End Semester Examination

Meet Maratha (19190)

November 25, 2022

1 Problem Statement

The environment consists of 3 robots, out of which 2 are active Create robot obstacles that move in a straight line. Our main robot is a turtle bot. The environment just consists of a single wall in the middle of the rectangle arena which has a narrow opening at the top. Our robot is currently in the section neighboring the section containing the goal. We need to move our robot such that it reaches its goal without colliding with the obstacles or the wall. Here a collision with the obstacle robot is considered when our robot is at a distance of 0.1 units from the obstacle robot, from their bodies.

2 Method

The algorithm used is simple and as follows:

Main Algorithm:

1. Initialize all the robots.
2. Set $TIMESTEP = 32$, $BASE\_SPEED = 6.67$, and a list named $WAYPOINT$ which contains the location of waypoints that the robot needs to follow to reach the goal.
3. Create two variables that will store two consecutive values of the positions of obstacle robots.
4. Set $goal_idx = 0$, which will represent index of the current goal from variable $WAYPOINT$ for the robot.
5. Start a **WHILE** loop, till we reach the end.
6. If 2 points have not been sampled for the obstacles, set the velocity of the robot as $BASE\_SPEED$ and return to step 5
7. Check if the robot has reached the current goal location. If it has, move to the next goal. If there is no next goal, stop the robot and let the user know that robot has reached the end goal.
8. If 2 points have been sampled for moving obstacles and the robot's x-coordinate is less than or equal to 0, then perform the following.
   (a) Set variable $dist\_0 =$ *distance between robot and obstacle* 1

(b) If $idst\_0 < 0.4$ and the obstacle is in a radius of 0.8 *units* to the robot, set $collision = True$ else set $collision = False$

(c) If there is a collision do the following:

i. Set variable $foot = position\ of\ the\ foot\ of\ perpendicular\ to\ the\ trajectory\ of\ obstacle\ 1$

ii. If the distance between the robot position and the foot of the perpendicular is less than 0.4 then set the goal as the foot of the perpendicular and move away from the goal.

iii. Else set the current position of obstacle 1 as the goal and move away from it.

(d) As there is no collision go towards the current goal.

9. If 2 points have been sampled for moving obstacles and the robot's x-coordinate is greater than 0, then perform the following.

(a) Set variable $dist\_1 = distance\ between\ robot\ and\ obstacle\ 2$

(b) If $idst\_0 < 0.4$ and the obstacle is in a radius of 0.8 *units* to the robot, set $collision = True$ else set $collision = False$

(c) If there is a collision do the following:

i. Set variable $foot = position\ of\ the\ foot\ of\ perpendicular\ to\ the\ trajectory\ of\ obstacle\ 2$

ii. If the distance between the robot position and the foot of the perpendicular is less than 0.4 then set the goal as the foot of the perpendicular and move away from the goal.

iii. Else set the current position of obstacle 2 as the goal and move away from it.

(d) As there is no collision go towards the current goal.

10. Return to step 5

Algorithm $goToGoal(translation\_field, rotation\_field, max\_speed, goal, flag)$**:**

1. The variable $flag$ is used to represent whether we want to go to the goal or move away from it, which is used in obstacle avoidance,

2. get *position* and *rotation* of the robot from $translation\_field$ and $rotation\_field$ respectively.

3. Calculate the heading angle of the robot.

4. Calculate desired heading angle for the robot which is given as $desired\_angle = atan2(goal_y - position_y, goal_x - position_x)$

5. If the difference between desired heading angle and the current heading angle of the robot is significant and we need to rotate to move toward the goal do the following:

(a) If $goal_x - position\ of\ robot_x < 0$ set left wheel velocity as $-max\_speed$ and right wheel velocity as max_speed

(b) Else set left wheel velocity as max_speed and right wheel velocity as $-max\_speed$

6. If the difference between desired heading angle and the current heading angle of the robot is somewhat significant and we need to rotate to move away from the goal do the following:

(a) If $goal_x - position\ of\ robot_x < 0$ set left wheel velocity as $-max\_speed$ and right wheel velocity as max_speed

(b) Else set left wheel velocity as max_speed and right wheel velocity as $-max\_speed$

7. As we don't need to rotate, set the velocity of the robot as $-max\_speed$ if we need to move away from the goal and set it to max_speed if we need to move towards the goal.

8. Return the speed for each wheel of the robot

For detecting the position field of all 3 robots and getting the rotation field of our robot supervisor was used.

3 Contribution

This project was made in collaboration with Marut Priyadarshi who helped develop the algorithm for the project.

4 Conclusion

The robot was able to avoid all moving obstacles and safely reach the goal present in the neighboring sector after passing through the narrow opening.