The goal of the project was to implement Model Predictive Control (MPC) so that a car simulator would drive around a track safely.

<u>The model:</u> *Student describes their model in detail. This includes the state, actuators and update equations.*

The state consisted of following parameters: vehicle's x-coordinate (x), vehicle's y-coordinate (y), orientation angle (psi), velocity (v), cross-track error (cte), and psi error (epsi).

Actuators consisted of throttle (acceleration) and steering (delta).

The model uses following formulas to update to the next timestep.

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

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

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

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

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

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

<u>Timestep Length (N) and Elapsed Duration (dt)</u>

For N and dt values, I tried a few different combinations. I initially started out with N=10 and dt=0.1 since that seemed to be most common values suggested on the forum. This was okay at lower speeds but turns were little too sharp and nearly touched edges at higher speeds. I tried increasing N to 15, and I did not observe noticeable improvement. When I lowered the N to 8, sharps turns still remained. After a few other trials, I felt N=8 and dt=0.08 provided best result.

<u>Polynomial Fitting and MPC Preprocessing</u>

The waypoints are transformed to the vehicle's coordinate system. This makes the vehicle's coordinate to be (0,0) and the psi to be also zero. Then, a third order polynomial is fitted to the waypoints.

<u>Model Predictive Control with Latency</u>

To account for the latency delay, a 100ms latency was implemented. To implement this, state values are calculated using the latency delay. The impact of latency was definitely observable at higher speeds.