# CS 747: Programming Assignment 3 Report

## Task 1:

After a lot of deliberation and trying out different reinforcement learning and path planning algorithms like A*, which were not working for me, I implemented a simple controller in Task 1. Essentially, I turned the car on the spot and aligning the heading angle of the car such that it points to the centre of the road, I accelerated the car so that it reaches the road.

First, I derived the values of the x and y coordinates, velocity, and heading angle of the car from the state vector. I calculated the target angle, i.e. the angle which the heading angle of the car should align with to reach the goal. Then I defined the error between the heading angle and the target angle and used it to steer the car in the clockwise or anticlockwise direction. Once the heading angle of the car is almost equal to the target angle, with some tolerance/error bound, I accelerate the car at the maximum acceleration possible and thus it reaches the road/goal.

One problem that I faced here was that the car was taking the longer angle while turning, because of the discontinuity between the 0 degree and 360 degree angles. I solved this by tweaking the error between the heading angle and the target angle to incorporate this discontinuity.

## Task 2:

When I tried out a modified A* algorithm in Task 2, which gave me not the entire path to reach the goal but the next coordinate the car should move to reduce the Euclidean distance between the state and the goal, I was not able to design a controller which met the exact coordinates because of the discrete control inputs. The problem was that even if the algorithm gave me the next best coordinate, I was not able to accurately move because of the constrain of 3 degrees and also the discrete accelerations. So I implemented a simple controller.

Since the mud pits spawned in a defined area, I noticed that there was a gap of 40 units in the centre of the map, 20 above and below the x axis, and similar for the y axis. So, from the initial point of the car, I got the car to the x axis, and then turned it towards the road and accelerated simply. For this, I first derived the values of the x and y coordinates, velocity, and heading angle of the car from the state vector. I introduced some extra variables to keep track of the current trajectory of the car, i.e. if it is headed for the x axis (1st part) or the road (2nd part). Based on the initial y coordinate of the car, I defined the initial target angle as 90 degree, 270 degree or 0 degree. Then I defined the error between the heading angle and the target angle and used it to steer the car in the clockwise or anticlockwise direction. Once the car steers on the initial spawn position, I accelerate it at the maximum acceleration till it reaches close to the x axis, with a set tolerance. Once it reaches there, I decelerate at the maximum deceleration possible so that the velocity becomes zero in the next few timesteps. Once the velocity becomes zero, I again steer the car towards the road, keeping the acceleration zero. Once that is done, I accelerate at the maximum acceleration possible so that the car reaches the road.

I faced quite a few problems here. Most of them were code related issues, where the car was either not stopping at the x axis due to a very low tolerance set by me, or the car was stopping but not turning, or it was stopping and turning but not moving. I solved this by running the car on various random seeds and experimenting various approaches to see which one works, which finally resulted in the code I wrote. A problem that I tried to solve at the end but failed to do so were the cases when the car spawned below or above the mud pits, on the opposite side of the x axis.