Udacity SDCND Term 2 Model Predictive Controller Project

Davinder Chandhok

October 9, 2017

In this project I built a model predictive controller to control a car in a simulator. An additional challenge was to impose latency in the commands, making it necessary for the controller to actuate on the car considering a delay between the sending of the command and the actuation.

1 Model

The car's state is described by its position, orientation, velocity, cross-track error, and orientation error, represented by the following variables, respectively: $x_1, y_1, \psi_1, v_1, cte_1, e\psi_1$. The actuators are steering and acceleration, respectively, δ_t, a_t. The update equations are as follows:

$$
\begin{aligned}
x_{t+1} &= x_t + v_t * cos(\psi_t) * dt \\
y_{t+1} &= y_t + v_t * sin(\psi_t) * dt \\
\psi_{t+1} &= \psi_t + \frac{v_t}{L_f} * \delta_t * dt \\
v_{t+1} &= v_t + a_t * dt \\
cte_{t+1} &= f(x_t) - y_t + v_t * sin(e\psi_t) * dt \\
e\psi_{t+1} &= \psi_t - \psi des_t + \frac{v_t}{L_f} * \delta_t * dt
\end{aligned}
$$

2 Timestep

The timestep length was chosen to have the polynomial in the controller fit the available waypoints. A length of 5 was too short to accurately predict far enough out in front of the car for the velocity the car drove, while a length of 20 was too long and caused the controller to predict a crazy path, especially on curves, after going much further past the last waypoint. An elapsed duration between timesteps of 0.1 was chosen to predict frequently enough, but not be too computationally demanding.

3 MPC Preprocessing

In order to facilitate polynomial fitting, the coordinates were rotated by 90 degrees in lines 106-112 of main.cpp. This is so that the polynomial we are fitting will be a third order polynomial with dependent variable y and independent variable x.

Here, when considering a latency of 100 ms in the controller, the vehicle state is predicted out to 100 ms ahead of the vehicle's current state, so that the controller may create control signals based on the predicted state.

$$
\begin{aligned}
x &= x + v * cos(\psi) * latency \\
y &= y + v * sin(\psi) * latency \\
\psi &= \psi + \frac{v}{L_f} * \delta * latency \\
v &= v + a * latency
\end{aligned}
$$

4 Latency

Latency is dealt with as described in the previous section.