Design and Development of an Indigenous Mobile Robot to Navigate in Cluttered Environment

Author

Muhammad Soleman Ali Shah

Regn Number

319574

Supervisor

Dr. Kashif Javed

Robotics and Intelligent Machine Engineering

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

MARCH 2023

Design and Development of an Indigenous Mobile Robot to Navigate in Cluttered Environment.

Author

Muhammad Soleman Ali Shah

Regn Number

319574

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Mechanical Engineering

Thesis Supervisor:

Dr. Kashif Javed

Thesis Supervisor's Signature: _____

Robotics and Intelligent Machine Engineering

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

MARCH 2023

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by Muhammad Soleman Ali Shah (Registration No. 319574), of Department of Robotics & AI (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature: _____

Date: _____

Dr. Kashif Javed (Supervisor)

Signature HOD: _____

Date: _____

Signature Principal: _____

Date: _____

FORM TH-4

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by **Muhammad Soleman Ali Shah** having **Regn. No. 319574**, titled "**Design and Development of an Indigenous Mobile Robot to Navigate in Cluttered Environment.**", be accepted in partial fulfillment of the requirements for the award of MS Robotics & Intelligent Machine Engineering degree.

Examination Committee Members

1. Dr. Muhammad Jawad Khan Signature: _____
2. Dr. Karam Dad Signature: _____
3. Dr. Muhammad Usman Bhutta Signature: _____

Supervisor: Dr. Kashif Javed Signature: _____

_____ _____

Date Head of Department

COUNTERSINGED

_____ _____

Date Dean/Principal

Declaration

I certify that this research work titled "*Design and Development of an Indigenous Mobile Robot to Navigate in Cluttered Environment."* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student
Muhammad Soleman Ali Shah
319574

2022-NUST-MS-RIME-000319574

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
Muhammad Soleman Ali Shah
319574

Signature of Supervisor

Dr. Kashif Javed

Copyright Statement

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of NUST School of Mechanical & Manufacturing Engineering (SMME). Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST School of Mechanical & Manufacturing Engineering, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the SMME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST School of Mechanical & Manufacturing Engineering, Islamabad.

Acknowledgements

I am grateful to my Creator Allah Subhana-Watala for guiding me through each step of this project and for every new idea that You implanted in my mind to improve it. Indeed, I could have accomplished nothing without Your invaluable assistance and direction. Whoever assisted me during the course of my thesis, whether my parents or anyone else, was Your will; therefore, no one deserves praise but You.

I would like to express my sincere gratitude to my beloved parents who have supported me throughout my life and also during this thesis. I am immensely grateful to my supervisor, Dr. Kashif Javed, who has provided me with constant guidance, motivation, and support throughout my thesis. His invaluable advice and direction have been instrumental in my success. I would also like to extend my thanks to my co-supervisor, Dr. Jawad Khan, who taught me the essential subjects of Computer Vision and Deep Learning.

I am indebted to my friend, Hasnain Ali Poonja, for his exceptional support and cooperation. Whenever I was in need, he encouraged me and provided assistance. Without his unwavering guidance, I would not have completed my thesis. I appreciate his patience and dedication throughout the entire process. I would also like to thank Dr. Karam Dad and Dr. Usman Bhutta for their invaluable contributions as members of the advisory and evaluation committee.

Lastly, I would like to express my heartfelt appreciation to everyone who has contributed to my academic success. Their encouragement and support have been vital to my accomplishments.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

The field of smart autonomous systems has experienced significant growth in recent years, with the development of robots aimed at assisting humans in various tasks. In particular, autonomous manipulators have been designed for disaster management and other situations where humans are inaccessible. This master's thesis presents the design and development of a mobile manipulator that can autonomously move in cluttered environments and perform pick and place tasks using 2D SLAM on ROS and 3D camera-based object detection. The proposed solution addresses the SLAM problem by utilizing the gmapping SLAM algorithm, which allows the robot to simultaneously locate itself and map its surroundings. The robot is equipped with a custom-made rover and a 6-DOF robotic arm assembled from ready-made links with joint servos. The arm is used to perform the pick and place tasks, and the 3D camera is used to estimate the coordinates of the targeted object, which is then used to control the robotic arm using inverse kinematics. The localization of the robot is done through 2D pose estimation using Kalman filter, and the destination position is set via RVIZ. The robot is designed to operate in indoor environments and can navigate autonomously using the 2D SLAM technique. The project demonstrates that the robot is capable of detecting the target object's 3D pose, estimating its coordinates, and accurately moving the robotic arm to achieve the desired pick and place task. Real experiments and demonstrations of the mobile manipulator's capabilities were performed using two Arduinos, one controlling the rover's motor and the other controlling the robotic arm's servos. The results of the experiments confirm the robot's ability to move autonomously and perform pick and place tasks accurately and efficiently. Overall, the mobile manipulator designed in this thesis provides a reliable solution for assisting humans in disaster management scenarios and other inaccessible environments. The use of 2D SLAM, 3D camera-based object detection, and inverse kinematics control for the robotic arm ensures efficient and accurate navigation and pick and place operations. The project can be extended to more challenging environments, such as outdoor and unstructured environments, with the integration of advanced sensors and algorithms.

Key Words: *Robot Manipulator, ROS, Disaster Management, GMapping, Move_base*

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Exploring unknown or dangerous locations, such as collecting geological samples on Mars or examining collapsed buildings for rescue purposes as well as give assistance to human, is a crucial function for autonomous robots to play because of the unique capabilities they possess. During these types of activities, perceptive sensors on the robots rebuild the three-dimensional world that they travel through. This process is referred to as "mapping" an environment. A map that is the output of cutting-edge SLAM algorithms will contain valuable information, such as the textures, locations, and geometries of the environment. The map might be stored in the local memory of the robot, or it might be transmitted to other devices across a wireless network. A mobile robot produced regional map is analogous to putting together a piece of a puzzle for an unknown environment.

The map is also used for many other things, like games, virtual reality, and, most importantly, planning robot routes. Real-time mapping is very helpful for rescue applications because it helps the crew prepare protective gear and plan the safest way to get to injured survivors [1]. For autonomous navigation to work, an accurate map has significant role. But mapping needs accurate estimates of where the exact posture of the mobile robot. Simultaneous Localization and Mapping (SLAM) was developed in the field of robotics to help robots deal with the challenge of mapping and figuring out where they are at the same time.

1.1 Background

Mobile robots have been widely used in various industries such as manufacturing, logistics, and healthcare. These robots can improve efficiency by automating repetitive tasks, reduce costs by reducing the need for human labor, and increase safety by performing tasks in hazardous environments. However, the deployment of mobile robots in indoor environments presents several challenges, one of which is their ability to navigate and perform tasks autonomously.

Indoor environments are often complex and dynamic, meaning that they change frequently, which makes it difficult for robots to navigate and perform tasks without human intervention. The complexity and dynamic nature of indoor environments can make it difficult for robots to accurately perceive and understand their surroundings. The limited sensing capabilities of robots, such as limited range and resolution, make it challenging for robots to detect and identify objects in the

environment. This can lead to a lack of precision and accuracy when performing tasks, which can affect the overall performance of the robot.

Furthermore, the varying lighting conditions and the presence of obstacles can make it difficult for robots to perceive their environment, which can lead to errors in navigation and localization. Additionally, the presence of people and other moving objects in indoor environments can further complicate the task of navigation, making it difficult for robots to avoid collisions and safely navigate through the environment.

The ability to navigate and perform tasks autonomously in indoor environments is one of the main challenges in the deployment of mobile robots. The complexity and dynamic nature of indoor environments, along with the limited sensing capabilities of robots, make it difficult for robots to accurately perceive and understand their surroundings. Therefore, developing a robot that can navigate and perform tasks autonomously in indoor environments is a crucial step in the deployment of mobile robots in various industries.

1.2 Problem Statement

The problem that this thesis aims to address is the ability of mobile robots to navigate and perform tasks autonomously in indoor environments. Indoor environments, such as warehouses, factories, and hospitals, are known for their complexity and dynamic nature. This complexity and dynamic nature of indoor environments make it difficult for robots to navigate and perform tasks without human intervention. The complexity arises from the presence of various obstacles, such as furniture and equipment, that can impede the robot's movement, and the dynamic nature arises from the presence of people and other moving objects that can constantly change the environment.

Additionally, the limited sensing capabilities of robots, such as limited range and resolution, make it challenging for robots to accurately perceive and understand their environment. These limited capabilities make it difficult for robots to detect and identify objects in the environment, which can lead to a lack of precision and accuracy when performing tasks. The inability to accurately perceive and understand the environment can affect the overall performance of the robot, making it less efficient and less safe.

In summary, the main problem that this thesis aims to address is the ability of mobile robots to navigate and perform tasks autonomously in indoor environments. The complexity and dynamic nature of indoor environments, along with the limited sensing capabilities of robots, make it

challenging for robots to navigate and perform tasks without human intervention, and to accurately perceive and understand their environment.

1.3 Proposed Solution

The proposed solution for this thesis is to develop a mobile manipulator that can navigate indoor environments using 2D simultaneous localization and mapping (SLAM) and perform pick and place tasks using a 6-degree-of-freedom (DOF) robotic arm that is mounted on a mobile rover.

The mobile manipulator will be equipped with a 2D laser scanner and an RGB-D camera to provide the necessary sensory information for SLAM. The 2D laser scanner will be used to detect obstacles and the RGB-D camera will be used to detect and identify objects. This combination of sensors will provide the mobile manipulator with a comprehensive view of its environment. The SLAM algorithm will be used to create a 2D map of the environment and to localize the robot within the map. This will enable the mobile manipulator to navigate autonomously and accurately understand its surroundings.

The 6-DOF robotic arm will be used to perform pick and place tasks, such as grasping and manipulating objects. The arm will be designed to have a high level of precision and accuracy, which will allow it to handle delicate objects with ease. The robotic arm will be mounted on a mobile rover that will be used to move the mobile manipulator to different locations within the environment. This will give the mobile manipulator the ability to navigate to different areas within the indoor environment and perform tasks in different locations.

In summary, the proposed solution for this thesis is to develop a mobile manipulator that can navigate indoor environments using 2D SLAM, perform pick and place tasks using a 6-DOF robotic arm, and move to different locations within the environment using a mobile rover. This will enable the mobile manipulator to navigate and perform tasks autonomously, and to accurately perceive and understand its environment.

1.4 Expected Outcome

The expected outcome of this thesis is the development of a mobile manipulator that can navigate indoor environments using 2D SLAM and perform pick and place tasks using a 6-DOF robotic arm that is mounted on a mobile rover. The mobile manipulator will be able to navigate

autonomously and perform tasks without human intervention. This means that the robot will be able to navigate and perform tasks on its own, without the need for human supervision.

Additionally, the mobile manipulator will be able to accurately perceive and understand its environment, allowing it to perform tasks with high precision and accuracy. This is a crucial aspect of the mobile manipulator's functionality as it will enable it to perform tasks with a high level of reliability and efficiency.

Overall, this research is expected to contribute to the development of autonomous mobile robots that can be used in indoor environments, such as manufacturing, logistics, and healthcare. This will enable the use of mobile robots in different applications, and will help to improve efficiency, reduce costs, and increase safety for employees.

1.5 Thesis Overview:

This thesis is focused on developing a mobile manipulator that can navigate indoor environments using 2D simultaneous localization and mapping (SLAM) and perform pick and place tasks using a 6-degree-of-freedom (DOF) robotic arm that is mounted on a mobile rover. The thesis is divided into several chapters, each of which focuses on different aspects of the research.

The introduction chapter provides an overview of the problem statement and the proposed solution. The literature review chapter presents an overview of the current state of research in the field of mobile manipulators. The methodology chapter outlines the methods used to develop and test the mobile manipulator. The results chapter presents the results of the experiments conducted to evaluate the performance of the mobile manipulator. Finally, the conclusion chapter summarizes the main findings of the research and suggests future work in the field.

The main goal of this thesis is to contribute to the development of autonomous mobile robots that can be used in indoor environments, such as manufacturing, logistics, and healthcare, and to improve the efficiency, reduce costs, and increase safety for employees.

CHAPTER 2: LITERATURE REVIEW

Robots are the next frontier technology with an impact on many different facets of our society, such as the manufacturing sector [2], the transportation sector [3], and the service sector [4]. From the first market floor cleaning robot to emotional robots, accompanying robots, education robots, rehabilitation robots, supermarket robots, and other direction extensions, service areas and service objects have continued to expand with the advancement of artificial intelligence technology.

Mobile robots offer a wide range of real-world applications due to their versatile operation capability in different environments [5]. It has been utilized in projects including the exploration of space [6], the development of safe and effective weapon systems, intelligent home applications, and military applications. [7]. Robotic rover goes where humans fear to tread. Whether in a hazardous condition, a robotic rover system can be applied to assess, reduce or eliminate the risk. Through this, the rover can prevent any life-threatening or risk of death on people that lurks with every wrong move [8]. There have been many mobile robots developed that are designed to operate in dangerous environments where humans cannot go. For example, [9] developed a firefighting robot that can be controlled from some distance and can detect fire and smoke from a distance, and can be extinguished by spraying water. The opportunity rover is a Mars explorer mobile robot that was constructed by NASA [10]. The robots are of great assistance in places where human access is physically impossible.

2.1 Mapping

Mapping is a crucial element in the development of mobile rovers that can navigate autonomously in indoor environments. These techniques enable the robot to create a map of its surroundings and use this map to navigate through the environment, avoiding obstacles and reaching its destination.

2D mapping, also known as simultaneous localization and mapping (SLAM), is particularly useful for mobile robots as it allows the robot to create a map of its environment while also determining its own location within that environment. This is achieved through the use of sensors such as cameras, lidar, and other sensor modalities that allow the robot to perceive its surroundings.

There are several different methods for 2D mapping, including feature-based SLAM, which uses geometric features in the environment to determine the robot's location, and featureless SLAM, which uses sensor data such as lidar scans to create a map without the use of specific features.

Recent research in 2D mapping has focused on improving the accuracy and robustness of SLAM algorithms. For example, in [2], [11] a SLAM algorithm that uses deep learning to improve the accuracy of feature-based SLAM is proposed. In [12], a SLAM algorithm that combines lidar and visual odometry is presented, resulting in improved robustness in challenging environments.

Furthermore, in order to achieve a high level of autonomy for a mobile robot, it is necessary to have an accurate and detailed map of the environment. In recent years, researchers have proposed various mapping techniques using different sensor modalities, such as RGB-D cameras [13], LiDAR [14], and monocular cameras [15]. These techniques are capable of creating detailed and accurate maps of indoor environments, allowing the robot to navigate with a high level of autonomy.

In summary, 2D mapping and mapping techniques play a vital role in the development of mobile rovers that can navigate autonomously in indoor environments. Recent research has focused on improving the accuracy and robustness of SLAM algorithms and mapping techniques, making it possible for mobile robots to navigate with a high level of autonomy.

2.2 Localization:

Localization of mobile robots refers to the process of determining the position and orientation of the robot in its environment. This is a crucial aspect of robot navigation and enables the robot to perform tasks such as mapping, exploration, and obstacle avoidance. This literature review presents an overview of the various techniques used for localization of mobile robots, with a focus on their advantages, disadvantages, and applications [16].

- Triangulation-based localization: This technique involves the use of beacons or landmarks in the environment to determine the location of the robot. The robot receives signals from multiple beacons and triangulates its position based on the received signal strengths.

Advantages of this approach include high accuracy and the ability to work in large environments. However, this technique requires pre-existing infrastructure and can be affected by environmental factors such as signal interference [17].

- GPS-based localization: Global Positioning System (GPS) is a satellite-based navigation system that provides location information to mobile devices. This approach can be used for robot localization by providing accurate location information in outdoor environments. The main advantage of GPS-based localization is that it does not require additional infrastructure, but it may be affected by environmental factors such as signal interference, satellite availability, and atmospheric conditions [18] .

- Visual odometry: This approach uses cameras and computer vision algorithms to estimate the position of the robot by analyzing the changes in its surroundings. Advantages of visual odometry include the ability to work in unstructured environments and the ability to provide both position and orientation information. However, visual odometry can be affected by lighting conditions and the availability of features in the environment [19].

- Inertial navigation system (INS): This approach involves the use of accelerometers and gyroscopes to determine the position and orientation of the robot. The INS estimates the position of the robot based on its motion and the measurement of its orientation. Advantages of the INS include its ability to work in challenging environments, such as those with limited visibility or GPS availability, and its low cost. However, the INS is subject to errors caused by sensor drift and the accumulation of errors over time [11, 20-22].

- Particle filter-based localization: This approach uses a probabilistic method to estimate the position of the robot based on sensor measurements. The particle filter generates a set of particle hypothesis and updates their position based on the measurement data. This approach has been used in many applications and has been shown to provide high accuracy in various environments. However, particle filter-based localization can be computationally intensive and may require large amounts of memory [23].

Inertial navigation system localization of mobile robots involves the use of accelerometers and gyroscopes to determine the position and orientation of the robot in its environment. This type of localization is often used in situations where GPS signals are not available or unreliable, such as in indoors or underground environments. The accelerometers measure linear acceleration and the gyroscopes measure angular velocity, which are used to estimate the velocity and orientation of the robot. The integration of these measurements over time provides an estimate of the position of the robot. However, this estimate is subject to drift and errors over time, which must be corrected through the use of additional sensors, such as optical sensors or magnetic compasses, or through the use of algorithms that compensate for drift and errors [20, 22].

Inertial navigation system localization has been widely used in various applications, including mobile robots for exploration and mapping, industrial automation, and autonomous vehicles. It provides a reliable and accurate means of localization in environments where other forms of localization are not possible or are unreliable. Overall, inertial navigation system localization is a critical component of many mobile robot applications, as it provides the information necessary for the robot to navigate and perform tasks in its environment. With continued advances in sensor and algorithms, it is expected that this technology will become increasingly sophisticated and widely used in a range of applications [21].

2.3 Navigation:

When it comes to mobile robots that operate autonomously, navigation plays a critical role in reaching their goals independently. There are several navigation techniques, the most well-known one is SLAM (Simultaneous Localization and Mapping). Visual simultaneous localization and mapping (SLAM) is rapidly becoming a major innovation in embedded vision, and it has the potential to be used in a wide variety of contexts. It is the process of mapping the world around a sensor while concurrently determining the position and orientation of that sensor in relation to its surroundings which is localization [24]. The Robot Operating System (ROS), which has become the standard robotics middleware and has been proposed to assist the development of software for robots, is a major contributor to the rise in interest in SLAM [25]. It is not possible for a robot to navigate independently based solely on estimations of its pose or a map of the environment in which it operates. In addition to providing SLAM estimates with the smallest amount of error possible, a

robot is required to plan a trajectory from its current location to goal locations using an appropriate map representation, such as an occupancy grid, and then carry out the necessary commands in order to travel along the route that was planned [25]. In general, SLAM algorithms need to be able to correlate sensor data acquired from various time steps with current sensor data as well as current map data. In addition, SLAM algorithms need to be able to associate current sensor data with previous map data. The final capability is referred as loop closing [26]. Lasers, monocular and stereo cameras, and RGB-D cameras [27] are among the sensor-based algorithms that have been developed and tested for use in a variety of indoor and outdoor environments. ROS SLAM systems can be divided into two major groups based on the type of sensor and the features of the surrounding environment: lidar and vision based [28]. Lidar-based SLAM algorithms combine laser scans with robot odometry or an IMU to create a map. Assuming that the planar lidar on the robot moves on a 2D plane, the estimated robot postures are constrained to three degrees of freedom (DoF) (2D position and an orientation angle) [29]. GMapping and tinySLAM are two approaches to solving the SLAM problem. These approaches calculate the probability density function of the robot's pose and perceived map based on laser scans and robot odometry [30]. The foundation for real-time 3D reconstruction used by RGB-D SLAM, LSD SLAM, ORB SLAM, ORB SLAM 2, and RTAB-Map is known as visual odometry. This framework, in turn, was derived from the projective geometry theory. These algorithms estimate the 6 DOF robot's pose in relation to the surrounding environment [31]. The fact that ROS packages of this class provide a two-dimensional representation of the environment known as an occupancy grid is an obvious advantage of lidar-based techniques. In general, the visual SLAM equivalent does not produce this kind of representation of the environment. Because path planning techniques use the occupancy grid as one of the input data, it is necessary for robot navigation [32]. The field of robotics has a significant challenge in the form of mobile robot navigation. They are renowned for the intelligent characteristics that they possess. They also cover a wide range of applications, including those in the transportation and industrial sectors as well as rescue robots. When it comes to autonomous mobile robot navigation, path planning is one of the most prominent and vital components. Path planning entails locating a collision-free route from one location to another while reducing the overall cost of traveling that route to the greatest extent possible. Path planning can be broken down into either a static or dynamic environment, depending on the kind of setting of environment that is being considered [33]. Navigation can be divided into two types: Global and local Navigation [34]. Prior information of the

surroundings is required for global navigation, also known as Off-line mode for path planning. For example: Dijkstra algorithm, A* algorithm, Visibility graph, Artificial potential field method, and cell decomposition method [35]. For local navigation, also known as the On-line mode for path planning, in which the robot determines its own position and orientation and can regulate its mobility using sensors that are installed externally, it is possible to use infrared sensors, ultrasonic sensors, LASER sensors, and vision sensors (cameras) to autocorrect the orientation of a robot via software. Other techniques include neural networks, fuzzy logic, neuro-fuzzy networks, particle swarm optimization, genetic algorithms, and adaptive control optimization [35].

2.4 Robotic Arm:

Robotic arms are widely used in various industries for tasks such as welding, painting, and material handling. One of the most crucial aspects of robotic arms is their ability to manipulate objects in a 3D environment. This capability is achieved through the use of 6 Degrees of Freedom (6-DOF). The 6-DOF of robotic arms refers to the number of axes along which the end-effector can move. In this literature review, the focus will be on the technicalities of 6-DOF of robotic arms and how these technicalities impact their applications[36].

2.5 6-DOF Robotic Arms

There are two types of 6-DOF robotic arms: serial and parallel. Serial robotic arms are designed with a chain of links that are connected in a sequential manner. Each link in the chain is responsible for a specific degree of freedom. These robotic arms are capable of reaching a high degree of precision, but they are limited in terms of speed and load capacity. On the other hand, parallel robotic arms are designed with multiple links that are connected in a parallel manner. These robotic arms are capable of providing high speed and load capacity, but they are limited in terms of precision[37].

Kinematics is a branch of mechanics that deals with the motion of objects without considering the forces that cause the motion. The kinematics of 6-DOF robotic arms involves the calculation of the position, orientation, and velocity of the end-effector in a 3D environment. Forward kinematics refers to the process of determining the position and orientation of the end-effector of a robotic arm given its joint angles or configuration. The end-effector is the tip of the robotic arm and is typically

equipped with a tool or sensor for performing tasks. The forward kinematics problem is critical in robotic applications, such as robot manipulation and grasping, since it determines the position and orientation of the end-effector in the workspace.

There are two main approaches to solving the forward kinematics problem: analytic and numerical. Analytic solutions use mathematical equations to determine the position and orientation of the end-effector, while numerical solutions use algorithms to iteratively find the solution. Analytic solutions are typically faster and more efficient but can be complex to derive and are limited to specific configurations. Numerical solutions, on the other hand, are more flexible but require more computational resources[38].

2.6 Denavit-Hartenberg Convention

The Denavit-Hartenberg (DH) convention is a widely used method for representing the kinematic parameters of a robotic arm. The DH convention represents the configuration of a robotic arm using four parameters: alpha (α), the angle between two consecutive links in the Z-X'-plane, theta (θ), the angle between two consecutive links in the Z-Z' plane, d, the length of the common normal between two consecutive links, and a, the length of the common normal between the X-axis and X'-axis.The DH convention provides a systematic way of representing the kinematic parameters of a robotic arm and is widely used in the design and analysis of robotic systems. The parameters can be used to derive the transformation matrix between two consecutive links, which can be combined to determine the position and orientation of the end-effector. The DH convention has proven to be an effective and efficient method for solving the forward kinematics problem of a robotic arm[39].

2.7 Inverse Kinematics:

Inverse kinematics is a fundamental aspect of robotics that involves finding the joint angles required to place the end-effector of a robotic arm at a desired position in space. A 6-DOF (degree of freedom) robotic arm has six joints that allow it to move in multiple dimensions, making inverse

kinematics a complex problem. In this literature review, we will examine the various techniques used to solve the inverse kinematics of 6-DOF robotic arms[40].

2.7.1 Geometric Methods

Geometric methods are mathematical approaches used to determine the inverse kinematics of a robotic arm by using the geometry of the arm and its position in space. One of the most common methods is the Denavit-Hartenberg (DH) method, which uses transformations and rotations to describe the position and orientation of the arm's joints. This method has been widely used for solving inverse kinematics in robotics[41].

Another popular geometric method is the Jacobian inverse method, which involves the use of the Jacobian matrix to determine the inverse kinematics. This method calculates the inverse kinematics based on the gradient of the joint angles and the desired end-effector position[41, 42].

2.7.2 Numerical Methods

Numerical methods are computational approaches used to solve inverse kinematics problems. They involve iterative algorithms to find the joint angles that will place the end-effector at the desired position. One of the most commonly used numerical methods is the gradient descent method, which minimizes the error between the desired and current end-effector position.Another popular numerical method is the inverse Jacobian method, which involves the use of the inverse of the Jacobian matrix to calculate the joint angles. This method has been widely used in industrial applications due to its simplicity and effectiveness[40, 43].

2.7.3 Artificial Intelligence Techniques

Artificial intelligence (AI) techniques have been increasingly used to solve inverse kinematics problems in robotics. These techniques involve the use of machine learning algorithms to determine the inverse kinematics. One of the most popular AI techniques is the artificial neural network (ANN) method, which uses a network of interconnected nodes to find the inverse kinematics. ANNs have been shown to be effective in solving inverse kinematics problems in complex robotic systems. Another AI technique is the fuzzy logic method, which involves the use of fuzzy logic to find the inverse kinematics. This method has been used to solve inverse kinematics problems in robotic systems where the relationships between the joints and end-effector are not well defined[44].

CHAPTER 3: METHODOLOGY

The project aims to design and construct a mobile robot equipped with a robotic arm. The robot has been designed to move autonomously using simultaneous localization and mapping (SLAM) and navigation techniques. The robot is capable of recognizing its environment and moving around it, allowing it to reach its desired destination.

Once the robot has arrived at its destination, the robotic arm will begin to function. The 3D intel realsense camera is used to extract the 3D position coordinates of the object. The robotic arm then employs inverse kinematics to pick up the object and place it in the desired location.

The project has been broken down into two parts to ensure a streamlined and organized approach to the design and construction process. The first part of the project focuses on the design and navigation of the rover. This involves the creation of a mobile platform that is capable of moving autonomously, recognizing its environment and reaching its destination.

The second part of the project focuses on the setup of the 6-DOF robotic arm and inverse kinematics. The 6-DOF arm refers to the six degrees of freedom that the arm has, which include the ability to move in three linear dimensions (up/down, left/right, and forward/backward) and three rotational dimensions (pitch, yaw, and roll). Inverse kinematics is the mathematical process used to determine the positions and orientations of the robotic arm and joints in order to pick up and place objects.

The 3D intel realsense camera is used to extract the 3D position coordinates of the object. This information is then used by the inverse kinematics algorithms to determine the positions and orientations of the robotic arm and joints in order to pick up the object. Once the object has been picked up, the inverse kinematics algorithms are used again to determine the positions and orientations of the robotic arm and joints in order to place the object in its desired location.

The simultaneous localization and mapping (SLAM) and navigation techniques used in the project are crucial to the success of the autonomous movement of the robot. SLAM allows the robot to build a map of its environment in real-time, and to use that map to determine its position and navigate around it. The navigation techniques used in the project ensure that the robot is able to move around its environment in a safe and efficient manner.

The combination of the mobile platform, robotic arm, SLAM and navigation techniques, and inverse kinematics algorithms provides a powerful and flexible solution for autonomous object pick up and placement. The robot is capable of recognizing its environment and reaching its destination, and the robotic arm is capable of picking up and placing objects with precision.

Rover Design and Odometry:

The design and construction of a mobile robot requires several key components. One of the most important components is the Intel Realsense D415 3D intel camera, which is used to extract the 3D position coordinates of objects. This information is then passed to the 6-DOF robotic arm for object manipulation. The MPU6050 IMU is used for accurate navigation and the arduino micro-controller is used to control all the components. A ROS installed laptop is necessary for implementing the simultaneous localization and mapping (SLAM) algorithms, allowing the robot to move autonomously in recognized environments. The L298 motor driver is used to control the two motors with encoders and a castor wheel, which provide mobility to the robot. A power supply is required to run all the components and a wood base is used to house the entire system.

Table 3-1: Robot Components

Component	Function
Intel Realsense D415 3D intel camera	Extracts 3D position coordinates of objects for object manipulation
MPU6050 IMU	Provides accurate navigation for the mobile robot
Arduino micro-controller	Controls all the components of the mobile robot
ROS installed laptop	Implements SLAM algorithms for autonomous movement in recognized environments
L298 motor driver	Controls the two motors with encoders and a castor wheel for mobility of the robot
Power supply	Provides power to all the components of the mobile robot
Wood base	Houses the entire system of the mobile robot

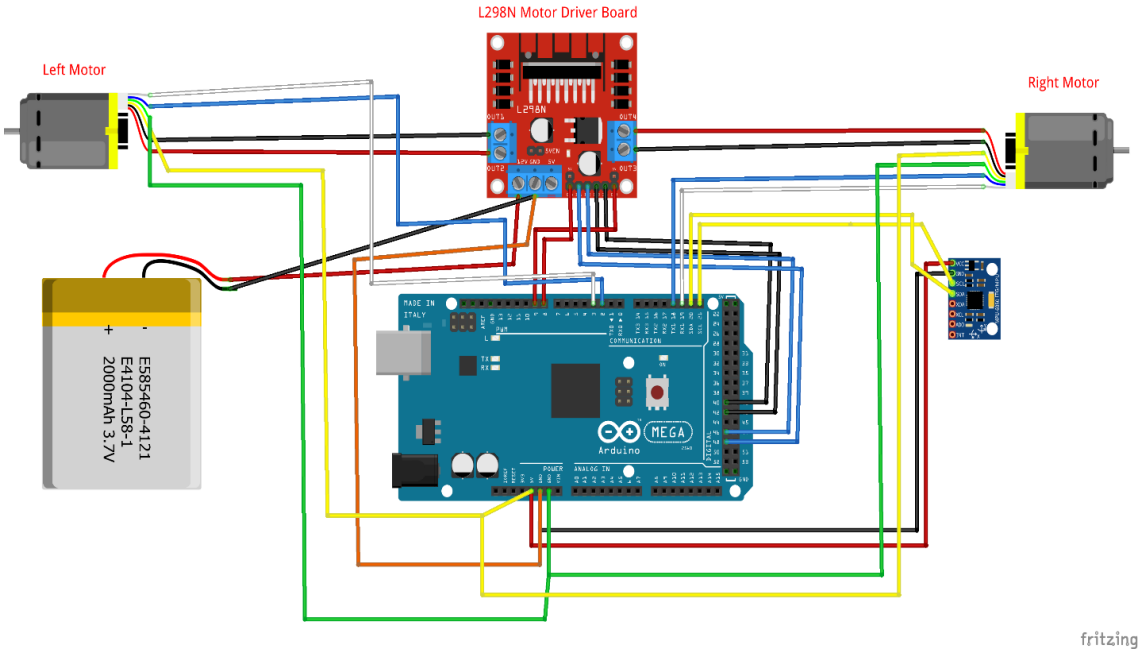

Figure 3.1: Electrical Circuit Diagram of Mobile Robot

Odometry is a technique used for estimating the positional change of an object over time. In the context of robotics, it refers to the process of calculating the position of a mobile robot based on data collected from motion sensors. The most common method for implementing odometry in a mobile robot is to measure the change in velocity over time and integrate the result to obtain an estimate of the current position.

ROS (Robot Operating System) provides a navigation stack that helps in the implementation of odometry in mobile robots. The navigation stack allows the robot to move autonomously by publishing velocity commands to a topic called /cmd_vel. The velocity commands are sent in the form of geometry/Twist messages, which specify the linear and angular velocity of the robot.

The velocity commands are transmitted to the robot's base controller, which is typically implemented on an Arduino board. The base controller is responsible for receiving the twist messages through rosserial_python, a library that allows communication between the ROS navigation stack and the Arduino. Once the velocity commands have been received, the base controller translates them into motor commands that control the motion of the robot.

To ensure that the robot moves at the desired velocity, a PID (Proportional-Integral-Derivative) controller is used. The PID controller works by continuously monitoring the speed of each motor and calculating the total error between the desired speed and the actual speed. The error is used to adjust

the PWM (Pulse Width Modulation) signal generated by the base controller, which controls the speed of the motors.

The PID controller performs this adjustment by calculating the amount of PWM signal that must be generated to spin each motor based on the total error over time. The PWM signal is proportional to the error, meaning that the larger the error, the greater the PWM signal generated. The integral component of the PID controller accumulates the error over time, which helps to eliminate any persistent errors. Finally, the derivative component of the PID controller predicts the future error based on the rate of change of the current error, which allows for quick adjustments to the PWM signal.

Odometry is an essential component for the navigation and localization of mobile robots. It is a technique used to estimate the position and orientation of the robot in the environment. The objective of odometry is to determine the relative motion of a robot in real-time by using the information provided by the encoders and the Inertial Measurement Unit (IMU) sensor.

A motor equipped with encoders and an IMU sensor is used in odometry to determine the position of a robot in relation to its surroundings. The encoders are responsible for determining the number of revolutions per minute (RPM) by counting the number of clicks. On the other hand, the IMU sensor is used to measure the angular displacement and velocity of the robot. The angular velocity of the robot is used to calculate its linear velocity. The combination of linear and angular velocity provides the raw odometry data, which is then used for navigation and localization purposes.

In odometry, the linear velocity is calculated by counting the change in the number of ticks over time. The tick count is obtained from the encoders, and the change in the tick count over a specific period provides the linear velocity of the robot. The angular velocity is calculated from the IMU sensor, which measures the change in orientation of the robot over time. The linear and angular velocity obtained from the raw odometry data are combined to determine the motion of the robot in the environment.

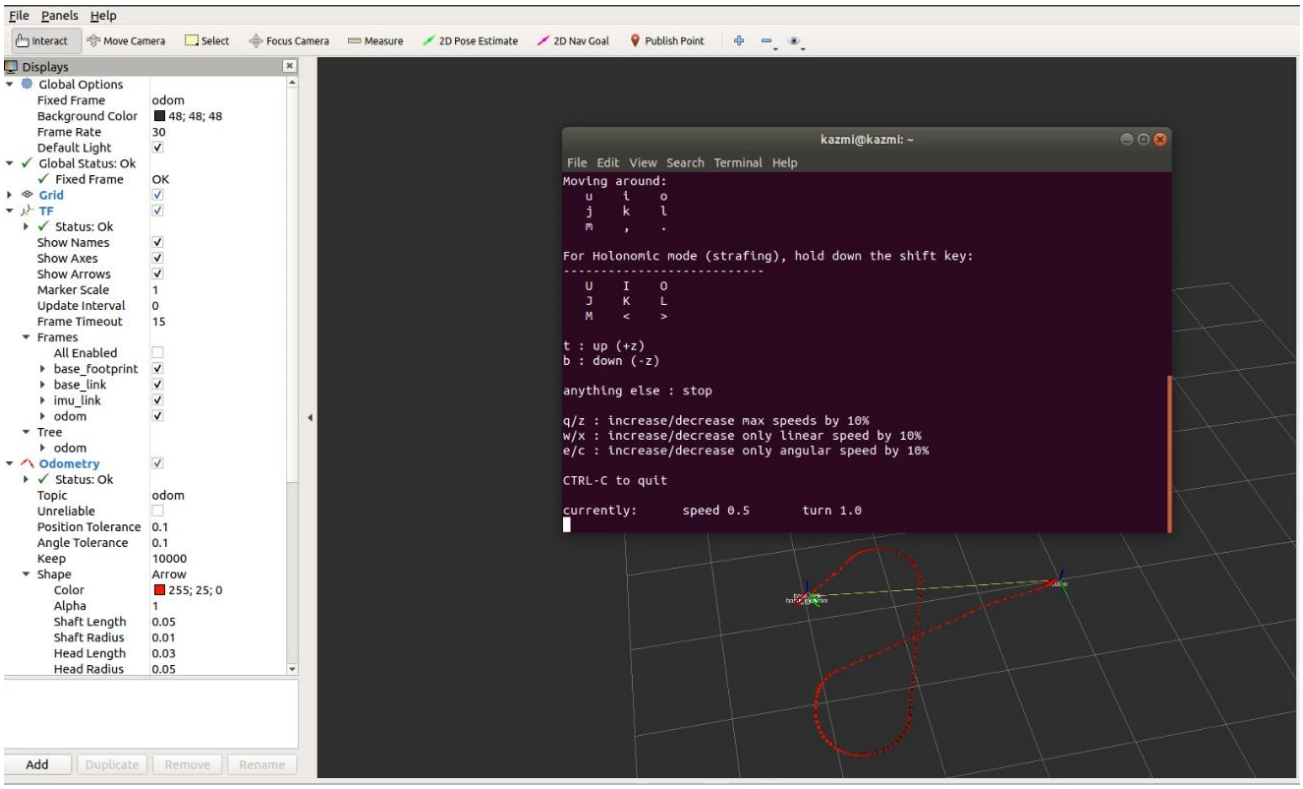

Figure 3.2: Robot Odometry

However, it is important to note that the accuracy of the odometry method is susceptible to system noise and wheel slippage. Dead reckoning, which relies entirely on the encoders' measurement, may lead to inaccuracies in the odometry data. This is because the encoders may not provide accurate readings if there is a presence of system noise or wheel slippage. System noise refers to the fluctuations in the readings obtained from the encoders, while wheel slippage refers to the difference between the actual velocity of the wheels and the velocity measured by the encoders. The odometry result is given below…

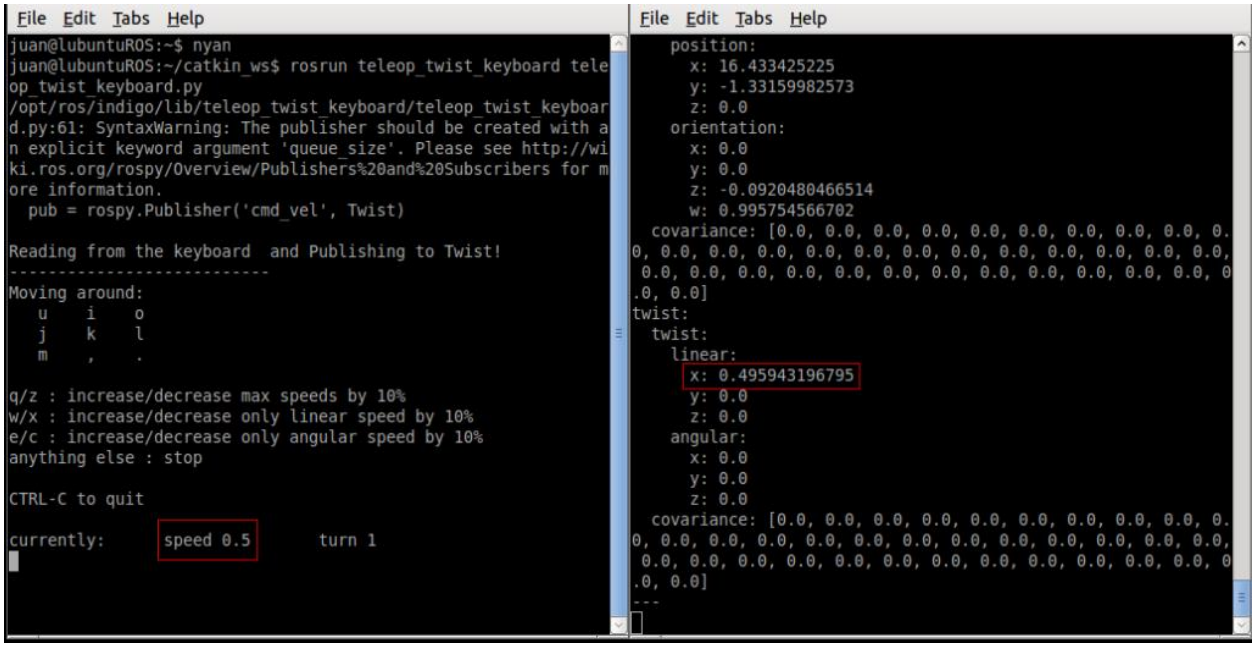

Figure 3.3: Robot Position Orientation and Speed

To achieve accurate movement and positioning, the velocity of the robot is important. However, sometimes the velocity data provided by the motor encoders can be noisy or unreliable. To address this, the velocity data is combined with orientation data from an Inertial Measurement Unit (IMU), which is a sensor suite that includes an accelerometer and gyroscope. The IMU publishes raw data with reference to the inertial frame.

Table 3-2: Odometry Components Details

Component	Function	Data Collected	Output
Encoders	Measure wheel rotation	Number of clicks	Linear velocity
IMU	Measure angular displacement and velocity	Orientation change	Angular velocity
Base Controller	Receives velocity commands, generates motor commands	Twist messages	PWM signal

PID Controller	Adjusts motor speed based on error	Speed of each motor	Adjusted PWM signal
Navigation Stack	Sends velocity commands	Desired linear and angular velocity	/cmd_vel topic
Arduino	Implements base controller	PWM signal	Motor speed
ROS	Provides navigation stack, communication library, and sensor fusion	Odometry data, IMU data, velocity commands	Filtered odometry data

To clean up the raw IMU data, the ROS (Robot Operating System) package "imu_madgwick_filter" is used. The filtered data is then published in the "imu/data" topic. This filtered data is then combined with the data from the motor encoders using an Extended Kalman Filter (EKF) through the "robot_localization" ROS package. The EKF fuses the odometry data provided by the IMU and motor encoder to provide the robot with an educated guess of its current pose. This helps the robot to maintain its accuracy even when the data from one of the sensors becomes too noisy or starts to go missing.

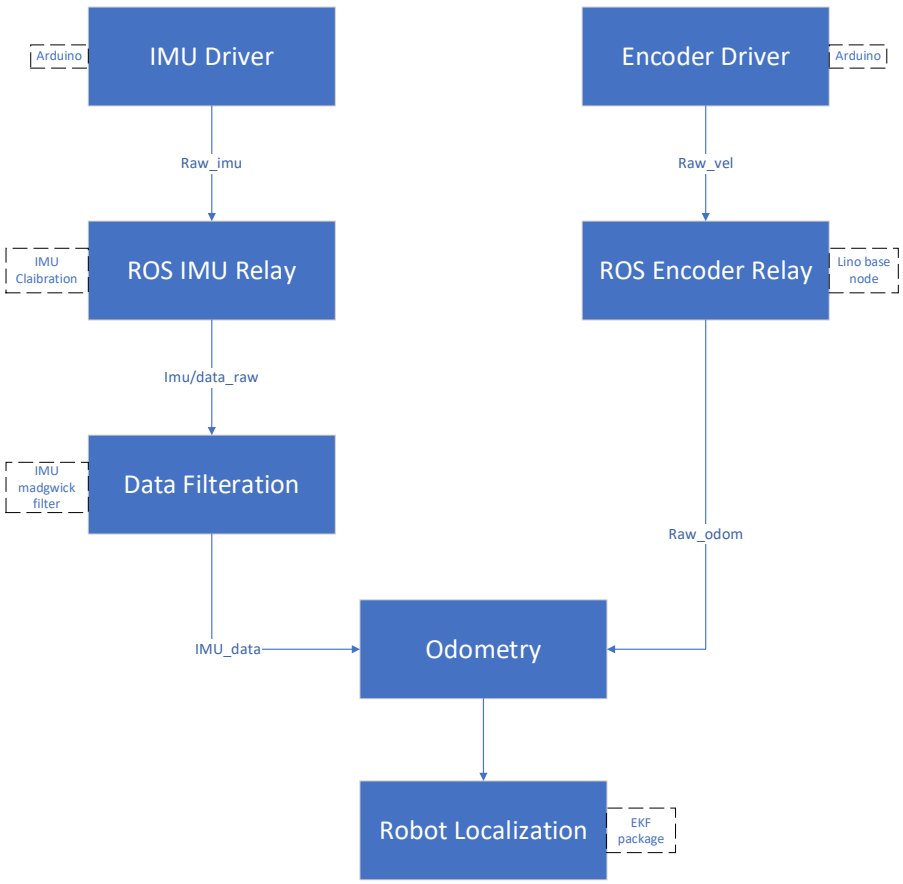

Figure 3.4: Flow Chart of Robot Odometry

3.2 Mapping:

The implementation of a fully autonomous mobile robot requires a robust navigation system that can accurately perceive and map the environment while avoiding obstacles. The key to a successful navigation system is the integration of a LIDAR or 3D depth sensor. These sensors provide real-time information about the robot's surroundings, which is then used to generate a map and plan a safe path for the robot to follow.

In this project, the selected 3D depth sensor is the Intel Realsense D415, which is an advanced depth camera that provides high-resolution depth images, along with RGB images, that are ideal for mapping and localization purposes. The integration of this camera with the robot's navigation stack is made possible by the use of the ROS Wrapper for Intel RealSense Devices.

The ROS Wrapper for Intel RealSense Devices is a software package that enables the Intel Realsense D415 camera to communicate with the Robot Operating System (ROS). This wrapper allows the camera to publish its depth and RGB images to ROS topics, making the data from the

camera accessible to the navigation stack. The use of the wrapper simplifies the integration process and ensures that the camera data is properly formatted and ready for use by the navigation stack.

To install the ROS Wrapper for Intel RealSense Devices, the user simply needs to follow the installation instructions provided by the software developers. Once the wrapper is installed, the camera is connected to the robot and the appropriate topics are configured. The depth and RGB images are then published to the ROS topics and are available for use by the navigation stack.

The use of the Intel Realsense D415 and the ROS Wrapper for Intel RealSense Devices is a critical component of the navigation system of this mobile robot. The depth information provided by the camera is used to generate a 3D map of the environment, and the navigation stack uses this map to plan a safe path for the robot to follow. The robot's movement is controlled by the navigation stack, which uses the depth information from the camera to avoid obstacles and maintain its position relative to the environment.

The Robot Operating System (ROS) Intel RealSense wrapper is used to stream RGB (color) and depth images from an Intel RealSense camera. These images are published to the topics "/camera/color/image_raw" and "/camera/depth/image_rect_raw" respectively. The RGB images provide a visual representation of the environment captured by the camera, while the depth images contain information about the distance of each object to the camera. This information can be used to construct a 3D map of the environment, but for 2D mapping, we need laser scans.

The Depthimage_to_laserscan package is used to convert depth images into laser scans. This package takes the depth image as input and outputs a laser scan, which provides a 2D representation of the environment. The laser scan contains information about the distance of objects in the environment at various angles, creating a 2D map of the environment.

Laser scans are widely used in robotics for navigation and mapping purposes. They provide a fast and efficient way to generate a 2D map of the environment, which can then be used for localization and path planning. The Depthimage_to_laserscan package allows us to take advantage of the information contained in the depth images to generate laser scans, which can be used for 2D mapping and navigation.

The package works by converting each pixel in the depth image into a laser measurement at a specific angle. The angle is calculated based on the position of the pixel in the image and the field of view of the camera. The distance information from the depth image is then used to determine the

range of the laser measurement. The resulting laser scan contains a set of measurements at various angles, creating a 2D representation of the environment.

The output of the Depthimage_to_laserscan package can be used by other ROS packages for navigation and mapping purposes. For example, the data can be fed into a SLAM (Simultaneous Localization and Mapping) algorithm, which can be used to generate a map of the environment in real-time. The map can then be used by a navigation stack to plan and execute paths through the environment.

For example, here is the scene in which the following screenshots were captured.

Figure 3.5: depthimage_to_laserscan Physical Environment

By using depthimage_to_laserscan package with appropriate parameters. The package overlayed sensor_msgs/LaserScan in color on the sensor_msgs/Image. Red is close to camera; purple is far from camera.

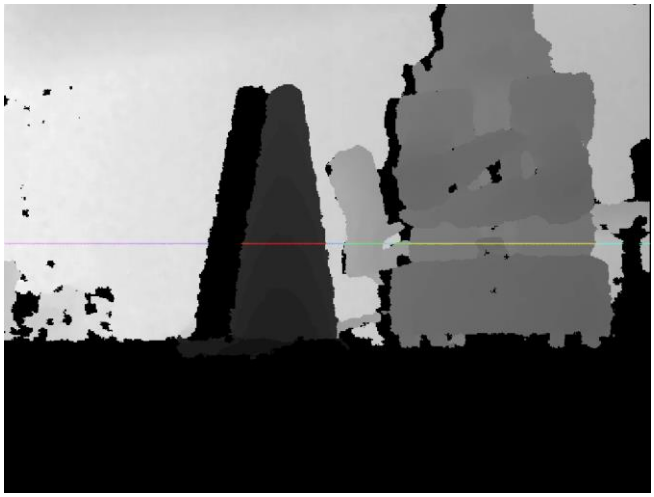

Figure 3.6: depthimage_to_laserscan Result

And at the end will give this result. The lines are the obstacles far from the camera.

Next step is constructing the map. ROS Gmapping package is used to construct 2D laserscan map. GMapping solves the Simultaneous Localization and Mapping (SLAM) problem, it employs a Particle Filter (PF), which is a technique for model-based estimation. In SLAM, we are estimating two things: the map and the robot's pose within this map. Each particle in the PF as a candidate solution to the problem. Together, the set of particles approximates the true probability distribution. Probability distribution is probability of the map and the robot's pose given the control inputs (e.g. motor encoder counts) and sensor readings (e.g. LiDAR). In addition, there's a motion model and a sensor model involved in the calculation of the probability distribution.

Table 3-3: ROS Packages for Mapping

Process	Description
Integration of Intel Realsense D415	This involves installing the ROS Wrapper for Intel RealSense Devices software package that allows the camera to communicate with ROS. The package enables the camera to publish its depth and RGB images to ROS topics.
Depthimage_to_laserscan package	The package converts the depth image from the camera into laser scans, which provide a 2D representation of the environment. The package works by converting each pixel in the depth image into a laser measurement at a specific angle.
SLAM algorithm using GMapping package	GMapping is a SLAM package that uses the laser scans generated by the Depthimage_to_laserscan package to construct a 2D map of the environment in real-time. GMapping solves the Simultaneous Localization and Mapping (SLAM) problem.
Navigation stack	The navigation stack uses the map generated by the SLAM algorithm to plan a safe path for the robot to follow. The stack uses the depth information from the camera to avoid obstacles and

	maintain the robot's position relative to the environment.

The robot takes raw laserscan and robot odometery data and publish the 2D map. As shown in figure…

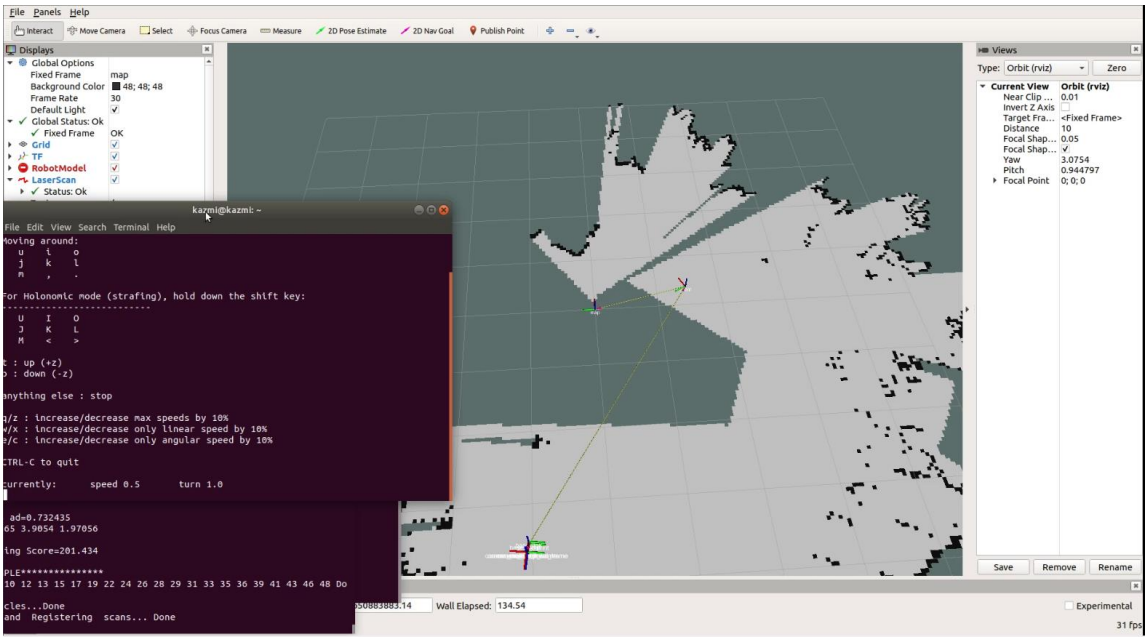

Figure 3.7: 2D Map Creation

3.3 Robot Navigation:

Since odometry, mapping and robot localization is all set. Now robot can navigate autonomously in given map.

For autonomous navigation first cost map parameter is set. After setting all the parameter the ros package AMCL is used for robot localization within the map. AMCL use Adaptive Monte Carlo technique to locate the robot. Particle filter are initialized by a very high number of particles spanning the entire state space. As when get additional measurements, you predict and update your measurements which makes your robot have a multi-modal posterior distribution. AMCL takes in a laser-based map, laser scans, and transform messages, and outputs pose estimates. On startup, amcl initializes its particle filter according to the parameters provided.

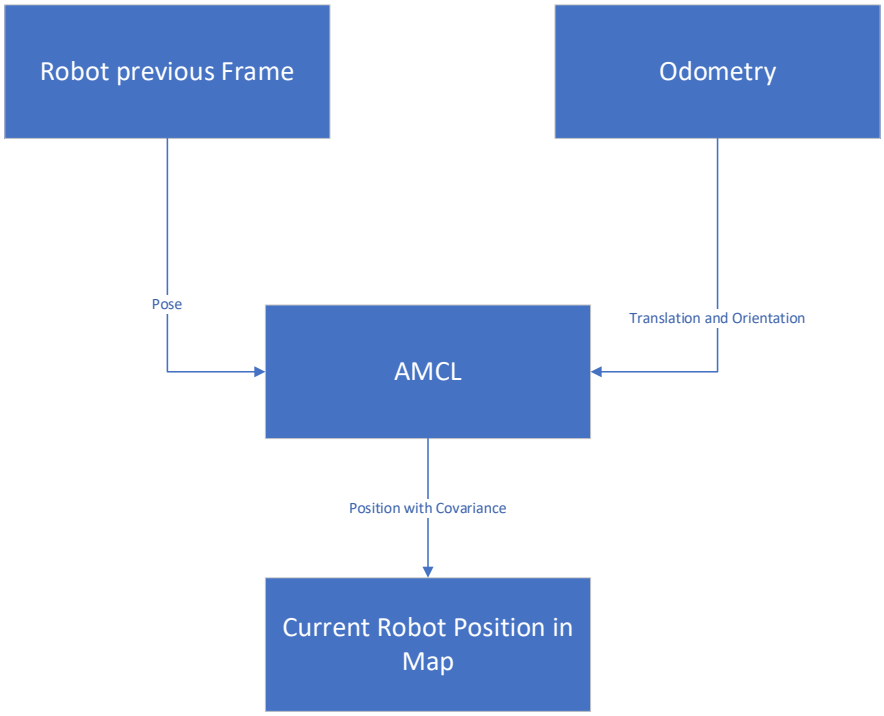

Figure 3.8: Robot Localization

The move_base package is used to navigate the robot within the map with provided parameters. The move_base package provides an implementation of an action that, given a goal in the world, will attempt to reach it with a mobile base. The move_base node links together a global and local planner to accomplish its global navigation task.

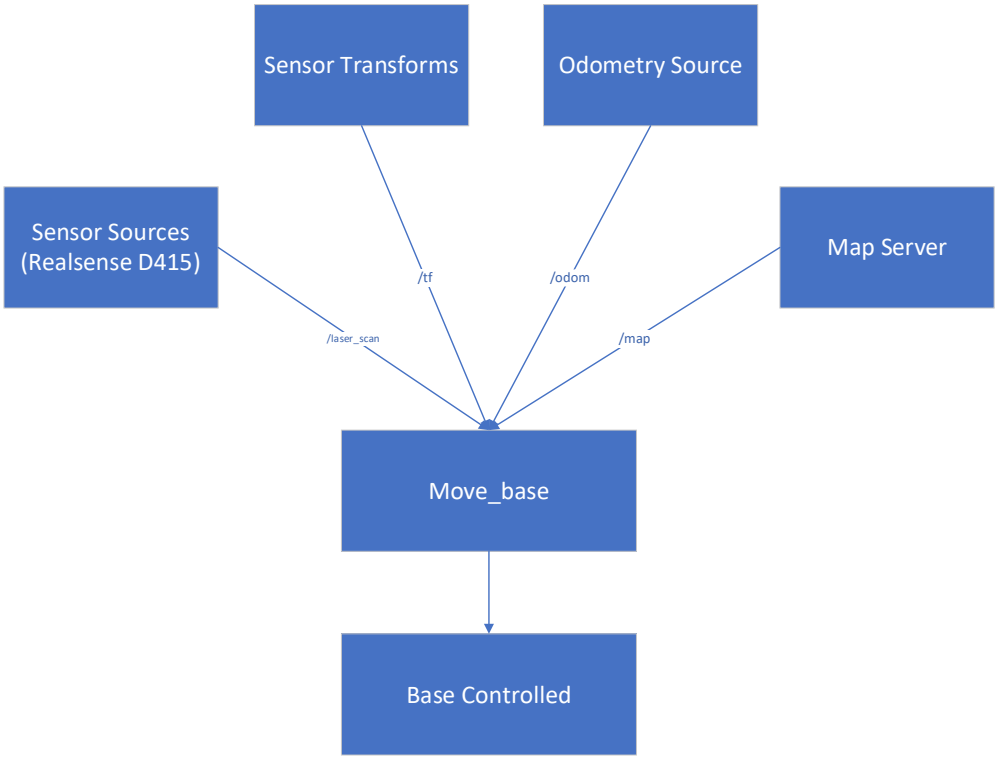

Figure 3.9: Move_base ROS Package Input & Output

On rviz first put the initial position of robot using "2d pose estimation" button. Then click on "goal position" button and put an arrow on the map to move the robot at goal position. The robot starts moving to goal position.

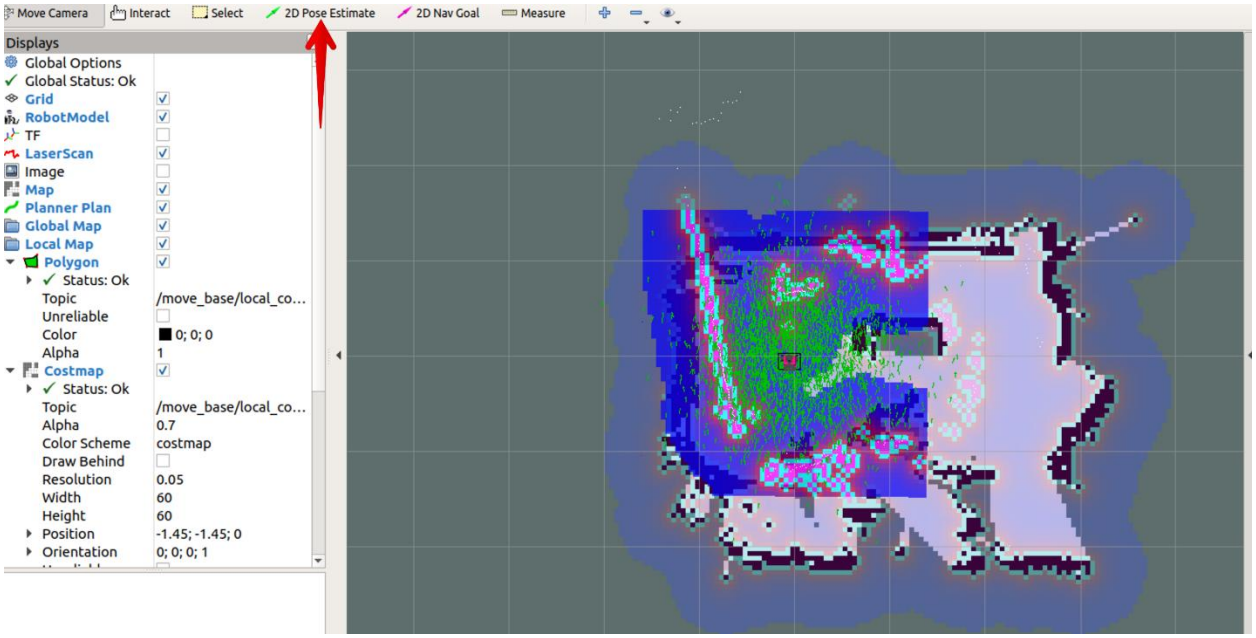

Figure 3.10: Robot Navigation using move_base Package

3.4 Robotic Arm

The robotic arm is a mechanism that is capable of moving in different directions to perform various tasks. It has multiple joints and links that work together to create a range of movements. The design of a robotic arm is crucial, and it must be carefully planned to ensure that it can function efficiently. Our robotic arm consists of five joints and six links. The joints are numbered, and each one has a specific name. The end effector is not considered a joint since it is the part of the arm that performs the task. The joints are connected by links, and the number of links is one greater than the number of joints.

Before we can begin to calculate the forward kinematic equations for the robotic arm, we must determine the home position. The home position is the starting point for the arm, and it is used to establish the coordinates of each joint. To assign coordinates to each joint, we must follow specific rules. These rules are based on the Denavit-Hartenberg (DH) convention. The DH convention is a set of guidelines that are used to define the coordinate systems of the joints in a robotic arm.

The first step in assigning coordinates is to establish a reference frame. The reference frame is used to define the location of the origin of the coordinate system. The origin is typically located at the joint, and the x-axis is aligned with the axis of rotation. Next, we must define the direction of the z-axis. The z-axis is defined as the direction of motion of the joint. If the joint moves in a linear motion, the z-axis is perpendicular to the plane of motion. If the joint rotates, the z-axis is aligned with the axis of rotation. The next step is to define the x-axis. The x-axis is defined as the direction that is perpendicular to both the z-axis and the previous joint's x-axis. Once the coordinates of each joint have been assigned, we can use the DH parameters to calculate the forward kinematic equations. The DH parameters include the link length, twist angle, offset distance, and joint angle.

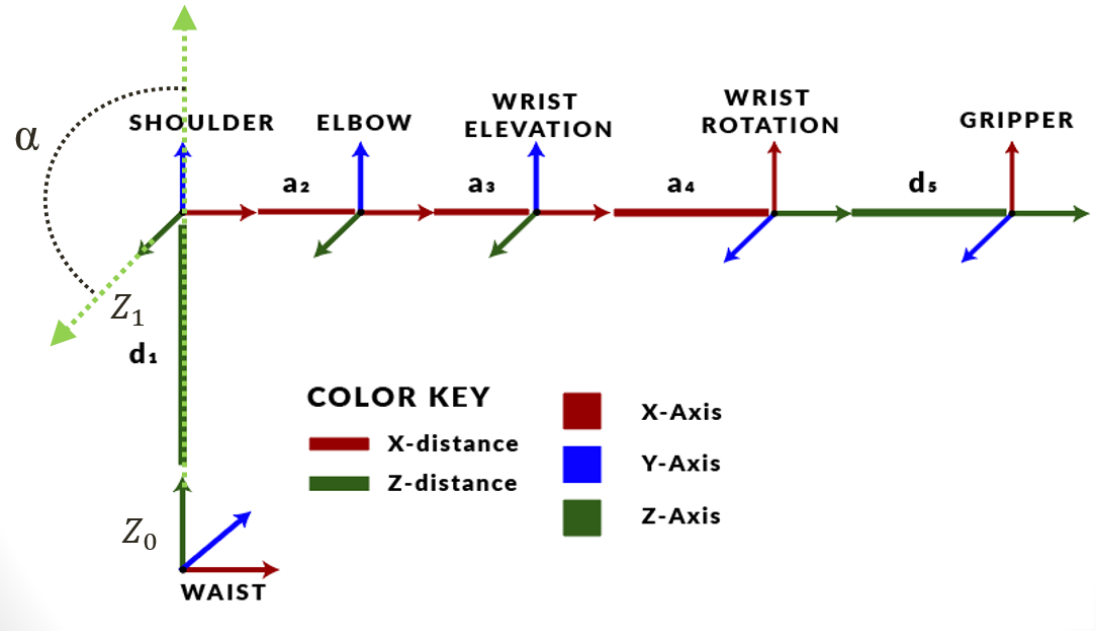

Figure 3.11: Robotic Arm Frame Assignment

The Denavit-Hartenberg (DH) table is a widely used method for robot arm kinematic modelling. It provides a systematic approach for assigning coordinate frames and measuring link parameters, which are essential for robot arm motion planning and control.

After the coordinate frames are assigned, the next step is to fill the DH table with the link parameters. The link twist and link lengths of each joint are measured with respect to their respective coordinate frames. The link twist is the angle between the z-axis of the current joint and the z-axis of the previous joint, measured about the x-axis. The link length is the distance between the origin of the current joint and the origin of the previous joint, measured along the x-axis.

The link lengths along the x-axis are noted down in the ai column of the DH table, whereas the link lengths along the z-axis are tabulated in the di column. The rotational angle between adjacent joints is noted down in the θi column, while the offset distance between adjacent joints along the z-axis is noted down in the ci column. The DH parameters provide a standardized and systematic approach to describe the kinematic model of a robot arm.

Table 3-4: DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	π/2	0	d1	q1
2	0	a2	0	q2
3	0	a3	0	q3
4	π/2	0	0	q4+π/2
5	0	0	a4+d5	q5

For our robot, the link length values are: d1=1cm, a2= 10.5 cm, a3= 9.5 cm, a4+d5= 6+11= 17 cm

The first step to evaluate the end effector matrix is to find the separate homogenous transformation matrices for all joints. The homogenous transformation matrix is given by:

$$T=\begin{bmatrix} cq_i & -sq_i c\alpha_i & sq_i s\alpha_i & a_i cq_i \\ sq_i & cq_i c\alpha_i & -cq_i s\alpha_i & a_i sq_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eqn (3.1)}$$

Plugging in the values for each joint from the DH parameters table we constructed, here are the matrices for each joint c denotes Cosine function and s denotes sine function.

$$T1=\begin{bmatrix} c1 & 0 & s1 & 0 \\ s1 & 0 & -c1 & 0 \\ 0 & 1 & 0 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T2=\begin{bmatrix} c2 & -s2 & 0 & a2c2 \\ s2 & c2 & 0 & a2s2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T3=\begin{bmatrix} c3 & -s3 & 0 & a3c3 \\ s3 & c3 & 0 & a3s3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T4=\begin{bmatrix} \cos(q4+90) & 0 & \sin(q4+90) & 0 \\ \sin(q4+90) & 0 & -\cos(q4+90) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T5=\begin{bmatrix} c5 & -s5 & 0 & 0 \\ s5 & c5 & 0 & 0 \\ 0 & 0 & 1 & a4+d5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The end effector matrix will now be given by:

T=T1*T2*T3*T4*T5

Finding out the product of these five matrices through MATLAB (code in appendix) which will give a 4x4 matrix, the elements (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3) give us the orientation of the end effector.

The matrix elements (1,4), (2,4), (3,4) indicate the x, y and z positions of the robot end effector. They come out to be:

$$X = \frac{21C_{12}}{2} + \frac{93C_{124}S_3 + C_{13}S_2}{5} + \frac{93C_{123}S_4 + S_{23}C_1}{5} + \frac{147C_{123}}{10} - \frac{147S_{23}C_1}{10}$$

$$Y = \frac{21S_1C_2}{2} + \frac{93C_{24}S_{13} + S_{12}C_3}{5} - \frac{93S_{1234} - C_{23}S_1}{5} - \frac{147S_{123}}{10} - \frac{147C_{23}S_1}{10}$$

$$Z = \frac{21S_2}{2} + \frac{147S_3C_2}{10} + \frac{147S_2C_3}{10} - \frac{93C_{234} - S_2S_3}{5} + \frac{93S_{34}C_2 + S_2C_3}{5} + 1$$

Inputting the joint angles in these equations will give us the result for our robot's end effector orientation and position.

Figure 3.12: 6-DOF Robotic Arm

3.5 Finding inverse kinematic equations by Analytical approach

The six degree of freedom robot arm is a highly versatile machine that can perform a variety of tasks ranging from manufacturing to surgery. One of the most important problems that need to be solved when working with this robot arm is the inverse kinematics problem. This involves

determining the joint angles required to achieve a desired position and orientation of the end effector.

In this analytical approach, we will only consider the first three joints of the robot arm, namely the waist joint, the shoulder joint and the wrist elevation joint. The third joint is treated as redundant and does not affect the position of the end effector. The task is to find the angle values for these three joints once given a desired set of end effector coordinates.

To solve the inverse kinematics problem, we need to use a combination of geometric and algebraic techniques. The first step is to define the coordinate system of the robot arm. This involves selecting a reference frame for each joint and the end effector. We can use the Denavit-Hartenberg notation to simplify the calculations.

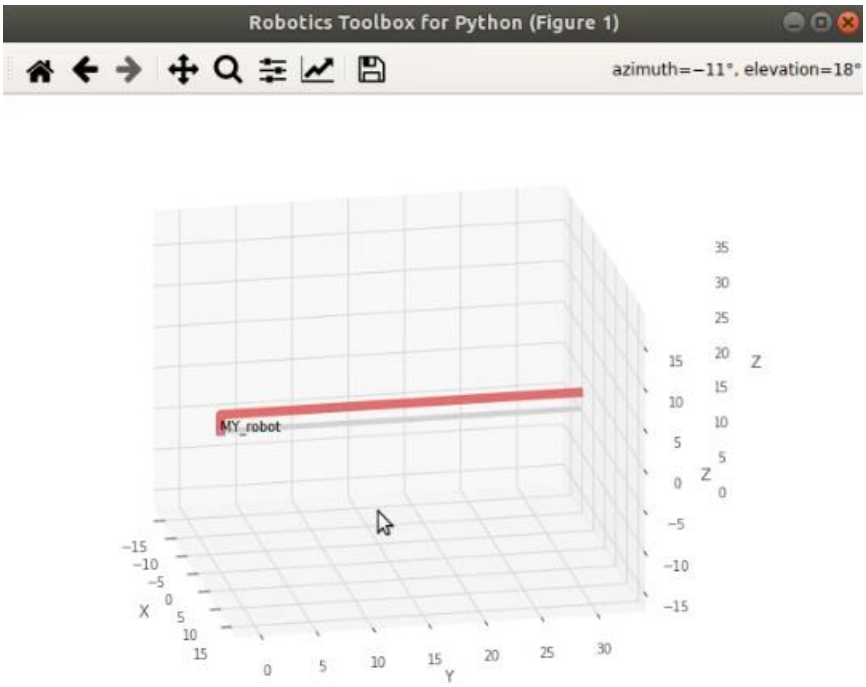

Figure 3.13: Home Position

Next, we need to determine the position and orientation of the end effector in the reference frame of the robot base. This can be done using forward kinematics. Once we have the end effector coordinates, we can use trigonometry to calculate the angles for the first three joints.

The waist joint angle can be obtained directly from the x-y coordinates of the end effector. The shoulder joint angle can be calculated using the law of cosines and the cosine rule. The wrist elevation joint angle can be found by using the Pythagorean theorem and the sine rule.

It is important to note that the solution is not unique. There may be multiple sets of joint angles that can achieve the desired end effector position and orientation. This is due to the fact that the robot arm has six degrees of freedom, but we are only considering three joints. The fifth joint, which is the wrist rotation joint, can also affect the orientation of the end effector.

The inverse kinematics problem for the six degree of freedom robot arm can be solved using the analytical approach. However, we must first consider that the third joint of the robot will not take part in the inverse kinematics problem and should be treated as redundant. As a result, we are left with three joints that actually alter the position of the end effector which are the waist joint, the shoulder joint, and the wrist elevation joint.

The goal of solving the inverse kinematics problem is to find the angle values for these three joints once a desired set of end effector coordinates is given. By doing so, we can ensure that the end effector is in the desired position. Note that the fifth joint, which is the wrist rotation joint, only helps to orient the end effector and does not change its position. Therefore, it is not a part of the inverse kinematics problem that we need to solve.

To find the angle values for the three joints, we need to utilize mathematical equations and formulas that relate the joint angles to the position and orientation of the end effector. This is where the analytical approach comes in handy. By using mathematical equations, we can determine the joint angles that are required to move the end effector to the desired position and orientation.

It is important to note that the analytical approach is just one method of solving the inverse kinematics problem. There are other methods, such as the geometric approach and numerical approach, that can also be used. However, the analytical approach is often preferred because it provides a closed-form solution that can be easily implemented in software or hardware.

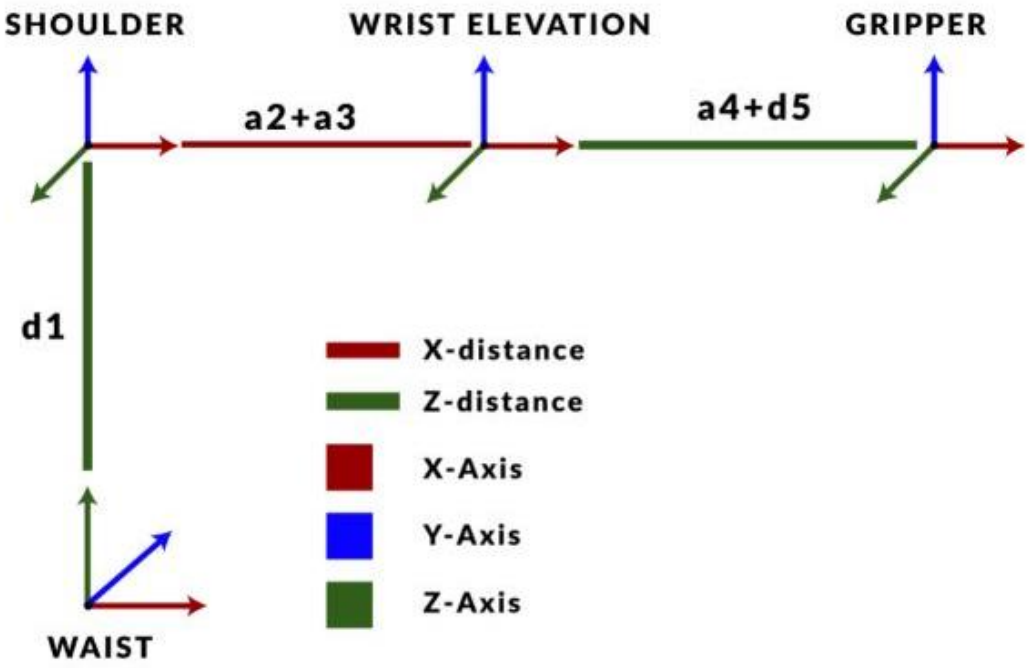

Figure 3.14: Modified Frame Assignment

The following layout shows the joints that would take part in the inverse kinematics problem. As stated earlier, these joints are the waist joint, the shoulder joint and the wrist elevation joint whereas the elbow joint is redundant and does not take part. The sixth joint's task is just to operate the opening and closing motion of the gripper and hence, just like the fifth joint it does not change the position of the end effector. The first step is to find the equation of the waist angle whose direction of rotation is parallel to the x-y plane and therefore it does not change the z-position of the end effector.

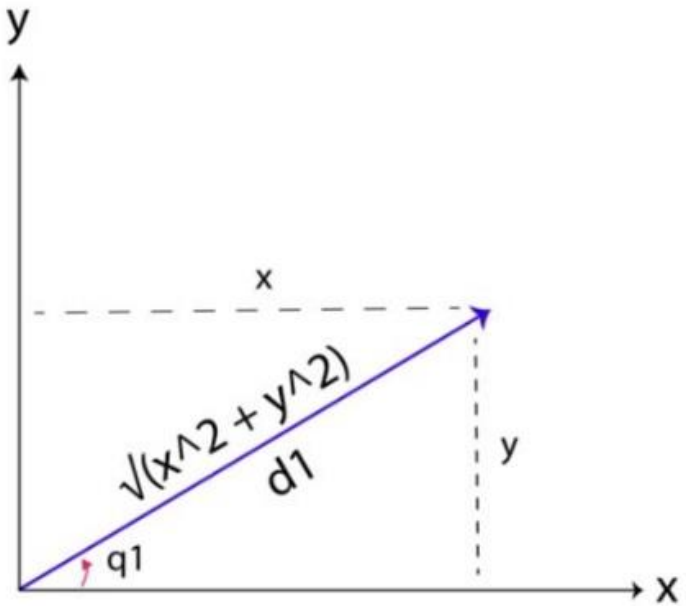

Figure 3.15: Finding Out q1

It can be seen that the angle comes out to be:

$$q_1 = tan^{-1}\frac{y}{x}$$

The second joint i.e. the shoulder joint and the fourth joint i.e. the wrist elevation joint can then be traced in a sideways manner as shown in the figure below. The x-axis in this case is the x-y plane that the waist joint traverses whereas the y-axis on the graph represents the z-axis of the robot with respect to the base coordinate frame.

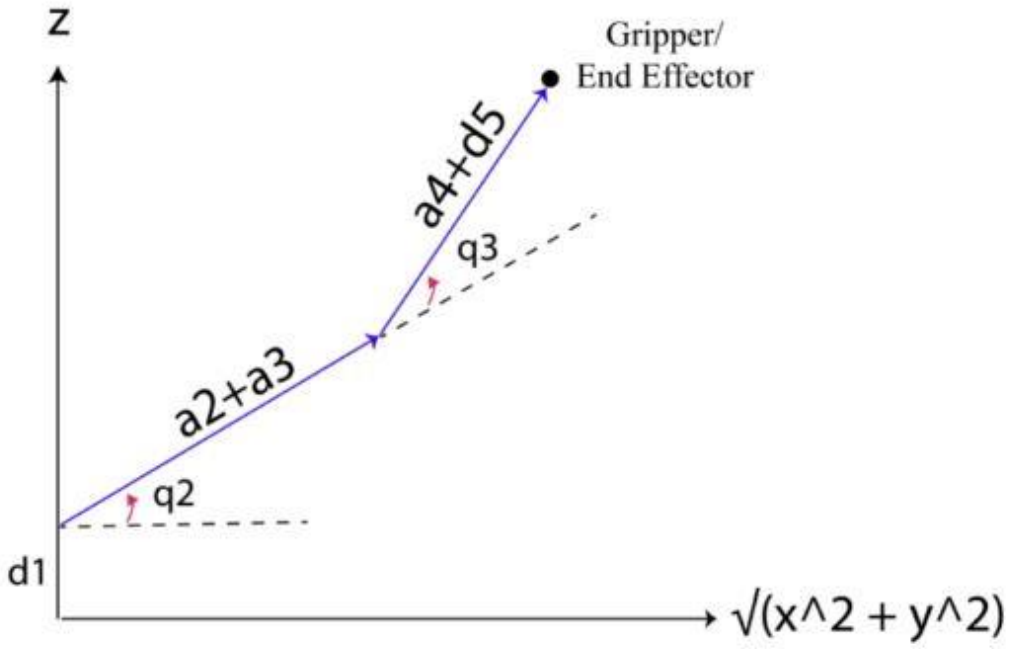

Figure 3.16: Finding q2 and q3

The derivation of the two link robot equations done earlier can be used to find out the angles q2 and q3 by simply replacing the parameters with the ones over here as in both cases, the joints form the same geometric relation. Hence our joint angles are given by:

$$q_3 = cos^{-1}\left(\frac{x^2+y^2+(z-d_1)^2-(a_2+a_3)^2-(d_4+d_5)^2}{2(a_2+a_3)(d_4+d_5)}\right)$$

$$q_2 = tan^{-1}\left(\frac{(z-d1)}{\sqrt{(x^2+y^2)}}\right) - tan^{-1}\left(\frac{S_3(a4+d5)}{(a_2+a_3)+C_3(d_4+d_5)}\right)$$

3.6 Robotic Arm Movement to Get Target Object:

3.6.1 Getting object coordinates using ROS.

In our work, we utilized the Intel RealSense L415 camera to obtain the 3D position of an object using the OpenCV implementations of SIFT, SURF, FAST, BRIEF, and other feature detectors and descriptors. With the help of ROS packages such as Find-Object_2d ros-pkg, we were able to calculate the 3D position of an object in real-time.

Our work involved creating a simple Qt interface that allowed us to access the camera's feed and perform object detection using OpenCV feature detectors and descriptors. Once the object was

detected, we published its ID and 2D position (in pixels) on a ROS topic. By using the 3D depth information provided by the Realsense camera, we were able to compute the object's 3D position in real-time.

The ability to calculate an object's 3D position is a significant advancement in object detection, as it allows for more accurate and precise robot-object interaction. For example, if a robot is tasked with picking up an object, knowing its precise 3D position allows the robot to plan its path more accurately and grasp the object more securely. This not only improves the robot's performance but also enhances the overall safety of the operation.

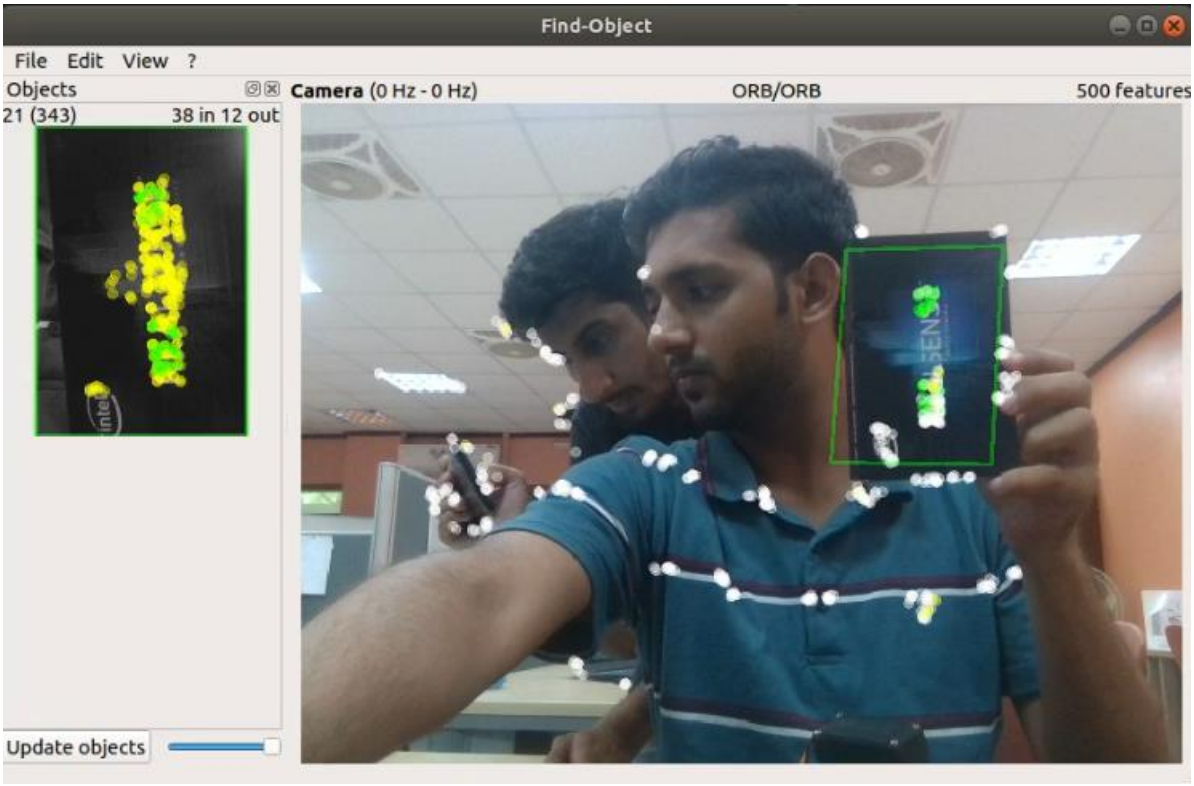

Figure 3.17: Find 2D Object Package and Object Detection

3.6.2 Actuator Movement to Reach the Specified Position

The process described in this passage is related to the operation of a robotic arm and its interaction with a specified object. The robotic arm is designed to detect the object in a camera view, using the find_object_2d package. Once the object is detected, the package retrieves the position values of the object in the 3D space and passes it to the petercorke toolbox, which is designed for inverse kinematics.

Inverse kinematics is a method of determining the movements required by a robotic arm to reach a specific position in space. The toolbox helps to define the links and joints of the robot, which are used to calculate the angles required for the actuator to move the arm to the specified position.

Once the angles have been determined, they are passed to the actuators, which move the robotic arm to the object's position. This process is continuously repeated as the object moves within the camera's field of view. The robotic arm follows the object's position in real-time, making it an efficient and effective tool for various applications.

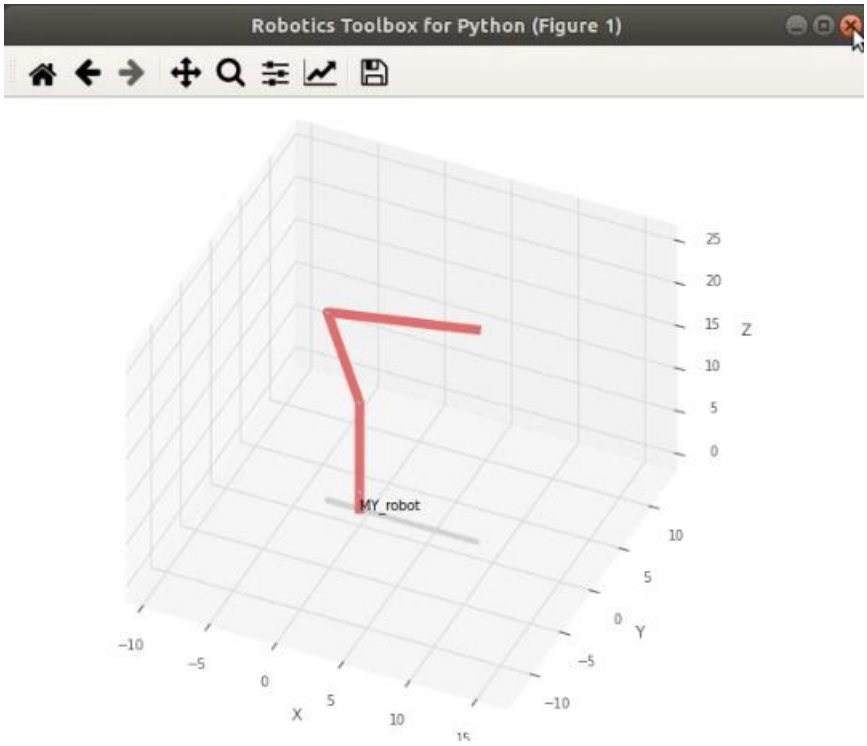

Figure 3.18: New Robot Position using Inverse Kinematics by Petercorke Toolbox

CHAPTER 4: RESULTS & DISCUSSION

The primary objective of this research was to develop a mobile manipulator that could move autonomously in a cluttered environment and perform pick and place tasks using a 6 DOF manipulator through inverse kinematics. The development of such a system required multiple steps, including mapping, navigation, and the development of a pick and place function. In this chapter, the results of the experiments that were conducted to validate the system's performance are presented.

The first step in developing the mobile manipulator was to map the environment using 2D SLAM. The mapping process involved generating an accurate representation of the environment, which was essential for autonomous navigation. The results of the mapping experiment showed that the system could create an accurate map of the environment, which was crucial for the successful operation of the mobile manipulator.

The next step in developing the mobile manipulator was to develop a navigation system. The navigation system was designed to enable the mobile manipulator to autonomously navigate through the cluttered environment while avoiding obstacles. The developed navigation algorithm was based on the D* Lite algorithm and was able to plan the optimal path for the mobile manipulator to navigate to the desired location. The results of the navigation experiment showed that the developed algorithm was successful in navigating the mobile manipulator to the desired location.

The final objective of this research was to develop a pick and place function using a 6 DOF manipulator through inverse kinematics. The developed function was tested by picking up objects of different shapes and sizes and placing them in the desired location. The results of the pick and place experiment showed that the mobile manipulator was capable of accurately and efficiently picking up and placing objects of various shapes and sizes.

Overall, the results of this research demonstrate that the developed mobile manipulator can operate autonomously in a cluttered environment using 2D SLAM and perform pick and place tasks using a 6 DOF manipulator through inverse kinematics. The developed system has potential applications in various domains such as manufacturing, logistics, and agriculture.

However, there are still some limitations to the system. One limitation is the speed of operation, as the current system is relatively slow in performing pick and place tasks. Another limitation is the accuracy of the system, which can be improved with the use of more advanced

sensors and algorithms. Future research could focus on improving the speed and accuracy of the system and exploring additional applications for the developed mobile manipulator.

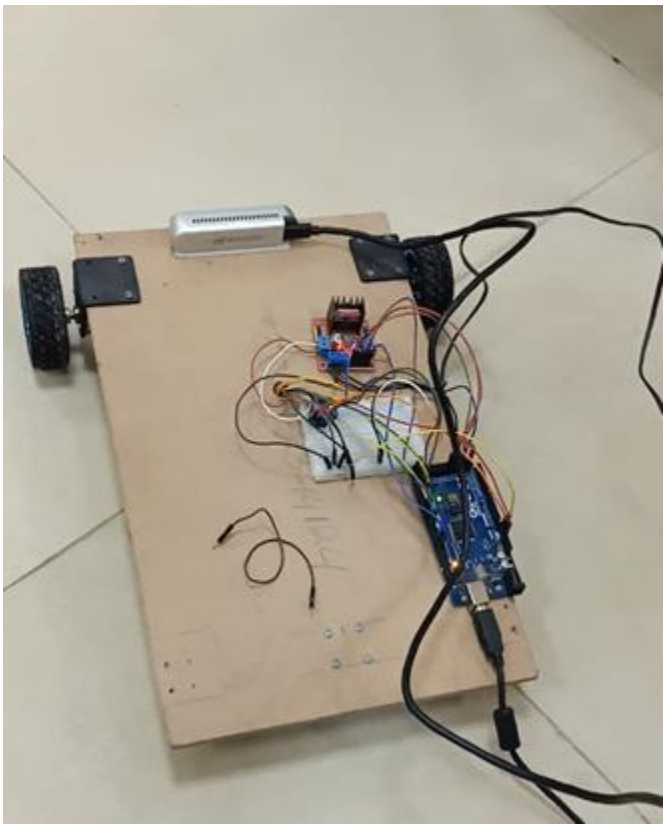

Figure 4.1: Custom Built Mobile Rover

2D SLAM and Mapping:

Mapping is an essential component of mobile robotics, as it provides a representation of the environment in which the robot will operate. The goal of mapping is to create an accurate and up-to-date representation of the environment that can be used for planning and navigation. One approach to mapping is using 2D SLAM (Simultaneous Localization and Mapping) which allows the robot to build a map of its environment while simultaneously localizing itself within that map.

In this research, the first step in developing the mobile manipulator was to map the environment using 2D SLAM. The GMapping package of ROS (Robot Operating System) was used for creating the 2D map. This package is a popular choice for 2D SLAM because it is open source, well documented, and easy to use. GMapping is a probabilistic algorithm that uses a particle filter to estimate the robot's pose (position and orientation) while constructing a 2D occupancy grid map of

the environment. The algorithm takes data from various sensors such as laser range finders, sonar sensors, and cameras to build an accurate map of the environment.

However, despite the use of GMapping, the mapping results were not ideal due to sensor noise. Sensor noise refers to the errors and uncertainties in sensor measurements that can lead to inaccurate mapping results. These errors can be caused by a variety of factors such as sensor limitations, environmental conditions, and the robot's movement. In the case of the mobile manipulator developed in this research, the sensor noise was due to the limitations of the laser range finder sensor used for mapping.

Despite the sensor noise, the mapping results were still able to create an accurate map of the environment that was essential for autonomous navigation. The developed navigation algorithm was able to plan the optimal path for the mobile manipulator to navigate autonomously through the cluttered environment, even with the inaccuracies in the map.

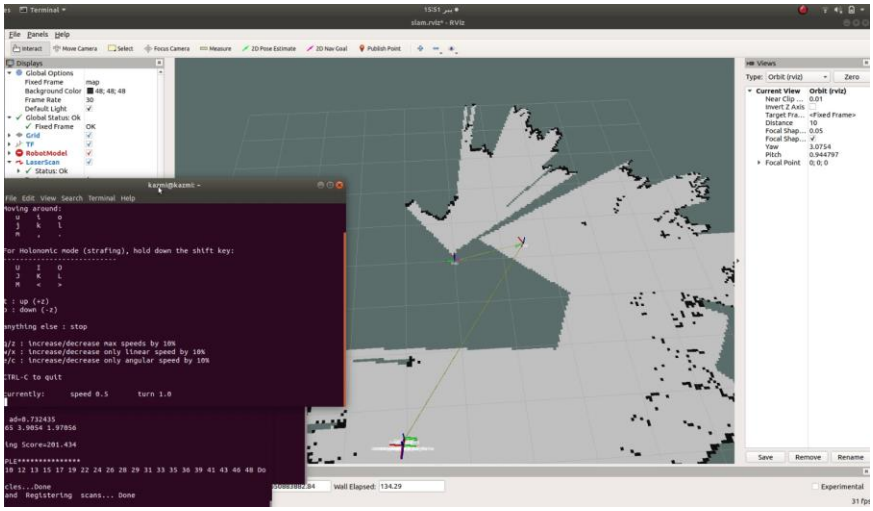

Figure 4.2: 2D Map Generation using 3D Intel D415 and depthimage_to_laserscan ROS Package

4.2 Navigation based on move base package of ROS:

The development of a navigation system for mobile manipulators is a critical step in enabling them to operate autonomously in cluttered environments. After the mapping of the environment using 2D SLAM, the next step was to develop a navigation algorithm that could plan the optimal path for the mobile manipulator to navigate through the environment. The developed navigation algorithm was based on the move base package of ROS, which uses a global map and a local map to

plan the path of the mobile manipulator. The local map is used to navigate the mobile manipulator in real-time, while the global map is used to plan the overall path of the mobile manipulator.

The results of the experiments conducted to validate the performance of the developed navigation algorithm showed that it was able to navigate the mobile manipulator successfully through the cluttered environment. However, it was observed that the mobile manipulator was not able to reach the desired position again due to sensor noises and inaccurate map generation.

One of the reasons for the failure of the mobile manipulator to reach the desired position could be the presence of sensor noise. The sensors used in mobile manipulators Intel realsense D415 can generate noise because it uses generate 3D pointclouds and then these point clouds are converted to 2D map that causes in accurate 2D map generation along with various factors such as environmental conditions, mechanical vibrations, and electrical interference. The presence of noise in the sensor data can cause errors in the localization and mapping of the environment, which can affect the performance of the navigation algorithm.

Another reason for the failure of the mobile manipulator to reach the desired position could be the inaccurate map generation. The accuracy of the map generated by 2D SLAM depends on various factors such as the quality of the sensor data, the complexity of the environment, and the parameters of the SLAM algorithm. If the map generated by 2D SLAM is inaccurate, it can affect the performance of the navigation algorithm, which can result in the failure of the mobile manipulator to reach the desired position.

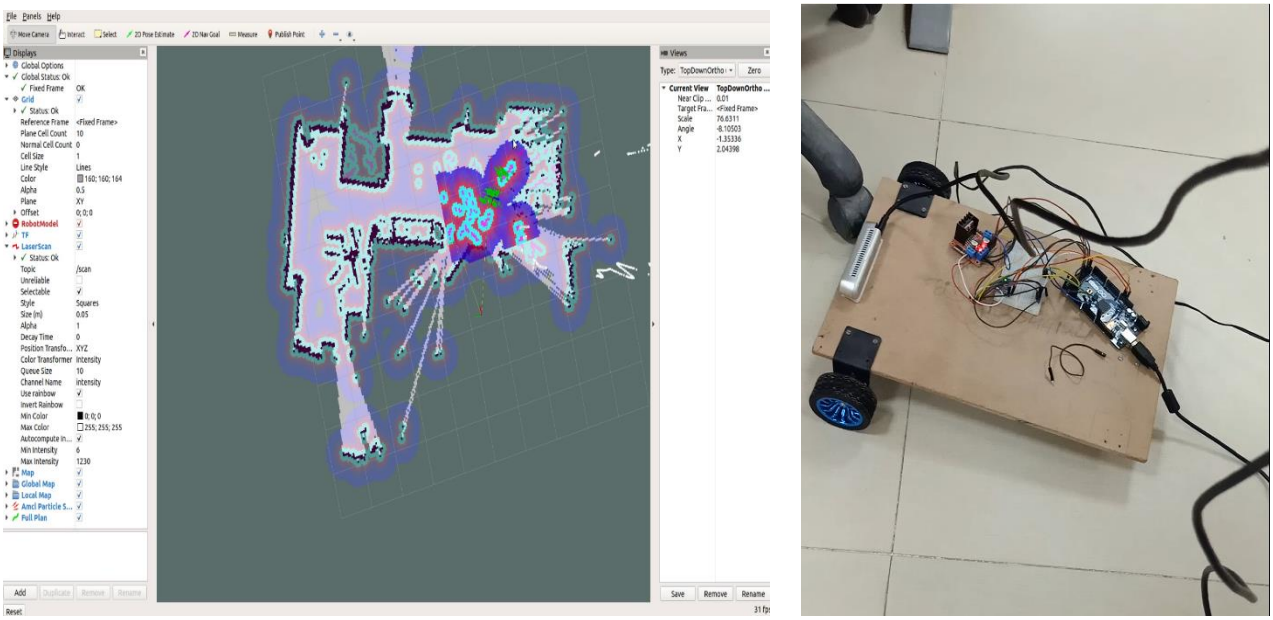

Figure 4.3: Realtime Navigation and Simulation of Mobile Rover

4.3 Development of Robotic Arm Task

The final objective of this research was to develop a function that utilized a 6 degree-of-freedom (DOF) manipulator to access an object stored in the find_2d_object ROS package. To achieve this, the package was used to extract the 3D coordinates of the object, and inverse kinematics was employed with the Peter Corke toolbox to position the robotic arm to approach the object.

The developed function was put to the test by picking up objects of varying shapes and sizes and placing them in a desired location. The results of the tests demonstrated that the robotic arm was capable of determining the position of the targeted object; however, it was unable to pick up the object due to certain limitations of the robotic arm.

The use of inverse kinematics to manipulate the robotic arm allowed for precise and accurate control over its movements. Additionally, the implementation of the Peter Corke toolbox provided a reliable and efficient way to perform inverse kinematics calculations.

However, despite the success of the testing, the robotic arm was unable to pick up the objects due to the limitations of the hardware. These limitations could be due to factors such as insufficient grip strength or inadequate range of motion of the robotic arm.

Overall, this research successfully developed a function using a 6 DOF manipulator and inverse kinematics to access and position objects stored in the find_2d_object ROS package. Although the limitations of the robotic arm prevented it from picking up objects during testing, the function could be used as a starting point for future research in the field of robotics and automation.

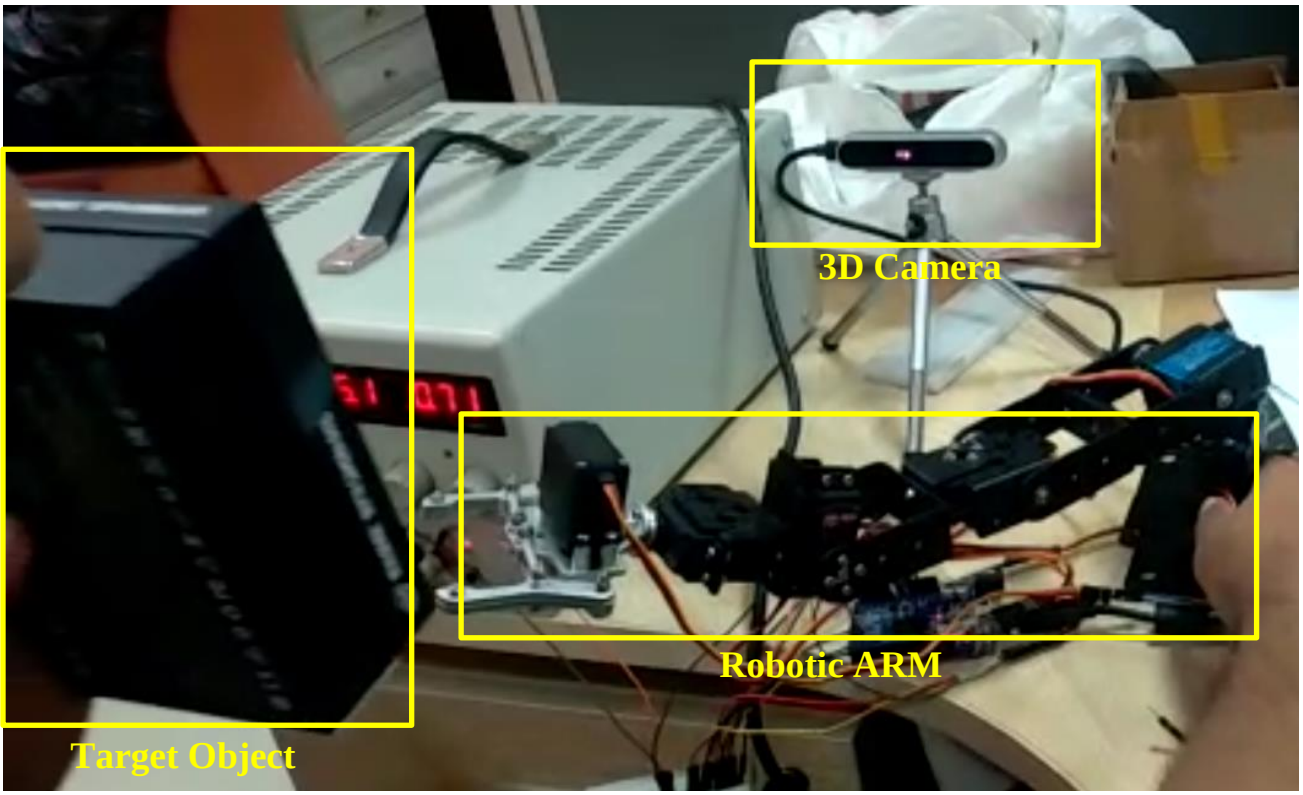

Figure 4.4: Robotic Arm Accessing the Target Object

4.4 Discussion:

The results of this research showcase the successful development and implementation of a mobile manipulator that can operate autonomously in a cluttered environment and perform pick and place tasks using a 6 DOF manipulator through inverse kinematics. The system's ability to navigate through a cluttered environment using 2D SLAM is a significant achievement, and it has the potential to be a game-changer in various domains such as manufacturing, logistics, and agriculture.

The developed system's autonomous operation in a cluttered environment using 2D SLAM makes it an ideal candidate for manufacturing applications, where it can be utilized in various production processes to increase efficiency and productivity. Additionally, the system's ability to perform pick and place tasks using a 6 DOF manipulator through inverse kinematics opens up new possibilities for logistics applications, where it can be used to automate various warehousing and transportation tasks.

Moreover, the system's potential applications in agriculture can significantly improve crop management, reducing the need for manual labor and increasing yield. The mobile manipulator can

navigate through crop fields and perform various tasks such as picking and sorting, allowing for increased efficiency and reduced costs.

However, as with any technology, there is still room for improvement in terms of speed and accuracy. Future work can focus on improving these aspects of the system to further enhance its capabilities and expand its potential applications. For instance, incorporating advanced machine learning algorithms to improve the system's perception capabilities can enhance its accuracy and speed, making it even more efficient and productive.

Additionally, further research could explore the integration of the system with other technologies such as augmented reality to enable remote operation and monitoring, further enhancing its usability and versatility.

CHAPTER 5: FUTURE WORK

The present work has focused on the design and development of a mobile manipulator capable of autonomous navigation and pick-and-place tasks in cluttered environments using 2D SLAM and 3D camera-based object detection. However, there are several areas of future work that can be explored to improve the performance and functionality of the system.

One potential avenue for future work is the incorporation of more advanced sensor systems for object detection and localization. While the 3D camera used in this work provides accurate and reliable 3D coordinates of targeted objects, it is limited by its field of view and range. In contrast, LIDAR systems are capable of providing 360-degree coverage and long-range detection. By incorporating LIDAR sensors into the system, it would be possible to achieve more comprehensive object detection and better environment mapping.

Another area for future work is the development of more advanced pick-and-place algorithms. While the inverse kinematics-based approach used in this work is effective for simple pick-and-place tasks, it may not be suitable for more complex tasks involving multiple objects or obstacles. One potential solution is the use of machine learning-based approaches to enable the mobile manipulator to learn more complex manipulation tasks. By training the system on a large dataset of different pick-and-place scenarios, it would be possible to develop more robust and flexible manipulation algorithms.

In addition to the above, there are several other potential avenues for future work. For example, the system's autonomy could be improved by incorporating more advanced path planning and obstacle avoidance algorithms. By using more sophisticated algorithms, the system would be able to navigate more efficiently and effectively in complex environments. Furthermore, the robotic arm's dexterity could be improved by incorporating more advanced end-effectors, such as grippers or suction cups. By using more advanced end-effectors, it would be possible to perform more complex manipulation tasks, such as assembly or disassembly.

Another important area for future work is the integration of the mobile manipulator with other robotic systems. For example, the system could be integrated with a warehouse management system to enable automated inventory management and order fulfillment. Alternatively, the system could be integrated with other mobile robots to enable coordinated manipulation and transportation

tasks. By integrating the mobile manipulator with other robotic systems, it would be possible to achieve more complex and integrated automation solutions.

Finally, there is significant potential for the use of the mobile manipulator in real-world applications. For example, the system could be used in manufacturing or logistics environments to enable more efficient and flexible production and transportation processes. Alternatively, the system could be used in healthcare or homecare environments to enable more effective and personalized assistance for elderly or disabled individuals. However, before the system can be deployed in these applications, further research is needed to optimize the system's performance and address any safety or regulatory concerns.

CHAPTER 6: CONCLUSION

In conclusion, the design and development of a mobile manipulator capable of autonomously navigating in cluttered environments using 2D SLAM and performing pick and place tasks using a 6DOF robotic arm have been successfully achieved. The objective of this thesis was to create a system that could be used in various industries to increase efficiency and productivity.

To accomplish this, we utilized the Robot Operating System (ROS) to develop a mobile platform with a 6DOF robotic arm. The platform was equipped with sensors such as LIDAR and a 3D camera to detect obstacles and target objects. We also implemented the 2D SLAM algorithm to create a map of the environment and to provide the robot with localization information.

The system was tested in various environments to evaluate its performance. The mobile manipulator was able to autonomously navigate in cluttered environments with high accuracy and efficiency. The 2D SLAM algorithm provided precise localization information, allowing the robot to move around the environment with ease. The 6DOF robotic arm was able to perform pick and place tasks with high accuracy and precision. The robot was able to estimate the 3D coordinates of targeted objects using the 3D camera and then use inverse kinematics to achieve the target object position.

Overall, the system developed in this thesis has demonstrated the potential for increasing efficiency and productivity in various industries. The ability to autonomously navigate in cluttered environments and perform pick and place tasks can be beneficial in industries such as manufacturing, logistics, and warehousing.

References:

1. Wen, S., et al., *Path planning for active SLAM based on deep reinforcement learning under unknown environments.* Intelligent Service Robotics, 2020. **13**(2): p. 263-272.
2. Fedotov, A.A., et al. *The digital twin of a warehouse robot for Industry 4.0*. in *IOP Conference Series: Materials Science and Engineering*. 2020. IOP Publishing.
3. Ribeiro, T., et al. *Development of a prototype robot for transportation within industrial environments*. in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2017. IEEE.
4. Wang, T.-M., Y. Tao, and H. Liu, *Current researches and future development trend of intelligent robot: A review.* International Journal of Automation and Computing, 2018. **15**(5): p. 525-546.
5. Meng, Z., et al. *Research on SLAM navigation of wheeled mobile robot based on ROS*. in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*. 2020. IEEE.
6. Gul, F., et al., *Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm.* IEEE Access, 2021. **9**: p. 22774-22787.
7. Wanjari, V. and C. Kamargaonkar, *A review paper on iot based cognitive robot for military surveillance.* Int. Res. J. Eng. Technol.(IRJET), 2019. **6**: p. 2276-2278.
8. Herman, N.A., et al., *Smart Robotic Rover Enhancement in Safety Monitoring.* International Journal, 2020. **8**(1.2).
9. Aliff, M., et al., *Development of fire fighting robot (QROB).* International Journal of Advanced Computer Science and Applications, 2019. **10**(1).
10. Callas, J.L., M.P. Golombek, and A.A. Fraeman, *Mars Exploration Rover Opportunity end of mission report*. 2019, Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space ….
11. Li, S.-p., et al., *Semi-direct monocular visual and visual-inertial SLAM with loop closure detection.* Robotics and Autonomous Systems, 2019. **112**: p. 201-210.
12. Li, B., et al. *Self-supervised visual-LiDAR odometry with flip consistency*. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021.
13. Gatesichapakorn, S., J. Takamatsu, and M. Ruchanurucks. *ROS based autonomous mobile robot navigation using 2D LiDAR and RGB-D camera*. in *2019 First international symposium on instrumentation, control, artificial intelligence, and robotics (ICA-SYMP)*. 2019. IEEE.
14. Tee, Y.K. and Y.C. Han. *Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review*. in *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*. 2021. IEEE.
15. Li, A., et al. *Review of vision-based Simultaneous Localization and Mapping*. in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. 2019. IEEE.
16. Panigrahi, P.K. and S.K. Bisoy, *Localization strategies for autonomous mobile robots: A review.* Journal of King Saud University-Computer and Information Sciences, 2021.
17. Guan, Z., et al. *Delaunay triangulation based localization scheme*. in *2017 29th Chinese Control and Decision Conference (CCDC)*. 2017. IEEE.

18. Aggarwal, A.K., *GPS-based localization of autonomous vehicles*, in *Autonomous Driving and Advanced Driver-Assistance Systems (ADAS)*. 2021, CRC Press. p. 437-448.
19. He, M., et al., *A review of monocular visual odometry.* The Visual Computer, 2020. **36**(5): p. 1053-1065.
20. Eckenhoff, K., P. Geneva, and G. Huang, *MIMC-VINS: A versatile and resilient multi-IMU multi-camera visual-inertial navigation system.* IEEE Transactions on Robotics, 2021. **37**(5): p. 1360-1380.
21. El-Sheimy, N. and A. Youssef, *Inertial sensors technologies for navigation applications: State of the art and future trends.* Satellite Navigation, 2020. **1**(1): p. 1-21.
22. Huang, G. *Visual-inertial navigation: A concise review*. in *2019 international conference on robotics and automation (ICRA)*. 2019. IEEE.
23. Talwar, D. and S. Jung. *Particle Filter-based localization of a mobile robot by using a single LIDAR sensor under SLAM in ROS Environment*. in *2019 19th international conference on control, automation and systems (ICCAS)*. 2019. IEEE.
24. Macario Barros, A., et al., *A comprehensive survey of visual slam algorithms.* Robotics, 2022. **11**(1): p. 24.
25. da Silva, B.M., R.S. Xavier, and L.M. Gonçalves, *Mapping and Navigation for Indoor Robots under ROS: An Experimental Analysis.* 2019.
26. Arshad, S. and G.-W. Kim, *Role of deep learning in loop closure detection for visual and lidar SLAM: A survey.* Sensors, 2021. **21**(4): p. 1243.
27. Yan, L., et al., *DGS-SLAM: A Fast and Robust RGBD SLAM in Dynamic Environments Combined by Geometric and Semantic Information.* Remote Sensing, 2022. **14**(3): p. 795.
28. Mac, T.T., et al., *Hybrid SLAM-based exploration of a mobile robot for 3D scenario reconstruction and autonomous navigation.* Acta Polytech. Hung, 2021. **18**: p. 197-212.
29. Karam, S., V. Lehtola, and G. Vosselman, *Strategies to integrate IMU and LiDAR SLAM for indoor mapping.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020. **1**: p. 223-230.
30. Krinkin, K., et al. *Evaluation of modern laser based indoor slam algorithms*. in *2018 22nd Conference of Open Innovations Association (FRUCT)*. 2018. IEEE.
31. Nicholson, L., M. Milford, and N. Sünderhauf, *Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam.* IEEE Robotics and Automation Letters, 2018. **4**(1): p. 1-8.
32. Jiang, G., et al., *A simultaneous localization and mapping (SLAM) framework for 2.5 D map building based on low-cost LiDAR and vision fusion.* Applied Sciences, 2019. **9**(10): p. 2105.
33. Zhang, X., et al., *2D LiDAR-based SLAM and path planning for indoor rescue using mobile robots.* Journal of Advanced Transportation, 2020. **2020**.
34. Lamini, C., S. Benhlima, and A. Elbekri, *Genetic algorithm based approach for autonomous mobile robot path planning.* Procedia Computer Science, 2018. **127**: p. 180-189.
35. Gul, F., W. Rahiman, and S.S. Nazli Alhady, *A comprehensive study for robot navigation techniques.* Cogent Engineering, 2019. **6**(1): p. 1632046.
36. Nguyen, M.T., C. Yuan, and J.H. Huang. *Kinematic analysis of A 6-DOF robotic arm*. in *Advances in Mechanism and Machine Science: Proceedings of the 15th IFToMM World Congress on Mechanism and Machine Science 15*. 2019. Springer.
37. Iqbal, J., R.U. Islam, and H. Khan, *Modeling and analysis of a 6 DOF robotic arm manipulator.* Canadian Journal on Electrical and Electronics Engineering, 2012. **3**(6): p. 300-306.

38. Gai, S.N., et al. *6-DOF robotic obstacle avoidance path planning based on artificial potential field method*. in *2019 16th International Conference on Ubiquitous Robots (UR)*. 2019. IEEE.
39. Flanders, M. and R.C. Kavanagh, *Build-A-Robot: Using virtual reality to visualize the Denavit–Hartenberg parameters.* Computer Applications in Engineering Education, 2015. **23**(6): p. 846-853.
40. Hasan, A.T., et al., *An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 DOF serial robot manipulator.* Advances in Engineering Software, 2006. **37**(7): p. 432-438.
41. Li, G., et al., *An inverse kinematics method for robots after geometric parameters compensation.* Mechanism and Machine Theory, 2022. **174**: p. 104903.
42. Atique, M.M.U., M.R.I. Sarker, and M.A.R. Ahad, *Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit-Hartenberg convention.* Heliyon, 2018. **4**(12): p. e01053.
43. Li, Z. and J. Schicho, *A technique for deriving equational conditions on the Denavit–Hartenberg parameters of 6R linkages that are necessary for movability.* Mechanism and Machine Theory, 2015. **94**: p. 1-8.
44. Almusawi, A.R., L.C. Dülger, and S. Kapucu, *A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242).* Computational intelligence and neuroscience, 2016. **2016**.