NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Introduction to MPC

Cyber-Physical Control Systems

B. Guerreiro

2021

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Outline

From LQR to MPC

Tracking and integral action

Incremental and augmented models

Constrained Optimal Control

Constr. MPC

Outline

Outline

Outline

Outline

Outline

From LQR to MPC

Tracking and integral action

Incremental and augmented models

Constrained Optimal Control

Constr. MPC

Problem definition

Problem (LQR optimal control problem)

Find the optimal control sequence u_k^* for $k \in \{0, \ldots, N-1\}$ that drives the system along a state trajectory x_k^* for $k \in \{0, \ldots, N\}$ according to the linear system dynamics, such that the specified performance index J_0 is minimized. That is,

$$
\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \\
s.t. \quad & x_{k+1} = A x_k + B u_k \\
& x_0 = z_0
\end{aligned}
$$

The closed-form LQR controller

- Replacing the costate expression in the controller yields

$$u_k = -(B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A x_k$$

- Defining the (Kalman) gain:

$$K_k = (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A$$

- The optimal LQR control law is simply

$$u_k^* = -K_k x_k$$

- While the closed-loop system and Riccati equation become

$$x_{k+1} = (A - BK_k) x_k$$

$$P_k = A^T P_{k+1} (A - BK_k) + Q$$

$$P_k = A^T (P_{k+1} - P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1}) A + Q$$

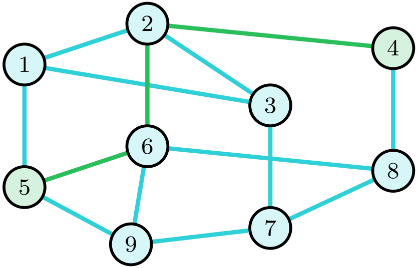
Dynamic programming approach

- Principle of optimality: Any tail of an optimal trajectory is optimal too.
- General cost-to-go from time i to horizon N:

$$J_i = J_{i \to N} = p(x_N) + \sum_{k=i}^{N-1} q(x_k, u_k)$$

- Optimal cost can be stated as

$$J_i^* = \min_{u_i} (q(x_i, u_i) + J_{i+1}^*)$$



- A recursive solution for the optimal control problem can be found.

Cost-to-go for LQR

- LQR optimal cost-to-go at time i:

$$J_i^*(x_i) := \min_{u_0,\ldots,u_{N-1}} \quad x_N^T P x_N + \sum_{k=i}^{N-1} x_k^T Q x_k + u_k^T R u_k$$
$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k$$

- Optimal cost at time $N-1$, with $P_N = P$:

$$J_{N-1}^*(x_{N-1}) := \min_{u_{N-1}} \quad x_N^T P x_N + x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1}$$
$$\text{s.t.} \quad x_N = A x_{N-1} + B u_{N-1}$$
$$J_{N-1}^*(x_{N-1}) = \min_{u_{N-1}} \quad x_{N-1}^T (A^T P_N A + Q) x_{N-1} + u_{N-1}^T (B^T P_N B + R) u_{N-1} + 2 x_{N-1}^T A^T P_N B u_{N-1}$$

First step of recursive solution of LQR

- 1st order optimality condition:

$$2(B^T P_N B + R)u_{N-1} + 2B^T P_N A x_{N-1} = 0$$

- Resulting control policy:

$$u_{N-1} = -K_{N-1}x_{N-1} \qquad K_{N-1} = (B^T P_N B + R)^{-1} B^T P_N A$$

- Results in the optimal cost-to-go:

$$J^*_{N-1}(x_{N-1}) = x^T_{N-1} P_{N-1} x_{N-1}$$

where

$$P_{N-1} = A^T (P_N - P_N B(B^T P_N B + R)^{-1} B^T P_N) A + Q$$

Recursive solution of LQR

- Optimal cost-to-go at time $N-2$:

$$
\begin{aligned}
J^*_{N-2}(x_{N-2}) := \min_{u_{N-2}} \quad & x^T_{N-2} Q x_{N-2} + u^T_{N-2} R u_{N-2} + J^*_{N-1}(x_{N-1}) \\
\text{s.t.} \quad & x_{N-1} = A x_{N-2} + B u_{N-2}
\end{aligned}
$$

- Similar solution to the previous case, as it relies on the previous optimal cost.
- In general we recover the solution we already knew:

$$
\begin{aligned}
u^*_k &= -K_k x_k \\
K_k &= (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A \\
P_k &= A^T (P_{k+1} - P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1}) A + Q \\
J^*_k(x_k) &= x^T_k P_k x_k
\end{aligned}
$$

Batch solution of an LTI system

- Recall the solution for a discrete-time LTI system:

$$x(k) = A^k x(0) + \sum_{m=0}^{k-1} A^{k-1-m} B u(m)$$

- Stacking the solution equations up to time N we get

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix}$$

- Which can be rewritten as $X = Fx(0) + GU$.

Batch cost functional

- Using the same batch notation, the LQR cost with horizon N becomes

$$J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k = X^T \bar{Q} X + U^T \bar{R} U$$

where $\bar{Q} = \mathrm{diag}\,(Q, Q, \ldots, Q, P)$ and $\bar{R} = \mathrm{diag}\,(R, R, \ldots, R)$.

- Replacing the LTI solution into this cost functional results in

$$\begin{aligned} J_0 &= (Fx(0) + GU)^T \bar{Q} (Fx(0) + GU) + U^T \bar{R} U \\ &= U^T (G^T \bar{Q} G + \bar{R}) U + 2U^T G^T \bar{Q} F x(0) + x(0)^T F^T \bar{Q} F x(0) \end{aligned}$$

- Considering $\tilde{R} = G^T \bar{Q} G + \bar{R}$, $\tilde{Q} = F^T \bar{Q} F$, and $\tilde{S} = G^T \bar{Q} F$, yields

$$J_0 = U^T \tilde{R} U + 2U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0)$$

noting that assuming $R > 0$ implies that $\tilde{R} > 0$.

Batch LQR problem and solution

- The LQR optimal control problem can then be formulated as an unconstrained minimization problem with a positive definite quadratic cost function:

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0)$$

- First order condition of optimality yields

$$\nabla_U J_0 = 2\tilde{R}U + 2\tilde{S}x(0) = 0$$

- Defining $K = \tilde{R}^{-1}\tilde{S}$, the optimal cost, control, and state sequences are

$$U^* = -Kx(0)$$

$$X^* = (F - GK)x(0)$$

$$J_0^* = x(0)^T[\tilde{Q} - \tilde{S}^T \tilde{R}^{-1} \tilde{S}]x(0)$$

Model-based Predictive Control (MPC)

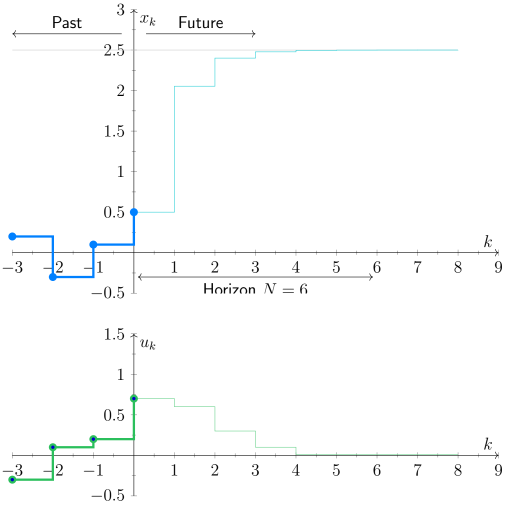
Ingredients:

- Cost function
- Model/input/state constraint
- Receding horizon

Solve optimization problem at each instant

$$\min_{u_i,\ldots,u_{i+N-1}} \quad J_i(x_i) = p(x_N) + \sum_{k=i}^{i+N-1} q(x_k, u_k)$$

$$\begin{aligned} s.t. \quad & x_{k+1} = f(x_k, u_k) \ \forall_{k=i,\ldots,i+N-1} \\ & f_{eq}(x_k, u_k) = 0 \\ & f_{in}(x_k, u_k) \leq 0 \end{aligned}$$



Model-based Predictive Control (MPC)

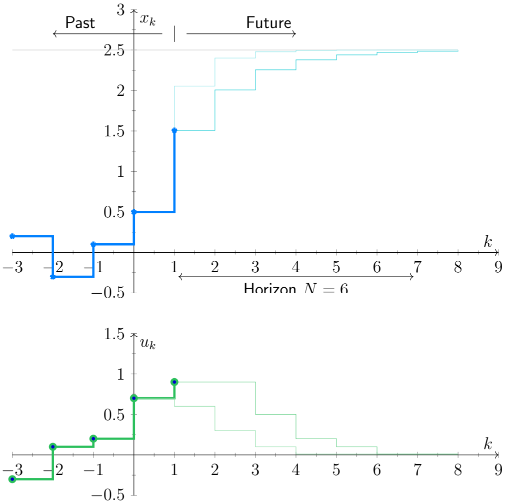
Ingredients:

- Cost function
- Model/input/state constraint
- Receding horizon

Solve optimization problem at each instant

$$\min_{u_i,\dots,u_{i+N-1}} \quad J_i(x_i) = p(x_N) + \sum_{k=i}^{i+N-1} q(x_k, u_k)$$

$$\begin{aligned} s.t. \quad & x_{k+1} = f(x_k, u_k) \ \forall_{k=i,\dots,i+N-1} \\ & f_{eq}(x_k, u_k) = 0 \\ & f_{in}(x_k, u_k) \leq 0 \end{aligned}$$



Model-based Predictive Control (MPC)

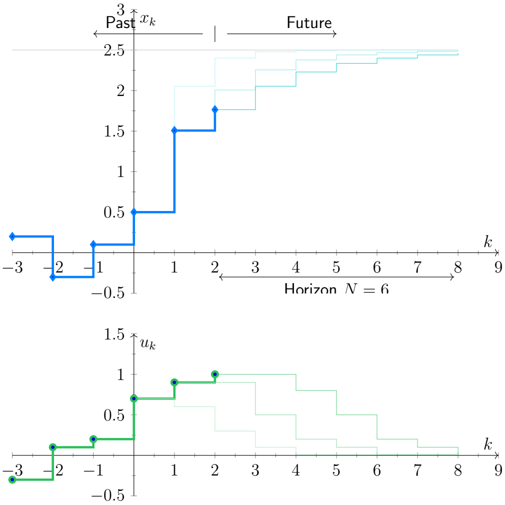
Ingredients:

- Cost function
- Model/input/state constraint
- Receding horizon

Solve optimization problem at each instant

$$\min_{u_i,\dots,u_{i+N-1}} \quad J_i(x_i) = p(x_N) + \sum_{k=i}^{i+N-1} q(x_k, u_k)$$

$$\begin{aligned} s.t. \quad & x_{k+1} = f(x_k, u_k)\ \forall_{k=i,\dots,i+N-1} \\ & f_{eq}(x_k, u_k) = 0 \\ & f_{in}(x_k, u_k) \leq 0 \end{aligned}$$



Model predictive control for LTIs

- Unconstrained predictive control problem:

$$\min_{U} \quad J_0(x_0, U) = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \; \forall_{k=0,\dots,N-1} s.t. \qquad x_0 = x_{t_k}$$

- Equivalent batch formulation:

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2 U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0)$$

- Optimal control sequence: $U^* = \{u_0^*, u_1^*, \dots, u_{N-1}^*\}$
- Receding horizon control policy:

$$u(t_k) = u_0^* = -\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} K x(t_k) = -K_0 x(t_k)$$

Receding horizon control algorithm

1. Consider the current state $x(t_k)$ as the initial condition x_0;
2. Obtain the optimal control sequence U^* solving the optimal control problem;
3. If problem unfeasible, terminate algorithm;
4. Apply the first control action from U^* to the system,
   $u(t_k) = u_0^* = -K_0 x(t_k)$;
5. Repeat from 1 in the next time instant, t_{k+1}.

MPC Example 2.1

Problem (MPC Example 2.1 - double integrator)

Consider the time-invariant discrete linear system described by

$$x(k+1) = Ax(k) + Bu(k) \qquad A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Compute the model-based predictive controller considering the quadratic cost

$$J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k)$$

with $N = 3$, $P = Q = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 10$ *and initial condition* $x(0) = \begin{bmatrix} -4.5 \\ 2 \end{bmatrix}$.

MPC Example 2.1

MPC Example 2.1 - double integrator (part 1)

Using the batch approach, the first step is to obtain the equivalent unconstrained quadratic optimization problem

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0)$$

which provides at each time step the optimal control sequence

$$U^* = -\tilde{R}^{-1} \tilde{S} x(0) = \begin{bmatrix} u^*(0)^T & u^*(1)^T & u^*(2)^T \end{bmatrix}^T$$

Considering that $X = Fx(0) + GU$ and noting that $\bar{Q} = \operatorname{diag}(Q, Q, Q, P) = I_8$ and $\bar{R} = \operatorname{diag}(R, R, R) = 10 I_3$, the final matrices are $\tilde{R} = G^T \bar{Q} G + \bar{R} = G^T G + 10 I_3$, $\tilde{Q} = F^T \bar{Q} F = F^T F$, and $\tilde{S} = G^T \bar{Q} F = G^T F$.

MPC Example 2.1

MPC Example 2.1 - double integrator (part 2)

The state batched matrices are given by

$$F = \begin{bmatrix} I \\ A \\ A^2 \\ A^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 2 \\ 0 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 & 0 \\ B & 0 & 0 \\ AB & B & 0 \\ A^2B & AB & B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

MPC Example 2.1

MPC Example 2.1 - double integrator (part 3)

The quadratic cost matrices are

$$\tilde{Q} = F^T \bar{Q} F = F^T F = \begin{bmatrix} 4 & 6 \\ 6 & 18 \end{bmatrix}$$

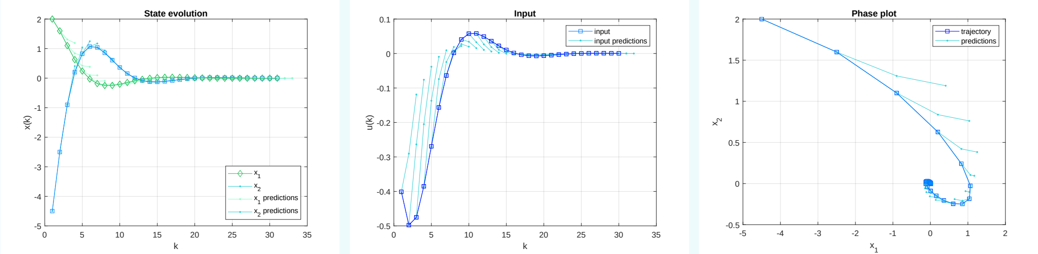
$$\tilde{R} = G^T \bar{Q} G + \bar{R} = G^T G + \bar{R} = \begin{bmatrix} 18 & 4 & 1 \\ 4 & 13 & 1 \\ 1 & 1 & 11 \end{bmatrix}$$

$$\tilde{S} = G^T \bar{Q} F = G^T F = \begin{bmatrix} 3 & 11 \\ 1 & 5 \\ 0 & 1 \end{bmatrix}$$

MPC Example 2.1

MPC Example 2.1 - double integrator (part 3)

Simulation results for 30 sampling intervals:



Outline

Linear quadratic tracking (LQT)

Problem (LQT optimal control problem)

Find the optimal control sequence u_k^* for $k \in \{0, \ldots, N-1\}$ that drives the system along a state trajectory x_k^* for $k \in \{0, \ldots, N\}$ according to the linear system dynamics, such that the specified performance index J_0 is minimized relative to a reference output signal $\bar{y}_k$. That is,

$$\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + u_k^T R u_k \\
s.t. \quad & x_{k+1} = A x_k + B u_k \\
& y_k = C x_k \\
& x_0 = z_0
\end{aligned}$$

Batch output of an LTI system

- Recall the output solution for a discrete-time LTI system:

$$y(k) = CA^k x(0) + \sum_{m=0}^{k-1} CA^{k-1-m} B u(m)$$

- Stacking the output equations up to time N we get

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix} x(0) + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}$$

- Defining $H = \operatorname{diag}(C, C, \ldots, C)$, with $Y = HX$, $\bar{F} = HF$, and $\bar{G} = HG$:

$$Y = \bar{F}x(0) + \bar{G}U$$

Batch cost functional

- Using the same batch notation, the LQT cost with horizon N is

$$J_0 = (Y - \bar{Y})^T \bar{Q} (Y - \bar{Y})^T + U^T \bar{R} U$$

where $\bar{Q} = \mathrm{diag}\,(Q, Q, \ldots, Q, P)$, $\bar{R} = \mathrm{diag}\,(R, R, \ldots, R)$, and the batched reference output is $\bar{Y} = \begin{bmatrix} \bar{y}_0^T & \cdots & \bar{y}_N^T \end{bmatrix}^T$.

- Replacing the LTI solution into this cost functional results in

$$\begin{aligned} J_0 &= (\bar{F}x(0) + \bar{G}U - \bar{Y})^T \bar{Q} (\bar{F}x(0) + \bar{G}U - \bar{Y}) + U^T \bar{R} U \\ &= U^T (\bar{G}^T \bar{Q} \bar{G} + \bar{R}) U + 2U^T \bar{G}^T \bar{Q} (\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q} (\bar{F}x(0) - \bar{Y}) \end{aligned}$$

- Considering $\tilde{R} = \bar{G}^T \bar{Q} \bar{G} + \bar{R}$ and $\tilde{S} = \bar{G}^T \bar{Q}$ we can simplify the cost as

$$J_0 = U^T \tilde{R} U + 2U^T \tilde{S} (\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q} (\bar{F}x(0) - \bar{Y})$$

noting that assuming $R > 0$ implies that $\tilde{R} > 0$.

Batch LQT problem and solution

- The LQT optimal control problem is formulated as an unconstrained minimization quadratic problem:

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y})$$

- First order condition of optimality yields

$$\nabla_U J_0 = 2\tilde{R}U + 2\tilde{S}(\bar{F}x(0) - \bar{Y}) = 0$$

- Defining the gains $K_y = \tilde{R}^{-1}\tilde{S}$ and $K = K_y \bar{F}$, the optimal control and state sequences are

$$U^* = -Kx(0) + K_y \bar{Y} = \{u_0^*, \ldots, u_{N-1}^*\} \quad , \quad X^* = (F - GK)x(0) + GK_y \bar{Y}$$

- Receding horizon control policy (assuming $\bar{y}$ constant):

$$u(t_k) = u_0^* = -\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} (Kx(t_k) + K_y \bar{Y}) = -K_0 x(t_k) + K_{y0} \bar{Y}$$

MPC Example 2.2

Problem (MPC Example 2.2 - tracking double integrator)

Consider the time-invariant discrete linear system described in Example 2.1. Compute the model-based predictive controller with quadratic tracking cost

$$J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + u_k^T R u_k$$

with $N = 3$, $P = Q = 1$, $R = 10$ *and initial condition* $x(0) = \begin{bmatrix} -4.5 \\ 2 \end{bmatrix}$, *and a reference signal* $\bar{y}(k) = \varepsilon(k - 30)$, *that is,* $\bar{y}(k) = 0$ *for all* $k < 30$ *and* $\bar{y}(k) = 1$ *for all* $k \geq 30$.

MPC Example 2.2

MPC Example 2.2 - tracking double integrator (part 1)

Using the batch approach, the first step is to obtain the equivalent unconstrained quadratic optimization problem

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y})$$

which provides at each time step the optimal control sequence

$$U^* = -(Kx(0) - K_y \bar{Y}) = \begin{bmatrix} u^*(0)^T & u^*(1)^T & u^*(2)^T \end{bmatrix}^T$$

Considering that $Y = \bar{F}x(0) + \bar{G}U$ and noting that $\bar{Q} = \operatorname{diag}(Q, Q, Q, P) = I_4$ and $\bar{R} = \operatorname{diag}(R, R, R) = 10 I_3$, the final matrices are $\tilde{R} = \bar{G}^T \bar{Q} \bar{G} + \bar{R}$, $\tilde{S} = \bar{G}^T \bar{Q}$, whereas $K_y = \tilde{R}^{-1}\tilde{S}$ and $K = K_y \bar{F}$.

MPC Example 2.2

MPC Example 2.2 - tracking double integrator (part 2)

The state batched matrices are given by

$$\bar{F} = HF = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

$$\bar{G} = HG = \begin{bmatrix} 0 & 0 & 0 \\ CB & 0 & 0 \\ CAB & CB & 0 \\ CA^2B & CAB & CB \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 2 & 1 & 0 \end{bmatrix}$$

MPC Example 2.2

MPC Example 2.2 - tracking double integrator (part 3)

The quadratic cost matrices are

$$\tilde{R} = \bar{G}^T \bar{Q} \bar{G} + \bar{R} = \bar{G}^T \bar{G} + \bar{R} = \begin{bmatrix} 15 & 2 & 0 \\ 2 & 11 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \tilde{S} = \bar{G}^T \bar{Q} = \bar{G}^T = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

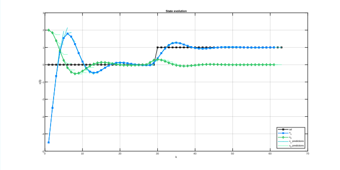
while the control gains are

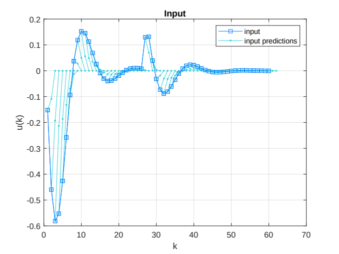
$$K_y = \bar{R}^{-1} \tilde{S} = \begin{bmatrix} 0 & 0 & 0.0683 & 0.1242 \\ 0 & 0 & -0.0124 & 0.0683 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad K = \bar{R}^{-1} \tilde{S} \bar{F} = \begin{bmatrix} 0.1925 & 0.5093 \\ 0.0559 & 0.1801 \\ 0 & 0 \end{bmatrix}$$

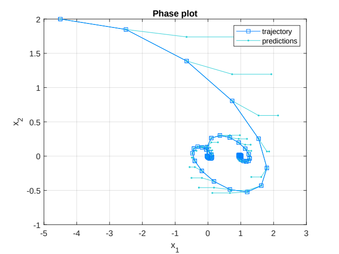
MPC Example 2.2

MPC Example 2.2 - tracking double integrator (part 3)

Simulation results for 60 sampling intervals:

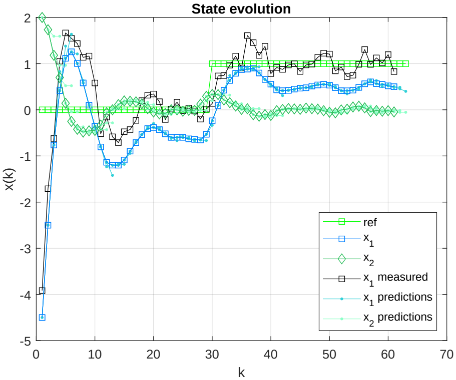






Effects of disturbances

- What if the measurements or models have errors?
- Adding a bias and noise to the state measurement in Example 2.2 **compromises** performance!
- Possible solutions:
  - Better models
  - Estimate disturbances
  - Add integrators!
- Our approach:
  - Define an incremental system
  - Add integrators to the inputs



Incremental state-space model

- Recall the discrete-time LTI state-space model:

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_d u(k) \quad , x_d(0) = x_0 \\ y_d(k) &= C_d x_d(k) \end{aligned}$$

- Define the incremental state and input vectors:

$$\Delta x_d(k) = x_d(k) - x_d(k-1) \qquad \Delta u(k) = u(k) - u(k-1)$$

- The incremental state equation becomes

$$\Delta x_d(k+1) = A_d \Delta x(k) + B_d \Delta u(k)$$

- Note that the output satisfies

$$\begin{aligned} y_d(k+1) - y_d(k) &= C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) \\ y_d(k+1) &= y_d(k) + C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) \end{aligned}$$

Integral action state-space model

- Define the augmented state vector $x(k) = \begin{bmatrix} \Delta x_d(k)^T & y_d(k)^T \end{bmatrix}^T$
- Then, the augmented state-space model with integral action is

$$
\begin{aligned}
x(k+1) &= Ax(k) + B\Delta u(k) \\
y(k) &= Cx(k)
\end{aligned}
$$

where

$$
A = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} \qquad B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} \qquad C = \begin{bmatrix} 0 & I \end{bmatrix}
$$

- For implementation, the actual control is used, $u(k) = \Delta u(k) + u(k-1)$.
- Defining $\Delta U = \begin{bmatrix} \Delta u(0)^T & \cdots & \Delta u(N-1)^T \end{bmatrix}^T$ for an horizon N, we have

$$
X = Fx(0) + G\Delta U
$$

Integral action LQT

Problem (LQTi optimal control problem)

Find the optimal control increment sequence Δu_k^* *for* $k \in \{0, \dots, N-1\}$ *that drives the system along a state trajectory* x_k^* *for* $k \in \{0, \dots, N\}$ *according to the linear system dynamics, such that the specified performance index* J_0 *is minimized relative to a reference output signal* $\bar{y}_k$*. That is,*

$$\min_{\Delta u_0, \dots, \Delta u_{N-1}} \quad J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

$$\begin{aligned} s.t. \quad & x_{k+1} = A x_k + B \Delta u_k \\ & y_k = C x_k \\ & x_0 = z_0 \end{aligned}$$

Batch LQTi problem and solution

- The LQTi optimal control problem is formulated as an unconstrained minimization quadratic problem:

$$\min_{\Delta U} \quad J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y})$$

- First order condition of optimality yields

$$\nabla_{\Delta U} J_0 = 2\tilde{R}\Delta U + 2\tilde{S}(\bar{F}x(0) - \bar{Y}) = 0$$

- Defining the gains $K_y = \tilde{R}^{-1}\tilde{S}$ and $K = K_y\bar{F}$, the optimal control and state sequences are

$$\Delta U^* = -Kx(0) + K_y\bar{Y} = \{\Delta u_0^*, \ldots, \Delta u_{N-1}^*\} \quad , \; X^* = (F - GK)x(0) + GK_y\bar{Y}$$

- Receding horizon control policy (assuming $\bar{y}$ constant):

$$\Delta u(t_k) = \Delta u_0^* = -\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} (Kx(t_k) + K_y\bar{Y}) = -K_0 x(t_k) + K_{y0}\bar{Y}$$

MPC implementation for LQTi

1. Measure the system current state $x_d(t_k)$;
2. Get incremental state initial condition $x_0 = \begin{bmatrix} x_d(t_k) - x_d(t_{k-1}) \\ C_d x_d(t_k) \end{bmatrix}^T$
3. Obtain the optimal control sequence ΔU^* solving the LQTi problem;
4. If the problem is unfeasible, terminate algorithm;
5. Get the first incremental control action from ΔU^*, defining $\Delta u(t_k) = \Delta u_0^*$;
6. Apply the actual control action $u(t_k) = u(t_{k-1}) + \Delta u(t_k)$ to the system;
7. Repeat from 1 in the next time instant, t_{k+1}.

MPC Example 2.3

Problem (MPC Example 2.3 - LQTi for double integrator)

Consider the time-invariant discrete linear system described in Example 2.1. Compute the model-based predictive controller with quadratic tracking cost

$$J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

with $N = 3$, $P = Q = 1$, $R = 4$ *and initial condition* $x(0) = \begin{bmatrix} -4.5 \\ 2 \end{bmatrix}$, *and a reference signal* $\bar{y}(k) = \varepsilon(k - 30)$, *that is,* $\bar{y}(k) = 0$ *for all* $k < 30$ *and* $\bar{y}(k) = 1$ *for all* $k \geq 30$.

MPC Example 2.3

MPC Example 2.3 - LQTi for double integrator (part 1)

Using the batch approach, the first step is to obtain the equivalent unconstrained quadratic optimization problem

$$\min_{U} \quad J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y})$$

which provides at each time step the optimal incremental control sequence

$$\Delta U^* = -(Kx(0) - K_y \bar{Y}) = \begin{bmatrix} \Delta u^*(0)^T & \Delta u^*(1)^T & \Delta u^*(2)^T \end{bmatrix}^T$$

Considering that $Y = \bar{F}x(0) + \bar{G}U$ and noting that $\bar{Q} = \mathrm{diag}\,(Q, Q, Q, P) = I_4$ and $\bar{R} = \mathrm{diag}\,(R, R, R) = 4I_3$, the final matrices are $\tilde{R} = \bar{G}^T \bar{Q} \bar{G} + \bar{R}$, $\tilde{S} = \bar{G}^T \bar{Q}$, whereas $K_y = \tilde{R}^{-1}\tilde{S}$ and $K = K_y \bar{F}$.

MPC Example 2.3

MPC Example 2.3 - LQTi for double integrator (part 2)

Considering the original system matrices as A_d, B_d, and C_d, the augmented matrices defined as

$$A = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$$

Then, the batched output matrices are given by

$$\bar{F} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 2 & 3 & 1 \\ 3 & 6 & 1 \end{bmatrix} \qquad \bar{G} = \begin{bmatrix} 0 & 0 & 0 \\ CB & 0 & 0 \\ CAB & CB & 0 \\ CA^2B & CAB & CB \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 3 & 1 & 0 \end{bmatrix}$$

MPC Example 2.3

MPC Example 2.3 - LQTi for double integrator (part 3)

The quadratic cost matrices are

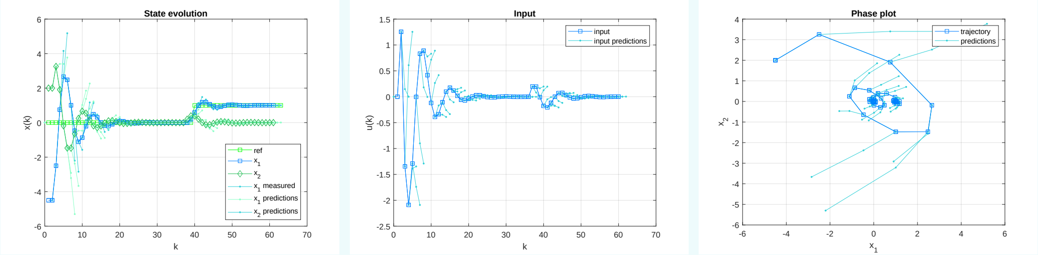
$$\tilde{R} = \bar{G}^T\bar{Q}\bar{G} + \bar{R} = \bar{G}^T\bar{G} + \bar{R} = \begin{bmatrix} 20 & 3 & 0 \\ 3 & 11 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \tilde{S} = \bar{G}^T\bar{Q} = \bar{G}^T = \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

while the control gains are

$$K_y = \bar{R}^{-1}\tilde{S} = \begin{bmatrix} 0 & 0 & 0.0521 & 0.1422 \\ 0 & 0 & -0.0142 & 0.0521 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad K = K_y\bar{F} = \begin{bmatrix} 0.5308 & 1.0095 & 0.1943 \\ 0.1280 & 0.2701 & 0.0379 \\ 0 & 0 & 0 \end{bmatrix}$$

MPC Example 2.3 - LQTi for double integrator (part 3)

Simulation results for 60 sampling intervals, including a disturbance in the measured state, x_m, used for computing the control action, defining at each sampling time, $x_m(t_k) = x_d(t_k) + 0.5 + n(t)$, where $n(t)$ is white gaussian noise.



Summary 2.1

The main learning outcomes of this class are:

- Define the LQR optimal control problem (OCP)
- Solve the LQR problem using dynamic programming
- Solve the LQR problem using the batch approach
- Implement the receding horizon control strategy for an OCP to obtain an MPC strategy
- Define and solve the LQ tracking (LQT) OCP
- Define and solve the incremental version (LQTi)

Exercise 2.2

Problem (Exercise 2.2 - MPC regulator)

Consider the time-invariant discrete linear system described by

$$x(k+1) = Ax(k) + Bu(k) \qquad A = \begin{bmatrix} 2 & 1 \\ 0 & 0.5 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Compute the model-based predictive control sequence for the first three sample times, considering the quadratic cost

$$J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k$$

with $N = 4$, $P = Q = I_2$, $R = 1$, *and* $x(0) = \begin{bmatrix} -4 & -6 \end{bmatrix}^T$.

Exercise 2.3

Problem (Exercise 2.3 - MPC tracker)

Consider the time-invariant discrete linear system described by

$$x(k+1) = Ax(k) + Bu(k) \qquad A = \begin{bmatrix} 0.4 & 1 \\ 0 & 3 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$y(k) = Cx(k) \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Compute the model-based predictive control sequence for the first three sample times, considering the quadratic tracking cost

$$J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \textstyle\sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + u_k^T R u_k$$

with $N = 3$, $P = Q = 1$, $R = 2$, $x(0) = \begin{bmatrix} 2 & 2 \end{bmatrix}^T$, *and* $\bar{y}(k) = 1$.

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Outline

Incremental state-space model

- Recall the discrete-time LTI state-space model:

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_d u(k) \quad , x_d(0) = x_0 \\ y(k) &= C_d x_d(k) \end{aligned}$$

- Define the incremental state and input vectors:

$$\Delta x_d(k) = x_d(k) - x_d(k-1) \qquad\qquad \Delta u(k) = u(k) - u(k-1)$$

- The incremental state equation becomes

$$\Delta x_d(k+1) = A_d \Delta x(k) + B_d \Delta u(k)$$

- Note that the output satisfies

$$\begin{aligned} y(k+1) - y(k) &= C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) \\ y(k+1) &= y(k) + C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) \end{aligned}$$

Integral action state-space model

- Define the augmented state vector $x(k) = \begin{bmatrix} \Delta x_d(k)^T & y(k)^T \end{bmatrix}^T$
- Then, the augmented state-space model with integral action is

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ y(k) &= Cx(k) \end{aligned}$$

where

$$A = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} \qquad B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} \qquad C = \begin{bmatrix} 0 & I \end{bmatrix}$$

- For implementation, the actual control is used, $u(k) = \Delta u(k) + u(k-1)$.
- Defining $\Delta U = \begin{bmatrix} \Delta u(0)^T & \cdots & \Delta u(N-1)^T \end{bmatrix}^T$ for an horizon N, we have

$$X = Fx(0) + G\Delta U$$

Properties of incremental model

- What happens to the state matrix eigenvalues?
  - Original state matrix A_d has characteristic polynomial:

$$\rho(A_d) = |\lambda I - A_d|$$

  - Incremental state matrix A characteristic polynomial:

$$\rho(A) = \left| \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} \right| = |\lambda I - A_d| \, |\lambda I - I| = |\lambda I - A_d| (\lambda - 1)^{n_y}$$

  - Adds n_y eigenvalues with magnitude 1 (marginally stable).

Batch LQTi problem and solution

- The LQTi optimal control problem is formulated as an unconstrained minimization quadratic problem:

$$\min_{\Delta U} \quad J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y})$$

- First order condition of optimality yields

$$\nabla_{\Delta U} J_0 = 2\tilde{R}\Delta U + 2\tilde{S}(\bar{F}x(0) - \bar{Y}) = 0$$

- Defining the gains $K_y = \tilde{R}^{-1}\tilde{S}$ and $K = K_y \bar{F}$, the optimal control sequence is

$$\Delta U^* = -Kx(0) + K_y \bar{Y} = \{\Delta u_0^*, \ldots, \Delta u_{N-1}^*\}$$

- Receding horizon control policy (assuming $\bar{y}$ constant):

$$\Delta u(t_k) = \Delta u_0^* = -\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} (Kx(0) + K_y \bar{Y}) = -K_0 x(0) + K_{y0} \bar{y}(0)$$

Integral action error state-space model

- Augmented state $x(k) = \begin{bmatrix} \Delta x_d(k)^T & e(k)^T \end{bmatrix}^T$, with $e(k) = y(k) - \bar{y}(k)$.
- Using the output computed above

$$
\begin{aligned}
y(k+1) &= y(k) + C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) \\
e(k+1) + \bar{y}(k+1) &= e(k) + \bar{y}(k) + C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) \\
e(k+1) &= e(k) + C_d A_d \Delta x_d(k) + C_d B_d \Delta u(k) - \Delta \bar{y}(k+1)
\end{aligned}
$$

- Then, assuming $\Delta \bar{y}(k+1) \approx 0$, the augmented state-space model is

$$
\begin{aligned}
x(k+1) &= Ax(k) + B\Delta u(k) \\
e(k) &= Cx(k)
\end{aligned}
$$

where matrices A, B, and C are the same as before.

- In batch notation, $X = Fx(0) + G\Delta U$ and $E = \bar{F}x(0) + \bar{G}\Delta U$.

LQT with integral action

Problem (LQTi optimal control problem)

Find the optimal control increment sequence Δu_k^* for $k \in \{0, \ldots, N-1\}$ that drives the system along a state trajectory x_k^* for $k \in \{0, \ldots, N\}$ according to the augmented linear system dynamics, such that the specified performance index J_0 is minimized relative to a reference output signal $\bar{y}_k$. That is,

$$\begin{aligned}
\min_{\Delta u_0, \ldots, \Delta u_{N-1}} \quad & J_0 = e_N^T P e_N + \sum_{k=0}^{N-1} e_k^T Q e_k + \Delta u_k^T R \Delta u_k \\
s.t. \quad & x_{k+1} = A x_k + B \Delta u_k \\
& e_k = C x_k = y_k - \bar{y}_k \\
& x_0 = z_0
\end{aligned}$$

Batch error-space LQTi problem and solution

- Using the same batch notation, the cost becomes

$$
\begin{aligned}
J_0 &= e_N^T P e_N + \textstyle\sum_{k=0}^{N-1} e_k^T Q e_k + u_k^T R u_k \\
&= x_N^T C^T P\, C x_N + \textstyle\sum_{k=0}^{N-1} x_k^T C^T Q\, C x_k + \Delta u_k^T R \Delta u_k \\
&= X^T \bar{Q}\, X + \Delta U^T \bar{R}\, \Delta U
\end{aligned}
$$

  where $\bar{Q} = \operatorname{diag}\left(C^T Q C, \ldots, C^T Q C, C^T P C\right)$ and $\bar{R} = \operatorname{diag}\left(R, \ldots, R\right)$.
- The LQTi optimal control problem is formulated as an unconstrained minimization quadratic problem:

$$
\min_{\Delta U} \quad J_0 = \Delta U^T \tilde{R} \Delta U + 2 \Delta U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0)
$$

  where $\tilde{R} = G^T \bar{Q} G + \bar{R}$, $\tilde{Q} = F^T \bar{Q} F$, and $\tilde{S} = G^T \bar{Q} F$.

Batch error-space LQTi solution

- First order condition of optimality yields

$$\nabla_{\Delta U} J_0 = 2\tilde{R}\Delta U + 2\tilde{S}x(0) = 0$$

- Defining $K = \tilde{R}^{-1}\tilde{S}$, the optimal control and state sequences are

$$\Delta U^* = -Kx(0) = \{\Delta u_0^*, \ldots, \Delta u_{N-1}^*\} \qquad X^* = (F - GK)x(0)$$

- Receding horizon control policy (also assuming $\bar{y}$ constant):

$$\Delta u(t_k) = \Delta u_0^* = -\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} Kx(0) = -K_0 x(0)$$

Properties of incremental error model

- Augmented system also adds n_y eigenvalues with magnitude 1 (marginally stable).
- A regulator is enough to drive original system to reference.
- MPC regulator control of error dynamics:
  - Control policy: $\Delta u(k) = -K_0 x(k) = -\begin{bmatrix} K_x & K_e \end{bmatrix} \begin{bmatrix} \Delta x_d(k) \\ e(k) \end{bmatrix}$
  - Note that MPC control gain K_0 does not depend on state or reference.
  - Obtain actual control:

$$\begin{aligned} u(k) &= u(k-1) + \Delta u(k) \\ &= u(k-1) - K_0 x(k) \\ &= u(k-1) - K_x \Delta x_d(k) - K_e e(k) \\ &= u(k-1) - K_x \Delta x_d(k) - K_e[y(k) - \bar{y}(k)] \end{aligned}$$

Resulting integral MPC policy

- Control policy with integral effect (simplified notation):

$$
\begin{aligned}
u_k &= u_{k-1} - K_x \Delta x_k - K_e e_k \\
&= u_{k-2} - K_x \Delta x_{k-1} - K_e e_{k-1} - K_x \Delta x_k - K_e e_k \\
&= u_{k-2} - K_x [\Delta x_{k-1} + \Delta x_k] - K_e [e_{k-1} + e_k] \\
&= u_{k-2} - K_x [x_{k-1} - x_{k-2} + x_k - x_{k-1}] - K_e [e_{k-1} + e_k] \\
&= u_{k-2} + K_x x_{k-2} - K_x x_k - K_e [e_{k-1} + e_k] \\
&= u_{k-3} + K_x x_{k-3} - K_x x_k - K_e [e_{k-2} + e_{k-1} + e_k] = \cdots \\
&= u_0 + K x_0 - K_x x_k - K_e \sum_{j=1}^{k} e_j
\end{aligned}
$$

- Assuming $u_0 = -K x_0 - K_e e_0$ results in the PI control law

$$u(k) = -K_x x(k) - K_e \sum_{j=0}^{k} [y(k) - \bar{y}(k)]$$

Example 2.4

Problem (MPC Example 2.4 - Error LQTi for double integrator)

Consider the time-invariant discrete linear system described by

$$x_d(k+1) = A_d x_d(k) + B_d u_d(k) \qquad A_d = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad B_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C_d = \begin{bmatrix} 1 & 0 \end{bmatrix}$$
$$y(k) = C_d x_d(k)$$

Using an augmented error state-space system, compute the model-based predictive controller with quadratic tracking cost

$$J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

with $N = 3$, $P = Q = 1$, $R = 4$ *and initial condition* $x(0) = \begin{bmatrix} -4.5 & 2 \end{bmatrix}^T$, *and a reference signal* $\bar{y}(k) = \varepsilon(k-30)$.

Example 2.4

Example 2.4 - Error LQTi for double integrator (part 1)

Considering the original system matrices as A_d, B_d, and C_d, the augmented matrices are defined as

$$A = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & I \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

The batch state matrices for $X = Fx(0) + G\Delta U$ are given by (see next slide)

Example 2.4

Example 2.4 - Error LQTi for double integrator (part 2)

$$F = \begin{bmatrix} I \\ A \\ A^2 \\ A^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \\ 1 & 3 & 0 \\ 0 & 1 & 0 \\ 3 & 6 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 & 0 \\ B & 0 & 0 \\ AB & B & 0 \\ A^2B & AB & B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \\ 3 & 1 & 0 \end{bmatrix}$$

Example 2.4

Example 2.4 - Error LQTi for double integrator (part 3)

The equivalent unconstrained quadratic optimization problem is then

$$\min_{U} \quad J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S} x(0) + x(0)^T \bar{Q} x(0)$$

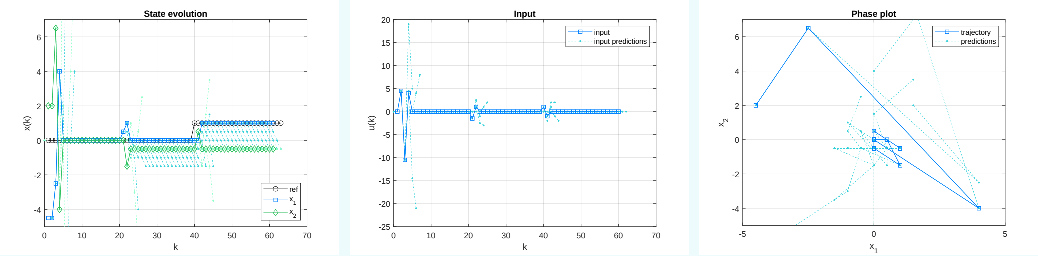
which results in the optimal incremental control sequence $\Delta U^* = -Kx(0)$. As $\bar{Q} = \mathrm{diag}\left(C^T QC, C^T QC, C^T QC, C^T PC\right)$ and $\bar{R} = \mathrm{diag}\left(R, R, R\right) = 4I_3$, the gain is $K = \tilde{R}^{-1}\tilde{S}$ with

$$\tilde{R} = G^T \bar{Q} G + \bar{R} = \begin{bmatrix} 14 & 7 & 4 \\ 7 & 5 & 4 \\ 4 & 4 & 4 \end{bmatrix} \qquad \tilde{S} = G^T \bar{Q} F = \begin{bmatrix} 11 & 21 & 4 \\ 3 & 6 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

MPC Example 2.4

MPC Example 2.4 - Error LQTi for double integrator (part 4)

Simulation results for 60 sampling intervals, including a constant bias disturbance of 0.5 in the applied input from $k = 20$, and a reference stap at $k = 40$, are provided below.



Outline

Problem definition

Problem (Constrained LQR optimal control problem)

Find the optimal control sequence u_k^* for $k = 0, \ldots, N-1$ that drives the system along a state trajectory x_k^* for $k = 0, \ldots, N$ according to its linear system dynamics and within the input and state constraints, while minimizing the performance index J_0. That is,

$$
\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \\
s.t. \quad & x_{k+1} = A x_k + B u_k \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_0 = z_0 \\
& x_k \in \mathcal{X} \ , \ u_k \in \mathcal{U} \ , \ \forall_{k=0,\ldots,N-1} \\
& x_N \in \mathcal{X}_f
\end{aligned}
$$

Batch approach to constrained LQR

- The constrained LQR optimal control problem can be reformulated as an minimization problem:

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0)$$

$$s.t. \quad x_k \in \mathcal{X} \ , \ u_k \in \mathcal{U} \ , \ \forall_{k=0,\dots,N-1} \ , \ x_N \in \mathcal{X}_f$$

- How to define the input and state constraints?
  - Use polyhedrons to approximate the sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{X}_f$.
  - Polyhedrons can be defined as a set o linear constraints.
- Typical constraints:
  - Input: $u_{min} \leq u(k) \leq u_{max}$
  - Input variations: $\Delta u_{min} \leq \Delta u(k) \leq \Delta u_{max}$
  - State: $x_{min} \leq x(k) \leq x_{max}$
  - Output: $y_{min} \leq y(k) \leq y_{max}$
- How to define them in batch notation?

Batch input constraints

- Considering $U = \begin{bmatrix} u_0^T & \cdots & u_N^T \end{bmatrix}^T$ as the optimization variables, the input constraint can be defined as

  $$U_{min} \leq U \leq U_{max}$$

  where $U_{max} = \begin{bmatrix} u_{max}^T & \cdots & u_{max}^T \end{bmatrix}^T$ and U_{min} is defined similarly.
- Separating into two inequalities results in

  $$-U \leq -U_{min} \quad \text{and} \quad U \leq U_{max}$$

  which can be stated in matrix form as

  $$\begin{bmatrix} -I \\ I \end{bmatrix} U \leq \begin{bmatrix} -U_{min} \\ U_{max} \end{bmatrix} \Leftrightarrow M_u U \leq w_u$$

Batch input variation constraints

- How to define the constraint on $\Delta u(k)$?
- Express ΔU in terms of U and some initial condition $u(-1)$:

$$\begin{aligned}
\Delta u(0) &= u(0) - u(-1) \\
\Delta u(1) &= u(1) - u(0) \\
&\vdots \\
\Delta u(N-1) &= u(N-1) - u(N-2)
\end{aligned}$$

which results in the batch equation

$$\begin{bmatrix} \Delta u(0) \\ \Delta u(1) \\ \vdots \\ \Delta u(N-1) \end{bmatrix} = \left(\begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & 0 \\ I & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \right) \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} - \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(-1)$$

Batch input variation constraints

- As in matrix form we have that $\Delta U = M_1 U + M_2 u(-1)$, the constraint becomes

$$\Delta U_{min} \leq \Delta U \leq \Delta U_{max} \Leftrightarrow \Delta U_{min} \leq M_1 U + M_2 u(-1) \leq \Delta U_{max}$$

where $\Delta U_{max} = \begin{bmatrix} \Delta u_{max}^T & \cdots & \Delta u_{max}^T \end{bmatrix}^T$ and ΔU_{min} is defined similarly.

- Separating into two inequalities results in

$$-M_1 U \leq -\Delta U_{min} + M_2 u(-1) \quad \text{and} \quad M_1 U \leq \Delta U_{max} - M_2 u(-1)$$

which can be written as

$$\begin{bmatrix} -M_1 \\ M_1 \end{bmatrix} U \leq \begin{bmatrix} -\Delta U_{min} + M_2 u(-1) \\ \Delta U_{max} - M_2 u(-1) \end{bmatrix} \Leftrightarrow M_{\Delta u} U \leq w_{\Delta u}$$

Batch state and output constraints

- Noting that $X = Fx(0) + GU$, the state constraint can be defined as

  $$X_{min} \leq X \leq X_{max} \Leftrightarrow X_{min} \leq Fx(0) + GU \leq X_{max}$$

  where $X_{max} = \begin{bmatrix} x_{max}^T & \cdots & x_{max}^T \end{bmatrix}^T$ and X_{min} is defined similarly.
- Separating into two inequalities results in

  $$-GU \leq -X_{min} + Fx(0) \quad \text{and} \quad GU \leq X_{max} - Fx(0)$$

  which can be stated in matrix form as

  $$\begin{bmatrix} -G \\ G \end{bmatrix} U \leq \begin{bmatrix} -X_{min} + Fx(0) \\ X_{max} - Fx(0) \end{bmatrix} \Leftrightarrow M_x U \leq w_x$$
- A similar procedure for the output, with $Y = \bar{F}x(0) + \bar{G}U$, results in

  $$\begin{bmatrix} -\bar{G} \\ \bar{G} \end{bmatrix} U \leq \begin{bmatrix} -Y_{min} + \bar{F}x(0) \\ Y_{max} - \bar{F}x(0) \end{bmatrix} \Leftrightarrow M_y U \leq w_y$$

Batch equivalent to constrained LQR

- Stacking all the above constraints results in

$$\begin{bmatrix} M_u \\ M_{\Delta u} \\ M_x \\ M_y \end{bmatrix} U \leq \begin{bmatrix} w_u \\ w_{\Delta u} \\ w_x \\ w_y \end{bmatrix} \Leftrightarrow MU \leq w$$

- The constrained LQR optimal control problem can be reformulated as:

$$\begin{aligned} \min_{U} \quad & J_0 = U^T \tilde{R} U + 2U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0) \\ s.t. \quad & MU \leq w \end{aligned}$$

- Convex problem: quadratic positive definite cost and linear constraints.
- Solutions computed using Lagrange multipliers for active constraints, etc.
- What about the tracking and incremental optimal control problems?

Problem definition

Problem (Constrained LQTi optimal control problem)

Find the optimal control increment sequence Δu_k^* for $k = 0, \ldots, N-1$ that drives the system along a state trajectory x_k^* for $k = 0, \ldots, N$ according to its linear system dynamics and within the input and state constraints, such that the cost J_0 is minimized relative to a reference output signal $\bar{y}_k$. That is,

$$\min_{\Delta u_0, \ldots, \Delta u_{N-1}} \quad J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_0 = z_0$$

$$x_k \in \mathcal{X} \ , \ u_k \in \mathcal{U} \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_N \in \mathcal{X}_f$$

Batch input variation constraints

- Considering $\Delta U = \begin{bmatrix} \Delta u_0^T & \cdots & \Delta u_N^T \end{bmatrix}^T$ as the optimization variables, the input variation constraint can be defined as

  $$\Delta U_{min} \leq \Delta U \leq \Delta U_{max}$$

  where $\Delta U_{max} = \begin{bmatrix} \Delta u_{max}^T & \cdots & \Delta u_{max}^T \end{bmatrix}^T$ and ΔU_{min} is defined similarly.
- Separating into two inequalities results in

  $$-\Delta U \leq -\Delta U_{min} \quad \text{and} \quad \Delta U \leq \Delta U_{max}$$

  which can be stated in matrix form as

  $$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U_{min} \\ \Delta U_{max} \end{bmatrix} \Leftrightarrow M_{\Delta u} \Delta U \leq w_{\Delta u}$$

Batch input constraints

- How to define the constraint on $u(k)$?
- Express U in terms of ΔU and some initial condition $u(-1)$:

$$
\begin{aligned}
u(0) &= u(-1) + \Delta u(0) \\
u(1) &= u(0) + \Delta u(1) = u(-1) + \Delta u(0) + \Delta u(1) \\
&\vdots \\
u(N-1) &= u(N-2) + \Delta u(N-1) = u(-1) + \textstyle\sum_{k=0}^{N-1} \Delta u(k)
\end{aligned}
$$

which results in the batch equation

$$
\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \begin{bmatrix} \Delta u(0) \\ \Delta u(1) \\ \vdots \\ \Delta u(N-1) \end{bmatrix} + \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} u(-1)
$$

Batch input constraints

- As in matrix form we have that $U = M_3 \Delta U + M_4 u(-1)$, the constraint becomes

$$U_{min} \leq U \leq U_{max} \Leftrightarrow U_{min} \leq M_3 \Delta U + M_4 u(-1) \leq U_{max}$$

where $U_{max} = \begin{bmatrix} u_{max}^T & \cdots & u_{max}^T \end{bmatrix}^T$ and U_{min} is defined similarly.

- Separating into two inequalities results in

$$-M_3 \Delta U \leq -U_{min} + M_4 u(-1) \quad \text{and} \quad M_3 \Delta U \leq U_{max} - M_4 u(-1)$$

which can be written as

$$\begin{bmatrix} -M_3 \\ M_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} -U_{min} + M_4 u(-1) \\ U_{max} - M_4 u(-1) \end{bmatrix} \Leftrightarrow M_u \Delta U \leq w_u$$

Batch state and output constraints

- Noting that $X = Fx(0) + G\Delta U$, the state constraint can be defined as

  $$X_{min} \leq X \leq X_{max} \Leftrightarrow X_{min} \leq Fx(0) + G\Delta U \leq X_{max}$$

  where $X_{max} = \begin{bmatrix} x_{max}^T & \cdots & x_{max}^T \end{bmatrix}^T$ and X_{min} is defined similarly.
- Separating into two inequalities results in

  $$-G\Delta U \leq -X_{min} + Fx(0) \quad \text{and} \quad G\Delta U \leq X_{max} - Fx(0)$$

  which can be stated in matrix form as

  $$\begin{bmatrix} -G \\ G \end{bmatrix} \Delta U \leq \begin{bmatrix} -X_{min} + Fx(0) \\ X_{max} - Fx(0) \end{bmatrix} \Leftrightarrow M_x \Delta U \leq w_x$$
- A similar procedure for the output, with $Y = \bar{F}x(0) + \bar{G}\Delta U$, results in

  $$\begin{bmatrix} -\bar{G} \\ \bar{G} \end{bmatrix} \Delta U \leq \begin{bmatrix} -Y_{min} + \bar{F}x(0) \\ Y_{max} - \bar{F}x(0) \end{bmatrix} \Leftrightarrow M_y \Delta U \leq w_y$$

Batch equivalent to constrained LQTi

- Stacking all the above constraints results in

$$\begin{bmatrix} M_{\Delta u} \\ M_u \\ M_x \\ M_y \end{bmatrix} \Delta U \leq \begin{bmatrix} w_{\Delta u} \\ w_u \\ w_x \\ w_y \end{bmatrix} \Leftrightarrow M\Delta U \leq w$$

- The constrained LQR optimal control problem can be reformulated as:

$$\begin{aligned} \min_{\Delta U} \quad & J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y}) \\ s.t. \quad & M\Delta U \leq w \end{aligned}$$

- Convex problem: quadratic positive definite cost and linear constraints.
- Solutions computed using Lagrange multipliers for active constraints, etc.

Example 2.5

Problem (Example 2.5 - Constrained minimization)

Find the optimal solution of the constrained minimization problem

$$\begin{aligned}\min_{x} \quad & J = 1/2x^T Q x + x^T s + c \\ s.t. \quad & Mx \leq w\end{aligned}$$

where the cost function is defined by $Q = I_3$, $s = \begin{bmatrix} -2 & -3 & -1 \end{bmatrix}^T$, *and* $c = 0$,

whereas the constraints are defined by $M = \begin{bmatrix} 1 & 1 & 1 \\ 3 & -2 & -3 \\ 1 & -3 & 2 \end{bmatrix}$ *and* $w = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$.

Example 2.5

Example 2.5 - Constrained minimization (part 1)

Considering the parameters defined above, we can rewrite compute the augmented cost functional considering the Lagrange for the constraints as

$$J = 1/2x^TQx + x^Ts + \lambda^T(Mx - w)$$

for which the gradient results in

$$J_x = Qx + s + M^T\lambda$$
$$J_\lambda = Mx - w$$

Notice that the system of equations represented by $Mx = w$ has at least one solution, which means that the set of constraints are feasible.

Example 2.5

Example 2.5 - Constrained minimization (part 2)

Using the active set method [1, Example 2.9], we first assume all three constraints are active, using them as equality constraints, yielding

$$
\begin{aligned}
J_x &= Qx + s + M^T\lambda = 0 \\
& Qx + s + M^T\lambda = 0 \\
x &= -Q^{-1}(M^T\lambda + s)
\end{aligned}
\qquad \text{and} \qquad
\begin{aligned}
J_\lambda &= Mx - w = 0 \\
-MQ^{-1}(M^T\lambda + s) &= w \\
MQ^{-1}M^T\lambda &= -(MQ^{-1}s + w)
\end{aligned}
$$

which result in

$$\lambda = (MQ^{-1}M^T)^{-1}(MQ^{-1}s + w) = \begin{bmatrix} 1.6873 & 0.0309 & -0.4352 \end{bmatrix}^T$$

As the 3rd element of λ is negative, the respective constraint is inactive.

Example 2.5

Example 2.5 - Constrained minimization (part 3)

As such, considering now that the active constraint is $M_{act}x \leq w_{act}$ where $M_{act} = \begin{bmatrix} 1 & 1 & 1 \\ 3 & -2 & -3 \end{bmatrix}$ and $w_{act} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, we obtain

$$\lambda = (M_{act}Q^{-1}M_{act}^T)^{-1}(M_{act}Q^{-1}s + w_{act}) = \begin{bmatrix} 1.6452 & -0.0323 \end{bmatrix}^T$$

As the 2nd element of λ is negative, the respective constraint is inactive, leaving just one active constraint, with $M_{act} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ and $w_{act} = 1$.
Thus, the resulting optimal solution is $\lambda^* = 1.6667$ and

$$x^* = -Q^{-1}(M_{act}^T\lambda^* + s) = \begin{bmatrix} 0.3333 & 1.3333 & -0.6667 \end{bmatrix}^T$$

Example 2.5

Example 2.5 - Constrained minimization (part 4)

In practice, we use state-of-the-art solvers, such as those in Matlab, as shown in the code below using the function `quadprog`, which yields the same solution.

```
Q = eye(3); s = [-2;-3;-1];
M = [1 1 1; 3 -2 -3; 1 -3 2]; w = [1;1;1];

Ma = [1 1 1]; wa = 1;
lbd_opt = -( Ma*Q^(-1)*Ma' )^(-1)*(Ma*Q^(-1)*s + wa ),
x_opt = -Q^-1*(Ma'*lbd_opt + s),

x_opt2 = quadprog(Q,s,M,w),
```

Outline

Model-based Predictive Control (MPC)

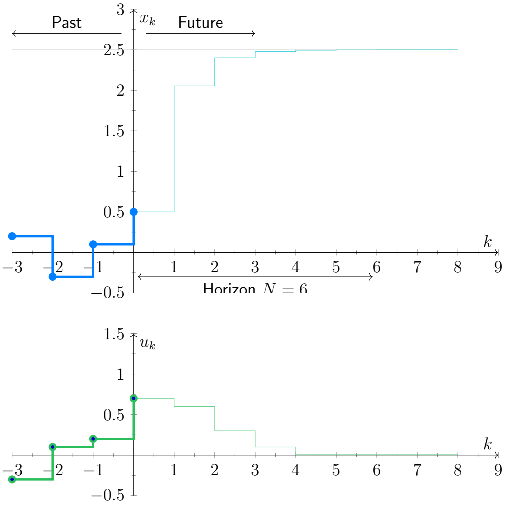
Ingredients:

- Cost function
- Model, input, and state constraint
- Receding horizon

Solve optimization problem at each instant

$$\min_{u_i,\ldots,u_{i+N-1}} \quad J_i(x_i) = p(x_N) + \sum_{k=i}^{i+N-1} q(x_k, u_k)$$

$$\begin{aligned} s.t. \quad & x_{k+1} = f(x_k, u_k) \; \forall_{k=i,\ldots,i+N-1} \\ & f_{eq}(x_k, u_k) = 0 \\ & f_{in}(x_k, u_k) \le 0 \end{aligned}$$



Model-based Predictive Control (MPC)

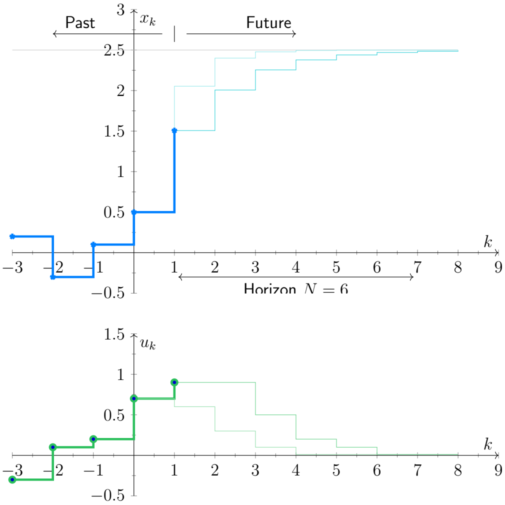
Ingredients:

- Cost function
- Model, input, and state constraint
- Receding horizon

Solve optimization problem at each instant

$$\min_{u_i,\ldots,u_{i+N-1}} \quad J_i(x_i) = p(x_N) + \sum_{k=i}^{i+N-1} q(x_k, u_k)$$

$$\begin{aligned} s.t. \quad & x_{k+1} = f(x_k, u_k) \ \forall_{k=i,\ldots,i+N-1} \\ & f_{eq}(x_k, u_k) = 0 \\ & f_{in}(x_k, u_k) \leq 0 \end{aligned}$$



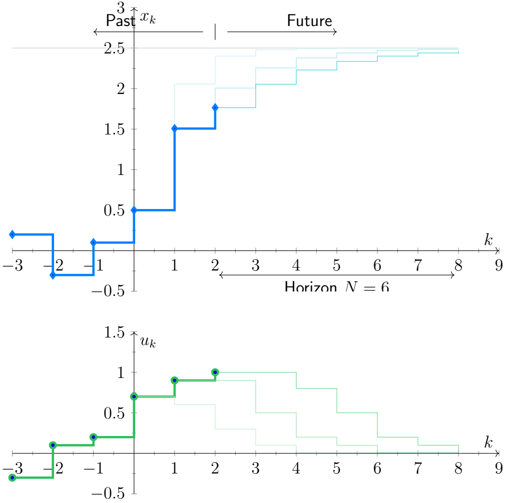
Model-based Predictive Control (MPC)

Ingredients:

- Cost function
- Model, input, and state constraint
- Receding horizon

Solve optimization problem at each instant

$$
\begin{aligned}
\min_{u_i,\ldots,u_{i+N-1}} \quad & J_i(x_i) = p(x_N) + \sum_{k=i}^{i+N-1} q(x_k, u_k) \\
s.t. \quad & x_{k+1} = f(x_k, u_k) \; \forall_{k=i,\ldots,i+N-1} \\
& f_{eq}(x_k, u_k) = 0 \\
& f_{in}(x_k, u_k) \leq 0
\end{aligned}
$$



Constrained MPC

- Constrained predictive control problem:

$$
\begin{aligned}
\min_{U} \quad & J_0(x_0, U) = x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_0 = x_{t_k} \\
& x_k \in \mathcal{X} \ , \ u_k \in \mathcal{U} \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_N \in \mathcal{X}_f
\end{aligned}
$$

- Equivalent batch formulation:

$$
\begin{aligned}
\min_{U} \quad & J_0(x_0, U) = U^T \tilde{R} U + 2 U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0) \\
s.t. \quad & MU \leq w
\end{aligned}
$$

- Constrained optimal control sequence: $U^* = \{u_0^*, u_1^*, \ldots, u_{N-1}^*\}$
- Receding horizon control policy: $u(t_k) = u_0^*$.

Receding horizon control algorithm

1. Consider the current state $x(t_k)$ as the initial condition x_0;
2. Obtain the constrained optimal control sequence U^* solving the optimal control problem;
3. If problem unfeasible, terminate algorithm;
4. Apply the first control action from U^* to the system, $u(t_k) = u_0^*$;
5. Repeat from 1 in the next time instant, t_{k+1}.

Example 2.6

Problem (Example 2.6 - Constrained MPC for double integrator)

Consider the time-invariant discrete linear system described by

$$x(k+1) = Ax(k) + Bu(k) \qquad A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Compute the model-based predictive controller considering the quadratic cost

$$J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k)$$

with $N = 3$, $P = Q = I$, $R = 10$ *and initial condition* $x(0) = \begin{bmatrix} -4.5 & 2 \end{bmatrix}^T$, *considering the input constraint* $-0.3 \leq u(k) \leq 0.3$ *and the state constraint* $-x_{max} \leq x(k) \leq x_{max}$ *for all* $k \geq 0$, *where* $x_{max} = \begin{bmatrix} 5 & 5 \end{bmatrix}^T$.

Example 2.6

Example 2.6 - Double integrator MPC (part 1)

Using the batch approach, the first step is to obtain the equivalent constrained quadratic optimization problem, given by

$$\min_{U} \quad J_0 = U^T \tilde{R} U + 2U^T \tilde{S} x(0) + x(0)^T \tilde{Q} x(0) \qquad\qquad s.t. \quad MU \leq w$$

where, considering that $X = Fx(0) + GU$ and noting that $\bar{Q} = \text{diag}\,(Q, Q, Q, P) = I_8$ and $\bar{R} = \text{diag}\,(R, R, R) = 10I_3$, the final cost matrices are $\tilde{R} = G^T \bar{Q} G + \bar{R} = G^T G + 10I_3$, $\tilde{Q} = F^T \bar{Q} F = F^T F$, and $\tilde{S} = G^T \bar{Q} F = G^T F$.

Regarding the constraint, we have that $M = \begin{bmatrix} M_u \\ M_x \end{bmatrix}$ and $w = \begin{bmatrix} w_u \\ w_x \end{bmatrix}$ corresponding to the input and state constraints, as given below.

Example 2.6

Example 2.6 - Double integrator MPC (part 2)

The state batched matrices are given by

$$F = \begin{bmatrix} I \\ A \\ A^2 \\ A^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 2 \\ 0 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} 0 & 0 & 0 \\ B & 0 & 0 \\ AB & B & 0 \\ A^2B & AB & B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Example 2.6

Example 2.6 - Double integrator MPC (part 3)

The quadratic cost matrices are

$$\tilde{Q} = F^T \bar{Q} F = F^T F = \begin{bmatrix} 4 & 6 \\ 6 & 18 \end{bmatrix}$$

$$\tilde{R} = G^T \bar{Q} G + \bar{R} = G^T G + \bar{R} = \begin{bmatrix} 18 & 4 & 1 \\ 4 & 13 & 1 \\ 1 & 1 & 11 \end{bmatrix}$$

$$\tilde{S} = G^T \bar{Q} F = G^T F = \begin{bmatrix} 3 & 11 \\ 1 & 5 \\ 0 & 1 \end{bmatrix}$$

Example 2.6

Example 2.6 - Double integrator MPC (part 4)

Considering $U_{max} = \begin{bmatrix} 0.3 & 0.3 & 0.3 \end{bmatrix}^T$ and $X_{max} = \begin{bmatrix} x_{max}^T & x_{max}^T & x_{max}^T & x_{max}^T \end{bmatrix}^T$, the constraints matrices are given by

$$M_u = \begin{bmatrix} -I_3 \\ I_3 \end{bmatrix} \qquad w_u = \begin{bmatrix} U_{max} \\ U_{max} \end{bmatrix}$$

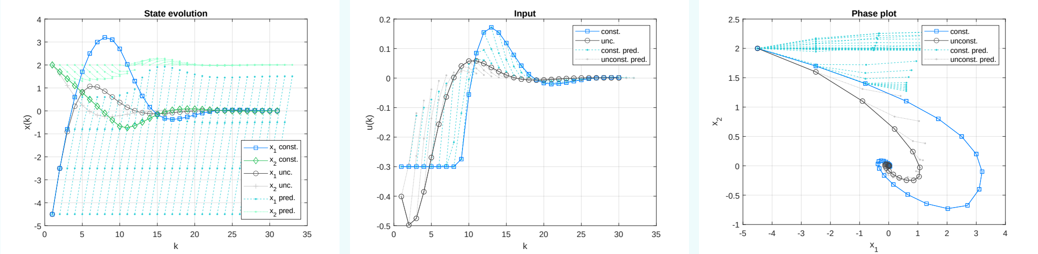
$$M_x = \begin{bmatrix} -G \\ G \end{bmatrix} \qquad w_x = \begin{bmatrix} X_{max} + Fx(0) \\ X_{max} - Fx(0) \end{bmatrix}$$

Now, we have everything we need to solve the constrained quadratic programming problem using Matlab function `quadprog(2*Rt,2*St*xk,M,w)`.

Example 2.6

Example 2.6 - Double integrator MPC (part 5)

Simulation results for 30 sampling intervals:



Constrained tracking MPC

- Constrained tracking predictive control problem:

$$\min_{\Delta U} \quad J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k \ , \ y_k = Cx_k \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_0 = x_{t_k}$$

$$x_k \in \mathcal{X} \ , \ u_k \in \mathcal{U} \ , \ \forall_{k=0,\ldots,N-1} \ , \ x_N \in \mathcal{X}_f$$

- Equivalent batch formulation:

$$\min_{\Delta U} J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y})$$

$$s.t. \, M\Delta U \leq w$$

- Constrained optimal control sequence: $\Delta U^* = \{\Delta u_0^*, \ldots, \Delta u_{N-1}^*\}$
- Receding horizon control policy: $u(t_k) = u(t_{k-1}) + \Delta u_0^*$.

Example 2.7

Problem (Example 2.7 - Tracking MPC for double integrator)

Consider the time-invariant discrete linear system described in Example 4.3, with $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$. *Compute the model-based predictive controller with quadratic tracking cost*

$$J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

with $N = 3$, $P = Q = 1$, $R = 4$ *and initial condition* $x(0) = \begin{bmatrix} -4.5 & 2 \end{bmatrix}^T$, *and a reference signal* $\bar{y}(k) = \varepsilon(k - 40)$, *considering the input constraint* $-0.3 \leq u(k) \leq 0.3$ *for all* $k \geq 0$.

Example 2.7

Example 2.7 - double integrator tracking MPC (part 1)

The first step is to obtain the equivalent constrained quadratic optimization problem

$$
\begin{aligned}
\min_{\Delta U} \quad & J_0 = \Delta U^T \tilde{R} \Delta U + 2\Delta U^T \tilde{S}(\bar{F}x(0) - \bar{Y}) + (\bar{F}x(0) - \bar{Y})^T \bar{Q}(\bar{F}x(0) - \bar{Y}) \\
s.t. \quad & M\Delta U \leq w
\end{aligned}
$$

Considering that $Y = \bar{F}x(0) + \bar{G}U$ and noting that $\bar{Q} = \operatorname{diag}(Q, Q, Q, P) = I_4$ and $\bar{R} = \operatorname{diag}(R, R, R) = 10I_3$, the final matrices are $\tilde{R} = \bar{G}^T \bar{Q} \bar{G} + \bar{R}$, $\tilde{S} = \bar{G}^T \bar{Q}$.
Regarding the constraint, we have that $M = M_u$ and $w = w_u$ corresponding to the input and state constraints, as given below.

Example 2.7

Example 2.7 - double integrator tracking MPC (part 2)

Considering the original system matrices as A_d, B_d, and C_d, the augmented matrices defined as

$$A = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$$

Then, the batched output matrices are given by

$$\bar{F} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 2 & 3 & 1 \\ 3 & 6 & 1 \end{bmatrix} \qquad \bar{G} = \begin{bmatrix} 0 & 0 & 0 \\ CB & 0 & 0 \\ CAB & CB & 0 \\ CA^2B & CAB & CB \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 3 & 1 & 0 \end{bmatrix}$$

Example 2.7

Example 2.7 - double integrator tracking MPC (part 3)

The quadratic cost matrices are

$$\tilde{R} = \bar{G}^T \bar{Q} \bar{G} + \bar{R} = \bar{G}^T \bar{G} + \bar{R} = \begin{bmatrix} 20 & 3 & 0 \\ 3 & 11 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \tilde{S} = \bar{G}^T \bar{Q} = \bar{G}^T = \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Example 2.7

Example 2.7 - double integrator tracking MPC (part 4)

Considering $U_{max} = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0.3 \end{bmatrix}^T$, the input constraint matrices are given by

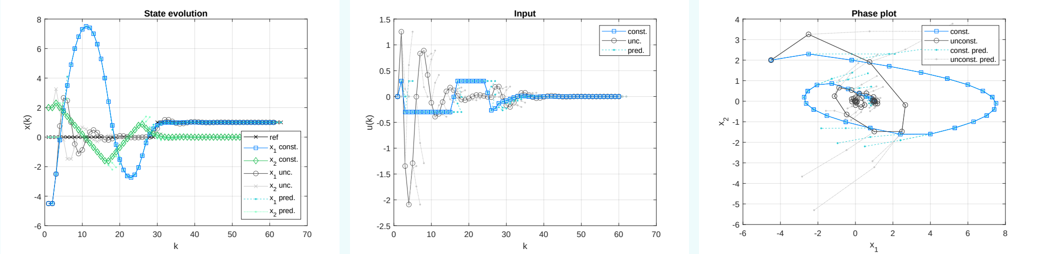
$$M_u = \begin{bmatrix} -M_3 \\ M_3 \end{bmatrix} \quad w_u = \begin{bmatrix} U_{max} + M_4 u(t_{k-1}) \\ U_{max} - M_4 u(t_{k-1}) \end{bmatrix} \quad M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad M_4 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Now, we have everything we need to solve the constrained quadratic programming problem using Matlab function `quadprog(2*Rt,2*St*(Fb*xk-Yb),M,w)`.

Example 2.7

Example 2.7 - double integrator tracking MPC (part 5)

Simulation results for 60 sampling intervals are given below.



Summary 2.2

The main learning outcomes of this class are:

- Define an augmented model to obtain tracking and integral action from and LQR OCP.
- Define linear state and input constraints using the batch approach
- Define and solve basic optimal control problems
- Define and solve linearly constrained MPC problems

Exercise 2.8

Problem (Exercise 2.8 - Constrained MPC regulator)

Consider the time-invariant discrete linear system described by

$$x(k+1) = Ax(k) + Bu(k) \qquad A = \begin{bmatrix} 1.3 & 1 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Compute the first 3 samples of the MPC control sequence considering the quadratic cost

$$J_0 = x_N^T P x_N + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k)$$

with $N = 3$, $P = Q = I$, $R = 15$ *and initial condition* $x(0) = \begin{bmatrix} -2 & 2 \end{bmatrix}^T$, *considering the input constraint* $-0.8 \leq u(k) \leq 0.8$ *and the state constraint* $-x_{max} \leq x(k) \leq x_{max}$ *for all* $k \geq 0$, *where* $x_{max} = \begin{bmatrix} 2.5 & 2.5 \end{bmatrix}^T$.

Exercise 2.9

Problem (Exercise 2.9 - Constrained MPC tracker)

Consider the time-invariant discrete linear system described by

$$x(k+1) = Ax(k) + Bu(k) \qquad A = \begin{bmatrix} 0.3 & 0.6 \\ 0 & 1.2 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Compute the first 3 samples of the MPC control sequence considering

$$J_0 = (y_N - \bar{y}_N)^T P (y_N - \bar{y}_N) + \sum_{k=0}^{N-1} (y_k - \bar{y}_k)^T Q (y_k - \bar{y}_k) + \Delta u_k^T R \Delta u_k$$

with $N = 3$, $P = Q = 1$, $R = 4$ *and initial condition* $x(0) = \begin{bmatrix} -2 & 2 \end{bmatrix}^T$, *and a reference signal* $\bar{y}(k) = 2$, *considering the input constraint* $-0.8 \leq u(k) \leq 0.8$ *for all* $k \geq 0$.

Bibliography

Liuping Wang.
Model Predictive Control System Design and Implem. Using MATLAB
Springer, 2009. [Chapter 1]

F. Borrelli, A. Bemporad, M. Morari.
Predictive Control for Linear and Hybrid Systems
Cambridge, 2017. [Chapter 8 and 12]

Frank L. Lewis, Draguna Vrabie, and Vassilis L. Syrmos.
Optimal control (Chapter 2)
John Wiley & Sons, 3rd edition, 2012.