SC 627
Assignment 3: Velocity Obstacle Avoidance

Shubham Agrawal | 180040100

Implementation Details

To avoid any obstacle we need to construct a collision cone which tells us about the velocities which would lead to a collision with that given obstacle. Hence selecting any velocity outside this cone would ensure that we avoid the obstacle. Extending the idea when there exist multiple obstacle we can take multiple such obstacles and find there intersection to be the collision area and then sample a velocity vector outside this region.

We can impose actuator constraint on linear and angular velocity while sampling the velocity vector to ensure that we select a collision free as well as feasible velocity vector. There would exist multiple such velocity vector, selecting one from the same is based on a heuristic which ensure the bot reaches goal and takes lesser time (optimality not guaranteed).

To implement the above algorithm each obstacle is stored in form of the collision cone itself, that is collisioConeAxis, collisionConeOrigin and collisionConeAngle which are the axis connecting obstacle pose to bot pose, the velocity of the obstacle and the swept angle between the axis and the tangent respectively.

With the obstacle information and the odometry of the bot available the next velocity vector is sampled and choosen iteratively unless we reach with some bound of the goal.

For every step(sampling next velocity) I started with the direction of the goal. In this direction I found intersection of this ray starting from current velocity to the curve defining the reachable velocity (rectangle in this case as there is a acceleration limit in x and y direction). One that was found a uniform sampling was done starting from the point on the curve to the current velocity point. For each sampled point the conditions checked are collision free, within maximum angle deviation in robot heading and within Velocity max.The moment all condition is true for any of the sampled point that point is chosen as the next velocity vector. If no such velocity is found in the direction to goal, we then take a ray at Small angle deviation from that direction and repeat the above process.

With the above technique we ensure that the velocity vector we end up choosing is the maximum possible and with least deviation from the direction to goal. Hence we are always directed towards the goal with max velocity unless an obstacle requires us to deviate from the path. And avoidance of obstacle if guaranteed by the fact that collision cone velocities are not chosen while sampling.

A sampled point is within a collision cone or not is checked by finding the angle between the cone axis and the axis connecting the cone origin and the velocity vector. If the angle is less that cone angle it is within cone else it is not.

P.T.O

Simulation results

We get the following results upon simulation: