Project Rubric

- **Student describes their model in detail. This includes the state, actuators and update equations?**

We use the kinematic model here. The variables used are

x coordinate

y coordinate

orientation angle psi,

velocity v,

cross-track error cte

psi error epsi

Actuator outputs:

acceleration a

steering angle delta

The model combines the state and actuations from the previous timestep to calculate the state for the current timestep.

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

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

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

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

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

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

- **Timestep Length and Elapsed Duration (N & dt)**

The values chosen for N and dt are 10 and 0.1. The other values tried are 25/0.05, 10/0.05. From those other values the values chosen were working fine.

- **Polynomial Fitting and MPC Preprocessing**

This is done in main.cpp. The waypoints are preprocessed by transforming them to the vehicle's perspective

- **Model Predictive Control with Latency**

The idea for handling latency is taken from jeremy-shannon github.

This is implemented in the code mpc.cpp. in the line 109-113 and 67.

the original kinematic equations depend upon the actuations from the previous timestep. There is a delay of 100ms (which happens to be the timestep interval) the actuations are applied another timestep later. The code in those lines are implemented to overcome this delay.