****

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**HIMALAYA COLLEGE OF ENGINEERING**

A

MAJOR PROJECT PROPOSAL

ON

**P.A.R.R.O.T  
Parallel Asynchronous Robots, Robustly Organizing Trucks**

BY:

NABIN KHADKA(HCE076BEI005)

NISHAN GNAWALI(HCE076BEI008)

PRASHANT BASHYAL(HCE076BEI009)

RAMU ACHARYA(HCE076BEI010)

A MAJOROR PROJECT MID-TERM REPORT TO DEPARTMENT OF ELECTRONICS AND COMPUTETR ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR BACHELOR’S DEGREE IN ELECTRONICS COMMUNICATION AND INFORMATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

January, 2024

**P**.**A**.**R**.**R**.**O**.**T**  
**Parallel Asynchronous Robots, Robustly Organizing Trucks**

BY:

NABIN KHADKA(HCE076BEI005)

NISHAN GNAWALI(HCE076BEI008)

PRASHANT BASHYAL(HCE076BEI009)

RAMU ACHARYA(HCE076BEI010)

A MAJOR PROJECT MID-TERM REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR IN ELECTRONICS COMMUNICATION AND INFORMATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

HIMALAYA COLLEGE OF ENINEERING

TRIBHUVAN UNIVERSITY

LALITPUR, NEPAL

January, 2024

# ABSTRACT

A warehouse is a place used for the storage of goods. It serves as a centralized location for inventory management, distribution, and logistics operations. Warehouses often have facilities for loading and unloading trucks and include storage systems such as racks or shelves. Where there are various items in various different places. These items are to be put into a specific place at certain time for the delivery of the order or to manage the warehouse.

This project is aimed to design and build a multi-agent robotic system capable of transporting pallets or items from a warehouse floor onto delivery area in an efficient manner. We aim to design algorithms and prototypes for computer vision, planning, and communication that is hoped to result in increased task-completion efficiency in a multi robot system that can be directly used in real-world systems in warehouses.

Keywords: *Efficiency; Multi-agent; Prototype; Robot; System; Warehouse.*

# TABLE OF CONTENTS

[ABSTRACT I](#_Toc156245068)

[TABLE OF CONTENTS II](#_Toc156245069)

[LIST OF ABBREVIATION IV](#_Toc156245070)

[LIST OF FIGURES V](#_Toc156245071)

[1. INTRODUCTION 1](#_Toc156245072)

[1.1 PROBLEM STATEMENT 1](#_Toc156245073)

[1.2 OBJECTIVE 2](#_Toc156245074)

[1.3 SCOPE AND APPLICATION 2](#_Toc156245075)

[2. LITERATURE REVIEW 4](#_Toc156245076)

[3. FEASIBILITY STUDY 8](#_Toc156245077)

[3.1 TECHNICAL FEASIBILITY: 8](#_Toc156245078)

[3.2 ECONOMICAL FEASIBILITY: 8](#_Toc156245079)

[3.3 OPERATIONAL FEASIBILITY. 8](#_Toc156245080)

[4. PROJECT METHODOLOOGY 9](#_Toc156245081)

[4.1. BLOCK DIAGRAM 9](#_Toc156245082)

[4.2. ALGORITHM 10](#_Toc156245083)

[5. REQUIREMENTS 12](#_Toc156245084)

[5.1 HARDWARE REQUIREMENTS 12](#_Toc156245085)

[5.1.1 NodeMCU1 12](#_Toc156245086)

[5.1.2 Electromagnet 13](#_Toc156245087)

[5.1.3 360o Servo motors 13](#_Toc156245088)

[5.1.4 USB camera 14](#_Toc156245089)

[5.1.5 Aruco Marker 15](#_Toc156245090)

[5.2 SOFTWARE REQUIREMENTS 16](#_Toc156245091)

[5.2.1 C++ 16](#_Toc156245092)

[5.2.2 Arduino IDE 16](#_Toc156245093)

[5.2.3 Open CV 17](#_Toc156245094)

[4.2.4 Python 17](#_Toc156245095)

[6. SYSTEM IMPLEMENTATION 18](#_Toc156245096)

[6.1 COMPUTER VISION 18](#_Toc156245097)

[6.1.2 Camera Stack 18](#_Toc156245098)

[6.1.3 Sandbox Detection 18](#_Toc156245099)

[6.1.4 Robot Localization 19](#_Toc156245100)

[6.2 Robot Planner 19](#_Toc156245101)

[6.2.1 Task Planner 20](#_Toc156245102)

[6.2.2 Path Planner 20](#_Toc156245103)

[7.PROGRESS MADE 21](#_Toc156245104)

[7.1 WORKS COMPLETED 21](#_Toc156245105)

[7.1.1 Aruco Marker Generation and Recognition: 21](#_Toc156245106)

[7.1.2 Object Detection: 21](#_Toc156245107)

[7.1.3 Hardware Setup for Robot: 21](#_Toc156245108)

[7.1.4 Camera Calibration: 22](#_Toc156245109)

[7.1.5 Localization. 22](#_Toc156245110)

[7.2. WORKS REMAINING 23](#_Toc156245111)

[REFERENCES 24](#_Toc156245112)

# LIST OF ABBREVIATION

AGVs Automated Guided Vehicles

ArUco Augmented Reality University of Cordoba

AS/RS Automated Storage and Retrieval Systems

BOM Bill of Materials

CNNTT Convolutional Neural Network Trajectory Tracking

CPSO Chaotic Particle Swarm Optimization

Etc Et Cetera

I2C Inter-Integrated Circuit

IDE Integrated Development Environment

i.e. Id Est

LCD Liquid-crystal Display

LED Light-emitting Diode

MATLAB Matrix Laboratory

MCU Micro-controller Unit

MPC Model Predictive Control

Navlab Navigation Laboratory

PWM Pulse Width Modulation

STAMC Simultaneous Task Allocation and Motion Coordination

STL Signal Temporal Logic

USB Universal Serial Bus

Wi-Fi Wireless Fidelity

# LIST OF FIGURES

[Figure 1: Block Diagram of Overall System 9](#_Toc156253130)

[Figure 2: Node Mcu 11](#_Toc156253131)

[Figure 3: Electromagnet 12](#_Toc156253132)

[Figure 4: Continuous Servo motor 13](#_Toc156253133)

[Figure 5: USB Camera 14](#_Toc156253134)

[Figure 6: Aruco Marker 15](#_Toc156253135)

[Figure 7: Arduino IDE 16](#_Toc156253136)

[Figure 8: Marker Recognition 21](#_Toc156253137)

[Figure 9: Schematic diagram of hardware setup 22](#_Toc156253138)

# 1. INTRODUCTION

Warehouse is a facility used for storing goods or products before they are distributed or sold, typically equipped with storage systems and logistics infrastructure. Warehouses play a crucial role in supply chain management by providing storage, inventory management, and order fulfillment services. They help ensure product availability, facilitate efficient order processing, and enable cost-effective transportation and distribution. Effective warehouse management and productivity is essential to optimize inventory levels, minimize storage costs, streamline order fulfillment processes, reduce order errors, and improve overall operational efficiency, leading to better customer satisfaction and business profitability.

Warehouse productivity is inversely proportional to how long it takes to load/unload a truck or delivery vehicle. As demands on warehouses grow, the first step to increasing warehouse productivity is often automating warehouses with robots. However, while humans might be able to move around each other fairly intuitively, having multiple robots running around a warehouse in an efficient and collision-free manner is quite difficult and complex. Companies such as Tesla Factory robots, Amazon Robotics, etc are just a few of the companies deploying swarm robots to factories and warehouses. This proposed project is supposed to explore some of these challenges in warehouses from both a technical and systems perspective and aims to design, integrate and develop a system for increasing the efficiency of warehouse operation and loading and unloading the vehicle.

## 1.1 PROBLEM STATEMENT

Warehouse is extremely curtail these days. Warehouse servers many small and large scale industries, producers, retailers, online stores and wholesalers. They keep their products in stock in the warehouse so that they can reach customers conveniently when required. As the warehouse has variety of items kept in stock and in large quantity warehouses has to face multiple problems. Inventory management, space utilization and timely dispatch of orders are some of them.

* Space Utilization: Optimizing the use of warehouse space can be difficult, especially when dealing with varying product sizes and storage requirements. Inefficient space utilization can lead to congestion and decreased operational efficiency.
* Order Dispatch: Efficiently picking, packing, and shipping orders in a timely manner can be complex, especially during peak periods. Inaccurate order fulfillment can result in customer dissatisfaction and increased costs.

To solve these problems robots are used in warehouse. These robots are responsible for finding the place of correct products to be dispatched or to be stored and keep the products to and from designated place. This increases the speed of work and also the efficiency of the work. But robots too have a lot of problems.

The problem with robots are these are very-very expensive and the initial cost of implementation is colossal. Hence only large warehouses like amazon can afford them. Also there are other challenges like integration of robots and finding best and shortest route every time for the pickup and dispatch of certain item to a designated place without collision.

## 1.2 OBJECTIVE

Every project is started with some objective. Our project has following objectives:

* To design and develop the algorithms and prototype for a warehouse management robot.
* To implement the existing and developed algorithms for integration of cost-effective prototype of the inventory management robots.

# 1.3 SCOPE AND APPLICATION

As this project aims to design and develop the low cost warehouse management/ order dispatch robots; it has vast field of application and huge scope. Following are the scopes and applications of the project:

**SCOPE**:

* Can be utilized to automate the process of tracking and managing inventory within a warehouse. They are aimed to accurately identify and locate items, reducing the need for manual inventory checks.
* Can efficiently navigate through warehouses to pick and pack items for order fulfillment resulting in faster and efficient order dispatch.
* Can sort and consolidate items based on various criteria, such as destination, order, or specific requirements.

**APPLICATIONS**:

* In e-commerce warehouses to handle the high volume of orders i.e. to assist in order picking, sorting and dropping resulting in improving order accuracy and delivery speed.
* In manufacturing units to automate the movement and management of raw materials, work-in-progress items, and finished goods within manufacturing facilities.
* In cold storage facilities where temperature-controlled environments are required, and robots can operate effectively without being affected by low temperatures.
* In various small medium and large warehouses to manage, pick and drop items in and out of the store.

# 2. LITERATURE REVIEW

Over the decades sopping through the online stores and other business and production companies has been extensively using the warehouses for the purposes of storing and holding goods. Also they have realized the importance of the warehouse management for the timely and efficient dispatch of the goods. Various researches related to similar field and the fields that can contribute in this project.

Trajectory tracking of a differential drive nonholonomic mobile robot is presented [1]. In addition to the complex relations of the control system, the nonholonomic system adds complexity to the system which has been solved using the feed-forward and feedback fuzzy logic controllers. An innovative scheme has been developed to track the reference trajectory in the presence of model uncertainties and disturbances. The performance comparison of the proposed controller is done with the standard back stepping controller and the simulation results show that the developed controller is best suited for the tracking trajectory problems.

This report describes the structure, implementation, and operation of a real-time mobile robot controller which integrates capabilities such as: position estimation, path speciation and tracking, human interfaces, fast communication, and multiple client support. The benefits of such high-level capabilities in a low-level controller was shown by its implementation for the Navlab autonomous vehicle. In addition, performance results from positioning and tracking systems are reported and analyzed. [2]

The smoother and quicker the journey between the customer clicking the online buy button and the delivery driver arriving at their door with the goods, the faster the retailer’s profits will roll in Online grocery retailer Ocado has created a system called the Ocado Smart Platform where robots bring the shelf stacks to human workers, who pick out the right products and package them up to be sent out. These robots travel many miles a day at speeds faster than a human can walk.

Nathan Wrench, head of the industrial and energy business at Cambridge Consultants, which has worked on the Ocado system, says an early pioneer of the robotic “goods to man” warehouse process was KIVA Systems, bought by Amazon in 2012. They use a flat warehouse floor to bring shelves of goods to human pickers. [3]

Warehouse robotics refers to the use of [automated systems](https://6river.com/3-key-advantages-of-robotic-warehouse-systems/), robots and specialized software to transport materials, perform various tasks and streamline/automate warehouse processes. In recent years, robotics has gained eminence in supply chain, distribution center, and warehouse management circles and continues to play a significant role in warehouse automation. The different types of warehouse robots are; Automated Guided Vehicles (AGVs); automated storage and retrieval systems (AS/RS), etc. [4]

The main purpose of this technical report is to describe in detail the implementation of the pure pursuit path tracking algorithm. Given the general success of the algorithm over the past few years, it seems likely that it will be used again in land-based navigation problems. This report also includes a geometric derivation of the method, and presents some insights into the performance of the algorithm as a function of its parameters. [5]

Out-of-the-box swarm solutions powering industrial logistics will need to adapt to the tasks at hand, coordinating in a distributed manner to transport objects of different sizes. This work designs and evaluates a collective transport strategy to move large and arbitrarily shaped objects in warehouse environments. The strategy uses a decentralized recruitment and decision-making process, ensuring that sufficient robots are in place for a coordinated, safe lift and transport of the object. Results show robots having no prior knowledge about the object’s size and shape were successfully able to transport them in simulation. [6]

The demand for application of mobile robots in performing boring and extensive tasks are increasing rapidly due to unavailability of human workforce. Navigation by humans within the warehouse is one among such repetitive and exhaustive task. Autonomous navigation of mobile robots for picking and dropping the shelves within the warehouse will save time and money for the warehousing business. In order to autonomously navigate within the warehouse, the mobile robot requires a path planning algorithm. This path planning algorithm generates a collision free path from the start point to goal point for the mobile robot. Robot’s charging area, shelves and consolidation area would be the possible start point and the goal point would be the shelves which depends on the customer’s order. In this algorithm, the robot will move towards the goal point if the robot’s neighboring point closer to the goal point is vacant. The developed collision free path planning algorithm is simulated in MATLAB and the developed algorithm generates feasible path for traditional parallel, traditional vertical, traditional horizontal, fishbone types of layouts in warehouse with various start and goal point. [7]

With the increasing applications of autonomous vehicles in dynamic and strictly constrained environments such as automated container terminals, efficient task/resource allocation and motion coordination (i.e., path and speed planning) of multi-autonomous vehicles has become the critical problem and have therefore been recently recognized as the key research issues by both academics and industry. This chapter addresses a generic approach for integration of task allocation, path planning and collision avoidance, which has so far not attracted much attention in the academic literature. A Simultaneous Task Allocation and Motion Coordination (STAMC) approach is presented. Two metaheuristic algorithms, i.e. simulated annealing and ant colony, and an auction algorithm are investigated and applied. The proposed approach is able to solve the scheduling, planning and collision avoidance problems simultaneously; improve the usage of bottleneck areas; handle dynamic traffic conditions and avoid deadlock. Simulation results demonstrated the effectiveness and efficiency of this approach. [8]

We tackle the challenging problem of multi-agent cooperative motion planning for complex tasks described using signal temporal logic (STL), where robots can have nonlinear and nonholonomic dynamics. Existing methods in multi-agent motion planning, especially those based on discrete abstractions and model predictive control (MPC), suffer from limited scalability with respect to the complexity of the task, the size of the workspace, and the planning horizon. We present a method based on timed waypoints to address this issue. We show that timed waypoints can help abstract nonlinear behaviors of the system as safety envelopes around the reference path defined by those waypoints. Then the search for waypoints satisfying the STL specifications can be inductively encoded as a mixed integer linear program. The agents following the synthesized timed waypoints have their tasks automatically allocated, and are guaranteed to satisfy the STL specifications while avoiding collisions. We evaluate the algorithm on a wide variety of benchmarks. Results show that it supports multi-agent planning from complex specification over long planning horizons, and significantly outperforms state-of-the-art abstraction-based and MPC-based motion planning methods. [9]

This paper presents an improvement of the actual output trajectory tracking performance of a mobile robot based on convolutional neural network controller with off-line and on-line tuning Back-Propagation algorithms. The goals of this strategy are to find the optimal path to direct its movement and to design Convolutional Neural Network Trajectory Tracking (CNNTT) controller in order to control the nonlinear kinematics mobile robot system. Therefore, a hybrid swarm optimization algorithm uses for solving the two most important problems of path planning; the first is that the path must avoid collision with obstacles, and the second it must reduce the length of the path to a minimum. This paper will discuss the finding of the shortest path with the optimum cost function by using three optimizations’ algorithms; Chaotic Particle Swarm Optimization (CPSO) algorithm, A-star algorithm, and a hybrid swarm optimization algorithm (ACPSO). The task of the proposed feedback (CNNTT) controller is to obtain precisely and quickly the robust left and right wheels velocity which are used to control the position and orientation of the mobile robot system. The (CNNTT) controller is accurate in terms of the mobile robot follows the desired paths quickly through fast obtaining the (CNNTT) controller’s parameters and a smooth linear wheels velocity actions are generated for mobile robot system with minimum number of cost-function evolutions that minimized the tracking error x-position around 4 cm and y-position around 2.5 cm and zero approximately orientation error as well as no oscillation in the responses. Finally, confirmed the effectiveness of the numerical simulation results of the proposed control strategy through comparison other types of controller simulation results. [10]

# 3. FEASIBILITY STUDY

The basic feasibility study for the project was done during the primary research phase from which we came into the conclusion that the project is both technically and economically feasible.

## 3.1 TECHNICAL FEASIBILITY:

The algorithms needed for the development of the backend and firmware is available and some new algorithms might be needed to be developed or existing ones might need some simple redesign. Also the hardware parts are available in the market. Hence the proposed system is technically feasible and can be developed.

## 3.2 ECONOMICAL FEASIBILITY:

The parts and the technology needed to develop the system are readily available in the market. These parts can be connected as per the requirement of the design and blueprint of the system. Hence we can conclude that the proposed system is economically feasible.

## 3.3 OPERATIONAL FEASIBILITY.

Designing and developing a system that can’t function well is not a practical thing. The projects or the system being designed should work in real life condition. The project is aimed to be operational in real warehouses or similar conditions and is intended to solve the ever existing problem of warehouse management.

# 4. PROJECT METHODOLOOGY

## 4.1. BLOCK DIAGRAM

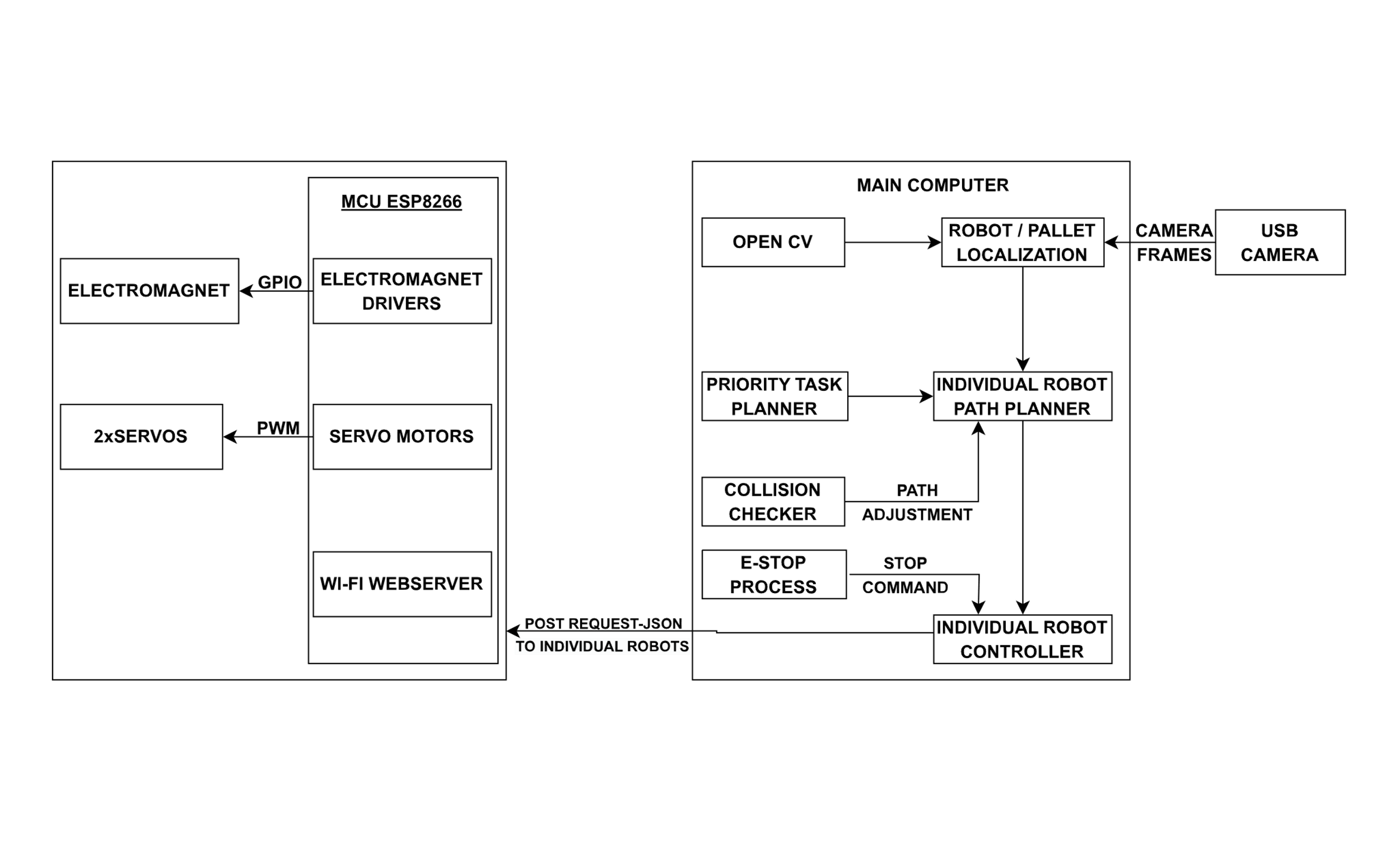


Figure 1: Block Diagram of Overall System

Above is the block diagram of the overall system of the proposed project. This diagram describes the working and requirements of the project clearly and in simple manner. Here we can see that the picture has two different segments. Left part represents the robot itself and the right one represents the remaining system which is actually the brain of the system and represents how overall system works.

Left part explains about the actual robot which is responsible for doing the actual task in the ware house. The warehouse management robot shall consist of electromagnet, NodeMCU and servo motors. The robot will be responsible for picking and dropping the goods with the help of electromagnets. The electromagnets will be triggered when the robot reaches near the specified item. The task of triggering shall be automated by the help of software and shall be done without human interaction. The servo motors help robots to move back and forth and in other direction. It shall be controlled by NodeMCU and is also responsible for receiving the command from main computer via wi-fi connection. The right part of the system showcases the main computer where actual code is running. The USB camera is responsible for taking video footage of the warehouse. From the footage pictures are taken into for processing and via image processing; the specific place of ArUco code is determined and also if the code is of goods or the robot is determined. Then the package which is to be picked is determined; the shortest distance between the robot and the package is then determined and if the robot is free the robot is assigned the task of doing that job. The shortest route for the robot carrying the package to the destination is determined in such a way that it don’t interfere the path of other robots. Each robot also has its own controller that keeps the robot along its specified path. The package is taken to the determined location & the electromagnet is turned off which detaches the package from the robot.

## 4.2. ALGORITHM

1. Start
2. Load ArUco detector and object detector
3. Establish connection between Robot and computer system in same network
4. If camera is connected; get ArUco markers from the corners else goto xiv
5. Discretize the floor into grid
6. Detect pallet and robots using open cv & Input pallets+goal pair and position of robots
7. Calculate the distance between Robots and pallets using formula

Distance=∣x2−x1∣+∣y2−y1∣

1. Allocate highest priority pallet to the robot with shortest distance & Plan the path with highest priority robot and loop over all robots
2. If intersection is present in the path; loop through other path and generate the new path
3. Send PWM signal to the robots and robot act accordingly
4. If position of pallet is same as the position of robot; initiate electromagnet
5. Plan the path of robot from current position to the goal position of pallet
6. If current position of the robot and goal position is same turn off the electromagnets
7. End

# 5. REQUIREMENTS

In the first look the design and the working of this project seems to be simple. But it consists of multiple hardware components and software counterparts for the proper and efficient functioning of the entire system.

## 5.1 HARDWARE REQUIREMENTS

### 5.1.1 NodeMCU1

Node MCU is a low-cost open source IoT platform. It initially included firmware, which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware, which was based on the ESP-12 module. It serves as the main control unit or the microchip for the robots in the project. And is also responsible for connecting the robots to the system via internet.



Figure 2: Node Mcu

**PINS**

NodeMCU provides access to the GPIO (General Purpose Input/Output) and a pin-mapping table is part of the API documentation.

|  |  |
| --- | --- |
| I/O Index | ESP8266 Pin |
| 0 | GPIO16 |
| 1 | GPIO5 |
| 2 | GPIO4 |
| 3 | GPIO0 |
| 4 | GPIO2 |
| 5 | GPIO14 |
| 6 | GPIO12 |
| 7 | GPIO13 |
| 8 | GPIO15 |
| 9 | GPIO3 |
| 10 | GPIO1 |
| 11 | GPIO9 |
| 12 | GPIO10 |

### 5.1.2 Electromagnet

Unlike a permanent magnet, the strength of an electromagnet can be changed by changing the amount of electric current that flows through it. If the current flow is cut, the property of magnetism ceases to exist. Electromagnets shall be connected to the robots and shall be controlled by the system itself to carry the pallets from their current position to destination, the electromagnets in the robots are used.

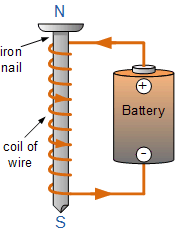


Figure 3: Electromagnet

### 5.1.3 360o Servo motors

A 360-degree servo motor, also known as a continuous rotation servo, is a type of servo motor that can rotate continuously in both directions. Unlike standard servo motors that have a limited range of motion typically around 180 degrees, continuous rotation servos can rotate indefinitely in either direction, making them suitable for applications where continuous movement is required.



Figure 4: Continuous Servo motor

**Specifications**:

|  |  |
| --- | --- |
| Control System | Analog |
| Operation Voltage | 4.8 – 7.2V |
| Servo type | analog servo |
| Speed | 4.8V: 0.20 sec/60°, 6.0V: 0.16 sec/60° |
| Rotation angle | 360° |
| Torque | 4.8V: 130.5 oz-in (9.40 kg-cm), 6.0V: 152.8 oz-in (11.00 kg-cm) |
| Operating temperature: | 30 ~ + 60 ° |
| Dimension | 40.7 × 19.7 × 42.9 mm |
| Weight | 55g |

### 5.1.4 USB camera

It is a digital camera that connects to a computer through a USB (Universal Serial Bus) port. USB cameras are commonly used for various purposes, including video conferencing, online streaming, video recording, and real-time video capture.



Figure 5: USB Camera

**Specification:**

|  |  |
| --- | --- |
| Dimension (W×H mm) | 49.9\*56.7\*63.8mm |
| Weight(g) | 124.8g |
| Input | USC-A to USB-C |
| Capture screen | 1920\*1080/1280\*720/640\*480 |
| Frame rate | Up to VGA 30 FPS |
| Interface | USB2.0, USB1.1 compatible |

### 

### 5.1.5 Aruco Marker

An ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier (id). The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques. The marker size determines the size of the internal matrix.

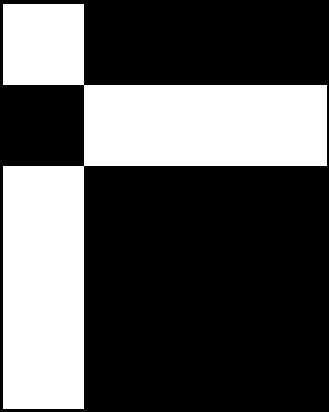


Figure 6: Aruco Marker

It must be noted that a marker can be found rotated in the environment, however, the detection process needs to be able to determine its original rotation, so that each corner is identified unequivocally. This is also done based on the binary codification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **0** | **0** | **0** |
| **0** | 1 | 1 | 1 | 1 |
| **1** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 |

(0011000000)2 = (192)10

A dictionary of markers is the set of markers that are considered in a specific application. It is simply the list of binary codifications of each of its markers. The main properties of a dictionary are the dictionary size and the marker size.

## 5.2 SOFTWARE REQUIREMENTS

### 5.2.1 C++

C++ is a general-purpose programming language that was developed as an extension of the C programming language. C++ provides a combination of high-level and low-level features, making it suitable for a wide range of applications, including system/software development, game development, embedded systems, and more. C++ language is being used for firmware coding in Arduino IDE in this very project..

### 5.2.2 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

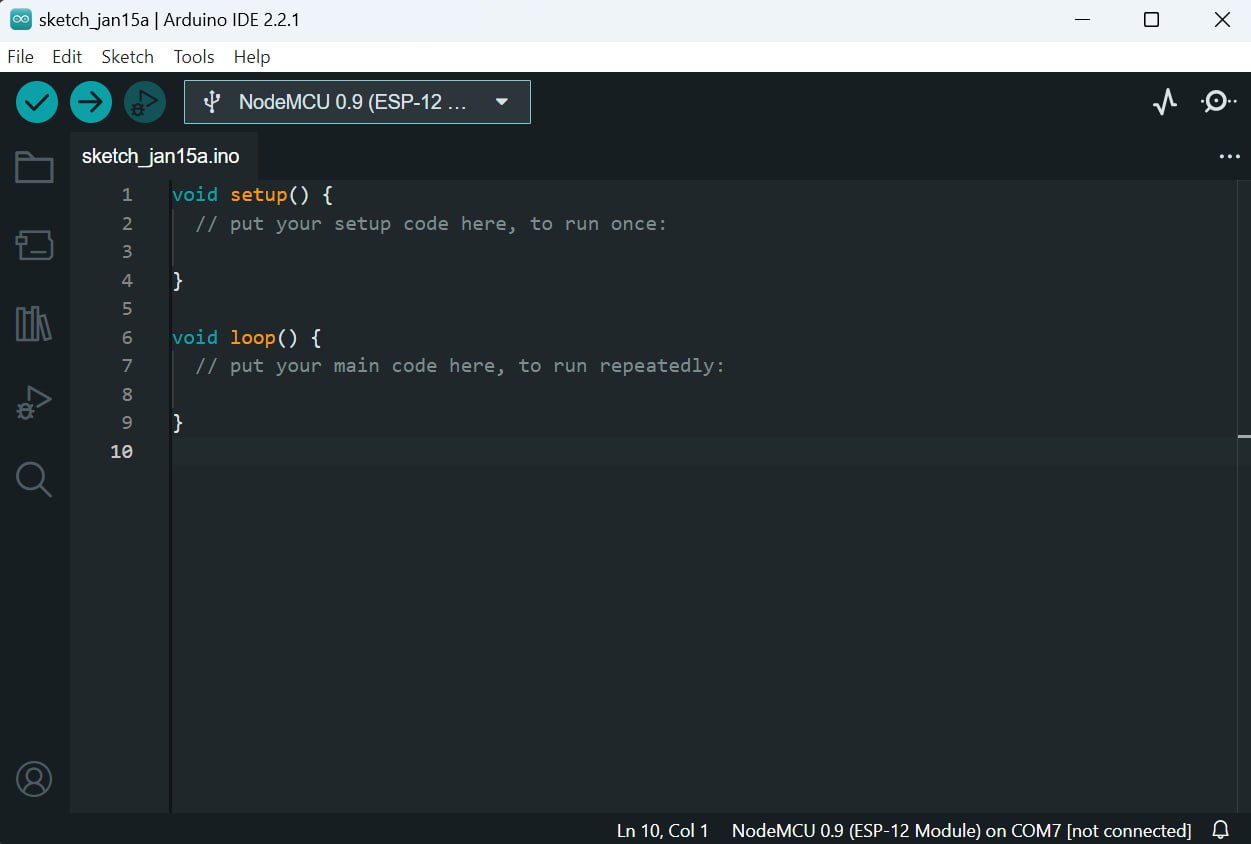


Figure 7: Arduino IDE

### 5.2.3 Open CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV provides a common infrastructure for computer vision applications and accelerate the use of machine perception in the commercial products.

It is being used as an efficient tool tool for image processing, finding the positions of pallets, goal positions and the total area of the playground in the project. It made easy for discretization and camera calibration too.

### 5.2.4 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python has a reputation as a beginner-friendly language.

It is being used for the main system development in open cv for using the functionalities like image and video processing, object detection, feature extraction, etc. Also, it provides flexibility for image detection, image processing for development of multi-agent robotic system. Therefore, the backend of the system is being developed using python programming language in openCV.

# 6. SYSTEM IMPLEMENTATION

Our main subsystems are the Robots, the Computer Vision section, the Robot Planner, and the Controller. The robots include hardware as well as firmware design, while the rest of the components are all software running on the main computer. Other major components of the system are described in this report.

## 6.1 COMPUTER VISION

The high-level goal for the computer vision algorithm is to determine where the field is and then determine where the robot is within that field. Our Computer-Vision stack work on Arcuomarker (QR-style) localization system. All of the Arcuomarker localization logic is implemented via OpenCV’s helper functions.

The robots have an Aruco marker attached on the top of them. Similarly, the goal position and pallets also have similar looking, yet unique Aruco markers attached. Also the field is defined using Aruco markers in all four corners. We use computer vision for our localization because the goal of our project is centered around the scalability of multi-agent robotic systems and we have budget/time restrictions. Computer vision enables us to not only localize our robots, but also localize our pallets via simple QR-fiducials, greatly simplifying pallet detection. It reduces the BOM cost of each robot.

### 6.1.2 Camera Stack

Our Camera hardware till date is our phone camera which can shoot 1080p video. The 1080p video is important for us to be able to hit our target localization accuracy. A 1920x1080p image across a 3m x 2m field means each pixel represents roughly 2mm in the real world

### 6.1.3 Sandbox Detection

The field detection is done with the use of a fiducial. Fiducials are like QR code tags that are supported by the OpenCV library for easy detection and localization. In particular, the cv2.aruco library is being used.

The computer vision stack is provided with a set of 4 fiducials that will mark the edges of the field. When the camera turns on, the system will look for these 4 fiducials and use the pixel positions to find which fiducial is the bottom left, top right, etc.

### 6.1.4 Robot Localization

The robot localization works on this transformed and flattened sandbox image. Using the flattened image, we can use OpenCv’s Aruco localization functions to find the x, y. theta positions of each of the pallets, goals, and robots. All this information is overlayed on a custom visualizer that draws the robot, pallet, and goal poses on the video stream with additional information like the robot paths and feedback + feedforward.

From this transformation matrix, the 3rd column represents the robot’s x and y position, and its angle is calculated with

θ = atan2 (sin θc(t), cos θc(t))

# 6.2 Robot Planner

When it comes to the robot planner, there are several approaches that are possible to achieve multi-agent coordination. Ultimately, it is a factor of implementation complexity that was the driving factor behind settling on our final strategy.

We assign robots priorities and then precompute all of their paths: planning in space and time. In this way, we can ensure that none of the robot paths intersect with each other in space-time configuration space. If a robot has completed its task early, it can drop to the lowest priority and plan around the motion of all the other robots. The only negative to this approach is that the robots must stick to their assigned path. Otherwise, the system is at risk of a collision. We can mitigate this risk by designing a robust controller and including an emergency stop in the form of the user killing the program as a safety measure against collisions.

It is divided into two parts i.e. task planner and path planner.

### 6.2.1 Task Planner

The task planner takes as input a list of pallet+goal pairs plus the positions of all the robots and assigns each robot a pallet and goal task. It prioritizes robots by minimizing the overall distance of the trajectory (i.e. robot to pallet and pallet to goal). The distance used is the straight line distance between the two sets of points. Overall, the set of assigned tasks should be such that the straight-line distance collectively traveled by all of the robots is minimized.

### 6.2.2 Path Planner

The path planner is the part of the system that coordinates the motion of all of the robots such that they can collaboratively work without colliding. Each robot has a sub-planner that is responsible for planning its motion given the paths of all other higher-priority robots (as decided by the task planner). The main planner loops over all of the robots in order of priority, planning the path of each of them around static obstacles (i.e. pallets) and the paths of higher-priority robots.

# 7.PROGRESS MADE

Our effort in the development of 'Parallel Asynchronous Robots, Robustly Organizing Trucks' has yielded notable progress in various critical domains. The following milestones reflect our efforts and achievements:

## 7.1 WORKS COMPLETED

### 7.1.1 Aruco Marker Generation and Recognition:

Developed and deployed ArUco markers, allowing robots to detect and recognize specific markers for navigation and coordination.



Figure 8: Marker Recognition

### 7.1.2 Object Detection:

Achieved the ability for robots to detect objects within their environment, enabling them to interact with their surroundings effectively.

### 7.1.3 Hardware Setup for Robot:

Successfully established the hardware configuration for robot, ensuring its operational readiness. We successfully completed designing of the power circuit for the robots. These circuits controls the power to be supplied to the electromagnet and servos.

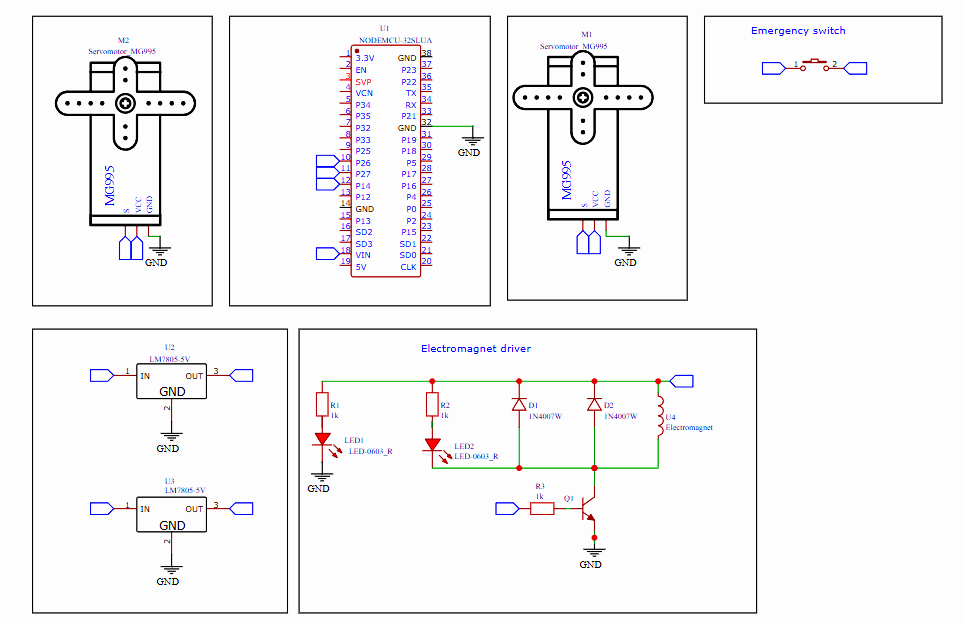


Figure 9: Schematic diagram of hardware setup

### 7.1.4 Camera Calibration:

Completed the calibration process for camera enhancing the accuracy in perceiving the environment.

### 7.1.5 Localization.

In this, the robots and pallets have been located in 2D grid.

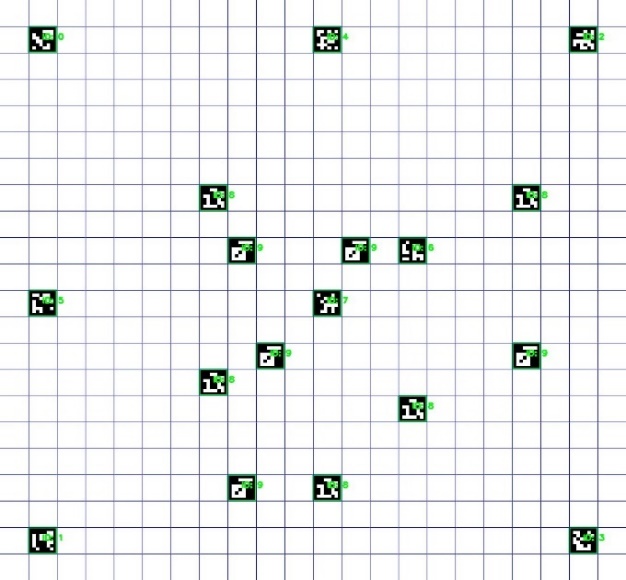


Figure 7.1.3: Localization

## 7.2. WORKS REMAINING

* **Path Planning and Task Planning Algorithm Implementation:** Currently working on implementing algorithms that enable the robots to plan efficient paths and tasks autonomously.
* **Hardware and Software Integration:** Continuing efforts to seamlessly integrate both hardware components and software modules for cohesive robot functionality.
* **Advancement of Robots**: Here the robot has used bo-motor for testing purposed but shall be shortly replaced with servos in the final form of the robot.

# REFERENCES

[1] Asif, M. *et al.* (2015) ‘Feedforward and feedback kinematics controller for wheeled mobile robot trajectory tracking’, *Journal of Automation and Control Engineering*, 3(3), pp. 178–182. doi:10.12720/joace.3.3.178-182.

[2] Omead Amidi. Integrated Mobile Robot Control. Tech. rep. CMU-RI-TR-90-17. Pittsburgh, PA: Carnegie Mellon University, 1990.

[3] Benady, D. (2022) *How robotics are Optimising Warehouse Operations*, *Raconteur*. Available at: https://www.raconteur.net/technology/automation/how-robotics-are-optimising-warehouse-operations/ (Accessed: 07 July 2023).

[4] Glynn, F. (2023) *Warehouse Robotics & different types of robots*, *6 River Systems*. Available at: https://6river.com/what-is-warehouse-robotics/ (Accessed: 07 July 2023).

[5] Coulter, R.C. (1992) *Implementation of the pure pursuit path tracking algorithm*. Pittsburgh, PA: Carnegie Mellon University, the Robotics Institute.

[6] Jurt, M. *et al.* (2022) ‘Collective transport of arbitrarily shaped objects using robot swarms’, *Artificial Life and Robotics*, 27(2), pp. 365–372. doi:10.1007/s10015-022-00730-5.

[7] Kumar, N.V. and Kumar, C.S. (2018) ‘Development of Collision Free Path Planning Algorithm for Warehouse Mobile Robot’, *Procedia Computer Science*, 133, pp. 456–463. doi:10.1016/j.procs.2018.07.056.

[8] Liu, D.K. and Kulatunga, A.K. (2007) ‘Simultaneous planning and scheduling for multi-autonomous vehicles’, *Evolutionary Scheduling*, pp. 437–464. doi:10.1007/978-3-540-48584-1\_16.

[9] Sun, D. *et al.* (2022) ‘Multi-agent motion planning from Signal Temporal Logic Specifications’, *IEEE Robotics and Automation Letters*, 7(2), pp. 3451–3458. doi:10.1109/lra.2022.3146951.

[10] Wahhab, O. and Al-Araji, A. (2021) ‘Path Planning and Control Strategy Design for mobile robot based on hybrid swarm optimization algorithm’, *International Journal of Intelligent Engineering and Systems*, 14(3), pp. 565–579. doi:10.22266/ijies2021.0630.48.

[11]*5.10 electromagnets* *Mrs. Foster’s Fourth*. Available at: http://fostersfourth.weebly.com/510-electromagnets.html (Accessed: 25 November 2023).

[12]*MG995 Metal Gear Servo Motor (180° rotation)* (2023) *Rytronics.in*. Available at: https://www.rytronics.in/product/mg995-metal-gear-servo-motor-180-rotation/ (Accessed: 25 November 2023).

[13]*Nodemcu ESP8266* *Components101*. Available at: https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet (Accessed: 25 November 2023).

[14]*Rapoo C260 1080 full-HD USB webcam price in Nepal* *ITTI*. Available at: https://itti.com.np/product/rapoo-c260-1080p-web-camera-price-nepal (Accessed: 25 November 2023).