Note on Model Predictive Control Project

1. Outline

In this short report, I explain some key points of my implementation of the model predictive control (MPC). The performance of the model I built can be seen in the videos, `https://youtu.be/dryBAmfa6ZY` (reference velocity, $v_{ref} = 80$) and `https://youtu.be/h9dsa6Rnf5w` (faster reference velocity, $v_{ref} = 100$).

2. Description of Model

2.1. Kinematic Model

In this project, I considered a simple Kinematic model whose state at time step t is denoted by (the subscript t of the variables denotes that they are the values at time step t)

$$(x_t,\, y_t,\, \psi_t,\, v_t, cte_t, \psi_t)\,.$$

Here (x_t, y_t) is the position of the vehicle in the two-dimensional Cartesian coordinate, ψ_t is the orientation angle of the vehicle, v_t is the velocity, cte_t is the cross-track error (CTE) and $e\psi_t$ is the orientation error. Time evolution is controlled by the following equations:

$$x_{t+1} = x_t + (v_t\, dt)\, \cos\psi_t\,, \quad y_{t+1} = y_t + (v_t\, dt)\, \cos\psi_t\,, \quad \psi_{t+1} = \psi_t - \left(\frac{v_t\, dt}{L_f}\right)\, \delta_t\,, \quad v_{t+1} = v_t + a_t\, dt\,,$$

$$cte_{t+1} = f(x_t) - y_t + (v_t\, dt)\, \sin\psi_t\,, \quad e\psi_{t+1} = \psi_t - \psi des_t - \left(\frac{v_t\, dt}{L_f}\right)\, \delta_t\,.$$

Here the actuators (δ_t, a_t) are the steering angle and acceleration, dt is the time elapse during a single time step, L_f (set to 2.67 throughout this project) is the distance between the center of mass of the vehicle and its front axle, $f(x_t)$ is the fitting polynomial $f(x)$ evaluated at x_t and $\psi des_t = \arctan f'(x_t)$.

2.2. Cost Function

In MPC, the optimal values of the actuators for the next time step are determined by minimizing a cost. For this purpose, I designed the cost as follows:

$$\begin{aligned}(\text{cost}) \quad = \quad & \sum_{t=0}^{N-1} \left[20\,(cte_t)^2 + 100\,(e\psi_t)^2 + 0.02\,(v_t - v_{ref})^2\right] \\ & + \sum_{t=0}^{N-2} \left[1000\,(\delta_t)^2 + (a_t)^2\right] + \sum_{t=0}^{N-3} \left[5000\,(\delta_{t+1} - \delta_t)^2 + 5\,(a_{t+1} - a_t)^2\right]\,.\end{aligned}$$

Here N is the number of time steps used for MPC (the prediction horizon is given by $T = N dt$ and the choice of N and dt is discussed below) and v_{ref} is the reference velocity (set to 80 or 100 in this project). I note that the coefficient of $(\delta_{t+1} - \delta_t)^2$ term is large. In the case with a small coefficient for this term, the vehicle oscillates a lot.

3. Choice of Parameters N and dt

In order for the vehicle to be controlled well with MPC, the prediction horizon $T = Ndt$ needs to be tuned carefully. I tried $T = 0.1$, 1, 2 and tried $N = 5$, 10, 15, 20, 30 for each T (note that the time elapse for each time step is given by $dt = T/N$). A small prediction horizon is not enough for predicting the next time step. On the other hand, when T is large, the vehicle takes into account the information very away (which is inaccurate at that time step) and thus the vehicle could not be controlled properly. For N, a small value leads to an unstable behavior of the vehicle (too small input for prediction), while a large value requires more numerical costs. I thus started with a small N and gradually made it large. Then I have chosen the smallest N with which the vehicle drives safely and stably. In the end, I set these parameters to $(N, dt) = (20, 0.05)$ (i.e. $T = 1$).

4. Polynomial Fitting and MPC Preprocessing

For the fitting function $f(x)$, I considered a third-order polynomial and fitted the coefficients by using a set of the waypoints. The third-order polynomial worked well, and thus I did not use more complicated fitting function.

Before fitting the polynomial, I carried out the change of the coordinate for the waypoints. Since the waypoints are originally given in the map coordinate, I carried out the transformation such that the waypoints are given in the vehicle frame (in which the vehicle is located at the origin and moves in the positive x-direction). Please refer to the lines 101-108 of `main.cpp` for the implementation.

5. Dealing with Latency

The 100-millisecond latency for the actuators was dealt with by setting the input state of of the vehicle for MPC to be the one after 100 milliseconds. Please refer to the lines 124-130 of the `main.cpp` for the implementation.