Model Predictive Control -Reflections

The model

The model used is Kinematic model as described in the class. Following are the equations used:

$$x_{t+1} = x_t + v_t * cos(\psi_t) * dt$$

$$y_{t+1} = y_t + v_t * sin(\psi_t) * dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{L_f} * \delta * dt$$

$$v_{t+1} = v_t + a_t * dt$$

$$cte_{t+1} = cte_t + v_t * sin(e\psi_t) * dt$$

$$e\psi_{t+1} = e\psi_t + \frac{v_t}{L_f} * \delta_t * dt$$

The steering angle here is **delta** and throttle is **alpha.** Throttle here is a combination of brake and acceleration with values -1 and 1 respectively. Delta is kept between -25 to +25 degrees. This model gives us the prediction of the next state quite accurately using the equations above.

Lf is a pre-determined constant denoting the center of gravity of the vehicle.

x,y,psi and velocity are used to model the state of the car and errors **cte** and **psi_error** are the cross-track and orientation errors respectively. The vehicle coordinates x and y are in global co-ordinates which are converted to car co-ordinates.

Timestep length(N) and elapsed duration(dt)

Initially, I started with N = 25 and dt =0.05 as given in quiz. I thought 1.25 secs would be a good starting point, but I realized soon that higher N tended to swerve the car out of the track. I tried experimenting with lower values of N keeping dt constant. But keeping dt constantly low, the speed was very slow, and I thought as I will be incorporating latency , I should probably increase dt. After some experimentation, I settled for following values:

N =10

dt = 0.1

<u>Polynomial Fitting and MPC Preprocessing</u>

To make calculations simple, the waypoints are converted from global to car's coordinates using the following equations:

Waypoint_x = (ptsx – px) * cos(-psi) – (ptsy – py) * sin (-psi);

Waypoint_y = (ptsx – px) * sin(-psi) + (ptsy – py) * cos (-psi);

This is run in a loop for the length of the ptsx. This makes sure the vehicles coordinates are now at origin and psi is zero. The polynomial is fitted using 3^{rd} degree equation and cross track error is obtained by evaluation at this "fitted" coefficients

<u>Model Predictive Control with Latency</u>

Latency is introduced in the model as there will always be a delay between the command (brake, steering etc) and the action to be performed. To account for this, I predicted the car's positions and state using a 0.1sec delay on top of added 100 ms delay by default. I initially tried a value of 0.2s for latency, but it immediately became clear that latency > dt does not make much sense. I modified the state equations to account for this delay and fed it to my mpc.solve()