Udacity Term2 Project MPC Controller

The Model

The model is the Kinematic model that neglecting the complex of the tires and the road. The model update equations are:

```
x[t] = x[t-1] + v[t-1] * cos(psi[t-1]) * dt
y[t] = y[t-1] + v[t-1] * sin(psi[t-1]) * dt
psi[t] = psi[t-1] + v[t-1] / Lf * delta[t-1] * dt
v[t] = v[t-1] + a[t-1] * dt
cte[t] = f(x[t-1]) - y[t-1] + v[t-1] * sin(epsi[t-1]) * dt
epsi[t] = psi[t] - psides[t-1] + v[t-1] * delta[t-1] / Lf * dt
```

Where:

- `x, y` : Car's position.
- `psi` : Car's heading direction.
- `v` : Car's velocity.
- `cte` : Cross-track error.
- `epsi` : Orientation error.

Those values are the State of the model, and constant Lf is the distance between the car of mass and the front wheels, usually provided by oem. The two model outputs are:

- `the steering angle:` use to control the steering wheel
- `the acceleration:` use to control the throttle pedal

The objective is to find the both output values in the way it will minimize an objective function that is the combination of different factors:

In MPC.cpp file go to line from 55 to 71.

Some parameters are needed to make sure these values are turned smoothly, these parameters such as 1000, 50, and 5000 are manually turned by observing the vehicle driving behavior in the simulator. The baseline of turning these parameters is to keep the vehicle driving in the line not go out of it.

Timestep Length and Elapsed Duration (N & dt)

The number of points N and the time interval dt are selected at the beginning. These numbers impact the controller performance. N is not too big to keep the simulation fast, and 10 is a good choice. For dt, it should larger than the system latency as defined as 100ms in this project, so I choose 0.12s.

Polynomial Fitting and MPC Preprocessing

The waypoints provided by the simulator are transformed to the car coordinate system at line 110 to 115 in main.cpp. A 3rd-degree polynomial is used to filter the transformed waypoints. These polynomial coefficients are used to calculate cte and epsi. They are used by the solver as well to create a reference trajectory.

Model Predictive Control with Latency

To handle actuator latency, the state values are calculated and by introducing the latency parameter. Please see the line from 125 to 130 in main.cpp

Simulation

The vehicle must successfully drive a lap around the track.

The demo video of the vehicle driving can be found here:

https://www.youtube.com/watch?v=hfFQHQHvBzg&feature=youtu.be