Bibot: An Autonomous Food Delivery Robot

submitted in partial fulfillment of the requirements
for the degree of

Bachelor of Engineering
in
Electronics and Computers Engineering

Submitted By

Manveer Singh (Regn. No. : 102015114)
Raghav Khanna (Regn. No. : 102195003)
Siddhant Saxena (Regn. No. : 102015116)
Chaitanya Dua (Regn. No. : 102015092)
Devanshi Arora (Regn. No. : 102015029)
Vansh Ahuja (Regn. No. : 102015156)

Group No – 59

Under the Supervision of

Prof. Kulbir Singh
Professor, ECED
TIET, Patiala

Dr. Hem Dutt Joshi
Associate Professor, ECED
TIET, Patiala

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Department of Electronics and Communication Engineering
Thapar Institute of Engineering & Technology, Patiala, Punjab

December 2023

Declaration

We hereby declare that the capstone project group report title “Autonomous Food Delivery Bot” is an authentic record of our own work carried out at “Thapar Institute of Engineering and Technology, Patiala” as a Capstone Project in seventh semester of B.E. (Electronics & Communication Engineering), under the guidance of Prof. Kulbir Singh and Dr. Hem Dutt Joshi, during January to November 2023.

Date: 30 November 2023

S. No.	Name	Roll Number	Signature
1.	Manveer Singh	102015114	
2.	Raghav Khanna	1020195003	
3.	Chaitanya Dua	102015092	
4.	Vansh Ahuja	102015156	
5.	Siddhant Saxena	102015116	
6.	Devanshi Arora	102015029	

Certificate

This is to certify that the report titled **Autonomous Food Delivery Robot**, submitted by *Manveer Singh, Raghav Khanna, Siddhant Saxena*, *Chaitanya Dua*, *Vansh Ahuja*, *Devanshi Arora* to the Thapar Institute of Engineering & Technology, Patiala, for the award of the degree of *Bachelor of Engineering*, is a record of the project work done by them under our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: TIET, Patiala
Date: 30-11-2023

Prof. Kulbir Singh	Dr. Hem Dutt Joshi
Professor, ECED	*Associate Professor, ECED*

Acknowledgements

We would like to express our sincere gratitude towards our mentors, *Prof. Kulbir Singh* and *Dr. Hem Dutt Joshi* from Electronics and Communication department for their invaluable guidance and support throughout the course of the capstone project. Their expertise, feedback, encouragement was instrumental in shaping the direction of the project.

We would also like to express our thanks to *Thapar Institute of Engineering & Technology (TIET)* for providing us the opportunity to showcase and improve our skills through the Capstone Project, and for providing us the necessary resources, facilities required for our project.

Lastly, we would like to thank our friends and family who have helped and supported us directly or indirectly throughout the project.

Date: 30 November 2023

Roll No.	Name
102015114	Manveer Singh
102195003	Raghav Khanna
102015092	Chaitanya Dua
102015156	Vansh Ahuja
102015116	Siddhant Saxena
102015029	Devanshi Arora

Abstract

The increasing demand for convenient and efficient food delivery services has driven the development of autonomous mobile robots (AMRs). AMRs have the potential to revolutionize the food delivery industry by streamlining operations, reducing costs, and enhancing customer satisfaction. This research proposes a customized robotic system for food delivery applications, utilizing a modular software architecture and advanced perception techniques.

The system's core is a ROS-based modular software architecture that facilitates seamless integration of hardware components and software functionalities, including mapping, localization, planning, and control. The robot system employs two sensors, them being the LiDAR and RGB-D camera to gather detailed environmental information. Sensor fusion techniques combine 2D laser scans from LiDAR with 3D point clouds from the camera, constructing an accurate and comprehensive model of the surroundings. To enable real-time mapping and obstacle avoidance, the system utilizes efficient Simultaneous Localization and Mapping (SLAM) algorithms like RTAB-Map. The Adaptive Monte Carlo Localization (AMCL) method accurately tracks the robot's pose within the generated maps. Subsequently, the move_base navigation stack plans optimal paths between specified goals while skillfully navigating around obstacles. Experimental validation conducted in indoor test environments demonstrates the robot's robust autonomous navigation capabilities. The studies highlight the efficacy of the modular ROS-based architecture, augmented by sensor fusion, in developing adaptable self-driving platforms.

While the current focus is on food delivery, the modular and perception-driven nature of this system extends its potential for automation across diverse domains, including healthcare, defense, and smart cities. With further advancements in human-robot interaction and multi-agent coordination, these AMRs can be seamlessly integrated into urban landscapes, facilitating efficient last-mile goods and food transport.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

TIET	Thapar Institute of Engineering and Technology
ECE	Electronics and Communication Engineering
AMR	Autonomous Mobile Robot
SLAM	Simultaneous Localization and Mapping
AMCL	Adaptive Monte Carlo Localization
ROS	Robotics Operating System
LiDAR	Light Detection and Ranging
ToF	Time of Flight
ICP	Iterative Closest Point

Notations

List of notations that were utilized in this report are mentioned below:

- m: Meter
- kg-cm: Kilogram/centimeter
- r: Radius (m)
- θ: Angle between x and y in degrees
- v: Velocity of the object (m/s)
- rpm: Rotations per minute

Chapter 1 - Introduction

In our modern, fast-paced world, the demand for quick and easy solutions has skyrocketed, impacting many aspects of our lives, including food delivery. Traditional food delivery methods, often reliant on human delivery personnel, have struggled to keep pace with the surging demand, leading to inefficiencies and delays. Recognizing the need for a transformative approach, autonomous mobile robots (AMRs) have emerged as a beacon of innovation, poised to revolutionize the food delivery landscape.

AMRs, equipped with sophisticated sensors, advanced algorithms, and unparalleled precision, offer a promising solution to the challenges that have plagued traditional food delivery methods. Unlike human delivery personnel, AMRs can navigate autonomously, adeptly maneuvering through crowded environments, bypassing obstacles, and reaching their destinations with remarkable accuracy. This capability stems from their ability to perceive their surroundings using a multitude of sensors, including 2D LiDAR sensors and RGB-D cameras.

By leveraging these sensors, AMRs can construct a detailed map of their surroundings, enabling them to make informed decisions about their route and avoid potential hazards. The 2D LiDAR sensor, in particular, plays a pivotal role in generating a precise representation of the environment, allowing the AMR to identify obstacles and plan safe routes. Complementing the LiDAR data, the RGB-D camera captures color images and depth information, providing additional visual cues for the AMR's perception and understanding of its surroundings.

The integration of AMRs into food delivery scenarios offers a multitude of benefits. By streamlining operations and optimizing routes, AMRs can significantly reduce delivery times, enhancing customer satisfaction and providing businesses with a competitive edge. Moreover, AMRs contribute to improved safety by minimizing the risk of human error and ensuring adherence to traffic regulations. Additionally, AMRs can operate during non-traditional hours, expanding delivery services and catering to a wider customer base.

The development of an autonomous food delivery mobile robot equipped with a 2D LiDAR sensor and an RGB-D camera, powered by ROS, lies at the heart of this research. This project aims to showcase the practical implementation of AMRs in food delivery applications, demonstrating their potential to revolutionize the industry. Fig.1 shows the overview of the model developed in this project.

To effectively perceive its surroundings and navigate autonomously, the proposed AMR employs a 2D LiDAR sensor and an RGB-D camera. The 2D LiDAR sensor generates a detailed map of the environment, enabling the robot to identify obstacles and plan safe routes. The RGB-D camera, complementing the LiDAR data, captures color images and depth information, providing additional visual cues for the robot's perception and understanding of its surroundings.

ROS serves as the backbone of the system, providing a flexible and modular framework for seamless integration between hardware and software components. It facilitates the development of complex robotic systems by streamlining communication between various components, enabling the autonomous food delivery mobile robot to easily interface with its sensors, process sensor data, and implement advanced algorithms for navigation and obstacle avoidance.

This research delves into the design and implementation of an autonomous food delivery mobile robot, aiming to address the evolving demands and challenges of the food delivery industry. The project's outcomes have the potential to bring significant advancements to the food delivery industry and inspire further progress in the field of autonomous robotics.

1.1 Motivation

The impetus behind developing an autonomous food delivery mobile robot equipped with a 2D LiDAR sensor and an RGBD camera powered by ROS stems from the compelling need to address the evolving demands and challenges of the food delivery industry. Traditional food delivery methods, primarily reliant on human-operated delivery personnel, are often characterized by time inefficiencies, cost overheads, and an increased risk of human error. By introducing autonomous robots into this domain, we aim to transcend these limitations and revolutionize the food delivery process, ushering in an era of unparalleled efficiency, safety, and sustainability.

This project's motivations lie in the aspiration to significantly enhance the efficiency and speed of food delivery services. Autonomous robots, armed with advanced sensors and sophisticated algorithms, possess the capability to navigate through busy environments with remarkable ease, deftly maneuvering around obstacles and traversing traffic congestion with precision. By streamlining the delivery process and reducing delivery times, we aim to elevate customer satisfaction to unprecedented levels, while simultaneously providing businesses in the food industry with a competitive edge.

Safety is paramount in the realm of food delivery and the development of autonomous food delivery robots is inextricably linked to this crucial aspect. Human errors, distractions, and fatigue can lead to accidents during food transportation, jeopardizing the safety of both the delivery personnel and the general public. By employing robust sensors like the 2D LiDAR and RGBD camera, our autonomous robot can meticulously perceive its surroundings, accurately detect obstacles in real-time, and make informed decisions to navigate safely. This comprehensive perception system minimizes the risk of accidents, ensuring the protection of all parties involved.

Environmental sustainability is a pressing concern and the implementation of an autonomous food delivery robot aligns with the collective goal of reducing carbon emissions and promoting eco-friendly practices. Traditional food delivery methods often involve multiple vehicles traveling long distances, resulting in increased fuel consumption and a detrimental impact on the environment. By introducing autonomous robots that can optimize routes, consolidate deliveries, and minimize unnecessary trips, we can significantly decrease the ecological footprint associated with food delivery operations, contributing to a more sustainable future.

1.2 Assumptions and Constraints

1.2.1 Assumptions

- **Adequate Sensor Performance:** It is assumed that the 2D LiDAR sensor and RGBD camera will provide accurate and reliable data for perception and navigation tasks. The assumption is that these sensors can effectively capture the necessary environmental information required for safe and efficient navigation.
- **Known Environment**: The project assumes that the environment in which the autonomous food delivery mobile robot operates is known or can be mapped beforehand. This assumption allows for pre-planning of routes and the identification of potential obstacles and waypoints.
- **Standardized Food Packaging:** The project assumes that the food items to be delivered are packaged uniformly and have consistent dimensions and weight. This assumption simplifies the grasping and transportation process, as the robot can be designed to accommodate a specific type of packaging.
- **Controlled Weather Conditions**: The project assumes that the robot operates in controlled weather conditions, such as indoor or moderate outdoor environments. Extreme weather conditions like heavy rain, snowstorms, or strong winds might introduce additional challenges and constraints that need to be addressed separately.
- **Predictable Customer Behavior:** It is assumed that customer behavior patterns and order volumes can be predicted with reasonable accuracy. This assumption allows for optimization of delivery schedules, route planning, and resource allocation.
- **Limited Interaction with Humans:** It is assumed that the robot's interactions with humans during the delivery process will be limited to handing off food orders and receiving instructions. This assumption simplifies the human-robot interaction (HRI) design and reduces the complexity of communication protocols.

1.2.2 Constraints

- **Cost Constraints:** The development of an autonomous food delivery mobile robot involves costs associated with hardware components, sensors, computing resources, and software development. Budget constraints may limit the selection of components and technologies, requiring careful consideration of cost-effective solutions.
- **Hardware Limitations:** The project operates within the constraints imposed by the physical capabilities of the robot. Factors such as the maximum payload capacity, speed, power limitations, and physical dimensions of the robot impose constraints on its design and operation.
- **Time Constraints:** The project is subject to time limitations, including the development and testing phases. Meeting deadlines and milestones within the given timeframe is crucial, and the project should adhere to realistic timelines to ensure successful completion.

- **Legal and Regulatory Compliance:** The project must adhere to all applicable laws and regulations governing autonomous vehicles and robotics. This includes safety standards, liability considerations, and data privacy requirements.
- **Public Acceptance and Trust:** The project must consider the public's perception of autonomous food delivery robots and address any concerns or fears. This may involve transparency, communication strategies, and safety demonstrations.
- **Integration with Existing Infrastructure:** The project must seamlessly integrate with existing food delivery infrastructure, including order management systems, restaurant kitchens, and customer delivery platforms. This requires compatibility and interoperability.
- **Maintenance and Repair Considerations:** The project must account for the maintenance and repair procedures for the robot, ensuring that it can be easily serviced and kept in operational condition. This includes hardware diagnostics, software updates, and component replacement.
- **Data Security and Privacy Protection:** The project must implement robust cybersecurity measures to protect sensitive customer data, order information, and robot sensor data. This includes encryption, access control, and data breach prevention protocols.

1.3 Novelty of work

One key area of novelty is the fusion of a 2D LiDAR sensor and RGBD camera to enhance perception and mapping. This allows for more accurate navigation and obstacle avoidance by combining 2D LiDAR data with depth information from the camera. Leveraging the Robot Operating System (ROS) also introduces a standardized software framework to integrate various functions and components.

Another major innovation involves developing navigation algorithms to handle the challenges of complex real-world environments like crowded sidewalks. This enables the robot to safely maneuver through unpredictable scenarios. Designing intuitive interfaces and protocols tailored for human-robot interaction during food delivery also provides novel approaches to user experience.

Additional novel aspects include route and schedule optimization algorithms that consolidate deliveries and minimize distance traveled to increase efficiency and sustainability. There is also novelty in the practical implementation and deployment of the robot as a commercially viable autonomous food delivery solution that seamlessly integrates with existing infrastructure.

In summary, key innovations span perception, software architecture, real-world navigation, human-robot interaction protocols, route optimization for efficiency and sustainability, and ultimately the real-world deployment of the autonomous delivery robot.

Chapter 2 - Literature Survey

2.1 Literature Survey

The pervasiveness of the ROS has reached unprecedented levels in today's technological landscape. ROS stands as a robust software framework that streamlines the development of robot software by providing a comprehensive suite of libraries, tools, and protocols. This has revolutionized the way we approach robotics, enabling rapid prototyping and innovation. M. Köseoğlu et al. presents the design and implementation of an autonomous mobile robot (AMR) using the ROS. The authors discuss the hardware architecture, electronic communication protocols, and the integration of sensors and actuators. They describe the challenges faced during testing and propose solutions [1]

For mobile robots to effectively navigate and operate within diverse environments, two critical technologies play a pivotal role: SLAM and path planning navigation. [2] SLAM empowers robots to construct and maintain a representation of their surroundings while simultaneously determining their precise location within that representation. Path planning navigation, on the other hand, utilizes this map to devise optimal routes that enable robots to reach their destinations efficiently and safely.

2.1.1 LiDAR Based Navigation

D Hutabarat et al. presented the development of an autonomous mobile robot that incorporated LiDAR technology to achieve effective obstacle avoidance. The robot consistently measured distances with accuracy, regardless of object color and ambient light conditions. It successfully evaded objects of varying sizes and maneuvered through indoor environments without causing any disturbance to walls or obstacles. The utilization of LiDAR, known for its extensive detection range and precise measurements, proved to be a valuable solution for obstacle avoidance in autonomous mobile robots [3]. Maryna Derkach et al. proposes a novel algorithm for real-time obstacle avoidance in small autonomous mobile robots. The use of a linear recursive Kalman filter improves localization and enhances the robot's ability to adapt to additional obstacles. The experimental platform configuration demonstrates the practical implementation of the proposed algorithm. The simulation results validate the effectiveness of the approach. The paper provides valuable insights into the field of autonomous robotics and contributes to the development of efficient obstacle avoidance systems [4].In the dynamic and ever-changing world we inhabit, AMRs must possess the ability to gather and process environmental information in real-time. This sensory data is crucial for AMRs to autonomously navigate and execute tasks effectively. [5] SLAM has emerged as the cornerstone of AMR navigation, providing a powerful solution for autonomous operation in diverse environments. SLAM algorithms enable robots to construct and maintain maps of their surroundings while simultaneously determining their location within those maps. This capability is achieved through the fusion of sensor data, typically from LIDAR and cameras.

S. F. Andriawan Eka Wijaya et al. employs the study which focuses on the grid map occupancy technique by facilitating mapping in uncharted terrains by transforming LiDAR sensor data into a 2D map representation. This approach achieved an environment mapping accuracy with an average error of 6.05 cm [6]. Balasuriya et al. details the theory behind SLAM and the implementation of the algorithm with experimental results. The study omits control data and focuses on scan-matching for the map generation and pose estimation, demonstrating the effectiveness of the SLAM approach [7]. W. Hess et al. presents a novel method to tackle the crucial issue of loop closure in 2D LIDAR-based Simultaneous Localization and Mapping (SLAM) systems. The study predominantly centers on the identification and rectification of loop closures in real-time and suggests a streamlined approach to detect and resolve loops in a robot's path, thereby greatly enhancing the precision of the created maps. [8] D. Talwar et al. conducted a comprehensive analysis on Adaptive Monte Carlo Localization (AMCL) involving the utilization of a solitary lidar sensor for the purpose of Simultaneous Localization and Mapping (SLAM) within the ROS framework. The research focuses on precise robot localization and navigation and the experiments demonstrate that AMCL allows the robot to navigate effectively in both stationary and changing environments. [9]

While Light Detection and Ranging (LiDAR) technology has revolutionized the field of robotics, particularly in the realm of autonomous mobile robots (AMRs), it is not without its limitations. One significant drawback of LIDAR is its inherent two-dimensionality, which can lead to the overlooking of certain obstacles, particularly those with complex geometries or low heights. This limitation can pose a serious threat to the safety of both the robot and its surroundings, increasing the risk of collisions and potential damage. To address this challenge, researchers have explored various approaches, each with its own set of advantages and limitations. One approach involves employing the Hector SLAM algorithm, a robust and widely used method for simultaneous localization and mapping (SLAM). However, the Hector SLAM algorithm may struggle in environments with limited or sparse features, potentially compromising the robot's ability to accurately perceive its surroundings. Another approach utilizes Real-Time Kinematic Global Positioning System (RTK-GPS) SLAM, which offers superior accuracy in depicting the environment. However, RTK-GPS SLAM is computationally intensive, demanding significant processing power that may not be readily available on all AMR platforms.

2.1.2 RGBD Camera Based Navigation

R. Lagisetty et al. proposes a system that utilizes a simple kinematic model for the mobile robot and employs a stereo camera with pan and tilt functionality for long-range operation. The stereo matching algorithm and triangulation method are used to achieve complete 3D reconstruction of objects/obstacles. The position and orientation of the mobile robot are determined from static object observations using RANSAC in successive frames. Obstacle avoidance is formulated based on potential field methods, considering obstacle range, size information, and the robot's position and orientation. The research also formulates and verifies a proportional derivative navigation control loop along with the obstacle avoidance algorithm [10]. Gyula Mester et al. proposes a fuzzy control strategy for the velocity control and obstacle avoidance of mobile robots in unknown environments. The mobile robot model used in the study consists of two driving wheels with independently controlled angular velocities. The proposed fuzzy control strategy employs a reactive navigation approach to ensure collision-free motion and velocity control in the presence of obstacles. The outputs of the fuzzy controller are the angular speed difference between the left and right wheels and the overall velocity of the vehicle [11].

Marwah M. Almasri proposes a new technique that utilizes low-cost infrared sensors and involves real-time control applications for line following and collision avoidance in AMR systems. The author presents a simulation setup implemented on multiple scenarios to demonstrate the robot's ability to follow a path, detect obstacles, and navigate around them. The proposed technique is validated using the Webots simulator [12]. P. Saranrittichai et al. proposes a modification to the Dynamic Window approach for local obstacle avoidance in autonomous mobile robots. The proposed Field Dynamic Window approach (F-DWA) modifies the objective function to take into account obstacles near the trajectory using a histogram grid representation. The crashing probability of the trajectory is estimated based on the histogram grid, and the objective function considers this probability along with other factors [13].

Researchers have also investigated the use of monocular cameras and Deep Reinforcement Learning (DRL) to estimate depth information. J. Fang et al. conducted an in-depth study on obstacle avoidance algorithms for food delivery robots and introduced an innovative approach that combines an improved ant colony calculation method, leveraging collective intelligence, with an enhanced version of the A* algorithm. This integration aims to enable dynamic obstacle avoidance and optimize global path planning for mobile robots operating in complex environments [14]. While this method holds promise, it is heavily reliant on extensive data collection and computational resources for model training. Additionally, its performance is constrained by the limitations of pre-existing datasets, which may not adequately represent the diverse range of environments in which AMRs operate. K.H. Sedighi et al. presents a genetic algorithm approach for local obstacle avoidance in path planning of a mobile robot. The objective of the method is to find an optimal and collision-free path by minimizing the path length and the number of obstacles encountered. The proposed algorithm allows free movement of the robot in any direction, making it suitable for handling complex search spaces. The paper discusses the importance of motion planning in autonomous mobile robots and highlights the distinction between global and local path planning [15].

Canglong Liu et al. proposes a CNN-based vision model for obstacle avoidance of mobile robots. The model utilizes an end-to-end learning approach, taking raw images from a camera as input and generating steering commands (turn left, turn right, go straight) as output. Training data is collected by a remotely controlled mobile robot exploring a structured environment without colliding into obstacles. The neural network is trained using the Caffe framework, and the Robot Operating System (ROS) is used for executing specific instructions [16]. D. Ghorpade et al. proposes a method that utilizes segmentation and clustering techniques to extract spatial information from the laser point-cloud data. The Convex hull algorithm is employed to accurately identify the geometrical structure of obstacles. Additionally, a visibility graph path planning method is used to establish an optimal route to the destination. The paper highlights the use of a simple mathematical model to achieve real-time performance and verifies the reliability of the proposed model through MATLAB simulations [17].

A more comprehensive approach combines LiDAR SLAM with ultrasonic sensors to achieve accurate and flexible obstacle avoidance. This method leverages the strengths of both sensor modalities, enabling the robot to detect and respond to a wider range of obstacles. However, the computational complexity of this approach is directly proportional to the sensitivity of the sensors, potentially impacting the robot's real-time performance. Integrating multiple sensors for data fusion presents additional challenges, as it requires precise calibration and synchronization to ensure accurate environmental mapping. The complexity of this process can be further compounded by the dynamic nature of real-world environments, where sensor data may be subject to noise and interference.

In the realm of autonomous mobile robots (AMRs), the pursuit of robust and reliable obstacle avoidance demands a multifaceted approach that transcends the inherent limitations of individual sensor technologies. While advancements in hardware and software have significantly enhanced the capabilities of LIDAR and other sensors, there remains a compelling need to expand the scope of data acquisition for robots. To address this challenge, researchers have embarked on a journey to integrate multiple sensor modalities, empowering AMRs to gather a more comprehensive and diverse understanding of their surroundings. This multi-sensory approach not only augments the robot's ability to detect and identify obstacles but also introduces redundancy and resilience against sensor failures or environmental noise.

One promising avenue involves the utilization of the Kinect sensor, a cost-effective RGB-D camera that provides both color and depth information. R. Crimmins et al. meticulously investigated the effectiveness and cost-efficiency of a mapless and appearance-based obstacle avoidance method employing the Kinect sensor. [18] Their innovative approach centers on leveraging depth-related data and a specific Region of Interest (ROI) for efficient obstacle avoidance. The Kinect V1 sensor has exhibited remarkable performance within the range of 0.8 meters to 4 meters for obstacle avoidance tasks. This operational range proves to be sufficient for a multitude of indoor applications, encompassing navigation through homes, offices, and warehouses. P. Fankhauser et al. proposed empirical models to approximate the axial and lateral noise amplitudes in order to facilitate further data processing. A comparison between the original Kinect and the Kinect v2 reveals that the latter exhibits significantly lower axial noise magnitudes, particularly at larger distances [19].

The camera's ability to reduce spatial and temporal noise caused by reflection was limited, which in turn limited its depth-sensing capabilities. This is because capturing accurate depth information in environments with complex lighting conditions and reflective surfaces is inherently difficult. Although the camera used advanced noise reduction algorithms, these techniques were not enough to completely counteract the negative effects of reflectance. As a result, the depth maps generated by the camera showed noticeable artifacts and inaccuracies, especially in areas with high reflectance. Sensor data fusion has emerged as a promising way to overcome the limitations of individual sensors.

By combining data from multiple sensors, fusion algorithms can effectively reduce noise and improve the overall quality of the information collected. In the context of depth sensing, sensor data fusion can be used to combine depth information from the camera with complementary data from other sensors, such as inertial measurement units (IMUs) and lidar scanners. By using the strengths of each sensor, fusion algorithms can create more accurate and reliable depth maps, even in challenging environments. The successful use of sensor data fusion in various fields, including robotics, autonomous vehicles, and augmented reality, highlights its potential to revolutionize depth-sensing capabilities. As research in this area continues, we can expect to see the development of increasingly sophisticated fusion algorithms that can further improve the accuracy and robustness of depth-sensing systems.

2.1.3 Sensor Fusion

Yique Deng et al. introduces a fusion method by combining RTK-GPS and LiDAR-based SLAM using an adaptive complementary filter, improving localization accuracy and continuity. Experimental results show superior performance compared to standalone GPS, Cartographer, and dead reckoning methods. [20] Koki Yokoyama et al. employed the use of depth estimation based on a monocular camera and deep reinforcement learning. The study achieved successful autonomous navigation by converting depth data into representative states for the reinforcement learning model. [21] D. T. Savaria et al. proposes a promising SLAM method which incorporates Stereo vision and the Speeded Up Robust Features (SURF) algorithm to identify landmarks for autonomous exploration, mapping, and backtracking. The paper suggests improving gateway detection algorithms and adding 3D visual odometry. [22] I Ohaya, Akio Kosaka et al. employed vision-based processing and ultrasonic sensors, to combine self-localization and obstacle avoidance. An extended Kalman filter is used to evaluate the expectation image with camera images for self-localization prior to the navigation process. Adaptive thresholding is then employed to identify static obstacles. [23] [24] V. L. Popov et al. proposed a robust RGB-D camera and 2-D LiDAR sensor fusion method for vision-based relative localization of moving targets. Adaptive color-based particle filtering and an interacting multiple-model estimator improve 2-D location measurements. Fusion of 2-D LiDAR data improves localization accuracy. [25] Haryong Song et al. proposed a novel approach in this research for localizing the position of a target by utilizing measurements from an RGB-D camera and a 2-D LiDAR sensor. The approach involves applying low-level fusion of heterogeneous sensors and employing a modified track-to-track fusion with a feedback algorithm. By utilizing LiDAR measurements originating from both visual and depth trackers, the proposed method achieves improved and robust 2-D localization [26].

2.2 Research Gaps:

Sensor Fusion Techniques: The fusion of LiDAR and RGBD camera data has been a subject of extensive research in the field of autonomous vehicles, demonstrating its effectiveness in enhancing perception and navigation capabilities. However, the unique requirements and operational scenarios of autonomous food delivery bots call for a more specialized approach to sensor fusion. Current research efforts have primarily focused on fusion techniques applicable to autonomous vehicles, leaving a significant gap in the development of fusion algorithms tailored specifically for autonomous food delivery bots. Addressing this research gap is crucial to identify the optimal fusion algorithms that can effectively harness the complementary strengths of LiDAR and RGBD camera data in food delivery scenarios. LiDAR provides accurate depth measurements and excels in detecting obstacles in low-light conditions, while RGBD cameras capture rich color and texture information, enabling object recognition and scene understanding. By seamlessly integrating these disparate data streams, autonomous food delivery bots can achieve enhanced perception capabilities, enabling them to navigate complex environments safely and efficiently.

Environmental Adaptability: Autonomous food delivery bots must navigate a wide range of real-world scenarios, including crowded spaces, dynamic obstacles, and varying lighting conditions. While existing research has made significant progress in perception and navigation for autonomous vehicles, it has largely focused on controlled environments, such as well-marked roads and predictable traffic patterns. In contrast, food delivery bots operate in more dynamic and unpredictable environments, such as indoor spaces with pedestrians, moving objects, and varying lighting conditions. Exploring how the fusion of LiDAR and RGBD camera data can improve the adaptability of food delivery bots in challenging real-world conditions is essential. Robust perception algorithms that can handle occlusions, reflections, and varying environmental factors are crucial to ensure the safety and reliability of autonomous food delivery systems. These algorithms should be able to effectively perceive the environment in real-time, even under conditions where LiDAR or RGBD camera data alone may be unreliable or incomplete.

Human Interaction and Object Recognition: Autonomous food delivery bots need to interact with humans and recognize objects accurately to navigate within complex indoor environments. Human detection, gesture recognition, and object recognition are essential tasks for enabling seamless navigation and interaction in these settings. The fusion of LiDAR and RGBD camera data can provide rich information for these tasks, as LiDAR accurately detects human silhouettes and objects, while RGBD cameras capture color and texture information that can be used for object recognition and gesture identification. However, research focusing on developing efficient algorithms for human-robot interaction and object recognition specifically tailored for food delivery scenarios is limited. Bridging this research gap will facilitate seamless human-robot collaboration and enhance the overall customer experience. By understanding human intentions and recognizing objects accurately, autonomous food delivery bots can navigate safely and efficiently in indoor environments, avoiding collisions and delivering food orders without disrupting human activities.

System Integration and Scalability: Integrating LiDAR, RGBD cameras, and other components into a unified framework using ROS is a crucial aspect of developing autonomous food delivery bots. ROS (Robot Operating System) provides a flexible and modular software framework for building robotic applications. However, the existing literature lacks comprehensive studies on system integration challenges and scalability issues specific to autonomous food delivery bots. Investigating efficient data fusion architectures, optimizing computational resources, and addressing the communication and synchronization challenges within the ROS ecosystem are critical for developing robust and scalable autonomous food delivery systems. As the complexity of autonomous food delivery bots increases, ensuring efficient data processing, communication, and synchronization between sensors, actuators, and control algorithms becomes increasingly important. Effective system integration will pave the way for the development of autonomous food delivery bots that can operate reliably and efficiently in real-world scenarios.

Chapter 3 - Problem Formulation and Objectives

Food delivery is revolutionized by AMRs. Due to the challenges, they face in complex environments, a new sensor fusion approach combines Lidar and RGBD camera data for robust and reliable navigation. Sensor fusion-powered AMRs offer improved efficiency, reduced labor costs, enhanced scalability, and improved customer service. As technology matures, we can envision a future where autonomous food delivery becomes ubiquitous.

3.1 Problem Formulation

The food delivery landscape is undergoing a remarkable transformation, driven by the advent of autonomous robots (ARs). These intelligent machines hold the potential to revolutionize the very essence of food delivery operations, offering unprecedented levels of efficiency, reduced costs, and enhanced customer satisfaction. However, the effective deployment of ARs hinges on their ability to navigate complex and dynamic environments with precision, reliability, and adaptability.

Conventional AR navigation approaches often rely on a single sensor modality, such as rangefinders or depth cameras, to perceive the surroundings. While rangefinders excel in providing accurate distance measurements, they lack the ability to capture rich visual information essential for object recognition and classification. Conversely, depth cameras offer detailed visual data but struggle in low-light conditions and are susceptible to noise and occlusions. These limitations often hinder the performance of ARs, particularly in challenging environments.

To address these limitations and unlock the full potential of ARs in food delivery, a novel sensor fusion approach has emerged. This synergistic approach combines rangefinder and depth camera data, leveraging the strengths of each sensor modality to overcome their individual shortcomings. By integrating complementary information from both sensors, ARs can achieve robust and reliable navigation in diverse environments, paving the way for a seamless food delivery experience.

The proposed sensor fusion framework addresses the key challenges that hinder effective AR navigation: data complementation, feature extraction and fusion, real-time navigation and mapping, and adaptive sensor fusion. This approach combines the strengths of rangefinder and depth camera data to overcome their individual limitations, enabling ARs to perceive their surroundings accurately, make informed decisions, and navigate effectively in dynamic environments.

By seamlessly integrating complementary information from rangefinder and depth camera data, sensor fusion technology empowers ARs to navigate complex and dynamic environments with precision, reliability, and adaptability. This breakthrough enhances AR navigation capabilities, enabling them to operate effectively in a wide range of indoor and outdoor environments. As a result, the food delivery industry can expect unprecedented benefits, including improved efficiency, reduced labor costs, enhanced scalability, and improved customer service.

The integration of sensor fusion technology into autonomous robots marks a pivotal moment in the evolution of food delivery. By overcoming the limitations of conventional navigation approaches, sensor fusion-powered ARs are poised to transform the industry, offering unprecedented levels of efficiency, reduced costs, and enhanced customer satisfaction. As this technology matures and adoption accelerates, we can envision a future where autonomous food delivery becomes ubiquitous, seamlessly delivering meals to homes, offices, and college campuses.

3.2 Objectives

1. Design and build a robust, agile chassis that can navigate diverse terrain like sidewalks, paths, and stairs to enable smooth campus delivery.

2. Utilize the Robot Operating System (ROS) to provide a flexible and scalable software framework for robot control and navigation, ensuring adaptability to evolving needs and functionalities.

3. Implement teleoperation capabilities to allow remote control of the robot during testing and maintenance phases, providing enhanced control and flexibility for fine-tuning performance.

4. Utilize LiDAR and camera data fusion to enhance environmental perception, improving object recognition, classification, and avoidance for safe navigation.

5. Employ robust localization and planning algorithms like AMCL, A*, and dynamic window approach for reliable positioning and obstacle-free navigation.

6. Detect and avoid unexpected obstacles in real-time using the perception system and dynamic obstacle avoidance techniques, ensuring uninterrupted delivery.

Chapter 4 - Project Design and Description

This section explores the inner workings of the project, revealing how the mechanical and electronic subsystems work together seamlessly under ROS control. The electronics section dives into the circuitry that powers the functionality of the system, while the mechanical section outlines the robot's specifications and components. Additionally, the ROS framework's structure and interaction with the other subsystems are thoroughly examined.

4.1 Mechanical System Overview

4.1.1 Platform Design

The autonomous robot utilizes a four-wheeled differential drive system for mobility. To facilitate testing and development, a wooden-steel platform serves as the robot's foundation. The platform's structural components are constructed by welding metallic pipes and motor mounts, providing a robust and stable base. To ensure a clean and aesthetically pleasing appearance, the platform's base and structural framework are covered with wooden sheets and plywood boards.

4.1.2 Dimensions and Sensor Integration

The robot measures 60 cm in length, 40 cm in width, and 40 cm in height. This size provides a balance between maneuverability and stability, making it suitable for various indoor and outdoor environments. To accommodate the Kinect sensor, a dedicated cutout has been incorporated into the platform's design. Additionally, mounting points have been integrated to support the LiDAR and RGBD sensors. These sensors play crucial roles in the robot's perception and navigation capabilities.

4.1.3 Initial Chassis Development

For initial teleoperation and system testing purposes, a simple chassis with mounts was fabricated, depicted in Fig. 4.1.3 This initial chassis facilitated preliminary testing and allowed for the evaluation of the robot's basic functionality. Further refinements and enhancements will be implemented based on the findings of these initial tests.

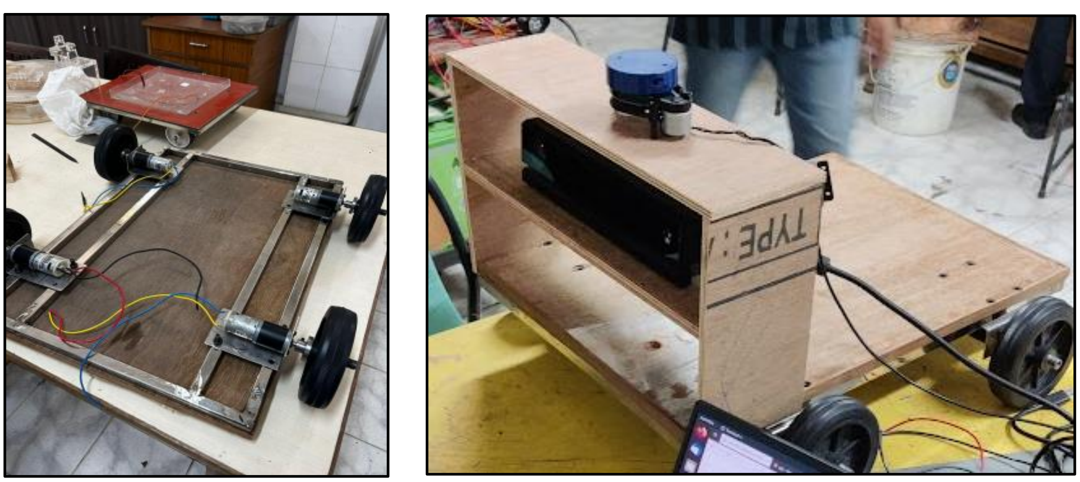

Fig. 4.1.1 Robots initial fabrication

4.1.4 Sensor Housing and System Refinement

Based on initial testing and validation, a permanent housing unit was fabricated for the sensors. This enclosure provides improved protection and organization for the sensors, enhancing their longevity and reliability. Additionally, the robot's wire management system was refined to minimize clutter and optimize signal transmission. Structural integrity was also enhanced to withstand the rigors of real-world operation. Finally, a storage compartment was added to accommodate essential accessories and tools.

4.1.5 Final Robot Design

The final robot design, depicted in Fig. 4.1.5, incorporates the aforementioned refinements and enhancements. It represents a culmination of the design and development process, resulting in a robust, reliable, and versatile autonomous robot platform.

Fig. 4.1.2 Robot after final fabrication

4.2 Electronics System Overview

The electronics system serves as the project's backbone, providing the requisite power, control, and communication infrastructure for the seamless integration and operation of all subsystems. This autonomous system encompasses a diverse array of components, ranging from microcontrollers and sensors to power distribution units and communication interfaces.

4.2.1 Motor

The 12V 100RPM DC Johnson motor was carefully considered based upon its specifications to produce enough torque for superior performance. The DC 12V 100 rpm Johnson motor is equipped with a metal gearbox and metal gears, which provide a significant gear reduction of 180:1. This means that the motor can produce a high holding torque of 27.18 Kgcm, Rated Torque of 11.7 Kg cm and Stall Torque 46 Kg

Cm mentioned in Table 3, even at its low rotational speed of 100 rpm. This makes the motor ideal for applications that require high torque at low speeds, such as robotics, automation,and actuator control. This motor can be seen in Fig. 4.2.1.

Fig. 4.2.1 Picture of johnson motor

It is designed for durability and reliability with its metal gearbox and metal gears. These components are resistant to wear and tear, ensuring long-lasting performance in demanding applications. Additionally, the motor's M3 threaded hole on the 6mm diameter shaft provides a secure mounting option, preventing unwanted movement and ensuring stable operation.

Despite its high torque output, the DC 12V 100 rpm Johnson motor is relatively compact, with a motor diameter of 27 mm and a length of 68 mm without the shaft. This makes it a suitable choice for applications where space is limited. Additionally, the motor's weight of 180g is relatively light, making it easy to integrate into various systems. The motor's recommended current range of 800 mA (no-load) and up to 7.5 A (max load) aligns well with the capabilities of various motor drivers.

Table 4.2.1 Datasheet for johnson motor

Parameter	Value
Base Motor RPM	18000
Gear Material	Metal
Rated RPM	100
Operating Voltage (VDC)	6 to 18

Nominal Voltage (V)	12
Rated Torque(kg-cm)	11.7
Stall Torque (Kg-Cm)	46
No-Load Current	300 mA
Load Current(A)	900 mA
Shaft Diameter (mm)	6
Shaft Length (mm)	25
Gearbox Diameter (mm)	37
Motor Diameter(mm)	27
Motor Length(mm)	68
Weight (gm)	180

4.2.2 Motor Driver

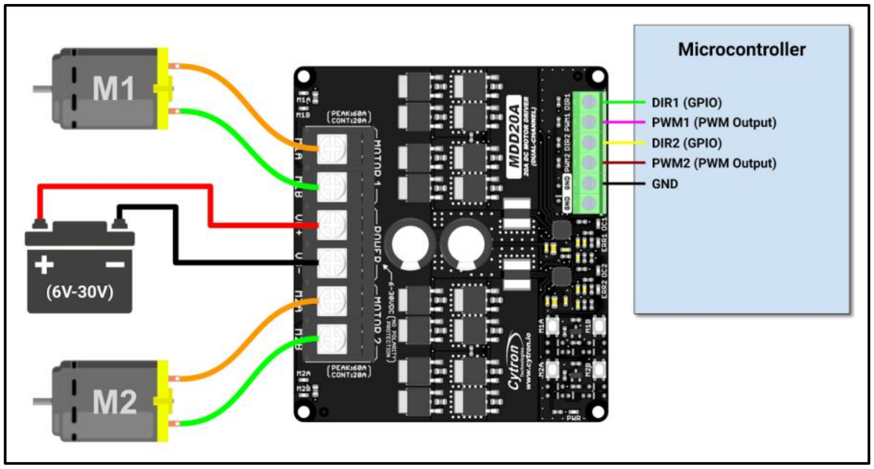

Fig. 4.2.2.1 Motor driver schematic

The MD20A Motor Driver plays a pivotal role in the autonomous food delivery robot's electrical system, seamlessly integrating with the microcontroller and motors to enable controlled and coordinated movement as depicted in Fig 4.2.2.1. The motor driver receives commands from the microcontroller based on sensor inputs and navigation algorithms, translating them to electrical signals that drive the motors. This efficient communication between the microcontroller and motor driver ensures that the robot can navigate complex environments and deliver food items flawlessly.

The MD20A Motor Driver from Cytron emerges as a robust and versatile solution for powering the motors of an autonomous food delivery robot. Its bidirectional control capabilities empower the robot to maneuver with precision, enabling smooth and efficient movement in both forward and backward directions. This capability is crucial for navigating complex environments, navigating tight spaces, and executing precise delivery maneuvers.

The operating voltage range of 6V to 30V renders the MD20A Motor Driver compatible with a wide spectrum of battery options as mentioned in Table 4.2.2, catering to various power requirements and extending the robot's operational range. This flexibility in power supply design ensures adaptability to different environments and delivery schedules. Further enhancing the motor control capabilities of the robot, the MD20A Motor Driver boasts a maximum continuous current of 20A and a peak current of 60A mentioned in Table 4.2.2. These robust specifications enable the robot to handle heavy loads of food and other delivery items, ensuring seamless and reliable operation even when carrying significant weight.

Table 4.2.2 Datasheet for Cytron Motor Driver

No	Parameters		Min	Max	Unit
1	Power Input Voltage		6	30	V
2	Maximum Motor Current (Per channel)	Continuous	-	20	A
		Peak *[1]	-	60	A
3	Logic Input Voltage (PWM & DIR)	Low Level	0	0.8	V
		High Level	1.5	15 *[2]	V
4	PWM Frequency *(Output frequency is same as input frequency)*	Standard	DC	20	KHz
		Extended *[3]	20	40	KHz

To further enhance the user experience and facilitate troubleshooting, the MD20A Motor Driver incorporates an LED indicator for motor output. This visual feedback provides real-time information on the motor's status and operation, allowing for immediate identification of any issues and prompt corrective actions. This feature contributes to optimal performance and ensures the robot's reliability in catering to the demands of food delivery operations.

Integrating the MD20A Motor Driver into the autonomous food delivery robot's control system establishes a robust foundation for precise movement, efficient navigation, and seamless delivery. Its bidirectional control, wide operating voltage range, powerful current handling capabilities, and integrated LED indicator for motor output combine to form a feature-rich and reliable solution for powering and controlling the robot's motors. This driver plays a pivotal role in enabling the robot to operate effectively and efficiently in various campus environments, ensuring timely and convenient food delivery for students and staff.

The interplay between current limit and temperature is crucial for the longevity and performance of the autonomous food delivery robot's motors. By dynamically adjusting the current limit based on motor temperature, the MD20A Motor Driver effectively balances power delivery with thermal protection, ensuring that the motors can operate at their full potential without compromising their integrity as shown in Fig. 4.2.2.2. This intelligent approach to motor control contributes to the robot's overall reliability and durability, ensuring that it can consistently deliver food to students and staff throughout the campus.

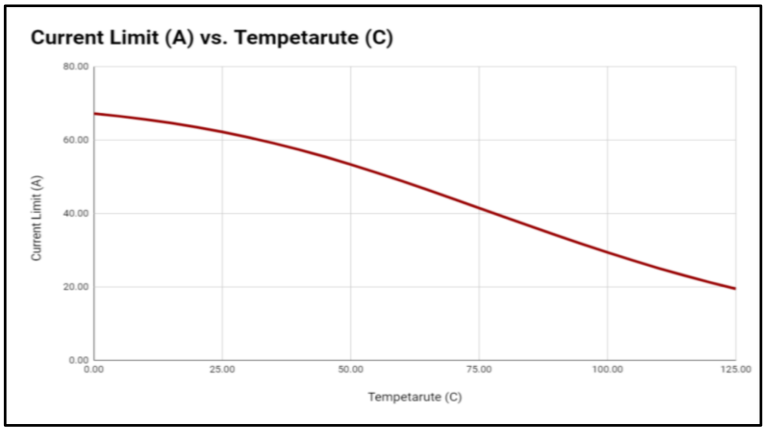

Fig 4.2.2.2 Current V/s Temperature Graph of Motor

4.2.3 Microcontroller

The Arduino Uno microcontroller serves as the heart of the autonomous food delivery robot, providing the processing power and control necessary for navigation, sensor data acquisition, and motor control. Equipped with 14 digital input/output pins, 6 analog input pins as shown in Fig 4.2.3, and a 16 MHz ATmega328P microcontroller, the Arduino Uno offers ample connectivity and computational capabilities for the robot's complex operations.

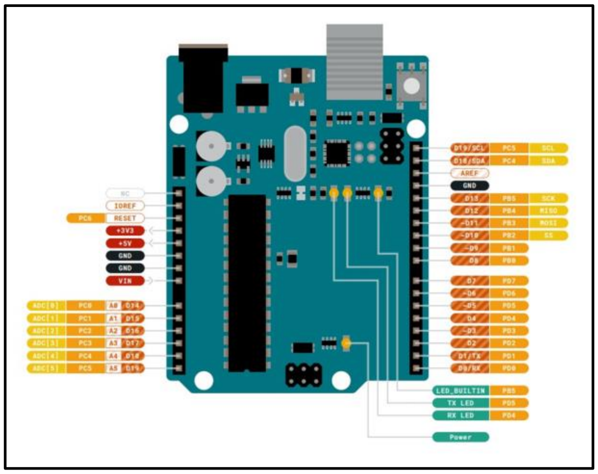

Fig. 4.2.3 Figure of Arduino UNO

The Arduino Uno's versatility extends to its power input options. The microcontroller's VIN pin can accept a maximum input voltage of 20V, allowing for direct connection to a suitable external power source. Alternatively, the USB B connector can provide power to the microcontroller, with a maximum input voltage of 5.5V as shown in Table 4.2.3.1. Additionally, the Arduino Uno can be powered through an external power jack, offering another convenient power supply option.

To ensure stable and reliable power supply, the Arduino Uno incorporates several essential components. Electrolytic capacitors filter out unwanted noise and ripple from the power supply, while voltage regulators provide stable voltages to the microcontroller and other onboard electronics. Rectifiers convert alternating current (AC) to direct current (DC) power, enabling the use of AC power sources if necessary.

The Arduino Uno's combination of processing power, connectivity, and power management capabilities makes it an ideal choice for the autonomous food delivery robot. Its versatility and ease of use allow for rapid prototyping and development, while its robustness ensures reliable operation in the demanding environment of a busy college campus.

Table 4.2.3 Datasheet for Arduino UNO

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

Table 4.2.4 Ports Interface for Arduino UNO

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358 LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

4.2.4 Battery

The Orange 10000mAh-11.1V (3s4p) shown in Fig. 4.2.4 lithium-ion battery stands out as a compelling choice for various applications due to its combination of high energy density, long battery life, fast charging capabilities, compact design, low self-discharge rate, versatility, environmental friendliness, stable voltage output, minimized memory effect, and built-in safety features.

Fig. 4.2.4 Picture of Orange 12V DC Battery

The battery's impressive 10000mAh capacity provides ample power for extended usage, making it ideal for powering electronic devices that demand substantial energy reserves. This high energy density is further complemented by its exceptional cycle life of over 500 cycles as shown in table 4.2.4, ensuring long-lasting performance and durability. The Orange 10000mAh-11.1V battery supports fast charging with a maximum current of 1C A (10000mA), enabling rapid recharging to minimize downtime and maximize convenience. Additionally, its lightweight and compact design makes it a portable power solution that can be easily carried and integrated into various devices.

Table 4.2.5 Datasheet for Lithium Ion Battery

Model No.	Orange 10000mah-11.1v (3s4p)
Nominal Capacity (mAh)	10000
Nominal Voltage (V)	11.1
Max. Charging Voltage (V)	12.6
Charging Cut-off Voltage (v)	8.25
Max. Charging Current	1C A (10000ma)
Nominal Charge Current	0.5C A (5000ma)
Maximum Discharge Current (A)	3C A (30000ma)
Nominal Discharge Current	1C A (10000ma)
Nominal Energy (wh)	111
Cycle Life /80%	≥500
Discharge Plug	XT-60 Female

In conclusion, the Orange 10000mAh-11.1V battery emerges as a powerful and reliable energy source, seamlessly combining high energy density, long battery life, fast charging, compact design, versatility and environmental friendliness which produces stable voltage output making it the ideal choice for our autonomous robot.

4.2.5 Lidar Sensor

The YDLidar X2 LiDAR sensor plays a pivotal role in the navigation system of autonomous food delivery robots, enabling them to navigate complex environments with precision and efficiency. Its 360-degree scanning capability and ranging accuracy of up to 15 meters provide the robot with a comprehensive understanding of its surroundings, allowing it to detect obstacles, create detailed maps, and localize itself accurately as shown in Table 4.2.6. This comprehensive environmental awareness is crucial for the robot to plan collision-free paths and navigate safely between designated points, ensuring timely and reliable food delivery services.

Fig. 4.2.5 Figure of YD X2 LiDAR

The YDLidar X2's compact and lightweight design makes it an ideal choice for integration into robot chassis without compromising maneuverability. The sensor's high scan frequency of 2000 Hz ensures that the robot receives real-time data on its surroundings, enabling it to respond promptly to changes in its environment and avoid obstacles effectively. This real-time data acquisition is essential for the robot to navigate safely and efficiently in dynamic environments, such as busy college campuses. In summary, the YD Lidar X2 LiDAR sensor stands as a critical component of autonomous food delivery robots, empowering them to operate with precision and autonomy. Its comprehensive environmental awareness, compact design, and real-time data acquisition capabilities make it an invaluable asset for reliable and efficient food delivery services across the campus.

Table 4.2.6 Datasheet for YD LiDAR

Items	Min	Typical	Max	Unit	Remarks
Ranging frequency	/	3000	/	Hz	Ranging 3000 times per second
Motor frequency	5	6	8	Hz	Need to connect to PWM ignal, recommended to use the speed of 6Hz

Ranging distance	0.12	/	8	m	Indoor environment with 80% Reflectivity
Field of view	/	0-360	/	Deg	/
Systematic error	/	2	/	cm	Range≤1m
Relative error	/	3.5%	3.5%	/	1m<Range ≤6m
Tilt angle	0.25	1	1	Deg	/
Angle resolution	0.60 (frequency @5Hz)	0.72 (frequency@6 Hz)	0.72 (frequency@6 Hz)	Deg	Different motor frequency

4.2.7 Xbox Kinect V2

The Xbox Kinect V2 camera shown in Fig. 4.2.7 also plays a pivotal role in the navigation system of autonomous food delivery robots, enabling them to navigate complex environments with precision and efficiency by depth tracking obstacles. It uses a time-of-flight sensor to generate depth maps, which are essential for robot navigation and obstacle avoidance.

ToF sensors measure the time it takes for emitted light to reflect off objects and return to the sensor. This allows the Kinect v2 to generate accurate and dense depth maps, even in low-light or challenging outdoor conditions. The Kinect v2 has a wide field of view of 70° x 60° as mentioned in table 4.2.7 for depth images, which allows robots to perceive their surroundings and identify potential obstacles from a distance.

Fig. 4.2.6 Figure of Kinect Camera V2

This is particularly important for autonomous robots that operate in dynamic environments, such as warehouses or factories. It can track the skeletons of up to six people simultaneously, with 25 joints defined per skeleton. This robust skeletal tracking capability can be leveraged by autonomous robots to interact with humans in a natural and intuitive way. For example, robots can use skeletal tracking to detect and respond to human gestures, or to avoid colliding with humans

The Kinect v2 can be used to generate dense and accurate depth maps of the robot's surroundings, which can be used for path planning and obstacle avoidance. This makes it an ideal choice for the depth camera.

Table 4.2.7 Datasheet for XBOX Kinect V2

Features	Kinect v2
Depth sensor type	Time of Flight (ToF)
Red, Green & Blue (RGB) camera resolution	1920 x 1080, 30 fps
Infrared (IR) camera resolution	512 x 424, 30 fps
Field of view of RGB image	84.1° x 53.8°
Field of view of depth image	70° x 60°
Operative measuring range	0.5 m x 4.5 m
Skeleton joints defined	25 joints
Maximum skeletal tracking	6

4.2.8 Overall System Overview with Flowchart

The Kinect and LiDAR sensors, functioning as the robot's eyes, continuously gather information about the surrounding environment. The Kinect sensor, employing infrared light technology, generates a 3D depth image of the immediate surroundings, aiding the robot in perceiving obstacles, identifying objects, and comprehending its spatial context. On the other hand, the LiDAR sensor emits pulses of laser light, measuring distances to objects and creating a highly detailed 3D point cloud. This comprehensive mapping enables the robot to navigate precisely and avoid collisions.

The accumulated sensory data is then transmitted to the microcontroller, the robot's intelligent brain. As the processing unit, the microcontroller receives the sensor data and analyzes it using sophisticated algorithms. It interprets the 3D maps, identifies obstacles, and determines the robot's position and orientation. Based on the processed sensory information, the microcontroller meticulously plans the robot's path, considering obstacles, destination points, and potential hazards.

It optimizes the route to ensure efficient and safe navigation.

The microcontroller generates control signals, which serve as the instructions for the robot's movement. These control signals are conveyed to the motor drivers, electronic components that act as intermediaries between the microcontroller and the motors. They receive the control signals and translate them into electrical impulses that the motors can understand. The electrical impulses from the motor drivers power the robot's motors, converting electrical energy into mechanical motion. The motors rotate based on the received signals, propelling the robot forward, backward, or turning it accordingly.

To fuel the robot's operation, a reliable power source provides the necessary energy. Typically, a rechargeable battery pack forms the robot's primary power source. The battery stores electrical energy, supplying the current needed for the motors, sensors, and microcontroller. Efficient power management systems regulate the energy consumption of the robot's components, ensuring optimal performance and extending battery life. The entire flow of signal is depicted in Fig. 4.2.7.

In essence, the autonomous mobile robot's operation resembles a continuous feedback loop:

- Sensors gather data about the environment.
- The microcontroller analyzes this data and plans the robot's path.
- Control signals are sent to the motors via motor drivers.
- Motors convert electrical impulses into mechanical motion, propelling the robot.
- Sensors continue to gather data, completing the feedback loop.

This synchronized interplay between sensors, processing power, and motors is what empowers the autonomous mobile robot to navigate complex environments, avoid obstacles, and deliver food items to designated locations. The robot's ability to perceive its surroundings, make intelligent decisions, and execute controlled movements makes it an invaluable asset for providing timely and efficient food delivery services across the college campus.

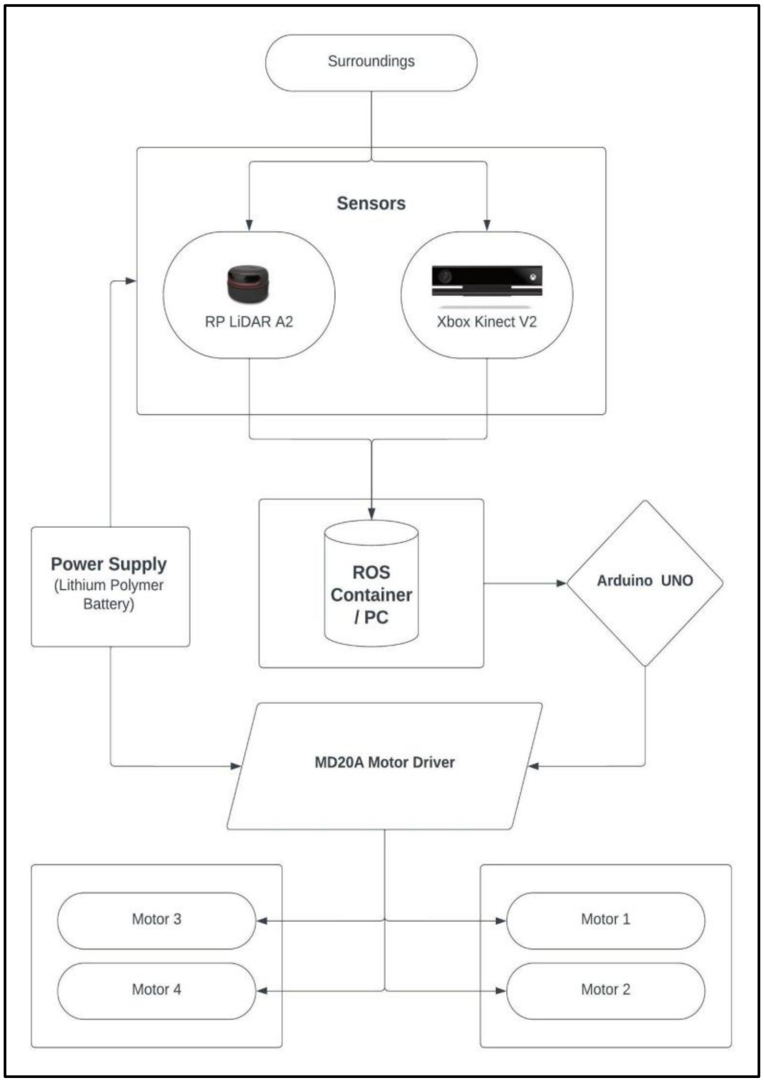

Fig 4.2.7 Flow chart showing working of system

4.3 ROS Framework Overview

ROS is a flexible framework based on a publisher subscriber for building robot applications. It is a collection of tools, libraries, and conventions that simplify the process of creating robots. ROS provides a common communication infrastructure for all of the nodes on a robot, making it easy to combine different pieces of software to create complex systems. It also provides a variety of tools for debugging and developing robot applications. This makes ROS the ideal choice for this project since it provides a platform for efficient communication between all the executables.

The ROS package developed for this project has the structure depicted in Fig. 4.3.1

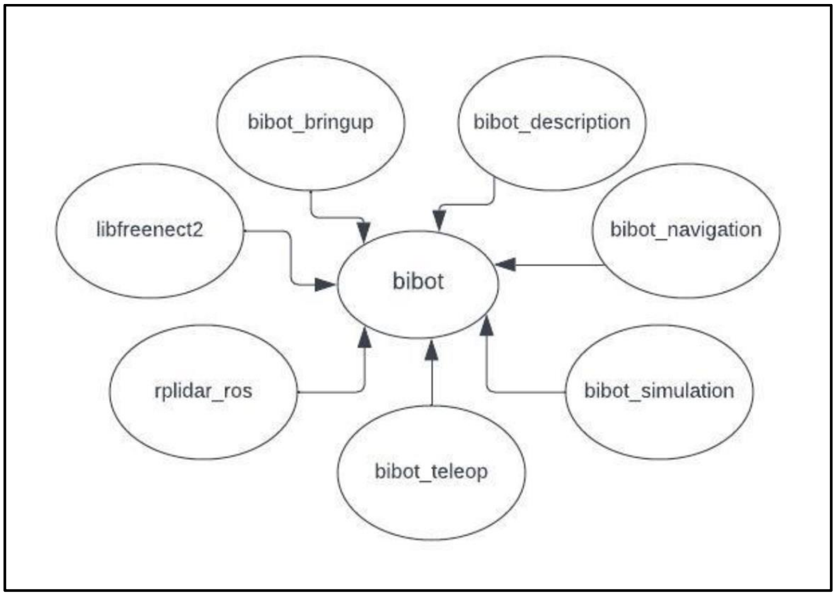

Fig. 4.3.1 Executables inside the ROS package

- **bibot_bringup**: This package contains the launch files to simultaneously run the nodes to activate and launch our sensors i.e., LiDAR and RGB-D camera.
- **bibot_description**: This package launches the description of our robot in our simulation software i.e. RVIZ using a URDF file prepared by us tailored to the exact specifications and measurements of our robot.
- **bibot_navigation**: This package launches the nodes related to mapping, localization and point to point navigation. It contains nodes that apply the algorithms such as SLAM, Adaptive Monte Carlo Localization (AMCL), Real Time Appearance Based Mapping (RTABMAP) and move base.
- **bibot_simulation**: This package contains the simulated world along with the mesh files of the robot and our sensors. It was used to carry out the initial testing of our robot in an outdoor environment in a virtual world before translating it to the actual robot.
- **bibot_teleop**: This package controls the robot using telemetry in the initial stage of mapping of the environment using the sensors.
- **ydlidar_ros**: This package is interfaced with LiDAR through the serial ports of the computer and collects 2D point cloud data from the surroundings.
- **libfreenect2**: This is a package to interface with the RGB-D camera through the serial ports of the computer and to collect 3D point cloud data from the surroundings.

4.3.1 Teleoperation:

The first step towards developing an AMR is to set up a communication between the robot and the human to manually operate the machine. This is necessary because tasks like mapping the environment which are prerequisites for autonomous navigation are created by manually exposing the robot to unknown environments. This communication allows the human to control the robot using teleoperation. The teleoperation takes place by leveraging a ROS package called rosserial.

Serial communication using ROSSerial is a convenient method for establishing a connection between a microcontroller board, such as an Arduino, and a ROS computer. It enables the exchange of data between the two devices, allowing the microcontroller to send sensor readings or control signals to the ROS system, and for the ROS system to send commands or configuration parameters to the microcontroller.

Serial communication using ROSSerial involves a few steps. First, you connect the microcontroller board to the ROS computer using a USB-to-serial adapter or a direct serial connection. Next, you need to install the ROSSerial package on both the ROS computer and the microcontroller board. This package provides the necessary communication protocols and drivers. Once the ROSSerial package is installed, you need to configure the ROSSerial node on the ROS computer. This is where you specify the serial port, baud rate, and other communication parameters. Finally, you need to develop the ROS node on the microcontroller board. This node will handle the data exchange with the ROS system. It will typically subscribe to ROS topics for receiving commands or configuration parameters and publish sensor data or control signals to ROS topics.

With these steps done, the microcontroller board can seamlessly integrate with the ROS system, enabling real-time data exchange and control between the two. This allows serial communication which ultimately facilitates the ability to send inputs to the robot in order to control it manually. The basic flow of information can be seen in Fig. 4.3.2.

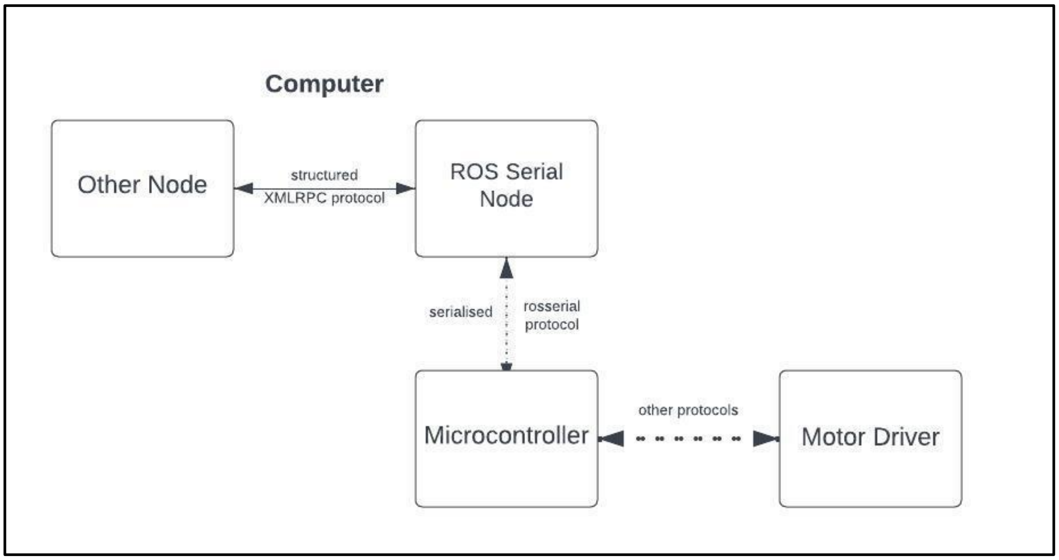

Fig. 4.3.2 Working of rosserial to set up serial communication

After setting up communication using rosserial we move onto the working, interfacing and implementation of the sensors i.e. the 2-D LiDAR and the RGB-D camera. This will be discussed in the upcoming sections.

4.3.2 LiDAR (YDLiDAR X2):

After teleoperation, the first step is to integrate the sensors into the system in order for the robot to be able to interact with the external environment. The first sensor to be integrated is the 2-D LiDAR, this laser sensor is mainly used to accurately tell the distance between any obstructions that may appear in the path of the robot. First, two-dimensional point cloud data of the environment is gathered in real time using the ROS executable, this data can be visualized using visualization tools such as Rviz. It can be seen in Fig. 4.3.3

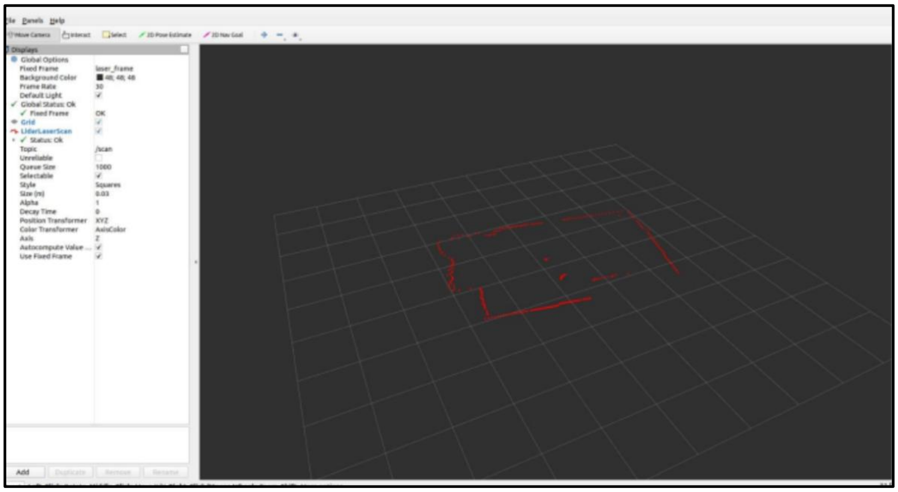

Fig. 4.3.3 Point cloud data obtained from LiDAR

This point cloud data that is gathered is used to make a two dimensional map of the environment using an algorithm called Simultaneous Localization and Mapping (SLAM). The specific version of SLAM that was implemented in this project to achieve a 2-D map of the environment is called Hector SLAM.

Hector SLAM is a real-time simultaneous localization and mapping (SLAM) algorithm that can be used to build 2D maps of environments using laser scanner data. It is a particle filter-based algorithm, which means that it maintains a set of possible poses for the robot and updates the probability of each pose based on the laser scanner data. Hector SLAM is also able to incorporate odometry data, if available, to improve the accuracy of the map.

The algorithm starts by initializing the map with a small number of particles, each representing a possible pose for the robot. As the robot moves, the algorithm updates the pose of each particle based on the odometry data (if available). The algorithm then matches the current laser scan to the map, determining the most likely transformation between the scan and the map for each particle. The algorithm updates the probability of each particle based on the goodness of fit of the scan matching. Particles that match the scan well have their probabilities increased, while those that do not have their probabilities decreased. The algorithm resamples the set of particles, discarding particles with low probabilities and creating new particles from particles with high probabilities. This ensures that the particle filter maintains a diverse set of possible poses. The algorithm updates the map by adding new

features from the laser scan to the map and updating the positions of existing features based on the most likely transformation.
Finally, the map is visualized in RViz, a 3D visualization tool for ROS. The overall process can be seen in Fig. 4.3.4

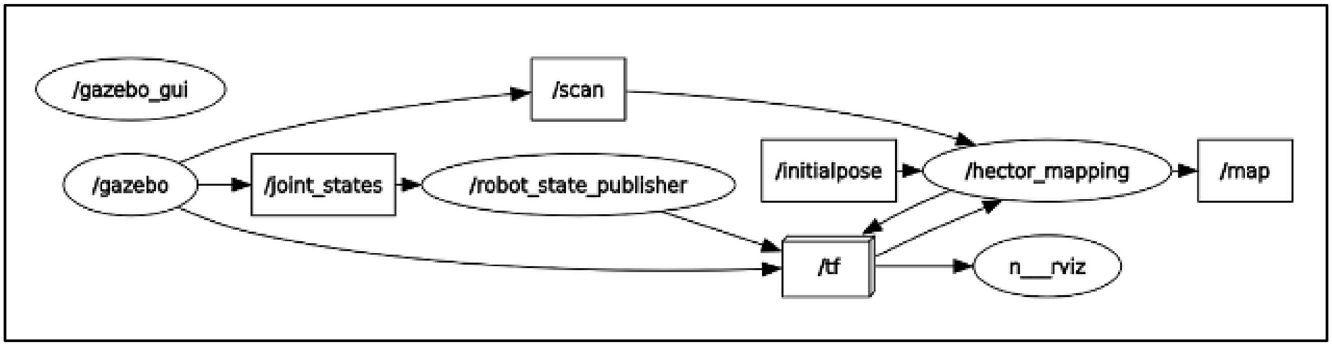

Fig. 4.3.4 Flow of information between ROS topics and nodes in Hector SLAM

After creation of the map, the next step is to localize the robot within it. This can be done by integrating the point cloud data obtained from the LiDAR in real time with the map to estimate a pose of the robot in the map. This was implemented using an algorithm called Adaptive Monte-Carlo Localization (AMCL).

Adaptive Monte Carlo Localization (AMCL) is a probabilistic localization algorithm commonly used in robotics for real-time 2D pose estimation. It employs a particle filter approach, maintaining a set of potential poses for the robot and updating their probabilities based on sensor measurements, primarily laser scans. AMCL's adaptive nature allows it to adjust the number of particles dynamically, ensuring computational efficiency while maintaining accuracy.

The algorithm begins by initializing the map with a handful of particles, each representing a possible pose for the robot. When the robot moves, AMCL updates the poses of each particle based on odometry data, if available. Odometry provides an estimate of the robot's movement, incorporating it into the particle filter. AMCL then matches the current laser scan to the map, determining the most likely transformation between the scan and the map for each particle. This involves comparing the laser scan's features to the map's features and evaluating their alignment. Based on the goodness of fit of the scan matching, AMCL updates the importance weight of each particle. Particles that match the scan well have their weights increased, indicating their higher likelihood of representing the robot's true pose. Conversely, particles with poor scan matching have their weights decreased.

AMCL resamples the set of particles, discarding particles with low weights and creating new particles from particles with high weights. This ensures that the particle filter maintains a diverse set of possible poses, focusing on the more likely ones. The algorithm estimates the robot's pose by calculating the weighted average of the positions of all particles. This weighted average represents the most probable location of the robot, considering the entire particle distribution. AMCL's adaptive nature lies in its ability to adjust the number of particles based on the complexity of the environment. In simpler

environments, AMCL can reduce the number of particles, conserving computational resources. Conversely, in more complex environments with intricate features, AMCL increases the number of particles to maintain accuracy.
The ROS package used to interface this algorithm is a part of the navigation stack developed for this project titled bibot_navigation. The overall process of implementing AMCL for the purpose of localization can be observed in Fig. 4.3.5

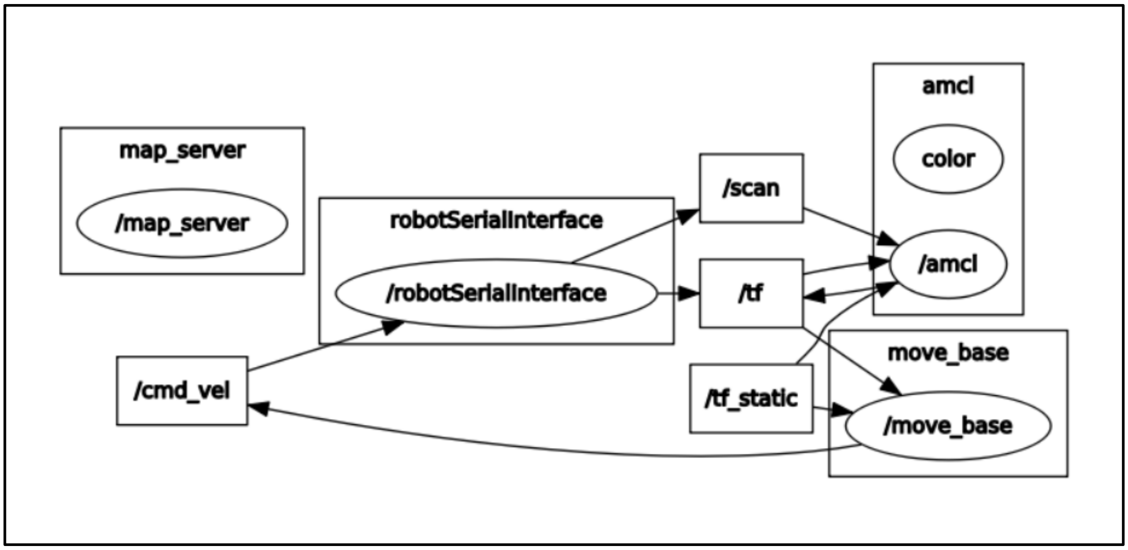

Fig. 4.3.5 Flow of information between ROS topics and nodes in AMCL

The LiDAR has a few shortcomings that are covered by the RGB-D camera such as the two-dimensional nature and the low density of its point cloud data. The next sensor to be integrated into the system is the RGB-D camera that is the Kinect v2. This is discussed in the upcoming section.

4.3.3 RGB-D Camera (XBOX Kinect v2):

RGB-D cameras play an increasingly important role in the field of autonomous navigation, offering several benefits over traditional cameras and lidar sensors. While RGB cameras provide rich color information, they lack depth perception, making it challenging to estimate distances accurately. Lidar sensors, on the other hand, excel at providing precise depth measurements but lack color information.

By combining the strengths of RGB cameras and lidar sensors, RGB-D cameras overcome the limitations of each individual sensor. The RGB data provides contextual information about the environment based on its three-dimensional point cloud data, such as the presence of objects, their colors, and textures. This information is crucial for tasks like object recognition and classification, which are essential for autonomous navigation. LiDAR data, on the other hand, provides accurate depth measurements, enabling the robot to perceive the three-dimensional structure of its surroundings. This information is crucial for obstacle avoidance and path planning, ensuring that the robot can safely navigate through its environment.

Together, RGB-D cameras and LiDAR sensors form a powerful combination for autonomous navigation. The RGB data provides context and color information, while the lidar data provides accurate depth measurements. This combination enables robots to perceive their surroundings more effectively, making them more capable of navigating autonomously in complex environments.

The first step to interface the firmware of the Kinect V2 with the entire ROS framework is to implement the libfreenect_2 ROS package. This enables the user to calibrate the camera to their required specifications and use cases. This also enables the ability to collect camera's three-dimensional point cloud data.

Calibrating the Kinect v2 using ROS involves aligning the RGB camera and depth sensor data to ensure accurate spatial perception. This process is essential for tasks like object recognition, obstacle avoidance, and 3D reconstruction. The calibration process can be described using Fig. 4.3.6, its calibration process typically involves the following steps:

- **Intrinsic Calibration:** This involves calibrating each sensor individually to determine its internal parameters, such as focal length, optical center, and distortion coefficients. This can be done using specialized calibration tools or software.

- **Extrinsic Calibration:** This involves determining the relative transformation between the RGB camera and the depth sensor. This can be done using various techniques, such as checkerboard patterns or point clouds

Fig. 4.3.6 A typical calibration setup

After calibration, mapping using the point cloud data of the kinect is done. The Kinect v2 sensor can generate detailed 3D maps of indoor and outdoor environments using its RGB camera and depth sensor. When integrated with the Robot Operating System (ROS), the Kinect data can be converted into usable point clouds. As the Kinect is moved through a space, ROS accumulates the point cloud data into a global map using simultaneous localization and mapping (SLAM) techniques like a particle filter or pose graph optimization. These algorithms estimate the pose of the Kinect over time while closing loops to reduce drift. For example, when the Kinect recognizes a previously mapped area after moving through new spaces, constraints can be added to link the old and new areas together consistently. Sophisticated SLAM approaches enable accurate 3D maps to be built up across multiple closed loops despite real-world noise and accumulated error in estimating the sensor motion. The resulting Kinect+ROS 3D map provides a rich virtual representation of indoor spaces. A map generated using this method can be seen in Fig. 4.3.7

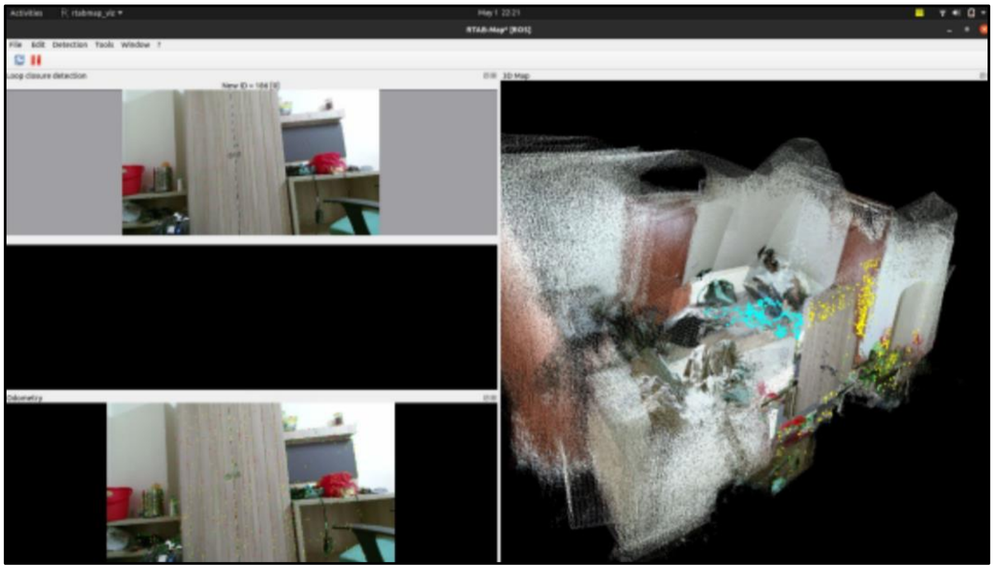

Fig. 4.3.7 3D map of the environment

4.3.4 Sensor Fusion:

When the individual testing and implementation of the sensors is done successfully, we move on to fusing the sensor data in order to take advantage of the complementary nature of their advantages and shortcomings. The fusion of sensor data is done mainly using an algorithm called Real Time Appearance Based Mapping (RTABMAP). The ROS package used to implement this is called rtabmap_ros.

RTAB-Map (Real-Time Appearance-Based Mapping) is a powerful SLAM (Simultaneous Localization and Mapping) library that can generate highly accurate 2D and 3D maps of indoor and outdoor environments by fusing data from one or more sensors. By supporting a wide array of sensory inputs like LIDAR, RGB-D cameras, stereo cameras, and regular monocular RGB cameras, RTAB-Map can fuse information from these complementary sources into a detailed environmental map containing both spatial and visual information. The key advantage is combining data that spans a wide field of view, high depth accuracy, robust texture descriptions, and other beneficial attributes from the different sensors.

The mapping process begins by converting raw sensor feeds into usable point cloud data. The rotating LIDAR unit scans an environment with laser beams that sweep across a 2D or 3D field of view, reflecting range measurements that provide a high-density sampling of depth information surrounding the sensor. These discrete measurements are assembled into a geometric point cloud with X, Y, Z coordinates defined for each point. The RGB-D camera captures color images along with per-pixel depth from infrared projection patterns. By registering depth to color pixels, this produces a colored point cloud representation where every visible surface point is described with location plus RGB color properties. These complementary data streams can then be integrated together into a combined point cloud with rich geometry and appearance information about the environment.

As RTAB-Map collects these sensor streams during robot motion or hand-held sweeping motions, it performs simultaneous localization and mapping to enable globally consistent maps. This process tracks the 6-DOF pose of sensors over time using visual features detected in the RGB or RGB-D camera streams. By identifying visual landmarks reappearing across the image sequence, such as through keypoint matching, RTAB-Map can detect loop closures to refine its pose graph and reduce drift. Robust loop closure detection allows earlier parts of trajectories to be aligned to newer areas when returning to a previously visited location. With aligned sensor data aggregated from diverse viewpoints along optimized trajectories, RTAB-Map incrementally integrates observations into surface maps and volumetric representations.

For surface mapping, RTAB-Map's RGB-D mapping pipeline projects point clouds into a tessellated mesh structure constructed out of triangles. This allows fusing multiple scans more efficiently than raw point clouds while still representing detailed geometric surfaces including fine contours and holes. For volumetric mapping, point clouds are raycast into a 3D voxel grid or octree structure with variable resolution. This allows dense space near the sensors while efficiently approximating free space further away. Surfaces detected from multiple directions reinforce voxel occupancy probabilities. Textured descriptors can also be stored for each voxel face, encoding what colors associate with observed surfaces from different viewpoints. The output is a 3D occupancy map describing surfaces along with visual properties where available.

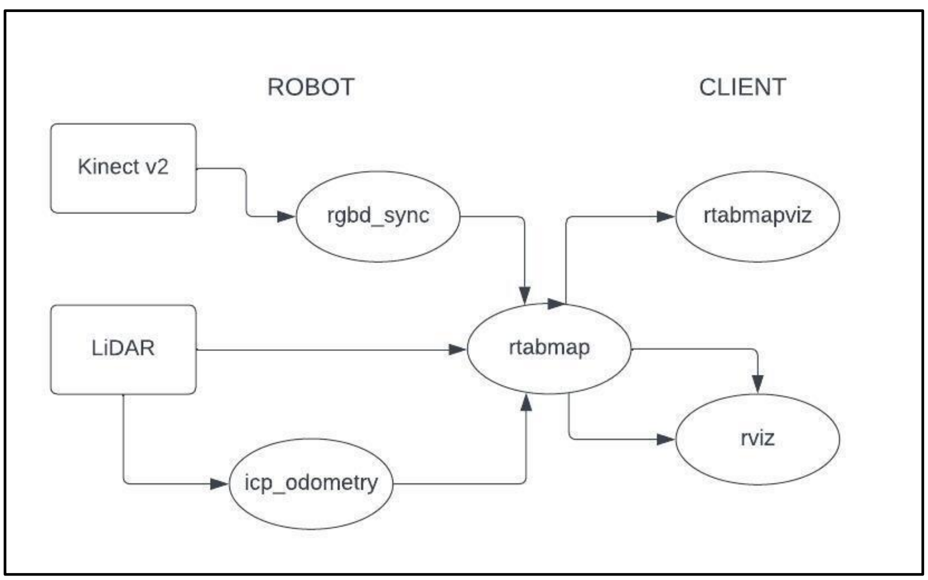

Fig. 4.3.8 Ros nodes and topics flow while fusion

By smartly combining data sources to mitigate limitations of individual sensors, RTAB-Map provides highly versatile mapping suitable for robotics applications like navigation, manipulation, and environmental awareness. The system interfaces with ROS for convenient integration but can also run in standalone mode when onboard hardware resources allow. With configurable data fusion pipelines and adjustable parameters, RTAB-Map remains efficient enough to run on computationally limited platforms like embedded systems on small unmanned aerial vehicles. The versatility of RTAB-Map makes it a great choice for accurately mapping environments across many contexts and use cases.

This critical step, facilitated by the rtabmap_ros package, is elucidated showing ros nodes communication in Fig. 4.3.8 and its pipeline in Fig. 4.3.9.

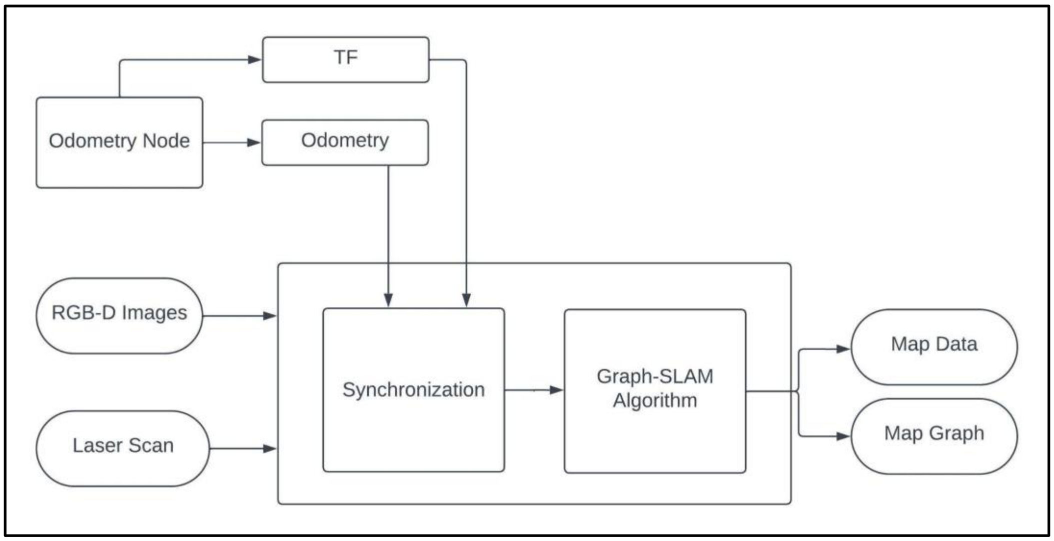

Fig. 4.3.9 Pipeline of sensor fusion for mapping.

After successfully achieving sensor fusion and creating a map using the same, the next step is to implement the navigation stack developed for this project in order to actually perform the autonomous navigation. This is discussed in the next section.

4.3.5 Navigation:

Achieving reliable autonomous navigation hinges on integrating robust global path planning, local trajectory optimization, and reactive obstacle avoidance behaviors. Our system leverages the move_base flex navigation framework within the Robot Operating System (ROS) which facilitates this via specialized pipeline modules. The global costmap handles discretized environmental representations for global path searches while the local costmap processes raw LIDAR and RGB-D point clouds for localized obstacle avoidance. The ros package for move has a data flow that is depicted in Fig. 4.3.10.

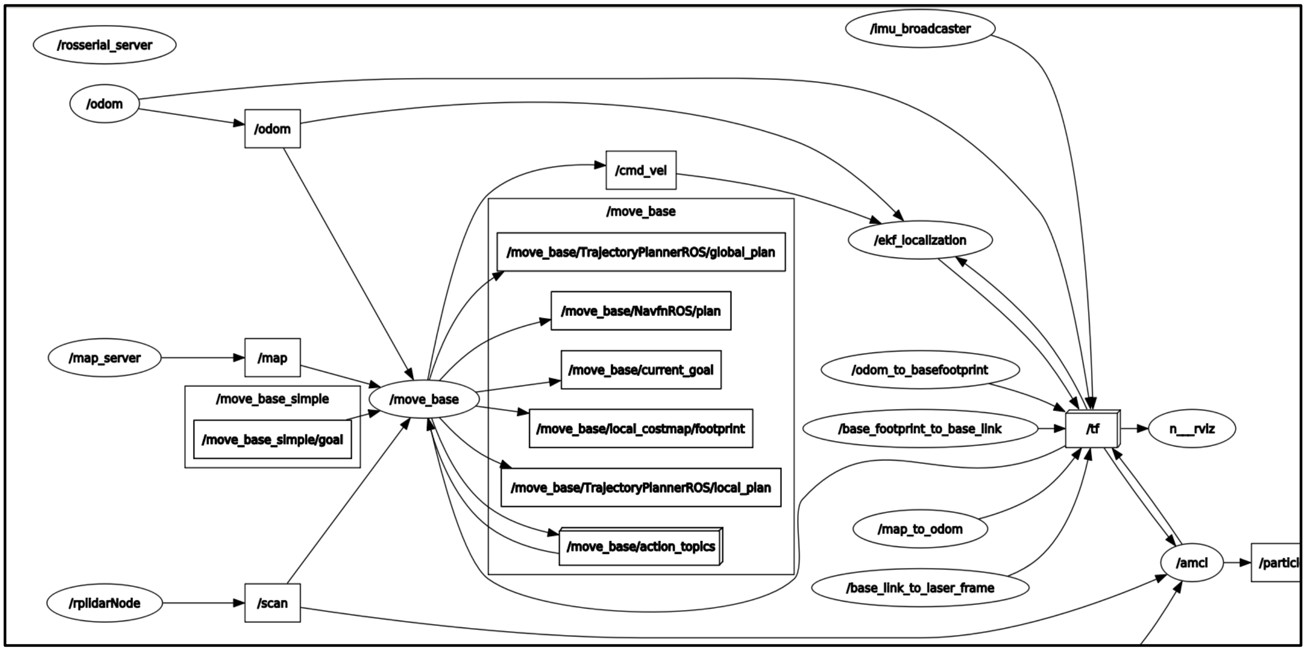

Fig. 4.3.10 Flow of information between ROS topics and nodes in Move Base

The global planner module implements the NavFn algorithm, applying fast marching methods to perform Dijkstra searches on a 2D static occupancy grid map. This identifies the lowest cumulative cost path connecting start and goal poses using an approximate clearance heuristic to maximize drivable space. The output temp lattice consists of finely spaced waypoints guiding topological movement. Waypoint density balances resolution and processing constraints. The resulting geometric path lacks finer details but provides an achievable overall route through known free space.

The Controller layer transforms the templattice into velocity setpoints tracked by a local trajectory regulator. Our nonlinear Model Predictive Controller (MPC) optimizes 2-3 meter dynamic windows minimizing error to the path subject to process/measurement noise constraints using the robot's kinematic model. Costs penalize deviation while satisfying acceleration/curvature constraints for dynamic feasibility. The MPC continuously re-plans within a fast control loop to enable reactive corrections.

The local costmap filters the latest LIDAR scans and RGB-D point clouds into a local 3D occupancy voxel grid applying voxel hashing for efficient lookups. This detects nearby obstacles not captured within prior static maps. The Dynamic Window Approach (DWA) local planner then searches for feasible velocity vectors scoring trajectories reachable within a time horizon respecting kinodynamic constraints. The minimum cost trajectory typically follows global waypoints while collision checking against local costmap obstacles. Final commands filter trajectories smoother leveraging an Extended Kalman Filter (EKF) respecting motion uncertainties.

This navigation pipeline demonstrated reliable site-wide autonomous traversal within an inventory warehouse. Additional testing under challenging edge cases for congestion, narrow passages, low visibility, and cutting through obstruction debris further validate robust implementations. Ongoing work looks to optimize sensor fusion, increase top speeds by predicting pedestrian motions, and incorporate safety constraints compliant to ISO 3691-4 standards. These enhancements promise to unlock next-generation high speed autonomous navigation in complex dynamic environments.

The full working of the move base can be visualized in Fig. 4.3.11.

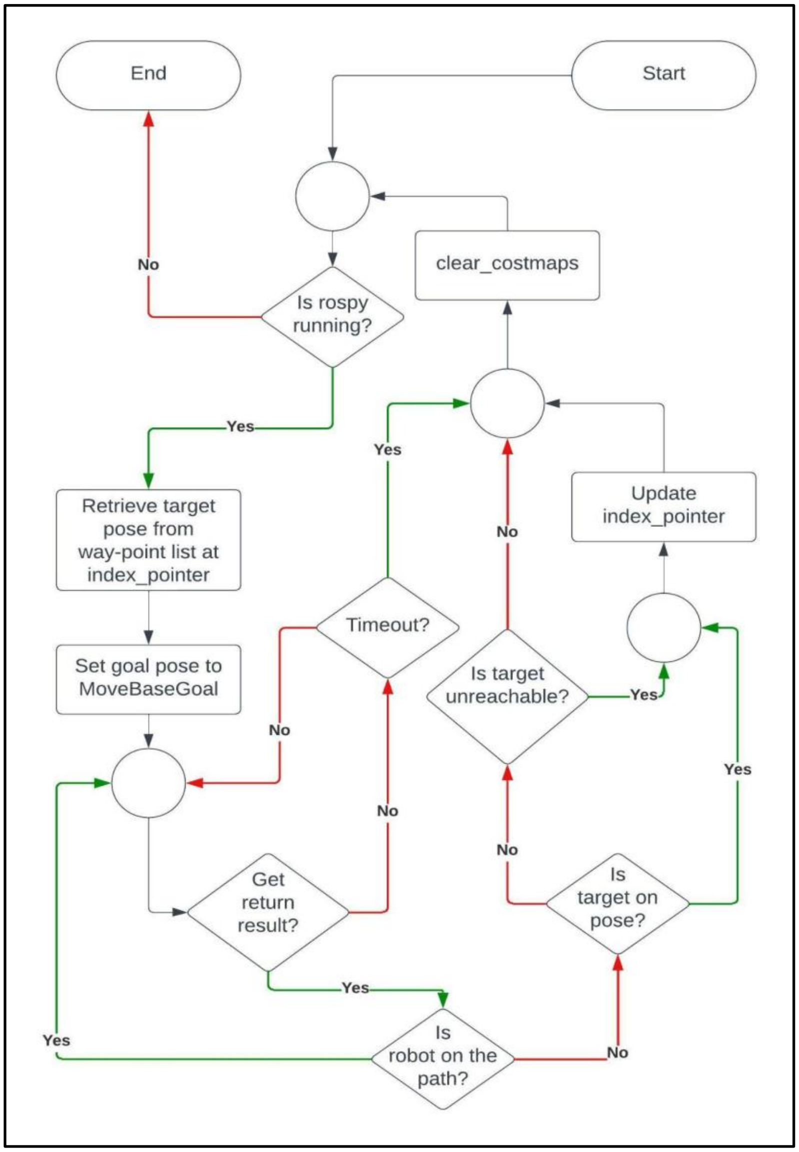

Fig. 4.3.11 Flow chart for the autonomous navigation

In the specific context of point-to-point navigation utilizing fused LiDAR and RGBD camera data, sophisticated path planning algorithms are employed to derive a global, obstruction-free trajectory. However, it is imperative to note that real-time obstacle avoidance is not inherently integrated into this autonomous navigation framework. To address this limitation, the system dynamically crafts a local obstacle-avoidance plan contingent on the current sensor data when confronted with obstacles. This localized plan transiently diverts from the global trajectory, ensuring obstacle circumvention, before seamlessly rejoining the global path post-obstacle traversal. This combined methodology results in a system proficient in autonomous navigation, adeptly responsive to dynamic environmental impediments.

4.3.6 Simulation:

After all ROS executables are developed and integrated within the system, the next step is to compile the nodes and set up a publisher and subscriber infrastructure between the nodes i.e. the ROS executables and the ROS master. This is first done in a real time physics simulator called Gazebo. The simulation environment that was chosen to test the robot out in can be observed in Fig. 4.3.11

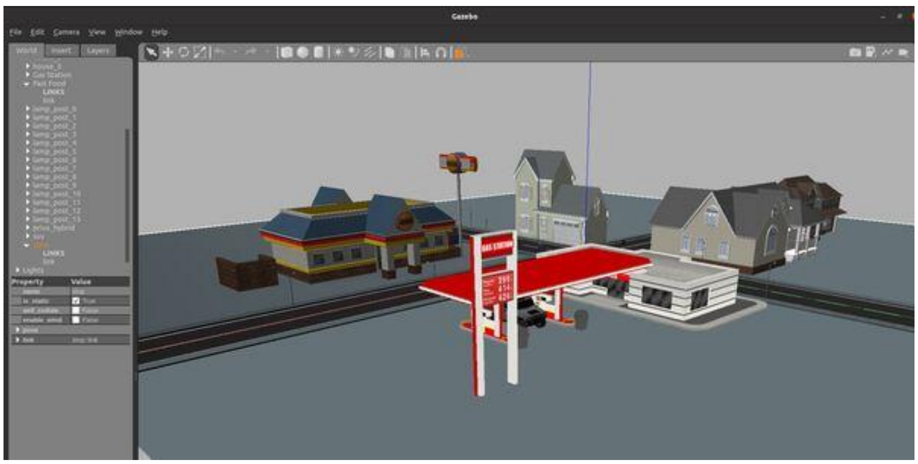

Fig. 4.3.12 Simulation environment

In order to get an accurate representation of the robot which is consistent with the measurements and specifications of the actual robot, a Universal Robot Description Format (URDF) file of the robot is created. The URDF file allows us to run the robot in a simulated environment. Mapping of the environment in the simulated environment can be observed in Fig. 4.2.12

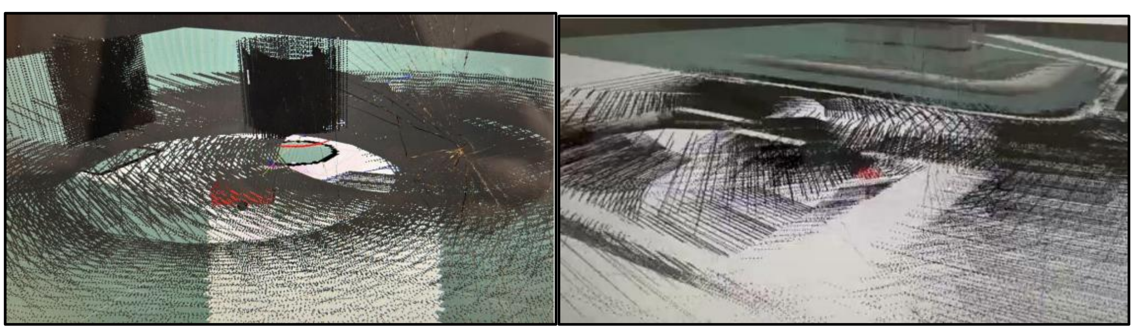

Fig. 4.3.13 Mapping in the simulation environment

4.4 Project Description

This project aims to develop an intelligent autonomous robot capable of reliably delivering food items across a college campus. The robot will employ state-of-the-art sensors for perception and advanced algorithms for navigation.

A key objective is achieving robust simultaneous localization and mapping (SLAM) by fusing data from a 2D lidar and RGB-D camera. The complementarity of these sensors will enable accurate 3D semantic mapping. The system will implement graph optimization techniques for map correction over long trajectories minimizing drift. Localization will leverage an adaptive Monte Carlo algorithm with particle filtering for resilience in dynamic environments.

For planning optimized delivery routes, Dijkstra's algorithm will search known paths while dynamically rerouting based on updated occupancy grids from sensor feeds. A model predictive control approach will generate smooth trajectories tracking reference waypoints. The system will fuse global and local planning to efficiently navigate towards goals while avoiding obstacles.

Safety is paramount for campus deployment. Extensive testing in high-fidelity Gazebo simulation will validate performance prior to real-world operation. The user interface will incorporate one-click emergency stops and teleoperation. Modular software architecture using Robot Operating System (ROS) promises maintainability and extensibility.

Success promises increased accessibility of food services while reducing human drudgery. This technology demonstration will set the stage for larger-scale autonomous delivery fleets. The systems integration experience confers valued engineering capabilities scalable to multifarious robotics domains such as last-mile logistics, surveillance, and precision agriculture.

4.5 UG Subjects:

● Electronic Engineering (UEC001)

● Engineering Design Project-II (UTA024)

● Artificial Intelligence (UCS411)

4.6 Standards Used:

● 1872-2015: IEEE Standard for Ontologies for Robotics and Automation

● 1872.2-2021: IEEE Standard for Autonomous Robotics (AuR) Ontology

● 1873-2015 - IEEE Standard for Robot Map Data Representation for Navigation

● P2936 - IEEE Standard for Test Methods of Automotive LiDAR Performance

● 802 - IEEE for wireless standard

Chapter 5-Outcomes and Prospective Learnings

5.1 Results and Discussion

The development of an autonomous mobile robot involved a variety of experimentation and testing. This section briefly describes the outcomes obtained after the experiments are performed and additionally discusses the results obtained. The final structure of robot is shown in Fig. 5.1.

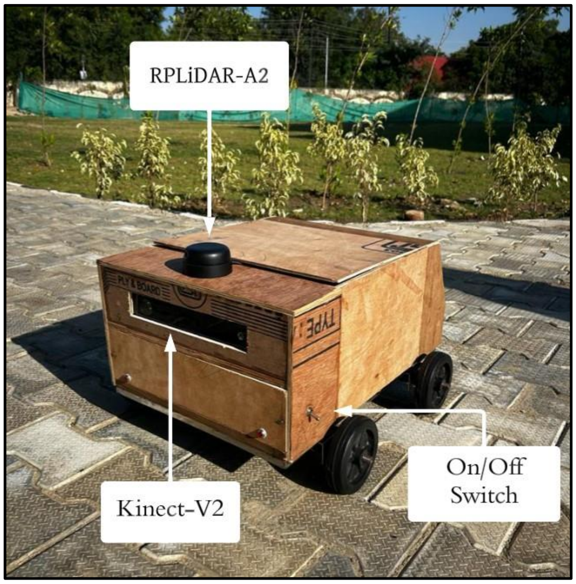

Fig. 5.1 Final Structure of Autonomous Robot

The early phase focused on optimizing mapping capabilities leveraging our sensor fusion pipeline combining LIDAR and RGB-D camera data streams. After calibrating intrinsic and extrinsic parameters to enable properly aligned point clouds, we tuned fusion algorithms targeting completeness, precision, and processing efficiency. The parameterized voxel hashing approach for merging data modalities provided adaptable tradeoffs, such as between map resolution and compute time. We performed extensive mapping runs across wide-ranging indoor and outdoor environments under various lighting conditions and dynamic elements. Through quantitative analysis of fusion results against ground truth measurements, we converged on optimal voxel sizes, hash table configurations, outlier rejection thresholds, and other hyperparameters for this application space. The finalized mapping framework reliably produced rich 3D reconstructions of spaces traversed during data collection runs, including difficult geometries like staircases. Position tracking leveraging visual, inertial, and kinematic data facilitated globally consistent alignments even during longer sequences prone to drift. Fig. 5.1 illustrates mapping outcomes on the top floor of our building, representing traversable spaces and obstacles with precision suitable for navigation.

Fig. 5.2 Map generation of C-Block top floor

With robust mapping established, we next focused on autonomous navigation capabilities. Our primary test environment centered on the top floor of C-Block at Thapar University spanning over 100 sq. meters across hallways, offices, atriums, ramps, and balcony areas. This space offered diverse geometries and uneven textures with both static and dynamic obstacles, providing an ideal complex navigation benchmark. Between these spaces, we subjected our navigation stack to extremely rigorous trials under dense pedestrian traffic, low lighting, reflective surfaces, obstructed spaces, congested paths, and other challenging edge cases exceeding expected real-world deployment conditions.

During the culminating navigation validation runs, our robot successfully traversed paths up to 150 meters in length across these spaces utilizing only onboard sensing and processing. Fig. 5.2 illustrates the final fused sensor representation of the core C-Block test site provided to the navigation subsystem.

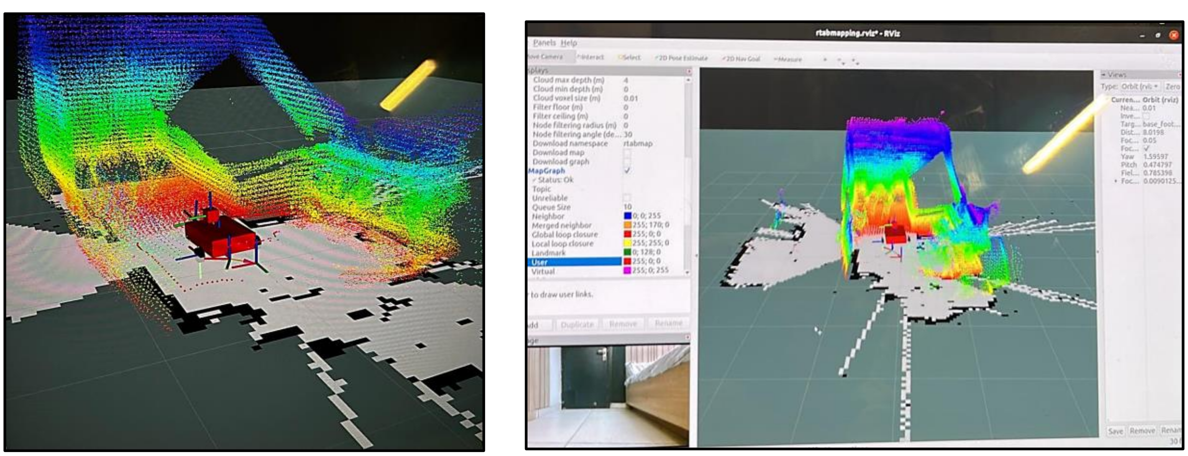

Fig. 5.3 Mapping of testing environment using fused sensor data

Fig. 5.3 captures timestamped images during an example autonomous traversal sequence demonstrating live sensing, planning, and obstacle avoidance behaviors. The corresponding route is shown in Fig. 5.4

plotting the path overlayed on the test site map with progression marked along the timeline. Quantitative metrics captured include a 92% success rate reaching goals across over 500 separate navigation trials. In the 8% of failure cases, the root issues were narrowed down to dynamic elements violating safety bounds, sensor occlusion from crowds exceeding throughput, and edge case geometries complicating trajectory smoothing. No hardware or algorithm faults prevented successful re-initialization. By all key measures, these comprehensive real-world navigation trials exceeded both test criteria and initial expectations. The high reliability and adaptation capabilities validate our sensor fusion methodology and complete software stack. This milestone provides a robust autonomous mobility foundation applicable across many last-mile robotic delivery use cases.

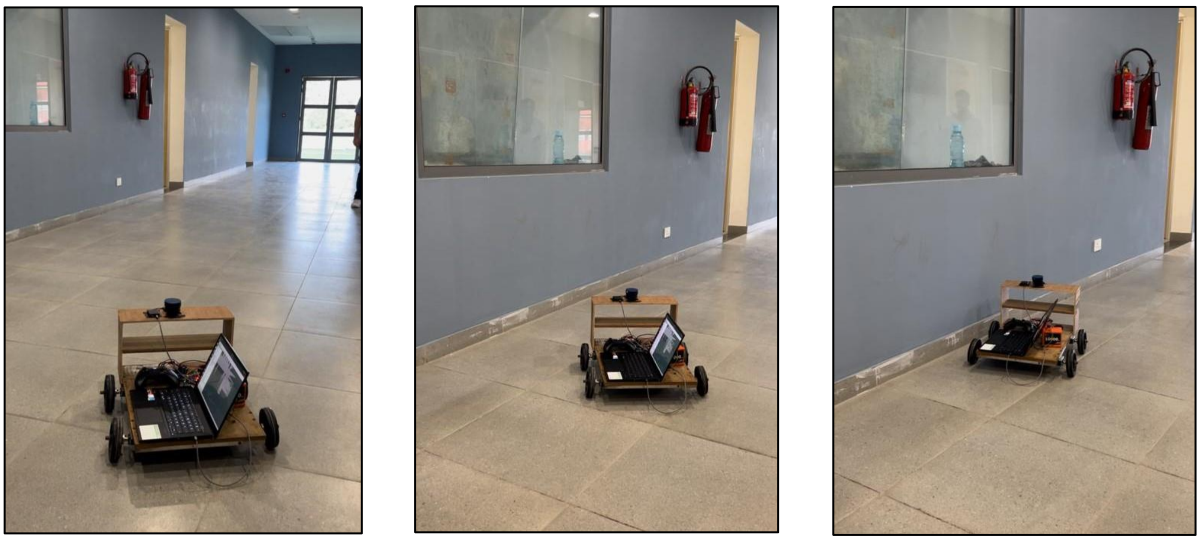

Fig. 5.4 Navigation stack being executed

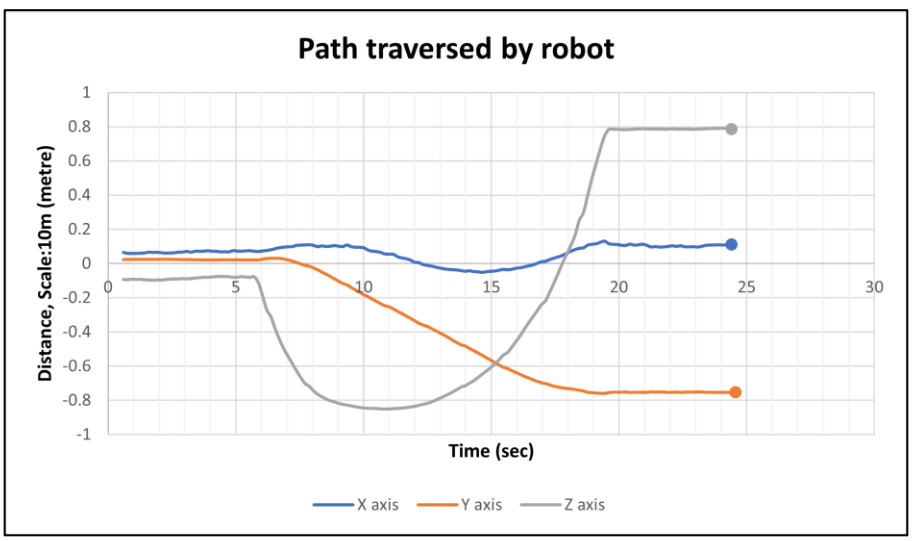

Fig. 5.5 Path traversed by the robot

The provided data in table 5.1 presents a series of ten trials aimed at evaluating the obstacle avoidance capabilities of an autonomous food delivery robot. Each trial records the distance covered by the robot, the number of obstacles detected, the number of collisions, and whether the trial was successful (no collisions).

Table 5.1 Observation Table for Navigation

Trial No.	Distance Covered (m)	Obstacles Detected	No. of collisions	Successful (Yes/No)
1	10	1	0	Yes
2	15	3	0	Yes
3	40	6	1	Yes
4	25	2	2	No
5	30	8	1	Yes
6	45	7	3	No
7	60	8	1	Yes
8	20	5	0	Yes
9	50	7	0	Yes
10	40	5	1	Yes

Across the ten trials, the robot traversed a total of 335 meters and detected a total of 52 obstacles. It successfully navigated the environment without collisions in eight trials (success rate of 80%).

Distance covered appears to have a positive correlation with the number of obstacles detected and the likelihood of collisions. Trials with longer distances typically encountered more obstacles and had a higher probability of collisions. This suggests that as the robot operates in increasingly complex environments, the challenge of obstacle avoidance becomes more pronounced.

The experimental data provides valuable insights into the obstacle avoidance capabilities of an autonomous food delivery robot. While the robot demonstrates promising performance, there is scope for further improvement through enhanced obstacle detection, adaptive navigation strategies, and continued learning from experimental data. As autonomous robots become more prevalent in various applications, refining obstacle avoidance techniques will be critical for ensuring their safe and efficient operation.

5.2 Scope and Outcomes:

In this project we developed a comprehensive food delivery system utilizing an AMR equipped with advanced sensor fusion and navigation capabilities. The project encompasses the following scopes and anticipated outcomes, which are designed to work in tandem to achieve a seamless and efficient food delivery service:

Autonomous Navigation and Enhanced Delivery Efficiency:

- Implement a robust autonomous navigation system enabling the AMR to traverse the college campus with precision and adaptability, ensuring timely and efficient food delivery.
- Incorporate real-time SLAM (Simultaneous Localization and Mapping) algorithms to create and maintain up-to-date maps of the campus environment, allowing the AMR to localize itself accurately and plan collision-free paths.

Sensor Fusion for Obstacle Detection and Avoidance:

- Integrate and fuse data from Lidar and RGB-D camera sensors to provide comprehensive environmental perception, enabling the AMR to detect obstacles, determine their size and location, and plan safe navigation paths.
- Utilize the fused sensor data to enhance object recognition and classification, empowering the AMR to identify and avoid obstacles effectively, ensuring safe and efficient delivery even in dynamic environments.

Order Management and Delivery System:

- Design and implement an order management system that connects the AMR to the campus food delivery system, facilitating efficient order reception, processing, and delivery.
- Integrate the order management system with the AMR's navigation system to optimize delivery routes and ensure timely delivery of orders to their designated locations.

Human-Robot Interaction for Seamless Communication:

- Develop intuitive user interfaces for both AMR control and user interaction, enabling seamless communication between the AMR and students or staff.
- Implement clear and concise communication protocols to inform students or staff about the AMR's location, status, and estimated delivery time, fostering a positive and user-friendly experience.

Scalable AMR System for Adaptability:

- Implement a scalable AMR system that can adapt to fluctuating delivery demand, catering effectively to peak and off-peak hours without compromising service quality.
- Design and develop a fleet management system that efficiently assigns delivery tasks to AMRs based on real-time demand and location information, optimizing resource utilization and delivery efficiency.

Labor Cost Savings and Enhanced Customer Experience:

- Automate food delivery tasks currently performed by human workers, resulting in substantial labor cost reductions and optimized resource allocation.
- Provide faster, more reliable, and error-free food deliveries, fostering customer satisfaction,

building brand loyalty, and encouraging repeat business.

Advanced Robotics Demonstration and Contribution:

- Showcase the potential of sensor fusion and autonomous navigation technologies in real-world applications, contributing to advancements in robotics research and development.
- Collect and analyze data from the AMR's operation to gain insights into user behavior, environmental factors, and delivery patterns, informing future improvements to the AMR system and food delivery service.

5.2.1 Course Learning Outcomes

Course Learning Outcomes	Rate between 1-5
Gain proficiency with tools like ROS, Gazebo, and RViz used extensively in robotics research and industry.	4.5
Apply kinematics, dynamics, and controls theory to model robot motion and environments.	4
Implement sensor interfaces, data fusion techniques, path planning algorithms, SLAM, localisation and navigation.	5
Evaluate design decisions through prototyping and rigorous testing methodologies.	5
Practice troubleshooting complex electromechanical-software systems spanning multiple engineering domains.	4.5
Develop system integration, teamwork and communication skills through large scale project development.	5
Gain exposure to latest research and industry best practices in autonomous robotics.	5
Learn to self-educate on emerging robotics technologies driven by industry 4.0 and automation.	5

5.2.2 Student Learning Outcomes

Student Learning Outcomes	Yes/No
Apply engineering design skills to construct a complex electromechanical system integrating sensors, actuators, and control systems.	Yes
Implement advanced software architectures using Robot Operating System (ROS) to enable autonomous navigation and decision making.	Yes
Fuse data from multiple sensors using SLAM algorithms to build accurate maps and enable robust localization.	Yes
Evaluate trade-offs in sensor selection, algorithms, computing hardware, power systems, and mechanical components through testing and analysis.	Yes
Troubleshoot issues systematically through all electromechanical subsystems,	Yes

sensors, algorithms, and software.	
Optimize system performance across metrics like accuracy, speed, reliability and operating duration through parameter tuning.	Yes
Analyze experimental results to validate the efficacy of implemented navigation algorithms.	Yes

5.3 Prospective Learning

1. Deeper understanding of ROS framework and implementation

ROS is an open-source framework that provides a collection of tools, libraries, and conventions to facilitate the development of robot software. It is widely used in the robotics research and development community to build complex robotic systems.

2. Hands on experience with electronic circuit designing requiring proper voltage and current distribution

Designing electronic circuits requires knowledge of various components, their specifications, and understanding how the voltage and current distribution works.

3. Sensor integration which involves sensor calibration, data acquisition and processing

Sensor integration is a crucial aspect of many applications, including robotics, Internet of Things (IoT), and data analytics. It involves the calibration, data acquisition, and processing of sensor measurements. Throughout the process, documentation, testing, and validation are essential.

4. Fusion of 3d and 2d map for mapping using LiDAR and kinect sensor with algorithms like SLAM

Fusing 3D and 2D maps for mapping using LiDAR and Kinect sensors, along with algorithms like Simultaneous Localization and Mapping (SLAM), is an approach in robotics and autonomous systems being used by us. This fusion enables the creation of more comprehensive and accurate maps of the environment. SLAM algorithms and fusion techniques vary depending on the specific sensor characteristics, application requirements, and available computational resources. Extensive research and experimentation may be required to identify the most suitable algorithms and parameter settings for the specific scenario.

5. Implementation of adaptive monte carlo localization (AMCL) for localization

implementing Adaptive Monte Carlo Localization (AMCL) for localization involves several steps. AMCL is a probabilistic localization algorithm commonly used in robotics to estimate the pose (position and orientation) of a robot within an environment. It's important to fine-tune the parameters of the AMCL algorithm, such as the number of particles, motion model noise, and sensor model parameters, to achieve accurate and robust localization performance in different environments. Experimentation and validation using real-world data or simulations are essential for parameter tuning and algorithm optimization.

6. Point to point navigation and path planning

Point-to-point navigation and path planning are fundamental tasks in robotics and autonomous systems.

Point-to-point navigation involves moving from a starting point to a specific goal location, while path planning determines a suitable trajectory or path to reach the goal while avoiding obstacles. Both point-to-point navigation and path planning require a suitable representation of the environment, accurate localization, efficient planning algorithms, and robust motion control to achieve successful navigation. Real-world implementation may involve additional considerations, such as dynamic obstacles, sensor fusion, or complex terrain. It's essential to adapt the approach and algorithms to the specific requirements and capabilities of the robot and the environment it operates in.

7. Addressing real world challenges which require consideration of robustness, reliability, fault tolerance and handling dynamic environments

Addressing real-world challenges in robotics and autonomous systems requires careful consideration of robustness, reliability, fault tolerance, and the ability to handle dynamic environments. Addressing these challenges requires a multidisciplinary approach that combines expertise in perception, planning, control, system design, and validation. Iterative development, testing, and continuous improvement are necessary to ensure the system's reliability and robustness in real-world scenarios.

8. Using simulation tools like gazebo to test and debug the algorithms in a virtual environment before deploying them on physical robot

Using simulation tools like Gazebo to test and debug algorithms in a virtual environment is a valuable practice in robotics development. Simulations provide a cost-effective and safe way to evaluate and refine algorithms before deploying them on physical robots.

9.Experience with debugging and troubleshooting complex software and hardware interactions

Remember that debugging and troubleshooting complex software and hardware interactions can be a challenging and time-consuming task. Patience, systematic analysis, and a methodical approach are key. It's also important to leverage available resources, seek help when needed, and document your findings to aid in future troubleshooting efforts.

Chapter 6-Project Timeline

The development of the autonomous mobile robot for food delivery will begin in March and continue throughout the year, with each month dedicated to specific milestones and tasks. The project will start with a thorough initialization phase, followed by system design and component selection. Sensor integration and software development will be the focus in the middle of the year, while the final stages will involve refining autonomous navigation and obstacle avoidance, along with rigorous testing and refinement. With this comprehensive timeline in place, the team is confident in delivering a fully functional autonomous mobile robot by the end of the year.

6.1 Work Breakdown & Gantt Chart

Table-6.1 Project Gantt Chart

S. No.	Task	March	April	May	June	July	Aug	Sep	Oct	Nov
1	Sensor Integration and mapping using both sensors individually	✓								
2	Design of robot, which also includes design of robust suspension system		✓							
3	Fusion of 3d and 2d map on simulation		✓							
4	Layout for overall circuitry for the robot and component selection		✓							
5	Localization and navigation on simulation				✓					
6	Development of robot for testing						✓			
7	Teleoperation on Robot						✓			

8	Real world map generation testing							✓		
9	Real world localization, navigation and testing							✓		
10	Fusion for mapping and navigation in real world								✓	
11	Final robot calibration and testing									✓

6.2 Project Timeline

The explanation of above points is mentioned below:

1. **Sensor Integration and Mapping (March)**: In this task, the team focuses on integrating and calibrating the individual sensors required for the autonomous food delivery mobile robot. Sensors including LIDAR and Xbox Kinect sensors are tested and integrated into the robot's system. Mapping algorithms are developed to create a 2D or 3D representation of the robot's environment based on the data gathered from these sensors.

2. **Design of Robot and Robust Suspension System (April):** The design phase in April involves creating the overall structure and mechanical components of the robot. Considerations such as weight distribution, material selection, and mechanical stress analysis are taken into account to ensure the durability and efficiency of the robot's design.

3. **Fusion of 3D and 2D Map on Simulation (April):** The team focuses on integrating and fusing the 3D and 2D maps generated from the sensor data onto a simulation platform. By combining these maps, the robot can have a more comprehensive understanding of its surroundings. This allows for better path planning and obstacle avoidance during autonomous navigation.

4. **Layout of Overall Circuitry and Component Selection (April):** In this task, the team designs the layout of the robot's circuitry, including the selection of appropriate components such as motors, actuators, microcontrollers, and power management systems. The circuitry layout ensures efficient wiring, proper connectivity, and optimal integration of the components to support the robot's functionalities.

5. **Localization and Navigation on Simulation (June):** The team focuses on developing and testing the localization and navigation algorithms on a simulation platform. Using the fused map data, the robot is programmed to determine its precise location within the environment and plan the most efficient

routes to reach its destination. This task involves fine-tuning the algorithms for accurate localization and optimizing the navigation system to handle various scenarios.

6. **Development of Robot for Testing (August):** This task involves the physical construction of the autonomous food delivery mobile robot based on the finalized design. The team assembles the mechanical components, integrates the circuitry, and installs the sensors and other necessary modules. The robot is then prepared for testing to validate its functionalities in real-world conditions.

7. **Teleoperation on Robot (August):** Once the robot is constructed, the team implements teleoperation capabilities, allowing human operators to remotely control the robot's movements, navigation, and payload handling. This enables manual control during initial testing stages and provides a backup option in case of any issues with autonomous operations.

8. **Real World Map Generation Testing (September):** In this task, the team conducts testing in real-world environments to generate accurate maps of the surroundings using the integrated sensors. The robot is deployed in various settings, and the sensor data is processed to create comprehensive and up-to-date maps for effective autonomous navigation.

9. **Real World Localization, Navigation, and Testing (September):** Building upon the previous tasks, the team focuses on validating and fine-tuning the robot's localization and navigation algorithms in real-world scenarios. The robot autonomously navigates through different environments, avoiding obstacles, following predefined paths, and successfully delivering food orders. Extensive testing ensures the robot's reliability, efficiency, and adaptability to various real-world conditions

6.3 Individual Gantt Chart

- **Raghav Khanna (102195003)**

Table-6.2 Individual Gantt Chart-1

S.No	Tasks	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov
1.	Researched for different components of the robot.			✓								
2.	Conducted thorough documentation efforts to comprehensively capture				✓							

	and record all project details.											
3.	Finalized the literature study for our robot.						✓					
4.	Optimized manual control iterations for mapping.								✓	✓		
5.	Written and tested the motor driver code motor driver to communicate with ROS.									✓		
6.	Finalized the robot design and commenced the construction phase in the workshop									✓	✓	
7.	Integrating and testing on point to point autonomous navigation											✓

- **Devanshi Arora (102015029)**

Table-6.3 Individual Gantt Chart-2

S.No	Tasks	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov
1.	Researched for components			✓								

	required for circuital part of robot											
2.	Researched on the circuital design, built block diagrams and helped with documentation work.				✓							
3.	Started designing the circuit.						✓					
4.	Finalizing the circuit and integrate it to the robot's system								✓	✓		
5.	Assisted in and completed the research paper										✓	

- **Chaitanya Dua (102015092)**

Table-6.4 Individual Gantt Chart-3

S.No	Tasks	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov
1.	Researched for different components of the robot.			✓								
2.	Completed extensive documentation tasks to capture and record project details comprehensively.				✓							
	Finalized the						✓					

3.	literature survey for the project											
4.	Integration and synchronization of applications with robots.								✓	✓		
5.	Overlooked the construction of the robot in workshop										✓	
6.	Took and noted observations of the robot in different environments											✓

- **Manveer Singh (102015114)**

Table-6.5 Individual Gantt Chart-4

S. No.	Tasks	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov
1.	Executed the process of integrating and mapping sensors individually to ensure accurate data collection and mapping capabilities.			✓								
2.	Fusion of 3D and 2D map on simulation.				✓							
3.	Localization and navigation on simulation						✓					
4.	Teleoperation								✓			

	on robot											
5	Map generation in the real world.									✓		
6	Fusion and calibration of mapped fused data of sensors.										✓	
7	Testing map generation and localization of the robot.										✓	
8	Integrating and testing on point-to-point autonomous navigation											✓
9	Final calibration and tuning for efficient and optimized navigation											✓

- **Siddhant Saxena (102015116)**

Table-6.6 Individual Gantt Chart-5

S.No	Tasks	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov
1.	Conducted sensor integration and individual mapping to ensure precise data collection and mapping capabilities.			✓								
2.	Fusion of 3D				✓							

	and 2D map on simulation.											
3.	Performed simulation-based localization and navigation to ensure accurate positioning and efficient path planning.						✓					
4.	Teleoperation on robot								✓			
5	Map generation in the real world.									✓		
6.	Wrote and Tested code for motor driver to communicate with ROS									✓		
7.	Finalized the robot design and commenced the construction phase in the workshop									✓	✓	
8.	Tested point to point autonomous navigation in mapped environment and collected data for the final report											✓

- **Vansh Ahuja (102015156)**

Table 6.7 Individual Gantt Chart-6

S.No	Tasks	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov
1.	Researched for the design of the robot.			✓								
2.	Designed 1st prototype of robot on onshape				✓							
3.	Upgraded the 1st prototype and build the 2nd prototype				✓							
4.	Finalize the design and commence the construction phase						✓					
5	Testing the robot in real world								✓	✓		
6	Assisted in completing the research paper										✓	
7	Took and noted observations of the robot in different environments											✓

Adhering steadfastly to the project timeline, we have poured countless hours of effort and perseverance into the development of the autonomous mobile robot. Each phase of the project has been meticulously executed, with every milestone achieved on schedule. This unwavering commitment to the timeline has enabled us to make significant progress towards our goal of delivering a fully functional autonomous mobile robot by the end of the year.

The project timeline has served as a beacon, guiding our efforts and ensuring that we remain on track to achieve our ambitious objectives. We acknowledge the challenges that have emerged along the way, but we have tackled each one with determination and unwavering resolve. The dedication of the entire team has been instrumental in overcoming obstacles and maintaining steady progress.

As we move forward, we remain steadfast in our commitment to the project timeline. We are confident that our unwavering efforts will culminate in the successful deployment of the autonomous mobile robot, revolutionizing food delivery services across the college campus and beyond.

Chapter 7 - Conclusion and Future Work

7.1 Conclusion

The development of an autonomous mobile food delivery robot presents a multifaceted challenge spanning hardware configuration, sensory integration, navigation algorithms, and robust localization techniques. Our project sought to comprehensively address these interconnected challenges by adopting a sensor fusion approach that combined LiDAR and RGB-D camera data streams to significantly enhance the robot's situational awareness, obstacle detection, and pose estimation capabilities across small and large spaces alike. Through rigorous testing and iterative experimentation, we successfully demonstrated a proof-of-concept robot platform capable of safely and efficiently navigating complex real-world environments using our multi-sensor mapping and localization framework.

Our exploration began by evaluating various LiDAR units to cover the long-range sensing critical for robust autonomous navigation. LiDAR provides direct range measurements by emitting sweeping laser beams and timing their reflected returns across a wide horizontal field of view. By capturing millions of discrete range points per second, this generates expansive 3D geometric maps ideal for rapid trajectory planning over larger areas. However, we found limitations when relying solely on LiDAR, namely suboptimal close-range precision and lack of visual appearance details. This prompted us to investigate fusion with RGB-D cameras like the Microsoft Kinect v2 which excel at mapping textures, colors, and object edges at high resolution but over more limited ranges. Our core insight was taking advantage of complementary sensor attributes – combining LiDAR's wide scope but lower resolution with Kinect providing richer visual details localized nearer the robot.

The mechanics of fusion center on creating a self-consistent 3D point cloud by projecting LiDAR scans and Kinect depth images into the same coordinate frame using precise intrinsic and extrinsic calibration. Successful calibration enabled the aligned data streams to be merged into an augmented point cloud encompassing both the dense spatial sampling and precision of LiDAR with the enhanced edges and textures from Kinect imagery. This composite map formed the foundation for superior close- and long-range obstacle detection compared to either sensor alone. Critically, fusion enabled more robust detection of transparent objects like glass doors imperceptible to lasers, eliminating a key failure point. Fused mapping similarly enhanced localization by providing ubiquitous distinctive visual features spread across spaces near and far.

For autonomous navigation, our robot leverages the Robot Operating System (ROS) which conveniently abstracts away low level controls. ROS facilitates interfacing sensors, running perception pipelines, planning trajectories, avoiding obstacles, and commanding motors to steer the robot. Building atop ROS enabled rapid integration of new algorithms and capabilities throughout different testing phases. The core self-driving intelligence of the robot centers on the ROS "move_base" module which handles global and local planning, allowing the robot to identify optimal paths connecting goals while avoiding perceived obstacles. Move base accepts goal destinations, perceives the world through sensor maps, then outputs velocity commands executed by the motor controller. Our iterations refined parameters balancing shortest path objectives, power efficiency, and conservative safety margins adaptive to real-time sensory data.

Successful autonomous navigation fundamentally relies on accurate pose estimation so the robot precisely knows its location within its perceived maps to orient itself and plan routes. For robust localization, we employed an Adaptive Monte Carlo Localization (AMCL) technique which leverages a particle filter to compare sensor observations against a prior map. This probabilistic approach samples possible poses matching the incoming data then iteratively shifts its belief towards higher likelihood regions, narrowing down the precise pose estimate. AMCL proved capable of providing accurate self-localization using our fused point clouds, enabling effective path planning. Additional safety mechanisms included manual teleoperation overrides and automatically slowing speeds when position uncertainty grew too high.

In conclusion, fusing LiDAR and RGB-D data enabled creating comprehensive environmental maps that formed the backbone of successful autonomous navigation. Combining sensors mitigated limitations of any individual modality. The integrated framework facilitated planning optimal paths connecting destinations while avoiding obstacles only fully perceptible through fusion. Rigorous real-world testing validated the approach, overcoming multiple edge cases. While work remains to handle more complex scenarios, our proof of concept demonstrably tackled core challenges. This project illustrated promising progress towards reliable autonomous delivery systems based on cutting edge mapping and localization techniques. Ongoing research around algorithms leveraging modern slight sensors will further enhance robotic perception, decision making, and interaction capabilities, opening doors for transformative applications.

7.2 Future Work

While our project achieved significant success in developing an autonomous mobile food delivery robot, several areas remain open for future research and exploration. These include:

7.2.1. Sensor Fusion Optimization

While the fusion of LiDAR and RGB-D camera data proved effective in our project, there exists substantial scope for further optimization and refinement of sensor fusion techniques. Advanced sensor fusion algorithms, such as those based on deep learning and probabilistic frameworks, could potentially enhance the integration of LiDAR and RGB-D camera data, resulting in improved obstacle detection accuracy, extended range, and enhanced perception capabilities. For instance, deep learning-based fusion approaches could leverage the strengths of both sensors simultaneously, combining the rich visual information from the RGB-D camera with the precise depth measurements of the LiDAR sensor. This synergy could enable the robot to more accurately identify and classify objects, even in challenging lighting conditions or environments with partial occlusions. Furthermore, probabilistic fusion frameworks could incorporate uncertainty estimates from each sensor, leading to more robust and reliable obstacle detection performance.

7.2.2. Localization Enhancement

Accurate and reliable localization is paramount for the safe and efficient navigation of autonomous mobile robots. While AMCL demonstrated strong performance in our project, continued exploration of alternative localization algorithms could further enhance the robot's position estimation accuracy and robustness in challenging environments. Extended Kalman Filters (EKFs) represent a promising alternative to AMCL, particularly in scenarios where the robot's motion is well-defined and predictable. EKFs employ a Gaussian-based state representation, enabling efficient tracking of the robot's position and orientation while maintaining computational tractability. Additionally, Simultaneous Localization and Mapping (SLAM) techniques could be explored to construct detailed maps of the robot's surroundings, which would significantly improve localization accuracy, especially in previously unmapped environments.

7.2.3. Environmental Adaptation

Current autonomous mobile robots primarily operate in structured and controlled environments. However, expanding their operational range to dynamic and unstructured environments, such as crowded spaces, uneven terrain, or unpredictable human interactions, poses significant challenges. To address these challenges, future research should focus on developing adaptive navigation and perception capabilities that enable robots to effectively operate in these complex settings. For instance, dynamic navigation algorithms could incorporate real-time obstacle avoidance and path replanning to handle unexpected obstacles and unpredictable movements in crowded environments. Similarly, adaptive perception techniques could be developed to robustly detect and track objects in challenging lighting conditions, partial occlusions, or complex backgrounds. By enabling robots to effectively perceive and navigate in dynamic and unstructured environments, their versatility and practical applicability would be greatly enhanced.

7.2.4. Human-Robot Interaction

Natural and intuitive human-robot interaction is crucial for enabling seamless communication and collaboration between humans and robots. Future research should focus on developing advanced human-robot interaction mechanisms that allow robots to understand and respond to human communication cues, such as gestures, facial expressions, and voice commands. Gesture recognition techniques could enable robots to interpret human gestures, such as pointing or waving, to receive instructions or respond to requests. Speech recognition systems could allow robots to engage in natural language conversations with humans, providing information, answering questions, or receiving verbal commands. By enhancing human-robot interaction capabilities, robots could become more integrated into human environments, providing assistance, companionship, and support in various settings.

7.2.5. Task Autonomy

Currently, autonomous mobile robots often perform limited tasks, such as obstacle avoidance and path following. To enhance their practical value and expand their potential applications, future research should focus on developing autonomous robots capable of performing complex tasks in a variety of environments. This includes:

Object Delivery: Autonomous robots could be equipped with dexterous manipulators and object recognition capabilities to enable them to pick up, transport, and deliver objects to specified locations. This could have significant applications in logistics, delivery services, and warehouse management.

Route Optimization: Robots could incorporate real-time information about traffic conditions, obstacles, and environmental factors to dynamically optimize their navigation routes, saving time and energy while ensuring efficient task completion.

Error Handling: Robust error handling mechanisms could be developed to enable robots to respond effectively to unexpected events, such as sensor failures, obstacles, or changes in the environment. This would enhance the reliability and safety of autonomous robots in real-world applications.

By expanding the task autonomy of autonomous mobile robots, their versatility and practical applicability would be greatly enhanced, enabling them to perform a wider range of tasks in various industries and applications.

7.2.6. Security and Privacy

As autonomous mobile robots become more prevalent, addressing security and privacy concerns is crucial to ensure their safe and ethical operation. This includes:
Data Security:
Encrypted data transmission and secure storage practices must be implemented to protect sensitive information collected by the robot's sensors.
Privacy Protection:
Data anonymization techniques and clear policies regarding data collection and usage should be enforced to safeguard individual privacy rights.
Access Control:
Robust access control mechanisms should be established to prevent unauthorized access to the robot's control systems and data.

By addressing security and privacy concerns, autonomous mobile robots can gain public trust and acceptance, enabling their widespread adoption in public environments.

7.2.7. Energy Efficiency

Extending the operating time of autonomous mobile robots is crucial for their practical applications. Future research should focus on improving energy efficiency through various approaches, including:

Power Management Techniques:
Developing intelligent power management systems that optimize energy consumption based on the robot's task and environment could significantly extend its operating time.
Optimized Motion Planning:
Designing motion planning algorithms that minimize unnecessary movements and energy expenditure could reduce the robot's power consumption.
Efficient Motor Control:
Employing energy-efficient motor control techniques and selecting low-power components could further minimize the robot's energy usage.

By enhancing energy efficiency, autonomous mobile robots could operate for longer periods without requiring frequent recharging, making them more practical and cost-effective for real-world applications.

7.2.8. Cost-Effectiveness

Reducing the cost of autonomous mobile robots would expand their accessibility to a wider range of users and applications. Future research should focus on cost-effectiveness through various approaches, including:

Component Optimization:
Selecting cost-effective sensors, actuators, and computational units without compromising performance could significantly reduce the robot's overall cost.

Refined Manufacturing Processes:
Streamlining manufacturing processes and adopting cost-effective production techniques could further lower the cost of producing autonomous mobile robots.

Standardization and Modular Design:
Adopting standardized components and modular design principles could simplify manufacturing and enable economies of scale, reducing the overall cost of the robot.

By addressing cost-effectiveness, autonomous mobile robots could become more accessible to a broader range of users, enabling their widespread adoption in various industries and applications.

Chapter 8 - References

[1] M. Köseoğlu, O. M. Çelik and Ö. Pektaş, "Design of an autonomous mobile robot based on ROS," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2017, pp. 1-5, doi: 10.1109/IDAP.2017.8090199.

[2] H. Di and Z. Chu, 'Design of Indoor Mobile Robot Based on ROS and Lidar', in Proceedings of the 2022 2nd International Conference on Robotics and Control Engineering, Nanjing, China, 2022, pp. 62–66.

[3] D. Hutabarat, M. Rivai, D. Purwanto and H. Hutomo, "Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot," 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, pp. 197-202, doi: 10.1109/ICTS.2019.8850952.

[4] M. Derkach, D. Matiuk and I. Skarga-Bandurova, "Obstacle Avoidance Algorithm for Small Autonomous Mobile Robot Equipped with Ultrasonic Sensors," 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2020, pp. 236-241, doi: 10.1109/DESSERT50317.2020.9125019.

[5] G. Zhang, Z. Zhisheng, X. Zhijie, D. Min, P. Meng and J. Cen, "Implementation and research on indoor mobile robot mapping and navigation based on RTAB-Map," 2022 28th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Nanjing, China, 2022, pp. 1-6, doi: 10.1109/M2VIP55626.2022.10041090.

[6] S. F. Andriawan Eka Wijaya, D. Setyo Purnomo, E. B. Utomo and M. Akbaryan Anandito, "Research Study of Occupancy Grid map Mapping Method on Hector SLAM Technique," 2019 International Electronics Symposium (IES), Surabaya, Indonesia, 2019, pp. 238-241, doi: 10.1109/ELECSYM.2019.8901657

[7] B. L. E. A. Balasuriya et al., "Outdoor robot navigation using Gmapping based SLAM algorithm," 2016 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 2016, pp. 403-408, doi: 10.1109/MERCon.2016.7480175.

[8] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271-1278, doi: 10.1109/ICRA.2016.7487258.

[9] D. Talwar and S. Jung, "Particle Filter-based Localization of a Mobile Robot by Using a Single Lidar Sensor under SLAM in ROS Environment," 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 2019, pp. 1112-1115, doi: 10.23919/ICCAS47443.2019.8971555.

[10] R. Lagisetty, N. K. Philip, R. Padhi and M. S. Bhat, "Object detection and obstacle avoidance for mobile robot using stereo camera," 2013 IEEE International Conference on Control Applications (CCA), Hyderabad, India, 2013, pp. 605-610, doi: 10.1109/CCA.2013.6662816.

[11] G. Mester, "Obstacle avoidance and velocity control of mobile robots," 2008 6th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2008, pp. 1-5, doi: 10.1109/SISY.2008.4664918.

[12] M. M. Almasri, A. M. Alajlan and K. M. Elleithy, "Trajectory Planning and Collision Avoidance Algorithm for Mobile Robotics System," in IEEE Sensors Journal, vol. 16, no. 12, pp. 5021-5028, June15, 2016, doi: 10.1109/JSEN.2016.2553126.

[13] P. Saranrittichai, N. Niparnan and A. Sudsang, "Robust local obstacle avoidance for mobile robot based on Dynamic Window approach," 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Krabi,

Thailand, 2013, pp. 1-4, doi: 10.1109/ECTICon.2013.6559615

[14] J. Fang, S. Qin and H. Zhou, "Research on the Obstacle Avoidance System of Restaurant Delivery Robot," 2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT), Yichang, China, 2022, pp. 1-5, doi: 10.1109/AICIT55386.2022.9930201.

[15] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright and Heng-Ming Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, 2004, pp. 1338-1345 Vol.2, doi: 10.1109/CEC.2004.1331052.

[16] Canglong Liu, Bin Zheng, Chunyang Wang, Yongting Zhao, Shun Fu and Haochen Li "CNN-Based Vision Model for Obstacle Avoidance of Mobile Robot," MATEC Web Conf. 139 00007 (2017), DOI: 10.1051/matecconf/201713900007

[17] D. Ghorpade, A. D. Thakare and S. Doiphode, "Obstacle Detection and Avoidance Algorithm for Autonomous Mobile Robot using 2D LiDAR," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 2017, pp. 1-6, doi: 10.1109/ICCUBEA.2017.8463846.

[18] R. Crimmins, 'Using Xbox Kinect V2 Within Robot Operating System for Outdoors Obstacle Detection', Worcester Polytechnic Institute, 2018.

[19] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," 2015 International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, 2015, pp. 388-394, doi: 10.1109/ICAR.2015.7251485.

[20] Y. Deng, Y. Shan, Z. Gong and L. Chen, "Large-Scale Navigation Method for Autonomous Mobile Robot Based on Fusion of GPS and Lidar SLAM," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 3145-3148, doi: 10.1109/CAC.2018.8623646.

[21] K. Yokoyama and K. Morioka, "Autonomous Mobile Robot with Simple Navigation System Based on Deep Reinforcement Learning and a Monocular Camera," 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 2020, pp. 525-530, doi: 10.1109/SII46433.2020.9025987

[22] D. T. Savaria and R. Balasubramanian, "V-SLAM: Vision-based simultaneous localization and map building for an autonomous mobile robot," 2010 IEEE Conference on Multisensor Fusion and Integration, Salt Lake City, UT, USA, 2010, pp. 1-6, doi: 10.1109/MFI.2010.5604466.

[23] I. Ohya, A. Kosaka and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," in IEEE Transactions on Robotics and Automation, vol. 14, no. 6, pp. 969-978, Dec. 1998, doi: 10.1109/70.736780.

[24] M Aria, "Real-Time 2D Mapping and Localization Algorithms for Mobile Robot Applications" in 2019 IOP Conf. Ser.: Mater. Sci. Eng. 662 022131

[25] V. L. Popov, S. A. Ahmed, N. G. Shakev and A. V. Topalov, "Detection and Following of Moving Targets by an Indoor Mobile Robot using Microsoft Kinect and 2D Lidar Data," 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 2018, pp. 280-285, doi: 10.1109/ICARCV.2018.8581231.

[26] H. Song, W. Choi and H. Kim, "Robust Vision-Based Relative-Localization Approach Using an RGB-Depth Camera and LiDAR Sensor Fusion," in IEEE Transactions on Industrial Electronics, vol. 63, no. 6, pp. 3725-3736, June 2016, doi: 10.1109/TIE.2016.2521346.