Report

Yash Patil

For this assignment, we have move from discrete path planning to continuous path planning where the kinematics of the vehicle should be considered for exploration. I first started by looking up ways to explore in a continuous space.

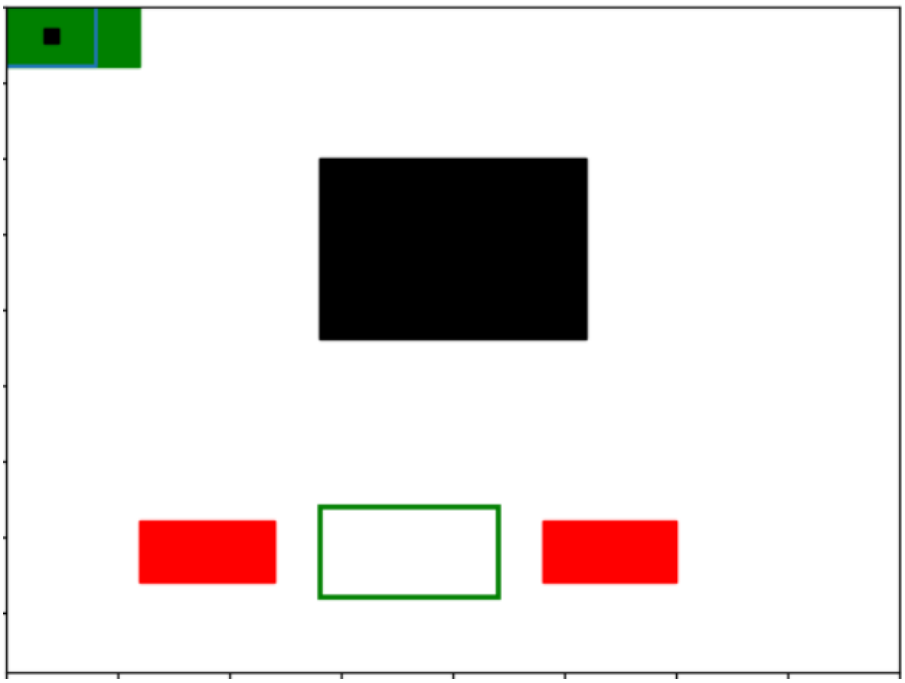
For my implementation I used hybrid A* algorithm that I referred from a blog linked in references. So, the idea here was to find the potential neighbors of the graph by moving fixed distance and using kinematic equations to calculate the possible points where the robot can travel next. And then calculate cost and heuristic for that neighbor. Then iterate over the neighbors, if the neighbor is valid i.e. at that position the robot is not colliding with any car or obstacle then move ahead and based on cost and heuristic continue the path further until the goal position is reached.

For collision checking, I simply checked if the 2 polygons are intersecting with each other, if they are then we can say that the two objects are colliding. I referred to a github repository for getting an insight on the approach.

For Differential Drive:

For this robot, I considered [5,2,1,0,-1,-2,-5] as a set of possible speed values for each wheel and I find point for each combination of the two wheels.

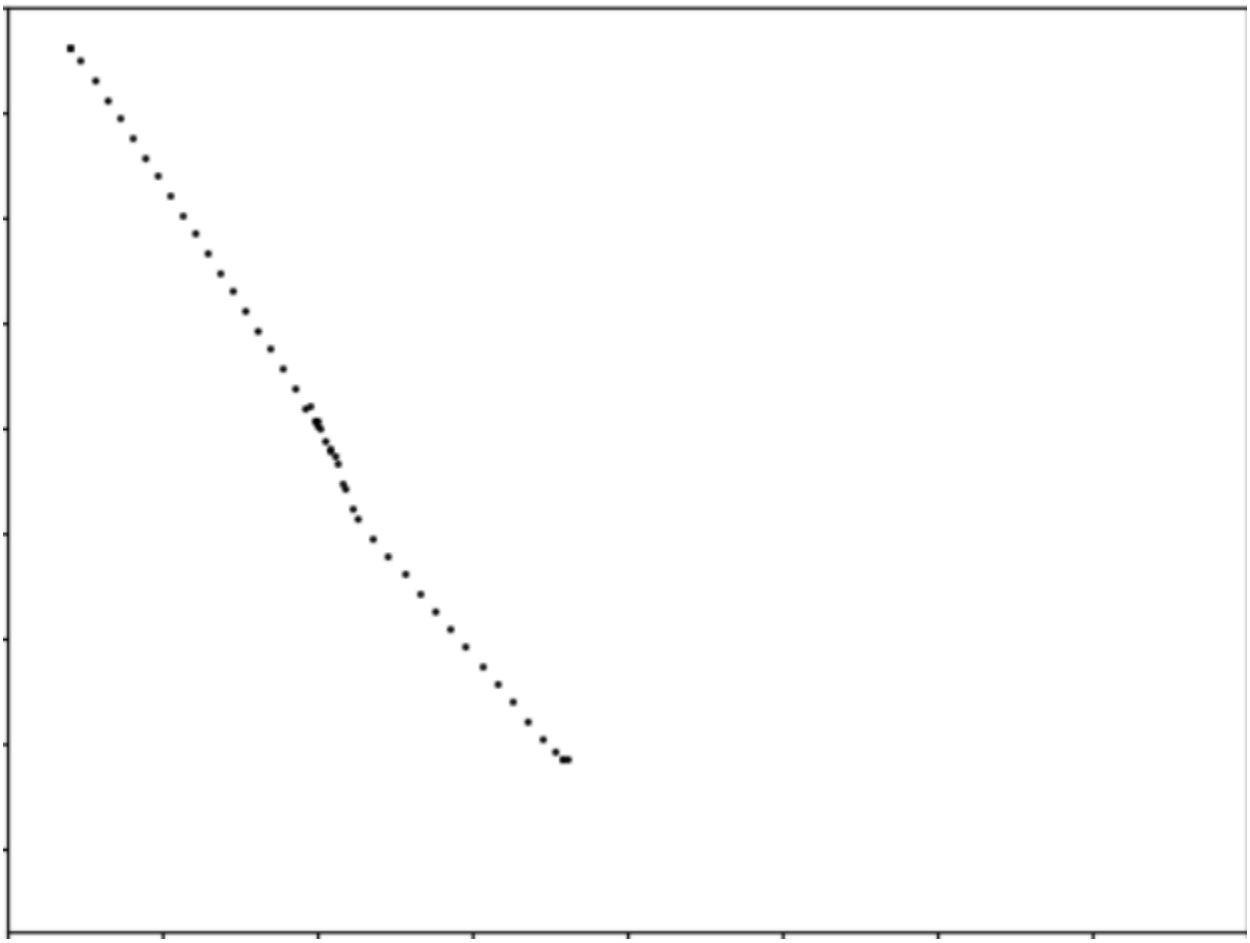
The environment:



Here the green are on upper left corner is the starting point and the hollow green rectangle in the bottom part is the goal position for the robot. The black rectangle is obstacle and two red rectangles are other cars.

The video of the robot following the path can be found in submitted files name "diff_drive.mp4"

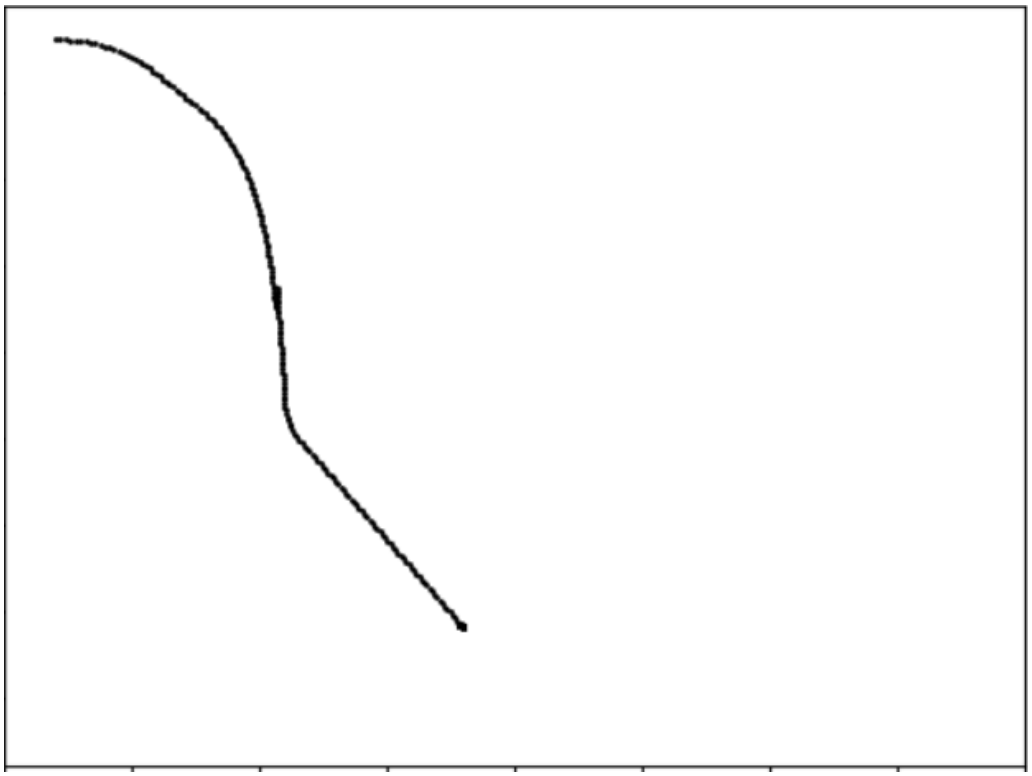
The path:



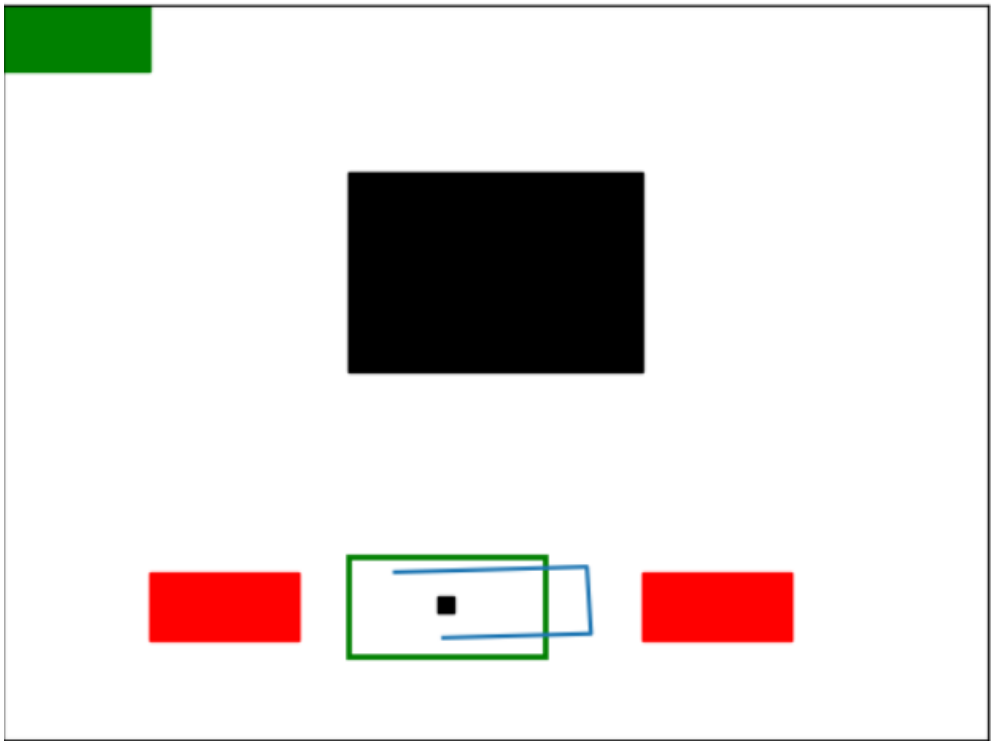
As we can see that robot almost followed a straight path just met with an obstacle in between and avoided it by turning.

For Ackermann Steering:

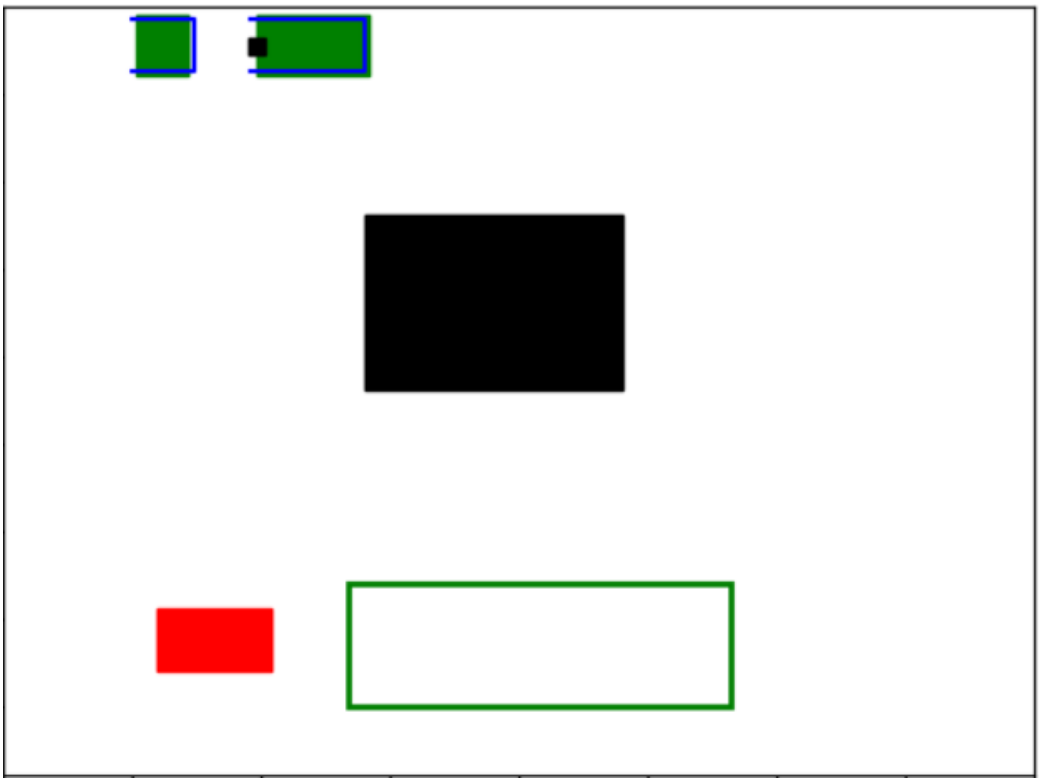
The basic skeletal code remained the same just the kinematic equations were changed to consider a steering angle and now the neighboring nodes were searched in space where steering angle ranged from 50 to -50 with an increment of 5.



The Path followed by the vehicle to reach the position.

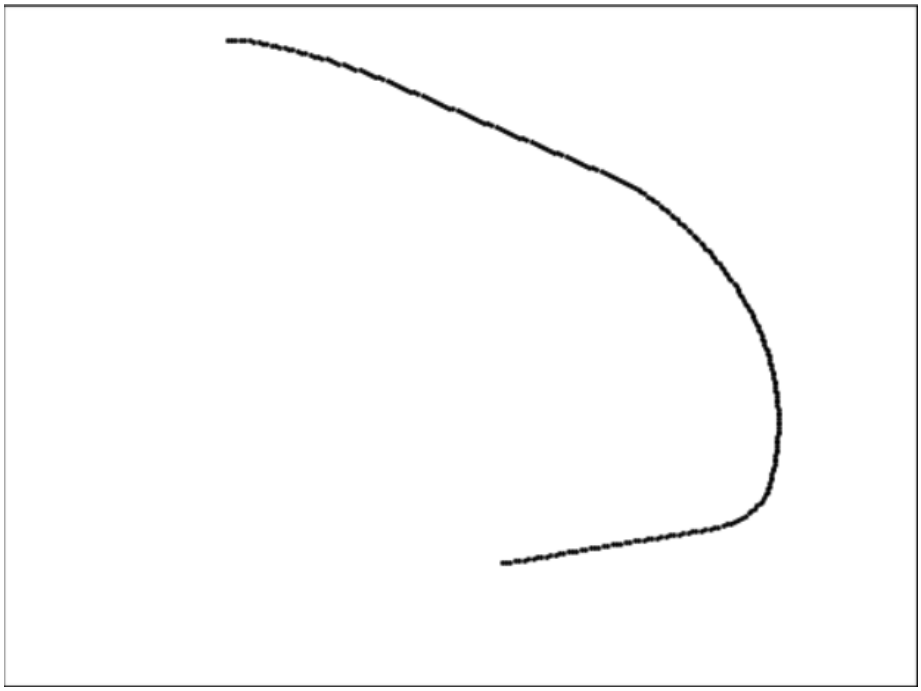


For Trailer:



Here we can see the starting point of the trailer and the goal position. I had to remove vehicle 2 to get it to work.

Path followed:



References:

- https://blog.habrador.com/2015/11/explaining-hybrid-star-pathfinding.html
- https://github.com/RitikJain12/Parking_different_vehcles