Assignment- 3
Collision Avoidance

Shashwat Shukla

213234001

Scenario:

- In the given scenario here, the robot starts from (0,0,0) position and is required to travel to x=5, y=0 coordinate.
- The robot has to navigate avoiding dynamic as well as static obstacles.
- The max robot velocity magnitude is 0.15m/s
- And max deviation in orientation is 10° in any step.

Algorithm:

- To begin with all robots are given to be circular with diameter= 0.15m.
- Since the velocity executed by the robots are subject to errors compared to the command, very close motion may result in robots grazing each other.
- To avoid robot grazing and to have clearer margins, the robots in the simulation are considered to be of diameter 0.2m.
- To set of feasible velocity vectors of the robot are discretized by in steps of 0.01m/s in the magnitude and the 1° in angle.
- At any given position, the set of feasible vectors is scanned to find vectors that do not end in collision with any of the obstacles.

  Collision check:

  - The robot is considered to be point robot and the obstacle is enlarged by adding robot radius to the obstacle radius.
  - From the point robot, the orientation of the obstacle is calculated and the tangent lines orientation is also calculated.

- The tangent lines are oriented to the line joining the centre of obstacle and robot by an angle given by:

$$\alpha = sin^{-1}(\frac{Radius\ of\ the\ enlarged\ obstacle}{Distance\ between\ obstacle\ and\ robot})$$

  - From the set of feasible vectors, a candidate is picked up and the relative velocity of the robot is calculated with respect to the obstacle.
  - If the velocity is oriented between the tangent lines it will lead to collision.
  - Such check is done for all obstacles.

- From the non-colliding velocities, the best among them is selected.

  Selection of best velocity:

  - The selection of the best velocity is based on the TG strategies given in literature.
  - In essence the best velocity is the one most closely oriented towards the goal with the maximum velocity.
  - This velocity is selected and commanded to the robot.

- Flowchart in figure 1 shows the overall algorithm used.
- The code is ended when the robot reaches reasonably close to the target.

Results:

- The robot can be seen successfully navigating to a point near the goal.
- The robot is able to avoid collision with any of the obstacle robots with clear margins.
- Figure 2 shows the path of the robot in the environment.
- Figure 3 shows the X and Y axis translations of the robot with respect to time.
- The model to drift around the goal and hence the robot steers slightly off.

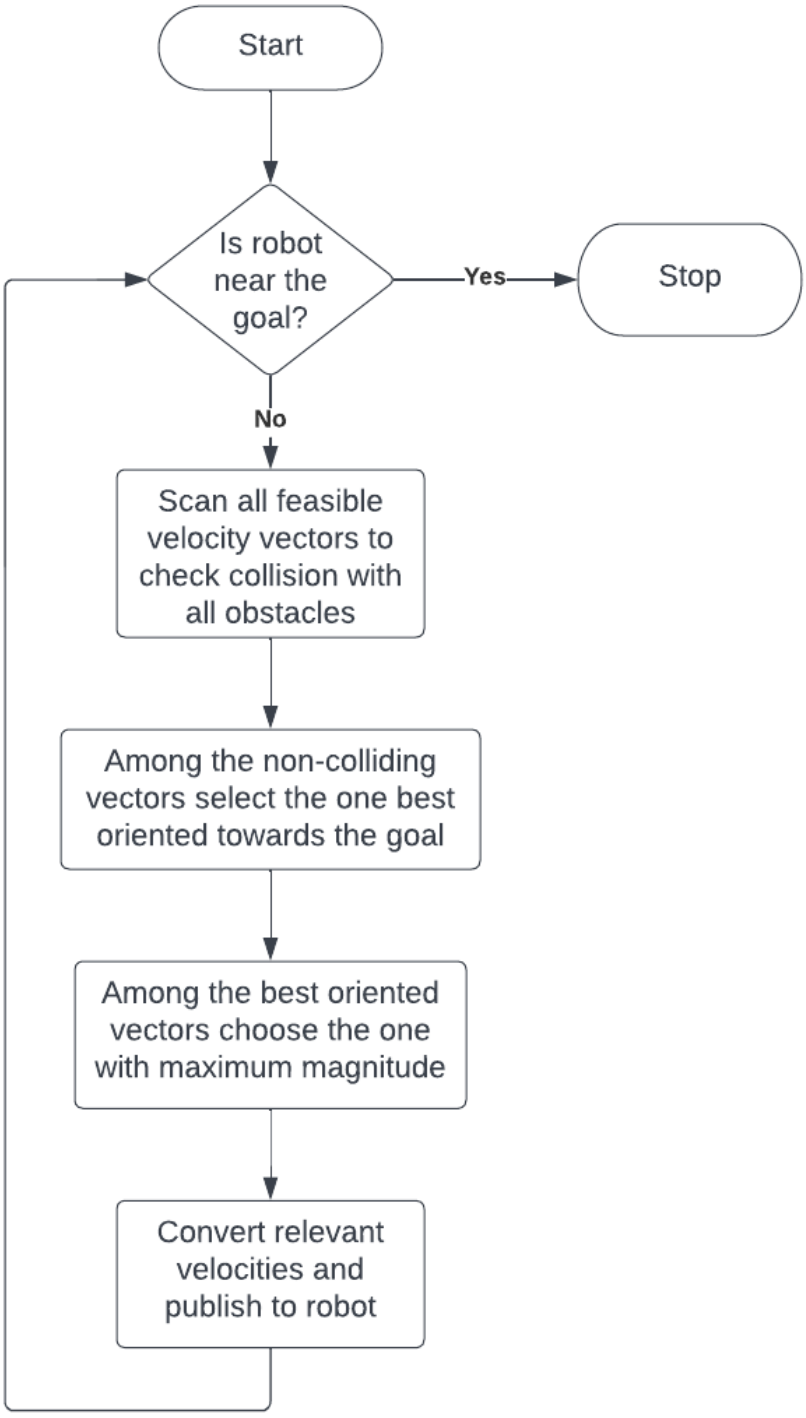

Fig 1: Flowchart of algorithm followed

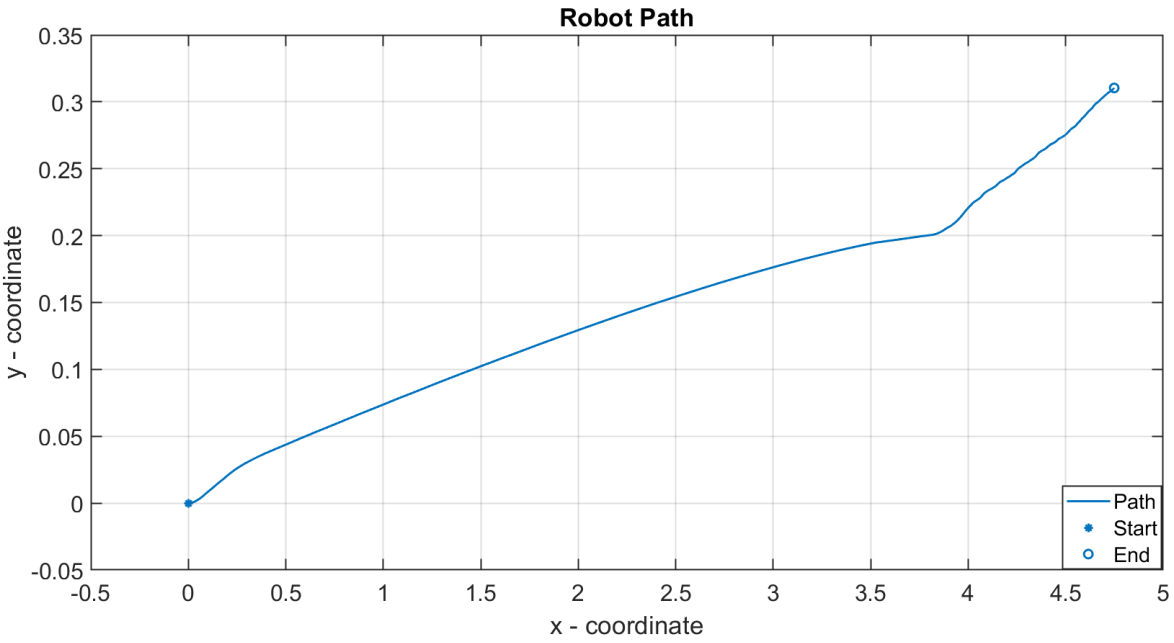

Fig 2: Path followed by the robot

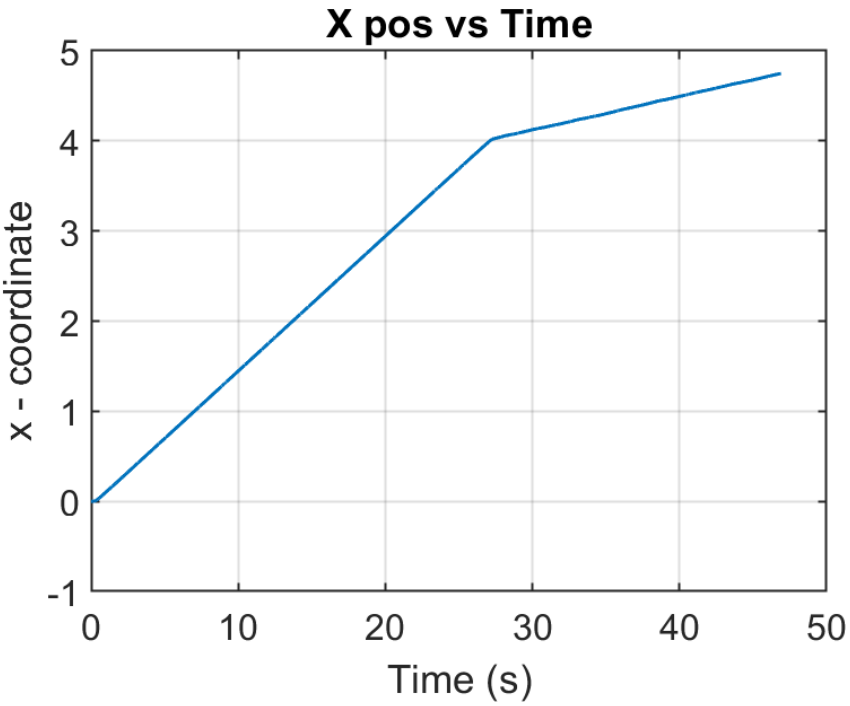

Fig 3: Variation in x - coordinate