1 How MPC works?

Controller gets reference path from path planner and it's job is to follow it. In our case the path is represented as waypoints, so we transform it to slightly more abstract representation by fitting a polynomial to it.

Additionaly, the controller has a model of behaviour of the car. Using this model it can simulate what will happen with the car given current state and planned actuations. Based on this, within a window of prediciton, the controller chooses actions that minimize the cost. The cost is composed of many parts, penalizing various aspects of controller behaviour:

- Most importantly, the "distance" from reference wanted state:
  - distance from center of the track
  - distance from reference yaw
  - distance from reference velocity
- Usage of actuators
- Rate of change in usage of actuators

In order for controller to be successful, a balance has to be struck between all of these constraints.

2 Model

The vehicle state contains 6 variables:

- x, y - position coordinates
- ψ - heading angle
- v - velocity
- cte - cross-track error
- $e\psi$ - heading angle error

Controlable actuators are:

- δ - steering angle

- a - acceleration

We update state based on simple kinematic model presented in the lectures:

$$x_{t+1} = x_t + v_t \cdot cos(\psi_t) \cdot dt \tag{1}$$
$$y_{t+1} = y_t + v_t \cdot sin(\psi_t) \cdot dt \tag{2}$$
$$\psi_{t+1} = \psi_t + \frac{v_t}{L_f} \cdot \delta_t \cdot dt \tag{3}$$
$$v_{t+1} = v_t + a_t \cdot dt \tag{4}$$
$$cte_{t+1} = f(x_t) - y_t + v_t \cdot sin(e\psi_t) \cdot dt \tag{5}$$
$$e\psi_{t+1} = \psi_t - \hat{\psi}_t + \frac{v_t}{L_f} \cdot \delta_t \cdot dt \tag{6}$$

The only preprocessing I do is transforming the waypoints into the vehicle coordinate system.

2.1 Choice of N and dt

Part of the project is choice of the parameters N and dt. Following the sugestion in the project statement, I have chosen $N = 10$ and $dt = 100$ ms. Choosing dt equal to the expected latency allows me to simplify how I handle latency a lot.

2.2 How to deal with latency?

I have chosen a very simple way of dealing latency. While calculating state, instead of using actuation values from one timestep before, I use actuations from two timesteps before. Timestep in MPC is chosen to be equal to expected latency of 100 ms.