

SC627 Assignment 1 report

180100049 – Hemanth Dontamsetti

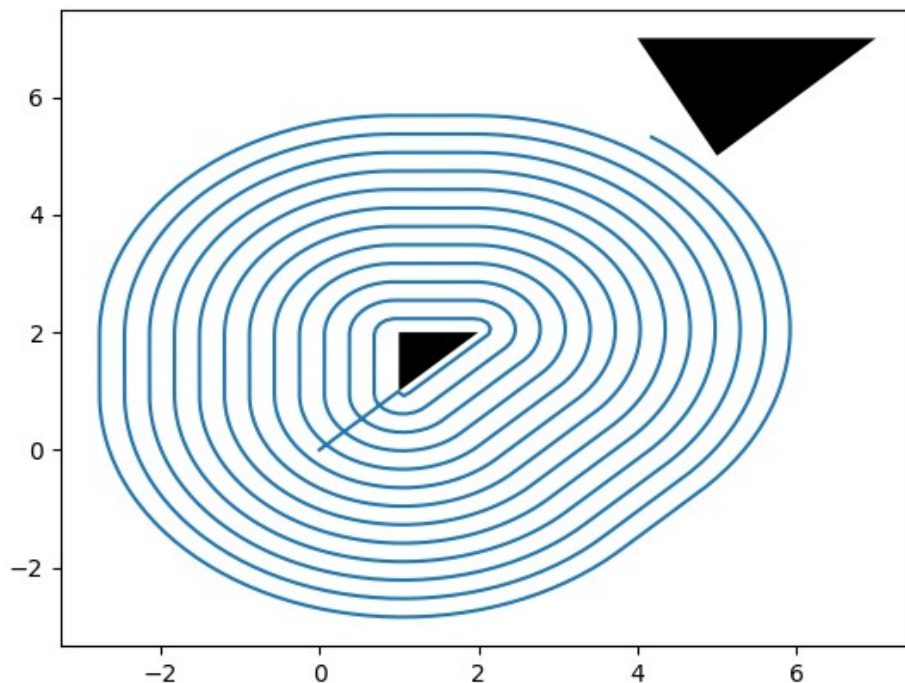
Implementation :

After implementing 1.6, 1.7, 1.8. I encountered following problems

1) Going inside virtual obstacles and provided tolerance

Solution : Tangent vector is changed if current position is inside obstacle or outside above a tolerance. This tolerance can be used to incorporate width of the robot.

2) Spiraling outwards as the robot moves around the obstacle.



Solution : While moving around the obstacle if the robots is away from it beyond a certain distance then

- 1,2 made computeTangentVectorToPolygon function pretty complex

3) Unwanted continuous rotating of the robot at place

Solution : added a timeout to wait_for_result method and if resulted in timeout another goal is picked which is the middle of current_position and next position; this eliminated the problem (but if timeout is resulted in second time as well then program stops with a warning)

4) Going round the obstacle more than once

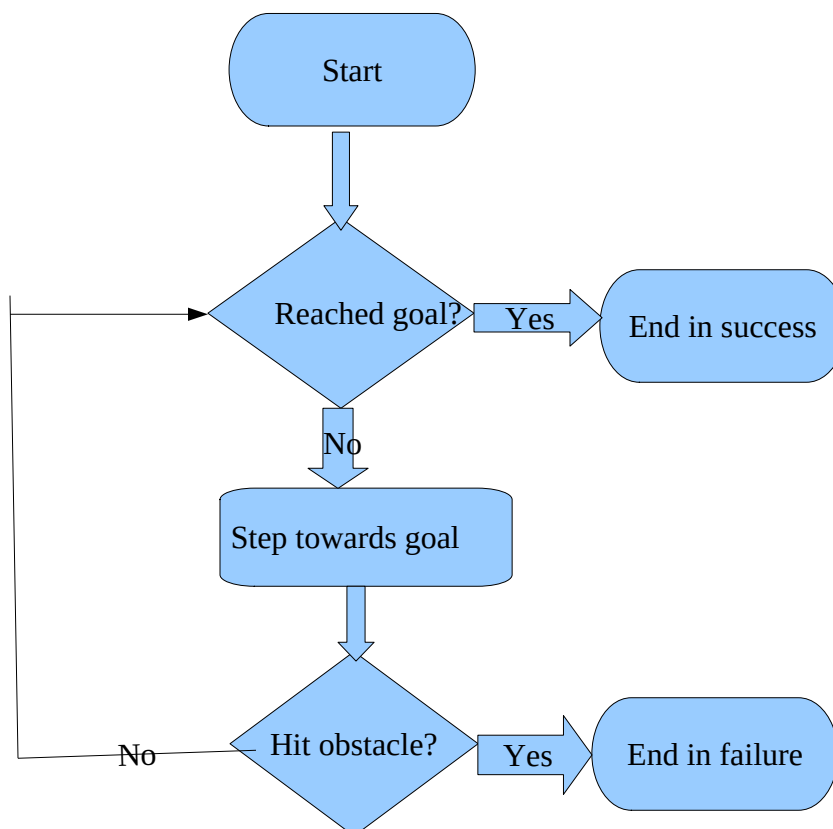
This happens as a result of finite step length. So there is possibility of missing the key points (obstacle's start point or obstacle's exit point) as result of moving away or near the obstacle After one loop around the obstacle.

Solution : Tuned few parameters like tolerance ; step length if inside obstacle or outside beyond certain range (Also value of this range beyond which robot is considered "far")

5) Few feedbacks/log infos are provided while running so as to get a better estimate of performance and errors in the program

1.8 Answers

i. Flow chart of bug_base



- ii. Instead of ending in failure after hitting the obstacle, we have to make the robot go around the obstacle once while storing their distances from goal. After circling once go around the obstacle till min distance point from goal. Then if we are able to proceed to goal from this point then continue with going towards goal until hit by obstacle, else end in failure that no path to reach goal.
 - To check whether the bot hit an obstacle within a given tolerance, `ComputeDistancePointToPolygon` is needed
 - While going around obstacle `computeTangentToPolygon` is needed
- iii. A video of trace of last ten steps of the path followed by the turtlebot_3 is the same folder named `bug_1motionROS.mp4`. Fig2 shows total trace of the bot.
- iv. For the given example
 - Bug_base terminated in *29 secs* as obstacle was present. (I,e it took 29secs to reach first obstacle while going towards the goal)
 - Bug_1 took *12min 14 secs* to complete with Success Flag , traveling a distance of ~ 25m

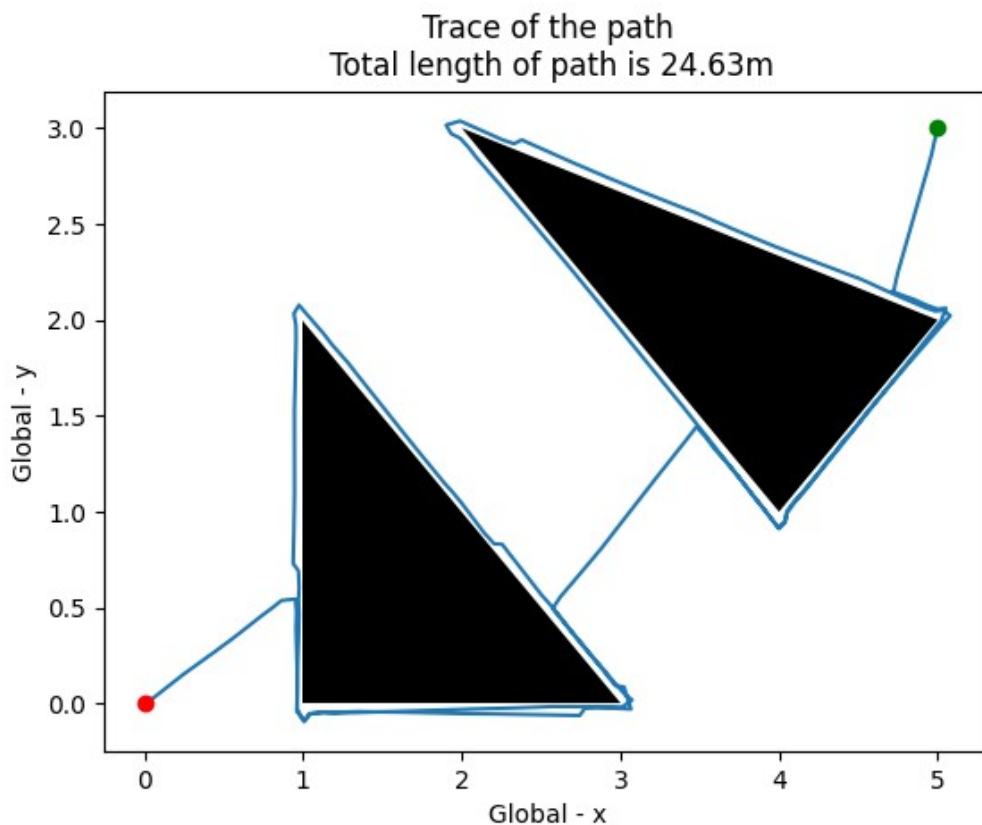


Fig 2: Total Trace of the turtlebot3 robot using bug_1 algorithm