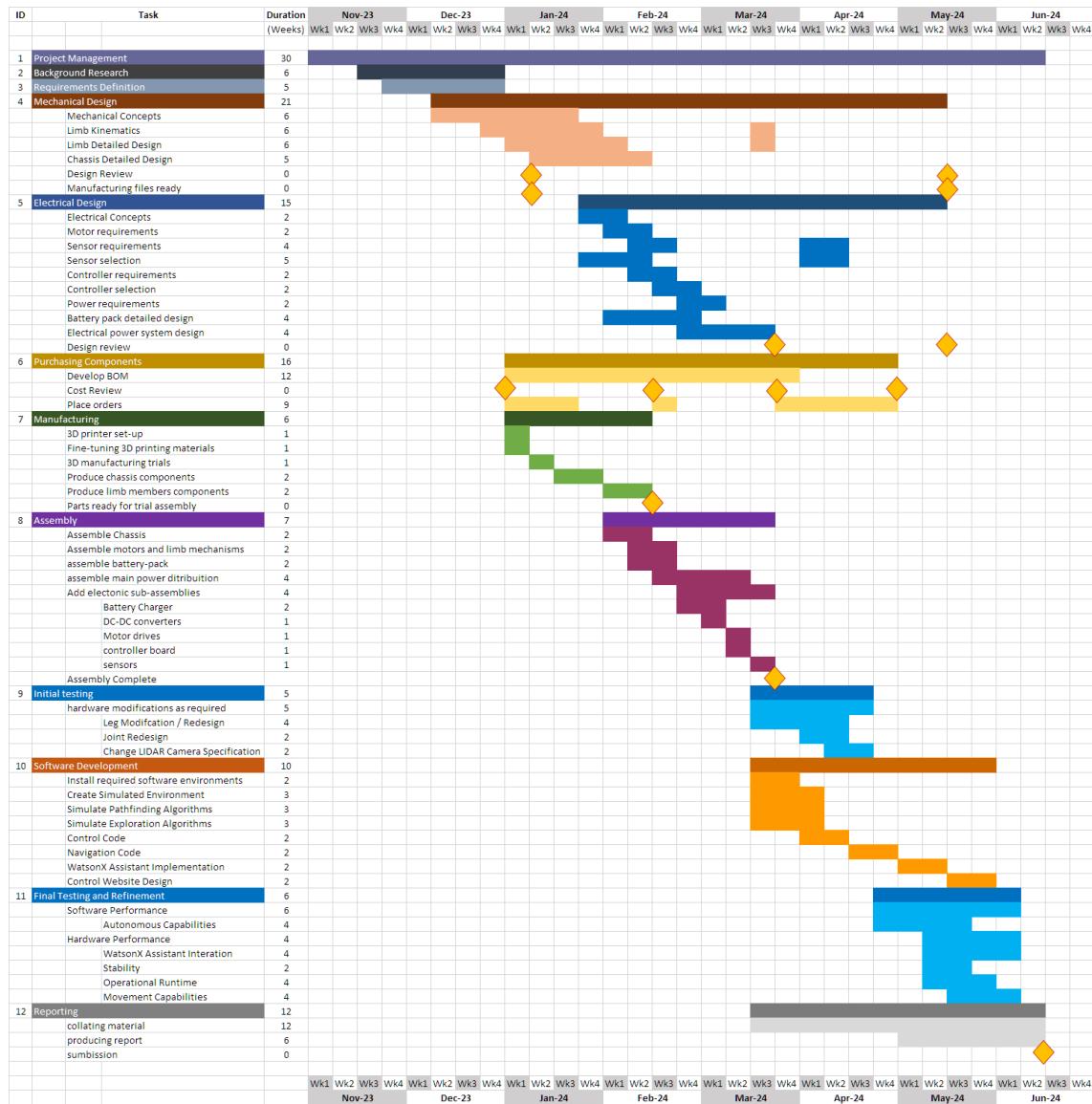


Figure 7.1: Project Gantt Chart

7.1.1 Key Milestones

The key milestones are shown in the Gannt Chart. Because some of the tasks were running in parallel, there were few definite Milestones.

The completion of the Requirements Definition would normally signal the beginning of the concepts design stages, but in practise there was some overlap in these tasks.

A major milestone was the completion of the initial mechanical design phase, which permitted the start of manufacturing. The next milestone was having the parts ready for assembly, followed by another for completing the hardware-build of the robot. This last milestone enabled the software development to start in earnest.

A further milestone signals the completion of this report.

7.2 Evaluation of Project Success

Please see the Conclusions in the next chapter for a discussion.

7.3 Safety Consideration

Ensuring safety throughout the development and testing phases of this project was of great importance.

7.3.1 Lab Access and Safety Protocols

Authorised access to the laboratory facilities was granted for the duration of this project. These facilities were equipped with the necessary tools and equipment for the design, assembly, and testing of the quadruped robot. Prior to commencing work in the lab, a thorough orientation was conducted for familiarisation with the lab's safety protocols, emergency procedures, and the proper use of equipment.

7.3.2 Risk Assessment

A detailed risk assessment was completed to identify potential hazards associated with the project activities. This assessment covered a broad range of risks, including electrical safety, mechanical operations, and potential chemical exposures. Each identified risk was evaluated for its severity and likelihood, leading to the development of tailored mitigation strategies. Particular regard was placed on the handling and assembly of the Lithium Ion battery packs.

7.4 Ethical Considerations

The development and deployment of autonomous quadruped robots in disaster response and other critical applications raised several ethical considerations around responsible innovation and application of this new technology. These considerations included safety, privacy, accountability, and the socio-economic implications of integrating such technology into human-centric operations.

7.4.1 Safety and Reliability

Ensuring the safety and reliability of autonomous robots was paramount, particularly when operating in disaster-stricken areas where human lives may be at risk. The robot is equipped with fail-safe mechanisms and decision-making algorithms designed to be robust to prevent malfunction or unintended harm. Ethical design principles necessitated rigorous testing under diverse conditions to ascertain the robots' performance and to minimize the risk of failure in critical situations.

7.4.2 Privacy and Data Protection

The use of sensors and cameras, essential for navigation and task execution, involved the collection of potentially sensitive data, including facial recognition information. It was necessary to implement stringent data protection measures to safeguard the privacy of individuals and to comply with relevant GDPR legislation. Ethical deployment required transparent data handling policies, ensuring that all collected data was used solely for humanitarian purposes and was protected against misuse.

7.4.3 Dual-Use Considerations

The development of an autonomous quadruped robot, while primarily aimed at disaster response and humanitarian assistance, inherently carries the potential for dual-use applications, including military deployment. This duality necessitated a thorough examination of ethical implications, responsible use, and the establishment of robust oversight mechanisms to ensure that the technology was used in a manner consistent with international laws and ethical standards. The potential military use of autonomous systems raised significant ethical concerns, particularly regarding the decision-making autonomy in conflict situations. It would be important to define clear operational boundaries for the robot's use in military contexts, ensuring that any deployment strictly adhered to the principles of distinction, proportionality, and necessity as outlined in the laws of armed conflict.

7.4.4 Accountability in Autonomous Actions

With the robot's ability to make decisions autonomously, establishing accountability frameworks was essential, especially in dual-use scenarios where the stakes could be considerably higher. This involved creating transparent protocols for decision-making processes, ensuring that there were

always clearly defined human oversight and intervention capabilities to prevent unintended or unethical use.

7.4.5 Socio-Economic Impacts

The adoption of advanced robotics in disaster response also entailed socio-economic considerations, such as the potential displacement of human workers and the accessibility of such technology across different regions and economic strata. Ethical development should be inclusive, ensuring that the benefits of robotic assistance in disaster management are equitably distributed and that the technology complemented human efforts rather than replacing them.

7.4.6 Human-Robot Interaction

The dynamics of human-robot interaction in high-stress environments like disaster zones required careful consideration. The design and operation of robots must prioritise empathy and support, ensuring that their presence and actions did not add to the distress of affected individuals. This involved user-friendly interfaces, predictable behaviors, and the inclusion of features that facilitated positive interaction and collaboration between humans and robots.

In summary, addressing these ethical considerations requires a multidisciplinary approach, involving stakeholders from diverse fields, including robotics, ethics, law, social sciences, and disaster management. By embedding ethical considerations into every stage of the design, development, and deployment processes, the project aimed to contribute positively to disaster response efforts, enhancing resilience and safety while upholding the highest ethical standards.

7.5 Environmental Considerations

Lastly, the environmental impact of manufacturing, deploying, and end-of-life disposal of such robotic systems was considered. Ethical practices involve minimising the environmental footprint of robots, from sustainable manufacturing processes to the recyclability of components.

During its use the robot requires only electrical power, which arguably could be from clean energy. The robot does not require any fuels nor liquids that could impact the environment

A variety of materials were used in the construction of the robot. While the use of Additive Manufacturing has greatly limited the material waste during manufacture, there is still a the consideration of disposal and recycling at the end of its useful life.

A list of the main materials/components is given in the table, below:

Most of the plastic components used in the robot are readily recyclable through long-established processes. Equally, this applies to all the metal components.

The proliferation of carbon-fibre reinforced composites used in the aerospace industry have increased the amount of waste of this material, which has led to the development of new processes

Table 7.1: Recycling and Disposal Information

Assembly	Material	Recycling / Disposal
Robot Chassis & panels	Carbon Fibre Composite Sintered Neodymium magnets Steel screws Brass inserts Nylon PLA PLA +	The carbon-fibre can be recovered from the epoxy-resin matrix. Recoverable and recyclable with hydrogen processing Readily recovered and recycled Need to separate from thermoplastic plastic material using heat Readily recycled Biodegradable Biodegradable
Battery Pack	Lithium Cells PLA PLA + Nickel strips Tin solder	Recyclable Biodegradable Biodegradable Readily recovered and recycled Readily recovered and recycled
Wiring	Copper PVC	Readily recovered and recycled Waste
Motors	Electrical, silicon, steel Carbon-steel Nylon Copper Sintered Neodymium magnets	Readily recovered and recycled Readily recovered and recycled Readily recovered and recycled Readily recovered and recycled Recoverable and recyclable with hydrogen processing
Control boards	Glass-fibre circuit board Copper Tin solder Electronic components	Waste Readily recovered and recycled Readily recovered and recycled Waste
Sensors	Glass-fibre circuit board Copper Tin solder Electronic components	Waste Readily recovered and recycled Readily recovered and recycled Waste

for its recovery. Most of the processes involve shredding the material and then separating the carbon-fibre from the epoxy resin through various chemical processes [41]. The carbon-fibre can then be re-used in new products.

The recycling of electronic waste is costly and time-consuming. For circuit boards there are a number of options [42]. At the simplest level, valuable components can be de-soldered and re-used [43], while electrochemical, hydro-metallurgy or smelting process can be used to recover base-metals

Lithium-ion batteries consist of a number of parts including an anode, a cathode, a separator, and an electrolyte. The anode is typically manufactured from graphite, the cathode consists of a metal oxide, and the separator is a thin polymer film that prevents the electrodes from touching each other, while the electrolyte is a liquid or gel that allows ions to move between the electrodes. These components are stacked or rolled together and placed in an outer packaging, which is normally either a steel can or an aluminium/polymer pouch material.

A number of new industries are emerging to deal with the recycling of these batteries or, rather, their materials. Figure 7.3, below shows a process developed by a US company, Elcan industries.

There is, however, as identified by the US Environmental Protection Agency, greater focus being placed on either re-purposing or re-using, where the battery may still be viable for another

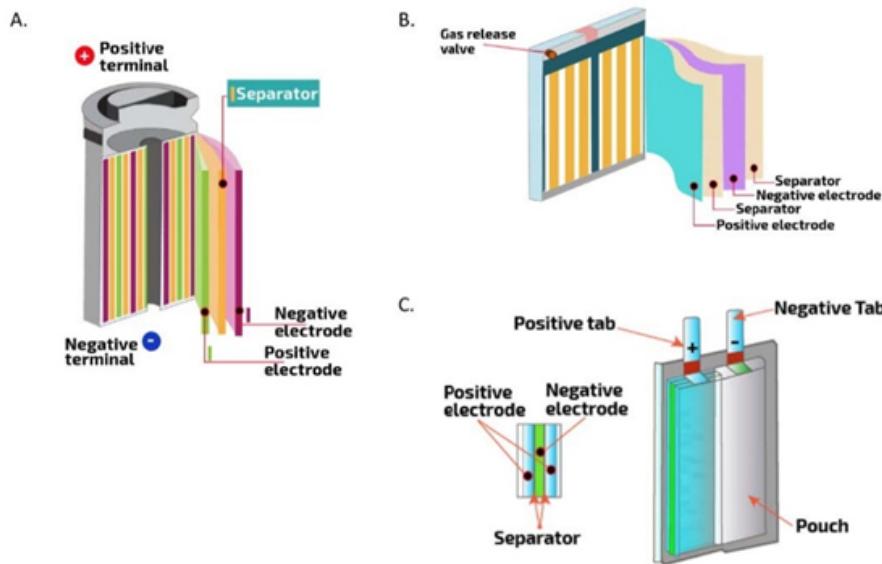


Figure 7.2: Schematics of various form factors for Li-ion cells, including: A) cylindrical, B) prismatic, and C) pouch. [Source: Sandia National Laboratory]

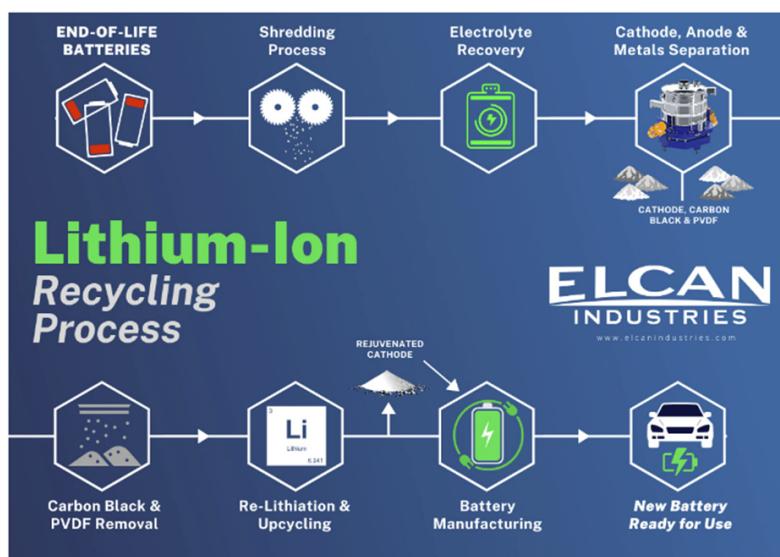


Figure 7.3: Example of Lithium-Ion battery recycling process [Source: Elcan Industries]

application, different from its original one.

Neodymium Iron Boron (NdFeB), rare-earth, magnets account for around 2/3 of the permanent magnet market and nowadays are almost as ubiquitous as Lithium Ion batteries.

Most of the magnets produced are of the sintered form and there are a number of methods [44] available to recover the valuable rare-earth materials. Because of their magnetised state, recycling can be problematic as they will attach themselves to the shredder. A initial heat-treatment is therefore required to demagnetise them (above their Curie temperature) and in most cases manual dismantling to remove them from the original motor or device.

An alternative method (hydro-metallurgy [45]) involves the use of a hydrogen reducing atmosphere, in a low-temperature flash-furnace, where the temperature effectively demagnetises the scrap magnets and H₂ atmosphere reduces the component oxides back to dust, for recycling.

Conclusions

7.6 Conclusions and Evaluations

The overall conclusion is that the project has successfully demonstrated the possibility of engineering a low-cost, quadruped robot. There were several technical areas covered in this work and we can assess each of them for individual conclusions.

Overall Problem Statement. Having reviewed the role of other robots in disaster recovery and management situations, it was clear that a quadruped design potentially offered greater agility over difficult terrain than, say, tracked vehicles that had been used in the past. Additionally, the limitations of a tethered system, either through wires or a radio-link, had to be overcome with autonomous capability to detect objects and situations, and respond accordingly. The integration of AI completes the requirements in providing an 'intelligent' link with potential disaster victims.

Mechanical Design The mechanical design of the quadruped robot was targeted to low-cost manufacture, ease of maintenance and light weight. These objectives were met with the careful design of parts and the use of Additive Manufacturing, which permitted complex part profiles to be easily produced and iteratively improved, as required, following experience from initial trials. For flexibility, CAD was used for the design of all the component and FEA was applied to validate the design for certain parts, ensuring the robot is adequately robust.

Electrical Design The objective here was not to 're-invent the wheel,' but to make use of COTS components where possible. Fortunately, robotics is a growing area of interest and generally it was easy to commercially source the required parts and components. The main controller was based on the ubiquitous Raspberry Pi platform to try to ensure continuity of supply and future upgrade paths. Many of the selected sensors provided standard interfaces both with respect to hardware and software, making integration straightforward in most cases.

Software Design The key elements of the software include the control system for locomotion; the interpretation of sensor data for the detection of objects and people; the use of this data in various algorithms for autonomous operation; the integration of IBM WatsonX AI; communication with potential victims; and the overall control and monitoring environment. All of these elements were successfully integrated and are operational. Some of them, such as the autonomous operation, require further testing and fine-tuning.

Limitations The inability to obtain accurate odometry data from the IMU's magnetometer, significantly impacted the robot's mapping performance. This means that over time, small errors in localisation calculations add up to large deviations from the true position. In the absence of the magnetometer data, the LIDAR can provide localisation through AMCL. However, this function would become inoperative if the environment were to be barren without distinctive features. Such situations could occur in open fields, featureless corridors, snow covered areas and deserts, some

of which, may be found in disaster recovery scenarios. Unfortunately, due to time constraints, the gyroscopic data from the IMU has yet not been fully integrated into the robot's stability algorithm. This is not expected to be a difficult implementation, and would likely require several days further work to code and test. This functionality would be particularly important to improve balance when walking, especially on uneven terrain.

7.7 Performance Evaluation

The robot was tested in various environments to evaluate its performance and reliability. The primary areas of focus included its ability to navigate outdoor terrain, accurately map indoor environments, and approach and interact with people.

7.7.1 Outdoor Navigation

The SLAM system proved mostly effective in creating real-time maps of the surroundings, enabling the robot to plan and follow efficient paths. The obstacle avoidance system was particularly effective, allowing the robot to detect and circumvent obstacles such as rocks and tree roots. However, as previously mentioned, the AMCL localisation would often fail when the environment lacked distinguishable features. This was seen when navigating on the lawn at Imperial College, in which the LIDAR's maximum range of 12m, would often fall short of detections.

7.7.2 Human Interaction

The robot's ability to approach and interact with people was evaluated in controlled indoor environments with fellow colleagues. The robot was able to detect humans using its integrated sensors and approach them smoothly. The communication interface was tested by engaging in simple interactions, where the robot would navigate towards people and ask questions, such as "Do you require any assistance?" These tests confirmed that the robot could effectively recognise and approach individuals, demonstrating its potential as a valuable tool and proof of concept in rescue scenarios where human detection and interaction are useful.

7.7.3 Overall Performance

The evaluation demonstrated that the quadruped robot performs reliably in indoor settings, whilst needing improvement for outdoor mapping. The autonomous navigation system, coupled with effective obstacle avoidance and mapping capabilities, ensures that the robot can operate efficiently in some environments. The ability to detect and approach humans adds a necessary functionality for rescue missions, enhancing the robot's overall utility. However, there are further improvements to be made, which are discussed in the next section.

7.8 Future Work

Whilst this project can be considered successful in the development of a first prototype, much was learned during the process, which would result in an improved, MkII, design. Some of the considerations are listed below:

Larger Size: Whilst adequate as a development platform, the size of REX is realistically too small for real-world disaster scenarios. REX MkII should be built larger to be at least able to cope with 20 mm step heights. Additionally a larger robot, would also be able to carry payloads such as medical kits or supplies. Of course, bigger size will translate into more cost, since the motors, drive systems and battery packs will also need to be scaled-up accordingly, and the chassis and leg design will need to be mechanically stronger.

Materials: For harsh environments, REX MkII would benefit from more robust materials. This could include the more extensive use carbon-fibre composites or light-weight magnesium or aluminum alloys, suitably treated for corrosion protection.

IP66 or IP67: It is likely that water will be encountered in many scenarios, either on the ground or from bad weather. An IP 66, water-proof, rating should be the minimum level requirement.

Extreme Temperature Rating: The working ambient temperature range of the robot should be reviewed to ensure a capability from -20°C to + 50°C. This fall short of military specifications of -40°C to +80°C, but will help keep the cost down.

Better Integration of Electrical Modules: The design successfully integrated COTS components into a working system. However, the system has 4 DC-DC converters (Charger, Drive-System, Rasbery Pi PSU, Audio Amplifier) and it would make sense to integrate these into perhaps a single circuit board. This would also reduce the amount of power wiring and improve EMC.

Better EMC Design: The use of button magnets for the convenient attachment of panels, whilst an attractive feature, should be avoided in any future implementations due to potential magnetic interference with sensitive magnetometer readings. In fact the whole topic of EMC would require more consideration to ensure that other potential EMI sources are also adequately suppressed through shielding or filtering.

Servo Motor Position Feedback: The servo-motors currently do provide position feedback, but this is only used locally within the built-in motor-drive. A position signal for each motor that could be read by the robot's main controller would permit better position control with a more 'elastic' movement in the legs.

Implement WLAN: In order to function in remote outdoor environments the robot will need the ability to communicate over LTE, 4G, 5G networks, or even specific, first-responder networks, such as FirstNet.

Facial Recognition: In rescue scenarios, it is often needed to locate specific individuals or keep track of those who have been accounted for. Integrating facial recognition capabilities into the robot would enhance its ability to identify and assist specific persons in need. This feature

would be particularly useful for locating missing individuals in disaster zones or verifying the identities of survivors. Additionally, incorporating this feature into the web application would allow remote operators to visualise and monitor the identities of the people detected, thereby improving the overall efficiency and effectiveness of rescue operations. This enhancement could significantly contribute to the success of search and rescue missions by providing more detailed and accurate information to first responders.

7.9 Design Files

The design files and source code is available in the project's public repository, providing comprehensive resources for replication and further development of the quadruped robot:

<https://github.com/SpaceBod/QuadREX>

A

Appendix

Table A.1: R&D Items

Item	Quantity	Total Price
Intel Realsense Camera	1	£153.00
Adafruit I2S Speaker Bonnet	2	£25.80
LED Plastic Bevel Holder	2	£1.80
LED Driver Board I2C	1	£4.80
Total		£185.40

Table A.2: Bill of Materials for REX

Category	Item	Quantity	Cost
Electronics			
	PCA9685 PWM Controller	2	£11.99
	Raspberry Pi 4B 8GB	1	£79.00
	LD19 D300 TOF Lidar	1	£72.70
	Dual Microphone Module	1	£27.31
	OLED Display 0.96	2	£7.99
	Speaker 3 Watt 4 Ohm	2	£6.99
	Dual Power Audio Amp Module	3	£5.31
	Oak-D LITE Camera (2nd Hand)	1	£100.00
	9-Axis Gyro Magnetometer	1	£10.00
	Various Power Switches	1	£4.00
	LG MJ1 18650 Cells	18	£56.00
Mechanical			
	Servo Motors (BLS-HV45MG-180)	12	£337.68
	Shoulder Bearing (6704-2RS)	4	£9.99
	Flanged Bearing (F696ZZ)	10	£9.17
	Knee Bearing (MR106ZZ)	10	£7.99
	M3/M2.5 Screws	100	£10.00
	Threaded Brass Inserts	100	£4.98
Materials			
	Carbon Fiber Tubes	4	£22.54
	3D Filament - Yellow PLA+ 1KG	1	£15.95
	3D Filament - Black ASA 1KG	1	£22.99
	Paint Set	1	£5.99
Electrical			
	Nichicon Capacitors 16V 2200uF	4	£7.01
	Converter 20A	1	£12.00
	Buck Converter CC/CV 5A	2	£10.60
	Schottky Rectifier Set	1	£8.99
	40mm 12V Fan	1	£4.00
	40A Battery Management Board	4	£8.99
	Wiring - Various Gauges	1	£15.00
	USB-C to USB-A Right Angle	2	£11.99
	USB-A 3.2 U-Bend Adapter	2	£10.00
Miscellaneous			
	DUPONT Crimping Set	1	£11.00
	Velcro for Battery	1	£7.99
	Magnets (3x6mm)	110	£4.99
	Aux Cable	2	£3.24
	Squash Balls	2	£6.50
	128GB microSDXC	1	£12.80
Total Expenses (GBP)			£964.67

Bibliography

- [1] B. J. D. S. CNN, *How 9/11 inspired a new era of robotics*, en. [Online]. Available: <http://www.cnn.com/2011/TECH/innovation/09/07/911.robots.disaster.response/index.html> (visited on 06/14/2024).
- [2] EU project successfully deploys robots following Italy earthquake, en. [Online]. Available: <https://cordis.europa.eu/article/id/120405-eu-project-successfully-deploys-robots-following-italy-earthquake> (visited on 06/14/2024).
- [3] Paris Firefighters Used This Remote-Controlled Robot to Extinguish the Notre Dame Blaze - IEEE Spectrum. [Online]. Available: <https://spectrum.ieee.org/colossus-the-firefighting-robot-that-helped-save-notre-dame> (visited on 06/14/2024).
- [4] (PDF) Searching for survivors of the Mexico earthquake—with snake robots. [Online]. Available: https://www.researchgate.net/publication/320230105_Searching_for_survivors_of_the_Mexico_earthquake-with_snake_robots (visited on 06/14/2024).
- [5] Inspection Robots Market Expected to Reach \$13.9 Billion by 2030. [Online]. Available: <https://www.alliedmarketresearch.com/press-release/inspection-robots-market.html> (visited on 06/14/2024).
- [6] Japan's Ground Self-Defense Force Deploys Robot Dogs To Aid Earthquake Relief Efforts, en-CA, Section: Japan, Jan. 2024. [Online]. Available: <https://www.overtdefense.com/2024/01/19/japans-ground-self-defense-force-deploys-robot-dogs-to-aid-earthquake-relief-efforts/> (visited on 06/14/2024).
- [7] J. Alvarez and M. Hunt, “Risk and resilience in canine search and rescue handlers after 9/11,” eng, *Journal of Traumatic Stress*, vol. 18, no. 5, pp. 497–505, Oct. 2005, ISSN: 0894-9867. DOI: 10.1002/jts.20058.
- [8] J. Casper and R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, Jun. 2003, Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), ISSN: 1941-

0492. DOI: 10.1109/TSMCB.2003.811794. [Online]. Available: <https://ieeexplore.ieee.org/document/1200160> (visited on 01/29/2024).
- [9] *Uncounted costs - Data gaps hide the true human impacts of disasters in 2023 / UNDRR*, en, Jan. 2024. [Online]. Available: <http://www.undrr.org/explainer/uncounted-costs-of-disasters-2023> (visited on 05/28/2024).
- [10] E. Cavallo, L. Giles Álvarez, and A. Powell, “Estimating the Potential Economic Impact of Haiti’s 2021 Earthquake,” Inter-American Development Bank, Sep. 2021. DOI: 10.18235/0003657. [Online]. Available: <https://publications.iadb.org/en/node/30923> (visited on 06/03/2024).
- [11] “Turkey Puts Economic Toll From Earthquakes at About \$104 Billion,” en, *Bloomberg.com*, Mar. 2023. [Online]. Available: <https://www.bloomberg.com/news/articles/2023-03-17/turkey-puts-economic-toll-from-earthquakes-at-about-104-billion> (visited on 06/14/2024).
- [12] K. Komiya and K. Komiya, “Japan warns of economic impact of Noto earthquake - govt report,” en, *Reuters*, Jan. 2024. [Online]. Available: <https://www.reuters.com/world/asia-pacific/japan-warns-economic-impact-noto-earthquake-govt-report-2024-01-25/> (visited on 06/14/2024).
- [13] *Funding & tenders*. [Online]. Available: <https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/horizon-cl3-2023-drss-01-05> (visited on 02/13/2024).
- [14] Y. Watanobe, R. Kabir, R. Aoba, *et al.*, “Disaster Rescue via Multi-Robot Collaboration: Development, Control, and Deployment,” en, *Journal of Robotics and Mechatronics*, vol. 35, no. 1, pp. 85–98, Feb. 2023, ISSN: 1883-8049, 0915-3942. DOI: 10.20965/jrm.2023.p0085. [Online]. Available: <https://www.fujipress.jp/jrm/rb/robot003500010085> (visited on 02/13/2024).
- [15] M. Wu, “Robotics Applications in Natural Hazards,” *Highlights in Science, Engineering and Technology*, vol. 43, pp. 273–279, Apr. 2023. DOI: 10.54097/hset.v43i.7429.
- [16] *Boston Dynamics’ SpotMini Is All Electric, Agile, and Has a Capable Face-Arm - IEEE Spectrum*, en. [Online]. Available: <https://spectrum.ieee.org/boston-dynamics-spotmini> (visited on 02/07/2024).

- [17] *Go2 quadruped robot by Unitree Robotics - ROS compatible / cost-efficient*. [Online]. Available: <https://www.generationrobots.com/en/content/156-go2-quadruped-robot-by-unitree-robotics> (visited on 02/07/2024).
- [18] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, “BigDog, the Rough-Terrain Quadruped Robot,” *IFAC Proceedings Volumes*, 17th IFAC World Congress, vol. 41, no. 2, pp. 10 822–10 825, Jan. 2008, ISSN: 1474-6670. DOI: 10.3182/20080706-5-KR-1001.01833. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016407020> (visited on 02/07/2024).
- [19] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Oct. 2018, pp. 2245–2252. DOI: 10.1109/IROS.2018.8593885. [Online]. Available: <https://ieeexplore.ieee.org/document/8593885> (visited on 02/07/2024).
- [20] A. Bouman, M. F. Ginting, N. Alatur, *et al.*, *Autonomous Spot: Long-Range Autonomous Exploration of Extreme Environments with Legged Locomotion*, arXiv:2010.09259 [cs], Nov. 2020. [Online]. Available: <http://arxiv.org/abs/2010.09259> (visited on 01/29/2024).
- [21] M. Sombolestan, Y. Chen, and Q. Nguyen, *Adaptive Force-based Control for Legged Robots*, arXiv:2011.06236 [cs, eess], Dec. 2021. [Online]. Available: <http://arxiv.org/abs/2011.06236> (visited on 01/29/2024).
- [22] D. Innovates and Q. P. a. D. Seeley, *AT&T’s Robotic Dogs Assist First Responders, the Military, Businesses, and More—With No Belly Rubs Needed*, en-US, Mar. 2023. [Online]. Available: <https://dallasinnovates.com/atts-robotic-dogs-assist-first-responders-the-military-businesses-and-more-with-no-belly-rubs-needed/> (visited on 06/14/2024).
- [23] A. Feldman, *Meet The Startup Behind The Robot ‘Dogs’ Set To Patrol The Southern Border*, en, Section: Manufacturing. [Online]. Available: <https://www.forbes.com/sites/amyfeldman/2022/02/15/meet-the-startup-behind-the-robot-dogs-set-to-patrol-the-southern-border/> (visited on 06/14/2024).
- [24] *DEEP Robotics - Global Quadruped Robot Leader*. [Online]. Available: <https://www.deeprobotics.cn/en> (visited on 06/14/2024).
- [25] T. Tanikawa, Y. Masuda, and M. Ishikawa, “A Reciprocal Excitatory Reflex Between Extensors Reproduces the Prolongation of Stance Phase in Walking Cats: Analysis on a Robotic

- Platform,” *Frontiers in Neurorobotics*, vol. 15, p. 636864, Apr. 2021. DOI: 10.3389/fnbot.2021.636864.
- [26] S. D. Potter, Z. J. Jackowski, and A. Young, “Screw Actuator for a Legged Robot,” en, US20180172121A1, Jun. 2018. [Online]. Available: <https://patents.google.com/patent/US20180172121A1/en> (visited on 02/07/2024).
- [27] M. H. Rahman, S. Alam, T. Das Mou, F. Uddin, and M. Hasan, “A Dynamic Approach to Low-Cost Design, Development, and Computational Simulation of a 12DoF Quadruped Robot,” *Robotics*, vol. 12, p. 24, Feb. 2023. DOI: 10.3390/robotics12010028.
- [28] S. Ramírez-Revilla, D. Camacho-Valencia, E. Gonzales-Condori, and G. Márquez, “Evaluation and comparison of the degradability and compressive and tensile properties of 3D printing polymeric materials: PLA, PETG, PC, and ASA,” *MRS Communications*, vol. 13, pp. 55–62, Feb. 2023. DOI: 10.1557/s43579-022-00311-4.
- [29] *Test of LG 18650 MJ1 3500mAh (Green)*. [Online]. Available: <https://lygte-info.dk/review/batteries2012/LG%2018650%20MJ1%203500mAh%20%28Green%29%20UK.html> (visited on 02/13/2024).
- [30] *Figure 5 : Comparison of Energy Density in Battery Cells [NASA-National...]* en. [Online]. Available: https://www.researchgate.net/figure/Comparison-of-Energy-Density-in-Battery-Cells-NASA-National-Aeronautics-and-Space_fig1_327237748 (visited on 06/08/2024).
- [31] Y. Liu, J. Zhang, R. Niu, *et al.*, “Manufacturing of high strength and high conductivity copper with laser powder bed fusion,” en, *Nature Communications*, vol. 15, no. 1, p. 1283, Feb. 2024, Publisher: Nature Publishing Group, ISSN: 2041-1723. DOI: 10.1038/s41467-024-45732-y. [Online]. Available: <https://www.nature.com/articles/s41467-024-45732-y> (visited on 06/14/2024).
- [32] S. D. Jadhav, L. R. Goossens, Y. Kinds, B. V. Hooreweder, and K. Vanmeensel, “Laser-based powder bed fusion additive manufacturing of pure copper,” *Additive Manufacturing*, vol. 42, p. 101990, Jun. 2021, ISSN: 2214-8604. DOI: 10.1016/j.addma.2021.101990. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221486042100155X> (visited on 06/14/2024).
- [33] K. Genç, S. Toyting, E. Galindo-Navia, I. Todd, and K. Mumtaz, “Laser powder bed fusion of NdFeB and influence of powder bed heating on density and magnetic properties,” en, *The International Journal of Advanced Manufacturing Technology*, vol. 132, no. 9, pp. 5017–

- 5038, Jun. 2024, ISSN: 1433-3015. DOI: 10.1007/s00170-024-13605-9. [Online]. Available: <https://doi.org/10.1007/s00170-024-13605-9> (visited on 06/14/2024).
- [34] *Figure 2.1: The life of Li-Ion batteries (cycle life) strongly depends...* en. [Online]. Available: https://www.researchgate.net/figure/The-life-of-Li-Ion-batteries-cycle-life-strongly-depends-on-the-operating_fig6_312376975 (visited on 06/08/2024).
- [35] A. Huang, *Hard Commutation of Power MOSFET*, English, 2014. [Online]. Available: https://www.infineon.com/dgdl/Infineon-Power_MOSFET_OptiMOS_FD_200V-250V_hard_diode_commutation-AN-v01_00-EN.pdf?fileId=db3a304344ae06150144b1d2f8250165 (visited on 06/13/2024).
- [36] A. Aristidou and J. Lasenby, *Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver*. Sep. 2009.
- [37] K. Daniel, A. Nash, S. Koenig, and A. Felner, “Theta*: Any-Angle Path Planning on Grids,” en, *Journal of Artificial Intelligence Research*, vol. 39, pp. 533–579, Oct. 2010, ISSN: 1076-9757. DOI: 10.1613/jair.2994. [Online]. Available: <https://jair.org/index.php/jair/article/view/10676> (visited on 06/11/2024).
- [38] D. Harabor and A. Grastien, “Online Graph Pruning for Pathfinding On Grid Maps,” en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, pp. 1114–1119, Aug. 2011, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v25i1.7994. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/7994> (visited on 06/11/2024).
- [39] RoboFoundry, *OAK-D Lite Camera — Basic Setup*, en, Feb. 2024. [Online]. Available: <https://robofoundry.medium.com/oak-d-lite-camera-basic-setup-38a563cd594f> (visited on 06/11/2024).
- [40] E. Uslu, F. Çakmak, M. Balçılار, A. Akıncı, M. F. Amasyalı, and S. Yavuz, “Implementation of frontier-based exploration algorithm for an autonomous robot,” in *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, Sep. 2015, pp. 1–7. DOI: 10.1109/INISTA.2015.7276723. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7276723> (visited on 06/13/2024).
- [41] E. Pakdel, S. Kashi, R. Varley, and X. Wang, “Recent progress in recycling carbon fibre reinforced composites and dry carbon fibre wastes,” *Resources, Conservation and Recycling*, vol. 166, p. 105340, Mar. 2021, ISSN: 0921-3449. DOI: 10.1016/j.resconrec.2020.105340. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921344920306558> (visited on 06/14/2024).

- [42] Z. Chen, M. Yang, Q. Shi, X. Kuang, H. J. Qi, and T. Wang, “Recycling Waste Circuit Board Efficiently and Environmentally Friendly through Small-Molecule Assisted Dissolution,” en, *Scientific Reports*, vol. 9, no. 1, p. 17902, Nov. 2019, Publisher: Nature Publishing Group, ISSN: 2045-2322. DOI: 10.1038/s41598-019-54045-w. [Online]. Available: <https://www.nature.com/articles/s41598-019-54045-w> (visited on 06/14/2024).
- [43] B. Niu, E. Shanshan, Z. Xu, and J. Guo, “How to efficient and high-value recycling of electronic components mounted on waste printed circuit boards: Recent progress, challenge, and future perspectives,” *Journal of Cleaner Production*, vol. 415, p. 137815, Aug. 2023, ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2023.137815. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095965262301973X> (visited on 06/14/2024).
- [44] M. Kaya, “An overview of NdFeB magnets recycling technologies,” *Current Opinion in Green and Sustainable Chemistry*, vol. 46, p. 100884, Apr. 2024, ISSN: 2452-2236. DOI: 10.1016/j.cogsc.2024.100884. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2452223624000051> (visited on 06/14/2024).
- [45] H. Wang, J. Wang, X. Lei, *et al.*, “Separation and Recovery of Rare Earths and Iron from NdFeB Magnet Scraps,” en, *Processes*, vol. 11, no. 10, p. 2895, Oct. 2023, Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2227-9717. DOI: 10.3390/pr11102895. [Online]. Available: <https://www.mdpi.com/2227-9717/11/10/2895> (visited on 06/14/2024).