Author: Siu Kei, Muk

Date: 6 July, 2017

The Model

State

The state of the model is a 6-element long vector: $[x, y, \psi, v, cte, e\psi]$, where (x, y) indicates the vehicle's position, ψ describes the angle from the x-axis increasing counter-clockwisely. cte is the cross track error, which is the displacement of the vehicle from the reference curve. $e\psi$ is the orientation error.

The actuator is represented by a 2-element vector $[\delta, a]$ where δ is the steering angle (limited to [-rad(25), rad(25)]) and a is the acceleration (limited to [-1, 1]).

The reference curve is approximated by a polynomial of degree 3, f.

Update

The update equations (line 114 through 119 in `MPC.cpp`), are as follows:

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

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

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

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

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

where L_f is the length of the vehicle.

Preprocessing

Each time when the MPC receives the vehicle's information $(px, py, \psi, v, \delta, a)$ and the way points (lines 125-132 in `main.cpp`), the way points are first converted to the vehicle's coordinate system by applying first a translation by $(-px, -py)$, then a rotation over $-\psi$ (lines 143-146, and 71-80 in `main.cpp`):

$$\begin{pmatrix} x_T \\ y_T \end{pmatrix} = \begin{pmatrix} \cos(-\psi) & -\sin(-\psi) \\ \sin(-\psi) & \cos(-\psi) \end{pmatrix} \begin{pmatrix} x_w - px \\ y_w - px \end{pmatrix}$$

where (x_w, y_w) is the original way point, (x_T, y_T) is the transformed way point.

Polynomial Fitting

The resulting reference points are then used to fit a degree-3 polynomial, f (lines 150-152 in `main.cpp`). The polynomial fitting is done as follows (lines 50-69 in `main.cpp`):

1. Define an n-by-4 matrix A, in which each row a_i is the vector $[1, x_i, x_i^2, x_i^3]$ for each way point (x_i, y_i).
2. Define a vector consisting of all the y-coordinates of the way points $(y_1, y_2, ..., y_n)$
3. Solve the least square problem

   $$\min \|Az - y\|^2$$

   where $z = (z_0, z_1, z_2, z_3)$ is the vector of coefficients of the resulting polynomial $\sum_{k=0}^{3} z_k x^k$, using QR decomposition.

Errors Computation

The errors $cte, e\psi$ are then computed using f and Newton's method (lines 82-102 in `main.cpp`) with 5 iterations (line 156 in `main.cpp`):

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n) + x_n}{f'(x_n)^2 + f(x_n)f''(x_n) + 1}$$

$e\psi$ is calculated by $e\psi = -\arctan(f'(x))$ where x is the point obtained from the Newton's method above (line2 101-102 in `main.cpp`). The first term vanishes as the vehicle's coordinate system is used, the vehicle always has $\psi = 0$.

The reason why Newton's method is used instead of taking the constant term of the fitted polynomial is that the coefficients of the polynomial depends on the coordinates system used, but the CTE should stay constant no matter which

direction the vehicle is heading as long as the vehicle's position is not changed. The approach above calculates the distance between the curve of the fitted path and the vehicle. The derivation is as follows:

1. Let the vehicle be at the origin (as we are using the vehicle's coordinate system), and f be the polynomial fitted to way points from above.
2. We want to find a point $p_f = (x_f, y_f)$ on the curve such that the line joining p_f and the original is orthogonal to the curve at p_f.
3. Mathematically, (x_f, y_f) satisfies:

   $$\frac{y_f}{x_f} f'(x_f) = -1$$

   which is equivalent to

   $$f(x_f)f'(x_f) + x = 0$$

   as $y_f = f(x_f)$.
4. Let $g(x) = f(x_f)f'(x_f) + x$, applying the Newton's method formula for one variable, we have

   $$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} = x_n - \frac{f(x_n)f'(x_n) + x_n}{f'(x_n)^2 + f(x_n)f''(x_n) + 1}$$

Latency Handling

The latency is handled by first predicting the state of the vehicle after 0.1s, using the current control values (lines 161-166 `main.cpp`), and this prediction together with the coefficients of f are used as the initial state in the optimal actuation computation.

Choice of N and dt

dt is chosen according to the latency as every time the command takes about 0.1s until it gets executed. Therefore it would be beneficial for the optimizer to take this into account while performing the simulations, which would model the situation more realistically.

N is chosen so that the horizon (1s) is not too far ahead, but not too close. This has an effect that the controller could have a greater focus on the near future, while limiting the effect of noise, such as floating point errors in computation and measurement errors.

Simulation

Please refer to the video (`mpc.mp4`) attached in the repository for the simulation result on the development machine.