DPC for IHM

Tudor Oancea

Jun 2024

1. System definition

We consider a simple kinematic bicyle model with the classical 4 DOF state $x = (X, Y, \varphi, v)$ and control input $u = (T, \delta)$, where X, Y are the position of the car, φ its heading, v its absolute velocity, T the throttle command, and δ the steering angle. The dynamics are given by the following ODE:

$$\begin{aligned} \dot{X} &= v\cos(\varphi+\beta), \\ \dot{Y} &= v\sin(\varphi+\beta), \\ \dot{\varphi} &= v\frac{\sin(\beta)}{L}, \\ \dot{v} &= \frac{F_x}{m}, \end{aligned} \tag{1}$$

where $\beta = \frac{1}{2}\delta$ denotes the kinematic slip angle, L the wheelbase, m the mass of the car, and $F_x = C_m T - C_{r0} - C_{r1} v_x - C_{r2} v_x^2$ the longitudinal force applied to the car.

In the following, we will denote $n_x = 4$ the state dimension, $n_u = 2$ the control dimension.

2. NMPC formulation

The optimal control problem (OCP) designed to track a state reference $\left\{x_k^{\text{ref}}\right\}_{k=0}^{N_f}$ starting from a current state x_0 reads:

$$\begin{aligned} \min \quad & F\left(x_{N_f}, x_{N_f}^{\text{ref}}\right) + \sum_{k=0}^{N_f-1} l\left(x_k, u_k, x_k^{\text{ref}}\right) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k),\ k = 0, ..., N_f - 1, \\ & -T_{\max} \leq T_k \leq T_{\max},\ k = 0, ..., N_f - 1, \\ & -\delta_{\max} \leq \delta_k \leq \delta_{\max},\ k = 0, ..., N_f - 1, \end{aligned} \tag{2}$$

where $l\left(x, u, x^{\text{ref}}\right)$ denotes the stage cost, $F\left(x, x^{\text{ref}}\right)$ the terminal cost, and f the discretized dynamics coming from the ODE above.

To define the costs, we define:

- the *longitudinal* and *lateral* position errors

$$\begin{aligned} e_{\text{lon}} &= \cos\left(\varphi^{\text{ref}}\right)\left(X - X^{\text{ref}}\right) + \sin\left(\varphi^{\text{ref}}\right)\left(Y - Y^{\text{ref}}\right), \\ e_{\text{lat}} &= -\sin\left(\varphi^{\text{ref}}\right)\left(X - X^{\text{ref}}\right) + \cos\left(\varphi^{\text{ref}}\right)\left(Y - Y^{\text{ref}}\right) \end{aligned} \tag{3}$$

  that correspond to rotated versions of the *absolute* position errors $e_{X,k} = \left(X_k - X_k^{\text{ref}}\right)$ and $e_{Y,k} = \left(Y_k - Y_k^{\text{ref}}\right)$.

  Using longitudinal and lateral errors instead of absolute errors allows us to tune the associated costs independently. Furthermore, these variables also allow us to formulate track constraints as

$$e_{\text{lat,min,k}} \leq e_{\text{lat},k} \leq e_{\text{lat,max,k}},\ k = 0, ..., N_f. \tag{4}$$

  Note however we do not make use of them in the current implementation.

- the *reference* throttle as the steady state contol input for a certain reference velocity v^{ref}

$$T^{\text{ref}} = \frac{1}{C_m}\left(C_{r0} + C_{r1} v^{\text{ref}} + C_{r2} {v^{\text{ref}}}^2\right) \tag{5}$$

The stage and terminal costs then read

$$
\begin{aligned}
l\left(x, u, x^{\text{ref}}\right) &= q_{\text{lon}} e_{\text{lon}}^2 + q_{\text{lat}} e_{\text{lat}}^2 + q_{\varphi}\left(\varphi - \varphi^{\text{ref}}\right)^2 + q_v\left(v - v^{\text{ref}}\right)^2 + q_T\left(T - T^{\text{ref}}\right)^2 + q_\delta \delta^2, \\
F\left(x, x^{\text{ref}}\right) &= q_{\text{lon},f} e_{\text{lon}}^2 + q_{\text{lat},f} e_{\text{lat}}^2 + q_{\varphi,f}\left(\varphi - \varphi^{\text{ref}}\right)^2 + q_{v,f}\left(v - v^{\text{ref}}\right)^2.
\end{aligned}
\tag{6}
$$

The OCP defined in Equation 2 is solved in closed-loop in a receding horizon fashion, using state references generated by a separate motion planner which:

- computes an offline reference trajectory by fitting a cubic spline to a set of waypoints on the center line of the track,
- generates online state references by projecting the current position on the interpolated center line and generates a set of discrete points based on a constant velocity profile and a heading given by the tangent of the center line.

3. DPC formulation

The objective of DPC is to learn an explicit control policy

$$
\pi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x \times (N_f+1)} \to \mathbb{R}^{n_u \times N_f}, \left(x_0, \left\{x_k^{\text{ref}}\right\}_{k=0}^{N_f}\right) \mapsto \left\{u_k\right\}_{k=0}^{N_f-1}
\tag{7}
$$

that returns the optimal solution of Equation 2. We model this policy by a neural network (NN) π_θ with parameters θ, that we will train in an unsupervised manner.

The learning procedure is based on a dataset of