Linear Tracking MPC

Timothy Werner

***Abstract*— This paper briefly introduces the implemented Linear Reference Tracking MPC scheme and a slight generalization to state trajectory tracking and nonlinear systems in the sense of linear time varying approximations.**

I. The Tracking Problem

We consider linear discrete time systems, sampled at rate δ of the form

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{1a}$$

$$\mathbf{y}_k = C\mathbf{x}_k, \tag{1b}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ are the states, $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^p$ are the inputs and $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^q$ are the outputs of the system. The goal of tracking is to steer the output along a known reference trajectory $\mathbf{r} \in \mathcal{Y}$, i.e. to render the reference error

$$\mathbf{e}_k := \mathbf{y}_k - \mathbf{r}_k \triangleq C\mathbf{x}_k - \mathbf{r}_k \to 0,$$

as $k \to \infty$. We assume a given trajectory $\mathcal{T}$, such that

$$\mathcal{T}(t_k) : [t_0, t_f] \to \mathcal{Y}, t_k \to \mathbf{r}_k.$$

Here $t_k = k\delta$, $k \in \mathbb{N}_0$ refers to a discrete point in time. In that sense $\mathcal{T}(t_k)$ can be a continuous mapping or a set of discrete points. Note that we consider the state-space, the input-space and the output-space to be constrained. We will also consider rate constraints because all physical systems are rate-constrained. This extension also appears natural for two more reasons which will be discussed briefly in the following. To this end, we define the rate as

$$\Delta\mathbf{u}_k := \mathbf{u}_k - \mathbf{u}_{k-1}. \tag{2}$$

The constraints on either signals are assumed to be of the form of box-constraints, i.e.

$$\underline{\mathbf{x}} \le \mathbf{x}_{j+k} \quad \le \overline{\mathbf{x}} \tag{3a}$$

$$\underline{\mathbf{y}} \le C\mathbf{x}_{j+k} \le \overline{\mathbf{y}} \tag{3b}$$

$$\underline{\mathbf{u}} \le \mathbf{u}_{j+k} \quad \le \overline{\mathbf{u}} \tag{3c}$$

$$\underline{\Delta\mathbf{u}} \le \Delta\mathbf{u}_{j+k} \le \overline{\Delta\mathbf{u}}. \tag{3d}$$

II. The Linear Reference Tracking MPC

It appears natural to consider the tracking error as defined above as the objective to be minimized. We can thus formulate the cost

$$\mathcal{J}_j := \sum_{k=0}^{N-1} \|\mathbf{e}_{j+k}\|_Q^2 + \|\Delta\mathbf{u}_{j+k}\|_{R_\Delta}^2 \tag{4}$$

where N is the discrete MPC horizon over which the optimal input sequence is to be found and we denoted $\|\mathbf{e}\|_Q^2 := \mathbf{e}^\top Q\mathbf{e}$. Note that a rate-regularization was introduced to the cost, which ensures a smooth tracking, which is one of the two main reasons to consider input rates.

III. Implementation

A. The Cost Function

To enforce stability on the controller, we introduce a terminal constraint that drives the predicted trajectory onto the desired trajectory $\mathbf{r}$ near the end of the prediction horizon. A reliable choice that guarantees stability under a sufficiently large optimization horizon is the solution to the Riccati equations P that stems from the infinite horizon LQR problem. We further introduce a direct penalty on the inputs $\mathbf{u}$ due to a state augmentation to be introduced later. We have thus

$$\begin{aligned}\mathcal{J}_j := \sum_{k=0}^{N-1} \|\mathbf{e}_{j+k}\|_Q^2 + \|\Delta\mathbf{u}_{j+k}\|_{R_\Delta}^2 + \\ \|\mathbf{u}_{j+k}\|_R^2 + \|\mathbf{e}_{j+N}\|_P^2\end{aligned} \tag{5}$$

If we further investigate the terms, omitting the constant terms w.r.t to $\mathbf{x}$ and $\mathbf{u}$, we find

$$\begin{aligned}\mathcal{J}_j := \sum_{k=0}^{N-1} & \mathbf{x}_{j+k}^\top C^\top QC\mathbf{x}_{j+k} \\ & + \mathbf{u}_{j+k}^\top R\mathbf{u}_{j+k} \\ & + \Delta\mathbf{u}_{j+k}^\top R_\Delta \Delta\mathbf{u}_{j+k} \\ & - 2\mathbf{r}_{j+k}^\top QC\mathbf{x}_{j+k} \\ & + \mathbf{x}_{j+N}^\top C^\top PC\mathbf{x}_{j+N} \\ & - 2\mathbf{r}_{j+N}^\top PC\mathbf{x}_{j+N}\end{aligned} \tag{6}$$

B. State Augmentation

We define the new state $\tilde{\mathbf{x}}_k := \begin{bmatrix}\mathbf{x}_k^\top & \mathbf{u}_{k-1}^\top\end{bmatrix}^\top$ and combining (1a) and (2), we find the new augmented system

$$\tilde{\mathbf{x}}_{k+1} = \begin{bmatrix} A & B \\ & I \end{bmatrix} \tilde{\mathbf{x}}_k + \begin{bmatrix} B \\ I \end{bmatrix} \Delta\mathbf{u}_k, \tag{7}$$

which we will refer to as

$$\tilde{\mathbf{x}}_{k+1} = \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\Delta\mathbf{u}_k.$$

This state augmentation was mainly performed to ensure zero steady state error independent of the weights chosen for error penalty and input penalty. With a mere error and input penalty, the steady state error would be directly dependent on the weights chosen for either term. Imagine to this end a high weight on the inputs and a low weight on the reference error. The controller would not put a lot of effort in tracking the reference, since in would cost a lot of input. A high weight on tracking error and a low weight on the inputs on the other hand would lead to a bang-bang behavior which is in general not a desired control strategy. Choosing the reference error and the rates to be minimized on the other hand resolves in a well posed problem, where the weights on error and input rates shift the pareto-solution between aggressive and smooth tracking. The drawback of this method is that the dimension of the problem is raised from $(N+1)n + Np$ to $(N+1)(n+p) + Np$. Rewriting the cost function in terms of (7), we find

$$\begin{aligned}\mathcal{J}_j := \sum_{k=0}^{N-1} & \tilde{\mathbf{x}}_{j+k}^\top \tilde{Q} \tilde{\mathbf{x}}_{j+k} \\ & + \Delta \mathbf{u}_{j+k}^\top R_\Delta \Delta \mathbf{u}_{j+k} \\ & - C_Q^\top \tilde{\mathbf{x}}_{j+k} \\ & + \tilde{\mathbf{x}}_{j+N}^\top \tilde{P} \tilde{\mathbf{x}}_{j+N} \\ & - C_P^\top \tilde{\mathbf{x}}_{j+N},\end{aligned} \tag{8}$$

with

$$\tilde{Q} := \begin{bmatrix} C^\top Q C & \\ & R \end{bmatrix},$$

$$\tilde{P} := \begin{bmatrix} C^\top P C & \\ & R \end{bmatrix},$$

$$C_Q^\top := \begin{bmatrix} 2\mathbf{r}_k^\top Q C & 0 \end{bmatrix},$$

and

$$C_P^\top := \begin{bmatrix} 2\mathbf{r}_k^\top P C & 0 \end{bmatrix}$$

C. The Optimization Problem

With this defined, we can formulate the QP to be solved at time instance $t = j$.

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}_j && \text{(9a)} \\ \text{subject to} \quad & \tilde{\mathbf{x}}_0 = \begin{bmatrix} \mathbf{x}_j^\top & (\mathbf{u}_{0|j-1}^*)^\top \end{bmatrix}^\top && \text{(9b)} \\ & \tilde{\mathbf{x}}_{j+k+1} = \tilde{A}\tilde{\mathbf{x}}_{j+k} + \tilde{B}\Delta \mathbf{u}_{j+k} && \text{(9c)} \\ & \underline{l} \le M \tilde{\mathbf{x}}_{j+k} \le \bar{l} && \text{(9d)} \\ & \underline{\Delta \mathbf{u}} \le \Delta \mathbf{u}_{j+k} \le \overline{\Delta \mathbf{u}} && \text{(9e)} \end{aligned}$$

Note that we combined (3a),(3b) and (3c) to form (9d). The solution of (9) yields the optimal input sequence $\mathbf{\Omega}_j^*$ and the corresponding predicted state trajectory $\mathbf{\Xi}_j^*$ at time instance j

$$\mathbf{\Omega}_j^* := \left\{ \mathbf{u}_{0|j-1}^* \quad \mathbf{u}_{0|j}^* \quad \mathbf{u}_{1|j}^* \quad \cdots \quad \mathbf{u}_{N-1|j}^* \right\} \tag{10}$$

$$\mathbf{\Xi}_j^* := \left\{ \mathbf{x}_{0|j}^* \quad \mathbf{x}_{1|j}^* \quad \cdots \quad \mathbf{x}_{N-1|j}^* \quad \mathbf{x}_{N|j}^* \right\} \tag{11}$$

(12)

where $\{k|j\}$ indicates the value at MPC stage $k \in [0, N]$ at time instance $t = j$. The controller will then output $\mathbf{u}(\mathbf{x}_j, j) = \mathbf{u}_{0|j}^*$ and apply it to (1a). Note that (9b) depends on the first stage of the optimal input sequence of the previous time step $\mathbf{u}_{0|j-1}^*$. This ensures satisfaction of the rate constraints of the resulting closed-loop input sequence

$$\mathbf{\Omega}_j^{cl} := \left\{ \mathbf{u}_{0|0}^* \quad \mathbf{u}_{1|\delta}^* \quad \cdots \right\}. \tag{13}$$

D. Different formulation

In general one directly calls a QP solver and handles the matrices defining the objective to be minimized and the constraints the problem is subject to. More sophisticated solvers allow for a slightly more loos problem formulation and handle the matrix stacking in the background. This increases the robustness and allows for a formulation that is tailored for the specific QP solver which increases the run time effectively. In this sense, another common formulation of the objective is

$$\begin{aligned}\mathcal{J}_j := \sum_{k=0}^{N-1} & \begin{bmatrix} C\mathbf{x}_{j+k} - \mathbf{r}_{j+k} \\ \mathbf{u}_{j+k} \end{bmatrix}^\top \begin{bmatrix} Q & \\ & R \end{bmatrix} \begin{bmatrix} C\mathbf{x}_{j+k} - \mathbf{r}_{j+k} \\ \mathbf{u}_{j+k} \end{bmatrix} \\ & + \Delta \mathbf{u}_{j+k}^\top R_\Delta \Delta \mathbf{u}_{j+k} \\ & + \begin{bmatrix} C\mathbf{x}_{j+N} - \mathbf{r}_{j+N} \\ \mathbf{u}_{j+N} \end{bmatrix}^\top \begin{bmatrix} P & \\ & R \end{bmatrix} \begin{bmatrix} C\mathbf{x}_{j+N} - \mathbf{r}_{j+N} \\ \mathbf{u}_{j+N} \end{bmatrix}\end{aligned} \tag{14}$$

IV. The Linear Trajectory Tracking MPC

A. General State Trajectories

A special case of a given trajectory is a complete state trajectory

$$\mathcal{T} = \left\{ \mathbf{u}^r(t_k) \in \mathcal{U} \mid \quad s.t. \, (1\text{a}) \forall t \in [t_0, t_f], \, t = \delta k, \, k \in \mathbb{N}_0 \right\}.$$

Other than in the reference tracking case, where the goal is to track a given output trajectory as good as possible subject to constraints, the goal is to track an a priori known state evolution. This evolution is given by a reference input sequence $\mathbf{u}^r$ and a reference state sequence $\mathbf{x}^r$, where $\mathbf{x}^r$ is the solution of (1a) under $\mathbf{u}^r$. Tracking such a state trajectory can be achieved by solving the MPC under the objective

$$\mathcal{J}_j := \sum_{k=0}^{N-1} \left\| \begin{bmatrix} \mathbf{x}_{j+k} - \mathbf{x}_{j+k}^r \\ \mathbf{u}_{j+k} - \mathbf{u}_{j+k}^r \end{bmatrix} \right\|_S^2 + \left\| \mathbf{x}_{j+N} - \mathbf{x}_{j+N}^r \right\|_P^2, \tag{15}$$

where $S := \text{blkdiag}([Q, R])$.

B. Generalization to Nonlinear Systems

Given a general nonlinear system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{16a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}), \tag{16b}$$

and a state trajectory of the form

$$\mathcal{T} = \left\{ \mathbf{u}^r(t_k) \in \mathcal{U} \mid \quad s.t. \, (1\text{a}) \forall t \in [t_0, t_f], \, t = \delta k, \, k \in \mathbb{N}_0 \right\},$$

one can compute a linearization of (16) around $\mathcal{T}$ and perform an exact discretization to arrive at a linear discrete-time-varying (LTV) system of the form

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{g}_k \tag{17a}$$

$$\mathbf{y}_k = C_k \mathbf{x}_k. \tag{17b}$$

This can be used to reformulate the now time-varying OCP (9) each time step, as a linearization around the nominal reference state trajectory.