# Model Predictive Controller Project

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

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

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

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

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

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

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

This kinematic model contains total 6 states.
‘x’ and ‘y’ positions and the orientation of the car as heading information represented by symbol psi.
It also contains cross track error and error in psi, as states.
In this model the actuations from the previous time step are added to the current state.

## *Student discusses the reasoning behind the chosen N (time step length) and dt (elapsed duration between time steps) values. Additionally the student details the previous values tried.*

For ‘dt’ to be very small we might be needing a super computer to compute the enormous number of points for even computing a 1 second of real time.
I once tried ‘dt’ to be 0.03 and it took me a lot of time for the simulation, and same experience goes for higher values of N. Tuning either ‘dt’ or N often produced unpredictable erratic behavior.

## *A polynomial is fitted to waypoints.*

The waypoints gathered, are processed by using the transformation matrix to transform into vehicle body axis. So, the shifting of origin was necessary to simplify the fitting problem.

## *The student implements Model Predictive Control that handles a 100 millisecond latency. Student provides details on how they deal with latency.*

As the simulator has 0.1 s delay so we had to incorporate 100 ms delay. So we have incorporated by actuator commands of steer and throttle by incrementing the index of time step. As simulation runs later.