Path Planning of autonomous omni-directional robot

Asia-Pacific Robot Contest 2020

Introduction

The Asia-Pacific Robot Contest 2020's problem statement is based on the rugby 7's game using two robots(Try robot and Pass robot) and five obstacles/poles as five defending players. The Pass robot needs to pass a try ball to the Try robot. On a successful receive of the ball by the Try robot, the Try robot needs to avoid the five defending players and place the ball in one of the Try spots. Therefore, the Try robot needs to plan it's own path to reach a given try spot. To achieve this task a planning algorithm is developed in accordance with the available data.

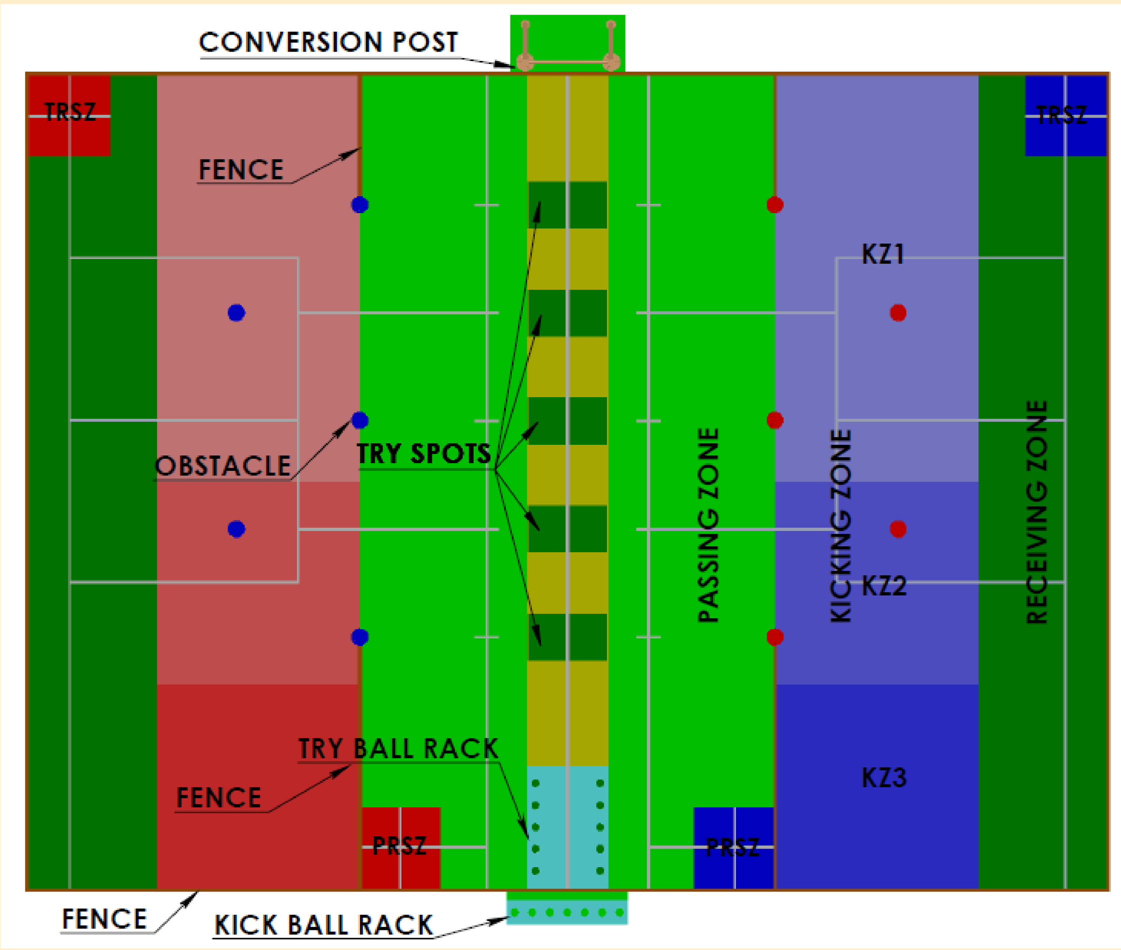


Figure 1: Arena(Top View)

*(TRSZ - Try robot start zone; PRSZ - Pass robot start zone)

Basic Overview

The algorithm is inspired by bug2 and visibility graphs. The available rudimentary data with the robot: 1. Pose (x, y, θ) 2. Arena

Below is the basic flow of the algorithm.

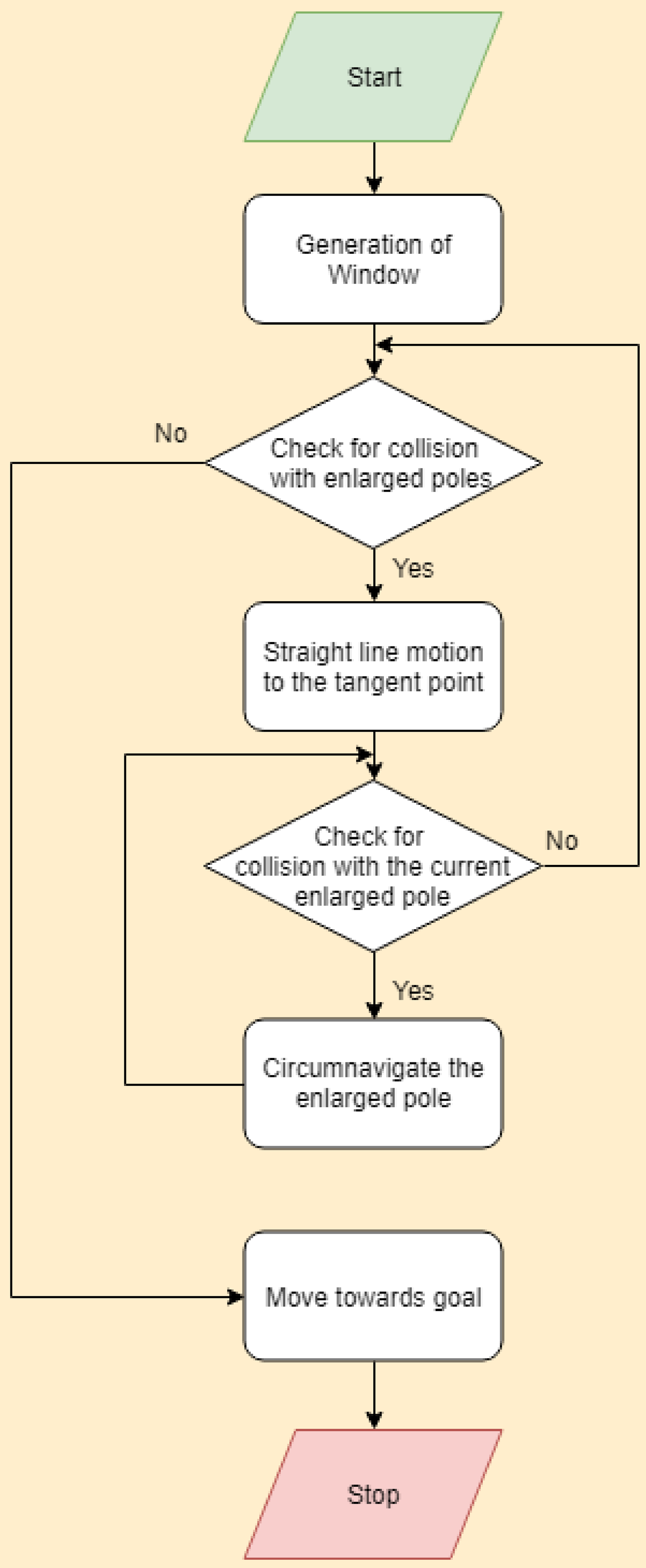


Figure 2: Flow diagram

Planning

Suppose the robot to be a circle of radius r=70cm. Given the work-space, the configuration space can be found by C-space transform. Let a goal position be assigned to the robot. Using the goal position and the current position, a rectangular window is generated. The enlarged poles contained in the rectangular window are then checked if they thwart the path of the robot. Rest of enlarged poles are eliminated. Once this process is completed, the robot executes a straight line motion to the tangent point(T_i) of the first enlarged pole(O_i). The robot on hitting the tangent point(T_i) checks if the current enlarged pole thwarts its path(current to goal). If yes, the robot will circumnavigate the enlarged pole and leave at (L_i) when the enlarged pole no longer thwarts the path(current to goal). This process is repeated until all the enlarged poles thwarting its path are avoided. *Note: Special cases for fence are added.*

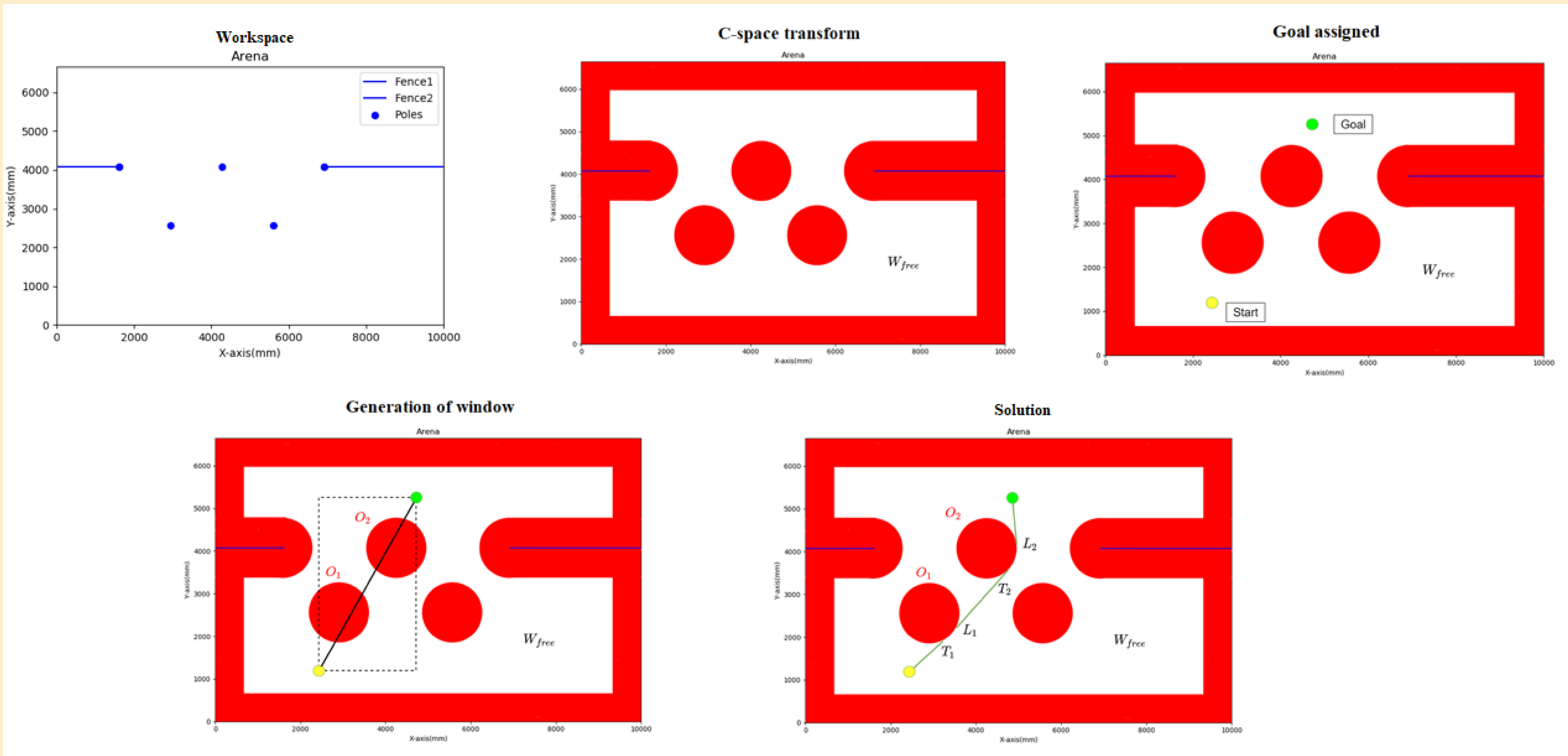


Figure 3: Visualization of algorithm

The Recipe

Let a goal position $G(x2, y2)$ be assigned to the robot situated at position $S(x1, y1)$. The robot has a prior information that a pole is located at position $C(h, k)$ and assumes it to be of a radius $'r'$. Before moving towards the goal position the robot makes sure its path is clear. For the pole to thwart the path of the robot, the equations (1) and (2) must have a solution.

$$y = m * x + c \quad (1)$$
$$where, m = \frac{(y2-y1)}{(x2-x1)}$$
$$(x-h)^2 + (y-k)^2 = r^2 \quad (2)$$
$$\alpha * x^2 + \beta * x - \gamma = 0 \quad (3)$$
$$\text{This gives solution when, } \beta^2 + 4 * \alpha * \gamma \geq 0$$
$$where, \alpha = 1 + m^2$$
$$\beta = 2 * (h - m * c + m * k)$$
$$\gamma = r^2 - h^2 - k^2 - c^2 + 2 * c * k$$

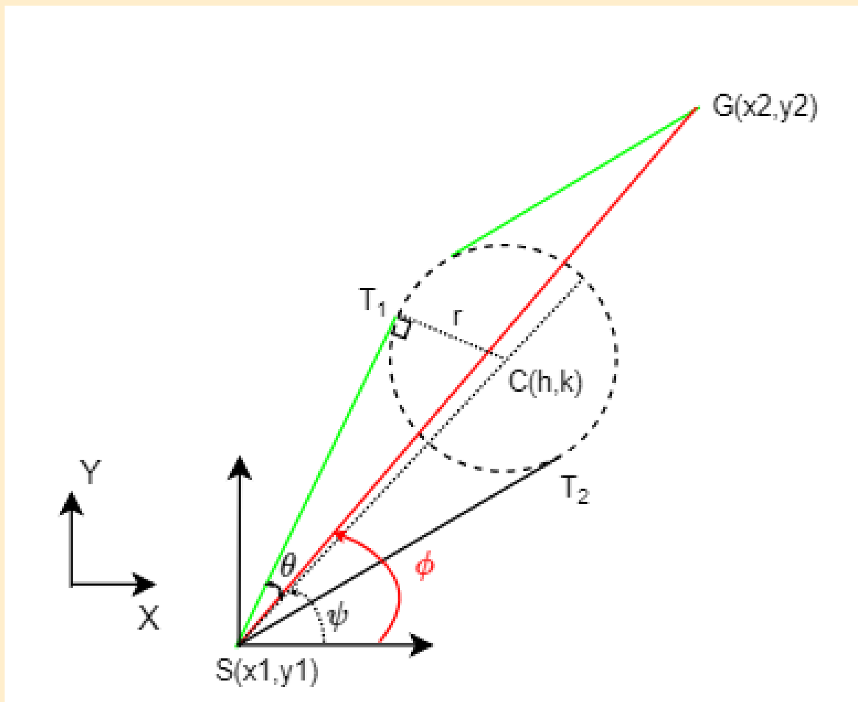


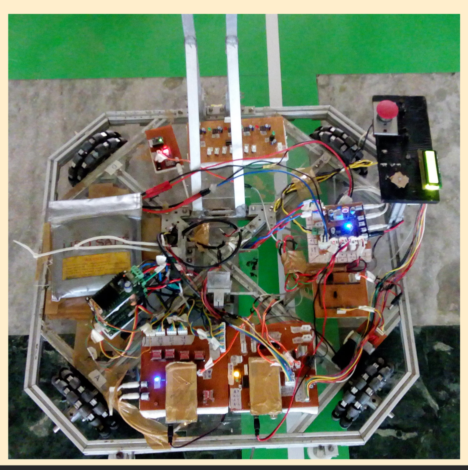
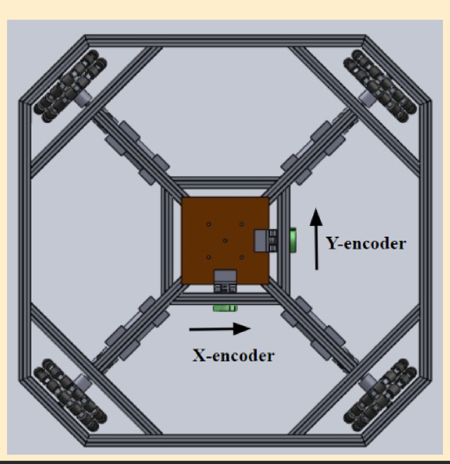
Figure 4: Enlarged pole

If the condition is satisfied, the following equations are computed for point T_1 or T_2.

$$angle_{SG}(\phi) = arctan(\frac{y2-y1}{x2-x1}) \quad (4)$$
$$\text{angle}_{SC}(\psi) = arctan(\frac{k-y1}{h-x1}) \quad (5)$$
$$\text{d}_{SC} = \sqrt{(x1-h)^2 + (y1-k)^2} \quad (6)$$
$$\theta = sin^{-1}(\frac{r}{d_{SC}}) \quad (7)$$
$$\text{angle}_{ST} = angle_{SC} + sign * \theta \quad (8)$$

When the constant $sign$ is positive, the line ST_1 is selected, otherwise line ST_2. For travelling the shortest path as a whole, $sign = \frac{fabs(\phi-\psi)}{\phi-\psi}$. Once the $angle_{ST}$ is known, the equation of line ST_1 and equation of line CT_1 can be solved for computing the point T_1. *Note: The special case when* $m = \infty$ *was also taken into consideration while implementing the algorithm on the robot.*

Testing Robot

Conclusion

A four wheeled omni-directional robot was able to avoid the obstacles and reach its goal using this algorithm. The algorithm has a remarkable feature of reducing the computations by generating a window, consequently eliminating the unnecessary obstacles to deal with.