



THE UNIVERSITY of ADELAIDE

The University of Adelaide
Faculty of Science, Engineering and Technology
May 29, 2023

ENG 4001 Research Project Progress Report

CaveX Exploration Robot 2023

Supervisors: A Prof David Harvey, A Prof Rini Akmeliawati

By

Luka Moran: a1726133

Riley Groome: a1766563

Tyler Groome: a1766574

Main Matter Page Limit: 28
Main Matter Page Count: 28

Executive Summary

Mapping cave systems is a task characterised by nuanced difficulties but it is essential to understanding such systems. Caves serve as time capsules of Earth itself by shielding matter from the weathering effects of the atmosphere. As such, they play a critical part in deciphering the history of Earth, the solar system it resides within, the civilisations which have existed, and prehistoric creatures that previously roamed the surface. Palaeontologists, geologists, and other researchers often bring digital scanners, such as Light Detection and Ranging (LiDAR) systems, with them on a cave exploration mission to create three-dimensional (3D) maps of them. However, the size, weight, and stationary nature of these systems prevents their use in hard to reach areas and, therefore, limits the maps' completeness. To combat this, the CaveX (Cave eXploration) honours project was initiated with its first iteration in 2021. The aim of the project is to design and build a bio-inspired autonomous robot which can traverse and map cave systems including the hard to reach areas. Dr Elizabeth Reed and Craig Williams from The University of Adelaide are paleontologists who spend considerable time mapping cave systems such as the Naracoorte Caves in Naracoorte, South Australia. Their input has formed the basis for prototype development.

The CaveX project spans multiple years and is currently in its third year. The first iteration, occurring in 2021, saw a bio-inspired hexapod robot designed and built with the ability to map its environment using a 3D LiDAR scanner. Additionally, the recorded LiDAR data was used to perform post-mission Simultaneous Localisation and Mapping (SLAM) to create a map of the explored environment. Additionally, it served as a proof of concept by mapping areas which were previously inaccessible. However, its use was constrained by difficulties with traversing non-uniform terrain and gait control. Subsequently, the 2022 project team assessed the issues with the existing prototype before designing and building an improved prototype. The new prototype maintained the bio-inspired hexapod structure of the 2021 prototype but saw improved servo motors, materials, and software. These improvements resulted in a 15% weight reduction, improved robustness, and faster movement. The 2023 iteration of the project aims to introduce autonomous functionality into the 2022 prototype to progress towards accomplishing more of the system requirements. Therefore, this iteration's objectives were defined in accordance with the system requirements and thus the user needs. Specifically, they are to implement a real-time localisation and mapping capability, a pathfinding algorithm, terrain sensing and gait control system, obstacle detection and avoidance, and a real-time data monitoring system. In pursuit of such objectives, a systems engineering approach is applied. This approach provides the necessary framework to track user needs and system requirements in relation to accomplishments of the project, thus permitting a detailed justification of user needs being fulfilled. A review of literature exposed a variety of options for accomplishing such fulfilment and, with consideration of the prototype and project's limitations, the most appropriate option for each objective was selected.

Through analysis and testing of the 2022 prototype, its physical and algorithmic limitations were identified. In particular, the prototype struggles to compensate for inclined and declined surfaces. Of the three gaits, tripod was the highest performer in terms of average speed on incline and decline slopes at various angles. This observation is likely due to the additional moving legs supplying a higher total pushing and pulling force depending on whether they are subjected to an incline or decline. Additionally, the wave gait failed to traverse some of the inclines and declines. It was also noted that the robot's feet would slip on the low-friction testing surface. Further, the

EXECUTIVE SUMMARY

prototype lacks autonomous functionality and currently has no real time understanding of its environment or its location relative to its environment. The latter is of particular significance as it fundamentally prevents three of the five objectives from being achieved. Thus, while the prototype represents a framework capable of fulfilling the user needs, it currently does not fulfill a key selection of them relating to autonomy. Critically, it was identified that the pursuit of such functionality must make careful considerations to computational optimisation due to the limited processing power of the onboard computer.

The 2023 CaveX project is currently in the initial phase of the implementation stage. During this stage, the initial algorithm design and implementation is taking place. This is a critical phase in development as multiple future functionalities rely on it. Therefore, it directly impacts the achievement of multiple project objectives and thus the fulfilment of system requirements. Additionally, a major consideration of this stage is the provisioning of hardware and software systems which are a suitable foundation for future iterations of the project. The stage's initial step involves a porting the Fast LiDAR Odometry and Mapping (F-LOAM) algorithm to conduct its primary processing on the robot's graphics processing unit (GPU). While this occurs, the selected dynamic gait optimisation algorithm is integrated with the existing open-source Syropod High-level Controller (OpenSHC). Subsequent to F-LOAM implementation, obstacle detection will be implemented. Finally, since pathfinding depends on an understanding of the obstacles in the environment, it will be implemented after the system has such information. DroneDeploy, the system monitoring platform, will be implemented upon receiving a license for the software. Throughout development, iterative testing and experimentation will occur to ensure the systems are functioning correctly as this allows for rapid debugging in the event of dysfunction. Following this stage is detailed testing. These tests will verify which system requirements have been met and underpin a discussion of the associated challenges in implementation.

The project's completion plan entails algorithm implementation, the associated hardware upgrades, implementation of third-party software for system monitoring and a streamlined development experience, and testing. Algorithm implementation is planned to follow the aforementioned sequence wherein F-LOAM comes first alongside dynamic gait optimisation, then obstacle detection and avoidance, followed by pathfinding, and the development and debugging suite being implemented when it is available. Appropriate testing of each new function will be conducted with the project's objectives, and therefore the system requirements and user needs, in mind. Where testing requires accurate measurement of position, velocity, or acceleration, the OptiTrack system in the Extraterrestrial Environment Simulation (EXTERRES) lab at The University of Adelaide will be used and the relevant data extracted. Completion of this iteration of the project will be marked by fulfillment of all five project objectives. Upon completion, the robot will be able to autonomously navigate complex environments including the Naracoorte Caves. In doing so, it is able to meaningfully contribute to the ongoing mapping and exploration work conducted by Williams, Reed, and other researchers exploring hard to reach areas of an environment.

Authorship Declaration

The authors of this report, as named and signed below, declare that all material in this assessment is our own except where there is clear acknowledgement and reference to the work of others. All authors have read the University Policy Statement on Plagiarism, Collusion and Related Forms of Cheating. All authors give permission for our assessment work to be reproduced and submitted to other academic staff for the purposes of assessment and to be copied, submitted, and retained in a form suitable for electronic checking of plagiarism. The authors of this report also advise that all intellectual property belongs to the University of Adelaide. The CaveX Honours Project is intended to be continued with different team members in 2024 and beyond.

By signing, each author is acknowledging that they have read the report in its entirety and are verifying the relevant quality assurance has been satisfied. Additionally, they are officially stating their contribution to the work.

Name	ID	Contribution	Signature
Luka Moran	a1726133	1/3	
Riley Groome	a1766553	1/3	
Tyler Groome	a1766574	1/3	

Contents

Executive Summary	i
Authorship Declaration	iii
Nomenclature	x
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 CaveX Mission Statement	3
1.4 2023 Project Aims, Scope, and Objectives	3
1.5 Project Management	5
1.6 Report Structure	5
2 Methodology	6
2.1 Systems Engineering	6
2.2 Agile Engineering	7
3 Literature Review	9
3.1 Mapping & Localisation	9
3.2 Obstacle Detection & Avoidance	10
3.3 Pathfinding	11
3.4 Dynamic Gait Optimisation	12
3.5 Findings	13
4 Problem Definition	15
4.1 Stakeholder Analysis	15
4.2 System Context	15
4.3 User Needs	15
4.4 System Requirements	16
4.5 Existing Prototype	16
4.5.1 Analysis	16
4.5.2 Analysis Outcomes	17
4.6 Prototype Performance Analysis	17
4.6.1 Incline and Decline Velocity Tests	18
4.6.2 Summary	19
5 Identified Prototype Issues	20
5.1 Mapping & Localisation	20
5.2 Autonomous Navigation	20
5.2.1 Obstacle Detection & Avoidance	20
5.2.2 Pathfinding	20
5.3 Dynamic Gait Optimisation	20
5.4 Computational Power	21
5.5 Development & Debugging	21
5.6 Summary	21

6 Preliminary Design	22
6.1 Mapping & Localisation	22
6.2 Navigation	23
6.2.1 Obstacle Detection & Avoidance	23
6.2.2 Pathfinding	24
6.3 Dynamic Gait Optimisation	24
6.4 Computational Power	24
6.5 Development & Debugging	25
6.6 Summary	25
7 Completion Plan	26
7.1 Mapping & Localisation	26
7.2 Autonomous Navigation	26
7.3 Dynamic Gait Optimisation	26
7.4 Development & Debugging	27
7.5 Summary	27
8 Conclusion	28
References	29
Appendices	34
A Project Management	34
A.1 SMART objective Justification	34
A.2 Stakeholder Analysis	36
A.3 Scenario Based Needs Analysis	37
A.4 System Context Diagram	38
A.5 CaveX 2021 Stakeholder Needs	39
A.6 CaveX 2021 System Requirements	40
A.7 CaveX 2023 User Needs	41
A.8 CaveX 2023 System Requirements	42
A.9 Work Breakdown Structure	46
A.10 Gantt Chart	47
A.11 Sprint Plan	50
A.12 Risk and Safety	51
A.12.1 Project Risks	52
A.12.2 Safety Risks	56
A.13 Quality Management	66
B Technical Datasheets	71
B.1 Jetson Nano	71
B.2 Jetson Orin NX 8GB	73
B.3 Velodyne Puck	75
B.4 Arduino Uno	77
B.5 Battery	81
B.6 Temperature & Humidity Sensor	82
B.7 Radio Frequency (RF) Module	84

B.8 WiFi Module	89
B.9 Servo Motors	92
B.10 Proximity Sensors	94
B.11 U2D2 Motor Board	95
C Preliminary Testing MATLAB Code	99

List of Figures

1	The current CaveX hexapod robot prototype, featuring upgraded actuators, a lighter structure, and more robust control software. (Bright et al. 2022)	1
2	The Systems Engineering "V" Diagram (Allouis et al. 2013)	6
3	Snapshot of the Jira taskboard used for project management	8
4	Example artificial potential field and the resulting optimal path trajectory (Kim et al. 2018)	12
5	Preliminary testing of the CaveX robot using the incline rig in the EXTERRES laboratory.	18
6	Gait velocity comparison of the CaveX robot for selected inclination angles	18
7	Gait velocity comparison of the CaveX robot for selected declination angles	19
8	CaveX 2023 technical objectives progress	27
A.1	Stakeholder map showing interactions between the project and those that have influence, or are influenced by the project.	36
A.2	Influence and interest grid of various stakeholders	36
A.3	Scenario-based User Needs Analysis	37
A.4	System context diagram displaying interfaces with key external entities	38
A.5	CaveX Stakeholder Needs Table - Needs gathered from stakeholder interviews and environmental constraints (Cooper et al. 2021).	39
A.6	CaveX System Requirements Table - Requirements with ID, Description, derivation from stakeholder needs and applicable traceability (Cooper et al. 2021).	40
B.1	Specifications for the Dualsky 11.1V 3-cell LiPo battery used in 2022 prototype (Dualsky 2022)	81

List of Tables

1	Abbreviations, acronyms, initialisms, and their definition.	ix
1.1	The project's technical objectives with target completion dates.	3
3.1	Potential solutions identified in literature organised by theme.	14
A.1	CaveX 2023 User Needs.	41
A.2	CaveX 2023 System Requirements.	42
A.3	Project risk assessment for CaveX 2023 Honours Project	52

Nomenclature

Table 1: Abbreviations, acronyms, initialisms, and their definition.

Abbreviation	Definition
AI	Artificial Intelligence
A-LOAM	Advanced LiDAR Odometry and Mapping
ARE	Average Rotational Error
ATE	Average Translational Error
CaveX	Cave Exploration
CD	Continuous Deployment
CI	Continuous Integration
CPU	Central Processing Unit
CSIRO	Commonwealth Scientific and Industrial Research Organisation
CUDA	Compute Unified Device Architecture
DOF	Degree of Freedom
DSC	Density-based Spatial Clustering
EKF	Extended Kalman Filter
EXTERRES	Extraterrestrial Environmental Simulation
FFSM	Foot Force Stability Criterion
F-LOAM	Fast LiDAR Odometry and Mapping
FLOPS	Floating Point Operations Per Second
FOV	Field of View
Gb	Gigabit
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
IQR	Interquartile Range
KD Tree	K-Dimensional Tree
LeGO-LOAM	Lightweight and Ground-Optimised LiDAR Odometry and Mapping
LiDAR	Light Detection and Ranging
LOAM	LiDAR Odometry and Mapping
MB	Megabyte
MFFSM	Modified Foot Force Stability Criterion
ML	Machine Learning
OpenSHC	Open-source Syropod High-level Controller

NOMENCLATURE

Abbreviation	Definition
RA	Risk Assessment
RADAR	Radio Detection and Ranging
RANSAC	Random Sample Consensus
RL	Reinforcement Learning
ROS	Robot Operating System
SLAM	Simultaneous Localisation and Mapping
SMART	Specific, Measureable, Attainable, Relevant, Time-based
SN	Stakeholder Need
SOP	Safe Operating Procedure
SR	System Requirement
SNN	Spiking Neural Network
UN	User Need
WiFi	Wireless Fidelity
2D	Two Dimensional
3D	Three Dimensional

1 Introduction

The Cave eXploration (CaveX) honours project aims to assist researchers in understanding cave systems through an improved ability to create maps of them. This iteration of the project serves to introduce autonomous functionality into the prototype from the project's 2022 iteration, shown in Figure 1. Such an expansion of capability is primarily going to be developed around the use of Light Detection and Ranging (LiDAR) data as it permits autonomous navigation without excessive disturbance of cave environments.

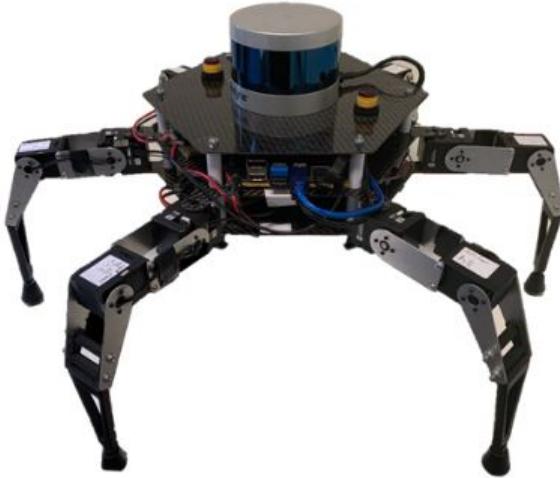


Figure 1: The current CaveX hexapod robot prototype, featuring upgraded actuators, a lighter structure, and more robust control software. (Bright et al. 2022)

1.1 Background

Caves serve as scientific time capsules due to their isolation from the world and minimal human disturbance. Consequently, they are of particular interest to palaeontologists, geologists, and researchers of other scientific fields to better understand important ecological artefacts (Kambesis 2009). Researchers however are often limited by unfavourable cave structures limiting their access to areas (Kambesis 2009). Palaeontologists from The University of Adelaide, Dr Elizabeth Reed and Craig Williams, are currently mapping South Australia's Naracoorte Caves which involves the mapping of 23 distinct caves, each with unique morphological features (Kambesis 2009). Accurate and detailed maps facilitate an improved understanding of the caves' structures, formation, and composition. Currently, the caves are mapped using a tripod-mounted LiDAR system that is manually moved with human involvement (Bright et al. 2022, p. 1). This system is unfavourable as it is time consuming, too large or cumbersome to scan some parts of the caves, and labour-intensive (Bright et al. 2022, p. 1). Thus, a system which improves on these qualities would considerably improve the quality of research and the rate at which research can be conducted (Bright et al. 2022, p. 1). A system that can also reduce the impact of human presence in natural cave environments is desirable.

The CaveX honours project, started in 2021, intends to create an autonomous biomimetic robot with cave mapping capabilities beyond that of current technologies. In particular, the autonomous system should be able to map difficult-to-reach sections of cave systems to expand researchers' ability to collect data and complete existing maps of cave systems. To improve the

system robustness a gait control system will also be implemented such that the movement of the robot will automatically adjust depending on the type of terrain. Implementing this feature will ensure the robot can adapt to different conditions throughout the cave environment improving its reliability. The current iteration of the robot can traverse cave systems under the control of a human operator but lacks any significant autonomous functionality. Implementing autonomy would not only improve the useability of the prototype but will also improve the system effectiveness and deliver its maximum potential value to researchers.

The robot currently uses a LiDAR sensor to generate a point cloud map of the surrounding cave environment. This point cloud map could also be used as a map to aid the robot in journey planning and autonomous traversal of the local environment. Simultaneous localisation and mapping (SLAM) algorithms are popularly paired with LiDAR data for autonomous movement. The aim of SLAM algorithms is to build maps of the environment surrounding the robot whilst also simultaneously determining the position and orientation of the robot within the map itself (Zou et al. 2022). Utilising the three-dimensional point cloud map produced by a LIDAR sensor allows for depth to be incorporated effectively into the map which makes it advantageous over computer vision navigation two-dimensional images from a camera (Zou et al. 2022).

1.2 Motivation

The current method of mapping the Naracoorte caves includes a large tripod mounted LiDAR sensor which is unfavourable in mapping tight cave passages. Previous iterations of the CaveX project have been able to produce a robot which is capable of mapping the cave environment under the control of a human operator. In order to manually control the robot, the operator needs a good understanding of the path the robot will take and needs to be aware of any obstacles in its path. This means that the robot's path is limited to what the operator can see thus effecting the robots effectiveness and robustness. The possibility of finding previously undiscovered cave entrances or formations is also significantly lowered. Autonomous operation of the robot can be implemented into multiple facets of its operation including movement, gait control, path planning and obstacle avoidance.

The current prototype utilises the LiDAR sensor to generate a three-dimensional (3D) point cloud map of the caves interior surface. Autonomous movement can be implemented through machine learning (ML) and artifical intelligence (AI) algorithms integrated into this data from the LiDAR sensor allowing the robot to understand its position in the map. Autonomous movement will allow the robot to move without direct control of the user which extends its utility and breadth of searching in the cave environment. An important milestone to achieving effective autonomous traversal of the cave environment is the ability for the robot to sense the terrain it is on and to adjust its gait accordingly. Different gaits are suited to different surface conditions as they provide the robot with specific measures of speed and stability. Traversing on surfaces with non-optimal gaits can significantly limit the speed of the robot and can also cause it to loose stability. Other complications such as motor stalling can also occur. The robot must also have a path planning capability to plan its next steps for its journey through the cave in real-time. Lastly, the cave system is far from a smooth, obstacle-free environment. This means that the robot must have the ability to detect obstacles and consequently alter its journey to avoid these obstacles. Implemented together, these abilities can supply the robot with an autonomous capability which would greatly improve its operation and value to researchers endeavouring to map the Naracoorte Caves.

1.3 CaveX Mission Statement

The 2022 CaveX project team developed a mission statement which is intended to represent the overarching goal for all CaveX project teams.

CaveX mission statement: To design and build a bio-inspired autonomous robot for the purpose of exploration and 3D mapping of cave systems.

The 2023 CaveX project team's aims are consistent with the mission statement. This year's project iteration will specifically focus on implementing an autonomous capability onto the robot. Previous years have not achieved this functionality which is directly mentioned in the CaveX mission statement.

1.4 2023 Project Aims, Scope, and Objectives

The aim of the 2023 CaveX project is to introduce autonomous capabilities, improve gait robustness, and real time system monitoring functionality into the prototype developed by the 2022 team to extend its utility in exploration and enable efficient development of the project's future iterations. This is to be achieved via the selection, design, implementation, and testing of software and algorithms on the prototype, as well as minor adjustments to its physical components as needed.

The project scope encompasses using the prototype to autonomously explore caves while collecting LiDAR scans and the production of maps from such data. Analysis and integration of these maps with existing data is intended to be conducted by Craig Williams or another third party. The software improvements are primarily intended to introduce autonomous capabilities into the prototype, although a secondary purpose is to facilitate efficient development and debugging. The intent of any physical modifications is to facilitate the software implementation.

The project's objectives signify specific development milestones and have been developed using the Specific, Measurable, Achievable, Relevant, Time-based (SMART) goal structure. This ensures they are justified, achievable, and meaningful. SMART objectives build a strong foundation for the project timeline, and maintain consistency with the 2022 CaveX project team's development process. The 2023 project objectives are summarised in Table 1.1 and defined comprehensively in Appendix A.1.

Table 1.1: The project's technical objectives with target completion dates.

ID	Objective Description	Specifications	Deliverables / Outcomes	Target Date
OB1	Implement real time localisation and mapping capability.	<ul style="list-style-type: none"> • The robot will be able to generate a 3D point cloud map and determine its position within this map in real time. 	<ul style="list-style-type: none"> • Custom made software to drive robot. 	18 June 2023

ID	Objective Description	Specifications	Deliverables / Outcomes	Target Date
OB2	Implement and optimise a pathfinding algorithm.	<ul style="list-style-type: none"> The robot will be able to plan its path for the next 25 steps in real time. The robot will be able to use data from sensors to explore unmapped sections during its mission with an error no greater than 10cm between predefined waypoints. 	<ul style="list-style-type: none"> Custom made software to drive robot. 	10 Sep 2023
OB3	Implement terrain sensing and gait control system.	<ul style="list-style-type: none"> The robot will be able to sense ground characteristics and adjust gait accordingly (hardness, rough terrain, slope). 	<ul style="list-style-type: none"> Ground sensing software. Gait optimisation algorithm. Leg force sensors. 	13 Aug 2023
OB4	Implement obstacle detection and avoidance capability.	<ul style="list-style-type: none"> The robot will be able to avoid 95% of the obstacles it encounters. 	<ul style="list-style-type: none"> Custom made software to drive robot. 	10 Sep 2023
OB5	Implement real-time data monitoring system.	<ul style="list-style-type: none"> The robot will be able to wirelessly transmit and receive data with a delay no greater than 3 seconds. The system should be able to display a real-time point cloud map to a connected user laptop as it is produced. 	<ul style="list-style-type: none"> System monitoring software. 	16 July 2023

1.5 Project Management

The 2023 CaveX project is managed using various project management tools. This year's project aims to have an emphasis on quality management and a quality assurance plan is included in Appendix A.13. The quality assurance plan is closely related to the project's risk management which involves identifying project risks and safety risks associated with specific implementation details. Appendix A.12 shows a risk assessment (RA) for project related risks and outlines mitigation strategies which help to ensure a high level of quality is achieved. Risk assessments and safe operating procedures (SOPs) for the safety risks associated with the technical work that has currently been undertaken is also presented in Appendix A.12. Jira is used to manage breaking down work and allocating tasks to group members which is discussed in Section 2.2.

1.6 Report Structure

This report is structured to outline the systems engineering approach undertaken by the team. The problem definition phase in Section 4 contains an analysis of key project stakeholders, the context of the system, and development of user needs and system requirements. Preliminary research is required to understand the current state of the art for autonomous robot operation using data from a LiDAR scanner, hence a literature review is included in Section 3. The next phase in defining the problem is to analyse the current prototype which includes various preliminary tests to identify issues or areas for improvement, this analysis is shown in Section 4.5. A concept design is then presented in Section 6 which includes implementing algorithms explored in the literature review to achieve the project's technical objectives. To ensure the objectives are met this report also details a completion plan in Section 7 for implementing all required functionality and a plan for additional hardware upgrades that may become necessary.

2 Methodology

The 2023 iteration of the CaveX project builds upon the existing work using robust project management strategies to ensure its objectives are achieved. A systems engineering approach is utilised to facilitate such accomplishments. Applying the systems engineering practices ensures that the engineering process is strictly informed by stakeholder needs. The stakeholder needs have been carried through both the 2021 and 2022 iterations of the CaveX project and have been adjusted according to stakeholder meetings. Periodic communication with the primary stakeholders ensures that the team is building the right system to meet stakeholder expectations which will ensure the system can successfully be validated. The implementation of the project objective's are to be done using an agile software engineering approach to allow for tasks to be completed in parallel.

2.1 Systems Engineering

The 2023 CaveX team will follow a systems engineering approach, shown in Figure 2, to fully break down and define the problem at hand and develop an innovative solution that meets the expectations of the final user.

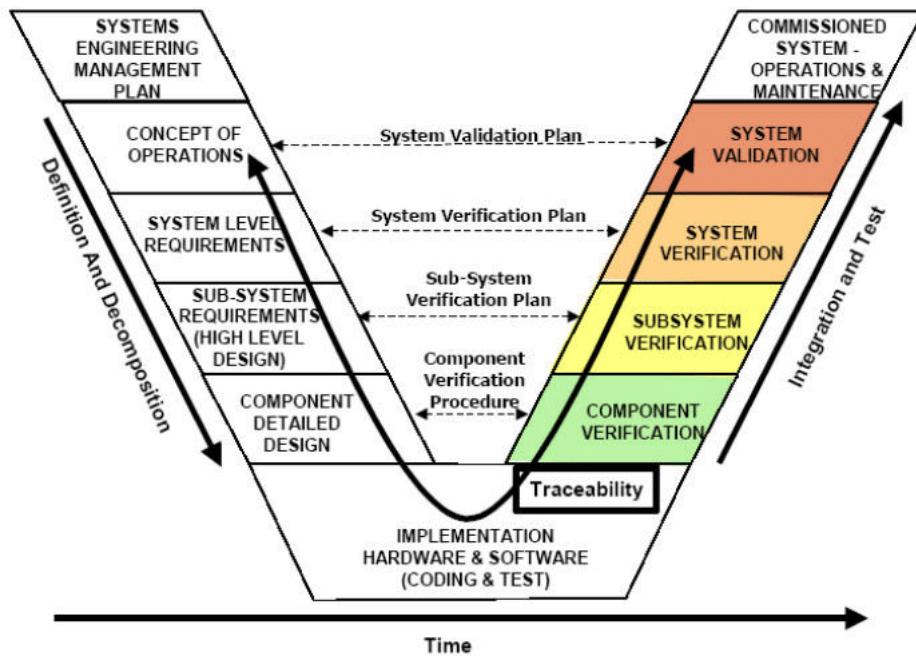


Figure 2: The Systems Engineering "V" Diagram (Allouis et al. 2013)

The problem definition phase at the beginning of the systems engineering "V" diagram will consist of a detailed stakeholder analysis to identify and understand all of the groups that can affect the project or are directly affected by the project. A stakeholder analysis is also important to understand potential restrictions from stakeholders and defines which groups of people should be closely worked with throughout the duration of the project. To better understand the entities that the system is interacting with a system context analysis will be conducted. The context of the system will assist in defining interface requirements for external systems that the CaveX robot will interact with.

The next important phase of the systems engineering approach is requirements development.

In this stage of the design process it is essential to understand the needs of the final user which will be transformed into system level requirements that guide the technical design. This project builds upon the user needs that were identified in the previous design iterations with slight adjustments due to discussions with primary stakeholders that focus on the autonomous direction of the project. A useful tool in understanding the fundamental needs of the user is to conduct a scenario based needs analysis where a typical use of the robotic system and all of the functions that are necessary to achieve the desired mission is analysed. After the user needs are identified system requirements which define functional and non-functional specifications for the final design are developed. The system requirements are developed in a similar way to the SMART framework used to describe the technical objectives in Appendix A.1. In particular, the requirements should be measurable to assist in the verification of the final design towards the end of the project.

After the development of system requirements a functional analysis will be conducted and concept solutions will be explored. The functional analysis will divide the system into smaller functional elements which assists in understanding key functionality and subsystem requirements. A literature review is essential to identify solutions to similar problems and how they could be applicable to this problem. In the literature review a range of different algorithms for autonomous functionality must be explored to understand the advantages and disadvantages of each approach. After a range of different algorithms are explored a preliminary concept design will be presented.

Verification and validation occurs towards the end of the engineering process. Full system validation is to be done at the Naracoorte Caves with the primary user Craig Williams to demonstrate the system has the required functionality to meet the needs of a user. The verification process involves ensuring the system has been built correctly and meets the technical objectives. Each of the system requirements listed in Table A.2 will be verified at the end of the project. There are four primary methods of verification which include analysis, inspection, demonstration, and testing. Demonstration and testing are the most costly but have the highest confidence in successfully assessing the quality of the final design. Inspection and analysis verification methods are of low confidence and should only be used in verification when it is infeasible to extensively test the system. To ensure a high level of quality, each system requirement will either be tested or the required functionality will be demonstrated. As outlined in the quality assurance plan in Appendix A.13, unit tests will be performed throughout the implementation phase to eliminate bugs early in the design process.

2.2 Agile Engineering

The implementation of the project's objectives is managed using an agile engineering methodology. This approach suits the project well, being a predominantly software-oriented project. The agile approach applied to software development allows for faster responses to changing requirements and situations which may present themselves during the project's duration. To compliment the agile software engineering approach, organised software development sprints have been devised. A plan for these sprints is shown in the Appendix A.11 and a schedule is shown in the Gantt chart in Appendix A.10. Some of the technical objectives in Table 1.1 are independent of others and, therefore, allows objective sub-tasks to be managed in a flexible manner. Hence, for an agile approach there is no definitive plan for the work to be conducted in each sprint. Deadlines have been set for technical objective completion to provide an

approximate timeline for task completion. These deadlines are also shown in the Gantt chart.

Jira is an issue tracking software developed by Atlassian that is utilised by many software engineering teams to manage workloads. It is used to assist the team in breaking down work into smaller tasks that can be assigned to individual group members. A snapshot of the 2023 Jira task board is shown in Figure 3.

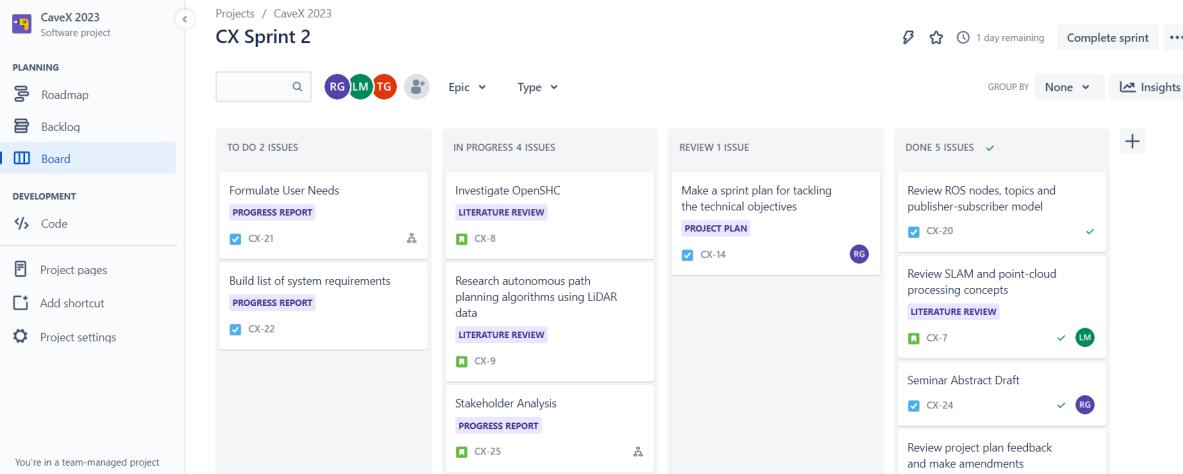


Figure 3: Snapshot of the Jira taskboard used for project management

Managing tasks using Jira has many benefits including a focus on project schedule and scope management. Therefore, using Jira allows for consistency with the work breakdown structure shown in Appendix A.9 and the project Gantt chart. Jira allows for tasks to be added to specific iterations of the software implementation which are the software development sprints. The sprints are four weeks in length and an approximate sprint plan is shown in Appendix A.11. This plan is used to manage the schedule of the project to ensure the objectives are completed by their deadlines. It is consistent with the schedule in the Gantt chart and the estimated objective completion dates stated in section 1.4. Where possible, objectives are designed such that they can be developed in parallel to mitigate the risk of unforeseen delays in early implementations which could be carried down the project timeline. These risks are prominent with a waterfall software engineering approach where functionality is implemented incrementally and depends heavily on predecessor milestones being completed. Hence, an agile methodology is justified to give the team flexibility in the schedule to investigate different functionality as it becomes necessary.

3 Literature Review

The exploration and documentation of caves underpins developments in the natural sciences, geographic databases, and conservation of subterranean fauna (Kambesis 2009; Bichuette & Trajano 2018). Historically, researchers have been constrained by dangerous or otherwise inaccessible cave features such as vertical drops and tight areas. Recent improvements in robotic technologies and their accessibility presents an opportunity to expand cave exploration capabilities (Dubowsky et al 2005). This literature review initially explores mapping and localisation technologies often implemented into robotic systems to determine which is most appropriate for the CaveX prototype. This analysis builds upon the conclusions of the 2022 CaveX team's literature review through a deeper assessment of SLAM algorithms and their analogous counterparts. Next, approaches to obstacle detection, avoidance, and pathfinding are considered in the context of the unique challenges imposed by caves. Finally, a review of gait optimisation strategies is conducted with consideration to its implementation with obstacle detection, avoidance, and pathfinding.

3.1 Mapping & Localisation

In robotics, mapping and localisation are the processes by which a robot constructs a representation of its environment and determines its location within an environment, respectively. Thus, SLAM is a process wherein a robot determines where it is located within its local environment while constructing a map of such environment (Mur-Artal, Montiel & Tardós 2015). Real time SLAM algorithms are attractive for systems requiring autonomous navigation due to a lack of reliance on outside data such as Global Navigation Satellite System (GNSS) information (Auat Cheein et al. 2010). The 2022 CaveX team determined that visual SLAM methods are unsuitable for cave exploration due to the presence of photosensitive fauna and flora. These methods would present a challenge in balancing the need for data to conduct SLAM with minimising damaging effects to cave inhabitants (Bright et al. 2022; Williams 2021). This, therefore, necessitates the implementation of LiDAR-based SLAM.

While there are several algorithms for SLAM using LiDAR, a common high-performance algorithm is the LiDAR Odometry and Mapping (LOAM) algorithm (Abdelaziz & El-Rabbany 2022). At a high level, it works by running two algorithms: one high-frequency but low-fidelity odometry algorithm to estimate robot velocity and one higher fidelity but lower frequency algorithm which determines the spatial transform between two LiDAR point clouds. The process of determining this spatial transform is known as matching. Separating these two functions facilitates reduced computational load at the expense of extremely high accuracy (Zhang & Singh 2014). Subsequent to LOAM's publication, improved variants such as Advanced LOAM (A-LOAM), Lightweight and Ground-Optimised LOAM (LeGO-LOAM), and Fast LOAM (F-LOAM) have been developed to provide improved performance (Abdelaziz & El-Rabbany 2022). A-LOAM focuses on general computational optimisation through the removal of redundant operations (Qin & Cao 2019). LeGO-LOAM targets computational performance and algorithm accuracy improvements for LOAM application in ground vehicles by windowing point clouds before analysis and reducing noise in data (Shan & Englot 2018). F-LOAM improves computational performance through a dynamic combination of feature extraction, distortion compensation, pose optimisation, and mapping. By assuming that the time between consecutive LiDAR scans is small, the linear and angular velocities in that time are assumed to be constant. This allows the distortion in a scan to be partially corrected by inferring the LiDAR scanner's motion

between the two scans. Then, after pose estimation, the distortion is re-computed to update the final map (Wang et al. 2021). In contrast, A-LOAM and LeGO-LOAM’s brute force iterative approaches to distortion compensation require greater computational resources. However, their approach avoids error due to variable linear and angular velocities between scans (Qin & Cao 2019; Shan & Englot 2018). Experimentally, Wang et al. (2021) showed that LeGO-LOAM has the smallest computational cost of LOAM, A-LOAM, LeGO-LOAM, and F-LOAM. However, LeGO-LOAM exhibited the highest average translational error (ATE) and average rotational error (ARE). At an ATE, ARE, and computing time of 0.80%, 0.0048deg/m, and 80ms, respectively, F-LOAM exhibits approximately 1.9 times less error than LeGO-LOAM in translation, 2.4 times less error in rotation, and is 12.5% slower. F-LOAM also outperformed LOAM and A-LOAM in all three metrics (Wang et al. 2021). Given the limited computational resources of the CaveX prototype’s NVIDIA Jetson Nano, F-LOAM presents the most ideal trade off between computing time and algorithm accuracy. However, all algorithms discussed have been tested on a central processing unit (CPU), which is not the ideal environment for point cloud processing (Wang et al. 2021; Anand et al. 2020). Thus, while porting them to a graphics processing unit (GPU) environment such as the Jetson’s CUDA cores is likely to yield a significant reduction in computing time, the exact improvements are unknown.

3.2 Obstacle Detection & Avoidance

Obstacle detection and obstacle avoidance are the phenomena wherein a system can identify obstacles in its path and make the appropriate adjustments to prevent a collision while continuing to accomplish the system’s objective(s). Therefore, solving these problems is critical to producing a system which is autonomy-capable. Obstacles can be broadly categorised as either positive or negative: positive obstacles are those which protrude up from the ground or otherwise physically impede a path, whereas negative obstacles are voids in the ground such as ditches, pits, or downward-sloped terrain (Shang et al. 2015, p. 591). Further classification of obstacles is to describe them as either static or dynamic, where static obstacles are stationary and dynamic obstacles are moving (Asvadi et al. 2016). This section describes and discusses existing LiDAR-based obstacle detection and avoidance methods to assess their utility in cave exploration.

Typically, obstacle detection is conducted using a sensor fusion approach with data from LiDAR or Radio Detection and Ranging (RADAR) combined with stereo and monocular cameras, inertial measurement units (IMUs), and GPS sensors. This is because much of the existing work done to tackle the problem is in the field of autonomous driving and advanced driver assistance systems (Badue et al. 2021; Asvadi et al. 2016). However, existing research has developed algorithms for LiDAR-only obstacle detection. One approach is to segment the ground surface from the overall LiDAR point cloud before voxelising the remaining obstacle point cloud for further static and dynamic obstacle segmentation. Due to the sparse and non-uniform nature of LiDAR point clouds, an interquartile range (IQR) gating strategy and random sample consensus (RANSAC)-based plane fitting is employed to minimise the impact of outliers. This algorithm showed promising results as it missed only 6.18% of obstacles and had a false-positive rate of 3.09%. (Asvadi et al. 2016). Critically, it conducts point cloud preprocessing in a manner similar to that of SLAM algorithms and, therefore, it may be possible to integrate them to reduce overall computational load. However, this method does not detect negative obstacles and therefore would need to be combined with an algorithm that does to be suitable for cave exploration. A lack of literature on combined algorithms means additional work is required to

determine an integration technique.

An alternative approach is a dynamic clustering algorithm which identifies objects through spatial clustering techniques. The approach developed by Gao, Li & Zhang (2021) builds upon previous findings regarding density-based spatial clustering (DSC) of applications with noise by accounting for the point cloud sparsity increase with distance from the sensor. This algorithm showed a true positive detection rate of 85.66% and a false negative rate of 5.98%. However, it suffers from the same issue as the previous algorithm in that it cannot detect negative obstacles (Gao, Li & Zhang 2021).

While previously-discussed algorithms have been based on hard-coded mathematical filters and geometric criteria, an emerging method of obstacle detection is applying deep learning techniques. Two deep learning models for point cloud classification and segmentation are PointNet and SECOND. PointNet focuses on object classification, part segmentation, and semantic segmentation whereas SECOND targets object and direction classification (Qi et al. 2017; Yan, Mao & Li 2018). Deep learning methods show promise for obstacle detection but are often inaccessible or suboptimal due to the specialised expertise needed, high-performance hardware required for training the models, or their non-deterministic nature rendering them a high risk endeavour. However, the SECOND model was trained in 19 hours on a single consumer NVIDIA GTX 1080Ti (Yan, Mao & Li 2018). Hence, if the two former issues can be overcome, then application of the model in a cave environment may be possible. One potential method to overcome the model's non-deterministic nature is to fuse it with a deterministic algorithm. However, this area is lacking in research.

Obstacle avoidance is dependent on obstacle data but can be independent of the algorithm used to acquire such data. A quintessential real time obstacle avoidance technique is the artificial potential field method developed by Khatib (1986). This method involves the construction of a model wherein virtual attractive and repulsive forces are produced by the goal location and the obstacles in the field, respectively. By applying an iterative approach where each iteration is triggered by the movement of a robot, the optimal path vector at the robot's position during the iteration can be determined. Thus, across a series of iterations, the robot can reach its goal position while avoiding obstacles in its path.

3.3 Pathfinding

In general, pathfinding is the process of determining the optimal path to take in space given a set of conditions (Le et al. 2018). Pathfinding algorithms can be categorised as global or local. Global pathfinding algorithms rely on complete data of the paths and obstacles in an environment, whereas local algorithms find a path from real time sensor data (Kim et al. 2018; Le et al. 2018). By definition, exploration robots are constrained by a lack of data and, therefore, rely on local pathfinding algorithms to navigate. A unique consideration of exploration robotics is that the system should prioritise unexplored paths over explored ones. Additionally, it should be capable of returning to its starting position via the path it originally took. Thus, existing local pathfinding algorithms are explored to examine their viability for cave exploration.

A modified version of the rapidly-exploring random tree (RRT) algorithm, executive rapidly-exploring random tree (ERRT), was implemented into a robot snake to facilitate LiDAR-based pathfinding. This algorithm involves setting a goal position before an iterative brute force approach is taken to find a path between the robot's current position and its goal. This approach

can be effective, but its brute force nature means the computational load grows extremely fast with distance to the goal position. This trend is due to the growth in the number of possible paths and the number of vertices within a path which, therefore, increases the number of states for the iterator to consider (Yang, Wang & Shen 2020; LaValle & Kuffner 2001).

A SLAM-based pathfinding algorithm for construction applications involves applying the artificial potential field method developed by Khatib (1986) based on the results of an obstacle detection algorithm to determine the optimal path to the goal position. By iteratively applying the Newton direction method while a robot moves, the optimal path vector at that location can be determined. As the robot approaches the goal position, the goal position can be re-allocated based on new sensor information or it can remain the same depending on what is most appropriate for the situation (Kim et al. 2018). Additionally, due to the artificial potential field method's virtual force behaviour, negative obstacles can be accounted for in the path planning process. However, such an implementation relies on the ability to detect such obstacles. Hence, detection is the major limitation of this algorithm.

The primary limitation of pathfinding algorithms in robotics, such as RRT, ERRT, and the artificial potential field method, is a lack of consideration of a robot's kinematic constraints; the algorithms focus solely on the generation of a path without obstacles. Limited literature in this area means that a custom algorithm would need to be developed wherein a kinematic model of the robot predicts its kinematic state at future iterations of a path and adjusts the path accordingly.

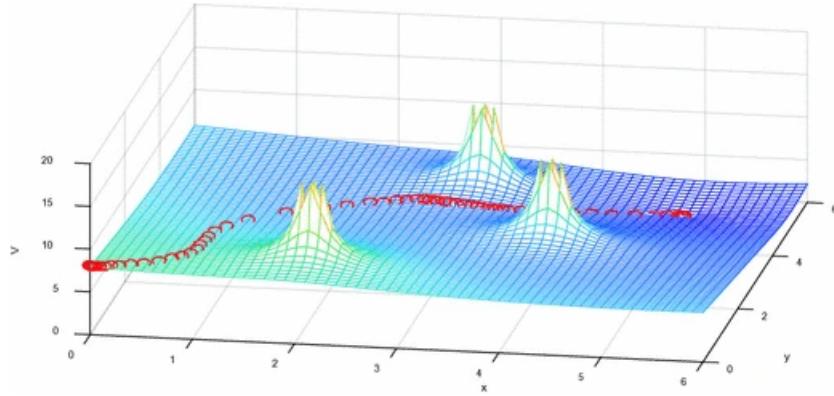


Figure 4: Example artificial potential field and the resulting optimal path trajectory (Kim et al. 2018)

3.4 Dynamic Gait Optimisation

The goal of dynamic gait optimisation in quasi-static multi-legged robots, such as a hexapod, is to improve their real time adaptability and stability in challenging terrain. This ability is of particular importance in cave exploration as cave surfaces are often non-uniform and unpredictable (Novak et al. 2021). This section explores existing gait optimisation algorithms to assess their suitability to a cave-exploring hexapod.

Previously, a basic style of gait optimisation in quadrupedal and hexapod robots has been to ensure its body is aligned with the terrain plane and its height kept constant. The primary issue with this technique is a lack of consideration to spatial constraints. Tight spaces and non-uniform terrain can result in a failure. Thus, this technique alone is suboptimal for unpredictable

cave terrain (Chen, Liu & Gan 2022; Bartsch et al. 2012). Chen, Liu & Gan (2022) found that employing a minimisation and pseudo-inverse optimisation technique improved a six-legged robot's walking speed by 17.1% and adaptability in slopes and steps by 48.5% and 96.0%, respectively. However, their robot structure, while six-legged, was different to the current CaveX prototype in that each leg is comprised of three sub-legs. Despite this, their kinematic analysis may be ported to an appropriate structure.

An alternative approach to gait optimisation is to directly measure the force applied to each of robot's feet. The Foot Force Stability Margin (FFSM) was developed to serve as a force-based test to assess a system's stability. Its low computational cost and sensitivity make it attractive for real time applications. However, its simplicity constrains it to act as an indicator of stability rather than a process to achieve stability. Subsequently, the Modified Foot Force Stability Margin (MFFSM) was developed to build upon FFSM by taking robot geometry and top-heaviness into account (Agheli & Nestinger 2012). Thus, MFFSM could be used to form an iterative stability algorithm wherein a robot's centre of gravity is lowered to increase stability when a low-stability state is detected. This process, however, may be suboptimal as it has only one course of action to correct a low-stability state.

More complex techniques fuse advanced forward and inverse kinematic analysis or artificial intelligence with sensor feedback to facilitate dynamic terrain compensation. One such algorithm involves using a dynamic model, motor torque feedback, and locomotion state information to iteratively adjust the endpoint positions of legs being lowered such that they account for ground surface inconsistencies (Zangrandi, Arrigoni & Braghin 2021). While this approach shows promising results, its complexity and reliance on a terrain elevation function present significant development challenges. An alternative approach is to implement a reinforcement learning (RL) algorithm which provides a control system with the ability to learn via trial and error. In one approach, a Spiking Neural Network (SNN) took visual and gyroscopic input to learn gait optimisations via determining whether stability was lost during a movement and factoring that into its reward function. The SNN frequently converged with the target gait pattern but, at a median of 207 iterations for convergence, spent considerable time learning (Lele et al. 2020). While RL has significant potential for future applications, its inherent stochastic nature significantly increases the expertise required for development and risk to the robot during its operation.

Overall, of the gait optimisation approaches discussed, the algorithm developed by Zangrandi, Arrigoni & Braghin (2021) provides the best combination of dynamic capability and algorithmic determinism. However, this comes at the cost of complexity. FFSM and MFFSM are attractive due to their simplicity but this simplicity also constrains the resulting algorithms. The approach adopted by Chen, Liu & Gan (2022) shows considerable improvements but for an alternative hexapod morphology. RL-based algorithms have the potential yield impressive dynamic capabilities but require considerable expertise to optimise.

3.5 Findings

The findings from the literature review are summarised in Table 3.1. The advantages and disadvantages of different solutions to implement the project's technical objectives are compared for a variety of different sources.

Table 3.1: Potential solutions identified in literature organised by theme.

Theme	Source	Summary	Outcome
Mapping & Localisation	Qin & Cao 2019	A-LOAM underperforms compared to LeGO-LOAM and F-LOAM	F-LOAM is the most appropriate SLAM algorithm given the computational constraints
	Shan & Englot 2018	LeGO-LOAM requires the least computational power but produces the most error	
	Wang et al. 2021	F-LOAM provides the best balance of accuracy and computational resources	
Obstacle Detection & Avoidance	Asvadi et al. 2016	Statistical RANSAC approach identified 94% of obstacles but cannot detect negative obstacles	The statistical RANSAC approach to obstacle detection and the Artificial Potential Field Method are the optimal techniques for obstacle detection and avoidance
	Ghao, Li & Zhang 2021	TP rate of 85% but cannot detect negative obstacles	
	Qi et al. 2017	PointNet can perform object classification but is complicated and non-deterministic	
	Yan, Mao & Li 2018	SECOND can perform object and direction classification but is complicated and non-deterministic	
Pathfinding	Yang, Wang & Shen 2020; LaValle & Kuffner 2001	Effective but computationally expensive	The Artificial Potential Field Method is the best approach as it allows for iterative updating of the ideal velocity vector and limits brute forcing a path
	Khatib 1986; Kim et al. 2018	Works for positive and negative obstacles, computationally cheap	
Dynamic Gait Optimisation	Chen, Liu & Gan 2022	Effective but for a different hexapod anatomy	The dynamic model developed by Zangrandi, Arrigoni & Braghin (2021) provides the optimal combination of capability and determinism in gait optimisation
	Agheli & Nestinger 2012	Too basic for a cave environment	
	Zangrandi, Arrigoni & Braghin 2021	Effective and for the appropriate hexapod anatomy	
	Lele et al. 2020	Somewhat effective but experimental and stochastic	

4 Problem Definition

The beginning of the project’s problem definition phase explores the project stakeholders and the effect they have on the project. Next, the context of the CaveX robot system is explored to identify the key interfaces required to implement the desired functionality. Finally a scenario-based needs analysis is conducted to understand a typical operation of the robot and the functionality that is required by the user. A detailed list of user needs are developed from this analysis which are then transformed into system level requirements which aim to guide the technical design to meet the user’s expectations.

4.1 Stakeholder Analysis

To gain an improved understanding of the key stakeholders, a stakeholder analysis has been conducted and is shown in Appendix A.2. The stakeholder analysis includes a map displaying significant interactions between groups and provides a grid depicting the varying influence and interest of the stakeholders. The stakeholders that have high interest and influence are the project supervisors, including The University of Adelaide paleontologists Craig Williams and Dr Elizabeth Reed, and the university technicians who assist with implementing hardware. These stakeholders are to be worked with closely throughout the project. Other important stakeholders include the cave inhabitants, legislative groups and Naracoorte Caves’ staff who have high influence on the design due to the constraints they place on the system’s operation.

4.2 System Context

A system context diagram, shown in Appendix A.4, defines all of the external entities the robot must interact with during its operation. These interactions define key interface requirements and behaviour that guides the system’s design. The system context diagram also defines the systems that place constraints on the design.

The key interface for the system design is the communication between the remote user and the CaveX robot which is a major distinction between the current and previous project iterations. There must exist a reliable communication system that allows data to be sent to a platform which can display real time cave maps which is outlined in OB5 in Table 1.1. The communication system must be two-way to also allow instructions to be received from a remote operator. Some important constraints which arise from the system context diagram are the Naracoorte Caves environment and natural wildlife as they impose special design considerations, namely noise reduction and obstacle avoidance (OB4).

4.3 User Needs

In keeping consistent with the systems engineering approach, a set of user needs are formulated which form the basis of the engineering design considerations for this year’s project. Much of the user needs developed by the 2021 and 2022 teams will be reused in this iteration and a full list of the previous iteration’s stakeholder needs are shown in Appendix A.5. A key distinction between the stakeholder requirements from previous years and the user needs from this year is that previous iterations developed requirements of all entities which interact with the system. Since these requirements have been validated by previous iterations, the focus this year is to develop needs from the direct users of the system. However, the stakeholder requirements will

still form a integral part to the design process to ensure that new functionality does not adversely effect these requirements.

To understand the needs of the end user, a typical use case of the robot system was considered through the scenario-based needs analysis shown in Appendix A.3. From this analysis and the previous stakeholder requirements, the 2023 user needs were collated and summarised in Appendix A.7. The needs are traceable to invalidated requirements from previous project iterations, but they have a refined focus on autonomy to align them with the CaveX mission statement. Additionally, some of the invalidated user requirements have not been included in this iteration since they are out of the scope of the project and do not directly relate to its technical objectives. These primarily include hardware-oriented upgrades, although hardware upgrades are within the scope of the project, they will not be considered unless software necessitates their inclusion.

4.4 System Requirements

The user needs serve as the fundamental basis of the system requirements. A complete list of the system requirements from 2022, which focused on detailed hardware design, are included in Appendix A.6. The additional system requirements that arise from this year's project scope are shown in Appendix A.8. These requirements were derived partly from the 2022 iteration's unverified requirements. Some requirements were not verified by the 2022 team but are not included in the 2023 requirements as they do not fall under this year's project scope, do not significantly effect the quality of the project, and do not stem directly from an essential user need. This year's system requirements are integral in the project meeting the technical objectives displayed in Table 1.1.

4.5 Existing Prototype

In this section a detailed analysis of the 2022 CaveX project team's prototype and the conceptual work it was built upon is conducted with a particular focus on the associated software. The purpose of this is to identify factors which influence the accomplishment of the 2023 objectives (see Table 1.1). Hardware is discussed where it constrains or otherwise impacts the implementation of new software.

4.5.1 Analysis

The current prototype functions using two primary software components: Robot Operating System (ROS) version 1 and the Commonwealth Scientific and Industrial Research Organisation's (CSIRO) open-source Syropod High-level Controller (OpenSHC). ROS serves as the backbone of all control and communication-related software and runs on a Linux operating system, Ubuntu 18.04 (Bright et al. 2022). Software within ROS has several layers of abstraction: packages, nodes, topics, and messages. Packages are repositories of executable files (nodes) which interface with ROS and the libraries required to run such nodes. Nodes are typically single-function executables launched by ROS, topics are the means of inter-node communication, and messages are the agreed format by which a topics communication occurs (ROS Wiki 2019a; ROS Wiki 2019b; ROS Wiki 2022). The current prototype runs three nodes to achieve its function: dynamixel_interface for interfacing with the servo motors, shc for running the OpenSHC software, and syropod_remote for listening to controller-based commands. The shc node receives a desired velocity vector from syropod_remote through the desired_velocity topic before being

used to calculate the desired position of each joint. These positions are then transmitted to dynamixel_interface via the desired_joint_states topic (CSIRO Robotics 2021; CSIRO Robotics 2020). The complexity of the existing control system may make integrating improved algorithms difficult by constraining it to function within the existing code base and, therefore, make achieving OB3 difficult. However, the kinematic model used by OpenSHC contains considerable overlap with many existing hexapod control algorithms (Tam et al. 2020; Zangrandi, Arrigoni & Braghin 2021). Hence, it is feasible to build from the existing model.

The prototype's LiDAR sensor is a Velodyne VLP-16 puck (Bright et al 2022). In its current configuration it transmits 1 megabyte (MB) of point data every second. The sensor provides 360° horizontal coverage and a vertical field of view (FOV) of $\pm 15^\circ$ relative to the sensor's horizontal plane. Additionally, the data is transmitted directly to the Jetson Nano via 10 gigabit (Gb) per second ethernet (Velodyne 2019). Thus, it is sufficient to support OB1, OB2, and OB4. However, the limited processing power of the Jetson Nano means significant care may be required in the implementation of new algorithms to optimise them and allow room for future development.

Currently, the prototype does not have a wireless interface with other devices (Bright et al 2022). A lack of WiFi connectivity means it is practically impossible to implement a real-time data monitoring system (OB5). To implement such a system, the prototype must be able to wirelessly transmit data at a high bit rate to a receiver system. Thus a WiFi module must be introduced to achieve OB5.

4.5.2 Analysis Outcomes

The analysis of the existing prototype shows that some changes are required to accomplish the 2023 objectives. Specifically, new software needs to be designed carefully to fit in with the existing system and to make the best use of its computational resources (OB1, OB2, OB3, and OB4), and a WiFi module is required to enable the implementation of a realtime system monitoring system (OB5).

4.6 Prototype Performance Analysis

To assist in identifying areas of improvement for the current prototype, preliminary testing has been undertaken using the OptiTrack system in the Extraterrestrial Environmental Simulation (EXTERRES) laboratory at The University of Adelaide. This testing will also form a baseline benchmark which can be compared to after algorithm implementation to conduct the validation and verification process outlined in Figure 2. The tests include using an adjustable ramp to measure the speed of the robot across different angles of incline and decline. The maximum incline and decline angles that the robot can travel can also be addressed. The ramp testing apparatus used for the preliminary testing in the EXTERRES laboratory is shown in Figure 5. A mulit-terrain testing rig is also planned to be used to assess the robot's velocity and versatility on different terrains. The velocity of the robot will also be measured using the OptiTrack system to accurately assess gait effectiveness on different terrains. This testing has not yet occurred but is planned for the 30th of May as shown in Appendix A.10.

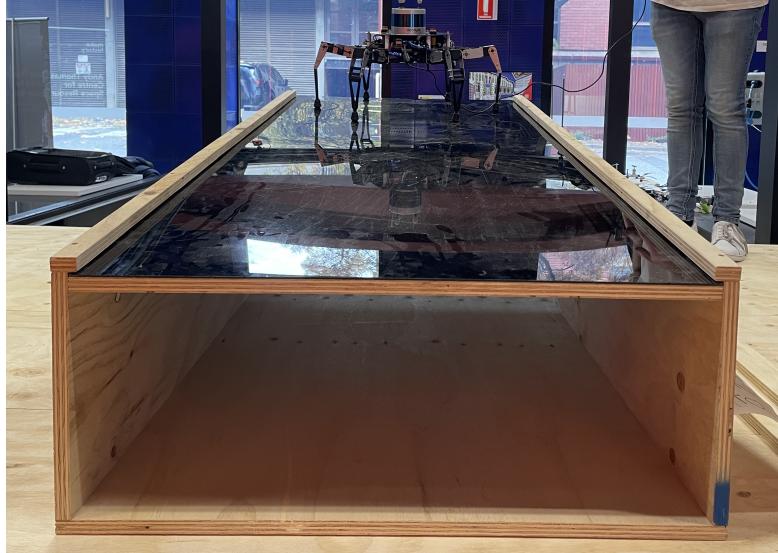


Figure 5: Preliminary testing of the CaveX robot using the incline rig in the EXTERRES laboratory.

4.6.1 Incline and Decline Velocity Tests

The measurements for the robot's speed at different angles of incline and decline were taken for each of its three gaits: tripod, ripple, and wave. The inclination speed testing is shown in Figure 6 and the MATLAB code used to process and plot the OptiTrack data is shown in Appendix C.

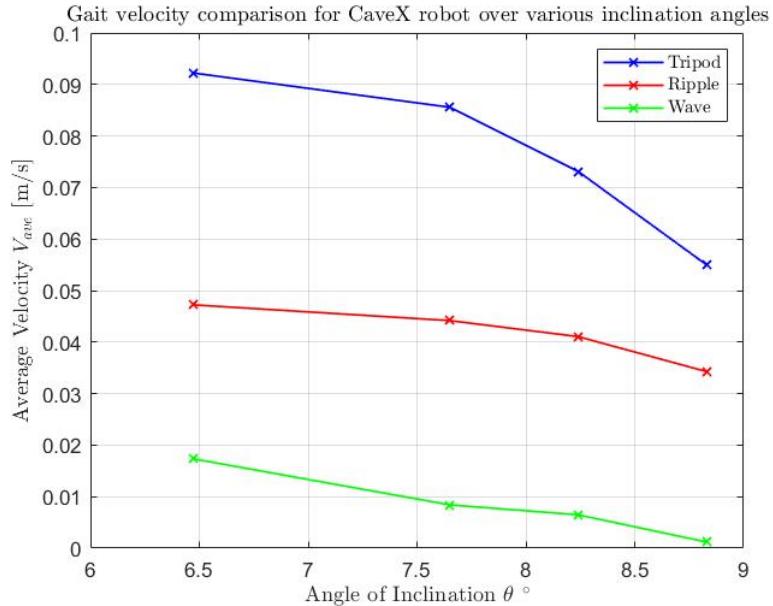


Figure 6: Gait velocity comparison of the CaveX robot for selected inclination angles

The measurements in Figure 6 show that as the angle of inclination increases, the speed of the robot decreases. Further, they show an inverse relation between static stability and velocity. The tripod gait has the highest speed of the three gaits but it reduces sharply as the angle increases. The ripple gait is the second fastest and appears to have a slight decrease in speed as

angle increases. Lastly, the wave gait has the slowest speed measured. As the angle increases to the last measured angle of 8.83° , the speed of the wave gait approaches a zero value which represents the angle that the robot can no longer traverse the ramp. The decreasing trend in speed between the gaits also follows a concordant decrease in the number of moving legs. Thus, it is likely that the reason for such change is a reduced net force applied to the robot. The speed measurements for the decline testing across each of its three gaits is shown in Figure 7.

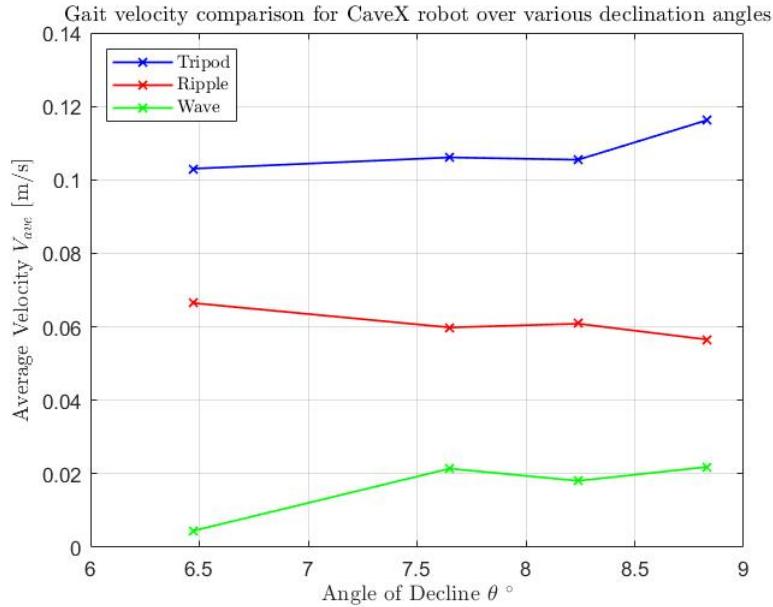


Figure 7: Gait velocity comparison of the CaveX robot for selected declination angles

The results in Figure 7 show that the tripod gait has the fastest movement speed across the decline angles, similar to the results shown in Figure 6. As the angle of declination is increased, the speed of the robot in its tripod gait increases. The ripple gait is again the second fastest gait. However, as the declination angle is increased its speed slightly decreases. Lastly, the wave gait is the slowest gait. As the declination angle increases the speed appears to increase. A noteworthy observation is that the change in speed between decline angles is relatively small compared to the inclined tests.

4.6.2 Summary

The testing results show consistency between the fastest and slowest gaits. The incline speed testing results demonstrate that the speed of the prototype decreases as the slope increases, an expected trend as the robot must overcome a larger vertical component of its own weight as the angle increases. In the declination results, the speed is observed to increase for the tripod and waves gaits but slightly decrease for the ripple gait. The difference in the trend of the results across the three gaits in the declination testing is due to the difference in stability and leg movements. This means that some gaits, such as the tripod and wave, are better suited to a declined slope as they allow for faster movement. The ripple gait is less impacted by the incline and decline angles as observable by the shallower slopes shown in the figures.

5 Identified Prototype Issues

The issues or areas for improvement with the current prototype are discussed in this section and form the basis for the project's technical objectives.

5.1 Mapping & Localisation

In the context of autonomy, mapping is where the robot has an understanding of the environment within which it resides and localisation is the state wherein a robot knows its location with respect to this environment. The current prototype collects map data using its LiDAR sensor but does not construct a map in real time and has no localisation capability. These functions underpin further autonomous abilities and, thus, the current prototype does not possess the ability to demonstrate autonomous behaviour.

5.2 Autonomous Navigation

The robot is currently manually controlled which limits its utility and ability to satisfy the system requirements. Autonomous navigation is the key objective of this year's iteration. As identified by the literature review in Section 3, the robot contains a LiDAR scanner which presents the possibility to implement a real time LiDAR-based SLAM algorithm. This functionality is yet to be implemented on the current prototype.

5.2.1 Obstacle Detection & Avoidance

The robot currently has a peripheral infrared (IR) sensor for detecting sharp extrusions on the cave ceiling, known as Stalactites (NPS 2015). This information is sent to a connected laptop so that the operator can adjust the robot's position accordingly. However, no autonomous function exists for the robot to adjust its path based on the data from the IR sensors. Autonomous obstacle detection and avoidance is most beneficial as it would allow for the robot to adjust its own path without the need for user intervention. This extends the mission capability of the robot, allowing it to venture into cave sections beyond what the user can observe directly. The data from the LiDAR sensor can also be used in conjunction with the IR sensor to achieve this capability. Other obstacles including rocks and sharp ascents or descents can also be detected with this method such that the robot can autonomously alter its gait accordingly. Another key limitation of the current prototype is its ability to maintain stability when a sharp descent or hole in the ground is encountered. Obstacles of this nature must also be detected to allow the robot to adjust its gait for stable and efficient traversal.

5.2.2 Pathfinding

Due to the lack of mapping and localisation the prototype has no ability to plan a path through the map of the cave environment. The robot must also have an ability to recognise sections of the cave which have already been explored in the map it is producing. This will ensure that the robot is producing a complete map of the area.

5.3 Dynamic Gait Optimisation

The current prototype is equipped with three different walking gaits. Each gait has different characteristics such as power consumption, movement speed, and stability. As a result, certain

gaits are better suited for different terrains (Bright et al. 2022). Additionally, incline and decline considerations further complicate the system's dynamics. Currently, the gaits are selected manually using a controller and there is no closed-loop feedback system to monitor its performance with a particular gait. The robot is, therefore, limited in the terrains it can traverse. During the preliminary testing of the robot, highlighted in Section 4.6, it was also observed that the robot legs were slipping along the acrylic sheet used on the incline testing rig. This significantly reduced the robot's stability and movement across the rig's surface. As the robot's intended operating environment is a cave, it is expected that some surfaces the robot must traverse will be characterised by limited friction and, therefore, a risk of slipping. A gait optimisation algorithm which dynamically responds to surface characteristics is liable to improve the robot's ability to traverse sub-optimal terrains.

5.4 Computational Power

The current onboard processor for the robot is the NVIDIA Jetson Nano. This processor is more powerful and AI-capable than the secondary Arduino Uno processor. The robot aims to use large data streams from the LiDAR scanner to conduct SLAM in real time. This means that the number of floating point operations per second (FLOPS) is a key parameter for performance. An uncertainty exists whether the Jetson Nano will have sufficient computational power to conduct the SLAM algorithms required for autonomous navigation. Potential processing power upgrades include the Jetson Orin Nano 8GB, Jetson Orin NX 8GB, or Jetson Orin NX 16G.

5.5 Development & Debugging

Deploying software onto the robot in its current state is an inefficient and convoluted process. To access the graphical user interface (GUI) of the Ubuntu operating system installed on the Jetson processor, the Jetson must be connected to a monitor. Once this connection is made software can be pulled from the relevant repository and built on the robot. This means that small changes in software cannot be deployed onto the robot in quick succession which limits the development process' efficiency. A possible objective to improve this process is to utilise a build software on the Jetson that facilitates continuous integration (CI) and deployment (CD) from software located on a repository with the code base stored on the machine. This will allow for changes in the code on the repository to be incorporated onto the robot without the need to manually pull and build the new code. Debugging code deployed on the robot is also a significant challenge due to the lack of a wireless communication system. One particular communication system that could be utilised is the DroneDeploy software which can facilitate data communication between the robot and a connected device over a network. This software can facilitate a visualisation of real time mapping, related to OB5, which allows bugs in software to be identified by the quality of the map produced. Having such a software will greatly speed up the debugging process rather than having to post process the LiDAR data after each run to inspect map quality.

5.6 Summary

The prototype issues mentioned in this section present key opportunities to improve the functionality of the robot. These limitations form the basis of the majority of this iteration's technical objectives. Improving the development and debugging process is not a technical objective of the project but will allow for a streamlined software development process to occur and will benefit future project iterations significantly.

6 Preliminary Design

As the 2023 iteration of the project primarily involves introducing autonomous functionality to the 2022 prototype, the design scope is largely limited to software improvements. Hardware changes only occur where they are necessary to facilitate the desired software implementation. As per this year's project objectives, the preliminary design work initially involves the selection of algorithms to support autonomous functionality. F-LOAM will serve as an accurate and computationally-efficient SLAM algorithm (OB1) in addition to underpinning the obstacle avoidance and path planning functionality described in OB4 and OB2, respectively. Further support of OB4 and OB2 will be provided through implementing a point cloud-based obstacle detection algorithm. In pursuit of autonomous navigation, a closed-loop terrain compensation algorithm will be implemented such that it integrates with the existing OpenSHC system to improve the prototype's traversal capabilities in a cave environment (OB4). Additionally, a real time system monitoring technology will be implemented to make the development and debugging pattern more efficient (OB5). These improvements are designed such that they facilitate the accomplishment of the project objectives as well as future iterations of the project.

6.1 Mapping & Localisation

Endowing the prototype with the ability to perform real time mapping and localisation is primarily a software-related process due to the robot's existing LiDAR sensor permitting implementation of SLAM. However, while SLAM can be performed solely using LiDAR data, it typically leads to non-negligible cumulative errors (Guo et al. 2021). Therefore, additional data is required to negate or minimise such errors. One way to achieve this is to implement an IMU. Concurrent analysis of LiDAR and IMU data allows SLAM accuracy to be maintained over the course of a mission. In particular, this sensor fusion technique permits loop closure, an error-reducing strategy wherein the robot is detected to have returned to a position it was at earlier, to be performed more easily and accurately. This technique to minimises cumulative map drift (Karam, Lehtola & Vosselman 2021).

The primary means of localisation is the implementation of a GPU-accelerated version of F-LOAM. F-LOAM determines the transform between two LiDAR scans via feature matching as it is more computationally-efficient than the classical Iterative Closest Point (ICP) method. Specifically, by extracting surfaces and edges, the number of points to consider in transform calculations is significantly reduced and therefore reduces run time (Wang et al. 2021).

A nine degree of freedom (DOF) IMU continuously provides discrete linear acceleration, angular velocity, and magnetic field data in three axes each (Yadav & Bleakley 2016). By taking the robot's starting position to be the origin of a fixed reference frame for both the LiDAR and IMU, the robot's pose can be continuously calculated for both in the same global coordinate system. Over time, these two representations of the robot's pose are liable to diverge due to unique measurement errors inherent to the sensors or algorithm (Guo et al. 2021; Yadav & Bleakley 2016). Since the data from an IMU is typically non-linear and numerically differentiable, the Extended Kalman Filter (EKF) can be applied to minimise error in the integration of the data (Wang & Li 2021). By maintaining a history of the robot's positions throughout a mission, the robot's current position can be continuously checked against the history for matches and the LiDAR data can be checked against it to correct error through loop closure. However, to ensure efficient matching the data must be structured appropriately. The k-dimensional tree (KD tree)

is a data structure which effectively stores spatial data for performant search (Bereczky et al. 2016). This will permit efficient integration with the LiDAR localisation algorithm.

As GPUs are designed for parallel computing, GPU acceleration can improve performance of parts of the F-LOAM algorithm where an identical arithmetic process is applied to or using a large number of points from the LiDAR point cloud (Xiao et al. 2022). Thus, the parts of the algorithm which are liable to realise performance gains through GPU acceleration are feature extraction, voxelisation, and cost factor calculation. As the robot’s processing occurs on the NVIDIA Jetson platform, NVIDIA’s GPU computing framework, Compute Unified Device Architecture (CUDA), will be used to introduce the aforementioned parallel processing (NVIDIA n.d.a).

6.2 Navigation

Providing an autonomous robotic system with navigational abilities involves overcoming three challenging tasks; implementing obstacle detection, obstacle avoidance, and path planning. Given a non-uniform and obstacle-laden terrain, obstacle detection underpins both avoidance and path planning. Obstacle avoidance and path planning have considerable overlap in purpose, but path planning aims to look further into the future and thus relies on short-term obstacle avoidance to do so.

6.2.1 Obstacle Detection & Avoidance

Given that the robot has a LiDAR sensor but cannot have a camera due to its operating environment, obstacle detection will be conducted solely using the LiDAR data. The selected obstacle detection algorithm uses point cloud pre-processing, segmentation, IQR gating, and RANSAC-based plane fitting of LiDAR data to identify obstacles (Asvadi et al. 2016). Since the pre-processing involves voxelisation and point cloud registration, two tasks which have already occurred for SLAM, the processing step can be skipped. The segmentation step involves slicing, gating, plane fitting, and validation. Slicing, the process of dividing the point cloud into region-based segments to facilitate future analysis, requires the consideration of a large number of LiDAR points and thus can benefit from GPU acceleration. Gating, however, relies on finding medians, a process which is not suited to parallel computing. Similarly, RANSAC plane fitting is better suited to serial computation on the CPU as it is a light and low-throughput operation, meaning the additional time required to copy relevant data to the GPU is detrimental to performance (Seekhao et al. 2019). Therefore, in the interest of optimising performance, slicing will be conducted as part of the SLAM analysis before the data is transferred back to the Jetson’s memory for further processing. For further optimisation, only points in a 150° horizontal FOV in front of the robot’s direction of motion will be analysed for objects as static objects outside of this range are unlikely to impede the robot’s operation. This can be realised by comparing a point’s position in the robot’s coordinate system to the robot’s velocity vector. As the final part of ground segmentation, the validation process is conducted on the CPU as it is very computationally-light. Next, on-ground obstacles are segmented from the remaining points, the obstacle point set is voxelised using a voxel grid, and dynamic obstacles are segmented from static ones. Ground/on-ground separation is conducted by determining the distance of an arbitrary point in a slice from the ground plane and comparing it to a maximum allowable value. Dynamic-static obstacle segmentation can be achieved by determining if an obstacle is mapped to different voxels over time. This is accomplished by comparing the voxels

in the broader map to the voxels in the obstacle voxel set. Finally, A two-dimensional (2D) counter and log-likelihood ratio are then applied to develop binary masks which assign voxels a label of stationary or moving. The 2D counter can be achieved by looping through and summing the relevant voxels on the CPU. Since the log-likelihood ratio depends on both the 2D counters, this will be calculated immediately after determination of the counters. The binary masks then compare the log-likelihood ratio to two arbitrary thresholds, one for static obstacles and one for dynamic obstacles, to label voxels accordingly (Asvadi et al. 2016). These thresholds, T_d and T_s , will be determined by iterative experimentation.

6.2.2 Pathfinding

Based on the results of the object detection algorithm, pathfinding will occur using the artificial potential field method. To do this, the obstacles will be considered in the robot's coordinate system. Through the artificial potential field method, the ideal velocity vector can be continuously determined as the robot moves through its environment. Only the components of the velocity vector which are in the horizontal plane will be considered as the dynamic gait algorithm will automatically adjust the robot's orientation for changes in the vertical direction. After calculation, the velocity vector will be communicated to the control system through the /syropod_remote/desired_velocity ROS topic using the geometry_msgs::Twist message type. Once the robot has avoided an obstacle, it will be removed from the obstacle list to reduce computational load.

6.3 Dynamic Gait Optimisation

The selected dynamic gait algorithm relies only on feedback from the Dynamixel motors and locomotion state information (Zangrandi, Arrigoni & Braghin 2021). Since the motors are capable of providing feedback on the current draw, which can be converted to motor torque, and the locomotion state is stored by OpenSHC, no hardware changes are required to implement the algorithm (Robotis 2023a; Robotis 2023b; Tam et al. 2020). Currently, OpenSHC's walk and pose controllers generate the leg movement by calculating the desired joint positions (Tam et al. 2020). As the selected dynamic gait algorithm only deals with modifying the gait according to feedback, it will be integrated into the existing OpenSHC code base. The gait to be modified is the tripod gait as this provides the best performance and is, therefore, the ideal candidate for serving as a baseline. To introduce the new gait algorithm, the walk and pose controllers will be modified. The kinematic model built into OpenSHC renders that aspect of implementation simpler as the new functionality can piggy back off it, but the complexity of the code base will make it difficult to determine exactly what to modify. An iterative development and testing cycle will be required to successfully implement the algorithm.

6.4 Computational Power

The new functionality introduced in this iteration of the CaveX project is almost entirely software-based. Additionally, it relies on heavy processing of point cloud data. To accommodate such changes, the current NVIDIA Jetson Nano will be upgraded to an NVIDIA Jetson Orin NX 8GB. The data on the existing Nano will be transferred to the Orin NX prior to installation. To avoid a change in the robot's mass distribution, the Orin NX will be installed in the same position as the Nano. However, the Orin NX comes with a fan which the Nano does not have and, therefore, requires additional vertical space to fit and provide sufficient airflow for cooling. While data on the height of the Orin NX combined with its carrier board is not available, it is

assumed that it matches the height of the Nano with its carrier board at 30.21mm (Seeed Studio n.d.; NVIDIA n.d.b). The additional fan adds up to 18mm and thus may necessitate an increase in the distance between the robot's middle and top layer of carbon fibre (DigiKey n.d.). In this case, the stand off screws which connect the three layers would need to be adjusted accordingly.

6.5 Development & Debugging

As mentioned in Section 5.5, the process of deploying software onto the robot is currently inefficient. To improve this process, the team will use the Jenkins open-source automation server which is compatible with the Ubuntu operating system installed on the Jetson Nano. This software will allow for code on the CaveX GitHub repository to be continuously integrated and deployed on the robot code base whenever changes are made to the repository. The JenkinsFile which sets up the configuration of the Jenkins software can be manipulated to only trigger an automatic build process when changes to the main branch are made on the repository's main branch. This will prevent undesired builds and eliminate the possibility of code conflicts. Enabling an efficient development and debugging experience also requires real time interaction with the system. Due to bit rate and connectivity limitations of non-WiFi wireless communications technologies, the detailed data required for development and debugging cannot be transmitted in a connectionless cave environment. However, as most development and debugging occurs in a connected environment, DroneDeploy can be installed onto the system with Rocos for ROS integration (DroneDeploy n.d.a; DroneDeploy n.d.b). To achieve this, the WiFi module must first be connected to the Jetson carrier board. The DroneDeploy ROS agent can then be installed onto the robot and configured to stream data to a device. DroneDeploy with Rocos and the ROS agent provide features such as topic monitoring, topic exploration, bandwidth monitoring, point cloud rendering, data dashboards, and data streaming (DroneDeploy n.d.b; DroneDeploy 2022). Where unavailable features are required, they can be introduced using a custom web server and the Rocos SDK or the ROS Node.js package (DroneDeploy n.d.c; ROS Wiki 2017).

6.6 Summary

Overall, with careful considerations to programmatic design, the various algorithms which are to endow the robot with autonomy can be implemented into the existing system. The SLAM algorithm, F-LOAM, is likely to be the most computationally expensive aspect of the system but it underpins most of the later autonomous decision making. Since the processing for SLAM partially overlaps with the processing for obstacle detection, the two processes can be merged for increased performance. Additionally, obstacle detection can be optimised by only considering points in the region ahead of the robot's direction of movement. The gait algorithm can be optimised by integrating it into OpenSHC's existing code base. Pathfinding and obstacle avoidance will be achieved using the artificial potential field method and build on the SLAM and obstacle detection data. To facilitate these systems, a new NVIDIA Jetson Orin NX will be installed along side a real time monitoring system for rapid development and debugging. These improvements also provide a strong basis for future iterations of the CaveX project.

7 Completion Plan

In this section, a completion plan outlines all of the remaining work to be completed by the end of the project to achieve the technical objectives. The completion plan aligns with the schedule presented in the sprint plan shown in Appendix A.11 and the Gantt chart shown in Appendix A.10. The target completion dates for each objective are included in Table 1.1.

7.1 Mapping & Localisation

Real time mapping and localisation is the first objective to be completed and is targeted to be done by the end of the third sprint on 18th June. Implementing a mapping and localisation capability necessitates hardware upgrades due to the drastic increase in required processing power for real time point cloud processing. A suitable option is to replace the Jetson Nano primary controller with a Jetson Orin NX 8GB which already in the process of being purchased. Additionally, to reduce cumulative errors involved with only using LiDAR data for SLAM, installing an IMU on the robot will be investigated. The team will begin integrating the F-LOAM open-source package with the data from the LiDAR and subsequently determine the robot's ability to perform SLAM. A specific focus of using F-LOAM is to accelerate the computational speed using the Jetson's powerful GPU. After SLAM is implemented, the robot will be tested in a controlled environment and the point cloud maps will be analysed to determine if they accurately represent the environment. To assess the algorithm's accuracy, its ARE and ATE will be determined.

7.2 Autonomous Navigation

Autonomous navigation encapsulates path planning as well as obstacle detection and avoidance. Due to a closely-coupled relationship between path planning and obstacle avoidance, these objectives will be developed simultaneously during the fourth and fifth sprint.

The first steps in obstacle detection rely on LiDAR point cloud preprocessing which is done by the SLAM algorithm. Thus, the obstacle detection algorithm will be implemented after SLAM implementation. Throughout the obstacle detection development process, iterative experimentation will be conducted in a controlled test environment with obstacles. This is to ensure that no premature optimisations excessively limit the algorithm's effectiveness. Following implementation and optimisation of the algorithm, true positive and false negative detection rates will be determined in the Naracoorte Caves. Much like obstacle detection is dependent on SLAM, pathfinding is dependent on obstacle detection. During implementation of the pathfinding algorithm, iterative testing will occur to identify changes which negatively impact its success rate. OB4 and OB2 will be considered completed when the prototype successfully navigates an obstacle-laden test course.

7.3 Dynamic Gait Optimisation

The implementation of a dynamic gait optimisation does not initially necessitate any additional hardware and will be implemented by setting up ROS topics for messages to be published about the motor's current draw. A control algorithm will subscribe to this topic and relate the current draw of each motor to its torque. Furthermore, a correlation between terrain and motor torque will be determined by testing the robot on a range of different surfaces and observing the motor torque. Ideally the control algorithm will automatically switch gaits depending on the torque

experienced by each motor. If a torque level becomes too high it will indicate that the gait should be changed and if a drop in torque is observed then the gait change was successful. This control algorithm will be integrated with the existing OpenSHC package currently used by the robot.

7.4 Development & Debugging

The key feature that will improve the software debugging process is the ability to transmit data to and from the robot during its operation. This capability will allow for the user to get a real time display of the map as it is produced by the LiDAR scanner (OB5). The capability will not be able to be used inside the cave environment due to signal limitations in such an environment, hence it serves purely to improve software development. The DroneDeploy software will be utilised to fulfill this capability. The DroneDeploy software allows for data communication and digital mapping through cloud-based software (Capterra 2023). A DroneDeploy license is in the process of being obtained for use by the 2023 CaveX team. To allow for this software to be used, the WiFi module shown in Appendix B.8 will be installed into the Jetson's carrier board. This module will facilitate network communication between the robot and a connected computer.

7.5 Summary

The progress of each technical objective is summarised in Figure 8. The work needed to implement each of the technical objectives has been extensively researched and appropriate algorithms have been selected for specific objectives. Hence, progress has been made towards every one of the objectives. Localisation and mapping (OB1) has slightly more progress as procurement of new hardware has begun which represents a major milestone in achieving this objective. More progress has also been made on a gait control system (OB3) through preliminary testing that has been undertaken to identify gait effectiveness for inclined and declined surfaces.

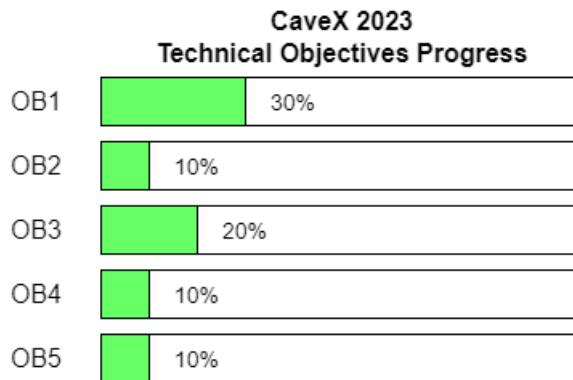


Figure 8: CaveX 2023 technical objectives progress

8 Conclusion

This report presented the current state of the 2023 CaveX project iteration. The technical objectives were outlined in Section 1.4 and form the fundamental basis for this iteration's project scope. To ensure that the technical objectives are met, a systems engineering methodology has been used. This approach facilitates the ability for the final design to fulfill the user's vision by constructing user needs and system requirements. The usage of an agile engineering approach in combination with the systems engineering methodology has also been outlined to improve the flexibility in project schedule.

The key theme of the 2023 CaveX project is to implement the current robot prototype with autonomous capabilities. The literature review in Section 3 discusses the current state of autonomous algorithms for robotic control, which is a key milestone for achieving autonomous functionality. Table 3.1 shows a summarised comparison between the findings from multiple sources in regards to themes which relate to the technical objectives. From this comparison, outcomes have been highlighted which direct the implementation specifics to meet the technical objectives.

A complete problem definition has been presented in Section 4. The information in this section highlights the systems engineering methodology and applies a stakeholder analysis to formulate a set of user needs. These user needs were translated into a set of system requirements which fulfill these needs. An analysis of the existing prototype is also presented. Initially, a theoretical analysis of the prototype exposed technical limitations. Subsequently, preliminary incline and decline testing was undertaken in the EXTERRES laboratory at The University of Adelaide. From this testing, the strengths and limitations of the gaits were highlighted. An analysis on the key limitations of the current prototype and identification of key prototype issues is displayed in Section 5. These limitations are consistent with the technical objectives of the project. A preliminary design has been outlined in Section 6 which provides the rationale for technical decisions to mitigate these limitations. Lastly, a completion plan has been outlined which illustrates the future work required to meet the objectives' deadlines shown in Table 1.1 and Appendix A.10. The current progress on the technical objectives is visually displayed in Figure 8. The agile development approach, elaborated on in Section 2.2, has allowed for progress to be made on each of the objectives in parallel rather than serially. This approach will allow the objectives to be met to the desired standards highlighted in Appendix A.13 by their deadlines.

References

- Abdelaziz, N & El-Rabbany, A 2022, 'An Integrated INS/LiDAR SLAM Navigation System for GNSS-Challenging Environments', *Sensors (Basel)*, vol. 22, no. 12, DOI:10.3390/s22124327
- Agheli, M & Nestinger, SS 2012, 'Study of the Foot Force Stability Margin for multi-legged/wheeled robots under dynamic situations', *Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, IEEE/ASME, Suzhou, China, pp. 99-104.
- Allouis, E, Blake, R, Gunes-Lasnet, S, Jorden, T, Maddison, B, Schroeven-Deceuninck, H, Stuttard, M, Truss, P, Ward, K, Ward, R, Woods, M 2013, 'A Facility for the Verification & Validation of Robotics & Autonomy for Planetary Exploration'
- Anand, B, Patil, AG, Senapati, M, Barsaiyan, V & Rajalakshmi, P 2020, 'Comparative Run Time Analysis of LiDAR Point Cloud Processing with GPU and CPU', in *2020 IEEE International Conference on Computing, Power and Communication Technologies*, IEEE, Greater Noida, India, pp. 650-654.
- Asvadi, A, Premevida, C, Peixoto, P & Nunes, U 2016, '3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes', *Robotics and Autonomous Systems*, vol. 83, pp. 299-311, DOI:10.1016/j.robot.2016.06.007.
- Atlassian n.d., *The Agile Coach*, Atlassian, viewed 19 April 2023,
<<https://www.atlassian.com/agile>>.
- Auat Cheein, FA, Lopez, N, Soria, CM, di Sciascio, FA, Lobo Pereira, FL & Carelli, R 2010, 'SLAM algorithm applied to robotics assistance for navigation in unknown environments', *Journal of NeuroEngineering and Rehabilitation*, vol. 7, DOI:10.1186/1743-0003-7-10.
- Badue, C, Guidolini, R, Carneiro, RV, Azevedo, P, Cardoso, VB, Forechi, A, Jesus, L, Berriel, R, Paixão, TM, Mutz, F, Veronese, L, Oliveira-Santos, T & De Souza, AF 2021, 'Self-driving cars: A survey', *Expert Systems with Applications*, vol. 165, DOI:10.1016/j.eswa.2020.113816.
- Bartsch, S, Birnschein, T, Langosz, M, Hilljegerdes, J, Kuehn, D & Kirchner, F 2012, 'Development of the Six-Legged Walking and Climbing SpaceClimber', *Journal of Field Robotics*, vol. 29, no. 3, pp. 506-532, DOI:10.1002/rob.21418.
- Bereczky, N, Duch, A, Németh, K & Roura, S 2016, 'Quad-kd trees: A general framework for kd trees and quad trees', *Theoretical Computer Science*, vol. 616, pp. 126-140.
- Bichuette, ME & Trajano, E 2018, 'Diversity of Potamolithus (Littorinimorpha, Truncatelloidea) in a high-diversity spot for troglobites in southeastern Brazil: role of habitat fragmentation in the origin of subterranean fauna, and conservation status', *Subterranean Biology*, vol. 25, pp. 61-88, DOI:10.3897/subtbiol.25.23778.
- Bright, H, Mahony, T, Verboon, N & Zilm, L 2022, 'CaveX Final Report', Honours thesis, The University of Adelaide.
- Capterra 2023, GIS Software DroneDeploy, viewed 28 May 2023,
<<https://www.capterra.com.au/software/197016/drondedeploy>>.
- Chen, Z, Liu, J & Gao, F 2022, 'Real-time gait planning method for six-legged robots to optimize the performances of terrain adaptability and walking speed', *Mechanism and Machine Theory*,

vol. 168, DOI:10.1016/j.mechmachtheory.2021.104545.

Cooper, J, King, M, Natt, C, Lee, H & Thomas, A 2021, 'CaveX Honours Final Report', Honours thesis, The University of Adelaide.

CSIRO Robotics 2020, *csiro-robotics/syropod_remote*, GitHub, viewed 28 May 2023, <https://github.com/csiro-robotics/syropod_remote/tree/master>.

CSIRO Robotics 2021, *csiro-robotics/syropod_highlevel_controller: OpenSHC: A Versatile Multi-legged Robot Controller*, GitHub, viewed 28 May 2023, <https://github.com/csiro-robotics/syropod_highlevel_controller/tree/master>.

DigiKey n.d., *114992687 Seeed Technology Co., Ltd | Fans, Thermal Management | DigiKey*, DigiKey, viewed 26 May 2023, <<https://www.digikey.com.au/en/products/detail/seeed-technology-co-ltd/114992687/14552876>>.

DroneDeploy 2022, *3D Point Clouds - DroneDeploy*, DroneDeploy, viewed 27 May 2023, <<https://help.dronedeploy.com/hc/en-us/articles/1500004860641-3D-Point-Clouds>>.

DroneDeploy n.d.a, *ROS - Docs*, DroneDeploy, viewed 27 May 2023, <<https://docs-automate.dronedeploy.com/docs/ros>>.

DroneDeploy n.d.b, *Key features - Docs*, DroneDeploy, viewed 27 May 2023, <<https://docs-automate.dronedeploy.com/docs/rocos-for-ros-robots>>.

DroneDeploy n.d.c, *Examples - Docs*, DroneDeploy, viewed 27 May 2023, <<https://docs-automate.dronedeploy.com/docs/examples>>.

Dualsky 2022, *Xpower HED series*, Dualsky, viewed 21 May 2023. <http://en.dualsky.com/Xpower_LiPos/64986f2b-c65ed53c- 6f56-99e22ad8c108.shtml>

Dubowsky, S, Iagnemma, K, Liberatore, S, Lambeth, DM, Plante, JS & Boston, PJ 2005, 'A Concept Mission: Microbots for Large-Scale Planetary Surface and Subsurface Exploration', in *AIP Conference Proceedings*, AIP, Canberra, Australia, pp. 1449-1458.

Guo, M, Zhang, L, Liu, X, Du, Z, Song, J, Liu, M, Wu, X & Huo, X 2021, '3D Lidar SLAM Based on Ground Segmentation and Scan Context Loop Detection', in *11th IEEE Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, IEEE, Jiaxing, China, pp. 692-697.

Jin, J, Zhang, Y, Zhou, Z, Jin, M, Yang, X & Hu, F 2023, 'Conflict-based search with D* lite algorithm for robot path planning in unknown dynamic environments', *Computers and Electrical Engineering*, vol. 105, DOI:10.1016/j.compeleceng.2022.108473.

Kambesis, P 2009, 'The importance of cave exploration to scientific research', *Journal of Cave and Karst Studies*, vol. 69, no. 1, pp. 46-58.

Karam, S, Lehtola, V & Vosselman, G 2021, 'Simple loop closing for continuous 6DOF LiDAR&IMU graph SLAM with planar features for indoor environments', *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 181, pp. 413-426, DOI:10.1016/j.isprsjprs.2021.09.020.

Khatib, O 1986, 'Real-Time Obstacle Avoidance for Manipulators and Mobile Robots', *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98.

Kim, P, Chen, J, Kim, J & Cho, Y 2018, 'SLAM-Driven Intelligent Autonomous Mobile Robot

Navigation for Construction Applications', in *25th Workshop of the European Group for Intelligent Computing in Engineering*, EG-ICE, Lausanne, Switzerland, pp. 254-269.

LaValle, SM & Kuffner, JJ 2001, 'Randomized Kynodynamic Planning', *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378-400.

Le, J, Pengcheng, Z, Wei, D, Jiayuan, L, Mingyao, A, Xuan, W & Qingwu, H 2018, 'An Eight-Direction Scanning Detection Algorithm for the Mapping Robot Pathfinding in Unknown Indoor Environment', *Sensors*, vol. 18, no. 12, DOI:10.3390/s18124254.

Lele, AS, Fang, Y, Ting, J & Raychowdhury, A 2020, 'Learning to Walk: Spike Based Reinforcement Learning for Hexapod Robot Central Pattern Generation', *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, IEEE, Genova, Italy, pp. 208-212.

Lewis, I 2019, 'South Australian geology and the State Heritage Register: an example of geo-conservation of the Naracoorte Caves complex and karst environment', *Australian Journal of Earth Sciences*, vol. 66, no. 6, pp. 785–792.

Mur-Artal, R, Montiel, JMM & Tardós, JD 2015, 'ORB-SLAM: A Versatile and Accurate Monocular SLAM System', *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, DOI:10.1109/TRO.2015.2463671.

National Park Service (NPS) 2015, *How Stalactites and Stalagmites Form*, U.S. Department of the Interior, viewed 21 May 2023,
<<https://www.nps.gov/ozar/learn/education/speleothems.htm>>.

Novak, R, Hrovat, A, Bedform, MD & Javornik, T 2021, 'Geometric Simplifications of Natural Caves in Ray-Tracing-Based Propagation Modelling', *Electronics*, vol. 10, no. 23, DOI:10.3390/embedded-syst10232914.

NVIDIA n.d.a, *Jetson Developer Tools | NVIDIA Developer*, NVIDIA Developer, viewed 26 May 2023, <<https://developer.nvidia.com/embedded/develop/tools>>.

NVIDIA n.d.b, , NVIDIA, viewed 26 May 2023, <<https://www.nvidia.com/en-au/autonomous-machines/embedded-systems/jetson-nano-developer-kit/>>.

Qi, CR, Su, H, Mo, K & Guibas, LJ 2017, 'PointNet: Learning on Point Sets for 3D Classification and Segmentation', in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, Hawaii, pp. 652-660.

Qin, T & Cao, S 2019, *GitHub - HKUST-Aerial-Robotics/A-LOAM: Advanced implementation of LOAM*, GitHub, viewed 25 March 2023, <<https://github.com/HKUST-Aerial-Robotics/A-LOAM>>.

Robotis 2023a, *2XC430-W250*, Robotis, viewed 26 May 2023,
<<https://emanual.robotis.com/docs/en/dxl/x/2xc430-w250/>>.

Robotis 2023b, *XC430-W240-T*, Robotis, viewed 26 May 2023,
<<https://emanual.robotis.com/docs/en/dxl/x/xc430-w240/>>.

ROS Wiki 2017, *rosnodejs - ROS Wiki*, ROS Wiki, viewed 27 May 2023,
<<https://wiki.ros.org/rosnodejs>>.

ROS Wiki 2019a, *Packages - ROS Wiki*, ROS Wiki, viewed 28 May 2023,
<<https://wiki.ros.org/Packages>>.

ROS Wiki 2019b, *Topics - ROS Wiki*, ROS Wiki, viewed 28 May 2023,
<<http://wiki.ros.org/Topics>>.

ROS Wiki 2022, *ROS/Tutorials/UnderstandingNodes - ROS Wiki*, ROS Wiki, viewed 28 May 2023,
<<https://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>>.

Seeed Studio n.d., *J401-datasheet.pdf*, Seeed Studio, viewed 26 May 2023,
<<https://files.seeedstudio.com/wiki/reComputer-J4012/Carrier-Board-J401/J401-datasheet.pdf>>.

Seekhao, N, Yu, G, Yuen, S, JaJa, J, Mongeau, L & Li-Jessen NYK 2019, 'High-Performance Host-Device Scheduling and Data-Transfer Minimization Techniques for Visualization of 3D Agent-Based Wound Healing Applications', *PDPTA*, vol. 19, pp. 69-76.

Shan, T & Englot, B 2018, 'LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain', in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Madrid, Spain, pp. 4758-4765.

Shang, E, An, X, Wu, T, Hu, T, Yuan, Q & He, H 2015, 'LiDAR Based Negative Obstacle Detection for Field Autonomous Land Vehicles', *Journal of Field Robotics*, vol. 33, no. 5, pp. 591-617, DOI:10.1002/rob.21609.

Tam, B, Talbot, F, Steindl, R, Elfes, A & Kottege, N 2020, 'OpenSHC: A Versatile Multilegged Robot Controller', *IEEE Access*, vol. 8, pp. 188908-188926, DOI:10.1109/ACCESS.2020.3031019.

Velodyne 2019, *VLP-16 User Manual*, Velodyne LiDAR, viewed 28 May 2023,
<<https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf>>.

Wang, H, Wang, C, Chen, L & Xie, L 2021, 'F-LOAM: Fast LiDAR Odometry and Mapping', in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Prague, Czech Republic, pp. 4390-4396.

Wang, Y & Li, X 2021, 'An improved robust EKF algorithm based on sigma points for UWB and foot-mounted IMU fusion positioning', *Journal of Spatial Science*, vol. 66, no. 2, pp. 329-350, DOI:10.1080/14498596.2019.1632754.

Williams, C 2021, *Interview*, The University of Adelaide North Terrace Campus, 23 March.

Xiao, J, Ji, Q, Shen, Q, Jiang, J, Huang, Y & Lou, J 2022, 'Accelerating stochastic-based reliability estimation for combinational circuits at RTL using GPU parallel computing', *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8309-8326, DOI:10.1002/int.22940.

Yadav, N & Bleakley, C 2016, 'Fast calibration of a 9-DOF IMU using a 3 DOF position tracker and a semi-random motion sequence', *Measurement*, vol. 90, pp. 192-198.

Yan, Y, Mao, Y & Li, B 2018, 'SECOND: Sparsely Embedded Convolutional Detection', *Sensors*, vol. 18, no. 10, DOI:10.3390/s18103337.

Yang, W, Wang, G & Shen, Y 2020, 'Perception-Aware Path Finding and Following of Snake Robot in Unknown Environment', in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Las Vegas, USA, pp. 5925-5930.

8 CONCLUSION

Zangrandi, M, Arrigoni, S & Braghin, F 2021, *Control of a Hexapod Robot Considering Terrain Interaction*, arXiv, viewed 15 May 2023, <<https://arxiv.org/pdf/2112.10206.pdf>>.

Zhang, J & Singh, S 2014, 'LOAM: Lidar Odometry and Mapping in Real-time', in D Fox, LE Kavraki & H Kurniawati, *Robotics: Science and Systems*, Robotics Science and Systems, Berkeley, California, USA.

Zou, Q. Sun, Q, Chen, L, Nie, B & Li, Q 2022, A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles, *IEEE Transactions on Intelligent Transportation Systems*, vol 23, no. 7, pp. 6907-6921. DOI:10.1109/TITS.2021.3063477.

Appendices

A Project Management

A.1 SMART objective Justification

This section expands on the objectives listed in the “Technical objectives” section. It specifies the SMART (Specific, Measurable, Achievable, Relevant, and Time-bound) nature of the objectives.

Objective 1: Implement real time localisation and mapping capability

S	Implement the ability to localise the robot’s position within the 3D point cloud generated by the LiDAR sensor.
M	This capability will be complete when the map produced by the LiDAR sensor correctly resembles the testing environment. A 2D overlay image of the map can be analysed to the known testing environment map to address accuracy and completeness.
A	This capability is commonly achieved in robotics by utilising a SLAM algorithm, these algorithms are heavily supported and documented within ROS.
R	For the robot to autonomously navigate its way through the cave environment, it must be able to recognise its position within the cave point cloud map.
T	This is a key capability that will allow for autonomous behaviour to be built on top of, hence it is aimed to be completed by June 18 th . This objective however can still be completed in parallel with the other technical objectives in an agile approach

Objective 2: Design, Implement & Optimise a Pathfinding Algorithm

S	Implement an algorithm which can generate an ideal path based on LiDAR data, recent movement, and generated way points.
M	Objective is completed when the robot can successfully plan paths fully autonomously (no prior instruction) and semi-autonomously (between pre-defined waypoints).
A	This is achievable as there is a wealth of information on the internet regarding robotics control algorithms using LiDAR. Pathfinding algorithms already exist and can, therefore, form a basis for the robot’s system.
R	For the robot’s autonomous control algorithms to work, they require input from a pathfinding algorithm.
T	Implementing autonomy to the robot is the overarching goal of this year’s iteration of the CaveX project. However, testing will be required therefore, this objective should be completed with time left to conduct testing. Hence, it should be completed by September 10 th .

Objective 3: Implement terrain sensing and gait control system

S	Implement hardware and software which can sense the surface characteristics beneath each foot and adjust the gait of each leg accordingly.
M	Objective is completed when the robot's speed or stability, which can be measured using optical tracking, is improved using motor feedback or force sensing resistors.
A	This is achievable as an iterative test-based approach will allow for the development of a system which can account for ground characteristics.
R	To increase the robot's ability to traverse cave structures, it should adjust its gait according to the variety of surfaces it will traverse.
T	Ideally, this objective should be complete before OB2 to allow for optimal performance, the target date for completion is 13 th August. OB2 does not directly depend on this objective meaning agile development is conserved.

Objective 4: Implement obstacle detection and avoidance capability

S	Detect obstacles using peripheral sensors & adjust the robot's path to avoid them.
M	The robot can be tested with obstacles to determine if the path planning algorithm takes obstacles into consideration. The object is complete when the robot demonstrates its ability to avoid obstacles.
A	This objective is achievable as it is a problem in robotics which has been solved using a variety of different methods. There is a range of literature for using data from a LiDAR scanner in ROS to detect and avoid obstacles.
R	Obstacle detection and avoidance is a very important aspect of autonomous systems. Without obstacle avoidance the robot risks failing to effectively search a large cave area and potential collisions with undetected obstacles may cause damage to the robot's hardware.
T	This objective is planned to be completed by 10 th Sept. A precursor to obstacle avoidance is having a successful path planning algorithm in place which is outlined in objective 2, however the obstacle detection software can be done independently.

Objective 5: Implement Real-time Monitoring System

S	Implement a convenient interface for the user to send commands to the robot and receive sensor data remotely in real time.
M	Objective is completed when the robot can continuously send its state wirelessly to and receive commands from another device.
A	This is achievable as pre-made software exists for wireless communication with ROS-based robots and software can be written for communication via the internet.
R	For operators to ensure the robot is functioning correctly and for efficient software development, real-time two-way communication with the robot is necessary.
T	This objective has a completion date by 16 th July. If this objective is completed early, it will significantly help the team with testing and debugging.

A.2 Stakeholder Analysis

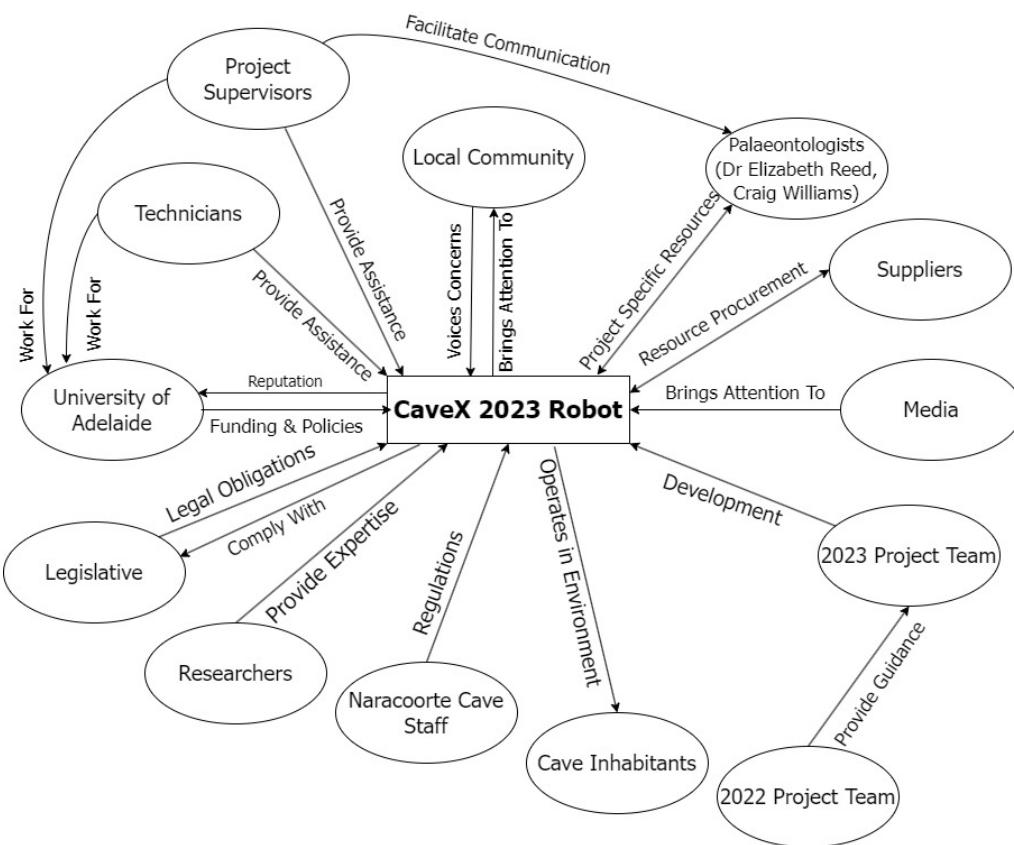


Figure A.1: Stakeholder map showing interactions between the project and those that have influence, or are influenced by the project.

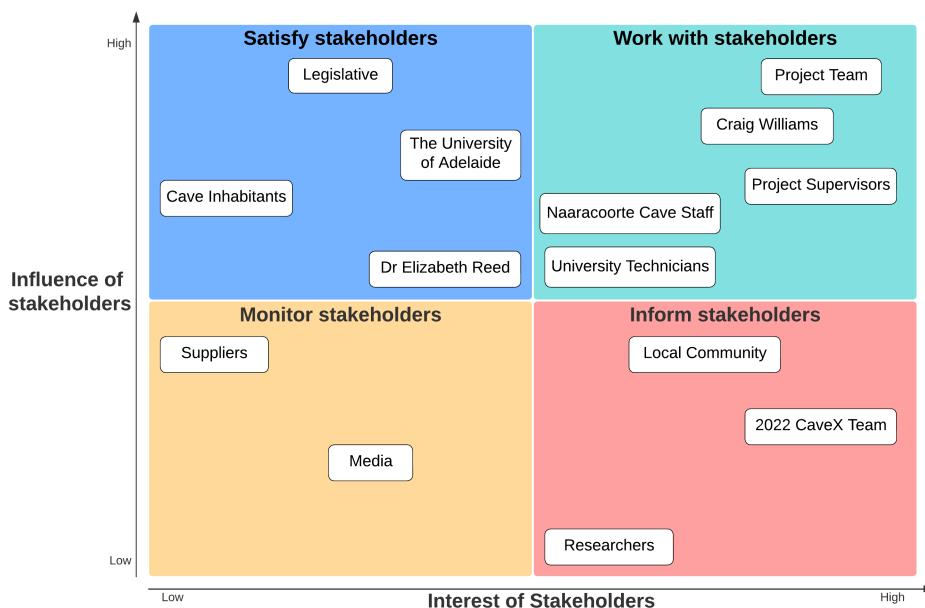


Figure A.2: Influence and interest grid of various stakeholders

A.3 Scenario Based Needs Analysis

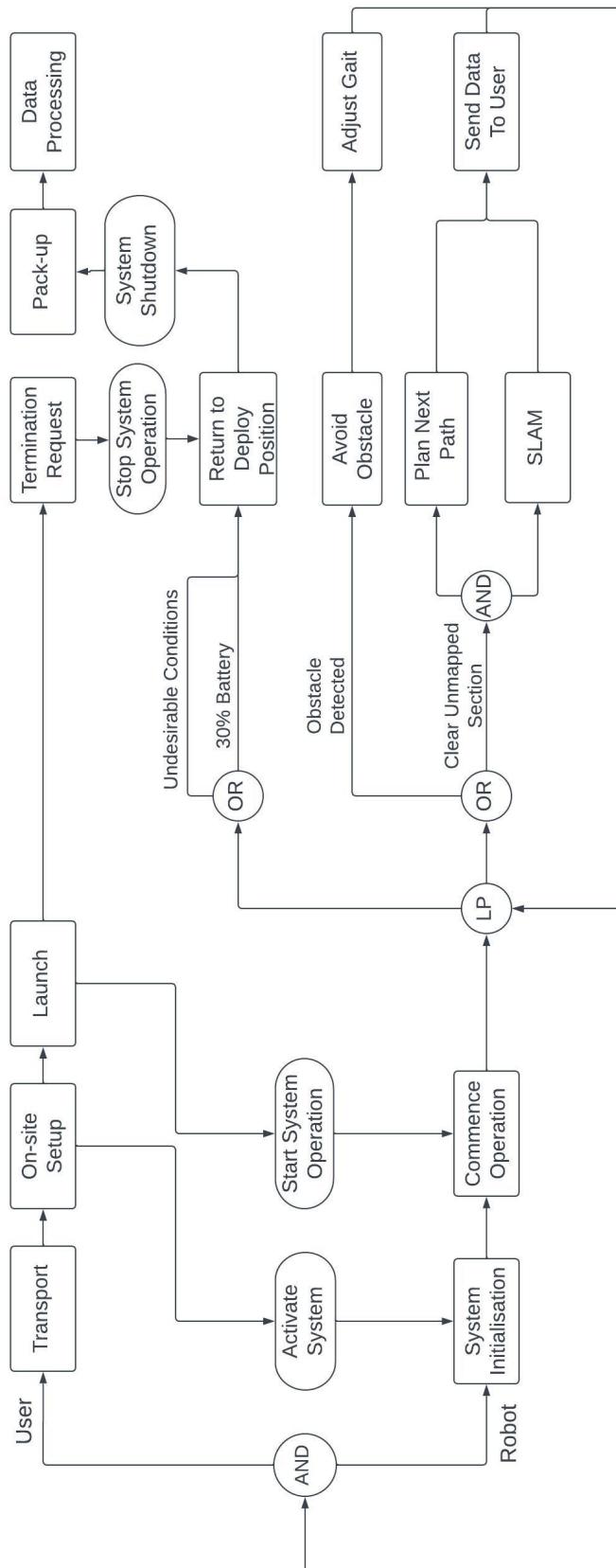


Figure A.3: Scenario-based User Needs Analysis

A.4 System Context Diagram

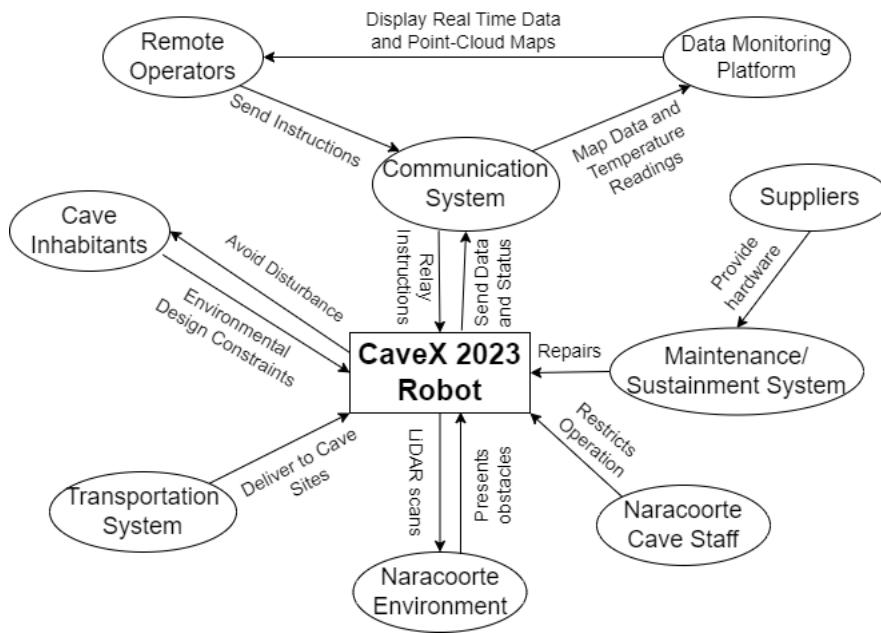


Figure A.4: System context diagram displaying interfaces with key external entities

A.5 CaveX 2021 Stakeholder Needs

#	Source	△ Name	Text	Owner
1	Craig Williams	<input checked="" type="checkbox"/> SN01 Portability	User needs system to include appropriate protective case for transportation purposes	<input type="checkbox"/> User
2	Craig Williams	<input checked="" type="checkbox"/> SN02 Carrying Weight	User needs system to be light enough for single user to carry into and out from the cave system	<input type="checkbox"/> User
3	Craig Williams	<input checked="" type="checkbox"/> SN03 Cave Conditions	User needs system to perform effectively in all cave environment conditions	<input type="checkbox"/> User
4	Craig Williams	<input checked="" type="checkbox"/> SN04 Size	User needs system to be of an appropriate size to map low, narrow spaces and cave ceiling alcoves	<input type="checkbox"/> User
5	Craig Williams	<input checked="" type="checkbox"/> SN05 Dynamic Mapping	User needs system to perform high spatial frequency scans	<input type="checkbox"/> User
6	Craig Williams	<input checked="" type="checkbox"/> SN06 Deployment	User needs system to be rapidly deployable from a backpack by a singular user	<input type="checkbox"/> User
7	Craig Williams	<input checked="" type="checkbox"/> SN07 Recovery	User needs a reliable method of recovering system within a short time frame	<input type="checkbox"/> User
8	Craig Williams	<input checked="" type="checkbox"/> SN08 Movement	User needs system to traverse cave floor	<input type="checkbox"/> User
9	Craig Williams	<input checked="" type="checkbox"/> SN09 Autonomy	User needs system to remain operational out of reach and line of sight of user	<input type="checkbox"/> User
10	Craig Williams	<input checked="" type="checkbox"/> SN10 Reusability	User needs to be able to use the system multiple times whilst underground	<input type="checkbox"/> User
11	Craig Williams	<input checked="" type="checkbox"/> SN11 Laptop Interface	The computing system shall be able to interface with a laptop to transfer and receive data	<input type="checkbox"/> User
12	Academic Supervisors	<input checked="" type="checkbox"/> SN12 Biomimpiration	Design team needs system movement mechanism to mimic that of a biological organism	<input type="checkbox"/> University of Adelaide and Design Team
13	Academic Supervisors and UoA	<input checked="" type="checkbox"/> SN13 Life Cycle	University of Adelaide needs system to be upgradeable for future years' projects and useful at the project's conclusion	<input type="checkbox"/> University of Adelaide and Design Team
14	Craig Williams	<input checked="" type="checkbox"/> SN14 Mapping	Researchers need system to create 3D maps of cave	<input type="checkbox"/> Researchers
15	Craig Williams	<input checked="" type="checkbox"/> SN15 Map Usability	Researchers need system maps to be extractable and of 3D point cloud format	<input type="checkbox"/> Researchers
16	Craig Williams	<input checked="" type="checkbox"/> SN16 Soundlessness	Bats need system to make minimal sound	<input type="checkbox"/> Cave Environment and Land Owner
17	Craig Williams and Liz Reed	<input checked="" type="checkbox"/> SN17 Environmental Impact	Cave land owner needs system to avoid touching cave walls and decorations	<input type="checkbox"/> Cave Environment and Land Owner
18	UNESCO and Liz Reed	<input checked="" type="checkbox"/> SN18 World Heritage Site Requirements	Cave land owner needs system to comply with cave-specific equipment impact restrictions	<input type="checkbox"/> Cave Environment and Land Owner
19	Craig Williams and Liz Reed	<input checked="" type="checkbox"/> SN19 Ground Impact	Cave land owner needs system to have minimal ground contact surface area	<input type="checkbox"/> Cave Environment and Land Owner
20	Craig Williams and UNESCO	<input checked="" type="checkbox"/> SN20 Delicate Area Avoidance	User and cave land owner needs system to avoid damaging delicate areas with known locations	<input type="checkbox"/> Cave Environment and Land Owner

Figure A.5: CaveX Stakeholder Needs Table - Needs gathered from stakeholder interviews and environmental constraints (Cooper et al. 2021).

A.6 CaveX 2021 System Requirements

#	Name	Text	Derived From	Validated by 2021 Prototype
1	REQ1 Storage Size	System shall be no larger than 40x30x20cm when in storage	SN01 Portability SN04 Size	Y
2	REQ2 Operation Noise	System shall be no louder than 60 dB during operation	SN16 Soundlessness SN03 Cave Conditions	Y
3	REQ3 Life Span	System shall have an overall functional life span of at least 3 years	SN10 Reuseability SN13 Life Cycle	N/A
4	REQ4 Weight	System shall have a total mass no larger than 10 kg	SN02 Carrying Weight SN13 Life Cycle	Y
5	REQ5 Modular	System shall be modular to allow for individual upgrades to the body, actuators and sensors	SN13 Life Cycle	Y
6	REQ6 Transport Damage	System shall remain undamaged during transportation in protective case and vehicles	SN01 Portability SN15 Map Usability	Y
7	REQ7 Map Generation	System shall create 3D point clouds in las format	SN04 Size	Y
8	REQ8 Operating Size	System shall be no higher than 50cm and no wider than 1m when operating	SN19 Ground Impact	Y
9	REQ9 Ground Surface Area	System shall have a total ground surface area less than 100cm^2	SN06 Deployment SN07 Recovery	Y
10	REQ10 Deployment Time	System shall be deployable from protective case to mission start within 5 minutes	SN20 Delicate Area Avoidance	N
11	REQ11 Wireless Communication	System shall include two-way wireless communication	SN08 Movement SN19 Ground Impact	Y
12	REQ12 Configurable Missions	System shall be able to accept constraints on movement based on SLAM coordinates during deployment	SN12 Biomimicry REQ13 Legged Locomotion	Y
13	REQ13 Legged Locomotion	System shall be able to maneuver limbs in order to locomote on the natural cave floor	REQ13 Legged Locomotion	N
14	REQ13.1 Inclined Traversability	System shall be able to manoeuvre across slopes of at least 10deg incline	REQ13 Legged Locomotion	N
15	REQ13.2 Surface Traversability	System shall be able to manoeuvre across all types of cave floor materials of varying roughness (excluding water)	SN09 Autonomy SN14 Mapping	Y
16	REQ14 Onboard SLAM Processing	System shall be able to execute a real-time SLAM algorithm via onboard microcontrollers and sensors	SN14 Mapping SN09 Autonomy	Y
17	REQ15 Operating Time	System shall have be able to operate autonomously for up to 20 minutes	SN08 Movement	N
18	REQ16 Stability	System shall minimise contact with the cave floor at all times by remaining upright	SN14 Mapping SN15 Map Usability	Y
19	REQ17 Map Data Storage	System shall have sufficient secure data storage to store 3D map data generated over 5 to 20 minute mission time	SN08 Movement SN14 Mapping	N
20	REQ18 Ambient Temperature	System shall be able to operate in ambient temperatures between 5 and 25 degrees celsius	SN15 Map Usability SN03 Cave Conditions	Y
21	REQ19 Humidity	System shall be able to operate in humid conditions with relative humidity up to 90%	SN03 Cave Conditions	Y
22	REQ20 Avoiding Objects	System shall be able to autonomously avoid collisions with walls, ceilings, and decorations	SN17 Environmental Impact SN18 World Heritage Site	N
23	REQ21 Multi-Mission	System shall be easily resettable such that missions can be repeated several times without surfacing	SN10 Reuseability SN07 Recovery	N
24	REQ22 Always Recoverable	System shall prioritise being able to return to the mission starting location above cave mapping when required	SN10 Reuseability	N

Figure A.6: CaveX System Requirements Table - Requirements with ID, Description, derivation from stakeholder needs and applicable traceability

(Cooper et al. 2021).

A.7 CaveX 2023 User Needs

Table A.1: CaveX 2023 User Needs.

Identifier	Description	Traceability	Criticality
UN-01	The user needs a system that shall be operational in cave climates.	SN03 (2021)	Essential
UN-02	The user needs a system that is easily deployable.	SN06 (2021) & Scenario-based Needs Analysis	Very Important
UN-03	The user needs a system that can quickly be recovered after a mission.	SN07 (2021) & Scenario-based Needs Analysis	Very Important
UN-04	The user needs a system that is capable of traversing complex cave terrain.	SN08 (2021) & Scenario-based Needs Analysis	Essential
UN-05	The user needs an autonomous system that can remain operational out of reach and sight.	SN09 (2021)	Essential
UN-06	The user needs a reliable system that can be deployed for multiple missions in a single day.	SN10 (2021)	Very Important
UN-07	The user needs a system that can provide real-time mapping on a connected remote operator laptop.	SN11 (2021) & System Context	Important
UN-08	The user needs an autonomous system that can successfully navigate around obstacles inside a cave.	SN17 (2021) & Scenario-based Needs Analysis	Essential
UN-09	The user needs a system that can maintain two-way communication during its operation.	Scenario-based Needs Analysis	Essential

A.8 CaveX 2023 System Requirements

Table A.2: CaveX 2023 System Requirements.

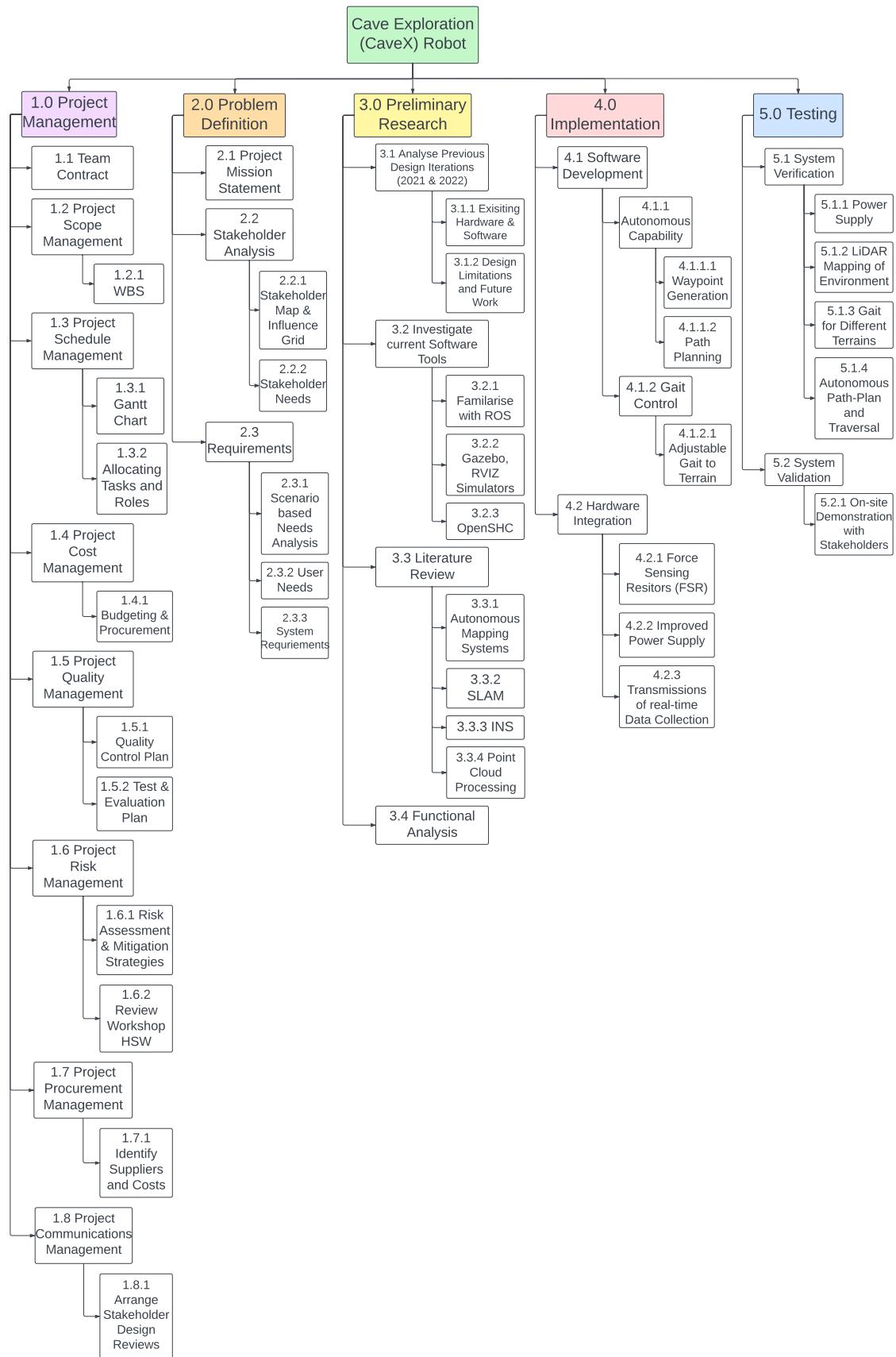
Identifier	Description	Traceability	Functionality	Criticality	Verification Method
SR-01	The system shall remain operational in a temperature range of 5°C to 25°C.	UN-01	Non-Functional	Essential	Testing
SR-02	The system shall fully initialise and autonomously begin its cave mapping mission when the user switches the power supply on.	UN-02	Functional	Very Important	Demonstration
SR-03	The system shall immediately stop mapping the cave and return to its deployed location within 0.1m when the user sends a termination request.	UN-03 UN-09	Functional	Very Important	Testing
SR-04	The system shall be capable of two-way communication with a remote user 99% of the time during operation.	UN-03 UN-09	Functional	Essential	Testing
SR-05	The system shall be able to automatically adjust its gait when required to traverse different terrains.	UN-04	Functional	Essential	Testing

SR-06	The system shall be able to sense low ceilings and avoid passages which are too narrow.	UN-04 UN-08	Functional	Essential	Testing
SR-07	The system shall be able to detect step inclines or declines which are unsafe to traverse.	UN-04	Functional	Essential	Testing
SR-08	The system shall independently plan a path for the next 25 steps to explore unmapped cave sections.	UN-05	Functional	Essential	Demonstration
SR-10	The system shall be capable of uploading point cloud data to a remote storage server for real-time mapping display.	UN-07	Functional	Important	Testing
SR-11	The system shall successfully avoid 95% of the obstacles it encounters.	UN-08	Functional	Essential	Testing
SR-12	The system shall be capable of toggling between autonomous and manual control.	UN-03	Functional	Important	Demonstration
SR-13	The system shall be able to prioritise different cave sections at the user's request.	UN-06	Functional	Important	Testing

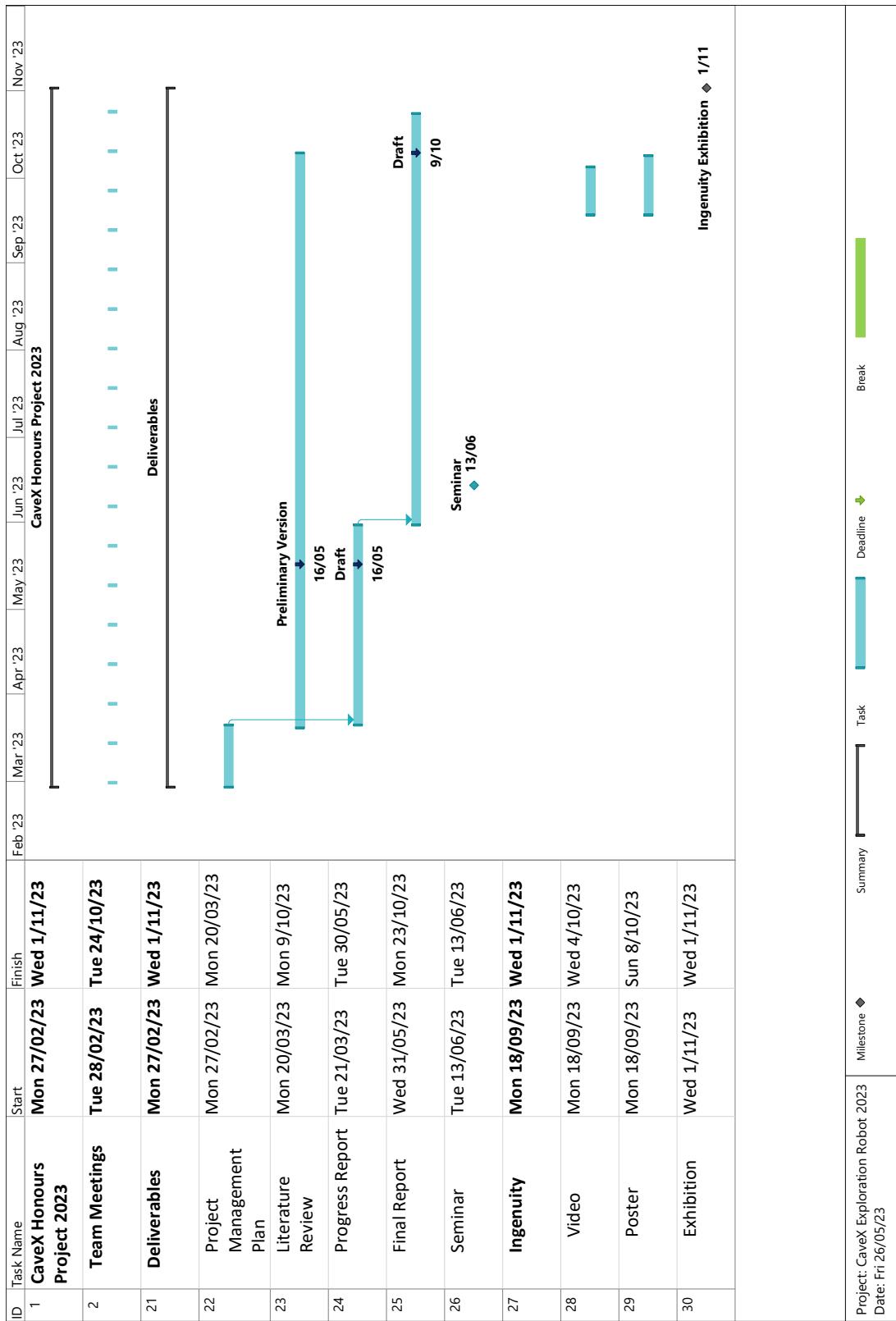
SR-14	The system shall be capable of autonomously adjusting its field of view for complete mapping.	UN-05	Functional	Important	Testing
SR-15	The system shall be able to operate at 100% relative humidity for the temperature range 5°C to 25°C.	UN-01	Non-Functional	Very Important	Testing
SR-16	The system shall be capable of sensing when its legs are in contact with the ground.	UN-04 UN-05	Functional	Very Important	Testing
SR-17	The system shall be able to use LiDAR data to determine its own position in the map.	UN-05 UN-07	Functional	Essential	Testing
SR-18	The system shall wirelessly transmit or receive data with a delay no greater than 3 seconds.	UN-05 UN-07 UN-09	Non-Functional	Important	Testing
SR-19	The system shall be capable of storing which cave sections have been already mapped.	UN-08	Functional	Very Important	Testing
SR-20	The system shall alert the user in the event of a compromise in its ability to conduct its mission.	UN-05 UN-09	Functional	Important	Testing

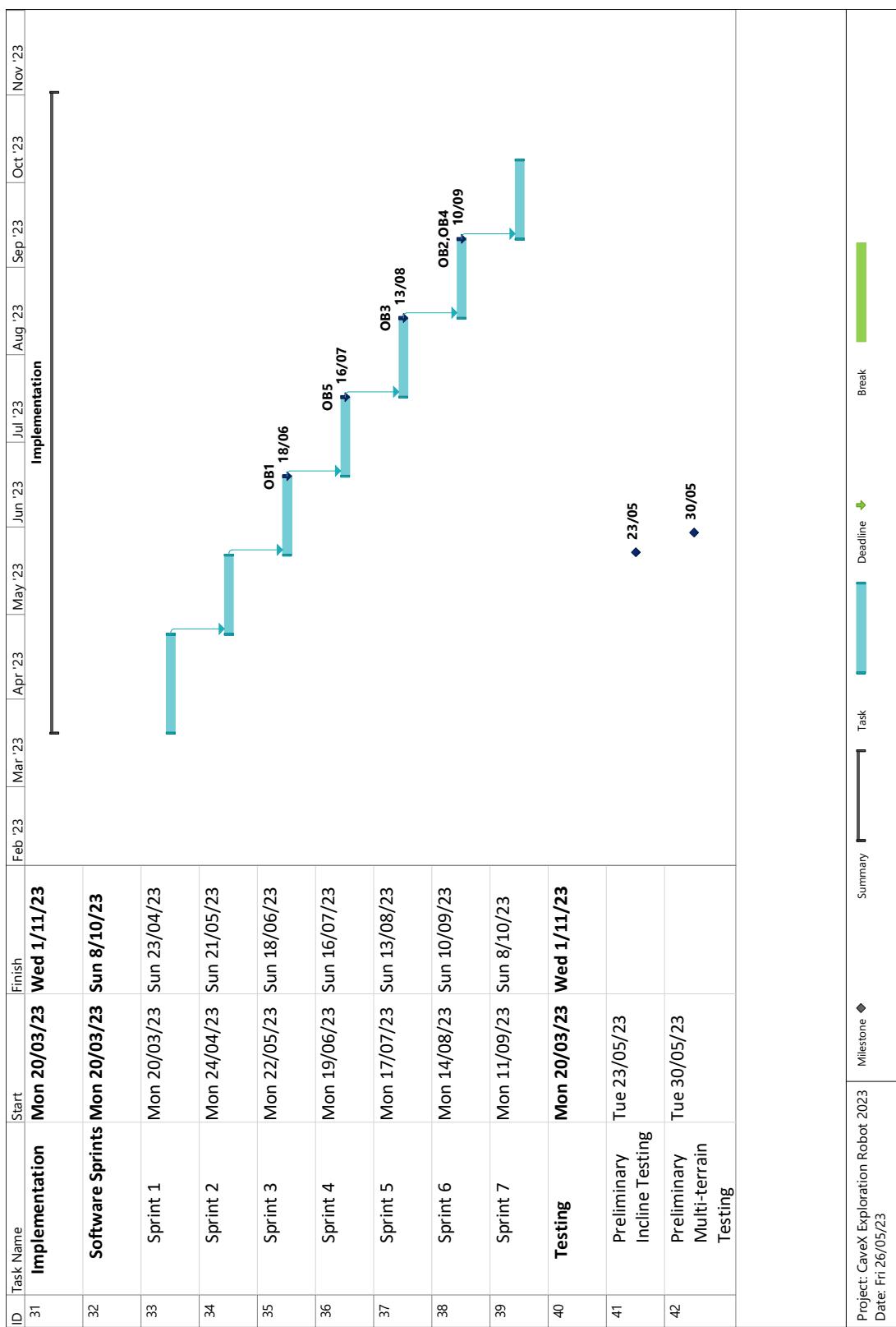
SR-21	The system shall have at least 7GB storage space for two full mission maps	UN-06	Non-Functional	Very Important	Demonstration
-------	--	-------	----------------	----------------	---------------

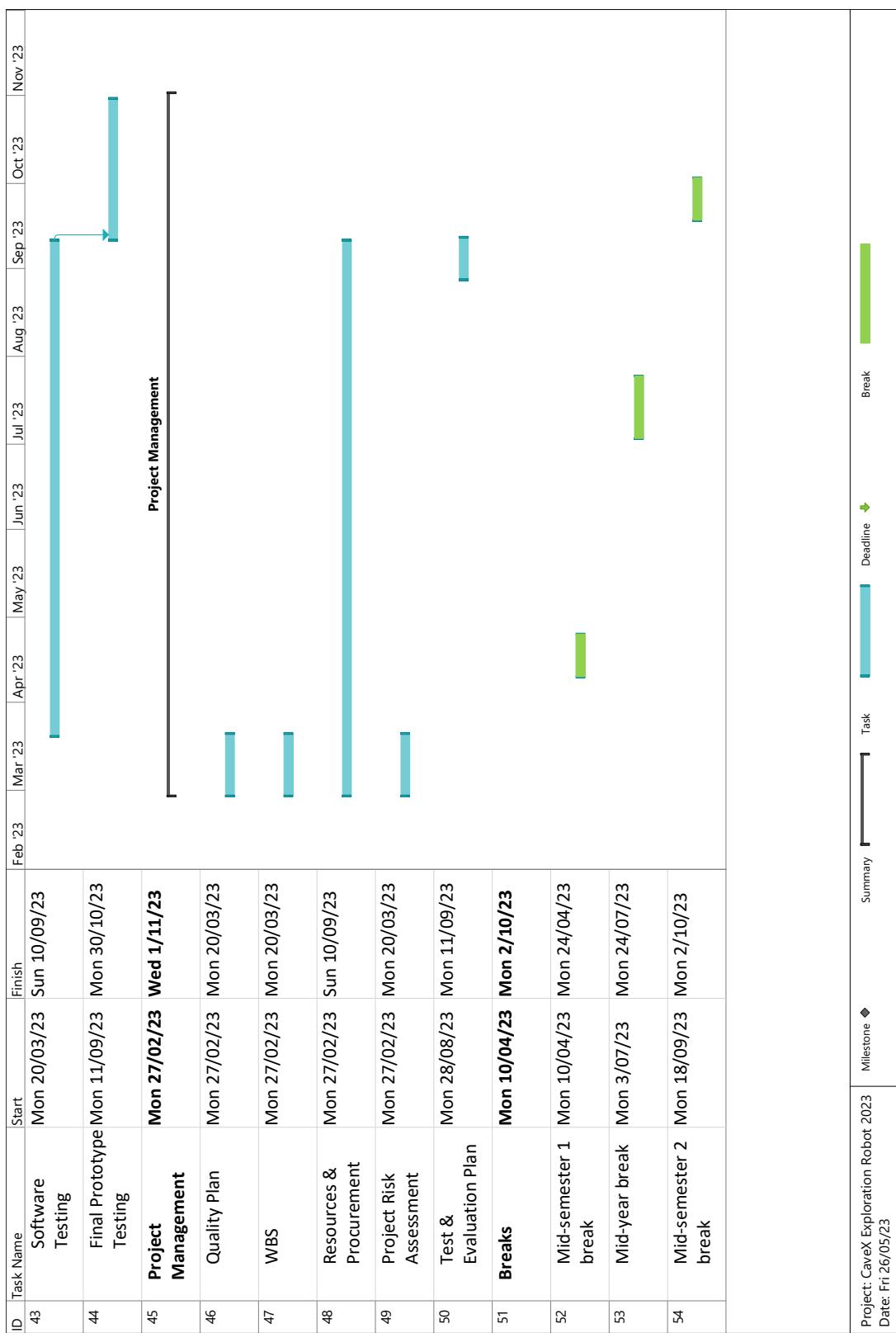
A.9 Work Breakdown Structure



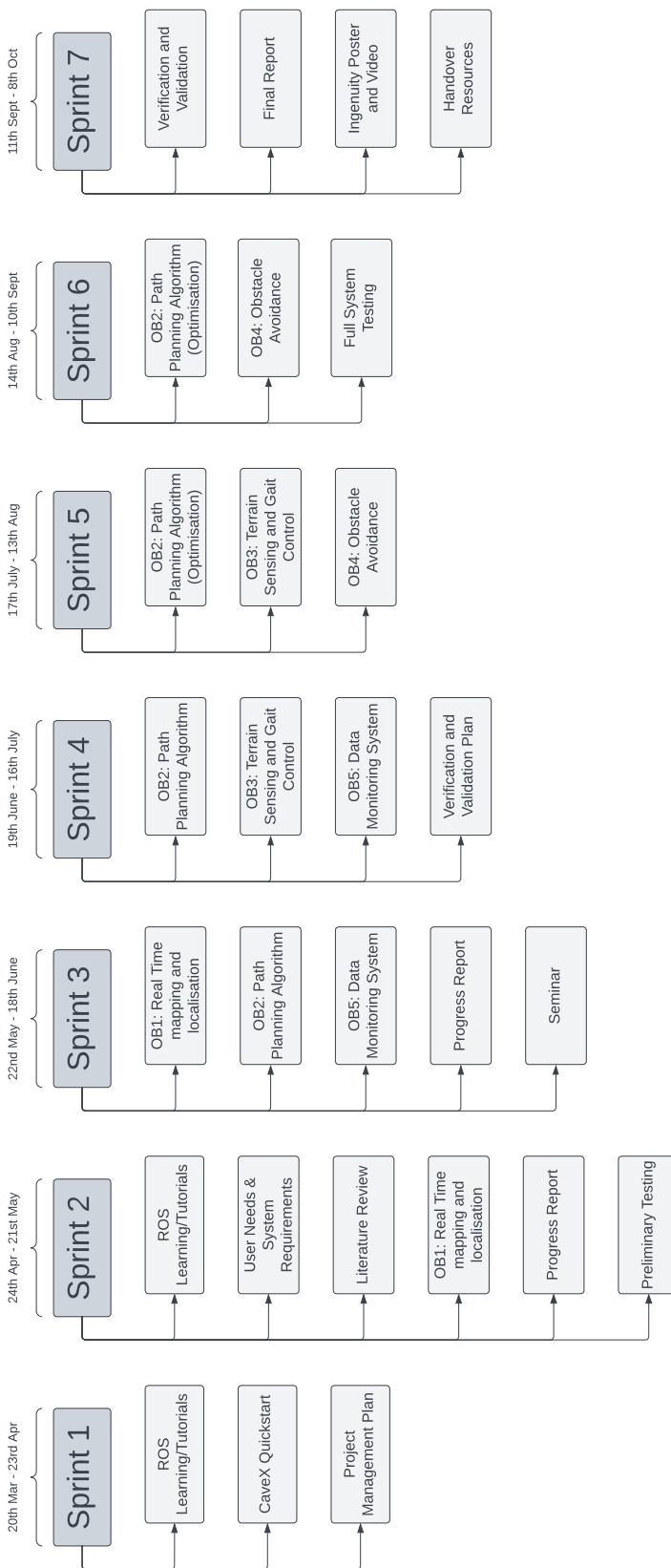
A.10 Gantt Chart







A.11 Sprint Plan



A.12 Risk and Safety

Two types of risks, project risks and safety risks, were assessed to minimise the chance of negative outcomes throughout the project. Project risks are those which could impact the progress of the project as a whole. Safety risks are those which could adversely effect the health of anyone involved in or in the vicinity of project operations.

A.12.1 Project Risks

Table A.3: Project risk assessment for CaveX 2023 Honours Project

Event Context:		CaveX 2023 Honours Project Plan						
Identified Risk	Impact	Initial Likelihood	Initial Severity	Initial Risk	Mitigation Strategies	New Likelihood	New Severity	Residual Risk
University shutdown due to COVID-19.	The robotics lab cannot be accessed. Meetings cannot be conducted in person.	Unlikely (2)	Major (4)	Medium (8)	Store all relevant files and software on multiple devices. Store CaveX laptop at a group member's house.	Rare (1)	Major (4)	Low (4)
Key robotics hardware is damaged.	Experimental testing with the prototype becomes impossible.	Unlikely (2)	Catastrophic (5)	High (10)	Where not reasonable, a replacement will be ordered as soon as possible. Hardware will be used in accordance with the relevant specifications.	Unlikely (2)	Minor (2)	Low (4)

Risk Register						
Risk ID		Risk Description		Risk Response		
Risk Type	Risk Sub-Type	Impact	Probability	Action	Owner	Priority
Laptop is lost or damaged.	Software may be lost. Project cannot proceed.	Unlikely (2)	Catastrophic (5)	High (10) Backups of all relevant files and software will be stored externally to the laptop and updated at the end of the day following an update or change.	Major (4)	Low (4)
Robot causes human harm.	Robot electronics could cause harm or autonomous behaviour could present danger.	Unlikely (2)	Catastrophic (5)	High (10) Electrical component testing with approved University of Adelaide. Extensive testing in a controlled environment to ensure controlled and safe autonomous movement.	Major (4)	Low (4)
Project objectives not completed in time for assessment.	Impact on the stakeholder's research and mapping process. Impacted project quality and representation at Ingenuity Exhibition.	Possible (3)	Moderate (3)	Medium (9) Create detailed Gantt chart showing deadlines of major assessments as well as processes to be undertaken to fulfill deadlines. Weekly team meetings to track and monitor progress.	Unlikely (2)	Minor (2)

Project expenditure goes over budget.	Loss of essential funds to acquire hardware necessary to complete the project objectives.	Next year's group will have an impacted understanding of the project which will significantly lower quality.	Robot runs out of power whilst in cave system.	Extensive research on suppliers and hardware requirements to compare prices and gain a detailed understanding of what is needed.
Possible (3)	Likely (4)	Moderate (3)	High (12)	High (12)
Possible (3)	Possible (3)	Moderate (3)	Medium (9)	Moderate (3)
Equipment loss which may not be retrievable.	Robot runs out of power whilst in cave system.	Increase robot battery life to support extended autonomous ventures. Implement a return to deployment position command when battery life falls below minimum threshold.	Robot runs out of power whilst in cave system.	Create handover documents to update next year's team of the completed work so far and what still needs to be done.

Robot gets loss or pathfinding capability malfunctions.	Equipment loss which may not be retrievable.	Unlikely (2)	Major (4)	Medium (8)	Rare (1)	Major (4)	Low (4)
							Conserve the robots ability to be manually controlled, signal to toggle between autonomous and manual control.

A.12.2 Safety Risks

The key safety risks which have needed to be considered regard the safe operation of the robot and the incline testing rig in the EXTERRES laboratory. The Risk Assessment (RA) and Safe Operating Procedure (SOP) for the robot use and the incline testing rig is included in this appendix.



HAZARD MANAGEMENT – RISK ASSESSMENT

Date: 10/05/2023

SINGLE TASK

RECORD THE HIGHEST RESIDUAL RISK RATING

Ensure the appropriate level of authority to complete the activity can be evidenced.
(e.g. a signature or formal approval attached)

- Low
- Medium
- High
- Very high

Title of the task	Use of the CaveX robot
Physical location(s) or operational unit	The University of Adelaide
Names of workers involved in completing the risk assessment	Author: Luka Moran (a1726133) Other workers : N/A

Supervisors/person in control of the area/activity

- Ensure that the control measures address the hazards identified for each step in the process for this task.
- Ensure that there is a system for retaining this Risk assessment. (See section 5.1 of the Handbook chapter)
- Ensure that workers who undertake this task have access to this Risk assessment, are provided with the relevant, information, instruction and training required before they undertake the task. (This includes any other guidance material (e.g. Safe operating procedures) where required by this Risk assessment.)
- Ensure that if there is a requirement for instruction (Level 2 proficiency) and/or training (Level 3 competency/qualification) the information is added to the Training Plan.

Hazard identification: Stop and think. What could cause harm from start to finish?	Assess the harm	What needs to be in place before you start?	Re-assess the level of risk	
Identify and list each hazard that is part of this work process	Record how/when the worker is exposed to the hazard (e.g. what is the route of exposure when completing the task)	Calculate the risk rating without controls in place (See descriptor table overleaf)	The measures you select must address the hazard, be selected in accordance with the Hierarchy of Control and be clear to the worker. (Refer to the Hierarchy of Control Appendix A page 6 for guidance.)	i.e. the residual risk rating after controls are in place
Manual handling	The worker is required to lift and move the robot.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	The testing rig should be lifted by at least two workers at all times. The testing rig should be oriented vertically on its larger end before lifting and placing it on the trolley. The load should be kept as close to the body as practical. Lifting should be done primarily with the legs. Workers should warm up before lifting if necessary. Workers should discuss the plan to lift the rig before doing so. Verbal communication should be used while lifting where necessary.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high
Electrical shock	The worker is required to handle the robot. The robot is comprised of many exposed wires, pins, and electrical components. This presents a risk of electrocution.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	The robot should be turned off when changing components (batteries, microprocessors, sensors, etc). Wires should be contained within the chassis as much as possible. When physically handling the robot, the worker's hands should be placed in an area assessed to be safe at the time.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 1 of 3
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

HSW Handbook



Trip	The worker is required to manually control the robot or set it to an autonomous mode. In either circumstance, the robot may become a trip hazard to anybody in the area of operation.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	When under manual control, the robot's path should be continually monitored to prevent it from causing people to trip over it. When autonomous, the area it is being used in should ideally be cleared of people other than the operator(s). If this is not possible or practical, then the other people in the area should be made aware of when the robot starts being operated and when it's no longer being operated.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high
Sharp objects	The worker is required to handle the robot. The robot consists of many wires and rigid edges which present a risk of sustaining cuts.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	Worker will ensure that the robot is picked up with two hands at all times. Hands will be placed on the robot such that the chance of contacting sharp surfaces or objects is minimised.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 2 of 3
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			



Appendix B1 (Page 2 of 2)

Authorisation for staff and student related tasks			
Residual risk rating	Authorisation	Name and signature (or attach evidence of authorisation)	
Low & medium risk	Supervisor/Person in control of the area/activity	Assoc. Prof. David Harvey	
High risk	Head of School/Branch		
Very high risk	Executive Dean/Divisional Head		

Proof of hazard identification and risk assessment is required for this task

- File your completed Risk assessment as instructed by the Supervisor/Person in control of the area/activity
- Ensure there is a system for retaining formal Risk assessments in accordance with the State Records of SA, General disposal [Schedule No 30](#) issued under the State Records Act 1997. (Contact the University's [Records Management Office](#) for further assistance/information if required.)

For activities with a Residual risk rating of high or very high risk

- The Head of School/Branch or Executive Dean/Divisional Head is to raise a risk under the [University's Risk management framework](#) through the [University Risk Register](#).

DESCRIPTORS FOR ASSESSING THE LEVEL OF RISK

Likelihood Table

CATEGORY	DESCRIPTION
Almost certain	There is an expectation that an event/incident will occur.
Likely	There is an expectation that an event/incident could occur but not certain to occur.
Possible	This expectation lies somewhere in the midpoint between "could" and "improbable". May happen occasionally.
Unlikely	There is an expectation that an event/incident is doubtful or improbable to occur.
Rare	There is no expectation that the event/incident will occur.

Consequences Table

CATEGORY	DESCRIPTION				
	Negligible	Minor	Moderate	Major	Severe
Severe	Injury resulting in death, permanent incapacity.				
Major		Injury requiring extensive medical treatment (e.g. hospitalisation), or activities could result in a Notifiable occurrence.			
Moderate			Injury requires formal medical treatment (e.g. hospital outpatient/doctors visit) Activities could result in an Improvement/Prohibition Notice.		
Minor				Injury requires first aid treatment.	
Negligible					Injury requires minor first aid (e.g. bandaid), or result in short term discomfort (e.g. bruise, headache, muscular aches), no medical treatment.

Likelihood of exposure	Consequences – level of seriousness of the injury following exposure to the hazard(s) -					
	Negligible	Minor	Moderate	Major	Severe	
Almost certain	<input type="checkbox"/>	Medium	<input type="checkbox"/>	High	<input type="checkbox"/>	Very High
Likely	<input type="checkbox"/>	Medium	<input type="checkbox"/>	High	<input type="checkbox"/>	Very High
Possible	<input type="checkbox"/>	Low	<input type="checkbox"/>	Medium	<input type="checkbox"/>	High
Unlikely	<input type="checkbox"/>	Low	<input type="checkbox"/>	Medium	<input type="checkbox"/>	Very High
Rare	<input type="checkbox"/>	Low	<input type="checkbox"/>	Low	<input type="checkbox"/>	Medium

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 3 of 3
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

HAZARD MANAGEMENT – SAFE OPERATING PROCEDURE (SOP)

Only to be completed where required as a control measure under a Risk Assessment

<p>A document setting out the requirements to carry out the work in a safe and healthy manner and in a logical sequence.</p> <p>It must be able to be easily read by those who need to know what has been planned.</p> <p>It is relevant to the following people:</p> <ul style="list-style-type: none"> • the worker carrying out the work; and • the person who has management and control over the work. 		<p>A SOP, if identified as a control measure, is to:</p> <ul style="list-style-type: none"> • identify the work; • specify/address the identified hazards relating to the work; • describe the measures to be implemented to control the risks; • take into account the circumstances at the workplace that may affect the way in which the work is carried out; • take into account emergency management arrangements where applicable; and • be communicated to all workers who carry out the work.
NAME OF THE TASK/ACTIVITY	USE OF CAVEX ROBOT	DATE: 10/5/2023
LOCATION	THE UNIVERSITY OF ADELAIDE	
RISK ASSESSMENT (RA) NAME	Robot Operation Risk Assessment	
Residual risk rating on the RA	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very High	
Hazards identified on the RA	<ul style="list-style-type: none"> • Manual handling • Electrical shock • Trip • Sharp objects 	
PERSONAL PROTECTIVE EQUIPMENT		
 <input checked="" type="checkbox"/> Long hair must be contained or covered  <input type="checkbox"/> Other:  <input checked="" type="checkbox"/> Enclosed footwear: <input type="checkbox"/> Footwear that is resistant to spills of hazardous substances <input type="checkbox"/> Boots with steel caps <input type="checkbox"/> Other:		
DESCRIBE, IN SEQUENCE, STEPS TO COMPLETE THE ACTIVITY SAFELY		
Pre-operational checks <p>Operational checks/steps to complete the activity from start to finish (including transport and waste disposal where relevant)</p> <ol style="list-style-type: none"> 1. Handle robot with care (follow controls described in risk assessment) 2. Ensure area is clear 3. Place robot on surface 4. Extend robot's legs 5. Connect battery to robot 6. Turn robot on 7. Wait for system to initialise 8. Operate robot (manual or autonomous) 9. When finished, stop robot operation (stop manually moving it via controller or turn off autonomous functionality) 10. Retrieve robot 11. Turn robot off while holding it up 12. Remove battery from robot 13. Pack up system 		
On completion of work – steps to make safe (including clean up, any waste disposal & service/maintenance requirements) <ol style="list-style-type: none"> 1. Tidy up and dispose of any rubbish 2. Clean up working area 		

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 1 of 2
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

HSW Handbook**Emergency and Spill Procedures, Transport or storage requirements (where relevant), First aid/Medical****Emergency Procedures:**

- In the event of an emergency, the priority of contacts is:
 1. 000 (triple zero) (if necessary)
 2. S226/EXTERRES Lab (NG40-41) Area Manager (depending on location of test)
 3. Campus security

Spill Procedures: N/A

Transport:

- Keep robot in storage case (regular suitcase or Pelican case)

Storage:

- Ensure the robot/system won't damage itself or any nearby equipment when in storage

First Aid/Medical:

- For minor cuts, refer to the S226 or EXTERRES Lab (NG40-41) documentation (if applicable) or rinse area and apply band-aid
- For severe medical issues, call 000 (triple zero) immediately

Prepared by

People involved in the drafting of this SOP	Luka Moran (a1726133)	
Person authorising the SOP	Name: Assoc. Prof. David Harvey	Signature
	Position: Supervisor	

This SOP must be reviewed after any incident/injury associated with this activity or when a Risk assessment is reviewed.

File your completed SOP as instructed by the Supervisor/Person in control of the area/activity and retain the SOP in accordance with the State Records of SA, General disposal [Schedule](#) No. 30 issued under the State Records Act 1997. (Contact the University's [Records Management Office](#) for further assistance/information if required.)

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 2 of 2
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

Appendix B1 (Page 1 of 2)

HAZARD MANAGEMENT – RISK ASSESSMENT

Date: 10/05/2023

SINGLE TASK

(This template or equivalent template can be used)

(If you have not completed a risk assessment before refer to the [Handbook Chapter Appendix A](#) for guidance)

		RECORD THE HIGHEST RESIDUAL RISK RATING	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high
		Ensure the appropriate level of authority to complete the activity can be evidenced. (e.g. a signature or formal approval attached)	

Supervisors/person in control of the area/activity	
<ul style="list-style-type: none"> • Ensure that the control measures address the hazards identified for each step in the process for this task. • Ensure that there is a system for retaining this Risk assessment. (See section 5.1 of the Handbook chapter) • Ensure that workers who undertake this task have access to this Risk assessment, are provided with the relevant, information, instruction and training required before they undertake the task. (This includes any other guidance material (e.g. Safe operating procedures) where required by this Risk assessment.) • Ensure that if there is a requirement for instruction (Level 2 proficiency) and/or training (Level 3 competency/qualification) the information is added to the Training Plan. 	

Hazard identification: Stop and think. What could cause harm from start to finish?	Assess the harm	What needs to be in place before you start?	Re-assess the level of risk
Identify and list each hazard that is part of this work process	Record how/when the worker is exposed to the hazard (e.g. what is the route of exposure when completing the task)	Calculate the risk rating without controls in place (See descriptor table overleaf)	The measures you select must address the hazard, be selected in accordance with the Hierarchy of Control and be clear to the worker. (Refer to the Hierarchy of Control Appendix A page 6 for guidance.)
Lifting of heavy equipment	The worker is required to lift the testing rig onto and off of a trolley for transportation as well as placing it in and removing it from the sand pit. Improper lifting technique and insufficient communication present a risk of injury.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	The testing rig should be lifted by at least two workers at all times. The testing rig should be oriented vertically on its larger end before lifting and placing it on the trolley. The load should be kept as close to the body as practical. Lifting should be done primarily with the legs. Workers should warm up before lifting if necessary. Workers should discuss the plan to lift the rig before doing so. Verbal communication should be used while lifting where necessary.
Sliding of heavy equipment	The worker may be required to slide the testing rig on a surface to position it correctly. Improper consideration of the surroundings and improper push/pull technique present a risk for injury.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	The testing rig should be moved by at least two workers at all times. The surrounding environment should be assessed prior to sliding the rig.

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 1 of 3
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

HSW Handbook



Relocating heavy equipment	The worker needs to relocate the testing rig between two labs. Improper consideration of the path between them presents a possibility of dropping the rig or colliding with another object.	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	Where possible, relocation should be done with the rig on a trolley. The path between the two locations should be planned and discussed with all involved parties prior to relocation.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high
Placing the robot on and removing it from the testing rig	The worker is required to place the robot on and remove it from the testing rig	<input type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	Worker will ensure that the testing rig is stable before placing the robot on it. The robot will be slowly let go of to ensure it remains stable on the rig.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high
Conducting tests with the robot	The worker will control the robot to perform tests	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high	Worker will ensure the various components of the robot are secured safely within its chassis. The path of the robot will be continuously monitored to ensure it does not fall off the edge of the rig.	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very high

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 2 of 3
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

Appendix B1 (Page 2 of 2)

Authorisation for staff and student related tasks			
Residual risk rating	Authorisation	Name and signature (or attach evidence of authorisation)	
Low & medium risk	Supervisor/Person in control of the area/activity	Assoc. Prof. David Harvey	
High risk	Head of School/Branch		
Very high risk	Executive Dean/Divisional Head		

Proof of hazard identification and risk assessment is required for this task

- File your completed Risk assessment as instructed by the Supervisor/Person in control of the area/activity
- Ensure there is a system for retaining formal Risk assessments in accordance with the State Records of SA, General disposal [Schedule No 30](#) issued under the State Records Act 1997. (Contact the University's [Records Management Office](#) for further assistance/information if required.)

For activities with a Residual risk rating of high or very high risk

- The Head of School/Branch or Executive Dean/Divisional Head is to raise a risk under the [University's Risk management framework](#) through the [University Risk Register](#).

DESCRIPTORS FOR ASSESSING THE LEVEL OF RISK

Likelihood Table

CATEGORY	DESCRIPTION
Almost certain	There is an expectation that an event/incident will occur.
Likely	There is an expectation that an event/incident could occur but not certain to occur.
Possible	This expectation lies somewhere in the midpoint between "could" and "improbable". May happen occasionally.
Unlikely	There is an expectation that an event/incident is doubtful or improbable to occur.
Rare	There is no expectation that the event/incident will occur.

Consequences Table

CATEGORY	DESCRIPTION
Severe	Injury resulting in death, permanent incapacity.
Major	Injury requiring extensive medical treatment (e.g. hospitalisation), or activities could result in a Notifiable occurrence.
Moderate	Injury requires formal medical treatment (e.g. hospital outpatient/doctors visit) Activities could result in an Improvement/Prohibition Notice.
Minor	Injury requires first aid treatment.
Negligible	Injury requires minor first aid (e.g. bandaid), or result in short term discomfort (e.g. bruise, headache, muscular aches), no medical treatment.

Likelihood of exposure	The level of risk will increase as the likelihood of harm and its severity increases				
	Consequences – level of seriousness of the injury following exposure to the hazard(s) -				
	Negligible	Minor	Moderate	Major	Severe
Almost certain	<input type="checkbox"/> Medium	<input type="checkbox"/> High	<input type="checkbox"/> Very High	<input type="checkbox"/> Very High	<input type="checkbox"/> Very High
Likely	<input type="checkbox"/> Medium	<input type="checkbox"/> High	<input type="checkbox"/> Very High	<input type="checkbox"/> Very High	<input type="checkbox"/> Very High
Possible	<input type="checkbox"/> Low	<input type="checkbox"/> Medium	<input type="checkbox"/> High	<input type="checkbox"/> High	<input type="checkbox"/> Very High
Unlikely	<input type="checkbox"/> Low	<input type="checkbox"/> Low	<input type="checkbox"/> Medium	<input type="checkbox"/> Medium	<input type="checkbox"/> High
Rare	<input type="checkbox"/> Low	<input type="checkbox"/> Low	<input type="checkbox"/> Low	<input type="checkbox"/> Medium	<input type="checkbox"/> Medium

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 3 of 3
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

Appendix C

HAZARD MANAGEMENT – SAFE OPERATING PROCEDURE (SOP)

Only to be completed where required as a control measure under a Risk Assessment

A document setting out the requirements to carry out the work in a safe and healthy manner and in a logical sequence. It must be able to be easily read by those who need to know what has been planned. It is relevant to the following people: <ul style="list-style-type: none">• the worker carrying out the work; and• the person who has management and control over the work.	A SOP, if identified as a control measure, is to: <ul style="list-style-type: none">• identify the work;• specify/address the identified hazards relating to the work;• describe the measures to be implemented to control the risks;• take into account the circumstances at the workplace that may affect the way in which the work is carried out;• take into account emergency management arrangements where applicable; and• be communicated to all workers who carry out the work.	
NAME OF THE TASK/ACTIVITY	USE OF THE INCLINE TESTING RIG	DATE: 10/5/2023
LOCATION	EXTERRES LAB SAND PIT (NG40-41), ROBOTICS LAB (S226)	
RISK ASSESSMENT (RA) NAME	Incline Test Rig Risk Assessment	
Residual risk rating on the RA	<input checked="" type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Very High	
Hazards identified on the RA	<ul style="list-style-type: none"> • Lifting of heavy equipment • Sliding of heavy equipment • Relocating heavy equipment • Placing the robot on and removing it from the testing rig • Conducting tests with the robot 	
PERSONAL PROTECTIVE EQUIPMENT		
	<input checked="" type="checkbox"/> Long hair must be contained or covered <input type="checkbox"/> Other:	
	<input checked="" type="checkbox"/> Enclosed footwear: <input type="checkbox"/> Footwear that is resistant to spills of hazardous substances <input type="checkbox"/> Boots with steel caps <input type="checkbox"/> Other:	
DESCRIBE, IN SEQUENCE, STEPS TO COMPLETE THE ACTIVITY SAFELY		
<p>Pre-operational checks</p> <p>Operational checks/steps to complete the activity from start to finish (including transport and waste disposal where relevant)</p> <ol style="list-style-type: none"> 1. If necessary, transport the testing rig to the desired location (following the controls described in the risk assessment) 2. Ensure the testing rig is on a stable surface 3. If using the sand pit in the EXTERRES Lab (NG40-41), ensure that the testing rig is positioned such that it's parallel with OptiTrack's x or y axis 4. When placing the robot on the rig, ensure that it is stable and as far towards the base or top of the rig as possible (whether it's supposed to be at the base depends on the type of test) 5. When testing, ensure the robot is not going to fall off the rig 6. Upon completion of tests, turn off and pack up the robot before moving the test rig 7. If in EXTERRES Lab (NG40-41) then transport testing rig back to S226 (following the controls described in the risk assessment) 8. If in S226 then simply move the testing rig back to its original place in S226 (following safe lifting/moving practices, as described in risk assessment) <p>On completion of work – steps to make safe (including clean up, any waste disposal & service/maintenance requirements)</p> <ol style="list-style-type: none"> 1. Tidy up and dispose of any rubbish 2. If using sand pit then use a broom to clean the surrounding floor 3. Clean up working area 		

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 1 of 2
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

HSW Handbook**Emergency and Spill Procedures, Transport or storage requirements (where relevant), First aid/Medical****Emergency Procedures:**

- In the event of an emergency, the priority of contacts is:
 1. 000 (triple zero) (if necessary)
 2. S226/EXTERRES Lab (NG40-41) Area Manager (depending on location of test)
 3. Campus security

Spill Procedures: N/A

Transport:

- Follow controls described in the risk assessment

Storage:

- Ensure the testing rig won't damage any surrounding structures or equipment when in storage

First Aid/Medical:

- For minor cuts, refer to the S226 or EXTERRES Lab (NG40-41) documentation
- For severe medical issues, call 000 (triple zero) immediately

Prepared by

People involved in the drafting of this SOP	Luka Moran (a1726133)		
Person authorising the SOP	Name:	Assoc. Prof. David Harvey	Signature

This SOP must be reviewed after any incident/injury associated with this activity or when a Risk assessment is reviewed.

File your completed SOP as instructed by the Supervisor/Person in control of the area/activity and retain the SOP in accordance with the State Records of SA, General disposal [Schedule](#) No. 30 issued under the State Records Act 1997. (Contact the University's [Records Management Office](#) for further assistance/information if required.)

HSW Handbook	Hazard Management	Effective Date:	1 December 2020	Version 4.0
Authorised by	Chief Operating Officer (University Operations)	Review Date:	1 December 2023	Page 2 of 2
Warning	This process is uncontrolled when printed. The current version of this document is available on the HSW Website.			

A.13 Quality Management

Faculty of Sciences, Engineering and Technology

**ENG 4001 Honours Research Project
Project Quality Management Plan**

CaveX Exploration Robot 2023

Supervisors: A Prof David Harvey, A Prof Rini Akmeliawati

By

Luka Moran: a1726133
Riley Groome: a1766563
Tyler Groome: a1766574

Date: 20/03/2023

Group: 2023s1-EME.Me-DJH-UG-13428

1 Scope

This document outlines the quality management plan for the 2023 CaveX Honours project team. The fundamental processes and procedures in place to ensure the production of a high-quality system are identified and closely related to system testing and evaluation planning in verification of the system level requirements. A carefully considered quality management plan also assists in the team's validation of the design to ensure the project outcome aligns with the end users' expectations.

2 Overview

The Project Management Body of Knowledge (PMBOK) guide outlines the key tools and techniques for quality management of a project. Quality management can be broken into three distinct processes as shown in figure 1 below. The processes include quality planning, quality assurance and quality control. Quality planning details the key stages in the project and techniques to ensure the project outcome has a high quality. Quality assurance provides the framework for fulfilling the predefined quality metrics and quality control lists the testing processes which ensure these requirements are satisfied.

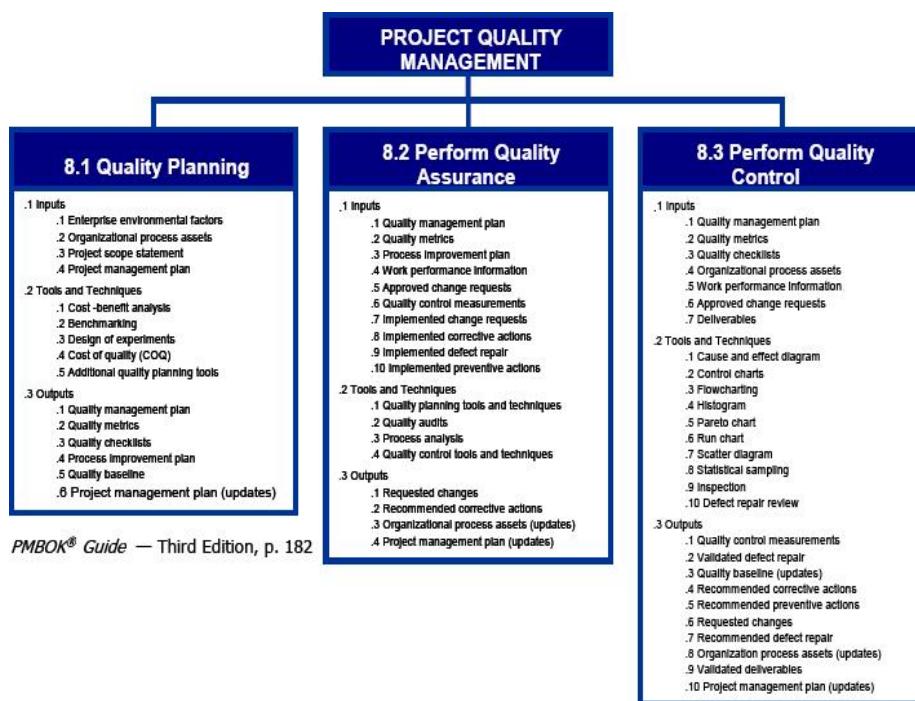


Figure 1: Project Quality Management Overview (PMI 2004) adapted from PMBOK Guide - Third Edition

3 Quality Planning

Planning processes to ensure the quality of the project outcome meets a high standard is essential in ensuring the clients' expectations are met. The team has devised a brief plan using systems engineering tools that will address the quality of the project shown in figure 2 below.

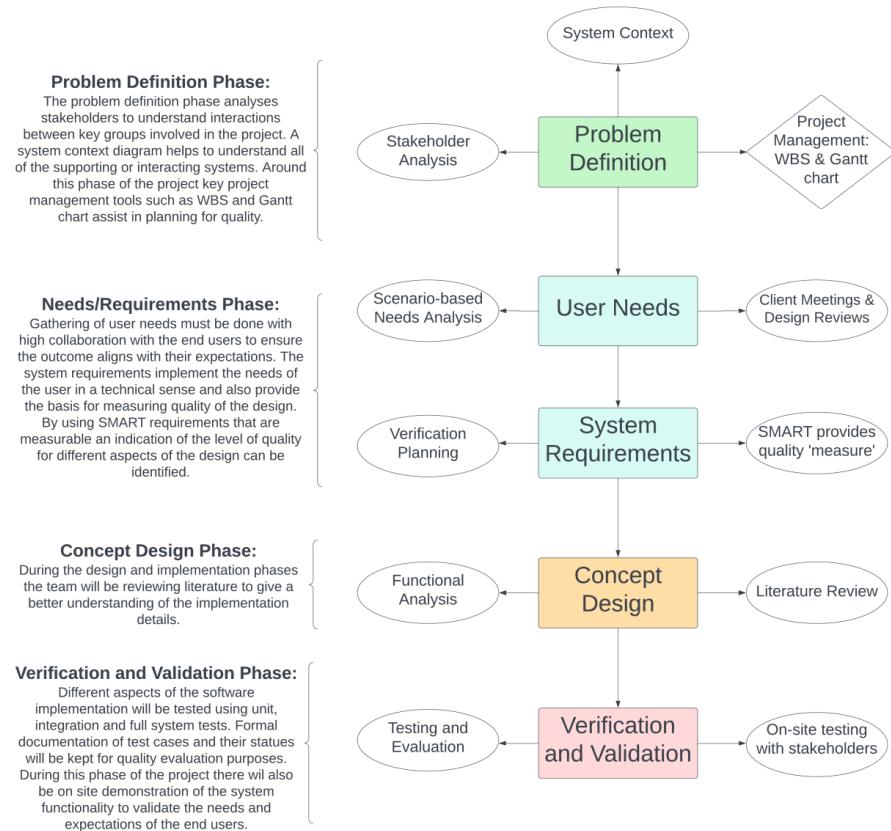


Figure 2: CaveX quality planning for different phases of the project

The first stage is to properly define the problem. This will include a detailed analysis of the current project state and what features need to be improved or added. Furthermore, a stakeholder analysis will be conducted to understand the external groups that are impacted by or can impact the project. The next stage is to accurately identify user needs using systems engineering tools such as scenario-based needs analysis and frequent communication with the end users of the product (Dr Elizabeth Reed and Craig Williams). After a list of fundamental user needs are developed, the team will begin translating these into system requirements that are traceable back to

the needs of the user. The system requirements will be constructed such that they adhere to the SMART (Specific, Measurable, Achievable, Relevant and Time-bound) guideline. The quality of different aspects of the system will be based on the measurable characteristics of the design that are specified to perform to a threshold in the system requirements. To better understand the system's functionality, a system context analysis and functional analysis will be conducted. The group also plans to conduct a literature review on environment mapping and autonomous systems to aid in understanding the implementation details. Finally, a detailed test and evaluation document which identifies fundamental test cases for the software and their corresponding statuses (fail or pass) will be written. This formal record will assist the group in identifying areas which need to be improved or fixed throughout the design phases that will ultimately help the group improve the quality of the final product.

4 Quality Assurance

Quality assurance is the subsection of project quality management that is concerned with providing confidence that the project outcome will fulfill the predefined system requirements which are the basis for the quality metrics (ASQ 2023). The first step in assuring quality for the end users is to carefully construct system requirements that will guide the project design in the right direction. Quality, by definition, is the degree to which a set of inherent characteristics fulfill some set of quality requirements or metrics (Roseke 2018). Therefore, it is imperative that the requirements are specific, measurable, attainable, realistic and time-bound. SMART requirements will not only make the verification processes significantly more efficient, but they will also make it easier to identify areas of the design that do not comply with the desired level of quality.

Another crucial aspect of assuring quality is having carefully considered project management tools in place. The most important include schedule, scope and budget management. The team will follow a systems engineering approach with a strong emphasis on project management to assist in breaking down the project and keeping on the right track. A strict project schedule with soft deadlines, captured in the team's Gantt chart, will ensure the group starts implementation aspects on time which will mitigate risk associated with some project aspects exceeding their allocated time budget. Following the schedule closely will ensure there is adequate time for testing and evaluation of the system which will identify areas for improvement and overall ensure a higher quality product is produced.

5 Quality Control

Quality control refers to the processes and techniques in place to fulfill the quality requirements (ASQ 2023). This project will follow an agile software engineering process where functionality will be developed iteratively in sprints. During each sprint, items will be moved from the product backlog, which contains a set of all tasks to be completed, into the sprint backlog to be completed in the current sprint. When new functionality is added it will be tested immediately. Testing during the sprints will allow early detection of bugs and will mitigate risks associated with poor quality and

exceeding the project's schedule. The most common type of test performed in the sprints will be unit tests that isolate individual software components and attempt to identify any flaws in its implementation. As the software implementation progresses, it will become necessary to perform integration tests which test the interaction between different components in the system such as the LiDAR-based mapping system and path planning software. Finally, towards the end of the project there will be full system testing where the entire robot system will be tested in an environment similar to the caves it will be deployed in to verify each of the system requirements. The system requirements should also be relevant to the needs of the user and, therefore, there must be consistent communications between the end users and CaveX team throughout the design stages. This will help to ensure the final product aligns with the clients' expectations.

6 References

ASQ 2023, *Quality Assurance & Quality Control*, American Society for Quality, viewed 18/03/2023.
<<https://asq.org/quality-resources/quality-assurance-vs-control>>.

Roseke, B 2018, *Project Quality Management According to the PMBOK*, ProjectEngineer, viewed 18/03/2023.
<<https://www.projectengineer.net/project-quality-management-according-to-the-pmbok/>>.

B Technical Datasheets

B.1 Jetson Nano



Join the Revolution and Bring the Power of AI to Millions of Devices

The NVIDIA® Jetson Nano™ Developer Kit delivers the compute performance to run modern AI workloads at unprecedented size, power, and cost. Developers, learners, and makers can now run AI frameworks and models for applications like image classification, object detection, segmentation, and speech processing.

The developer kit can be powered by micro-USB and comes with extensive I/Os, ranging from GPIO to CSI. This makes it simple for developers to connect a diverse set of new sensors to enable a variety of AI applications. It's incredibly power-efficient, consuming as little as 5 watts.

Jetson Nano is also supported by NVIDIA JetPack™, which includes a board support package (BSP), Linux OS, NVIDIA CUDA®, cuDNN, and TensorRT™ software libraries for deep learning, computer vision, GPU computing, multimedia processing, and much more. The software is even available using an easy-to-flash SD card image, making it fast and easy to get started.

The same JetPack SDK is used across the entire NVIDIA Jetson™ family of products and is fully compatible with NVIDIA's world-leading AI platform for training and deploying AI software. This proven software stack reduces complexity and overall effort for developers.



KEY FEATURES

Jetson Nano Module

- > 128-Core NVIDIA Maxwell™ GPU
- > Quad-Core ARM® A57 CPU
- > 4 GB 64-Bit LPDDR4
- > 10/100/1000BASE-T Ethernet

Power Options

- > Micro-USB 5V 2A
- > DC Power Adapter 5V 4A

I/O

- > USB 3.0 Type A
- > USB 2.0 Micro-B

- > HDMI/DisplayPort

- > M.2 Key E
- > Gigabit Ethernet
- > GPIOs, I²C, I²S, SPI, UART
- > MIPI-CSI Camera Connector
- > Fan Connector
- > PoE Connector

Kit Contents

- > NVIDIA Jetson Nano Module with Heatsink and Reference Carrier Board
- > Quick Start Guide and Support Guide

NVIDIA JETSON NANO DEVELOPER KIT

TECHNICAL SPECIFICATIONS

DEVELOPER KIT

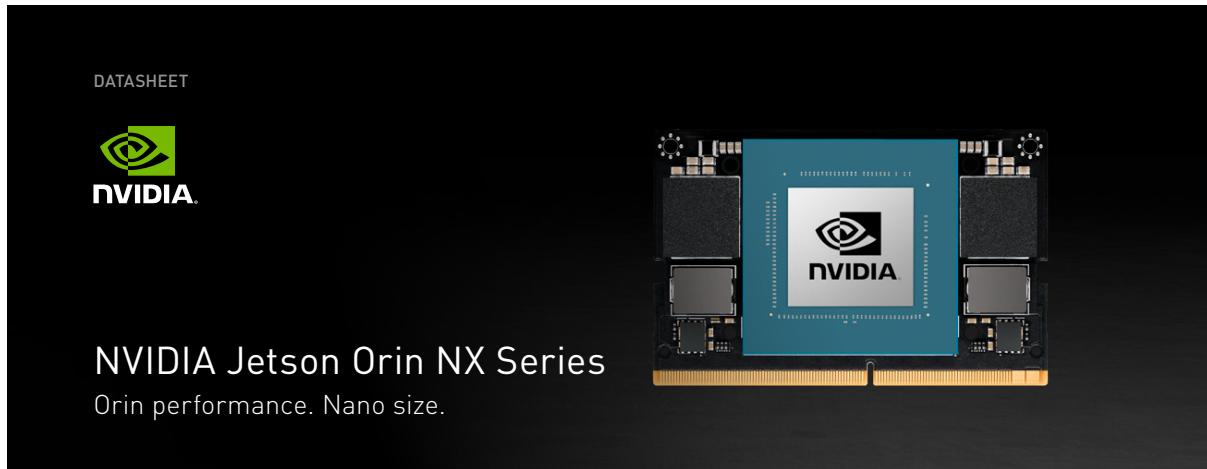
GPU	128-Core Maxwell
CPU	Quad-Core ARM A57 @ 1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD [Not Included]
Video Encoder	4K @ 30 4x 1080p @ 30 9x 720p @ 30 [H.264/H.265]
Video Decoder	4K @ 60 2x 4K @ 30 8x 1080p @ 30 18x 720p @ 30 [H.264/H.265]
Camera	2x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI 2.0 and eDP 1.4
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I²C, I²S, SPI, UART
Mechanical	100 mm x 80 mm x 29 mm

*Please refer to NVIDIA documentation for what is currently supported.

©2020 NVIDIA Corporation. All rights reserved. NVIDIA, the NVIDIA logo, CUDA, Jetson, Jetson Nano, Maxwell, NVIDIA JetPack, and TensorRT are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated. ARM, AMBA and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc; its operating company, ARM Limited; and the regional subsidiaries, ARM Inc.; ARM KK; ARM Korea Limited.; ARM Taiwan Limited; ARM France SAS; ARM Consulting [Shanghai] Co. Ltd.; ARM Germany GmbH; ARM Embedded Technologies Pvt. Ltd.; ARM Norway AS and ARM Sweden AB. JAN20



B.2 Jetson Orin NX 8GB



The most advanced AI computer for smaller, lower-power autonomous machines.

NVIDIA® Jetson Orin™ NX series modules deliver up to 100 TOPS of AI performance in the smallest Jetson form-factor, with power configurable between 10W and 25W. This gives you 3X the performance of NVIDIA Jetson AGX Xavier™ and 5X the performance of Jetson Xavier™ NX, making it ideal for small form-factor, low-power products like drones and handheld devices.

These system-on-modules support multiple concurrent AI application pipelines with an NVIDIA Ampere architecture GPU, next-generation deep learning and vision accelerators, high-speed IO, and fast memory bandwidth. Now, you can develop solutions using your largest and most complex AI models to solve problems such as natural language understanding, 3D perception, and multi-sensor fusion.

Jetson runs the NVIDIA AI software stack, and use case-specific application frameworks are available, including NVIDIA Isaac™ for robotics, DeepStream for vision AI, and Riva for conversational AI. You can also save significant time with NVIDIA Omniverse™ Replicator for synthetic data generation (SDG), and with NVIDIA TAO Toolkit for fine-tuning pretrained AI models from the NGC™ catalog.

Jetson ecosystem partners offer additional AI and system software, developer tools, and custom software development. They can also help with cameras and other sensors, as well as carrier boards and design services for your product.

Jetson Orin modules are unmatched in performance and efficiency for robots and other autonomous machines, and they give you the flexibility to create the next generation of AI solutions with the latest NVIDIA GPU technology. Together with the world-standard NVIDIA AI software stack and an ecosystem of services and products, your road to market has never been faster.

Key Features

Jetson Orin NX 8GB

- > 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores
- > 1x NVDLA v2.0
- > 6-core Arm® Cortex®-A78AE v8.2 64-bit CPU
- > 8GB 128-bit LPDDR5
- > PVA v2.0

Jetson Orin NX 16GB

- > 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores
- > 2x NVDLA v2.0
- > 8-core Arm® Cortex®-A78AE v8.2 64-bit CPU
- > 16GB 128-bit LPDDR5
- > PVA v2.0

Power

- > Voltage input 5V-20V
- > Module Power: 10W-20W

Power

- > Voltage input 5V-20V
- > Module Power: 10W-25W

P3767 | NVIDIA JETSON ORIN NX SERIES | Datasheet

NVIDIA JETSON ORIN NX SERIES MODULES**TECHNICAL SPECIFICATIONS**

	JETSON ORIN NX 8GB	JETSON ORIN NX 16GB
AI Performance	70 TOPS (INT8)	100 TOPS (INT8)
GPU	NVIDIA Ampere architecture with 1024 NVIDIA CUDA® cores and 32 Tensor Cores	
Max GPU Freq	765MHz	918GHz
CPU	6-core Arm® Cortex®-A78AE v8.2 64-bit CPU 1.5MB L2 + 4MB L3	8-core Arm® Cortex®-A78AE v8.2 64-bit CPU 2MB L2 + 4MB L3
CPU Max Freq		2GHz
DL Accelerator [DLA]	1x NVDLA v2.0	2x NVDLA v2.0
DLA Max Frequency		614MHz
Vision Accelerator		PVA v2.0
Memory	8GB 128-bit LPDDR5 102.4GB/s	16GB 128-bit LPDDR5 102.4GB/s
Storage		Supports external NVMe
CSI Camera		Up to 4 cameras (8 via virtual channels*) 8 MIPI CSI-2 lanes D-PHY 1.2 (20Gbps)
Video Encode		1x 4K60 3x 4K30 6x 1080p60 12x 1080p30 (H.265) H.264, AV1
Video Decode		1x 8K30 2x 4K60 4x 4K30 9x 1080p60 18x 1080p30 (H.265) H.264, VP9, AV1
UPHY		3 x1 + 1 x4 PCIe Gen 4 3x USB 3.2 Gen2
Networking		1x GbE
Display		1x 8K60 multi-mode DP 1.4a (+MST)/eDP 1.4a/HDMI 2.1
Other I/O		3x USB 2.0 3x UART 2x SPI 4x I²C 1x CAN DMIC DSPK 2x I²S 15x GPIOs
Power	10W-20W	10W-25W
Mechanical		69.6mm x 45mm 260-pin SO-DIMM connector

* Virtual channel-related camera information for Jetson Orin NX is not final and subject to change.

Refer to the Software Features section of the latest NVIDIA Jetson Linux Developer Guide for a list of supported features.

[Learn more](#)

Learn more at www.nvidia.com/jetson-orin

© 2022 NVIDIA Corporation. All rights reserved. NVIDIA, the NVIDIA logo, CUDA, Jetson, Jetson Xavier NX, NVIDIA NGC, NVIDIA Clara, NVIDIA Isaac, NVIDIA JetPack, and NVIDIA Jetson Orin NX are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. ARM, AMBA and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc, its operating company ARM Limited, and the regional subsidiaries ARM Inc., ARM KK, ARM Korea Limited., ARM Taiwan Limited, ARM France SAS, ARM Consulting (Shanghai) Co. Ltd., ARM Germany GmbH, ARM Embedded Technologies Pvt. Ltd., ARM Norway, AS and ARM Sweden AB. Other company and product names may be trademarks of the respective companies with which they are associated. APR22



B.3 Velodyne Puck

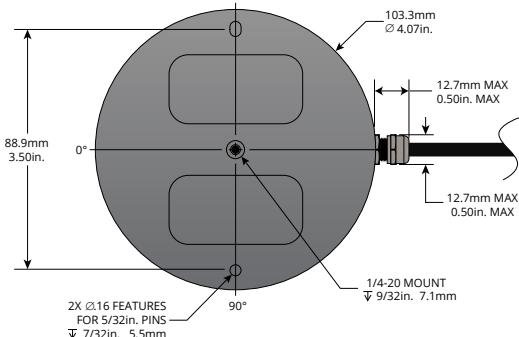
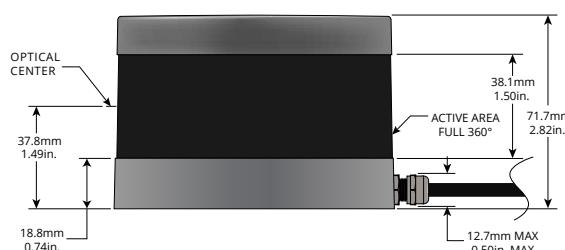
The page features the Velodyne Lidar logo and the word "Puck™". Below it is the text "VERSATILE REAL-TIME LIDAR SENSOR". To the right is a photograph of a silver car with the Puck sensor mounted on its front bumper. Below the car are six icons representing different applications: Automotive, Robotics, Mapping, UAV, Security, and Industrial.

Puck

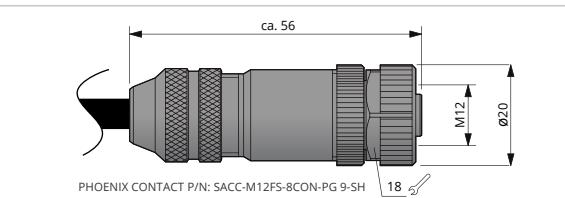
Velodyne Lidar's Puck is a small and compact lidar that is performance and power optimized for usage across a variety of applications ranging from automotive, mapping, robotics, security, smart cities and more. The Puck is attractively priced and built on the foundations of Velodyne's leadership in lidar, enabling real-time, surround view, 3D distance and calibrated reflectivity measurements.

The Puck has a range of 100 m and generates up to ~600,000 points/second, across a 360° horizontal field of view and a 30° vertical field of view. It uses proven, Class 1 eye-safe 905 nm technology with substantial autonomous fleet validation, making the Puck a sensor of choice for lower speed autonomous vehicle (AV) applications. The Puck has best-in-class power, which enables operation over a wide temperature range. Its use of off-the-shelf components enables enhanced scalability and attractive volume pricing. Like other Velodyne sensors, the Puck has world-class technical support available across North America, Europe & Asia from the world's leading lidar company.

DIMENSIONS (Subject to change)



M12 CONNECTOR OPTION



For other connector options contact
Velodyne Sales (sales@velodyne.com)

velodynelidar.com

Puck**Real-Time Lidar Sensor**

The Puck provides high definition 3-dimensional information about the surrounding environment.



Specifications:	
Sensor:	<ul style="list-style-type: none"> • 16 Channels • Measurement Range: 100 m • Range Accuracy: Up to ± 3 cm (Typical)¹ • Field of View (Vertical): +15.0° to -15.0° (30°) • Angular Resolution (Vertical): 2.0° • Field of View (Horizontal): 360° • Angular Resolution (Horizontal/Azimuth): 0.1° – 0.4° • Rotation Rate: 5 Hz – 20 Hz • Integrated Web Server for Easy Monitoring and Configuration
Laser:	<ul style="list-style-type: none"> • Laser Product Classification: Class 1 Eye-safe per IEC 60825-1:2007 & 2014 • Wavelength: 903 nm
Mechanical/ Electrical/ Operational	<ul style="list-style-type: none"> • Power Consumption: 8 W (Typical)² • Operating Voltage: 9 V – 18 V (with Interface Box and Regulated Power Supply) • Weight: ~830 g (without Cabling and Interface Box) • Dimensions: See diagram on previous page • Environmental Protection: IP67 • Operating Temperature: -10°C to +60°C³ • Storage Temperature: -40°C to +105°C
Output:	<ul style="list-style-type: none"> • 3D Lidar Data Points Generated: <ul style="list-style-type: none"> - Single Return Mode: ~300,000 points per second - Dual Return Mode: ~600,000 points per second • 100 Mbps Ethernet Connection • UDP Packets Contain: <ul style="list-style-type: none"> - Time of Flight Distance Measurement - Calibrated Reflectivity Measurement - Rotation Angles - Synchronized Time Stamps (μs resolution) • GPS: \$GPRMC and \$GPGGA NMEA Sentences from GPS Receiver (GPS not included)

63-9229 Rev-K VLP-16

For more details and ordering information, contact Velodyne Sales (sales@velodyne.com)

1. Typical accuracy refers to ambient wall test performance across most channels and may vary based on factors including but not limited to range, temperature and target reflectivity.
2. Operating power may be affected by factors including but not limited to range, reflectivity and environmental conditions.
3. Operating temperature may be affected by factors including but not limited to air flow and sun load.



CLASS 1 LASER PRODUCT

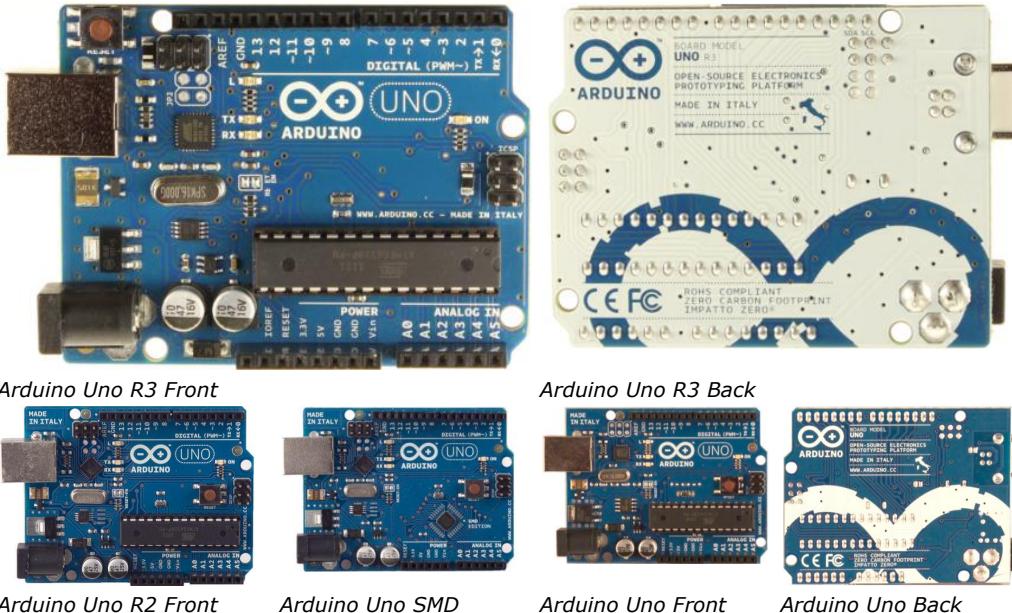
Copyright ©2019 Velodyne Lidar, Inc. Specifications are subject to change. Other trademarks or registered trademarks are property of their respective owners.

Velodyne Lidar, Inc. 5521 Hellyer Ave, San Jose, CA 95138 / lidar@velodyne.com / 408.465.2800

velodynelidar.com

B.4 Arduino Uno

Arduino Uno



Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

| [Revision 2](#) of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#).

| [Revision 3](#) of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the `analogWrite()` function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

B.5 Battery

The technical data for the 5050mAh XP50503HED Dualsky battery is shown in Figure B.1.



XP5050-HED

Specification 参数

5050mAh

POWERED BY DUALSKY

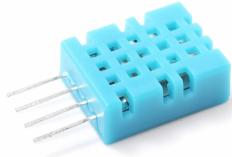
NAME 型号	XP50502HED	XP50503HED	XP50504HED	XP50505HED	XP50506HED
SKU 商品编码	31835	31836	31837	31838	31839
Capacity(mAh) 容量	5050				
Voltage(V) 电压	7.4	11.1	14.8	18.5	22.2
Energy(Wh) 电量	37.4	56.1	74.7	93.4	112.1
Number of Cell 电池节数	2S1P	3S1P	4S1P	5S1P	6S1P
Dimensions L x W x T(mm) 尺寸	140x44x18	140x44x27	140x44x36	140x44x45	140x44x54
Weight (g) 重量	245	355	465	578	688
Wiring & Connector 线材&插头	12AWG (XT60)				
Max Charge 充电C数	5C				
Max Charge Current (A) 最大充电电流	25				
Max Cont. Discharge 放电C数	50C				
Max Cont. Current (A) 最大放电电流	253				
Max Cont. Output (W) 最大放电功率	530	795	1061	1326	1591
Aerobatic Models 特技飞机	2.7kg	4.0kg	5.3kg	6.6kg	8.0kg
Extreme 3D Models 3D机	1.8kg	2.7kg	3.5kg	4.4kg	5.3kg
Applications 应用级别	Surface models 船模和越野车模型爱用	Off-road cars	Drones 轻量长航时无人机较多选择	2x FAI F3A	90E 3D Airplane, 90mm EDF or 2x 700E 3D Helicopter 90-120级标准动力 (5-6S) 10S版本为F3A标准动力组, 推荐使用

Figure B.1: Specifications for the Dualsky 11.1V 3-cell LiPo battery used in 2022 prototype (Dualsky 2022)

B.6 Temperature & Humidity Sensor



Your specialist in innovating humidity & temperature sensors



Digital relative humidity & temperature sensor DHT11

1. Feature & Application:

- *Good precision
- *Resistive type
- *Full range temperature compensated
- *Relative humidity and temperature measurement
- *Calibrated digital signal
- *Outstanding long-term stability
- *Extra components not needed
- *Long transmission distance, up to 100 meters
- *Low power consumption
- *4 pins packaged and fully interchangeable

2. Description:

DHT11 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(100m) enable DHT11 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT11	
Power supply	3.3-5.5V DC	
Output signal	digital signal via Aosong 1-wire bus	
Sensing element	Polymer humidity resistor	
Operating range	humidity 20-90%RH;	temperature 0~50Celsius
Accuracy	humidity +5%RH;	temperature +2Celsius
Resolution or sensitivity	humidity 1%RH;	temperature 1Celsius
Repeatability	humidity +/-2%RH;	temperature +/-1Celsius
Humidity hysteresis	+/-1%RH	
Long-term Stability	+/-1%RH/year	
Interchangeability	fully interchangeable	

4.Dimensions: (unit---mm)

- 1 -

Aosong Electronics Co., Ltd.

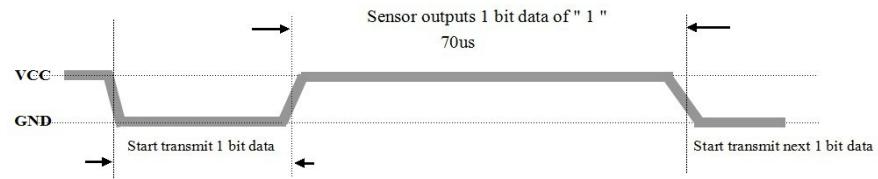
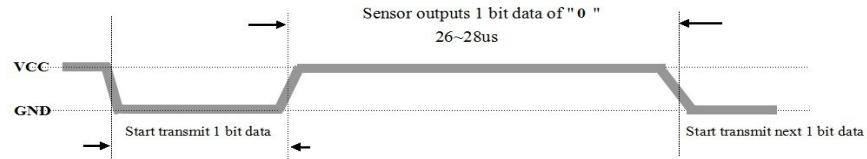
<http://www-aosong.com>

Thomas Liu (Sales Manager)

Email: thomasliu198518@aliyun.com , sales@aosong.com

AOSONG

Your specialist in innovating humidity & temperature sensors



Attention:

If signal from DHT11 is always high-voltage-level, it means DHT11 is not working properly, please check the electrical connection status.

7. Electrical Characteristics:

Items	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

8. Error and sources of error:

Measure values maybe influenced by follow factors:

Humidity errors

Equilibration time too short, steam, water sprays, dripping water or condensation at the sensor, etc.

Temperature errors

Equilibration time too short, cold or hot outside wall, sunlights, heating elements, etc.

B.7 Radio Frequency (RF) Module

PRODUCT SPECIFICATION



Single chip 433/868/915 MHz Transceiver nRF905

FEATURES

- True single chip GFSK transceiver in a small 32-pin package (32L QFN 5x5mm)
- ShockBurst™ mode for low power operation
- Power supply range 1.9 to 3.6 V
- Multi channel operation – ETSI/FCC Compatible
- Channel switching time <650µs
- Extremely low cost Bill of Material (BOM)
- No external SAW filter
- Adjustable output power up to 10dBm
- Carrier detect for "listen before transmit" protocols
- Data Ready signal when a valid data package is received or transmitted
- Address Match for detection of incoming package
- Automatic retransmission of data packages
- Automatic CRC and preamble generation
- Low supply current (TX), typical 11mA @ -10dBm output power
- Low supply current (RX), typical 12.5mA

APPLICATIONS

- Wireless data communication
- Alarm and security systems
- Home Automation
- Remote control
- Surveillance
- Automotive
- Telemetry
- Industrial sensors
- Keyless entry
- Toys

GENERAL DESCRIPTION

nRF905 is a single-chip radio transceiver for the 433/868/915 MHz ISM band. The transceiver consists of a fully integrated frequency synthesiser, receiver chain with demodulator, a power amplifier, a crystal oscillator and a modulator. The ShockBurst™ feature automatically handles preamble and CRC. Configuration is easily programmable by use of the SPI interface. Current consumption is very low, in transmit only 11mA at an output power of -10dBm, and in receive mode 12.5mA. Built in power down modes makes power saving easily realizable.

QUICK REFERENCE DATA

Parameter	Value	Unit
Minimum supply voltage	1.9	V
Maximum transmit output power	10	dBm
Transmitted data rate (Manchester-encoder embedded)	100	kbps
Supply current in transmit @ -10dBm output power	11	mA
Supply current in receive mode	12.5	mA
Temperature range	-40 to +85	°C
Typical Sensitivity	-100	dBm
Supply current in power down mode	2.5	µA

Table 1 nRF905 quick reference data.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

ORDERING INFORMATION

Type Number	Description	Version
nRF905 IC	32L QFN 5x5mm	-
nRF905-EVKIT 433	Evaluation kit 433MHz	1.0
nRF905-EVKIT 868/915	Evaluation kit 868/915MHz	1.0

Table 2 nRF905 ordering information.

BLOCK DIAGRAM

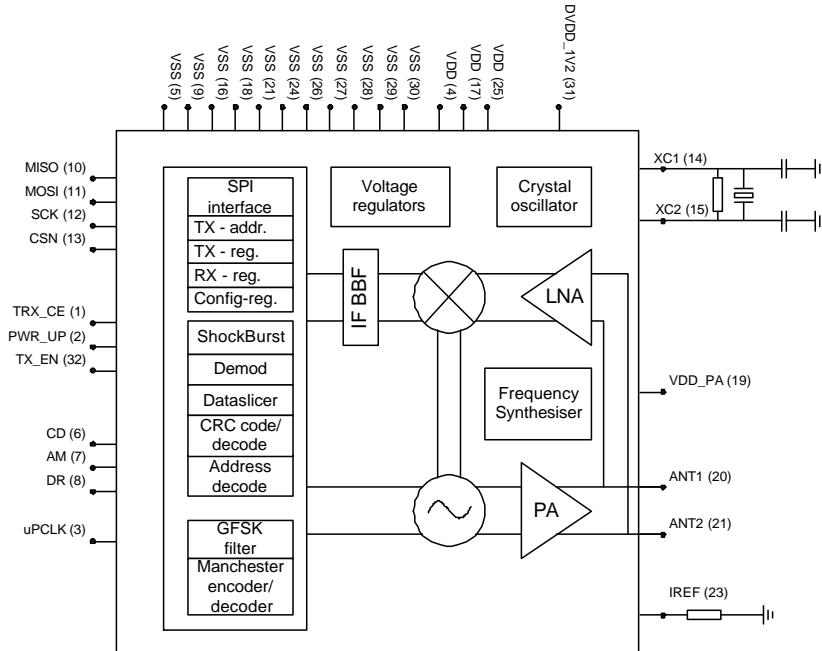


Figure 1 nRF905 with external components.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

PIN FUNCTIONS

Pin	Name	Pin function	Description
1	TRX_CE	Digital input	Enables chip for receive and transmit
2	PWR_UP	Digital input	Power up chip
3	uPCLK	Clock output	Output clock, divided crystal oscillator full-swing clock
4	VDD	Power	Power supply (+3V DC)
5	VSS	Power	Ground (0V)
6	CD	Digital output	Carrier Detect
7	AM	Digital output	Address Match
8	DR	Digital output	Receive and transmit Data Ready
9	VSS	Power	Ground (0V)
10	MISO	SPI - interface	SPI output
11	MOSI	SPI - interface	SPI input
12	SCK	SPI - Clock	SPI clock
13	CSN	SPI - enable	SPI enable, active low
14	XC1	Analog Input	Crystal pin 1/ External clock reference pin
15	XC2	Analog Output	Crystal pin 2
16	VSS	Power	Ground (0V)
17	VDD	Power	Power supply (+3V DC)
18	VSS	Power	Ground
19	VDD_PA	Power output	Positive supply (1.8V) to nRF905 power amplifier
20	ANT1	RF	Antenna interface 1
21	ANT2	RF	Antenna interface 2
22	VSS	Power	Ground (0V)
23	IREF	Analog Input	Reference current
24	VSS	Power	Ground (0V)
25	VDD	Power	Power supply (+3V DC)
26	VSS	Power	Ground (0V)
27	VSS	Power	Ground (0V)
28	VSS	Power	Ground (0V)
29	VSS	Power	Ground (0V)
30	VSS	Power	Ground (0V)
31	DVDD_1V2	Power	Low voltage positive digital supply output for de-coupling
32	TX EN	Digital input	TX_EN="1" TX mode, TX_EN="0" RX mode

Table 3 nRF905 pin function.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

PIN ASSIGNMENT

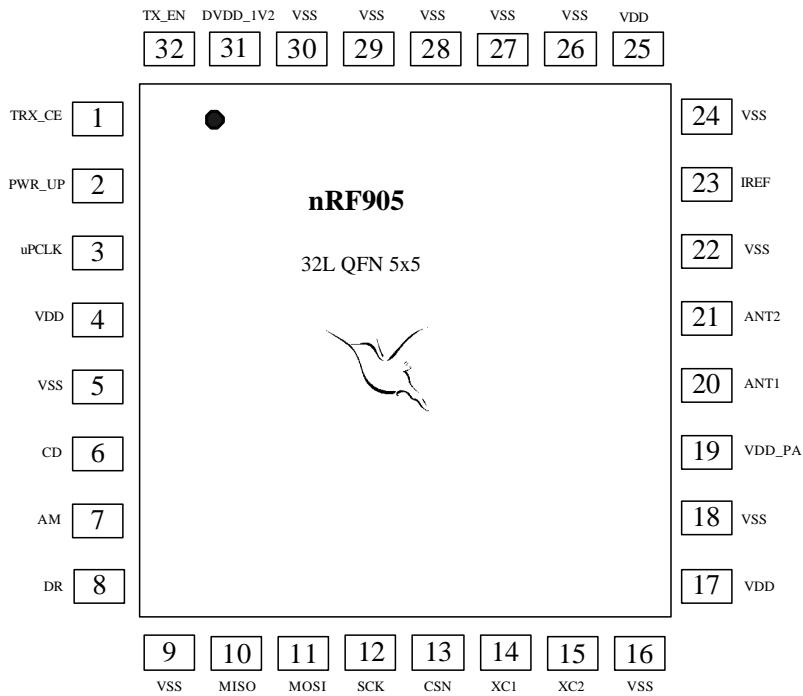


Figure 2 nRF905 pin assignment (top view) for a 32L QFN 5x5 package.

PRODUCT SPECIFICATION



nRF905 Single Chip 433/868/915 MHz Radio Transceiver

ELECTRICAL SPECIFICATIONS

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
Operating conditions						
VDD	Supply voltage		1.9	3.0	3.6	V
TEMP	Operating temperature		-40	27	85	°C
Digital input pin						
V _{IH}	HIGH level input voltage		VDD-0.3		VDD	V
V _{IL}	LOW level input voltage		VSS		0.3	V
Digital output pin						
V _{OH}	HIGH level output voltage (I _{OH} =-0.5mA)		VDD-0.3		VDD	V
V _{OL}	LOW level output voltage (I _{OL} =0.5mA)		VSS		0.3	V
General electrical specification						
I _{stby_eclk}	Supply current in standby, uCLK enabled	1)		100		µA
I _{stby_dclk}	Supply current in standby, uCLK disabled	2)		12.5		µA
I _{PD}	Supply current in power down mode			2.5		µA
I _{SPI}	Supply current in SPI programming	3)		20		µA
General RF conditions						
f _{OP}	Operating frequency	4)	430		928	MHz
f _{XTAL}	Crystal frequency	5)	4		20	MHz
Δf	Frequency deviation		±42	±50	±58	kHz
R _{GFSK}	GFSK data rate, Manchester-encoded			100		kbps
f _{CH433}	Channel spacing for 433MHz band			100		kHz
f _{CH868/915}	Channel spacing for 868/915MHz band			200		kHz
Transmitter operation						
P _{RF10}	Output power 10dBm setting	6)	7	10	11	dBm
P _{RF6}	Output power 6dBm setting	6)	3	6	9	dBm
P _{RF-2}	Output power -2dBm setting	6)	-6	-2	2	dBm
P _{RF-10}	Output power -10dBm setting	6)	-14	-10	-6	dBm
P _{BW}	20dB bandwidth for modulated carrier			190		kHz
P _{RF1}	1 st adjacent channel transmit power	7)		-27		dBc
P _{RF2}	2 nd adjacent channel transmit power	7)		-54		dBc
I _{TX10dBm}	Supply current @ 10dBm output power			30		mA
I _{TX-10dBm}	Supply current @ -10dBm output power			11		mA
Receiver operation						
I _{RX}	Supply current in receive mode			12.5		mA
RX _{SENS}	Sensitivity at 0.1%BER			-100		dBm
RX _{MAX}	Maximum received signal		0			dBm
C/I _{CO}	C/I Co-channel	8)		13		dB
C/I _{1ST}	1 st adjacent channel selectivity C/I 200kHz	8)		-7		dB
C/I _{2ND}	2 nd adjacent channel selectivity C/I 400kHz	8)		-16		dB
C/I _{IM}	Image rejection	8)		-30		dB

Table 4 nRF905 electrical specifications.

1) Output frequency is 4MHz load of external clock pin is 5pF, Crystal is 4MHz.

2) Crystal is 4MHz.

3) Chip in power down, SPI_SCK frequency is 1MHz.

4) Operates in the 433, 868 and 915 MHz ISM band.

5) The crystal frequency may be chosen from 5 different values (4, 8, 12, 16, and 20MHz).

6) Optimum load impedance, please see peripheral RF information.

7) Channel width and channel spacing is 200kHz.

8) Channel Level +3dB over sensitivity, interfering signal a standard CW, image lies 2MHz above wanted.

B.8 WiFi Module



Description

This WiFi card, based on Intel 8265AC chip, is specially designed for Jetson Nano. It supports 4.2 Bluetooth, 2.4G and 5G dual-band WiFi. The transmission speed can reach up to 867Mbps. Coupled with external 3db large gain dual antenna, even if in the complex wireless environment with multi-interference, it can also have good wireless penetration and wider coverage, making data reception smoother and signal transmission farther. The antenna supports 90-degree rotation. You can adjust the angle of antenna to improve signal sensitivity according to the actual use environment and signal strength. Give full play to the performance of this WiFi card, and enjoy high network speed at any time.



This WiFi module can be directly plugged into the WiFi port reserved on Jetson Nano board, easy to use.

Installation

1. Remove the two fixing screws on the Jetson Nano heatsink module, release the side latches located on either side of the module to remove it.



2. Insert the WiFi card into the card slot, and seat the card. Then reinstall the heatsink module onto the carrier board.



SPECIFICATION

- ## **SPECIFICATION**

- Network Interface: NGFF (m.2)
- WiFi Frequency Band: 2.4GHz-5GHz
- Standard Protocol: IEEE 802.11a/b/g/n/ac
- Wireless Speed: 300Mbps+867Mbps
- WiFi Connector: IPEX
- Bluetooth: 4.2
- Compatible with Linux, Windows 10/8.1/8/7
- Dimension: 22*30*2.4mm/0.87*1.18*0.09"

Customer Reviews

Peter S.

★★★★★ Verified Purchase

WiFi Module for Jetson Nano

Reviewed in QLD, Australia on 12 Jul 2021

I give the WiFi Module for Jetson Nano a rating of 5 stars!

[Show more](#)

Ask us about the WiFi Module for Jetson Nano

 Call (1300 240 817) >

 Email (help@littlebird.com.au) >

 © 2023 Little Bird Electronics Pty Ltd.
Made with ❤️ in SYD. All prices inc GST. ABN 15 634 521 449. We're  @lbhq on Twitter.

[Powered by Koi](#) [About Us](#) [Privacy Policy](#) [Terms of Service](#)

B.9 Servo Motors

ROBOTIS e-Manual

DYNAMIXEL DYNAMIXEL EDUCATIONAL SOFTWARE PARTS FAQ

Enter Search Terms

2XC430-W250

- 1. Specifications >
- 2. Control Table >
- 3. How to Assemble >
- 4. Reference >

2XC430-W250

[Edit on GitHub](#)

1. Specifications

2XC430-W250 is a ground breaking DYNAMIXEL that allows to control 2 axis(2 DOF) with a single module. In order to control 2 axis at the same time, each axle should be assigned with different ID while sharing an identical [Baud rate\(8\)](#). Since the Control Table for each axle is separated except the Baudrate, **2XC430-W250** can be applied in various applications.
The usage is identical to other DYNAMIXEL's, but be aware that Firmware Recovery will reset both axis to factory settings.

Item	Specifications
MCU	ARM CORTEX-M3 (72 [MHz], 32Bit)
Position Sensor	Contactless absolute encoder (12Bit, 360 [°]) Maker : ams(www.ams.com), Part No : AS5601
Motor	Coreless
Baud Rate	9,600 [bps] ~ 4.5 [Mbps]
Control Algorithm	PID control
Resolution	4096 [pulse/rev]
Operating Modes	Velocity Control Mode Position Control Mode (0 ~ 360 [°]) Extended Position Control Mode (Multi-turn) PWM Control Mode (Voltage Control Mode)
Weight	102 [g]
Dimensions (W x H x D)	36 x 46.5 x 36 [mm]
Gear Ratio	257.4 : 1
Stall Torque	1.3 [N.m] (at 9.0 [V], 1.1 [A]) 1.6 [N.m] (at 11.1 [V], 1.3 [A]) 1.8 [N.m] (at 12.0 [V], 1.4 [A])
No Load Speed	48 [rev/min] (at 9.0 [V]) 59 [rev/min] (at 11.1 [V]) 64 [rev/min] (at 12.0 [V])
Operating Temperature	-5 ~ +72 [°C]
Input Voltage	6.5 ~ 14.8 [V] (Recommended : 12.0 [V])
Command Signal	Digital Packet
Physical Connection	TTL Multidrop Bus (5V Logic) TTL Half Duplex Asynchronous Serial Communication (8bit, 1stop, No Parity)
ID	253 ID (0 ~ 252)
Feedback	Position, Velocity, Load, Realtime tick, Trajectory, Temperature, Input Voltage, etc
Case Material	Engineering Plastic(Front, Middle, Back)
Gear Material	Full Metal Gear
Standby Current	49 [mA]

DANGER
(May cause serious injury or death)

- Never place items containing water, flammables, and solvents near product.
- Never place fingers, arms, toes, and other body parts near product during operation.
- Cut power off if product emits strange odors or smoke.
- Keep product out of reach of children.
- Check the power's polarity before wiring.

CAUTION
(May cause injury or damage to product)

ROBOTIS e-Manual

DYNAMIXEL SYSTEM EDUCATIONAL KITS SOFTWARE PARTS FAQ

Enter Search Terms

[Edit on GitHub](#)

XC430-W240-T

1. Specifications >
2. Control Table >
3. How to Assemble >
4. Reference >

XC430-W240

1. Specifications

Item	Specifications
MCU	ARM CORTEX-M3 (72 [MHz], 32Bit)
Position Sensor	Contactless absolute encoder (12bit, 360 [°]) Maker : ams(www.ams.com), Part No : AS5601
Motor	Coreless
Baud Rate	9,600 [bps] ~ 4.5 [Mbps]
Control Algorithm	PID control
Resolution	4096 [pulse/rev]
Operating Modes	Velocity Control Mode Position Control Mode (0 ~ 360 [°]) Extended Position Control Mode (Multi-turn) PWM Control Mode (Voltage Control Mode)
Weight	65 [g]
Dimensions (W x H x D)	28.5 x 46.5 x 34 [mm]
Gear Ratio	245.22 : 1
Stall Torque	1.4 [N.m] (at 9.0 [V], 1.1 [A]) 1.7 [N.m] (at 11.1 [V], 1.3 [A]) 1.9 [N.m] (at 12.0 [V], 1.4 [A])
No Load Speed	52 [rev/min] (at 9.0 [V]) 65 [rev/min] (at 11.1 [V]) 70 [rev/min] (at 12.0 [V])
Operating Temperature	-5 ~ +80 [°C]
Input Voltage	6.5 ~ 14.8 [V] (Recommended : 12.0 [V])
Command Signal	Digital Packet
Physical Connection	TTL Level Multidrop Bus Half Duplex Asynchronous Serial Communication (8bit, 1stop, No Parity)
ID	253 ID (0 ~ 252)
Feedback	Position, Velocity, Load, Realtime tick, Trajectory, Temperature, Input Voltage, etc
Case Material	Engineering Plastic (Body)
Gear Material	Full Metal Gear
Standby Current	46 [mA]

DANGER
(May cause serious injury or death)

- Never place items containing water, flammables, and solvents near product.
- Never place fingers, arms, toes, and other body parts near product during operation.
- Cut power off if product emits strange odors or smoke.
- Keep product out of reach of children.
- Check the power's polarity before wiring.

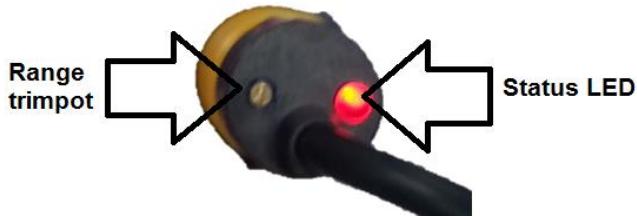
CAUTION
(May cause injury or damage to product)

- Do not operate the product at a temperature exceeding -5 ~ +80 [°C] range.
- Do not insert sharp blades nor pins during product operation.

B.10 Proximity Sensors

ZD1906 IR Adjustable Proximity Sensor

ZD1906 is an IR proximity sensor which can operate over an adjustable range of 3-80cm (depending on object reflectiveness) and has an active low output which can sink up to 100mA. It is fitted with two plastic nuts, and so can be mounted in a panel up to 30mm thick.



The range trimpot can be used to adjust the trigger sensitivity. Clockwise corresponds to greater sensitivity and detection range. The status LED illuminates when the sensor is triggered and the output is active.



The red end is the sensor and has an emitter and receiver behind an IR filter. The field of view is 15 degrees.

Diameter	18mm
Length	45mm
Supply Voltage	5V nominal
Brown Wire	5V input
Blue Wire	Ground connection
Black Wire	Signal

The output is NPN, so it is floating when inactive and pulled to GND when active. To use as an input to a logic level device, connect a pullup resistor to the Signal wire, or enable pullups on the microcontroller input pin, eg for Arduino:

```
pinMode(SIGNALPIN, INPUT_PULLUP);
```

To read the pin, use:

```
state=digitalRead(SIGNALPIN);
```

In this case, state will be 0 when active and 1 when inactive.

B.11 U2D2 Motor Board

ROBOTS e-Manual

DYNAMIXEL DYNAMIXEL EDUCATIONAL SOFTWARE PARTS FAQ

Enter Search Terms

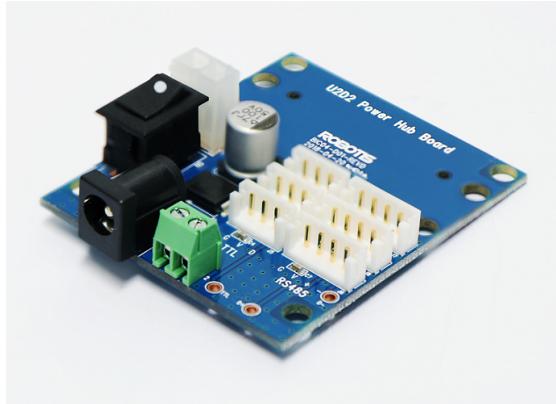
[Edit on GitHub](#)

U2D2 Power Hub

- 1. Introduction
- 2. Specifications
- 3. Layout
- 4. Cautions
- 5. How to Assemble
- 6. Connector Information

1. Introduction

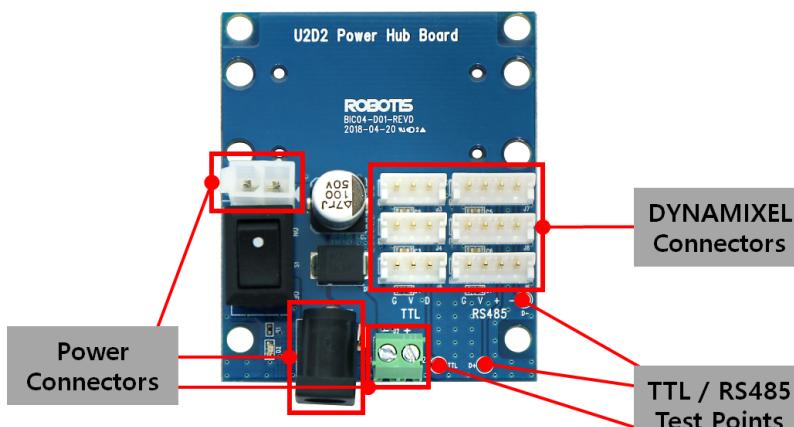
Combined with U2D2, it can supply various kinds of external power supply to supply stable power to DYNAMIXEL.



2. Specifications

Item	Details
Operating Voltage	3.5V ~ 24.0V
Maximum Current	10.0A

3. Layout



3.1. Description

- DYNAMIXEL Connectors:** 3 and 4-pins JST connectors are installed for communication. See [Connector Information](#) for more details.
- TTL/RS485 Test Points:** TTL data line, RS485 D+, D-line signals can be easily identified via the test points
- Power Connectors:** U2D2 PHB provides three different types of power connectors to supply a diverse voltage and current to DYNAMIXEL.
 - SMPS DC Connector (2.5mm ID / 5.5mm OD, **Center Positive**)
 - MOLEX Power connector (See [Connector Information](#) for more details)
 - PCB Screw Terminal Block Connector
 - Left: -
 - Right: +

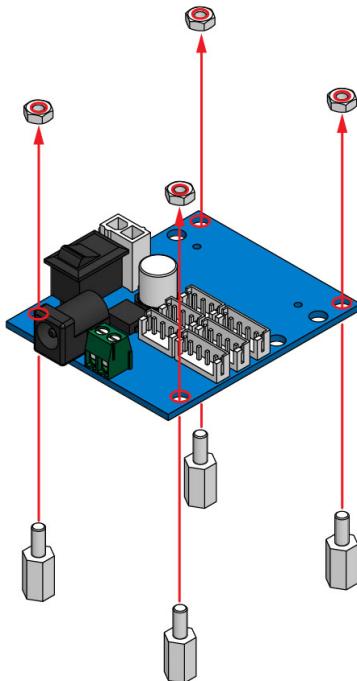
CAUTION:

**U2D2 Power Hub**

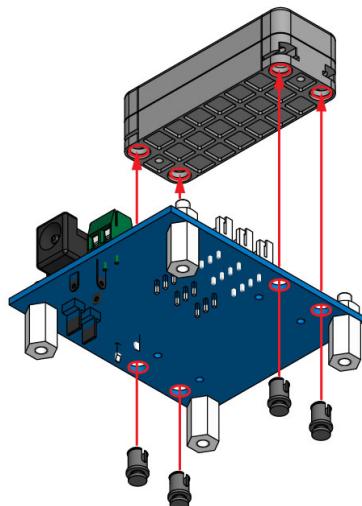
- 1. Introduction
- 2. Specifications
- 3. Layout
- 4. Cautions
- 5. How to Assemble
- 6. Connector Information

4. How to Assemble

1. Assemble four(4) plastic supporters at each corner of the **U2D2 Power Hub Board** as shown below.

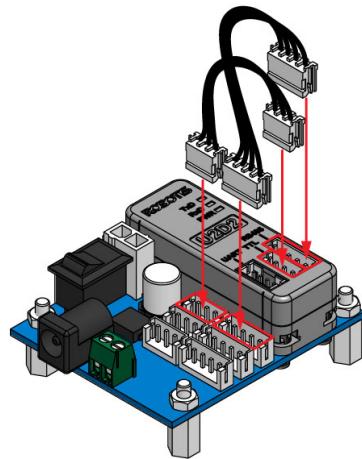


2. Use plastic rivets to mount **U2D2** on the **U2D2 Power Hub Board**.

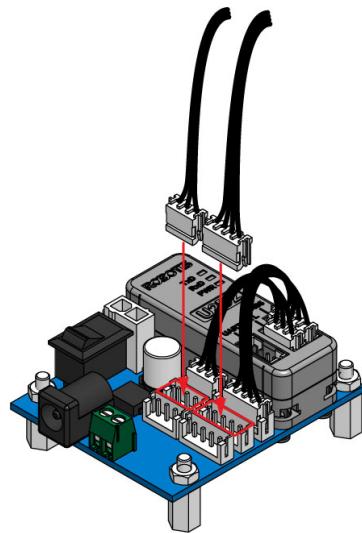


3. Connect **U2D2** and **U2D2 Power Hub Board** with 3P or 4P cables (Both 3P and 4P cables can be connected at the same time).

U2D2 Power Hub	
1.	Introduction
2.	Specifications
3.	Layout
4.	Cautions
5.	How to Assemble
6.	Connector Information



4. Connect additional cables that will connect **U2D2** and **DYNAMIXEL** (Depending on DYNAMIXEL model, converting cables may be required).



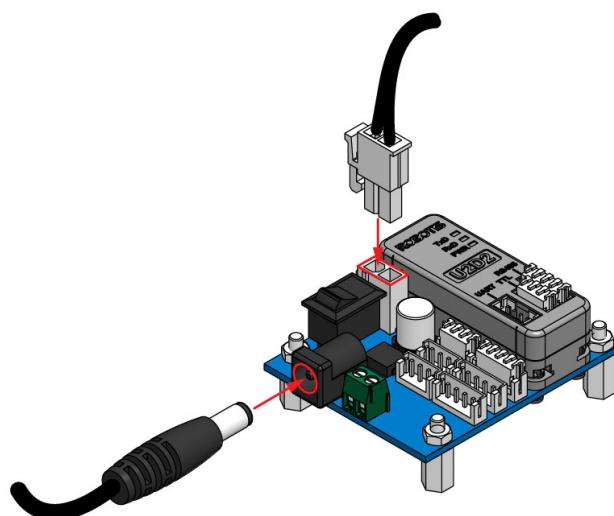
5. Connect the other side of the cable to **DYNAMIXEL** (Depending on DYNAMIXEL model, converting cables may be required).



6. Connect power to [U2D2 Power Hub Board](#).

DANGER : Do NOT use multiple power inputs at the same time. Select ONLY one of the power source to connect.

WARNING : Check the Recommended Voltage for DYNAMIXEL before supplying the power.



7. Turn the power switch on [U2D2 Power Hub Board](#) to supply power to DYNAMIXEL

C Preliminary Testing MATLAB Code

```
% CaveX Honours Project 2023
% Riley Groome, Tyler Groome & Luka Moran
% Preliminary Testing results

% angles corresponding to different blocks of wood
angles = [6.47, 7.65, 8.24, 8.83]; % deg [0 blocks, 2, 3 4]
% extract opt-track csv data
% Tripod Gait data
Tripod_Decline_0 = csvread("CaveX - Tripod - Decline Test (0 blocks of
wood).csv");
Tripod_Decline_2 = csvread("CaveX - Tripod - Decline Test (2 blocks of
wood).csv");
Tripod_Decline_3 = csvread("CaveX - Tripod - Decline Test (3 blocks of
wood).csv");
Tripod_Decline_4 = csvread("CaveX - Tripod - Decline Test (4 blocks of
wood).csv");

Tripod_Incline_0 = csvread("CaveX - Tripod - Incline Test (0 blocks of
wood).csv");
Tripod_Incline_2 = csvread("CaveX - Tripod - Incline Test (2 blocks of
wood).csv");
Tripod_Incline_3 = csvread("CaveX - Tripod - Incline Test (3 blocks of
wood).csv");
Tripod_Incline_4 = csvread("CaveX - Tripod - Incline Test (4 blocks of
wood).csv");

% Ripple Gait Data
Ripple_Decline_0 = csvread("CaveX - Ripple - Decline Test (0 blocks of
wood).csv");
Ripple_Decline_2 = csvread("CaveX - Ripple - Decline Test (2 blocks of
wood).csv");
Ripple_Decline_3 = csvread("CaveX - Ripple - Decline Test (3 blocks of
wood).csv");
Ripple_Decline_4 = csvread("CaveX - Ripple - Decline Test (4 blocks of
wood).csv");

Ripple_Incline_0 = csvread("CaveX - Ripple - Incline Test (0 blocks of
wood).csv");
Ripple_Incline_2 = csvread("CaveX - Ripple - Incline Test (2 blocks of
wood).csv");
Ripple_Incline_3 = csvread("CaveX - Ripple - Incline Test (3 blocks of
wood).csv");
Ripple_Incline_4 = csvread("CaveX - Ripple - Incline Test (4 blocks of
wood).csv");

% Wave Gait Data
Wave_Decline_0 = csvread("CaveX - Wave - Decline Test (0 blocks of
wood).csv");
Wave_Decline_2 = csvread("CaveX - Wave - Decline Test (2 blocks of
wood).csv");
Wave_Decline_3 = csvread("CaveX - Wave - Decline Test (3 blocks of
wood).csv");
```

```

Wave_Decline_4 = csvread("CaveX - Wave - Decline Test (4 blocks of
wood).csv");

Wave_Incline_0 = csvread("CaveX - Wave - Incline Test (0 blocks of
wood).csv");
Wave_Incline_2 = csvread("CaveX - Wave - Incline Test (2 blocks of
wood).csv");
Wave_Incline_3 = csvread("CaveX - Wave - Incline Test (3 blocks of
wood).csv");
Wave_Incline_4 = csvread("CaveX - Wave - Incline Test (4 blocks of
wood).csv");

% Decline Results for each gait and each angle
% work out average velocity for Tripod initial angle (0 blocks)
Time_Tripod_0 = Tripod_Decline_0(:,1); % s
X_Tripod_0 = Tripod_Decline_0(:,2); % m
Y_Tripod_0 = Tripod_Decline_0(:,3); % m
Z_Tripod_0 = Tripod_Decline_0(:,4); % m
delta_t_tripod_0 = Time_Tripod_0(length(Time_Tripod_0))-Time_Tripod_0(1); % s
delta_x_tripod_0 = X_Tripod_0(length(X_Tripod_0))-X_Tripod_0(1); % m
delta_y_tripod_0 = Y_Tripod_0(length(Y_Tripod_0))-Y_Tripod_0(1); % m
delta_z_tripod_0 = Z_Tripod_0(length(Z_Tripod_0))-Z_Tripod_0(1); % m
Vx_ave_Tripod_0 = delta_x_tripod_0/delta_t_tripod_0; % m/s [average x
velocity]
Vy_ave_Tripod_0 = delta_y_tripod_0/delta_t_tripod_0; % m/s [average x
velocity]
Vz_ave_Tripod_0 = delta_z_tripod_0/delta_t_tripod_0; % m/s [average x
velocity]
V_ave_tripod_0 =
sqrt((Vx_ave_Tripod_0)^2+(Vy_ave_Tripod_0)^2+(Vz_ave_Tripod_0)^2);

% work out average velocity for Tripod second angle (2 blocks)
Time_Tripod_2 = Tripod_Decline_2(:,1); % s
X_Tripod_2 = Tripod_Decline_2(:,2); % m
Y_Tripod_2 = Tripod_Decline_2(:,3); % m
Z_Tripod_2 = Tripod_Decline_2(:,4); % m
delta_t_tripod_2 = Time_Tripod_2(length(Time_Tripod_2))-Time_Tripod_2(1); % s
delta_x_tripod_2 = X_Tripod_2(length(X_Tripod_2))-X_Tripod_2(1); % m
delta_y_tripod_2 = Y_Tripod_2(length(Y_Tripod_2))-Y_Tripod_2(1); % m
delta_z_tripod_2 = Z_Tripod_2(length(Z_Tripod_2))-Z_Tripod_2(1); % m
Vx_ave_Tripod_2 = delta_x_tripod_2/delta_t_tripod_2; % m/s [average x
velocity]
Vy_ave_Tripod_2 = delta_y_tripod_2/delta_t_tripod_2; % m/s [average x
velocity]
Vz_ave_Tripod_2 = delta_z_tripod_2/delta_t_tripod_2; % m/s [average x
velocity]
V_ave_tripod_2 =
sqrt((Vx_ave_Tripod_2)^2+(Vy_ave_Tripod_2)^2+(Vz_ave_Tripod_2)^2);

% work out average velocity for Tripod third angle (3 blocks)
Time_Tripod_3 = Tripod_Decline_3(:,1); % s
X_Tripod_3 = Tripod_Decline_3(:,2); % m

```

```

Y_Tripod_3 = Tripod_Decline_3(:,3); % m
Z_Tripod_3 = Tripod_Decline_3(:,4); % m
delta_t_tripod_3 = Time_Tripod_3(length(Time_Tripod_3))-
Time_Tripod_3(1); % s
delta_x_tripod_3 = X_Tripod_3(length(X_Tripod_3))-X_Tripod_3(1); % m
delta_y_tripod_3 = Y_Tripod_3(length(Y_Tripod_3))-Y_Tripod_3(1); % m
delta_z_tripod_3 = Z_Tripod_3(length(Z_Tripod_3))-Z_Tripod_3(1); % m
Vx_ave_Tripod_3 = delta_x_tripod_3/delta_t_tripod_3; % m/s [average x
velocity]
Vy_ave_Tripod_3 = delta_y_tripod_3/delta_t_tripod_3; % m/s [average x
velocity]
Vz_ave_Tripod_3 = delta_z_tripod_3/delta_t_tripod_3; % m/s [average x
velocity]
V_ave_tripod_3 =
sqrt((Vx_ave_Tripod_3)^2+(Vy_ave_Tripod_3)^2+(Vz_ave_Tripod_3)^2);

% work out average velocity for Tripod fourth angle (4 blocks)
Time_Tripod_4 = Tripod_Decline_4(:,1); % s
X_Tripod_4 = Tripod_Decline_4(:,2); % m
Y_Tripod_4 = Tripod_Decline_4(:,3); % m
Z_Tripod_4 = Tripod_Decline_4(:,4); % m
delta_t_tripod_4 = Time_Tripod_4(length(Time_Tripod_4))-
Time_Tripod_4(1); % s
delta_x_tripod_4 = X_Tripod_4(length(X_Tripod_4))-X_Tripod_4(1); % m
delta_y_tripod_4 = Y_Tripod_4(length(Y_Tripod_4))-Y_Tripod_4(1); % m
delta_z_tripod_4 = Z_Tripod_4(length(Z_Tripod_4))-Z_Tripod_4(1); % m
Vx_ave_Tripod_4 = delta_x_tripod_4/delta_t_tripod_4; % m/s [average x
velocity]
Vy_ave_Tripod_4 = delta_y_tripod_4/delta_t_tripod_4; % m/s [average x
velocity]
Vz_ave_Tripod_4 = delta_z_tripod_4/delta_t_tripod_4; % m/s [average x
velocity]
V_ave_tripod_4 =
sqrt((Vx_ave_Tripod_4)^2+(Vy_ave_Tripod_4)^2+(Vz_ave_Tripod_4)^2);

% work out average velocity for Ripple initial angle (0 blocks)
Time_Ripple_0 = Ripple_Decline_0(:,1); % s
X_Ripple_0 = Ripple_Decline_0(:,2); % m
Y_Ripple_0 = Ripple_Decline_0(:,3); % m
Z_Ripple_0 = Ripple_Decline_0(:,4); % m
delta_t_ripple_0 = Time_Ripple_0(length(Time_Ripple_0))-
Time_Ripple_0(1); % s
delta_x_ripple_0 = X_Ripple_0(length(X_Ripple_0))-X_Ripple_0(1); % m
delta_y_ripple_0 = Y_Ripple_0(length(Y_Ripple_0))-Y_Ripple_0(1); % m
delta_z_ripple_0 = Z_Ripple_0(length(Z_Ripple_0))-Z_Ripple_0(1); % m
Vx_ave_Ripple_0 = delta_x_ripple_0/delta_t_ripple_0; % m/s [average x
velocity]
Vy_ave_Ripple_0 = delta_y_ripple_0/delta_t_ripple_0; % m/s [average x
velocity]
Vz_ave_Ripple_0 = delta_z_ripple_0/delta_t_ripple_0; % m/s [average x
velocity]
V_ave_ripple_0 =
sqrt((Vx_ave_Ripple_0)^2+(Vy_ave_Ripple_0)^2+(Vz_ave_Ripple_0)^2);

```

```
% work out average velocity for Ripple second angle (2 blocks)
Time_Ripple_2 = Ripple_Decline_2(:,1); % s
X_Ripple_2 = Ripple_Decline_2(:,2); % m
Y_Ripple_2 = Ripple_Decline_2(:,3); % m
Z_Ripple_2 = Ripple_Decline_2(:,4); % m
delta_t_ripple_2 = Time_Ripple_2(length(Time_Ripple_2))-Time_Ripple_2(1); % s
delta_x_ripple_2 = X_Ripple_2(length(X_Ripple_2))-X_Ripple_2(1); % m
delta_y_ripple_2 = Y_Ripple_2(length(Y_Ripple_2))-Y_Ripple_2(1); % m
delta_z_ripple_2 = Z_Ripple_2(length(Z_Ripple_2))-Z_Ripple_2(1); % m
Vx_ave_Ripple_2 = delta_x_ripple_2/delta_t_ripple_2; % m/s [average x velocity]
Vy_ave_Ripple_2 = delta_y_ripple_2/delta_t_ripple_2; % m/s [average x velocity]
Vz_ave_Ripple_2 = delta_z_ripple_2/delta_t_ripple_2; % m/s [average x velocity]
V_ave_ripple_2 =
sqrt((Vx_ave_Ripple_2)^2+(Vy_ave_Ripple_2)^2+(Vz_ave_Ripple_2)^2);

% work out average velocity for Ripple third angle (2 blocks)
Time_Ripple_3 = Ripple_Decline_3(:,1); % s
X_Ripple_3 = Ripple_Decline_3(:,2); % m
Y_Ripple_3 = Ripple_Decline_3(:,3); % m
Z_Ripple_3 = Ripple_Decline_3(:,4); % m
delta_t_ripple_3 = Time_Ripple_3(length(Time_Ripple_3))-Time_Ripple_3(1); % s
delta_x_ripple_3 = X_Ripple_3(length(X_Ripple_3))-X_Ripple_3(1); % m
delta_y_ripple_3 = Y_Ripple_3(length(Y_Ripple_3))-Y_Ripple_3(1); % m
delta_z_ripple_3 = Z_Ripple_3(length(Z_Ripple_3))-Z_Ripple_3(1); % m
Vx_ave_Ripple_3 = delta_x_ripple_3/delta_t_ripple_3; % m/s [average x velocity]
Vy_ave_Ripple_3 = delta_y_ripple_3/delta_t_ripple_3; % m/s [average x velocity]
Vz_ave_Ripple_3 = delta_z_ripple_3/delta_t_ripple_3; % m/s [average x velocity]
V_ave_ripple_3 =
sqrt((Vx_ave_Ripple_3)^2+(Vy_ave_Ripple_3)^2+(Vz_ave_Ripple_3)^2);

% work out average velocity for Ripple fourth angle (4 blocks)
Time_Ripple_4 = Ripple_Decline_4(:,1); % s
X_Ripple_4 = Ripple_Decline_4(:,2); % m
Y_Ripple_4 = Ripple_Decline_4(:,3); % m
Z_Ripple_4 = Ripple_Decline_4(:,4); % m
delta_t_ripple_4 = Time_Ripple_4(length(Time_Ripple_4))-Time_Ripple_4(1); % s
delta_x_ripple_4 = X_Ripple_4(length(X_Ripple_4))-X_Ripple_4(1); % m
delta_y_ripple_4 = Y_Ripple_4(length(Y_Ripple_4))-Y_Ripple_4(1); % m
delta_z_ripple_4 = Z_Ripple_4(length(Z_Ripple_4))-Z_Ripple_4(1); % m
Vx_ave_Ripple_4 = delta_x_ripple_4/delta_t_ripple_4; % m/s [average x velocity]
Vy_ave_Ripple_4 = delta_y_ripple_4/delta_t_ripple_4; % m/s [average x velocity]
Vz_ave_Ripple_4 = delta_z_ripple_4/delta_t_ripple_4; % m/s [average x velocity]
```

```

V_ave_ripple_4 =
sqrt((Vx_ave_Ripple_4)^2+(Vy_ave_Ripple_4)^2+(Vz_ave_Ripple_4)^2);

% work out average velocity for Wave gait initial angle (0 blocks)
Time_Wave_0 = Wave_Decline_0(:,1); % s
X_Wave_0 = Wave_Decline_0(:,2); % m
Y_Wave_0 = Wave_Decline_0(:,3); % m
Z_Wave_0 = Wave_Decline_0(:,4); % m
delta_t_wave_0 = Time_Wave_0(length(Time_Wave_0))-Time_Wave_0(1); % s
delta_x_wave_0 = X_Wave_0(length(X_Wave_0))-X_Wave_0(1); % m
delta_y_wave_0 = Y_Wave_0(length(Y_Wave_0))-Y_Wave_0(1); % m
delta_z_wave_0 = Z_Wave_0(length(Z_Wave_0))-Z_Wave_0(1); % m
Vx_ave_Wave_0 = delta_x_wave_0/delta_t_wave_0; % m/s [average x
velocity]
Vy_ave_Wave_0 = delta_y_wave_0/delta_t_wave_0; % m/s [average x
velocity]
Vz_ave_Wave_0 = delta_z_wave_0/delta_t_wave_0; % m/s [average x
velocity]
V_ave_wave_0 =
sqrt((Vx_ave_Wave_0)^2+(Vy_ave_Wave_0)^2+(Vz_ave_Wave_0)^2);

% work out average velocity for Wave gait second angle (2 blocks)
Time_Wave_2 = Wave_Decline_2(:,1); % s
X_Wave_2 = Wave_Decline_2(:,2); % m
Y_Wave_2 = Wave_Decline_2(:,3); % m
Z_Wave_2 = Wave_Decline_2(:,4); % m
delta_t_wave_2 = Time_Wave_2(length(Time_Wave_2))-Time_Wave_2(1); % s
delta_x_wave_2 = X_Wave_2(length(X_Wave_2))-X_Wave_2(1); % m
delta_y_wave_2 = Y_Wave_2(length(Y_Wave_2))-Y_Wave_2(1); % m
delta_z_wave_2 = Z_Wave_2(length(Z_Wave_2))-Z_Wave_2(1); % m
Vx_ave_Wave_2 = delta_x_wave_2/delta_t_wave_2; % m/s [average x
velocity]
Vy_ave_Wave_2 = delta_y_wave_2/delta_t_wave_2; % m/s [average x
velocity]
Vz_ave_Wave_2 = delta_z_wave_2/delta_t_wave_2; % m/s [average x
velocity]
V_ave_wave_2 =
sqrt((Vx_ave_Wave_2)^2+(Vy_ave_Wave_2)^2+(Vz_ave_Wave_2)^2);

% work out average velocity for Wave gait third angle (3 blocks)
Time_Wave_3 = Wave_Decline_3(:,1); % s
X_Wave_3 = Wave_Decline_3(:,2); % m
Y_Wave_3 = Wave_Decline_3(:,3); % m
Z_Wave_3 = Wave_Decline_3(:,4); % m
delta_t_wave_3 = Time_Wave_3(length(Time_Wave_3))-Time_Wave_3(1); % s
delta_x_wave_3 = X_Wave_3(length(X_Wave_3))-X_Wave_3(1); % m
delta_y_wave_3 = Y_Wave_3(length(Y_Wave_3))-Y_Wave_3(1); % m
delta_z_wave_3 = Z_Wave_3(length(Z_Wave_3))-Z_Wave_3(1); % m
Vx_ave_Wave_3 = delta_x_wave_3/delta_t_wave_3; % m/s [average x
velocity]
Vy_ave_Wave_3 = delta_y_wave_3/delta_t_wave_3; % m/s [average x
velocity]
Vz_ave_Wave_3 = delta_z_wave_3/delta_t_wave_3; % m/s [average x
velocity]

```

```

V_ave_wave_3 =
sqrt((Vx_ave_Wave_3)^2+(Vy_ave_Wave_3)^2+(Vz_ave_Wave_3)^2);

% work out average velocity for Wave gait fourth angle (4 blocks)
Time_Wave_4 = Wave_Decline_4(:,1); % s
X_Wave_4 = Wave_Decline_4(:,2); % m
Y_Wave_4 = Wave_Decline_4(:,3); % m
Z_Wave_4 = Wave_Decline_4(:,4); % m
delta_t_wave_4 = Time_Wave_4(length(Time_Wave_4))-Time_Wave_4(1); % s
delta_x_wave_4 = X_Wave_4(length(X_Wave_4))-X_Wave_4(1); % m
delta_y_wave_4 = Y_Wave_4(length(Y_Wave_4))-Y_Wave_4(1); % m
delta_z_wave_4 = Z_Wave_4(length(Z_Wave_4))-Z_Wave_4(1); % m
Vx_ave_Wave_4 = delta_x_wave_4/delta_t_wave_4; % m/s [average x
velocity]
Vy_ave_Wave_4 = delta_y_wave_4/delta_t_wave_4; % m/s [average x
velocity]
Vz_ave_Wave_4 = delta_z_wave_4/delta_t_wave_4; % m/s [average x
velocity]
V_ave_wave_4 =
sqrt((Vx_ave_Wave_4)^2+(Vy_ave_Wave_4)^2+(Vz_ave_Wave_4)^2);

% plot Decline results for each angle
figure
plot(angles, [V_ave_tripod_0, V_ave_tripod_2, V_ave_tripod_3,
V_ave_tripod_4], '-bx', 'LineWidth', 1)
ylim([0 0.14])
hold on
plot(angles, [V_ave_ripple_0, V_ave_ripple_2, V_ave_ripple_3,
V_ave_ripple_4], '-rx', 'LineWidth', 1)
hold on
plot(angles, [V_ave_wave_0, V_ave_wave_2, V_ave_wave_3,
V_ave_wave_4], '-gx', 'LineWidth', 1)
grid on
legend('Tripod','Ripple','Wave','interpreter','latex', 'Location', 'NorthWest')
xlabel('Angle of Decline $\theta$ $\circ$', 'interpreter','latex')
ylabel('Average Velocity $V_{ave}$ [m/s]', 'interpreter','latex')
title('Gait velocity comparison for CaveX robot over various
declination angles','interpreter','latex')

% Incline Results for each gait and each angle
% work out average velocity for Tripod initial angle (0 blocks)
Time_Tripod_0 = Tripod_Incline_0(:,1); % s
X_Tripod_0 = Tripod_Incline_0(:,2); % m
Y_Tripod_0 = Tripod_Incline_0(:,3); % m
Z_Tripod_0 = Tripod_Incline_0(:,4); % m
delta_t_tripod_0 = Time_Tripod_0(length(Time_Tripod_0))-
Time_Tripod_0(1); % s
delta_x_tripod_0 = X_Tripod_0(length(X_Tripod_0))-X_Tripod_0(1); % m
delta_y_tripod_0 = Y_Tripod_0(length(Y_Tripod_0))-Y_Tripod_0(1); % m
delta_z_tripod_0 = Z_Tripod_0(length(Z_Tripod_0))-Z_Tripod_0(1); % m
Vx_ave_Tripod_0 = delta_x_tripod_0/delta_t_tripod_0; % m/s [average x
velocity]
Vy_ave_Tripod_0 = delta_y_tripod_0/delta_t_tripod_0; % m/s [average x
velocity]

```

```

Vz_ave_Tripod_0 = delta_z_tripod_0/delta_t_tripod_0; % m/s [average x
    velocity]
V_ave_tripod_0 =
sqrt((Vx_ave_Tripod_0)^2+(Vy_ave_Tripod_0)^2+(Vz_ave_Tripod_0)^2);

% work out average velocity for Tripod second angle (2 blocks)
Time_Tripod_2 = Tripod_Incline_2(:,1); % s
X_Tripod_2 = Tripod_Incline_2(:,2); % m
Y_Tripod_2 = Tripod_Incline_2(:,3); % m
Z_Tripod_2 = Tripod_Incline_2(:,4); % m
delta_t_tripod_2 = Time_Tripod_2(length(Time_Tripod_2))-Time_Tripod_2(1); % s
delta_x_tripod_2 = X_Tripod_2(length(X_Tripod_2))-X_Tripod_2(1); % m
delta_y_tripod_2 = Y_Tripod_2(length(Y_Tripod_2))-Y_Tripod_2(1); % m
delta_z_tripod_2 = Z_Tripod_2(length(Z_Tripod_2))-Z_Tripod_2(1); % m
Vx_ave_Tripod_2 = delta_x_tripod_2/delta_t_tripod_2; % m/s [average x
    velocity]
Vy_ave_Tripod_2 = delta_y_tripod_2/delta_t_tripod_2; % m/s [average x
    velocity]
Vz_ave_Tripod_2 = delta_z_tripod_2/delta_t_tripod_2; % m/s [average x
    velocity]
V_ave_tripod_2 =
sqrt((Vx_ave_Tripod_2)^2+(Vy_ave_Tripod_2)^2+(Vz_ave_Tripod_2)^2);

% work out average velocity for Tripod third angle (3 blocks)
Time_Tripod_3 = Tripod_Incline_3(:,1); % s
X_Tripod_3 = Tripod_Incline_3(:,2); % m
Y_Tripod_3 = Tripod_Incline_3(:,3); % m
Z_Tripod_3 = Tripod_Incline_3(:,4); % m
delta_t_tripod_3 = Time_Tripod_3(length(Time_Tripod_3))-Time_Tripod_3(1); % s
delta_x_tripod_3 = X_Tripod_3(length(X_Tripod_3))-X_Tripod_3(1); % m
delta_y_tripod_3 = Y_Tripod_3(length(Y_Tripod_3))-Y_Tripod_3(1); % m
delta_z_tripod_3 = Z_Tripod_3(length(Z_Tripod_3))-Z_Tripod_3(1); % m
Vx_ave_Tripod_3 = delta_x_tripod_3/delta_t_tripod_3; % m/s [average x
    velocity]
Vy_ave_Tripod_3 = delta_y_tripod_3/delta_t_tripod_3; % m/s [average x
    velocity]
Vz_ave_Tripod_3 = delta_z_tripod_3/delta_t_tripod_3; % m/s [average x
    velocity]
V_ave_tripod_3 =
sqrt((Vx_ave_Tripod_3)^2+(Vy_ave_Tripod_3)^2+(Vz_ave_Tripod_3)^2);

% work out average velocity for Tripod initial angle (0 blocks)
Time_Tripod_4 = Tripod_Incline_4(:,1); % s
X_Tripod_4 = Tripod_Incline_4(:,2); % m
Y_Tripod_4 = Tripod_Incline_4(:,3); % m
Z_Tripod_4 = Tripod_Incline_4(:,4); % m
delta_t_tripod_4 = Time_Tripod_4(length(Time_Tripod_4))-Time_Tripod_4(1); % s
delta_x_tripod_4 = X_Tripod_4(length(X_Tripod_4))-X_Tripod_4(1); % m
delta_y_tripod_4 = Y_Tripod_4(length(Y_Tripod_4))-Y_Tripod_4(1); % m
delta_z_tripod_4 = Z_Tripod_4(length(Z_Tripod_4))-Z_Tripod_4(1); % m

```

```

Vx_ave_Tripod_4 = delta_x_tripod_4/delta_t_tripod_4; % m/s [average x
    velocity]
Vy_ave_Tripod_4 = delta_y_tripod_4/delta_t_tripod_4; % m/s [average x
    velocity]
Vz_ave_Tripod_4 = delta_z_tripod_4/delta_t_tripod_4; % m/s [average x
    velocity]
V_ave_tripod_4 =
    sqrt((Vx_ave_Tripod_4)^2+(Vy_ave_Tripod_4)^2+(Vz_ave_Tripod_4)^2);

% work out average velocity for Ripple initial angle (0 blocks)
Time_Ripple_0 = Ripple_Incline_0(:,1); % s
X_Ripple_0 = Ripple_Incline_0(:,2); % m
Y_Ripple_0 = Ripple_Incline_0(:,3); % m
Z_Ripple_0 = Ripple_Incline_0(:,4); % m
delta_t_ripple_0 = Time_Ripple_0(length(Time_Ripple_0))-%
    Time_Ripple_0(1); % s
delta_x_ripple_0 = X_Ripple_0(length(X_Ripple_0))-X_Ripple_0(1); % m
delta_y_ripple_0 = Y_Ripple_0(length(Y_Ripple_0))-Y_Ripple_0(1); % m
delta_z_ripple_0 = Z_Ripple_0(length(Z_Ripple_0))-Z_Ripple_0(1); % m
Vx_ave_Ripple_0 = delta_x_ripple_0/delta_t_ripple_0; % m/s [average x
    velocity]
Vy_ave_Ripple_0 = delta_y_ripple_0/delta_t_ripple_0; % m/s [average x
    velocity]
Vz_ave_Ripple_0 = delta_z_ripple_0/delta_t_ripple_0; % m/s [average x
    velocity]
V_ave_ripple_0 =
    sqrt((Vx_ave_Ripple_0)^2+(Vy_ave_Ripple_0)^2+(Vz_ave_Ripple_0)^2);

% work out average velocity for Ripple second angle (2 blocks)
Time_Ripple_2 = Ripple_Incline_2(:,1); % s
X_Ripple_2 = Ripple_Incline_2(:,2); % m
Y_Ripple_2 = Ripple_Incline_2(:,3); % m
Z_Ripple_2 = Ripple_Incline_2(:,4); % m
delta_t_ripple_2 = Time_Ripple_2(length(Time_Ripple_2))-%
    Time_Ripple_2(1); % s
delta_x_ripple_2 = X_Ripple_2(length(X_Ripple_2))-X_Ripple_2(1); % m
delta_y_ripple_2 = Y_Ripple_2(length(Y_Ripple_2))-Y_Ripple_2(1); % m
delta_z_ripple_2 = Z_Ripple_2(length(Z_Ripple_2))-Z_Ripple_2(1); % m
Vx_ave_Ripple_2 = delta_x_ripple_2/delta_t_ripple_2; % m/s [average x
    velocity]
Vy_ave_Ripple_2 = delta_y_ripple_2/delta_t_ripple_2; % m/s [average x
    velocity]
Vz_ave_Ripple_2 = delta_z_ripple_2/delta_t_ripple_2; % m/s [average x
    velocity]
V_ave_ripple_2 =
    sqrt((Vx_ave_Ripple_2)^2+(Vy_ave_Ripple_2)^2+(Vz_ave_Ripple_2)^2);

% work out average velocity for Ripple third angle (3 blocks)
Time_Ripple_3 = Ripple_Incline_3(:,1); % s
X_Ripple_3 = Ripple_Incline_3(:,2); % m
Y_Ripple_3 = Ripple_Incline_3(:,3); % m
Z_Ripple_3 = Ripple_Incline_3(:,4); % m
delta_t_ripple_3 = Time_Ripple_3(length(Time_Ripple_3))-%
    Time_Ripple_3(1); % s

```

```

delta_x_ripple_3 = X_Ripple_3(length(X_Ripple_3))-X_Ripple_3(1); % m
delta_y_ripple_3 = Y_Ripple_3(length(Y_Ripple_3))-Y_Ripple_3(1); % m
delta_z_ripple_3 = Z_Ripple_3(length(Z_Ripple_3))-Z_Ripple_3(1); % m
Vx_ave_Ripple_3 = delta_x_ripple_3/delta_t_ripple_3; % m/s [average x
velocity]
Vy_ave_Ripple_3 = delta_y_ripple_3/delta_t_ripple_3; % m/s [average x
velocity]
Vz_ave_Ripple_3 = delta_z_ripple_3/delta_t_ripple_3; % m/s [average x
velocity]
V_ave_ripple_3 =
sqrt((Vx_ave_Ripple_3)^2+(Vy_ave_Ripple_3)^2+(Vz_ave_Ripple_3)^2);

% work out average velocity for Ripple fourth angle (4 blocks)
Time_Ripple_4 = Ripple_Incline_4(:,1); % s
X_Ripple_4 = Ripple_Incline_4(:,2); % m
Y_Ripple_4 = Ripple_Incline_4(:,3); % m
Z_Ripple_4 = Ripple_Incline_4(:,4); % m
delta_t_ripple_4 = Time_Ripple_4(length(Time_Ripple_4))-
Time_Ripple_4(1); % s
delta_x_ripple_4 = X_Ripple_4(length(X_Ripple_4))-X_Ripple_4(1); % m
delta_y_ripple_4 = Y_Ripple_4(length(Y_Ripple_4))-Y_Ripple_4(1); % m
delta_z_ripple_4 = Z_Ripple_4(length(Z_Ripple_4))-Z_Ripple_4(1); % m
Vx_ave_Ripple_4 = delta_x_ripple_4/delta_t_ripple_4; % m/s [average x
velocity]
Vy_ave_Ripple_4 = delta_y_ripple_4/delta_t_ripple_4; % m/s [average x
velocity]
Vz_ave_Ripple_4 = delta_z_ripple_4/delta_t_ripple_4; % m/s [average x
velocity]
V_ave_ripple_4 =
sqrt((Vx_ave_Ripple_4)^2+(Vy_ave_Ripple_4)^2+(Vz_ave_Ripple_4)^2);

% work out average velocity for Wave gait initial angle (0 blocks)
Time_Wave_0 = Wave_Incline_0(:,1); % s
X_Wave_0 = Wave_Incline_0(:,2); % m
Y_Wave_0 = Wave_Incline_0(:,3); % m
Z_Wave_0 = Wave_Incline_0(:,4); % m
delta_t_wave_0 = Time_Wave_0(length(Time_Wave_0))-Time_Wave_0(1); % s
delta_x_wave_0 = X_Wave_0(length(X_Wave_0))-X_Wave_0(1); % m
delta_y_wave_0 = Y_Wave_0(length(Y_Wave_0))-Y_Wave_0(1); % m
delta_z_wave_0 = Z_Wave_0(length(Z_Wave_0))-Z_Wave_0(1); % m
Vx_ave_Wave_0 = delta_x_wave_0/delta_t_wave_0; % m/s [average x
velocity]
Vy_ave_Wave_0 = delta_y_wave_0/delta_t_wave_0; % m/s [average x
velocity]
Vz_ave_Wave_0 = delta_z_wave_0/delta_t_wave_0; % m/s [average x
velocity]
V_ave_wave_0 =
sqrt((Vx_ave_Wave_0)^2+(Vy_ave_Wave_0)^2+(Vz_ave_Wave_0)^2);

% work out average velocity for Wave gait second angle (2 blocks)
Time_Wave_2 = Wave_Incline_2(:,1); % s
X_Wave_2 = Wave_Incline_2(:,2); % m
Y_Wave_2 = Wave_Incline_2(:,3); % m
Z_Wave_2 = Wave_Incline_2(:,4); % m

```

```

delta_t_wave_2 = Time_Wave_2(length(Time_Wave_2))-Time_Wave_2(1); % s
delta_x_wave_2 = X_Wave_2(length(X_Wave_2))-X_Wave_2(1); % m
delta_y_wave_2 = Y_Wave_2(length(Y_Wave_2))-Y_Wave_2(1); % m
delta_z_wave_2 = Z_Wave_2(length(Z_Wave_2))-Z_Wave_2(1); % m
Vx_ave_Wave_2 = delta_x_wave_2/delta_t_wave_2; % m/s [average x
    velocity]
Vy_ave_Wave_2 = delta_y_wave_2/delta_t_wave_2; % m/s [average x
    velocity]
Vz_ave_Wave_2 = delta_z_wave_2/delta_t_wave_2; % m/s [average x
    velocity]
V_ave_wave_2 =
sqrt((Vx_ave_Wave_2)^2+(Vy_ave_Wave_2)^2+(Vz_ave_Wave_2)^2);

% work out average velocity for Wave gait third angle (3 blocks)
Time_Wave_3 = Wave_Incline_3(:,1); % s
X_Wave_3 = Wave_Incline_3(:,2); % m
Y_Wave_3 = Wave_Incline_3(:,3); % m
Z_Wave_3 = Wave_Incline_3(:,4); % m
delta_t_wave_3 = Time_Wave_3(length(Time_Wave_3))-Time_Wave_3(1); % s
delta_x_wave_3 = X_Wave_3(length(X_Wave_3))-X_Wave_3(1); % m
delta_y_wave_3 = Y_Wave_3(length(Y_Wave_3))-Y_Wave_3(1); % m
delta_z_wave_3 = Z_Wave_3(length(Z_Wave_3))-Z_Wave_3(1); % m
Vx_ave_Wave_3 = delta_x_wave_3/delta_t_wave_3; % m/s [average x
    velocity]
Vy_ave_Wave_3 = delta_y_wave_3/delta_t_wave_3; % m/s [average x
    velocity]
Vz_ave_Wave_3 = delta_z_wave_3/delta_t_wave_3; % m/s [average x
    velocity]
V_ave_wave_3 =
sqrt((Vx_ave_Wave_3)^2+(Vy_ave_Wave_3)^2+(Vz_ave_Wave_3)^2);

% work out average velocity for Wave gait fourth angle (4 blocks)
Time_Wave_4 = Wave_Incline_4(:,1); % s
X_Wave_4 = Wave_Incline_4(:,2); % m
Y_Wave_4 = Wave_Incline_4(:,3); % m
Z_Wave_4 = Wave_Incline_4(:,4); % m
delta_t_wave_4 = Time_Wave_4(length(Time_Wave_4))-Time_Wave_4(1); % s
delta_x_wave_4 = X_Wave_4(length(X_Wave_4))-X_Wave_4(1); % m
delta_y_wave_4 = Y_Wave_4(length(Y_Wave_4))-Y_Wave_4(1); % m
delta_z_wave_4 = Z_Wave_4(length(Z_Wave_4))-Z_Wave_4(1); % m
Vx_ave_Wave_4 = delta_x_wave_4/delta_t_wave_4; % m/s [average x
    velocity]
Vy_ave_Wave_4 = delta_y_wave_4/delta_t_wave_4; % m/s [average x
    velocity]
Vz_ave_Wave_4 = delta_z_wave_4/delta_t_wave_4; % m/s [average x
    velocity]
V_ave_wave_4 =
sqrt((Vx_ave_Wave_4)^2+(Vy_ave_Wave_4)^2+(Vz_ave_Wave_4)^2);

% plot Incline results for each angle and gait
figure
plot(angles, [V_ave_tripod_0, V_ave_tripod_2, V_ave_tripod_3,
    V_ave_tripod_4], '-bx', 'LineWidth', 1)
hold on

```

```
plot(angles, [V_ave_ripple_0, V_ave_ripple_2, V_ave_ripple_3,
    V_ave_ripple_4], '-rx', 'LineWidth', 1)
hold on
plot(angles, [V_ave_wave_0, V_ave_wave_2, V_ave_wave_3,
    V_ave_wave_4], '-gx', 'LineWidth', 1)
grid on
legend('Tripod','Ripple','Wave','interpreter','latex', 'Location', 'NorthEast')
xlabel('Angle of Inclination $\theta$'
    '$^{\circ}$','interpreter','latex')
ylabel('Average Velocity $V_{ave}$ [m/s]','interpreter','latex')
title('Gait velocity comparison for CaveX robot over various
    inclination angles','interpreter','latex')
```

Published with MATLAB® R2018b