MPC Project - Ahmad Chatha

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

The model consists of the x and y position of the vehicle along with the velocity, angle of orientation (psi), the cross-track error (cte) and the orientation error(epsi). The actuators are the steering angle and the throttle value. Following equations are used to calculate the next state of the model:

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

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

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

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

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

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

2) Student discusses the reasoning behind the chosen N (timestep length) and dt (elapsed duration between timesteps) values. Additionally the student details the previous values tried.

The final values that were chosen are N = 25 and dt=0.03 through trail and error. The values for N and dt were chosen in order to maximize the prediction accuracy and minimize the calculation time.

Other combinations were tried such as (10, 0.1) (20, 0.05), (15, 0.1), (25, 0.1). Larger values of N increases the complexity of the optimization problem and increases the calculation time. Larger values of dt does't allow the car to recover from the mistakes. For example: For smaller and larger values of N (10, dt=0.1) and (30, 0.05), the car just eventually drives of the track.

3) Polynomial Fitting and MPC Preprocessing

Before fitting the polynomial, the points were converted from map coordinates to vehicle coordinates. After the conversion, we fit a 3rd order polynomial.

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

The latency was handles by predicting the state of the car 100 ms in the future. The code for that is in main.cpp around lines 121-129.