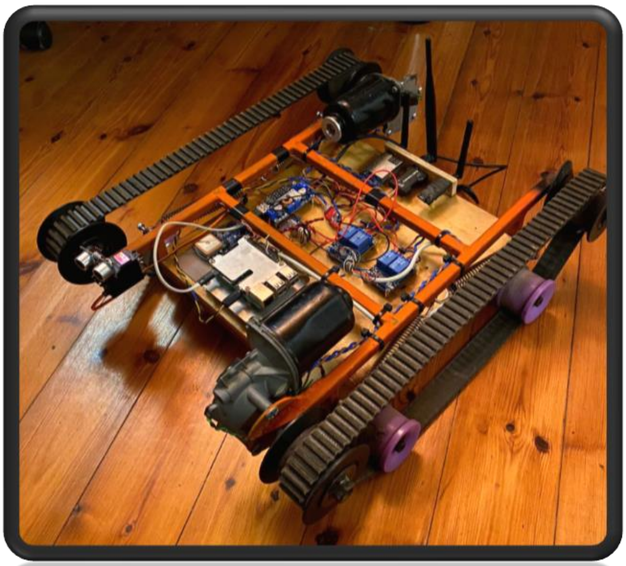

ITPPA3-B12 (PAYLOAD ROBOT)

Pretoria Campus

MARKKU EBERSOHN (JB78BP7P3)

RONALD MOMMSEN (SHGFNS3P8)

AMOGELANG SEHLAKO (WKKY6F1G4)

NOBUHLE SONGO (5K6BF2TB2)

GUSTAV STIERLIN (Z3WJG4VV9)

ACKNOWLEDGEMENTS:

Ronald Mommsen:

I would like to extend my sincere gratitude to the members of my team who played a vital role in the success of this project. Your contributions have been invaluable and greatly appreciated. Each team member brought unique skills and perspectives that helped us overcome challenges and achieve our goals.

Gustav Stierlin:

I would like to extend my heartfelt gratitude to my parents for their unwavering support, love, and encouragement throughout this project. Their constant belief in me and their willingness to lend a helping hand whenever needed have been instrumental in my journey.

I am also deeply grateful to my dedicated team members, whose collaboration and collective efforts have been invaluable. Their commitment, hard work, and expertise have greatly contributed to the success of this project. Their passion and enthusiasm for our shared goal have been a constant source of inspiration and motivation.

Amogelang Sehlako:

My deepest gratitude and appreciation go out to everyone who helped with our project. Your efforts, commitment, and knowledge have been crucial to the success of this project. We have worked together to drive robotics innovation to new heights. Your attitude of cooperation, innovative approach to solving problems, and efforts have been incredibly motivating. I appreciate your great efforts, thank you.

Nobuhle Songo:

Words cannot express my gratitude to our lecturer for his invaluable guidance and patience throughout this journey. I had the pleasure of collaborating with a dedicated and passionate team to which I'd like to extend a special thanks to for their hard work and perseverance.

Lastly, I would like to express my deepest appreciation to my family, sponsor and close friends for their support and motivation throughout this journey.

Markku Ebersohn:

I would like to thank each of my team members who have contributed to the success of this project, who without, this robot would not have come into existence. These individuals have taught me a lot and I would not hesitate to take on another project with them in the future. I've learned a great deal during the development of this project and will carry all the lessons with me throughout the rest of my life.

I would like to thank all the people who have inspired and supported me throughout this journey. Thanking my wonderful parents that have always motivated me to reach for goals that seems unreachable and teaching me all the skills needed to take on this project without fear. Also thanking my brothers who have always supported me and provided me with vital advice when needed.

Lastly, I would like to thank our lecturer as he has provided us with the necessary skills to complete the project successfully.

ABSTRACT:

This abstract summarizes the key features and functionalities of our group project, which focuses on the development of a versatile payload carrying robot equipped with a tank track drive train. The robot is designed to perform various tasks such as 3D mapping, obstacle avoidance, object following, and carrying payloads of up to 15 kilograms.

Our team conducted a comprehensive analysis of the problem statement to identify the user requirements and functions necessary for the robot. Based on this analysis, we assigned relevant specifications to each function of the robot. The design process began with the tank, followed by the design of individual components.

During the manufacturing phase, we encountered several challenges and discovered potential drawbacks in the different designs. These findings prompted us to refine and improve the designs to ensure optimal performance and functionality.

Once all the components were manufactured and the relevant electrical components were tested, we proceeded with the assembly of the tank. The assembly process involved meticulously integrating all the components to create the final robot.

To validate the performance and functionality of the tank, a series of rigorous tests were conducted. These tests assessed various aspects, such as mobility, obstacle avoidance, object following, and the robot's ability to carry payloads. The results of these tests provided valuable insights and allowed us to fine-tune the robot's capabilities. Overall, our iterative design and testing process ensured that the tank met the specified requirements and performed effectively in real-world scenarios.

TABLE OF CONTENTS

LIST OF FIGURES:

CHAPTER 1 (Background & Problem Setting):

1.1 INTRODUCTION

Approaching an era where automation is the drastically taking over various lines of work and the need for more technological intervention is needed over human intervention. However, automation is not just about taking a way job but also adding value to current lines of work. Many physically strenuous jobs have been neglected and left with jobs that could be much easier and safer with a few robotic improvements. This study will focus on how to implement robotic improvement physically strenuous work environments.

Focusing on mining, where low back pain has been reported to be affecting a lot of miners. Mining has continually been observed as being so hazardous environment job that it was the preserve of only slaves and criminals. Though there has been some degree of mechanization of the mining activities, there remains manual work involving lifting of heavy burdens. To address these challenges, mines are using industrial automation technologies like load-carrying mobile robots, automated guided vehicles (AGVs), and autonomous systems (Mohd Javaid, 2021). Even though these technologies have the potential to change industrial processes, they currently face several obstacles.

For instance, it might be challenging to integrate new technologies into current processes and logistical systems, and there are worries about job displacement and employee safety. This study aims to explore the challenges and opportunities associated with industrial mobile robots. The study will, specifically, examine how payload autonomous mobile robots can improve efficiency, reduce costs, and increase productivity in industrial sectors. The study ultimately, by addressing these issues, aims to provide insight and recommendations for companies looking to transition from traditional methods of working to autonomous mobile robots. The outcome of our study is to offer improved worker efficiency for workers in many industries with employees that need to transport heavy equipment, tools, and instruments over long distances to complete their tasks. Our aim is to provide efficiency while reducing the number of injuries at work.

1.2 BACKGROUND AND JUSTIFICATION

In the late year of 2022 South Africa alone consisted of over a million people working in construction workers and over 400 000 miners. These jobs are part of the most physically demanding line of work. These jobs require employees to walk uneven terrain and often involve carrying various loads from one point to another. It is shocking that the OSHA an acronym for the Occupational Safety and Health Administration “does not have a standard which sets limit on how much a person may lift” at work (Roux, 2021). In environments like construction the primary goal is to reduce the occurrences of work injuries however heavy lifting casualties continue to remain high on the percentage of work injuries.

Work injuries occur because of various reasons, including:

- Fatigue
- Improper lifting
- Repetition
- Hazardous materials
- Insufficient training

heavy lifting is the leading cause of injuries in the workplace. The most common areas affected by heavy lifting injuries include:

- Back sprains
- Pulled muscles.
- Joint injuries
- Foot and ankle injuries
- Spinal injuries

These injuries can be long term or short term, from musculoskeletal discomfort (MSD) to permanent disability. Studies found that carrying heavy objects depending on how heavy they are, how long/often they are carried are directly related to the potential backpain and disability. Payload autonomous mobile robots could potentially improve the efficiency and safety of mining operations. However, there are several challenges involved. This study was conducted to highlight those challenges. Mines can be enormously challenging environments with uneven terrain which can make it difficult for robots to navigate and carry loads without encountering obstacles.

Mines are also generally large and complex spaces which then require high-level mapping and localization capabilities to ensure the robot can move around the mine safely and efficiently. Finally, since payload robots are designed to transport heavy loads, it should be able to do that without tipping over or causing damage. The problem of developing payload mobile robots for use in mines arose because of the challenges faced by the mining industry in transporting heavy loads of materials and equipment in a safe, efficient, and cost-effective manner.

The conventional method of manually moving big items by human laborers is physically demanding and sometimes dangerous since it requires operating in small areas, on uneven ground, and around powerful gear and potentially dangerous materials. This poses a danger to the wellbeing of the workforce and may lead to accidents, injuries, and long-term health issues.

1.3 PROBLEM STATEMENT

It is known that payload robots could potentially mitigate injuries and improve safety and efficiency in mining and manufacturing industries. By automating the transport of heavy loads, these robots can reduce the risk of workplace injuries and accidents, such as back injuries, slips, trips, and falls, that can occur when workers manually transport heavy items. By easing physical strain on employees and eliminating the need for them to execute jobs in dangerous conditions, this can increase overall safety. To do this, however, several major issues associated with the robots must be resolved, including load capacity and stability, navigation and obstacle avoidance, autonomy and power management, coordination and communication, resilience, and dependability.

The final design must be both scalable and cost-effective, and it must take into consideration the unique demands and conditions of the area in which the robot will work. The project team will need to use a multi-disciplinary strategy that integrates knowledge in robotics, engineering, computer science, and other pertinent domains to handle these issues. This study is done to discover new information, information that will help researchers with their own studies or projects. The robotics industry is an ever-growing industry where there are constantly new possibilities, new outcomes, and new issues that they can solve. Studies are essential for providing the society with that information for integration and possible expansion.

1.4 OBJECTIVES

The current research objective in the field of robotics is to develop collaborative robots that can safely share their workspace with human coworkers. While there are working solutions to protect humans from damages induced by the body of the robot there are still challenges in the field of work and tool safety. While we consider mobile autonomous robots as mobile robots that can perceive their surrounding via various sensors and having the ability to react to changes in their environment, their application in mining environments may yield several beneficial effects including:

- Enhanced efficiency.
- Ensured safety.
- Optimized resource utilization.
- Enhanced scalability and adaptability.

To be able to design a planning method for the implementation of mobile autonomous robots the first step is to find use cases in a mining environment. During further research it will be possible to evaluate the needed processes and services to implement the use cases and derive shared requirements for a mine.

Considering these and other factors derived from subsystems of mobile robots and programming requirements will give insight in the steps needed to plan and deploy mobile robots. Summarizing and generalizing all acquired research and studies will lead to a planning methodology to support engineers in designing mines that are well suited for automatization with mobile robots.

1.5 SCOPE OF THE STUDY

This dissertation will concentrate on payload autonomous mobile robots intended for mining applications, with a particular focus on the movement of large loads in subterranean mines. This dissertation will consider literature on related technologies and features that will fulfil the objectives of the project's payload aims. The payload system and structure design as well as their implementation details. The tests, challenges, results, and conclusion of the design implementation.

1.6 METHODOLOGY

The task of deducting use cases for mobile robots has been split into several steps that are shown in (figure). To further develop the drafted use cases and methodically consolidate them they were checked against common mining setups in mines. This process created six application potentials. To include additional knowledge and opinions from not only a theoretical point of view but also from practice, expert interviews have been organized. The overall result of this research process are five use cases which will serve as foundation for further research.

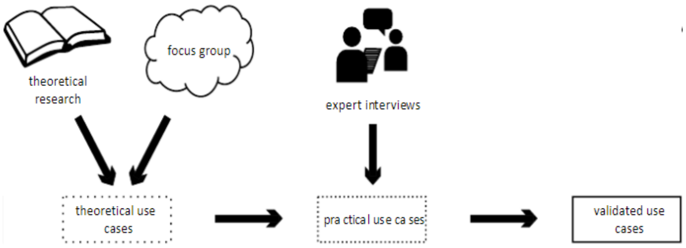

Figure 1 Use Case Methodology

I. Theoretical deduction of Use Cases: The theoretical background to find new use cases for mobile robots in the mining industry is based on the problem setting identified. Using the system theoretic background of modelling processes as separate work packages enables a systematic path to find new use cases for mobile robotic platforms. The focus in this article is set upon load bearing and navigation in mining areas. The following list gives an overview of the considered process steps:

Exploration:

- *Geophysical surveys*

Site Preparation:

- *Acquisition of permits*

Processing:

- *Crushing and grinding*
- *Separation*
- *Concentration*
- *Dewatering*

Smelting:

- *Smelting*
- *Refining*

Transportation:

- *Loading*

Reclamation and Closure:

- *Monitoring and maintenance*

The analysis of the work packages and specific tasks for the mining process shows that there are several stages of the mining process. The main problem is to decide where automatization will create the most benefit in the process chain. The literature study and a critical review of the mining process within the mines yielded the following areas that would benefit the most by an augmentation by mobile robots.

- Payload handling
- Material transportation
- Surveying and Mapping
- Hazardous Environment Exploration

II. Interviews: To validate the deducted application potentials the method of expert interviews was chosen.

1.7 SIGNIFICANCE OF THE STUDY

The implementation of a payload robot can benefit employees and businesses globally in Scientific research, Mining operations, Sports, Trades, Civil Engineering, and many industries with employees that need to transport heavy loads (robotsdoneright, 2023).

Increased productivity: payload robots in the work area increase the productivity of the business in less time than the average human. With these robots' businesses can run for longer at a progressive constant speed with minimal breaks as they can transport overwhelming objects more rapidly, permitting workers to turn their attention to more complex errands.

Saves costs and reduced injuries: Businesses keeping the employees safe from harm and alleviating the risk of injuries would save cost in the long run as less employees are likely to get hurt onsite. this would reduce the amount of employee compensation claims from disgruntled/injured employees. Robots are designed to function in a secure and controlled way, minimizing the potential for casualties caused by human mistake or fatigue.

Payload robots can save costs in numerous ways. They reduce need for manual labour or specialized hardware for overwhelming lifting, saving businesses labour costs and the injuries that can lead to additional expenses. Furthermore, Robots optimize workflow and reduce business downtime, leading to increased operational efficiency as well as savings cost in the long run. Employees would benefit greatly as they would be operating in a much safer environment. the implementation of this robot would save many workers from excessively overworking their bodies to the point of a dangerous decline.

Enhanced precision and accuracy: These robots are designed to perform tasks with accuracy and precession. They can take after pre-defined ways or utilize sensors and calculations to explore and position payloads with most extreme accuracy, minimizing mistakes and improving operational quality.

Adaptability and flexibility: Payload robots are often designed to be versatile and easily adaptable to different environments and tasks. They can be modified to handle various load sizes and shapes, making them reasonable for a wide range of applications. Their adaptability permits businesses to quickly react to changing requests and optimize their operations appropriately.

Scalability: As businesses evolve or payload requirements change, payload robots can be made to suite the requirements of different businesses. they can be effectively scaled up or down to meet the future business needs. This versatility gives businesses with the capacity to grow their operations without noteworthy extra speculation in labour or gear.

Data analysis and collection: Numerous payload robots are designed with sensors and data collection features. That can accumulate useful data about the loads they carry, such as weight, measurements, and movement patterns. This information can be analysed to optimize workflows, recognize inefficiencies, and make data-driven choices for business process improvement.

Robots play an important role in contribution to the body of knowledge and practice in many fields. A few ways in which robots contribute:

- In research and exploration data collected form sensors can be used to contributes to scientific research, allowing researchers to study and understand various terrain, environments etc.
- Robots are designed to inspire and educate the public. Introducing robots in public and educational environment. These educational payloads be used as hands-on learning tools and encourage students to pursue careers paths that contribute to the overall dissemination of knowledge.
- Can be used in various application for different markets. Robots can be used in different real-world problems. In areas such as medicine, robot can carry medical equipment, research materials, or telemedicine devices to support healthcare initiatives in remote or underserved regions.

One could say that a that the implementation of these robots could lead to job losses as less workers would be needed in to carry loads, and this may spark a trend in continuous automation in many more fields of work. It is important to confront this line of thought with the implementation of such technologies designed to help and not endanger jobs. As well as training workers in the integration of digital /autonomous systems to the workplace. Companies and Educational systems can promote training for workers whose jobs may be in danger.

1.8 STRUCTURE OF THE DISSERTATION

This dissertation is organized into five chapters. The study's history, problem description, aims, scope, methodology, and importance were all included in Chapter 1's introduction. A thorough analysis of the pertinent literature and cutting-edge developments in the field of payload autonomous mobile robots will be presented in Chapter 2. The design utilized in project, including system design, mechanic design, electronic design, software design and integration design will be covered in Chapter 3.

The project problems that came up during the system prototype development, solutions, data readings and test results the will be covered in Chapter 4. The dissertation will be concluded in Chapter 5 by summarizing the main conclusions, results, offering suggestions, and going through potential directions for further study.

CHAPTER 2 (Literature Review):

2.1 INTRODUCTION

The use of robots in various industries has been on the rise in recent years due to their ability to perform repetitive tasks, work in hazardous environments, and increase productivity. One application of robots is in the field of payload transportation. Payload-carrying robots have the potential to reduce the physical strain on workers and increase efficiency in various industries.

In this literature review, we will explore the use of a payload-carrying robot that drives on tank tracks/treads to navigate off-road terrain. The inspiration for this robot came from the need to assist workers in the mining industry, specifically in decline tunnel mines, where workers are required to carry heavy instruments up and down long declines. This type of work can lead to fatigue and reduced productivity, which is why a robot that can carry loads could be of significant benefit.

However, the potential uses of this type of robot are not limited to the mining industry alone. The ability to navigate difficult terrain and carry payloads could also prove useful in other fields, such as construction, agriculture, and military operations. By reducing the physical strain on workers, these robots could increase efficiency and productivity in a range of industries.

In this literature review, we will examine previous research on payload-carrying robots, including their design, capabilities, and potential applications. We will also explore the challenges associated with their use, such as navigation in complex environments and interaction with human operators. Ultimately, this review aims to contribute to the ongoing development of payload-carrying robots and their potential to improve efficiency and safety in various industries.

Overview of payload-carrying robots

Between the 1960s and 1970s the concept of terrain robots began to emerge with the development of small, mobile robots used in space exploration. The Mars Rover was developed by NASA in the 70s to explore hostile or unknown terrain. The robot was a platform that integrated a mechanical arm, proximity sensors, a laser telemetry device, and stereo devices.

- In 2007, an article titled “Design and prototype of a six-legged walking insect robot” was published based on the studies of legged robots. (Alli, 2007) The motivation for studying legged robots is to give access to terrains that are inaccessible or too dangerous for human beings. Legged robots can be used for rescue work after earthquakes and in hazardous places such as the inside of a nuclear reactor.
- Aiming at the crossing problem of complex terrain, to further improve the ability of obstacles crossing, all-terrain wheel-legged hybrid robot were developed (Jianwei Zhao 1, 2021). According to the article, the robot operates in different road conditions. It adopts a wheel and leg compound structure, which can realize the transformation of wheel movement and leg movement to adjust its motions state.
- In recent years studies have been also focused on developing tracked robots. Tracked robots are designed with a reliable mechanical platform, scalability of the control system, and the flexibility of their self-diagnosis and error recovery mechanism. In 2022, a paper on such a system was published and titled "DESIGN AND DEVELOPMENT OF ALL TERRAIN TRACK ROBOT” (Dr. Y.R. Kharde*1, 2022)

All-terrain robots have remained a focus of research and development in recent years, with uses that include disaster response, environmental monitoring, and exploration. Advances in artificial intelligence and machine learning are also being implemented into all-terrain robots to enable more autonomous and intelligent behaviour.

Overall, this literature review demonstrates that all-terrain robots have been a focus of research for many years and that there has been a steady advancement in their development, capabilities, and design.

Importance and applications in various industries

In many industries, the most common solution for carrying heavy objects is to use your brain instead of brawn. (Institutional Integrity and risk Management, n.d.) Workers are encouraged to use the right posture when picking things up and carrying them around. Workers should use mechanical means where possible like hand trucks or push carts.

In many situations it's not possible to use mechanical means to move around an object thus workers need to carry it by them self. In this case the workers are encouraged lift a heavy object properly:

- Take your time, when picking up heavy objects haste fully it could strain back muscles.
- Ensure that your body is facing the object that you're lifting.
- Never reach out to pick up a heavy object.
- Bend your knees, instead of bending your back.
- When carrying an object, try to keep it between your waist and shoulders.

Although this is the most common solution to prevent injuries in the workplace, there are other solutions that involves robotics. These solutions could contribute towards the safety of the workers in the workplace, by reducing fatigue. When fatigue is reduced the workers will work more efficiently, have more motivation and better concentration.

2.2 DRIVE TRAIN SYSTEMS

Types of **suspension** systems used in payload-carrying robots.

Small off-road vehicles can have a variety of different suspension types to improve performance and handling on uneven terrain. These suspension types include:

- Independent suspension
- Solid Axle suspension
- Leaf spring suspension
- Coil spring suspension
- Air suspension
- Torsion bar suspension
- Hydro-Pneumatic suspension
- Hydrostatic suspension
- Rubber band tracks

Independent suspension allows for each wheel to move independently of the others which in return can improve traction and handling on rough terrain. This suspension is usually used in off road motor sports where high-speed performance is required. (Moe, 2017)

Solid Axle suspension is a suspension type where two wheels on each side are connected via a single strong beam. This type of suspension is common on trucks and sport utility vehicles. This suspension's strong suit is slow and steady off roading and carrying heavy loads.

Leaf spring suspension is a type of suspension that uses curved metal strips that are stockpiled on top of each other and then attached to an axle. This suspension is durable and provide for a smooth ride.

Coil spring suspension are used in most off-road vehicles. Compared to leaf spring suspension, coil springs are lighted and more flexible which improves the vehicles' handling on uncertain terrain.

Air suspension utilizes bags that are filled with air to absorb shock which leads to a smoother ride. This suspension is mostly used in luxury sports utility vehicles, this suspension can adjust ride height.

Torsion Bar suspension is commonly used in tracked vehicles like tanks. The torsion bar is mounted to the hull of the vehicle and connected to the wheel via a suspension arm. When the vehicle moves over uncertain terrain the torsion bar twists and then stores the energy, when the wheel moves back down the energy is then released. In return this helps with shock absorbance.

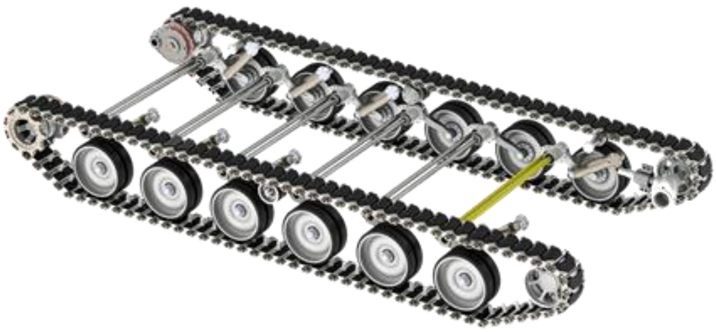

Figure 2 Torsion bar Suspension (MENATEK, 2023)

Hydrostatic suspension utilizes a hydraulic system to ensure a smooth ride. This suspension is commonly used tracked vehicles like bulldozers and excavators. It is made up of a hydraulic pump, hydraulic motor, and a set of hydraulic cylinders.

Hydro-Pneumatic suspension utilizes hydraulic fluid and compressed air to absorb shock. This suspension type is mainly used in military vehicles like tanks. This suspension is much like Hydrostatic suspension in terms of build. It consists of a hydraulic pump, pneumatic compressor, and hydraulic cylinders. This suspension is mounted to the hull of the vehicle and the suspension arms. When rough terrain is encountered, the hydraulic cylinders compress, and pneumatic compressor will then inflate which in return will absorb shock for a smoother ride.

Rubber band tracks are commonly used in small and light weight vehicles like snowmobiles and ATVs. The rubber bands are flexible by design which provides a smoother ride over rough terrain. This suspension system consists of a set of rollers that guide the tracks and a set of springs to absorb shock. In contrast to other tank track designs, this suspension is made up of rubber belts instead of metal tracks. The flexibility of the rubber tracks allows these tracks to conform to the contours of the terrain, this results in less bouncing for the people or payload on the vehicle. (Wright, 2021) One of the biggest advantages of the rubber tracks suspension is that the tracks won't damage surfaces when trans versing over. The rubber also allows for more grip in slippery conditions. A disadvantage of this suspension is the fact that the rubber isn't as durable as metal tracks, they could also be more expensive to replace. (Lombardo, 2018)

Drive train systems used in payload-carrying robots.

In an article by Harry Lye, it is explained that a track drivetrain offers the best solution for a vehicle that must be versatile and is required to operate over diverse terrain, which includes difficult ground. Tank tracks has an increased surface area and drive which provides a better solution. Tracked vehicles excel in carrying a payload, this is due to the increased ground pressure. According to LiteTrax (Company that produces tracks for multiple vehicles), tracks have advantages and disadvantages over wheels. (LiteTrax)

Advantages:

- Power efficiency
- Traction
- Moving on rough terrain
- Aesthetics
- Ground impact
- Weight growth potential

Disadvantages:

- Lower speed

Tracks are Power efficient; this is since the force of the vehicle is placed over a wider ground surface area. When transversing over uneven terrain like mud, sand, snow or rugged surfaces, the tracks drivetrain distributes the weight of the vehicle over a larger surface area. This results in the reduction of ground pressure while simultaneously increasing traction. More traction enables the vehicle to have better grip on a surface, which results in less spinning or slipping. This leads to better power performance when compared to wheels. Tracks can conform to the shape of the terrain; this keeps the vehicle from tipping while reducing energy loss from lateral movements which in turn leads to power efficiency.

One of the main reasons Tanks in the military use a track drive train is due to this type of drive train's ability to navigate obstacles that would have stopped vehicles with wheels. (Mizokami, 2018) . This is due to the tracks' ability to conform to the contours of the terrain. This allows military tanks to climb over large obstacles without getting stuck. Tracks allow for improved traction compared to wheels; tracks provide more contact with the ground. This increases the amount of friction and traction between the tracks and the terrain. This allows for military tanks to transverse over rough terrains such as loose soil, steep inclines, and rocks. Paired with this is the track's ability to distribute weight more evenly than the wheels. Since the weight is distributed more evenly, the possibility of the tracks sinking into soft or uneven ground is improbable.

This results in greater stability and maneuverability on difficult terrain. Military tanks can pivot in place, which allows the tanks to be able to turn in place without needing to move forward or backward. This allows the tanks to change direction instantaneously (Mullen, 2020). Lastly, the tracks on military tanks are equipped to handle shocks and vibrations that can occur from traversing rough terrain. Meaning that when compared to wheels tracks provide a more cushioned ride, which in return can reduce the risk of damage to the payload, the tank and improve the comfort of the crew. (hambling, 2008)

2.3 MOTORS

Role of motors in payload-carrying robots

In the world of automation and autonomous robots, the actuation and motion of mechanical components are one of the fundamental core's that these systems rely on to perform tasks. In this section, we will look at different actuators/motors and decide what type to use. After the actuator type is chosen, we will look at the different motor controllers, needed to make these actuators/motors function. We will look at topics such as the complexity of integration, Implementation complexity, operational voltages, and ease of programming.

Types of motors commonly used (e.g., DC motors, brushless motors)

Stepper Motors

Stepper motors are motors that move in precise steps, or angles (see Figure 3). This is done by a series of programmed steps, which have timed pulses that supply the motor with the needed current to output rotational motion. This type of motor does not work like a conventional Dc motor, which rotates when a DC current is supplied. Rather it will lock in the current position that it is in if power is supplied. Each series of pulses rotates the motor in one increment steps, depending on the direction that was chosen. (See Figure 4) These steps are variable, with a standard step being 1.8 deg, relating to about 200 steps per revolution. (ENGINEERING, n.d.)

Usually, these types of motors are used in high-precision averments, where calculated distances need to be travelled to achieve the desired outcome e.g., CNC machines.

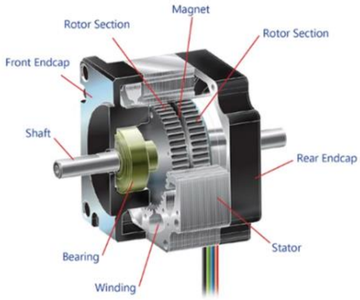

Figure 3 Basic Configuration of Hybrid Stepper Motor (ENGINEERING, n.d.).

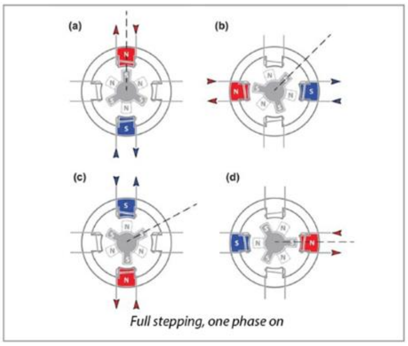

Figure 4 Phasing Sequence of Each Step (ENGINEERING, n.d.).

It should be noted that with these steps that are precise, comes an unintended drawback to using these types of motors. With the increase in rotational speed comes a proportional decrease in the holding torque that the motor can deliver. (See Figure 3) This should be considered when choosing a motor that operates at high RPMs.

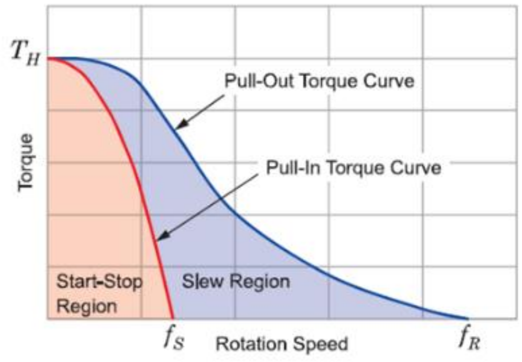

Figure 5 Stepper Motor Torque vs Speed Graph (MOTOR, n.d.)

Servo Motors

A servo motor is a closed-loop electromechanical device, that produces precise torque and rotational motion depending on the current and voltage supplied. These motors consist of 3 parts: A motor, a Feedback system, and a control system (See Figure 4). These motors usually work with a 3-wire harness. The first 2 are the power supply wires, which supply the motor with the operational voltages it needs. The last wire is the signal wire for delivering the signal of which position, and revolution/rotation count are transmitted through. This signal is usually a PWM (Pulse Width Modulation) Signal which is generated by the microcontroller controlling the system. (KOLLMORGEN, 2020)

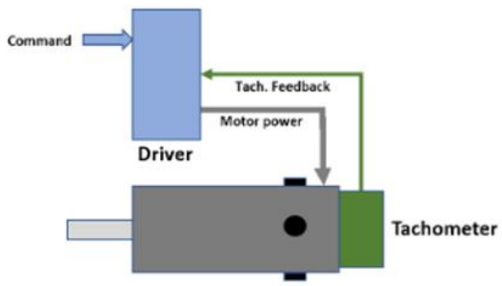

Figure 6 DC Servo Motor System (KOLLMORGEN, 2020)

Geared DC Motors

Geared DC Motors are basically normal DC that provides a torque and rotational motion when a voltage and current are supplied. However, these motors are paired with gearboxes to increase the motor’s torque. It comes with one drawback though, which is that when the torque increases a proportional decrease in rotational speed will be seen, depending on the gearbox type that will be selected. (See Figure 5) It should be noted that to reverse the direction of these motors, the current flow needs to be reversed. In other words, the poles of the batter need to switch from positive to negative. This motor’s speed can also be controlled by using a feature called PWM (Pulse Width Modulation). PWM controls the percentage of time the motor is active through pulses.

Figure 7 DC Geared Motor (Bosh, 2023, p. 32)

BLDC Brushless Motors

BLDC (Brushless Direct Current) motor can be seen as a mini version of a 3-phase motor used in industrial factories (See Figure 6). However, these motors are shrunken to be utilized in robotic, or hobbyists. These motors are widely used in the RC (Remote Control) field, because of their amazing power-to-weight ratio. Coupled with the power-to-weight eight ratios, it should be noted that this motor usually operates at high rpm. The RPM can be checked by multiplying the Battery voltage with the motor’s KV rating.

For instance, if we are using a 12v battery and a motor with a KV rating of 1000, the operational RPM of the motor will be +- 12000 rpm. These motors come with another benefit, which is that there are no brushes that need to be replaced after extensive use. It comes with the disadvantage of being a quite expensive option when looking at bigger motors**.** See the example of the BLDC motor in Figure 7.

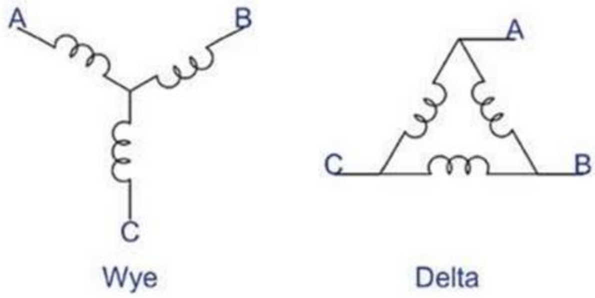

Figure 8 BLDC Winding configuration (Staff, 2008) .

Figure 9 Standard BLDC Motor (SPEKTRUM, n.d.)

Motor specifications and selection criteria

It should be noted that all the motors that will be reviewed in this table, will operate on a voltage of 12V. This is due to the constraint, of the type of power source we have chosen and should be considered when selecting any type of electronics, that will be integrated into the system.

TYPE	VOLTAGE	AMPS	RPM	TORQUE	PRICE
STEPPER	12	3	2500	19 kg-cm	R644,00
SERVO	12	42	30	0.075-50 NM	R1 498,07
DC GEARED	12	6	+-60	40NM	R520,00
BLDC	12	130	46800	NAN	R 1 798,70

Table 1 Motor Comparison

Motor control techniques for efficient operation

In this section, we will now look at the different motor drivers that are available for driving DC motors. Also, keep in mind the requirements such as 12V and rated for current that the motors pull. We will look at 3 different motor controllers listed below.

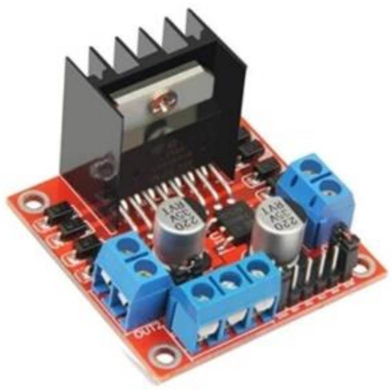

Figure 10 HKD DUAL MOTOR DRIVER L298 SCREW

(Figure 10) This motor controller is a very popular option, in the robotics hobbies community. This motor controller provides bidirectional control for 2 motors whilst having variable speed functionalities. It should be noted that this motor controller can only control small motors, which can only handle small loads.

Figure 11 DFR 40A BI/DIR BRUSHED MOTOR ESC

(Figure 11) This motor controller provides a bidirectional output for a single motor; it is also controlled by a standard RC PWM signal. Meaning that all wires that are needed for control are a GND and 1 signal wires. Which is nice when developing a robot and channels aren't readily available.

Figure 12 BMT MONSTER M OTO SHIELD-VNH2SP30

(Figure 12) This motor controller features various functionalities, from bidirectional control for 2 motors, whilst having a speed control function utilizing PWM. Also, this option has AMP meters for each motor, which can be utilized to calculate how long the batteries will last under load.

TYPE	VOLTAGE	AMPS	PRICE
Figure 8	35	2	R45,00
Figure 9	24	40	R430,00
Figure 10	16	14	R355,01

Table 1 Motor Driver Comparison

2.4 LOCALIZATION

Importance of localization for payload-carrying robots

Localization is a critical aspect in the development and operation of payload-carrying robots. It refers to the ability of a robot to determine its position and orientation in each environment accurately. In the context of payload-carrying robots, localization plays a pivotal role in ensuring precise navigation, effective load transportation, and overall operational efficiency.

Localization Techniques Used In Robots (E.G., GPS, Inertial Navigation)

LOCATION

In the world of automation and autonomous robots, the need of knowing the location sometimes arises. Knowing the location of the robot can help you achieve various things, like knowing where to collect the robot when it gets stuck or when it has broken down on the job. It also allows the developer to calculate sophisticated algorithms, such as 3D terrain mapping, live geofencing, or ETA calculations.

In this section, we will look at different ways of finding a known object's location. There are various ways of finding the location of an object.

These include:

- Global Positioning System (GPS)
- Cellular Location Positioning
- Radio Frequency Identification (RFID) positioning

Global Positioning System (GPS)

What is GPS?

GPS stands for Global Positioning System and is the most widely used geolocation technology. This technology has been used in various devices over the years, from smart mobiles all the way to waypoint calculators and flight controllers. The technology is based on the usage and communication between satellites that orbit the Earth (INCOGNIA, 2023). The satellites most used for this purpose are MEO (Medium Earth Orbit) satellites, 10 of which will provide global coverage (Duddell, 2015). Originally when GPS was invented, its purpose was to be used in military applications, with the research being conducted by the US Department of Defence. Its purpose was to serve as a navigational tool for missiles and the guidance of bombs, whilst also being used in map-making and search and rescue operations. (Rifandi, 2013) Luckily for us, this technology is available for civilian use all over the world, making it a useful tool for the development of robots.

How does GPS Work?

The whole GPS System consists of 3 components:

- Space Segment
- Control Segment
- User Segment

Space Segment

The space segment of the GPS system consists of a constellation of about 24 satellites, located at an altitude of about 20192 km above the Earth, and work together to provide global coverage. The satellites each consist of a platform of radio transceivers, computers, and additional equipment. One of which is an atomic clock providing incredibly high accuracy. The accuracy comes from a decaying substance such as rubidium or caesium, which provides a consistent steady resonance. (Rifandi, 2013)

Control Segment

The control segment of the system is located on the Earth. Sixteen to be precise, all distributed in such a way as to allow the monitoring of each satellite by at least three control stations at a time. The purpose of each control station is to synchronize the satellites which it has in its control and ensure that the satellites remain in a constant orbit around the Earth. (Rifandi, 2013)

User Segment

The user segment of the system consists of an antenna to receive radio frequency signals from the satellites and convert them into a usable digital format. (Rifandi, 2013)

These three works in conjunction with one another to achieve the result of outputting the user's precise location. How exactly? Firstly, the satellites transmit signals towards the earth continuously, which are then received by the ground station. The signals consist of 2 types: coarse/acquisition (A/C) and precession (P).

The information contained within these codes consists of when the codes were transmitted, and what type of position the satellites are in. This data is all that is needed by the User section of the system to complete all the calculations necessary to find the location of the GPS receiver. The process works because of the global coverage of the satellite constellation. At any time and at any location on the Earth, a GPS receiver will have a clear line of sight of at least 4 satellites. This allows the Receiver to perform its calculation, which is based on a method of triangulation, solving the unknown node of N, from the received information. This method relies on the intersection of 3 spherical surfaces on the earth achieving an accuracy of around a few millimetres all the way up to 100 meters. (Rifandi, 2013)

Cellular Location Position

What is Cellular Location Position?

Cellular location positioning is a technology that uses cellular communication towers to reach highly effective localization when GPS-level accuracy is not applicable. This form of localization works out the position of the object based on the geometric constraints between where the base station (BS) is located and where the object position is. This technique is called Time of arrival (ToA) or Time difference of Arrival (TDoA) (Rappaport, 2021).

How does ToA and TDoA Work?

These techniques use the fundamental principles of transitions to calculate the position of the object. The fundamental principle used is that data have a certain rate that it can be transmitted. For instance, how the speed of sound is around 343m/s. This principle also applies to cellular data communication between the towers. By applying this principle, the data communication time between the different towers and the object can be measured, which can then be multiplied by the speed of the data transmission to get the estimated distance between the one communication tower and the object.

This can then be done between all of the communication towers in the object’s current location. From there, a representation can be generated of which tower it has communicated to. And then circles/hyperbolas with the known estimated positions can be drawn up onto the map, with the intersection of all these circles being the estimated position of the device’s location (Rappaport, 2021). This process is illustrated in Figure 13.

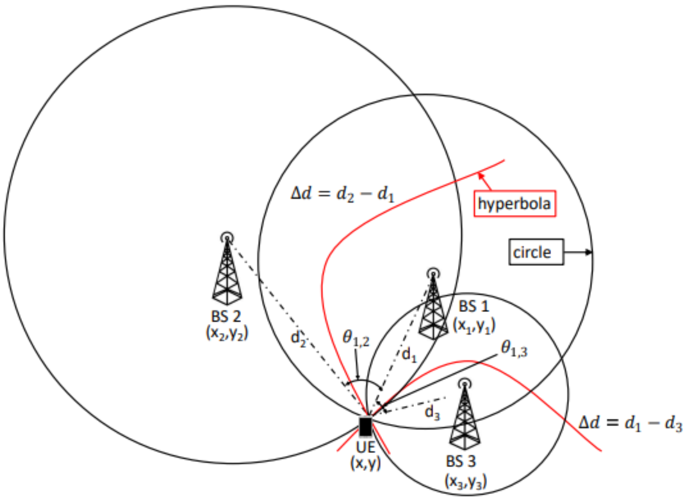

Figure 13 AoA localization technique in black dotted lines (Rappaport, 2021).

The Drawback to using a localization technique like this is that it is not possible in places where there Is no signal, and then also it will require the device/object that wants to be found to also have airtime or another form of communication currency like data.

RADIO FREQUENCY IDENTIFICATION (RFID)

What is RFID Localization?

RFID stands for Radio Frequency Identification and is widely used in the business industry for electronic identification and tracking. These systems are made of 3 main parts: Server, Readers, and Tags with different IDs see Figure 14 for the illustration.

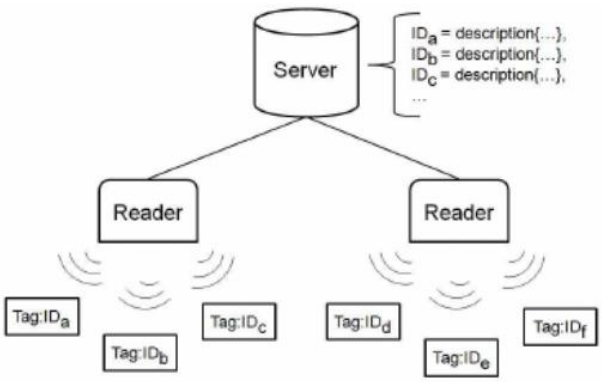

Figure 14 Classic RFID System Architecture (Mathieu Bouet, 2008)

There exist various methods for powering these circuits, however, we will only look at the method which utilizes an internal battery for the power source. This type of RFID tag is called a semi-passive tag. To communicate with these tags there exists 2 interfaces:

Radio Frequency Interface and then also a Server communication interface. The RF interface can only identify the tags if they are in the read range of the system. Whereas with the Server communication system, the identities of the tags are collected and then a localization algorithm is applied to work out the tags' precise location.

These systems work very similarly to the systems explained in the cellular localization techniques, and some of the methods are even shared across them, like ToA and TDoA algorithms. Having said that, there is no need for further research on this topic as it is very limited in what range of devices and in what environment it can be utilized.

Usually, these types of systems are employed indoors where there are localized servers for optimal communication, and card readers at known positions, to properly do the math to find the tags.(Mathieu Bouet, 2008). All these restrictions this system imposes for the implementation of it, make it a nonviable system to be used in our project.

3D-MAPPING & MODELLING

Under the concept of Mobile robots, we often see robots that must operate in various environments, while completing certain operations. These operations, which need to be completed, lead to a functionality requirement that must be fulfilled. This requirement is Localization or route mapping. This concept is based on the 3D Modelling of the terrain around the robot, whilst navigating through the identified obstacle features the terrain possesses. The operation is to build a generalized map of the robot's current surroundings, whilst developing a navigational route that the robot can navigate according to.

With the industry moving more towards the automation of various tasks, this problem has been solved in various ways. The first involves the process of generating a map from a previously scanned, operational environment.

This process involves the combining and stitching of multiple forms of sensory data, which makes up a precompiled 3D environment, whilst presenting the robot its route that has been generated for it. The second form is called Dynamic mapping. This process involves the robot recognizing certain features in the current environment that the robot can interact with and recognize.

There are various beneficial advantages and disadvantages that come into play when making use of these functionalities. For instance, when it comes to the prior map development, the capture process, and recording process of the environment, will require a lot of effort from a team, to develop.

This process is also preferred to be used in an environment that doesn't change much. This process involves the use of an operational server that feeds the robot with its route that it needs to take the whole time.

With frequently changing environments, the process of dynamic mapping is typically used. When it comes to dynamic mapping, the processing capabilities of the robot will become quite hefty, which is quite a task when wanting to develop a compact unit. SLAM is widely used in the industry and stands for Simultaneous Localization and Mapping. This process is also based on dynamic processing and developing a map of its current surroundings.

The process is to use the combination of all the sensors that are attached to the robot and stitch together a map according to its estimated position. This process can map entire buildings while performing route calculations, over time.

There are several different types of these methods for generating 3D terrain mapping models. Which we will investigate and understand within this chapter. (Michael Shneier, 2015)

DEM

Digital Elevation Model (DEM) is a method used to model and present a representation of a chosen area, displaying only the bare surface of the earth i.e., excluding trees, buildings, and other objects (USGS, 2023). The selected areas to be mapped are divided into cells, which then store various height values, building a complete altitude matrix according to the size of the selected areas.

It should be noted that there are a few drawbacks when it comes to this approach. Mainly the ability to map overhanging structures and then also structures that consist of multiple height levels spanning a single cell area (D. C. Guastella, 2017).

The generation of such DEM maps is reliant on various steps that need to be done before the results are modelled. One of these processes includes the Pre-Processing of the data that produces the DEM map. There exist various sources for the creation of these DEM maps. These data sources include topographical maps which DEM maps are mainly derived from, also there is a steady increase in the use of high-resolution (LIDAR) and (IFSAR) data (USGS, 2023).

Topographical Maps are used as a data source for DEM Maps, utilizing the On Demand Topes Data feature which produces USGS-style maps from a request. Such a request usually supplies the algorithm with current location information and the desired STD of output. (USGS, 2023).

Light Detection and Ranging (LIDAR) is a relatively new technology that is used in DEM mapping. This new technology produces high-resolution models of ground elevation with incredible accuracy. Around about +-10 cm.

This technology is based on a laser scanner that works with a Global Positioning System (GPS) and an Inertial Navigation System (INS) system to produce the complete setup. This system is then attached to a small aircraft to capture data.

The system works by transmitting pulses of light towards the ground, which reflects to produce a delay of which a traveled distance can be calculated. This data is then stored in a point cloud paired with its respective GPS data to produce a datapoint surface, including structures and vegetation forms. To Produce a DEM map from this, the data needs to be fed through an algorithm to remove the unwanted structures to reveal the bare Earth surface underneath (USGS).

Interferometric Synthetic Aperture Radar (IFSAR) is a new type of radar technology that has been developed to produce clear, sharp map images which are high-quality and highly accurate. This technology produces highly accurate representations in 3d digital maps. This technology is also highly effective during nonideal conditions.

For instance, extreme weather conditions, or thick cloud cover. IFSAR works by transmitting a radar signal toward the ground at an oblique angle to accentuate the landscape details. Hidden topographical features, like subtle faults and folds, will be clearly seen using this technology versus conventional satellite imaging. The Z-axis resolution reached is between 0.625 – 2.5 depending on the region (USGS, n.d.).

SLAM

Simultaneous Localization and Mapping (SLAM) is a mapping method used by robots and autonomous vehicles to map and localize themselves in various environments. By mapping the unfamiliar environment, the robot can determine its position based on a dynamic algorithm. Dynamic meaning that it plots its surrounding, whilst positioning itself within the map, to create an accurate representation of a real-life scenario.

Slam systems are based on various types of software solutions and can be mixed and matched depending on your needs. SLAM however consists of 2 parts. These include some type of Range Measurement functionality and then also a Data Abstraction feature.

Range Measurement is a highly needed functionality in a robot. It allows the robot to measure and observe its environment, and the different characteristics that it possesses. This type of input can be achieved in various ways using different sensors.

Ranging from cameras, and other types of imaging devices, all the way to LiDAR scanners or sonar. Basically, any type of measurement sensor that can interact with its environment can be used.

Data Extraction is another essential function of a SLAM system. This is where the extraction of sensory data is collected and then processed to identify landmarks within the unfamiliar environment.

Fully functional SLAM systems will have a constant interchange between the Range Measurement, and Data Extraction centres. Whilst updating communications between the robot and other additional hardware/software technologies that are involved (FLYABILITY, n.d.).

LRFs

Laser Rangefinders (LRFs) are a technology relatively new in the world of terrain mapping. This technology was implemented in the early nineties. There exist 2 types of rangefinders, mainly 3-D and 2-D finders (Borenstein, 2003).

What are Laser Rangefinders? LRFs are a type of measuring equipment/sensor that is used to measure precise distances. It works by transmitting a laser pulse in the form of a narrow beam targeting the object of interest. This beam of light then reflects on the surface of the target and is received by a light-sensitive sensor attached to the system. This process produces a delay which is then used to calculate the distance between the system and the target, based on the bounce-back time of the light. These modules can be sensitive enough to detect the slightest change in distance resulting in a millimetre’s precision level, even over further distances (INFINITI, n.d.).

The difference between 2d and 3d LRFs?

2-D LRF

2-D laser range finders are basically what can be found on the market today. They are also known in the literature as LIDAR or LADAR sensors. These types of sensors provide a highly accurate range of measurements with increased angular resolution. This produces a plane of depth measurements that can be computed to gain information from the environment (Michal Krumnikl, 2016).

3-D LRF

A 3-D laser range finder is used to map 3d environments, as the name states. This is achieved by adding another degree of freedom to the z-axis angular rotation axis. Ideally, the sensor would be placed within a rotation mechanism that should be concentric on the axis of rotation. Enabling the light-sensitive surface to pass right through the axis of rotation, elimination of any deviation as measurements are taken.

This expands the capabilities of the 2-D range finder to be able to map a complete 3d environment. This however is paired with complex stitching algorithms to translate the data into useful information (Nagla, 2020). For the design of a rather simple robot, 3-D laser rangefinders are not usually recommended due to the complexity level, and the fact that they are very costly, bulky, and heavy (Borenstein, 2003).

Methods used to achieve terrain mapping with LRFs modules.

To implement LRFs in the use of terrain mapping, the sensor of the range finder would have to be positioned in a certain way. Specifically, the sensor should be facing diagonally towards the ground to pick up height differences, the angle which is about -11deg depending on the user's setup.

When the vehicle is moving, the laser sweeps the ahead terrain of the vehicle which produces a continuous stream of distance measurements of the terrain. This data is then fed into a Proprioceptive Pose Estimation (PPE) System which returns a digital representation of the mapped coordinates in a 2-D grid map (multi-dimensional array) described in (Borenstein, 2002).

Each of these slots within the grid contains a value that represents the height of each cell producing an accurate representation of an elevation map (Borenstein, 2003).

It should be noted that we will aim to generate, a 3D map of the previously driven terrain. No form of localization or route generation calculations will be performed. It will also not be used for navigation whilst commuting through the terrain. The ability of navigation and route generation will be considered in future development.

TRANSLATION/ROTATION ORIENTATION

In this project we will be utilizing an accelerometer/gyro to calculate this pitch and roll of the robot, at any given time. The purpose of this is so that when thew user is driving the robot manually, he/she does not tip the robot over. Avoiding an accident like this is crucial. By measuring the pitch and roll of the robot, the user will know the status of the robot’s stability while it is being driven manually by the user.

Accelerometers and gyroscopes are sensors that measure different types of motion. An Accelerometer measures proper acceleration, or better known as linear velocity. It measures acceleration of an object relative to inertia or freefall. Whereas a gyroscope measures the rate of angular velocity of an object around an axis, so this means it measures rate of angular rotation of an object.

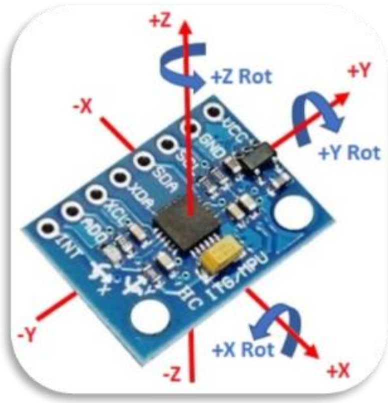

Figure 15 Example 6 Axis Accelerometer

There are 3 different types of Accelerometers. These are Capacitive accelerometers, Piezoelectric Accelerometers and Piezo resistance Accelerometers.

Piezoelectric accelerometers utilize the piezoelectric effect to get change in acceleration. What this effect is, is that piezoelectric materials produce electricity when put under physical stress. These types of accelerometers are commonly used for vibration and shock measurements. (Jost, 2019)

Piezo resistance accelerometers are a lot less sensitive than piezoelectric sensors. These types are better suited for car crash testing. How this type of accelerometer works is that it increases its resistance in proportion to the pressure applied. (Jost, 2019)

Capacitive accelerometers are the most used. How a Capacitive accelerometer works is that it uses electrical capacitance to determine an object's acceleration. When this type of accelerometer is accelerated in a direction, the distance between its capacitor plates changes as the diaphragm of the sensor moves. (Jost, 2019)

There are advantages and disadvantages to each type of accelerometer. (ResearchGate, 2023)

Type	Advantages	Disadvantages
Capacitive	• High Sensitivity • Low noise • Low drift • Low Temperature sensitivity	• Susceptible to electromagnetic interference
Piezoresistive	• Simple design structure • Simple fabrication process • Simple readout circuit	• Large temperature sensitivity • Large design
Piezoelectric	• Self – powered • Digital Output • Simple Interface circuitry	• Large design size

Table 2 Accelerometers Accelerometer Comparison

Just like there are different types of accelerometers, there are also different types of gyroscopes. Namely Mechanical, MEMS (Micro Electromechanical System), RLG (Ring Laser Gyroscope) and Optical Gyroscopes.

Mechanical gyroscopes are the oldest type of gyroscopes. These types of gyroscopes are dependent on the gravitational pull to get the stability and motion in an object. An example of this is a spinning-disk gyroscope. This example can also explain how it works. A spinning-disk gyroscope generates gyroscopic forces that resist changes in its axis and stabilize the outer frame to which it is attached.

A better way of understanding it is Precession. Precession is the change in orientation of a spinning wheel due to gyroscopic forces when an external force is applied. The right-hand rule can be used to determine the direction of angular velocity and moment of a rotating disk. Precession occurs due to gyroscopic forces that resist any change in the axis of the spinning wheel. (Smlease, 2023)

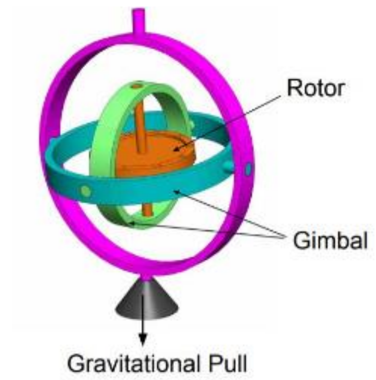

Figure 16 Gyroscope Example (Smlease, 2023)

MEMS gyroscopes are small gyroscopic sensors used in electronic devices to determine the angular velocity of a rotating body in three directions. Unlike mechanical gyroscopes, they do not stabilize moving bodies. They consist of a resonating structure or spring-mass system that converts the movement of the gyroscope into an electric current. MEMS gyroscopes are suitable for calculating yaw, roll, and pitch in a freely moving body but cannot be used to determine the angular speed of rotating shafts. Their sensitivity is proportional to the oscillating mass. (Smlease, 2023)

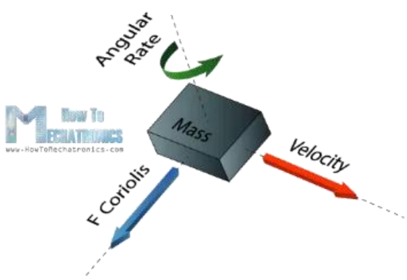

Figure 17 MEMS Gyroscope (Dejan, 2015)

A ring laser gyroscope (RLG) is an optical gyroscope that uses the Sagnac effect to detect rotation. The Sagnac effect is something that happens when you split a beam of light and make it go in opposite directions on the same path. The way the light waves meet and overlap changes depending on how fast the apparatus is spinning.

The RLG utilizes a closed-loop laser cavity filled with helium-neon gas to perform measurements, with the laser integrated within the chamber. The externally observed interference pattern is directly proportional to the rotation angle. These gyroscopes are the highest-performance available but are also the most expensive due to their complexity. (Vectornav, 2023)

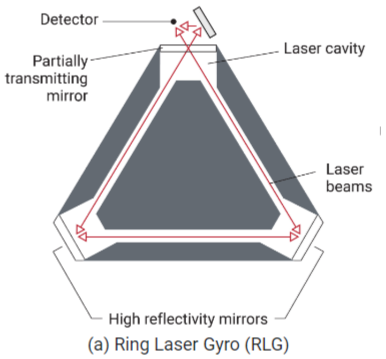

Figure 18 RLG (Ring Laser Gyroscope) (Vectornav, 2023)

A fibre optic gyroscope (FOG) is a device that uses the Sagnac effect to detect rotation. It consists of a laser, beam splitters, a detector, and an optical fibre coil. The light from the laser splits into two wavelengths that travel through the coil in opposite directions. When the beams reach the detector, it calculates the rotation rate.

The performance of a FOG depends on the length of the fibre used, with longer fibres resulting in better performance. FOGs are newer than RLGs and are cheaper, although they have slightly lower performance. (Vectornav, 2023)

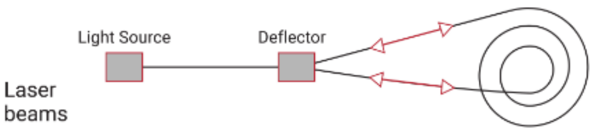

Figure 19 FOG (Fibre Optic Gyroscope) (Vectornav, 2023)

As with the accelerometers, there are advantages and disadvantages for each type of gyroscope. The advantages and disadvantages being the following (Soffar, 2020):

Type	Advantages	Disadvantages
MEMS	• Extremely small and light-weight • High sensor resolution • Indicates True North as opposed to magnetic north • Measures relative orientation on all three axes • Fast Operation	• Less stable over certain temperatures, humidity and stress • Offers more integration errors relative to FOG or RLG • Varied performance across lots of units
RLG	• High performance and accuracy • No mechanical or moving parts to create friction, so there is no drift	• Most expensive option • Doesn’t measure static angle of orientation • It is subject to relative azimuth drift
FOG	• Very Precise rotational rate information • No moving parts • Higher resolution than RLG	• High calibration requirements • Can only be used for one axis, so to get multiple readings across multiple axes, you would need multiple gyros

Table 3 Gyroscope Types

The sensor of our choice is an MPU6050. It is a Capacitive accelerometer, it is an IMU (Inertial Measurement Unit) that combines a MEMS (Micro Electromechanical System) gyroscope and accelerometer which also uses a standard I2C bus to communicate data, which a Raspberry Pi has. This accelerometer is also budget friendly, making this the sensor of choice. It is also a widely used sensor amongst hobbyists and content creators, meaning there is a lot of support and help on the internet for this specific sensor. (mjwhite8119, 2019)

To calculate the roll and pitch of the robot, from the data sent by the sensor. Some maths would need to be done based off the values given by the sensor. These values can then be used to calculate the roll and pitch angles.

Roll = ϕ = arctan (ay/az)

= atan2 (ay, az)

Pitch = Θ = arcsin (ax/g)

$g = 9.8\, m \cdot s^2$

Where yaw = 0°, or ψ = 0°

$$\begin{bmatrix} ax \\ ay \\ az \end{bmatrix} = \begin{bmatrix} cos\Theta & 0 & -sin\Theta \\ sin\Theta sin\phi & cos\phi & cos\Theta sin\phi \\ sin\Theta cos\phi & -sin\phi & cos\Theta cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

Figure 20 Pitch and Roll Equations (Wrona, 2020)

POSITIONAL RELATIONSHIP TO ALTITUDE

Altitude in robotics refers to the height of a robot or an object above a reference point (the ground or a specific target). Altitude information can be used in many ways in mobile robots, such as navigation, mapping, obstacle detection, and flight control. In mobile robots, altitude is typically measured using sensors such as accelerometers, gyroscopes, barometers, or laser rangefinders.

There are several common uses of altitude, and each use depends on the task the robot is designed for. For example, Aerial robots such as drones use altitude to fly and navigate through the air; Underwater robots such as submarines or autonomous underwater vehicles (AUVs) may use altitude sensors to maintain constant depth while navigating through the water; Autonomous Mobile Robots (AMRs) may use altitude sensors essentially to detect changes in elevation, such as climbing up and down a ramp, or traversing over obstacles.

Through the advancement of technology, the altitude of mobile robots can be measured in various ways and using a wide range of sensors. The first to be discussed is by using Acceleration translation but is also known as Inertial Navigation System which is a system that uses both accelerometers and gyroscopes to measure the altitude of a moving object. The next method proposed using a Barometric altimeter sensor or officially described as, Barometric pressure measurement, to determine altitude. Other systems have used Laser Altimetry that includes instances like Light Detection and Ranging (LIDAR) technology to measure the distance between the instrument and the ground, thus finding altitude. The final method is by, which will also be implemented in our mobile robot, is the GPS sensor.

Acceleration Translation

As mentioned, Acceleration translation is one of the methods for measuring altitude, positioning, and velocity (Tsai, 2006). It is necessary, however, to employ the dead-reckoning method for a relative positioning system. This is positioning with gyroscopes and accelerometers (Liu, 2001). The gyroscopes measure the angular rate, and the accelerometer senses the accelerations. The accelerometer that has been evaluated in this experiment is the ADXL202 produced by Analog Devices, Inc. It is a low-cost, low-power, two-axes micromachined accelerometer.

It can measure both dynamic acceleration and static acceleration. The outputs are digital signals with duty cycles proportional to the acceleration in each of the two axes and can be measured directly with a micro-controller unit (MCU) timer system. The selection of the accelerometer was convenient due to its small size, low cost, and acceptable performance. The experimental findings have helped to evaluate a low-cost solid-state accelerometer.

The accelerometer's performance is demonstrated to be adequate as a short-duration distance-measuring tool for mobile robots. With an appropriate periodic recalibration using outside measurements of position, velocity, and attitude, the low-cost accelerometer might be judged adequate for high dynamic robot operation with only 2mg of random bias drift.

Altitude using Barometric Altimeter

In addition to sensors, a barometric altimeter is another one of which can be used to determine altitude (Tsai, 2006). This specific sensor measures the atmospheric pressure to determine the altitude of a mobile robot. The principle behind a barometric altimeter is that air pressure decreases as altitude increases.

By measuring the difference in air pressure between current altitude and a reference altitude, the sensor can calculate the robot's current height above the reference altitude. The observation that was made based off the research paper was that the barometric altimeter measurements resemble the kinematics GPS altitude closely except an apparent low frequency difference.

This basically means that, like a kinematic GPS, the barometric altimeter had a short-term accuracy but is nonetheless an extremely accurate sensor for altitude in the sense of short-term accuracy.

Laser Altimetry or LiDAR

Laser altimetry, also known as LIDAR (Light Detection and Ranging), is a popular method for find the altitude of mobile robots and even more popular in Aerial robots (Ramamurthy, 2005). It is important, however, to note that it cannot be used in tanked mobile robots because of their constant height from the ground. How it works is that A LIDAR sensor is mounted on the mobile robot to provide a 360-degree view of the environment.

The sensor then emits laser pulses and measure the time it takes for the pulse to bounce back from objects in the environment. The sensor can determine the distance to objects in the environment by measuring the time it takes for the laser pulses to return and the results will create a 3D map of the robot's surroundings. Therefore, by analysing the 3D map, the robot can determine its altitude relative to the ground or other objects in the environment. One study is localizing a robot in space and the LiDAR sensor was used for detection.

The study mentioned that a horizontal-looking 2D-LiDAR sensor was the best suited for that setup instead of using the downward-looking 2D-LiDAR sensor. The advantages of a horizontal-looking 2D LiDAR sensor compared to a downward-looking 2D LiDAR sensor include:

1. Coverage: The horizontal placement detects obstacles the same height as the sensor or higher. On the other hand, downward placement enables detection of short obstacles and portholes which makes it more difficult to detect dynamic objects.

2. Obstacle detection: Horizontal-looking 2D LiDAR sensors are better suited for detecting obstacles on the ground or at low heights, such as speed bumps or portholes. Downward looking 2D LiDAR sensors are better suited for detecting objects directly below the sensor, such as the ground or a floor.

The algorithm has a relatively low computational complexity, making it suitable for use in projects where autonomous mobile robots have limited computing power. The downside to using a LiDAR sensor for our robot is the cost variable, both initial costs and maintenance costs. Because, to use LiDAR accurately for altitude measurements in mobile robots, the robot must be equipped with a LiDAR sensor and the appropriate software to interpret the sensor data. The accuracy of the altitude measurement will depend on the quality of the LiDAR sensor which will ultimately make it even more expensive for this project.

GPS

A common method that was found to be used to measure altitude on autonomous robots is through GPS. To use GPS to find altitude (Ladislav Jurišica, 2012), a GPS receiver must be installed that will receive signals from numerous GPS satellites. The robot's position is determined by the GPS receiver by comparing the timestamps of signals received from different satellites. The receiver can compute the robot's height by utilizing this data to calculate the distances between each satellite and the robot.

Overall, altitude is a critical parameter in the design, operation, and control of robotic systems. Our tank mobile robot will make use of the Barometer altimeter to measure altitude because it is relatively simple and inexpensive.

Unlike other height measuring techniques like laser altimetry or GPS, the barometer altimeter is unaffected by obstructions or interference from other sensors in the robot's surrounding which will make it suitable for our robot even in challenging locations like cities or woods. Ultimately, the choice of altitude measurement method depends on factors such as accuracy requirements, environmental conditions, cost, and available technology.

2.5 AUTONOMOUS FEATURES

Object Following

1.Method for Object Following

The objective of this capability is to enable the robot to operate autonomously, without requiring manual control from the user. To accomplish this, we aim to utilize computer vision and machine learning techniques to teach the Raspberry Pi to track a specific object. By doing so, we can provide the robot with a target to follow. In this project we will utilize OpenCV. What is OpenCV you may ask?

OpenCV is an open-source computer vision and machine learning software library. We will be using OpenCV to achieve an object detection capability in the robot and then to follow set given object to follow. (SimpliLearn, 2023)

What is Object detection? Object detection is a computer vision technology that involves identifying and localizing objects of interest within an image or a video. It is challenging, because it involves not only recognizing the presence of an object but also detecting its location and size within the image or video. (SimpliLearn, 2023)

In this project we might also make use of TensorFlow to train a machine learning or deep learning model to use with OpenCV. What exactly is TensorFlow? TensorFlow is an open-source software library that is used for machine learning and artificial intelligence. TensorFlow is a rich system for managing all the aspects of machine learning systems.

To achieve what we want to achieve in this project we may need to make use of the TensorFlow APIs to train a model for our project. To be specific, we will be using the Keras API for our project should we3 need to make a new model for our project and there isn't one that is already available for us to use. (Learning, 2023)

Now you may ask what is Keras? Keras is a high-level, deep-learning or machine learning model made by Google to implement neural networks. Keras used to be a separate API but is now embedded into TensorFlow. This specific API is slow compared to other deep learning frameworks but is a lot more beginner friendly. Thus, making it a go to for this project. (SimpliLearn, 2022)

Based on the environments the robot will be used. It would be easier to implement an object or shape following feature instead of user or human following. If you were to implement human following, then the robot could get confused as to who to follow in an industrial setting as there are many people weaking around. And even in the underground mine set-up, you might not always be the only person doing a job underground, this could leave the robot confused, and make it useless.

The robot needs a specific object to follow so that it does not get confused by surrounding objects, shapes, or people. The robot will already struggle with processing power alone to run this kind of feature, so any confusions should be out of the question entirely.

Based off some research, I have found that some roboticists make use of a co-processor to run a smoother object detection process. As the Raspberry Pi alone struggles with FPS or frames per second, when running this process alone. The FPS average out at approximately 0.9 FPS. This is very poor and will lead the robot to have very delayed reactions based on the object it has to detect/follow.

A popular choice is the Google Coral USB Accelerator. This device is a co-processor which aids in computer vision and plugs into the Raspberry Pi via USB. This would be a great and useful additive electronic device/module to our robot, however due to budget constraints and scarcity, we will not be able to make use of this device. Unfortunately, our robot will have reasonably delayed reactions when it comes to the computer vision and the FPS. This is indeed a limiting factor to its usability in a work environment. (Tayal, 2019)

When it comes to machine learning models and deep learning, we will use OpenCV and TensorFlow. How this works is that you can use TensorFlow for training a machine learning model and then use OpenCV for the prediction. What are the benefits of using OpenCV as just the prediction model? The first advantage will be that you get rid of the TensorFlow overhead, making the code for OpenCV a lot more compact and simpler. Second, someone who has no idea how the TensorFlow framework works can still just use OpenCV for predictions as they are separate, they do not need to understand how the model was trained, they can just use OpenCV for the predictions.

OpenCV also clocks at much better speeds in terms of CPU usage versus that of TensorFlow. Making use of this method will help a lot in terms of process speed for the raspberry pi. OpenCV cannot be used for training any models. There are other alternative deep learning frameworks, such as the use of Caffe, PyTorch, Darknet, and ONNX, but TensorFlow is widely used and there is more support for the use of TensorFlow as it is the popular choice for this kind of situation.

It is also possible to use an already made machine learning model. If this were available to us then we would not need to train a model with the use of TensorFlow, then we could just import the model into OpenCV which would make it easier for us to implement. To load a deep learning model, we make use of the DNN module in OpenCV, so that is how it will be done. This would be done as follows: (Academy, 2021)

Net = cv2.dnn.readNetFromCaffe (architecture, weights), this would be used if a deep learning model was made with the use of the Caffe framework, whereas if it were from TensorFlow, the architecture and weight would be swopped around in that line of code. As well as readNet would change to TensorFlow, instead of Caffe. (Academy, 2021) (autonaticdai, 2020) The architecture refers to the model structure, so this includes the layers of the model, how many nodes are in the model, what types of layers are in the model, etc. (Academy, 2021) Weights refers to the trained weights this includes parameters made up in the model. The outputs that come from this file are then the predictions that are needed.

Both the weights and the architecture are two separate files that are needed for this model to work, so both would need to be present. OpenCV works with images in BGR and TensorFlow works with images in RGB, so a conversion would need to take place to ensure more accuracy and to ensure the best outcome. Another form of processing we need to do is to apply a softmax function to get an output.

A softmax function is one that converts a vector of K real numbers into a probability distribution of K possible outcomes. It acts as an activation function of a neural network to normalize the output. The softmax function changes the values given by the network and makes it a number between 0 and 1 so that it is interpreted as a probability. (Academy, 2021)

Most common optimiser is Adam, this ensures that the model learns. A common practice is to save the TensorFlow model and save it as a H5 file format and then to convert it to a ONNX format. This is because OpenCV does not work directly with .h5 file formats.

ONNX stands for Open Neural Network exchange. ONNX is a industrial standard format for changing model frameworks, this means that you can train a model in a framework such as TensorFlow and then convert it to ONNX and then convert back to any other format of a framework.

The goal is to use a custom trained model in the DNN module of OpenCV, but the problem with this is that it does not support the .h5 Keras model directly. So, a conversion would have to take place from .h5 to. onnx, after this being done we will then be able to take the trained model and to plug it into the DNN module of OpenCV. (Academy, 2021) (LearnOpenCV, 2023) Running a continues live feed of object detection causes a raspberry pi to heat up very quickly, so in our build we will need to ensure that there is a form of cooling provided for the raspberry pi, so it does not overheat. (Rosebrock, 2017) (arjun2000ananth, 2022) (Fluxwood, 2017) (Automaticaddison, 2022) (Raj, 2017)

2.Sensors for Object Following

In this section, we will perform camera sensor selection, for the robot. This is a vital, part of the process because 2 of our features, listed above rely on the camera. This includes the Live video streaming, and then also the User following section. We will look at 3 different camera modules all supported by the Raspberry Pi. It is vital that the camera modules are supported by the raspberry pi because of the integration with machine learning libraries, to be used in the object following sections.

Figure 21 USB Camera Module for Raspberry Pi (Pi-Shop.CH, 2020)

Looking at Figure 21, we see that this module utilizes a serial form of communication, this camera also supports night vision capabilities. Surprisingly this camera has multiple OS support, from Windows, Mac, Linux, and Android. (Pi-Shop.CH, 2020)

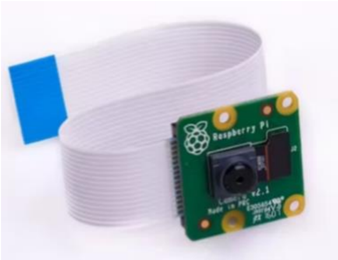

Figure 22 Raspberry Pi Camera Module 2 (Pi, 2013)

The raspberry pi module v2 camera is a widely used module in the communities. This is a nice option when developing any form of project, because of the large community base that provides support and the amazing libraries that have been developed to work with this type of camera (Pi, 2013).

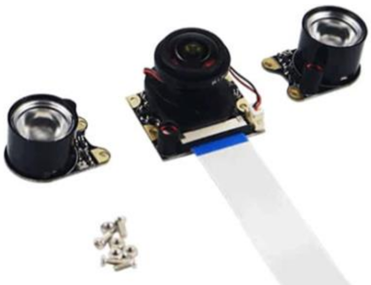

Figure 23 5MP Adjustable Focus Night Vision Camera for Raspberry PI (SHOP, 2023)

The Night Vision Camera for the Raspberry Pi is not a conventional choice of camera. This camera works on the Legacy version of Raspbian and features amazing night vision capabilities. (SHOP, 2023)

TYPE	INTERFACE	RESOLUTION	View Angle	MP	PRICE
USB RPI	USB	1920 x 1080	100	2	R1 017,82
RPI Module V2	CSI	1080p30, 720p60	NAN	8	R491,41
RPI Night Vision	CSI	1080p	75.7	5	R477,25

Table 5 Camera Comparison

Obstacle avoidance

Object avoidance is a critical capability for many types of autonomous robots, including mobile robots, drones, and autonomous vehicles. The ability to detect and avoid obstacles in real-time is essential for safe and efficient navigation in complex environments. Over the years, advances in sensor technology and machine learning algorithms have enabled the development of more sophisticated and capable obstacle avoidance systems. However, despite these advances, there are still many challenges and open research questions in the field of object avoidance in robotics.

In this section we will examine the object avoidance systems for robots, including their underlying approaches, limitations, and sensors. By providing a comprehensive overview of the current state-of-the-art in object avoidance, this review aims to provide researchers and practitioners with a better understanding of the challenges and opportunities in this field, and to stimulate further research and development in this important area of robotics.

What is obstacle avoidance?

Obstacle avoidance in robots is the ability of a robot to distinguish and navigate around obstacles.

This can be approached in different methods on the robot's design, sensors, processing capabilities and it environment.

A few Methods include:

- Relative obstacle avoidance – this method employs the use of sensors such as ultrasonic sensors, infrared or laser detectors to locate objects in the robot's immediate vicinity.

- Sensor Fusion: The method used the combination of data from multiple sensors, such as cameras, lidar, and radar, to acquire accurate perception of the environment. By integrating the data collected from different sensors, the robots gain better understanding of the size, shape, and distance of obstacles, contributing to more effective avoidance strategies.

- Simultaneous Localization and Mapping (SLAM): SLAM techniques allow a robot to build a map of its environment while simultaneously estimating its own position within that map. By continuously updating the map and localizing itself, the robot can navigate around obstacles based on its understanding of the environment.

- Machine Learning: Machine learning methods can be utilized in the training of robots to recognize and classify on sensor data. The robot can learn from large datasets and develop the ability to recognize and avoid various obstacles in real-time.

Types of Collision Avoidance Sensors

Ultrasound:

These sensors utilize sound waves to detect objects. These sound waves usually have frequencies of 20hHz. These sensors are best used for objects moving at slow speeds and short ranges.

These sensors have been used for high end reversing systems. However, has the limitation of a doppler effect when compromised by wind or external noises. because of this high-speed mobile objects/robot are not suited for these sensors.

Millimetre wavelength radar:

Radar sensors detect objects with a millimetre wavelength and can detect things up to 200 meters away. They travel at nearly light speed and have a high energy level that allows them to avoid radio frequency interference, this technology is the best overall. Most of the contemporary self-driving or assistive vehicles use millimetre-wavelength radar.

Laser:

Laser Sensors are sensors frequently used in controlled systems where reflectors are placed in the path of the machine. When reading from things with little to no reflection, it may provide false readings. Although laser detection offers incredibly fast speed and does away with radio frequency problems, the sensor limitations are that they are unable to pick up light reflected from cloth-like surfaces.

LED sensors:

These sensors utilize infrared transmitting light radiating diodes/LEDs, at around 880 nm wavelength for location. Like ultrasound sensors, they are utilized for brief extend location of less than 10 feet. Driven sensors are conservative, but they do have a limitation in zones with high temperature sources. These are primarily utilized in the industrial field of application and a few mobile reverse function auto sensors utilize them.

GPS RF detection:

These sensors utilize a centralized framework where each robot contains a radio transmitter that's associated to a GPS based following framework. When two robots are near to each other as well, notices are sent to both robots. This method is easily scalable and cover a generous number of mobile robots at high speeds, however these sensors are generally costly.

Camera:

A camera would visually recognize objects and avoid the object based on visual data. However, the camera will be used in the implementation of object following.

<u>Methods of object Avoidance</u>

- virtual attractive forces were used in a study by Khatib et al. an algorithm that produced the virtual attractive forces allocated towards a desired target as well as a virtual repulsive force against the objects that the robot will encounter along its path. the robot then redirects its movements according to the most optimal path (Khatib, 1993).

- Eun et al’s study was also an algorithm that performed obstacle avoidance through the determined the distance between a robot and an obstacle based on a virtual force exerted by an impedance force control (S. Jang, 2005).

- In visual obstacle avoidance approach cameras that face forward are utilized to identify and avoid obstacles in real-time. While the paper experiments with a small quadrotor air robot, the given fuzzy control-based solution is in theory applicable to many types of unmanned vehicles.
  The authors employ visual surveying and image processing to prevent collisions using a camera mounted in front of the robot. In this method, the data is wirelessly transferred to a laptop for more processing, where obstacles are designated with colours.
  The robot is then guided around the obstacles using this information. By moving the barriers to the left or right side of the image, the algorithm can avoid them. A potential limitation could be with the design as obstacles close by or is moving in the direction of the robot could experience communication lags between the drone and the controlling computer could result in an accident (Olivares-Mendez MA, 2012).

The LiDAR aka Light Detection and Ranging technique in object avoidance is used to measure the distances to the earth by producing light in the form of a pulsed laser. (US Department of Commerce NOAAA, 2020)

Research shows object avoidance in a proposed feed-forward centred algorithm that made use of the evaluation of a model Ultrasonic Air Vehicle embedded with a LiDAR sensor. The operator controls the UAV to a large extent, and the algorithm predicts its course for a predetermined amount of time using the operator's present inputs and future inputs. When necessary, the algorithm scans for any potential object collisions and alter the robot's path to keep it as close to the operator's desired destination as possible. (Bareiss D, 2015)

A multiple sensor approach with MATLAB consisted of the utilization of 5 ultrasonic sensors as well as a Neural Network from locate the obstacles edges and reroute itself. in this study they tested the avoidance on three various shaped. the five sensors where place on the body of the robot at various angles however five is an excessive number and the study indicated that the first two sensors to detect obstacles, the third sensor is used to locate object the as well as the depth of the object.

This study also revealed that avoidance was more efficient /effective when obstacles were found to be regular shaped as opposed to irregular shaped. Study reports indicate that sensors are unable to precisely Identify obstacles (Simone, 2018).

2.6 COMMUNICATION

Data Transmission

Transmission via Radio Frequency

A. Sivasoundari and his team done a literature review on a surveillance robot equipped with a motion detection sensor, gas sensor, fire sensor, metal detector, wireless cameras, and a video perception device. The robot communicates bidirectionally with a PC using a high-speed RF modem with a 9600-bps data rate and a transmission distance of up to 100 meters. The RF modem is self-controlled and transparent to the user interface and can be easily embedded into existing designs. The module operates in half-duplex mode and has a packet buffer of 128 bytes. After each transmission, the module switches to receiver mode automatically, and LED indicators show whether the module is transmitting or receiving data. The module is also very reliable, small, and easy to mount.

In addition to the RF modem, the robot also uses USB communications to provide multiple device functionalities, such as a modem, fax machine, and network port. The communication device class is primarily used for modems, but also supports ISDN and fax machines, as well as plain telephony applications for performing regular voice calls. The data interfaces are generally used to perform bulk data transfer.

The surveillance robot is designed to detect motion and happenings in high-jacked environments, such as nuclear power plants and other hazardous environments. It can also be used for automatic motion detection in underground coal mining, as well as to detect and visualize motions on the moon, Mars, and other planets. The robot can even be used to detect motion in earthquake-affected areas. The various sensors on the robot allow the user to detect any poisonous gas, smoke, or fire in the environment, and take appropriate measures to deal with these problems. Metal objects such as guns and rocket launchers can also be detected with robots, allowing users to be alerted to their presence in certain areas.

Overall, the wireless surveillance robot with motion detection and live video transmission is a highly adaptable and versatile device, capable of performing a wide range of functions in a variety of environments. Its high-speed RF modem and USB communication capabilities, combined with its motion detection sensor and other sensors, make it a powerful tool for surveillance and detection purposes. (Sivasoundari, 2013)

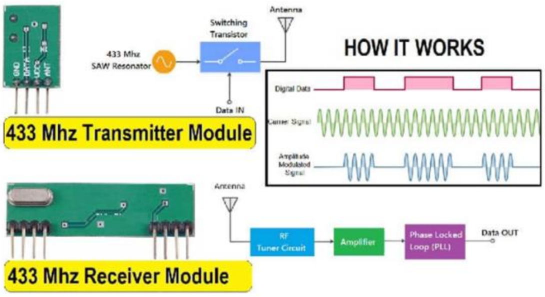

Figure 24 Radio Frequency Modules

Transmission via Bluetooth

Bluetooth wireless technology is a new standard for short-range radio communications. Bluetooth is an open standard specification for radio frequency for short range connectivity.

Jan Nadvornik developed a mobile robot that can communicate through Wi-Fi or Bluetooth. Bluetooth has a transmission speed of 720kbps, when compared to Wi-Fi with a transmission speed of up to 11Mbps Bluetooth is several times slower. Wi-Fi has a greater reach than Bluetooth while Bluetooth has a lower power requirement than Wi-Fi. It was decided due to the lower power requirement Bluetooth should be used.

Before any communication can be done the user needs to be sure that the mobile device and the Bluetooth module in the NXT are connected to each other. This is ensured by a function (BTComCheck) that the developer wrote. Once the connection is established the mobile device is ready to send and receive data. (Nadvornik, 2014)

In another research paper Prof Smita Bhuyarkar and his team aims to design a robot that can be operated using an Android App. The robot will then be controlled wirelessly through an android application via Bluetooth. The micro controller is an Arduino Uno. Paired with the Uno is a HC-05 Bluetooth module. The Bluetooth module then acts as a bridge between the Mobile devices and the electronics. This module uses a frequency of 2.45GHz and an operation voltage of 5V while maintaining a current of 30mA. The max range that this module can transfer and receive data is 10 meters. (Dipawale, 2022)

Transmission via GSM

The Malaysian Nuclear Agency used a GSM module for wireless radiation monitoring via SMS. In the design of this system the developers used the Arduino platform paired with a GSM shield for Arduino. In the hardware perspective they designed a custom PCB that houses all the necessary design blocks. The voltage regulator provides voltage supply to all the components on the board based on the voltage specifications. The PCB is designed on Telit design recommendation. The voltage regulators specified for 3.8 Volts for GSM module and the Atmega328P microcontroller. The Microcontroller controls the operations of the **GSM** module and handles data and commands from the external device. The GSM module that is used is the Telit GL8565-DUAL V3 module. This module is a compact GSM/GPRS module that is suitable for a portable and battery-operated device. The Integrated circuit acts as a GSM model that can transmit and receive all the SMSs for the GSM module.

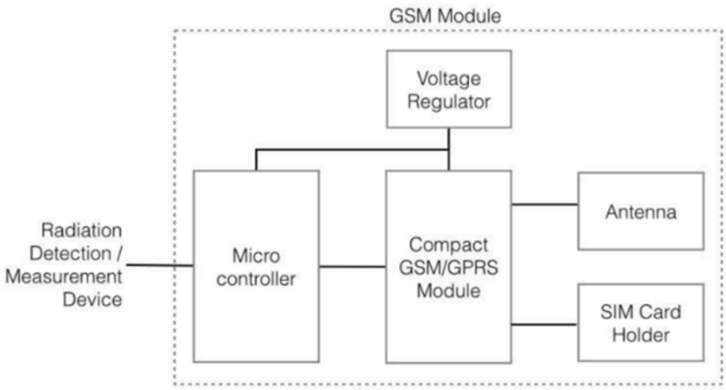

Figure 25 Block diagram of GSM PCB Module

Communication between the microcontroller and the GSM module is done by using AT commands by using standard serial connection. The GSM module is designed to enable wireless communication for a radiation monitoring instrument intended for continuous monitoring of data and alert emergency services if a disaster would have to take place. In this design there are three parameters that are necessary to complete the task, the Host number, time interval, and threshold level for alert SMS. These parameters are then stored in EEPROM of the microcontroller.

This system will then have multiple states that it could be in:

- **State Ready** is the default state. This state will then determine the other state priorities and initiate the state event execution.
- **State SMS Data Out** will send an SMS to the host at every fixed time according to the parameter.
- **State SMS IN** will receive all incoming SMS's and then execute further processes accordingly.
- **State Status Check** handles queries from connected instruments. This could be useful if the user on-site would like to view the information. This information could then also be displayed on the instrument display.
- **State SMS alert** handles alert notification SMSs to host when radiation data reach or exceed threshold parameter.

AT commands are standard instructions that are used to control a modem. For example: AT+CMGD is a command that is used to delete a SMS. In the default state the microcontroller will frequently check its serial receive buffer in case there is new notification or a new status from the Telit GSM. Incoming SMSs are temporarily stored in a firmware array buffer. Then it is crucial for the microcontroller to identify and process the SMSs that are intended for the system. The microcontroller is then expected to receive three types of SMS's:

- SMS from host/sender to check status.
- SMS from host to change configuration parameter.
- Unrelated SMS's

The GSM module enables the user to query the status of the device and data at any time. The microcontroller is programmed to identify keywords for status query and then SMS's the status to the original sender. (Oancea, 1998)

Due to the range constraints of Bluetooth. This mode of communication cannot be used for our intended purpose.

Video Transmission

In the world of autonomous robots and automation, robots are constantly evolving to replace human tasks that are repetitive and enduring. But to do so these robots are developed with highly complex and sophisticated systems. These systems are made of a mixture of electronic, electrical, and mechanical parts, that work in unison with one another to reach the end goals automatically (without human intervention).

Unfortunately, these systems are still limited to when and where they can be implemented. So sometimes when the robot can't complete its task fully, some type of human supervision is necessary. In the world of robotics and automation, the camera is seen as the eye of the robot, also called robovision. The camera takes in information about the current environment of the robot and then is translated into readable information to the robot.

This camera data can also be sent to the user's location for inspection if something goes wrong. (Sai, 2017) In this chapter, we will look at the different ways that a video link can be transmitted via a signal and then received at the other end at the user's control centre side. It should be noted that the signal must be readable by an STD computer to be later displayed on the Graphical User Interface (GUI) where all sensory information about the robot will be displayed.

Common methods for transmitting and receiving live video include:

- IP/WIFI Streaming (Most common)
- AV video stream
- WiMAX video stream

IP/WIFI Streaming

One form of streaming Video is called IP/WIFI streaming, and arguably it is the most popular form of streaming live video. This form of streaming utilizes an internet protocol called RTSP (Real-Time Streaming Protocol) and its syntax closely resembles HTTP protocol.

This protocol is an application-level protocol used for real-time data delivery. RTSP provides on-demand delivery of video or audio data by utilizing its extensible framework. Data that can be transmitted can either be a live form of video data or stored clips. RTSP is used with the purpose of controlling multiple data delivery sessions whilst providing a way of selecting the delivery channels for instance (UDP, TCP, and Multicast UDP). Running parallel to the selection of delivery channels is the delivery method. RTSP delivers a way of choosing said mechanisms based on RTP (Transport Protocol for Real-Time Applications) (Schulzrinne, 1998)

The Following Operations are supported by RTSP:

- Recaptures media data from Media Server.
- Invites a media server to a conference to either present or record all forms of media present. This functionality can be controlled by several participants.
- Adding additional media to an existing presentation.

This operations process can be visualized in Figure 26, describing the request orders of the TSTP protocol (Reindl, 2017) .

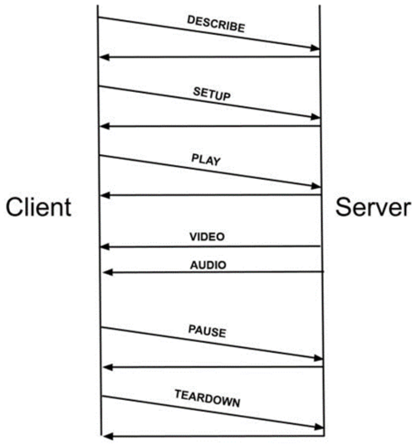

Figure 26 Real Time Streaming Protocol Request Order (Reindl, 2017)

RTSP Streaming Platform

RTSP is based on the platform of P2P (Peer to Peer) communications and by default uses the UDP or TPC protocols depending on the case. It can be utilized in places where 2 or more users can access the same WLAN network. When no common WLAN networks are available the users can utilize Live555 and its ability to act like a RTSP server to provide a streaming platform. This form of streaming supports all the most common types of media files and allows the user to stream multiple streams from the same or totally different files if need be. (Reindl, 2017) It should be noted though that when the user exits the range of the WIFI, streaming capability will no longer be available. This limits the streaming capabilities to small areas like home or office buildings.

AV video stream

Before we look at how the composite video is transmitted. We need to understand what Analog video is. To do this we get a basic understanding of what analog video is how analog video signals work and then the types of systems they use to function.

What is analog video?

Analog video is seen in the literature is seen as an electrical signal. Specifically, a one-dimensional signal, which samples the intensity of the video patterns i.e. (vertical and temporal coordinates) and converts them to an electrical representation. This one-dimensional signal can be thought of as an array, with each space representing a certain time frame, the information stored in it needs to be displayed. The sampling of this array uses a method called raster scanning. Raster scanning involves the process of progressively scanning a frame in a horizontal manner with a slight downward trajectory. This process continues until the bottom of the screen is reached. From there the process starts all over again in the top left corner of the frame. The speed at which Raster scanning completes the screen is called the frame rate and usually ranges from 25-30 frames/s. Paired with this is a functionality called the refresh rate, which is used to eliminate the amount of flickering that the human eye picks up. This is achieved by having a refresh rate above 50Hz. (R.Bull, n.d., pp. 32-34)

Analog video systems.

Looking at the different analog video systems there exist 3 main types.

Types:

- 625/50 PAL
- 625/50 SECAM
- 525/60 NTSC

The different types of systems are liked by different countries and what standards they implement. These systems are all based on composite video systems. Composite video is made from single-channel analog video transmission. It works by first band-limiting the chroma components and then combining the results with the luma components, which results in a video signal. (R.Bull, n.d., p. 35)

How Analog video is transmitted.

After the light is translated to a composite video format by the camera. The signal is then fed by a wire towards the Radio transmission module. This Video Transmission works by taking the video signals from the camera and transforming them into radio waves, with the respective frequency. Typically, this frequency is around 5.8 gigahertz (GHz) for most composite video transmitters. The transformation method used to convert the video signals to radio waves is known as modulation. Radio transmission works by transmitting the encoded analog video signals into the air by amplifying them and then passing them through the antenna on the transmitter side. The process works by passing a current through the antenna, which in turn produces an electromagnetic field/radio wave. This signal is the result of the current flowing through the antenna and is linked to the frequency directly. For example, if the system works on the frequency of 5.8 GHz, the antenna will experience around 5,800,000,000 current pulses per second through it to produce the chosen video frequency (OSprey, 2018).

How Analog video is received.

After the Radio waves have been transmitted. It is then picked up by the receiver which is paired on the same frequency as the transmitter. The signals that are picked up by the RX antenna carry the encoded video feed of the transmitter side of the system. The radio waves moving across the antenna produces a change in current flow through it. Because current and voltage is linked by the equation (voltage = current * resistance) The antenna will also produce a change in voltage, which is measured and then decoded to recover all the transmitted video signals, resulting in a composite video output. (OSprey, 2018).

How to convert Analog video to USB?

As mentioned in the intro of this section, we talked about how all the data and sensor readouts will be displayed in a suitable Graphical User Interface (GUI). Keeping this in mind we need to convert the above-mentioned composite video outputs into some type of format that a computer can read it. From what we know, the output of the receiver is a composite video format. This format produces analog signals and the type of signals that a computer usually read via its USB ports is a form of serial communication, which is a digital type of format. From this, we can build or source some type of analog-to-digital converter. However luckily for us, there exists a product on the market specifically for the purpose of converting Composite Video into a readable USB signal for computers. This device can support all types of composite video formats and is relatively easy to come by and use. It works by translating the composite video format into a suitable serial communication/digital signal so that the computer can read the signals. From there software that is integrated on the computer can communicate with the USB device and display the received composited video output. (Startech, n.d.)

WiMAX Video Steaming

Before we jump into the detail of how WiMAX can be used for streaming a live video feed, we need to understand what WiMAX is.

What is WiMAX?

WiMAX was introduced in 2004 (Anon., 2016) and stands for (Worldwide Interoperability for Microwave Access) and was developed according to the IEEE802.16 Standards. WiMAX is an emerging telecommunications technology that utilizes the same principles as WIFI technology but improves it. This technology is not only more efficient than Wi-Fi but provides higher speeds and covers larger areas of reach/signal. WiMAX can handle thousands of communications to a single base station while providing a speed connection of up to 70 Mbps over an area spanning as large as 48 Km. (freewimaxinfo)

WiMAX video TX and RX Fundamentals.

Now that we understand what WiMAX is, we will look at how we can utilize this technology to stream video feed over a distance wirelessly. Note that all the information that is referred to below is from the (International Journal of Technical Research and Application) which has performed research on the topic of Video Transmission Using Wireless Networks utilizing WiMAX. (Omkar Ketkar, 2016). Because of the similarities between WiMAX and conventional WIFI, the methods used to perform transmission and receival of data are very similar to how any other form of wireless data transmission works, like how AV transmission works discussed in the previous chapter. What I mean by this is that both sides of the system receive a form of input (either an analog signal from a camera or a radio wave signal from the transmitter) which is then processed and transformed into the necessary format to complete the next step which will be to output the data (either as an analog signal on the receiving side of the system or a radio wave that will be transmitted). See (Figure 27) to understand WiMAX Communication Fundaments.

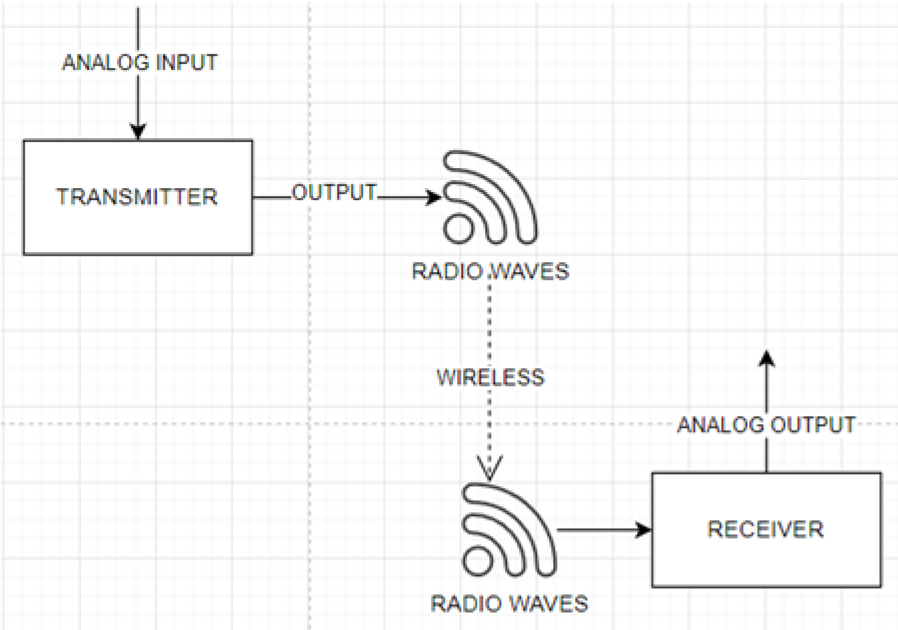

Figure 27 WiMAX Communication Fundamentals

How does WiMAX Video Transmission work?

BLOCK DIAGRAM

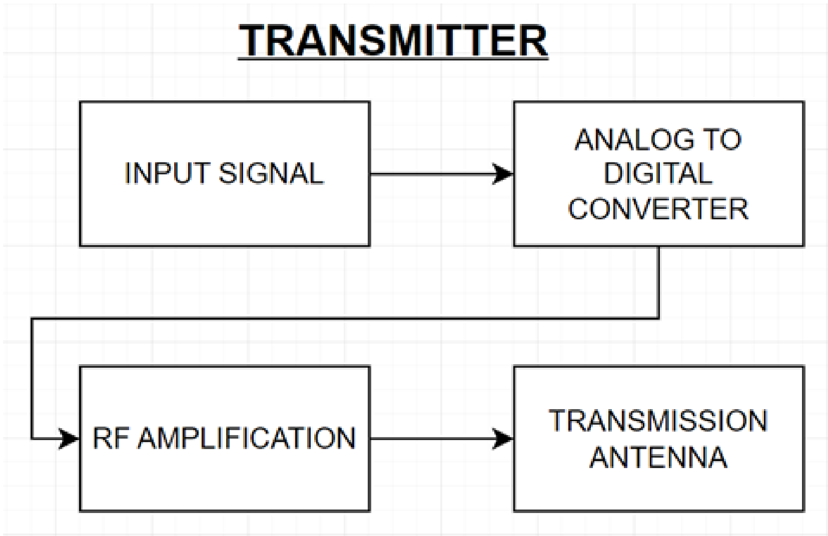

Figure 28 WiMAX transmission Block Diagram (Omkar Ketkar, 2016)

INPUT SIGNAL:

Analog signals are received from the camera and will be transmitted to the analog-to-digital converter.

ANALOG TO DIGITAL CONVERTER:

Analog Signals are converted to Digital Signals by converting the physical analog quantities to digital pulses representing bits.

RF AMPLIFICATION:

This is where the encoded signals are amplified to high-frequency radio communications. The frequency of the maximum can is changeable, by varying the inductance and capacitance of the tuned circuit.

TRANSMISSION ANTENNA:

This is where the electric power is converted into radio waves using the antenna/aerial.

WiMAX transmission works by receiving an analog signal (input) form a camera. The camera signal is then directed into a fixed 12MHz filter, which allows for high-quality low-interference video transmission. After this process has been completed the signal is sent to a modulator. The modulator then transforms the analog signal into a usable digital signal which can be amplified by the RF Amp and transmitted via the antenna. By utilizing this process WiMAX allows for the transmission of heavy-load data or high-quality video transmission. This is achieved by the utilization of high-frequency radio waves paired with a wide bandwidth. This system utilizes 2.4GHz as its operational frequency.

This allows WiMAX to achieve long-distance transmission whilst utilizing very small antenna sizes. When analog data from the camera is fed to the WiMAX transmitter, the signal that was input is typically a lower frequency, around about 12MHz. Because of the quite substantial difference in frequency, between the WiMAX transmitter and the camera. And then the large bandwidth of WiMAX. The signal is sent at an alarming rate, resulting in an almost instantaneous receival of footage at the receiving end of the system.

How does WiMAX Video Receival work?

BLOCK DIAGRAM

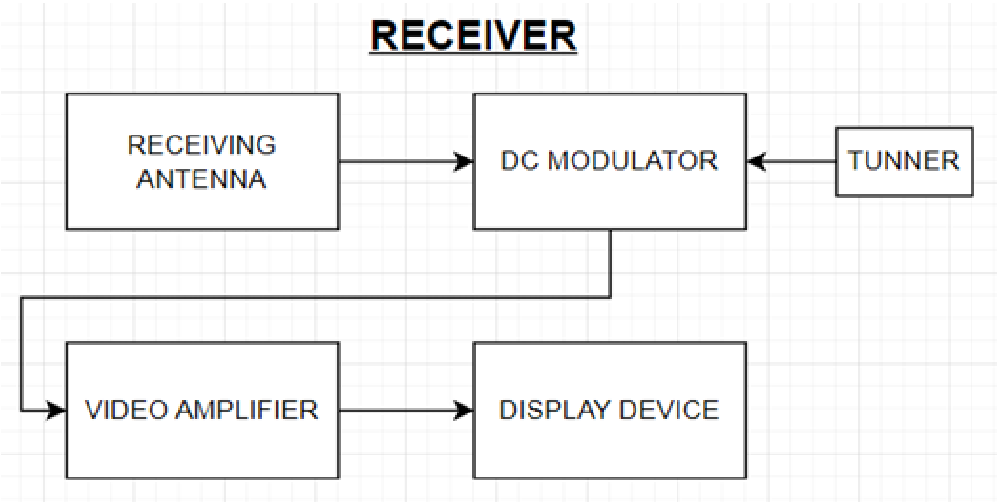

Figure 29 WiMAX Receival Block Diagram (Omkar Ketkar, 2016)

RECEIVING ANTENNA:

The antenna/aerial gets input in the form of radio waves and converts the signals into electric power, that can be read by the radio receiver.

DC MODULATOR:

This device receives input from the antenna and translates the signals to the correct voltages for the video amplifier.

RF VIDEO AMPLIFIER:

This device handles the filtration of video signals, to ensure the product can be displayed in high resolution.

TUNER:

Allows the configuration of the device so that the system can be optimally paired for the best output possible.

DISPLAY DEVICE:

This is a device used for the presentation of the footage that has been decoded at the receiving end of the system.

After the video signals have been transmitted, they should be received. This is done by utilizing the opposite process that was used in the transmission of the analog video signals. The process starts with an antenna receiving the radio waves that have been transmitted, utilizing a monopole antenna.

The signals generated by the antenna can range from 2.4 GHz to 3 GHz. This is why the demodulator is paired with a tuner, to synchronize the frequency received, with the demodulator’s operational frequency. After this data have been successfully demodulated, it will continue its way from the demodulator to the video amplifier. The video amplifier has the final job of amplifying the video signals so that they can be read by a displaying device, such as a TV.

2.7 POWER SOURCE

Robots are electronic devices that require a stable and reliable source of power to operate. A power supply is an essential component of any robot due to its provision of the electrical power needed to drive the robot's motors, sensors, processors, and other components. The power supply ensures that the robot's electrical components receive the correct voltage and current levels, which are necessary for them to function correctly and without damage.

Types of power sources used (e.g., batteries, fuel cells, hybrid systems)

Battery power supply

Batteries are the most important component of a mechanical framework. They can be classified into categories: rechargeable or non-rechargeable. Non-rechargeable batteries provide more power based on their measure and are the more reasonable option for certain implementations. Alkaline batteries are budget friendly, and lithium batteries, on the other hand display a longer rack life and superior execution.

Rechargeable batteries consist of lithium-ion, nickel metal hydride and silver zinc batteries as well as lead batteries and nickel-cadmium (NiCd) batteries. lead and NiCd batteries usually provide a lower voltage than alkaline batteries. They come in packs with their unique connectors. Gelled lead batteries are broadly utilized and able to give up to 40Wh/kg of power. While Lithium-ion, nickel metal hydride and silver zinc batteries produce a lot more energy density [2]. A battery is competent of conveying current.

The material utilized to allow the negative anode to discharge electrons amid a chemical response. The chemical response discharges electrons at the anode and at the cathode another chemical response consumes them. electrolytes are a fluid that acts as a separator that isolates the anode and cathode or electrodes. oxidation and lessening are the results of the negative electrode losing electrons and the positive terminal has absorbs the electrons. Batteries need both chemical reactions this is called a redox response pair. (Rohit kumar, 2014).

Battery Technologies

Lead Acid

- Lead acid had been around for the longest of all the battery technologies.
- most advanced of the battery choices
- Gelled lead acid the least hazardous
- broadly utilized.
- gives energy densities of up to 40Wh/kg.
- have easily built linear discharge curve.
- good shelf life
- big and heavy compared to other types of battery technologies.

Nickel Cadmium (NiCd)

- broadly utilized in rechargeable implementations
- used when high current is required.
- have good energy density.
- have high amount of discharge cycles in comparison to lead acid batteries.
- have the highest specific power.
- easily accessible
- heavier than lead batteries
- inexpensive

Nickel Metal Hydride (NiMH)

- provides advances improvement in the energy to weight ratio compared to NiCd.
- Environmentally friendly
- High self-discharge rate
- Unreliable for long term storage
- power density is lower than NiCad's.

Lithium Ion (Li+)

- provide more energy density than NiCd and NiMh
- have a low weight and low delf discharge for their output of energy compared other batteries.
- similar volume to NiMH
- limited power, only for low current usage
- hazardous if not charged cautiously as Li+ becomes volatile as its temperature increases and the batteries internal resistance decreases this may ignite if left unsupervised.

Silver Zinc (AgZn) and Silver Cadmium (AgCd)

- Used in electric devices where high-power density is mandatory.
- Up to 15 times more expensive than other batteries
- Used in military application.
- Limited to 20% of the volume and mass of an equivalent lead battery
- Relevantly shorter shelf life than
- lifetimes of 100-250 cycles.
- Have stable voltages.
- Independent of state of discharge.

Sodium Sulfer (NaS) and Lithium Chloride (Li-Cl)

- Used in electric devices that need high power density and energy.
- Need elevated temperature operations.
- Sodium Sulfer systems work at 250C.
- Lithium Chloride systems work at 350C.
- Are both hazardous when involved in collusions or accidents.

Nickel Zinc (Ni-Zn)

- Have advanced capability in power specific energy.
- cycle life limitations (Dowling, 1997).

Batteries utilized in the FU-fighters robotic soccer team in 2002 were in packs containing 8 NiMH units averaging a 2000mAh powering the robots several for several hours . The measure of the batteries permitted them to be set at the underneath the robot. The NiMH batteries discharge curve allowed energy extraction from the batteries and the voltage to only drop drastically if the capacity limit is reached.

This was often vital for applications, as it would be undesirable for the voltage to degrade continuously during the use of the battery's energy. This energy had a rate of 3C with slight voltage drops. "The NiCd batteries are appraised at 700 and 1200 mAh. The NiMH batteries at 1600 and 2000 mAh." Considering that NiMH stored greater energy per gram than the poisonous NiCd batteries, this lead to a progressive substitution of NiCd by NiMH batteries.

Generator Systems

Generator power supplies utilizes gasoline and other fuels to produce the high energy densities in electric devices, the produce more energy density than batteries. these systems contain gasoline energy content that supersedes lead-acid batteries 200 times in weight comparison basis. [Kalhammer95] this indicates the superiority of generator power supply over battery power supply.

This system utilizes the gasoline and converts it into energy through combustion in an engine. This engine output is utilized to power an alternator that gives power for many applications, commonly spark ignition in various engines. The diesel cycle is an exception to this process as it utilizes compression of air and heat to ignite fuel. Once converted from fuel energy to mechanical energy it is only 30-40 percent efficient, the rest of the energy is wasted to heat and shed proper engine operation.

These systems operate on a dual stage system that continually charges a battery. Whenever the system is immobile the system continues utilizing fuel even with no -load conditions. An automatic starting system uses other technology as well as small storage systems to mitigate this. Elastic storage were used an auto-start that eliminates idling periods [Killgoar80]. unfortunately, even in the implementation of the smallest generation, they are much bigger and heavier than most small robot to bear. Honda’s lightest generator at the time of its release “the EX350” it was considered relatively quiet.

Characteristics:

- 300-watt unit
- 2 hours runtime
- units provided 5.4kWh
- 80Wh/kg power source on a full tank

This made the generator a good candidate compared to batteries however its size was still an issue. Additionally with the increase of the fuel the energy to weight ratio increases therefore worsening the power to weight ratio. This resulted in the disadvantage of high fuel usage on a small engine resulting in little power but the production of lots of energy.

This therefore eliminates the use of this power supply for our system as our robot will need a good power to weight ratio to navigate various terrains. Another factor to be considered is the fact that running such a small robot on fuel would be expensive not only with the hiking fuel prices but also the initial purchase of a generator opposed to other power supply methods. In future implantations of a bigger payload that needs a significantly greater amount of power a generator may be option. (Dowling, 1997)

Hybrid power supply

Power sources can have restrictions, these include the sources' ability to convey little sums of energy for extended periods of time but incapability to convey high power for short time periods. When designing such robotics systems, load variances can drive specifications well past normal needs because of "momentary high loads" that every so often happen. Automobiles are a example of this: high-power is as it were required for speeding up or slopes but for most steady driving only a small part of accessible control is required. For this reason, a two-stage or hybrid power system is the most practical approach.

The 2-stage approach suggests the first stage to "continuously feed energy to the second stage or alternatively have 2 systems implemented parallelly as required. hybrid systems approach can be implemented in two ways: parallel or series, series systems the primary stage gives a little sum of energy continuously while the second stage gives high power for brief durations.

In electric car series hybrid systems, a parallel arrangement gives a mechanical association between the motor and wheels as well as an electric engine to drive the wheels. With this setup an electric engine can be utilized for flat surface diving where the mechanical output power is less for increasing speed. The advantages of this hybrid approach is that in series configuration the primary stage can be measured to average loads can be optimized for steady power output.in small robots the overall engine efficiency can be doubled. (Dowling, 1997)

Battery Level detection and Longevity techniques

In this project we would like to display the robots Battery Percentage on the GUI, so that when the user is operating the robot manually. The user can see the battery percentage of the robot while operating it, then the user can decide when to drive it back based off the battery percentage, as the user will not want to walk and go fetch the robot when it runs flat.

The C rating on the battery represents the discharge rating of the battery. To work out the Max current you can do the following calculation:

C rating x Capacity (AH) = Max Current (A)

45 C x 3,3 Ah = 148,5 A

(Cowan, 2020)

There are a few facts to know about LiPo batteries to understand them, and to maintain them properly to keep longevity of the batteries. These facts being the following:

- A fully charged LiPo battery is 4.2 V per cell.
- A LiPo cell should never be discharged lower than 3.0V
- Proper LiPo storage voltage is 3.8 V per cell.
- A LiPo cell nominal voltage is 3.7. (Ampow, 2018)

Based off the nominal voltage per cell and based off the values given in the tables below, a function was drawn up by (Ibrahim, 2015), this function is used to calculate the percentage of the LiPo battery. The function, as well table of values are below:

$$123 - \frac{123}{(1 + (v/3.7)^{80})^{0.165}}$$

Figure 30 Percentage of Battery Equation

Seeing that we are using a 3S battery, we then change the 3.7 value to 11.1. This represents the voltage of the battery. This is the only alteration to the function we would need to make.(Gibson, 2021)

Lipo Voltage Chart						
Voltage	1S	2S	3S	4S	5S	6S
Voltage	3.7V	7.4V	11.1V	14.8V	18.5V	22.2V
Fully Charged Voltage	4.2V	8.4V	12.6V	16.8V	21V	25.2V

Figure 31 Lipo Voltage Chart (Ampow, 2018)

Capacity %	3S Lipo Voltage
100	12.6
95	12.45
90	12.33
85	12.25
80	12.07
75	11.95
70	11.86
65	11.74
60	11.62
55	11.56
50	11.51
45	11.45
40	11.39
35	11.36
30	11.30
25	11.24
20	11.18
15	11.12
10	11.06
5	10.83
0	9.82

Table 6 Battery Voltage Percentage (Ampow, 2018)

Using the function given, and table of values. We can then make an accurate Percentage calculation of the Battery Level. This will ensure that the user can respond accordingly to the Battery Level.

To read the voltage of the batteries, we would need a Voltage divider to get 12 V down to 5 V to be able to read on analog pins. When you get this right, you will be able to then read the voltage on the micro controller.

The formula used to get your voltage output is as follows:

Vout = Vin x R2/R1 + R2 (Tesalex, 2017).

Using this formula, we can determine just how much resistance we would need to change our 12 V into 5 V so that it can be read on the module.

Keeping all this information in mind. We can then apply all functions and modules needed to the robot, to be certain of accurate battery voltage, and percentage data. This can prevent the inconvenience of going to physically having to fetch the robot if it were to run flat a distance away from the user. It can also ensure longevity of the batteries so that the user does not damage the batteries by over draining them.

CHAPTER 3 (System Analysis & Design):

3.1 INTRODUCTION

Before delving into the construction of this payload we need to be accepting of the numerous challenges and opportunities in the development of the payload. With the previously mentioned robotic features we are planning to implement our robot needs sensing, perception, planning, and control capabilities to navigate various terrains. Moreover, the payload needs to integrate mechanical, electrical, and software aspects seamlessly to ensure efficient and reliable operation.

In this chapter we focus on the system modelling and architectural design of our autonomous payload robot. we will be addressing the design of the payload in relation to mechanical, electrical and software applications of the robot. The mechanical designs contain CAD models and design drawings of the robot's physical build and components, specifying materials, dimensions, assembly instructions, and optimization techniques for seamless implementation. The electrical schematics focuses on the interconnections and wiring of electronic components of the payload, this includes sensors, motors, power supplies, and control systems. Lastly, we will investigate the software architecture aspect of the robot, its software modules, interfaces, and data flow diagrams that enable its autonomous capabilities.

In essence, this chapter aims to contribute to further advance autonomous payload systems, and provide insight, methodology, and best practice for our design, modelling, and architectural development of the payload robot.

3.2 DESIGN METHODOLOGY

To achieve our project objectives, we will be using an adapted combination of methodology. Tailored to the previously mentioned objects of the payload the project methodologies of choice will be Design Thinking, Case Study and Agile Methodology.

Design Thinking:

Design Thinking is based on a user-oriented approach it aims to develop innovative ideas or model (Muller, n.d.). this methodology encourages the three Ps of Design Thinking People, Process and Place when den coming us with an idea. A fourth P can be Partnerships, since many partners must be involved in the development and implementation of ideas (Muller, n.d.).

Case Study:

The case study methodology involves an in-depth examination a problem. It uses may differently sources and document analysis to gain a comprehensive understanding of the problem (Sarah Crowe, 2011).

Agile Methodology uses the approach of segmenting a project into various individual small tasks. It uses iterative and incremental development and continuous collaboration, flexibility, and adaptability. The small tasks also known as sprints. After each sprint there is a team reflection on the task to see if there is need for improvement and plans for strategy adjustments for the next sprint (Laoyan, 2022).

Due to the structure of the block and the time limitation we must complete the robot. An adapted methodology would work best to achieve the project objectives. Implementing Agile methodology would help in accomplishing our project deliverables in a timely manner. After each deliverable is complete the team meets to improve our work and plan for the next approach, also giving time for our lecturer to provide useful input.

Case study is important in gathering information related to the project. the use of prior studies and data is important when designing a cost effective and efficient payload model. Case studies help prevent pitfalls that could been avoided, and good research and data will allow us to leverage on the discoveries of related studies. With the implementation of the features, we will be implementing case studies have information on how to apply these features our payload.

Our project design is a payload robot that operate manually or autonomously. Briefly summarized the payload is designed with 8 wheels and 2 tracks, this allows the robot to navigate various terrains. the payload is powered by 2 motors that spin the top wheels and suspension on the bottom wheels. The centre body frame will hold all the essential electrical appliances as well as a load carrying platform.

The sensors of the payload will be paced in their respective positions to achieve their respective feature objectives. Our payload will be linked to user interface system where data can be observed. The body construction of the robot was made using the design thinking methodology, if design flaws were found those issues would have to be brainstormed to resolve them. While elements like the features used case study methodology, regular consultation with published data.

3.3 MECHANICAL DESIGNS

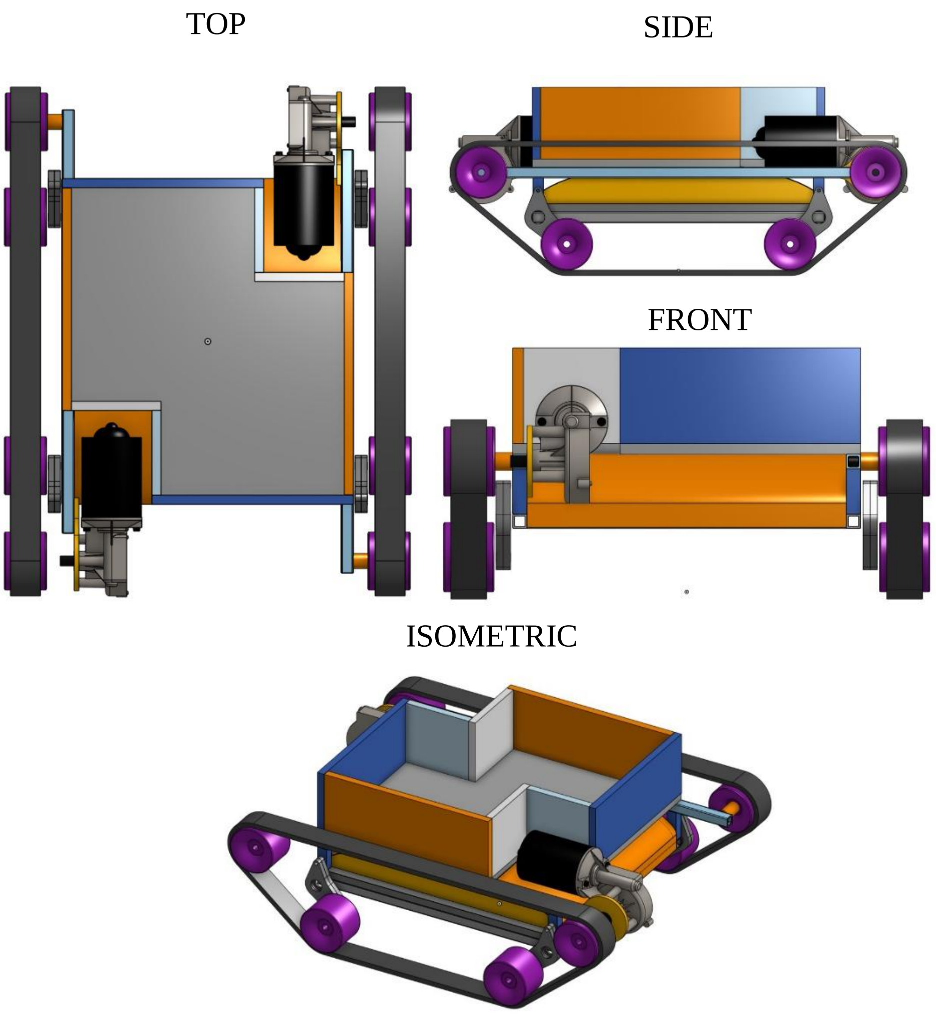

Figure 32 Robot Design

The mechanical design of the payload robot is broken down into:

- Body
- Motors
- Suspension
- Frame
- Wheels

The body of the robot is designed to carry the payload electrical components and the loading bay that will be used to transport loads on. This bay is design to sustain this without causing damage to itself while also ensuring that these loads can carry objects without dropping and damaging these sensitive electrical components from exposure to harmful substances.

The rectangular body is supported by a chassis designed ladder frame. our selection of is frame was to distribute the weight efficiency the frame is designed with the strongest materials to support the robot and its load supporting the rest of the body the frame. Our main objective is to carry loads of up to 5kg. without through various terrain.

The design dimensions:

- Width 400
- Length 500
- Height 210
- Weight: 8kg

To move the payload our design makes use of motors in each wheel that will steer and drive the robot. The size of these motors is determined by the mean drive power of the robot. The mean drive power of the robot will then be determined by the terrain conditions. Paired with these motors are the wheels, when compared to flexible tire rubber the wheels are rigid with grousers to improve the performance.

Wheels and Track

Our payload utilizes skateboard wheels with a rubber track. The skateboard wheels provide a cheap functional method of mobility and are easily attainable. They have good traction and manoeuvrability across many different terrains. The rubber track is useful in navigating the robot on uneven terrain including stairs, while skateboard wheels offer the robot stability and control on flat surfaces. Skateboard wheels used with a rubber track, contribute to a low centre of gravity, enhanced stability, and balance. This system allows the robot better control and mitigates the potential risks of the pay load tipping over when navigating uneven terrain.

3.4 SUSPENSION

Through the case study methodology, we filtered the various suspension types to improve performance and handling on uneven terrain on smaller payloads. Our chosen suspension types include:

- Coil spring suspension

This is a common suspension method discussed in the literature review. Our selection was primarily due to ease of implementation access and budget constraints. However, Coil springs have excellent shock absorption capabilities, this means they can absorb and dampen vibrations and impacts the robot may face as it moves. This is effective in protecting the payload from potential damage caused by rough terrain or sudden movements. springs also offer more versatility allowing the for customization or alteration in the cases where performance optimization is necessary, we can exchange springs to achieve the desired outcome.

3.5 SYSTEM MODELLING

In this section we will look at the system of the robot and how the robot has been split into subsystems to allow for the efficient and effective communication between all the system that are incorporated into the design. The process will begin with a context diagram, giving the reader some insight into whole picture of how the system will look. From there the system will be broken into more detailed parts, going a level deeper into how the final system will operate. From there a functional wiring diagram can be drawn up to help with the development of the robot.

Context Diagram

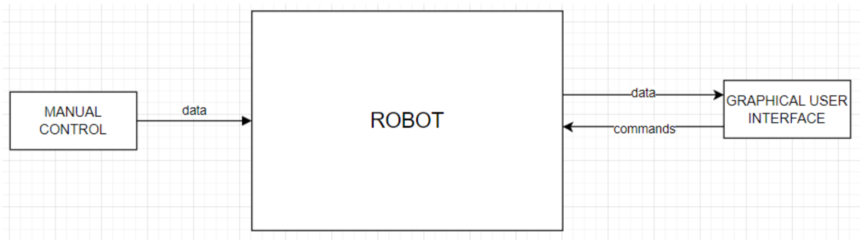

Figure 33 Context Diagram

Level 1 DFD Diagram

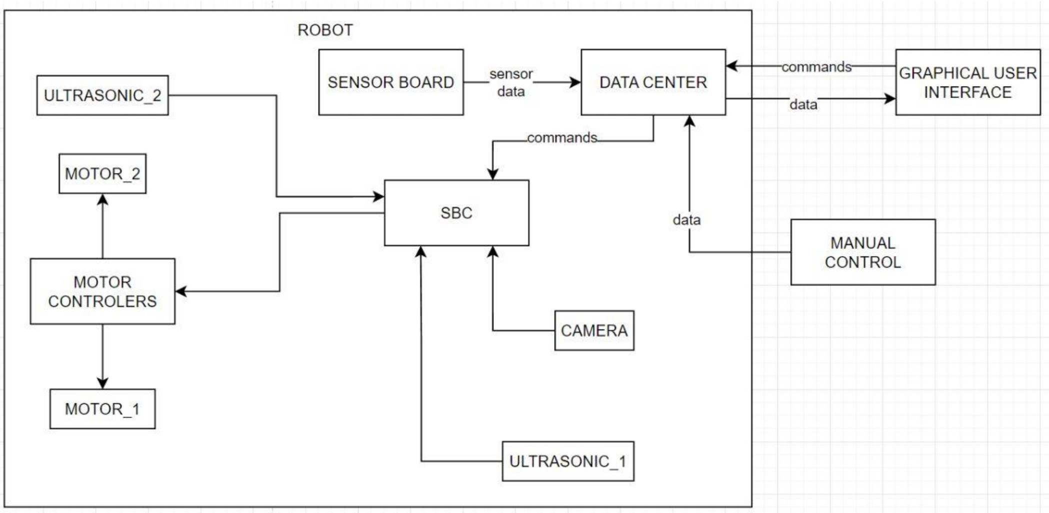

Figure 34 L1 DFD

Level 2 DFD Diagram

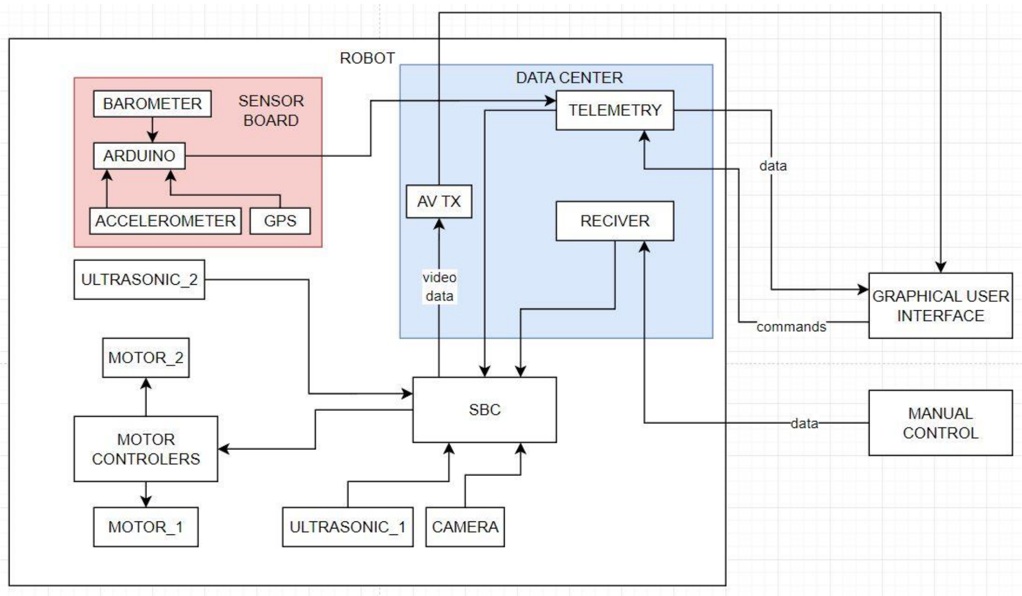

Figure 35 L2 DFD

3.6 ELECTRICAL SCHEMATICS

All the circuits shown in this section was derived from the Dataflow diagrams, and follows the same principles discussed in the System Modelling section.

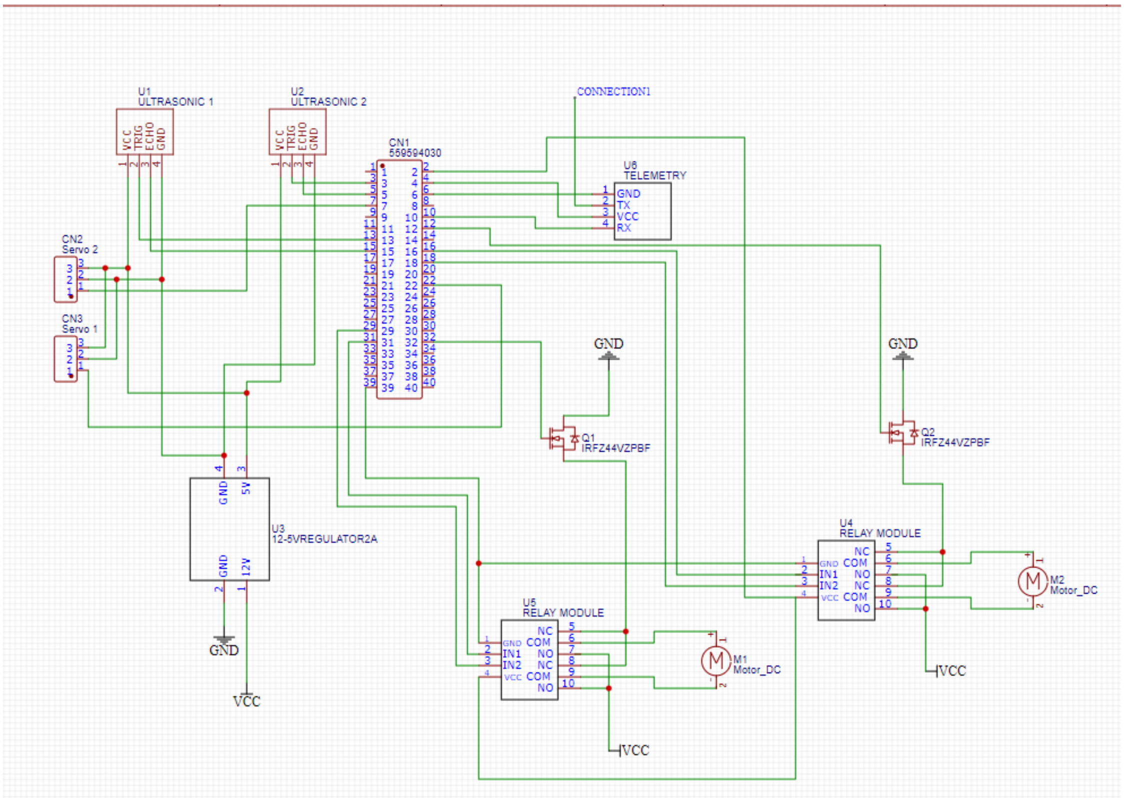

Figure 36 Main System Board Circuit

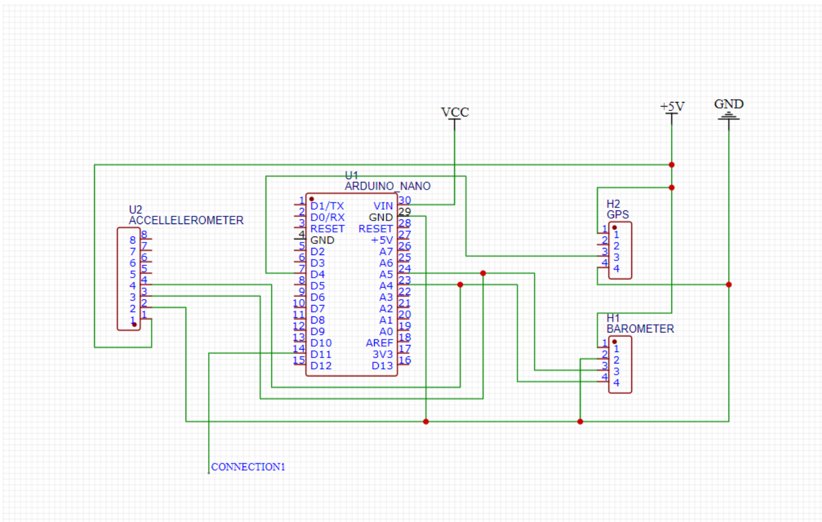

Figure 37 Sensor Board Circuit

The electronics components design for the robot are responsible for the communication devices, computers, and sensors necessary to operate the robot. Paired with the electronics is the thermal management system. This would include any cooling necessary to maintain the hardware temperature in the robot at operating levels.

POWER SOURCE

The power supply of our payload robot was decided with the consideration of the following:

- Robot size.
- Cost
- Necessity for recharge/refuel.
- Lifetime
- Weight
- Energy and Power per volume required.
- Voltage supply and conversions

With the consideration of our limited budget, resources and the best option was a battery power system this met our power supply requirements. Our proposed payload is small and will need a power supply that will weigh the least to allow for the payload to carry more weight. having any other previously mentioned power supply methods would be too expensive and complex while batteries provide a straightforward method to fulfil the payload power requirement.

Our implementation will use two lithium-ion polymer batteries. These are rechargeable batteries that use polymer and lithium. An estimated voltage of 12V is needed and will be supplied by both batteries connected in parallel.

Our Battery Specification:

- Minimum Capacity: 3300mAh
- Configuration: 3S1P / 11.1v / 3Cell
- Constant Discharge: 45C
- Pack Weight: 271g
- Pack Size: 138 x 43 x 24mm
- Charge Plug: JST-XH

MOTORS & MOTOR CONTROLLERS

Taking the power supply design into consideration it should be noted that all the motors that will operate on a voltage of 12V. This is due to the constraint, of the type of power source we have chosen and should be considered when selecting any type of electronics, that will be integrated into the system. They will make use of 2 motors; one drives the left-hand side of the robot and the other the right-hand side. This drastically impacts the amount of budget. we opted to design the model with DC Geared motor for, as it they are easily obtainable and relatively simple to work with.

SENSORS

1. GPS

Our payload robot will utilize the NEO-6M GPS module. its inexpensive cost, simple integration and suitable performance make the NEO-6M a good candidate for GPS tracking of the pay load. additionally, the modules are graded and used for electronic devices on road vehicles.

2. ACCELEROMETER /GYRO

The MPU6050 is also very cheap and attainable. Making it a go to choose for our project.

3. CAMERA

The design uses RPI Night Vision cameras as they are in a reasonable price range as other models and this camara incorporates a night vision capability. This is a very useful functionality that can come in handy, essentially since we want to develop a robot that will operate in mining conditions where there might not always be adequate lighting. Because of this reason, we will choose the Night vision camera for the use of this project.

4. ULTRASONIC

When designing for object avoidance, sensors are important to detect and trigger an output to avoid obstructing objects. Due to multiple factors taken into consideration such as the cost, accuracy, and compatibility with our single board computer. Our ultrasonic sensor of choice will be the HC-SR04 ultrasonic sensor. Our payload robot will use 2 sensors for object avoidance as mentioned earlier in previous sections.

3.7 SOFTWARE ARCHITECTURE

Our proposed software architecture is shown in figure 27. The computer system will consist of two layers. The high-level layer with task-specific vision, planning and control modules, and the low-level layer which serves as an interface between hardware (sensors and actuators) and the high-level layer. The low-level modules execute commands published by high-level modules, receive, and process sensor data, and publish processed sensor data for consumption by high level modules.

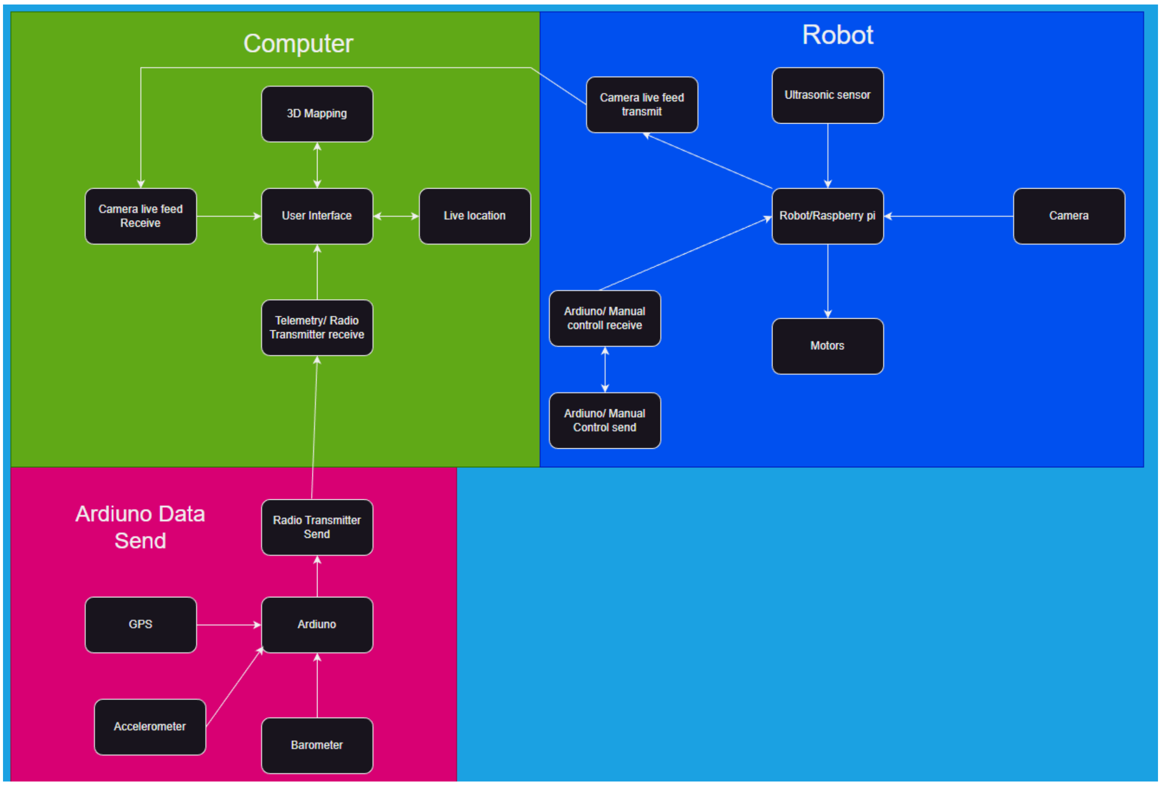

Figure 38 Software architecture

3.8 IMPLEMENTATION OF THE SOFTWARE ARCHITECTURE:

In this section we describe the implementation of the system architecture's infrastructure. The robot's task is to drive autonomously, follow its user and mapping. Along the way, it is required to avoid obstacles.

USER FOLLOWING

The method discussed that we would use if we needed to make a model is a common practice among people trying to achieve the same capability. It is a general way to do this. There are alternative frameworks that can be used that were mentioned, but it all boils down to the same practice to get this object detection functionality.

OBSTACLE AVOIDANCE

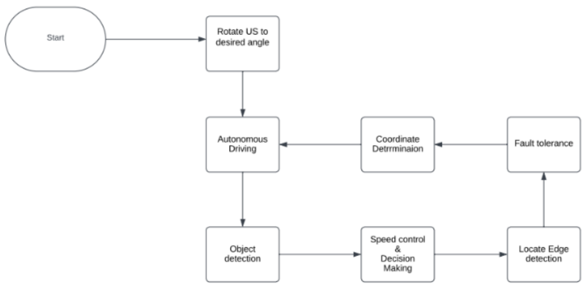

Figure 39 Obstacle Avoidance Architecture

For object recognition and collision avoidance in our robot, there are several possible strategies and methodologies available. Based on case study methodology, we will use an ultrasonic (US)-based collision avoidance. The suggested collision avoidance algorithm operates on a rather straightforward basis. The algorithm oversees managing the horizontal speed to prevent collisions in the XY plane.

- Object detection: in charge of warning of nearby obstacles.

- Speed Control and Decision: in charge of lowering the payload's speed as it approaches obstacles and stopping it when it is too close to the obstacle.

- Locate and edge detection: in charge of scanning the item in front of it and identifying its edges, - Fault tolerance: eliminating any erroneous readings,

- Triangulation: the process of determining an obstacle’s precise angles and distances

- Coordinates determination and choice look for detours around the item and reroute the robot as necessary (JN, 2017).

3D MAPPING

Our design will implement DEM mapping as the primary 3d mapping algorithm. This approach revolves around the storage of height measurements, within a grid like pattern. It should be noted that the implementation of DEM mapping algorithms into our project will not include any path finding capabilities. We will focus on the development of the Map, and storing the data collected for future use. Further improvements will include path finding abilities based on the data recorded within each DEM Map.

3.8 INTEGRATION OF DESIGN DOCUMENTS

In the overall system design, the mechanical designs, electrical designs, and software designs of the robot must interconnect and manually influence each other.

1. Mechanical design: The physical framework, elements, and mechanisms of the robot are all included in the mechanical designs. They decide how the robot moves, stays stable, and engages with its surroundings. The architecture and location of electrical components, such as the wiring, connections, and circuit boards, are influenced by the mechanical designs. The mechanical structure determines where to put sensors, motors, and other actuators to maximize their use and accessibility.

2. Electrical Schematics: The electrical schematics are the designs and layouts of electrical circuits and power systems within the robot including the appropriate component selection, power distribution, sensors, actuators, and control modules integration. These electrical schematics are influenced by the mechanical designs in the aspect of accommodation of the physical constraints and requirements of the robot's structure. For example, the mechanical layout determines the wiring paths and cable lengths to ensure efficient routing and to avoid interference.

3. Software architecture: The software architecture, as defined in the previous section, is the design of the robot's software systems. This includes vision, planning, control, and graphical user interface. This architecture is influenced by electrical schematics. The capabilities and limitations of sensors and actuators integrated into the robot's mechanical structure determines the software algorithms developed. The software architecture also considers the computational resources, memory constraints, and real-time requirements of the systems in the robot.

3.9 CONCLUSION

In this paper, the design of a payload robot system was presented, with the focus of the paper being based on the autonomous, manual, and user following features. These features have undergone several iterations to improve their functionality on the robot. This paper also considered the robot’s vision, planning, and control modules that affected its effectiveness. Structural analysis was carried out to ensure the user friendliness of the modules in the proposed design.

CHAPTER 4 (Prototype Development & Testing):

4.1 INTRODUCTION

In our previous chapter we looked at the design of our payload robot. In this chapter we go through the implementation of this design. We will be highlighting the challenges faced in the prototype creation, the development of the payload and the test processes implemented in the building process of the robot as well as the payload's autonomous features.

4.2 SYSTEM PROTOTYPE DEVELOPMENT

4.2.1 Mechanical Development:

Figure 40 Welding of Frame

Figure 41 Complete Frame

The robot’s chassis was built to achieve 2 objectives. The first objective is to travel on various terrains and, the second, is to be able to carry loads of 5 kg and more. To accomplish this the robot’s frame had to be able to support not only the load but also the robot itself. The chassis was made with steel, which was accomplished this by assembling and TIG welding the metal pieces of the robot to make up the frame.

Figure 42 Removing Tabs from CNC process.

Figure 43 Checking Fitment of Tubing

To complete the body a CNC router was used to manufacture the payload’s wooden pieces, this is a machine that carves out complex shape. Robot body was assembled to fit the prototype frame, these wooden pieces (below the loading bay) attach to each other as a puzzle and can be disassembled when necessary. The loading bay was made using the same method as the body however the loading bay was bolted and glued together.

Figure 44 Wheels Marching on Lathe

The wheels of the robot needed a surface that allowed the rubber track to rotate around. To customise the wheels to fit these requirements the skateboard wheels, a lathe machine was used to shave down the middle of the wheels to the width of the rubber track. The purpose of this was to ensure that the rubber track remains in the ridges created from shaving down the wheels, however when tested the rubber track would eventually to climb out the ridges and off the wheels after a certain period. To rectify this problem disks were bolted onto the sides of the top wheels; this blocked the track from climbing off the wheels and allowed the track to continue navigating terrain without any track issues.

Figure 45 Motor Mount Machining on CNC Mill

The suspension of the robot includes 2 elements the springs and the point of pivot. The robot's suspension arms were manufactured using a CNC milling machine, aluminium parts were cut by the means of a machining process like drilling and cutting the robot suspension with absolute precision. To complete the suspension, heavy duty springs were attached from the frame to the eccentrics on the suspension arms. In the first attempt at the suspension implementation the springs used were not strong enough to efficiently work for a payload, to improve this the original springs were replaced with trampoline springs.

4.2.2 Electrical development:

This section will cover the Electrical Development of the robot. It will cover the development of the Sensor board, all the power distribution and wiring. Finally, it will how the motor controllers was developed.

Sensor Board

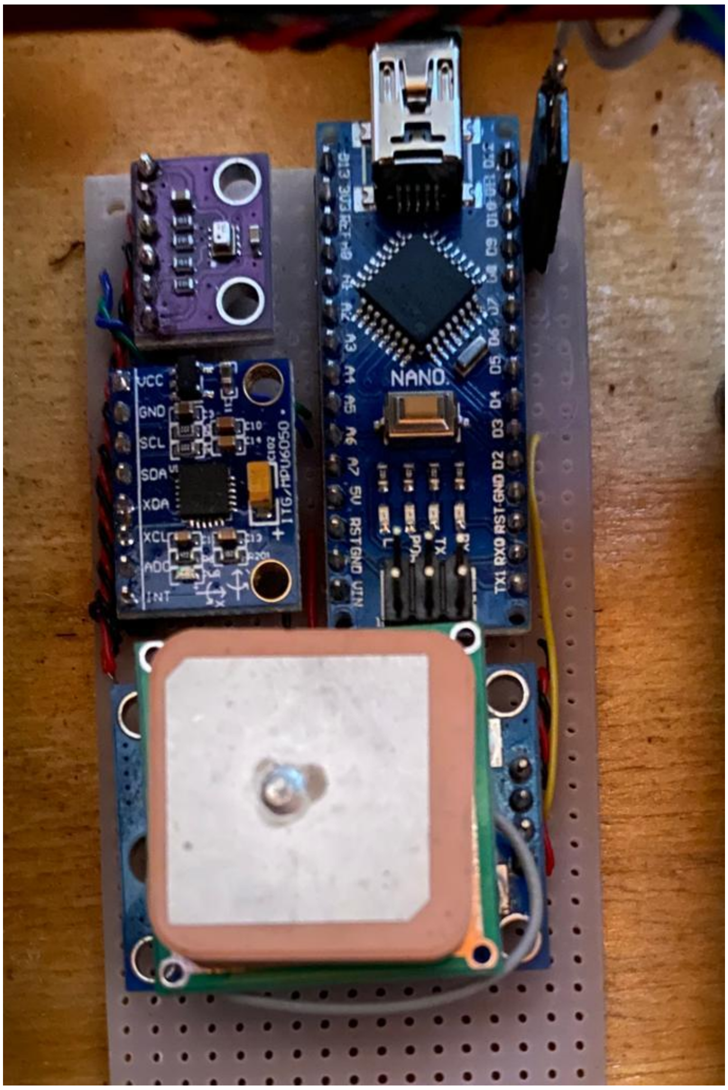

Figure 46 Sensor Board

Looking at Figure 19 we see the developed Sensor Board, this circuit was developed around prototyping PCB, this means that all the wiring was done and soldered by hand to achieve the necessary connections. The board is designed around 3 sensors, these include GPS (Neo 6m), Accelerometer (MPU 6050) and then finally a barometer (BME280).

The board was developed around the Arduino nano (Microcontroller) that handles data processing, and communication. The sensor data is received, via all the channels and busses, in the wiring and then converted into readable data. From there the data is sent to the Telemetry module via on cable, in the form of serial communication. It should be noted that this board is sufficiently powered by just USB power alone and does not require any other forms of power input.

<u>Motor controllers</u>

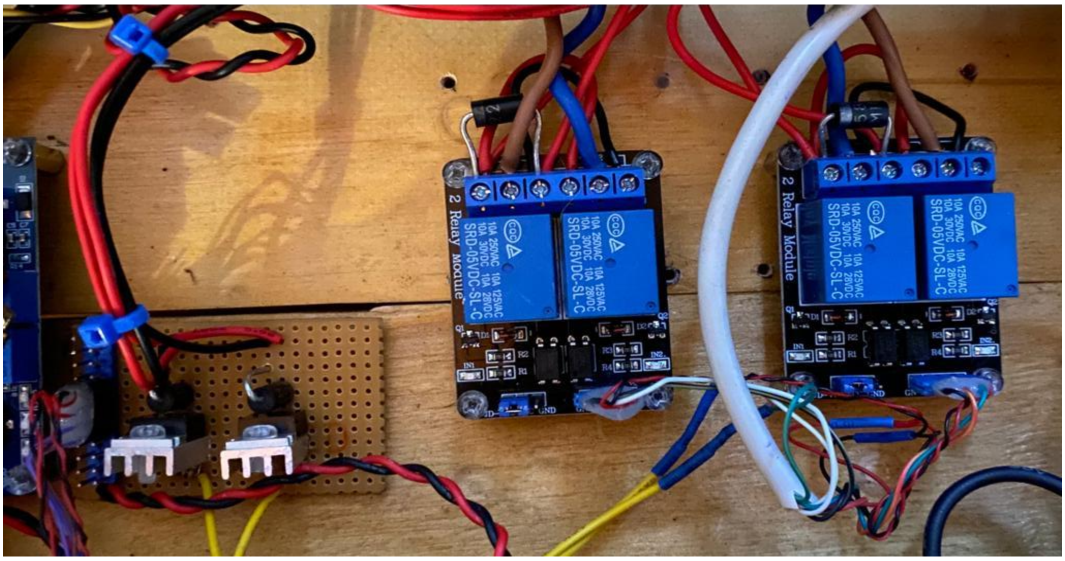

Figure 47 Motor Controllers

Looking at Figure 21 we see the final development of the motor controllers. This circuit revolves around the forwards and reverse directional control that one of the motors require of the robot. This is achieved by utilizing 2 relays, that are developed into a module to perform forwards and reverse (bidirectional) control. The result of this is that bidirectional control is achieved by means of 2 signal wires. See Below Table:

Wire 1	Wire 2	Signals WIRES
HIGH	LOW	FORWARDS
LOW	HIGH	REVERSE

Table 7 Bidirectional Control Results

The speed control is achieved by the Pulse modulation of the power through the ground terminal of the relay board. A mosfet (power transistor) is used to handle the high current draw of a motor. This adds another wire to the wiring of a motor, resulting in 3 signal wires to fully control a single motor. This configuration of dual relay boards and mosfet to achieve motor control is then duplicated to achieve dule motor control on the left and right side of the robot. Resulting in the utilisation of 6 signal wires to achieve full control of the final robot.

Power Distribution:

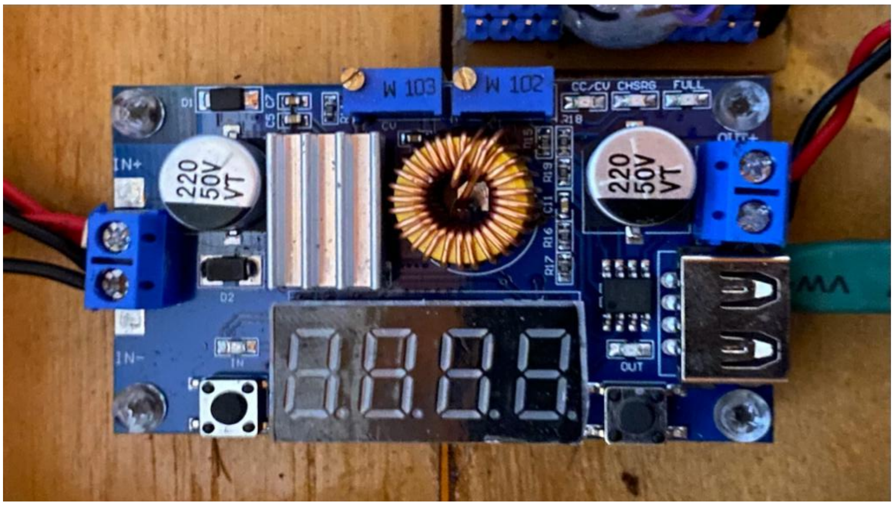

Figure 48 Power Distribution

Power Distribution of the robot is handled by XL4015 5A Step Down Module with Digital Voltmeter. This Module is used to deliver a current limited, 5v power source to all the necessary electronics. It should be noted that all the electronics utilising this 5v supply is connected in parallel to ensure effective distribution across all the electronics. This module was chosen because of its current limiting functionalities, ensuring that none of the electronics will be burned out in the case of a mishap or short.

Overall Wiring:

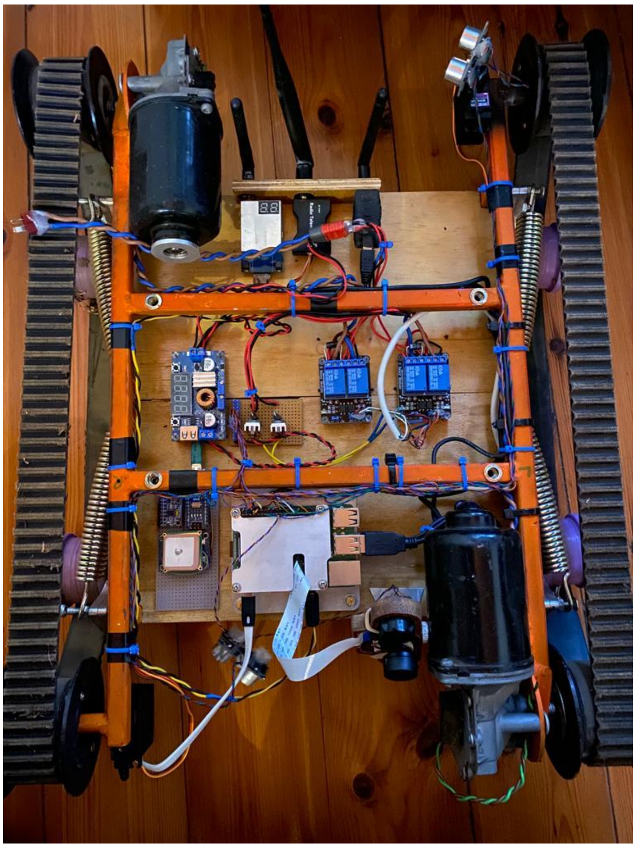

Figure 49 Overall Wiring

Overall Wiring was achieved by utilising a relatively small gauge wires for the signal's wires and connections between the less power-hungry sensor modules, or servo motor. Whilst tick gauge wires were utilized in the motors and battery power cables, this is done so that the wires do not melt due to high current pull. The overall wire management was achieved by utilizing the frame and routing the wires against the frame according to its final position on the robot. Cable Ties were used to secure the wiring to the robot's frame, achieving a neat result, that directs wires out of high-risk zones.

4.2.3 Software development:

Object following

The process of object following was achieved by utilizing the following steps. Firstly, OpenCV and a Pi camera were employed to capture the camera view and set the desired display size for the window. Next, the colour space of the image was converted from BGR to HSV values. By defining a specific range of HSV values, the camera was restricted to detecting only those values within the programmed range.

Once an object within the desired HSV range was detected, the middle pixel was designated as the point of interest. OpenCV libraries and functions were then utilized to draw contours around the object. These contours were further used to create a bounding rectangle that enclosed the outlined object. The tracking of the object within the frame was made possible by extracting its spatial information, including x and y coordinates, as well as the width and height of the bounding box.

To align the object with the centre of the frame, a calculation was performed using the formula: error = (x + w/2) - dispW/2. This calculation allowed for the determination of the error in relation to the centre pixel. Based on the width and height values of the bounding box, which varied depending on the distance of the object from the camera, the error value was adjusted.

To control the robot's movement, an area calculation was performed using the width and height of the bounding box. The resulting area variable determined the appropriate action for the robot, such as moving forward, stopping, or reversing. This method facilitated the successful implementation of object following, enabling the robot to track and respond to a designated object in the frame.

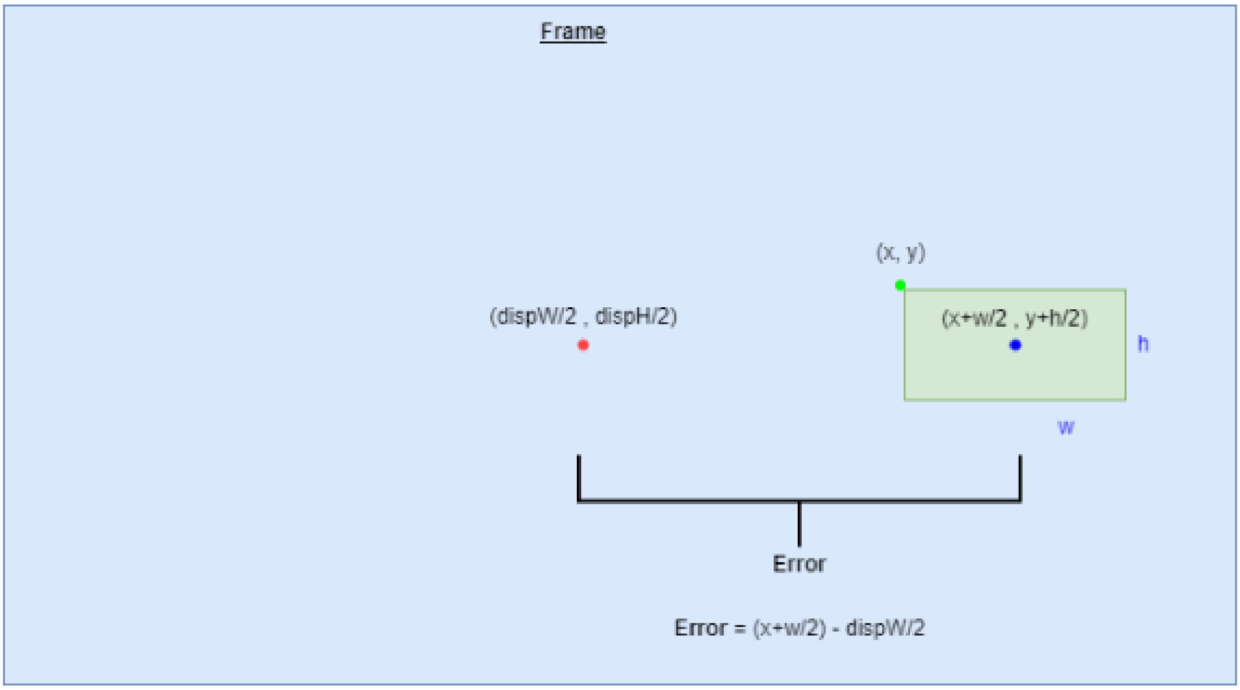

Figure 50 Error Calculation OBJ Following

3d Mapping

3D Mapping is where a digital representation of the terrain is, generated based on sensory data that was recorded. This section will cover the software development to achieve 3D mapping. First we need to understand how the wiring is done to envision the dataflow.

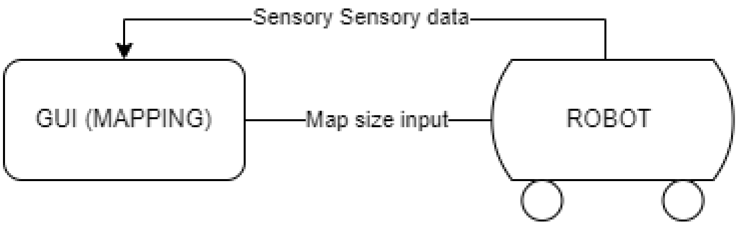

Figure 51 Base Level Systems Communication

Looking at the diagram above we see that there are 2 sections. 1st The GUI section that is used to initiate the mapping sequence by transmitting the necessary data to the robot so that it knows what the size of the gird should be that the user wants to map. 2nd

The robot, which receives data from the sensors, and then relays it back towards the GUI. It should be noted that the Robot does not run any data processing algorithms like (Slam Mapping), as the SBC is heavy under powered for this task. The mapping sequence is achieved, by recording height measurements, based on the previously driven terrain (DEM Mapping).

The algorithm works as follows.

There are 2 programs that are simultaneously initiated. The first one is an algorithm on the robot, that receives the map size via Telemetry from the GUI. From there the robot generates an equal sided matrix depending on the size (NxN) given by the user on the GUI side (See Figure 52).

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Figure 52 Matrix Grid

ROBOT SOFTWARE:

Motor control:

This matrix grid pattern, driven by the robot based on an algorithm that utilizes 2 for loops with one another. The Fist loop is initiated, with a repeating cycle of N (given by the GUI/user), this takes care of all the rows within the grid. From there the Second Loop is initiated each time the First loop is initiated. With the same number of repeating cycles as the previous loop. The second loop takes care of all the columns within the grid. When the second loop is initiated, both motors are driven 1 unit.

The cycle is then repeated the number of times NxN gives you. So, if the user Inputs a grid size of 5 the motors will be initiated roughly 25 times. It should be noted that there is a left hand and right-hand hairpin sequence that is initiated after the motor commands are run. This sequence is initiated, before every repeating cycle of the first For loop. The initiation of the sequence selects every odd number of times a lefthand sequence and then a right-hand sequence every equal number of times.

Data Output:

As the Motor control algorithm is initiated, data is continuously sent via a Telemetry channel towards the GUI, with minimal delay, and no processing required.

GUI MAPPING SOFTWARE:

Data Output:

When the mapping function is initiated by the user, the GUI takes the value given by the user and sends a program initiation, bit with a value that the Matrix is based on.

Data recording:

This section is based on the same algorithm used within the Motor control section of the Robot. However instead of the motor control functions that are initiated within the second for loop, a data recording algorithm is run. This data recording algorithm takes a height recording with a delay based on the duration the robot takes to drive a single Unit or cell within the matrix. This records a height measurement, within each cell in the matrix, giving a complete NxN matrix with data in it.

Data smoothing:

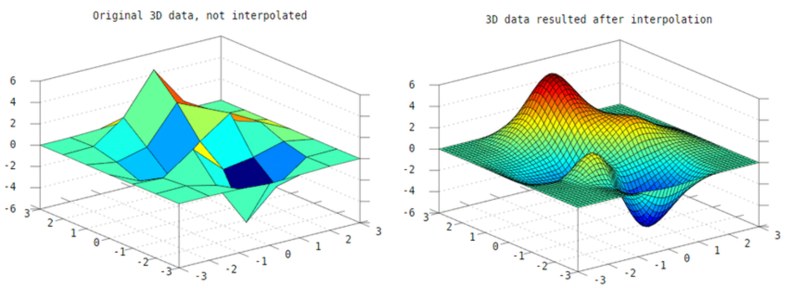

Figure 53 Interpolation Algorithm (Interpolation, n.d.)

After a Matrix with data is recorded, the data within the matrix will be very rough and jagged. This is due to the resolution of data measurement within the XY plane. The data can be given a higher resolution and smoothed by running an interpolation algorithm on the matrix and the data within it. An example of which can be seen in figure 24, showing a Data Plot before interpolation (on the Left) and then a Data Plot after interpolation (on the Right)

Map Generation

Finally, after the improved Matrix is received after Interpolation. The data is displayed by means of a graphing plugin within Python, called (MATPLOTLIB), this program stretches a surface over the Matrix resulting in a 3D Terrain representation, of the environment that the robot had just driven over.

Obstacle avoidance

The ultrasonic sensors, mounted on servos, play a vital role in the system. These servos enable the rotation of the ultrasonic sensors to the three programmed positions: right, middle, and left.

Within the programmed range of 50 cm (or 500 mm) perpendicular to the robot, the ultrasonic sensors continuously monitor the surroundings. When an object is detected within this range, the corresponding value in the output array is set to "1". Conversely, if no object is detected, the value is recorded as "0".

These output values provide valuable insights into the proximity of objects rather than a simple binary representation. A value of "1" signifies the presence of an object within the programmed range, while "0" indicates the absence of an object.

Based on these output values, the system initiates appropriate motor control sequences. These motor control actions are tailored according to the position and proximity of the detected objects. This allows the robot to dynamically respond to its environment, enabling it to navigate and avoid obstacles with precision and safety.

Telemetry (Send & Receive)

```
-25.724412,28.236143,130790,10,0.04,1130.36
,0.92,-1.13
-25.724414,28.236145,130790,10,0.17,1130.36
,0.91,-1.14
-25.724414,28.236145,130790,10,0.17,1130.14
,0.89,-1.14
-25.724414,28.236145,130790,10,0.17,1130.19
,0.88,-1.14
-25.724414,28.236145,130790,10,0.17,1130.19
,0.87,-1.14
-25.724416,28.236145,130790,10,0.14,1130.19
,0.86,-1.14
-25.724416,28.236145,130790,10,0.14,1130.24
,0.84,-1.14
```

Figure 54 Data Received Via Telemetry

The telemetry done through a Arduino and a Radio Frequency transmitter.

The Arduino collects data from multiple sensors which includes:

- Accelerometer
- Barometer
- GPS NEO-6M

The data collected from these sensors include:

- Longitude
- Latitude
- Altitude
- Satellites
- Speed
- Pitch
- Roll

The data is processed and then stored in a list in a specific order. The data is then encoded and using the write line method sent to the RF module. From the RF module the data is sent to the receiving RF module which is plugged in to the base computer. The user's computer then reads this data through the COM port where the RF module is plugged into. The data is then read by using the read line method.

To avoid the misinterpretation of data we implemented an algorithm that takes the length of the list and when the list is the incorrect length, we would use the previous sent data that is correct length. The data is then split from the list into the respective pieces of data. From there the data is then displayed in the GUI and used for 3D mapping.

The telemetry can take a while to take place, the Arduino that handles this section waits for GPS signal to be used as input, only when the GPS signal is picked up the transmission may begin.

Graphical User Interface (GUI)

The Graphical User Interface is coded in Python using the Tkinter library. Serving as the central control centre for the Robot, the GUI plays a crucial role in the operation of the Robot. A key feature of the GUI is to represent real time data transmitted by the Robot, enabling users to monitor its status and performance. Additionally, the GUI facilitates the initiation of various protocols, such as the 3D mapping feature.

For the 3D Mapping section, the GUI takes in a value in an input box, from there on the input is then sent via telemetry to the robot to use as input. Paired with this when the 3D mapping is initiated the GUI will then display the Area that has been mapped providing users with a visual representation of the Robot's exploration.

When the Location button has been pressed a Google Earth map will open with a pin on the Location of the Robot, this will only happen if the Robot is sending location data to the user's computer. If a location is not received there will be an Error message Output. The GUI supports a live video feed from the Robot's camera, which enables users to remotely view the surroundings and control the Robot from a distant location. By providing this real-time visual feedback, users can make informed decisions and interact with the robot effectively.

To ensure a smooth and responsive user interface, the Robot Control centre updates the labels displaying received data every 100 milliseconds, which minimizes any perceptible delay or lag between the Robot's operations and the displayed information. This frequent update rate ensures that users have access to the most current and accurate data in a timely manner.

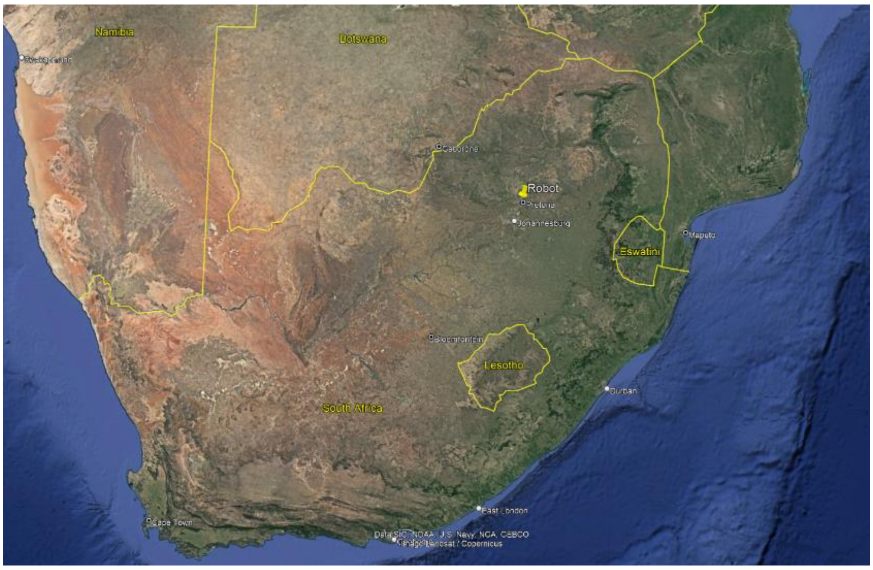

Figure 55 Screenshot of Robot Location

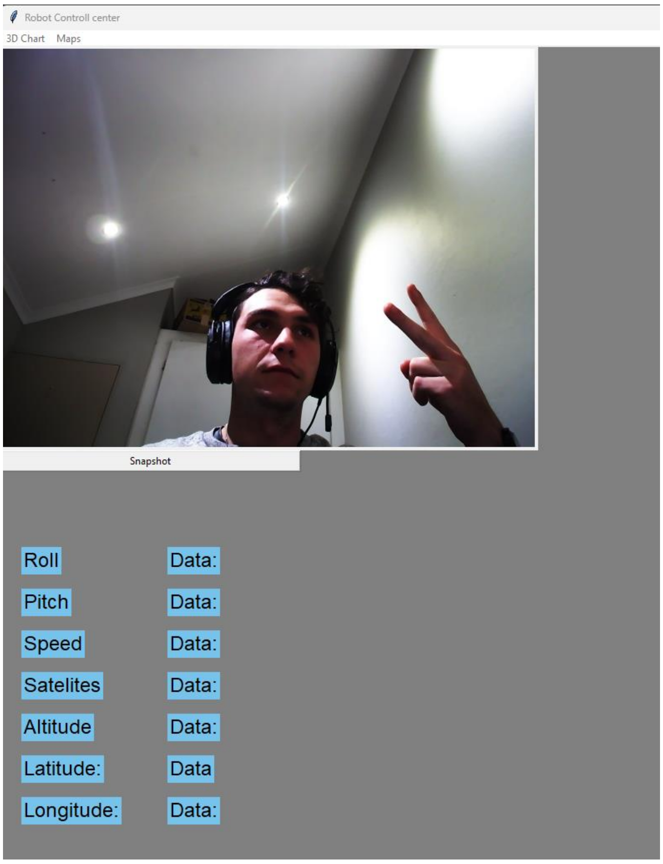

Figure 56 Screenshot of GUI

4.3 TESTING

4.3.1 Tests to be done:

To ensure the robot can function effectively. We will conduct tests to evaluate the performance of the robot in various conditions these tests include.

Battery, working duration (with battery monitoring {each minute reading of voltage for graph}) With load.

To test the battery, we would like to measure the following parameters:

- Duration of continuous operation
- Voltage drops.

To prepare for the battery test, we would need to ensure that the robot is in optimal working condition meaning that the battery should be fully charged and that all sensors are working. Once the preparations are done the tests can be conducted.

To measure the power consumption and duration of continuous operation of the robot we are going to use a stopwatch and take measurements the volts of the battery every 60 seconds until the battery is discharged. The battery produces 12.6 Volts at full charge and 11.4 volts at low charge.

To simulate a working environment this test will consist of various other tests:

- Speed test with payload
- Obstacle Avoidance
- Object Following
- 3D Mapping

To calculate the life span of the battery we use the following calculations:

Effective battery capacity = Battery capacity × (Remaining voltage range÷ Rated voltage range)

$$3.3Ah \times (12.6V - 11.4V) \div (12.6V - 11.4V)$$

$$= 3.3Ah$$

Next, we must determine the discharge current which is 12A. The next calculation is to calculate the discharge time.

Discharge time = Effective battery capacity ÷ Discharge current)

$$3.3Ah \div 12A = 0.275$$

The battery will discharge in 0.275 hours which is equal to approximately 16.5 minutes.

Our hypothesis is that the battery will have a life span approximately 15 minutes from full charge to discharge under a full working load.

Speed test (With load)

In this test we aim to determine the maximum speed at which the robot can travel while carrying a 7.5 kilograms payload. To conduct this test, we will employ a stopwatch and a tape measure. The robot will commence its journey at point A and travel 2 meters to reach point B. Out objective it to measure the time taken by the robot to travel from A to B.

The test will be repeated twice under different battery conditions. The first test will be conducted when the battery is fully charged, while the second test will be performed with the battery on low charge. In both tests we will carefully measure the time it takes to complete the distance.

Throughout the experiment we expect to record a faster speed when operating on a full battery compared to a low battery. The recorded times from both tests will be used to confirm this hypothesis and determine the impact of battery voltage on the robot's speed.

To calculate the speed of the robot we will then use the recorded times to get the metres per second the robot travels.

Calculations:

$$\text{Speed} = \text{distance} \div \text{time}$$

Weight Capacity test

This test will involve the robot's weight capacity of the robot while stationary versus when the robot is in motion.

1. Determination of stationary weight capacity: This test will be performed by loading the maximum weight the robot can safely carry when it is stationary. This will assist in defining the baseline for the payload capacity test.

2. Payload preparation for mobility: Choose an appropriate payload ranging in predefined the specification of 5kg. Ensure the payload is securely placed in the loading bay.

3. Assess the payload weight capacity in motion: Next, analyse the mobility of the robot carrying loads. Start with a lower weight (under 5Kg) and gradually increase it until the robot reaches its operational limit or indicates any signs of instability or significant reduced performance.

4. Record the weight capacity: Once the payload weight is at a point in which the robot indicates signs of strain or instability is determined, record this value as the payload weight capacity in motion.

5. Repeat test: conduct the previous stems multiple times to ensure consistent results.

6. Document the findings: Document the results, with the maximum weight tolerance the robot can safely carry both stationary and in motion. This information will ensure that the robot is used within its safe weight limits without damage to the robot.

Ultrasonic Test

To evaluate the performance of the ultrasonic sensors, we will conduct tests to determine the maximum accurate detection range and assess their field of view.

The first aspect we will examine is the accurate detection range of the ultrasonic sensors. By systematically testing different distances, we will identify the point at which the sensors can reliably detect objects with precision. This testing process will allow us to establish the effective range at which the sensors provide accurate distance measurements.

Additionally, we will assess the field of view of the ultrasonic sensors. This involves testing the sensors' ability to detect objects within their viewing angle. By placing objects at different positions and orientations relative to the sensors, we can determine the extent of their field of view and identify any potential blind spots or limitations.

By performing these comprehensive tests, we can gain a thorough understanding of the capabilities and limitations of the ultrasonic sensors. This knowledge will enable us to optimize their usage and ensure accurate object detection and avoidance in various scenarios.

Terrain limitation test

This test measures the highest surface the payload can navigate over.

to perform this test the robot will be placed on multiple surfaces with and without the payload to determine the robot's ability to navigate over different levels/heights. Starting from the lowest height the robot's ease in motion will be analysed, steadily increasing the surface height until the robot is unable to climb the terrain or the load in the loading bay cannot be carried safely.

3D map test

This section will cover the theoretical tests to be done on the 3D Mapping section. The 2 tests to be done:

- Still Standing Level ground test
- Actual Mapping of terrain

Still Standing:

The still standing test revolves around the robot standing stationary at a single location (with deactivated motors). This test is done to check the accuracy of the barometer data. What this means is that when the robot is stationary and the program is initiated, the 3D Graph that it produces, should be a flat surface parallel to the XY plane. If this result is produced within a range of 250mm the sensor data is optimal for the usage in terrain mapping. If not, then more calculations and improvements are needed to get a more precise result.

Actual Mapping of terrain:

It should be noted that this test will only happen after the Still standing test are successfully completed. If this is not done, then the mapping of the actual terrain will be inaccurate.

The Actual Mapping of the terrain is based on the environment that surrounds the robot. The expected test results should produce a 3D chart of the driven terrain. It should be noted that, for a successful test to occur, the robot should run parallel lines to the previous row that it has recorded and then all the turns that the Robot takes, should be 90 degrees.

If this is not achieved, then the output of the terrain mapping will not be an accurate representation of the terrain driven.

Object following:

To validate the object following functionality, a comprehensive test will be conducted using various objects of different colours. The primary objective of this test is to ensure that the system accurately detects and focuses on the specific colour we have programmed it to recognize. By systematically testing different colours, we can verify the reliability and specificity of the colour detection algorithm.

In addition to colour testing, the camera's field of view will also be thoroughly examined. Understanding the camera's field of view allows us to determine the width and breadth of the image that the camera can perceive within its frame. This information is crucial for precisely tracking and following objects within the camera's visual range. By conducting these tests and evaluating the camera's performance with different colours and objects, we can gain a comprehensive understanding of the system's capabilities and limitations. This comprehensive approach ensures that the object following feature is robust and effective in a variety of scenarios and environments.

We can make predictions based off the camera's given specs which are provided from the supplier we bought it from.

- Camera specifications
  - CCD size: 1/4inch
  - Aperture (F) : 1.8
  - Focal Length: 3.6mm
  - Angle of View (diagonal) : 75.7 degree
  - Sensor best resolution : 1080p
- 4 screw holes
  - Used for attachment
  - Provides 3.3V power output
  - Supports connecting infrared LED and/or fill flash LED
- Dimension: 31mm × 32mm

Figure 57 Camera Specs

Obstacle Climbing Test

To assess the obstacle climbing capability of the robot, we will conduct a series of tests to determine its height limitations.

The first phase of testing will involve placing a smaller obstacle with a relatively lower height. By carefully observing the robot's ability to navigate and overcome this obstacle, we can establish a baseline for its climbing performance.

Subsequently, we will introduce a higher obstacle to evaluate the robot's capabilities in handling increased height challenges. Through systematic testing, we will determine the maximum height at which the robot can effectively climb over obstacles.

By conducting these tests and documenting the robot's performance with obstacles of varying heights, we can ascertain its obstacle climbing limitations. This information will help us understand the robot's capabilities and enable us to make informed decisions regarding its application in real-world scenarios that involve overcoming obstacles of different heights.

4.3.2 Testing Environment:

To ensure best analysis of the robot's capabilities, the robot undergo testing on various surfaces ranging from grass, gravel, sand, tiles, and bricks. For the software testing of the robot certain environmental criteria will need to be met, these software implementations are mainly obstacle avoidance and user following.

Tests will be conducted in a well-lit environment to ensure optimised camera performance and accurate colour recognition in the implementation of user following. Additionally, to cater to obstacle avoidance the robot will navigate through paths with various obstacles of different lengths and sizes, providing insight on the efficiency or limitations of its obstacle avoidance capabilities.

4.4 Project Cost

PROJECT FINAL COST			
Electronics	**COST**	**QTY**	**TOTAL**
Raspberry pi	810	1	810
Batteries	770	2	1540
Motors	517,5	2	1035
Motor contorllers	308,7	2	617,4
Motor contorllers V2	92,42	2	184,84
RC TX, RX	83,32	2	166,64
Accelerometer	45,36	1	45,36
arduino nano	147	2	294
Ultrasonic Sensor	26,09	2	52,18
Power Distrobution	145,43	1	145,43
Battery Sensor	79,95	1	79,95
Camera	477,25	1	477,25
GPS	134,78	1	134,78
Serial, TX RX	1539	1	1539
Video to USB	200	1	200
battery connectors	108	1	108
Video rx tx	1042,99	1	1042,99
Switch	128	1	128
MISC	739,14	1	739,14
Mechanics			
LongBoard Wheels	200	2	400
Tracks	209	2	418
9mm Ply	669	1	669
13 mm Square Tubing	59,9	1	59,9
Springs	170	1	170
MISC	34	1	34
Equipment			
varnish	159	1	159
hacksaw blades	31,9	3	95,7
spray paint	144	1	144
MISC	300	1	300
		TOTAL	**11090,86**

Figure 58 Project Cost

4.5 Conclusion

This chapter was focused on the hardware and software design of our robot as well as the experience gained through the design project. Developed by five undergraduate students we found the development and design process of the project challenging and motivating. In conclusion, our system prototype development serves as a foundation for further development and gives insights for future iterations. Upon successful completion of this chapter, the project will be closer to its goal of bringing a robust and functional system.

CHAPTER 5 (Results, Conclusions & Recommendations):

5.1 Introduction:

Following the testing section in chapter 4 is the results from the tests and the analysis. This chapter covers the working duration of the battery, the speed test, weight test, climbing test, 3D mapping test, and the ultrasonic testing which includes the field of view, pickup and distance test. This phase is essential to evaluate the performance of the subsystems and validate the design and implementation of the robot.

5.2 Results:

Battery, working duration with load.

With the experiment conducted, the following is the results:

The following table shows how the voltage dropped over time in real working conditions:

Time in seconds (S)	Voltage (V)
0	12,60
60	12,50
120	12,50
180	12,50
240	12,40
300	12,40
360	12,40
420	12,40
480	12,30
540	12,30
600	12,30
660	12,20
720	12,20
780	12,00
840	11,50
900	11,50

Table 8 Voltage Over Time

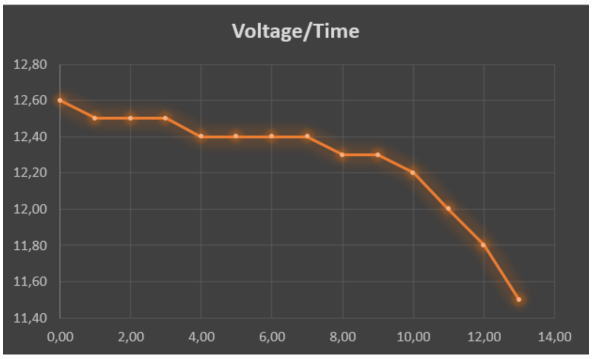

Figure 59 Voltage over time

With real working conditions simulated with all the tests that has been conducted the voltage has dropped over time till the battery was discharged. The voltage was read by using a voltage meter over time.

An interesting observation that we made, is, that when voltage drops, the Ultrasonic sensors became less reliable as they were giving out inconsistent readings.

Furthermore, our hypothesis was wrong as the battery lasted just under 15 minutes. Meaning that under a full workload and going at max speed the Robot could travel for up-to 14 minutes. The experiment was successful as we could calculate the duration of continuous operation, and accurately take readings of how the voltage drops as the robot was operating.

Speed test (With load)

From this test we gathered the robot maximum speed, using a set distance and measuring the time taken to complete the distance. We aim to determine the maximum speed at which the robot can travel while carrying a 7.5 kilograms payload. The robot will commence its journey at point A and travel 2 meters to reach point B.

Figure 60 Speed Test

Flat Battery: 11.4v

Speed =19.9 sec for 2000mm

Approximate speed ≈ 0.1005 m/s

Full Battery: 12.6v

Speed =18.26 sec for 2000mm

Approximate speed ≈ 0.1094 m/s

Average speed between Full and Flat Battery

Speed =19.08 sec for 2000mm

Approximate speed ≈ 0.1048 m/s

0.377 **km/h**

Weight test (Without driving)

Without Load:

Figure 61 Height without load

With Load (15kg):

Figure 62 Load 15kg

Figure 63 Height measurement with Load

Overload/ Stress (65kg):

Figure 64 Overload 65kg

During the weight test or stress test, we conducted rigorous evaluations to assess the robot's carrying capacity and overall performance. Remarkably, the robot exceeded our initial goal of a 5 kg carry load by a substantial margin.

To thoroughly challenge the robot's capabilities, we subjected it to a load of 7.5 kg during the weight test. Surprisingly, this additional weight had minimal impact on the robot's suspension or its overall performance. The robot demonstrated remarkable stability and maintained its functionality even under this increased load.

Furthermore, we conducted a stress test by placing a team member's full weight, measuring 65 kg, on the robot. The suspension system impressively withstood this substantial load without experiencing any failure or compromising its functionality. The robot demonstrated its robust design and ability to handle significant stress without compromising its performance.

These findings indicate that the robot possesses exceptional carrying capacity and structural integrity, surpassing our initial expectations. Its suspension system and overall design enable it to handle substantial weights and stresses effectively, ensuring reliable performance even under demanding conditions.

Ultrasonic (pickup test, FOV and distance {battery voltage significant role in readings})

In our comprehensive testing, we evaluated the accuracy and range of the ultrasonic sensors. The results indicated that the sensors provide reliable and accurate measurements within a range of 1.5 meters (or 1500 mm). At the optimal position, we were able to obtain measurements up to 1.6 meters (or 1600 mm), while at the middle position, measurements were recorded up to 1.4 meters (or 1400 mm).

However, when testing the left position, we encountered some inaccuracies in the measurements, particularly when objects were detected at distances up to 0.95 meters (or 950 mm). This discrepancy can be attributed to the presence of a motor in the vicinity. The ultrasonic sensor's wide field of view (FOV) captures not only the desired object but also intermittently detects the motor, leading to alternating readings between the two.

Given the wide FOV of the ultrasonic sensor and the potential interference from the nearby motor, we acknowledge that the left position measurements may be less reliable compared to the other positions. It is important to consider this limitation when utilizing the ultrasonic sensors in scenarios where accurate distance measurements are critical.

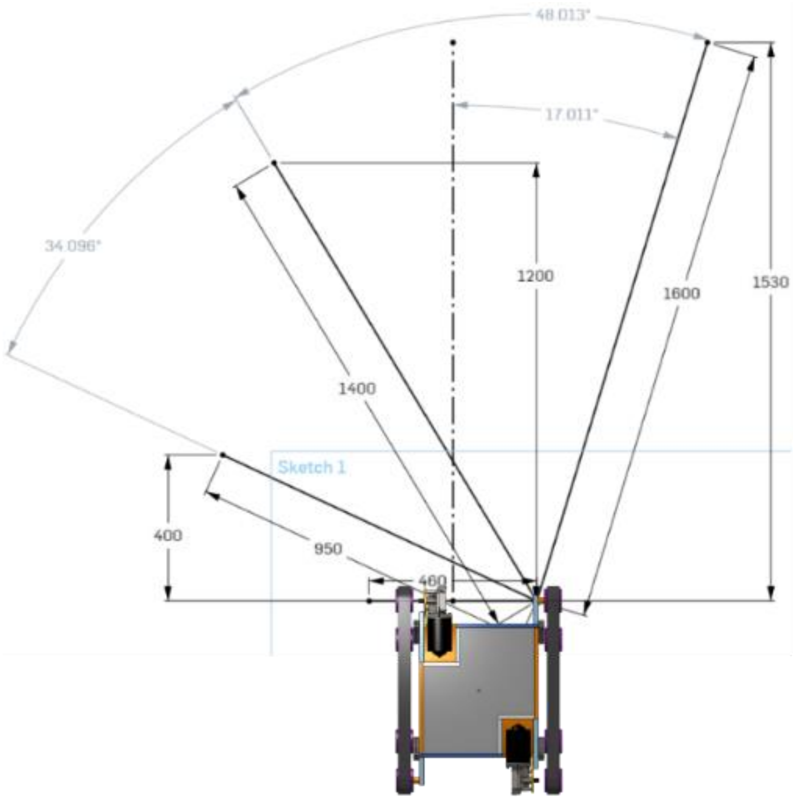

Figure 65 Ultrasonic View angle

Obstacle climbing limitation test (height)

Climbing Test:

Figure 66 Obstacle with height of 50 mm

Figure 67 Obstacle with height of 70 mm

During the conducted tests, we obtained significant findings regarding the robot's obstacle climbing capabilities. It was observed that the robot could smoothly drive over objects with a height of up to 50 mm. However, as the height increased to 70 mm, the robot faced some difficulties in climbing over the obstacle, although it eventually managed to surmount it.

Based on these observations, it can be concluded that the robot's comfortable climbing limit lies around 60 mm. At this height, the robot exhibits reliable and efficient climbing performance without significant challenges. Therefore, for optimal functionality and consistent performance, it is advisable to consider obstacles with a maximum height of approximately 60 mm for the robot to climb comfortably.3D map test.

Still Standing Level ground test:

The Predicted Results for this test, is a flat Plane parallel to the XY plane.

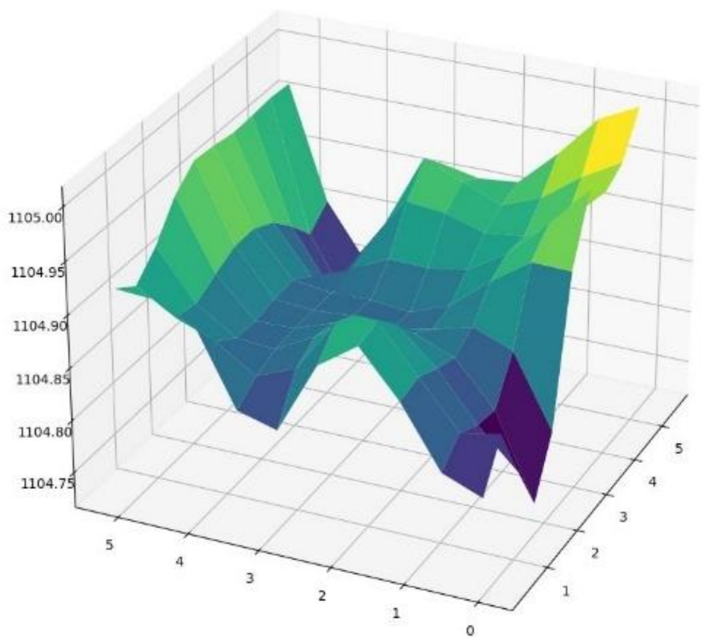

Figure 68 Test 1 Still Standing

Looking at the first test results (see fig 29), we see that the data recorded within the matrix is very, spiky and un consistent, this means that the sensor data recorded form the barometer is inconsistent and should be average out to achieve a more consistent result.

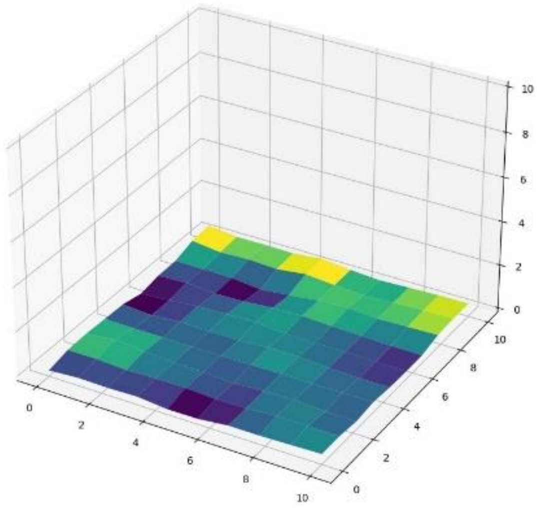

Figure 69 Test 2 Still Standing Top View

<u>Averaging Algorithm:</u>

The averaging algorithm that was applied to the terrain mapping algorithm works as follows. It should be noted that the Sensor data is received at an alarmingly high bit rate, with gives us the ability to apply averaging. The Algorithm works by taking, 10 height readings per second and adding it to one another. From there the result is divided by 10 giving us 1 unit of the cells that was driven (averaged out). This Sequence runs the same number of times for the number of sells within the matrix. The result that is produced is an averaged-out height readings matrix, producing a consistent result.

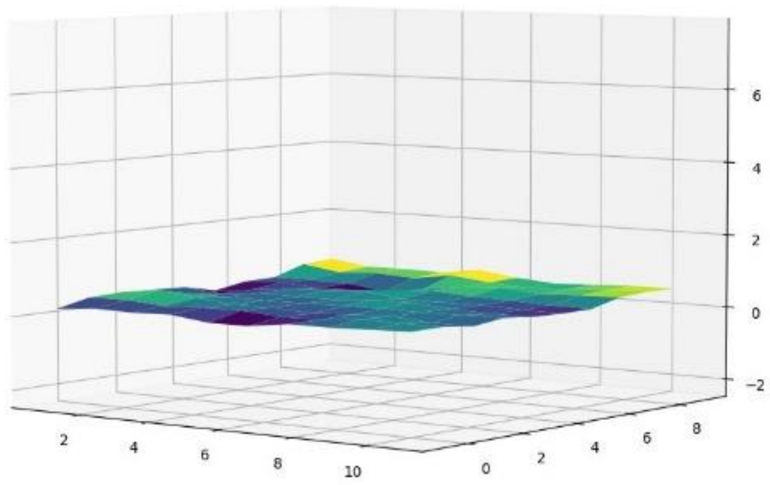

Figure 70 Test 2 Still Standing Side View

After the Averaging Algorithm has been applied to the Mapping of the terrain, the 3D Graph that is produced is like the predicted results (See fig 31). This means that the test was successfully completed, and an actual terrain mapping test can be done.

Actual Mapping of terrain:

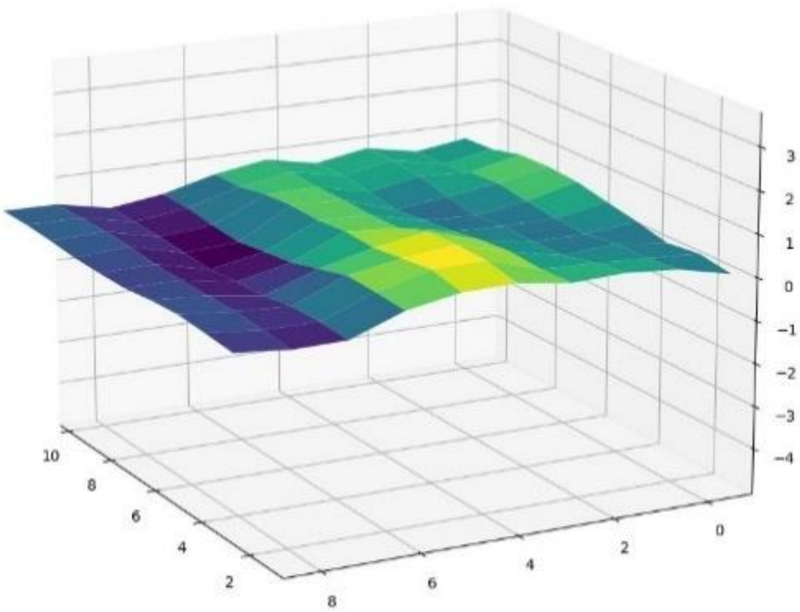

Figure 71 Actual Map of Terrain

Looking at Figure 32 we see a map of the actual terrain, the result produced seems to be quite accurate in terms of the mapping of the terrain. Terrain that was mapped is relatively even in terms of height. However, we see that there is to low sections almost running parallel to one another, with a high spot in the middle section reducing in high. The representation of hight can be accurately visualized by means of the colour change depending on the height. The High spots are displayed as Yellow, whilst the Low spots are represented by the colour Blue.

Object following:

The conducted tests provided conclusive evidence that the camera successfully detects only the specified object based on its colour. By utilizing the programmed HSV value range, the camera effectively filters out any other colours or objects that fall outside of this defined range. This ensures the system's accuracy in selectively identifying and tracking the desired object while disregarding irrelevant elements.

Furthermore, the tests included an evaluation of the camera's field of view. The results indicated that the field of view measured approximately 74 degrees. This measurement aligns closely with the supplier's specifications, which state a field of view of 75.7 degrees. The camera's observed field of view confirms its adherence to the expected performance, further validating its suitability for the intended application.

The field of view (FOV) test was conducted by employing a measuring tape placed parallel to the camera's centre axis. To determine the FOV, the object of interest was positioned at 120 cm (or 1200 mm) from the camera. Subsequently, we moved to one side until the object reached the edge of the frame, measuring the distance from the centre line to our position. This process was repeated on the opposite side.

Using specialized software and calculations based on these measurements, we arrived at a conclusive result: the FOV was determined to be approximately 74 degrees. This finding demonstrates the camera's ability to capture a substantial visual range and confirms its alignment with the specifications provided by the supplier, which stated a FOV of 75.7 degrees. Thus, the camera effectively covers a wide field of view, making it suitable for the intended application.

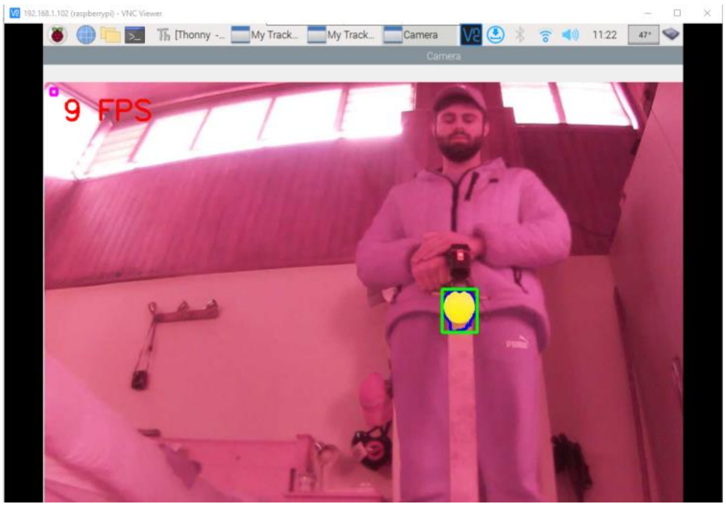

Figure 72 Camera FOV TEST 1

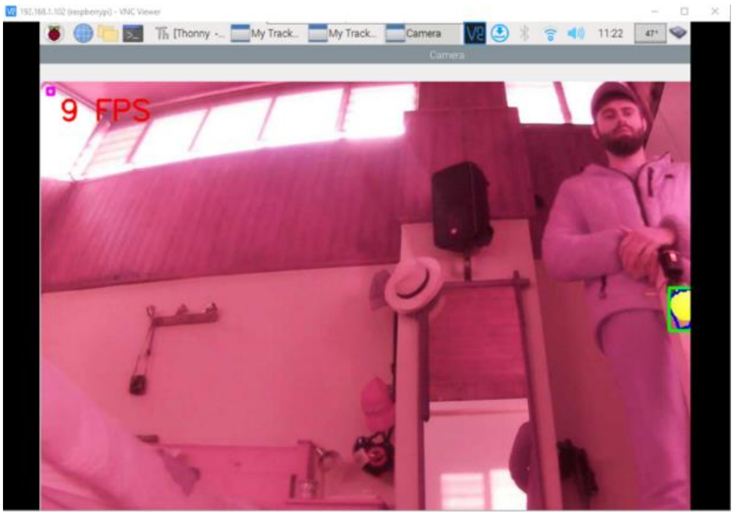

Figure 73 Camera FOV TEST 2

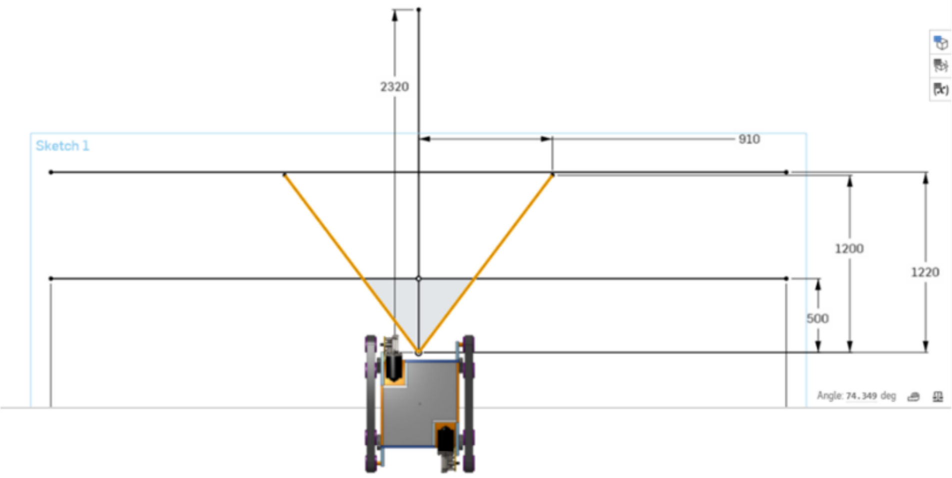

Figure 74 Camera FOV Calculation Diagram

The zones shown above represent the following zones:

The 500 mm range represents the backwards moving range, where if an object comes within that range the robot moves backwards.

The next range between 500mm and 1200mm, represents the dead zone range, where the robot does not move forward or backward, but instead, turns left or right, to align to the object.

The next range 1200 mm and above, represents the forward moving range, so when an object is in that range the robot then moves forward.

5.3 Data Analysis:

3D mapping (angle slow due to battery level)

Based on all the test that where deducted, with the terrain mapping section of the robot. We have noticed that, with the drop in voltage of the battery level, the angles that are turned are not as accurate due to the RPM of the motors reducing with the battery level. This means that the terrain mapping function can only be initiated when the battery is still full, otherwise inaccurate results will be produced.

Ultrasonic Sensors (Battery level affecting readings)

Through extensive testing, we evaluated the range and performance of the ultrasonic sensors in detecting objects. The results of our tests revealed that the sensors are capable of accurately sensing objects at distances of approximately 1.5 meters. This range aligns with our expectations and provides reliable obstacle detection capabilities.

However, an interesting observation was made when the battery voltage dropped below 12 volts. During these instances, we noticed a slight inconsistency in the sensor readings.

We hypothesize that this inconsistency may be attributed to the overall power supply of the system. As the battery voltage decreases, it may not be supplying sufficient power to adequately support all components, including the sensors.

Considering this finding, it is important to note that while the sensors can still function as obstacle avoidance mechanisms when the voltage drops below 12 volts, this discrepancy highlights a limitation in our system.

Object following (Night fall)

From the conducted tests, an important observation emerged regarding the camera's performance in varying lighting conditions. It became apparent that the camera's ability to accurately detect the colour of objects diminished in low-light environments. This dependency on reflection for colour detection posed a challenge, particularly during darker periods.

To address this issue and enhance the camera's reliability in object detection, a practical solution was implemented. A flashlight was strategically attached to the ball, illuminating it and causing it to emit a glow. This innovative addition significantly improved the camera's ability to detect the ball without any difficulties, irrespective of the lighting conditions encountered, whether it be daytime or nighttime.

By incorporating the flashlight and enabling the ball to emit a consistent glow, the reliability and effectiveness of the object-following system were greatly enhanced. This modification ensured that the camera could consistently detect the ball's presence and accurately track its movements regardless of the lighting conditions, thereby overcoming the limitations associated with colour detection in low-light environments.

Barometer (wind gusts)

With the 3D mapping, tests that were completed, we have noticed the barometer sensor readings, spiking in heavy wind conditions, this is due to the sudden increase in pressure generated by the wind gust. A precautionary measure to be taken to this can be to apply a sponge to the sensor chip, distributing the air pressure over time rather than in a sudden burst. There is another problem, that comes with the barometer reading, which is that the temperature, affects the readings significantly. This can be adjusted for by adding another section to the height calculation equation.

5.4 Conclusions and Recommendations

Conclusions

Based on the results and overall analysis we can conclude that the robot can carry out its intended tasks supported the effective features it contains. Considering the navigation system, obstacle avoidance, terrain mapping, and autonomous movement, the robot is more than capable of sustaining most of what it will face. The only concerns being larger obstacles in mining industries that the robot cannot climb over, inaccuracies in measurements which will affect obstacle avoidance, and limited battery life in the case of longer operations. Future work will improve and ultimately eliminate these concerns.

Recommendations

Better SBC

With the implementation and testing of the robot, we have notices significant performance issues with the raspberry pi, causing restriction in the object following section and, in the sensor, reading section. The way to solve this problem is to rather implement an SBC with much higher performance. SBC's that can be implemented are the JETSON NANO, which features deep learning algorithms and improved AI specifically generated for the board. Ensuring the best performance.

Better camera

After careful consideration and analysis, it is recommended to explore alternative camera options for improved performance in the object following system. The camera utilized in our setup was an infrared version 2 picamera, which unfortunately introduced some challenges with the accuracy of the HSV values used for object detection.

To mitigate these issues, we opted for a bright yellow object as it presented fewer colour interference problems compared to other colours. However, to enhance the overall performance and reliability of the system, it is advisable to switch to a regular camera that does not employ infrared technology. This change would allow for more precise and consistent colour detection, resulting in improved object tracking capabilities.

Furthermore, upgrading to a newer camera model, such as a version 3 camera, could provide additional benefits. Newer cameras often boast higher megapixel counts, which can enhance image quality and detail. By leveraging a camera with improved megapixel capabilities, the system can capture finer details of the objects being tracked, facilitating more accurate and robust object following.

In summary, to optimize the object following functionality, it is recommended to transition to a non-infrared camera and consider upgrading to a newer camera model with superior megapixel performance. These adjustments will help mitigate the challenges encountered with the infrared camera and enhance the overall effectiveness and reliability of the object following system.

Feedback system for motors

The robot also showed significant flaws, within the autonomous features. This problem is that when the robot is given an angle to turn or a specific distance to be travelled. The robot does not always and repeatedly produce the wanted results. An improvement to this will be to make a closed loop system, that gets feedback from either a haul effect sensor or a tachometer. This will drastically improve the repeatability of the robot.

Brushless motors (increased speed)

The Robot that was produce, showed significant signs of under rated motors. This can be seen with the speed test that where done. A better solution to this would be to make use of brushless motors. These types of motors are very powerful in correlation to their size and are easily controllable with ESC's readily available on the market.

Standard Tank Tracks.

The design we used was to accommodate for the tracks that we had on hand. We noticed that the tracks could be too hard to use for every type of terrain. The tracks often lose traction due to slim width of the tracks when compared to the robot. Our tracks also are put under a lot of tension due to our strong suspension; this makes it difficult for the tracks to conform to the terrain that is the main issue that causes loss of traction.

Appendix

A) Sensor Board Code:

```
#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include <BME280I2C.h>

#include <Wire.h>



SoftwareSerial ser(10, 11);

String Data[5] = "";

String DATA = "";

String Final_Data = "";

//GPS

TinyGPSPlus gps;

SoftwareSerial gpsSerial(4, 5);

String dataString="";

//BAROMETER

BME280I2C bme;

const float sea = 101325;

const float grav = 9.8;

float height = 0.0;

//ACCELLEROMETER
```

```
int gyro_x, gyro_y, gyro_z;

long acc_x, acc_y, acc_z, acc_total_vector;

int temperature;

long gyro_x_cal, gyro_y_cal, gyro_z_cal;

long loop_timer;

int lcd_loop_counter;

float angle_pitch, angle_roll;

int angle_pitch_buffer, angle_roll_buffer;

boolean set_gyro_angles;

float angle_roll_acc, angle_pitch_acc;

float angle_pitch_output, angle_roll_output;



void setup(){



  Serial.begin(57600);

  gpsSerial.begin(9600);

  BME_SETUP();

  mpu_setup();

  Serial.print("Setup done");



}



void loop(){
```

```
//Serial.println("loop");

Data[0] = displayInfo();

Data[1] = BME280Data();

Data[2] = ",";

Data[3] = roll();

Data[4] = ",";

Data[5] = pitch();



for(int i=0; i<= 5; i++){

  if(i == 0){

    Final_Data = "";

    Final_Data += Data[0];

  }

  if(i == 1){

    Final_Data += Data[1];

  }

  if(i == 2){

    Final_Data += Data[2];

  }

  if(i == 3){

    Final_Data += Data[3];

  }

  if(i == 4){
```

```
      Final_Data += Data[4];

    }

    if(i == 5){

      Final_Data += Data[5];

    }

  }



  Serial.println(Final_Data);

  delay(10);



}



String displayInfo(){

  while (gpsSerial.available() > 0)

    if (gps.encode(gpsSerial.read()))

  if (gps.location.isValid()){

    long scale=10000000UL;

                                    long                    lat                          =
gps.location.rawLat().deg*scale+gps.location.rawLat().billionths/100UL;

    if(gps.location.rawLat().negative) lat=-lat;

                                    long                    lon                          =
gps.location.rawLng().deg*scale+gps.location.rawLng().billionths/100UL;

    if(gps.location.rawLng().negative) lon=-lon;

    dataString = "";
```

```
    dataString += lon;

    dataString += (",");

    dataString += lat;

    dataString += (",");

    dataString += ("?");

    dataString += (",");

    dataString += (gps.satellites.value());

    dataString += (",");

    dataString += (gps.speed.mps());

    dataString += (",");

  }

  else

  {

    dataString = "0,0,?,0,0,";

  }

  return(dataString);

}



void BME_SETUP(){

  while(!Serial) {}

  Wire.begin();

  while(!bme.begin()){

    Serial.println("Could not find BME280 sensor!");
```

```
    delay(1000);

  }

  switch(bme.chipModel()){

     case BME280::ChipModel_BME280:

       Serial.println("Found BME280 sensor! Success.");

       break;

     case BME280::ChipModel_BMP280:

       Serial.println("Found BMP280 sensor! No Humidity available.");

       break;

     default:

       Serial.println("Found UNKNOWN sensor! Error!");

  }

}



float BME280Data(){

  float temp(NAN), hum(NAN), pres(NAN);

  BME280::TempUnit tempUnit(BME280::TempUnit_Celsius);

  BME280::PresUnit presUnit(BME280::PresUnit_Pa);

  bme.read(pres, temp, hum, tempUnit, presUnit);

  height = (sea-pres)/(1.293*grav);

  return (height);



}
```

```
void mpu_setup(){
  setup_mpu_6050_registers();


  for (int cal_int = 0; cal_int < 2000 ; cal_int ++){
    if(cal_int % 125 == 0);
    read_mpu_6050_data();                                        //Read the raw acc and gyro data from the MPU-6050
    gyro_x_cal += gyro_x;                                         //Add the gyro x-axis offset to the gyro_x_cal variable
    gyro_y_cal += gyro_y;                                         //Add the gyro y-axis offset to the gyro_y_cal variable
    gyro_z_cal += gyro_z;                                         //Add the gyro z-axis offset to the gyro_z_cal variable
    delay(3);                                                    //Delay 3us to simulate the 250Hz program loop
  }
  gyro_x_cal /= 2000;                                          //Divide the gyro_x_cal variable by 2000 to get the avarage offset
  gyro_y_cal /= 2000;                                          //Divide the gyro_y_cal variable by 2000 to get the avarage offset
  gyro_z_cal /= 2000;                                          //Divide the gyro_z_cal variable by 2000 to get the avarage offset


  loop_timer = micros();                                       //Reset the loop timer
```

```
}


float pitch(){
  read_mpu_6050_data();                                                          //Read the raw acc and gyro data from the MPU-6050


  gyro_x -= gyro_x_cal;                                                          //Subtract the offset calibration value from the raw gyro_x value
  gyro_y -= gyro_y_cal;                                                          //Subtract the offset calibration value from the raw gyro_y value
  gyro_z -= gyro_z_cal;                                                          //Subtract the offset calibration value from the raw gyro_z value


  //Gyro angle calculations
  //0.0000611 = 1 / (250Hz / 65.5)
  angle_pitch += gyro_x * 0.0000611;                                             //Calculate the traveled pitch angle and add this to the angle_pitch variable
  angle_roll += gyro_y * 0.0000611;                                              //Calculate the traveled roll angle and add this to the angle_roll variable


  //0.000001066 = 0.0000611 * (3.142(PI) / 180degr) The Arduino sin function is in radians
  angle_pitch += angle_roll * sin(gyro_z * 0.000001066);                         //If the IMU has yawed transfer the roll angle to the pitch angel
```

```
  angle_roll -= angle_pitch * sin(gyro_z * 0.000001066);                //If the IMU has yawed transfer the pitch angle to the roll angel


  //Accelerometer angle calculations
  acc_total_vector = sqrt((acc_x*acc_x)+(acc_y*acc_y)+(acc_z*acc_z));     //Calculate the total accelerometer vector
  //57.296 = 1 / (3.142 / 180) The Arduino asin function is in radians
  angle_pitch_acc = asin((float)acc_y/acc_total_vector)* 57.296;       //Calculate the pitch angle
  angle_roll_acc = asin((float)acc_x/acc_total_vector)* -57.296;       //Calculate the roll angle


  //Place the MPU-6050 spirit level and note the values in the following two lines for calibration
  angle_pitch_acc -= 0.0;                                              //Accelerometer calibration value for pitch
  angle_roll_acc -= 0.0;                                               //Accelerometer calibration value for roll


  if(set_gyro_angles){                                                 //If the IMU is already started
    angle_pitch = angle_pitch * 0.9996 + angle_pitch_acc * 0.0004;     //Correct the drift of the gyro pitch angle with the accelerometer pitch angle
    angle_roll = angle_roll * 0.9996 + angle_roll_acc * 0.0004;        //Correct the drift of the gyro roll angle with the accelerometer roll angle
```

```
  }

  else{                                                                 //At first start

    angle_pitch = angle_pitch_acc;                                      //Set the gyro pitch angle equal to the accelerometer pitch angle

    angle_roll = angle_roll_acc;                                        //Set the gyro roll angle equal to the accelerometer roll angle

    set_gyro_angles = true;                                             //Set the IMU started flag

  }



  //To dampen the pitch and roll angles a complementary filter is used

  angle_pitch_output = angle_pitch_output * 0.9 + angle_pitch * 0.1;   //Take 90% of the output pitch value and add 10% of the raw pitch value

  angle_roll_output = angle_roll_output * 0.9 + angle_roll * 0.1;

  //Serial.print(angle_pitch_output);

  //Serial.println(angle_roll_output);



  return(angle_pitch_output);
 //Write the roll and pitch values to the LCD display

}

float roll(){

  read_mpu_6050_data();                                                 //Read the raw acc and gyro data from the MPU-6050

```

```
  gyro_x -= gyro_x_cal;                                         //Subtract
the offset calibration value from the raw gyro_x value

  gyro_y -= gyro_y_cal;                                         //Subtract
the offset calibration value from the raw gyro_y value

  gyro_z -= gyro_z_cal;                                         //Subtract
the offset calibration value from the raw gyro_z value



  //Gyro angle calculations

  //0.0000611 = 1 / (250Hz / 65.5)

  angle_pitch += gyro_x * 0.0000611;                           //Calculate
the traveled pitch angle and add this to the angle_pitch variable

  angle_roll += gyro_y * 0.0000611;                            //Calculate
the traveled roll angle and add this to the angle_roll variable



  //0.000001066 = 0.0000611 * (3.142(PI) / 180degr) The Arduino sin function is
in radians

  angle_pitch += angle_roll * sin(gyro_z * 0.000001066);          //If the
IMU has yawed transfer the roll angle to the pitch angel

  angle_roll -= angle_pitch * sin(gyro_z * 0.000001066);          //If the
IMU has yawed transfer the pitch angle to the roll angel



  //Accelerometer angle calculations

                              acc_total_vector                            =
sqrt((acc_x*acc_x)+(acc_y*acc_y)+(acc_z*acc_z));    //Calculate   the   total
accelerometer vector

  //57.296 = 1 / (3.142 / 180) The Arduino asin function is in radians
```

```
  angle_pitch_acc = asin((float)acc_y/acc_total_vector)* 57.296;       //Calculate the pitch angle
  angle_roll_acc = asin((float)acc_x/acc_total_vector)* -57.296;       //Calculate the roll angle


  //Place the MPU-6050 spirit level and note the values in the following two lines for calibration
  angle_pitch_acc -= 0.0;                                              //Accelerometer calibration value for pitch
  angle_roll_acc -= 0.0;                                               //Accelerometer calibration value for roll


  if(set_gyro_angles){                                                 //If the IMU is already started
    angle_pitch = angle_pitch * 0.9996 + angle_pitch_acc * 0.0004;     //Correct the drift of the gyro pitch angle with the accelerometer pitch angle
    angle_roll = angle_roll * 0.9996 + angle_roll_acc * 0.0004;        //Correct the drift of the gyro roll angle with the accelerometer roll angle
  }
  else{                                                                //At first start
    angle_pitch = angle_pitch_acc;                                     //Set the gyro pitch angle equal to the accelerometer pitch angle
    angle_roll = angle_roll_acc;                                       //Set the gyro roll angle equal to the accelerometer roll angle
```

```
    set_gyro_angles = true;                                              //Set
the IMU started flag

  }



  //To dampen the pitch and roll angles a complementary filter is used

  angle_pitch_output = angle_pitch_output * 0.9 + angle_pitch * 0.1;    //Take
90% of the output pitch value and add 10% of the raw pitch value

  angle_roll_output = angle_roll_output * 0.9 + angle_roll * 0.1;

  //Serial.print(angle_pitch_output);

  //Serial.println(angle_roll_output);



  return(angle_roll_output);
//Write the roll and pitch values to the LCD display

}




void read_mpu_6050_data(){                                     //Subroutine
for reading the raw gyro and accelerometer data

  Wire.beginTransmission(0x68);                                     //Start
communicating with the MPU-6050

  Wire.write(0x3B);                                                  //Send
the requested starting register

  Wire.endTransmission();                                             //End
the transmission

  Wire.requestFrom(0x68,14);                                       //Request
14 bytes from the MPU-6050
```

```
  while(Wire.available() < 14);                                                   //Wait
until all the bytes are received

  acc_x = Wire.read()<<8|Wire.read();                                              //Add
the low and high byte to the acc_x variable

  acc_y = Wire.read()<<8|Wire.read();                                              //Add
the low and high byte to the acc_y variable

  acc_z = Wire.read()<<8|Wire.read();                                              //Add
the low and high byte to the acc_z variable

  temperature = Wire.read()<<8|Wire.read();                                        //Add
the low and high byte to the temperature variable

  gyro_x = Wire.read()<<8|Wire.read();                                             //Add
the low and high byte to the gyro_x variable

  gyro_y = Wire.read()<<8|Wire.read();                                             //Add
the low and high byte to the gyro_y variable

  gyro_z = Wire.read()<<8|Wire.read();                                             //Add
the low and high byte to the gyro_z variable



}



void setup_mpu_6050_registers(){

  //Activate the MPU-6050

  Wire.beginTransmission(0x68);                                                   //Start
communicating with the MPU-6050

  Wire.write(0x6B);                                                                //Send
the requested starting register

  Wire.write(0x00);                                                                 //Set
the requested starting register
```

```
  Wire.endTransmission();                                                    //End
the transmission

  //Configure the accelerometer (+/-8g)

  Wire.beginTransmission(0x68);                                              //Start
communicating with the MPU-6050

  Wire.write(0x1C);                                                          //Send
the requested starting register

  Wire.write(0x10);                                                          //Set
the requested starting register

  Wire.endTransmission();                                                    //End
the transmission

  //Configure the gyro (500dps full scale)

  Wire.beginTransmission(0x68);                                              //Start
communicating with the MPU-6050

  Wire.write(0x1B);                                                          //Send
the requested starting register

  Wire.write(0x08);                                                          //Set
the requested starting register

  Wire.endTransmission();                                                    //End
the transmission

}
```

B) Robot Main Program:

```
from time import sleep

import RPi.GPIO as GPIO

import serial

from Get_Info import *

import cv2

import numpy as np

from picamera2 import Picamera2

import time

#from gpiozero import AngularServo

picam2 = Picamera2()

#Obstacle Avoidance

#servo = AngularServo(37, min_pulse_width=0.0006, max_pulse_width=0.0023)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(22, GPIO.OUT)

pwm_f=GPIO.PWM(22,50)

pwm_f.start(0)

GPIO.setup(37, GPIO.OUT)

pwm_r=GPIO.PWM(37,50)

pwm_r.start(0)

# attribute the pins from the ultrasonic sensor

TRIG_f = 3
```

```
ECHO_f = 5

GPIO.setup(TRIG_f,GPIO.OUT)

GPIO.setup(ECHO_f,GPIO.IN)

GPIO.output(TRIG_f, False)

TRIG_r = 13

ECHO_r = 15

GPIO.setup(TRIG_r,GPIO.OUT)

GPIO.setup(ECHO_r,GPIO.IN)

GPIO.output(TRIG_r, False)

#OBJECT FOLLOWING



#pan=Servo(pin=13)

#tilt=Servo(pin=12)

panAngle=0

tiltAngle=0



#pan.set_angle(panAngle)

# tilt.set_angle(tiltAngle)



def TrackX(val):

    global xPos

    xPos=val

    #print('x Poistion', xPos)
```

```
def TrackY(val):
    global yPos
    yPos=val
    #print('y Poistion', yPos)


def TrackW(val):
    global boxW
    boxW=val
    #print('Box width', boxW)


def TrackH(val):
    global boxH
    boxH=val
    #print('Box height', boxH)


dispW=1280
dispH=720
picam2.preview_configuration.main.size = (dispW,dispH)
picam2.preview_configuration.main.format = "RGB888"
picam2.preview_configuration.controls.FrameRate=30
picam2.preview_configuration.align()
picam2.configure("preview")
```

```
picam2.start()


fps=0

pos=(30,60)

font=cv2.FONT_HERSHEY_SIMPLEX

height=1.5

weight=3

myColor=(0,0,255)


upperLeft=(250,20)

lowerRight=(800,650)

rColor=(255,0,255)

thickness=3


# hueLow=115

# hueHigh=130

# satLow=100

# satHigh=255

# valLow=100

# valHigh=255


#lowerBound=np.array([hueLow,satLow,valLow])

#upperBound=np.array([hueHigh,satHigh,valHigh])
```

```
cv2.namedWindow('My Trackbars')
cv2.createTrackbar('X Pos','My Trackbars',10,dispW-1,TrackX)
cv2.createTrackbar('Y Pos','My Trackbars',10,dispH-1,TrackY)
cv2.createTrackbar('Box Width','My Trackbars',10,dispW-1,TrackW)
cv2.createTrackbar('Box Height','My Trackbars',10,dispH-1,TrackH)



def onTrack1(val):
    global hueLow
    hueLow=val
    #print('hueLow', hueLow)

def onTrack2(val):
    global hueHigh
    hueHigh=val
    #print('hueHigh', hueHigh)

def onTrack3(val):
    global satLow
    satLow=val
    #print('satLow', satLow)
```

```
def onTrack4(val):
    global satHigh
    satHigh=val
    #print('satHigh', satHigh)

def onTrack5(val):
    global valLow
    valLow=val
    #print('valLow', valLow)

def onTrack6(val):
    global valHigh
    valHigh=val
    #print('valHigh', valHigh)


cv2.namedWindow('My Trackbar Colours')
cv2.createTrackbar('hueLow','My Trackbar Colours',1,179,onTrack1)
cv2.createTrackbar('hueHigh','My Trackbar Colours',68,179,onTrack2)
cv2.createTrackbar('satLow','My Trackbar Colours',71,255,onTrack3)
cv2.createTrackbar('satHigh','My Trackbar Colours',255,255,onTrack4)
cv2.createTrackbar('valLow','My Trackbar Colours',100 ,255,onTrack5)
cv2.createTrackbar('valHigh','My Trackbar Colours',255,255,onTrack6)
```

```
GPIO.setmode(GPIO.BOARD)
button1=19
button2=21
GPIO.setup(button1,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(button2,GPIO.IN,pull_up_down=GPIO.PUD_UP)

PWR1, ENA1, IN1, IN2, GND = 2, 32, 31, 29, 39
GPIO.setup(ENA1, GPIO.OUT)
GPIO.setup(IN1, GPIO.OUT)
GPIO.setup(IN2, GPIO.OUT)
PWMA = GPIO.PWM(ENA1, 100)
PWMA.start(0)

PWR2, ENA2, IN3, IN4, GND = 4, 12, 18, 16, 34
GPIO.setup(ENA2, GPIO.OUT)
GPIO.setup(IN3, GPIO.OUT)
```

```
GPIO.setup(IN4, GPIO.OUT)

PWMB = GPIO.PWM(ENA2, 100)

PWMB.start(0)


ser = serial.Serial(

port='/dev/ttyUSB1',

baudrate = 9600,

parity=serial.PARITY_NONE,

stopbits=serial.STOPBITS_TWO,

timeout=1

)


data = [0,0]


def calculate_distance_f():

    pulse_duration = 0.0

    pulse_end = 0.0

    GPIO.setwarnings(False)

    GPIO.output(TRIG_f, True)

    time.sleep(0.00001)

    GPIO.output(TRIG_f, False)

    while GPIO.input(ECHO_f)==0:

        pulse_start = time.time()
```

```
    while GPIO.input(ECHO_f)==1:

        pulse_end = time.time()

                       pulse_duration          =          pulse_end          -
pulse_start


    distance = pulse_duration * 17150

    distance = round(distance+1.15, 2)

    return distance



def calculate_distance_r():

    pulse_duration = 0.0

    pulse_end = 0.0

    GPIO.setwarnings(False)

    GPIO.output(TRIG_r, True)

    time.sleep(0.00001)

    GPIO.output(TRIG_r, False)

    while GPIO.input(ECHO_r)==0:

        pulse_start = time.time()

    while GPIO.input(ECHO_r)==1:

        pulse_end = time.time()

                       pulse_duration          =          pulse_end          -
pulse_start


    distance = pulse_duration * 17150
```

```
    distance = round(distance+1.15, 2)

    return distance



def O_V(p):

    if(p == 1.1):

        print("left front")

        #Motor_control(0, 0, 1, 0)

    if(p == 1.2):

        print("center front")

        Motor_control(-100, -100, 2, 1)

        Motor_control(100, -100, 3, 1)

        Motor_control(100, 100, 3, 1)

        Motor_control(-100, 100, 3, 1)

    if(p == 1.3):

        print("right front")

        #Motor_control(0, 0, 1, 0)

    if(p == 2.1):

        print("left rear")

        #Motor_control(0, 0, 1, 0)

    if(p == 2.2):

        print("center rear")

        #Motor_control(0, 0, 1, 0)
```

```
    if(p == 2.3):

        print("right rear")

        #Motor_control(0, 0, 1, 0)


def Motor_control(M1, M2, Run, Mode):


    if(M1 == 0):

        GPIO.output(IN1, GPIO. LOW)

        GPIO.output(IN2, GPIO.LOW)

    if(M2 == 0):

        GPIO.output(IN3, GPIO.LOW)

        GPIO.output(IN4, GPIO.LOW)

    if(M1 > 0):

        if(Mode == 0):

            pwm_f.ChangeDutyCycle(10)

            if(calculate_distance_f()< 50):

                print("Obstical Front Left")

                O_V(1.1)

        PWMA.ChangeDutyCycle(M1)

        GPIO.output(IN1, GPIO.HIGH)

        GPIO.output(IN2, GPIO.LOW)


    if(M1 < 0):
```

```
        #print(-1*M1)

        if(Mode == 0):

            #servo.angle = 90

            pwm_r.ChangeDutyCycle(5)

             #sleep(0.5)

            if(calculate_distance_r()< 50):

                print("Obstical Back Left")

                O_V(2.1)

        PWMA.ChangeDutyCycle(-1*M1)

        GPIO.output(IN1, GPIO.LOW)

        GPIO.output(IN2, GPIO.HIGH)


    if(M2 > 0):

        if(Mode == 0):

             pwm_f.ChangeDutyCycle(5)

             if(calculate_distance_f()< 50):

                 print("Obstical Front Right")

                 O_V(1.3)

        PWMB.ChangeDutyCycle(M2)

        GPIO.output(IN3, GPIO.LOW)

        GPIO.output(IN4, GPIO.HIGH)


    if(M2 < 0):
```

```
        if(Mode == 0):

             pwm_r.ChangeDutyCycle(10)

             if(calculate_distance_r()< 50):

                 print("Obstical Back Right")

                 O_V(2.3)


        PWMB.ChangeDutyCycle(-1*M2)

        GPIO.output(IN3, GPIO.HIGH)

        GPIO.output(IN4, GPIO.LOW)

    if(Mode == 0):

        if(M1 < 0 and M2 < 0):

            pwm_r.ChangeDutyCycle(8)

            if(calculate_distance_r()< 50):

                print("Obstical Back")

                O_V(2.2)


    if(Mode == 0):

        if(M1 > 0 and M2 > 0):

            pwm_f.ChangeDutyCycle(8)

            if(calculate_distance_f()< 50):

                print("Obstical Front")

                O_V(1.2)

    sleep(Run)
```

```python
def Mapping(n):
    print(n, "map size")
    for x in range(n):
        print("line", x)
        Motor_control(100, 100, n, 0)
        if((x%2) == 0):
            print("end1")
            Motor_control(100, -100, 3, 0)
            Motor_control(100, 100, 2.5, 0)
            Motor_control(100, -100, 3, 0)
        else:
            print("end2")
            Motor_control(-100, 100, 3, 0)
            Motor_control(100, 100, 2.5, 0)
            Motor_control(-100, 100, 3, 0)


#im= picam2.capture_array()
#cv2.imshow("Camera", im)


while(True):
    line=str(ser.readline())
    print(line)
```

```
    im= picam2.capture_array()

    cv2.setWindowProperty("window",cv2.WND_PROP_FULLSCREEN,cv2.WINDOW_FULLSCRE
EN)

    cv2.imshow("window", im)


    if cv2.waitKey(1)==ord('q'):

        break

    if GPIO.input(button1)==0:

        #print ("Program 1")

        #im= picam2.capture_array()

        #cv2.imshow("Camera", im)


        dataBuffer = data

        #line=str(ser.readline())

        #print(line)

        if len(line)>7:

            data = (line[2:-5].replace(","," ").split())

            if len(data) == 3:

                data = [data[0],data[-1]]


            try:

                data = [int(data[0]),int(data[1])]

            except:
```

```
            print('Using previous value')

            data = dataBuffer


        if data[0] > 100:

            data[0] = 100

        if data[0] < -100:

            data[0] = -100

        if data[1] > 100:

            data[1] = 100

        if data[1] < -100:

            data[1] = -100

        Motor_control(data[1],data[0], 0, 1)


if GPIO.input(button2)==0:


    tStart=time.time()

    im= picam2.capture_array()

    imHSV=cv2.cvtColor(im,cv2.COLOR_BGR2HSV)




    lowerBound=np.array([hueLow,satLow,valLow])

    upperBound=np.array([hueHigh,satHigh,valHigh])
```

```
        myMask =cv2.inRange(imHSV,lowerBound,upperBound)

        myMaskSm=cv2.resize(myMask,(int(dispW/2),int(dispH/2)))




        objectofInterest=cv2.bitwise_and(im,im,mask=myMask)

        objectofInterestSm=cv2.resize(objectofInterest,(int(dispW/2),int(dispH
/2)))


                                                            contours,           junk=
cv2.findContours(myMask,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

        if len(contours)>0:

                  contours=sorted(contours,  key=lambda  x:cv2.contourArea(x),
reverse=True)

            cv2.drawContours(im,contours,0,(255,0,0),3)

            contour=contours[0]

            x,y,w,h=cv2.boundingRect(contour)

            cv2.rectangle(im,(x,y),(x+w,y+h),(0,255,0),3)

            error=(x+w/2)-dispW/2


            robot_dir = int(error/6)

            area = int((w*h)/100)

            #print(area)

            if(area == 0):

                Motor_control(0, 0, 0, 0)
```

```
            if((area > 30) and (area < 130)):

                print("dead zone")

                if(robot_dir < 0):

                    #print("Right motor ON")

                    Motor_control(100, 0, 0, 0)

                if(robot_dir > 0):

                    #print("Left motor ON")

                    Motor_control(0, 100, 0, 0)

            if(area < 30) and (area > 1):

                print("forward")

                Motor_control(100, 100, 0, 0)

            if(area > 130):

                print("backward")

                Motor_control(-100, -100, 0, 0)


        cv2.putText(im,str(int(fps))+' FPS',pos,font,height,myColor,weight)

        cv2.rectangle(im,(xPos,yPos),(xPos+boxW,yPos+boxH),rColor,thickness)

        ROI=im[yPos:yPos+boxH,xPos:xPos+boxW]



        cv2.setWindowProperty("window",cv2.WND_PROP_FULLSCREEN,cv2.WINDOW_FULL
SCREEN)

        cv2.imshow("window", im)

            #cv2.imshow("Object of Interest",objectofInterestSm)
```

```
            #cv2.imshow("myMask",myMaskSm)

        if cv2.waitKey(1)==ord('q'):

            break

        tEnd=time.time()

        loopTime=tEnd-tStart

        fps=.9*fps + .1*(1/loopTime)











    if(GPIO.input(button2)==1) and (GPIO.input(button1)==1):

        p_data = int(read())

        if(p_data > 0):

            Mapping(p_data)

        else:

            print("Map Data Not Initiated")



            #except:

            #    print("Mapping Data Not Recived")
```

C) Get Info Program:

```
import serial

import numpy as np

from time import sleep

from PLOTTING import plot

try:

    ser = serial.Serial(port='COM5',baudrate=57600,parity=serial.PARITY_NONE,stopbits=serial.STOPBITS_ONE,bytesize=serial.EIGHTBITS,timeout=0)

except:

    print("no serial")

def get_info():

    StringBuffer = ""

    new_data = []

    buffer_data = []



    i = 0



    while True:

            StringBuffer += str(ser.read(1)).replace('b','').replace('\\',",").replace("'","").replace('r','').replace('n','').replace(',',' ')

        StringBuffer2 = StringBuffer.split()

        for count, value in enumerate(StringBuffer2):
```

```
            if value == "?":

                idx = count-2

                try:

                    new_data = StringBuffer2[idx:idx+8]

                    new_data[0] =float(new_data[0][:-7]+"."+new_data[0][-7:])

                    new_data[1] = float(new_data[1][:-7]+"."+new_data[1][-7:])

                    buffer_data = new_data

                    #print(StringBuffer2)

                    StringBuffer2 = []

                except:

                    #print("error")

                    new_data = buffer_data

                    #print(StringBuffer2)



                if len(new_data) == 8:

                    #print(new_data)



                    return new_data

                else:

                    pass




def GetCoordinates():
```

```
    data = get_info()

    print(data)

    return data[0], data[1]


def GetAltitude():

    data = get_info()

    print(data)

    return data[5]


def GetSat():

    data = get_info()

    return data[3]


def GetSpeed():

    data = get_info()

    return data[4]


def GetPR():

    data = get_info()

    return data[6], data[7]


def GetMap(n):

    dim = str(n)
```

```
ser.write(dim.encode('utf-8'))

a = np.zeros(shape=(n,n))

for i in range(n):

    for j in range(n):

        val = 0.0

        for k in range(10):

            data = get_info()

            val += float(data[5])

            #print(val)

            #sleep(0.1)



        val = round((val/10), 2)

        #print(val)

        a[i, j] = val


        #data[5].append(a[i, j])


    print("ping")

    sleep(8.5)


print(a.max())

nor = lambda i: a.max() - i
```

```
    vectorized_nor= np.vectorize(nor)

    map_nor= vectorized_nor(a)

    print(map_nor)

    plot(n, map_nor)

    #data[5]
```

D) Mapping Program:

```
import matplotlib.pyplot as plt

import numpy as np

from scipy import interpolate



ax = plt.axes(projection = "3d")



def plot(n, arr):

    x_data = np.arange(0, n, 1)

    y_data = np.arange(0, n, 1)

    X, Y = np.meshgrid(x_data, y_data)



    Z = arr



    Z_I = interpolate.interp2d(x_data, y_data, Z, kind="cubic")

    Xnew =np.linspace(0, n, 10)
```

```
Ynew =np.linspace(0, n, 10)

X_I, Y_I = np.meshgrid(Xnew, Ynew)



Z_new = Z_I(Xnew, Ynew)



Interp = ax.plot_surface(X_I, Y_I, Z_new, cmap ="viridis")

ax.set_zlim(0, 10)

plt.show()
```

Bibliography

Academy, B. A., 2021. *YouTube.* [Online]
Available at: https://www.youtube.com/watch?v=y4Y1R2wHeIY
[Accessed 30 04 2023].

Alli, H., 2007. Design and prototype of a six-leggedwalking insect robot. *Industrial Robot,* p. 12.

Anon., n.d. [Online]
Available at: https://www.researchgate.net/figure/Interpolation-in-3D-space_fig3_319653273

arjun2000ananth, 2022. *Autodesk Instrucatbales.* [Online]
Available at: https://www.instructables.com/Object-Detection-on-Raspberry-Pi/
[Accessed 2 05 2023].

Automaticaddison, 2022. *Automatic Addison.* [Online]
Available at: https://automaticaddison.com/how-to-make-an-object-tracking-robot-using-raspberry-pi/
[Accessed 30 04 2023].

autonaticdai, 2020. *Github.* [Online]
Available at: https://github.com/automaticdai/rpi-object-detection
[Accessed 30 04 2023].

Bareiss D, v. d. B. J. L. K., 2015. Stochastic automatic collision avoidance for tele-operated unmanned aerial vehicles.. *In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 4818-4825.

Borenstein, C. Y. a. J., 2003. *A new terrain mapping method for mobile robots obstacle negotiation.* [Online]
Available at: http://www-personal.umich.edu/~johannb/Papers/paper97.pdf

Borenstein, L. O. a. J., 2002. Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain. In: *Fuzzy Logic Expert Rule-based Position Estimation for*

Mobile Robots on Rugged Terrain. s.l.:FLEXnav Proc. IEEE International Conference on Robotics and Automation , pp. 317-322.

D. C. Guastella, L. C. C. D. M. a. G. M., 2017. *A Global Path Planning Strategy for a UGV from Aerial Elevation Maps for Disaster Response.* [Online]
Available at:
https://pdfs.semanticscholar.org/58a7/1b72ef67bfff9be8c367366ddef9cba51147.pdf

Dejan, 2015. *How to Mechatronics.* [Online]
Available at: https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/
[Accessed 05 05 2023].

Dr. Y.R. Kharde*1, A. M. P. P. S. P. P. P., 2022. DESIGN AND DEVELOPMENT OF ALL TERRAIN TRACK ROBOT. *International Research Journal of Modernization in Engineering Technology and Science* , 04(06), p. 5.

Duddell, S. L. a. D., 2015. *Computer Science.* 1 ed. Cambridge: Cambridge University Press.

Fluxwood, 2017. *Fluxwood.* [Online]
Available at: https://fluxwood.org/front/electronics/printing/2017/12/31/Tracking-Tank.html
[Accessed 27 04 2023].

FLYABILITY, n.d. *WHAT IS SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)?.* [Online]
Available at: https://www.flyability.com/simultaneous-localization-and-mapping

hambling, D., 2008. *Tanks: Why tracks beat wheels.* [Online]
Available at: https://www.bbc.com/future/article/20150611-tanks-why-tracks-beat-wheels

INCOGNIA, 2023. *All you need to know about location technology [The Ultimate Guide].* [Online]
Available at: https://www.incognia.com/the-authentication-reference/all-there-is-to-know-about-location-technology-the-ultimate-guide
[Accessed 08 05 2023].

INFINITI, n.d. *What is an LRF (Laser Rangefinder) and what is its purpose?.* [Online]
Available at: https://www.infinitioptics.com/glossary/lrf-laser-rangefinder#:~:text=An%20LRF%20or%20Laser%20Rangefinder,narrow%20beam%20towards%20the%20object

Institutional Integrity and risk Management, n.d. *Lifting and Material Handling.* [Online]
Available at: https://ehs.unc.edu/topics/ergonomics/lifting-and-material-handling/#:~:text=Where%20possible%2C%20use%20mechanical%20means,the%20person%20doing%20the%20lifting.

Interpolation, M., n.d. [Online]
Available at: https://www.researchgate.net/figure/Interpolation-in-3D-space_fig3_319653273

Jianwei Zhao 1, *. ,. T. H. 1. ,. S. W. 2. C. L. 1. J. F. 1. a. S. L. 1., 2021. Design and Research of All-Terrain Wheel-Legged Robot.

JN, Y., 2017. *Collision avoidance algorithm for unmanned aerial or ground vehicles.*, bo AKademi University: s.n.

Jost, D., 2019. *Fierce Electronics.* [Online]
Available at: https://www.fierceelectronics.com/sensors/what-accelerometer
[Accessed 05 05 2023].

Khatib, S. Q. a. O., 1993. Elastic Bands Connecting Path Planning and Control. *Proc. of the IEEE Int. Conf. on Robotics and automtion*, pp. 802-807.

Ladislav Jurišica, F. D. D. K. a. A. B., 2012. High Precision GNSS Guidance. p. 10.

Laoyan, S., 2022. *asana.* [Online]
Available at: https://asana.com/resources/agile-methodology
[Accessed 11 June 2023].

Learning, G. -. M., 2023. *Machine Learning.* [Online]
Available at: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit#:~:text=TensorFlow%20is%20an%20end%2Dto,and%20train%20machine%20learning%20models
[Accessed 02 05 2023].

LearnOpenCV, 2023. *LearnOpenCV.* [Online]
Available at: https://learnopencv.com/deep-learning-with-opencvs-dnn-module-a-definitive-guide/#:~:text=The%20OpenCV%20DNN%20module%20supports%20the%20famous%20Darknet%20framework%20as,file%20will%20always%20be%20a%20
[Accessed 03 05 2023].

LiteTrax, n.d. *Wheels vs Tracks: Advantages and disadvantages,* s.l.: s.n.

Liu, H., 2001. Accelerometer for mobile robot positioning. p. 10.

Michael Shneier, R. B., 2015. *Literature Review of Mobile Robots.* [Online]
Available at: https://www.govinfo.gov/content/pkg/GOVPUB-C13-f0af4a0660a95893452e7ee39defce71/pdf/GOVPUB-C13-f0af4a0660a95893452e7ee39defce71.pdf

Michal Krumnikl, P. M. D. S., 2016. *Calibration of Short Range 2D Laser Range Finder for 3D SLAM Usage.* [Online]
Available at: https://www.hindawi.com/journals/js/2016/3715129/

Mizokami, K., 2018. *Popularmechanics.* [Online]
Available at: Why Tanks Have Tracks Instead of Wheels

mjwhite8119, 2019. *Github.* [Online]
Available at: https://mjwhite8119.github.io/Robots/mpu6050
[Accessed 05 05 2023].

Mullen, C., 2020. *The Advantages of Tracks Over Wheels on a Tank.* [Online]
Available at: https://nationalinterest.org/blog/buzz/advantages-tracks-over-wheels-tank-118071

Muller, C., n.d. *handbook of design thinking.* [Online]
Available at:
https://www.researchgate.net/publication/329310644_Handbook_of_Design_Thinking/link/5c3d987b299bf12be3c8b626/download
[Accessed 10 July 2023].

Nagla, A. K. &. K. S., 2020. *An Improved Method for Extrinsic Calibration of Tilting 2D LRF.* [Online]
Available at: https://link.springer.com/article/10.1007/s10846-020-01147-7#:~:text=Tilting%202D%20LRF%20(a%20low,centre%20of%20the%202D%20LRF

Olivares-Mendez MA, C. P. M.-B. I. M. L., 2012. See-and-avoid quadcopter using fuzzy control optimized by cross entropy. *2012 IEEE internationak Conference on fuzzy Systems* , pp. 1-7.

Plosila1, J. N. Y. •. S. A. S. M. •. M.-H. H. •. J. H. •. H. T. •. J., n.d. *Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots.* [Online].

Priyaranjan Biswal, P. K. M., 2021. *Development of quadruped walking robots: A review.* [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S2090447920302501

Raj, A., 2017. *Circuit Digest.* [Online]
Available at: https://circuitdigest.com/microcontroller-projects/raspberry-pi-ball-tracking-robot-using-processing
[Accessed 30 04 2023].

Ramamurthy, M., 2005. Indoor Geo-location And Tracking Of Mobile Autonomous Robot. p. 96.

ResearchGate, 2023. *ResearchGate.* [Online]
Available at: https://www.researchgate.net/figure/Advantages-and-disadvantages-of-various-transduction-schemes_fig2_329007197
[Accessed 05 05 2023].

Rifandi, R., 2013. *An Insight About GPS.* [Online]
Available at: https://www.researchgate.net/publication/256545966_An_Insight_About_GPS
[Accessed 08 05 2023].

Rosebrock, A., 2017. *pyimagesearch.* [Online]
Available at: https://pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/
[Accessed 2 05 2023].

S. Jang, S. J. a. T. C. H., 2005. Collision Avoidance of a Mobile Robot for Moving Obstacles Based on Impedance Force Control Algorithm. *Proc. of the IEEE/RSJ Inte. Conf. on Intelligent Robots and Systems*, pp. 277-285.

Sarah Crowe, c. a. K. C. A. R. G. H. A. A. a. A. S., 2011. *The case study approach.* [Online]
Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3141799/
[Accessed 9 June 2023].

Simone, D., 2018. *Machines.* [Online]
Available at: https://doi.org/10.3390/machines6020018

SimpliLearn, 2022. *SimpliLearn.* [Online]
Available at: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras#:~:text=ProgramExplore%20Program-,Why%20Do%20We%20Need%20Keras%3F,and%20explains%20user%20error%20clearly
[Accessed 30 04 2023].

SimpliLearn, 2023. *Object Detection Using OpenCV Python.* s.l.:Simplilearn.

Smlease, 2023. *Smlease.* [Online]
Available at: https://www.smlease.com/entries/mechanical-design-basics/what-is-gyroscope-how-gyroscopes-work-and-their-applications/
[Accessed 05 05 2023].

Soffar, H., 2020. *Online Sciences.* [Online]
Available at: https://www.online-sciences.com/technology/what-is-gyroscope-gyroscopes-review-use-types-advantages-disadvantages/
[Accessed 05 05 2023].

Tayal, S., 2019. *Autodesk Instructables.* [Online]
Available at: https://www.instructables.com/Object-Finding-Personal-Assistant-Robot-Ft-Raspber/
[Accessed 28 04 2023].

Tsai, Y.-H., 2006. Barometric altimeter short-term accuracy analysis. Issue Barometric Altimeter Short-Term Accuracy Analysis, p. 4.

US Department of Commerce NOAAA, 2020. *US Department of Commerce NOAAA (2020) Lidar, national.* [Online]
Available at: . https://oceanservice.noaa.gov/facts/lidar.html.

USGS, 2023. *Topographic Maps.* [Online]
Available at: https://www.usgs.gov/programs/national-geospatial-program/topographic-maps

USGS, 2023. *What is a digital elevation model (DEM)?.* [Online]
Available at: https://www.usgs.gov/faqs/what-digital-elevation-model-dem#:~:text=A%20Digital%20Elevation%20Model%20(DEM)%20is%20a%20representation%20of%20the,derived%20primarily%20from%20topographic%20maps

USGS, 2023. *What is Lidar data and where can I download it?.* [Online]
Available at: https://www.usgs.gov/faqs/what-lidar-data-and-where-can-i-download-it?qt-news_science_products=0#qt-news_science_products

USGS, n.d. *EROS Archive - Radar - IFSAR Orthorectified Radar Image (ORI) Alaska.* [Online]
Available at: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-radar-ifsar-orthorectified-radar-image-ori-alaska?qt-science_center_objects=0#qt-science_center_objects

Vectornav, 2023. *Vectornav.* [Online]
Available at: https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-gyros#:~:text=A%20ring%20laser%20gyroscope%20(RLG,Sagnac%20effect%20to%20detect%20rotation.
[Accessed 05 05 2023].

Wrona, M., 2020. *Mwrona.* [Online]
Available at: https://mwrona.com/posts/accel-roll-pitch/
[Accessed 06 05 2023].

Yueheng Zhou, M. L. C. S. J. L., 2022. *Kirin: A Quadruped Robot with High Payload Carrying Capability.* [Online]
Available at:

https://www.researchgate.net/publication/358687621_Kirin_A_Quadruped_Robot_with_High_Payload_Carrying_Capability