Introduction

This document provides a brief explanation of key points as outlined in the project rubric.

The Model

The Kinematic model is characterized by the State variables, Actuator variables, and update equations.

The table below lists state variables describing the vehicle's kinematic state.

State Variables	
Variable	Description
x	X coordinate
y	Y coordinate
Ψ	orientation angle
v	Velocity in meters / second
cte	Cross track error
e_Ψ	Orientation angle error

The vehicle in this project has two actuators. Their input values are described in the table below

Actuator Variables	
Variable	Description
δ	Steering angle
a	Acceleration value

Below are equations that model the way the state variables are updated after each actuator input *a* is applied over a specified time interval *dt*.

$$x_{t+1} = x_t + v_t \cdot \cos(\Psi_t) \cdot dt$$

$$y_{t+1} = y_t + v_t \cdot \sin(\Psi_t) \cdot dt$$

$$\Psi_{t+1} = \Psi_t + \frac{v_t}{L_f} \cdot \delta \cdot dt$$

$$v_{t+1} = v_t + a_t \cdot dt$$

$$cte_{t+1} = (f(x_t) - y_t) + v_t \cdot \sin\left(err_{\Psi_t}\right) \cdot dt$$

$$e_{\Psi_{t+1}} = (\Psi_t - \arctan(f'(x_t))) + \left(\frac{v_t}{L_f} \cdot \delta_t \cdot dt\right)$$

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

N represents the total number of future points the algorithm will pass to the solver. *dt* represents the elapsed time between the aforementioned future points. I ended up choosing a value of 10 for *N* and 0.05 for *dt.* I chose 0.05 because it is ½ of the system latency specified in the project. I felt like this number would allow me to correct for the system latency in a more granular manner. I tried the following higher numbers for N, 20 and 40. The system performed really poor with 40. With this value, the optimizer seemed to give a strange output. This may have been caused by additional latency introduced by the optimizer itself. The value 20 worked much better, but I notice slight oscillations about the centerline. I believe this was also due to additional latency caused by the solver. Setting N to 10 seemed to give satisfactory results so I stopped there. I also tried setting *dt* to 0.10. This worked ok, but the shorter duration seemed to yield better results.

Polynomial Fitting and MPC Preprocessing

As described in class, a 3rd degree polynomial is fit to waypoints obtained from the simulator. The coefficients of this polynomial are ultimately passed to the optimizer. Since the waypoints are in the map coordinate system, they are first converted to the vehicle's coordinate system. This conversion includes a translation and rotation. As a result the x, y, and orientation angle state variables passed to the algorithm are always 0.

Model Predictive Control with Latency

The system has a 100 millisecond latency period. To deal with this, I used the actuator values of the third timestamp instead of the first. In my implementation this would be the actuation commands projected out 150 milliseconds. (*dt* = 50 ms) One may ask, why not use the commands at 100 milliseconds. The answer is, in my case 150 milliseconds yielded smoother driving on my system. Perhaps this is because there was additional latency in my system. The simulator itself may have some latency on some computers depending on graphics settings or other things.