MPC Project Report

October 2, 2017

1 The Model

The state of the vehicle at time t consists of (x_t, y_t, ψ_t, v_t) where (x_t, y_t) is the location of the vehicle, ψ_t is the orientation of the vehicle, v_t is the velocity of the vehicle. The actuators are (δ_t, a_t) where δ_t is the steering angle, a_t is the acceleration. The kinematic model of the vehicle is the follwoings:

$$\begin{aligned}
x_{t+1} &= x_t + v_t cos(\psi_t) dt \\
y_{t+1} &= y_t + v_t sin(\psi_t) dt \\
\psi_{t+1} &= \psi_t + \frac{v_t}{L_f} \delta_t dt \\
v_{t+1} &= v_t + a_t dt
\end{aligned}$$

, where dt is the time step, Lt is the distance between the front of the vehicle and the center of gravity.

Let ψ_{des_t} be the desired ψ. This is the derivative of polynomial, $f'(x)$. The cross track error between the path and the vehicle's position is computed by the following equation

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

,where $e\psi_t = \psi_t - \psi_{des_t}$.

Orientationtion error is computed as follows:

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

2 Time step Length and Elapsed Duration

The final values are $N = 14$ and $dt = 0.07$.

2.1 The effect of N

When we used big numbers for N (such as $N \geq 20$), the computation for the optimization took time, and we had an additional latency to the system. In this case, the vehicle was not able to respond to any situations quickly, which led the vehicle to go off the road.

2.2 The effect of dt

When we used the small numbers for dt such as $dt \leq 0.05$, we needed a bigger N to plan a longer horizon. Then, the optimization took a long time to find a good solution. When we kept N to be small (Ndt is also small), the MPC tries to plan the control inputs that work for a very short path. This did not work especially for the steep curves where the longer planning is necessary.

When we used $dt \geq 0.1$, the MPC sometimes could not find reasonably good control inputs. This is because we have a big step in the optimization, and it is hard for the optimization algorithm to find good control inputs with such a big step.

3 Polynomial Fitting and MPC Preprocessing

We first transformed the way-points in the world frame into the vehicle's frame. Then, we fit the way-points to a third order polynomial.

$$f(x) = ax^3 + bx^2 + cx + d$$

We use the vehicle's state in the vehicle's frame for MPC.

4 Model Predictive Control with Latency

We have a 100 ms latency. In order to deal with the issue, we first update the current state to the state at $t + 100ms$. We use this state as the initial state for MPC. Thus, the control inputs computed for the state at $t + 100ms$ will be executed at around $t + 100ms$.
We used the following cost functions for MPC.

The cost based on the reference state:

$$J_1 = w_1 \sum_{t=0}^{N-1} cte_t^2 + w_2 \sum_{t=0}^{N-1} e\psi_t^2 + w_3 \sum_{t=0}^{N-1} (v_t - v_{ref})_t^2$$

The penalty on the use of actuators:

$$J_2 = w_4 \sum_{t=0}^{N-2} \delta_t^2 + w_5 \sum_{t=0}^{N-2} a^2$$

The penalty on the change of the actuations:

$$J_3 = w_6 \sum_{t=0}^{N-3} (\delta_{t+1} - \delta_t)^2 + w_7 \sum_{t=0}^{N-3} (a_{t+1} - a_t)^2$$

The total cost for MPC is:

$$J = J_1 + J_2 + J_3$$

We tuned the parameters $w_1, \cdots, w_7$ by hand. We used $w_1 = 1.0, w_2 = 20.0, w_3 = 1.0, w_4 = 1.0, w_5 = 1.0, w_6 = 4000.0, w_7 = 1.0$.
We focused on adjusting w_2 and w_6. This is because w_2 and w_6 are important to generate smooth steering angle transitions.

Ideally, the parameter search should be done using twiddle and other optimization methods. As a future work, we will apply the optimization techniques (for example twiddle and gradient descent) to find better parameters as we did in the PID project.