CarND MPC Project

Jason Selfe

HARDWARE USED
MacBook Pro
2.7Ghz intel Core i5
8GB RAM
Intel Iris Graphics 6100 1536MB

COMPILATION
The code compiled all ok on my hardware.

IMPLEMENTATION
The Model

I employed the Kinematic model which doesn't use more sophisticated inputs like mass, gravity, tire profiles and forces, mainly as I knew the hyper parameter tuning is a tricky process, almost an art, and didn't want to get lost in too many parameters.
And the kinematic model will definitely produce a workable solution, which after a lot of tuning and spectacular crashes into the lake, it did get a good result.

State
The vehicle was set up as a state vector ingesting ;

Position : (x,y)
Heading : (psi)
Velocity : (v)

Actuators
The vehicle had two actuator inputs ingested ;

Steering angle : (delta)
Throttle/Brake combo : (a)

Update Equations
In the kinematic model we build the equations to use the current state with actuators and predict the state on the next time step, as outlined here;
(" Lf " is the UDACITY provided number for distance between front and center of gravity on vehicle)

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) + v(t)/Lf * delta * dt
v(t+1) = v(t) + a(t) * dt

then we can start making a cost function to supply to the MPC using cross track error and error of psi (epsi), which looks like this;

cte(t+1) = cte(t) + v(t) * sin(epsi(t)) * dt
epsi(t+1) = epsi(t) + v(t)/Lf * delta(t) * dt

Timestep Length and Elapsed Duration(N & dt)
So here we start playing with hyper parameters. The course outline suggested N=25 (which is 2.5seconds… e.g.; 100milliseconds times 25 = 2.5 sec)

I found after testing smaller and larger numbers that 2seconds or N=2 was enough to capture the future turns on the course.
For the timestep duration (dt) number i started with dt=0.1 but later on in the testing after finetuning all the other hyperparameters I saw a thread in the discussions forum talking about trying different numbers to help calculate latency better… sure enough tweaking this number i settled on dt=0.15 which made a great improvement to my model.

Polynomial Fitting and MPC Preprocessing
I took the waypoints data and moved it into cars coordinate space, then ran a polynomial fitting in the 3rd order.

Model Predictive Control with Latency
The supplied latency of 100milliseconds is there as simulation of the real world that might incorporate things like sensors gathering and processing data, or the response time a human driver might have.
The actual state of the car was moved into the 100ms latency space by using dt=0.15 and N=20 to help smooth and balance out the controller response. This was trail and error in the simulator to see how it would perform, but numbers were starting and moving around that 100ms latency number as a guide.

SIMULATION
The vehicle must successfully drive a lap around the track

Supplied a quicktime movie file called “ simulator_movie.mov ” that shows the code running the car around the track once successfully in the simulator.