MPC

The vehicle model at the next time step (t+1) in relation to current time step is given by the following equations below. These equations assume SI units.

1. $x_{t+1} = x_t + v_t \cos(\psi_t)\, dt$
2. $y_{t+1} = y_t + v_t \sin(\psi_t)\, dt$
3. $\psi_{t+1} = \psi_t + \frac{v_t}{Lf}\, \delta_t\, dt$
4. $v_{t+1} = v_t + a_t dt$

5. $cte_{t+1} = f(x_t) - y_t + v_t \sin(e\psi_t)\, dt$
6. $\psi_{des_t} = \text{atan}(f'(x_t))$
7. $e\psi_{t+1} = \psi_t - \psi_{des_t} + \frac{v_t}{Lf}\, \delta_t\, dt$

Equation 5 and 7 are important to understand when we account for latency and handling different coordinates between the "world" and the car. Equation 5 describes cross-track error where f(x) describes Y-coordinate points as a function of X-coordinate points. Cross-track error is difference between our desired path (f(x)) and the car's perpendicular position. Equation 7 is like cross-track error; however, it is the difference between desired vehicle heading (ψ) and current vehicle heading. The desired heading is the slope, of the desired path, at point X.

I started with dt = 0.1 to match approximate delay with N = 20 steps. With 20 steps, the calculated trajectory was following the desired path for too long. This means that cost contribution after a certain N steps is negligible and a waste in computation. So, I decrease N to allow calculated trajectory to get to the desired path but not for too long. Based on the simulation, N = 12 steps looks good.

To account for delay, I used equations 1 through 4 to transform world points with respect to vehicle state after time delay. If the vehicle state was accurately calculated then desired path (yellow line) would not move away from center of the road. Our equations are simplified from the Physics in real life. I also looked at average delay time between calculate and set MPC initial state after average delay of 0.113 s. Since we fit f(x) to the delayed car position with respect to world points, cte and epsi is simplified to coefficients of f(0) and f'(0).