Machine Learning Based Industrial Sorting Platform

Final Year Design Project Report
(EE491 & EE492)

Submitted By

Atta Ul Haleem	2019093
Inshal Ahmed Khan	2019185
Mustafa Anique Ansari	2019415
Shaheer Ahmed	2019465

Advisor: Dr. Muhammad Irfan

Co-Advisor: Engr. Umar Afzaal

Faculty of Electrical Engineering
GIK Institute of Engineering Sciences and Technology
Topi 23640, District Swabi, Khyber Pakhtunkhwa, Pakistan
www.giki.edu.pk

May 2023

Certificate of Ownership

It is certified that the work presented in this report, regarding our senior year design project titled,

Machine Learning Based Industrial Sorting Platform

has been carried out through the collective efforts of all the undersigned team members. This work is carried out under the supervision of the undersigned faculty members, using number of available resources including research articles, books, and expert opinion.

Atta Ul Haleem
2019093

Inshal Ahmed Khan
2019185

Mustafa Anique Ansari
2019415

Shaheer Ahmed
2019465

Advisor
Dr. Muhammad Irfan
Assistant Professor

Co-Advisor
Engr. Umar Afzaal
Lecturer

Dr. Mohammad Akbar
Dean, Faculty of Electrical Engineering,
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology
Topi 23640, District Swabi, Khyber Pakhtunkhwa, Pakistan

Submission Date: 17th May, 2023

Final Year Design Project as a Complex Engineering Problem

It is to certify here that the final year design project (FYDP) entitled *Machine Learning Based Industrial Sorting Platform* is categorized as a complex engineering problem (CEP) based on the preamble (in-depth engineering knowledge) and involvement of the following attributes.

1. Depth of knowledge required
2. Depth of analysis required
3. Familiarity of issues
4. Interdependence

The above listed attributes are thoroughly assessed after conducing meeting on 7th September 2022 with the following final year students, who proposed the idea of the titled FYDP.

1. Atta Ul Haleem 2019093
2. Inshal Ahmed Khan 2019185
3. Mustafa Anique Ansari 2019415
4. Shaheer Ahmed 2019465

This project is going to be conducted in fall semester 2022 and spring semester 2023. Further, it is submitted that the proposed idea is worthy, and the required efforts are up to the level of a final year design project.

FYDP Advisor
Dr. Muhammad Irfan
Assistant Professor
Date:

FYDP Co-Advisor
Engr. Umar Afzaal
Lecturer
Date:

Acknowledgements

To start off, we owe all our gratitude to the Almighty Allah, who has given us wisdom, bravery, and power. May Allah guide us and help us all in our lives here and hereafter. Without his blessings, none of this would have been possible.

We would like to thank our advisor, Dr. Muhammad Irfan, for the continued motivation and help he has given us throughout the completion of this project, and for how he never doubted our abilities. Furthermore, we would like to thank the Faculty of Electrical Engineering, for giving us this opportunity to present our project and make it successful. Lastly, to our parents, without whom we would have given up a long time ago.

May Allah guide us and help us all in our lives here and hereafter.

Group 7, May 2023

Contents

List of Figures

List of Tables

Abstract

Throughout the scope of this project priority was set to improving the efficiency of traditional conveyor systems in use today. We proposed a solution through an omnidirectional, Machine Learning integrated, package handling platform which aids in package grouping and sorting. The platform is designed to be modular so that it may be reconfigurable to cater versatile needs. The platform consists of hexagonal, independent modules linked together to create the overall platform with omnidirectional wheels installed at alternative sides of each module. This provides omnidirectional movement of packages. Furthermore, Reinforcement learning was explored as a viable option to increase sorting and grouping efficiency via shortest path planning and collision avoidance in the cases of multiple packages. Q – Learning, Double Q – Learning and Deep Q – Learning and algorithms were rigorously evaluated through simulations to find out the most viable option. Simulations were run for large- and small-scale platforms. The proposed algorithms were also evaluated for object avoidance after a series of episodes were simulated. On the basis of simulations and results, as shared below, Q – Learning algorithms was selected.

In order to reduce the complexity of the project a Centralized Control System was designed for the platform modules with Raspberry Pi as master controller. For motor control dedicated motor driver circuits were implemented independently for each module alongside shift registers to reduce the hardware complexity and specially to reduce the number of pins on Raspberry Pi for motor control due to shift register implementation we were able to handle all modules via a single pin.

Keywords: Omnidirectional, Q-Learning, Double Q-Learning, Deep Q-Learning, Raspberry Pi.

1 Introduction

This chapter presents a brief overview of the project rationale, related problem statement, and overall scope and expected outcome of the project. This chapter also covers the aspects of sustainable development goals and complex engineering problem in the context of this work. The last section of the chapter highlights the overall contribution of this project work. [1-5]

1.1 Background and Motivation

Product flow in the logistic hubs of the current era largely depends on the usage of traditional roller conveyors which are inflexible and non-digitized. This is due to the fact that the package movement possible on these platforms are either uni-directional or difficult to set up. In order to be more capable, these conveyors have to be incorporated with more moving parts without adding to the complexity of controlling those extra moving parts. The following are the main difficulties that the upcoming conveyor technologies must overcome.

- They are inflexible: A shift in logistics operations often necessitates an alteration of equipment setup, from the displacement of hundreds to thousands of metres of conveyor systems, to the installment of fresh components, to a total revamp of the conveyor systems intended for the transportation of items.
- They are Maintenance-unfriendly: As systems become more intricate, the need for additional spare parts increases. Every technical maintenance procedure needs to be completely reevaluated. This process can be quite lengthy.
- They take a lot of Space & are cost intensive: The cost of floor space is rising, especially in the vicinity of urban centers. At the same time, the volume of logistics is growing, which means that more material needs to be moved on the available space. While computer chip technology has decreased in size over time, conveyor technology has not fundamentally altered. Existing frameworks make it challenging to save space while keeping up with productivity

The solution to these challenges is: Modular and software-driven conveyor technology, which is built around a single component: the cell. A german based company by the name Cellumation created a design of conveyor system which incorporated all of the aforementioned qualities. The design was industrial grade, sturdy and with less chance of failure. Our design is very similar to the Cellumation, integrating more features to make it novel.

1.2 Problem Statements

We aim to modernize the currently used traditional method of conveyor systems by bringing innovation to the existing machines. Our goal is to develop a conveyor system which allows the movement of packages in multiple directions. This will improve not only improve the efficiency and capability of the logistics industry but also solve the issues currently faced by the Pakistani shipping industry hindering the growth of FMCGs and other consumer goods industry.

1.3 Scope of Work and Expected Outcome

Our work aims to produce accurately working software integrated with the hardware. We divided the project into 4 main phases: Design, Procurement, Assembly and Testing. All of the phases were planned using project management techniques. The expected outcome of the project was to produce a tool that had a positive impact on the society.

1.4 Sustainable Development Goals

This section presents a brief overview of all the SDGs and mainly justifies the contribution of the project to the sustainable development goals (SDGs). Detailed justification of the mentioned points is presented in Table 1.1.

Table 1.1 SGD Table of the Project

Sr. No	**Title**	**Compliance (Y/N)**	**Remarks/Justification**
1	No poverty	N	NA
2	Zero hunger	N	NA
3	Good health/wellbeing	N	NA
4	Quality education	N	NA
5	Gender equality	N	NA
6	Clean water and sanitation	N	NA

7	Affordable and clean energy	N	NA
8	Decent work and economic growth	N	NA
9	Industry, innovation and infrastructure	Y	Idea which is integrating innovative features to produce a novel machine
10	Reduced Inequalities	N	NA
11	Sustainable Cities and Communities	N	NA
12	Responsible consumption and production	N	NA
13	Climate action	N	NA
14	Life below water	N	NA
15	Life on land	N	NA
16	Peace, Justice and strong institutions	N	NA
17	Partnerships for the goals	N	NA

1.5 Complex Engineering Problem

This project satisfies the attributes of the complex engineering problem, in the given context, this section presents the justification that how the presented work addresses different attributes of the complex engineering problem. The details are presented in

Table 1.2.

Table 1.2 CEP Attributes Mapping

Sr. No	**Attribute**	**Justification**
1	Preamble - In-depth engineering knowledge	Design of hardware and software used technical knowledge of programming and robotics.
2	Range of conflicting requirements	There was an opportunity cost in overall cost against control system design, where keeping cost low was preferred.
3	Depth of analysis required	Issues faced during the project were dealt with several new techniques developed from critical analysis, such as communication system between modules, hardware design.
4	Depth of knowledge required	Development of the project included the theoretical and practical knowledge acquired from the technical courses studied over the course of degree, especially control systems and programming.
5	Familiarity of issues	Project selection was done on the basis of research done on current issues faced by the logistics and warehouse industry in Pakistan from research articles and papers.
6	Extent of applicable codes	Required going beyond the technical knowledge acquired in degree for design of the modules.

7	Extent of stakeholder involvement and level of conflicting requirement	Advisor and Co Advisor were appropriately involved as well as the faculty members during presentations for critique.
8	Consequences	Knowledge of project management and design was acquired during the project life cycle.
9	Interdependence	Control Systems, Microcontrollers, CAD Modelling, software programming were involved.

1.6 Contribution

The entire project was a team effort. To ensure all aspects of the project demands are being looked up at all times, the team was divided into 2 groups, one overlooking the hardware section including designing, fabrication, procurement and assembly while the other looked over the software development including algorithm for motor control, GUI, Open CV and integration with the motor controller. The hardware section was looked over by Inshal Ahmed and Shaheer Ahmad, while the software work was handled by Mustafa Ansari and Atta ul Haleem. All four group members were responsible for the documentation of their respective domains.

2 Literature Review

This section of the project aims to provide a comprehensive review of the literature related to machine learning-based industrial sorting platforms. It examines the current state-of-the-art techniques and technologies used for sorting tasks in various industries, including their strengths and limitations.

By surveying the existing research and publications, this section seeks to identify the key challenges and opportunities in the field of machine learning-based industrial sorting platforms. It also explores the various applications of these platforms in different industries and their potential impact on industrial processes and operations.

The literature review section serves as a critical foundation for the rest of the project, providing a solid understanding of the current state-of-the-art and potential future developments in the field.

2.1 Literature Review

According to Keek et al. [1], there exist two types of control systems for omnidirectional cellular conveyors namely Decentralized Control System (DCS) and Centralized Control System (CCS). Both systems have their own advantages and disadvantages.

Decentralized systems are more complex since they require extra hardware such as photosensitive sensors to detect package movement whereas a centralized system can achieve the same using a single camera. With the rapid advancement of computer vision techniques, it has become quite trivial to identify the carton location, size, and orientation. A detailed comparison of DCS and CCS is shown in Table 3.

Decentralized Control System (DCS)	**Centralized Control System (CCS)**
Separate controller for each module	Single controller for complete system
Communication system in each module	Communication directly via controller
Feedback via integrated sensors	Feedback through camera
Complex hardware	Simple hardware
Complex programming algorithms	Computer vision techniques

Table 3: Comparison between Centralized and Decentralized Control Systems

Furthermore, decentralized systems necessitate communication systems within each module and the algorithms needed for path planning are increasingly complex. Examples of such algorithms include the Von Neumann and Moore neighborhood methods which are based on the concept of cellular automata. In contrast, computer vision techniques are simple to implement and provide sufficient accuracy.

Most existing literature on omnidirectional cellular conveyors implements two types of control systems. In the case of the E-pattern Omnidirectional Cellular Conveyor (EOCC), there is one control system for yaw axis movement and another for orientation control for each wheel. Similarly, the hexagonal-shaped cellular conveyor proposed by Uriarte et al. [2] utilizes two control systems: one for velocity control and another for angle control. The proposed systems are usually PD (Proportional-Derivative) control systems. Based on this literature, we will also implement two control systems to ensure accurate movement of cartons on the platform. An overview of the system design is summarized in the block diagram in Figure 1: Block Diagram of the Platform.

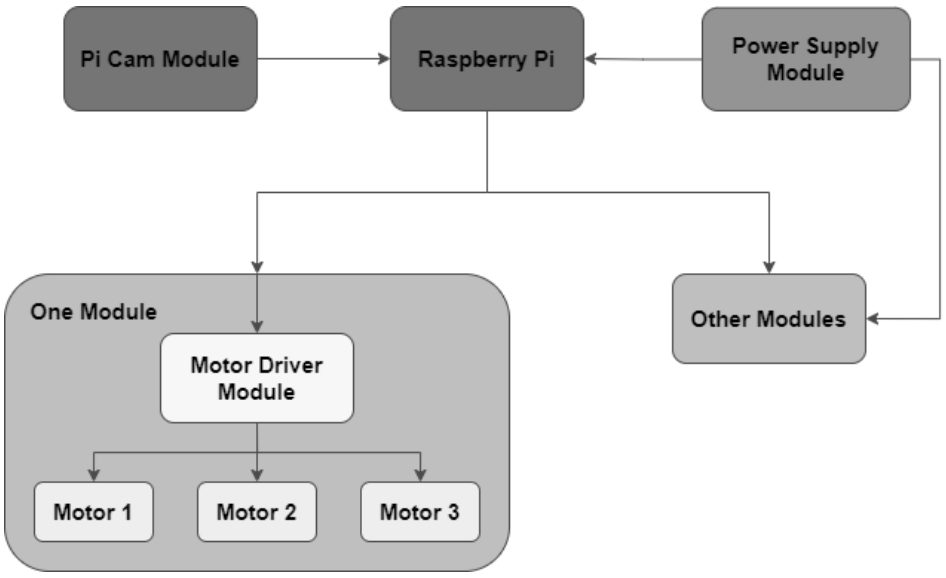

Figure 1: Block Diagram of the Platform

As shown in Figure 1, Raspberry Pi is the main controller being used to control the motor driver modules, which are adjusting the velocities of motor. The only input given to the whole system is the video feed by the Pi Cam Module. That input is used to deploy the control system techniques mentioned in the literature above. The Power Supply Module is used to power the whole system and consists of the over-voltage and over-current protection systems to prevent system damage.

There are different configurations of wheel placements possible, the ideal being hexagon, as given in the paper by J. S. Keek et al. [1] and penta-wheel designs, proposed in the paper by Sun, T. et al. [3]. Hexagonal Design uses less motors and wheels, while allowing for 3 axes of movements. This allows cost saving and energy efficiency in the system, and a simplified control system, as there are less motors being controlled. Both designs are illustrated in Figure 2.

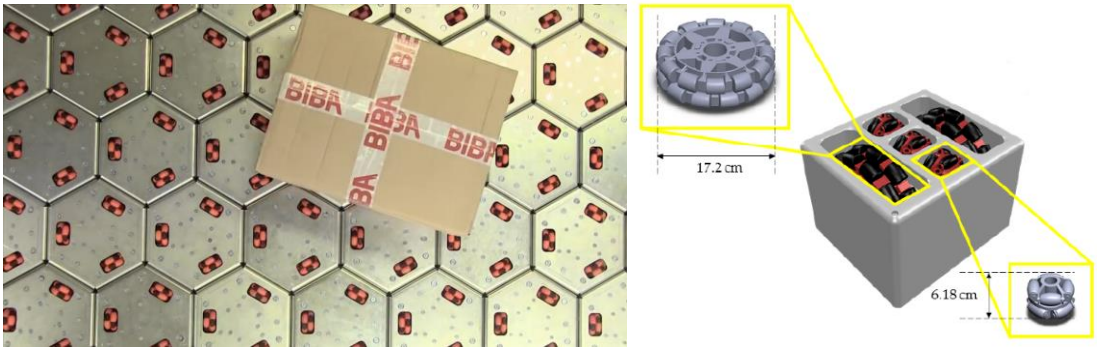

Figure 2: (a) Hexagonal Tri-Wheel Design [1] and (b) Penta-Wheel Design [2]

An ideal design of the conveyer unit cell is analyzed by Yongguo Z. et al. [3], where the cell design is a hexagon with 3 wheels, their axis placed in 3 directions, 120 degrees apart. The omni-directional wheel can be said to be placed in a circle centered in the hexagon, with each wheel having radius r from the center. The relation of the length l of a side of hexagon and radius r is:

$$l = \sqrt{3} \times r$$

Equation 1: Relation between length and radius in hexagonal design.

This is ideal for the package to be stable in movement on the conveyor belt. In addition, there is a caster wheel in the center to support the movement. The wheel's rotation velocity is calculated for all three wheels separately, as required for the package to have 3 axis movement. This concept is illustrated in Figure 3.

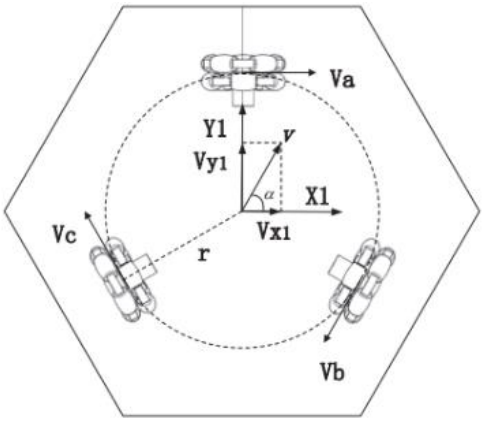

Figure 3: Velocity Components in Hexagonal Design [3]

The matrix equation (Equation 2) relates ω , the rotation velocity of package, V_{x1} and V_{y1} , the x and y component of straight-line velocity of package, r , radius of wheels from center, with V_a , V_b and V_c the rotational speed of the omnidirectional wheels, a , b and c :

$$V_a = V_{x1} + \omega r$$

$$V_b = -\frac{1}{2}V_{x1} - \frac{\sqrt{3}}{2}V_{y1} + \omega r$$

$$V_c = -\frac{1}{2}V_{x1} + \frac{\sqrt{3}}{2}V_{y1} + \omega r$$

Equation 2: Velocity Components as illustrated in Figure 3

2.2 Machine Learning

Machine learning is an overly broad term in today's day and age. To narrow it down, the type of machine learning that is applicable to our project is reinforcement learning. Reinforcement learning is a type of machine learning that rewards desired behaviors and punishes undesired ones. It trains agents to optimize their actions in an environment to maximize a cumulative reward. It is widely used in robotics, gaming, and autonomous systems. [4]. This model learns by trial and error. We can set rewards and penalties for the actions taken by the model and train it to seek the maximum overall reward and achieve an optimal solution. With time and more and more trials, this agent keeps improving.

For our project, the RL algorithm we have studied is set up by considering a Markov Decision Problem (MDP). This takes 4 variables into account which are, S (states), A (action), R (rewards), S' (next states). This RL algorithm rewards the correct movement of the package and penalizes collisions or faults. The three types of algorithms used in the case study are, Q-Learning, Double Q-Learning, Deep Q-Learning, and Double Deep Q-Learning.

2.2.1 Q-Learning

Q-learning is a reinforcement learning algorithm used for solving Finite Markov Decision Processes (FMDP). It finds the optimal policy that maximizes the expected value of the total reward over all successive steps, starting from the current state.

The working principle of Q-learning involves evaluating all possible actions taken by the system to move to possible next states. It calculates the rewards associated with each action, including short-term and

long-term rewards, and chooses the action/state pair that maximizes the rewards. The Q-values, which represent the expected rewards for each action/state pair, are stored in the Q-table. A flowchart can be used to illustrate the working principle of Q-learning, showing the process of selecting actions based on the Q-values in the Q-table [5].

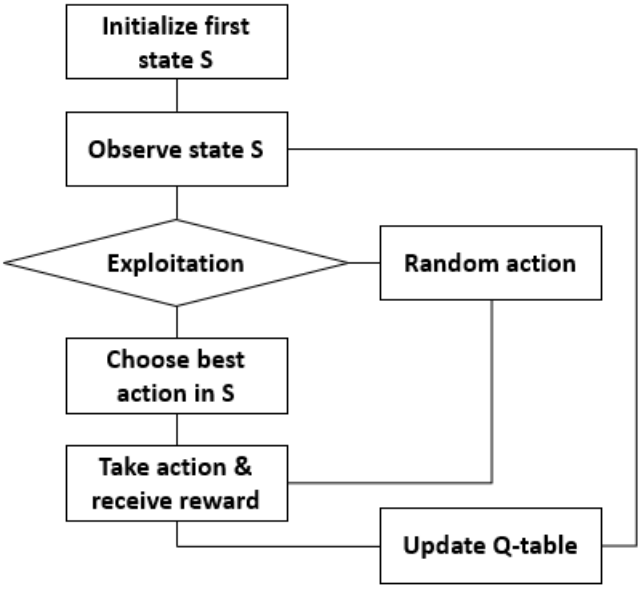

Figure 4: Flowchart for updating Q-Table

The Q-table is then modified based on the new value obtained through Equation 3.

$$Q(S, A) = Q(S, A) \times (1 - \alpha) + \alpha[R + \gamma \operatorname{argmax} Q(S', a)]$$

Equation 3: Q-Value formula for updating Q-Table

Here, S represents the current state, A represents the current action, α denotes the learning rate, γ denotes the discount factor, and the term "argmax Q(S', a)" denotes the state-action pair that maximizes the Q-value in the next state.

2.2.2 Double Q-Learning

Double Q-Learning enhances the Q-Learning algorithm by eliminating the issue of maximization bias. It follows a similar approach as Q-Learning but uses two separate Q-tables to reduce overestimation. During each time step, only one table is randomly updated. Actions are then selected in the same way as Q-

Learning, but based on a table that results from combining the two updated Q-tables [5]. Both tables are updated using Equation 4.

$$Q1(S,A) = Q1(S,A) \times (1 - \alpha) + \alpha[R + \gamma\, Q2(S, \operatorname{argmax} Q1(S', a)]$$

$$Q2(S,A) = Q2(S,A) \times (1 - \alpha) + \alpha[R + \gamma\, Q1(S, \operatorname{argmax} Q2(S', a)]$$

Equation 4: Formulas for updating Q-Table in Double Q-Learning

2.2.3 Deep Q-Learning

The combination of reinforcement learning and neural networks is known as Deep Q-Learning. This algorithm shares similarities with Q-Learning but incorporates an experience replay memory D to store transitions consisting of the current state, current action, current reward, and next state. Additionally, a second network, Q', referred to as the target network, is introduced. [5] At each time-step, the next state is randomly selected from the experience replay memory and overestimated using the target network. This estimate is then employed to update the original network using Equation 5.

$$Q(S,A;\ \theta) = Q(S,A) \times (1 - \alpha) + \alpha[R + \gamma \operatorname{argmax} Q0\ (S', a;\ \theta-)]$$

Equation 5: Formula for updating network in Deep Q-Learning

2.3 Inference from Literature

From the comprehensive literature review preformed, the following deductions can be inferred:

1. The combination of efficiency, uniformity, structural integrity, and flexibility make hexagons an excellent choice for modular systems.
2. The most optimum size for omni-wheels would be 40mm for the purpose of sorting packages.
3. Reinforcement learning algorithms would be most suitable for our project, as they are capable of further expansion. For example, if the platform size was to increase, the algorithm would still remain the same.

2.4 Chapter Summary

In this section, we have explored several types of control systems, namely Centralized and Decentralized. Existing hardware was also extensively researched to find the most optimum arrangement to fabricate the modular system. Furthermore, the use of Machine Learning in robotic systems like this was also explored, and different types of reinforcement learning models have been reviewed.

3 Design and Methodology

This section focuses on the hardware design and implementation of our project. It discusses in depth the reasoning for all the design decisions made along with the integration of software with hardware.

3.1 Hardware Design

At the core of this project are the Omnidirectional wheels, shown in Figure 5. Also known as mecanum wheels, are a type of wheel that allows a vehicle or robot to move in any direction without changing its orientation. They are made up of a series of small rollers arranged at an angle to the wheel's axis, which allows them to move both forwards and sideways. This makes them particularly useful for applications where maneuverability is key, such as in our case where we require free movement on the whole platform. By enabling a system to move in any direction, omnidirectional wheels provide greater flexibility and efficiency, allowing tasks to be completed more quickly and effectively.

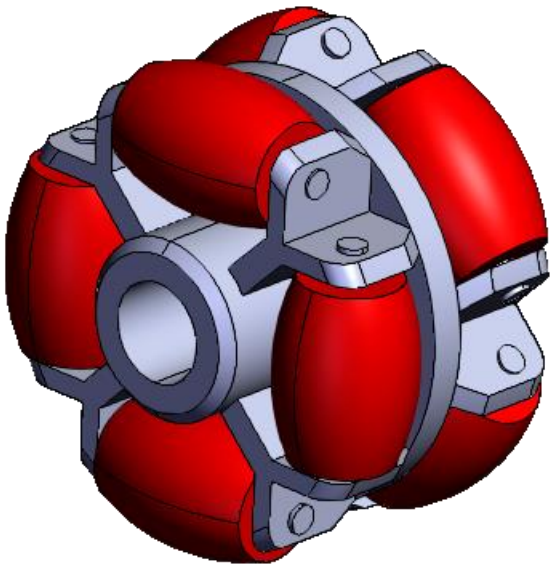

Figure 5: Omni-Wheel [6]

Given that the platform is composed of 10 individual modules, it is crucial that each module is modular, and that they can be easily attached and repositioned as needed. To facilitate this, we have designed a magnetic alignment system, shown in Figure 6, that enables seamless attachment of the different

modules. With this system, the modules can be easily and quickly attached, allowing for efficient repositioning and modification of the platform's layout. This approach ensures that the platform remains versatile and adaptable to a wide range of applications. The final CAD model of our hardware design is shown in Figure 7.

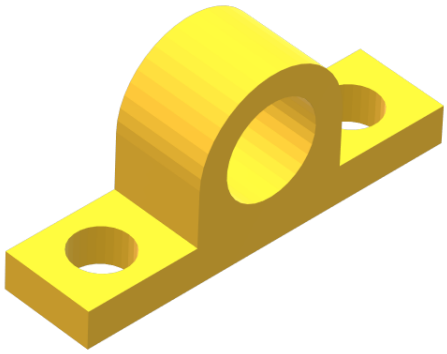

Figure 6: Magnet mount

Figure 7. CAD Model

3.2 Methodology

3.2.1 Shift Register for GPIO

The L293D Arduino Motor Driver Shield is a popular motor driver board used to control DC motors and stepper motors with an Arduino microcontroller. It features two L293D H-bridge driver chips, each of which

can drive up to 600 mA of current per channel and up to 36V of voltage, making it suitable for a wide range of motor applications.

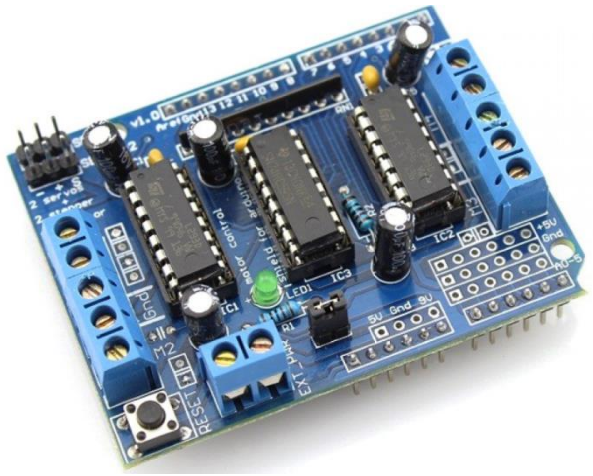

Figure 8: L293D Motor Driver Shield [6]

The board is designed to be mounted directly onto an Arduino board, but we have retrofitted it to integrate with Raspberry Pi. It also features a range of connectors, including screw terminals for power and motor connections, and male headers for easy access.

One of the key features of the L293D Arduino Motor Driver Shield is its ability to control the direction of the motor. The H-bridge design of the L293D driver chips allows the polarity of the motor to be reversed, enabling the motor to run in either direction. This is achieved by sending a signal from the Arduino to the driver chip, which then adjusts the voltage across the motor accordingly. The L293D Arduino Motor Driver Shield can also control the speed of the motor using pulse width modulation (PWM). By varying the duty cycle of the PWM signal, the speed of the motor can be adjusted. This is particularly useful for applications such as robotics, where precise control of motor speed is essential.

The Raspberry Pi has 26 GPIO pins which can limit the number of motors that can be controlled. Since the Omniveyor is designed to be modular and extendable, we needed a way to increase the GPIO pins. A typical solution for such a problem is the use of shift registers in Serial In Parallel Out (SIPO) mode. Moreover, shift registers can be daisy chained i.e., connected in series with each other to provide an arbitrary number of pins.

To avoid the long design, fabrication, and testing time of a PCB, we utilized the L293D motor driver shield which was originally intended for use with Arduinos. The shield already contained the components we

needed and was available locally for low cost. Some small changes were made to the shields such as shorting the Ground pin to the D7 pin which corresponds to the Output Enable ($\overline{OE}$) of the Shift Register as well as connecting V_{in} to 5V. The schematic is shown below:

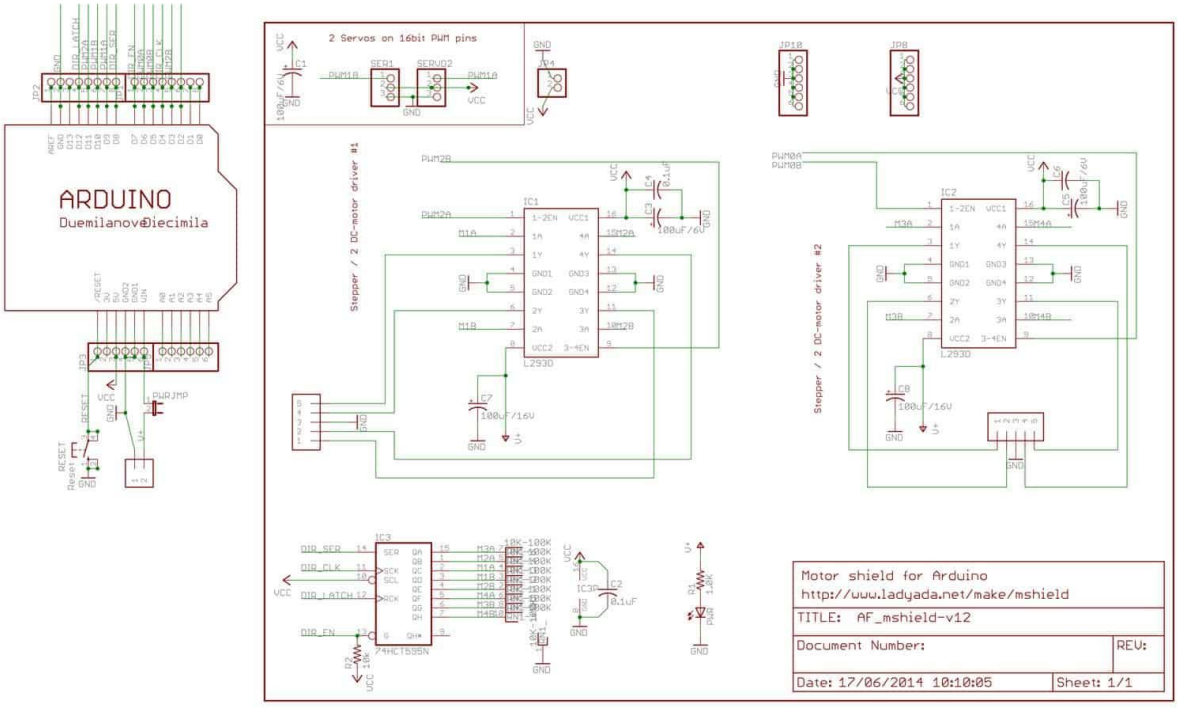

Figure 9: Motor shield schematic diagram [7]

The challenge was making the shield work with Raspberry Pi. For this OOP principles were applied in Python to create a library for interfacing with the shield. The library has the following methods and attributes:

ShiftRegister	
Attributes	**Methods**
data_pin	Pulse
clock_pin	shift_out
latch_pin	Clear

daisy_chain	
delay	

Table 4: Shift Register attributes and methods.

The shift_out function can be used to write to a chain of shift registers as defined by daisy_chain.

3.2.2 Implementing scalable Omniveyors

The rules of Object-Oriented Programming (OOP) were followed when writing the code for the modules. This allows for highly modular and scalable systems which is a key advantage of our project. The main components of the project were identified as the motors, the modules, and the Omniveyor itself. Therefore, we created classes for each of these objects representing their functionality. Moreover, to work with hexagonal modules, a Hexagon class was created representing some general properties of a hexagon.

Motor	
Attributes	**Methods**
pins	set_state
position	
state	

Module	
Attributes	**Methods**
position	set_action
motors	encode_sr_byte
action	is_below_package

Hexagon	
Attributes	**Methods**
is_flat_top	set_diffs
id	set_drawing_points
coord	set_center_position
diffs	get_direction
points	is_neighbor
position	

ShiftRegister	
Attributes	**Methods**
data_pin	pulse
clock_pin	shift_out
latch_pin	clear
daisy_chain	
Delay	

Omniveyor	
Attributes	**Methods**
hexagons	update_sr_data
num_of_modules	actuate
modules	update_module_actions
sr_data	create_hexagons
delay	get_neighbors
	get_nearest_index
	get_path_indexes

3.2.3 Machine Learning

The steps involved in package sorting using reinforcement learning are shown in Figure 10.

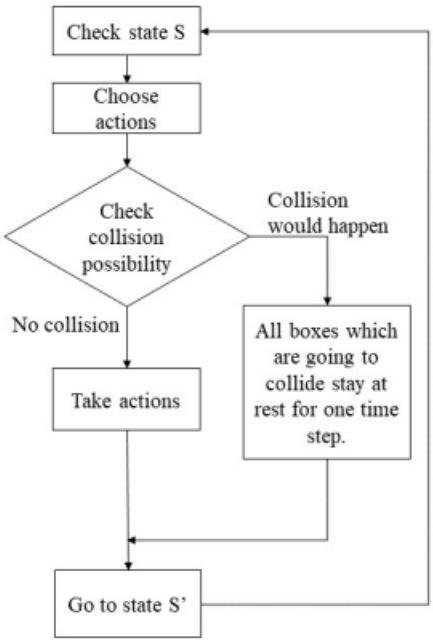

Figure 10: Sorting Q-Table

In this case, there are two types of collisions. Due to the large number of packages, if any collision occurs, all the packages involved will not move for one time-step. The following flowchart demonstrates this process.

The RL agent begins by assessing the current state of the system, including the current positions of the packages and their intended sorting destinations. Based on this information, the agent selects actions that will yield the highest rewards and checks for the possibility of package collisions. If no collision is expected, the actions are executed. However, if collisions are predicted, all packages that are likely to collide will remain stationary for one time-step. The algorithm then re-evaluates the system during the next time-step.

3.2.4 Image Processing

The image processing techniques required include image preprocessing to eliminate uneven illumination and shadows in low light conditions, contour detection of carton, calculation of the carton dimensions and the tilt angle, as well as identification of the actuators underlying the package.

The bulk of the work will be done by the cross-then-activate method shown in Figure 11, proposed by Keek et al. [1] which uses image thresholding and masking to identify which motors to actuate at any given time. In the case where multiple packages must move on the platform at the same time, the Hungarian optimization algorithm can be applied for assignment of motors to cartons.

Our project will be capable of identifying the dimensions of the package and move the packages accordingly to the destination nodes. The system must be able to process real-time to prevent any lag associated with activating the motors. As needed, color markers may be added on the platform for simplicity in the image-processing techniques.

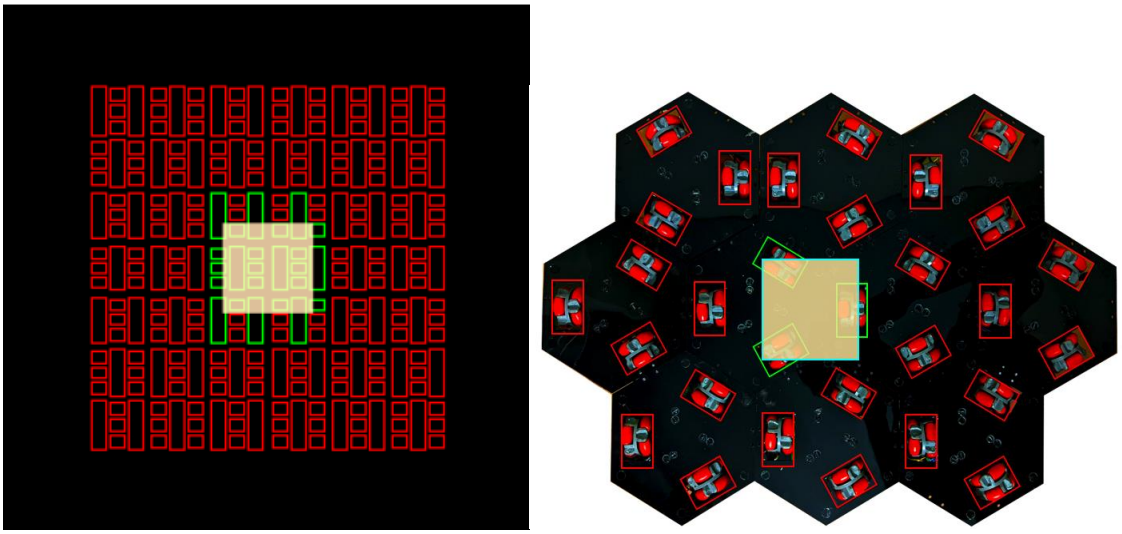

Figure 11: (a) Thresholding and masking to identify underlying actuators [1] and (b) Done in real-time

The open source computer vision library known as OpenCV is an effective tool for computer vision and image processing applications. We have opted to using OpenCV to tracking packages on the omnidirectional conveyor system. Packages can move in any direction on the platform thanks to the omnidirectional wheels. A camera is positioned above the conveyor platform to use OpenCV for tracking. When the packages go along the platform, the camera takes pictures of them.

These images are then processed using OpenCV to detect and track the packages. Finding the packages in the image is the first stage in the procedure. Object detection methods from OpenCV, including Haar cascades or deep learning models like YOLO, are used for this. The image processing features of OpenCV can be used to determine the packages' location and orientation after they have been identified. The parcels are then tracked as they move along the platform using their location and orientation. This is

accomplished by figuring out each package's speed and direction based on where it is in the video feed's subsequent frames. This data can be utilized to forecast each package's location in the following frame and modify the tracking as necessary.

The monitored packages can then be identified and have their data entered into a database or another tracking system. This enables warehouse managers to keep an eye on how goods are treated properly as they move across the facility. For tracking packages on an omnidirectional conveyor system, OpenCV is a crucial tool. It can effectively identify and track packages using object identification and image processing methods, enabling efficient and dependable package handling. Shown below are some stages of the image processing taking place.

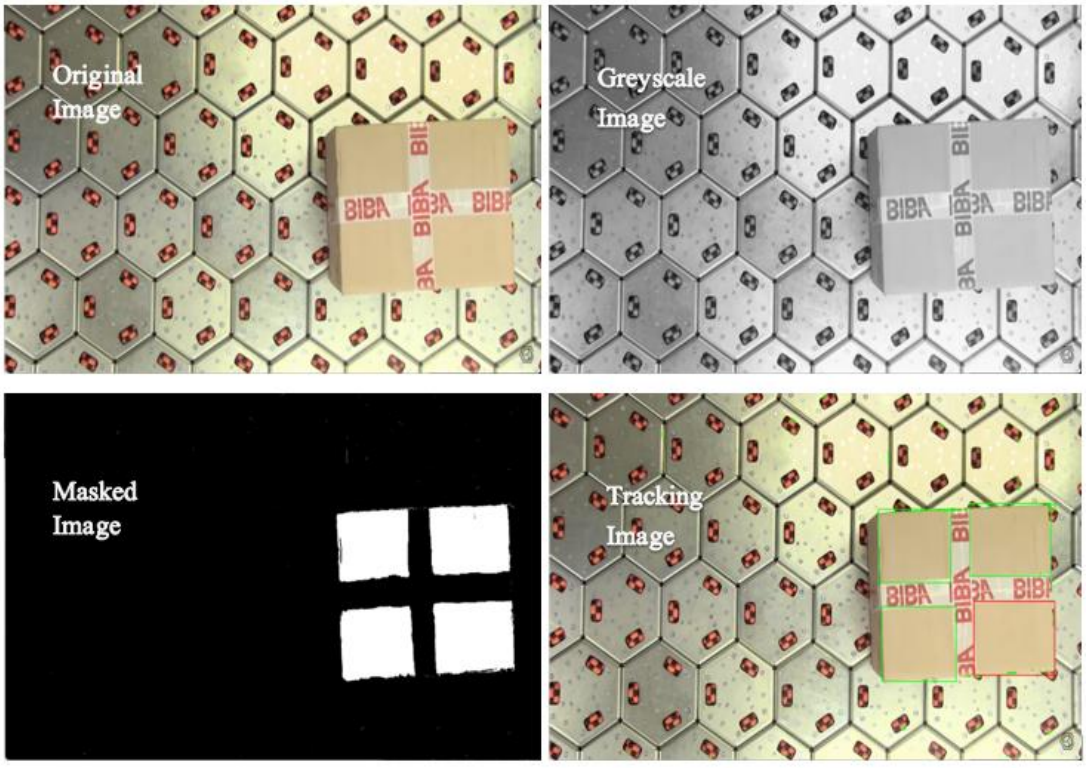

Figure 12. OpenCV testing

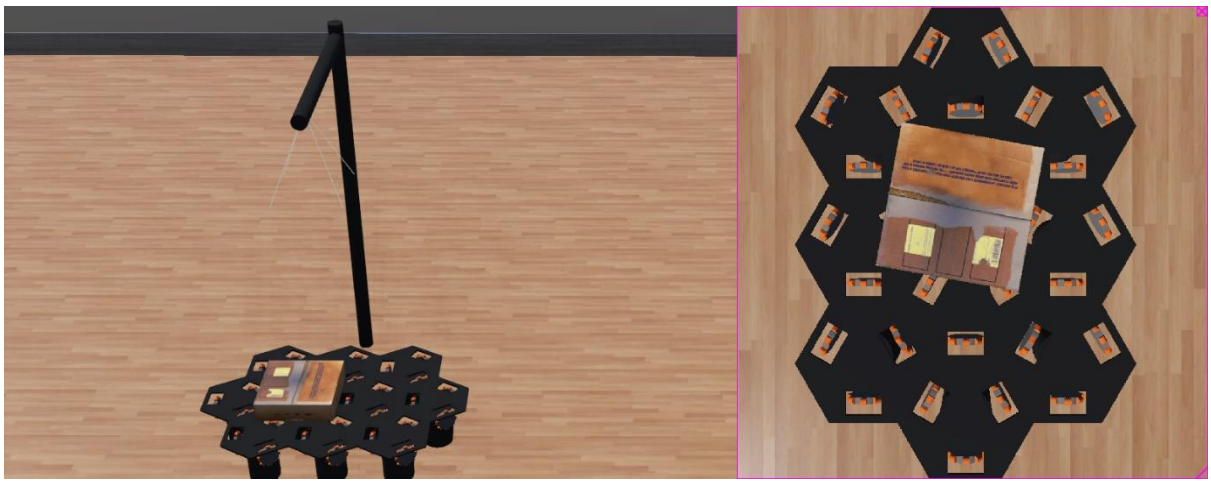

Figure 13. Camera mounting and point of view in WeBots

3.2.5 Graphical User Interface

Graphical User Interfaces (GUIs) are useful to an omnidirectional platform system in several ways. GUIs provide a visual interface for controlling and interacting with the system. With a GUI, operators can easily monitor the system's status and control its movements and functions with the click of a button. This makes it easier to operate the system and reduces the risk of errors or accidents.

For our project, we created a dashboard shown in Figure 14 that can send commands to the Omniveyor and show the live video feed. The tools used in creating the GUI were mainly Python and the Tkinter library. The Tkinter library is very detailed and can be used to create complex user interfaces. An OOP implementation was preferred to keep the code maintainable and easy to modify.

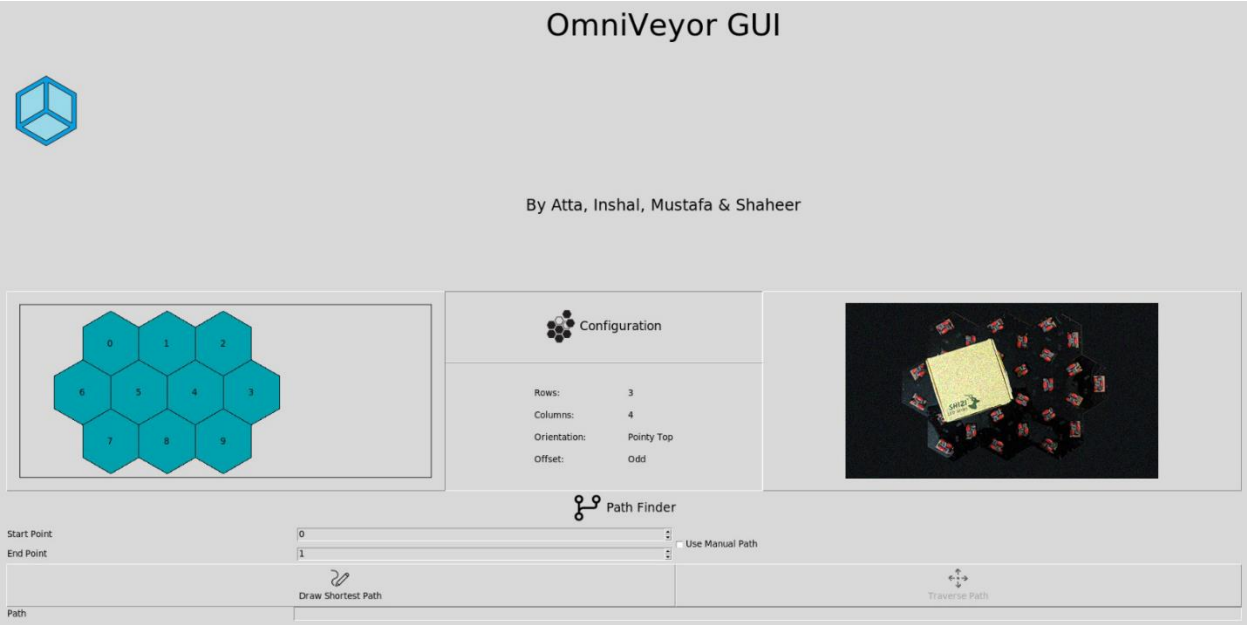

Figure 14: GUI

The GUI provides diverse options such as providing a start and end point to traverse. The GUI draws the shortest path on the screen and the package then travels this path on the Omniveyor. Another option in the GUI is to draw manual path which can be used to trace an arbitrary path using user input and then the path is moved by the package.

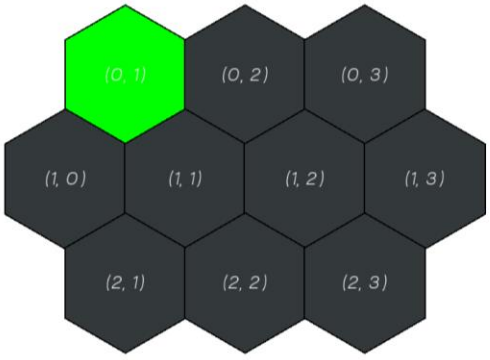

Figure 15: Modules arrangement in GUI

Furthermore, this GUI can be customized to suit the specific needs of the user as shown in Figure 17, the GUI can reform to display any possible arrangement of the modules. The interface can also be designed to display only the information that is relevant to the task at hand. For capturing the video stream, the frame

must be updated at regular intervals. For this a class called CameraStream was implemented which returns a frame to the GUI after regular intervals.

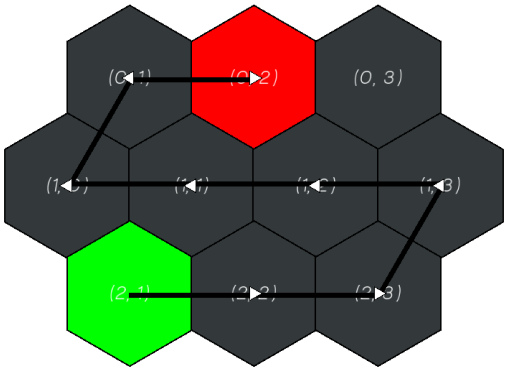

Figure 16. Path tracing on GUI

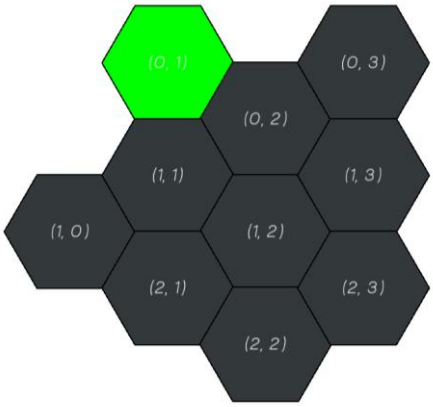

Figure 17. Alternate arrangements in GUI

3.2.6 Simulations

The software used for simulation was Webots since it is free, open-source, and can be used with many programming languages such as C, C++, Python, and MATLAB. Another advantage of Webots is the vast library of objects which can be directly imported to streamline the simulation process. Since we are using a complex hexagonal tri-wheel design, the CAD model of the hexagon and omni-wheel was first developed

in SolidWorks to adhere to the accurate dimensions. We followed the dimensions in Figure 18 16, keeping $L = 100mm$ and $r = 58mm$. The radius of the wheels was chosen to be $48mm$ which is the size we will be using in our hardware.

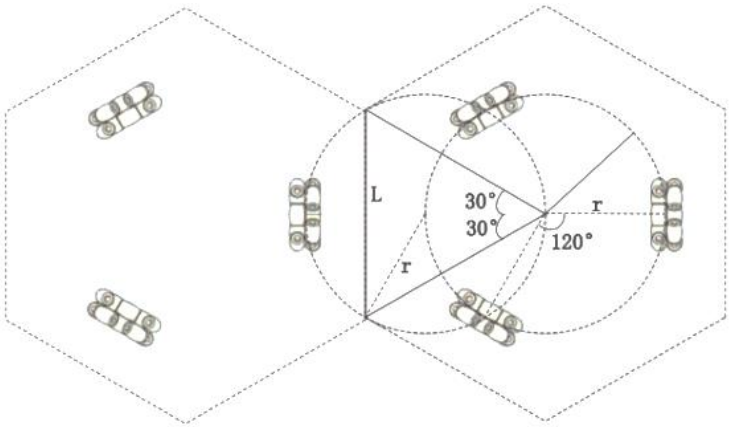

Figure 18: Hexagonal module dimensions [3]

In Webots R2022b, the CadShape node has been added which allows importing designs in the OBJ format. MeshLab was used to convert the STL file from SolidWorks to an OBJ file which could then be imported to Webots. For the omni-wheel implementation, we opted to use the omni-wheel tutorial provided by the Cyberbotics documentation [8] and modified the design to our size and needs. Similarly, for the carton, we used the CardboardBox object found in Webots Projects [9] and modified the size and mass to our application. The models of hexagonal module and cardboard box are shown in Figure 19.

Figure 19: Model of (a) Hexagonal module and (b) Cardboard Box in Webots

We grouped 10 hexagonal modules in Webots to create a small conveyor shown in Figure 20. Running the simulation showed that the package was capable of omnidirectional movement on the platform.

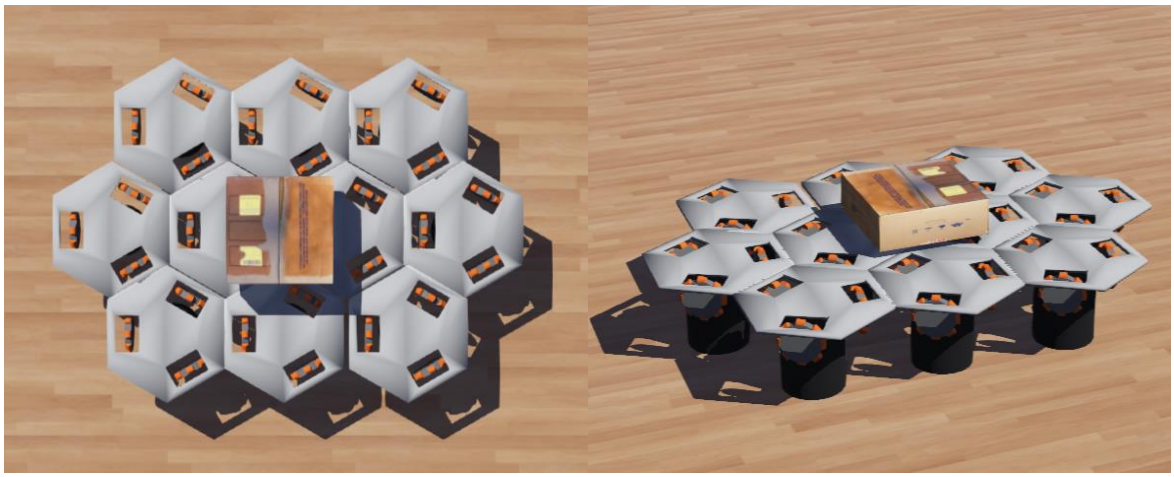

Figure 20: Conveyor in Webots: (a) Top View and (b) Perspective View

3.3 Chapter Summary

In this section we have discussed every aspect of the project in detail, from the hardware design to the software methodology and its integration with the GUI. The hardware uses omni-wheels paired with DC motors to move packages on its surface. Each individual module has a magnetic mounting system that

allows for easy reconfigurability. The control system consists of a Raspberry Pi, paired with motor drivers and shift registers which allow for data transfer serially. This drastically reduces the complexity of the hardware, as now all data is transferred through a single pin, rather than six pins per module, making the platform infinitely expandable. The details of the code to drive the motors is discussed in section 3.2.2.

Image processing is used to track the movement of the package, along with its color, size and all necessary data which can be used to sort the package. Computer vision also enables optimization for motor control.

For avoidance of collisions and path planning, reinforcement learning is used. Specifically, a Q-Learning model has been developed which allows for expansion of the platform size without requiring the need to rewrite or hard-code any pathways for the packages.

The computer vision and path planning come together in the GUI where the user has complete control of where he would like to send the package, the user can either draw any arbitrary path for the package to follow or use the built-in shortest path command which automatically draws up the shortest path between the specified start and end points.

To test the image processing capable machine learning model and its integration with the hardware we used Webots simulator. Codes for path planning and motor control were all tested thoroughly using this simulator. After which we moved over to the hardware fabrication and its testing.

4 Results, Discussion, and Troubleshooting

In this section we will discuss all of our results obtained through testing our hardware and software. Then a brief discussion on what led to such results, followed by the troubleshooting preformed to make the project work.

4.1 Hardware

For the fabrication of the hardware, we used 3mm acrylic sheets, which were laser cut to our exact dimensions. For the top plate we had to cut a few holes for mounting of the hardware, this is better represented in Figure 21: Top plate . The bottom plate is essentially exactly like the top plate, just without the motor cut-outs and mounting holes.

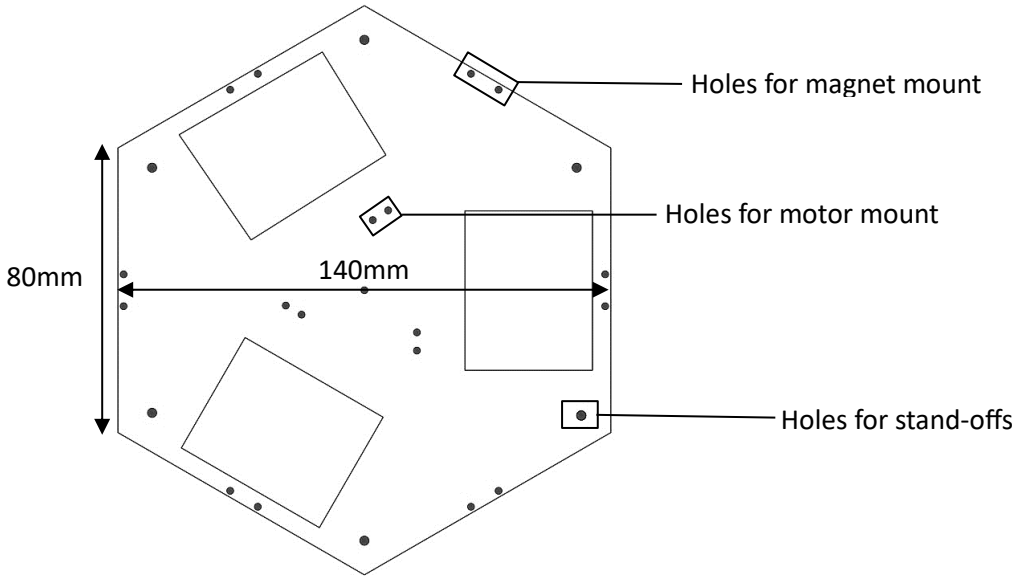

Figure 21: Top plate

After assembling all the hardware, we came to realize that is would have been easier and more efficient to 3D print the stand-offs between the plates, rather than using adjustable threaded 8mm metal rods. This made the fabrication process quite tinker-some.

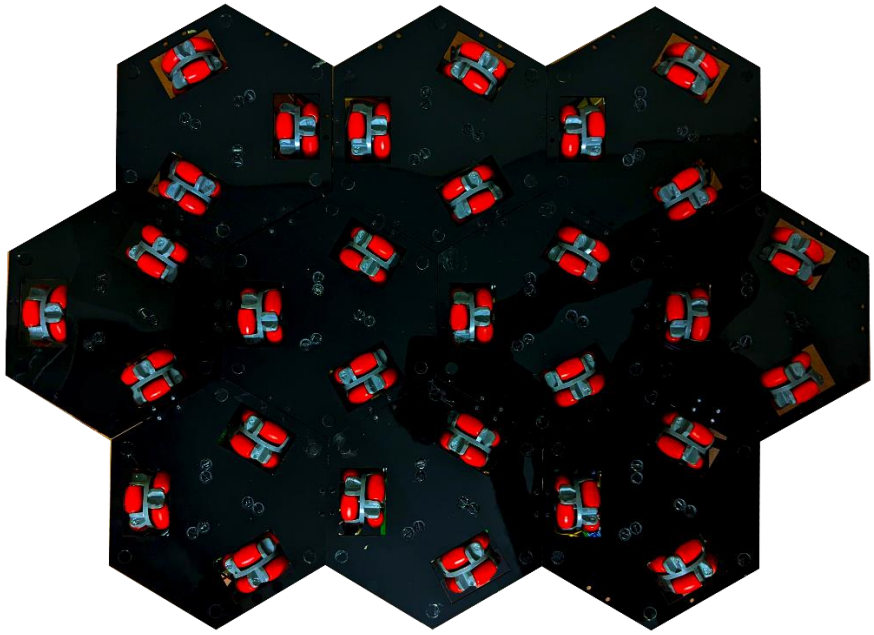

Figure 22: Top view of complete hardware

After testing all the motor drivers directly by supplying 5V from our 20A power supply, we realized that the voltage fluctuations were a problem, hence we fitted Buck convertors to each motor driver, which regulated the voltage at 5V to each module and this method worked just fine.

4.2 Machine Learning

The results from our machine learning tests conclude that Q-Learning would be most suitable for our use case. Q-Learning is a reinforcement learning algorithm that is well-suited for problems where there is no prior knowledge of the environment and the agent needs to learn through trial-and-error. Since our system is ever expanding and highly flexible, the data shows that Q-Learning specifically would be the easiest and quickest model to train for our application. No matter what the size of the platform is, the code will remain the same. It works by updating the expected reward of taking an action in a particular state, based on the observed reward and the expected reward of the next state.

Shown below in Table 6 are the results of the comparisons between all the different types of reinforcement learning models.

Parameters	Q-Learning	Double Q-Learning	Deep Q-Learning
No. of training runs	5	5	1
No. of episodes	10000	10000	10000
Time steps per episode	1000	1000	100
Learning rate	0.1	0.1	0.00005

Table 5: Algorithm parameters

Parameters	Q-Learning	Double Q-Learning	Deep Q-Learning
Attempts	19,709	19,679	7650
Time steps/episode	5.963	6.042	6.907
One collision	0.27/episode	0.29/episode	0.15/episode
Two collisions	0.013/episode	0.017/episode	0.09/episode

Table 6: Results for all Reinforcement learning algorithms

4.3 Chapter Summary

In this chapter we have discussed the results of our testing, including both software and hardware. The entire process has been thoroughly outlined above, including the specific challenges we faced and the corresponding solutions we developed to overcome them. The outcomes of our assessment for reinforcement learning algorithms are presented, and based on the results, we have concluded that Q-Learning is the most suitable option for our use case. Furthermore, we have provided detailed justifications for our choice of Q-Learning by analyzing its effectiveness and efficiency in addressing the challenges specific to our project.

5 Conclusion and Future Work

This chapter presents the conclusion of this project along with the potential directions in the context of future work.

5.1 Conclusion

The project was completed successfully within the given time period, and the team gained a considerable amount of knowledge and experience in the process. The project involved the acquisition of several new concepts, and the use of innovative techniques to solve various challenges. This not only makes the project more novel but also ensures that it is tailored to meet the needs of different logistics companies.

At present, the project has reached a state where it can be effectively commercialized and marketed to different logistics companies. With its unique features and ability to streamline various processes, it has the potential to transform the logistics industry and provide businesses with a competitive edge. Overall, the project has been a valuable learning experience for the team, and it has great potential for success in the market.

5.2 Future Work

Since the boom of E-Commerce we have seen a large increase in number of warehouses world wide. Within Pakistan we can see that a lot of online businesses and marketplaces have evolved and with it increases the number of ware houses. Recently Daraz has launched a smart warehouse in Karachi. This shows the need for in house package management utilities and inter warehouse package handling. The project is targeted towards such company's which need suitable solutions for large, real time package handling like Daraz, Fedex, TCS and others.

Improvements can be made in the microcontroller integration and using different computer vision methods. AI can be integrated for the model to perform tasks by prediction, such as adjusted sorting rate for high and low volume areas.

6 References

[1] J. S. Keek, S. L. Loh and S. H. Chong, "Design and Control System Setup of an E-Pattern Omniwheeled Cellular Conveyor," *Machines,* vol. 9, no. 2, p. 43, 2021.

[2] C. Uriarte, A. Asphandiar, H. Thamer, A. Benggolo and Michael Freitag, "Control strategies for small-scaled conveyor modules enabling highly flexible material," *Procedia CIRP,* vol. 79, no. 2212-8271, pp. 433-438, 2019.

[3] T. Sun, Y. Zhang, H. Zhang, P. Wang, Y. Zhao and G. Liu, "Three-wheel Driven Omnidirectional Reconfigurable Conveyor Belt Design," in *Chinese Automation Congress*, Hangzhou, China, 2019.

[4] J. M. Carew, "What is Reinforcement Learning?," 2021. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning#:~:text=Reinforcement%20learning%20is%20a%20machine,Eye%20on%20Tech. [Accessed 22 11 2022].

[5] W. ZAHER, A. W. YOUSSEF, L. A. SHIHATA and E. AZAB, Omnidirectional-Wheel Conveyor Path Planning and Sorting Using Reinforcement Learning Algorithms, 2022.

[6] D. Dyke, "Grabcad," [Online]. Available: https://grabcad.com/library/omni-wheel-41x29mm-diameter-x-width-1.

[7] [Online]. Available: http://wiki.sunfounder.cc/images/f/ff/L293D_schematic.png.

[8] Cyberbotics, "Webots documentation: Factory," [Online]. Available: https://www.cyberbotics.com/doc/guide/object-factory#cardboardbox. [Accessed 20 11 2022].

[9] Cyberbotics, "Webots documentation: How To," [Online]. Available: https://cyberbotics.com/doc/guide/samples-howto#omni_wheels-wbt. [Accessed 20 11 2022].

[10] Mujju, "https://www.majju.pk".