A

Project Report

on

Design and Development of Smart Warehouse Robot Prototype

Submitted to

Sant Gadge Baba Amravati University, Amravati

Submitted in partial fulfilment of

the requirements for the Degree of

Bachelor of Engineering in

Electronics and Telecommunication Engineering

Submitted by

Mr. Bhushan Nemade
(PRN: 193120226)

Ms. Namrata Kherde
(PRN: 193120334)

Mr. Abhay Tiwari
(PRN: 193120108)

Ms. Anushka Unhale
(PRN: 193120264)

Under the Guidance of

Dr. K. B. Khanchandani

Professor, E & TC Dept.

Department of Electronics & Telecommunication Engg.

Shri Sant Gajanan Maharaj College of Engineering,

Shegaon – 444 203 (M.S.)

2022-2023

Department of Electronics & Telecommunication Engineering
Shri Sant Gajanan Maharaj College of Engineering,
Shegaon – 444203, Maharashtra, India
(Recognized by AICTE, Accredited by N.B.A, New Delhi)

Certificate

This is to certify that the project report entitled **"Design and Development of Smart Warehouse Robot Prototype"** is hereby approved as a creditable study carried out and presented by

Mr. Bhushan Nemade	(PRN: 193120226)
Ms. Namrata Kherde	(PRN: 193120334)
Mr. Abhay Tiwari	(PRN: 193120108)
Ms. Anushka Unhale	(PRN: 193120264)

in a manner satisfactory to warrant its acceptance as a pre-requisite in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Electronics & Telecommunication Engineering of Sant Gadge Baba Amravati University, Amravati during the Session 2022-23.

Dr. K. B. Khanchandani
Project Guide

Dr. V. V. Ratnaparkhi
Internal Examinar

31-5-23 (Dr. S. V. Pattalwar)
External Examiner

Dr. M. N. Tibdewal
Professor & Head, E & TC Dept.

Dr. S. B. Somani
Principal

Abstract

The advancements in technology and science have led to the rapid growth of the robotics industry. Robotics and automation are being increasingly utilized in various fields to simplify and enhance processes. In situations where human intervention can be risky or dangerous, such as in the aftermath of human and natural disasters, robots can be used instead of humans to avoid further casualties.

Moreover, robotics can also be employed in warehouse management. Traditionally, humans have been responsible for picking and choosing items in warehouses, but this can be a time-consuming and error-prone process. In addition to improving efficiency and accuracy, the use of robotics in warehouse management can also help reduce labor costs and minimize human errors. With the increasing demand for faster and more efficient order fulfillment, robotics can help meet these demands while also reducing operational costs. With the help of a robotic arm or a simple robot system, the movement and picking up of items can be made faster and more efficient. Furthermore, the use of robotics in warehouse management can also lead to better utilization of available space in warehouses. Robots can navigate through narrow spaces and reach high shelves, which may be difficult or unsafe for humans to access. This can enable better use of vertical space in warehouse and increase storage capacity.

However, to achieve successful and accurate results, the robotic system must be correctly instructed to perform the necessary actions in the correct order.

Acknowledgement

We would like to take this opportunity to express our heartfelt thanks to our guide

Dr. K.B. Khanchandani for his esteemed guidance and encouragement, especially through difficult times. His suggestions broaden our vision and guided us to succeed in this work. We are also very grateful for his guidance and comments while designing part of our project and learned many things under his leadership. Also, we would like to thank

Dr. M. N. Tibdewal, Head of the Electronics and Telecommunication Department, and all teaching and non-teaching staff of the EXTC Department for their encouragement and suggestions for our project.

We extend our thanks to **Dr. S. B. Somani**, Principal, Shri Sant Gajanan Maharaj, College of Engineering, for his valuable support.

We sincerely thank all our friends, who helped us directly or indirectly complete our project work. We would like to express our appreciation for the wonderful experience while completions of this project work.

Mr. Bhushan Nemade

Ms. Namrata Kherde

Mr. Abhay Tiwari

Ms. Anushka Unhale

Abbreviations

AGV: - Automated Grounded Vehicle

APF: - Artificial Potential Field

GND: - Ground

IMU: - Inertia Measurement Unit

IR: - Infrared

MRS: - Multi Robot System

RISC: - Reduced Instruction Set Computer

RMF: - Robotic Mobile Fulfillment

RRT: - Rapid Random exploring Tree

RFID: - Radio Frequency Identification

RST: - Reset

RPM: - Revolution Per Minute

SLAM: - Simultaneous Localization and Mapping

SRAM: - Static Random Access Memory

VFH: - Vector Field Histogram

VFF: - Virtual Force Field

WMS: - Warehouse Management System

List of Figures

Contents

CHAPTER.1
INTRODUCTION

Chapter 1

Introduction

With the exponential growth in ecommerce transactions and market penetration, large retail businesses are incurring huge revenue losses due to the time wasted by employees moving around the warehouse to pick required items. The world's largest ecommerce businesses such as Amazon and Alibaba have started using thousands of stock management robots and this has increased delivery efficiency considerably. A robot is a versatile manipulator that can be programmed to perform various tasks by using different motions to move tools, materials, parts, or specialized devices. The term "robot" originated from "Robot," which means forced labor. The robot's control process includes sensing, thinking, and acting. Sensing involves detecting speech, vision, acceleration, temperature, position, distance, touch, force, magnetic field, light, sound, and position. Thinking includes task planning, plan classification, learning, data processing, path planning, and motion planning. Acting involves executing output information, such as movement, speech, text, visuals, wheels, legs, arms, tracks, and other physical actions. A software application known as a warehouse management system (WMS) is designed to assist in the management of warehouses or distribution centers. It helps with the planning, organizing, staffing, controlling, and directing of resources to efficiently store and move materials in and out of the warehouse. To improve productivity, robots and warehouse associates can work together using a new process. Industrial robots are used for material handling, welding, and inspection, while mobile robots move on legs, tracks, or wheels.

1.1 Motivation

Ecommerce and quick commerce have seen tremendous growth in the past few years. Ecommerce refers to the buying and selling of products or services over the internet, while quick commerce is a subset of ecommerce that focuses on the quick delivery of goods to customers. Warehouse Management robots play a vital role in the operation of these warehouses. Warehouse Management Robots:

- improve efficiency and productivity in the warehouse. With the use of robots, tasks such as material handling, inventory management, and order fulfillment can be performed faster and more accurately than with human labor alone.
- can work around the clock without the need for breaks or time off, which can help to reduce the number of human associates needed to operate the warehouse. This can help to reduce labor costs and make the warehouse more profitable.
- can also help to improve safety in the warehouse. Robots are designed to work in hazardous environments and can perform tasks that may be dangerous for human associates. By using robots for these tasks, the risk of injury to human associates can be reduced.

So, the motivation for designing a warehouse management robot is driven by the desire to improve efficiency, reduce costs, improve safety, and increase accuracy in the warehouse.

1.2 Objectives

Online shopping has been on a steady rise and consumers' demands are increasing day by day with the expectation of instant delivery as soon as possible at their work place or home. This means having a physical store would not guarantee a success without using online platform. This changed the way goods are needed to be distributed from distribution centers to the customers. To meet the everchanging expectations of the customers the logistics world and warehouse management is constantly evolving and changing. The shift towards creating a new intelligence is now being witnessed over several fields. Therefore, e-commerce giants are making major investments in adopting robotics and automation-based technology. A warehouse management robot can be used to automate and streamline the processes involved in managing a warehouse or distribution center. By utilizing robots for tasks such as material handling, inventory management, and order fulfillment, the objective is to improve efficiency, increase productivity, and reduce labor costs. The warehouse management robot should be capable of navigating the warehouse environment safely and efficiently, interacting with human associates when necessary, and performing tasks accurately and reliably. The ultimate goal of a warehouse management robot is to optimize warehouse operations and improve the overall performance of the facility.

1.3 System Overview

A warehouse management robot is a robotic system designed to automate the process of managing inventory in a warehouse. The robot should be able to move around the warehouse, locate and collect items, and deliver them to their designated location. The following is an overview of the system design for a prototype of a warehouse management robot:

Hardware component of the system utilizes the Arduino, sensors, motors, motor drivers.

These components help robot to work as per the requirement.

Software: The software for the robot should provide the necessary functionality to manage the inventory in the warehouse. This includes:

Localization and mapping: The robot should be able to create a map of the warehouse and localize itself within that map.

Gripper: The robot should be equipped with a gripper that can pick up and manipulate objects of various sizes and shapes.

Localization and mapping: The robot should be able to create a map of the warehouse and localize itself within that map.

Path planning and navigation: The robot should be able to plan an optimal path to navigate through the warehouse, avoiding obstacles and other robots.

Object detection and recognition: The robot should be able to detect and recognize objects in the warehouse using computer vision techniques.

Inventory management: The robot should be able to manage the inventory in the warehouse by locating items, picking them up, and delivering them to their designated location.

Communication with the warehouse management system: The robot should be able to communicate with the warehouse management system to receive instructions and report on its status.

Integration: The robot should be integrated with the warehouse management system to ensure that it is operating in sync with other warehouse processes.

In summary, the design of a warehouse management robot prototype should consider the unique requirements and constraints of the warehouse environment, such as the layout of the warehouse, the types of inventories being managed, and the operational procedures of the warehouse.

1.4 Outline of Project

The outline provides a general framework for a project plan for a warehouse management robot prototype. Depending on the specifics of the project, certain sections may be expanded or modified, and additional sections may be added as needed. The project is created by following below steps:

- Study of literature survey
- Advantages and Disadvantages of the product
- Deciding the methodology, Roadmap of design
- Finalization of the hardware and software
- Design development and performance analysis

At start complete study of papers and products were done thereafter the advantages and advantages of products were considered to find the improvement areas in existing designed. Then we surveyed D-MART and JIO-MART to get an idea about what they actually want from these kinds of products. Afterwards the mechanical design was fixed based on it mathematical and mechanical aspects. The hardware design was made then electronics, and software simulations were done and finally performance analysis was done.

CHAPTER.2
Literature Review

Chapter 2

Literature Review

This section of report provides the summary of published papers and products available in the market that were studied during the whole process.

Warehouse Management System using Microprocessor Based Mobile Robotic Approach:

This paper discusses the increasing use of robots in various fields, including warehouse management. The paper proposes a model to control a warehouse using robots, which can search for and retrieve objects from shelves, improving productivity and reducing travel time. The paper includes a literature survey, a discussion of robotics for warehouse management, an explanation of the proposed approach using a microprocessor-based mobile robotic system, and a conclusion. The literature survey mentions various applications of robotics, including warehouse navigation using cameras, bomb identification and diffusion, precision welding, and the use of RFID. The section on robotics for warehouse management explains that robots can work together with human associates to reduce travel time and increase productivity. The section also discusses the different types of robots used for material handling and domestic tasks, and includes a figure illustrating how mobile robots can be utilized for warehouse management. The block diagram of a warehouse management system using robots includes a microcontroller, sensor array, power supply, lifting motor, and motor drive is also given. The microcontroller is the robot's brain and consists of a processor, input/output peripherals, and memory. The sensor array has nine IR emitter-detector pairs for detecting obstacles and distinguishing colors. A regulated power supply provides power to the driver circuits and microcontroller. DC motors, driven by the motor driver circuit L293D, are used for the lifting mechanism. Infrared sensors, such as TSOP 1738, are employed for obstacle detection. IC 7805 is a DC regulated IC of 5V. An 8-bit Atmel AVR RISC microcontroller is used, with features such as SRAM, flash memory, A/D converter, and EEPROM. The motor driver circuit L293D works on the idea of an H-bridge, allowing voltage to flow in either direction. This paper proposes the use of robots for warehouse management. The robots are capable of searching for and retrieving objects from shelves with high precision, flexibility, and reliability, while also maximizing safety and efficiency. The proposed system is space-efficient, and can be adapted for use in various fields with different end effectors. Additional improvements, such as adding wheels, could allow for multitasking.

Adoption of next generation robotics: A case study on Amazon:

This case study discusses how Amazon faced issues related to time spent on customer orders and preparation for shipping, leading to deficiencies and a negative image. To solve the problem, Amazon enhanced its usage of robots in the warehouse and supply chain management process. The study concludes with a brief note on the impact of robots on improving the shipping network and quick delivery of products to customers. The study focuses on Amazon's use of robotic technology to address issues with product delivery and labor shortages. The case highlights the company's expansion of its robotic workforce in warehouses and supply chains to improve efficiency and meet customer demands for quick delivery. The study also explores the reasons behind Amazon's increased use of robotics to reduce labor costs and improve performance, as well as the challenges the company faces in the competitive global market. The case concludes with a discussion of potential areas where new generation robotics could help the company further improve its operations. Amazon faces several problems, including inventory management, distribution network and shipping service, positioning against business rivals, and failure to meet customer experience and order delivery. Inventory management includes the challenge of managing a large number of warehouses and offering more products on its site, which can become unmanageable. The distribution network and shipping service is another challenge, including failed deliveries and overcapacity and shortage of materials. Positioning against business rivals is a significant problem, including the faster delivery of products and services by competitors such as eBay and Netflix. Amazon has also faced difficulties meeting customer expectations, including hard return policies and delays in product delivery. Amazon invested in the Kiva robotic systems to improve order picking and speed up order turnaround times, increase flexibility and scalability, improve sales and customer satisfaction, and reduce electricity consumption while increasing inventory accuracy and security. The robotic systems were also beneficial in reducing training time, avoiding downtime, and providing continuous order fulfillment benefits. Adopting new innovations, such as using robots in Amazon's warehouses, had significant impacts on the company's performance and efficiency. Benefits included quicker delivery, reduced delays in product delivery, increased efficiency and financial gains, improved productivity, and customer assistance. Additionally, the use of robots allowed Amazon to shift resources from physical stores to warehouses, retain existing customers and attract new ones, and improve its competitive advantage. The adoption of new technology also allowed the company to offer high-touch personalized experiences to its customers, meeting their demands for faster order fulfillment and immediate gratification after a purchase.

The design of a low-cost automatic ground vehicle for warehouse stock management:

This project designed and developed a low-cost automated ground vehicle for stock management in ecommerce warehouses in African countries. The article discusses the use of automated ground vehicles (AGVs) in warehouse management and the challenges associated with their implementation. The article also presents the design specifications and system design for a proposed AGV, which includes a software system for navigation, a frame for supporting all parts, a mechanical lifting mechanism, and an electrical system for control. The Pugh matrix was used to select components and materials for the design. The design specifications for the AGV include low production, installation and operational costs, minimum changes to the environment, easy and safe operation, wireless communication, precise line following, and obstacle avoidance. The Pugh matrix was used to evaluate four microcontroller boards for the AGV project. ESP32 development board was selected for the project because it is cheap, easy to program, and has built-in wireless connectivity, making it easy to send, receive, and implement commands. The automatic ground vehicle (AGV) will be controlled through a flowchart that receives wireless commands via a WIFI connection. The AGV can be commanded to pick up a shelf or go to the charging station, and it will plan its path and navigate accordingly. The robot will relocalize using RFID tags at corners to improve accuracy, and once it reaches the goal, it will wait for another command. The AGV frame design will have a platform for mounting sensors at the front and a lifting mechanism on top. The robot will be close to the ground for stability and increased sensor accuracy. The working environment for the robot is defined using an occupational grid, where each coordinate represents nodes that can be free or occupied. The wavefront algorithm will be used for path planning and navigation in the warehouse workspace, where the workspace is divided into equal polygons, each assigned (x, y) coordinates and labelled with 0s and 1s for free and occupied spaces respectively. The wavefront algorithm will be implemented using Python, and the path will be sent to the ESP8266 via serial communication and then to the robot through the webserver. The robot's frame design will have a platform for mounting sensors in front and a lifting mechanism on top. The localization results indicate that the robot ESP8266 can read RFID information within a maximum range of 0.02m. The communication between ESP8266 and the webserver, as well as between the robot ESP8266 and the server, has been implemented successfully, including the transmission of ultrasonic sensor and RFID information. However, the navigation test could not be performed due to insufficient friction between the robot wheels and body, which prevents the wheels from turning 90 and 360 degrees. A solution is needed to address this friction problem before the navigation test can be performed.

<u>A comprehensive study for robot navigation techniques:</u>

This paper presents the design of a Brazilian Warehouse Mobile Robot, which is a Multi-Robot System (MRS) based on the concept of Robotic Mobile Fulfillment Center (RMFS). The paper outlines the mechanical, electrical, and computational configuration of the robot and provides a simulation environment based on ROS for a warehouse setting. The RMFS system is advantageous compared to traditional AGVs due to its ability to make real-time decisions with a level of autonomy based on computer vision and powerful sensors such as LIDAR. The system is made up of three main components: Robot Drive Units, Inventory Pods, and Workstations, where workers perform replenishment, picking, and packing. The article discusses the concept of RMF, a goods-to-person AMR system used for order-picking and sorting processes in warehouses. AMRs are mobile robots that can operate autonomously using sensors, cameras, and maps. Automated warehouses are characterized by a high degree of intelligence and minimum human interference, with various types of picking equipment used, such as layer pickers, dispensers, and robots. The recent field of AMRs applied to logistics and warehouses has grown considerably in recent years. The proposed robotic platform consists of two subsystems: the lift system and the traction system. The design is compact and allows for electronic devices to have enough space to avoid overheating. The dimensions of the robot were calculated based on the needed robustness and compactness to fit under the shelf racks it would carry. The mobile motion system has six wheels, with two normal wheels placed in a differential drive manner powered by electric DC motors. The gear transmission system was chosen for reliability and ease of maintenance. The lift system of the proposed robotic platform uses power screws to convert rotation motion into translation motion and maintain momentum without an actuator. Two power screws are used to lift the lift plat in a synchronous manner and are rotated by a pair of bevel gears. The system has a total reduction of 6 to 1 and uses angular contact bearings and roller bearings for support. The lift and mobility systems are united in the same structural part that acts as the backbone of the robot. The structural part consists of a horizontal square steel sheet that receives the vertical load from the power screws and transmits it to the six wheels through its shaft bearings. The sheet is twice as thick as any other sheet used in the project and is cut using a CNC cutting machine and welded together. All bearings are assembled in machined housings and then welded to the frame of the robot. This section describes the design of the electric and computational architecture for a proposed robotic system, which includes a power supply, the Intel NUC 8 Rugged minicomputer as the system core, and various sensors and peripherals. The system is powered by a bank of Li-Po batteries and includes two H-bridges for motor control, 9-axis IMU, Garmin LIDAR, camera module, and 125kHz RFID reader for shelf recognition. The system is connected to the management broker via IEEE 802.11ac wireless protocol for high bandwidth and low-latency operation. The authors created a simulation environment using ROS and the Gazebo simulator for testing and developing algorithms for their mobile robotic system. The environment consists of a warehouse layout with shelves, delivering, recharging, and picking areas. A robot with similar dimensions to the physical design was modeled with a differential drive, lifting tray, and all necessary sensors. The simulation package allows for monitoring and testing of the robot's health and performance. This section discusses the effort calculations for the motors to lift up to 150kg and its impact on power consumption. Additionally, the authors present a ROS-Gazebo package with a robot and shelves, which is available to the community. The paper describes the development of a prototype Brazilian robot for warehouse systems that can lift up to 150kg and operate for up to 12 hours without recharging. The robot was designed using locally available parts and a ROS-based simulation environment was created for testing different configurations and multi-agent planning algorithms. The simulation package is available to the community and future work will focus on improving the mechanical design, building the real robot, and testing the multi-agent simulation environment.

CHAPTER.3
Mechanical Design

Chapter 3

Mechanical Design

The stability and flexibility of any system depends on its mechanics and in case of pick & place operations it is vital to select proper mechanical design as it involves lifting an object from source to placing at destination. The lift system is basically the point of discussion and we will discuss various type of lift systems in this section.

- Simple Lifting Mechanism
- Screw-Jack Lifting Mechanism
- Forklift Mechanism

3.1 Simple Lifting Mechanism

There are multiple designs of pick and place robots, based on the specific application for which they are used. The basic principle of most of these designs is on similar lines.

These robots are typically mounted on a stable stand, and have a long arm that can reach their entire area of operation. The end of arm attachment is specialized to the type of objects the robot intends to move.

These robots can transfer items from a stationary surface to a stationary surface, stationary to a moving surface, moving to a stationary surface, and moving to a moving surface (such as between two conveyor belts).

A robot has several dedicated parts, such as:

Robot Arm tool: A robotic arm, also known as a manipulator, is the extension of the robot by using cylindrical or spherical parts. links, and joints.

End Effector: The end effector is the accessory at the end of the robotic arm, that does the required job such as gripping objects. The end effectors can be designed to perform different functionalities based on requirements.

Actuators: Actuators create the motion in the robotic arm and end effectors. The linear actuators are basically any type of motor, such as servo motor, stepper motor, or hydraulic cylinder.

Sensors: You can think of sensors as the eyes of the robots. The sensors do the tasks like identifying the position of the object.

Controllers: Controllers synchronize and control the movement of different actuators of a robot, thereby being the brain behind the smooth robotic operation.

Figure 3.1: Pick and Place Robotic Arm

3.2 Screw-Jack Lifting Mechanism

A screw jack often referred to as a jackscrew or a mechanical screw jack, is a mechanical power transmission equipment that is used in applications where linear motion is required. Jackscrews are used to lift heavy and moderate loads like raising and lowering of stages, provide adjustable support in construction applications, lift vehicles, lift machinery for assembly purposes etc. Jackscrews can be used as linear motors, linear actuators, or mechanical lifts depending on the type of motion. Jackscrews can be driven by either AC/DC motors or manual handwheels.

The main components of screw jacks are; trapezoidal lifting screw also known as lead screw, worm screw, worm gear and gear housing. A worm screw is rotated manually or by a motor. With the rotation of the worm gear, the lead screw in it moves upwards or downwards linearly. The feed rate of the screw depends on the turning speed, the number of teeth of the gears and the size of the screw pitch.

In some models of jackscrews, the lifting screw does not move up and down. It only rotates around its axis. A lifting nut (also known as a travelling nut) moves along the lead screw. The lifting nut of the screw jack is made of bronze to decrease friction.

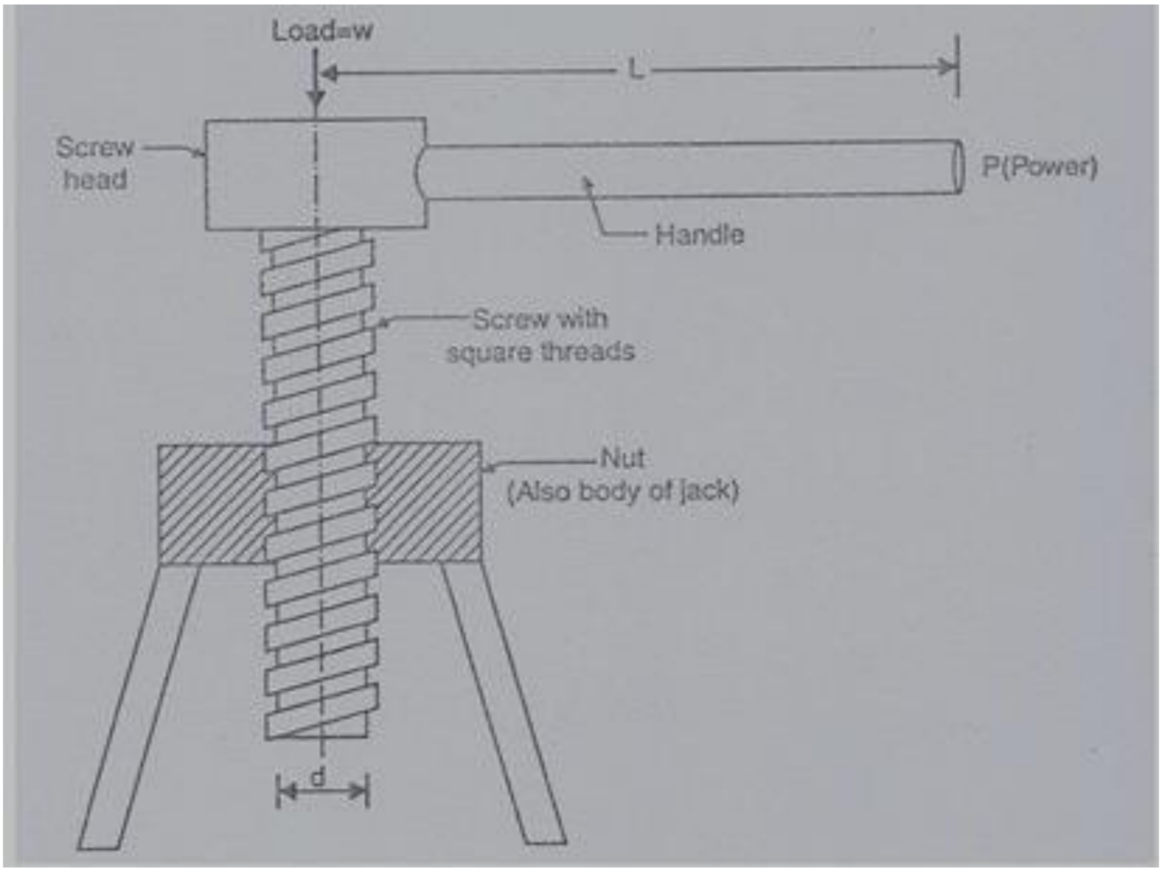


Figure 3.2: Screw-Jack Lifting Mechanism

3.3 Forklift Mechanism

Forklifts use a combination of hydraulics, a pulley system, and other things to lift heavy materials across different distances. They are commonly used in warehouses, construction sites and other places that need to transport lots of heavy materials. The lifting controls use one lever for moving the load up and down and another for tilting the load back and forth. Pushing the lifting lever forward moves the load up while moving it backwards moves it down. Tilting, however, is a bit more complex. The tilt function involves two more hydraulic cylinders found at the base.

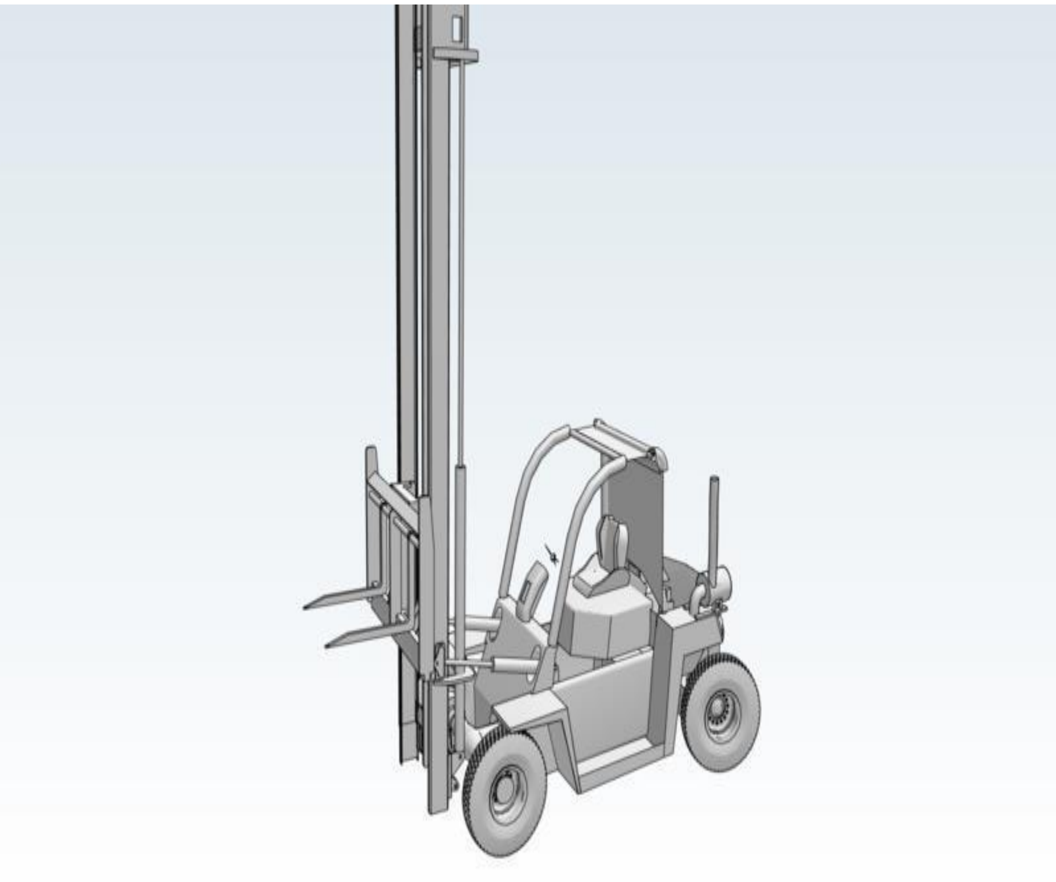


Figure 3.3: Forklift Mechanism

Here is what happens when the tilt lever is moved in both directions:

- **Forward:** Air is pumped into the chamber. The rise in pressure level pushes the piston head and results in the mast tilting away from the frame.
- **Back:** Air is gently released from the chamber and air is pumped in the other pair of cylinders we mentioned earlier from the hydraulic system. The masts lean towards the vehicle when those pistons push that pair of cylinders forward.

CHAPTER.4
Autonomous Navigation

Chapter 4

Autonomous Navigation

Even after completing the complex designing of a robot, the more serious problem is how it will navigate from source to destination? In this section we will try to describe how an autonomous robot navigate in dynamically changing environment, obstacle detection and avoidance, path planning between source and destnation.

Autonomous navigation is the ability of a vehicle/rover to learn and execute movement of its path without any human intervention. The automation facility liberates mobile robots to perform various tasks in known and unknown environment as they are not attached to a physical location as they have intelligence of navigating around automatically. Automation navigation has the ability to reduce manual errors and negligence which use to result in collisions. While performing navigational tasks, robotic systems, make use of capabilities that involve modelling the environment, locating, and mapping its position, planning path and motion planning.

4.1 Perception

Perception is the process of understanding the environment based on measurements. It is environmental perception that enables a system to respond intelligently to what is out there – even when it differs from any expectations. Perception can be divided in three categories:

- Image Processing: When a signal processing is applied to images it is often called as image processing. It deals with either appearance of data (which comes from cameras of various kinds) or geometry data (which comes from imaging range sensors).
- Geometric Computer Vision: It focuses on inferring shape or motion or both or on constructing models or maps. Also intended to focus on understanding spatial relationships, on localization of objects in the scene, or on relative motion of the sensor and of objects.
- Semantic Computer Vision: It is used to recognize the nature of objects or parts of the scene, and try to understand or interpret the content of the scene.

4.1.1 Sensors for Perception:

All the relevant perception sensor can be separated into three categories:

- Contact Sensor
- Non-Contact Sensor
- Inertial Sensor

Contact Sensor: Contact or touch sensors are one of the most common sensors in robotics. These are generally used to detect a change in position, velocity, acceleration, force, or torque at the manipulator joints and the end-effecter. There are two main types, bumper and tactile. Bumper type detect whether they are touching anything, the information is either Yes or No. Tactile sensor is more complex and provide information on how hard the sensor is touched, or what is the direction and rate of relative movement.

Non-Contact Sensor: These are often called as radiative sensors. Radiative sensors provide the basic capacity to interpret the visible area around the robot without having to make physical contact. It can be further classified into active and passive.

- An active sensor emits its own radiation.
- A passive sensor relies on ambient radiation.

There are mainly six types of non-contacting sensor are as:

(1). Visual and optical sensor.

(2). Magnetic and inductive sensor.

(3). Capacitive sensor.

(4). Resistive sensor.

(5). Ultrasonic and sonar sensor.

(6). Air pressure sensor.

A scanning sensor actively orients the sensing elements perhaps in order to generate a synthetically wide effective field. A proximity sensor is a binary detector of whether an object is in the field of view. A ranging sensor provides a range value to the objects or scene elements being sensed. Ranging sensor include SONAR, RADAR and also LASER RADAR called LIDAR or LADAR. These sensors are ultimately sensitive to types of radiation used as such as sound, light, infrared etc. Light Detection & Ranging is called Lidar, typically single range pixel sensor provides high resolution in both range and angle. Its works on the phenomenon of photoelectric effect.

4.2 Simultaneous Localization and Mapping

SLAM refers to the problem of trying to simultaneously localize (i.e., find the position/orientation) with respect to its surroundings, while at the same time mapping the structure of the environment. SLAM is a concept which solves a very important problem in mobile robotics. SLAM is not necessarily a computer vision problem and need not involve visual information at all. It can be done with LIDAR and IMU alone, however here we have considered visual SLAM. SLAM is made up of two parts: -

• ***Mapping:*** Building a map of the environment which the robot is in

• ***Localization:*** Navigating in its environment using map while keeping track of the robot's relative position and orientation.

There numerous open-source ROS packages available for SLAM like ORB SLAM and there are few integrated solutions which use a stereo camera and perform visual SLAM onboard like the Intel tracking camera T265.

4.2.1 Sensors for localization

We can broadly classify localization sensors into 2 categories:

a) external source-based sensors and

b) internal source-based sensors.

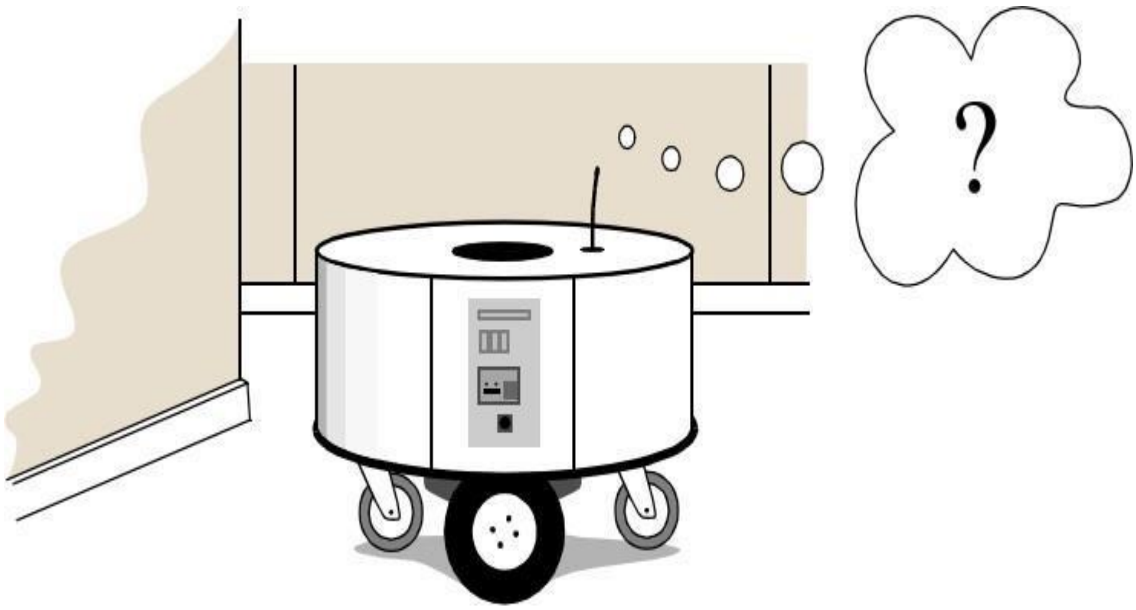


Figure 4.1: Where am I?

Internal sensors use rover's velocity and/or wheel motion to localize itself whereas external sensors use RF source for localization. The most popular external sensors are GNSS sensors [ex-GPS]. The accuracy of the GNSS sensors could vary between 3 meters and above, there are few GNSS devices with the accuracy of 10 mm but they require a precisely located base station and cannot work in indoor/sheltered environments. Other possibilities include using RF signals from WIFI or Bluetooth device to get rovers relevance position from signal source to localization itself. Onboard sensors like Inertia Measurement Units [IMU] and motor encoders are common internal sensors. IMU's works by sensing motion including the type, rate and direction of that motion using a combination of accelerometers and gyroscopes. Accelerometers are placed such that their measuring axes are orthogonal to each other. An IMU works by detecting the current rate of acceleration, as well as its changes in rotational attributes, including pitch, roll and yaw. This information is used by the rover to generate its motion trajectories and other purposes. A motor encoder is an electromechanical device that provides an electrical signal that is used for speed and/or position control. Encoders turn mechanical motion into an electrical signal these are used by the rover to monitor distance travelled, we have encoders on all wheels of the rover to understand its direction and speed of motion.

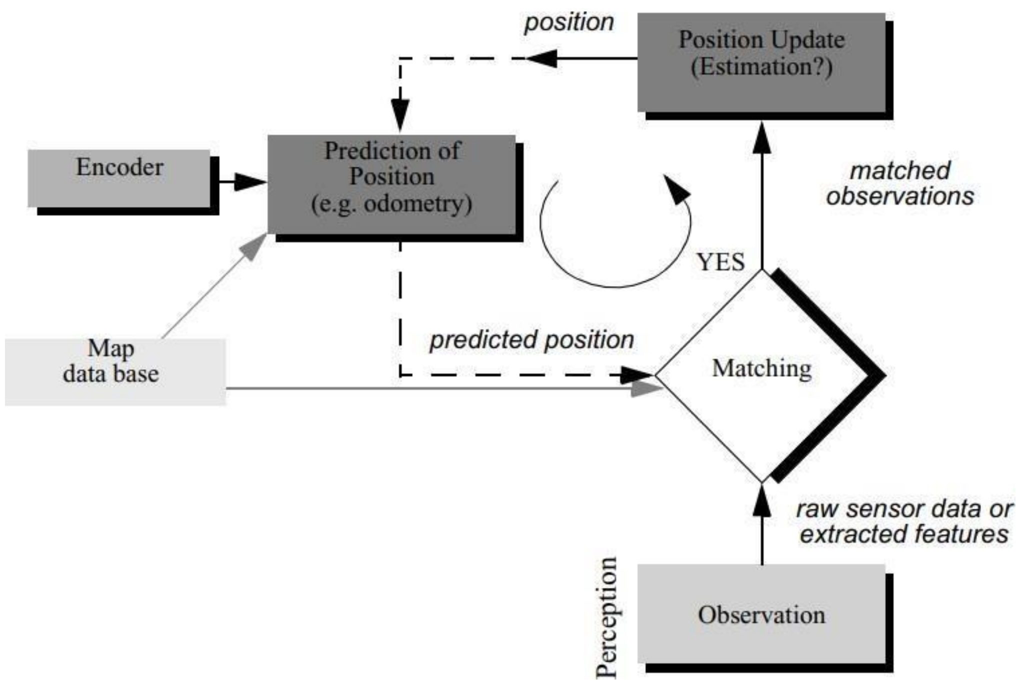


Figure 4.2: General schematic of robot localisation

4.2.2 Types of SLAM

- **Geometric SLAM:** Geometric Metric SLAM is to compute 3D maps with accurate mathematical equations. According to using different sensors, Geometric metric SLAM consists of monocular SLAM, multilocular SLAM, and multi-kind sensors SLAM.
- **Learning SLAM:** Learning SLAM is a new topic recently due to the deep learning development. It is a single category different from 3D metric SLAM and 2D topological SLAM. Learning SLAM can obtain camera pose and 3D map but need prior dataset to train a network.
- **Topological SLAM:** Topological SLAM does not need accurately computing 3D map and represents environment by connectivity or topology. It uses a hierarchical description of the spatial environment, where a topological network description mediates between a control and a metrical level, also distinctive places and paths are defined by their properties at the control level, and serve as the nodes and arcs of the topological model.
- **Marker SLAM:** The works of image base camera localization in both known and anonymous environments above. In addition, there are some works to localize cameras using some prior environment knowledge but not a 3D map such as markers. These works are regarded in semi-known environment.

4.3 Path Planning for Robot

Path planning is one of the most prominent and essential part of autonomous mobile robot navigation. For the past two decades, researchers are working on path planning problem for which several methods have been developed. Path planning involves the determination of collision-free path from one point to another while minimizing the total cost of the associated path. Depending on the nature of environment, path planning can be divided into static and dynamic environment. If obstacles change their position with respect to time, it is referred as static path planning and if obstacles change their position and orientation with respect to time, then it is referred as dynamic path planning. This knowledge can further be divided into online and offline algorithms. In online path planning, the information about surrounding is obtained from separately attached local sensor installed on robot, then robot construct the map of environment from the information being fed from the locally attached sensors. In offline path planning, robot has complete information of surrounding environment without the aid of sensors.

Navigation can be divided into two types: Global and local Navigation. For global navigation type, prior knowledge of environment should be known, which is also called as Off-line mode for path planning, For local navigation, also known as On-line mode for path planning in which robot decides its position and orientation and can control its motion using externally equipped sensors for example: Infrared sensor, ultrasonic sensor, LASER, and vision sensor (camera) can be employed to autocorrect the orientation of robot via software.

4.3.1 Global Navigation Methods

A. Artificial Potential Field: Artificial Potential Field (APF) is a nature inspired technique. The basic idea of APF is to fill the robot environment with the artificial potential field in which obstacles are repelled using repulsive force and robot is attracted toward the goal using attractive force. The potential field depends on two forces, that is, attractive and repulsive force. The goal produces the attractive force towards robot and obstacles produce repulsive force, which is inversely proportional to the distance from robot to the obstacles and directing toward obstacles. In APF field, robot travels from high potential to low potential.

In potential field approach, an attractive field is created to reach the goal. The potential field is usually defined across the entire free space, and in each time step, potential field is calculated at the robot position, which calculates the induced force applied by the field. The robot then should move according to above-mentioned forces. The major problem with APF is that robot may trap is local/global minima problem, that is, the robot is stuck at point where both fields/forces cancel out the effect of each other and does not allow the robot to move further or even backward.

B. Dijkstra Algorithm: The graph searching method is considered as the simplest method for finding a path for robot. It is considered as a well-defined, effective, and efficient method with less time and computational complexity in identifying a nonobstructive path. Environment is constructed for robot and the path is connected by line through robot easily reaches to target. The process continues until a better and optimal solution is achieved from one node to another. When the robot reaches the desired target, the robot is allowed to proceed to new location. Dijkstra algorithm is considered as graph searching method that solves the optimal path problem with non-negative edge path costs producing shortest path. It is used to find path cost from single point to single destination. This it is used for determining the shortest distance with lower cost between the initial node and other nodes in a graph. The main crux of the algorithm is to repetitively calculate the shortest distance from a initial point to end point at the same time excluding longer distance path.

C. A* Algorithm: A* is a search algorithm that can also be used to find path-finding. The algorithm continuously searches for unexplored location in graph. All the locations are searched in graph when the target location is reached, the algorithm stops. And if target is not achieved then it sets all the neighbours for exploration to search for the shortest path. A-star algorithm is the popular for path finding in games.

The A star (A*) is a search method that uses a heuristic function, h(n), where n represents a node n. To each node n is associated an approximation h(n) of the cost of a path from n to a goal node, while $h*$ (n) corresponds to the real distance (cost) from n to a goal node. A heuristic h is consistent if and only if: (i) h(n) = 0 (if n is the goal node); and (ii) for all nodes and their successors n`, the estimated cost of moving from node n to the goal node is not greater than the cost of moving from node n to node n 0 plus the estimated cost of moving from node n` to the goal node.

The Time Enhanced A* (TEA*) is an extension of the A*, used when there are multiple vehicles. It contains an additional component – time. This component allows a better prediction of the vehicles' movements during the run time. TEA* consists of an incremental algorithm that builds the path of each vehicle considering the movements of other mobile robots. This feature allows the algorithm to produce conflict free routes and, at the same time, deal with deadlock situations, since the paths are constantly recalculated and the map information is updated at each iteration. This way, the unpredictable events are considered in the input map, allowing to avoid the main challenges of any multi-robot approach, such as collisions and deadlocks. Each node on the map has three dimensions: the Cartesian coordinates (x, y) and a representation of the discrete time. The time is represented through temporal layers, k = [0; TMax], on which TMax represents the maximum number of layers. Each temporal graph is composed of a set of free and occupied/obstacles nodes. The path for each robot is calculated during the temporal layers. In each temporal layer, the position of each vehicle is known and shared with the other vehicles. This way, it is possible to detect possible future collisions and avoid them at the beginning of the paths' calculation. Each robot can only start and stop in nodes and a node can only be occupied by one vehicle on each temporal layer. The operation of the TEA* algorithm is similar to A*, since for each AGV, during the path calculation, the next neighbour node analysed depends on a cost function, f(n), given by the sum of two terms: the distance to the initial vertex, g(n), and the distance to the end point, h(n).

D. Rapidly exploring Random Tree: The Rapidly exploring Random Tree (RRT) algorithm incrementally constructs a search tree in the configuration space until the goal configuration can be connected to one of its nodes. The operation of the RRT, exemplified in involves the iterative execution of the following steps:

1) A random configuration, q_{rand}, is sampled in the configuration space.

2) The tree is searched for a configuration q_{near}, which is the nearest node in the tree to q_{rand}.

3) A new configuration q_{new} is created by moving a predefined distance d from q_{near} in the direction of q_{rand}.

4) If q_{new} is a valid configuration that falls in C_{free} (unobstructed space), and if the local path between it and q_{near} is collision-free, then q_{new} is added to the tree as a new node and an edge is created between q_{new} and q_{near}. However, if q_{new} falls in Cobs (obstacle space), and if the local path between it and q_{near} has collisions, then is not created an edge between q_{new} and q_{near}.

The most common metric for the nearest-neighbour selection is the Euclidean distance between points. In this case, the expansion pattern of the tree is modelled by the Voronoi diagram over the nodes within the tree. The probability of a node being expanded is directly proportional to the volume of its corresponding Voronoi region. Nodes that have a larger Voronoi region (i.e., the portion of the space that is closer to the node than to other nodes of the tree) are more likely to be chosen for expansion and are referred to as major nodes. This way, the tree is pulled towards unexplored areas, spreading rapidly in the configuration space (as the Voronoi regions of samples become approximately equal in size, the exploratory behaviour gradually shifts from expansion of the tree to refinement). In the case of the Euclidean metric, these nodes tend to lie on the outside of the tree during the initial exploration. Conversely, inner or minor nodes have smaller Voronoi regions and often lie on the inside of the tree. Once the tree has explored the state space, these become major nodes as the algorithm begins to fill in the space. This phenomenon of favouring some nodes over others is referred to as the Voronoi bias, and yields an initial preference towards the exploration of the state space.

4.3.2 Local Navigation Methods

A. LIDAR: In local navigation techniques, sensors are usually employed to control the orientation and position of robot. For such use, LIDAR sensor is frequently used for automation purpose. LIDAR works independently as compared to GPS system; therefore, it has the capability of mapping the environment. LIDAR can be used independently but when coupled with other sensors like GPS, Inertial navigation system, and camera, it gives improved results. For example, together with camera it provides powerful positioning tool. This system can be employed for mapping the local environment to locate and identify the landmarks position, which in general is called as SLAM (Simultaneous Localization and Mapping). With the help of this technique, mobile robot automatically corrects its position and orientation remotely. Motors encoder sensors employment with LIDAR can also improve and enhance accuracy.

. B. Vector Field Histogram (VFH): It is another local navigation method used to solve the path planning problem for mobile robots. The idea of VFH is based on VFF (Virtual Force Filed) method. As the name indicates it's a field, so obstacles detected at a certain distance from vehicle will apply repulsive force on vehicle to move away from obstacles and to draw the vehicle toward goal point an attractive force is used. VFH uses a certainty grid-like radar screen, where obstacles found by sensor will count up sum certainty value at the corresponding coordinates in the certainty grid. Which means, higher the certainty value reveals that the real object is detected by sensor range. In real time, the grid is continuously updated at every instant hence this method is suitable for sparse moving objects.

The path guidance for AGV can be classified as static path determination or dynamic path determination. For static path vehicle uses pre-defined path between its origin and destination point. In direct guidance system, embedded wire, magnetic tapes, radiofrequency identification (RFID) chips, and dead reckoning are used to guide vehicle. Static path can also be subdivided into two categories: unidirectional and bidirectional. In unidirectional system, vehicle is allowed to follow and travel in single direction whereas in bidirectional system, vehicles can navigate in any direction; this functionality is achieved by using turn-around points or bidirectional vehicles which can move forward or backward. In dynamic path system, vehicle behaves autonomously to determine path by detecting and avoiding obstacle. In this system, the vehicle knows its destination via some coordinates system. Internal navigation system is used by vehicle to reach its destination.

4.4 Obstacle Detection

Local obstacle avoidance focuses on changing the robot's trajectory as informed by its sensors during robot motion. The resulting robot motion is both a function of the robot's current or recent sensor readings and its goal position and relative location to the goal position. The obstacle avoidance algorithms presented below depend to varying degrees on the existence of a global map and on the robot's precise knowledge of its location relative to the map. Despite their differences, all of the algorithms below can be termed obstacle avoidance algorithms because the robot's local sensor readings play an important role in the robot's future trajectory. We first present the simplest obstacle avoidance systems that are used successfully in mobile robotics. The Bug algorithm represents such a technique in that only the most recent robot sensor values are used, and the robot needs, in addition to current sensor values, only approximate information regarding the direction of the goal. More sophisticated algorithms are presented afterward, considering recent sensor history, robot kinematics, and even dynamics.

4.4.1 Vector Field Histogram (VFH)

The VFH techniques overcome this limitation by creating a local map of the environment around the robot. This local map is a small occupancy grid, populated only by relatively recent sensor range readings. For obstacle avoidance, VFH generates a polar histogram as shown in figure below.

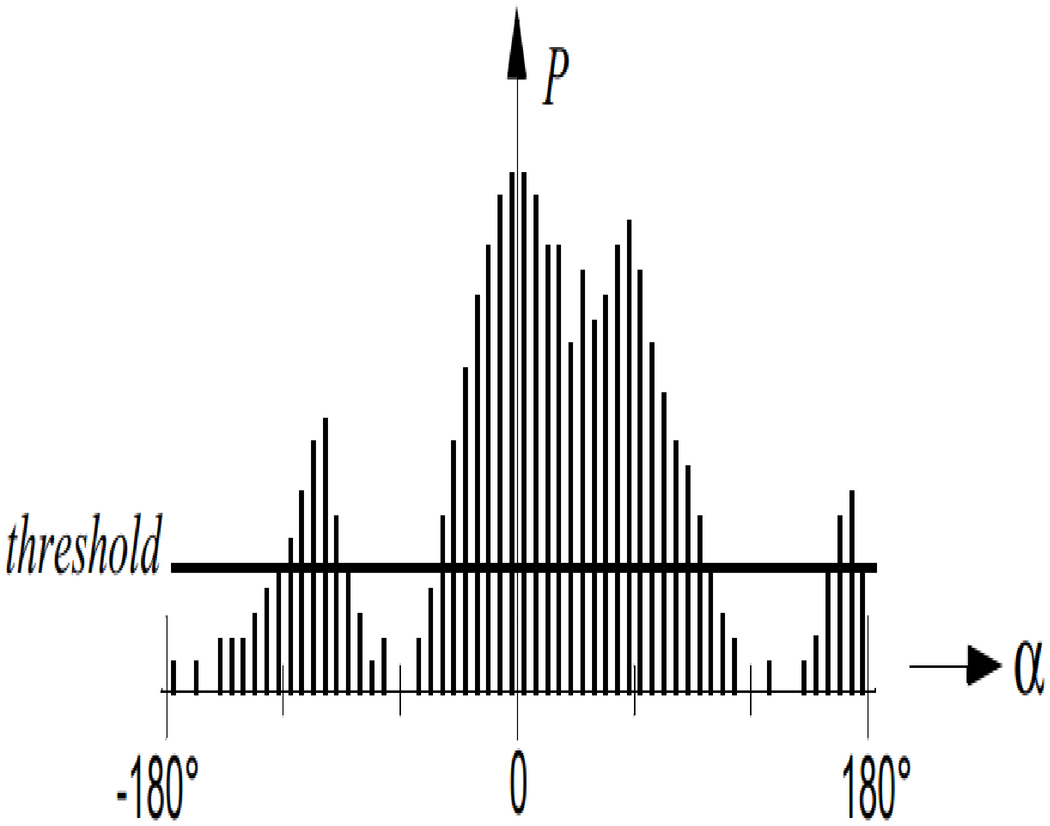


Figure 4.3: Polar Histogram

The x-axis represents the angle at which the obstacle was found and the y-axis represents the probability that there really is an obstacle in that direction based on the occupancy grid's cell values. From this histogram a steering direction is calculated. First all openings large enough for the vehicle to pass through are identified. Then a cost function is applied to every such candidate opening. The passage with the lowest cost is chosen. The cost function has three terms:

G = a · target_direction+b · wheel_orientation+c · previous_direction

target_direction = alignment of the robot path with the goal;

wheel_orientation = difference between the new direction and the current wheel orientation;

previous_direction = difference between the previously selected direction and the new direction.

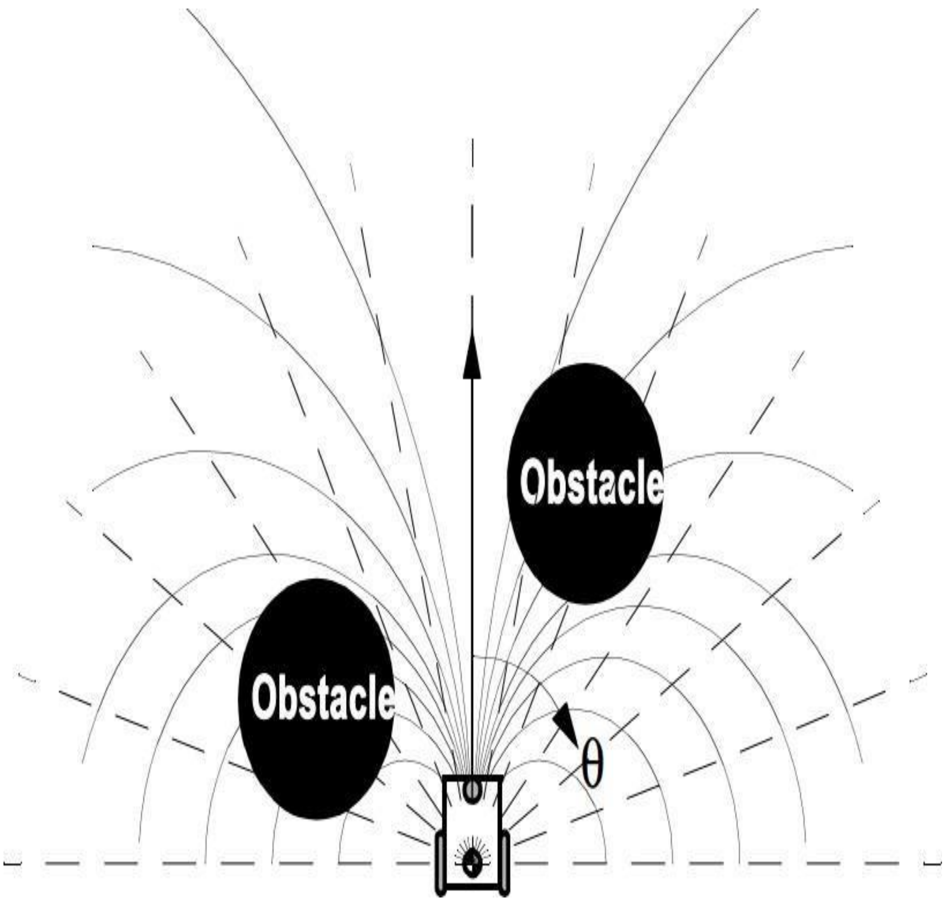


Figure 4.4: Robot and blocking Obstacles

In the VFH+ improvement one of the reduction stages takes into account a simplified model of the moving robot's possible trajectories based on its kinematic limitations (e.g., turning radius for an Ackerman vehicle). The robot is modelled to move in arcs or straight lines. An obstacle thus blocks all of the robot's allowable trajectories which pass through the obstacle. This results in a masked polar histogram where obstacles are enlarged so that all kinematically blocked trajectories are properly taken into account.

4.4.2 The Bubble Band Technique

A bubble is defined as the maximum local subset of the free space around a given configuration of the robot that which can be travelled in any direction without collision. The bubble is generated using a simplified model of the robot in conjunction with range information available in the robot's map. Even with a simplified model of the robot's geometry, it is possible to take into account the actual shape of the robot when calculating the bubble's size. Given such bubbles, a band or string of bubbles can be used along the trajectory from the robot's initial position to its goal position to show the robot's expected free space throughout its path. Clearly, computing the bubble band requires a global map and a global path planner. Once the path planner's initial trajectory has been computed and the bubble band is calculated, then modification of the planned trajectory ensues. The bubble band takes into account forces from modelled

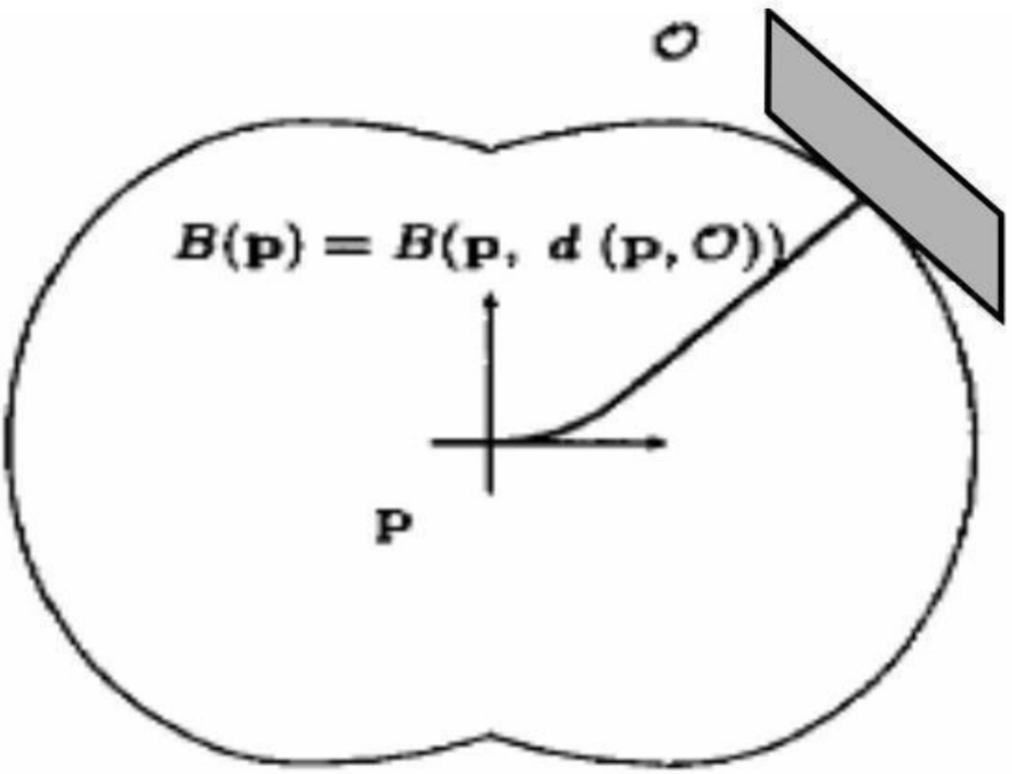


Figure 4.5: Shape of bubbles around vehicles

objects and internal forces. These internal forces try to minimize the "slack" (energy) between adjacent bubbles. This process, plus a final smoothing operation, makes the trajectory smooth in the sense that the robot’s free space will change as smoothly as possible during path execution. Of course, so far this is more akin to path optimization than obstacle avoidance. The obstacle avoidance aspect of the bubble band strategy comes into play during robot motion. As the robot encounters unforeseen sensor values, the bubble band model is used to deflect the robot from its originally intended path in a way that minimizes bubble band tension.

CHAPTER.5
Component Details

Chapter 5

Component Details

5.1 Arduino UNO

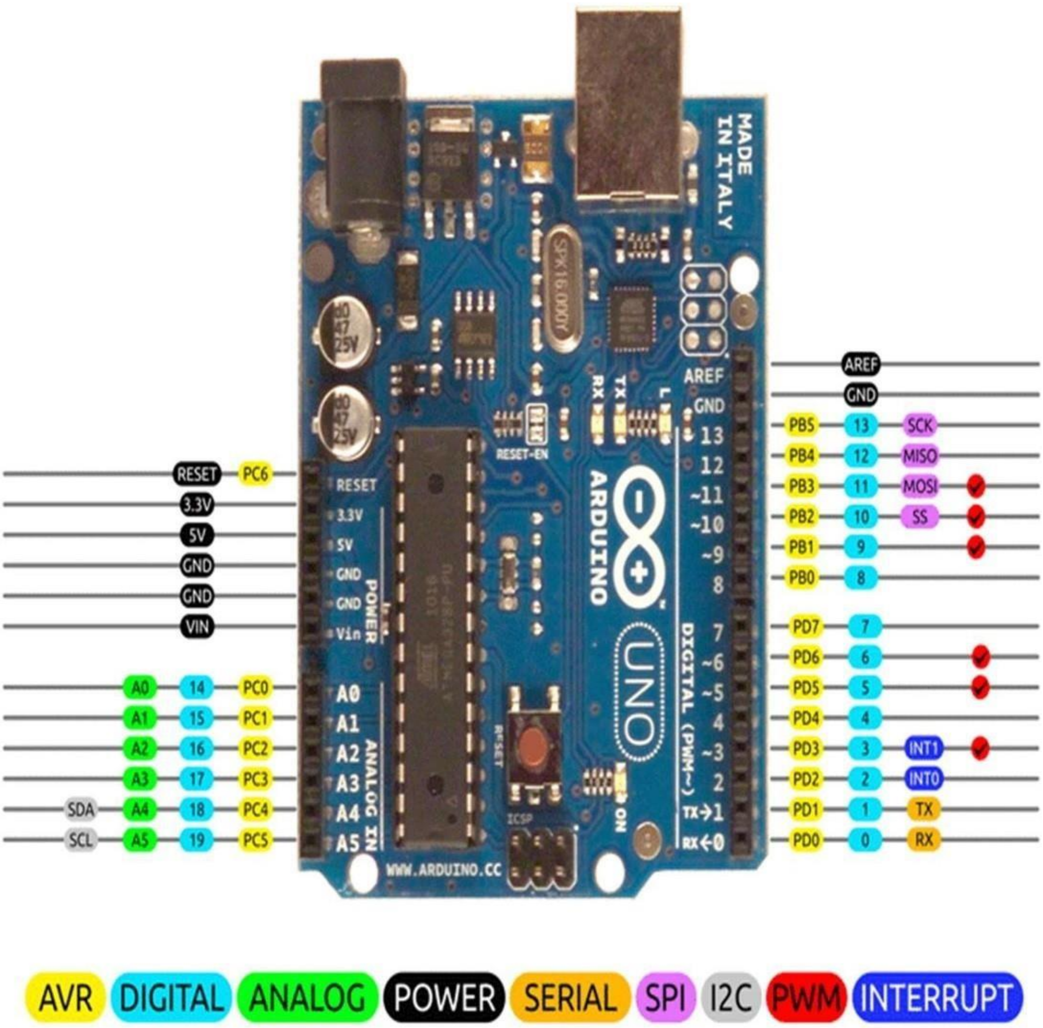


Figure 5.1: Arduino Uno Pinout

Arduino Uno is a popular microcontroller development board based on 8-bit ATmega328P microcontroller. Along with ATmega328P MCU IC, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. It is the perfect board to get familiar with electronics and coding. This versatile microcontroller is equipped with the well-known ATmega328P and the ATMega 16U2 Processor.

5.1.1 Arduino Uno Pinout Configuration:

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.

External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

5.1.2 Arduino Uno Technical Specifications:

Microcontroller	ATmega328P – 8-bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V

Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

5.2 Arduino Nano

Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328. This microcontroller is also used in Arduino UNO. It is a small size board and also flexible with a wide variety of applications.

This board has many functions and features like an Arduino Duemilanove board. However, this Nano board is different in packaging. It does not have any DC jack so that the power supply can be given using a small USB port otherwise straight connected to the pins like VCC & GND. This board can be supplied with 6 to 20volts using a mini-USB port on the board.

5.2.1 Arduino Nano Features:

- ATmega328P Microcontroller is from 8-bit AVR family
- Operating voltage is 5V
- Input voltage (Vin) is 7V to 12V
- Input/Output Pins are 22
- Analog i/p pins are 6 from A0 to A5
- Digital pins are 14
- Power consumption is 19 mA
- I/O pins DC Current is 40 mA
- Flash memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK speed is 16 MHz
- Weight-7g
- Size of the printed circuit board is 18 X 45mm
- Supports three communications like SPI, IIC, & USART

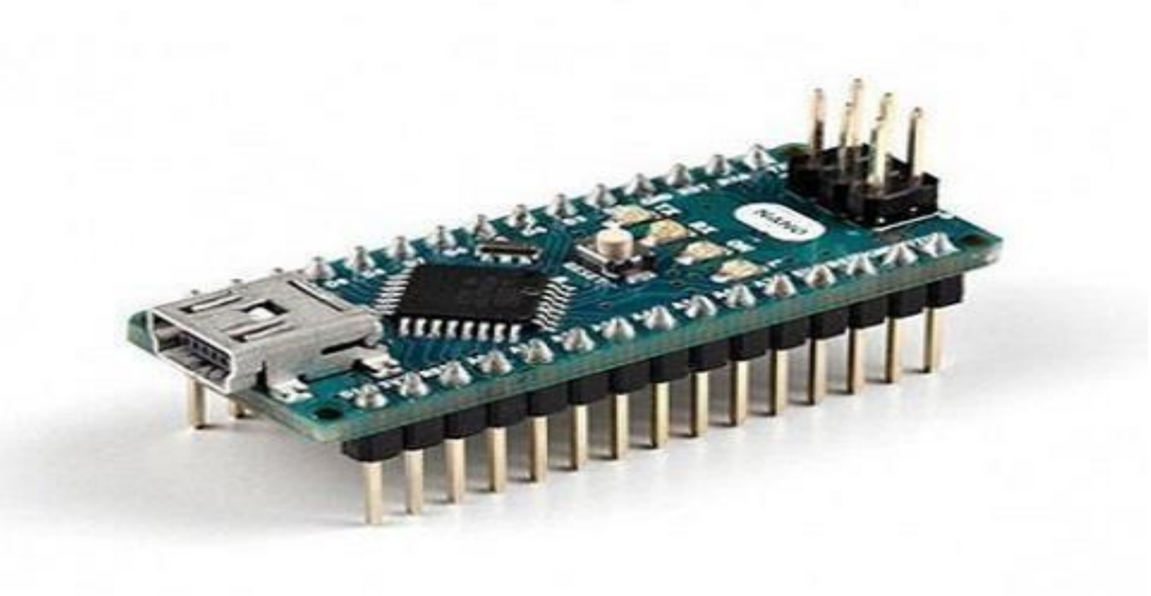


Figure 5.2: Arduino Nano Board

5.2.2 Arduino Nano Pinout:

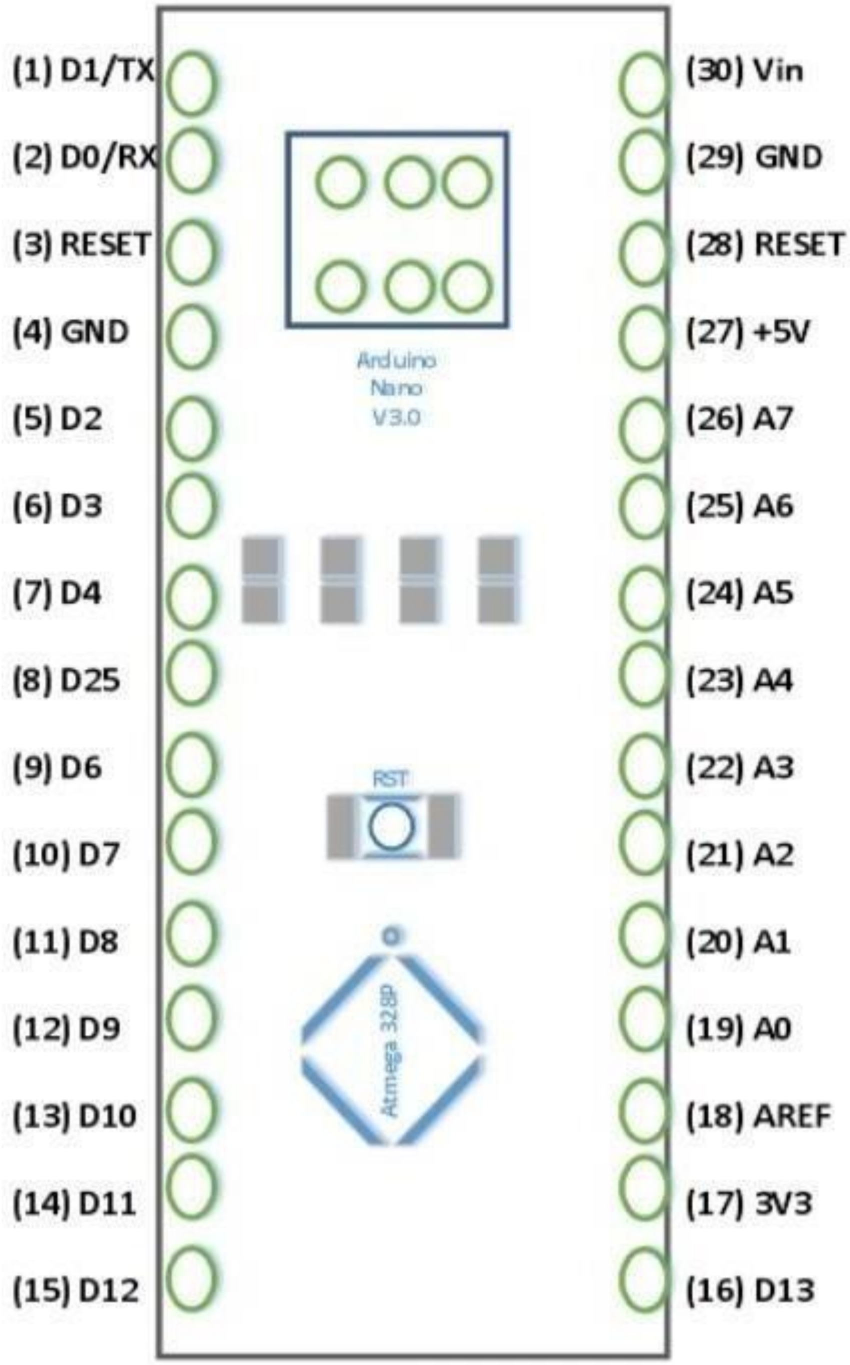


Figure 5.3: Arduino Nano Pinout

Power Pin (Vin, 3.3V, 5V, GND): These pins are power pins

- Vin is the input voltage of the board, and it is used when an external power source is used from 7V to 12V.

- 5V is the regulated power supply voltage of the nano board and it is used to give the supply to the board as well as components.
- 3.3V is the minimum voltage which is generated from the voltage regulator on the board.
- GND is the ground pin of the board

RST Pin(Reset): This pin is used to reset the microcontroller

Analog Pins (A0-A7): These pins are used to calculate the analog voltage of the board within the range of 0V to 5V

I/O Pins (Digital Pins from D0 – D13): These pins are used as an i/p otherwise o/p pins. 0V & 5V

Serial Pins (Tx, Rx): These pins are used to transmit & receive TTL serial data.

External Interrupts (2, 3): These pins are used to activate an interrupt.

PWM (3, 5, 6, 9, 11): These pins are used to provide 8-bit of PWM output.

SPI (10, 11, 12, & 13): These pins are used for supporting SPI communication.

Inbuilt LED (13): This pin is used to activate the LED.

IIC (A4, A5): These pins are used for supporting TWI communication.

AREF: This pin is used to give reference voltage to the input voltage.

5.3 Drivers L293 & L293D

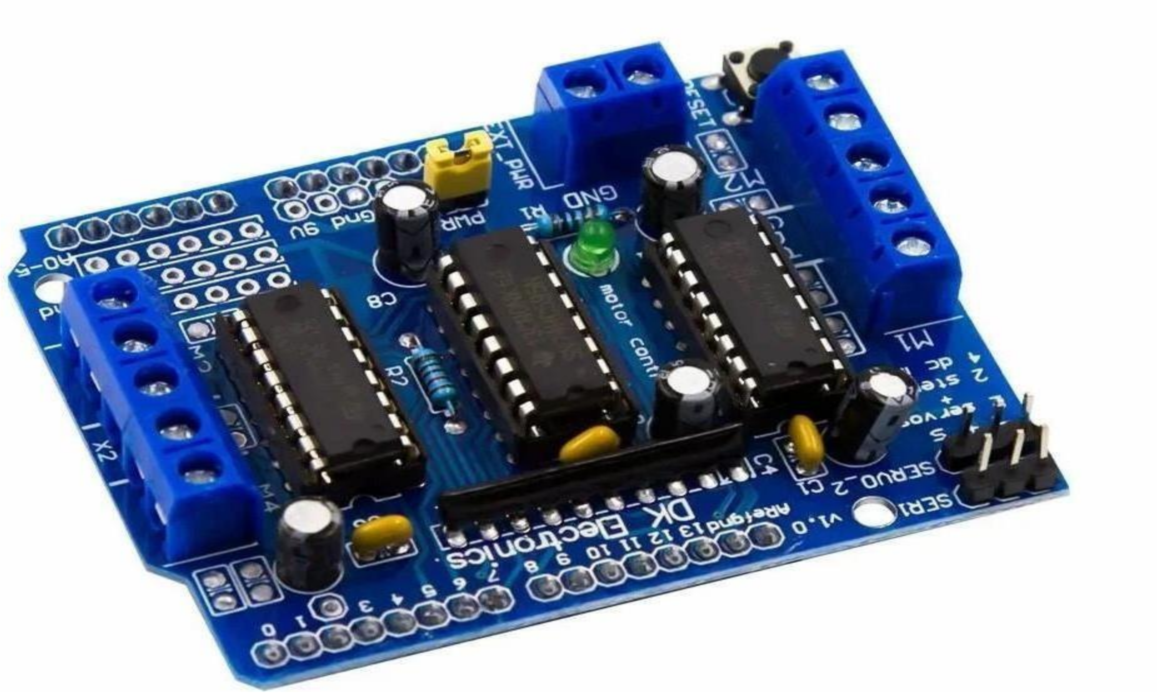


Figure 5.4: Driver L293D

Whenever we want to drive heavy loads with a small TTL signal you use a transistor, bit or a mosfet. In case the circuit becomes complex and you need to drive load with two, three or four TTL signals you use a combination of transistors or mosfet to provide high power output to load. For example, motors dc or stepper or servo require high power and 2, 3 control signals for rotation direction change and speed control. This transistor or mosfet combination is known as H-bridge circuit. Making a H-bridge circuit on bread board or pcb(printed circuit board) requires many wires to be connected and it seems like a mess also the circuit takes too much space.

L293d ic is same like an h bridge circuit with two channels. It has two half h bridge circuits residing in it. You can use it to drive unipolar, bi polar stepper motors, dc motors or even servo motors. The individual two channels can be use stand alone to drive solenoids/relays. A single channel can be used to drive a dc motor in forward (clock wise) or back word (anti clock wise) direction. Hence, we can drive two dc motors with l293d. It can also be used to output a pwm (pulse width modulation) signal. The ic came in two different versions L293 and L293D. Both have same number of operational pins with same pin names. But differ in voltage and current supply and control specifications. L293 can output more current. But the disadvantage is it has greater number of pins than L293D.

L293D

Total number of pins = 16
Ic Operating voltage = 5 – 7 v
Input voltage = 4.5 – 36 v
Output voltage = 4.5 – 36 v
Continuous Output Current = 600 mA or 0.6 A

L293

Total number of pins = 28
Ic Operating voltage = 5 – 7 v
Input voltage = 4.5 – 36 v
Output voltage = 4.5 – 36 v
Continuous Output Current = 1 A

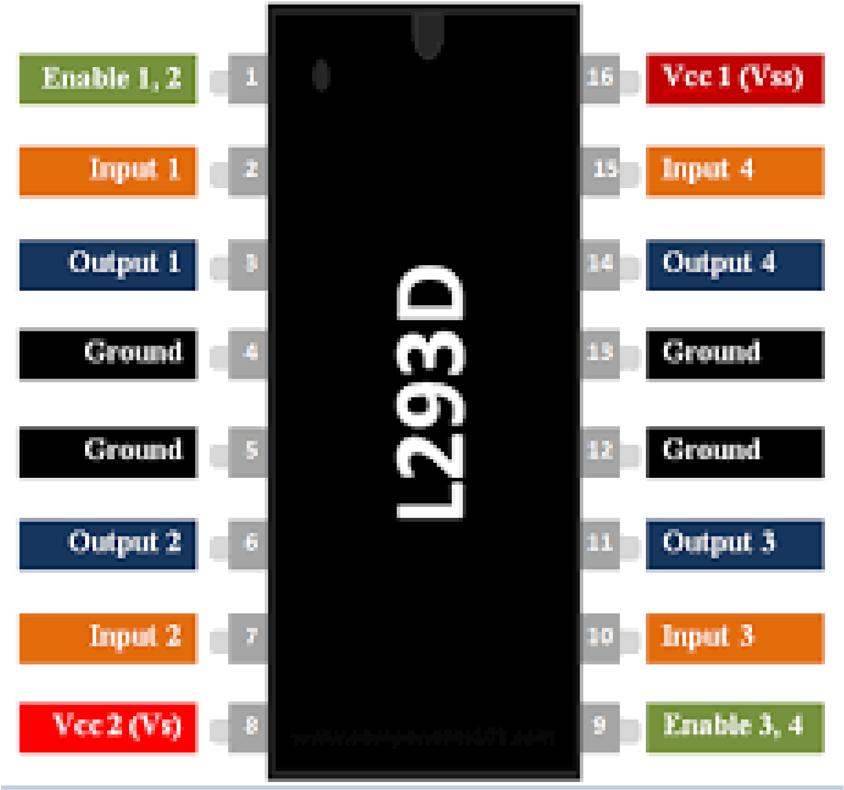


Figure 5.5: L293D Pinout

5.4 DC Stepper Motor

5.4.1 3.5RPM 12v Low Noise DC Motor with Metal Gears

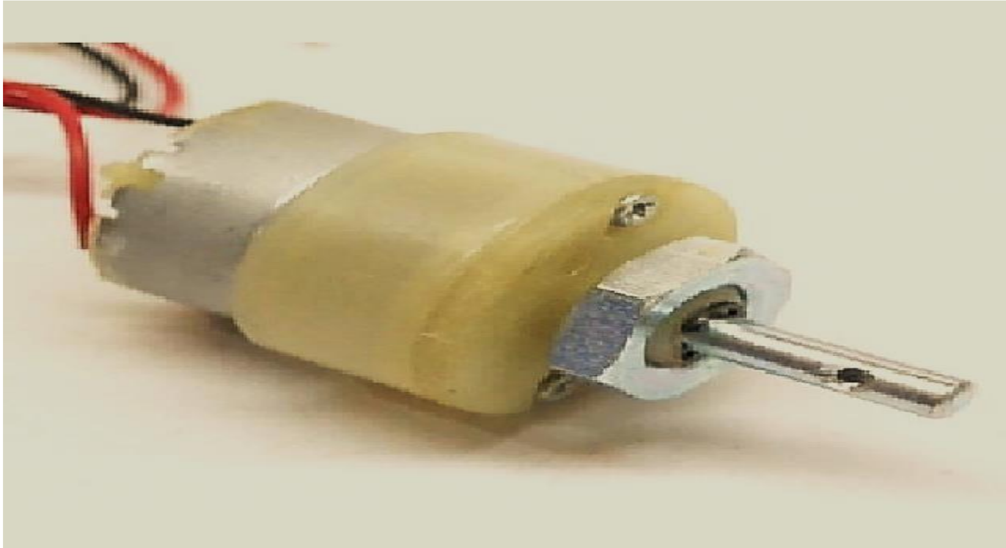


Figure 5.6: DC Stepper Motor

These motors are simple DC Motors featuring **Metal gears** for the shaft for obtaining the optimal performance characteristics. They are known as Center Shaft DC Geared Motors because their shaft extends through the center of their gearbox assembly.
This DC Motor – 10RPM – 12Volts can be used in all-terrain robots and a variety of robotic applications. These motors have a 3 mm threaded drill hole in the middle of the shaft thus making it simple to connect it to the wheels or any other mechanical assembly.

Features:

- The metal gears have better wear and tear properties.
- Gearbox is sealed and lubricated with lithium grease and requires no maintenance.
- Although motor gives 13.5 RPM at 12V, motor runs smoothly from 4V to 12V and gives the wide range of RPM, and torque.
- The shaft has a hole for better coupling.
- Operating Voltage(V): 12
- Rated Torque(kg-cm): 15.7
- Stall Torque(kg-cm): 52

5.4.2 30RPM 12v DC Motor with Metal Gears

30RPM 12V DC geared motors for robotics applications. Very easy to use and available in standard size. Nut and threads on shaft to easily connect and internal threaded shaft for easily connecting it to wheel.

Features:

- 30RPM 12V DC motors with Gearbox
- 3000RPM base motor
- 6mm shaft diameter with internal hole
- 125gm weight
- Same size motor available in various rpm
- 7kgcm torque
- No-load current = 60 mA(Max), Load current = 300 mA(Max)

5.5 Sensor Devices

5.5.1 IR Sensor

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation.

Figure 5.7: IR Sensor

5.5.2 Ultrasonic Sensor

Figure 5.8: Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

5.5.3 Colour Sensor TCS3200

The TCS3200 color sensor can detect a wide variety of colors based on their wavelength. This sensor is especially useful for color recognition projects such as color matching, color sorting, test strip reading and much more. The TCS3200 color sensor uses a TAOS TCS3200 RGB sensor chip to detect color. It also contains four white LEDs that light up the object in front of it.

Figure 5.9: TCS3200 Colour Sensor

The TCS3200 has an array of photodiodes with 4 different filters. A photodiode is simply a semiconductor device that converts light into current. The sensor has:

- 16 photodiodes with red filter – sensitive to red wavelength
- 16 photodiodes with green filter – sensitive to green wavelength
- 16 photodiodes with blue filter – sensitive to blue wavelength
- 16 photodiodes without filter

By selectively choosing the photodiode filter's readings, you're able to detect the intensity of the different colors. The sensor has a current-to-frequency converter that converts the photodiodes' readings into a square wave with a frequency that is proportional to the light intensity of the chosen color.

CHAPTER.6
Software Details

Chapter 6

Software Details

6.1 Arduino IDE Software

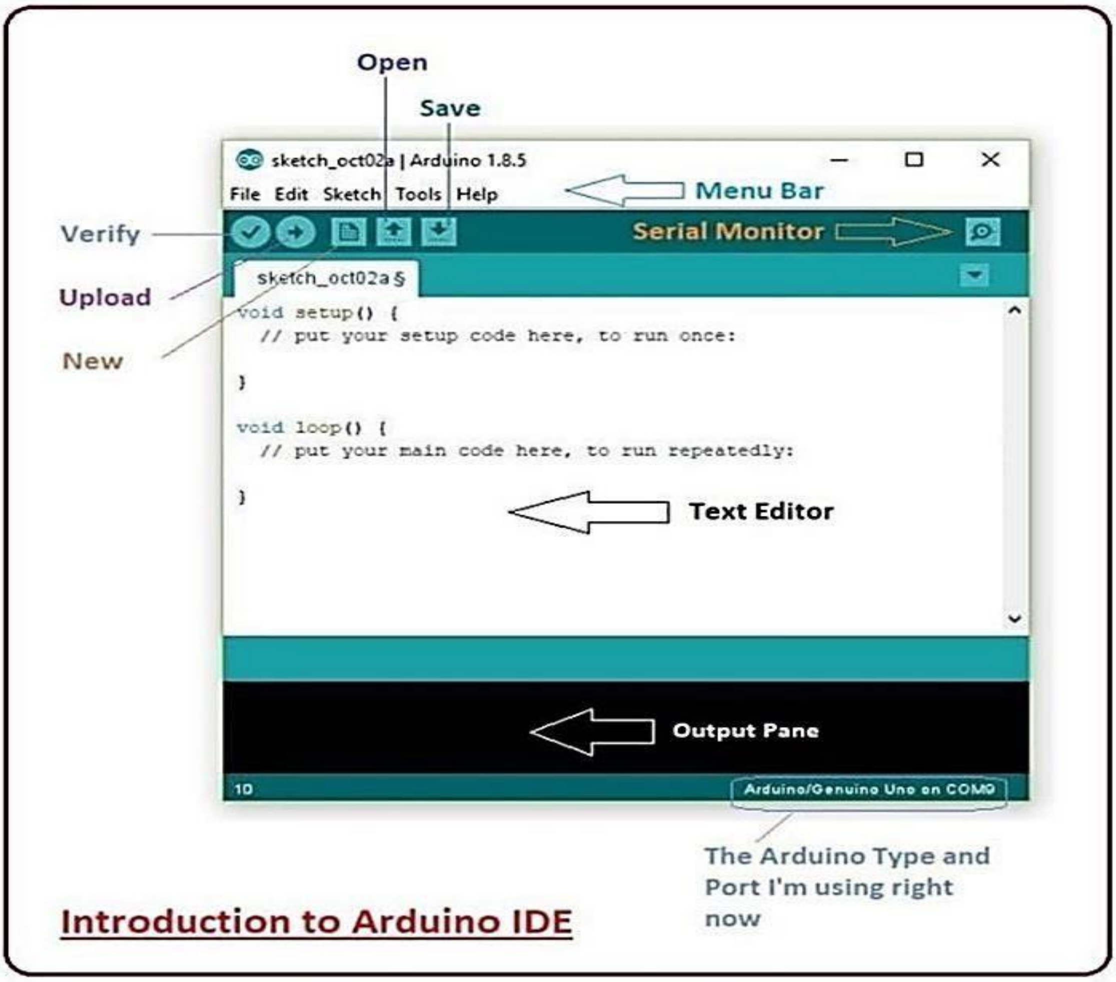


Figure 6.1: IDE Interface

An official software introduced by Arduino.cc, that is mainly used for writing, compiling, and uploading the code in almost all Arduino modules/boards. Arduino IDE is open-source software and is easily available to download & install.

- Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

- It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
- It is available for all operating systems i.e. MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code.
- A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
- Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
- The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
- The IDE environment mainly contains two basic parts: Editor and Compiler where the former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
- This environment supports both C and C++ languages.

- As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

- And at the end of the compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

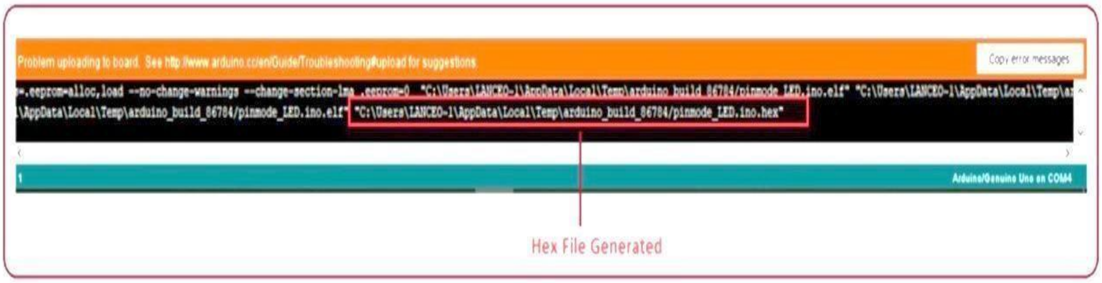


Figure 6.2: Compiler Interface

- Edit – Used for copying and pasting the code with further modification for font
- Sketch – For compiling and programming
- Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
- Help – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

The Six Buttons appearing under the Menu tab are connected with the running program as follows.

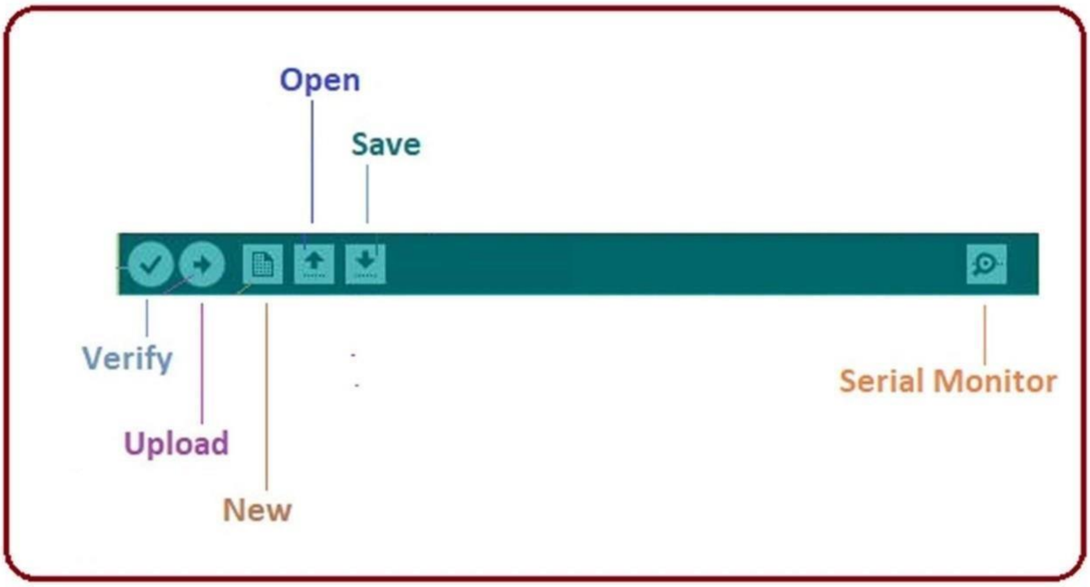


Figure 6.3: Menu Bar of IDE

- The checkmark appearing in the circular button is used to verify the code. Click this once you have written your code.
- The arrow key will upload and transfer the required code to the Arduino board.
- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a Serial Monitor.
- A separate pop-up window that acts as an independent terminal and plays a vital role in sending and receiving the Serial Data.
- You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly.
- The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating.
- Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
- You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.

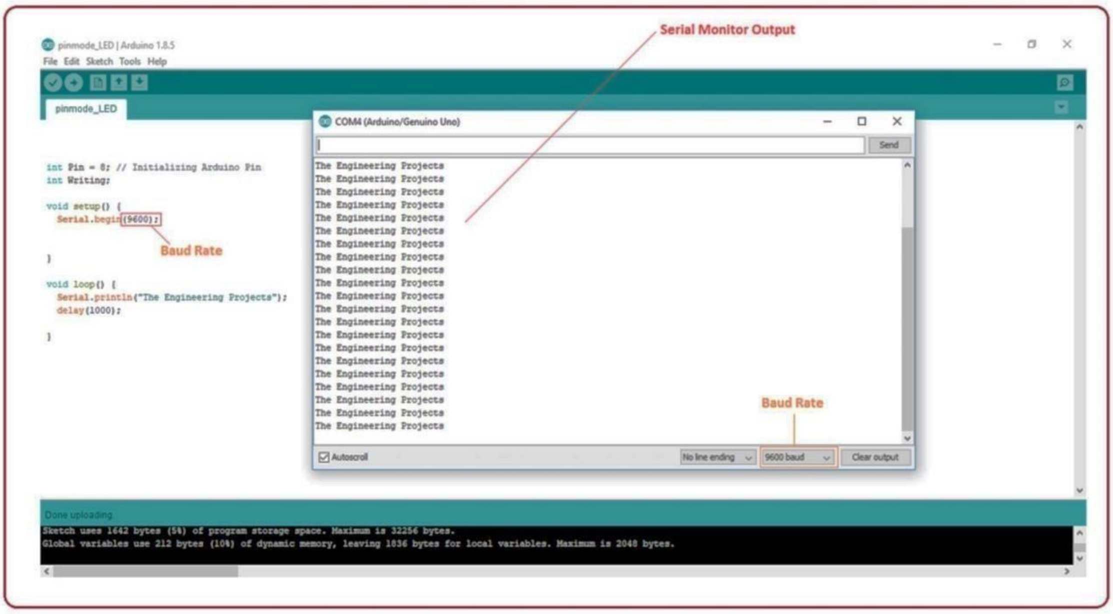


Figure 6.4: Serial Monitor Interface

CHAPTER.7
Proposed Design

Chapter 7

Proposed Design

In this chapter we will discuss about the designed prototype, how it works, its construction, mathematical formulation, sensing and actuation process step by step with flowcharts.

- **Body Construction:** The body of the robot is meticulously constructed using high-quality wood, selected for its durability, strength, and ability to withstand the demands of the robot's operation. The wood is carefully cut and shaped to form a robust base and frame, providing ample space for accommodating the electronic components, as well as ensuring stability and reliability during the robot's operation.
- **Lift System:** The lift system of the robot is designed with precision, incorporating a slider bar and gear and pinion mechanism. A 12V DC motor is utilized to power the lifting mechanism, which is connected to a gear system for efficient and reliable lifting capability. The slider bar enables smooth vertical movement of the lifting mechanism, allowing the robot to pick and drop objects with precision and accuracy.
- **Mobility:** The robot is equipped with four 12V DC motors, each with a speed of 30 RPM, for mobility. These motors are connected to a motor driver L293D, which is controlled by an Arduino Uno. The motor driver provides precise control over motor speed and direction, enabling the robot to move in various directions and navigate through obstacles with ease.
- **Color Recognition:** The robot is equipped with a TCS3200 color sensor for accurate color recognition. The color sensor is capable of detecting RGB colors and provides analog output signals. The Arduino Uno processes these signals to determine the color of the object placed in front of the sensor.
- **Path Following System:** The robot utilizes two infrared (IR) sensors positioned at the front of the bot for efficient path following. These IR sensors use infrared rays to detect the color of the surface and determine the path to follow. The IR sensors consist of a matched pair of infrared transmitter and receiver. The receiver measures the amount of reflected light and sends analog signals to the

Arduino Uno for processing. The robot is programmed to follow a specific path based on the color detected by the color sensor.

- **Color Recognition:** The robot initiates the color recognition process by using the TCS3200 color sensor to detect the color of the object in front of it. The sensor emits light on the object, and the reflected light is captured by the sensor. The sensor converts the RGB color information into analog signals, which are then processed by the Arduino Uno.

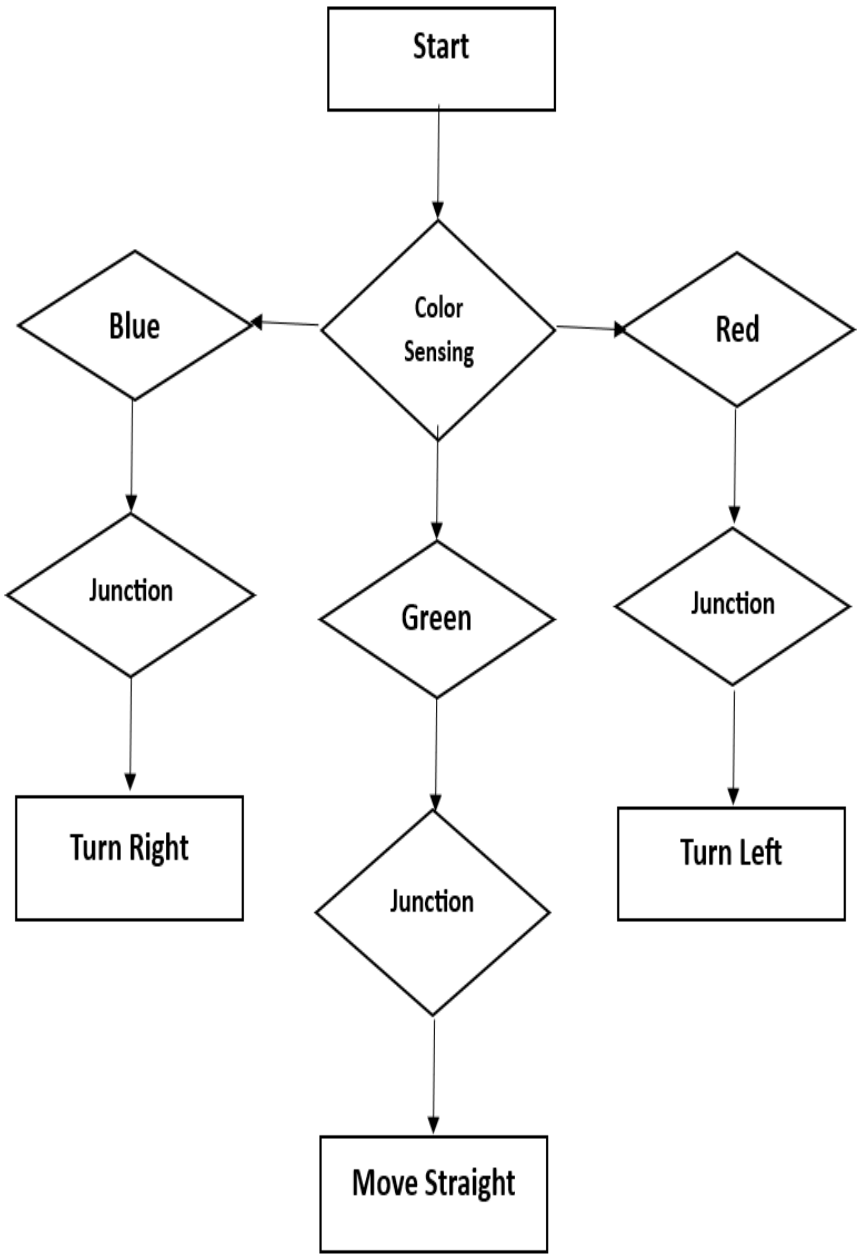


Figure 7.1: Flowchart for Color Sensing & Path Following

- **Object Lifting:** Once the color is recognized, the robot utilizes the lifting mechanism to pick up the object. The DC motor attached to the gear system drives the lifting mechanism, allowing it to slide up and securely grip the object. The robot can then move to the desired destination while safely holding the object with the lifting mechanism.

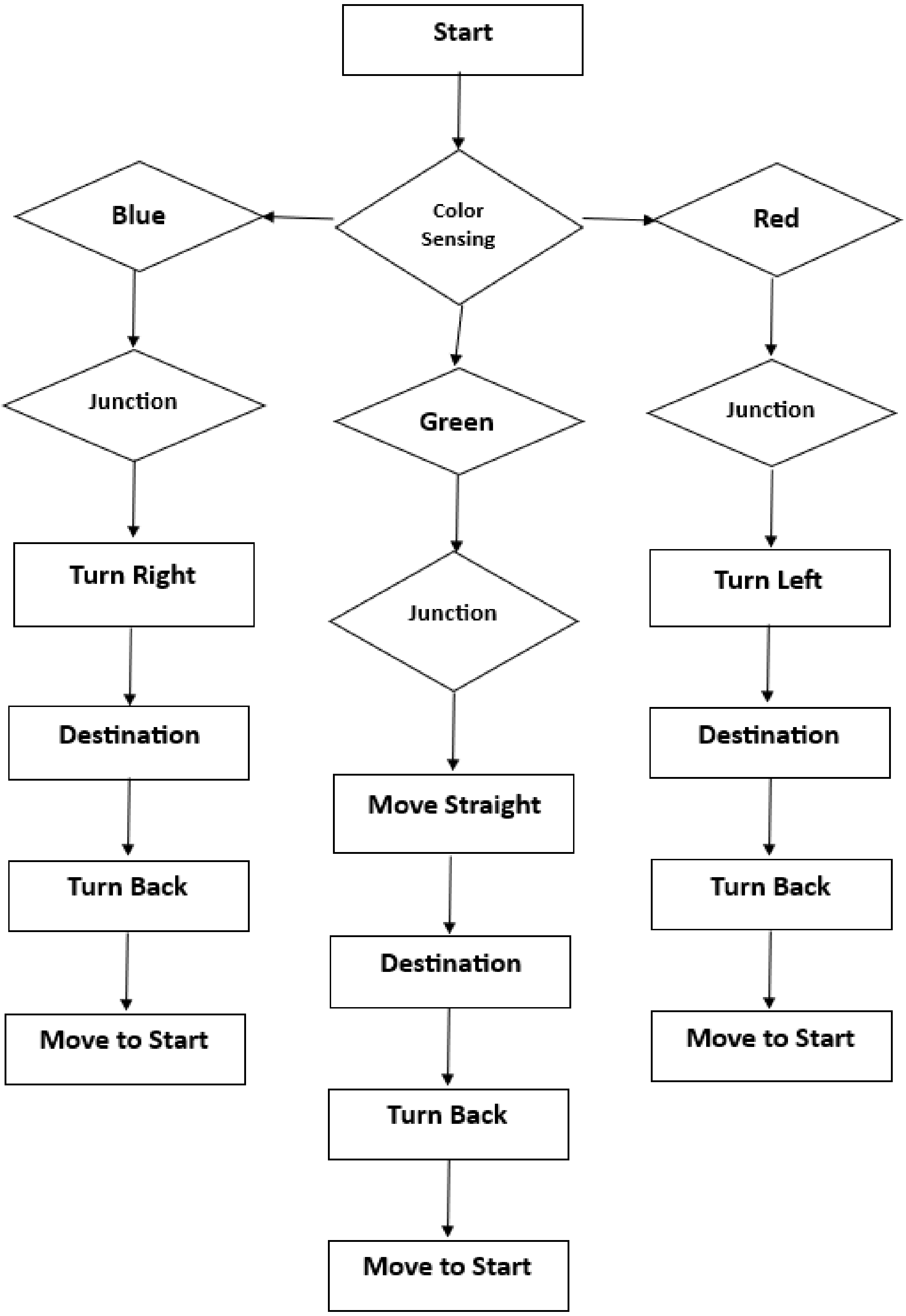


Figure 7.2: Flowchart for Robot Navigation

- **Path Following:** The robot employs the two IR sensors at the front to accurately follow the path. The IR sensors emit infrared rays on the surface, and the reflected rays are received by the sensors. The Arduino Uno processes the analog signals from the sensors and determines the color of the surface (white or black). Based on the color detected, the robot adjusts its motor speed and direction to correctly follow the designated path.
- **Object Dropping:** Once the robot reaches the designated destination based on the color, it uses the lifting mechanism to lower and drop the object. The lifting mechanism smoothly slides down, releasing the object at the intended location.
- **Mathematical Aspects:**

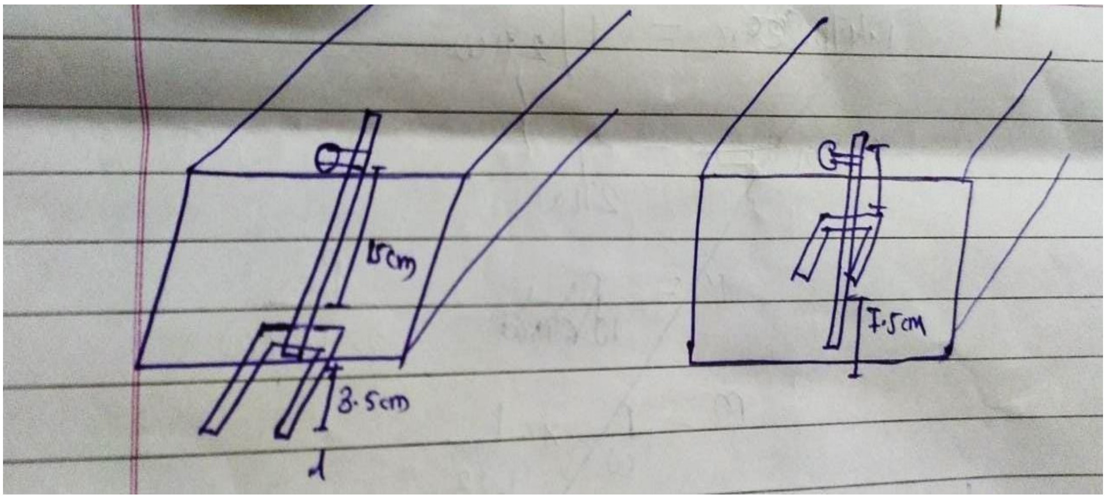


Figure 7.3: Diagram for Mathematical Calculation Of Lifting

Suppose an object of mass M to be picked.

The force acting on object is,

F= M x g

Where g is acceleration due to gravity

Therefore, F=9.8M -------- (1)

Force required to pull the fork upward suppose F` should be more than or equal to F to prevent fork from going downward.

$F` \geq 9.8M$ ---------- (2)

The applied torque T, should be

$T \geq r \times F$

Since r = 15cm = 0.15m

$T \geq 0.15 \times 9.8M$

$T \geq 1.47M$ ----------- (3)

The stall torque of motor is 0.52 Kg-m

Maximum torque that can be applied to the shaft and cause the motor stop rotating is known as stall torque.

Therefore, substituting T=0.52 Kg-m in equation 3 we get,

$0.52 \geq 1.47M$

$M \leq 0.353$ Kg

The designed robotic controlled model can lift an object of mass up to or less than 0.353kg.

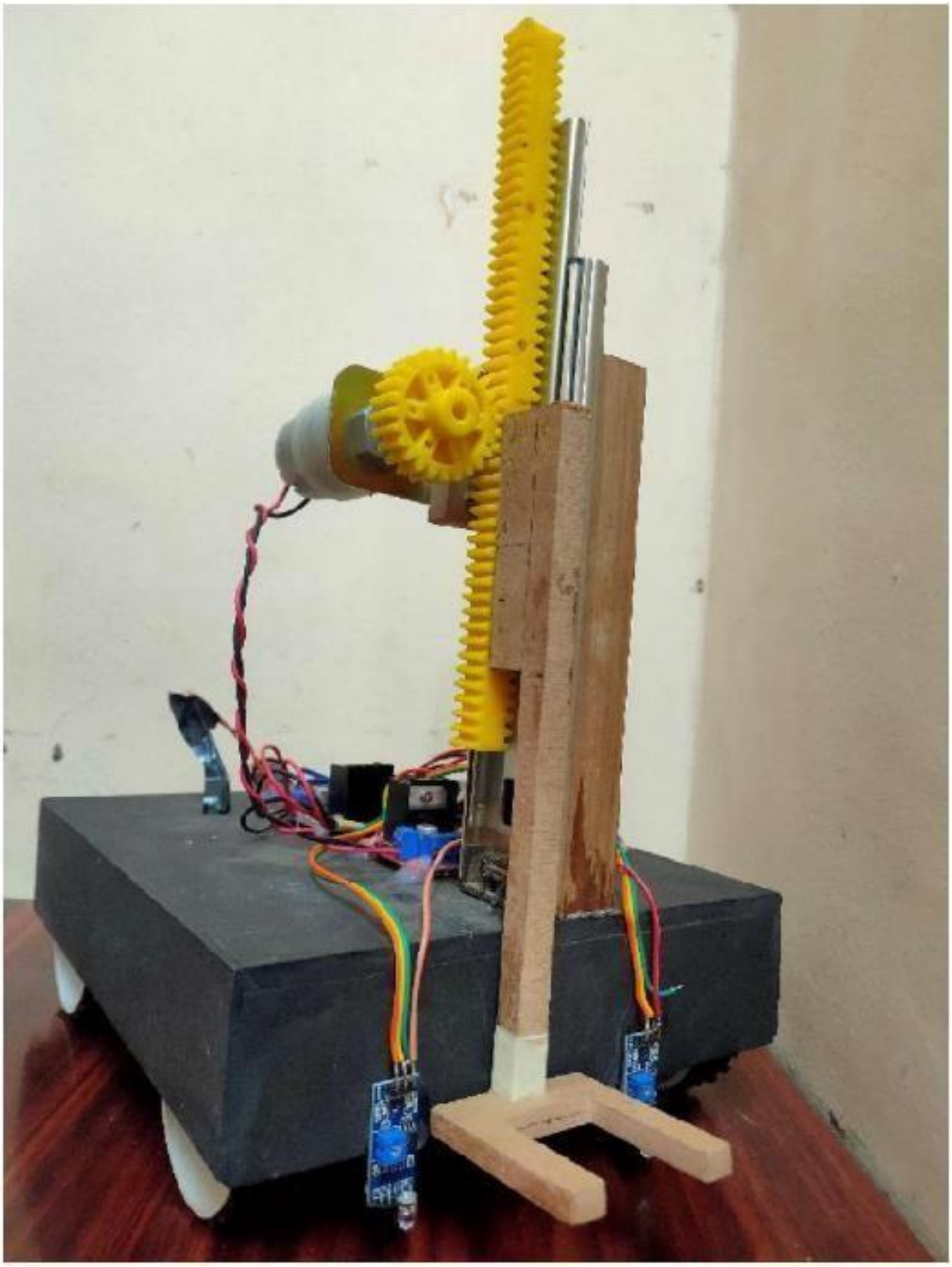


Figure 7.4: Prototype Design of Robot

CHAPTER.8
Future Scope & Conclusion

Chapter 8

Future Scope and Conclusion

In this chapter we will discuss about the improvements that can be made in created model, how it can be automated using AI algorithms, how it can be controlled via centralized approach etc in section below.

8.1 Future Scope

- **Full Automation with AI:** The warehouse robot can be made fully autonomous by incorporating advanced AI algorithms. The robot can use computer vision and image recognition technologies to identify and locate the boxes in the warehouse, and autonomously pick and transport them to their destinations. The robot can also learn from its interactions and experiences, continuously improving its efficiency and accuracy in box recognition, path planning, and navigation.
- **Real-time Order Management:** The warehouse robot can directly receive order information from a centralized database or an e-commerce platform. The robot can dynamically prioritize and optimize its tasks based on the order requirements, warehouse inventory, and delivery deadlines. This can enable real-time order management, reducing delays and improving customer satisfaction.
- **Path Planning and Navigation:** With the help of AI, the warehouse robot can intelligently plan and optimize its path within the warehouse without the need for external sensors like infrared (IR) sensors. The robot can analyse the layout of the warehouse, identify the shortest and safest routes, and navigate through the aisles, avoiding obstacles in real-time. This can result in improved efficiency, reduced collision risks, and optimized space utilization.
- **Advanced Sensing and Perception:** IoT sensors can be integrated into the robot to provide real-time data on the warehouse environment. This may include sensors for temperature, humidity, lighting, and even RFID sensors for tracking inventory. The robot can use this data to make informed decisions on box handling, storage, and retrieval, ensuring optimal conditions for the stored items.

- **Scalability and Flexibility:** The warehouse robot can be designed to be scalable and adaptable to changing warehouse layouts, inventory types, and order volumes. The robot can be easily reconfigured or expanded to accommodate new requirements or business growth. This can provide long-term sustainability and flexibility in meeting evolving warehousing needs.
- **Integration with Warehouse Management Systems:** The warehouse robot can be integrated with existing warehouse management systems (WMS) or enterprise resource planning (ERP) systems, creating a seamless flow of data and information. This can enable better coordination and synchronization of tasks, leading to improved overall warehouse operations and productivity.

8.2 Conclusion

In conclusion, we have successfully designed and developed a robot prototype for warehouse applications. The robot was designed to achieve the following objectives:

- Lift and navigate through the warehouse
- Make decisions on its own
- Reach its destination
- Return from the destination to the start point
- Avoid obstacles in its path

We achieved these objectives by using a variety of materials and components, including wood, 12V DC gear motors, IR sensors, TCS 3200 sensors, Arduino Uno, L293 and L293D motor drivers, rack and pinion, slider bar, and ultrasonic sensors.

To fulfil these objectives, we firstly designed the frame of the robot we choose the material as wood for it as it will be light weight and will not cost us much. For the lifting mechanism we make the lifting mechanism inspired from the forklift mechanism we designed it using the gear and a pinion and a slider bar the lifting mechanism was equipped with the 12v dc gear motor which will help the lift to move in up and down direction the motor of the lifting mechanism will be controlled by the Arduino nano and powered by the Li-ion battery. For mobility of the robot, we used 4 gear 12v dc motor with 30 rpm which will be controlled by the Arduino uno and powered by the Li-ion battery. For the navigation of the robot, we designed the map of our warehouse and to follow that map and navigate through the warehouse we use 2 IR sensors which will follow the map and the robot will be able to lift and navigate through the warehouse.

The second objective of our project is that it should be able to make decision at it own when needed for that we have used the Arduino uno which have ATmega 32 and with the help of the provided data from the sensors and the logic written in the code it is able to take own decision. While the robot is at starting position the colour is shown to it and then he has to choose the proper path to reach the destination while navigating it meets a junction from where the path get split in three ways each path for each colour at that junction the robot take the decision on its own on the base of data provided from the colour sensor tcs3200 and move forward on correct path.

The objective of reaching to the destination is fulfilled by the help of the overall system discussed above with the help of navigation system and colour sensing and making decision on its own it get fulfilled.

The objective to avoid the obstacles in the path is fulfilled with the help of ultrasonic sensor they detect the obstacle in the path and the system also have a buzzer which get activate when obstacle is detected in the path the system is operated by the Arduino nano and powered by Li-ion batteries

In conclusion, our warehouse robot project, leveraging IoT and mechanical systems, presents a practical and efficient solution for optimizing warehouse operations. With the integration of IoT sensors and a user-friendly interface, the robot can effectively handle box recognition, order management, and path planning, leading to improved efficiency and reduced manual labour in the warehouse. And cut down the time it takes to manage the warehouse and reduce the error that occurs while processing the orders from warehouse.

CHAPTER.9

Research Paper and Certificates

Design and Development of Smart Warehouse Robot Prototype

Dr.Kamlesh Khanchandani, Bhushan Nemade, Abhay Tiwari, Anushka Unhale, Namrata Kherde
Department of Electronics and Telecommunication, Shri Sant Gajanan Maharaj College of Engineering, Shegaon, India

Abstract:Efficient warehouse management depends heavily on accurate and reliable automation systems for inventory handling. This research paper presents a prototype of a smart warehouse robot designed to fulfil the objective of efficiently picking and placing boxes by utilising a colour-sensing system for identifying the correct placement in racks. The robot employs a lifting mechanism inspired by forklifts, consisting of a 12V DC motor, a rack and pinion, and a slider bar. The robot also features an obstacle avoidance system using an ultrasonic sensor and a buzzer, while navigation is done through two IR sensors. The colour-sensing system is done with the help of a TCS 3200 colour sensor and helps the robot to identify the colour codes on the boxes to determine their correct placement in the racks. The system is controlled by an Arduino Uno and uses five DC 12V motors, two L293D motor drivers, a TCS 3200 colour sensor, an ultrasonic sensor, a buzzer, and a rack and pinion with a slider bar for the lifting mechanism. The robot's performance was tested in a simulated warehouse environment, demonstrating accurate and efficient performance in picking and placing boxes. This research offers valuable insights into the development of smart warehouse robots, promoting sustainable warehouse management practices and enhancing overall efficiency.

Keywords:TCS Color Sensor,IR Sensor, 12V DC Motors, Arduino UNO, L293D Motor Driver.

I. INTRODUCTION

In recent years, the use of robots in various industries, including warehousing, has seen a significant increase. Warehouse robots are designed to automate processes, improve efficiency, and reduce labor costs. However, current warehouse robots have limitations, particularly in tasks such as lifting and transporting heavy boxes while avoiding obstacles. To address this issue, the primary objective of this research project is to design and develop a prototype warehouse robot with advanced sensing and navigation capabilities. The need for greater efficiency and productivity in the modern warehousing industry has made the development of a warehouse robot with advanced capabilities more important than ever. The use of robots not only reduces labor costs but also enhances workplace safety by automating hazardous tasks.

The proposed warehouse robot aims to fill the gap by incorporating obstacle avoidance technology using sensors to detect obstacles in its path and navigate around them. Additionally, the robot will be equipped with a lifting mechanism capable of lifting boxes of varying weights and sizes, making it versatile for use in different warehouse settings. To achieve the objectives outlined in the previous sections, a robotic prototype was designed and built using a variety of components and technologies. The robot was designed to be capable of lifting and transporting boxes while avoiding obstacles in its path, and it was equipped with a range of sensors and intelligent features to achieve this goal. The robot was built on a frame made from wood as it is a prototype and mounted on four high-torque 12VDC motors, which provided the power and mobility required to transport boxes across the warehouse floor.

The lifting mechanism consisted of a linear actuator that was capable of lifting boxes, the lift system is inspired by forklift. To ensure the robot could navigate around obstacles in its path, several sensors were integrated into the design. The robot used ultrasonic sensors mounted on the front of the chassis to detect obstacles in its path, and TCS3200 color sensors were used to identify the colors of boxes and determine their destinations. In addition, a microcontroller based on the Arduino Uno platform was used to control the robot's movements and coordinate its various functions. The robot was programmed using C, with code developed to control the motors, lift mechanism, and sensor inputs. The code was designed to be modular and scalable, allowing for easy modification and expansion as required.

II. COLOR SENSOR TCS3200

The TCS3200 color sensor can detect a wide variety of colors based on their wavelength. This sensor is especially useful for color recognition projects such as color matching, color sorting, test strip reading and much more. Here's the sensor specifications:

Power: 2.7V to 5.5V
Size: 28.4 x 28.4mm (1.12 x 1.12″) Interface:
digital TTL.
High-resolution conversion of light intensity to frequency
Programmable color and full-scale output frequency
Communicates directly to microcontroller

Construction:

The TCS3200 color sensor typically comes in a small rectangular package with four photodiodes arranged in a 2x2 matrix, along with color filters covering each photodiode. The color filters are designed to allow specific wavelengths of light to pass through, corresponding to red, green, blue, and clear (no filter) colors. The sensor package also includes pins for power supply, ground, and frequency output. The TCS3200 color sensor can be easily interfaced with microcontrollers or other electronic devices for color sensing applications. It requires an external light source, such as a white LED, to illuminate the object being sensed. The frequency output from the sensor can be connected to a microcontroller input pin, which can then be used to calculate the RGB components of the detected color based on the frequency of the output signal.

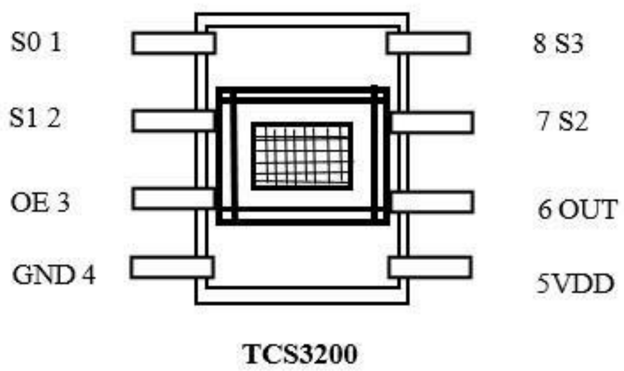


Figure 1: Circuit Diagram of TCS3200 Color Sensor

Pin Name	I/O	Description
GND (4)		Power supply ground
OE (3)	I	Enable for o/p frequency
OUT (6)	O	Output frequency
S0, S1 (1,2)	I	O/P frequency scaling selection inputs
S2,S3 (7,8)	I	Photodiode type selection inputs

VDD (5)	Voltage Supply	VDD (5)

Filter selection: The photodiodes are connected in parallel and can be selected by setting the control pins S2 and S3 to different combinations of LOW and HIGH states. Refer to the table below for the different color selections available.

Photodiode Type	S2	S3
RED	LOW	LOW
BLUE	LOW	HIGH
GREEN	HIGH	HIGH
No Filter (Clear)	HIGH	LOW

The Working Principle of TCS3200 Color Sensor: The TCS3200 color sensor is based on the principle of color filtering and photodiode array. It consists of a grid of color-sensitive filters, also known as the Bayer filter, and an array of photodiodes underneath. The color sensor module typically includes a high-intensity white LED that projects modulated light onto the object being sensed. The light reflected from the object passes through the Bayer filter, which consists of four filters - red, blue, green, and clear (no filter) - arranged in a pattern known as the Bayer pattern. The TCS3200 color sensor operates by detecting the intensity of light at different wavelengths. The photodiodes in the array are designed to detect specific colors of light depending on the filter they are placed under. When light passes through the filters, it is absorbed by the photodiodes, generating an electrical current proportional to the intensity of the light. The intensity of light is measured by the frequency of the square wave output generated by the sensor. Frequency Measurement

The TCS3200 color sensor has an internal current-to-frequency converter that converts the output from the photodiodes into a square wave whose frequency is proportional to the intensity of the chosen color. The output frequency of the sensor typically ranges from 2 Hz to 500 kHz, depending on the intensity of the detected color. The frequency of the square wave output is determined by the intensity of light falling on the photodiodes. Higher intensity of light results in a higher frequency output, while lower intensity of light results in a lower frequency output. This frequency measurement allows the sensor to quantify the intensity of each color component (red, green, and blue) in the reflected light from the object being sensed. Color Recognition: The color of the object being sensed is determined by measuring the relative level of red, green, and blue light using the frequency output of the TCS3200 color sensor.

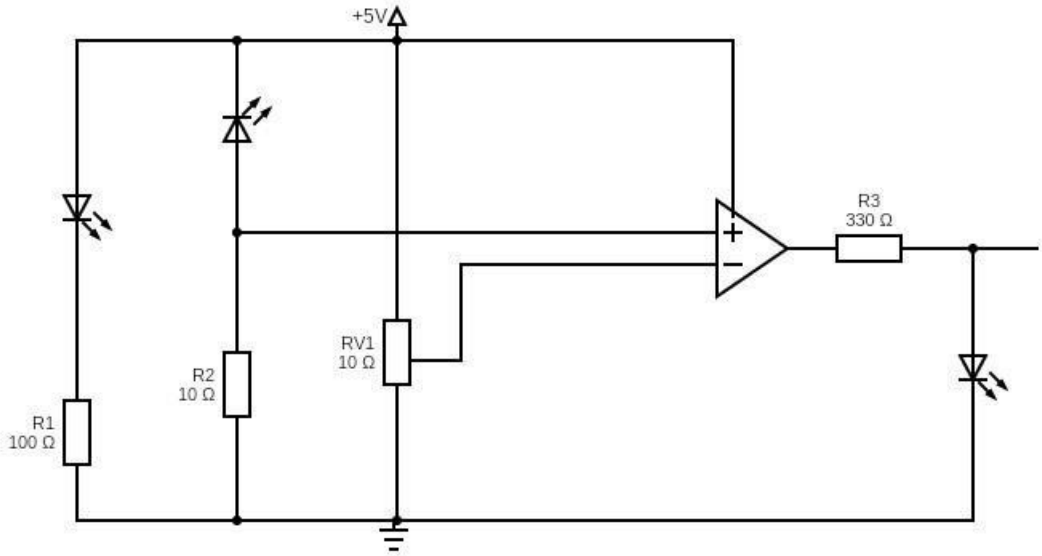
The frequency of each color component is compared to a reference value or a pre-defined threshold to determine the color of the object. The TCS3200 color sensor module provides control pins (S2 and S3) that allow selection of the color array to be read by enabling specific photodiodes. By selectively enabling the photodiodes under the red, blue, and green filters, the sensor can measure the intensity of each color component separately. The intensity of each color component is then converted into a frequency by the internal current-to-frequency converter, and the frequency output is used to determine the color of the object based on the pre-defined thresholds or reference values.

Conclusion: In conclusion, the TCS3200 color sensor works based on the principle of color filtering and photodiode array. It uses a grid of color-sensitive filters and an array of photodiodes to detect the intensity of light at different wavelengths. The internal current-to-frequency converter converts the output from the photodiodes into a square wave whose frequency is proportional to the intensity of the detected color. The frequency output is used to determine the color of the object being sensed based on pre-defined thresholds or reference values. The TCS3200 color sensor is widely used in various applications, such as color detection, color sorting, and color-based automation systems. Further research and development of the TCS3200 color sensor can lead to improved color sensing technologies and applications in fields such as robotics, industrial automation, and smart devices.

II. IR SENSOR

These sensors consist of a matched pair of an IR transmitter and an IR receiver. When the IR transmitter emits infrared rays onto the surface, the receiver measures the amount of light that is reflected back. White surfaces generally reflect a significant amount of light, while black surfaces reflect very little. By analyzing the reflected IR rays, the robot can determine whether it is on a white or black surface. To ensure accurate detection, the IR reflectance sensors need to be shielded from ambient light, and the distance between the sensors and the reflective surface should be small, ideally between 2 to 10 mm. In the robot design, eight sensors are used, and their positioning in relation to each other is critical. The distance between sensors needs to be adjusted depending on the line width of the path being followed. If the line width is thin, the distance between sensors must be reduced to ensure timely detection of curves in the line path. The analog signals from the IR sensors need to be converted to digital form for processing by the robot's microprocessor.

This can be achieved by using an external Analog-to-Digital Converter (ADC). In the design, the LM324 ADC is used, with two LM324 chips supporting eight sensors. The resistance of the receiver sensor decreases when exposed to IR radiation, and a good sensor will have near-zero resistance in the presence of the rays, and a high resistance in their absence. This property can be utilized to create a potential divider circuit, which allows for the conversion of the analog signals from the sensors to digital form for further processing by the robot's processor. In summary, the robot design incorporates IR reflectance sensors that emit and detect IR rays to follow a line path. The sensors are positioned close to the reflective surface and shielded from ambient light. The analog signals from the sensors are converted to digital form using an external ADC, and the resistance change of the sensors due to IR radiation is utilized in a potential divider circuit. This allows the robot to accurately detect the line path and navigate along it.



The IR sensor used in the line following robot consists of a matched pair of an IR transmitter and an IR receiver. These components are typically housed in a compact module. The IR transmitter emits infrared rays onto the surface, while the IR receiver measures the amount of light that is reflected. The module is designed to have a small distance between the transmitter and receiver, typically around 2 to 10 mm, to ensure accurate detection of the reflected rays.

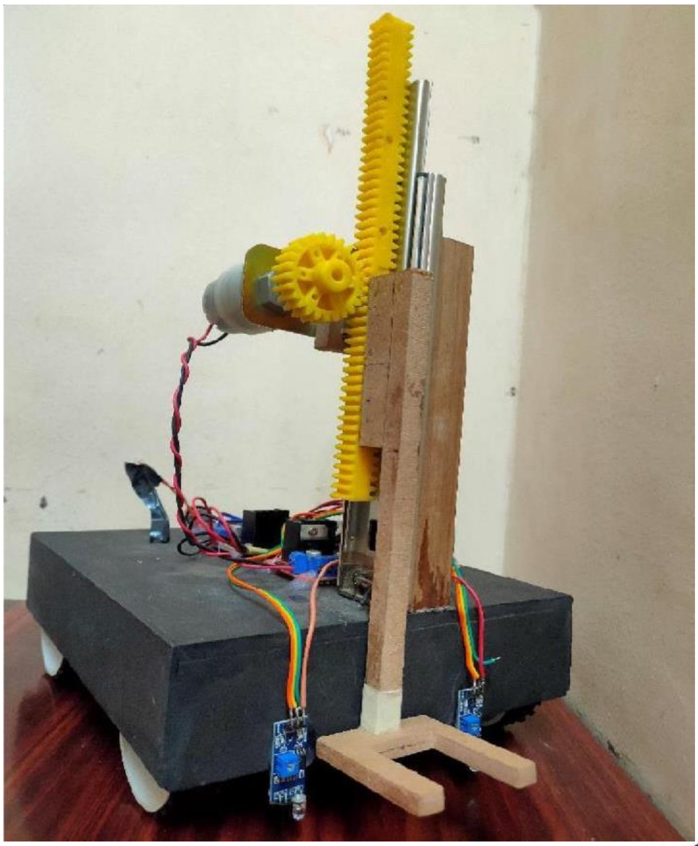
The IR sensor operates based on the principle of reflectance. When the IR transmitter emits infrared rays onto a surface, the number of reflected rays depends on the color and reflectivity of the surface. White surfaces generally reflect a significant amount of light, while black surfaces reflect very little. The IR receiver measures the intensity of the reflected rays and generates an analog signal that represents the detected light intensity. To convert the analog signals from the IR receiver to digital form for processing by the robot's microprocessor, an external ADC is used. In the design, the LM324 ADC is utilized, with two LM324 chips supporting eight sensors. The resistance of the IR receiver sensor changes with the intensity of the IR radiation it receives. A good sensor will have near-zero resistance in the presence of the rays, and a high resistance in their absence.

This property of the IR receiver can be utilized in a potential divider circuit. The analog signal from the IR receiver is connected to the input of the LM324 ADC, and the sensor's resistance is connected in series with a fixed resistor to create a voltage divider. The voltage across the IR receiver sensor is measured at the junction of the two resistors and is converted to a digital value by the ADC. This digital value represents the detected light intensity and can be processed by the robot's microprocessor to determine the position of the line path. The positioning of the IR sensors on the robot is critical for accurate line following. The sensors need to be positioned close to the reflective surface and shielded from ambient light to minimize interference. The distance between sensors may need to be adjusted depending on the line width of the path being followed. If the line width is thin, the distance between sensors needs to be reduced to ensure timely detection of curves in the line path. In summary, the IR sensor in the line following robot uses an IR transmitter and an IR receiver to emit and detect infrared rays for line detection. The analog signals from the IR receiver are converted to digital form using an external ADC, and the resistance change of the IR receiver due to IR radiation is utilized in a potential divider circuit. This allows the robot to accurately detect the line path and navigate along it. Proper positioning and shielding of the sensors are crucial for reliable operation.

III. EXPLANATION OF THE DESIGN AND IMPLEMENTATION PROCESS OF THE SMART WARE HOUSE ROBOT

Body Construction:

The body of the robot is meticulously constructed using high-quality wood, selected for its durability, strength, and ability to withstand the demands of the robot's operation. The wood is carefully cut and shaped to form a robust base and frame, providing ample space for accommodating the electronic components, as well as ensuring stability and reliability during the robot's operation.



ISSN 2581-9429 IJARSCT

Lift System:

The lift system of the robot is designed with precision, incorporating a slider bar and gear and pinion mechanism. A 12V DC motor is utilized to power the lifting mechanism, which is connected to a gear system for efficient and reliable lifting capability. The slider bar enables smooth vertical movement of the lifting mechanism, allowing the robot to pick and drop objects with precision and accuracy.

Mobility:

The robot is equipped with four 12V DC motors, each with a speed of 30 RPM, for mobility. These motors are connected to a motor driver L293D, which is controlled by an Arduino Uno. The motor driver provides precise control over motor speed and direction, enabling the robot to move in various directions and navigate through obstacles with ease.

Color Recognition:

The robot is equipped with a TCS3200 color sensor for accurate color recognition. The color sensor is capable of detecting RGB colors and provides analog output signals. The Arduino Uno processes these signals to determine the color of the object placed in front of the sensor.

Path Following System:

The robot utilizes two infrared (IR) sensors positioned at the front of the bot for efficient path following. These IR sensors use infrared rays to detect the color of the surface and determine the path to follow. The IR sensors consist of a matched pair of infrared transmitter and receiver. The receiver measures the amount of reflected light and sends analog signals to the Arduino Uno for processing. The robot is programmed to follow a specific path based on the color detected by the color sensor.

Working:

Color Recognition:

The robot initiates the color recognition process by using the TCS3200 color sensor to detect the color of the object in front of it. The sensor emits light on the object, and the reflected light is captured by the sensor. The sensor converts the RGB color information into analog signals, which are then processed by the Arduino Uno.

Object Lifting:

Once the color is recognized, the robot utilizes the lifting mechanism to pick up the object. The DC motor attached to the gear system drives the lifting mechanism, allowing it to slide up and securely grip the object. The robot can then move to the desired destination while safely holding the object with the lifting mechanism.

Path Following:

The robot employs the two IR sensors at the front to accurately follow the path. The IR sensors emit infrared rays on the surface, and the reflected rays are received by the sensors. The Arduino Uno processes the analog signals from the sensors and determines the color of the surface (white or black). Based on the color detected, the robot adjusts its motor speed and direction to correctly follow the designated path.

Object Dropping:

Once the robot reaches the designated destination based on the color, it uses the lifting mechanism to lower and drop the object. The lifting mechanism smoothly slides down, releasing the object at the intended location.

Conclusion:

The wood-based robot with color recognition and path following system is designed and implemented using Arduino Uno, DC motors, motor driver L293D, TCS3200 color sensor, and IR sensors. The robot showcases efficient color recognition and path following capabilities, making it suitable for various applications such as automation, logistics, and transportation. The meticulous construction of the wooden body, precise lift system, and reliable mobility mechanism allow the robot to accurately detect colors, pick and drop objects, and follow designated paths, making it a promising solution for diverse practical applications.

IV. OVERVIEW OF IOT AND MECHANICAL SYSTEMS

The Internet of Things (IoT) is a system of physical devices, vehicles, buildings, and other objects that are equipped with sensors, software, and network connectivity, allowing them to gather and exchange data. This technology is built upon the idea of linking ordinary objects to the internet, allowing them to communicate and exchange information with each other and with humans with proper interface.

When combined, IoT and mechanical systems can create powerful and intelligent systems that can automate processes, optimize operations, and improve efficiency in various domains. For example, IoT sensors can collect data on environmental conditions, machine performance, or product quality, which can then be processed and analyzed by the mechanical system to make informed decisions and take appropriate actions. The integration of IoT and mechanical systems can bring numerous benefits. Real-time data collection from IoT sensors allows for better monitoring, control, and optimization of mechanical systems, leading to increased efficiency, reduced downtime, and improved safety. Additionally, the ability of IoT devices to communicate with each other and with humans enables remote control, monitoring, and management of mechanical systems, allowing for more flexible and convenient operations. In various industries, the combination of IoT and mechanical systems has the potential to revolutionize processes and operations. For example, in manufacturing, IoT sensors can collect data on machine performance, predictive maintenance needs, and supply chain optimization, which can then be used by the mechanical system to optimize production processes and reduce downtime. In smart cities, IoT sensors can collect data on traffic patterns, environmental conditions, and energy usage, which can be used by the mechanical system to optimize transportation systems, manage resources, and improve sustainability.

The project revolves around the development of a warehouse robot that incorporates IoT sensors and a mechanical system. The IoT sensors are advanced sensors that are capable of collecting various types of data, such as motion, temperature, humidity, and pressure, and transmitting this data to other devices or the cloud for processing and analysis. These sensors are integrated into the robot to enable it to gather real-time data on the environment and its surroundings. The mechanical system of the warehouse robot includes mechanisms, actuators, and components that allow it to navigate, move, and perform tasks within the warehouse. The mechanical system is designed to work in tandem with the IoT sensors, allowing the robot to interpret the data collected by the sensors and make informed decisions based on the environment it operates in. The IoT sensors and mechanical system work collaboratively to optimize the warehouse robot's operations. For example, the sensors can provide real-time data on the location and status of inventory, environmental conditions, and obstacles in the robot's path.

The mechanical system can then use this data to adjust the robot's movements, optimize its path, and perform tasks efficiently. The warehouse robot project aims to demonstrate the potential of IoT and smart sensor technologies in combination with a mechanical system for warehouse automation. The use of these technologies can enable more efficient and streamlined warehouse operations, such as inventory management, order picking, and item tracking. The project also seeks to showcase the benefits of using IoT and smart sensors for real-time data collection, analysis, and decision-making in a warehouse setting. Overall, the warehouse robot project aims to leverage the power of IoT sensors and a mechanical system to improve the efficiency, accuracy, and sustainability of warehouse operations. By integrating cutting-edge technologies, the project seeks to provide a practical solution for optimizing warehouse management and automation in the modern era.

V. FUTURE SCOPE

1. Full Automation with AI: The warehouse robot can be made fully autonomous by incorporating advanced AI algorithms. The robot can use computer vision and image recognition technologies to identify and locate the boxes in the warehouse, and autonomously pick and transport them to their destinations. The robot can also learn from its

interactions and experiences, continuously improving its efficiency and accuracy in box recognition, path planning, and navigation.

2. Real-time Order Management: The warehouse robot can directly receive order information from a centralized database or an e-commerce platform. The robot can dynamically prioritize and optimize its tasks based on the order requirements, warehouse inventory, and delivery deadlines. This can enable real-time order management, reducing delays and improving customer satisfaction.
3. Path Planning and Navigation: With the help of AI, the warehouse robot can intelligently plan and optimize its path within the warehouse without the need for external sensors like infrared (IR) sensors. The robot can analyze the layout of the warehouse, identify the shortest and safest routes, and navigate through the aisles, avoiding obstacles in real-time. This can result in improved efficiency, reduced collision risks, and optimized space utilization.
4. Advanced Sensing and Perception: IoT sensors can be integrated into the robot to provide real-time data on the warehouse environment. This may include sensors for temperature, humidity, lighting, and even RFID sensors for tracking inventory. The robot can use this data to make informed decisions on box handling, storage, and retrieval, ensuring optimal conditions for the stored items.
5. Scalability and Flexibility: The warehouse robot can be designed to be scalable and adaptable to changing warehouse layouts, inventory types, and order volumes. The robot can be easily reconfigured or expanded to accommodate new requirements or business growth. This can provide long-term sustainability and flexibility in meeting evolving warehousing needs.
6. Integration with Warehouse Management Systems: The warehouse robot can be integrated with existing warehouse management systems (WMS) or enterprise resource planning (ERP) systems, creating a seamless flow of data and information. This can enable better coordination and synchronization of tasks, leading to improved overall warehouse operations and productivity.

In conclusion, the future scope of your warehouse robot project integrating IoT and mechanical systems is promising, with potential advancements in AI, automation, and sensing technologies. This can result in a fully autonomous, intelligent, and efficient warehouse robot that can recognize boxes, receive orders, plan paths, and navigate the warehouse, ultimately optimizing warehouse operations and enhancing customer satisfaction.

VI. CONCLUSION

In conclusion, our warehouse robot project, leveraging IoT and mechanical systems, presents a practical and efficient solution for optimizing warehouse operations. With the integration of IoT sensors and a user-friendly interface, the robot can effectively handle box recognition, order management, and path planning, leading to improved efficiency and reduced manual labor in the warehouse. And cut down the time it takes to manage the warehouse and reduce the error that occurs while processing the orders from warehouse.

REFERENCES

[1] Yihong Wu, Fulin Tang, Heping Li, “Image Based Camera Localization: an Overview” University of Chinese Academy of Sciences [2018].

[2] Karthik Karur, Nitin Sharma, Chinmay Dharmatti and Joshua E. Siegel, “A Survey of Path Planning Algorithms for Mobile Robots”, Halla Mechatronics, 3933 Monitor Rd, Bay City, MI 48706, USA [2021].

[3] Marcia M. Costa, “A Survey on Path Planning Algorithms for Mobile Robots”, INESC TEC and ISEP-IPP Porto, Portugal [2020].

[4] Suman Harapanahalli, Niall O Mahony, Gustavo Velasco Hernandez, Sean Campbell, Daniel Riordan, Joseph Walsh, “Autonomous Navigation of mobile robots in factory environment,” 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), [June 2019]

[5] Faiza Gul, Wan Rahiman & Syed Sahal Nazli Alhady, “A comprehensive study for robot navigation techniques”, Gul et al., Cogent Engineering [2019]

[6] Yuncheng Lu, Zhucun Xue, Gui-Song Xia & Liangpei Zhang, "A survey on vision-based UAV navigation", Geospatial Information Science, 2018 VOL. 21, NO. 1, 21–32.

[7] Rhaian J. F. Barros, Jorge L. P. Silva Filho, Joao V. S. Neto, and Tiago P. Nascimento, "An Open-Design Warehouse Mobile Robot", Lab of Systems Engineering and Robotics (LASER), Department of Computer Systems, Universidade Federal da Paraiba, Brazil.

[8] Matthew Gadd, Paul Newman, "A Framework for Infrastructure-Free Warehouse Navigation", IEEE International Conference on Robotics and Automation (ICRA), 2015.

[9] Jolly Shah, S.S.Rattan, B.C.Nakra, "End-Effector Position Analysis Using Forward Kinematics for 5DOF Pravak Robot Arm" ,International Journal of Robotics and Automation (IJRA),Vol. 2, No. 3, September 2013.

[10] Dr. Dhiraj Jain, Mr.Yuvraj Sharma, "Adoption of next generation robotics: A case study on Amazon", PERSPECTIVA A Case Research Journal Volume III (2017).

[11] Amandeep Dhaliwal, Manav Rachna, "Rise of Automation and Robotics in Warehouse Management", International Institute of Research and Studies.

[12] Ismot Sadik Peyas, Zahid Hasan, Md. Rafat Rahman Tushar, Al Musabbir, "Autonomous Warehouse Robot using Deep Q-Learning".

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

CERTIFICATE OF PUBLICATION

INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429

THIS IS TO CERTIFY THAT

Bhushan Nemade

Shri Sant Gajanan Maharaj College of Engineering, Shegaon, India

HAS PUBLISHED A RESEARCH PAPER ENTITLED

Design and Development of Smart Warehouse Robot Prototype

IN IJARSCT, VOLUME 3, ISSUE 4, MAY 2023

Certificate No: 052023-A997
www.ijarsct.co.in

Crossref
DOI: 10.48175/IJARSCT-9865
www.doi.org
www.crossref.org

7.301 Journal Impact Factor
www.sjifactor.com

Editor-in-Chief

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

CERTIFICATE OF PUBLICATION

INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429

THIS IS TO CERTIFY THAT

Namrata Kherde

Shri Sant Gajanan Maharaj College of Engineering, Shegaon, India

HAS PUBLISHED A RESEARCH PAPER ENTITLED

Design and Development of Smart Warehouse Robot Prototype

IN IJARSCT, VOLUME 3, ISSUE 4, MAY 2023

Certificate No: 052023-A1000
www.ijarsct.co.in

Crossref
DOI: 10.48175/IJARSCT-9865
www.doi.org
www.crossref.org

7.301 Journal Impact Factor
www.sjifactor.com

Editor-in-Chief

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

CERTIFICATE OF PUBLICATION

INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429

THIS IS TO CERTIFY THAT

Abhay Tiwari

Shri Sant Gajanan Maharaj College of Engineering, Shegaon, India

HAS PUBLISHED A RESEARCH PAPER ENTITLED

Design and Development of Smart Warehouse Robot Prototype

IN IJARSCT, VOLUME 3, ISSUE 4, MAY 2023

Certificate No: 052023-A998
www.ijarsct.co.in

Crossref
DOI: 10.48175/IJARSCT-9865
www.doi.org
www.crossref.org

7.301
Scientific Journal Impact Factor
Journal Impact Factor
www.sjifactor.com

Editor-in-Chief

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

CERTIFICATE OF PUBLICATION

INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429

THIS IS TO CERTIFY THAT

Anushka Unhale

Shri Sant Gajanan Maharaj College of Engineering, Shegaon, India

HAS PUBLISHED A RESEARCH PAPER ENTITLED

Design and Development of Smart Warehouse Robot Prototype

IN IJARSCT, VOLUME 3, ISSUE 4, MAY 2023

Certificate No: 052023-A999
www.ijarsct.co.in

Crossref
DOI: 10.48175/IJARSCT-9865
www.doi.org
www.crossref.org

Scientific Journal Impact Factor 7.301 Journal Impact Factor
www.sjifactor.com

Editor-in-Chief

References

References

[1] Yihong Wu, Fulin Tang, Heping Li, *"Image Based Camera Localization: An Overview"* University of Chinese Academy of Sciences [2018]

[2] Karthik Karur, Nitin Sharma, Chinmay Dharmatti and Joshua E. Siegel, *"A Survey of Path Planning Algorithms for Mobile Robots"*, Halla Mechatronics, 3933 Monitor Rd, Bay City, MI 48706, USA [2021]

[3] Marcia M. Costa, "*A Survey on Path Planning Algorithms for Mobile Robots*", INESC TEC and ISEP-IPP Porto, Portugal [2020]

[4] Suman Harapanahalli, Niall O Mahony, Gustavo Velasco Hernandez, Sean Campbell, Daniel Riordan, Joseph Walsh, "*Autonomous Navigation of mobile robots in factory environment,"* 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), [June 2019]

[5] Faiza Gul, Wan Rahiman & Syed Sahal Nazli Alhady, *"A comprehensive study for robot navigation techniques",* Gul et al., Cogent Engineering [2019]

[6] Yuncheng Lu, Zhucun Xue, Gui-Song Xia & Liangpei Zhang, *"A survey on vision-based UAV navigation",* Geo-spatial Information Science, 2018

[7] Rhaian J. F. Barros, Jorge L. P. Silva Filho, Joao V. S. Neto, and Tiago P. Nascimento, *"An Open-Design Warehouse Mobile Robot"*, Lab of Systems Engineering and Robotics (LASER), Department of Computer Systems, Universidade Federal da Paraiba, Brazil

[8] Matthew Gadd, Paul Newman, *"A Framework for Infrastructure-Free Warehouse Navigation", IEEE* International Conference on Robotics and Automation (ICRA), 2015

[9] Jolly Shah, S.S.Rattan, B.C.Nakra, *"End-Effector Position Analysis Using Forward Kinematics for 5DOF Pravak Robot Arm" ,*International Journal of Robotics and Automation (IJRA),Vol. 2, No. 3, September 2013.

[10] Dr. Dhiraj Jain, Mr.Yuvraj Sharma, "*Adoption of next generation robotics: A case study on Amazon*", Perspectiva A Case Research Journal Volume III (2017)

[11] Amandeep Dhaliwal, Manav Rachna,*"Rise of Automation and Robotics in Warehouse Management",* International Institute of Research and Studies

[12] Ismot Sadik Peyas, Zahid Hasan, Md. Rafat Rahman Tushar, Al Musabbir, *"Autonomous Warehouse Robot using Deep Q-Learning"*