Project Report: Autonomous maze-solving holonomic robot.

Shyam Varsani

Suryansh Ankur

Instructor: Martin Jagersand

Date: 15th Dec 2023

Table of Contents

Section 1: Introduction

In the rapidly evolving field of robotics, the ability to navigate through complex environments stands as a critical capability. This report presents our project, an “Autonomous Maze-Solving Holonomic Robot,” designed to autonomously navigate and solve a maze. The project encapsulates a blend of innovative design, precise engineering, and efficient algorithmic problem-solving.

Our robot is not just a product of off-the-shelf components; it is a testament to our commitment to customizability and precision. We designed and 3D printed the omni wheels ourselves, starting with a basic template and then modifying it to better fit our robot’s design. These omni wheels, a crucial element of our robot’s holonomic capabilities, allow it to move in any direction, thereby enhancing its maneuverability within the maze.

The maze, another pivotal aspect of our project, was meticulously designed and built using foam boards cut precisely to form the walls. We aimed to create a structured and adaptable environment for the robot’s navigation. To ensure modularity, we designed and 3D printed connector clips that allowed us to easily assemble the maze and join the maze walls together. This modular approach enabled us to create a versatile and easily reconfigurable structure, accommodating the robot’s navigation and problem-solving capabilities within the project’s timeline and constraints.

This project represents our endeavor to push the boundaries of what is possible with autonomous robots. Through a combination of innovative design, custom-built components, and intelligent algorithms, we have created a robot capable of solving a maze autonomously. This report will detail the design, construction, and operation of our robot, as well as the challenges we faced and the solutions we devised. We hope that our work will contribute to the broader field of robotics and inspire further innovation.

Section 2: Project Overview and Objectives

This project revolves around the design, construction, and programming of an autonomous, holonomic robot capable of solving mazes. The term “*holonomic*” refers to our robot’s ability to move in any direction on a 2D plane, a feature made possible by the specially designed omni wheels. These wheels, which were custom-built using 3D printing technology, provide the robot with superior mobility and maneuverability within the maze. The “*autonomous*” aspect of our robot is achieved through the use of Python and the Lego Mindstorms EV3 kit. The robot operates independently, using sensor data

to make decisions and navigate the maze without human intervention. This autonomous operation showcases the power of integrating hardware and software in robotics.

The primary task of our robot is to solve mazes, a complex task that requires spatial awareness and decision-making. To accomplish this, our robot uses a depth-first search algorithm to explore and map the maze, marking its path as it goes. Once it identifies a solution to the maze, the robot remembers the solution path, and then after being manually reset to the start position, it embarks on a second run. In this second run, the robot follows the predefined path to reach the goal square, optimizing its movement to traverse the maze in the fastest possible time.

The maze, a pivotal aspect of our project, was meticulously designed and built using foam boards cut precisely to form the walls. We aimed to create a structured and adaptable environment for the robot's navigation. To ensure modularity, we designed and 3D printed connector clips that allowed us to easily assemble the maze and join the maze walls together. This modular approach enabled us to create a versatile and easily reconfigurable structure, accommodating the robot's navigation and problem-solving capabilities within the project's timeline and constraints.

All in all, this project represents our endeavor to push the boundaries of what is possible with autonomous robots. Through a combination of innovative design, custom-built components, and intelligent algorithms, we have created a robot capable of solving a maze autonomously. The following sections will delve into the specifics of the robot's design, the construction of the maze, and the implementation of the depth-first search algorithm. We will also discuss the challenges we faced, the lessons we learned, and the potential for future improvements.

Section 3: Omni Wheel Design and Construction

The design and construction of the omni wheels were a crucial part of our project. The process was challenging, involving several steps from initial design to final assembly, but it provided us with valuable lessons and skills.

3.1 Initial Design

We started with a base template of the omni wheels from Thingiverse by RemRC. This template served as a starting point for our custom omni wheels. We modified the axle cutout in the inner half of the wheel, which was initially intended for stepper motors, to fit the Lego axles. To achieve this, we used SolidWorks software. This task was challenging as we had to learn the basics of SolidWorks, a new design software for us.

The learning curve was steep, but we managed to grasp the essential features of the software and modify the wheel design successfully.

3.2 3D Printing

Once we had the design ready, we registered for 3D printer training at the Elko Engineering Garage in the ETLC building. We experimented with different 3D printer types and settings to achieve the desired print for the parts of the omni wheels. This process was very involved and time-consuming. We aimed to obtain a very strong print, requiring an infill of 70% or higher, and as precise as possible, so we used a layer height of 0.1mm.

The 3D printing process was not without its challenges. We encountered many failed prints initially, primarily due to the high infill and low layer height settings. These settings, while intended to ensure a strong and precise print, increased the chances of print failure and significantly extended the printing time. However, through perseverance and fine-tuning of our print settings, we managed to slice our parts optimally. We then printed the parts required in three batches using PLA filament, a common choice for its ease of use and good detail. The images below show the 3D models of the wheel parts.

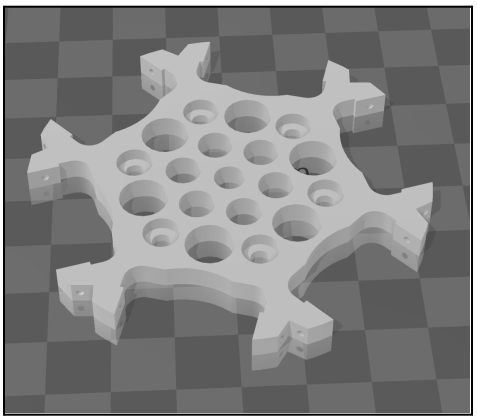

Image 1: Back Rim of The Wheel

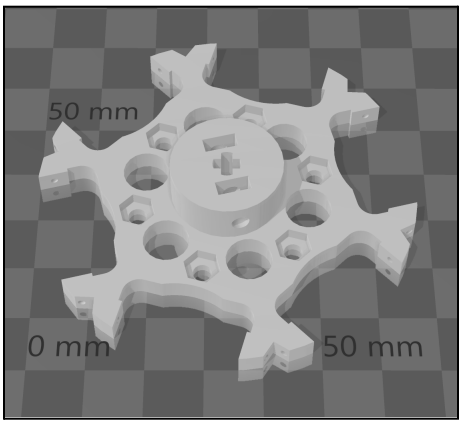

Image 2: Front Rim of The Wheel

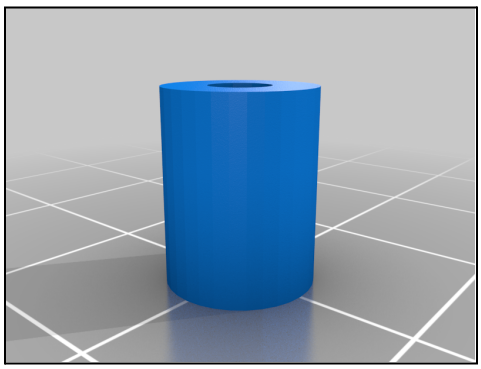

Image 3: Spacer

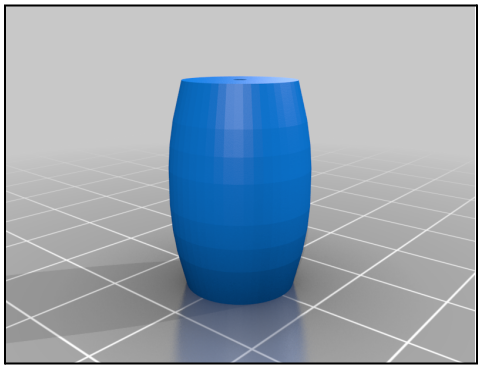

Image 4: Lateral Roller

The images below show some of our failed prints.

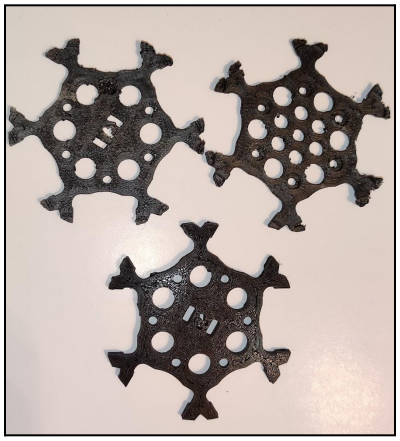

Image 4: Failed Rims

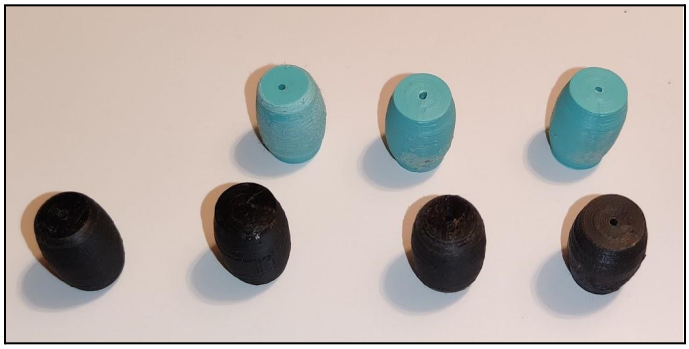

Image 5: Failed Roller prints

3.3 Assembly

With all the required parts printed, we began the assembly of the wheels. We bought 1.25-inch bolts and 4mm nuts, which fit perfectly in the cutouts of the wheel parts. We used files and sandpaper to smooth out any rough edges left by the printer. We bought medium-sized paper clips, about 1.5 inches long and 1/16 inches in diameter, and cut out the long pieces to serve as the axis for the lateral rollers of the wheel. To ensure smooth gliding of the wheels upon this axis, we drilled a hole in the center of each roller using a 1/16th inch drill bit, ensuring the hole was perfectly straight to avoid introducing wobble in the roller. This process was delicate and required precision, and we lost some rollers in the trial and error process. As a result, we had to print more rollers. Once all the rollers had been drilled, we sanded down the roller surface with coarse sandpaper to create a rough surface that would grip the floor better.

We assembled the first half of the wheel by fitting the first six rollers around it, choosing a black and turquoise blue alternate color system for aesthetic purposes. We then similarly fitted the back half of the wheels with the rollers. At this point, none of the roller axes were fit permanently. We used three 1cm spacers to join the front and back halves, ensuring that no roller was rubbing with another roller or the wheel rims. We slid in the three bolts into their cutouts and fastened the wheel using the nuts on the other side. After testing the wheel on the robot, we fixed the roller axes using hot glue. We repeated the same process to assemble the last two wheels.The entire process of designing and assembling the omni wheels from scratch was challenging and involved many setbacks. However, with commitment and dedication, we were able to successfully assemble three omni wheels. This process taught us many valuable lessons and skills, which we will discuss further in the following sections.

The images below show the fully assembled wheels;

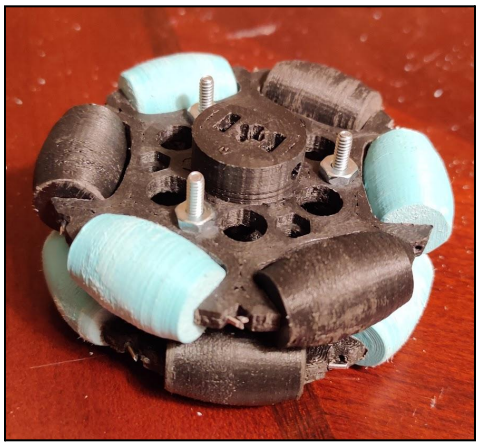

Image 6: Fully Assembled Front

Image 7: Fully Assembled Back

Section 4: Robot Design and Assembly

The design and assembly of our robot were meticulously carried out, drawing inspiration from the writings of Miguel on TheTechnicGear.com, particularly the “Lego Mindstorms Robots” section. Our primary focus was to create a robust and functional robot with a symmetrical structure, ensuring an even distribution of weight across the three wheels.

4.1 Frame Structure

The frame of our robot is a robust triangular structure, designed using the Lego Mindstorms EV3 kit. We initiated the design process by constructing a triangular scaffolding to provide a rigid base for the robot. To this structure, we added a large motor to each side of the triangle. The main EV3 brick was then attached on top, maintaining the symmetry of the structure and further ensuring an even weight distribution.

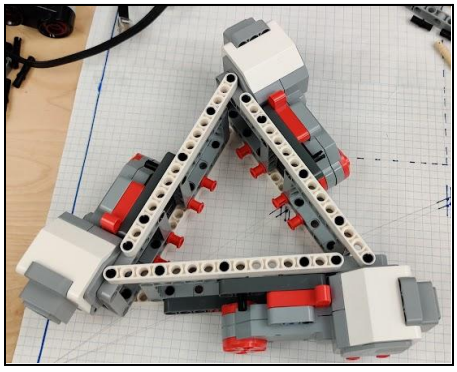

Image 8: Base Frame

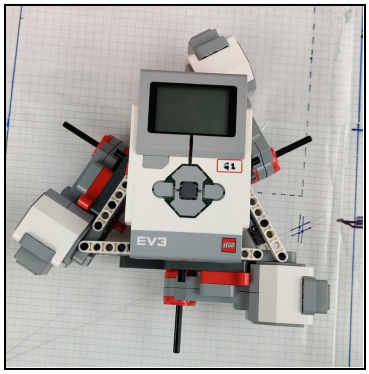

Image 9: Frame With Brick Attached

4.2 Sensor Integration

Our robot is equipped with four ultrasonic sensors, each corresponding to a cardinal direction (North, East, South, and West). These sensors provide the robot with comprehensive environmental perception, enabling it to detect obstacles and effectively navigate the maze. To maintain symmetry, we created a square contraption using available Lego pieces, which was secured to the long sides of the EV3 brick. An ultrasonic sensor was then added to each side of the square contraption. Although we initially considered using a gyro sensor, the four input ports of the brick were already utilized by the ultrasonic sensors, presenting a challenge that we will discuss in later sections.

Image 10: Structure With Sensors

4.3 Omni Wheel Attachment

Upon completion of the main structure, we attached three custom-made omni wheels, which we designed and 3D printed ourselves. These wheels enable the robot to move fluidly in any direction on a 2D plane, providing it with superior mobility and maneuverability within the maze.

4.4 Final Assembly Check

The final step in the assembly process involved a thorough check to ensure that each component was securely attached and correctly positioned to facilitate optimal performance. The culmination of this process was a fully functional holonomic robot, capable of autonomously navigating and solving mazes. The challenges faced, lessons learned, and potential for future improvements will be discussed in the following sections. The images below show the final iteration of our robot design.

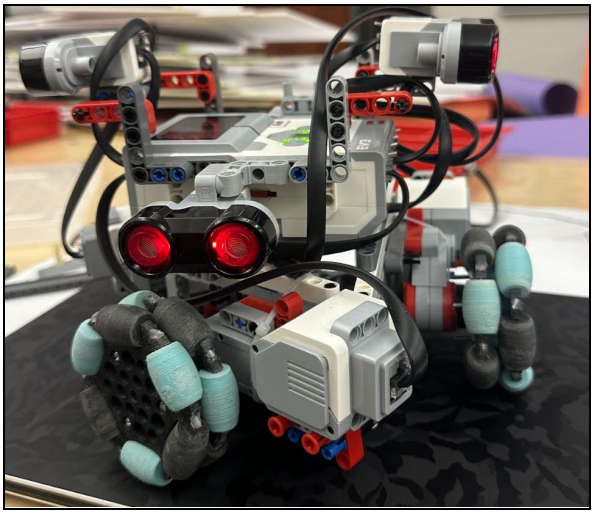

Image 11: Final Iteration Angle 1

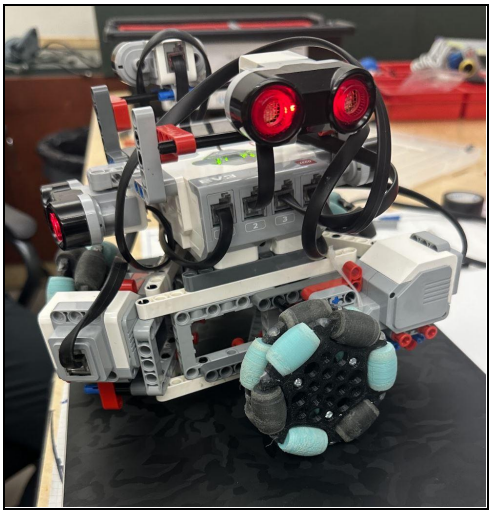

Image 12: Final Iteration Angle 2

Section 5: Mathematics Behind the Robot's Operation

The operation of our autonomous maze-solving holonomic robot is underpinned by a series of mathematical principles and calculations. This section will delve into the mathematics behind the robot's movement, from the design of the omni wheels to the implementation of the movement in Python code.

5.1 Omni Wheels

Omni wheels, or omnidirectional wheels, are a key component of our robot's holonomic movement. These wheels have small discs around the circumference which are perpendicular to the rolling direction. This unique design allows the robot to move in any direction without turning. The ability to move laterally, diagonally, rotate while moving and change direction instantly are all made possible by these wheels.

5.2 Vector Analysis

The robot's movement is based on vector analysis. Each wheel creates a motion vector, with the direction of the vector corresponding to the direction of the wheel's movement and the magnitude corresponding to the speed. The net movement of the robot is determined by the vector sum of these individual motion vectors. This principle allows the robot to move in any direction on a 2D plane, a characteristic of holonomic robots.

5.3 Force Calculation

The forces applied to the wheels are calculated using an inverse matrix of the direction cosines. This matrix relates the desired motion (in terms of direction and speed) to the necessary wheel forces. By calculating the inverse of this matrix, we can determine the forces that need to be applied to each wheel to achieve the desired motion.

5.4 Coding the Movement

These calculations are implemented in Python. The robot continuously calculates the necessary forces to apply to each wheel in order to achieve the desired movement. This involves reading sensor data, determining the desired motion, calculating the necessary wheel forces using the inverse matrix of direction cosines, and then applying these forces to the wheels. The images below show the mathematics on paper.

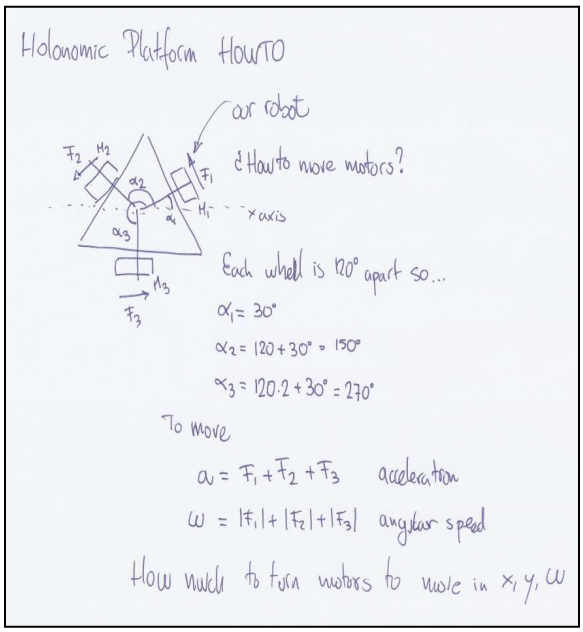

Image 13: Robot Math 1 (Source1)

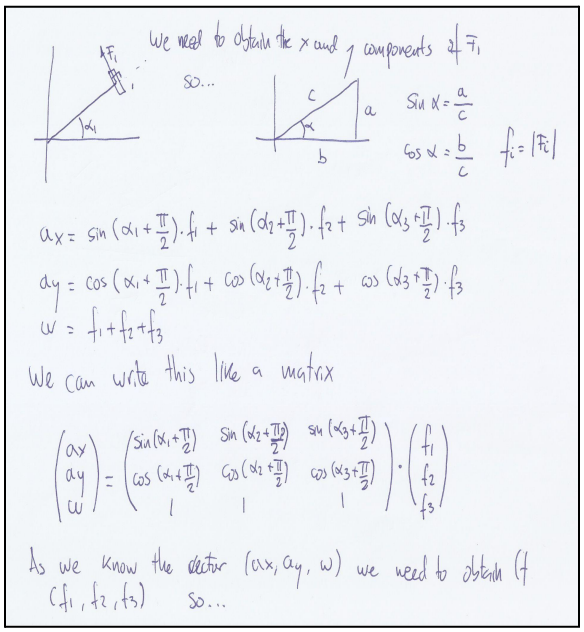

Image 14: Robot Math 2 (Source1)

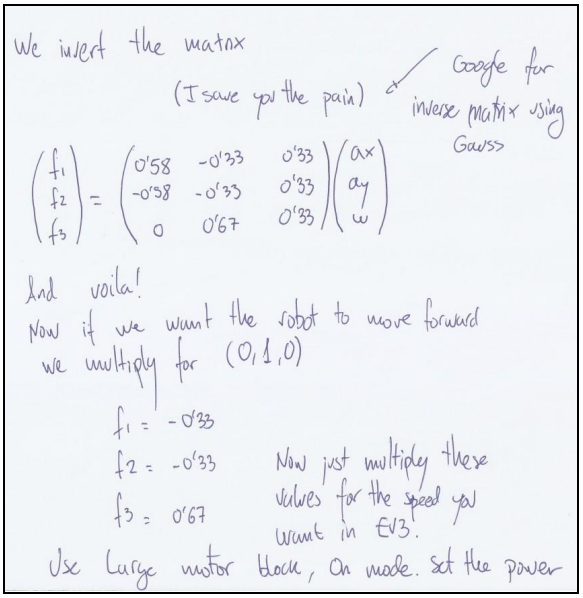

Image 15: Robot Math 3 (Source1)

```python
sensors = [front_sensor, back_sensor, left_sensor, right_sensor]

# Define the angles for each motor in radians.
a1 = math.radians(0)  # Motor 1 is at 0 degrees.
a2 = math.radians(150)  # Motor 2 is at 150 degrees.
a3 = math.radians(270)  # Motor 3 is at 270 degrees.

# Pre-calculated inverse matrix.
A_inv = [[0.56799, 0.34904, 0.33333], [-0.58627, 0.31738, 0.33333], [0.01827, -0.66642, 0.33333]]

# Function to calculate the forces.
def calculate_forces(ax, ay, w):
    B = [ax, ay, w]
    return [sum(A_inv[i][j] * B[j] for j in range(3)) for i in range(3)]

# Function to set the motor speeds.
def set_motor_speeds(f1, f2, f3):
    motor1.on(f1 * 50, 2)
    motor2.on(f2 * 50, 2)
    motor3.on(f3 * 50, 2)

# Calculate the forces to move straight forward.
F_foward = calculate_forces(1, 0, 0)
F_left = calculate_forces(0, -1, 0)
F_right = calculate_forces(0, 1, 0)
F_back = calculate_forces(-1, 0, 0)
wait_time = 1.4
```

Image 16: Robot Math Code Snippet

Section 6: Testing and Error Analysis

After the assembly of the robot, we embarked on a comprehensive testing phase to analyze the robot's movement and identify any potential errors. This section details the testing process, the error analysis, and the results obtained.

3.1 Surface Testing

Initially, our testing focused on how the robot performed on different surfaces. We tested its movement on various surfaces including the cement floor, the carpet available in the lab, the table surface of the lab tables, and different foam boards. This phase allowed us to determine the optimal surface for our robot. After careful observation, it was evident that the lab carpet provided the best performance.

3.2 Error Analysis

As part of our error analysis, we introduced a gyroscope to the robot's sensor array. This gyroscope played a crucial role in measuring and quantifying straight-line error and rotation error. We conducted multiple test runs, each time recording the actual and expected coordinates, the direction, the direction change, and the rotation error. The table below summarizes the results obtained:

To measure the accuracy/repeatability of holonomic movements.								
				Actual (x,y)		Expected (x,y)		
Test run	Direction	Direction Change	Rotation Error	x	y	x	y	Error (cm)
1	North		1	2.2	31	0	30	2.416609195
2	North		-2	-1.5	28.8	0	30	1.920937271
3	North	North	1	2.8	61.7	0	60	3.275667871
4	North	East	2	33.1	30.6	30	30	3.157530681
5	North	South	-1	-0.3	-0.5	0	0	0.583095189
6	North	West	3	-28.1	-32	-30	-30	2.758622845
7	East		0	30.3	0.2	30	0	0.360555128
8	East		1	31	-0.4	30	0	1.077032961
9	East	North	2	29.7	32.3	30	30	2.319482701
10	East	East	1	61.7	-0.8	60	0	1.878829423
11	East	South	-2	31.2	-28.9	30	-30	1.62788206
12	East	West	2	0.5	0.8	0	0	0.943398113
13	South		0	0.2	-29.7	0	-30	0.360555128
14	South		-1	-0.5	-30.3	0	-30	0.583095189
15	South	North	2	0.7	0.3	0	0	0.761577311
16	South	East	-1	31	-32	30	-30	2.236067977
17	South	South	1	-0.4	-62	0	-60	2.039607805
18	South	West	2	-29.7	-32.1	-30	-30	2.121320344
19	West		-1	-28	0.6	-30	0	2.088061302
20	West		-1	-28.6	0.4	-30	0	1.456021978
21	West	North	-1	-29.5	32	-30	30	2.061552813
22	West	East	2	0.8	1.2	0	0	1.44222051
23	West	South	1	-30.8	-31	-30	-30	1.280624847
24	West	West	0	-59.6	0.3	-60	0	0.5

Avg Rotation Error	0.458333333
Mean	1.635431193
Standard deviation	0.845053141
Variance	0.714114812

The average rotation error was found to be approximately 0.458 degrees. The mean error in the robot's movement was 1.635 cm, with a standard deviation of 0.845 cm, indicating a variance of 0.714 cm^2. These values provide a quantitative measure of the robot's accuracy and repeatability in its holonomic movements.

After the testing phase, we realized that our robot needed a way to correct its movement. Given the nature of maze navigation, the robot had to change its direction numerous times. This frequent change of direction could lead to movement errors, which we needed to correct for optimal performance.

However, we faced a challenge: our EV3 brick had only four input ports, all of which were already occupied by other sensors. This meant we couldn't include a gyro sensor in the robot's final sensor array.

To overcome this challenge, we implemented a centering algorithm that uses just the ultrasonic sensors. This algorithm helps the robot correct its movement and maintain its path within the maze. Here is the Python code for our centering algorithm:

```
def center_robot(direction):
    if direction in [Direction.NORTH, Direction.SOUTH]:
        # Use sensors 2 and 4 to center
        left_distance = left_sensor.distance_centimeters_continuous
        right_distance = right_sensor.distance_centimeters_continuous
        if abs(left_distance - right_distance) > 1:
            if left_distance > right_distance and right_distance > 2:
                movement.adjustLeft()
            else:
                movement.adjustRight()
    elif direction in [Direction.EAST, Direction.WEST]:
        # Use sensors 1 and 3 to center
        front_distance = front_sensor.distance_centimeters_continuous
        back_distance = back_sensor.distance_centimeters_continuous
        if abs(front_distance - back_distance) > 1:
            if front_distance > back_distance and back_distance > 2:
                movement.adjustForward()
            else:
                movement.adjustAround()
```

Image 17: Robot Centering Algorithm

This centering algorithm was instrumental in enhancing the robot's performance in maze navigation. It allowed the robot to correct its movement errors and maintain a straight path, significantly improving its efficiency in changing its Direction.

Section 7: Maze Design and Construction

The design and construction of the maze were integral to our project, providing a structured environment for our robot's navigation. The maze was designed with modularity in mind, allowing for easy reconfiguration and adaptability.

7.1 Material Selection

Initially, we planned to use plywood to build the maze walls. However, considering the relatively large size of our maze, we opted for foam boards instead. This decision was primarily driven by the need to keep the overall cost of the project down. Foam boards offered a cost-effective alternative to plywood, while still providing the sturdiness required for the maze walls.

7.2 Designing the Connecting Clips

The connecting clips, used to join the maze walls together, were designed from scratch. We created three types of clips: "T", "I", and "Corner/right angle" shapes. These clips

were designed to secure the maze walls both from the top and bottom, ensuring a stable and robust structure.

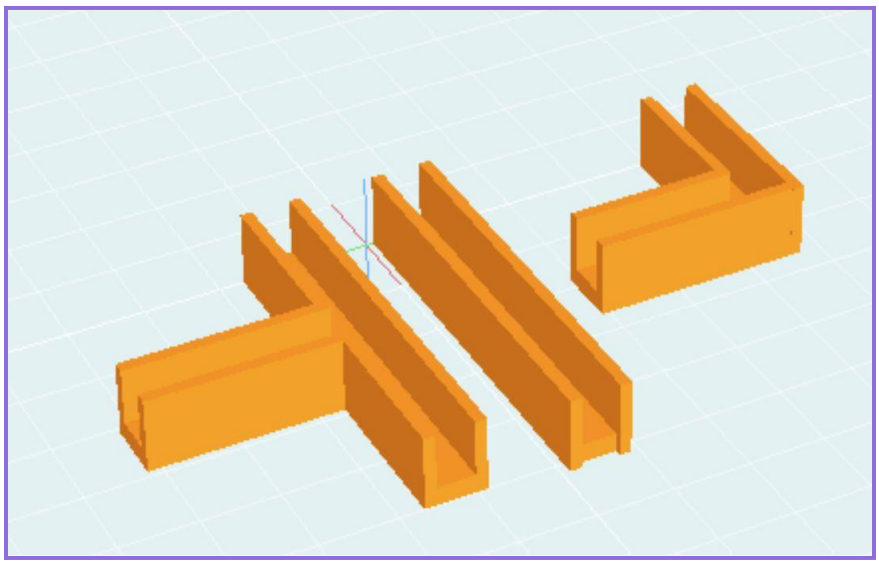

Image 18: Maze Connector Clips

7.3 3D Printing the Clips

We utilized 3D printing technology to produce the connecting clips. Multiple batches of these clips were printed to accommodate the size of our maze. The 3D printing process allowed us to create custom clips that perfectly fit our design requirements.

7.4 Maze Assembly

With the foam boards and 3D printed clips ready, we proceeded to assemble the maze. The modularity of our design allowed us to easily put together the maze walls and adjust the layout as needed.

The process of designing and constructing the maze was a challenging but rewarding experience. It not only required careful planning and design but also involved hands-on work in material selection, 3D printing, and assembly. This process was instrumental in providing a suitable environment for our robot to demonstrate its maze-solving capabilities.

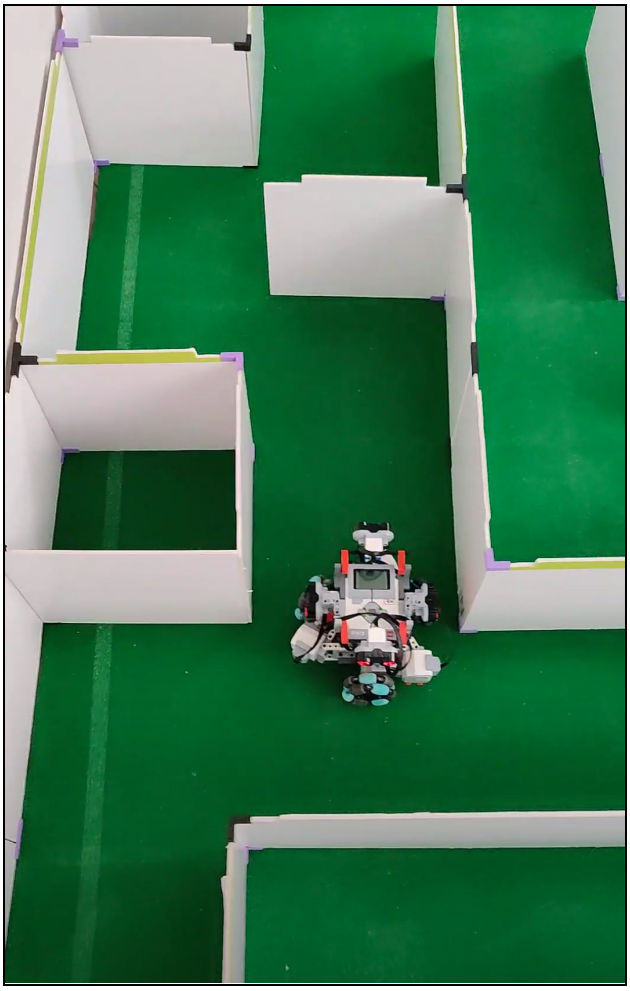

Image 19: Sample Maze Structure 1

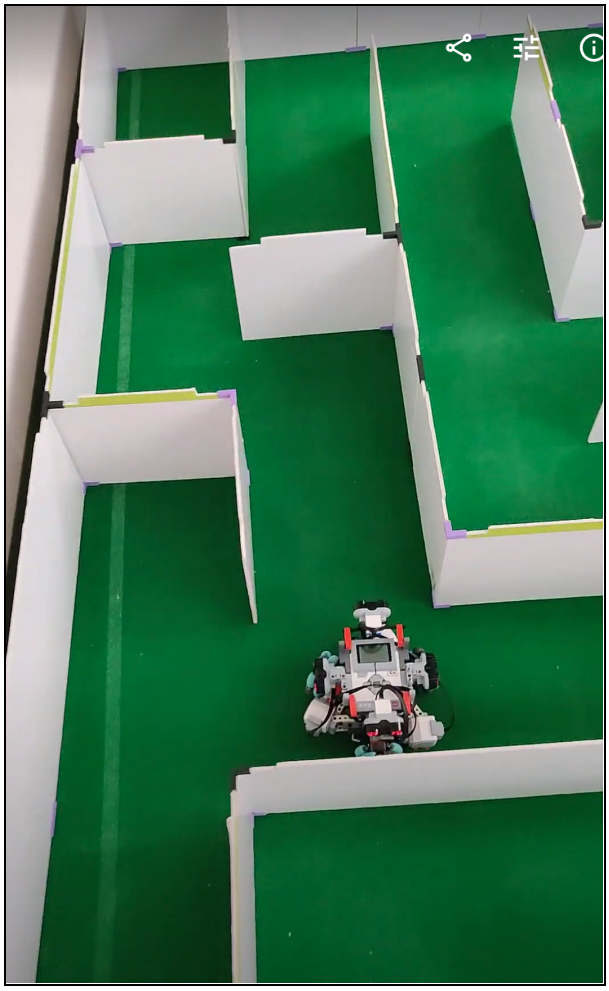

Image 20: Sample Maze Structure 2

Section 8: Maze Solving Strategy and Implementation

8.1 Maze Solving Strategy

To ensure that our robot does not face many errors while navigating the maze, we decided to keep our maze-solving strategy as simple as possible. We chose to implement a depth-first search (DFS) algorithm, a popular method for traversing or searching tree or graph data structures. The algorithm starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

```
def dfs(x, y):
    global visited
    visited.add((x, y))
    #print(x, y, file=sys.stderr)

    for a, b, d in getValidNeighbors(x, y):
        print(record_path,file=sys.stderr)
        moveInDirection(d)
        dfs(a, b)
        moveOppositeDirection(d)
```

Image 21: BFS Algorithm Snippet

8.2 Maze Navigation

After testing the robot in multiple different maze configurations, it was clear that the robot was able to autonomously navigate the maze and solve it. To begin with, we broke down the maze into a 30cm by 30cm grid and limited our robot's movement to 30cm at a time. Each time the robot moved 30cm in any direction, the robot paused for a very brief moment to check its ultrasonic sensor readings and use the centering algorithm discussed in the previous section to adjust itself if need be.

Before settling for this approach, we tried different approaches such as allowing the robot to move continuously until it reaches a wall in its current direction of movement. However, we found that the longer the robot moved continuously, the higher the chances of the robot crashing into the side walls. Therefore, we decided to limit the robot's movement to 30cm at a time, which significantly reduced the chances of crashing and improved the robot's ability to navigate the maze successfully.

In the following sections, we will discuss the challenges we faced, the lessons we learned, and the potential for future improvements.

Section 9: Challenges and Overcoming Them

Throughout the course of our project, we encountered several challenges. Each of these obstacles provided us with an opportunity to learn and adapt. In this section, we will discuss these challenges and how we overcame them.

9.1 3D Printing

As we were new to 3D printing, our initial challenge was learning how to use the 3D printers and how to optimize our print settings. This was a time-consuming process, and we had many failed prints at the start. However, with time and perseverance, we

managed to fine-tune our print settings and slice our parts optimally, leading to successful prints.

9.2 Maze Construction

The construction of the maze presented another challenge. Initially, we planned to use plywood for the maze walls, but due to the large size of our maze, we opted for foam boards to keep costs down. Cutting the foam boards precisely to form the maze walls was a challenging task. We had to measure and cut the foam boards accurately to ensure that the maze walls fit together perfectly. Despite the challenges, we successfully constructed the maze using foam boards.

9.3 Robot Movement

A significant challenge we encountered was related to the robot's movement. Ideally, the robot was designed to move in a straight line, but in some runs, it veered off to the right or left. After an extensive debugging process, which involved checking the wheel alignment, weight distribution, motor calibration, and sensor interference, we discovered that the issue was not due to our implementations or the structure of the robot.

Given the current setup of the maze and the robot, the robot was performing to the best of its abilities. The primary source of error in the robot's movement was the fact that the motors and sensors included in the EV3 kit were not 100% accurate. Additionally, the material used to print the wheel rollers was PLA plastic. While a rubber-like material would have been ideal for the wheel rollers to grip the surface better, we opted for PLA to keep the overall costs down. This material was provided for free by the ELKO garage.

Furthermore, we were limited by the number of input ports available, preventing us from using a gyro sensor to correct rotational errors. The inclusion of a gyro sensor could have significantly improved the accuracy of the robot's movement. Despite these challenges, we were able to navigate the robot through the maze successfully in most test runs.

9.4 Maze Solving Strategy

Our initial maze-solving strategy also presented challenges. We tried different approaches, such as allowing the robot to move continuously until it reached a wall in its current direction of movement. However, the longer the robot moved continuously, the higher the chances of the robot crashing into the side walls. Therefore, we decided to limit the robot's movement to 30cm at a time, which significantly reduced the chances of crashing and improved the robot's ability to navigate the maze successfully.

Despite these challenges, we were able to successfully complete our project. Each challenge provided us with valuable learning experiences and made our project journey

more interesting and rewarding. In the next section, we will discuss the lessons we learned from these challenges and how they can inform future projects.

Section 10: Lessons Learned and Future Improvements

Throughout the course of this project, we learned several valuable lessons. These lessons extended beyond the technical aspects of robotics and programming, encompassing teamwork, perseverance, and continuous learning. We learned the importance of planning and organization, the value of teamwork, the significance of perseverance, the necessity of learning from our mistakes, and the importance of continuous learning.

Looking forward, there are several improvements that could be made to enhance the robot's performance and capabilities:

10.1 Sensor Optimization

One prime improvement would be the addition of a medium motor to the robot, attached to a single ultrasonic sensor. This sensor could be rotated to measure the distance to the maze walls in all four directions. This modification would not only enhance the robot's awareness of its surroundings but also free up three input ports.

10.2 Incorporation of Additional Sensors

With the freed-up input ports, we could incorporate a gyro sensor and a color sensor. The gyro sensor would significantly improve the accuracy of the robot's movement by correcting any rotational errors. The color sensor would allow us to indicate the start and finish positions using different colors, as initially intended.

10.3 Maze Solving Strategy

Currently, the robot determines that it has solved the maze if all sensors show that it is not surrounded by any maze walls. With the proposed improvements, the robot could potentially identify the start and finish positions more accurately and navigate the maze more efficiently.

These proposed improvements, coupled with the lessons learned from this project, provide a promising direction for future work. They represent the next steps in our journey of exploration and learning in the field of robotics.

Section 11: Conclusion

In conclusion, our project of designing and constructing an autonomous, holonomic robot capable of solving mazes was a resounding success. Despite numerous challenges, we were able to navigate our way to a successful project outcome. The project not only provided us with technical knowledge in robotics and programming, but also taught us invaluable life skills such as teamwork, perseverance, and the importance of continuous learning.

The robot, with its custom-built omni wheels and depth-first search algorithm, was able to navigate and solve the maze autonomously. The modular design of the maze, made possible by the use of foam boards and 3D printed clips, provided a structured and adaptable environment for the robot's navigation.

While we are proud of what we have achieved, we acknowledge that there is always room for improvement. The lessons we learned from this project and the potential improvements we identified provide a promising direction for future work. We look forward to continuing our journey of exploration and learning in the field of robotics.

This project has been a testament to the power of determination, collaboration, and continuous learning. It has shown us that with commitment and dedication, we can overcome challenges and achieve our goals. We are excited about the possibilities that lie ahead and are eager to take on new challenges in the future.

Acknowledgement

We extend our gratitude to the following resources that significantly contributed to our project:

1. Holonomic Lego Mindstorms Robot by Dr. Christoph Bartneck - An insightful resource detailing the creation and design considerations for holonomic robots using Lego Mindstorms.
2. Omni-wheel vs Mecanum wheel from RobotDigg - A comparative analysis shedding light on the differences and advantages between omni-wheels and Mecanum wheels for robot mobility.
3. Holonomic Lego Mindstorms Robot by Dr. Christoph Bartneck - Another valuable resource from Dr. Christoph Bartneck, providing additional insights into the development of holonomic Lego Mindstorms robots.
4. Demonstration of Holonomic Robot by Jeremy Cole on YouTube - A demonstration video showcasing the functionality and movement capabilities of a holonomic robot.
5. Holonomic Robot Control System by ReM-RC on YouTube - A resource detailing the control system and operational aspects of a holonomic robot.
6. 3D Model for Omni-drive Wheels on Thingiverse - A 3D model resource offering designs for omni-drive wheels, a crucial component for our robot's mobility.
7. Micromouse wooden Maze - A retailer selling a modular wooden micromouse maze pieces

These resources have been instrumental in guiding and inspiring our project, providing invaluable insights and information throughout the development process.