**Dynamic Obstacles (RPN Assignment 3)**

Sachin Chandani (20161201)

Pratikkumar Bulani (2019201074)

Handling Multiple Obstacles:

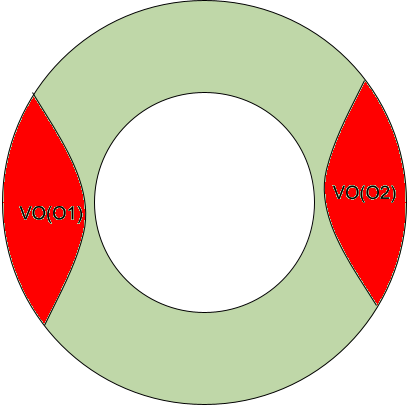
Due to presence of multiple dynamic obstacles, the velocity obstacle is calculated as: Union over the velocity obstacle of individual obstacles i.e. 

Feasibility radius:

Consider that the robot is present in Mumbai and the obstacle in Delhi (~1500 km). Then is this obstacle really an obstacle for us? Definitely no. An obstacle is considered as an obstacle only when it is within the feasibility radius of the robot. When the obstacle is inside the feasibility radius of the robot, our algorithm calculates the collision cone and its velocity obstacle.

Reachable Avoidance Velocity:

It is the set of velocities that the robot can take which ensures no collision. Example:



Here, the green region indicates the reachable avoidance velocities of the robot and red region indicates the velocity obstacles. So the sampling method can choose a random velocity within this green region.

Allowed Strategies:

Strategy 1: Allow the robot to travel straight to the goal with the maximum permissible speed within the permissible steering angle. This method suits well for the non-holonomic robots.

Strategy 2: Allow the robot to travel straight to the goal in any direction while taking the greedy velocity (using which we will reach the goal faster). This method suits well for the holonomic robots. We have used this method.

Algorithm:

Global variables: time\_step = 1sec, v\_min = 5 pixels/s, v\_max = 10 pixels/s, feaibility\_radius = 50 pixels.

While all the robots don’t reach their goals:

For ‘a’ in robots list:

Find the reachable velocities of ‘a’

For ‘b’ in robots list where a  b:

If ‘b’ in the feasibility radius of ‘a’:

Remove those velocities from ‘a’ list that lead to collision of ‘a’ with ‘b’ (collision cone approach)

From the updated reachable avoidance velocities list, choose a velocity such that the goal of ‘a’ is near

Make the robot ‘a’ go along this direction for the given time step.

If ‘a’ is near to its goal, ‘a’ has reached its goal.

\*\*\*\*\*\*\*\*\*\* THANKS \*\*\*\*\*\*\*\*\*\*