Department of Robotics and Automation
Jain Global Campus, Kanakapura Taluk - 562112
Ramanagara District, Karnataka, India

2019-2023

A Report on

"Mapping and Navigation Robot for Maze solving"

Submitted by

JAGAN V (19BTRRA012)

GIRISH P (19BTRRA019)

BHASKAR M (19BTRRA020)

THUMMALA DHEERAJ GOUD (19BTRRA021)

Under the guidance of
CHALUVARAJU PP
Professor

Department of Robotics and Automation
Faculty of Engineering and Technology JAIN
(Deemed-to-be University)

Faculty of Engineering & Technology

Department of Electronics & Communication Engineering

Jain Global campus
Kanakapura Taluk - 562112
Ramanagara District
Karnataka, India

CERTIFICATE

This is to certify that the project work titled **"MAPPING AND NAVIGATION ROBOT FOR MAZE SOLVING"** is carried out by **JAGAN V (19BTRRA012), GIRISH P (19BTRRA019), BHASKAR M (19BTRRA020), THUMMALA DHEERAJ GOUD (19BTRRA021)** are bonafide students of Bachelor of Technology at the Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY, Bengaluru in partial fulfillment for the award of degree in Bachelor of Technology in Electronics and Communication Engineering, during the academic year **2021-2022**.

CHALUVARAJU PP

Assistant/Associate Professor

Dept. of ECE,

Faculty of Engineering & Technology,

JAIN DEEMED-TO-BE UNIVERSITY

Date:

Dr. R. Sukumar

Head of the Department, Electronics and Communication,

Faculty of Engineering & Technology,

JAIN DEEMED-TO-BE UNIVERSITY

Date:

Dr. Hariprasad S.A

Director,

Faculty of Engineering & Technology,

JAIN DEEMED-TO-BE UNIVERSITY

Date:

Name of the Examiner — Signature of Examiner

1

2

DECLARATION

We, **JAGAN V (19BTRRA012), GIRISH P (19BTRRA019), BHASKAR M (19BTRRA020), THUMMALA DHEERAJ GOUD (19BTRRA021)** are student's of eighth semester B. Tech in **ROBOTICS AND AUTOMATION**, at Faculty of Engineering & Technology, **JAIN DEEMED-TO-BE UNIVERSITY**, hereby declare that the project titled **"MAPPING AND NAVIGATION ROBOT FOR MAZE SOLVING"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in ROBOTICS AND AUTOMATION Engineering** during the academic year **2022-2023**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Signature

JAGAN V
19BTRRA012

GIRISH P
19BTRRA019

BHASKAR M
19BTRRA020

THUMMALA DHEERAJ GOUD
19BTRRA021

Place: Bengaluru
Date :31/03/2023

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to **Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY** *for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.*

In particular we would like to thank ***Dr. HariPrasad S.A***, ***Director***, **Faculty of Engineering & Technology**, **JAIN DEEMED-TO-BE UNIVERSITY** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. R. Sukumar***, **Head of the department**, **Electronics and communication Engineering**, **JAIN DEEMED-TO-BE UNIVERSITY,** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***CHALUVARAJU PP.***, *Associate /* ***Assistant Professor***, **Dept. of Electronics and Communication Engineering**, **JAIN DEEMED-TO-BE UNIVERSITY**, *for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our Project Coordinator ***Mr. Sunil M P*** *and all the staff members of Electronics and Communication for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.

Signature of Students

ABSTRACT

The development of a LiDAR-based maze-solving robot is an exciting and innovative project that has the potential to revolutionize the field of robotics. The use of LiDAR sensors to create a 2D map of the maze is a cutting-edge technology that allows the robot to navigate through the maze with precision and accuracy. The robot's control system is designed to enable it to make decisions about which direction to take based on the information gathered from the LiDAR sensors. This means that the robot can operate autonomously without any human intervention, making it highly efficient and reliable.

One of the main challenges in developing this robot is to ensure that it can navigate through unknown maze environments with ease. To overcome this challenge, the LiDAR sensors must be highly accurate and capable of detecting obstacles in the robot's path. The robot's control system must also be designed to handle unexpected obstacles and make decisions about how to navigate around them. Additionally, the robot's software must be able to create an accurate and detailed map of the maze, which can be used to plan the robot's path towards the destination.

The successful development of this robot has the potential to revolutionize many different industries. For example, in the manufacturing industry, the robot could be used to navigate complex production lines, increasing efficiency and reducing the risk of human error. In the construction industry, the robot could be used to navigate complex building sites, improving safety and reducing construction times. Additionally, the robot could be used in search and rescue operations to navigate through complex and dangerous environments, improving the safety of rescuers and those in need of assistance.

In conclusion, the development of a LiDAR-based maze-solving robot is an exciting project with vast potential for numerous industries. The robot's ability to navigate through unknown maze environments without human intervention could have a significant impact on safety, efficiency, and productivity. The project's success will undoubtedly drive further research and development in the field of robotics, leading to even more innovative solutions in the future

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION TO TOPIC

1.1 Simultaneous localization and mapping (SLAM)

SLAM stands for Simultaneous Localization and Mapping. It is a technique that allows robots and other autonomous vehicles to create a map of an unknown environment while simultaneously keeping track of their own location within that environment.

SLAM works by using a variety of sensors, such as cameras, lidar, and inertial measurement units (IMUs), to gather information about the environment and the robot's movements. The robot then uses this information to create a map of the environment and to track its own location within that map.

The main components of SLAM are:

1. Sensor fusion: Sensor fusion is a technique that combines data from multiple sensors to create a more accurate and complete picture of the environment. In SLAM, sensor fusion is used to combine data from cameras, lidar, and IMUs to create a map of the environment and to track the robot's location within that map.
2. Localization: Localization is the process of determining the robot's location within a map. In SLAM, localization is achieved by using a variety of techniques, including dead reckoning, visual odometry, and feature matching.
3. Mapping: Mapping is the process of creating a map of an environment. In SLAM, mapping is achieved by using a variety of techniques, including feature extraction, feature matching, and map building.

SLAM is a powerful technique that has a wide range of applications. It is likely to become increasingly important in the future as robots and autonomous vehicles become more commonplace.

Here are some additional details about the challenges of SLAM:

1. Sensor noise and uncertainty: The accuracy of the sensor data can be affected by factors such as lighting conditions, sensor calibration, and environmental factors. This can make it difficult for SLAM to accurately localize the robot and create an accurate map of the environment.
2. Dynamic environments: SLAM can be challenging in dynamic environments, where objects are moving or changing. This can make it difficult for SLAM to track the robot's location and create an accurate map of the environment.
3. Occlusions: Occlusions can occur when objects block the robot's view of the environment. This can make it difficult for SLAM to track the robot's location and create an accurate map of the environment.

Despite these challenges, SLAM is a powerful technique that has a wide range of applications. It is likely to become increasingly important in the future as robots and autonomous vehicles become more commonplace.

Feature extraction: SLAM algorithms extract features, such as edges, corners, and key points, from sensor data to help identify and track objects and landmarks in the environment. These features are used to create a visual map of the environment that can be used for navigation and localization.

1. Mapping: SLAM algorithms use the sensor data and extracted features to create a map of the environment, including the locations of objects and landmarks. The map is continually updated as the robot or camera moves through the environment, allowing it to create a detailed and accurate representation of the surroundings.
2. Localization: SLAM algorithms use the map and sensor data to estimate the location and orientation of the robot or camera within the environment. This allows the robot or camera to navigate through the environment and avoid obstacles.

SLAM has a wide range of applications, including:

1. Autonomous vehicles: SLAM is used to create maps of the surrounding environment and localize the vehicle within the map, enabling it to navigate autonomously without human intervention.

2. Robotics: SLAM is used to create maps of indoor and outdoor environments, allowing robots to navigate and interact with the environment.
3. Augmented reality: SLAM is used to anchor virtual objects in the real world, creating an immersive and interactive experience for users.
4. Mapping and surveying: SLAM is used to create maps of indoor and outdoor environments, allowing for accurate and efficient mapping and surveying

Overall, SLAM is a powerful and important technology that has the potential to transform a wide range of industries and applications. As research in this field continues to advance, we can expect to see even more exciting developments in the years to come.

1.2 Robot operating system (ROS)

Robot Operating System (ROS) is an open-source software framework that helps developers build and reuse code between robotics applications. ROS is not an operating system (OS), but rather a middleware that sits on top of an existing OS (such as Linux) and provides a set of standardized communication protocols and message formats for different robot components to interact with each other. This allows developers to easily write modular and reusable code that can be used across a wide range of robot platforms and applications.

ROS provides a number of features that make it a powerful tool for robotics development, including:

Hardware abstraction: ROS provides a layer of abstraction that allows developers to write code that is independent of the specific hardware that the robot is using. This makes it easier to port code to new robots or to use different hardware components in the same robot.

Low-level device control: ROS provides a set of tools for controlling low-level devices, such as motors, sensors, and actuators. This makes it easier to write code that controls the robot's physical behavior.

Commonly used functionality: ROS provides a set of libraries that implement commonly used functionality, such as navigation, mapping, and object recognition. This makes it easier to write code that implements these features without having to reinvent the wheel.

Message-passing between processes: ROS uses a message-passing communication system to allow different components of a robot to communicate with each other. This makes it easier to design and implement complex robot systems.

Package management: ROS provides a package management system that makes it easy to install and manage ROS software. This makes it easier for developers to get started with ROS and to find the software they need for their projects.

ROS is a powerful tool that can be used to develop a wide range of robots. It is used by a large community of developers and is supported by a wide range of hardware and software. If you are interested in developing robots, ROS is a great place to start.

1.3Kalman filter

Kalman filtering is a mathematical technique for estimating the state of a dynamic system based on a series of measurements observed over time. The state of a system is defined as the set of variables that completely describe the system's behavior. For example, the state of a car might include its position, velocity, and acceleration.

Kalman filtering works by combining two types of information:

Measurements: Measurements are observations of the system's state. For example, a GPS sensor might measure the car's position.

Predictions: Predictions are estimates of the system's state based on its previous state and the laws of physics. For example, a Kalman filter might predict the car's position based on its current position and velocity.

Kalman filtering uses a combination of measurements and predictions to estimate the current state of the system. This estimate is more accurate than either the measurements or the predictions alone, because it takes into account the uncertainty in both.

Kalman filtering is a powerful tool that is used in a wide variety of applications, including:

Guidance, navigation, and control of vehicles: Kalman filtering is used to estimate the state of vehicles such as aircraft, spacecraft, and ships. This information is used to guide the vehicles to their destinations and to control their movements.

Signal processing: Kalman filtering is used to estimate the state of systems that generate signals, such as speech signals and radar signals.

Robotics: Kalman filtering is used to estimate the state of robots. This information is used to plan the robots' movements and to control their movements.

Kalman filtering is a powerful and versatile tool that can be used in a wide variety of applications. It is a valuable tool for engineers and scientists who need to estimate the state of dynamic systems.Kalman filtering is a tool that helps us estimate the state of a dynamic system based on a series of measurements observed over time. In the context of the central nervous system, Kalman filtering can be used to estimate the state of the motor system, which is the system of muscles and nerves that control movement.

The motor system is a complex system that is subject to a variety of factors, including noise and uncertainty. Kalman filtering can be used to take into account these factors and to provide an accurate estimate of the state of the motor system. This information can then be used to control movement more efficiently and accurately.

For example, Kalman filtering can be used to estimate the position of a limb based on measurements from sensors such as accelerometers and gyroscopes. This information can then be used to control the muscles that move the limb in order to achieve a desired goal.

Kalman filtering is a powerful tool that can be used to improve the control of movement. It is a valuable tool for engineers and scientists who are working to develop more efficient and accurate methods of controlling movement.

Here is a more concise version of the same information:

Kalman filtering is a mathematical technique that can be used to estimate the state of a dynamic system, such as the motor system, based on a series of measurements. This information can then be used to control movement more efficiently and accurately.

.

1.4 General introduction

Maze solving robots are autonomous robots designed to navigate through a maze or unknown environment without human intervention. These robots are equipped with sensors, algorithms, and other hardware and software technologies that enable them to scan their surroundings, build a map of the environment, and plan an optimal path through the maze to reach a specified destination.

Maze solving robots are a type of autonomous robot that use advanced technologies such as sensors, algorithms, and path planning to navigate through a maze or unknown environment. These robots are designed to operate without human intervention and can perform a wide range of tasks in hazardous or inaccessible environments.

One of the key technologies used in maze solving robots is SLAM, which stands for Simultaneous Localization and Mapping. This technology allows the robot to build a map of the environment and simultaneously localize itself within that map, enabling it to determine its position and orientation in real-time. This is achieved by using a variety of sensors such as cameras, Lidar, ultrasonic sensors, and infrared sensors to gather data about the environment.

Once the robot has built a map of the environment, path planning algorithms are used to determine the best route to the destination while avoiding obstacles and dead-ends. These algorithms use the map and sensor data to create a plan that optimizes the robot's movements, reducing the time it takes to navigate through the maze.

The development of maze solving robots has been driven by the increasing demand for automation and the need for robots that can perform tasks in hazardous or inaccessible environments. These robots have numerous applications in fields such as warehouse management, inspection, surveillance, search and rescue, and agriculture, among others.The project described in this context aims to develop a maze solving robot using Raspberry Pi and RP Lidar, which is a popular Lidar sensor that can be used for SLAM. The robot will use these technologies to navigate through a maze, build a map of the environment, localize itself within the map, and plan an optimal path to reach the destination. The project aims to showcase the potential of combining advanced hardware and software technologies to create a robot capable of autonomous navigation in unknown environments.

Overall, maze solving robots represent a promising area of research and development in the field of robotics, with numerous potential applications across a range of industries. As technology continues to advance, we can expect to see even more sophisticated and capable maze solving robots in the future.

The development of maze solving robots has been fuelled by the increasing demand for automation and the need for robots that can perform tasks in hazardous or inaccessible environments. These robots have various applications in fields such as warehouse management, inspection, surveillance, search and rescue, and agriculture, among others.

Maze solving robots use a range of sensors to detect their surroundings, including cameras, Lidar, ultrasonic sensors, and infrared sensors. They also use algorithms such as SLAM (Simultaneous Localization and Mapping) to build a map of the environment and localize themselves within it. Path planning algorithms are used to determine the best path to reach the destination while avoiding obstacles and dead-ends.

This project focuses on developing a maze solving robot using Raspberry Pi and RP Lidar and SLAM technology. The robot will be able to autonomously navigate through a maze, build a map of the environment, localize itself within it, and plan an optimal path to reach the destination. The project aims to showcase the potential of combining advanced hardware and software technologies to create a robot capable of autonomous navigation in unknown environments.

1.5 Objectives

- Navigation: The primary objective of the robot is to navigate through a maze using LiDAR technology for obstacle detection and localization.
- Mapping: The robot should be able to create a map of the maze as it navigates through it. The map could be used to optimize the robot's path, avoid obstacles, and potentially provide useful data for other applications.
- Speed and Efficiency: The robot should be able to solve the maze in a timely and efficient manner, taking the shortest possible path while avoiding obstacles.
- Accuracy: The LiDAR sensor should be accurate and reliable, allowing the robot to accurately detect obstacles and map the maze.

- Robustness: The robot should be robust and able to handle a variety of maze layouts and obstacles, including dead ends, sharp turns, and narrow passages.
- User-Friendly Interface: The robot should be easy to use and have a user-friendly interface that allows users to input maze information, view the robot's progress, and control its movements if necessary.
- Integration with other technologies: The robot should be able to integrate with other technologies, such as GPS and machine learning algorithms, to further optimize its performance and accuracy.

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

Table 1.1 LITERATURE REVIEW

SI no	TITLE	AUTHOR	DETAILS
1	A cross-correction LiDAR SLAM method for high accuracy 2D mapping of problematic scenario	Shoujun Jia, Chun Liu	This system summarizes the state-of the-art work on the et al. proposed and tested on Random maze, an LiDAR based library Tracking system All relevant positions features were supported using the ROS
2	A Robust Visual Odometry and Precipice Detection System Using Consumergrade Monocular Vision	H. Wu	In this system, a separate Maze application was developed so that the robot could use the path to solve the maze using Kalman Filter algorithms
3	Maze solving robot with automated obstacle avoidance	Rahul Kumar, Peni Jitoko	The Proposed system uses IR sensors and Arduino to create a proof-of concept smart Robot. To make robots mutinously intelligent with obstacle avoidance
4	Simultaneous Localization and Mapping in Maze Solving	Divya Bhaskaran, Charushree MB	Proposed system is based on navigation with SLAM, LIDAR and robot navigation algorithms

CHAPTER 3

RESEARCH GAPS

3.1 EXISTING SYSTEM

Maze solving robots have been an area of active research in recent years, with many different approaches proposed and studied in the literature. One of the research works involves the use of advanced sensors, such as ultrasonic sensors, IR sensors, GPS, and others, to enable the robot to navigate through the maze in real-time. In this approach, the robot is equipped with an array of sensors that continuously monitor the environment and provide information about obstacles, walls, and other features in the maze. The ultrasonic sensors, in particular, are used to detect objects and walls in the robot's vicinity, allowing it to avoid collisions and navigate through unknown environments .In addition to ultrasonic sensors, the research also incorporates IR sensors to detect the line on the maze floor, GPS to track the robot's location and movement, and other sensors to provide information about the maze configuration and environment .The use of multiple sensors and real-time monitoring enables the robot to make decisions and adjust its course quickly and accurately. The robot can detect and avoid obstacles, find the shortest path through the maze, and adapt to changes in the environment .Compared to the existing system that uses line-following technology, this approach is more robust and adaptable, and can handle complex maze configurations that are difficult to navigate using simple line-following techniques .Overall, the research work demonstrates the potential of using advanced sensors and real-time monitoring for maze-solving robots, and opens up new avenues for further research and development in this area.

3.2 PROPOSED SYSTEM

The LiDAR-based robot is a highly advanced autonomous robot that uses state-of-the-art technologies to navigate through unknown environments. With the Hector SLAM algorithm, the robot is capable of exploring its own path and moving from one place to another with great precision and accuracy.

Equipped with RP Lidar A1 M8, IMU BNO055, and Encoder DC motor, the robot can detect obstacles, measure distances, and track its own movements with exceptional accuracy. The Raspberry Pi 4 serves as the brain of the robot, receiving inputs from the load sensor cell and processing the data in real-time to guide the robot through the environment.

The use of Wi-Fi technology enhances the functionality of the robot, allowing it to transmit and receive data in real-time. By leveraging IoT technologies, the robot is able to communicate with other devices and sensors, enabling it to interact with its environment and adapt to changing conditions.

Overall, the LiDAR-based robot is a highly sophisticated device that represents the cutting edge of autonomous robotics. With its advanced sensors, powerful processing capabilities, and IoT integration, it has the potential to transform a wide range of industries, from manufacturing and logistics to healthcare and agriculture.

3.2.1 Applications

1. **Search and Rescue:**

In disaster scenarios, maze solving robots can navigate through hazardous environments to search for survivors. These robots can be equipped with sensors to detect heat signatures or other signs of life, making it easier to locate survivors. This can be especially useful in situations where it may be dangerous or impossible for humans to enter the area.

2. **Warehouse Management**

Maze solving robots can navigate through warehouses to locate and transport items. This can reduce the need for human intervention and improve the efficiency of the warehouse. In addition, these robots can operate 24/7, which can lead to faster and more efficient order fulfilment. They can also be programmed to optimize their routes, reducing the time it takes to transport items.

3. **Inspection**

Maze solving robots can be used for inspecting and monitoring critical infrastructure such as pipelines, tunnels, and bridges. These robots can navigate through tight spaces, take pictures or videos of the area, and report back to a central control room. This can help detect and prevent issues before they become major problems, increasing safety and reliability

4. Agriculture

Maze solving robots can be used in agriculture to navigate through fields, identify diseased plants, and treat them with the appropriate chemicals. These robots can also locate ripe fruit for harvesting, reducing the amount of time and labor required for manual harvesting.

5. Entertainment

Maze solving robots can be used as interactive exhibits in amusement parks or museums, providing an engaging experience for visitors. These robots can navigate through mazes, play games, or perform other tasks that provide entertainment and education.

6. Environmental Monitoring

Maze solving robots can be used for environmental monitoring, such as tracking the movement of wildlife, monitoring the water quality, and detecting the presence of pollutants. The robots can navigate through the natural terrain and collect data in real-time.

7. Mining

Maze solving robots can be used for mapping and exploring mining tunnels, identifying potential hazards, and collecting data on the geological structures. The robots can operate in hazardous and inaccessible environments, reducing the risk of human injury or exposure.

8. Medical

Maze solving robots can be used in medical applications, such as assisting in surgeries, delivering medicines, and performing diagnostics. The robots can navigate through the complex anatomy of the human body and reach the targeted areas with high precision.

3.3 COMPONENTS USED

3.3.1 HARDWARE COMPONENTS

- RP LiDAR A1 M8,
- IMU BNO055,
- ENCODER DC motor,
- Raspberry pi 4
- L298n motor driver
- Li-po Battery
- Arduino UNO

3.3.2 SOFTWARE COMPONENTS

- ROS
- GAZEBO
- Open CV
- Raspberry pi
- Python

3.3.1 Hardware Components

1) RP LiDAR A1 M8

Fig 3.3.1 RP LIDAR A1 M8

The RP LiDAR A1 M8 is a state-of-the-art laser range-finding sensor that is designed to enable accurate and reliable mapping and navigation in various robotics applications. It uses Time of Flight (ToF) technology to precisely detect and measure the distance to surrounding objects, creating a detailed 3D map of the environment in real-time.

One of the key features of the A1 M8 is its 360-degree scanning capability, which allows it to capture a complete view of the robot's surroundings. This is particularly useful in applications such as autonomous vehicles and drones, where a comprehensive view of the environment is critical for safe and efficient navigation.

The sensor also has a range of up to 12 meters, which makes it ideal for use in a wide range of environments, including both indoor and outdoor settings. Additionally, the A1 M8 is equipped with 8 channels of laser ranging, which enables it to capture a large amount of data quickly and accurately.

The A1 M8 is also lightweight and compact, which makes it easy to integrate into various robotics platforms. It supports multiple interfaces, including USB, UART, and CAN, which provides flexibility and ease of use in different system architectures.

Furthermore, the A1 M8 comes with an open-source SDK, which provides developers with the necessary tools and resources to customize and optimize the sensor for their specific applications. This enables developers to fine-tune the sensor's performance and ensure optimal performance in their specific use cases.

Overall, the RP LiDAR A1 M8 is a high-performance and versatile laser range-finding sensor that is well-suited for a wide range of robotics applications. Its advanced features, high accuracy, fast scanning speed, and flexible connectivity make it an ideal choice for autonomous navigation and mapping tasks in various industries, including logistics, agriculture, and construction.

2) IMU BNO055

Fig 3.3.2 IMU BNO055

The Inertial Measurement Unit (IMU) BNO055 is a state-of-the-art sensor designed for use in a wide range of robotics and navigation applications. This high-performance sensor integrates multiple sensors, including an accelerometer, gyroscope, and magnetometer, to provide accurate and reliable measurement of motion, orientation, and environmental conditions.

One of the key features of the BNO055 is its ability to provide precise orientation data in both static and dynamic conditions. The sensor's integrated accelerometer and gyroscope enable it to detect motion and measure the rate of change in orientation, while the magnetometer detects changes in magnetic field strength and direction. By combining data from these sensors, the BNO055 is able to provide accurate and reliable orientation data even in complex and dynamic environments.

Another important feature of the BNO055 is its low power consumption and small size, which makes it ideal for use in mobile robotics and other battery-powered applications. The sensor operates on a single 3.3V power supply and has a low power consumption of just 155uA, making it suitable for use in battery-powered applications where power efficiency is critical.

The BNO055 also features a built-in fusion algorithm that combines data from the accelerometer, gyroscope, and magnetometer to provide accurate and reliable motion tracking in various applications. The sensor's fusion algorithm uses sophisticated sensor fusion techniques to integrate data from multiple sensors and provide precise motion tracking even in challenging environments.

The sensor's compact size and low power consumption, combined with its advanced motion tracking capabilities, make it an ideal choice for a wide range of robotics and navigation applications. These include mobile robotics, drones, smart appliances, and other battery-powered devices.

Moreover, the BNO055 is easy to integrate into various robotics platforms, thanks to its support for multiple interfaces, including I2C, SPI, and UART. Additionally, the sensor comes with a comprehensive set of software tools and libraries that enable developers to quickly and easily integrate the BNO055 into their projects.

Overall, the IMU BNO055 is a versatile and high-performance sensor that is well-suited for a wide range of robotics and navigation applications. Its advanced features, low power consumption, and small size make it an ideal choice for use in mobile robotics and other battery-powered applications, while its sophisticated motion tracking capabilities enable precise and reliable motion tracking even in challenging environments

3) ENCODER DC MOTOR

Fig 3.3.3 ENCODER DC MOTOR

An encoder DC motor is a type of DC motor that is equipped with an encoder, a device that provides precise feedback on the motor's rotational position and speed. The encoder is typically a rotary incremental encoder that is attached to the motor shaft, and it generates a series of digital pulses that correspond to the motor's rotational movement.

The encoder DC motor is widely used in robotics and automation applications where precise control over motor speed and position is required. The encoder feedback enables the motor to be controlled with high precision, allowing it to be used in applications where accuracy and repeatability are essential.

The encoder DC motor operates by converting electrical energy into rotational energy through the interaction of magnetic fields. The motor consists of two main parts: the stator and the rotor. The stator is the stationary part of the motor and contains the windings that produce the magnetic fields. The rotor is the rotating part of the motor and is attached to the encoder shaft.

When electrical current is supplied to the motor, it creates a magnetic field that interacts with the magnetic field produced by the rotor. This interaction causes the rotor to rotate, generating torque and rotational movement. As the motor rotates, the encoder generates digital pulses that correspond to the rotational movement of the motor. This feedback is used to control the motor's speed and position accurately.

The encoder DC motor has several advantages over other types of motors. Its precise feedback enables it to be controlled with high accuracy, making it ideal for use in robotics and automation applications that require precise positioning and control. Additionally, the motor's brushless design makes it more reliable and efficient than traditional brushed DC motors.

In summary, an encoder DC motor is a type of DC motor equipped with an encoder that provides precise feedback on the motor's rotational position and speed. The motor is widely used in robotics and automation applications where precise control over motor speed and position is required. Its accurate feedback and efficient, reliable design make it an ideal choice for a wide range of applications.

4) Raspberry pi 4

Fig 3.3.4 RASPBERRY PI 4

The Raspberry Pi 4 is a single-board computer that was released in 2019. It is a powerful and versatile device that can be used for a wide range of applications, including home automation, media centers, robotics, and industrial automation.

The Raspberry Pi 4 is powered by a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC with a clock speed of 1.5GHz. It is available with either 1GB, 2GB, or 4GB of LPDDR4-3200 SDRAM, and it features a dual-band 802.11ac wireless networking, Bluetooth 5.0, Gigabit Ethernet, two USB 3.0 ports, two USB 2.0 ports, and a microSD card slot for storage.

The Raspberry Pi 4 also features two micro-HDMI ports capable of outputting 4K video at 60 frames per second, making it ideal for multimedia applications. Additionally, it has a 3.5mm audio jack and supports dual-display output.

The Raspberry Pi 4 runs on a variety of operating systems, including the official Raspberry Pi OS, Ubuntu, and other Linux distributions. It also supports a wide range of programming languages, including Python, C++, and Java.

One of the significant improvements in the Raspberry Pi 4 compared to previous models is its improved thermal management. The board features a dual-display heat sink and a cooling fan that helps to dissipate heat more efficiently, allowing it to run at higher clock speeds without overheating.

The Raspberry Pi 4 is an excellent platform for a wide range of applications. Its powerful processor, ample memory, and improved thermal management make it an ideal choice for demanding applications that require high performance and reliability. Its low cost and ease of use also make it an attractive option for hobbyists, students, and makers looking to experiment and learn about programming and electronics.

5) LIPO BATTERY

Fig 3.3.5. Lipo Battery

Lithium polymer batteries (LiPo) are a type of rechargeable battery that uses a polymer electrolyte instead of a liquid electrolyte. This makes them more lightweight and flexible than traditional lithium-ion batteries. LiPo batteries are also known for their high energy density, which means they can store more energy per unit volume than other types of batteries. This makes them ideal for use in applications where weight and space are limited, such as mobile devices, drones, and electric vehicles.

LiPo batteries are typically rated in terms of their voltage, capacity, and discharge rate. The voltage of a LiPo battery is typically between 3.0 and 4.2 volts. The capacity of a LiPo battery is typically measured in milliampere-hours (mAh). This indicates how much current the battery can deliver for one hour. The discharge rate of a LiPo battery is typically measured in amperes (A). This indicates how much current the battery can safely deliver.

LiPo batteries offer a number of advantages over traditional lithium-ion batteries. They are lighter, more flexible, and have a higher energy density. However, they also have some disadvantages. They can be more expensive than traditional lithium-ion batteries, and they can be more susceptible to damage from overcharging and overheating.

Overall, LiPo batteries are a versatile and powerful type of rechargeable battery. They are ideal for use in applications where weight and space are limited, and they offer a number of advantages over traditional lithium-ion batteries.

LiPo batteries are a popular choice for a variety of applications, including:

- Mobile devices, such as smartphones, tablets, and laptops
- Radio-controlled aircraft and other hobby equipment
- Electric vehicles, such as electric cars and scooters
- Power tools
- Flashlights
- Cameras
- Medical devices
- Industrial equipment

LiPo batteries are a safe and reliable choice for a variety of applications. However, it is important to follow the manufacturer's instructions when using LiPo batteries. Improper use can lead to fire or explosion.

Here are some safety tips for using LiPo batteries:

- Always use a LiPo battery charger that is specifically designed for LiPo batteries.
- Never charge a LiPo battery that is damaged or leaking.
- Do not expose LiPo batteries to extreme temperatures.
- Do not short-circuit a LiPo battery.
- Do not store a LiPo battery in a discharged state.
- Dispose of LiPo batteries properly.

6) L298N motor driver

Fig 3.3.6 L298N Driver

The L298N motor driver is a popular integrated circuit that is commonly used to control DC motors, stepper motors, and other types of motors in various applications. It is a dual H-bridge driver that can control the speed and direction of two DC motors simultaneously, or control the step and direction of a stepper motor.

The L298N consists of four power MOSFETs arranged in an H-bridge configuration, which allows for bidirectional control of the motor's rotation. It also has built-in flyback diodes to protect the circuit from voltage spikes generated by the motor when it is turned off.

The L298N requires an external power source to drive the motors, which can be between 7V and 35V DC. It also has separate inputs for controlling the speed and direction of each motor, with a logic voltage input range of 5V to 7V.

To use the L298N motor driver, the motor's positive and negative leads are connected to the corresponding outputs of the driver. The speed and direction of the motor are then controlled by sending the appropriate signals to the driver's input pins.

The L298N motor driver is commonly used in various applications, such as robotics, automation, and motor control projects. It is also popular in hobbyist projects, such as remote-controlled cars and drones.

Overall, the L298N motor driver is a versatile and reliable integrated circuit that provides a simple and effective way to control DC and stepper motors. Its bidirectional control and built-in flyback diodes make it a popular choice for a wide range of motor control applications.

7) Arduino UNO

Fig 3.3.7 Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P microcontroller. It is one of the most popular boards in the Arduino family and is widely used in various projects, such as robotics, home automation, and prototyping. The board is open-source and can be programmed using the Arduino Integrated Development Environment (IDE), which is available for free.

The ATmega328P microcontroller on the Arduino Uno board has 32KB of flash memory for storing the program code, 2KB of SRAM for storing variables, and 1KB of EEPROM for storing data that needs to be retained even after the board is powered off. It also has 14 digital input/output pins, 6 of which can be used as PWM outputs, and 6 analog input pins.

The board has a USB port for programming and serial communication with the computer. It can also be powered using an external power supply, such as a battery or a wall adapter.

The Arduino Uno board is compatible with various shields, which are add-on boards that extend its functionality. These shields can be used to add features such as Wi-Fi connectivity, Bluetooth communication, motor control, and more.
To use the Arduino Uno board, a program is written in the Arduino IDE and uploaded to the board using the USB cable. The program can read sensor data, control actuators, communicate with other devices, and perform various tasks based on the code written.

Arduino Uno is an open-source microcontroller board based on the ATmega328P microcontroller. It is one of the most popular boards in the Arduino family and is widely used for various DIY projects and prototyping. The board has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and an ICSP header for programming.

The digital pins can be used to interface with various electronic components such as LEDs, motors, and sensors, while the analog pins can be used to read signals from analog sensors such as temperature sensors and light sensors. The PWM pins can be used to control the brightness of LEDs and the speed of motors.

The board can be programmed using the Arduino IDE, which is a free software development environment based on the Processing programming language. The Arduino programming language is based on C/C++, and the Arduino IDE provides a simplified interface for writing and uploading code to the board.

Overall, the Arduino Uno board is a versatile and user-friendly microcontroller board that is well-suited for beginners and advanced users alike. Its open-source nature, compatibility with various shields, and large community of users and developers make it a popular choice for a wide range of projects.

3.3.2 SOFTWARE COMPONENTS

1) ROS

ROS (Robot Operating System) is a flexible and open-source framework for developing software applications for robots. It provides a collection of libraries and tools that allow developers to create complex robotic systems, from simple mobile robots to humanoid robots and autonomous vehicles.

ROS was originally developed at Stanford University in 2007 and has since been widely adopted in the robotics community. It is supported by a large community of developers, researchers, and companies, who contribute to its development and use it in a variety of applications.

ROS provides a distributed computing architecture that allows different parts of a robot system to run on different computers, communicating with each other over a network. This allows for a modular approach to software development, where different components can be developed independently and then integrated into a larger system.

ROS provides a set of tools for building, testing, and deploying robot applications. These tools include a package manager for managing dependencies, a build system for compiling code, a runtime environment for executing code, and a set of visualization and debugging tools.

ROS supports a wide range of programming languages, including C++, Python, and Java, which makes it accessible to a wide range of developers. It also provides a large collection of open-source libraries and packages for common robot tasks, such as mapping, localization, and navigation.

ROS has been used in a variety of applications, including mobile robotics, industrial automation, healthcare, and space exploration. Its flexibility and modularity make it well-suited for developing complex robot systems that require advanced sensing, perception, and decision-making capabilities.

Overall, ROS is a powerful and flexible framework for developing software applications for robots. Its open-source nature, large community, and wide range of tools and libraries make it a popular choice for roboticists and developers around the world.

2) GAZEBO

Gazebo is a 3D simulation environment for robotics and autonomous systems. It is an open-source project that provides a physics engine, a rendering engine, and a set of tools for creating realistic simulations of robots and their environments.

Gazebo is widely used in the robotics community for developing and testing robot software in a safe and controlled environment. It can simulate a wide range of robot systems, including wheeled robots, aerial robots, manipulators, and humanoid robots.

Gazebo provides a set of physics engines that allow developers to simulate the physical behaviour of objects in the environment. These engines support a range of physics models, including rigid body dynamics, articulated systems, and fluid dynamics. This allows developers to create realistic simulations of robot behaviour, including collisions, dynamics, and sensor readings.

Gazebo also provides a rendering engine that allows developers to create realistic 3D visualizations of their simulations. This includes support for textures, lighting, and shadows, which can help make simulations more immersive and realistic.

In addition to its physics and rendering engines, Gazebo provides a set of tools for creating and customizing simulations. This includes support for creating complex environments, importing 3D models, and defining robot models using URDF (Unified Robot Description Format).

Gazebo is integrated with the Robot Operating System (ROS), which allows developers to easily integrate their simulations with real-world robot systems. This includes support for sending sensor data to the simulation environment, and controlling the simulated robot using ROS messages.

Overall, Gazebo is a powerful and flexible simulation environment for robotics and autonomous systems. Its support for physics simulation, rendering, and customization, as well as its integration with ROS, make it a popular choice for robotics developers who want to test and refine their software in a safe and controlled environment.

Google Gazebo is a 3D simulation environment for robotics research and development. It is based on the open-source Gazebo simulator, and it provides a number of features that make it well-suited for robotics applications, such as: Support for a wide range of robot models Realistic physics simulation Integration with the Robot Operating System (ROS) A large library of plugins and extensions Google Gazebo is used by a wide range of robotics researchers and developers, including those working on autonomous vehicles, drones, and robots for manufacturing and logistics. It is a powerful tool that can be used to test and develop new robotics algorithms and applications. Here are some of the benefits of using Google Gazebo: It is a free and open-source software. It is highly customizable and extensible. It is easy to use and learn. It is supported by a large community of users and developers. Google Gazebo is a valuable tool for robotics research and development. It can be used to test and develop new robotics algorithms and applications in a safe and controlled environment.

Applications to do with Google Gazebo

- Test and develop new robotics algorithms and applications. Google Gazebo can be used to test and develop new robotics algorithms and applications in a safe and controlled environment. This is important because it allows you to experiment with new ideas without the risk of damaging your robot or harming yourself.
- Create realistic simulations of real-world environments. Google Gazebo can be used to create realistic simulations of real-world environments. This is useful for testing and developing new robotics algorithms and applications in environments that are difficult or impossible to recreate in the real world.
- Visualize and analyse robot motion. Google Gazebo can be used to visualize and analyze robot motion. This is useful for understanding how your robot moves and for identifying any potential problems with its motion.
- Train and evaluate robot controllers. Google Gazebo can be used to train and evaluate robot controllers. This is useful for ensuring that your robot can control its motion in a safe and predictable manner.

- Collaborate with other robotics researchers and developers. Google Gazebo is a popular tool among robotics researchers and developers. This means that you can collaborate with others to share ideas, learn from each other, and work together to solve problems.

3) Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed by the Raspberry Pi Foundation. The boards are designed to be low-cost and highly versatile, making them popular for a wide range of applications, including education, prototyping, and hobby projects.

The Raspberry Pi boards feature a variety of input/output interfaces, including USB, HDMI, Ethernet, and GPIO (General Purpose Input/Output). This allows them to be connected to a wide range of devices, such as displays, keyboards, cameras, sensors, and actuators.

One of the key features of the Raspberry Pi is its ability to run a variety of operating systems, including Linux-based distributions such as Raspbian and Ubuntu. This makes it easy to develop and run software on the boards, as well as to connect them to other systems and networks.

The Raspberry Pi has become popular in the field of robotics and automation, where it is used as a platform for controlling robots, collecting and processing sensor data, and communicating with other systems. Its small size and low power consumption make it well-suited for mobile and remote applications, such as drones and autonomous vehicles.

The Raspberry Pi also has a large and active community of developers and enthusiasts, who create and share a wide range of projects and resources. This includes tutorials, code libraries, and add-on boards, such as motor controllers and sensor interfaces.

Overall, the Raspberry Pi is a powerful and flexible platform for a wide range of applications, including robotics, automation, and education. Its low cost, versatility, and community support make it an ideal choice for hobbyists, students, and professionals alike.

4) PYTHON

Python is a high-level programming language that is easy to use and read. It is often described as "executable pseudo-code" because it resembles human language more than other programming languages. This makes it easier for programmers to write and understand code, even if they are not experts in the language.

Python is also a highly versatile language with a large and growing number of libraries and modules that extend its capabilities. These include libraries for web development, scientific computing, data analysis, machine learning, and more. The Python ecosystem is known for its ease of use, and many developers appreciate the large and supportive community that has grown around the language.

Another key advantage of Python is its portability. Python code can run on a wide range of platforms and operating systems, including Windows, macOS, Linux, and many more. This makes it easier to develop and deploy applications across multiple devices and environments.

Python is also known for its support for object-oriented programming (OOP) and functional programming (FP) paradigms. This makes it possible to write code that is organized, modular, and reusable, which can save time and improve the quality of code.

Overall, Python is a popular and powerful programming language that is well-suited for a wide range of applications. Its simplicity, readability, versatility, and large ecosystem make it a top choice for developers around the world.

5) Odometry

Odometry is a commonly used technique for estimating the position and orientation of a robot in its environment. The method involves measuring changes in position and orientation by integrating velocity data over time. This can be done using sensors such as encoders on wheels, gyroscopes, and accelerometers. However, the accuracy of the odometry estimates can be affected by various factors such as sensor noise, wheel slippage, and errors in sensor calibration.

To overcome these challenges and improve the accuracy of odometry, advanced techniques such as sensor fusion, mapping, and localization can be used. For example, sensor fusion involves combining data from multiple sensors, such as a camera and lidar, to obtain a more accurate estimate of position and orientation. Mapping involves constructing a map of the environment to aid in localization and path planning, while localization involves determining the robot's position and orientation within a map.

In addition to these techniques, machine learning and computer vision algorithms can also be used to improve odometry accuracy. These algorithms can help to identify and correct errors in sensor data, and can also be used to predict the robot's future position and orientation based on past data.

Overall, accurate and reliable odometry is essential for successful navigation and operation of robots in complex environments.

6) UBUNTU SERVER

Ubuntu Server 20.04 is a long-term support (LTS) release of the Ubuntu operating system, specifically designed for servers and cloud environments. It was released on April 23, 2020, and is supported until April 2025.

Some of the key features of Ubuntu Server 20.04 include:

1. Improved security: Ubuntu Server 20.04 includes several security enhancements, including support for UEFI Secure Boot, AppArmor profiles, and kernel signing.
2. New package versions: The release includes updated versions of popular server software, including Python 3.8, OpenJDK 11, and PostgreSQL 12.
3. ZFS support: Ubuntu Server 20.04 includes support for the ZFS file system, which provides advanced features such as data compression, snapshots, and data deduplication.
4. Enhanced container support: The release includes updates to Docker and Kubernetes, making it easier to run and manage containerized applications.

7) Ubuntu Desktop

Ubuntu Desktop is a user-friendly version of the Ubuntu operating system, designed for personal computers and laptops. It is based on the Debian Linux distribution and is known for its ease of use, stability, and security.

Some of the key features of Ubuntu Desktop include:

- User-friendly interface: Ubuntu Desktop features a modern and intuitive interface, with a dock for easy access to frequently used applications and a search bar for quickly finding files and applications.
- Software Center: The Ubuntu Software Center allows users to easily browse, install, and manage a wide range of software applications, including popular productivity tools, media players, and games.
- Customizable: Ubuntu Desktop is highly customizable, allowing users to easily change the appearance of the desktop, install themes and icons, and add or remove applications as needed.

8) ROS (Robot Operating System) Noetic

ROS (Robot Operating System) Noetic is the 12th and latest long-term support (LTS) release of the ROS software framework for robotics development. It was released on May 23, 2020, and is supported until May 2025.

Some of the key features of ROS Noetic include:

1. Python 3 support: ROS Noetic supports Python 3, which is the latest version of the popular programming language. This allows developers to take advantage of the latest Python features and libraries.
2. Improved performance: ROS Noetic includes several performance improvements, such as faster startup times and reduced CPU and memory usage.
3. New and updated packages: The release includes several new packages, such as the Robot Model Visualization (rviz) package, and updates to existing packages, such as the Navigation Stack and the Robot Localization package

9) ROS BASH

ROS (Robot Operating System) uses a customized version of the Bash shell, which includes several custom commands and environment variables specific to ROS.

Some of the custom Bash commands used in ROS include:

1. roscd: This command allows you to change the current directory to the package or stack directory for a specified ROS package or stack.
2. roslaunch: This command allows you to launch one or more ROS nodes, along with any required parameters and configuration files.
3. rosrun: This command allows you to run a specific ROS node by specifying its package and executable name.

10) ROS Serial

ROS Serial is a package in the ROS software framework that enables communication between a ROS node and a microcontroller over a serial connection. It is commonly used in robotics applications where a microcontroller is used to control hardware such as motors, sensors, and actuators.

ROS Serial provides a lightweight protocol for transmitting ROS messages over a serial connection, allowing a ROS node to communicate with a microcontroller in a platform-independent way. This makes it easier to integrate microcontrollers into ROS-based robotics systems.

Some of the key features of ROS Serial include:

1. Lightweight protocol: The ROS Serial protocol is designed to be lightweight and easy to implement on a microcontroller.
2. Platform-independent: ROS Serial can be used with any microcontroller that has a serial interface, regardless of the underlying hardware or operating system.
3. Efficient message serialization: ROS Serial uses an efficient message serialization format to minimize the amount of data that needs to be transmitted over the serial connection.

4. Support for multiple transports: ROS Serial supports several transport protocols, including USB, Bluetooth, and Wi-Fi, making it easy to connect to a microcontroller using a variety of interfaces.

11) ROS CORE

In ROS (Robot Operating System), the ROS core is the main component responsible for coordinating communication between ROS nodes. It provides several key services that enable nodes to communicate with each other and share data.

The ROS core consists of several components, including the ROS Master, Parameter Server, and Logging System.

1. ROS Master: The ROS Master is responsible for coordinating communication between ROS nodes. It provides a registry of active ROS nodes, and manages the connections between them.
2. Parameter Server: The Parameter Server is a key-value store that enables nodes to share configuration parameters with each other. It provides a centralized location for storing and retrieving parameter values, and allows nodes to dynamically update parameters at runtime.
3. Logging System: The Logging System provides a mechanism for nodes to log messages and errors. It includes several built-in log levels, ranging from debug to fatal, and allows users to configure the level of detail that is logged

12). Secure Shell

Secure Shell (SSH) is a network protocol that provides a secure way to access a computer over an unsecured network. SSH encrypts all data that is transmitted between the client and server, which makes it a much more secure option than other protocols, such as telnet or FTP.

SSH can be used for a variety of tasks, including:

1. Remote login: SSH can be used to log in to a remote computer and execute commands. This can be useful for system administration tasks, such as managing files or installing software.
2. File transfer: SSH can be used to transfer files between two computers. This is a more secure option than using a file sharing service, such as Dropbox or Google Drive.
3. Port forwarding: SSH can be used to forward ports from one computer to another. This can be useful for setting up a web server or VPN.
4. Tunneling: SSH can be used to create a secure tunnel between two computers. This can be useful for accessing resources on a remote network, such as a file server or database.
5. SFTP: SSH can be used to transfer files using the Secure File Transfer Protocol (SFTP). SFTP is a secure alternative to FTP that uses SSH encryption.

SSH is a powerful tool that can be used to securely access and manage computers over a network. It is a valuable tool for system administrators and anyone who needs to transfer files securely.

Here are some additional details about SSH:

1. Client-server architecture: SSH is a client-server protocol. This means that there is a client program that runs on the user's computer and a server program that runs on the remote computer.
2. Encryption: SSH uses encryption to protect the data that is transmitted between the client and server. This encryption is based on public key cryptography.
3. Security features: SSH has a number of features that make it a secure protocol. These features include:
    1. Strong authentication: SSH supports a variety of authentication methods, including passwords, public keys, and certificates.
    2. Data encryption: SSH encrypts all data that is transmitted between the client and server.
    3. Integrity protection: SSH protects the data from being modified in transit.
    4. Denial of service protection: SSH protects against denial of service attacks.

SSH is a valuable tool for anyone who needs to securely access and manage computers over a network. It is a secure and reliable protocol that can be used for a variety of tasks.

Here are some examples of how SSH can be used:

1. A system administrator can use SSH to log in to a remote server and perform system maintenance tasks. This can include tasks such as managing files, installing software, and configuring services.
2. A developer can use SSH to transfer files between their local computer and a remote server. This can be useful for transferring code, data, and other files between development environments.
3. A user can use SSH to access a web server that is running on a remote computer. This can be useful for accessing websites, web applications, and other resources that are hosted on a remote server.
4. A user can use SSH to connect to a remote computer and run a graphical application. This can be useful for accessing applications that are not available on the user's local computer.

SSH is a powerful tool that can be used for a variety of tasks. It is a secure and reliable protocol that is essential for anyone who needs to access or manage computers over a network.

13) TELEOP TWIST

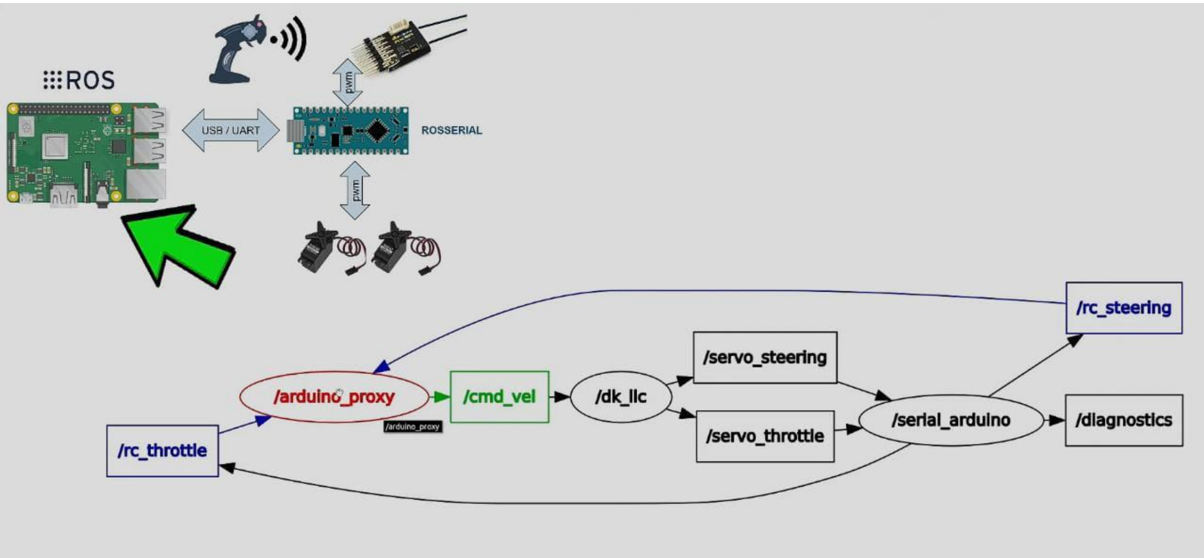

Fig 3.3.12. Teleop Twist

Teleop Twist is a ROS (Robot Operating System) package that provides a simple teleoperation interface for controlling a robot using a keyboard or joystick. It is commonly used in robotics applications where a human operator needs to manually control the movement of a robot.

The Teleop Twist package works by subscribing to the topic, which is the standard ROS topic for sending velocity commands to a robot. It then listens for keyboard or joystick input, and translates that input into velocity commands that are sent to the robot.

The twist component of the package refers to the linear and angular velocity commands that are sent to the robot. The package allows the operator to control the linear and angular velocity of the robot separately, enabling precise control of the robot's movement.

The Teleop Twist package can be easily customized to work with different types of input devices and robot platforms. It also includes several useful features, such as configurable velocity scales, Deadman switches to ensure safe operation, and support for multiple input devices.

Overall, Teleop Twist is a useful tool for teleoperating robots in a wide range of applications, and is a popular choice for robotics developers who need a simple and reliable teleoperation interface.

12) CATKIN WORKSPACE

Catkin Workspace is a build system used in ROS (Robot Operating System) for organizing, building, and managing ROS packages. It is the recommended build system for ROS since ROS Indigo and later versions.

Catkin Workspace is a directory hierarchy containing several folders, including:

1. **src**: This folder contains the source code for the ROS packages that are being developed.

2. **devel**: This folder is where the compiled packages are stored during development. The compiled packages are not installed system-wide, but are only available within the workspace.
3. **build**: This folder contains the build files generated during the compilation process.
4. **install**: This folder is where the compiled packages are installed when they are ready to be used system-wide

13) RP LIDAR PACKAGES

The RP Lidar package is a ROS (Robot Operating System) package that provides a driver for the RPLIDAR line of low-cost 2D laser scanners. These scanners are commonly used in robotics applications to provide real-time 2D range data for mapping, localization, and obstacle avoidance.

The RP Lidar package works by communicating with the RPLIDAR sensor over a serial port connection, and providing ROS nodes with 2D laser scan data. The package includes a **relearned** node that publishes the laser scan data on the **/scan** topic, as well as a **rplida No de Client** node that can be used to visualize the scan data in RViz.

The RP Lidar package also includes several other nodes and utilities, including a **rplidar Node ClientQt** node that provides a graphical user interface for visualizing the laser scan data, and a **rp lidar Node Service** node that provides a ROS service for starting and stopping the laser scanner.

In addition to the ROS driver, the RP Lidar package also includes a firmware update utility for updating the firmware on RPLIDAR sensors, as well as a calibration utility for calibrating the sensors.

Overall, the RP Lidar package is a useful tool for integrating RPLIDAR sensors into ROS-based robotics applications. Its driver provides easy access to 2D laser scan data, and its utilities make it easy to configure and calibrate the sensors for optimal performance.

14. RVIZ APPLIACTION

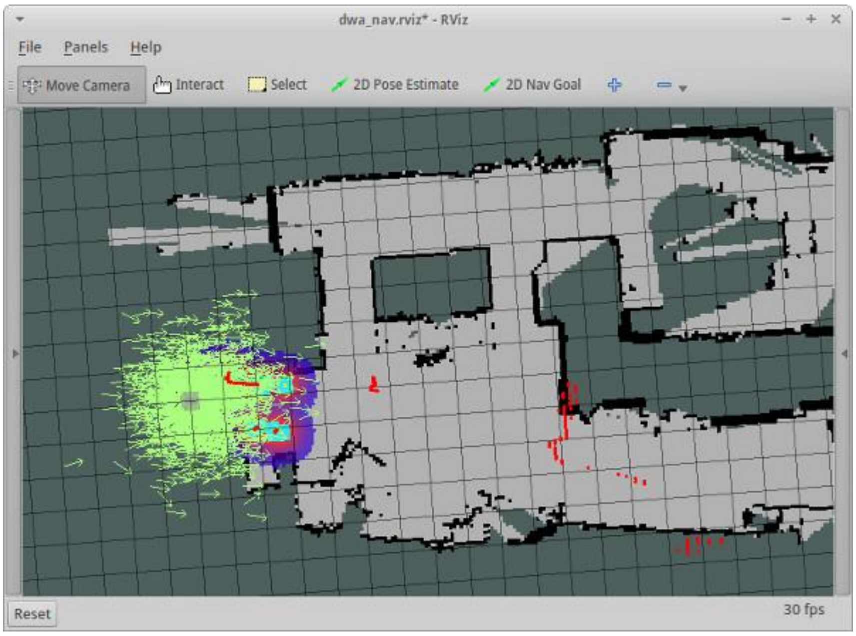

Fig 3.14 RVIZ

RVIZ is a 3D visualization tool for robots using ROS. It is a powerful tool that can be used to visualize a wide range of data, including robot models, maps, and sensor data. RVIZ is easy to use and can be customized to meet the needs of the user.

Visualizing robot models. RVIZ can be used to visualize robot models in 3D. This is useful for understanding the robot's geometry and for planning robot motions. To do this, you will need to create a robot model in a 3D modelling software, such as Blender or SolidWorks. Once you have created the model, you can export it in a format that RVIZ can understand, such as COLLADA or URDF. Then, you can load the model into RVIZ by clicking on the "Add" button and selecting "Robot Model."

Visualizing maps. RVIZ can be used to visualize maps in 3D. This is useful for understanding the robot's environment and for planning robot paths. To do this, you will need to create a map in a 2D mapping software, such as Cartographer or Gapping. Once you have created the map, you can export it in a format that RVIZ can understand, such as YAML or PLY. Then, you can load the map into RVIZ by clicking on the "Add" button and selecting "Map."

Visualizing sensor data. RVIZ can be used to visualize sensor data, such as camera images, laser scans, and odometry data. This is useful for understanding the robot's surroundings and for debugging robot sensors. To do this, you will need to connect the robot's sensors to your computer. Once the sensors are connected, you can start visualizing their data in RVIZ by clicking on the "Add" button and selecting the appropriate plugin. For example, to visualize camera images, you would select the "Image" plugin.

Customizing the appearance of RVIZ. RVIZ can be customized to meet the needs of the user. This includes changing the colours, fonts, and layouts of RVIZ. To customize RVIZ, you can use the "View" menu. From the "View" menu, you can select "Appearance" to change the colours and fonts of RVIZ. You can also select "Layout" to change the layout of RVIZ.

Creating and saving your own configurations. RVIZ configurations can be created and saved. This allows users to save their favourite settings and configurations for future use. To create a new configuration, click on the "File" menu and select "Save As." Then, enter a name for your configuration and click on "Save." To load a saved configuration, click on the "File" menu and select "Open." Then, select the configuration that you want to load and click on "Open."

Collaborating with other users. RVIZ can be used to collaborate with other users. This is useful for sharing data and for working together on projects. To collaborate with other users, you can use the "Share" menu. From the "Share" menu, you can select "Export" to export your RVIZ configuration to a file. You can then share this file with other users. You can also select "Import" to import an RVIZ configuration that has been exported by another user.

3.4 BLOCK DIAGRAM

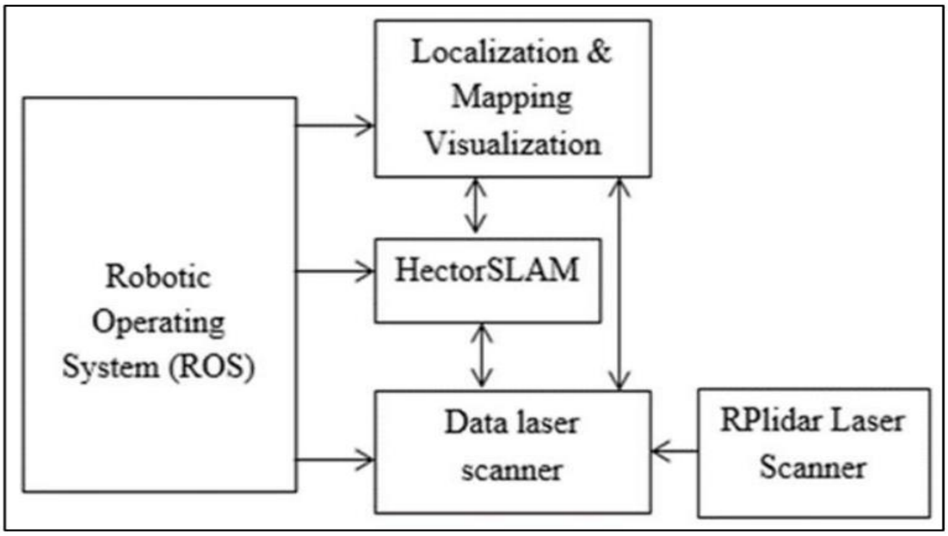

Fig 3.4 Block Diagram

A block diagram of a maze solving robot using RP Lidar, Odometry, and SLAM typically consists of the following components:

1. RP Lidar: The RP Lidar sensor is mounted on the robot to detect obstacles in the maze. It sends out laser beams and measures the time it takes for the beams to bounce back, creating a 2D map of the surrounding environment.
2. Odometry: Odometry is a method for measuring the robot's movement and determining its position and orientation. Encoders on the robot's wheels or an IMU (Inertial Measurement Unit) can be used to track the robot's movements and calculate its position.
3. Microcontroller: A microcontroller, such as a Raspberry Pi or Arduino, is used to control the robot's movements and process data from the sensors. It receives sensor data and uses it to make decisions about the robot's movements.
4. Motor Drivers: Motor drivers are used to control the robot's movement. They take signals from the microcontroller and convert them into signals that the motors can understand.
5. Battery: A battery provides power to the robot and its components.
6. SLAM Algorithm: The SLAM algorithm processes data from the RP Lidar and odometry to create a map of the maze and localize the robot within it. The algorithm

uses probabilistic techniques to estimate the robot's position and update the map as the robot moves through the maze.

7. Path Planning Algorithm: The path planning algorithm determines the optimal path for the robot to navigate through the maze based on the map generated by the SLAM algorithm. It uses the map to avoid obstacles and dead-ends and reach the destination.
8. Robot Actuators: Robot actuators, such as motors, servos, or grippers, are used to interact with the environment or complete specific tasks in the maze.

Overall, a maze solving robot using RP Lidar, Odometry, and SLAM combines sensors, algorithms, and hardware components to create a robust and autonomous system that can navigate through an unknown environment and solve complex problems.

3.5 Localisation and Mapping Visualisation

Localisation and mapping visualisation is the process of representing the environment and the position of a robot within that environment in a visual form. This is typically achieved through the use of maps and visualisations of sensor data.

In simultaneous localisation and mapping (SLAM), a robot uses its sensors to gather data about its surroundings and build a map of the environment while simultaneously determining its own position within that environment. The resulting map can be visualised in a variety of ways, including 2D and 3D maps, point clouds, and occupancy grids.

Visualisations of SLAM data can provide valuable insights into the performance and accuracy of a robot's localisation and mapping algorithms. They can also be used to identify areas of the environment that may be difficult for the robot to navigate, such as narrow passages or areas with poor sensor coverage.

In addition to SLAM, localisation and mapping visualisation can be used in a variety of robotics applications, including autonomous vehicles, surveillance, and search and rescue. For example, in autonomous vehicles, visualisations of sensor data and maps can be used to provide real-time feedback to the vehicle's control system, enabling it to make decisions about steering, acceleration, and braking.

Overall, localisation and mapping visualisation is an important tool for understanding and improving the performance of robots in a variety of applications. As robotics technology continues to advance, we can expect to see even more sophisticated and informative visualisations of SLAM data and other sensor information

3.6 HECTOR SLAM

HECTOR SLAM (Heterogeneous Robot-based Cooperative Localization and Mapping) is a robust and efficient algorithm for simultaneous localization and mapping (SLAM) that is designed to work in a variety of environments, including indoor and outdoor settings. Developed by the Technical University of Munich, HECTOR SLAM is an open-source algorithm that is widely used in robotics and computer vision applications.

HECTOR SLAM works by using laser range finder measurements to create a map of the environment and estimate the robot's position in real-time. It uses a probabilistic approach to deal with noise and uncertainties in the sensor measurements and the robot's motion. The algorithm works by continuously updating the map and the robot's position estimate based on new sensor measurements and odometry information.

One of the key advantages of HECTOR SLAM is its ability to work with low-cost and lightweight sensors, making it ideal for use in small and medium-sized robots. It also has low computational requirements, making it suitable for real-time applications. The algorithm is able to handle large environments and can generate accurate maps even in dynamic environments with moving obstacles.

HECTOR SLAM has been used in a variety of robotic applications, including autonomous navigation, inspection, and search and rescue. Its open-source nature has also enabled researchers and developers to modify and adapt the algorithm for their specific needs, making it a versatile tool for SLAM research and development.

In summary, HECTOR SLAM is a powerful SLAM algorithm that is widely used in robotics and computer vision applications. Its ability to work with low-cost sensors and handle large, dynamic environments make it a valuable tool for robotics research and development.

3.7 RP LIDAR DATA SCANNER

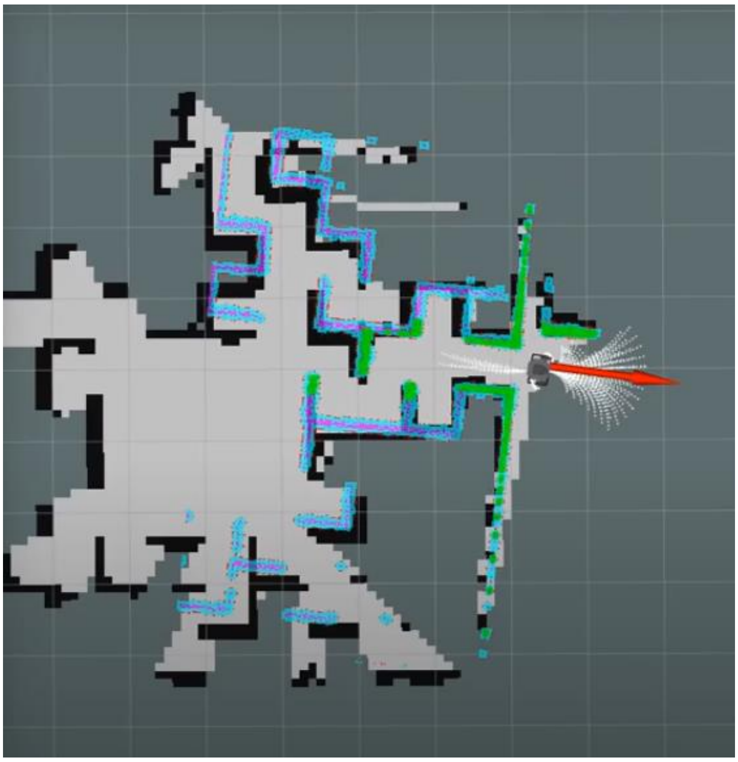

Fig 3.7 Rp Lidar Scanner

RP Lidar (Light Detection and Ranging) is a sensor that uses laser light to scan and measure distances to objects in the environment. The sensor emits a laser beam that reflects off objects in its path, and the time it takes for the beam to return to the sensor is used to calculate the distance to the object. This process is repeated multiple times per second, resulting in a 2D or 3D map of the environment.

The RP Lidar data scanner is an advanced technology that uses the RP Lidar sensor to scan and collect data about the environment. The data collected by the RP Lidar sensor is typically represented in the form of a point cloud, which is a set of 3D points that represent the objects in the environment. These points are often visualized in real-time using software tools such as ROS (Robot Operating System).

The RP Lidar data scanner is used in various applications, including robotics, autonomous vehicles, and mapping. In robotics, the data collected by the RP Lidar sensor can be used to help the robot navigate through the environment, avoid obstacles, and perform tasks. In autonomous vehicles, the RP Lidar data scanner is used to create high-resolution maps of the surrounding environment, which are then used by the vehicle's navigation system to determine the best path to take.

In mapping, the RP Lidar data scanner is used to create accurate maps of the environment, which can be used for a wide range of applications, including urban planning, construction, and environmental monitoring. The RP Lidar data scanner is particularly useful in areas where traditional mapping methods, such as satellite imagery, are not suitable, such as in densely populated urban areas.

Overall, the RP Lidar data scanner is an advanced technology that has the potential to transform various industries and applications. As research in this field continues to advance, we can expect to see even more exciting developments in the years to come.

3.8 MAZE

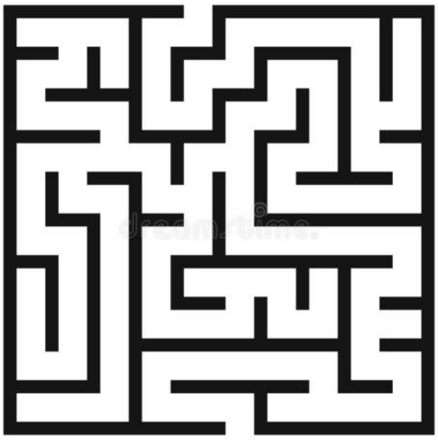

Fig 3.8 Maze

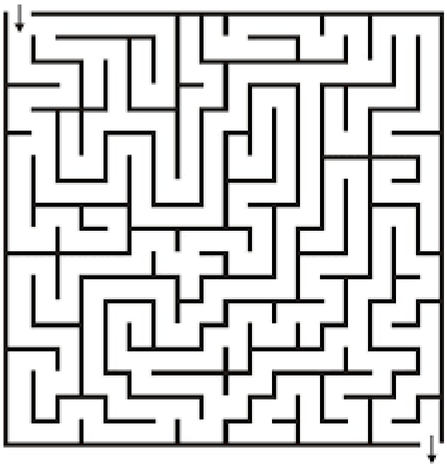

Fig3.8.1 Mazes

A maze is a complex structure of passages and dead ends, typically with a single path leading from the entrance to the exit. Mazes have been used for centuries as a form of entertainment and challenge, and they have also been used as a metaphor for life. There are many different types of mazes, including linear, branching, revolving, mirror, water, and haunted mazes.

Linear mazes are the simplest type of maze. They have a single path that leads from the entrance to the exit. Branching mazes have multiple paths that lead from the entrance to the exit. Revolving mazes have walls that rotate, making it difficult to find the exit. Mirror mazes have walls that are mirrored, making it difficult to tell which way is forward. Water mazes are filled with water, making it difficult to walk through them. Haunted mazes are decorated with scary objects and are designed to scare people.

Mazes have been used for centuries as a form of entertainment and challenge. People enjoy the challenge of trying to find their way through a maze, and they also enjoy the sense of accomplishment when they finally reach the exit. Mazes have also been used as a metaphor for life. The winding paths of a maze can represent the challenges and obstacles that we face in life, and the exit can represent the goal that we are striving to achieve.

In today's world, mazes are also being used in a variety of technical applications. One of the most important applications of mazes is in the field of robotics. Mazes are used to train robots to navigate complex environments. By training robots to solve mazes, they can learn how to avoid obstacles, find their way around, and make decisions. Mazes are also being used in the field of artificial intelligence. AI researchers are using mazes to develop new algorithms for solving problems. By studying how humans solve mazes, AI researchers can develop algorithms that can solve problems more efficiently and effectively. Mazes are also being used in the field of education. Mazes are used to teach children about different concepts, such as geometry, problem-solving, and spatial reasoning. By solving mazes, children can learn how to think logically and how to solve problems.

Mazes are a fascinating and challenging puzzle that have been enjoyed by people for centuries. They are a great way to test your problem-solving skills and to have some fun. In today's world, mazes are also being used in a variety of technical applications, such as robotics, artificial intelligence, and education.

Here are some specific examples of how mazes are being used in technical applications:

Robotics: Mazes are used to train robots to navigate complex environments. By training robots to solve mazes, they can learn how to avoid obstacles, find their way around, and make decisions. For example, Google has developed a robot that can solve mazes using its own sensors and artificial intelligence.

Artificial intelligence: Mazes are being used in the field of artificial intelligence to develop new algorithms for solving problems. By studying how humans solve mazes, AI researchers can develop algorithms that can solve problems more efficiently and effectively. For example, researchers at the University of California, Berkeley have developed an AI algorithm that can solve mazes faster than any human.

Education: Mazes are used in the field of education to teach children about different concepts, such as geometry, problem-solving, and spatial reasoning. By solving mazes, children can learn how to think logically and how to solve problems. For example, the Children's Museum of Indianapolis has a maze that teaches children about geometry and problem-solving.

Mazes are a versatile and powerful tool that can be used in a variety of applications. They are a great way to test your problem-solving skills, have some fun, and learn about different concepts.

3.9 FLOW CHART

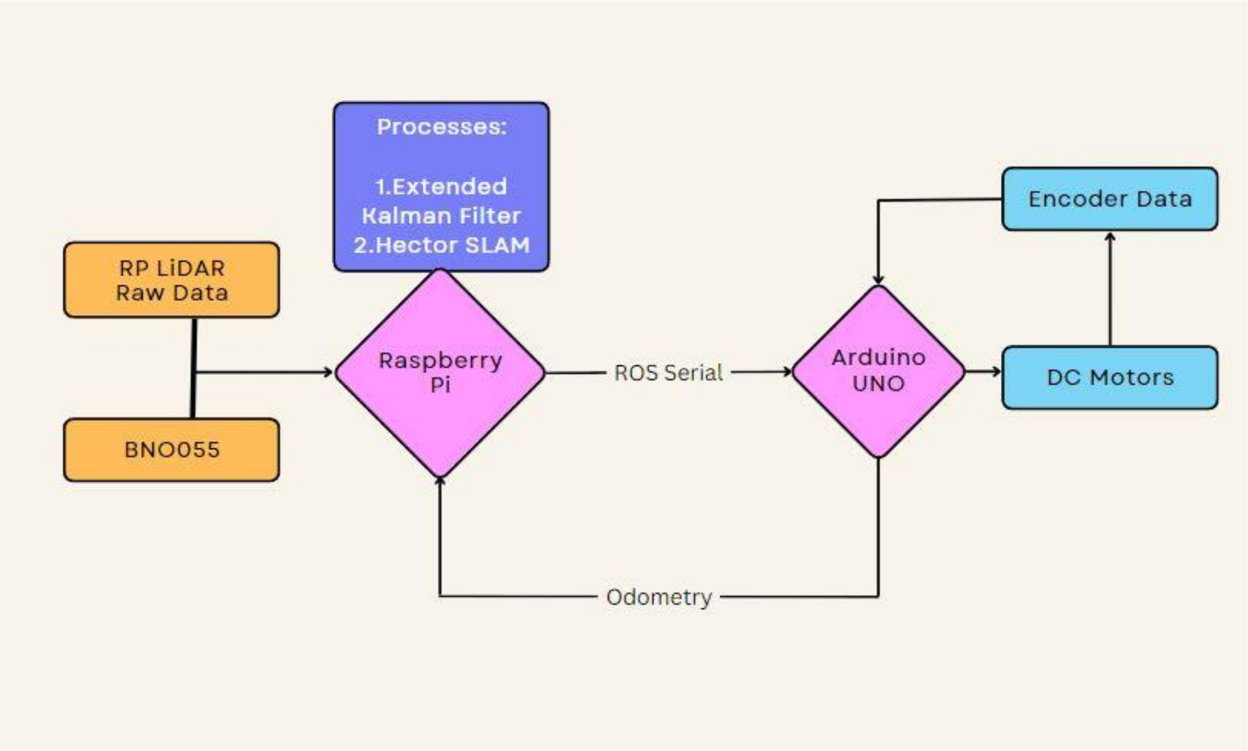

Fig 3.9 Flowchart

Flowchart for a Maze Solving Robot

Input:

1. RP LIDAR raw data
2. BNO055 data
3. Encoder data
4. Processes:
5. Extended Kalman Filter
6. Hector SLAM
7. Output:
8. Odometry
9. Map of the maze

Steps:

1. The RP LIDAR raw data is used to create a point cloud of the environment.
2. The BNO055 data is used to determine the robot's orientation.
3. The encoder data is used to determine the robot's position.
4. The Extended Kalman Filter is used to fuse the data from the RP LIDAR, BNO055, and encoders to create an accurate estimate of the robot's pose.
5. The Hector SLAM algorithm is used to build a map of the environment from the robot's pose estimates.
6. The odometry is used to track the robot's movement through the maze.
7. The map of the maze is used to plan the robot's path to the exit.

Output:

1. The robot will be able to navigate through the maze and reach the exit.

2. Here are some additional details about the components and processes used in this flowchart:

3. RP LIDAR: The RP LIDAR is a laser scanner that can be used to create a point cloud of the environment. The point cloud is a collection of points that represent the distance to objects in the environment. The RP LIDAR can be used to create a detailed map of the environment, which can be used by the robot to navigate.

4. BNO055: The BNO055 is an IMU (Inertial Measurement Unit) that can be used to determine the robot's orientation. The IMU measures the robot's acceleration and rotation, which can be used to calculate the robot's orientation. The BNO055 can be used to keep the robot's map aligned with the real world, which is important for accurate navigation.

5. Encoders: Encoders are devices that can be used to measure the rotation of a shaft. The encoders on the robot's wheels can be used to measure the robot's position. The encoders can be used to update the robot's map as it moves through the environment.

6. Extended Kalman Filter: The Extended Kalman Filter is a recursive filter that can be used to fuse data from multiple sensors to create an accurate estimate of a state. In this case, the state is the robot's pose. The Extended Kalman Filter uses the data from the RP LIDAR, BNO055, and encoders to create an accurate estimate of the robot's pose.

7. Hector SLAM: Hector SLAM is a Simultaneous Localization and Mapping algorithm that can be used to build a map of the environment while simultaneously tracking the robot's pose. Hector SLAM uses the data from the RP LIDAR and encoders to build a map of the environment. Hector SLAM also uses the data from the BNO055 to track the robot's pose.

8. Odometry: Odometry is the process of measuring the robot's movement through the environment. Odometry can be used to track the robot's position and orientation. The odometry can be used to update the robot's map as it moves through the environment.

3.10CIRCUIT DIAGRAM

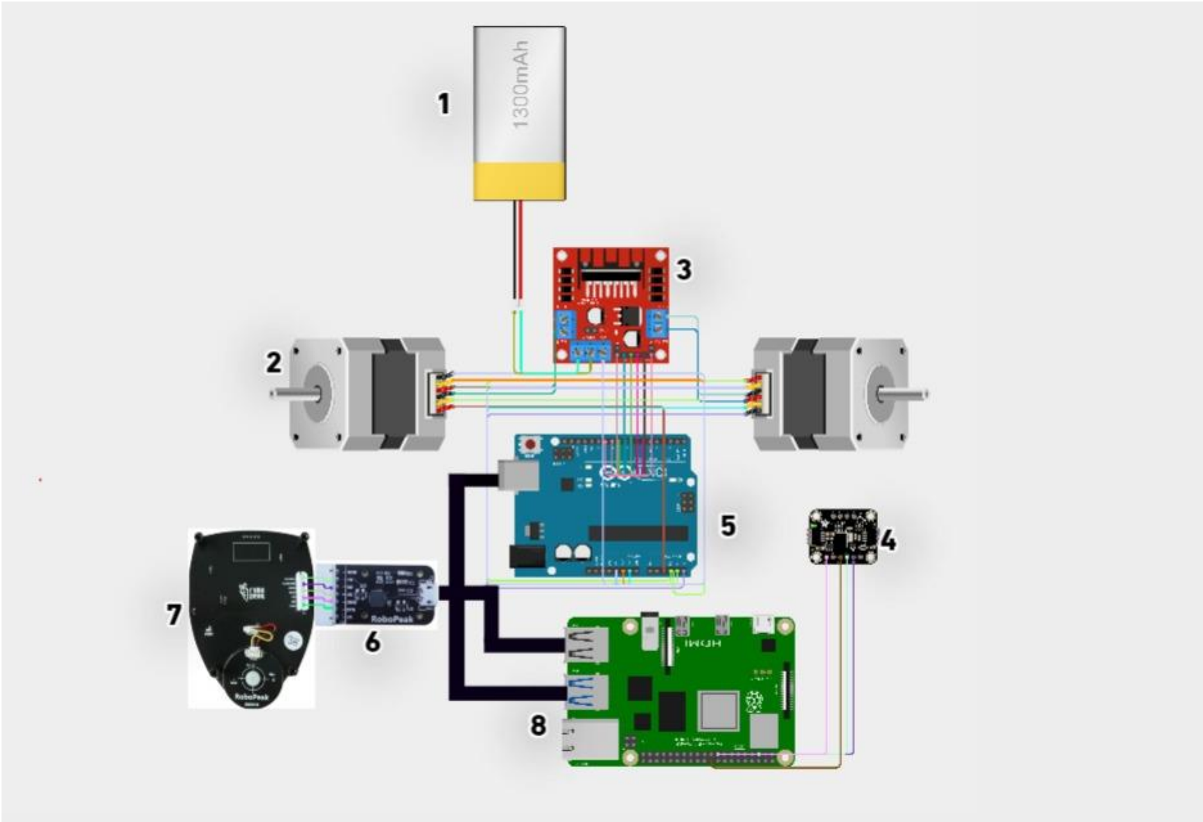

Figure 3.10 Circuit Diagram

1) Lipo Battery

2) Dc motors

3) Motor Driver

4) Imu sensor BN055

5) Arduino UNO

6) Lidar-USB interface

7) RP Lidar

8) Raspberry PI4

CHAPTER 4
RESULT AND DISCUSSION

The successful automation of maze-solving capabilities and obstacle detection and handling in a robot is a significant achievement that has numerous applications across various industries. This project used IR sensors to detect obstacles and walls, motor control, and a robotic arm to move objects out of the robot's path. The robot uses RP lidar A1 M8, IMU BNO055, ENCODER DC motor, and raspberry pi 4 calculations to make decisions, enabling it to traverse the maze autonomously while correcting any orientation errors arising from its physical motion within the maze.

The potential applications of this maze-solving robot are vast and can be seen in various industries. For instance, in the warehousing industry, automated crate moving is made possible by this robot, thus increasing efficiency and reducing the need for human labour. Military applications may include robots traversing unknown terrain while avoiding or moving obstacles out of the way, improving the safety of soldiers. In rescue or emergency scenarios, this robot can go into hazardous or hard-to-reach places and aid rescue operations.

Future improvements to this system could include incorporating a full visual system to enable the robot to have a better understanding of its environment. Additionally, a feedback memory system, as proposed in the analysis, can be implemented to enhance the robot's performance. Further studies can also be carried out to test the robot's maze-solving capability in larger and more complex mazes with reconfigurable features. Scaling down the model for easier testing and faster debugging would also be beneficial. Image processing could also be implemented to improve the robot's performance and increase its functionality.

CHAPTER 5
CONCLUSION

Simultaneous localization and mapping (SLAM) has become a key technology in the field of robotics, enabling robots to navigate and operate in unknown and dynamic environments. SLAM-based robots equipped with Raspberry Pi and LiDAR sensors can create detailed maps of their environment and navigate autonomously, opening up a wide range of applications in industries such as logistics, warehousing, and agriculture.

One of the major advantages of SLAM-based robots is their ability to work in dynamic and changing environments. With the help of advanced sensors and algorithms, they can quickly adapt to changes in their surroundings and adjust their navigation accordingly. This makes them ideal for applications such as search and rescue, where they can navigate through complex and hazardous environments to locate and rescue people in need.

In addition, SLAM-based robots can also be used in healthcare applications, such as assisting elderly and disabled individuals with daily activities, and in manufacturing, where they can operate in hazardous and high-risk environments to increase efficiency and productivity.
As technology continues to advance, the capabilities of SLAM-based robots are expected to improve further, with more advanced sensors, algorithms, and software being developed to enhance their performance. This will lead to the development of robots that can perform more complex tasks, such as autonomous vehicles that can navigate through city streets or drones that can perform detailed inspections of infrastructure.

Overall, SLAM-based robots with Raspberry Pi and LiDAR sensors are a promising area of research and development, with the potential to revolutionize a wide range of industries and improve the quality of life for people around the world. As research in this field continues to progress, we can expect to see even more exciting developments in the years to come.

Kalman filtering is a recursive algorithm that uses measurements and prior information to estimate the state of a system. The state of a system is defined as the set of variables that describe the system's current condition. Measurements are observations of the system that are used to update the state estimate. Prior information is information about the system's state that is known before any measurements are taken.

Kalman filtering is a powerful tool for estimating the state of systems that are subject to noise and uncertainty. It is widely used in a variety of fields, including:

1. Guidance, navigation, and control of vehicles
2. Signal processing
3. Econometrics
4. Robotics

Modelling of the central nervous system's control of movement

In guidance, navigation, and control of vehicles, Kalman filtering is used to estimate the current position, velocity, and other state variables of the vehicle based on measurements from sensors such as GPS, inertial measurement units (IMUs), and radar. By combining these measurements with predictions based on the vehicle's dynamics model, Kalman filtering can provide highly accurate estimates of the vehicle's state, even in the presence of noise and other sources of uncertainty.

In signal processing, Kalman filtering is used to estimate hidden state variables based on noisy measurements. For example, in speech recognition, Kalman filtering can be used to estimate the state of a hidden Markov model based on noisy observations of speech signals.

In robotics, Kalman filtering is used for motion planning and control. By using Kalman filtering to estimate the current state of a robot, it is possible to plan optimal trajectories that take into account the robot's dynamics and the uncertainty in its measurements. Kalman filtering is also used for robotic mapping and localization, where it is used to estimate the position and orientation of the robot based on sensor measurements.

Finally, Kalman filtering is used in the modelling of the central nervous system's control of movement. Due to the time delay between issuing motor commands and receiving sensory feedback, Kalman filtering provides a realistic model for making estimates of the current state of a motor system and issuing updated commands. By using Kalman filtering, it is possible to optimize the control of movement, leading to more efficient and accurate motor control.

Here are some additional details about Kalman filtering:

Kalman filtering is a linear estimator, which means that it assumes that the state of the system can be described by a linear model. This assumption is often not strictly accurate, but it can often be a good approximation.

Kalman filtering is a recursive estimator, which means that it can be used to update the state estimate as new measurements become available. This is in contrast to non-recursive estimators, which must be re-run from scratch whenever new measurements are available.

Kalman filtering is a very efficient estimator, which means that it can be used to estimate the state of a system quickly and accurately.

Kalman filtering is a powerful tool that can be used in a variety of applications. It is a versatile and efficient estimator that can be used to estimate the state of systems that are subject to noise and uncertainty.

FUTURE SCOPE:

Maze Solving Robots Will Become Even More Sophisticated And Capable. They Will Be Able To Solve More Complex Mazes, Navigate In More Difficult Environments, And Perform A Wider Range Of Tasks. This Will Lead To New And Innovative Applications For These Robots, Which Will Have A Positive Impact On Society.

SEARCH AND RESCUE

Maze Solving Robots Could Be Used To Search For Survivors In Collapsed Buildings Or Other Disaster Zones. These Robots Could Be Equipped With Cameras And Sensors To Help Them Identify Survivors And Assess The Damage. For Example, The Robots Could Be Equipped With Thermal Cameras To Detect Heat Signatures From Survivors, Or They Could Be Equipped With Microphones To Listen For Survivors' Voices.

INSPECTION

Maze Solving Robots Could Be Used To Inspect Infrastructure For Damage. These Robots Could Be Equipped With Sensors To Detect Cracks, Leaks, And Other Signs Of Damage. For Example, The Robots Could Be Equipped With Ultrasonic Sensors To Detect Cracks In Bridges, Or They Could Be Equipped With Gas Sensors To Detect Leaks In Pipelines.

DELIVERY

Maze Solving Robots Could Be Used To Deliver Goods To Customers In Difficult-To-Access Areas. These Robots Could Be Used To Deliver Food, Medicine, And Other Supplies To People In Remote Locations. For Example, The Robots Could Be Used To Deliver Supplies To People Who Live In Areas That Are Prone To Flooding, Or They Could Be Used To Deliver Supplies To People Who Live In Areas That Are Difficult To Reach By Car.

MANUFACTURING

Maze Solving Robots Could Be Used To Automate Tasks In Factories, Such As Picking And Placing Parts. This Could Help To Improve Efficiency And Reduce Costs. For Example, The Robots Could Be Used To Pick And Place Parts In Assembly Lines, Or They Could Be Used To Load And Unload Trucks.

HEALTHCARE

Maze Solving Robots Could Be Used To Assist With Surgery Or To Deliver Medication To Patients In Hospitals. This Could Help To Improve Patient Care And Reduce The Risk Of Errors. For Example, The Robots Could Be Used To Hold Surgical Tools, Or They Could Be Used To Deliver Medication To Patients In Their Rooms.

These Are Just A Few Of The Many Potential Applications For Maze Solving Robots. As Technology Continues To Advance, These Robots Will Become Even More Sophisticated And Capable. This Will Lead To New And Innovative Applications For These Robots, Which Will Have A Positive Impact On Society.

REFERENCES

International Conference on Promising Electronic Technologies (2021), Intelligent Maze Solving Robot Based on Image Processing and Graph Theory Algorithms.

1.IFSA World Congress and 20th NAFIPS International Conference, 2019. Joint 9th, pages 2781– 2785. IEEE, 2001Chatelais Q., Vulture H, and Canellas E., “Maze Solving by an Autonomous Robot”, Aalborg University, 2019.

2.A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision (2022).

3.International Journal of Engineering Research and General Science Volume 3, Issue 2, Part 2, March-April, 2021, “Simultaneous Localization and Mapping in Maze Solving”.

4.HECTORSLAM 2D MAPPING FOR SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM), Journal of Physics: Conference Series 1529 (2021).

5.ISPRS Journal of Photogrammetry and Remote Sensing 171 (2021) 367–384 “A cross correction LiDAR SLAM method for high-accuracy 2D mapping of problematic scenario “.

6.Introduction to Autonomous Robots" by Roland Siegwart, Lilah R. Nour bakhsh, and Davide Scarmorze: This textbook provides a comprehensive introduction to autonomous robots, including maze solving algorithms and techniques.

7."Programming Robots with ROS: A Practical Introduction to the Robot Operating System" by Morgan Quigley, Brian Geekery, and William D. Smart: This book covers the basics of robot programming using ROS and includes a section on maze solving robots.

8."Robotics: Modelling, Planning and Control" by Bruno Siciliano and Lorenzo Sciatica: This textbook covers the fundamentals of robotics, including kinematics, dynamics, control, and motion planning. It includes a section on maze solving robots and algorithms.

9."Robot Building for Beginners" by David Cook: This book provides a beginner-friendly introduction to building and programming robots, including maze solving robots.

10. "Microhouse: Maze Navigation and Mapping Strategies" by Matthew D. Barkeeper: This book focuses specifically on the design and implementation of maze solving robots for the microhouse competition. It covers various maze solving algorithms and strategies and includes practical tips and examples.

11. Autonomous Maze Navigation using LiDAR for Mobile Robots" by S. Ravi et al., 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

12."Development of Maze-Solving Robot Using LiDAR for Navigation" by J. M. Lee et al., 2017 IEEE International Conference on Mechatronics and Automation (ICMA)

13."A Maze-Solving Robot Using LiDAR and Machine Learning" by S. W. Kim et al., 2018 18th International Conference on Control, Automation and Systems (ICCAS)

14."Autonomous Navigation and Mapping of Unknown Environment using LiDAR" by S. K. Sahoo et al., 2018 International Conference on Computing, Power and Communication Technologies (GUCON)

15."ORB-SLAM: A Versatile and Accurate Monocular SLAM System" by R. Mur-Artal et al., 2018 IEEE International Conference on Robotics and Automation (ICRA)

16."Real-Time Loop Closure in 2D LIDAR SLAM" by J. Zhang et al., 2018 IEEE International Conference on Robotics and Automation (ICRA)

17."A Survey of SLAM Algorithms for Autonomous Robots" by T. Bailey et al., 2018 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)

18."A Comparison of SLAM Algorithms Based on a Graph of Relations" by L. Karlsson et al., 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)

19."Visual SLAM: Why Filter? And, Why Separate Initialization, Update, and Correction?" by A. Howard et al., 2019 IEEE International Conference on Robotics and Automation (ICRA)

20. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System" by R. Mur-Artal et al., 2015 IEEE International Conference on Robotics and Automation (ICRA)

21. "Real-Time Loop Closure in 2D LIDAR SLAM" by J. Zhang et al., 2018 IEEE International Conference on Robotics and Automation (ICRA)

22."A Survey of SLAM Algorithms for Autonomous Robots" by T. Bailey et al., 2006 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)

23."A Comparison of SLAM Algorithms Based on a Graph of Relations" by L. Karlsson et al., 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)

24."Visual SLAM: Why Filter? And, Why Separate Initialization, Update, and Correction?" by A. Howard et al., 2012 IEEE International Conference on Robotics and Automation (ICRA)

25."ROS Navigation Stack for Autonomous Mobile Robots in Indoor Environments" by M. Quigley et al., 2009 IEEE International Conference on Robotics and Automation (ICRA)

26."Navigation in Unknown Environments using ROS-based Autonomous Robot" by R. Singh et al., 2016 3rd International Conference on Signal Processing, Computing and Control (ISPCC)

27."Real-time Mapping and Navigation for Mobile Robots using ROS" by Y. Park et al., 2016 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)

28. "ROS-Based Navigation System for a Mobile Robot" by L. Paetz et al., 2019 International Conference on Unmanned Aircraft Systems (ICUAS)

29."Real-Time Odometry Estimation for Legged Robots Using Contact-Informed Dynamics and Stereo Vision" by N. Gopalan and M. C. Yip. IEEE Robotics and Automation Letters, vol. 5, no. 2, 2020.

30."Estimating Kinematics and Dynamics Parameters of a Wheeled Mobile Robot for Odometry Calibration" by R. Maier, S. I. Sheikh, and M. T. Hagan. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

31."Probabilistic Self-Calibration for Wheel Odometry using Maximum Likelihood Estimation" by M. Lauer, F. Scholten, and J. A. Stork. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

32."Distributed Fusion of Odometry and RGB-D Measurements for Accurate Localization in Complex Indoor Environments" by P. Huang, L. Zhang, and K. H. Low. 2019 IEEE International Conference on Robotics and Automation (ICRA).

33."Wheel Odometry Correction Based on a Machine Learning Approach" by X. Liu, H. Qiu, and Z. Wu. 2019 IEEE International Conference on Mechatronics and Automation (ICMA)

34. 1. Gul, F.; Rahiman, W.; Nazli Alhady, S.S. A comprehensive study for robot navigation techniques. Cogent Eng. 2019, 6, 1632046.

35. Pandey, A.; Pandey, S.; Parhi, D.R. Mobile robot navigation and obstacle avoidance techniques: A review. Int. Rob. Auto. J. 2017, 2, 00022

36. Coufal, P.; Hubálovský, Š.; Hubálovská, M. Application of Basic Graph Theory in Autonomous Motion of Robots. Mathematics 2021, 9, 919

37. Zhang, H.Y.; Lin, W.M.; Chen, A.X. Path planning for the mobile robot: A review. Symmetry 2018, 10, 450. [CrossRef]

38. Tullu, A.; Endale, B.; Wondosen, A.; Hwang, H.Y. Machine learning approach to real-time 3D path planning for autonomous navigation of unmanned aerial vehicle. Appl. Sci. 2021, 11, 4706.

39. Tirian, G.O. Maze-solving mobile robot. Ann. Fac. Eng. Hunedoara 2015, 13, 199

APPENDIX-I

PHOTOGRAPHS

APPENDIX - II

SOURCE CODE

Install the ROS Navigation Stack

```
sudo apt-get install ros-noetic-navigation
```

```
SUDO APT- sudo apt-get install ros-melodic-navigation
```

Create a Package

```
cd ~/catkin_ws/src/
mkdir jetson_nano_bot
cd ~/catkin_ws/src/jetson_nano_bot
roscd navstack_pub
mkdir param
roscd navstack_pub
gedit CMakeLists.txt
cd ~/catkin_ws/
catkin_make --only-pkg-with-deps navstack_pub
```

```
roscd navstack_pub
 mkdir launch
```

```
cd launch
gedit jetson_nano_bot.launch
```

Transform Configuration

```
roscd navstack_pub
cd launch
```

```
gedit jetson_nano_bot.launch
```

LIDAR Information

```
roscd navstack_pub
cd launch
gedit jetson_nano_bot.launch
```

Odometry Information

```
roscd navstack_pub
cd launch
gedit jetson_nano_bot.launch
```

Base Controller

```
roscd navstack_pub
cd launch
gedit jetson_nano_bot.launch
```

Mapping Information

```
roscd navstack_pub
cd launch
gedit jetson_nano_bot.launch
args="-d
/home/automaticaddison/catkin_ws/src/jetson_nano_bot/na
vigation_data_pub/maps/floorplan4.rviz"
```

Common Configuration (Global and Local Costmap)

```
cd ~/catkin_ws
roscd navstack_pub
cd param
gedit costmap_common_params.yaml
```

Global Configuration

```
cd ~/catkin_ws
roscd navstack_pub
cd param
gedit global_costmap_params.yaml
```

Local Configuration

```
cd ~/catkin_ws
roscd navstack_pub
cd param
gedit local_costmap_params.yaml
```

Base Local Planner Configuration

```
cd ~/catkin_ws
roscd navstack_pub
cd param
gedit base_local_planner_params.yaml
```

Move Base Node

```
cd ~/catkin_ws
```

```
roscd navstack_pub
cd launch
gedit jetson_nano_bot.launch
```

AMCL Configuration

```
cd ~/catkin_ws
roscd navstack_pub
cd launch
gedit jetson_nano_bot.launch
```

Launch the Autonomous Mobile Robot!

```
cd ~/catkin_ws/
catkin_make --only-pkg-with-deps navstack_pub
roslaunch navstack_pub jetson_nano_bot.launch
rosrun tf view_frames
evince frames.pdf
rqt_graph
```

APPENDIX-III

DATASHEETS

BN0055DATASHEET

Power Pins

- VIN: 3.3-5.0V power supply input
- 3VO: 3.3V output from the on-board linear voltage regulator, you can grab up to about 50mA as necessary
- GND: The common/GND pin for power and logic

I2C Pins

- SCL - I2C clock pin, connect to your microcontrollers I2C clock line. This pin can be used with 3V or 5V logic, and there's a 10K pullup on this pin.
- SDA - I2C data pin, connect to your microcontrollers I2C data line. This pin can be used with 3V or 5V logic, and there's a 10K pullup on this pin.

L298NDATASHEET:

Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low: -0.3V ≤ Vin ≤ 1.5V.
- High: 2.3V ≤ Vin ≤ Vss.
- Enable signal input voltage range :
    - Low: -0.3 ≤ Vin ≤ 1.5V (control signal is invalid).
    - High: 2.3V ≤ Vin ≤ Vss (control signal active).
- Maximum power consumption: 20W (when the temperature T = 75 ℃).
- Storage temperature: -25 ℃ ~ +130 ℃.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

Ardino Uno Datasheet:

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

RP LIDAR DATASET:

Laser Power Specification

- For Model A1M8 Only

Item	Unit	Min	Typical	Max	Comments
Laser wavelength	Nanometer(nm)	775	785	795	Infrared Light Band
Laser power	Milliwatt (mW)	TBD	3	5	Peak power
Pulse length	Microsecond(us)	TBD	110	300	

ENCODER DC MOTOR DATASET:

Order Option							
Order Code	Input Voltage	Rated		Weight (g)	Power (w)	Diameter (mm)	L (mm)
		Speed (RPM)	Torque (mN.m)				
SPG10-30K	6	440	29.4	10	-	12	24
SPG10-150K	6	85	107.9	10	-	12	24
SPG10-298K	6	45	176.5	10	-	12	24
SPG20-50K	12	130	58.8	60	0.6	27.2	-
SPG30-20K	12	185	78.4	160	1.1	37	22
SPG30-30K	12	103	127.4	160	1.1	37	22
SPG30-60K	12	58	254.8	160	1.1	37	25
SPG30-150K	12	26	588	160	1.1	37	27
SPG30-200K	12	17	784	160	1.1	37	27
SPG30-300K	12	12	1176	160	1.1	37	27
SPG50-20K	12	170	196	300	3.4	37	23
SPG50-60K	12	56	588	300	3.4	37	26
SPG50-100K	12	34	980	300	3.4	37	26
SPG50-180K	12	17	1960	300	3.4	37	28

RASPERI PI 4 DATASHEET:

	Raspberry Pi 4 Model B
Wireless connectivity	Yes
Display out	2 × micro HDMI port (supports up to 4Kp60)
Ethernet port	Yes (Gigabit Ethernet)
Processor	Quad-core 64-bit Cortex-A72 (Arm v8) BCM2711 @ 1.5GHz
RAM	1GB/2GB/4GB/8GB
CSI camera connector	Yes
microSD card slot	Yes
Bluetooth connectivity	Yes
Headphone/speaker/composite TV socket	4-pole 3.5mm stereo socket
USB ports	2 × USB 3.0 ports, 2 × USB 2.0 ports 1 × USB On-The-Go port
GPIO pins	Pre-soldered 40-pin GPIO header
DSI display connector	Yes
Power	5V/3A, USB-C

INFORMATION REGARDING STUDENTS

STUDENT NAME	EMAIL ID	PERMANENT ADDRESS	PHONE NUMBER	PLACEMENT DETAILS	PHOTOGRAPH
JAGAN V	jaganvenkatesh55@gmail.com	14 ,3RD A CROSS, SVG NAGAR,MUDALAPALYA, BANGALORE (560072)	9620417025	KYNDRYL	
GIRISH P	girishp0508@gmail.com	VIDYAMANYA NAGRA , HEROHALLI CROSS BANGALORE(560072)	9591351737	INTEVA PRODUCTS	
THUMMALA DHEERAJ GOUD	dheerajthummala2@gmail.com	BESIDE CHAITANYA SCHOOL , HUZURNAGAR (M),SURYAPETA(DIST), TELANGANA(ST),(508204)	9390029651	DXC TECHNOLOGIES	
BHASKAR M	bhaskarbass7722@gmail.com	MALLSANDRA OPP BANHGALORE1,DASARAHALLI, BANGALORE(560072)	7676600390	UNPLACED