The Model

The kinematic model is composed from the following (x coordinate, y coordinate ,vehicle orientation, velocity and cte). The actuator output is the acceleration and steering angle.

In our calculation we use the previous state(t) to calculate next step(t+1). The model try to predict the trajectory through the following steps:

1. Set everything required by MPC
2. Define Vehicle Model and constraints(solve function)
3. Define upper and lower bound for the constraints
4. Calculate the cost function(we need to optimum this function)

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

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

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

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

$$cte_{t+1} = f(x_t) - y_t + (v_t * sin(e\psi_t) * dt)$$

$$e\psi_{t+1} = \psi_t - \psi des_t + (\frac{v_t}{L_f} * \delta_t * dt)$$

Timestep Length and Elapsed Duration (N & dt)

The values chosen for N and dt are 12 and 0.05, they are tuned manually using the fact that the more dt smaller the much better

When I tested 10 and 0.01 the car get out from the lane

Polynomial Fitting and MPC Preprocessing

We transform the coordinates from global to local, also we assume that the vehicle at the origin. Then we use Polyfit function to get coeffs(3 rd order equation)

Model Predictive Control with Latency

Latency is taken into account by constraining the controls to the values of the previous iteration for the duration of the latency. Thus the optimal trajectory is computed starting from the time after the latency period. This has the advantage that the dynamics during the latency period is still calculated according to the vehicle model.

Implementation

Check this video

https://youtu.be/fgzTZEOb8II

Discuss the points of the reviewer

Blue: Reviewer

Red: My answer

After trying different values, you should also try to give the reasoning behind your values e.g. explaining why some values work better than others. You can try answering the following questions:

1. Why smaller dt is better? (finer resolution

   A smaller dt will make a continuous approximation, this should make the error smaller between the predicted and reference trajectory

   “MPC attempts to approximate a continuous reference trajectory by means of discrete paths between actuations. Larger values of *dt* result in less frequent actuations, which makes it harder to accurately approximate a continuous reference trajectory. This is sometimes called "discretization error".

2. Why larger N isn't always better? (computational time)

N determines the number of variables optimized by the MPC. An optimizer will tune these inputs until a low cost vector of control inputs is found so the more value haven for N the more processing is done the more time taken by the optimizer, the bigger T we end up to have(this will cause inaccurate delta measurement as the value is enough to have after it a lot of change)

3. How does time horizon (N*dt) affect the predicted path? This relates to the car speed too.

T is the product of two other variables, *N* and *dt*, *T* should be a few seconds as the environment will be changed a lot after this time especially if the vehicle speed is high

The most popular choice is N=10, dt=0.1 which is a good starting point for experimenting other values

These values (Nd =10 and dt=0.1) means that the optimizer is considering a one-second duration in which to determine a corrective trajectory. However when this has been tested the car has deviated to the line so have to tune dt to much smaller 0.05