Python-Control Testing Issue 929

JMK593

September 13, 2023

1 LTI Dynamics

Consider a deterministic, discrete-time, LTI system with state $x_k \in \mathbb{R}^n$ and dynamics

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

where $u_k \in \mathbb{R}^m$ is the control input at time k, and $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. Assume that the pair (A, B) is controllable.

From a given initial state value x_0 then the state at time $k+1$ can be written as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ &= A(Ax_{k-1} + Bu_{k-1}) + Bu_k \\ &= A^2 x_{k-1} + ABu_{k-1} + Bu_k \\ &\dots \\ &= A^{k+1}x_0 + A^k Bu_0 + \cdots + ABu_{k-1} + Bu_k \\ &= A^{k+1}x_0 + \sum_{j=0}^{k} A^j Bu_{k-j}. \end{aligned}$$

Alternatively the state at some time $k+\ell$ from an initial state at time k can be written as

$$x_{k+\ell|k} = A^\ell x_k + \sum j = 0^{\ell-1} A^{\ell-j} Bu_{k+j}. \tag{1}$$

Let us define the finite horizon state prediction of length N from time k and the accompanying control sequence as

$$\vec{x}_k = \begin{bmatrix} x_{k+1|k} \\ \vdots \\ x_{k+N|k} \end{bmatrix} \quad \text{and} \quad \vec{u}_k = \begin{bmatrix} u_{k|k} \\ \vdots \\ u_{k+N-1|k} \end{bmatrix},$$

then using (1) we can write the prediction conveniently as

$$\vec{x}_k = \Lambda x_k + \Phi \vec{u}_k \tag{2}$$

where

$$\Lambda = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \quad \text{and} \quad \Phi = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B. \end{bmatrix}$$

2 Cost Function

Let us consider a quadratic stage cost of

$$L(x,u) = (x - x^\star)^T Q (x - x^\star) + u^T R u$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are the state and control action penalties, and $x^\star$ is the state reference (or desired state). Let us consider a terminal cost of

$$L_F(x) = x^T Q_F x$$

where $Q_F \in \mathbb{R}^{n \times n}$ is the terminal state penalty.

Consider the finite horizon cost function

$$\begin{aligned} J(x_k, \vec{u}_k) &= \sum_{\ell=0}^{N-1} (L(x_{k+\ell|k}, u_{k+\ell|k})) + L_F(x_{k+N|k}) \\ &= \sum_{\ell=0}^{N-1} \Big((x_{k+\ell|k} - x^\star)^T Q (x_{k+\ell|k} - x^\star) + u_{k+\ell|k}^T R u_{k+\ell|k} \Big) \\ &\qquad + (x_{k+N|k} - x^\star)^T Q_F (x_{k+N|k} - x^\star) \\ &= (\vec{x}_k - \vec{x}^\star)^\mathsf{T} \bar{Q} (\vec{x}_k - \vec{x}^\star) + \vec{u}_k^\mathsf{T} \bar{R} \vec{u}_k + (x_k - x^\star)^\mathsf{T} Q (x_k - x^\star) \end{aligned} \tag{3}$$

where we define $\bar{Q}$ and $\bar{R}$ as block diagonal matrices

$$\bar{Q} = \begin{bmatrix} Q & 0 & \dots & 0 & 0 \\ 0 & Q & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & Q & 0 \\ 0 & 0 & \dots & 0 & Q_F \end{bmatrix} \quad \text{and} \quad \bar{R} = \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R \end{bmatrix}.$$

Utilising (2) then

$$\begin{aligned} J(x_k, \vec{u}_k) &= (\vec{x}_k - \vec{x}^\star)^\mathsf{T} \bar{Q} (\vec{x}_k - \vec{x}^\star) + \vec{u}_k^\mathsf{T} \bar{R} \vec{u}_k + (x_k - x^\star)^\mathsf{T} Q (x_k - x^\star) \\ &= (\Lambda x_k + \Phi \vec{u}_k - \vec{x}^\star)^\mathsf{T} \bar{Q} (\Lambda x_k + \Phi \vec{u}_k - \vec{x}^\star) + \vec{u}_k^\mathsf{T} \bar{R} \vec{u}_k + (x_k - x^\star)^\mathsf{T} Q (x_k - x^\star) \\ &= (\Lambda x_k - \vec{x}^\star)^\mathsf{T} \bar{Q} (\Lambda x_k - \vec{x}^\star) + 2\vec{u}_k^\mathsf{T} \Phi^\mathsf{T} \bar{Q} (\Lambda x_k - \vec{x}^\star) + \vec{u}_k^\mathsf{T} \Phi^\mathsf{T} \bar{Q} \Phi \vec{u}_k \\ &\qquad + \vec{u}_k^\mathsf{T} \bar{R} \vec{u}_k + (x_k - x^\star)^\mathsf{T} Q (x_k - x^\star) \\ &= (\Lambda x_k - \vec{x}^\star)^\mathsf{T} \bar{Q} (\Lambda x_k - \vec{x}^\star) + (x_k - x^\star)^\mathsf{T} Q (x_k - x^\star) \\ &\qquad + 2\vec{u}_k^\mathsf{T} \Phi^\mathsf{T} \bar{Q} (\Lambda x_k - \vec{x}^\star) + \vec{u}_k^\mathsf{T} (\Phi^\mathsf{T} \bar{Q} \Phi + \bar{R}) \vec{u}_k. \end{aligned}$$

In the context of minimizing $J(x_k, \vec{u}_k)$ with respect to the control sequence $\vec{u}_k$, we note that the first line

$$(\Lambda x_k - \vec{x}^\star)^\intercal \bar{Q}(\Lambda x_k - \vec{x}^\star) + (x_k - x^\star)^\intercal Q(x_k - x^\star)$$

is constant with respect to $\vec{u}_k$ and could be discarded in the minimization process.

We can write a simplified cost function of

$$J(x_k, \vec{u}_k) = 2\vec{u}_k^\intercal \Phi^\intercal \bar{Q}(\Lambda x_k - \vec{x}^\star) + \vec{u}_k^\intercal(\Phi^\intercal \bar{Q}\Phi + \bar{R})\vec{u}_k$$

which has a first and second order component. Writing in the form of MATLAB's `quadprog` or `mpcActiveSetSolver` we find

$$J(\vec{u}_k) = \frac{1}{2}\vec{u}_k^\intercal H \vec{u}_k + f^\intercal \vec{u}_k \tag{4}$$

where $H = 2(\Phi^\intercal \bar{Q}\Phi + \bar{R})$ and $f = 2\Phi^\intercal \bar{Q}(\Lambda x_k - \vec{x}^\star)$. Observe that H can be pre-computed but f must be computed each time step as a function of the current state x_k.

3 Constraints

Consider the following constraints

$$x_{\min} \leq x_k \leq x_{\max}$$

which apply to the state at all times. We let $\vec{x}_{\min}$ and $\vec{x}_{\max}$ denote the vector of constraints that match the prediction $\vec{x}_k$.

Consider the lower bound constraints and apply the prediction dynamics (2)

$$\begin{aligned}
\vec{x}_{\min} &\leq \vec{x}_k \\
\vec{x}_{\min} &\leq \Lambda x_k + \Phi\vec{u}_k \\
-\Phi\vec{u}_k &\leq \Lambda x_k - \vec{x}_{\min}.
\end{aligned}$$

Similar can be done for the upper bound to find

$$\begin{aligned}
\vec{x}_k &\leq \vec{x}_{\max} \\
\Lambda x_k + \Phi\vec{u}_k &\leq \vec{x}_{\max} \\
\Phi\vec{u}_k &\leq \vec{x}_{\max} - \Lambda x_k.
\end{aligned}$$

Consider the following control action constraints

$$u_{\min} \leq u_k \leq u_{\max}$$

which apply to the control action at all times. Let $\vec{u}_{\min}$ and $\vec{u}_{\max}$ denote the vector of constraints that match the prediction $\vec{u}_k$.

The constraints can be conveniently written as

$$
\begin{aligned}
-I_N \vec{u}_k &\le -\vec{u}_{\min} \\
I_N \vec{u}_k &\le \vec{u}_{\max}
\end{aligned}
$$

where I_N is the identity of size N.

Let us combine the four sets of constraints together to give linear inequality constraints

$$
\begin{aligned}
\begin{bmatrix} -\Phi \\ \Phi \\ -I_N \\ I_N \end{bmatrix} \vec{u}_k &\le \begin{bmatrix} \Lambda x_k - \vec{x}_{\min} \\ \vec{x}_{\max} - \Lambda x_k \\ -\vec{u}_{\min} \\ \vec{u}_{\max} \end{bmatrix} \\
A_{in} \vec{u}_k &\le B_{in}.
\end{aligned} \tag{5}
$$