

MINOR BIOMIMICRY

 UNIVERSITY OF AMSTERDAM
Makerspace

Quad-EX: Robotic cave exploration

Noe Nicolas Medina (15909433), Joost Weerheim (13769758),
Eva Bredeweg (13770551), Sep Claessen (14653907),
Margriet Frericks (15904474)

January 24, 2025

Supervisor(s): Morten Strømme, Mimi den Uyl, Pim Linnebank

Abstract

This report presents the design, development, and testing of Quad-EX, a quadruped robotic system aimed at addressing the challenges of human cave exploration. The robot is designed for autonomous navigation and data collection in difficult cave environments, such as the Rijckholt Prehistoric Flint Mine in Limburg. Through Biomimicry, the robot incorporates features inspired by canine anatomy, including a shock-absorbing paw design and a walking gait for improved stability. The system is equipped with the necessary cameras to perform photogrammetry, an IMU sensor for orientation tracking, and ultrasonic sensors for obstacle detection, all integrated into a lightweight 3D-printed body. The report outlines the iterative design process, including mechanical, software, and electronic subsystems, as well as performance evaluations through testing. Quad-EX demonstrates the potential of Biomimicry and robotics in overcoming the physical and safety barriers of human cave exploration, leading the way for more efficient archaeological, geological, and environmental research.

Contents

1	Introduction	5
2	Theoretical Framework	6
2.1	Literature Review - Technical	6
2.1.1	Locomotion mechanisms	6
2.1.2	Walking Engine	6
2.1.3	Principles of Photogrammetry	7
2.1.4	Navigation	8
2.1.5	Comparison Of Sensor Technology	8
2.1.6	3D Printing: Infill density and patterns	9
2.2	Literature Review - Biomimicry	10
2.2.1	Quadrupeds: Leg Geometry and Structure	10
2.2.2	Paw and Shock Absorption	11
3	Evolution and Prototype Design	12
3.1	Design Phase - Mechanical	12
3.1.1	Design Guidelines	12
3.1.2	Performance Specifications	12
3.1.3	Leg Design Stages	12
3.1.4	Paw Structure Implementation	17
3.1.5	Body Structure Implementation	18
3.1.6	Photogrammetry implementation	21
3.1.7	Electronics	23
3.2	Design Phase - Software	26
3.2.1	Design Guidelines	26
3.2.2	Performance Specifications	26
3.2.3	Walking Engine Implementation	27
3.2.4	Navigation	29
4	Testing Phase	32
4.1	Test Methods	32
4.1.1	Test 1: Foot Grip	32
4.1.2	Test 2: Basic Movements	32
4.1.3	Test 3: Walking Engine	33
4.1.4	Test 4: Center of Mass	33
4.1.5	Test 5: Photogrammetry	34
5	Stakeholders	35
5.1	Stakeholder Identification	35
5.1.1	Jitte Waagen	35
5.1.2	Alicia Walsh	36
5.1.3	Tycha Buekers	36
5.1.4	Arnoud Visser	37

5.1.5	Mels van Eck	37
6	Conclusions and Discussion	39
6.1	Design Evaluation	39
6.2	Performance Assessment	39
6.3	Comparison with Stakeholder Input	41
6.4	Evaluation of Biomimicry	41
6.4.1	Leg design	41
6.4.2	Paw structure	41
6.5	Ethical Considerations	41
6.6	Future Plans	42
6.6.1	Short Term Improvements	42
6.6.2	Long Term Improvements	43
6.6.3	Improvements Possible with Unlimited Resources	43
6.7	Reflective Insights	43
6.7.1	Points of Improvement	43
6.7.2	Positives	43
7	Documentation	45
7.1	GitHub	45
7.2	Quad-EX contact information	45
A	Appendix	49
A.1	Software	49
A.1.1	ROS 2	49
A.1.2	Docker	49
A.1.3	Switching to Python	50
A.2	Author contributions	50
A.3	Schematic overview of electronic components	50
A.4	Mechanical Design Guidelines	50
A.5	Mechanical Design Specifications	50
A.6	Software Design Guidelines	50
A.7	Software Design Specifications	50
A.8	Sensor Criteria	50
A.9	3D Printing Infill Patterns	50

CHAPTER 1

Introduction

The drive to explore and expand our understanding of the world is an inherent human instinct. For centuries humanity has continued to push the boundaries of the unknown, from unmapped forests to the far depths of the ocean. From previous scientific breakthroughs, to the discoveries that reshape our understanding of the past, the rewards of exploration are boundless. A large portion of the earth remains undiscovered, with around 50% of the Earth's land surface being classified as having low human influence or being relatively untouched (Paxman 2020).

Among these unexplored regions, caves stand out as a "by far the most evocative archaeological sites", given by their ability to exceptionally preserve organic materials and rock art (Conati Barbaro et al. 2024). Caves are also powerful archives of paleo-environmental records, retaining crucial information about past climates, vegetation changes, and geological events (Conati Barbaro et al. 2024). Despite this, caves remain largely unexplored due to the substantial risks they pose, such as darkness, low temperatures, extremely narrow passages, the possibility of getting lost, getting stuck, hypothermia, panic attacks and many more. According to the British Cave Rescue Council, the UK experienced 17 caving incidents, with 3 fatalities in 2023 alone (British Cave Rescue Council 2024), thus proving our limited ability to safely discover these territories. As a result, important knowledge about the Earth, its origins and history remain undiscovered, highlighting the need for new, more innovative exploration methods.

To address this challenge, robotic technologies offer a potential solution. Using automated robots, specifically designed for cave exploration, can help overcome the physical limits faced by human explorers, allowing for safer access to territories once deemed too remote and hazardous. Robotic explorers can navigate difficult terrain and collect data without putting humans at risk. This is particularly beneficial for fields such as archaeology or mining, where accessing these remote caves could lead to precious discoveries.

The Rijckholt Prehistoric Flint Mine in Limburg is a key area of interest for one of our stakeholders, Tycha Buekers. This 8-hectare mine contains approximately 2,000 connected shafts, where prehistoric humans mined flint between 6400 and 4700 years ago. In 1964, the Prehistoric Flint Mines Working Group of the Dutch Geological Society carried out excavations of flint mines at Rijckholt-St. Geertruid, which lasted 8 years. The excavation yielded over 14,000 artefacts, and it was calculated that 400,000 artefacts may still be present in the subsoil at Rijckholt-St. Geertruid (Vuursteenmijnen.nl 2025). However, the conditions within the mine present difficulties for further discovery. The shafts are narrow and require crawling, making it physically demanding and time-consuming for humans. The potential for rockfall and geological instabilities also pose a significant safety threat. According to Alicia Walsh (personal communication, 2025) current methodologies for mapping the mines include manual mapping and photogrammetry for digital reconstruction. However, this is a repetitive and time consuming process, which further slows research. These challenges highlight the potential of a robotic solution. A compact and maneuverable robot can more easily fit through the narrow shafts, and easily adapt to the uneven terrain, expanding the limit of exploration. Without the need for human intervention, humans are no longer put at risk. Furthermore, specialized sensors can be added to automate the task of data collection, effectively speeding up excavations of the Rijckholt Flint Mine.

This report thus focuses on the development of an advanced robotic system designed to explore the Rijkholt Prehistoric Flint Mine in Limburg, traverse its complex terrain, and collect data. The main objectives include:

1. To facilitate access to unexplored sites, thus uncovering new scientific knowledge in a safe manner.
2. To make exploration more efficient by automating data collection.

CHAPTER 2

Theoretical Framework

2.1 Literature Review - Technical

2.1.1 Locomotion mechanisms

The performance of a robot is largely determined by its locomotion mechanism. Mobility, robustness and efficiency are all factors that determine whether a robot can successfully traverse specific terrains. Let us consider four main types of locomotion; Legged, wheeled, crawler-type and aerial.

Aerial robots are often used in cave exploration, because of their ability to easily gather data from a top-down view. Collectively using aerial robots is argued to minimize flight time, while maximizing the surveilled area (Petráček et al. 2021). Other research indicates that using a combination of aerial and ground-based robots is effective for navigating caves. In this case, aerial drones can provide surveillance that assists ground robots in their navigation (Mueggler et al. 2014).

However, aerial robots have multiple drawbacks when used in caves. Firstly, the control system has to account for one more degree of freedom in comparison to a ground-based robot, requiring more computing power. Secondly, aerial robots make downward pushing winds, that can potentially disrupt delicate carvings or drawings in a cave.

Ground-based robots do not face these challenges. Additionally, a ground-based robot is better suited for navigation in an environment without connectivity, since it can carry more weight to include a lighting source, LiDAR and enough batteries. It also is significantly more robust to withstand a collision than an aerial robot. We have therefore chosen to focus our research on ground-based robots.

When analyzing ground-based robots, wheeled and crawler-type robots are more suited for flat terrain, while legged robots are more capable of maneuvering through obstacles-affected terrains (Lu et al. 2022). This is logical, as legged robots can more easily step over obstacles. And since caves consist mostly of uneven and rocky terrain, we can exclude wheeled and crawler-type robots from consideration. In conclusion, legged robots are the best choice for cave exploration due to their ability to navigate uneven terrain, carry essential equipment, and operate without connectivity.

With respect to the number of legs needed, there is a trade-off between stability and motion efficiency. More legs result in more stability, but decrease motion efficiency, since more legs are required to move to gain the same distance. A quadruped, meaning four legs, contains both aspects and forms a good middle ground (Lu et al. 2022).

2.1.2 Walking Engine

To enable a quadruped robot's locomotion, inverse kinematics is crucial. Inverse kinematics is the process of obtaining joint angles from known coordinates of end effector (the foot). Our quadruped's inverse kinematics is based on the methodology by (Sen, Bakircioglu, and Kalyoncu 2017). This forms the basis of our walking engine, which controls the robot's movement. We implemented a trot gait, where two feet are always on the ground while the other two are in the air. An offset is applied to the step trajectory, which follows a Bezier curve due to its effective trajectory shaping (Zeng et al. 2019). This allows the quadruped to move forward smoothly. Navigating uneven terrain for cave exploration requires continuous step adjustment to maintain stability. Although force-sensitive resistors on the feet were not feasible, an Inertial Measurement Unit (IMU) can provide orientation data to keep the robot balanced. Research has shown that IMU data can be utilized in reinforcement learning to dynamically adjust foot positioning

(Rahme et al. 2020). However, due to time constraints, this reinforcement learning approach was not implemented in our project.

2.1.3 Principles of Photogrammetry

To map the environment of the robot, a technique called photogrammetry will be used. Photogrammetry is a technique used to create detailed 3D digital models by analyzing images captured from different angles. It allows for precise documentation and analysis of subterranean structures that are often inaccessible. The result is a highly accurate 3D model of the photographed area. Photogrammetry is widely adopted in archaeology and is valued for its cost effectiveness and adaptability to field and lab settings (Awange et al. 2013). Photogrammetry is based on several key principles that enable the extraction of accurate measurements and 3D models from photographs (Awange et al. 2013).

Projective Geometry

One of the fundamental principles of photogrammetry is projective geometry, a branch of mathematics that describes how three-dimensional objects are projected onto a two-dimensional plane, such as a photograph. Camera properties and settings, including focal length, sensor dimensions, and lens parameters, play a crucial role in this process. These factors determine how objects within the camera's field of view are represented and directly influence the accuracy of the final 3D reconstruction. One of the most critical choices is the type of camera lens. Fixed focal length lenses, often referred to as "prime lenses," are particularly suitable for this work because of their optical quality and low distortion (Bisson-Larrivée and LeMoine 2022). Among these, lenses with a focal length of 40 to 60 mm strike the perfect balance. They provide high detail while minimizing potential errors in the reconstruction process.

Scale and perspective

Scale and perspective are key concepts in photogrammetry (Awange et al. 2013). Since photos are two-dimensional, they inherently contain perspective distortions that must be corrected to achieve realistic 3D models. This can be accomplished by capturing photos from different angles and distances. By capturing overlapping images of an environment from different perspectives, photogrammetry software algorithms can identify common points between these images, called tie points. Using these tie points, the software can calculate their spatial relationships and reconstruct the geometry of the captured object. This process is critical for achieving accurate models of the photographed area.

Image overlap

An essential aspect of photogrammetry is a sufficient amount of image overlap (Awange et al. 2013). Image overlap is needed because it ensures that there are common reference points between consecutive photos, which allows the photogrammetry software to correctly align the photos and reconstruct the 3D model. In an interview with archaeologist Alicia Walsh, she stated that a minimum overlap of 60% is required for effective model reconstruction. This ensures the best identification of shared features, such as corners and textures. These shared features form the basis for combining images into a coherent 3D model. Without sufficient overlap, it is challenging to create a reliable reconstruction.

Number of photographs

The number of photographs taken per rotation of the object is another crucial parameter for photogrammetry. Capturing a minimum of 18 images per rotation ensures sufficient overlap between images, which is necessary for the software to align them accurately. However, capturing 24 images per rotation is often seen as the sweet spot, offering optimal quality without unnecessarily increasing processing time or storage demands. Adding more images typically does not lead to noticeable improvements in the final model and only takes more time (Bisson-Larrivée and LeMoine 2022).

Lighting and reflections

Lighting plays an important role in photogrammetry. Inconsistent lighting conditions, such as sharp shadows, highlights, or overexposure, can introduce significant distortions. This can hinder the software's ability to identify and match key features across images (Jasiak, Schultz, and Ferrell 2023). These inconsistencies may lead to errors in the alignment and modeling process. Reflections can have the same effects, as they cause glare or reflections that interfere with the feature identification of the software. Uniform and consistent lighting reduces the risk of data artifacts and

loss of texture information. Photogrammetry with sufficient lighting leads to more reliable and precise image matching and improves the overall quality of the generated 3D models.

2.1.4 Navigation

Navigation is an essential component for any mobile robot, and when effective navigation is achieved, it enables effective path planning around obstacles in an environment. In caves specifically, the robot will be faced with two challenges that further complicate navigation:

1. **Lack of GPS-signal in underground structures:** GPS enables good localization, and precise localization is essential for mobile robots to operate autonomously and perform tasks (Macenski et al. 2022).
2. **No a priori map:** The robot operates in an unmapped environment and thus does not have an a priori map of the environment it operates in, while having access to a map of the environment is a requirement for the robot to obtain its position and successfully execute its path planning tasks (Taheri and Xia 2021).

To overcome the first challenge, an alternative to GPS is needed for localization. For the second, the robot would need to simultaneously construct a map of the unknown environment and determine its position within it. Both challenges are addressed by the Simultaneous Localization and Mapping (SLAM) algorithm (*SLAM* n.d.), which enables a robot to map their unknown surroundings while simultaneously determining its position within that map. This process relies on various sensor inputs, with laser scanner being a favored option due to their robustness in supporting SLAM algorithms for localization (Cole and Newman 2006). SLAM also benefits from compatibility with widely-used tools in robotics, such as the Robot Operating System (ROS) and Python.

However, implementing SLAM adequately introduces a few difficulties or challenges. For a more elaborate and detailed overview of the challenges and how they may arise, see Section 3.2.4.

SLAM may thus be not be viable due to the limitations imposed by both the environment and the robot, and alternative navigation methods must be considered. For example, inertial navigation, based on data from the Inertial Measurement Unit (IMU), can estimate a robot's position by tracking its orientation and motion relative to its starting point. Complementary approaches often include using distance sensors to detect obstacles and trigger a simple obstacle avoidance behavior. This allows the robot to navigate and explore without relying on optimal path planning.

2.1.5 Comparison Of Sensor Technology

Sensors are essential in robotic applications by collecting information to select the appropriate actions, based on and specific to the application they are used for. Given the diverse range of applications in robotics, a wide variety of sensors is available. Selecting the right sensors requires thorough research careful consideration of both its advantages and disadvantages. Therefore, a short description of the sensors is given below, along with their (main) advantages and disadvantages.

Light Detection And Ranging (LiDAR) sensor

The LiDAR sensor gathers spatial data to create 3D point clouds and subsequently 3D maps of the environment by emitting laser beams and calculating the distance to objects by measuring the time it takes for the reflected beams to return. It is often used with the Simultaneous Localization and Mapping (SLAM) algorithm to enable mapping and localization, and to facilitate navigation. LiDAR is one of the most robust sensors for SLAM integration, as it produces high-resolution 3D point clouds and creates accurate 3D representations of the environment, outperforming camera-based systems (Singh 2021). It performs well in low-light conditions, as it operates independently of lighting (Petrík, Krajiník, and Saska 2021).

Inertial Measurement Unit (IMU) sensor

The IMU is an inertial sensor, which means that this sensor determines the location, velocity and direction, thereby providing accurate real-time orientation information. It determines these from the data gathered by the accelerometer that determines the acceleration, which in turn can be used to determine the velocity and displacement. While it's possible to have only an accelerometer incorporated in an IMU sensor (commonly referred to as a 3 Degrees Of Freedom IMU (3-DOF) sensor), IMU sensors can also incorporate more sensors to provide more accurate orientation information. The 9-DOF IMU sensor combines a three-axis accelerometer, three-axis gyroscope and three-axis magnetometer;

accelerometers measure linear motion (up/down, left/right, forwards/backwards), gyroscopes rotational motion (pitch, yaw, roll), and magnetometers determine directional orientation (north/south, east/west, up/down). With the added redundancy that three sensors give, good accuracy is given for the 9-DOF IMU sensor. Therefore the selected version of the IMU sensor will probably be one with 9-DOF. The data from these sensors allows for motion tracking, inertial navigation and stabilization (*Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055* 2025). An IMU is a great sensor when GPS information becomes unavailable (such as in underground environments) (Toy, Durdu, and Yusefi 2022). It is a small and lightweight, cost-effective sensor which provides real time data with good accuracy (Shenoy, Gupta, and Varadhan 2022). In addition to that, it also incorporates a 1Hz temperature sensor which measures the temperature in degrees Celsius (*Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055* 2025), which provides interesting information. However, IMUs in general often have a low signal-to-noise ratio, leading to highly inaccurate velocity and orientation data that requires filtering (Toy, Durdu, and Yusefi 2022). If a 9-DOF IMU sensor with a micro controller for sensor fusion and Kalman filter is chosen, the effect that these inaccuracies have on the final orientation determination is decreased. When choosing a 9-DOF IMU sensor, which is a likely choice for selection, there are 9-DOF IMU sensors available that include a micro controller with Kalman filter that perform sensor fusion on the raw data of the sensors incorporated in the IMU sensor. This increases the accuracy of the orientation estimation considerably (Shenoy, Gupta, and Varadhan 2022). For other IMU sensors, this remains a more prominent drawback. Another drawback is the delicacy of IMU sensor; their magnetometers are particularly prone to errors from interference caused by nearby metallic components (Shenoy, Gupta, and Varadhan 2022; Zhang 2019).

Ultrasonic distance sensor

The ultrasonic distance sensor is a popular and cost-effective infrared sensor that uses sonar to measure distances to objects in the environment by measuring the time between the transmission and reflection of the ultrasonic wave (Gabriel and Kuria 2020). While laser sensors (including LiDAR sensors) are most optimal for acquiring accurate localization of objects in general, ultrasonic distance sensors are equally suitable for localizing objects at a short range (Zhmud et al. 2018). Some benefits are their high detection accuracy, often reaching up to 3mm (*Ultrasonic Ranging Module HC - SR04* 2025), their effectiveness in low light conditions (Gabriel and Kuria 2020), and their ease of integration with software. However, distance sensors are prone to error from interference with unwanted noise (Al Tahtawi 2018). Moreover, for long range measurements, distance sensors have lower sampling rates than laser based sensors, since sound travels at a lesser speed.

Camera

Cameras provide detailed visual information that can be used to reconstruct the texture and visual aspects of cave structures, complimenting spatial data from other sensors. Camera sensors work by capturing light and converting it into digital signals, producing images or video streams (*Mechanism Digital Camera* n.d.). This makes it possible to capture objects and surfaces with high precision. One significant advantage of camera sensors is their ability to provide high-resolution visual information, which makes detailed scene analysis possible. This is particularly useful for applications where a lot of details are required. In addition, cameras are relatively cost-effective compared to some advanced sensors, making them an accessible option for many robotic systems. However, cameras also have a large downside: they heavily rely on ambient lighting conditions. Their performance can degrade in low-light or overly bright environments without additional lighting or specialized technology (Moomaw 2007). Furthermore, processing visual data often requires substantial computational power, especially for complex tasks like real-time object recognition or scene reconstruction. Despite these flaws, camera sensors remain a versatile and essential tool in robotics, often used together with other sensors, such as LiDAR or IMUs

2.1.6 3D Printing: Infill density and patterns

3D printing is an emerging manufacturing method used for cheap, rapid and reliable prototyping. The range of printable designs are limitless, but what is often disregarded is the infill pattern and density of the design, which greatly affects the performance of the final product. Infill is the material that fills the interior of a 3D printed object, providing strength, weight, and durability (3DSourced 2023). Many parts of a robot experience loads of different magnitudes in different directions. It is therefore necessary to bring careful attention to the material choice and structure of the design. Nowadays, all 3D printing slicer software provides a range of different infill patterns and densities to choose from. Choosing the correct infill density depends greatly on the purpose of the part that is printed. A high infill percentage means more material is used and hence more strength, but takes longer to print. A low infill

means a lighter and faster print, but more prone to fracture. The majority of slicer programs have a default infill setting somewhere between 18% and 20%, which for most common prints is suitable (3DSourced 2023). However, for parts that need to sustain higher loads, it is acceptable to reach 30%, with 40% being the upper limit until no extra benefit is obtained from increasing the density (Kočí 2021). If more strength is needed, it is generally recommended to increase the number of perimeters (wall thickness) on the print. (Prusa Research 2024). Different infill patterns provide different internal structures to a part, and behave much like the internal microstructure of materials. An overview of the infill patterns for 3D printing including a description and use is shown in the Appendix, Figure 3.12. For robotic and load bearing applications, two infill patterns are covered in more detail. The gyroid infill pattern has isotropic mechanical properties (provides support in all directions), has a low weight to stiffness ratio and hence reduces material used. The honeycomb pattern is inspired by nature, great for high compressive loads, and provides low weight to stiffness ratios, but has high printing times (Kočí 2021).

2.2 Literature Review - Biomimicry

Biomimicry is an approach to innovation that seeks to solve human challenges by imitating nature's time-tested patterns, strategies, and designs. (Bloom 2024). Nature offers solutions that are efficient, resilient and adaptable. By studying it, engineers can create innovative solutions that replicate nature. This project uses a top-down approach to Biomimicry, which solves an identified technical problem by drawing inspiration from a biological system which addresses a similar problem (Ebbinghaus, Lang, and Scheibel 2023). The following section combines all Biomimicry-related literature, which will be referenced in the robot's design phase.

2.2.1 Quadrupeds: Leg Geometry and Structure

The leg geometry of a quadruped robot is an essential component for effective navigation in its environment, and directly influences the maneuverability of the body. Leg structures of quadruped robots can broadly be categorized into two types: linkage legs and scaled legs (Lu et al. 2022). Linkage legs are composed multiple hinges and beams interconnected together. Based on how these beams and joints are linked, linkages can be categorized between series and parallel. Series linkages have actuators placed in a chain like manner (see figure 2.1a). The series type offers advantages such as a simple structure, a large range of motion, and straightforward control. However, this means heavier legs, and unstable motion as the actuators have to be placed at every joint, which unbalances the center of mass. Parallel linkages have at least two independent chains that converge to a common point (see figure 2.1b). The parallel type provides advantages such as high stiffness, strong load-bearing capacity, high precision, and low inertia (Fan et al. 2024). For rough terrain, robustness and high load bearing capacity is chosen over simplicity. Designing a linkage mechanism is a complicated and involved process, so engineers typically look for pre-existing designs.

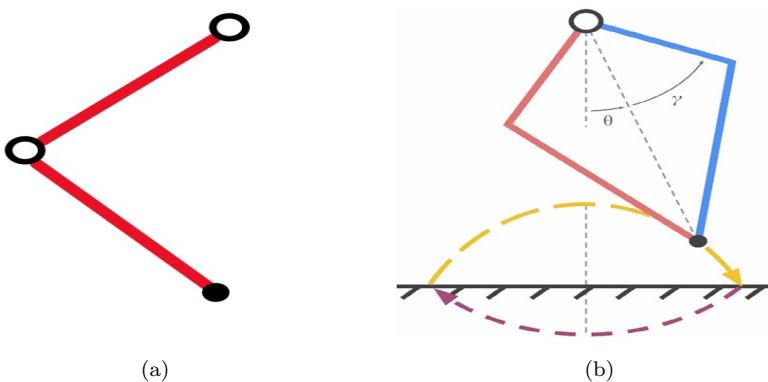


Figure 2.1: Linkage Mechanisms.

Leg topology structures are defined by different joint orientations (see figure 2.2). Knee joints have apex points that face in the direction of the movement. In contrast, elbow joints have apex points that face away from the direction of the movement. Most mammals make use of either all elbow joints, all knee joints (horses) or a combination of both (cats and dogs) (Fan et al. 2024). Having knee joints in the front legs limits the range of motion when stepping over

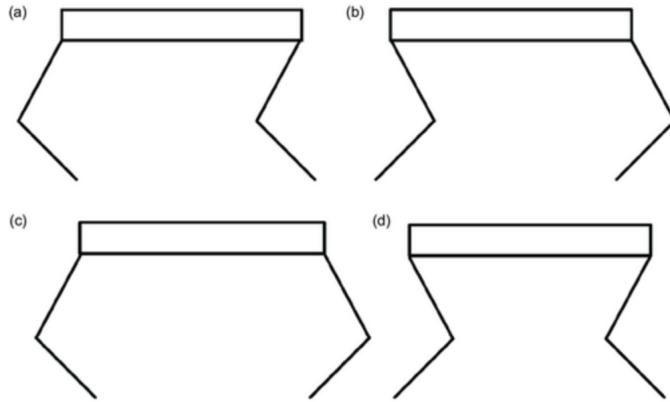


Figure 2.2: Joint formation for mammals. a) all knee joints b) all elbow joints c) front knee joints and back elbow joints d) front elbow joints and back knee joints.

obstacles, since it risks interference with the obstacle ahead (Fan et al. 2024). However, elbow joints do not face this problem and are therefore more suitable for cave exploration.

2.2.2 Paw and Shock Absorption

To improve grip on a rocky terrain, with humid surfaces, the foot structure and its material need to be considered. For the structure, a simple circular foot is ideal as it allows the robot to make contact from different angles. This multi-directional contact capability gives stability on uneven terrain (Fan et al. 2024). Using Biomimicry, inspiration for a specific foot structure can be taken from the morphology of canine feet. Canines can exert ground reaction forces up to four times their body weight per limb (Minetti 2000). Their paw structure is therefore optimized to absorb and dissipate these forces. In figure 2.3 a dog's paw is shown up close. It consists of a dermis layer, a natural honeycomb structure (Miao et al. 2017). Honeycomb structures have a high strength-to-weight ratio providing stiffness while minimizing weight. This makes the structure a preferred choice for design in industries like aerospace, automotive and robotics (Kee Paik, Thayamballi, and Sung Kim 1999).

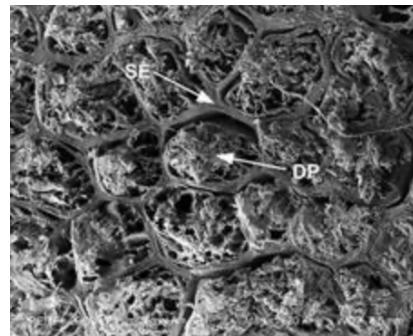


Figure 2.3: Dermis layer on a dog paw, containing a natural honeycomb structure. SE stands for stratified epithelium, forming a protective outer layer. DP stands for dermal papillae which contains fibers to increase strength.

CHAPTER 3

Evolution and Prototype Design

3.1 Design Phase - Mechanical

3.1.1 Design Guidelines

Based on the needs of both Alicia Walsh, Jitte Wagen and Tycha Buekers, the following design guidelines were produced. These outline the mechanical requirements for the prototype to function effectively, to meet the project objectives. Firstly, the size of the prototype must be constrained to ensure fitting in narrow passages, while still giving enough room for internal components. These components must be orderly arranged to maximize functionality and to leave room for component upgrades. Lightweight materials should be used like PLA to minimize structural weight while maintaining integrity. Weight distribution should be accounted for by placing heavier components, such as batteries, near the center of mass. Internal components must have resistance to wet and dusty environment in caves, hence using IP65-rated materials and sealed enclosures will keep these protected. Moreover, components must be able to function in the caves' cold temperatures. For low-light conditions, external lights or light-independent sensors like LiDAR should be used. For autonomous control, a sufficiently powerful and rechargeable battery should be included. To protect against uneven terrain and potential slipping, shock absorbing features must be integrated. Lastly, the leg must have 3 DOFs to allow for flexible and adaptive navigation through the environment. For further explanation as to why and how these guidelines are implemented, please refer to Figure A.2 in the Appendix.

3.1.2 Performance Specifications

Once the design guidelines are set, they must be converted into qualitative and measurable goals, called performance specifications. These will lay the foundation for the evaluation of the prototype once it is built. The specification are found in Figure 3.1. A reasoning and possible test methods for each specification can be found in Figure A.3 in the Appendix.

3.1.3 Leg Design Stages

Basic Leg Structure

Based on the literature found in section 2.2.1, the leg design will use a parallel linkage mechanism with three actuators. Inspiration for the rough design was taken from an open source CAD model (see figure 3.2a) found on YouTube (Panchal 2023). This model was chosen as a good starting point, particularly for its simplicity, which could easily be replicated on any CAD software. Moreover, the design closely resembles the morphology of a canine leg, with one actuator acting as the hamstring and quadriceps, and another acting as the hip flexor. The structure of the leg is composed of a few primary parts, which will often be referenced. Figure 3.2b is a labelled diagram to depict this in detail.

Design Guideline	Specification
Size Constraints	Maximum dimension of robot : 0.5 x 0.5 x 0.5 m
Internal Component Storage	Robot has at least 1L of internal storage for electronic components and around 0.3 L to spare
Overall Weight	The prototype must weigh no more than 3.3 kg
Weight Distribution	CAD model of robot must have a centre of mass within 10cm of its centre of symmetry .
Resistance to Environment	Robot is built with waterproof electrical components rated at IP65 or otherwise sealed/protected.
Low – Light Functionality	The robot must have a LED Ring Light attached to it
Temperature Resistance	Robot can perform test run in cold temperatures of 2 degrees C .
Battery Power	The selected battery must have allow the robot to run 15min, have a charge of 80% left and must be rechargeable
Schock Absorption	The prototype must have padding on the feet . It must prevent slipping, and absorb motion.
Manoeuvrability	The prototype must have a total of 18 DOFs including the body, or 12 servo motors.

Figure 3.1: Mechanical Performance Specifications.

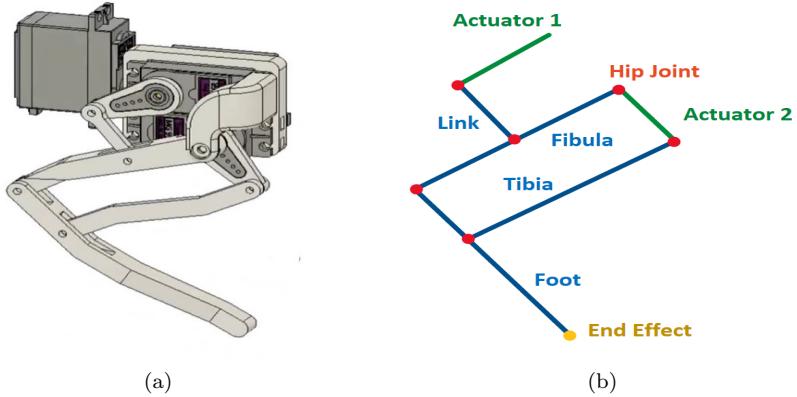


Figure 3.2: Basic Leg Structure Design.

Components and Manufacturing - 3D printing

The use of 3D printing provides a fast and cost-effective method for prototyping. It allows for multiple design iterations and quick testing. For the material, PLA (polylactic acid) was chosen because of its compatibility with most 3D printers and affordability. Unlike wood or metal, it is resistant to water, lightweight, and can withstand some deformation, preventing it from fracturing in a brittle manner, all of which are requirements for the robot.

Components and Manufacturing - Servo Motors

Servo motors were selected as the actuators for the quadruped due to their balance of cost, weight, torque, and ease of use. While high-end quadruped robots often use brushless DC (BLDC) motors for their high torque and acceleration, these were unnecessary for this project due to their cost, size, and overkill performance. The servo motors available in the Makerspace have a torque rating of 30 kg-cm, sufficient for this prototype. Considering the legs would be

approximately 20 cm long, a single motor could support a load of 1.5 kg. With two legs per side, the robot could theoretically support a maximum weight of 3 kg. This is sufficient for a prototype design.

Prototype 1: Simple 2 DOF Leg

The first prototype is a simple two-degree-of-freedom (DOF) design. It is the first design iteration, and is made for simplicity. The main components are the foot, tibia, fibula, link and hip holder. The hip holder acts as extra support for the hip joint, relieving stress on the joint, and reducing play. A plastic servo horn was used as it was immediately available in the Makerspace and good for initial testing. Components were printed with a rectangular infill pattern, and around 90% infill to maximize strength. In general, this prototype fulfilled its purpose and was testable. It was a good guide for a more focused design process.

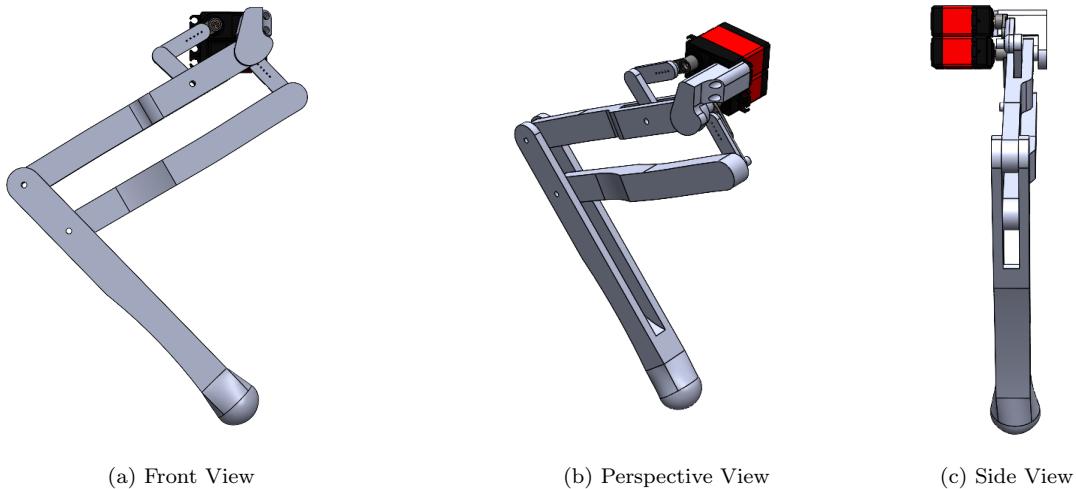


Figure 3.3: Prototype 1 Model.

Challenges with Prototype 1

1. **Misalignment and Error Propagation** Many parts had to be screwed into each other, which introduced a lot of play in the system. A small misalignment in the hip would introduce a large error at the foot. The leg was therefore unstable. This is especially noticeable at the servo horns (see Figure 3.4a).
2. **Excessive Weight** The prototype, weighing in at 200g, excluding servos, felt too heavy, most likely due to the high infill density. When tested, the inertia of the leg would cause a lot of vibration, and many of the bolts would loosen over time.
3. **Stress Concentrations** Many parts were poorly designed and had obvious areas where high stress would concentrate. For example, on the hip holder, especially, the pin holding the fibula experienced excessive stress, causing it to break during operation. (see figure 3.4b). The same happened for the pin of the link (see figure 3.4c).

Prototype 2 Features

The second prototype is the first 3 DOF design and incorporates many improvements based on feedback from a Mechanical Engineer at the Technology Center at the University of Amsterdam (hereafter referenced as "the expert"). A servo housing unit was made, to hold all servos together. This second prototype also addresses the challenges of prototype 1. Below are the key improvements.

1. **Servo Horn Upgrade** The plastic servo horn was replaced by a metal version (see figure 3.5a, greatly increasing the load-bearing capacity and reducing play, as advised by the experts.

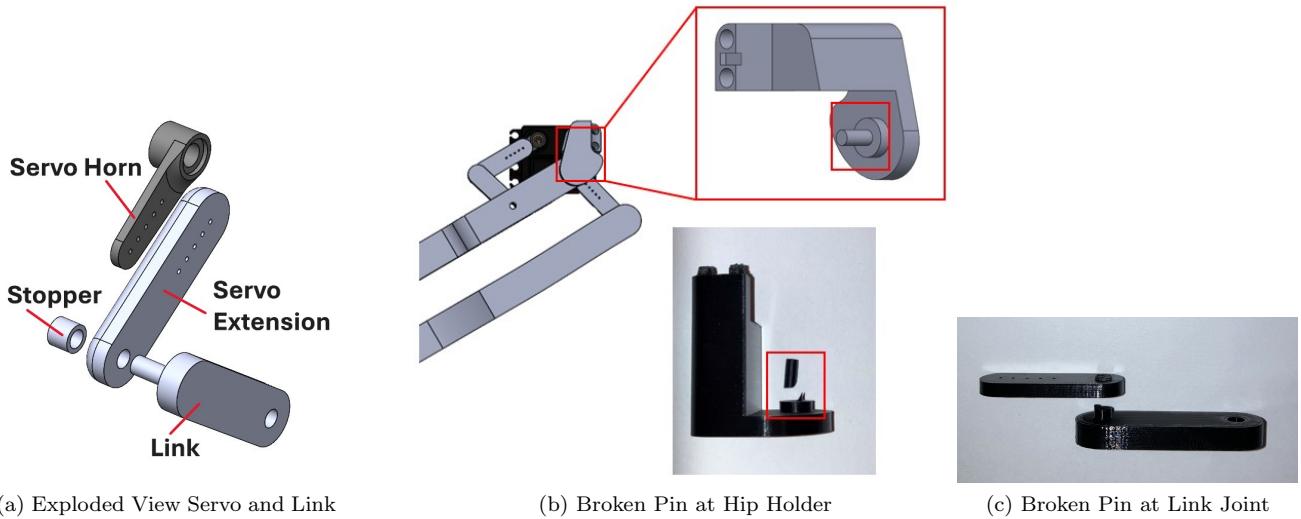


Figure 3.4: Challenges with Prototype 1.

2. **New Servo Horn Extension** An extension was added to the metal horn, improving stiffness and reducing play. Since the horn fits into the extension and isn't screwed on top (unlike the previous design, see figure 3.4a), this design is better at keeping the center of mass close to the hip, and is much more secure and rigid, see figure 3.5b.
3. **Bearings** Bearings were added to joints closest to the hip, which bear the highest loads during operation (see figure 3.5c). Joints near the knee and foot were left without bearings, as they bear less load, as advised by the experts. For extra support and rigidity, two bearings are placed per joint
4. **Infill Pattern and Density** Based on literature in section 2.1.6, the infill pattern was changed from rectangular to gyroid due to its anisotropic properties and for its ideal weight to stiffness ratio. Unlike the first prototype, where all parts had the same density, the second prototype had different infill densities for each part. High-load parts had around 40% infill for greater strength, and more outer walls were added in the slicing software. The same was done for parts that needed screws (servo housing). Other parts had infills reduced to around 15-20% to reduce weight.
5. **Weight Reduction** Cutouts were made on each part to reduce weight and material use without compromising their overall integrity (visible in figure 3.5). This was later verified by the experts. This change combined with the infill adjustments resulted in **over 40% decrease in weight** (117 g), excluding servos.
6. **Improved Mobility** The shapes of the fibula and large link were redesigned to create an asymmetrical parallelogram, increasing leg mobility as more angles were achievable, and the legs weren't restricted by their geometry (see figure 3.5d). This allowed the servos to move to a fully horizontal position without interference, increasing the leg's range of motion
7. **Reduced Size** The overall geometry of the leg design has been scaled down by 30% to bring the center of mass closer to the hip (see figure 3.5d). This will improve stability.

Prototype 3: Final Design

After another feedback session with the expert at the Technology Center, further improvements were made. Given the many iterations, only the most relevant improvements are shown. The final prototype for the leg is seen in Figure 3.8.

1. **Smaller Bearings** Bearings were reduced in size from 16x5 mm to 10x3 mm, decreasing weight and allowing for more support material around the bearings (see figure 3.7). The fit of the bearings was also made tighter as advised to reduce alignment errors.



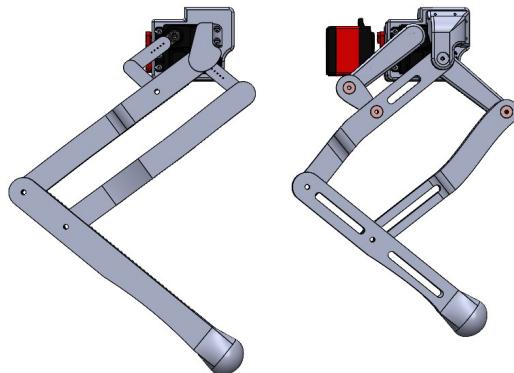
(a) Servo Horn Upgrade



(b) Servo Horn Extension Upgrade



(c) Bearing Addition



(d) Prototype 1 vs Prototype 2

Figure 3.5: Prototype 2 Upgrades.

2. Reinforced Hip Holder The hip joint was strengthened with more material (see figure 3.6). While this increases weight, the trade-off was necessary to enhance stiffness especially since this is a high load point.

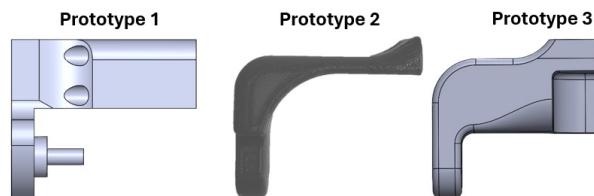


Figure 3.6: Evolution of Hip Holder.



Figure 3.7: Reduced Bearing Size.

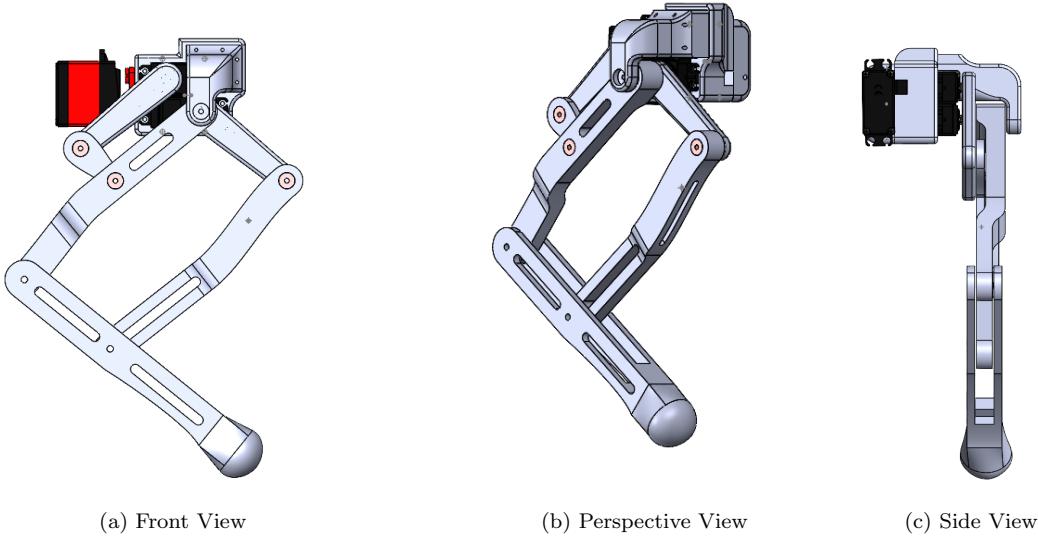


Figure 3.8: Prototype 3 Model.

3.1.4 Paw Structure Implementation

To replicate the shock absorbing feature, inspired by paw pads from dogs, we designed a spherical, hollow mold with hexagonal structures sticking out on its inner surface. This forms a negative impression of the honeycomb structure. The mold is then filled with liquid latex; a stretchy and squishy material. Besides being shock-absorbing it also provides a natural anti-slip feature due to the sticky nature of the latex. Two designs have been made:

Leaf shaped mold

Figure 3.9a is a quarter of the mold, needing four of these combined to form the final round shape around the feet. The mold is filled with liquid latex that can be removed after drying (see figure 3.9b). Then the four latex shapes are placed around the feet. Two shortcomings of this design are that the latex, despite being quite flexible, did not adhere well to the feet, and it was difficult to remove from the mold.

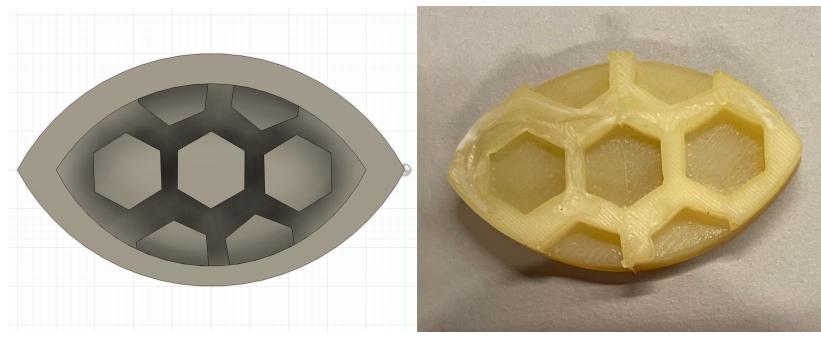


Figure 3.9: A leaf shaped mold design including a honeycomb pattern. Four molds together form a spherical socket around the foot.

Round shaped mold

Figure 3.10a is the mold where the foot is placed inside the hollow sphere, and the mold is then filled with liquid latex. Once the latex sets, it dries into a flexible rubber layer that adheres to the foot, addressing the first problem

with the prior mold design. It consists of two parts, allowing easy removal after the latex has dried (see figure 3.10a), solving the second problem with the prior design. The result shown in figure 3.10b is bouncy and sticky, indicating the shock-absorbing and anti-slip function are successfully implemented.

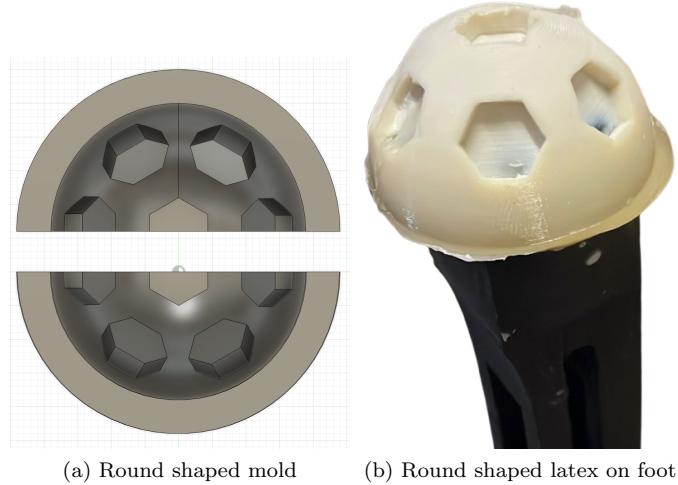


Figure 3.10: A round shaped mold design including a honeycomb pattern which folds around the foot. It consists of two parts for easy removal.

3.1.5 Body Structure Implementation

Structural requirements

1. **Robustness:** The body needs to be robust, capable of withstanding forces generated by leg movement and ground impact. To achieve this, we designed structural supports to reinforce areas where the servo motors exert the most force.
2. **Strength:** The body must support the weight of all components. We chose PLA filament for 3D printing the body and increased the infill density to 20% to enhance its strength.
3. **Lightweight:** It must be lightweight in order not to put too much strain on the servo motors. To reduce weight without compromising strength, we incorporated strategically placed holes in the design.

Functional requirements

1. **Surface area:** The body surface must be large enough to hold all electronics and sensors. The surface has to allow the servo motors to move freely. For this, we made specific cutouts.
2. **Protection:** The electronics inside the body must be protected from the environment. For this, we designed a shell that can be partially removed with magnets.

Prototype 1

In the first prototype the body consists of two parts that can be attached to each other using screws. In figure 3.11a a design of half of the body is shown. To create the full body, a mirrored design is screwed onto the three sockets at the edge. The total measurements are 20 cm wide, 40 cm long and 0.5 cm thick. The body is designed in a rectangular shape with four rectangular cutouts at the corners where two servo motors for each leg will be placed (see figure 3.11a). The legs are attached to the body at two points. The first place is by the third servo motor, which can be placed into a support on the body (see the door-like shapes in figure 3.11a). The second place is at the outer edge on the body where the socket for the two servo motors can be screwed onto the body. This setup allows for rotation of the entire leg.

Challenges with Prototype 1

1. **Stress concentrations:** The supports at the corner of the body were too weak. They broke when assembling the legs to the body. And when the robot attempted to walk the supports, connecting two halves of the body, broke.
2. **Lack of robustness:** When assembling the two parts of the body together, we noticed that the overall structure was quite bendable.

Prototype 2

When designing the second body, we added stronger supports to attach the servo motors (see figure 3.11b). Additionally, we designed rectangular wooden shapes to screw into the body for more support (see figure 3.12). The second body also contains a shell to protect the robot from moist (see figure 3.13). It consists of six parts where the two middle parts are attached to the body with magnets. There are specific holes designed in the front of the shell to allow a camera to take pictures. And there are holes designed for the distance sensors to measure through. The second design and prototype is shown in figure 3.14).

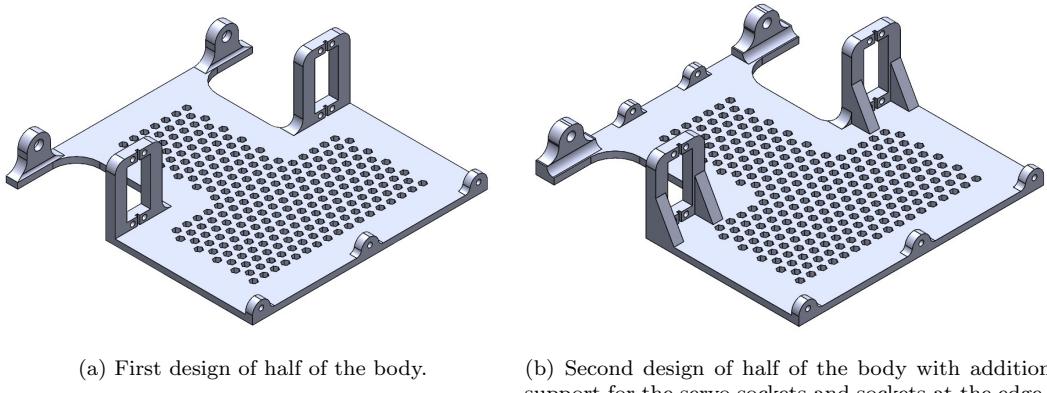


Figure 3.11: Two iterations on the body design.



Figure 3.12: Two wooden supports that are placed along the bottom of the body.

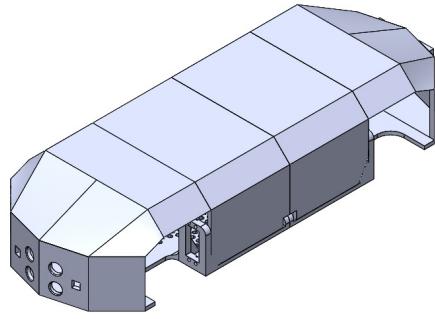
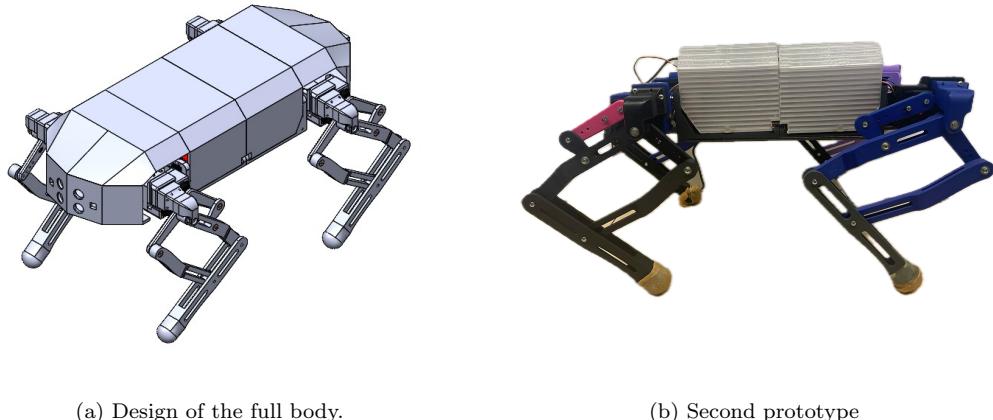


Figure 3.13: The shell of the body.



(a) Design of the full body.

(b) Second prototype

Figure 3.14: The second prototype includes latex feet, parts of the shell and runs on batteries.

Challenges with Prototype 2

1. **Weight:** The body is too heavy for the servo motors to adequately support the weight, leading to the legs bending too much when the robot attempts to walk.
2. **IMU sensor placement:** A smaller issue is that the sockets placed in the middle of the body to attach the two halves are in the way of placing an IMU sensor.

Prototype 3

The third design of the body is smaller than the previous iterations (see figure 3.15), as advised by our stakeholder Arnoud Visser (personal communication, 2025). The new measurements are 15 cm wide, 35 cm long and 0.5 cm thick. And there is space for the IMU sensor to be placed. We have yet to assemble this prototype.

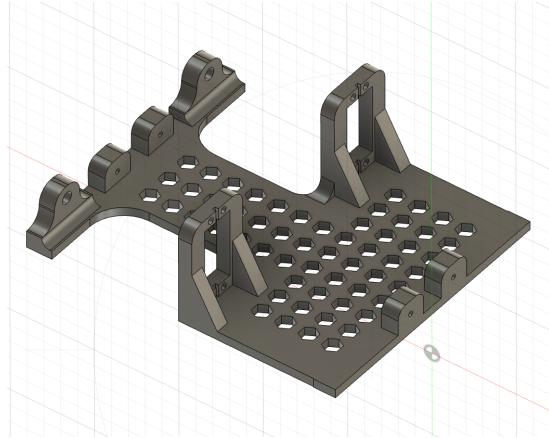
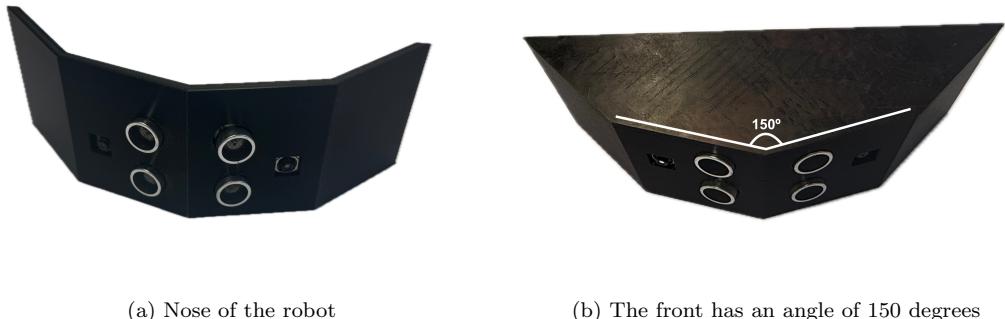


Figure 3.15: Third iteration of the body design.

3.1.6 Photogrammetry implementation

To implement photogrammetry into the robot, a dual-camera setup was selected, providing a robust solution for capturing detailed stereoscopic images. The setup consists of two Raspberry Pi Camera Module 3 units, which are mounted on the front of the robot. These cameras are positioned next to the two sonar sensors, which enables coordinated data acquisition for both distance and visual information. By using two cameras, the robot can generate stereoscopic images, which is beneficial for photogrammetry (Karami, Menna, and Remondino 2022). The dual-camera setup also offers a significantly wider total FOV compared to a single camera system. This expanded FOV provides more coverage for creating accurate 3D models of the robot's surroundings.



(a) Nose of the robot

(b) The front has an angle of 150 degrees

Figure 3.16: Camera and sensor setup on the nose of the robot.

However, the outward-facing orientation of the cameras also has a disadvantage. The camera orientation causes a small blind spot directly in front of the robot. To determine the size of this spot, specific details about the cameras and their placement are necessary. The following parameters are relevant:

- The cameras have a field of view (FOV) of 66 degrees.
- There is a 4 cm distance between the cameras and the nose's angular reference point of the robot.
- The nose of the robot has an angle of 150 degrees

Following these steps, the size of the size of the blind spot can be calculated.

In the drawing in Figure 3.17, the two cameras are indicated by points F and G . Both cameras have a field of view (FOV) of 66° , meaning 33° on each side. To calculate the blind spot, a line is drawn from the apex of triangle ABC that is perpendicular to line segment BC . Another line is then drawn from point F , perpendicular to line segment AD .

To determine the length of line segment FI , the following formula is used:

$$\sin(75^\circ) = \frac{FI}{4}$$

$$FI = 4 \cdot \sin(75^\circ) = 3.86 \text{ cm}$$

The same applies to the other camera, as everything is mirrored.

Since triangle FAI is a right triangle, the angle at point F in triangle FAI is calculated as follows:

$$180^\circ - 90^\circ - \frac{150^\circ}{2} = 15^\circ$$

Next, we need to determine the angles of triangle FEI . As it is a right triangle, the remaining two angles must add up to:

$$180^\circ - 90^\circ = 90^\circ$$

By drawing a straight line from point F that is perpendicular to line segment BA , the FOV of the camera is divided into two parts of 33° each. This forms a right triangle between points A , F , and the perpendicular line on BA , through F . From this, the remaining angle can be calculated as:

$$90^\circ - 33^\circ = 57^\circ$$

This means that angle F in triangle FEI is:

$$57^\circ + 15^\circ = 72^\circ$$

To calculate the length of line segment ED , the following formula is used:

$$\tan(72^\circ) = \frac{ED}{3.86}$$

$$ED = \tan(72^\circ) \cdot 3.86 = 11.88 \text{ cm}$$

From this, the length of line segment A can be calculated as:

$$A = (A + B) - B = 11.88 \text{ cm} - 1.03 \text{ cm} = 10.85 \text{ cm}$$

Thus, the robot's blind spot is 10.85 cm, measured from the tip of its nose.

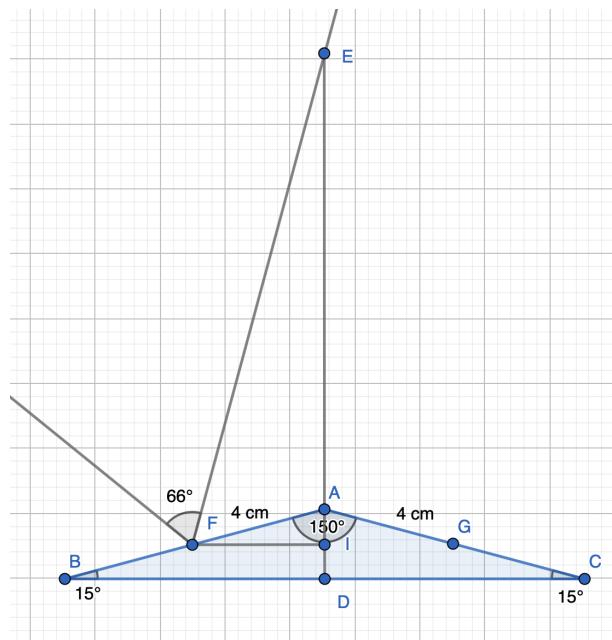


Figure 3.17: Schematic representation of robot's front end

3.1.7 Electronics

In this paragraph, the different electronic elements will be discussed, including the criteria and tradeoffs.

Overview

A complete overview of the sensors and other components that make up the electronic circuit has been created and can be seen in the figure below (Figure 3.18). Following this, the components will be introduced briefly. The sensors and their selected versions have been described as well, and reasons for their inclusion or exclusion are provided.

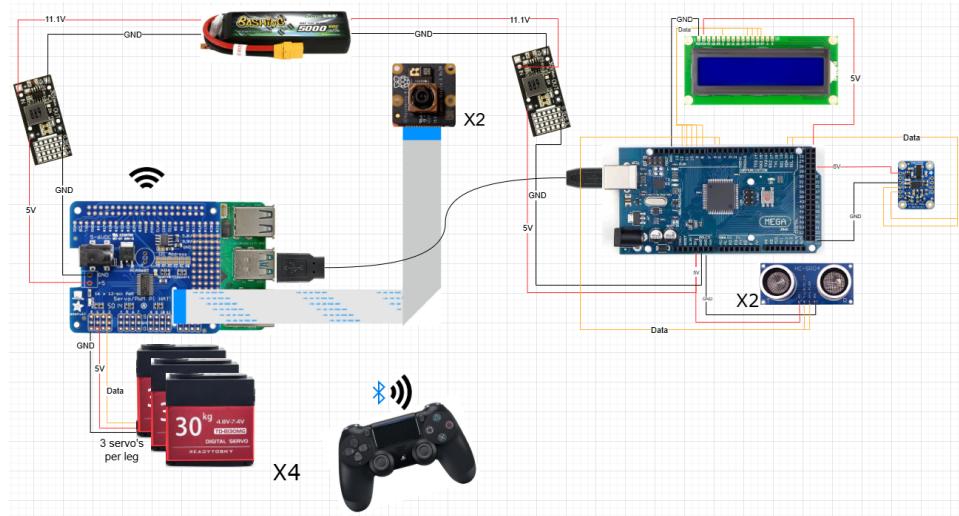


Figure 3.18: Schematic overview of the electrical components.

Raspberry Pi

A Raspberry Pi is a small, cheap computer that can be used for a lot of different tasks. It is a so-called single board computer that can easily integrate hardware with software. It can be programmed with Python, which is where the Pi in the name is coming from. The Raspberry Pi 5 has been selected for this project, as this version has CPU performance that is 2 to 3 times better than that of the raspberry pi 4 (*Raspberry Pi 5 2024*). It has 4 GB of memory, and draws a current of 5A from 5V (*Raspberry Pi 5 2024*).

Motor control board

The Adafruit 16-channel PWM/servo HAT for Raspberry Pi is an add-on board which is capable of 12-bit pulse-width modulation with a maximum speed of 1.6 kHz. The "16-channel" denotes that this board can serve up to 16 servo motors. All 16-channels of this board can be commanded from the Raspberry Pi over I_C with only two pins. In the electronics circuit of this project, this motor control board is used to control the 12 servo motors attached to the four legs. The board has its own power supply to make sure that voltage drops to the high currents of the servos will not disturb the sensors and other sensitive electronics (*Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi - Mini Kit 2025*).

Arduino Mega 2560

The Arduino Mega 2560 is a programmable microcontroller. It is used to read out the different sensors and to output information to an LCD display. It has 2560kB of flash memory, of which 8kB used by the bootloader. It has 8 kB of SRAM and 4 kB of EEPROM (*Arduino® Mega 2560 Rev3 2025*).

Sensor selection

As briefly mentioned in the theoretical framework, many sensors have been researched extensively to ensure an informed decision could be made on whether they should be included in the final prototype. These decisions have been based on a number of selection criteria, including whether they could perform their function appropriately, their cost, power

consumption and more. Their scores based on these criteria are found in Figure 3.19. For clarity, the sensors will be discussed below in the same order as they appear in the figure.

Criteria	Sensor types			
	LIDAR	Camera	Ultrasonic distance sensor	IMU sensor
Mapping	2	2	1	1
Localization	1	1	1	2
Navigation	2	2	1	1
Orientation	2	2	1	2
Stabilization	3	3	1	3
Resolution	3	2	3	2
Object detection	3	3	3	1
Performance in low light	3	1	3	3
Real time measuring	3	3	2	3
Consistency in changing orientation	1	1	1	2
Error and error propagation	1	3	2	1
Signal to noise	1	2	1	1
Weight and volume	1	3	2	3
Ease of use	1	2	3	2
Cost	1	2	3	2
Lead time	2	3	3	3
Power consumption mW	1	2	3	3
Total	31	37	34	35

Figure 3.19: Table showing sensor performance of the final sensors on criteria for selection.

- **LiDAR sensor** A wide range of LiDAR sensors exist, but few have been considered for this project. Most of the LiDAR sensors have a cost that fall outside the budget, and while many sensors have good Robotic Operating System (ROS) support, the only LiDARs that offered good Python support were the RPLiDAR A1, A2 and A3. This project only used Python and not ROS due to its complexity to master, meaning those would be the only LiDAR sensors to choose from (for an introduction to ROS and elaborate description of the considerations in this switch, see Appendix A.1). Despite the RPLiDAR versions A1, A2 and A3 being compatible with Python, the LiDAR sensor in itself is a complex sensor, and when mounted on a mobile robot with constantly changing orientations, it makes it nearly impossible to create accurate 3D maps of the environment when the LiDAR output is used for SLAM. Additional disadvantages of the LiDAR include its relatively high weight, power consumption and complexity, which makes the LiDAR an unsuited option for this robot that is constrained by weight, power consumption and complexity. These limitations outweigh its advantages in this project, which is reflected by the relatively low score shown in Figure 3.19. For the reasons mentioned above and shown in the figure, the LiDAR sensor has been excluded from the final prototype.
- **Camera** The Raspberry Pi Camera Module 3 was selected for the robot prototype. A range of factors influenced this decision, including sensor size, cost, power consumption, ease of integration, and versatility in different lighting conditions. Raspberry Pi offers several camera modules, with the Camera Module v2 and the Camera Module 3 being among the most used. The Raspberry Pi Camera Module 3 is a significant upgrade over the Camera Module v2, featuring a 12-megapixel Sony IMX708 sensor. This sensor offers a substantial improvement in resolution and low-light performance, which is important for photogrammetry. The autofocus feature is one of the standout additions in the Camera Module 3, allowing the camera to automatically adjust focus based on object distance, improving image quality in dynamic environments. The cost of the Camera Module 3 is relatively low compared to similar products on the market, making it great value for money. Despite its upgrades, the price difference between the Camera Module 3 and its predecessor is minimal. Additionally, the module is optimized for Raspberry Pi boards, ensuring seamless compatibility. With plenty of documentation and community support, it is a practical choice for developers with varying skill levels. Finally, the camera performs average to high on almost all requirements, as shown in Figure 3.19. Because of the reasons mentioned above, the Raspberry Pi Camera Module 3 was selected for the robot prototype, offering an optimal balance of

performance, affordability, and ease of integration (*Raspberry Pi* n.d.).

- **Ultrasonic Distance Sensor**

The HC-SR04 was chosen due to its intuitive usage, extensive documentation, and ease of integration, which were key after the late switch to ultrasonic distance sensors. Although it doesn't excel in all performance criteria 3.19, its range and accuracy are sufficient for simple obstacle avoidance. Its affordability, low power consumption, lightweight design, and ease of use make it a practical choice despite not being the most advanced option. If time constraints hadn't been a factor, the JSN-SR04T would have been preferred for its lower power consumption and waterproof, dust-resistant features, ideal for flint mines. However, given the time limitations, the HC-SR04 was the most suitable choice based on its cost-effectiveness and ease of integration.

- **IMU sensor**

Given the advantages of 9-DOF IMU sensor (briefly) mentioned in Sensor Comparison, a 9-DOF Imu sensor was selected; specifically the The BNO055 version of the 9-DOF IMU sensor was chosen for this project because it offers onboard sensor fusion and a Kalman filter, that most 9-DOF IMU sensors lack. This allows improved sensor readings without requiring expert knowledge of how to (manually) implement it. While the output of newer models like the BNO085 and BNO086 have the same onboard sensor fusion with Kalman filter and provide more accurate output, their communication protocol is more complex. This makes the BNO055 the preferred choice.

BNO055	Sensors		
	Accelerometers	Gyroscopes	Magnetometers
Axes	3	3	3
Bandwidth	62.5 Hz	32 Hz	10 Hz

Figure 3.20: The three sensors in the BNO055 IMU sensor.

The accuracy of the individual sensors is also a benefit of this sensor version, and its values can be seen in Figure 3.20 as well. The table outlines the three sensors in the 9-DOF BNO055 IMU sensor (accelerometers, gyroscopes, and magnetometers), and shows their axes and bandwidth, expressed in Hz. The bandwidth is a direct indicator of performance, as it indicates the maximum sampling frequency (amount of readings /s) it can handle. So the higher the bandwidth -and thus the sampling frequency- of a sensor, the more accurate it is. Accelerometers, with the highest bandwidth (62.5 Hz), measure acceleration for precise velocity and displacement determination. Gyroscopes (32 Hz) measure rotation, which requires less frequent updated sensor readings, while magnetometers (10 Hz) determine orientation relative to the Earth's magnetic field, needing the lowest frequency.

Servo motors

The motors used in the robot are the TD-8130MG Digital Servo Motors. These servos operate within a voltage range of 4.8V to 8.4V, providing a peak torque of 32kg*cm. They draw between 140mA and 200mA when unloaded, between 2600mA and 3400mA at stall. A total of twelve servos are used in the robot, with three servos dedicated to each leg. The servos enable movement in all directions and allow the robot to walk (*Tiny Tronics* n.d.).

Battery setup

To power the robot, a 5000 mAh 11.1V LiPo (Lithium Polymer) battery is used as the main power source. LiPo batteries are widely chosen in robotics and remote-controlled applications due to their high energy density and lightweight design. On average, LiPo batteries deliver the highest Watt-hour-per-kilogram (Wh/kg) ratio compared to other commonly used battery types such as Li-ion, NiMH, and NiCd. Additionally, LiPo batteries excel at delivering consistent power output over extended periods, making them highly suitable for demanding applications. Considering the cost of LiPo batteries, the robot is equipped with a capacity of 5000mAh. Despite being on the more expensive side, LiPo batteries have a great price-performance ratio. Compared to the alternative battery technologies named above, this LiPo battery provided the higher capacity at a reasonable price and was therefore the best choice for the robot (Marin-Garcia et al. 2020).

In order to determine run time of the robot, necessary data about the robot's power consumption must be discussed. The components that consume most of the power are the servo motors. According to the data sheet (*Tiny Tronics*

n.d.), a servo motor has a running current of 140mA and a stall current of 2600mA at a voltage of 4.8V. The robot is equipped with 12 motors. When all 12 motors are running simultaneously, the total current draw is calculated as follows:

$$\text{Total current} = I_{\text{motor running}} \times n_{\text{motor}}$$

$$\text{Total Current} = 140 \text{ mA} \times 12 = 1680 \text{ mA} = 1.68 \text{ A.}$$

Using the battery's capacity of 5000 mAh, the runtime can be estimated as follows:

$$\text{Runtime} = \frac{\text{Battery Capacity}}{\text{Total Current Draw}} = \frac{5 \text{ Ah}}{1.68 \text{ A}} \approx 2.98 \text{ hours.}$$

This calculation assumes that the motors operate continuously at their running current. However, in the worst-case scenario where all motors stall simultaneously, the current draw would spike significantly. The stall current for each motor is approximately 2600 mA, leading to a total current draw of:

$$\text{Total current at stall} = I_{\text{motor at stall}} \times n_{\text{motor}}$$

$$\text{Total current at stall} = 2600 \text{ mA} \times 12 = 31200 \text{ mA} = 31.2 \text{ A.}$$

Under these conditions, the battery runtime would plummet:

$$\text{Runtime at Stall} = \frac{\text{Battery Capacity}}{\text{Total Current Draw at Stall}} = \frac{5 \text{ Ah}}{31.2 \text{ A}} \approx 0.160 \text{ hours} \approx 9.62 \text{ minutes.}$$

Given that motors are unlikely to stall continuously during typical use, the operational runtime will usually fall within a range of 1.5 to 2 hours, assuming 80-90% efficiency for the LiPo battery (Marin-Garcia et al. 2020). This adjusted runtime accounts for energy losses and the slight voltage drop as the battery discharges.

To convert the 11.1V output from the battery into usable power for the Raspberry Pi components, two buck converters (step-down converters) are used. Buck converters are efficient voltage regulators that step down the input voltage to a lower, stable output voltage. In the battery setup, each buck converter reduces the 11.1V from the LiPo battery to the required voltage. One buck converter is dedicated to powering the Raspberry Pi. The Raspberry Pi, being the central processing unit of the robot, needs a reliable 5V power source to perform tasks such as processing sensor data, running algorithms, and managing communication. The second buck converter is used to supply power to the servo motor control board. This control board, commonly known as a "servo HAT," is responsible for managing the servo motors, which control the movement of the robot. The servo motors need 4.8V to operate. A schematic overview of the entire electronics system, including the integration of the LiPo battery, is provided in Appendix A.3.

3.2 Design Phase - Software

3.2.1 Design Guidelines

Based on the needs of Alicia Walsh, Jitte Wagen and Tycha Buekers, the following software design guidelines were produced. These outline the software requirements for the prototype to function effectively and meet the project objectives. The robot should be able to walk, in order to achieve this a walking engine needs to be build. The robot should be able to collect data while in the mine, for this a photogrammetry control sequence needs to be implemented. The robot should be able to move over various terrains, for this a dynamic step adjustment needs to be learned through reinforcement learning. Lastly, the robot should be able to autonomously navigate a cave and thus avoid obstacles, for this some obstacle avoidance software needs to be written.

3.2.2 Performance Specifications

Once the design guidelines are set, they must be converted into qualitative and measurable goals, called performance specifications. These will lay the foundation for the evaluation of the prototype once it is built. The specification are found in Figure ???. To find these, please see Figure A.6 and Figure A.7 in the Appendix.

3.2.3 Walking Engine Implementation

The term "walking engine" encompasses all components that enable a robot to walk, including the servo steering and dynamic adjustments of the feet to maintain balance. The foundation of our walking engine relies on inverse kinematics. Initially, we implemented inverse kinematics for a single leg in a 2D plane. We derived the analytical solution for the inverse kinematics, meaning we formulated equations to determine the joint angles based on the desired foot position. To validate this model, we visualized the leg using matplotlib, as illustrated in Figure 3.21.

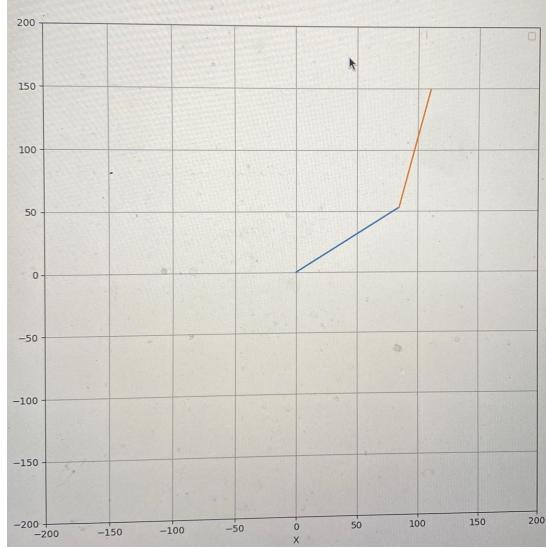


Figure 3.21: 2D Leg Kinematics Visualization.

The subsequent step involved extending the inverse kinematics to 3D for all four legs and the body. Existing research on quadruped inverse kinematics informed our implementation, which varied slightly from the referenced methodology (Sen, Bakircioglu, and Kalyoncu 2017). Each leg comprises three servos: two in the shoulder and one in the elbow. Consequently, the inverse kinematics for each leg needed to yield three joint angles to position the servos correctly, as depicted in Figure 3.22.

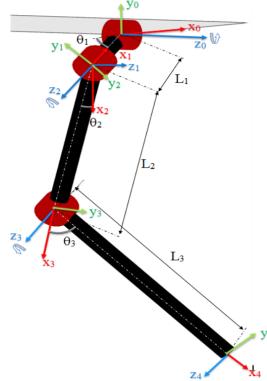
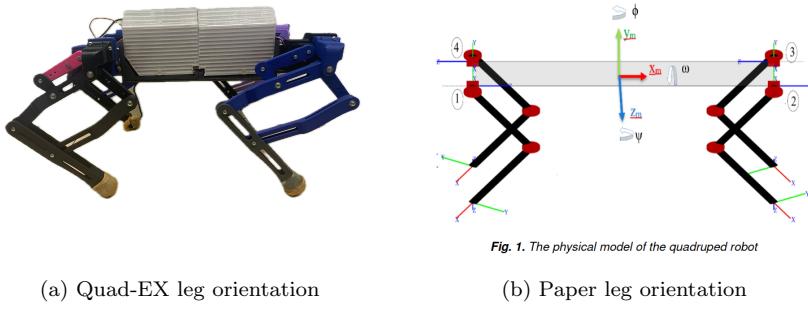


Figure 3.22: Inverse kinematics and coordinate system of a leg.

The inverse kinematics differed for the left and right legs due to the mirrored orientation of the servos. This distinction diverged from the methodology described in the referenced paper, as our leg orientations (Figure 3.23) necessitated a different approach to inverse kinematics, accommodating the right and left sides of the body respectively.



(a) Quad-EX leg orientation

(b) Paper leg orientation

Figure 3.23: Leg orientations.

The body inverse kinematics involved calculating the positions of all twelve degrees of freedom (three servos for each of the four legs) to facilitate movement along the yaw, pitch, and roll axes. We moved from matplotlib to Rerun, a state-of-the-art robotic visualization tool, due to matplotlib's limitations in handling 3D movements. Visualizations of the yaw, pitch, and roll axes using Rerun are presented in Figure 3.24.

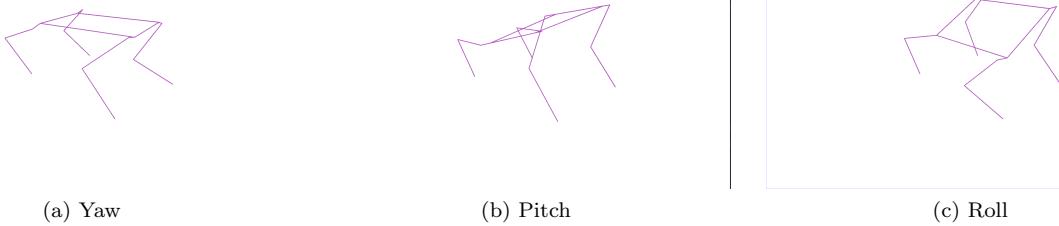


Figure 3.24: Body inverse kinematics in rerun.

With the inverse kinematics established for both the legs and the body, the next step was to integrate these components to enable walking through the implementation of a gait. A gait refers to the locomotion pattern of a quadruped robot, typically comprising two phases: swing and stance. During the swing phase, the foot moves through the air to its next position, while during the stance phase, the foot contacts the ground, propelling the robot using ground reaction forces. The gait alternates ground contact between diagonal legs, as shown in Figure 3.25, where the front left (FL) and back right (BR) legs initiate the stance phase, while the front right (FR) and back left (BL) legs commence the swing phase.

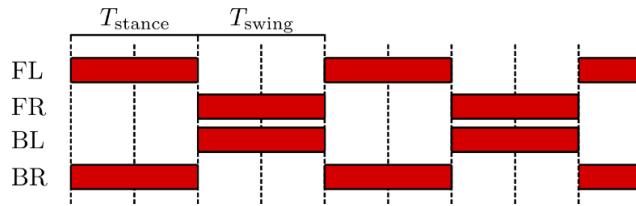


Figure 3.25: Stance and swing phases of a trot gait.

Both phases constitute a foot trajectory, described by a 10-point Bézier curve, which is advantageous due to its capability to minimize impact upon touchdown (Zeng et al. 2019). A representation of a Bézier curve is provided in Figure 3.26.

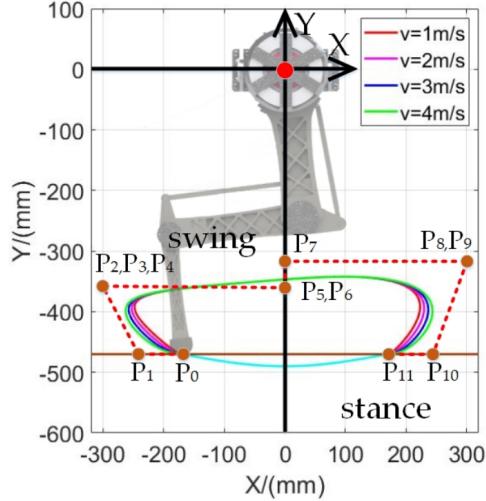


Figure 3.26: The Bézier curve trajectory.

However, it became apparent that walking on uneven terrain posed significant challenges, as the walking engine did not account for individual step adjustments essential for maintaining balance on such surfaces. Dynamic step adjustment can be achieved using reinforcement learning (RL) on IMU data, as demonstrated by (Rahme et al. 2020). Reinforcement learning involves an agent making adaptive decisions in a dynamic environment to maximize a reward signal. For our application, this implies a virtual model of Quad-EX learning to adjust the walking engine parameters to traverse uneven terrain effectively in a simulation. This promising approach only requires an IMU. We identified a potential challenge in that RL would necessitate many iterations to learn the desired behavior, and the current walking engine was not optimized for AI. To enhance efficiency, we adopted components from the SpotMicroAI walking engine, which was already optimized for RL. These optimizations included leveraging linear algebra for matrix calculations, suitable for GPU execution. Additionally, the inverse kinematics were standardized for all four legs rather than being distinct for left and right legs, as shown in Algorithm 1.

Algorithm 1 Inverse Kinematics Function

Require: (x, y, z) : Target position
Ensure: $(\theta_1, \theta_2, \theta_3)$: Joint angles

$$F \leftarrow \sqrt{x^2 + y^2 - l_1^2}$$

$$G \leftarrow F - l_2$$

$$H \leftarrow \sqrt{G^2 + z^2}$$

$$\theta_1 \leftarrow -\text{atan}2(y, x) - \text{atan}2(F, -l_1)$$

$$D \leftarrow \frac{H^2 - l_3^2 - l_4^2}{2 \cdot l_3 \cdot l_4}$$

$$\theta_3 \leftarrow \text{acos}(D)$$

$$\theta_2 \leftarrow \text{atan}2(z, G) - \text{atan}2(l_4 \cdot \sin(\theta_3), l_3 + l_4 \cdot \cos(\theta_3))$$
return $(\theta_1, \theta_2, \theta_3)$

Despite these optimization efforts, training and implementing the RL algorithm for dynamic step adjustment proved unfeasible within the project timeframe.

3.2.4 Navigation

SLAM Algorithm

The SLAM algorithm is the most ideal solution to use for navigation for the intended environment, as it solves the challenges posed by a GPS-denied, unknown location (no a priori map of the environment). It builds a map of the unknown environment while simultaneously localizing itself within it. When the robot is able to achieve this, it can also navigate effectively, and find the optimal path to a location and through the environment riddled with obstacles.

However, implementing the SLAM algorithm adequately comes with its challenges, which are aggravated by both the complexity of the environment as well as by the robot itself. These difficulties include:

1. **Data association**: difficulties in matching currently observed landmarks with previously observed landmarks, that may arise due to a number of different reasons (Taheri and Xia 2021):
 - (a) when the same landmark is observed, but seen from a different orientation.
 - (b) when the algorithm is unable to distinguish different landmarks that have similar features. This is a problem in environments where the different landmarks are alike in terms of their appearance and features, in cave-like environments.
 - (c) when actual movements deviate from the expected movements.
2. **Hardware uncertainty**: hardware errors and noise that result in inaccurate sensor readings. This may lead to inaccurate estimates of the pose, but also of landmarks.
3. **Location uncertainty**: uncertainty in the location of a robot, that arises from the existence of multiple paths leading from one location to another. This may lead to the robot coming back to a point it previously visited, without recognizing it, since it did not arrive there by following the known path to the previous location.
4. **Sensor noise and observation error**: imperfections in sensor data due to environmental conditions, such as lighting changes or sensor limitations, propagate the errors in sensor data over time. This leads to the quality and accuracy of the map and the localization to degrade. An example of this type of difficulty that is likely to arise in the environment this robot operates in, is insufficient or inaccurate lighting that results in shadows forming. When the same location is measured with different shadows, it has a high chance of being interpreted as a different location.
5. **Time complexity**: the SLAM algorithm must handle navigation, mapping and localization simultaneously, any delay in the computation of any of these tasks should be avoided, as they depend on each other; the mapping at time-frame $t=0$ should be linked to the computed location at $t=0$. However, when the number of landmarks increases, this becomes an increasingly apparent issue; more computations are required when the number of landmarks increases, and thus the time complexity increases with the computational complexity.

While some challenges are more prominent and important than others, all highlight the complexities of achieving accurate results with the SLAM algorithm, especially for this robot. Some challenges may be negligible, while others may form a considerable hurdle. These issues stem from the robot's gait, which affects the sensor data at changing orientations, and the monotony of the environment, which complicates landmark recognition. Since these challenges arise from constraints of the robot and the environment, they are inherently difficult to solve. Therefore, effective path planning and navigation may not be feasible, but simpler inertial navigation using the IMU sensor is possible.

Inertial Navigation

The information gathered from the IMU sensor can be used for motion tracking, stabilization, and navigation, e.g. for walking out of the cave via the same path taken to get to the current location. With inertial navigation, the location relative to the starting position is known at each point in time. Therefore, the way to any point within the environment can be determined at each time, as long as it's known where that point is relative to the starting location. Also the way back to the starting position can be extrapolated in this way from the robot's current position and its relation to the starting point. In this way, a simple navigation is possible.

It also allows the end-user to keep track of which path the robot is taking, and if it is heading for a place that is of little interest to the user, the robot can be led to another location within the environment with this inertial information. To ensure the end-user can exert this control over the robot, and can lead it wherever the user wants it to go, the option for remote control using a controller has been implemented as well.

Remote Control for Navigation

Remote control over the robot is established with a PlayStation 4 controller that is to be wirelessly connected to the robot via Bluetooth. Establishing wireless connection should be possible by putting the controller in pairing mode. For the robot to detect and react to button presses and joystick movement, input control and handling is needed. This is handled by the package Evdev (*evdev 1.7.1 2024*). Evdev is a minimal input library with no added dependencies

or functionalities other than input handling; it is used for handling input events from input devices, in this case the PlayStation controller. It provides raw and low-level input handling such as key presses or joystick axes, thereby also providing control over how the input events are processed and which behavior it generates.

CHAPTER 4

Testing Phase

4.1 Test Methods

Based on the performance specifications from section 3.1.2 and section 3.2.2, a series of tests were formed to assess the robot performance. Only the most relevant tests are included in this section. For a list of results of all tests, please refer to the Appendix (reference). Each test aims to target a specific design guideline.

4.1.1 Test 1: Foot Grip

This test aims to verify the effectiveness of the paw design, with objective to maximize the slip angle on a ramp. With more friction at the feet, the robot can better navigate its terrain. A wooden ramp was set up for the robot to be placed on with four feet. For this test, all servos were turned on and directed to their home position. The test was broken down into two parts, one with the latex paw design and one without. With each test, the ramp angle is progressively increased until the robot begins to slide.

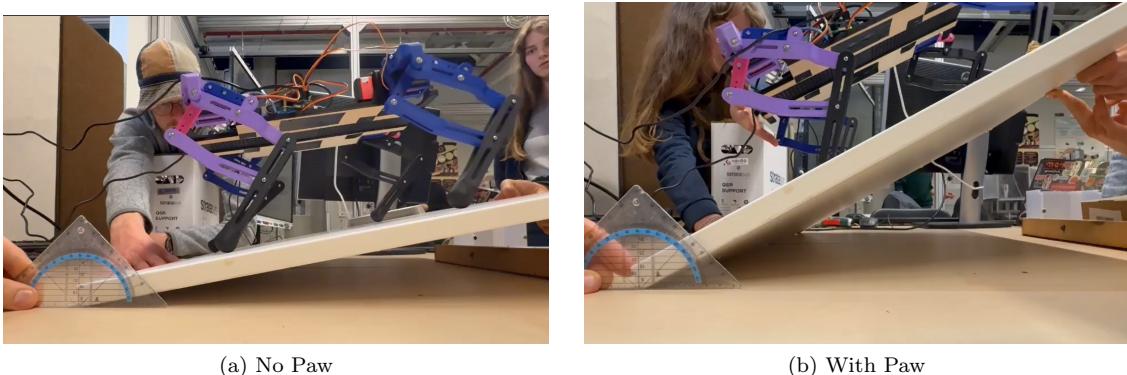


Figure 4.1: Foot Grip Tests.

From this experiment, the paws excelled in performance, with a slip angle of 34 degrees, against the 10 degrees slip angle without the paws in place. The paws are therefore **3 times more effective** at increasing grip.

4.1.2 Test 2: Basic Movements

This test aims at verifying all the degrees of freedom for the robot, with objective to complete all basic movements. The test is divided into full body motion and single leg motion. For the single leg, the ability to pan in all 3 planes (see Figure 4.2). For the body, the ability to pitch, yaw and roll is tested (see Figure 4.3). Note that is a test to verify the inverse kinematics; no motion is pre-programmed.

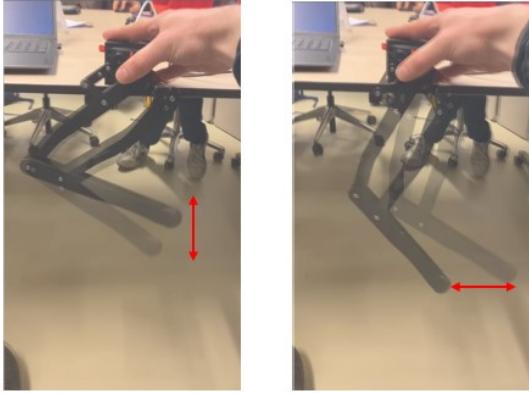


Figure 4.2: Individual Leg Testing.

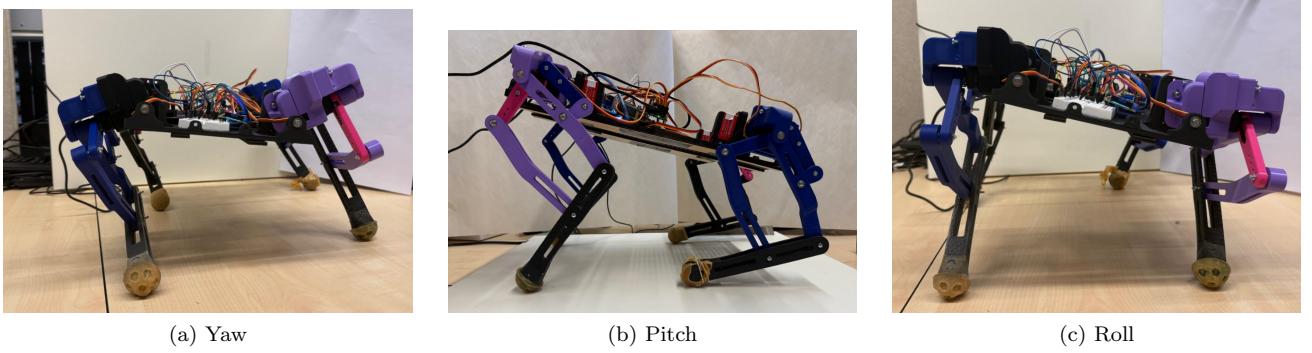


Figure 4.3: Basic Motion Tests.

The results indicate that the 3rd DOF of each servo motor (servo fixed on the body) is the least effective. This is especially noticeable in Figure 4.3a, which shows the robot struggling to perform the Yaw Motion. On the other hand, all motion that utilizes the two other DOFs, namely roll, pitch and planar motion of the individual leg is satisfactory. There are two main reasons for this, the first being faulty calibration on the servo motors attached to the body (PWM motor is offset from the real angles), causing an error between the input control and the physical output. The second reason may be that the 3rd servo motor cannot handle the weight of the body. The leg structure, being a parallel linkage, functions like a differential mechanism, essentially allowing the torque to be almost always shared between both motors. This isn't the case for the 3rd DOF motor.

4.1.3 Test 3: Walking Engine

To test the performance of the walking engine, a test was conducted in which the robot was programmed to walk forward over a flat surface. The criterion for successful completion of the test was defined as the robot being able to traverse a distance of 1.5 meters in a reasonably straight line. Although this may appear to be a rudimentary assessment, it serves to validate the trot gait and to ensure that the robot maintains sufficient stability to prevent tilting or falling. Initial attempts were unsuccessful. Through calibration of the robotic legs and the implementation of a pitch offset to compensate for a slightly uneven weight distribution, significant improvements were achieved. Following these adjustments, the robot was able to successfully complete the test, demonstrating both effective gait operation and body stability.

4.1.4 Test 4: Center of Mass

This test verifies that the center of mass is well aligned with the symmetrical center of the robot, with objective being to minimize their distance apart. The test set up consists of the body, with its legs and its internal components, hung up using a string that is to be attached to the symmetrical center of the body. This gives a simple and visually

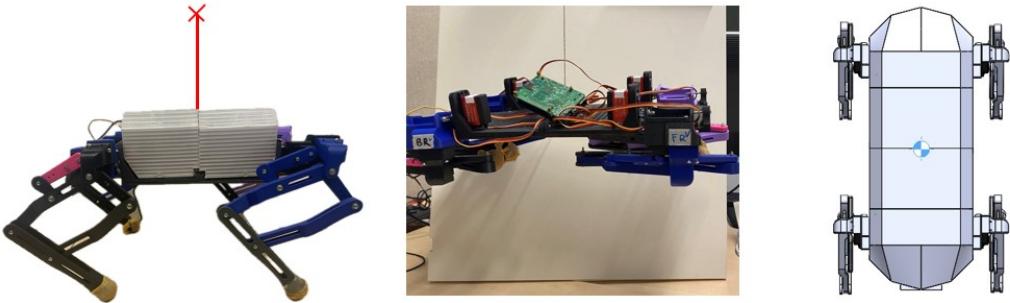


Figure 4.4: Center of Mass Tests.

intuitive way to observe if the center of mass is at the center. A protractor is then used to measure the angle of tilt. For further testing, the center of mass was also calculated for full CAD Model, see Figure 4.4

The simulation results show that the center of mass is within the 10 cm of the symmetrical center, only slightly offset due to some asymmetry in the back cover, where the LCD screen is to be placed (these are slightly heavier than the camera and the distance sensors at the front). When testing the model physically, an offset of the COM was observed, and the prototype would slightly tilt, by no more than 5 degrees. This was most likely due to the placement of the internal components, which at the time was not well organized. However, there was no major offset observed which would greatly impact the robot's stability.

4.1.5 Test 5: Photogrammetry

To test the photogrammetry potential of the robot, an experiment was done to evaluate how effectively the camera could capture the necessary data for generating a 3D model.

A simple object, a bottle of glue, was selected for the test due to its recognizable shape and surface detail. Using the Pi Camera, multiple images of the object were captured from various angles, making sure all sides and perspectives were covered. Consistent lighting conditions were maintained to minimize shadows and reflections that could interfere with the quality of the 3D model.

After capturing the images, they were imported into Agisoft Metashape, a photogrammetry software. Within the software, the photos got aligned and a dense point cloud was created. The result was a detailed 3D model of the object was generated. The process, from photo alignment, to creating a dense cloud, to the final textured model is displayed in Figure 4.5. As can be seen in Figure 4.5c, the model does display some flaws. There are some holes in the final textured model. this is likely because of an insufficient image overlap at that part of the model.

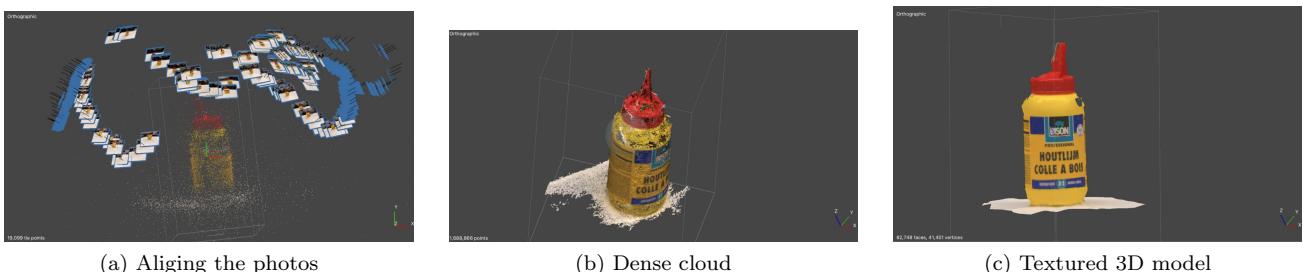


Figure 4.5: Photogrammetry tests.

CHAPTER 5

Stakeholders

5.1 Stakeholder Identification

Stakeholders hold an important role as they bring a unique perspective in any project's development. To identify the relevant stakeholders, a Stakeholder Map (see Figure 5.1) was created with axes representing power against support. The nature of the project is twofold. On one side, high financial interest, often from influential stakeholders, can allow for accelerated development. On the other hand, expertise in cave exploration, a rather niche field, can lead to a better design, high quality prototype. These axes ensure that both aspects are compared. From the map, the main stakeholders chosen were Archeologists and Technical Developers, as these were more easily approachable. A total of five stakeholders interviews were conducted. Two of them bring valuable expertise in archaeology and photogrammetry: Jitte Waagen and Alicia Walsh. Tycha Buekers regularly works in quarries and mines in Limburg, providing detailed information on the environment we focus on. Arnound Visser and Mels van Eck provide more technical expertise.

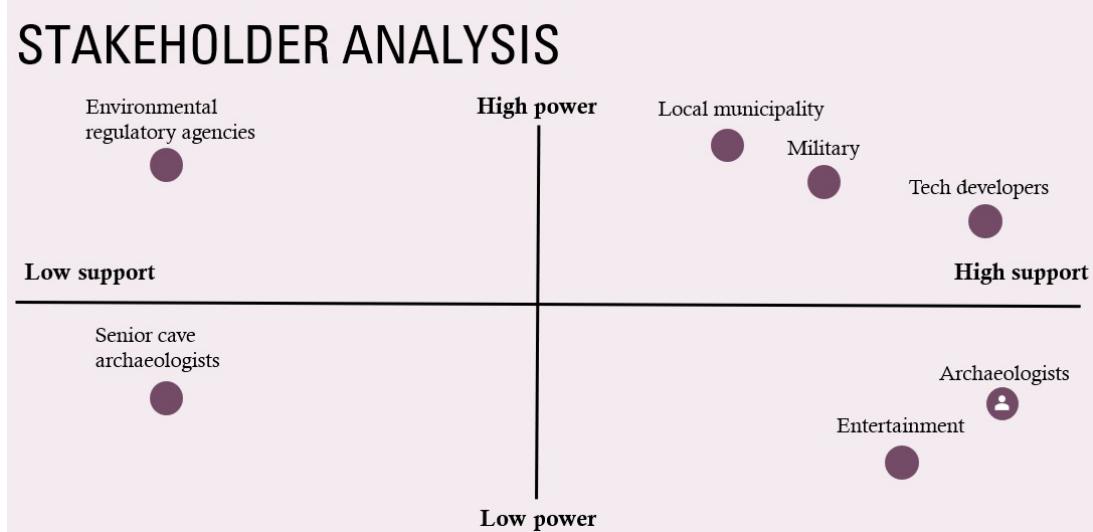


Figure 5.1: Stakeholder analysis.

5.1.1 Jitte Waagen

Digital archaeologist specialized in photogrammetry and 3D mapping.

Contact Information:

- Email: J.Waagen@uva.nl

Jitte gave a lecture that inspired us to design a robot for cave exploration, he mentioned problems that archaeologists face when documenting cave-like environments and that required solutions.

In the interview, conducted by Margriet, Jitte provided insights into those issues that could be addressed in our project, thereby shaping the direction and focus of the application of our robot. He elaborated on the problems and emphasized the inaccessibility of areas within caves during exploration. Narrow passages prevent human explorers from reaching areas that need to be documented, making size an essential component of the robot to consider. Another environmental constraint is its complex, uneven ground, which should not hinder the robot's mobility. From the interview, it became clear that designing a robot small and compact enough to traverse tight spaces, yet stable enough to navigate uneven terrain, was a top priority.

He also helped us with the practical requirements our design should meet for photogrammetry. His suggestions were to use a 3D LiDAR sensor and a camera with consistent lighting. The LiDAR would serve to complement the camera for capturing spatial data. The suggestion to ensure consistent lighting was particularly useful, as it made us aware that accurate 3D maps can only be generated when the photos can be overlapped, which can only happen when taken under uniform light conditions.

Additionally, he raised concerns about maintaining connectivity and retrieving the robot from unreachable areas. A safety cable would provide a good solution for power, data transfer and retrieval. Unfortunately, time did not allow implementation of Jitte's suggestions, but we gained a good understanding of the environment, the tangible problems we could address, and the importance of size and stability, and which issues need to be considered.

5.1.2 Alicia Walsh

Archaeologist.

Contact Information:

- Email: A.Walsh@uva.nl

Alicia Walsh is an archaeologist with relevant experience with photogrammetry. She also occasionally participates in cave exploration projects and is therefore familiar with the challenges one might encounter on cave exploration projects. This information is essential in designing the prototype to effectively handle or overcome the relevant challenges.

A minor inconvenience regarding the cave conditions Alicia encountered on her projects is the dust and moist present in caves. When dust gathers on sensors and on the camera, the data gathered includes some degree of noise as a result. Therefore, the aim is to seal off – or otherwise protect- the sensors from being affected by moisture and dust. This does, however, add a layer of complexity to the robot's design.

Some more prominent challenges were also raised, of which one major challenge regarding photogrammetry is adequately illuminating the area that is photographed. This requires a light that is bright enough and positioned at the right angle when photos are taken, so that no shadows are cast on the area. Protruding parts cast shadows that obscure the structure of surrounding areas and are therefore to be prevented. Another major challenge in photogrammetry is capturing photos with enough overlap between them to make a complete map of the cave. Alicia told us that an overlap of 60 percent is needed to overlay them and ultimately create a complete 3D representation of the cave.

To solve these photogrammetry problems, a ring light with appropriate brightness will be installed around the camera, so the light is cast directly on the area that the camera captures. This will provide the necessary illumination. An algorithm that measures the degree of overlap between the photos can be used to predict when the appropriate degree of overlap needed is reached and when the next photo needs to be taken. This will solve the problems that arise when the mapping is done manually and eliminate the use of manual mapping entirely.

5.1.3 Tycha Buekers

Archaeologist, specialized in 3D scanning.

Contact Information:

- Email: tycha.buekers@hotmail.com

Tycha Buekers focuses on 3D scanning in the prehistoric flintstone mines of Valkenburg. These mines, created over 5,000 years ago, are the subject of research aimed at understanding the ancient miners. Questions include: How old were they? How tall? What kinds of tools did they use? So far, 81 out of more than 2,000 mines have been mapped. In two of the mapped mines, remains of rope were found, indicating that the miners used rope to pull objects around the mines. The flintstone mines consist of long, narrow tunnels, with no steep drops. The height of the mines ranges between 0.5 meters and 1.5 meters. Tycha provided us with a 3D map of the mine, obtained by their archaeological data collection so far, see figure 5.2.

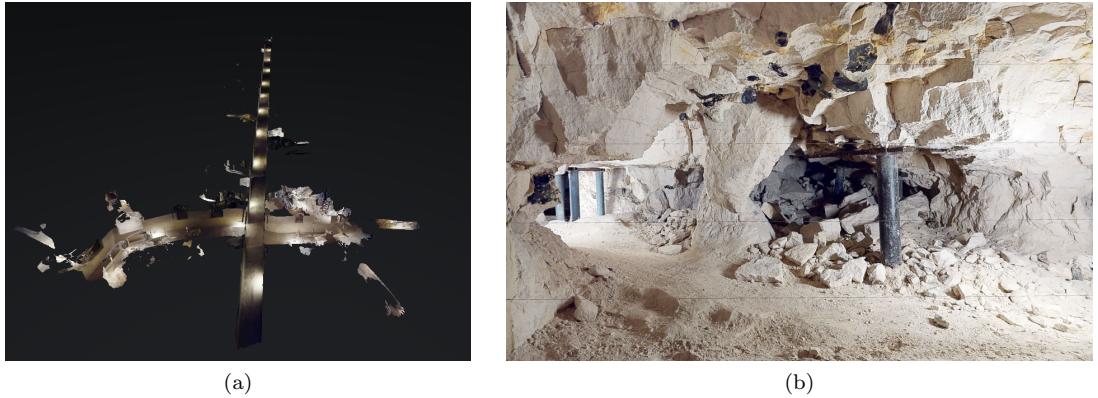


Figure 5.2: Rijckholt Prehistoric Flint Mine 3D map.

Tycha considers the use of walking robots to be a useful solution for mapping the mines, since exploring such confined spaces is labor-intensive. She also proposed a solution for maintaining communication with the robots: setting up a local Wi-Fi or Bluetooth network in the mines. And similar to Alicia Walsh, she advised being cautious with the sensors due to the high humidity levels in the mines.

5.1.4 Arnoud Visser

Roboticist, founder of the Intelligent Robotics Lab at the UvA.

Contact Information:

- Email: a.visser@uva.nl

Arnoud Visser provided us with insights regarding the upsides and downsides of different robot designs. He showed us two quadruped robots. The first robot, Spotmicro, had a leg design similar to ours, but with a servo motor placed halfway up the leg. A lower center of mass means less force is needed to lift the leg, increasing efficiency. The second robot, mimipupper2, was very lightweight, but could not carry as much weight. This is ofcourse a trade-off that should be considered. Arnoud also advised us to use two batteries instead of one—one for the servo motors and another for the Raspberry Pi. This will improve power management. Since robot weight is one of the main challenges we face, he offered additional advice on this topic. Instead of using a 3D-printed shell, we could use a plastic bag to protect the robot from moisture. This will lower the weight. Additionally, making the body of the robot smaller is another option to further reduce its weight.

5.1.5 Mels van Eck

Electrical engineer at the UvA Technology Centre.

Contact Information:

- Email: m.r.vaneck@uva.nl

Mels van Eck provided feedback on the robot's power supply. The main issue with the robot's power supply is that the servo motors do not receive enough current to support the robot's body. As a result, the robot collapses when under load.

The current buck converter, rated at 5A and 5V, cannot provide adequate current for the robot. This issue becomes particularly apparent when all twelve motors are active and under load. Mels explained that in electric motors, an increase in voltage raises speed, while an increase in current raises torque. To ensure proper operation, the robot requires additional current. The running current of a servo motor is 140 mA without load, but this value rises significantly under load. Accurate measurement of the robot's current consumption is essential.

To determine the required current, Mels proposed the use of a lab power supply. The lab power supply can be connected to the Servo HAT and set to 5V. It will then deliver the necessary current based on the robot's demand. The total current consumption can be read from the power supply, and the measured value should be multiplied by at least 1.5 to account for peak currents. The Servo HAT is unlikely to be a bottleneck, as it typically supports high currents. However, limitations in the screw terminals could potentially restrict performance, though this is rarely a primary issue.

The addition of a separate battery for the Raspberry Pi was suggested by Mels, to consistently provide it with enough current. If the servo motors draw excessive current from a single battery that is also powering the Raspberry Pi, insufficient current may reach the Pi. This can result in unsafe shutdowns that risk damaging the system.

Mels suggested that the best solution for the problem involves replacing the buck converter with a model capable of delivering higher currents. Connecting two buck converters in parallel was also a possibility to increase current delivery to the PWM controller. For improved cable management and power distribution, the use of breakout boards or Wago connectors was recommended. Alternatively, two PWM controllers could be stacked, with each receiving power from a separate buck converter.

CHAPTER 6

Conclusions and Discussion

6.1 Design Evaluation

The final design does not fully achieve our project objectives, however, it attempts to address the problem statement and significant progress has been made. It ultimately fails to navigate rugged cave terrain and conduct autonomous photogrammetry data collection missions. Although the leg design has undergone several iterations, further improvements are necessary. The primary issue with the current design is the presence of play, which results in a lack of stiffness and an unstable end position for the foot. When deployed, the leg is subject to numerous vibrations. The presence of play exacerbates these vibrations, progressively reducing leg stiffness. The use of bolts and nuts at each joint, along with excessive tolerances from 3D printing, likely contribute to this problem. While the addition of bearings has considerably mitigated the issue, switching to materials and manufacturing methods that allow for tighter tolerances (such as aluminum or resin) could reduce play to negligible levels. The walking engine was the most sizable and complex software component of the project. Despite the inverse kinematics, step planner, and various other aspects of the walking engine functioning flawlessly in visualization, real-world conditions introduced numerous additional challenges. As previously mentioned, joint play is a significant problem. This issue is extremely difficult to compensate for in the code, leading to unexpected behaviors as the legs deviate from their intended movements. Although we utilize several sensors and can access their data, we have not successfully integrated them into the system to function seamlessly together. The distance sensors determine whether the robot moves forward, left, or right based on obstacle detection. This system operates much more effectively in the virtual robot visualization compared to the actual robot. Additionally, one of the problems with the servos is that the torque is lower than it could be due to insufficient current. This power issue results in the robot being unable to fully utilize the servos' power, hindering basic actions such as standing up from the ground. There are obviously many 'weaknesses' which mostly stem from the project being unfinished but a quadruped is still fundamentally a good choice for cave exploration and data collection as shown in this report. The main strengths of the robot are definitely the design of the legs and the walking engine, while these aspects are both incomplete they form a solid basis to further build upon.

6.2 Performance Assessment

To evaluate the performance of the final prototype, an assessment method was created. The method is based on the performance specifications created in Section 3.1.2 and Section 3.2.2, condensed down to "performance indicators". Each indicator is given a grade out of 3 based on how successfully it met the guidelines from the testing phases, with 0 being not met at all, 1 being very poorly met, 2 being somewhat met and 3 being fully met. Furthermore, each specification is given an associated weight to scale the grade based on its level of importance. Grading weights were decided with navigation and locomotion being more relevant than data collection. A final combined score is then attributed to the overall performance of the robot.

Performance Indicator Mechanical	Grade	Weight	Weighted Grade
Max Build Volume: 0.5m x 0.5m x 0.5m	3	3	9
Dedicated Storage Volume: 2 Litres Spare Storage Volume: 1 Litre	2	1	2
Max Weight: 3.3kg	3	3	9
18 DOF Motion	2	3	6
Shock Absorption Padding	3	2	6
1h Battery Life	2	3	6
Temperature Resistant	3	1	3
Waterproof IP65	1	3	3
Low-Light Functionality	0	2	0
Equal Weight Distribution	2	2	4

Mechanical Performance Grade: = 48/69 = 69%

Software Performance Grade: = 74/93 = 79%

Overall Grade: 125/171 = 73%

Performance Indicator Software	Grade	Weight	Weighted Grade
LCD screen	3	2	6
Controller	2	2	4
Battery Warnings	0	2	0
Bezier foot trajectory	2	3	6
IMU updates	3	1	3
Trot gait	3	3	9
Creep gait	3	1	3
Resilient Real-time Communication & Fallback data handling	2	3	6
Built-in Get-up Function	0	2	0
Body inverse kinematics, roll, pitch, yaw	2	3	6
Leg inverse kinematics	3	2	6
Simple Obstacle Avoidance	1	3	3
Adaptive Gait and Speed	2	2	4
Incremental movements	2	2	4
Photogrammetry	1	3	3

Figure 6.1: Performance Assessment Results.

The overall grade reflects the performance of the robot compared to the design guidelines. Some guidelines were not met at all like the built-in get-up function or most of the user interface items, while others got full points. The weight for the grades is mostly based on their importance in achieving the major goals within the project. Some parts got three points, this does not mean they are absolutely perfect but it does mean that it is working at a level which can be considered sufficient for a prototype.

6.3 Comparison with Stakeholder Input

In this section we shortly discuss the relevant input given by our stakeholders.

1. **Jitte Waagen:** Suggested the need for a remote control, he mentioned the lack of GPS signal.
2. **Alicia Walsh:** Mentioned ethical technology, proposing that robot shouldn't necessarily replace humans, but rather assist them. Suggested the need for a LED ring light, suggested water and dust resistance.
3. **Tysha Buekers:** The robot should not be larger than the tunnels of the flint mines and there should be a shell to protect against the high humidity in the air.
4. **Arnoud Visser:** We might use two batteries to supply the servo motors and the Raspberry Pi. A lightweight replacement for the current shell on the robot will also be considered.
5. **Mels van Eck:** Mels he pointed out that the servo HAT could probably take a lot more current than it is receiving right now. This could solve the issue with the servo's not having enough torque.

6.4 Evaluation of Biomimicry

Two key implementations of biomimicry in our robot design are the leg and paw pad structures. In the following section, we discuss the advantages and disadvantages of these biomimetic designs.

6.4.1 Leg design

Biomimicry has played a significant role in shaping both the leg and paw pad designs of our robot. By drawing inspiration from natural systems, we aimed to enhance the robot's performance in terms of efficiency and mobility. The robot's leg design draws inspiration from the musculoskeletal system, specifically the action of biceps and triceps in human limbs. Similarly, our robot's leg uses this linkage system. In biological systems, muscles work in pairs to create controlled movement, where one muscle contracts while the other relaxes. We used servo motors to replicate this action, with one set of motors controlling the movement of the leg in one direction and another set controlling the opposite motion. By mimicking the bicep and tricep functions, the robot's legs make a more natural leg articulation, allowing a wide range of motion. Therefore our robot is more mobile and capable of navigating complex terrains. While the design and movement replicates the natural movement of these muscles, the joints deviate from biomimicry. The joints consist of metal bearings, which introduce rigidity instead of flexibility that comes from natural tissue. Further research could focus on creating flexible joints that replicate natural tissue.

6.4.2 Paw structure

The design of the paw pads draws inspiration from the dermis layer of dog paws, particularly the natural honeycomb structure found in paw pads on dogs. In nature, these structures provide a high strength-to-weight ratio, which is important for efficient energy absorption and load distribution. The robot's paw features a hexagonal pattern made from latex, which helps to distribute forces more evenly when the robot makes contact with the ground. Additionally, the anti-slip feature that the latex material introduced was very effective when testing the robot. A downside of the design can be the following: there is a scale difference between our paw structure and the natural honeycomb structure. In dog paws the structure is on micrometer scale, while in our design the structure is made on millimeter scale. This discrepancy could affect the robot's performance. Further research is needed to understand the impact on performance.

6.5 Ethical Considerations

The design of a robotic system requires careful consideration of the ethical values it embodies. Value sensitive design (VSD) is an established method for integrating societal values into a design process (Umbrello and Poel 2021). It has been applied to many different technologies, and provides a more robust design. The use of a value hierarchy chart can schematically depict how ethical values are incorporated into design guidelines (see Figure 6.2). For this project, four main core values have been identified. **Preservation** considers the idea that the robot may have disruptive effects on the environment it explores, especially in caves with delicate ecosystems or ones that hold particular cultural heritage.

Ancient artifacts and microorganisms are at risk of damage if a robot is present. This includes unforeseen mechanical errors such as falling, incorrect commands or environmental disturbances like sound and light flashes. To address this, non-invasive data collection methods such as avoiding material sampling, excessive light, or noise, were considered for the project. Moreover, shock absorbing feet were included to reduce impact on the ground, and prevent slipping, which minimizes dust particles from being scattered. **Accessibility** considers the importance of making the robot affordable and easy to use, and ensures that it is accessible to amateur archaeologists. High costs often limit access to advanced technologies, as mentioned by Jitte Wagen regarding drones (personal communication, 2025). This forces archaeologists to rely on traditional methods like photogrammetry. Moreover, the robot should require no special technical knowledge, to allow for more usability among non-experts. To achieve this, the robot implements low-cost components such as distance sensors and materials such as PLA. It also has intuitive user interface by including a PS4 controller to navigate the robot. **Ethical technology** considers collaboration between humans and robots instead of complete replacement. It specifically considers archaeologists who may have a passion in caving, and wouldn't want to be replaced by a robot (this was considered during a meeting with Alicia Walsh). It also questions whether robots can be fully trusted to perform autonomous tasks. Robots can enhance efficiency in data collection while humans maintain control over navigation. To address this, the robot incorporates human-robot interaction features, such as an LCD screen and real-time sensory data transmission to the operator. **Sustainability** considers the effect of the robot on the environment, both in and out of the cave. It emphasizes the need for careful consideration of sustainable materials, and the environmental cost of the robot's life cycle. To address this a rechargeable battery was chosen as the energy source, while the main material for production is PLA, a biodegradable and recyclable plastic.

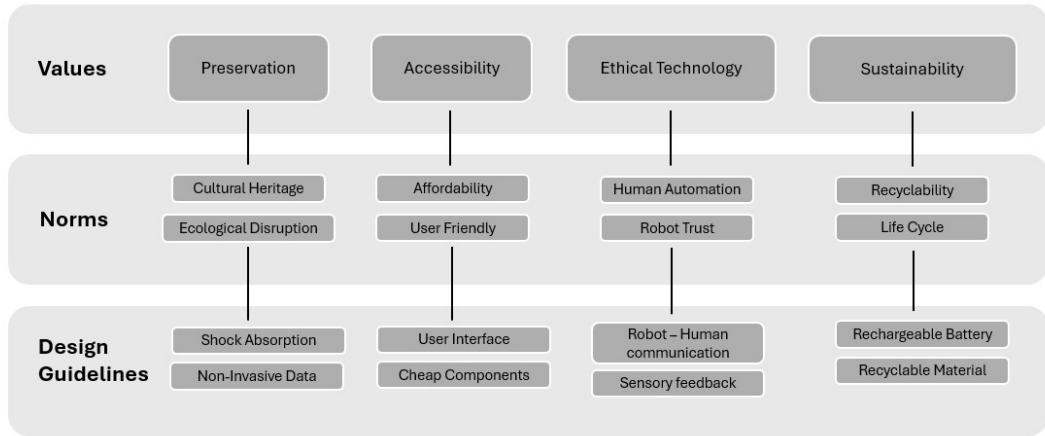


Figure 6.2: Value Hierarchy Schematic.

6.6 Future Plans

6.6.1 Short Term Improvements

- Improving the power management. The motor hat is currently a major movement bottleneck since it is not able to provide the servo's with enough voltage and ampere. This leads to the motors being too weak to properly move the body of the robot. This could be improved by directly feeding power to the servo's from the battery (with a buck converter), while still sending the PWM signal through the current servo hat or through an Arduino.
- Making a user friendly and consistent way to do servo calibration.
- Making the robot more user friendly. Adding quality of life upgrades like an indication for battery level or improved cable management.
- Find the optimal Bezier curve for the walking engine. This will significantly improve movement.
- Adding force sensitive resistors to the feet so that the robot can utilize feedback data in the walking engine.
- Add extra ultrasonic distance sensors or a LiDAR for improved obstacle avoidance.

- Change the leg motor providing the 3rd DOF, to one with more torque. The leg structure, being a parallel linkage, functions like a differential mechanism, essentially allowing the torque to be almost always shared between both motors. This isn't the case for the 3rd DOF motor.
- Changing the currently selected ultrasonic distance sensor to the version JSN-SR04T. This sensor is waterproof and dust-resistant, which protects its components from damaging in the environment.

6.6.2 Long Term Improvements

- Improving navigation. SLAM with LiDAR is unlikely to work in caves but something like visual SLAM could be a good option for navigation.
- Improving the stability of the walking engine through reinforcement learning as described by (Rahme et al. 2020).
- Redesign the body of the robot and add software so that the robot can recover gracefully from falling. In addition to this damage prevention code should be written to make sure the robot falls in a way that minimizes damage, perhaps by retracting it's leg when falling.
- Create compliant robotic legs, for better adaptability to uneven terrain and more shock absorption
- Change motors to BLDC motors, for extra torque, quicker accelerations and a smoother operation.

6.6.3 Improvements Possible with Unlimited Resources

- Add complex, robust autonomous behavior where the robot is able to traverse any terrain.
- Redesign the body to utilize brushless motors or custom actuators and improve the overall build quality by incorporating more expensive material like carbon fibre.
- Add advanced sensors to create extremely high quality 3D data from scans.

6.7 Reflective Insights

6.7.1 Points of Improvement

One key area for improvement is gathering feedback from stakeholders earlier in the project. Engaging with stakeholders earlier would have helped to better align our goals and expectations, potentially identifying issues and refining the design sooner. Another challenge we faced was spending too much time on the SLAM algorithm and getting ROS to work, which ultimately did not succeed. In hindsight, dedicating excessive time to SLAM diverted attention from other critical areas of the project, such as refining the robot's core functionality and exploring alternative navigation methods. Additionally, we spent too little time on the report, as much of the group's focus was directed toward developing the robot. While the robot was a priority for most of the team, the report should have been given more attention earlier. We should have spread the writing of the report better and over a longer time period. A stronger focus on documentation would also have helped us better track progress and refine our approach as the project advanced. Lastly, we should also have devoted more attention to testing our robot and conducting thorough analyses based on these tests. A structured approach to testing would have helped us to select specific variables as a point for optimization. Unfortunately, since our robot was not able to walk very well until the last few days, it was hard to conduct meaningful tests. It hindered our ability to evaluate the design's effectiveness under different conditions. As a result, we were not able to identify many trends and patterns from the test data. In the future we should develop a structured testing plan with clear methods and measurable variables to optimize.

6.7.2 Positives

Despite these challenges, several aspects of the project went well. The leg prototype was designed and built relatively early, which allowed for early testing and iterative improvements. This early prototype proved crucial in refining the robot's mobility and gave us valuable insights into how the components interacted. Another positive aspect was that the walking engine and visualization were ready for testing as soon as the robot's body was assembled. This

seamless integration allowed us to assess the robot's movement quickly and make necessary adjustments. We also made significant progress through iterations. At least three iterations of both the body and the legs were completed, leading to a highly optimized prototype. This iterative process enabled us to address issues as they arose, ultimately improving the robot's design and functionality.

Finally, the teamwork throughout the project was exceptional. We communicated effectively and ensured that everyone was involved, often meeting together nearly every day of the minor. The task division was efficient, and everyone was able to contribute to areas they were most passionate about, which helped foster a sense of ownership and motivation.

CHAPTER 7

Documentation

7.1 GitHub

The entire codebase is available here: Quad-EX github. The contribution guidelines, documentation and build instructions can be found in the README.

7.2 Quad-EX contact information

Joost Weerheim: weerheimjoost2000@gmail.com

Noe Nicolas Medina: noe.nicolas097@gmail.com

Eva Bredeweg: eva.bredeweg@student.uva.nl

Sep Claessen: sep.claessen@student.uva.nl

Margriet Frericks: margrietfrericks@gmail.com

Bibliography

- 3DSourced (Nov. 2023). *Understanding Infill Density Settings in 3D Printing*. Accessed: 2025-01-18. URL: <https://www.3dsourced.com/rigid-ink/3d-print-optimal-infill-settings/>.
- Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi - Mini Kit (2025). URL: <https://www.adafruit.com/product/2327> (visited on 01/24/2025).
- Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 (2025). URL: <https://www.adafruit.com/product/2472> (visited on 01/22/2025).
- Al Tahtawi, Adnan Rafi (2018). "Kalman filter algorithm design for hc-sr04 ultrasonic sensor data acquisition system". In: *IJITEE (International Journal of Information Technology and Electrical Engineering)* 2.1, pp. 15–19.
- Arduino® Mega 2560 Rev3 (2025). URL: <https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf> (visited on 01/24/2025).
- Awange, Joseph L et al. (2013). *Fundamentals of photogrammetry*. Springer, pp. 157–174.
- Al-Batati, Abdulrahman S, Anis Koubaa, and Mohamed Abdelkader (2024). "ROS 2 in a Nutshell: A Survey". In: *preprints*.
- Bisson-Larrivée, Alexandre and Jean-Baptiste LeMoine (2022). "Photogrammetry and the impact of camera placement and angular intervals between images on model reconstruction". In: *Digital Applications in Archaeology and Cultural Heritage* 26, e00224. ISSN: 2212-0548. DOI: <https://doi.org/10.1016/j.daach.2022.e00224>. URL: <https://www.sciencedirect.com/science/article/pii/S2212054822000133>.
- Bloom, Sandra (Oct. 2024). *What Is Biomimicry, and How Does It Work?* Accessed: 2025-01-18. URL: <https://scientificorigin.com/what-is-biomimicry-and-how-does-it-work>.
- Bonci, Andrea et al. (2023). "Robot Operating System 2 (ROS2)-based frameworks for increasing robot autonomy: A survey". In: *applied sciences* 13.23, p. 12796.
- British Cave Rescue Council (Apr. 2024). *Incident Report for 2023*. Tech. rep. Accessed: 2025-01-18. British Cave Rescue Council. URL: https://www.caverescue.org.uk/wp-content/uploads/2024/04/2023_BCRC-Incident-Report.pdf.
- Cole, David M and Paul M Newman (2006). "Using laser range data for 3D SLAM in outdoor environments". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE, pp. 1556–1563.
- Conati Barbaro, Cecilia et al. (2024). "Safeguarding archaeological excavations and preserving cultural heritage in cave environments through engineering geological and geophysical approaches". In: *Journal of Archaeological Science: Reports* 60, p. 104868. ISSN: 2352-409X. DOI: <https://doi.org/10.1016/j.jasrep.2024.104868>. URL: <https://www.sciencedirect.com/science/article/pii/S2352409X24004966>.
- Docker Desktop (2024). URL: <https://docs.docker.com/desktop/> (visited on 12/17/2024).
- Ebbinghaus, Thomas, Gregor Lang, and Thomas Scheibel (June 2023). "Biomimetic polymer fibers—function by design". In: *Bioinspiration Biomimetics* 18.4, p. 041003. DOI: [10.1088/1748-3190/acddc1](https://doi.org/10.1088/1748-3190/acddc1). URL: <https://doi.org/10.1088/1748-3190/acddc1>.
- evdev 1.7.1 (2024). URL: <https://pypi.org/project/evdev/> (visited on 12/20/2024).
- Fan, Yanan et al. (2024). "A Review of Quadruped Robots: Structure, Control, and Autonomous Motion". In: *Advanced Intelligent Systems* 6.6, p. 2300783. DOI: <https://doi.org/10.1002/aisy.202300783>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202300783>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202300783>.
- Gabriel, Mutinda M and Kamweru P Kuria (2020). "Arduino uno, ultrasonic sensor HC-SR04 motion detector with display of distance in the LCD". In:

- Jasiak, Caroline C, John J Schultz, and Morgan J Ferrell (2023). "Documenting Outdoor Simulated Scenes with Photogrammetry Methods for Improving Harsh Natural Lighting Conditions." In: *Forensic Anthropology (University of Florida)* 6.3.
- Karami, Ali, Fabio Menna, and Fabio Remondino (2022). "Combining photogrammetry and photometric stereo to achieve precise and complete 3D reconstruction". In: *Sensors* 22.21, p. 8172.
- Kee Paik, Jeom, Anil K Thayamballi, and Gyu Sung Kim (1999). "The strength characteristics of aluminum honeycomb sandwich panels". In: *Thin-Walled Structures* 35.3, pp. 205–231. ISSN: 0263-8231. DOI: [https://doi.org/10.1016/S0263-8231\(99\)00026-9](https://doi.org/10.1016/S0263-8231(99)00026-9). URL: <https://www.sciencedirect.com/science/article/pii/S0263823199000269>.
- Kočí, Jakub (Jan. 2021). *Everything you need to know about infills*. Accessed: 2025-01-18. URL: https://blog.prusa3d.com/everything-you-need-to-know-about-infills_43579/.
- Lu, Ming et al. (2022). "Design of a Small Quadruped Robot with Parallel Legs". In: *Complexity* 2022.1, p. 9663746. DOI: <https://doi.org/10.1155/2022/9663746>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/9663746>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/9663746>.
- Macenski, Steven et al. (2022). "Robot operating system 2: Design, architecture, and uses in the wild". In: *Science robotics* 7.66, eabm6074.
- Marin-Garcia, G et al. (2020). "Battery types and electrical models: A review". In: *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Vol. 4. IEEE, pp. 1–6.
- Mechanism Digital Camera (n.d.). URL: https://www.ricoh-imaging.co.jp/english/r_dc/photostyle/knowledge/preparation/digitalcamera.html#:~:text=With%20a%20digital%20camera%2C%20the%20on%20the%20picture%20display..
- Miao, Huaibin et al. (Nov. 2017). "How does the canine paw pad attenuate ground impacts? A multi-layer cushion system". In: *Biology Open* 6.12, pp. 1889–1896. ISSN: 2046-6390. DOI: 10.1242/bio.024828. eprint: <https://journals.biologists.com/bio/article-pdf/6/12/1889/1832158/bio024828.pdf>. URL: <https://doi.org/10.1242/bio.024828>.
- Minetti, AE (2000). "The three modes of terrestrial locomotion". In: *Biomechanics and biology of movement*, pp. 67–78.
- Moomaw, Butch (2007). "Camera technologies for low light imaging: overview and relative advantages". In: *Methods in cell biology* 81, pp. 251–283.
- Mueggler, Elias et al. (2014). "Aerial-guided navigation of a ground robot among movable obstacles". In: *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, pp. 1–8. DOI: 10.1109/SSRR.2014.7017662.
- Panchal, Priyam (2023). *Quadruped Leg*. Accessed: 2025-01-20. URL: https://www.youtube.com/watch?v=sP7Wvgz0Dfw&list=LL&index=40&ab_channel=PriyamPanchal.
- Paxman, Georgina (Aug. 2020). *50% of Earth's Land Surface Remains Relatively Untouched By Humans*. Accessed: 2025-01-18. URL: <https://earth.org/half-of-earths-land-surface-remains-relatively-untouched-by-humans/>.
- Petráček, Pavel et al. (2021). "Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles". In: *IEEE Robotics and Automation Letters* 6.4, pp. 7596–7603. DOI: 10.1109/LRA.2021.3098304.
- Petrlík, Matěj, Tomáš Krajník, and Martin Saska (2021). "LIDAR-based Stabilization, Navigation and Localization for UAVs Operating in Dark Indoor Environments". In: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 243–251. DOI: 10.1109/ICUAS51884.2021.9476837.
- Prusa Research (2024). *Infill*. Accessed: 2025-01-18. URL: https://help.prusa3d.com/article/infill_42.
- Rahme, Maurice et al. (2020). "Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion". In: *CoRR* abs/2010.12070. arXiv: 2010.12070. URL: <https://arxiv.org/abs/2010.12070>.
- Raspberry Pi* (n.d.). URL: <https://www.raspberrypi.com/documentation/accessories/camera.html>.
- Raspberry Pi 5* (Aug. 2024).
- Sen, Muhammed, Veli Bakircioglu, and Mete Kalyoncu (Jan. 2017). "Inverse Kinematic Analysis of a Quadruped Robot". In: *International Journal of Scientific Technology Research* 6, pp. 285–289.
- Shenoy, Prajwal, Anurag Gupta, and SKM Varadhan (2022). "Design and validation of an IMU based full hand kinematic measurement system". In: *IEEE Access* 10, pp. 93812–93830.
- Singh, J (2021). "The Future of Autonomous Driving: Vision-Based Systems vs. LiDAR and the Benefits of Combining Both for Fully Autonomous Vehicles". In: *Journal of Artificial Intelligence Research and Applications* 1.2, pp. 333–376.
- SLAM* (n.d.). URL: <https://nl.mathworks.com/help/nav/slam.html>.

- Taheri, Hamid and Zhao Chun Xia (2021). "SLAM; definition and evolution". In: *Engineering Applications of Artificial Intelligence* 97, p. 104032.
- Tiny Tronics* (n.d.). URL: https://www.tinytronics.nl/product_files/002405_Data%20Sheet%20of%20TD-8130MG%20Digital%20Servo%20Motor.pdf.
- Toy, Ibrahim, Akif Durdu, and Abdullah Yusefi (2022). "Improved dead reckoning localization using IMU sensor". In: *2022 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, pp. 1–5.
- Ultrasonic Ranging Module HC - SR04* (2025). URL: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> (visited on 01/22/2025).
- Umbrello, Steven and Ibo Poel (Aug. 2021). "Mapping value sensitive design onto AI for social good principles". In: *AI and Ethics* 1, p. 3. DOI: 10.1007/s43681-021-00038-3.
- Vuursteenmijnen.nl (2025). *Prehistorical Flint-Mining in the Netherlands: Rijckholt (Ryckholt) - St. Geertruid*. Accessed: 2025-01-20. URL: <https://www.vuursteenmijnen.nl/rijckholt/english-summary.html>.
- Zeng, Xuanqi et al. (Apr. 2019). "Leg Trajectory Planning for Quadruped Robots with High-Speed Trot Gait". In: *Applied Sciences* 9, p. 1508. DOI: 10.3390/app9071508.
- Zhang, Shengkai (2019). "Exploring imu attitude and position estimation for improved location in indoor environments". In.
- Zhmud, VA et al. (2018). "Application of ultrasonic sensor for measuring distances in robotics". In: *Journal of Physics: Conference Series*. Vol. 1015. 3. IOP Publishing, p. 032189.

APPENDIX A

Appendix

A.1 Software

For the Software, the initial attempt of programming the robot has been using Robot Operating System (ROS). Given its widespread use in the robotics industry and its good support for sensors and other hardware implemented in robots, it's a logical choice to turn to ROS for a project like this. However, with no experience with ROS, it proved to be an unrealistic choice. An explanation of ROS and why the switch to Python has been made has been included here. In case the choice to use ROS would be re-considered in the future, this information below can serve as introduction to ROS.

A.1.1 ROS 2

Robot Operating System (ROS) is a framework that has found widespread use in the robotics industry and has played a pivotal role in the development of a wide range of robotic applications (Macenski et al. 2022). Despite the name, it is not an operating system, but a framework that functions as so-called “middleware” on top of another operating system, Linux Ubuntu, bridging the gap in communication between different components and thus enabling different systems to work together and achieve complex tasks (Bonci et al. 2023). To enable the ROS framework to fulfill this purpose, it offers a wide toolbox of packages and libraries that are fundamental in many diverse robotic systems. However, as it gained a more prominent role in the field of robotics, the ROS framework’s weak spots needed to be addressed in order to secure its continuous use and widespread adoption. Addressing these limitations led to the new and improved version of the Robot Operating System, ROS 2 (Al-Batati, Koubaa, and Abdelkader 2024). This new version improved the autonomy and flexibility that can be achieved by applications built with ROS 2, and improved real-time performance (Al-Batati, Koubaa, and Abdelkader 2024).

In this project, the robot and its sensors need to be communicating accordingly, and proper support for the hardware (and sensors) is essential. Given that ROS 2 is developed for the purpose of facilitating the development of robotic applications and offers unmatched support for various hardware, we initially selected ROS 2 as the software framework. Within ROS 2, the Humble version was selected, since this version is stable and is well-documented.

A.1.2 Docker

However, ROS 2 operating on Linux Ubuntu raised an issue: the selected ROS 2 Humble version is specific to Ubuntu 22, which is incompatible with the selected raspberry pi 5, since Ubuntu 22 does not support the Raspberry Pi 5. This issue could be resolved in two ways: either replacing the raspberry pi 5 for a raspberry pi 4 – a version that is supported by Ubuntu 22- or using a Docker container.

Given that the raspberry pi 5 has CPU performance that is 2 to 3 times better than that of the raspberry pi 4 (*Raspberry Pi 5 2024*), switching to a raspberry pi 4 would be a significant downgrade and would not be the preferred solution. A Docker container, on the contrary, would provide a satisfactory solution. A Docker container is an isolated environment that shares the host operating system (OS) kernel, thus without requiring a separate guest OS, while keeping the environment isolated from the host kernel during processing (*Docker Desktop 2024*). Sharing the kernel means that Docker containers do not need additional memory or CPU even though a separate OS is being used. Therefore, Docker containers are a good solution to the problem in terms of their overhead, while also being

lightweight. It allows a Linux environment to be run on another host OS, thus also allowing Ubuntu 22 to be run on a raspberry pi 5.

A.1.3 Switching to Python

However, having no experience with ROS, gaining an understanding of the basic building blocks of any program that can be created with ROS, took considerable time and effort. Continuing this path to ultimately create a complex software package for the robot would be unrealistic. Therefore, we stepped away from using ROS, and focused on finding python-supported versions of the same implementation. This also eliminated the need to use Docker containers.

A.2 Author contributions

Eva: Literature Review (locomotion mechanisms), Literature Review - Biomimicry (leg geometry and structure, paw and shock absorption), Design phase - Mechanical (paw structure implementation, body structure implementation), Stakeholders (Tycha Buekers, Arnoud Visser), Conclusions and discussion (Comparison with Stakeholder Input, Evaluation of Biomimicry, Reflective Insights).

Margriet: Abstract (partly), Comparison of Sensor Technology (LiDAR, ultrasonic distance sensor, IMU), Navigation (Theoretical Framework), Stakeholder Analysis of Jitte Waagen, Electronics (Overview, Raspberry Pi, Motor Control Board, Arduino Mega 2560, Sensor selection (LiDAR, ultrasonic distance sensor, IMU)), Software Design Guidelines (except photogrammetry-related), Software Performance Specifications (except photogrammetry-related), Sensor Selection (LiDAR, ultrasonic distance sensor, IMU), Navigation (Design Phase - Software), Remote Control For Navigation, Performance Assessment (only partly for grading of Software).

Sep: Photogrammetry (Literature review, implementation in the robot, camera comparison, photogrammetry guidelines, photogrammetry), Electronics (Battery setup, camera configuration, sensor selection, servo motors), Software (Performing photogrammetry with camera, data transfer), Testing Phase (Photogrammetry), Stakeholder interview (Mels van Eck), Mechanical Design Phase (Photogrammetry implementation).

Noe: Introduction, Literature Review (3D Printing: Infill Density and Pattern, Quadrupeds: Leg Geometry and Structure), Design Phase Mechanical - (Design Guidelines, Performance Specifications, Leg Design Stages), Test Methods - (Test 1,2,4), Stakeholders - (Stakeholder Identification, Alicia Walsh), Conclusions and Discussions (Design Evaluation, Performance Assessment, Ethical Considerations).

Joost: Abstract, Introduction, Literature Review - Technical (Walking Engine), Design Phase - Mechanical (Electronics), Design Phase - Software (Design Guidelines, Performance Specifications, Walking Engine Implementation), Stakeholders (Tycha Buekers, Arnoud Visser), Testing Phase (Basic Movement, Walking Engine), Conclusion and Discussion (Performance Assessment, Comparison with Stakeholder Input, Future Plans), Documentation (All).

A.3 Schematic overview of electronic components

A.4 Mechanical Design Guidelines

A.5 Mechanical Design Specifications

A.6 Software Design Guidelines

A.7 Software Design Specifications

A.8 Sensor Criteria

A.9 3D Printing Infill Patterns

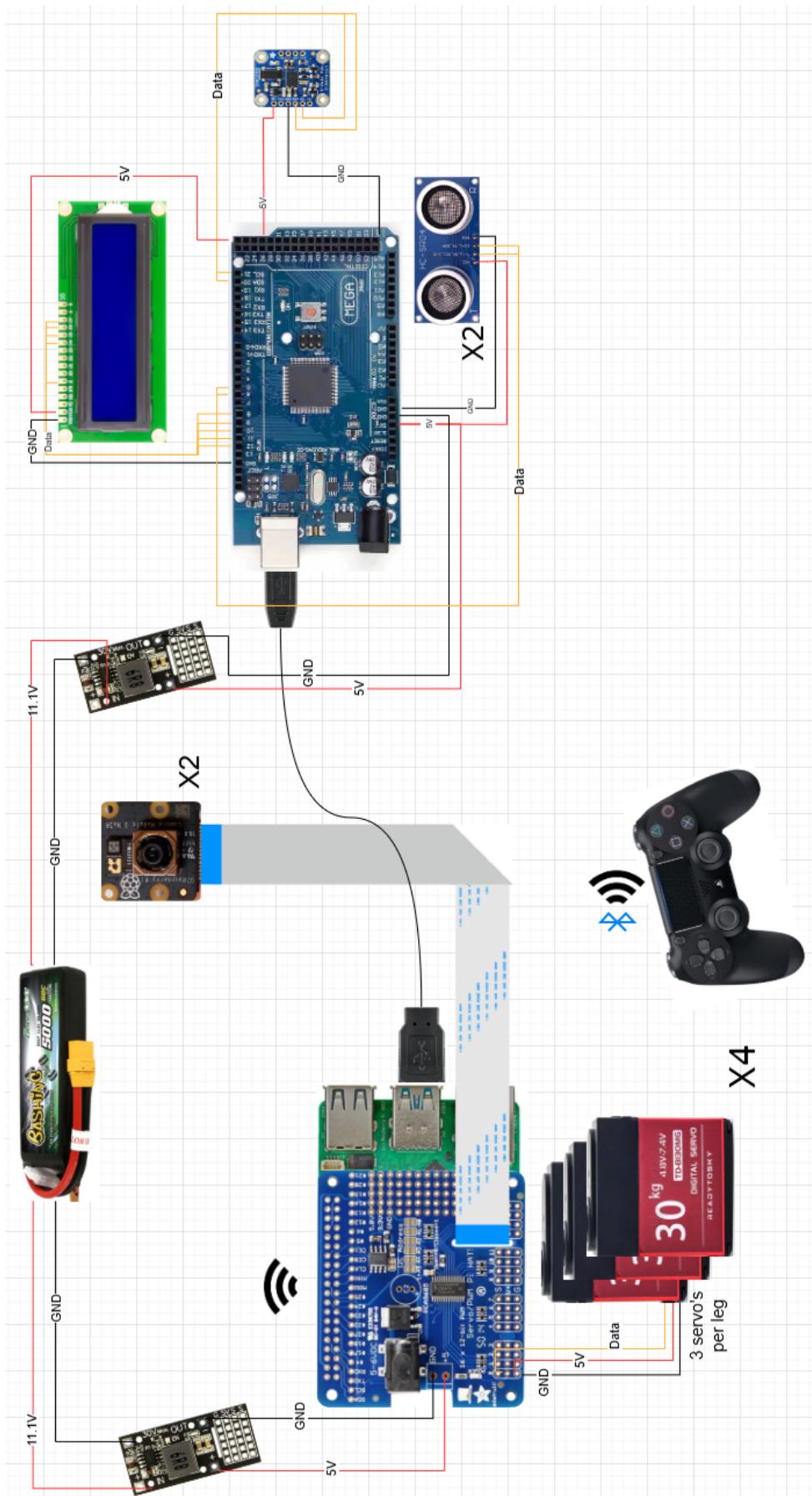


Figure A.1: Schematic overview of electronic components.

Design Guideline	What	Why	How
Size Constraints	Robot can fit and move inside a cave	Size impacts obstacle avoidance, and limits the range of exploration in tighter spaces.	Overall size of the robot must not exceed a certain dimension, based on narrowest passages.
Internal Component Storage	Electronic components must be stored inside the robot	To protect the electronics and improve performance. Too little space for internal components affects data quality by limiting upgrades, like LiDAR, and reduces exploration range due to constrained battery capacity	Measure dimensions of all electrical components, create an electrical circuit plan, arrange components as compact as possible, obtain minimum body dimensions
Overall Weight	Weight of robot structure must be minimized, while maintaining integrity	Reduces strain on motors and enables better manoeuvrability in caves. Puts less pressure on batteries for a longer expedition	Use lightweight materials such as PLA and include weight reduction techniques, including cutouts and optimized structural shapes like triangles for high strength to weight ratios.
Weight Distribution	Weight of overall robot must be evenly distributed	Good weight distribution provides balance and stability when navigating rough terrain	Position batteries and heavier components closer to the centre of mass.
Resistance to Environment	Robot must have resistance to a wet, muddy and dusty environment	Improper resistance will damage electronics, limit the lifespan of the robot and potentially stop the robot in wet caves	Use materials that are IP65 water resistant, design casings and covers for electronics, potentially use water resistant motors, use sealings like silicone.
Low – Light Functionality	Robot must function in complete darkness	Most caves are not equipped with lighting, this limits the range of sensory data that can be collected	Include an external light source mounted on the robot, or use sensors that require no light input (i.e. LiDAR)
Temperature Resistance	Robot must operate reliably in low temperatures	Caves have much colder temperatures than the outside environment. If cold enough, this can impact the responsiveness of electronics	Check operating temperature for each electric component, and set a limiting temperature.
Battery Power	Robot must have enough battery power, and be rechargeable	Insufficient battery power may leave the robot stuck mid expedition, with no return. It may also lead to failure in data collection.	Run all motors at full power to measure peak amperage and voltage, estimate expedition duration, select a suitable battery, and validate by testing in a cave to account for environmental impacts.
Shock Absorption	Shock absorption and anti-slip	Caves have uneven surfaces. High load shocks may directly damage the motors and components	Incorporate foot padding, use soft TPU or rubber materials for stability.
Manoeuvrability	Flexible robot with large range of motion	To navigate uneven terrain and obstacles. Flexibility impacts the range of exploration.	Design a 3DOF leg to allow for x-y-z, pitch, yaw and roll motion.

Figure A.2: Mechanical Design Guidelines.

Design Guideline	Specification	Reasoning	Test Method
Size Constraints	The prototype must fit the following size dimensions: 0.5 x 0.5 x 0.5 m	Based on the average size of caves in Limburg (Buekers, Tycha)	Measure dimensions of robot with measuring tape
Internal Component Storage	The prototype must have at least 1L of internal storage for electronic components around 0.3 L to spare	Measured the volume of each component, added 20% for cabling and empty space. Then estimated another 30-40% needed for extra additions (around the size of one extra battery)	Ensure at least one more battery can fit in the robot
Overall Weight	The prototype must weigh no more than 3.3 kg	Based on the length of the legs (18cm), and the torque provided by the servo motors (30kg-cm) _____	Weigh the robot on a scale
Weight Distribution	The full prototype must be modelled on SolidWorks, and its centre of gravity must be within 10cm of its centre of symmetry .	This particular rating holds for resistance to dust and occasional splashes of water, which is similar to what can be found in caves.	Hang the body of the robot at the symmetrical centre with a string, and record any tipping of the body.
Resistance to Environment	The prototype must be built with waterproof electrical components rated at IP65 . Otherwise, these should be properly sealed.	This solution is lightweight, and consumes very little power. It opens the gate to all sensors	Let the robot walk outdoors for 20 min and observe conditions of interior
Low – Light Functionality	The prototype must have a LED Ring Light attached to it	Based on the temperatures recorded in Limburg caves in the winter period, during test phase C.	The robot has a light source, and can perform photogrammetry with clear pictures
Temperature Resistance	The prototype must be able to navigate and collect data for the duration of one full expedition at a temperature of at least 2 degrees C.	Test outdoors in cold temperatures for 20 min	Test outdoors in cold temperatures for 20 min
Battery Power	The selected battery must have allow the robot to run 15min, have a charge of 80% left and must be rechargeable	15 min time limit makes the testing repeatable. Checking the battery level can give estimate on maximum battery life range.	Let the robot walk for 15 min, check battery level and recharge to full battery.
Schock Absorption	The prototype must have padding on the feet . It must prevent slipping, and absorb motion.	_____	Place robot on incline and measure at which angle it slips (with and without foot padding)
Manoeuvrability	The prototype must have a total of 18 DOFs including the body, or 12 servo motors.	This is the highest number of DOFs possible for a quadruped.	Robot is tested for all 18 DOFs. These should be smooth

Figure A.3: Mechanical Performance Specifications.

Design Guideline	What	Why	How
LCD screen for Feedback	Robot must have an LCD screen	Ensures clear communication with the user before expedition	Integrate an LCD to display feedback, status and warnings
Controller for Control	Robot must have a wireless controller	Enables manual control and override of the robot and its movements	Provide a handheld wirelessly connected controller
Battery Warning	Robot must give a warning message to the user in the case of low battery and safely shut down	Prevents corruption of the robot's main computer as a result of power loss	Implement a program that monitors the battery percentage, warns the user when it's low, and shuts down when it's too low
Bezier foot trajectory	The foot of the robot should follow a bezier curve.	It is an optimal trajectory and it's easy to add a stance and swing phase.	Create a bezier curve and sample 10 points from it continually.
IMU Updates	The robot should have access to the data from the IMU that is attached to it	IMU data provides important insights into the position of the robot body. It will allow the robot to, for example, detect when it's falling.	Add the IMU to the Arduino and send its data over serial to the Raspberry Pi
Trot gait	Walking with two feet in the air and two on the ground at all times.	It is a fast and stable walk for quadrupeds.	Use a certain offset in the walking engine, this indicates to the legs from which points they should sample at which time on the Bezier curve.
Creep gait	Walking with three feet on the ground and one in the air at all times.	It is a slow but very stable walk.	Use a certain offset in the walking engine, this indicates to the legs from which points they should sample at which time on the Bezier curve.
Resilient Real-time Communication & Fallback data handling	Communication must be real-time with minimal latency, maintaining $\geq 50\%$ connection /min. If unavailable, data collection should continue, with the fall-back option.local data storage or cable-supported communication as alternative to wireless transfer.	This design ensures precise control in tight spaces, enables immediate intervention during emergencies (malfunctions, safety risks, etc.), guarantees reliable communication, and preserves data with a redundant backup system.	Based on testing, implement wireless communication with local storage as fall-back option, or wired communication that doesn't require a fall-back option.
Built-in Get-up Function	Robot must be able to get up after falls	Improves and enables adaptability on uneven terrain	Equip the robot with a way to recover from falls.
Leg inverse kinematics	Moving the leg based on a desired position.	The leg should be able to move to a position, provided the coordinates for this position.	Finding the analytical solution to the inverse kinematics problem for a 3DoF leg

Figure A.4: Software Design Guidelines 1/2.

Body inverse kinematics, roll, pitch, yaw	Robot must have working body inverse kinematics allowing it to move over the axis of roll, pitch, and yaw.	The robot should be able to move over the axis of roll, pitch, and yaw.	Write the body inverse kinematics.
Simple Obstacle Avoidance	Robot should exhibit an autonomous obstacle avoidance behaviour	Prevents robot from sustaining damage without the need for human interference	Program simple obstacle avoidance behaviour based on proximity of objects
Adaptive Gait and Speed	Robot could optimise gait and speed for different terrains	Increases efficiency and stability on challenging terrain	Implement reactive algorithms to adapt movement to terrain
Incremental Movements	Robot should be able to move and turn in small steps	Ensures appropriate overlap between images for optimal photogrammetry can be established	Implement fine motion control for small, precise movements (with and without controller)

Figure A.5: Software Design Guidelines 2/2.

Design Guideline	Specification	Reasoning
LCD screen for Feedback	The robot has an LCD screen	
Controller for Control	Robot has a controller. The controller must have directional control, an emergency stop button, and at least 5m range wireless communication	A 5m range ensures basic operational distance
Battery Warning	The battery warning should appear at 20%, system should initiate shutdown sequence with a 15-second delay at 10%	20% threshold provides a buffer for the user to act. 10% cutoff ensures that the robot's main computer shuts down safely before risking corruption
Controller Range Alert	If the controller signal is lost for over 5 seconds, the message "Controller out of range. Please reconnect." appears	A 5-second delay prevents false alerts caused by temporary signal drops. Warning ensures the user is aware of the lost connection
Internal State Monitoring	A small window should be drawn on the connected device (laptop) and updated every 2 seconds with internal states like <ul style="list-style-type: none"> • CPU temperature (<80°C) • Battery level (robot and controller) • Motor and battery health (normal or overheating) 	Update interval allow the user to monitor performance in real time. CPU threshold selected based on typical hardware tolerances
Inaccessible Area Alert	The robot must pause and display "Going beyond retrieval point. Confirm to proceed." on the connected device when it accesses an area with a height of <0.75m or out of wireless range	Height limit based on comfortable crawl level for adults
Direct Data Transfer	Photos should be passed on to connected device upon capture, with a retry function if connectivity fails. If transfer was successful, photos should be deleted locally	Real-time transfer minimises local storage usage and ensures immediate backup of the data.
Resilient Real-time Communication & Fallback Data Handling	Communication should be real-time (max latency 300ms). If unavailable, collected data must be stored on local storage that can accommodate 8GB of data. Attempts to retry data transfer should be continued.	300ms latency ensures near real-time communication, aligns with accepted thresholds to feel instantaneous to users. 50% threshold accounts for occasional signal drops or interference while still maintaining functional operation. 8GB based on size of one image, and expected number of images.
Built-In Get-Up Function	The robot must be able to recover from falls within 30 seconds and resume operation	Quick recovery ensures minimal disruption of the exploration
Pan and Tilt Function	Pan must support a full 360° rotation, and tilt must achieve ±30° for optimal photogrammetry coverage	Based on values recommended by Tycha
Live Location Updates	Robot must transmit its live location every 1 second with an accuracy ≤0.5m to the connected device (not storing it locally). And refresh its own location locally every 10 sec	A 1-second interval ensures near-real-time updates for accurate tracking and mapping. Logging at 10-second intervals ensures up-to-date location data without overburdening storage
Simple Obstacle Avoidance	Robot must maintain obstacle detection in a 1m radius, and halt when distance is 0.6m	A detection range of 1m ensures obstacles are identified on time. The 0.6m limit ensures collisions are prevented provided that the localization accuracy is within range.

Figure A.6: Software Design Specifications 1/2.

Safe Manual Override	The robot must disable autonomous walking and return control to the user within 300ms but continue obstacle avoidance (triggered at 1m). Conflicting user input below 0.6m triggers a double-confirmation prompt: "Object dangerously close. Sure you want to continue? Press ... for Yes, ... for No."	300ms ensures near-instant override, crucial for emergencies. Aligns with industry standards. Obstacle avoidance algorithm remains active to maintain safety. Requiring user to confirm twice minimises accidental or uninformed risky actions.
Incremental Movements	Robot must support movements in increments of 5cm (forward/backward) and rotations of 15° per step	Small steps and precise rotations can ensure at least 60% overlap between images. (18-24 images necessary per rotation)

Figure A.7: Software Design Specifications 2/2.

Criteria	LIDAR	Camera	Ultrasonic distance sensor	IMU sensor
Mapping	"Eyes" enable to determine where it is	Only short distance eyes	Only relative movement	
Localization	"Eyes" see where to go and where not	Only short distance eyes	Wrt reference position	
Navigation	Scene shows orientation wrt environment	Cannot see angles	only inertial nav	
Orientation	Scene can be used to stabilize Quad EX	Cannot see angles	Due to noisy gyroscope data	
Stabilization	Laser source light	Worse in low light	Knows yaw, pitch and roll	
Resolution	"Eyes" can detect obstacles	Down to 3 mm	- Velocity, + Bandwidth	
Object detection	Laser source light	Uses "bat" principle	Has no "eyes"	
Performance in low light	Needs light to see	Uses sonar	Does not depend on light	
Real time measuring	Fast image processing	Lower sound velocity	Relative high bandwidth	
Consistency in changing orientation	Because of Robots Gait	Only short distance	Changing orientation is measured	
Error and error propagation	Noisy sensor readings	Susceptible to noise, no propagation	Data can be noisy	
Signal to noise	Noisy sensor readings	Susceptible to noise	Data can be noisy	
Weight and volume	Lidar light 77 cm^3, 38g	6.8 cm^3, 4 g	2.2 cm^3, 3 g	
Ease of use	Complicated data association	+ interpretation, - light	Simple and straightforward + sensor fusion, - inertial sensors	
Cost	\$ 150	€ 30	\$4	\$ 35
Leadtime	1-2 weeks	1-2 days	1-2 days	1-2 days
Power consumption mW	425	200-300	75	62,5

Figure A.8: Explanation of values given in the sensor performance criteria figure.

Infill Pattern	Description	Recommend Prints
Rectangular	This standard infill pattern provides a reasonable amount of rigidity in all directions. It is also one of the easiest infill patterns to print, requiring a minimum amount of bridging on the part of your print head.	Standard prints
Triangular	Appropriate when strength is required in the direction of the shell. However, it takes longer to print.	Strong standard prints
Wave or Wiggle	As the name implies, a waveform infill pattern. Particularly useful when an object needs to be able to twist or compress. This is great to use for flexible materials.	Flexible prints
3D Honeycomb Infill	One of the more popular infill patterns. Provides greater overall strength in all directions than a rectangular pattern, with very little increase in print time. It is generally considered the most commonly used, strongest infill pattern.	Standard prints
Lines	Consists of multiple parallel lines per layer. Each layer crosses over the previous one at a 90-degree angle. This strengthens the part in two dimensions.	Decorative prints, figurines, and models
Gyroid	Creates alternating wavy lines or curves. This infill pattern takes longer to print than the others. However, the unique gyroid internal structure allows for almost isotropic mechanical properties.	Functional and mechanical prints
Octet	Creates tetrahedral (pyramid-like) volumes inside the part. It is best for parts with large horizontal surfaces.	Functional and mechanical prints
Concentric	One of the fastest infill patterns to print, and it uses the least material. However, this comes at the cost of reduced part strength.	Flexible prints
Lightning	Provides the fastest possible print time at the expense of part strength. Supports are added in a lightning bolt structure and only placed where necessary.	Decorative prints, figurines, and models
Tri-Hexagon	The strongest infill pattern. Like the grid and triangular infill types, it will cross over itself to create a hexagonal pattern interspersed with triangles.	Functional and mechanical prints
Cross	Creates multiple cross shapes as infill. This 3D printing infill pattern is ideal for flexible part shapes as it allows for the part to bend and twist.	Flexible prints
Cross 3D	Similar to the cross pattern, except that it removes the vertical rigidity for added flexibility.	Flexible prints
Cubic	Based on a series of cubes piled on top of one another at an angle for added strength.	Strong standard and functional prints
Cubic Subdivision	Similar to Cubic, except that it's optimized to use less filament by printing cubes of different sizes while still maintaining strength.	Strong standard and functional prints
Quarter Cubic	Pieces together tetrahedrons for strength suitable for load-bearing parts, especially tall and thin prints	Strong standard and functional prints

Figure A.9: 3D printing infill patterns list, 3DSourced 2023.