Model Predictive Control(MPC)

Chen, Zhuhui

August 23, 2017

Contents

1 Rubric Points

1.1 The Model

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

The model I used is much like the one in the lecture. In the model there're 6 states and 2 acturator inputs:

- States:
  - x - x component of location of the vehicle, in vehicle coordinate
  - y - y component of location of the vehicle, in vehicle coordinate
  - ψ - angle between the current orientation of the vehicle and the initial longitudinal direction of the vehicle
  - v - longitudinal velocity of the vehicle
  - cte - cross track error
  - e_ψ - orientation error
- Inputs:

- δ - acturator for turning
- a - acturator for trottle

The update equations for the model are defined as following:

$$
\begin{align}
x_{t+1} &= x_t + v_t * \cos(\psi_t) * \Delta t \tag{1}\\
y_{t+1} &= y_t + v_t * \sin(\psi_t) * \Delta t \tag{2}\\
\psi_{t+1} &= \psi_t + \frac{v_t}{L_f} * \delta_t * \Delta t \tag{3}\\
v_{t+1} &= v_t + a_t * \Delta t \tag{4}\\
\text{cte}_{t+1} &= \text{cte}_t + v_t * \sin(e_{\psi_t}) * \Delta t \tag{5}\\
&= (y_t - y_t^{ref}) + v_t * \sin(e_{\psi_t}) * \Delta t \tag{6}\\
e_{\psi_{t+1}} &= e_{\psi_t} + \frac{v_t}{L_f} * \delta_t * \Delta t \tag{7}\\
&= (\psi_t - \psi_t^{ref}) + \frac{v_t}{L_f} * \delta_t * \Delta t \tag{8}\\
y_t^{ref} &= f(x_t) \tag{9}\\
\psi_t^{ref} &= \arctan(f'(x_t)) \tag{10}
\end{align}
$$

where f is the polynomial fit for waypoints. $x_0 = 0, y_0 = 0, \psi_0 = 0$ since I'm using vehicle coordinate. v_0 is set to the speed feedback from the simulator.

1.2 Timestep Length and Elapsed Duration(N & dt)

> 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.

Following with the MPC to line quiz, the initial N and dt I chose was 25 and 0.05. This was not a bad choice for driving slowly in this project. When I turned to higher speed, things got wrong.

The reasons for this are:

1. There's 100ms latency. The decision from MPC can only be actually carried out after 100ms. For setting dt=0.5, MPC would get wrong prediction of how the vehicle would be in the next coming future.

2. The reference trajectory comes from polynomial fitting for 6 waypoints(that's what the simulator give). The trajectory is only effective within the paths which the 6 waypoints cover. At higher speed, the greater N is, the deeper the future paths MPC will look into. When the paths MPC is predicting exceeds the effective paths, cte and e_ψ are no longer meaningful. Not suprisingly, MPC will give out a wrong path.

So I've reduced N and increased dt for my MPC implementation. The final N I used is 10 and dt is 0.1 + 0.05.

1.3 Polynomial Fitting and MPC Preprocessing

> A polynomial is fitted to waypoints.
>
> If the student preprocesses waypoints, the vehicle state, and/or actuators prior to the MPC procedure it is described.

Waypoints are transformed from map coordinate to vehicle coordinate before polynomial is fit. For two reasons polynomial is better fit in vehicle coordinate:

1. The fit polynomial is a single-value function in the form $y = f(x) = a_n * x^n + a_{n-1} * x^{n-1} + \cdots + a_1 * x + a_0$. f is a single-value function implies 1 value in codomain maps to at most 1 value in domain. In map coordinate, this is not an easy condition. While in vehicle coordinate, this can be meet without many difficulties as long as there is no extreme sharp turns and e_ψ won't getting to large.

2. The formulas for cte and e_ψ will be more involed in map coordinate. Take how to calculate the initial cte_0 for example. In map coordinate, that would be the euclidean distance between (x_0, y_0) and (x_0^{ref}, y_0^{ref}) :

$$\sqrt{(x_0 - x_0^{ref})^2 + (y_0 - y_0^{ref})^2}$$

   In vehicle coordinate, that would simply be $y_0^{ref} = f(x_0^{ref}) = f(x_0)$.

In vehicle coordinates, the initial states for vehicle are very simple. The vehicle is now located in the origin, with head direction matching the orientation of the x-axis, so $x_0 = 0, y_0 = 0, \psi_0 = 0$. v_0 is longitudinal speed so no conversions needed. Actuator values are not affected by coodinate system changing.

1.4 Model Predictive Control with Latency

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

The net effect of 100ms latency is, the decision from MPC can only be carried out after 100ms. I modeled the latency into the model in the following way.

Supposed in timestep t , the inputs are δ_t and a_t . With the existence of latency, the vehicle will keep moving for 100ms with δ_t and a_t . If cooperating the latency into dt, say, $dt' = dt + \text{latency}$, the MPC will then use inputs at timestep t and predict the trajectory after dt' seconds, finding an optimal action (δ_{t+1} and a_{t+1}) to perform at timestep $t+1$ and issue at timestep t. Since $dt' > 100\text{ms}$, the action issued at t can be effectively carried out at $t+1$. That should solve the latency problem.

Besides, I also takes inputs from MPC previously solved and set them as fixed initial inputs because now the MPC can only effectively change future inputs. Inputs at timestep $t = 1$ are used for actuators feeding to the simulator instead of those at timestep $t = 0$ (the initial timestep).