STEVENS

INSTITUTE OF TECHNOLOGY

1870

ME 598

INTRODUCTION TO ROBOTICS

FALL 2022

AUTONOMOUS OBJECT SORTING USING A MOBILE ROBOT

12/18/2022

ABSTRACT:

Robots are often used to complete tasks in a preset order. A simple algorithm-based command can get the robot to do the task; however, it becomes more complicated when the work requires decision-making. To minimize the manual effort needed, the robot must attain autonomic (autonomous nature), i.e., it must be able to make decisions based on the circumstances at the time. This may be accomplished by incorporating conditional mechanisms that allow the robot to select a path from a set of predetermined options based on the condition's outcome. It may be further simplified, and the robot can become autonomous by adding and utilizing appropriate sensors and data. The robot is said to be autonomous when these requirements are defined so that all conceivable processes are completed without mistakes.

This project intends to mimic an autonomous mobile robot under predetermined conditions and accomplish tasks without human interaction during execution. This work was completed in MATLAB using Simulink and State-flow. The algorithm designed for the task was sufficiently precise, and the robot obtained autonomy in sorting the pucks into their proper zones. The average trial time to sort both pucks and return to the origin was roughly 60 seconds. The team worked together to develop and implement the algorithm. Once the algorithm was defined, everyone had tasks to complete, and they were merged and tested in all potential starting conditions. According to the team, the model's performance is excellent since it precisely identified the color of pucks and placed them in their corresponding zones. The model is easily adjustable if the beginning conditions change, and it may be made more independent by scavenging the arena and sorting whatever number of things it finds.

INTRODUCTION:

A mobile robot can travel from one location in the real world to another to fulfill various activities. In contrast, Autonomous Mobile Robots (AMR) can comprehend and navigate their surroundings. The AMR can navigate any terrain without the need for human interaction. This is possible thanks to sophisticated sensors and mapping techniques allowing AMRs to adequately grasp and recognize their environment. The primary goal of this project is to move the robot and localize its path in a virtual environment for object sorting. We're guiding our AMR to find the things and transfer them to a particular location (Target). Specific requirements apply here, such as AMR navigating the colored pucks into a like-colored target within the arena. This will result in the color-detecting sensor on the AMR can accomplish. To satisfy the project objectives, an initial simulation file with the preset environment and arena settings was provided, which should be utilized as it is for the "available robot hardware.". There are three target places in the arena, each with a distinct color allocated to it. The arena's origin is set at [0 0], and the arena walls are each 5 meters long. The robot will always begin at [-1.5 0], and there will always be two pucks at [0, 1.5] and [0, -1.5]. Each of the two pucks will be red, green, or blue, with only one of each color present, albeit the order in which the pucks appear is arbitrary. The primary goal of this project is to accurately find the pucks and sort them into their appropriate zones based on their color (the red object goes to the red zone, and so on).

After sorting the objects into their corresponding zones, the robot must return to the origin (0,0). It is of utmost importance that the robot operates with care and responsibility, ensuring it does not collide with or scratch any walls, thereby maintaining its safety and the integrity of the environment.

THEORY AND EXPERIMENTAL PROCEDURE:

The robot's tasks may be repeatedly moving from point A to point B in an arena. And, because the robot only travels on the ground in a fixed environment, the environment is assigned a two-dimensional coordinate system. It is simple to complete the immediate goals and establish what the robot should perform using fundamental geometry and graph theory in the algorithm. Turning left or right, moving forward or backward, and stopping are the four operations of the robot. A pivoting motion is made for turning left or right by supplying the same speed but in opposing directions, and for going forward and backward, left and suitable motors are given the same speed.

PART 1: ARENA LAYOUT AND ROBOT OBJECTIVE:

The arena is 5x5 m in dimension and has three colored zones (blue, green, and red). It is provided with:

I. The two pucks will always be present at [0, 1.5] and [0, -1.5].
II. The robot will begin at the coordinates [-1.5 0].
III. Two zones are located at the corners of one side and the middle of the opposing side. Each zone is of size meter.

Because the objective is to sort the two pucks, the following stages are described in a predetermined algorithm specifying robot actions: The Robot will take the following route:

I. Begin at the coordinates [-1.5, 0].
II. Move to the first puck at [0, 1.5].
III. Determine the color and the zone in which the puck must be put.
IV. Avoid colliding with the puck as you push it to the allocated zone.

V. Return to the second puck at [0, -1.5] and repeat steps (iii) and (iv).

VI. Return to the arena's origin at [0, 0] and come to a halt.

PART 2: ROBOT ATTRIBUTES:

I. In all, the robot has four sensor inputs:

II. The color- detecting camera is mounted on the robot's head.

Three distance/proximity sensors are located on the robot's head, left, and right sides.

The following proximity sensor properties are fixed:

I. Maximum range = 2 m

II. Minimum range = 0.02 m

III. Resolution = 0

PART 3: METHODS AND ALGORITHMS:

When the project library is opened, a picture of the block diagram is attached below.

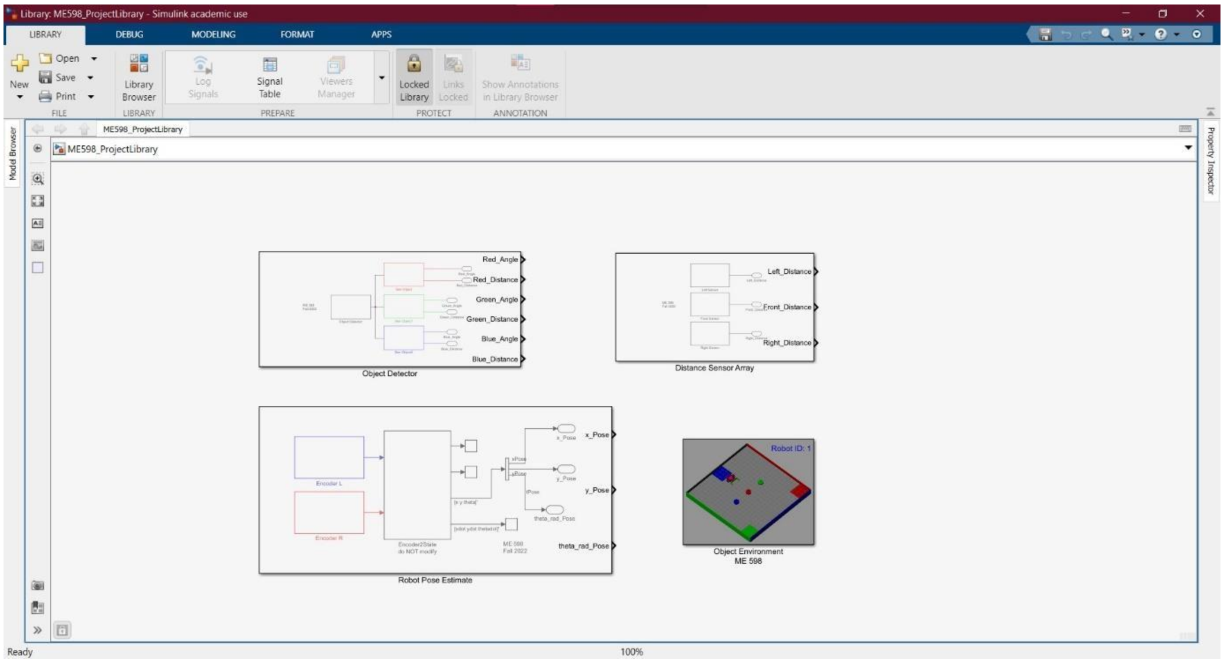


FIGURE-1 ROBOT LIBRARY

First, a model with just a single MATLAB function is created to solve the problem, achieving the robot's desired functionality. The model is partially automated but well-suited for this project. The same result will appear despite how many times the code runs.

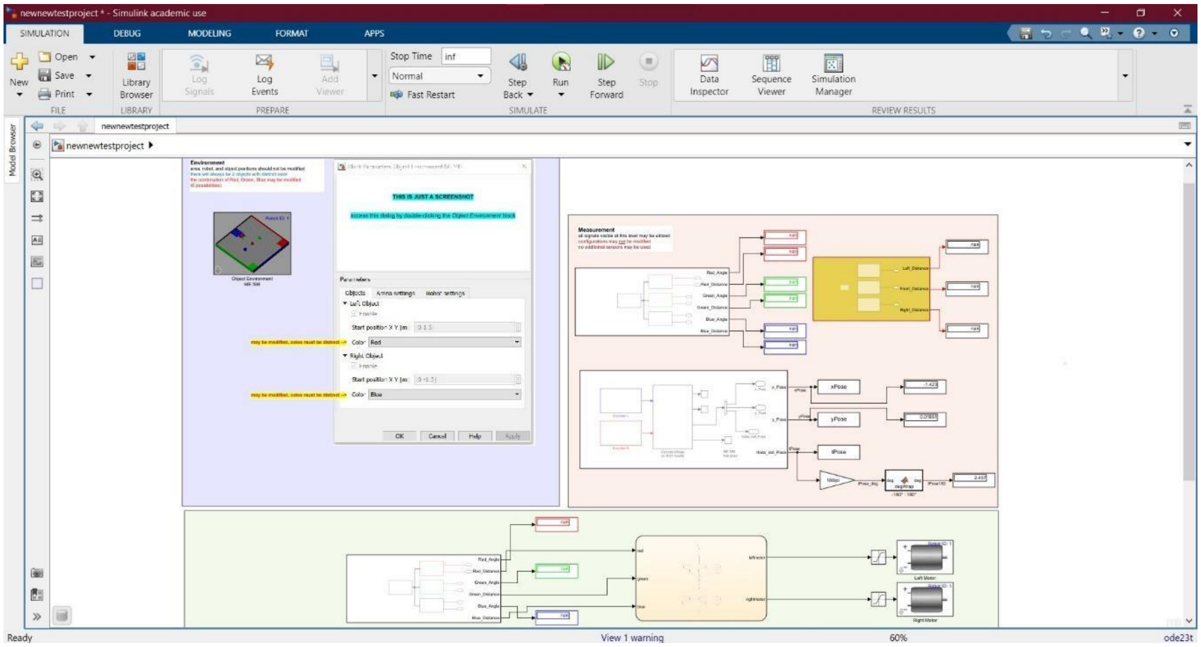


FIGURE 2: SNAPSHOT OF THE SIMULINK MODEL WITH A STATEFLOW CHART

Certain conditions must be met to achieve the robot's actions, and the simulation state must be defined. As the actions are repetitive, they can be modularized into MATLAB functions.

The experiment, a testament to the significance of Stateflow, was performed entirely using this tool. It commenced by identifying the color of the puck and then calculating the distance between the robot and the puck, all of which were implemented in Stateflow.

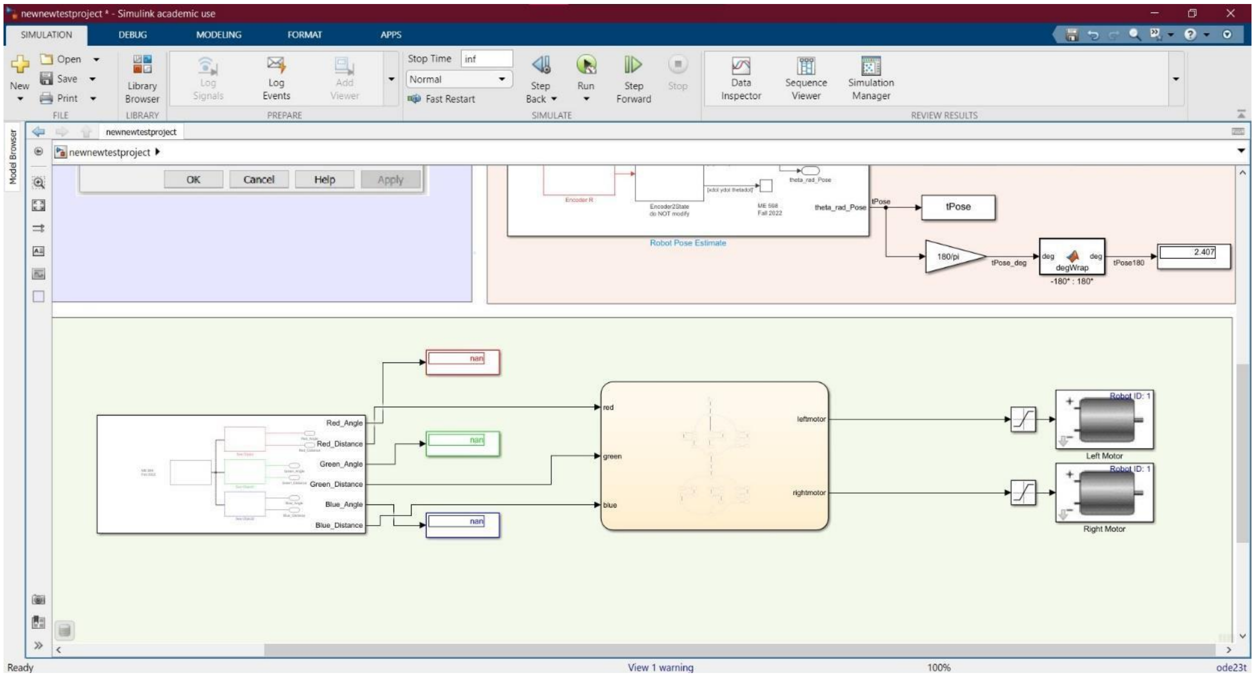
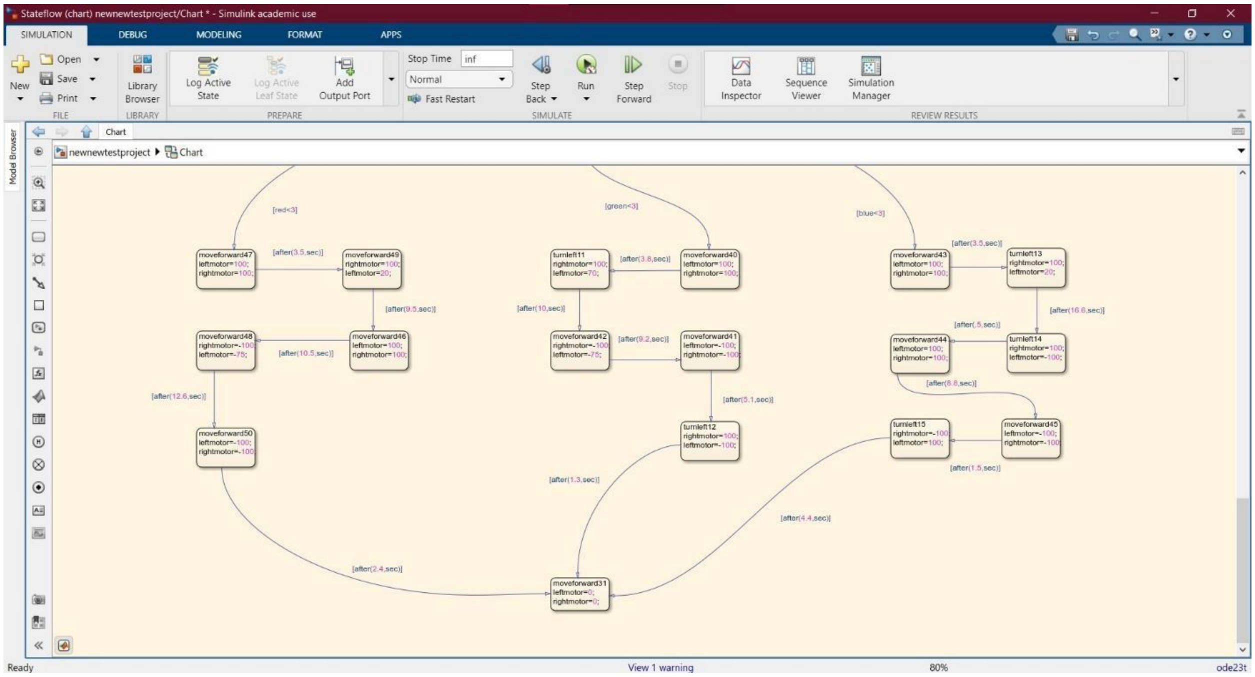


FIGURE 3: SNAPSHOT OF THE SIMULINK MODEL

When the state flow block is expanded, the flowchart appears, which defines the robot's path depending on the puck's color. The snapshot of the flowchart is attached below.



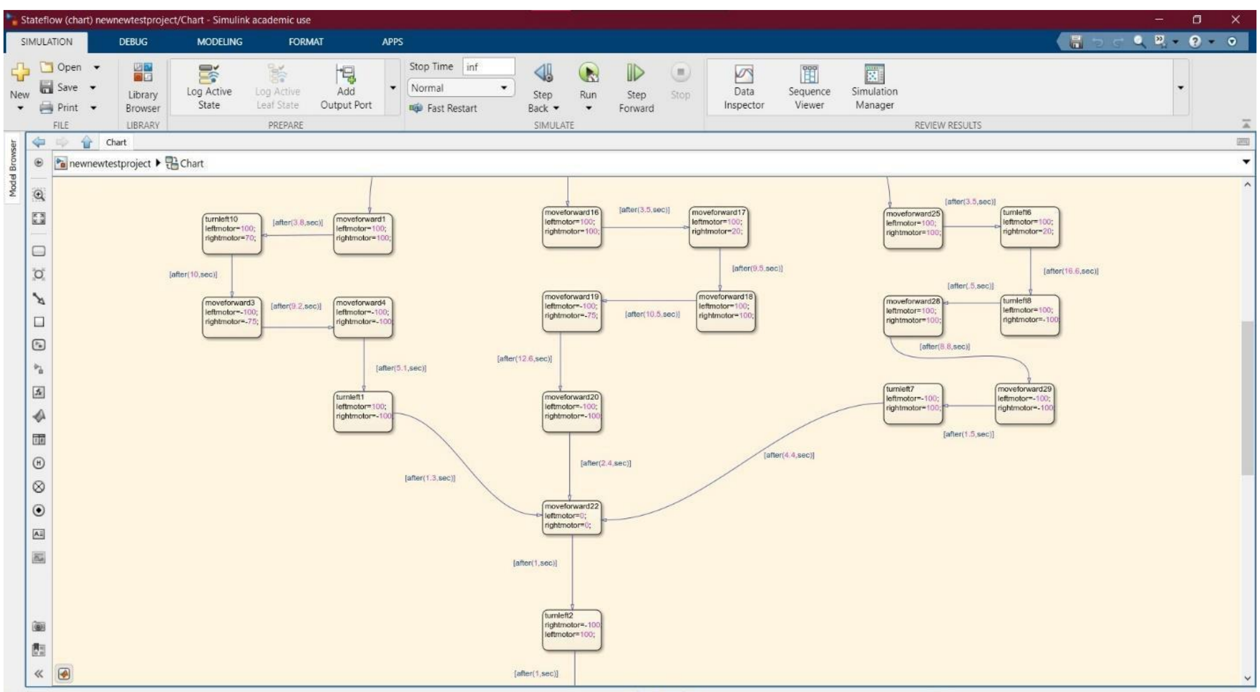


FIGURE 4: FLOWCHART

PART 4: RESULTS & DISCUSSION:

A. ROBOT TRIALS:

TRAIL 1:

In this trail puck, one was red, and the other was blue, as seen below. The robot successfully sorted the objects into their designated zones, as seen in the figure in 73 seconds. Additionally, there was no collusion between the robot and any obstacles, and the robot's trajectory was smooth, free of any jerky motions. The graph confirms the results as the robot returns to the home position after sorting.

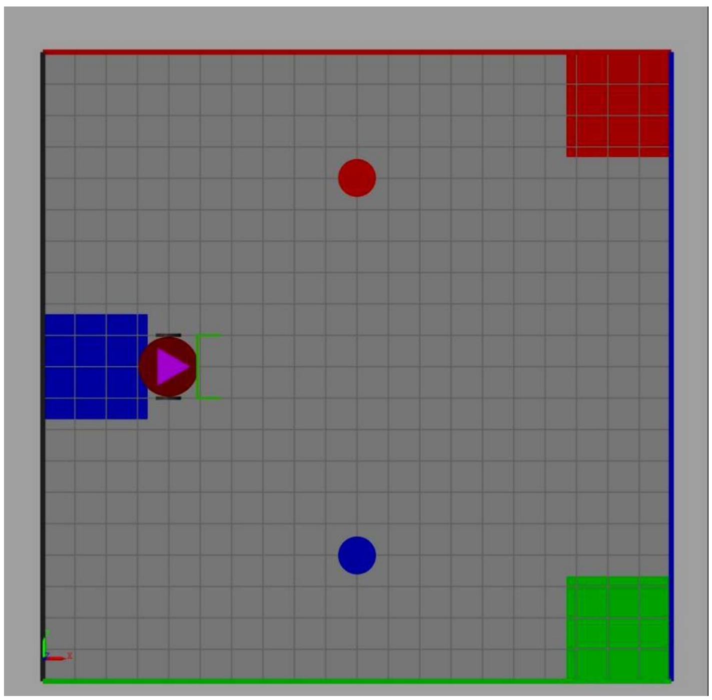


FIGURE 5-A: ARENA BEFORE SORTING THE PUCK FOR TRAIL 1

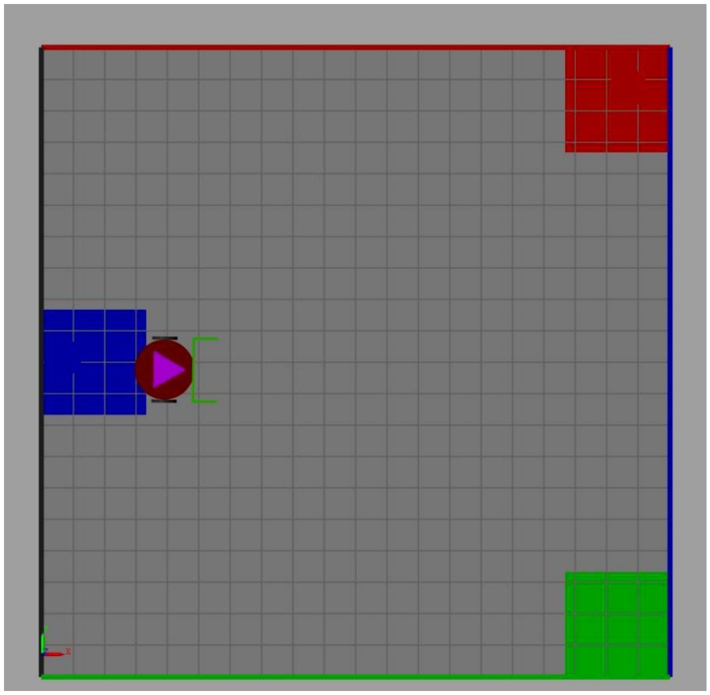
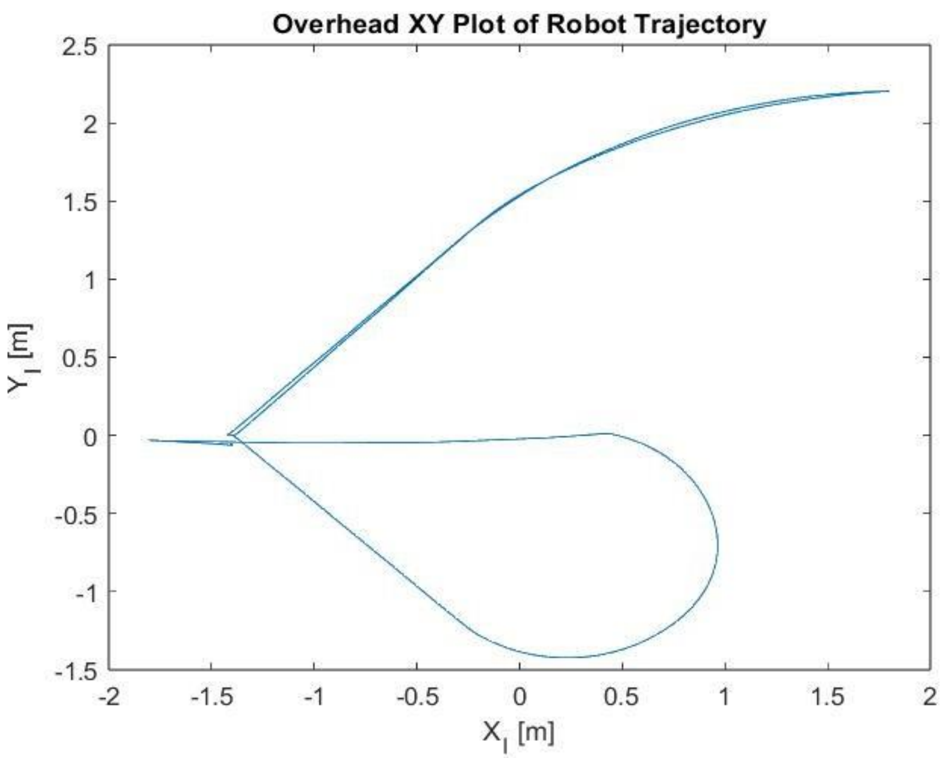


FIGURE 5-B: ARENA AFTER SORTING THE PUCK FOR TRAIL 1



GRAPH-1: ROBOT'S TRAJECTORY FOR THE TRAIL 1

TRAIL 2:

In this trail puck, one was blue, and another was red, as seen below in the figure. The robot successfully sorted the objects into their designated zones, as seen in the figure in 81.2 seconds. Additionally, there was no collusion between the robot and any obstacles, and the robot's trajectory was smooth, free of any jerky motions. The graph confirms the results as the robot returns to the home position after sorting.

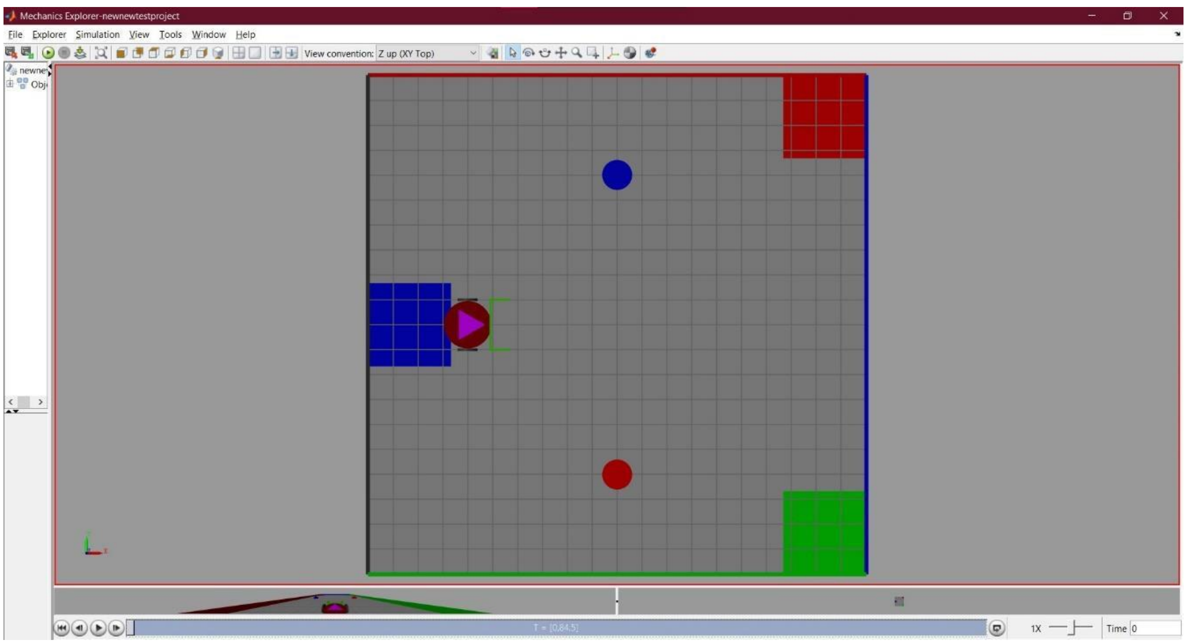


FIGURE 6-A: ARENA BEFORE SORTING THE PUCK FOR TRAIL 2

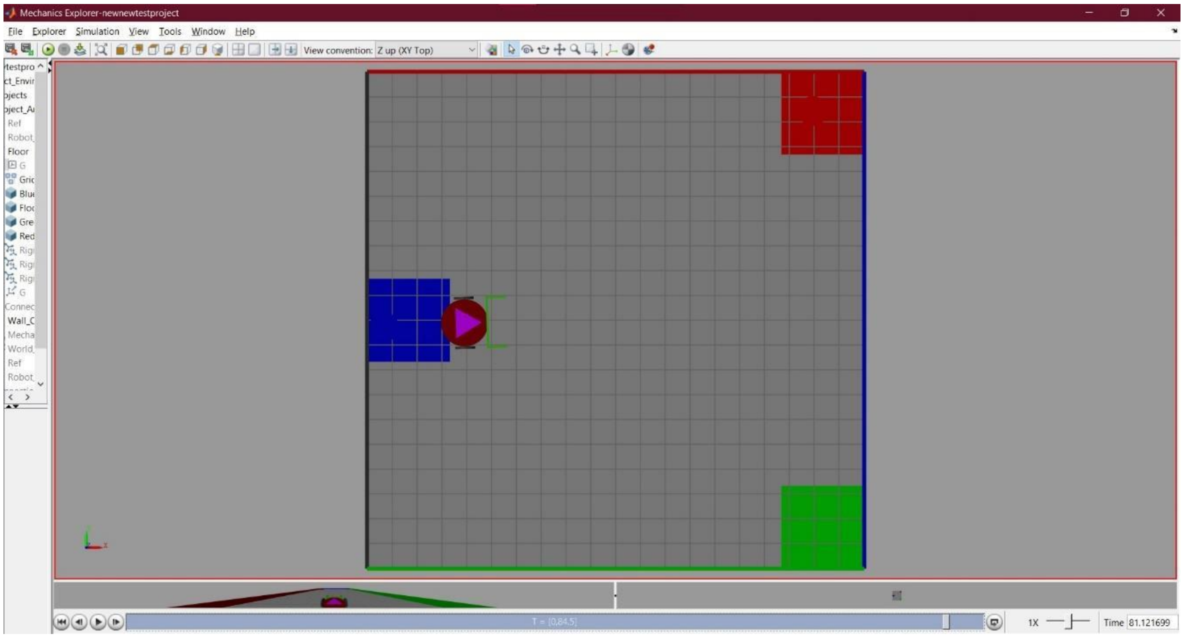
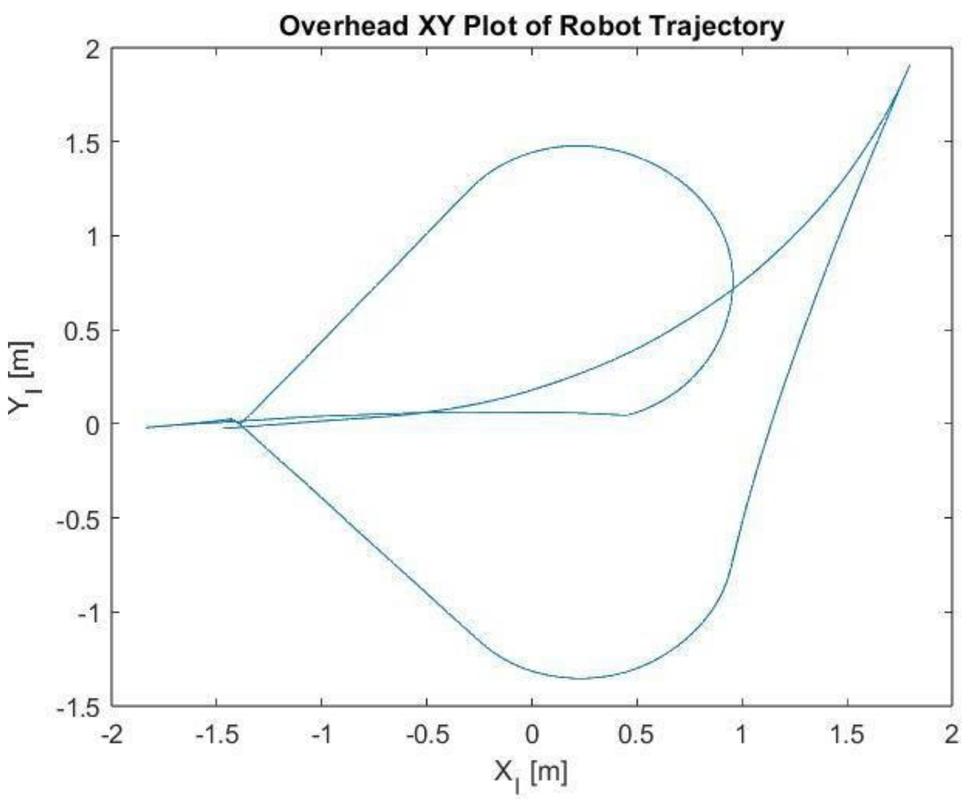


FIGURE 6-B: ARENA AFTER SORTING THE PUCK FOR TRAIL 2



GRAPH-2: ROBOT'S TRAJECTORY FOR THE TRAIL 2

TRAIL 3:

In this trail puck, one was red, and the other was green, as seen below in the figure. The robot successfully sorted the objects into their designated zones in 67.6 seconds. Additionally, there was no collusion between the robot and any obstacles, and the robot's trajectory was smooth, free of any jerky motions. The graph confirms the results as the robot returns to the home position after sorting.

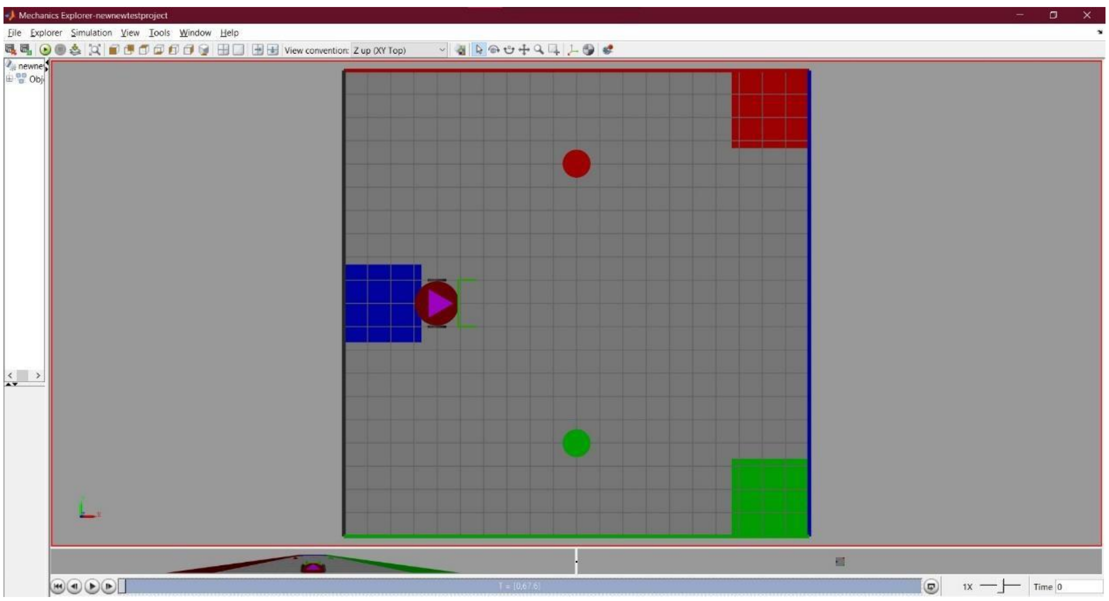


FIGURE 7-A: ARENA BEFORE SORTING THE PUCK FOR TRAIL 3

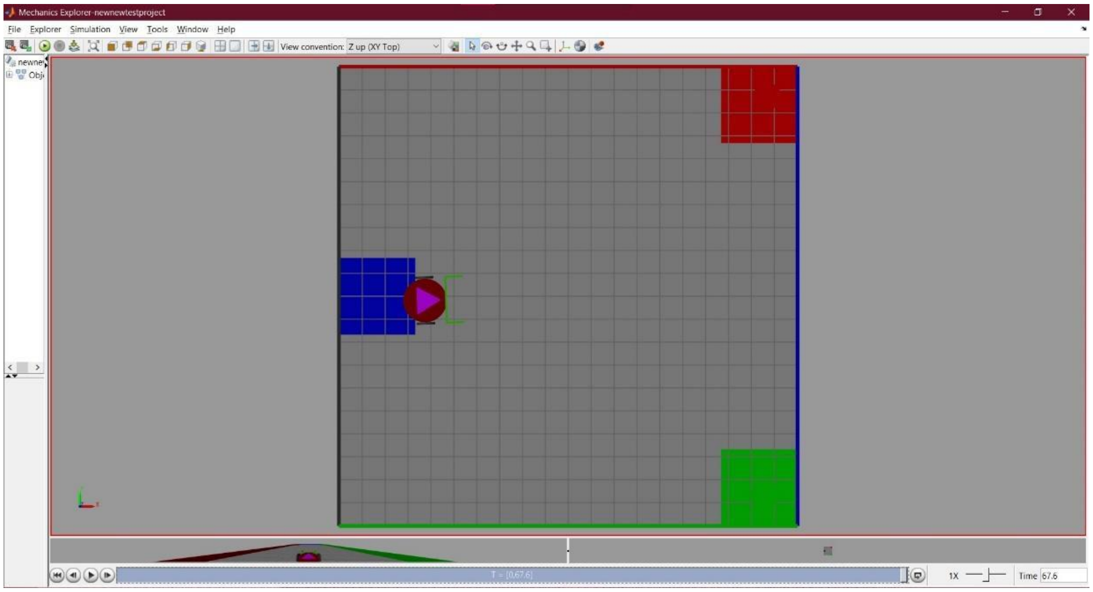
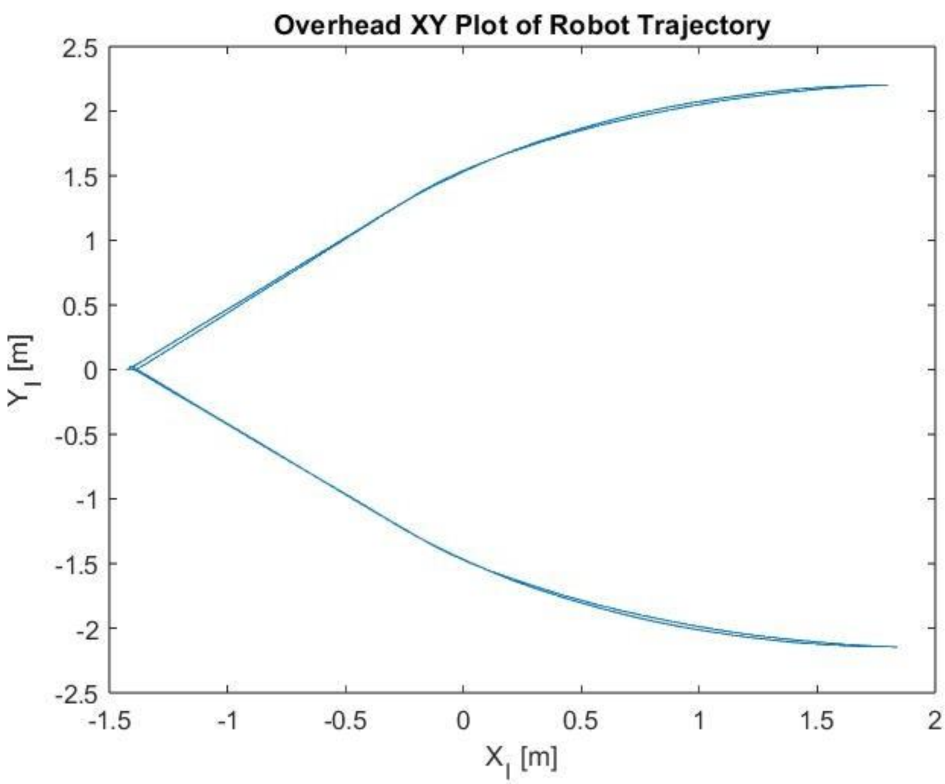


FIGURE 7-B: ARENA AFTER SORTING THE PUCK FOR TRAIL 3



GRAPH-3: ROBOT'S TRAJECTORY FOR THE TRAIL 3

B. ACCURACY AND REPEATABILITY:

From the trial presented in the above section, it can conclude that the robot is highly accurate in the:

- Identifying the color of the puck.
- Correctly identifying the place where the puck must be placed.
- Placing the entire puck in the rightly colored part of the arena.
- Avoiding a collision with the arena walls.

C. VIDEO LINK:

The video link of the trail using state flow is linked below:

https://www.youtube.com/watch?v=istiwEp0FrQ

CONCLUSION:

The primary purpose of this research is to understand the concepts of sorting, path planning, and autonomous mobile robots (AMR) in a digitally generated environment. For this project, a solid understanding of SIMULINK, the robot environment, and the robotic playground Library was required and obtained.

We were able to incorporate a variety of things in this project by utilizing their corresponding MATLAB functions:

1. Avoiding wall collisions with an algorithm we developed.
2. Programmed an autonomous robot to follow a certain course and do specific duties, such as moving a puck and depositing it in the same-colored zone (blue puck in blue zone, for example).
3. We ran six trials using the model we created, and all the data revealed that our robot was performing successfully.

REFERENCES

[1] Introduction to Autonomous Mobile Robots; Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza

[2] MATLAB Examples and API Reference:
https://www.mathworks.com/help/matlab/examples.html?s_tid=CRUX_topnav