Model Predictive Control Project

The Model

The MPC model has three main components/variables,

1. Current State - The state consists of the below variables
   px - current location on the x - axis
   py - current location on the y - axis
   psi - current orientation or heading of the vehicle
   v - current velocity of the vehicle

2. Actuators - These are components that control the vehicle system, it consists of 2 controls as described below,
   steering angle (delta) - This is limited between -25 to +25 degrees in this project
   acceleration of the vehicle (a) - limited between -1 to 1, negative values indicates deceleration/braking, and positive values indicate acceleration

3. Update equations for the Kinematic model
   pred_px = px + v * cos(psi) * dt
   pred_py = py + v * sin((psi) * dt
   pred_psi = psi + v / Lf * (-delta) * dt
   pred_v = v + a * dt

   where dt - duration of each timestep
   Lf - length from the front of the vehicle to center of gravity

The MPC receives an array of waypoints in world space, which are then converted to vehicle coordinate space and then used to calculate the cross track error (cte) and orientation error (epsi). The MPC approximates the trajectory of the transformed waypoints using a 3rd order polynomial and does predictions of the N states and the N-1 actuator controls. The state of the vehicle in vehicle coordinate space is described as below:
state << pred_px, pred_py, pred_psi, pred_v, cte, epsi

Choosing N and dt:

The values chosen are N = 10, and dt = 0.1
These values of N and dt worked best in terms of computation time and driving safely around the track. The speed chosen is 50.
Choosing a much larger value of N, like 30 or 50 makes the car unstable and changes direction frequently causing it to go off track. So in my case based on some trial and error, found that N = 10 and dt = 0.1 works best.

Model Predictive Control with Latency:

The 100ms latency chosen simulates the latency between the sensor measurement and the processing this sensor data. We use Kinematic model to predict the state 100ms in the future with the equations as mentioned above, and then feed this state to the MPC solver