SC 627 – Assignment 1

Nakul Rondad | 180010047

E 1.6 :

① Compute line through 2 pts

$$P_1 \equiv (x_1, y_1) \quad ; \quad P_2 \equiv (x_2 - y_2)$$

This gives, $\begin{vmatrix} x_2 - x_1 & x - x_1 \\ y_2 - y_1 & y - y_1 \end{vmatrix} = 0$

$$\Rightarrow (y_1 - y_2)x + (x_2 - x_1)y + (x_1 y_2 - x_2 y_1) = 0$$

We scale this by $[(y_1 - y_2)^2 + (x_2 - x_1)^2]^{-1}$ for $A^2 + B^2 = 1$

② Compute dist. of pt. to line

If 'P_1' & 'P_2' pass through line $ax + by + c = 0$

dist of 'q' from this line is

$$d = \left| \frac{a q_x + b q_y + c}{a^2 + b^2} \right|$$

③ Compute dist. of pt. from segment

$\overrightarrow{vec1}$ ≡ vector from start of line segment to end of line segment

$\overrightarrow{vec2}$ ≡ vector from pt to start of line segment

$$\text{rel length} = (\overrightarrow{vec1} \cdot \overrightarrow{vec2}) / |\overrightarrow{vec1}|^2$$

if rel length < 0 : dist = dist (start pt, pt.)
state ≡ left of start pt {1}

if rel length > 1 : dist = dist (end pt, pt.)
state = right of end pt {2}

else dist = ⊥ dist from pt to line
state = on the line segment {0}

E 1.7

① Compute distance pt to polygon

→ Compute the dist of the pt. from every 2 adj vertices of the polygon
→ Return the min dist of the above computed

② Compute tangent vector to polygon

→ Find the segment closest to the pt. by looping through every 2 adj vertices as edge

→ Depending on its state i.e. on the left, on the right or projection on this closest segment;

- if state = 0 { on the segment }

  tangent has slope equivalent to closest segment.

- if state = 1/2 :
  Let the closer pt of the segment be $\hat{p}$ and original pt. be $\hat{q}$

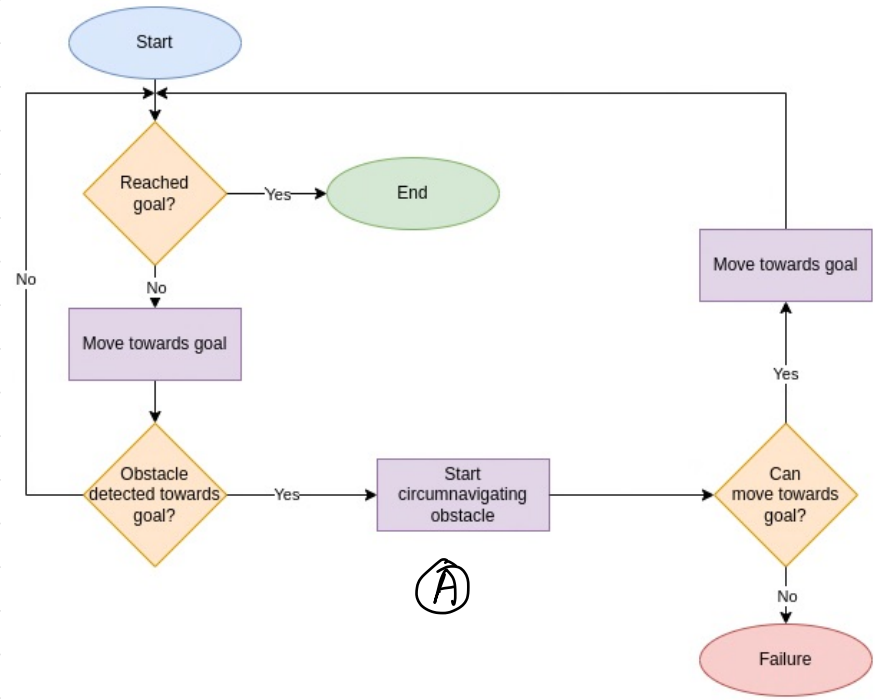
$$\overline{vec} = (\bar{q} - \bar{p}) \quad \text{from } p \text{ to } q$$

$$\text{tangent} = \hat{k} \times \overline{vec}$$

↳ ensures anticlockwise

E 1.8

i) Flowchart for implementation of BugBase algorithm

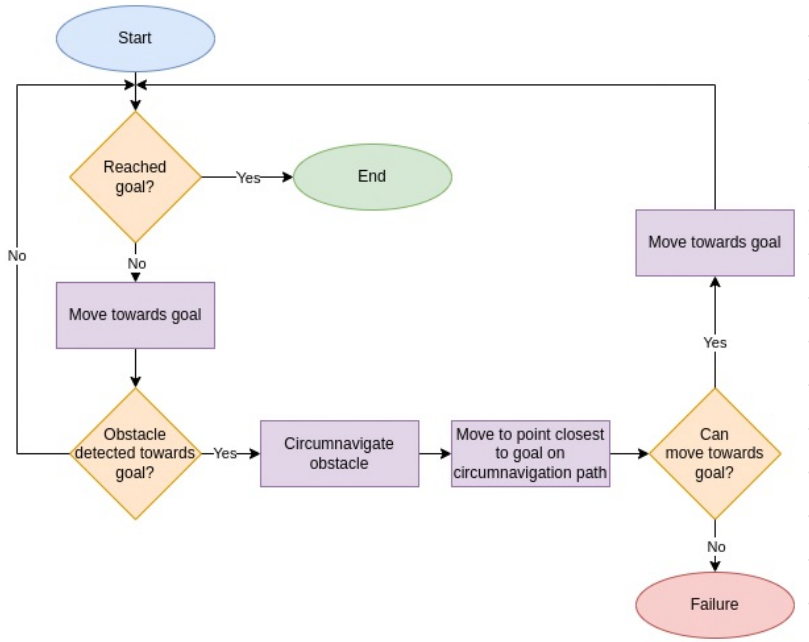


ii) → Task Ⓐ of BugBase algorithm is changed/modified to transform this to Bug1 algorithm.

→ In BugBase, the robot circumnavigates the obstacle only till the point where it is able to move towards the goal.

→ Whereas in Bug1 algo, the robot fully circumnavigates the obstacle and departs towards the goal only from the point on circumnavigation path that is closest to the goal

Flowchart of implementation of Bug 1 algorithm



iii) Implementation of functions in E 1.7 and E 1.6 are given on earlier pages. These will be used to:

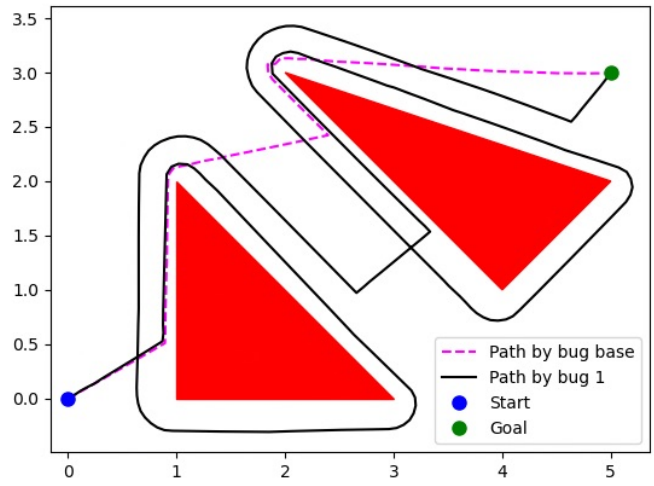
a) Detect the distance from obstacle

b) Align the robot about the boundary of obstacle

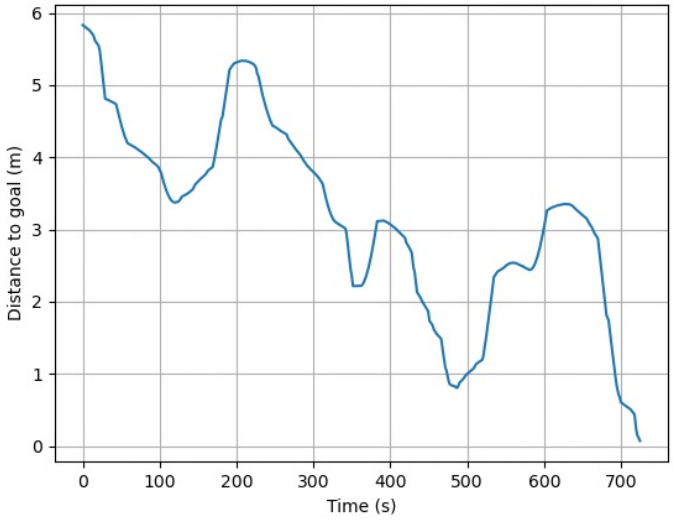
c) Move in a circular path when close to vertices

d) Calculate distance between two points

iv) • Trajectory followed by the robot :



• Plot showcasing the distance from bug's position to the goal as a function of time :



- By Bug 1 algorithm:

→ Time taken to compute and travel the path is

724.66 secs

→ Length of total path travelled by robot is

29.2 m