MPC: Tracking, Soft Constraints, Move-Blocking

M. Morari, F. Borrelli*, C. Jones†

Institut für Automatik
ETH Zürich

*UC Berkeley

† EPFL

Fall Semester 2015

Table of Contents

Table of Contents

Tracking problem

Consider the linear system model

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

Goal: Track given reference r such that $y_k \to r$ as $k \to \infty$.

Determine the steady state target condition x_s, u_s:

$$\begin{aligned} x_s &= Ax_s + Bu_s \\ Cx_s &= r \end{aligned} \qquad \Longleftrightarrow \qquad \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Table of Contents

Steady-state target problem

- In the presence of constraints: (x_s, u_s) has to satisfy state and input constraints.
- In case of multiple feasible u_s, compute 'cheapest' steady-state (x_s, u_s) corresponding to reference r:

$$
\begin{aligned}
\min \quad & u_s^T R_s u_s \\
\text{s.t.} \quad & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\
& x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}.
\end{aligned}
$$

- In general, we assume that the target problem is feasible
- If no solution exists: compute reachable set point that is 'closest' to r:

$$
\begin{aligned}
\min \quad & (Cx_s - r)^T Q_s (Cx_s - r) \\
\text{s.t.} \quad & x_s = Ax_s + Bu_s \\
& x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}.
\end{aligned}
$$

RHC Reference Tracking

We now use control (MPC) to bring the system to a desired steady-state condition (x_s, u_s) yielding the desired output $y_k \to r$.

The MPC is designed as follows [1]

$$\begin{array}{ll} \min\limits_{u_0,\ldots,u_{N-1}} & \|y_N - Cx_s\|_P^2 + \sum\limits_{k=0}^{N-1} \|y_k - Cx_s\|_Q^2 + \|u_k - u_s\|_R^2 \\ \text{subj. to} & \text{model} \\ & \text{constraints} \\ & x_0 = x(t). \end{array}$$

Drawback: controller will show **offset** in case of unknown model error or disturbances.

[1]Notation: $\|u - v\|_M^2 = (u - v)' M (u - v)$

Table of Contents

RHC Reference Tracking without Offset (1/6)

Discrete-time, time-invariant system (possibly nonlinear, uncertain)

$$\begin{aligned} x_m(t+1) &= g(x_m(t), u(t)) \\ y_m(t) &= h(x_m(t)) \end{aligned}$$

Objective:

- Design an RHC in order to make $y(t)$ track the reference signal $r(t)$, i.e., $(y(t) - r(t)) \to 0$ for $t \to \infty$.
- In the rest of the section we study step references and focus on zero steady-state tracking error, $y(t) \to r_\infty$ as $t \to \infty$.

Consider augmented model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + B_d d(t) \\ d(t+1) &= d(t) \\ y(t) &= Cx(t) + C_d d(t) \end{aligned}$$

with constant disturbance $d(t) \in \mathbb{R}^{n_d}$.

RHC Reference Tracking without Offset (2/6)

State observer for augmented model

$$\begin{bmatrix} \hat{x}(t+1) \\ \hat{d}(t+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ \hat{d}(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (-y_m(t) + C\hat{x}(t) + C_d\hat{d}(t))$$

Lemma

Suppose the observer is stable and the number of outputs p equals the dimension of the constant disturbance n_d. The observer steady state satisfies:

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d\hat{d}_\infty \\ y_{m,\infty} - C_d\hat{d}_\infty \end{bmatrix}.$$

where $y_{m,\infty}$ and u_∞ are the steady state measured outputs and inputs.

$\Rightarrow$ The observer output $C\hat{x}_\infty + C_d\hat{d}_\infty$ tracks the measured output $y_{m,\infty}$ without offset.

RHC Reference Tracking without Offset (3/6)

For offset-free tracking at steady state we want $y_{m,\infty} = r_\infty$. The observer condition

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ y_{m,\infty} - C_d \hat{d}_\infty \end{bmatrix}$$

suggests that at steady state the MPC should satisfy

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_{target,\infty} \\ u_{target,\infty} \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ r_\infty - C_d \hat{d}_\infty \end{bmatrix}$$

RHC Reference Tracking without Offset (4/6)

Formulate the RHC problem

$$
\begin{array}{lll}
\min\limits_{U_0} & \|x_N - \bar{x}_t\|_P^2 + \sum\limits_{k=0}^{N-1} \|x_k - \bar{x}_t\|_Q^2 + \|u_k - \bar{u}_t\|_R^2 & \\
\text{subj. to} & x_{k+1} = Ax_k + Bu_k + B_d d_k, & k = 0, \dots, N \\
& x_k \in \mathcal{X},\ u_k \in \mathcal{U}, & k = 0, \dots, N-1 \\
& x_N \in \mathcal{X}_f & \\
& d_{k+1} = d_k, & k = 0, \dots, N \\
& x_0 = \hat{x}(t) & \\
& d_0 = \hat{d}(t), &
\end{array}
$$

with the targets $\bar{u}_t$ and $\bar{x}_t$ given by

$$
\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_t \\ \bar{u}_t \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}(t) \\ r(t) - C_d \hat{d}(t) \end{bmatrix}
$$

RHC Reference Tracking without Offset (5/6)

Denote by $c_0(\hat{x}(t), \hat{d}(t), r(t)) = u_0^*(\hat{x}(t), \hat{d}(t), r(t))$ the control law when the estimated state and disturbance are $\hat{x}(t)$ and $\hat{d}(t)$, respectively.

Theorem

Consider the case where the number of constant disturbances equals the number of (tracked) outputs $n_d = p = r$. Assume the RHC is recursively feasible and unconstrained for $t \geq j$ with $j \in \mathbb{N}^+$ and the closed-loop system

$$\begin{aligned} x(t+1) &= f(x(t), c_0(\hat{x}(t), \hat{d}(t), r(t))) \\ \hat{x}(t+1) &= (A + L_x C)\hat{x}(t) + (B_d + L_x C_d)\hat{d}(t) \\ &\quad + B c_0(\hat{x}(t), \hat{d}(t), r(t)) - L_x y_m(t) \\ \hat{d}(t+1) &= L_d C \hat{x}(t) + (I + L_d C_d)\hat{d}(t) - L_d y_m(t) \end{aligned}$$

converges to $\hat{x}_\infty,\ \hat{d}_\infty,\ y_{m,\infty}$, i.e., $\hat{x}(t) \to \hat{x}_\infty,\ \hat{d}(t) \to \hat{d}_\infty,\ y_m(t) \to y_{m,\infty}$ as $t \to \infty$.

Then $y_m(t) \to r_\infty$ as $t \to \infty$.

RHC Reference Tracking without Offset (6/6)

Question: How do we choose the matrices B_d and C_d in the augmented model?

Lemma

The augmented system, with the number of outputs p equal to the dimension of the constant disturbance n_d, and $C_d = I$ is observable if and only if (C, A) is observable and

$$\det \begin{bmatrix} A - I & B_d \\ C & I \end{bmatrix} = \det(A - I - B_d C) \neq 0.$$

Remark: If the plant has no integrators, then $\det(A - I) \neq 0$ and we can choose $B_d = 0$. If the plant has integrators then B_d has to be chosen specifically to make $\det(A - I - B_d C) \neq 0$.

Table of Contents

Table of Contents

Soft Constraints: Motivation

- Input constraints are dictated by physical constraints on the actuators and are usually “hard”
- State/output constraints arise from practical restrictions on the allowed operating range and are **rarely hard**
- Hard state/output constraints always lead to *complications in the controller implementation*
  - Feasible operating regime is constrained even for stable systems
  - Controller patches must be implemented to generate reasonable control action when measured/estimated states move outside feasible range because of disturbances or noise
- In industrial implementations, typically, state constraints are **softened**

Table of Contents

Mathematical Formulation

- Original problem:

$$\begin{array}{ll} \min\limits_{z} & f(z) \\ \text{subj. to} & g(z) \leq 0 \end{array}$$

  Assume for now $g(z)$ is scalar valued.

- "Softened" problem:

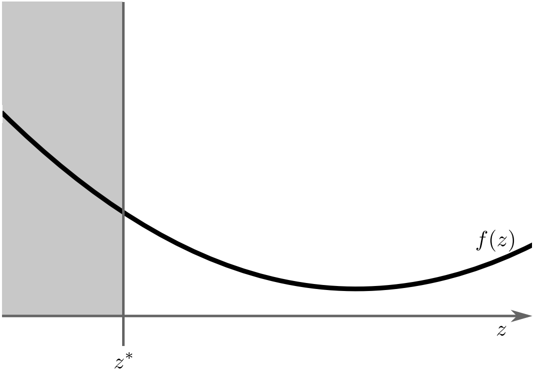
$$\begin{array}{ll} \min\limits_{z,\epsilon} & f(z) + l(\epsilon) \\ \text{subj. to} & g(z) \leq \epsilon \\ & \epsilon \geq 0 \end{array}$$

Requirement on $l(\epsilon)$

If the original problem has a feasible solution z^*, then the softened problem should have the same solution z^*, and $\epsilon = 0$.

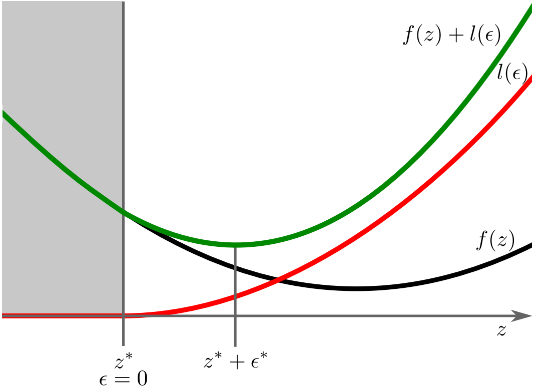
Note: $l(\epsilon) = v \cdot \epsilon^2$ does not meet this requirement for any $v > 0$ as demonstrated next.

Quadratic Penalty



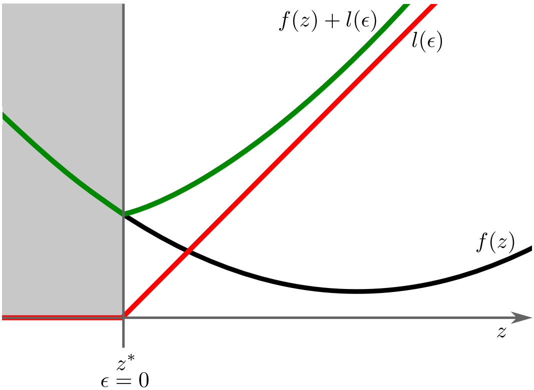
- Constraint function $g(z) \triangleq z - z^* \leq 0$ induces feasible region (grey) $\Longrightarrow$ minimizer of the original problem is z^*
- **Quadratic penalty** $l(\epsilon) = v \cdot \epsilon^2$ for $\epsilon \geq 0$ $\Longrightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^* + \epsilon^*, \epsilon^*)$ instead of $(z^*, 0)$

Quadratic Penalty



- Constraint function $g(z) \triangleq z - z^* \leq 0$ induces feasible region (grey) $\Longrightarrow$ minimizer of the original problem is z^*
- **Quadratic penalty** $l(\epsilon) = v \cdot \epsilon^2$ for $\epsilon \geq 0$ $\Longrightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^* + \epsilon^*, \epsilon^*)$ instead of $(z^*, 0)$

Linear Penalty



- Constraint function $g(z) \triangleq z - z^* \leq 0$ induces feasible region (grey) $\Longrightarrow$ minimizer of the original problem is z^*
- **Linear penalty** $l(\epsilon) = u \cdot \epsilon$ for $\epsilon \geq 0$ with u chosen large enough so that $u + \lim_{z \to z^*} f'(z) > 0$
  $\Longrightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^*, 0)$

Comments

- **Disadvantage:** $l(\epsilon) = u \cdot \epsilon$ renders the cost non-smooth.
- Therefore in practice, to get a smooth penalty, we use

$$l(\epsilon) = u \cdot \epsilon + v \cdot \epsilon^2$$

with $u > u^*$ and $v > 0$.

- Extension to multiple constraints $g_j(z) \leq 0,\ j = 1, \ldots, r$:

$$l(\epsilon) = \sum_{j=1}^{r} u_j \cdot \epsilon_j + v_j \cdot \epsilon_j^2 \tag{1}$$

where $u_j > u_j^*$ and $v_j > 0$ can be used to weight violations (if necessary) differently.

Table of Contents

Generalizing the Problem

Modify problem as:

$$
\begin{array}{lll}
\min\limits_{U_0} & \left\{ \|x_{N_y}\|_P^2 + \sum\limits_{k=0}^{N_y-1} \left[\|x_k\|_Q^2 + \|u_k\|_R^2 \right] \right\} & \\
\text{subj. to} & y_{\min}(k) \le y_k \le y_{\max}(k), & k = 1, \ldots, N_c \\
& u_{\min} \le u_k \le u_{\max}, & k = 0, 1, \ldots, N_u \\
& x_0 = x(t) & \\
& x_{k+1} = Ax_k + Bu_k, & k \ge 0 \\
& y_k = Cx_k, & k \ge 0 \\
& u_k = Kx_k, & N_u \le k < N_y
\end{array}
$$

with $N_u \le N_y$ and $N_c \le N_y - 1$.

- Many applications require time-varying constraints, e.g. $y_{\min}(k)$, $y_{\max}(k)$
- Complexity can be reduced by introducing separate horizons N_u, N_c, N_y
- But, all theoretical feasibility and stability guarantees are lost!

Generalizing the Problem

- More effective way to reduce the computational effort: **Move-blocking**
- Manipulated variables are fixed over time intervals in the future
  $\Rightarrow$ degrees of freedom in optimization problem are reduced
- By choosing the blocking strategies carefully RHC stability results remain applicable