End Semester Final Report

ECS418/618 Intelligent Robotics

Course Instructor: Dr. Sujit PB

Manoj Kumar (19188)

Ritik (19245)

Saswata Sarkar (19279)

Departmet of Data Science and Engineering

IISER Bhopal

Submission date: 25th of November 2022

Contents

1 Description of the environment

The given work is in the **final.wbt** file is to move the robot from start to goal while avoiding static and dynamic obstacles. The given robot is the **turtlebot3**, while the robot created in the environment is the dynamic obstacle. The obstacles move randomly in the environment.

1.1 A brief description of the simulation environment

- A floor is a rectangular object with dimensions of 3 meters cross 3 meters.
- The floor tile is square with dimensions 0.5 meters cross 0.5 meters.
- Two dynamic obstacles, i.e., the moving robots, and a static obstacle, i.e., a wall.
- Our robot turtleBot3Burger.

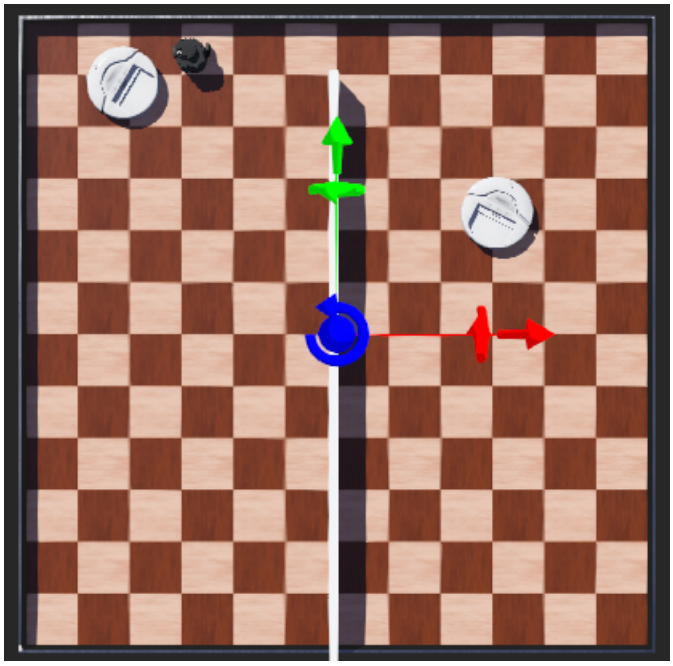

Figure 1: The World Environment

1.2 The objective

- Move the robot from the starting position of (-0.7,-1.2) to the goal position of (0.7,-1.2).
- The robot should not collide with any of the robots.

2 Obstacle Detection

The following methods do obstacle detection:

- Getting the position of all the dynamic and static obstacles from the supervisor.
- Check the distance between the dynamic robots and our **turleBot**.
- With the help of **real time position and coordinates of obstacles and robot** (Laser imaging, Detection, and Ranging), obstacles are detected when the distance between the dynamic robots and **turtleBot** is less than 0.8 meters.
- We check the block in the rectangular arena also. We divide the arena into left and right blocks concerning the static obstacle, i.e., the wall, which eases the simulation.

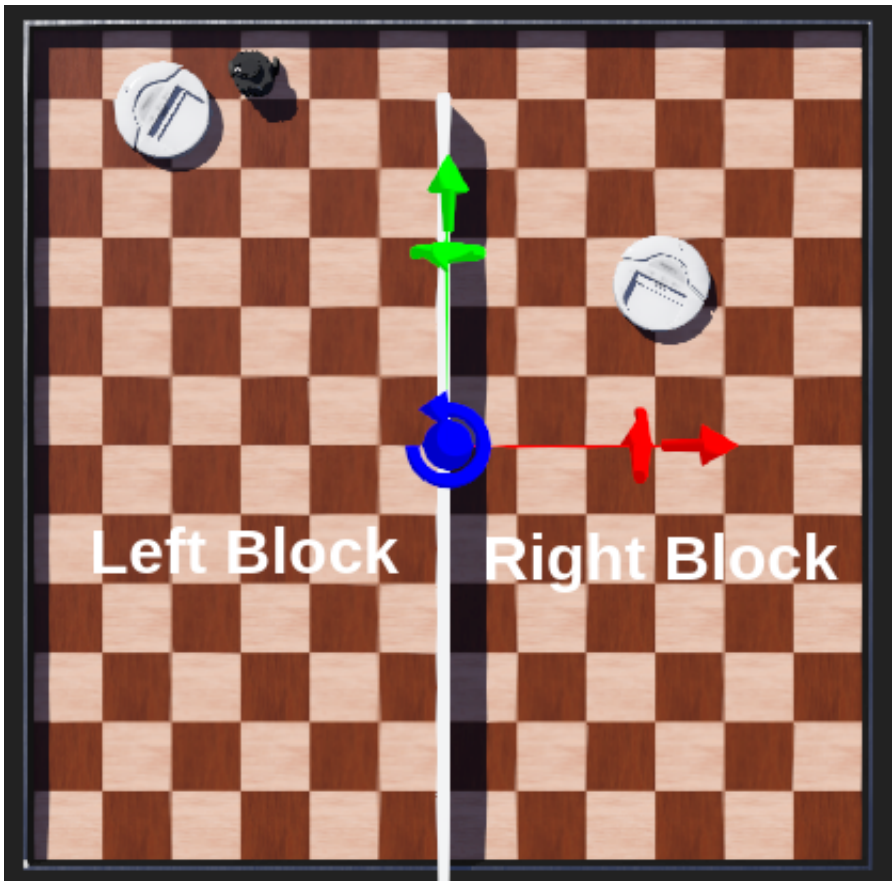

Figure 2: The World Environment with Separated Blocks

3 Obstacle Avoidance:

For the left block, if the dynamic obstacle, i.e., the robot roaming in the left block, is detected, then the turtleBot checks the heading direction of the roaming robot. Here are the following two cases:

- If the moving robot is coming towards the direction of the front of the turtleBot, then turtleBot moves backward in the opposite direction with negative velocity.
- If the moving robot is coming towards the direction of the back of the turtleBot, then turtleBot moves forward with the maximum speed in the direction it was following.

For the right block, if the dynamic obstacle, i.e., the robot roaming in the right block, is detected, then the turtleBot checks the heading direction of the roaming robot. Here are the following two cases:

- If the moving robot is coming towards the direction of the back of the turtleBot, then turtleBot moves forward with the maximum velocity in the direction it was following.
- If the moving robot is coming towards the direction of the front of the turtleBot, then turtleBot moves backward in the opposite direction with negative velocity.

4 Algorithm Used

To achieve the objective of the task, we use a modified version of the **Bug2** algorithm as the following:

- The robot have two states: "Go towards the Goal" and "Follow the Wall."
- Initially, the robot is in the state "Go towards the Goal," and in this state, the robot will head directly towards the goal until it encounters any static obstacle. whenever it encounters a stationary obstacle, its state changes to "Follow the wall" whenever the state of the robot changes (either from "Go towards the goal" to "Follow the Wall" or vice-versa), it will turn "left." If the robot is in the state "Follow the wall," it will follow the wall by turning "right."
- To sense whether the robot came near the wall, we use a LiDAR (Laser imaging, Detection, and Ranging) sensor. Whenever the LiDAR value of the front is less than a certain threshold, our robot has come very close to the wall and must turn left or right.

5 Contributions

- **Ritik** Implemented obstacle detection and obstacle avoidance algorithm.
- **Manoj** Implemented Path planning algorithm for the robot (modified Bug2).
- **Saswata** Got LiDAR values and position coordinates and implemented the path planning algorithm.