Model Predictive Control

4. Model Predictive Control without Constraints

Jun.-Prof. Dr.-Ing. Daniel Görges
Juniorprofessur für Elektromobilität
Technische Universität Kaiserslautern

System Model

- **Discrete-Time Linear Time-Invariant (LTI) System**

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{w}(k) \qquad \text{state equation} \tag{4.1}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) + \boldsymbol{v}(k) \qquad \text{output equation} \tag{4.2}$$

- **Symbols**

$\boldsymbol{x}(k) \in \mathbb{X} \subseteq \mathbb{R}^n$ state vector

$\boldsymbol{u}(k) \in \mathbb{U} \subseteq \mathbb{R}^m$ input vector

$\boldsymbol{y}(k) \in \mathbb{Y} \subseteq \mathbb{R}^p$ output vector

$\boldsymbol{w}(k) \in \mathbb{R}^n$ system disturbance vector

$\boldsymbol{v}(k) \in \mathbb{R}^p$ measurement noise vector

$\boldsymbol{A} \in \mathbb{R}^{n \times n}$ system matrix

$\boldsymbol{B} \in \mathbb{R}^{n \times m}$ input matrix

$\boldsymbol{C} \in \mathbb{R}^{p \times n}$ output matrix

$\boldsymbol{D} \in \mathbb{R}^{p \times m}$ feedthrough matrix

System Model

- **Assumptions**
  - $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable and $(\boldsymbol{C}, \boldsymbol{A})$ is detectable
  - No constraints ($\mathbb{X} = \mathbb{R}^n, \mathbb{U} = \mathbb{R}^m, \mathbb{Y} = \mathbb{R}^p$) — removed in Chapter 5
  - State feedback ($\boldsymbol{C} = \boldsymbol{I}_{n \times n}$) — removed in Chapter 7
  - No disturbance and noise ($\boldsymbol{w}(k) = \boldsymbol{0}, \boldsymbol{v}(k) = \boldsymbol{0}$) — removed in Chapter 7
  - Regulation of the state to the origin ($\boldsymbol{x}(k) \to \boldsymbol{0}$ as $k \to \infty$) — removed in Chapter 7
  - No uncertainties ($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ known exactly) — removed in Chapter 8

Cost Function

- **Discrete-Time Quadratic Cost Function**

$$V_N\big(\boldsymbol{x}(k),\boldsymbol{U}(k)\big) = \boldsymbol{x}^T(k+N)\boldsymbol{P}\boldsymbol{x}(k+N) + \sum_{i=0}^{N-1} \boldsymbol{x}^T(k+i)\boldsymbol{Q}\boldsymbol{x}(k+i) + \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i) \quad (4.3)$$

- **Symbols**
  - $\boldsymbol{U}(k) = (\boldsymbol{u}^T(k) \quad \boldsymbol{u}^T(k+1) \quad \cdots \quad \boldsymbol{u}^T(k+N-1))^T \in \mathbb{R}^{Nm}$ input sequence
  - $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ symmetric and positive semidefinite ($\boldsymbol{Q} = \boldsymbol{Q}^T \succcurlyeq \boldsymbol{0}$) state weighting matrix
  - $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ symmetric and positive semidefinite ($\boldsymbol{R} = \boldsymbol{R}^T \succcurlyeq \boldsymbol{0}$) input weighting matrix
  - $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ symmetric and positive semidefinite ($\boldsymbol{P} = \boldsymbol{P}^T \succcurlyeq \boldsymbol{0}$) terminal weighting matrix
  - $N \geq 1$ finite prediction horizon
- **Remarks**
  - Besides quadratic cost functions also linear cost functions can be considered, cf. [Mac02, Section 5.4]
  - For linear cost functions the computation time is smaller but the behavior is different

Cost Function

- **Selection of the Weighting Matrices**
  - $\boldsymbol{Q}$ punishes the state vector $\boldsymbol{x}(k+i)$ and thus state deviations from $\boldsymbol{x}(k+i) = \boldsymbol{0}$
  - $\boldsymbol{R}$ punishes the input vector $\boldsymbol{u}(k+i)$ and thus a large control energy
  - $\boldsymbol{P}$ punishes the terminal state vector $\boldsymbol{x}(k+N)$ and thus state deviations from $\boldsymbol{x}(k+N) = \boldsymbol{0}$
  - For receding horizon control $\boldsymbol{P}$ can be selected such that the closed-loop system is stable (cf. Ch. 6)
  - For simplicity the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are often selected as diagonal matrices with diagonal elements $q_v \geq 0, v \in \{1, \dots, n\}$ and $r_w \geq 0, w \in \{1, \dots, m\}$. For selecting the diagonal elements a good guess can be based on the magnitudes of the states and inputs, i.e.

$$x_v(k+i) \in [\underline{x}_v, \overline{x}_v], v \in \{1, \dots, n\} \quad \rightarrow q_v = \frac{1}{\max\left(\underline{x}_v^2, \overline{x}_v^2\right)}$$

$$u_w(k+i) \in [\underline{u}_w, \overline{u}_w], w \in \{1, \dots, m\} \quad \rightarrow r_w = \frac{1}{\max\left(\underline{u}_w^2, \overline{u}_w^2\right)}$$

Bryson‘s rule

The diagonal elements are then fine-tuned according to the importance of the states and inputs.

Optimization Problem

Problem 4.1 For the discrete-time linear time-invariant system (4.1) and the current state $\boldsymbol{x}(k)$ find an input sequence $\boldsymbol{U}^*(k)$ such that the discrete-time quadratic cost function (4.3) is minimized, i.e.

$$\min_{\boldsymbol{U}(k)} V_N(\boldsymbol{x}(k), \boldsymbol{U}(k))$$

$$\text{subject to } \boldsymbol{x}(k+i+1) = \boldsymbol{A}\boldsymbol{x}(k+i) + \boldsymbol{B}\boldsymbol{u}(k+i), i = 0,1,\dots,N-1$$

- **Remarks**
  - Problem 4.1 can be solved in a "recursive" way using dynamic programming (cf. Optimal Control)
  - Problem 4.1 can be solved in a "batch" way using quadratic programming (considered here)
- **Solution based on Quadratic Programming**
  - Construct a prediction model describing the states over the whole prediction horizon ("batch")
  - Reformulate the cost function $V_N(\boldsymbol{x}(k), \boldsymbol{U}(k))$ in terms of $\boldsymbol{x}(k)$, $\boldsymbol{U}(k)$ using the prediction model
  - Set ${}^{\partial}/_{\partial \boldsymbol{U}(k)} V_N\big(\boldsymbol{x}(k), \boldsymbol{U}(k)\big) = \boldsymbol{0}$ and solve for $\boldsymbol{U}^*(k)$ (analytical solution possible in unconstrained case)

Construction of the Prediction Model

- **Solution of the State Equation (4.1)**

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$$

$$\boldsymbol{x}(k+2) = \boldsymbol{A}\boldsymbol{x}(k+1) + \boldsymbol{B}\boldsymbol{u}(k+1) = \boldsymbol{A}^2\boldsymbol{x}(k) + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{B}\boldsymbol{u}(k+1)$$

$$\boldsymbol{x}(k+3) = \boldsymbol{A}\boldsymbol{x}(k+2) + \boldsymbol{B}\boldsymbol{u}(k+2) = \boldsymbol{A}^3\boldsymbol{x}(k) + \boldsymbol{A}^2\boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}(k+1) + \boldsymbol{B}\boldsymbol{u}(k+2)$$

$$\vdots$$

$$\boldsymbol{x}(k+N) = \boldsymbol{A}^N\boldsymbol{x}(k) + \boldsymbol{A}^{N-1}\boldsymbol{B}\boldsymbol{u}(k) + \cdots + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}(k+N-2) + \boldsymbol{B}\boldsymbol{u}(k+N-1)$$

- **Representation in Matrix Form**

$$\underbrace{\begin{pmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{x}(k+2) \\ \vdots \\ \boldsymbol{x}(k+N) \end{pmatrix}}_{\boldsymbol{X}(k)} = \underbrace{\begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N \end{pmatrix}}_{\boldsymbol{\Phi}} \underbrace{\boldsymbol{x}(k)}_{\boldsymbol{x}(k)} + \underbrace{\begin{pmatrix} \boldsymbol{B} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \cdots & \boldsymbol{B} \end{pmatrix}}_{\boldsymbol{\Gamma}} \underbrace{\begin{pmatrix} \boldsymbol{u}(k) \\ \boldsymbol{u}(k+1) \\ \vdots \\ \boldsymbol{u}(k+N-1) \end{pmatrix}}_{\boldsymbol{U}(k)} \qquad (4.4)$$

Reformulation of the Cost Function

- **Representation in Matrix Form**

$$V_N\big(\boldsymbol{x}(k), \boldsymbol{U}(k)\big) = \boldsymbol{x}^T(k+N)\boldsymbol{P}\boldsymbol{x}(k+N) + \sum_{i=0}^{N-1} \boldsymbol{x}^T(k+i)\boldsymbol{Q}\boldsymbol{x}(k+i) + \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i) =$$

$$\underbrace{\boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) +}_{\boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k)+} \underbrace{\begin{pmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{x}(k+2) \\ \vdots \\ \boldsymbol{x}(k+N) \end{pmatrix}^T}_{\boldsymbol{X}^T(k)} \underbrace{\begin{pmatrix} \boldsymbol{Q} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{P} \end{pmatrix}}_{\boldsymbol{\Omega}} \underbrace{\begin{pmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{x}(k+2) \\ \vdots \\ \boldsymbol{x}(k+N) \end{pmatrix}}_{\boldsymbol{X}(k)} + \underbrace{\begin{pmatrix} \boldsymbol{u}(k) \\ \boldsymbol{u}(k+1) \\ \vdots \\ \boldsymbol{u}(k+N-1) \end{pmatrix}^T}_{\boldsymbol{U}^T(k)} \underbrace{\begin{pmatrix} \boldsymbol{R} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{R} \end{pmatrix}}_{\boldsymbol{\Psi}} \underbrace{\begin{pmatrix} \boldsymbol{u}(k) \\ \boldsymbol{u}(k+1) \\ \vdots \\ \boldsymbol{u}(k+N-1) \end{pmatrix}}_{\boldsymbol{U}(k)} = \tag{4.5}$$

- **Remarks**
  - Note that $\boldsymbol{P} \succcurlyeq \boldsymbol{0}$ and $\boldsymbol{Q} \succcurlyeq \boldsymbol{0}$ implies $\boldsymbol{\Omega} \succcurlyeq \boldsymbol{0}$ and furthermore $\boldsymbol{P} \succ \boldsymbol{0}$ and $\boldsymbol{Q} \succ \boldsymbol{0}$ implies $\boldsymbol{\Omega} \succ \boldsymbol{0}$
  - Note that $\boldsymbol{R} \succcurlyeq \boldsymbol{0}$ implies $\boldsymbol{\Psi} \succcurlyeq \boldsymbol{0}$ and furthermore $\boldsymbol{R} \succ \boldsymbol{0}$ implies $\boldsymbol{\Psi} \succ \boldsymbol{0}$

Reformulation of the Cost Function

- **Substitution of the Prediction Model** (4.4)

$$
\begin{aligned}
V_N\big(\boldsymbol{x}(k),\boldsymbol{U}(k)\big) &= \boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{X}^T(k)\,\boldsymbol{\Omega}\boldsymbol{X}(k) + \boldsymbol{U}^T(k)\boldsymbol{\Psi}\boldsymbol{U}(k) \\
&= \boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \big(\boldsymbol{\Phi}\boldsymbol{x}(k)+\boldsymbol{\Gamma}\boldsymbol{U}(k)\big)^T\boldsymbol{\Omega}\big(\boldsymbol{\Phi}\boldsymbol{x}(k)+\boldsymbol{\Gamma}\boldsymbol{U}(k)\big) + \boldsymbol{U}^T(k)\boldsymbol{\Psi}\boldsymbol{U}(k) \\
&= \boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{x}^T(k)\boldsymbol{\Phi}^T\boldsymbol{\Omega}\boldsymbol{\Phi}\boldsymbol{x}(k) + \boldsymbol{x}^T(k)\boldsymbol{\Phi}^T\boldsymbol{\Omega}\boldsymbol{\Gamma}\boldsymbol{U}(k) + \boldsymbol{U}^T(k)\boldsymbol{\Gamma}^T\boldsymbol{\Omega}\boldsymbol{\Phi}\boldsymbol{x}(k) \\
&\quad + \boldsymbol{U}^T(k)\boldsymbol{\Gamma}^T\boldsymbol{\Omega}\boldsymbol{\Gamma}\boldsymbol{U}(k) + \boldsymbol{U}^T(k)\boldsymbol{\Psi}\boldsymbol{U}(k) \qquad \boldsymbol{x}^T\boldsymbol{M}\boldsymbol{U} = (\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{U})^T = \boldsymbol{U}^T\boldsymbol{M}\boldsymbol{x} \text{ Scalar!} \\
&= \boldsymbol{x}^T(k)(\boldsymbol{Q}+\boldsymbol{\Phi}^T\boldsymbol{\Omega}\boldsymbol{\Phi})\boldsymbol{x}(k) + \boldsymbol{U}^T(k)(\boldsymbol{\Psi}+\boldsymbol{\Gamma}^T\boldsymbol{\Omega}\boldsymbol{\Gamma})\boldsymbol{U}(k) + 2\boldsymbol{U}^T(k)\boldsymbol{\Gamma}^T\boldsymbol{\Omega}\boldsymbol{\Phi}\boldsymbol{x}(k) \\
&= \frac{1}{2}\boldsymbol{U}^T(k)\underbrace{2(\boldsymbol{\Psi}+\boldsymbol{\Gamma}^T\boldsymbol{\Omega}\boldsymbol{\Gamma})}_{\boldsymbol{H}}\boldsymbol{U}(k) + \boldsymbol{U}^T(k)\underbrace{2\boldsymbol{\Gamma}^T\boldsymbol{\Omega}\boldsymbol{\Phi}}_{\boldsymbol{F}}\boldsymbol{x}(k) + \boldsymbol{x}^T(k)(\boldsymbol{Q}+\boldsymbol{\Phi}^T\boldsymbol{\Omega}\boldsymbol{\Phi})\boldsymbol{x}(k) \\
&= \frac{1}{2}\boldsymbol{U}^T(k)\,\boldsymbol{H}\,\boldsymbol{U}(k) + \boldsymbol{U}^T(k)\,\boldsymbol{F}\,\boldsymbol{x}(k) + \boldsymbol{x}^T(k)(\boldsymbol{Q}+\boldsymbol{\Phi}^T\boldsymbol{\Omega}\boldsymbol{\Phi})\boldsymbol{x}(k)
\end{aligned} \tag{4.6}
$$

- **Remarks**
  - Note that $\boldsymbol{\Psi} \succcurlyeq \boldsymbol{0}$ and $\boldsymbol{\Omega} \succcurlyeq \boldsymbol{0}$ implies $\boldsymbol{H} \succcurlyeq \boldsymbol{0}$. Then $V_N\big(\boldsymbol{x}(k),\boldsymbol{U}(k)\big)$ is convex.
  - Note that $\boldsymbol{\Psi} \succ \boldsymbol{0}$ and $\boldsymbol{\Omega} \succ \boldsymbol{0}$ implies $\boldsymbol{H} \succ \boldsymbol{0}$. Then $V_N\big(\boldsymbol{x}(k),\boldsymbol{U}(k)\big)$ is strictly convex.

Analytical Solution

- **Determination of the Derivative**

$$\frac{\partial}{\partial \boldsymbol{U}(k)} V_N\big(\boldsymbol{x}(k), \boldsymbol{U}(k)\big) = \frac{\partial}{\partial \boldsymbol{U}(k)} \left(\frac{1}{2} \boldsymbol{U}^T(k) \boldsymbol{H} \boldsymbol{U}(k) + \boldsymbol{U}^T(k) \boldsymbol{F} \boldsymbol{x}(k) + \boldsymbol{x}^T(k)(\boldsymbol{Q} + \boldsymbol{\Phi}^T \boldsymbol{\Omega} \boldsymbol{\Phi}) \boldsymbol{x}(k)\right)$$

$$= \boldsymbol{H}\boldsymbol{U}(k) + \boldsymbol{F}\boldsymbol{x}(k)$$

$$\stackrel{!}{=} \boldsymbol{0}$$

$\frac{\partial}{\partial \boldsymbol{U}} \boldsymbol{U}^T \boldsymbol{M} \boldsymbol{U} = 2\boldsymbol{M}\boldsymbol{U}, \frac{\partial}{\partial \boldsymbol{U}} \boldsymbol{U}^T \boldsymbol{M} \boldsymbol{x} = \boldsymbol{M}\boldsymbol{x}$

- **Optimal State Feedback Control Law**

$$\boldsymbol{U}^*(k) = -\boldsymbol{H}^{-1} \boldsymbol{F} \boldsymbol{x}(k)$$

- **Remarks**
  - Note that $\boldsymbol{\Phi} \in \mathbb{R}^{Nn \times n}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{Nn \times Nm}$, $\boldsymbol{\Omega} \in \mathbb{R}^{Nn \times Nn}$, $\boldsymbol{\Psi} \in \mathbb{R}^{Nm \times Nm}$, $\boldsymbol{H} \in \mathbb{R}^{Nm \times Nm}$ and $\mathbf{F} \in \mathbb{R}^{Nm \times n}$
  - $\boldsymbol{H} = 2(\boldsymbol{\Psi} + \boldsymbol{\Gamma}^T \boldsymbol{\Omega} \boldsymbol{\Gamma})$ is invertible if $\boldsymbol{R} \succ \boldsymbol{0}$ (then $\boldsymbol{\Psi} \succ \boldsymbol{0}$) or $\boldsymbol{P} \succ \boldsymbol{0}$, $\boldsymbol{Q} \succ \boldsymbol{0}$, $\boldsymbol{\Gamma}$ full rank (then $\boldsymbol{\Gamma}^T \boldsymbol{\Omega} \boldsymbol{\Gamma} \succ \boldsymbol{0}$)
  - $\boldsymbol{\Gamma}$ full rank is guaranteed if $(\boldsymbol{A}, \boldsymbol{B})$ is controllable

Illustrative Example

- **System Model**

$$\boldsymbol{A} = \begin{pmatrix} 1.1 & 2 \\ 0 & 0.95 \end{pmatrix},\ \boldsymbol{B} = \begin{pmatrix} 0 \\ 0.0787 \end{pmatrix},\ \text{unstable due to } \rho(\boldsymbol{A}) = 1.1 > 1, \text{ controllable}$$

- **Cost Function**

$$\boldsymbol{Q} = \boldsymbol{P} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \succcurlyeq \boldsymbol{0},\ R = 0.01 \succ 0,\ N = 4$$

- **Construction of the Prediction Model**

$$\boldsymbol{\Phi} = \begin{pmatrix} 1.1 & 2 \\ 0 & 0.95 \\ 1.21 & 4.1 \\ 0 & 0.9025 \\ 1.331 & 6.315 \\ 0 & 0.8574 \\ 1.4641 & 8.6612 \\ 0 & 0.8145 \end{pmatrix},\ \boldsymbol{\Gamma} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.0787 & 0 & 0 & 0 \\ 0.1574 & 0 & 0 & 0 \\ 0.0748 & 0.0787 & 0 & 0 \\ 0.3227 & 0.1574 & 0 & 0 \\ 0.0710 & 0.0748 & 0.0787 & 0 \\ 0.4970 & 0.3227 & 0.1574 & 0 \\ 0.0675 & 0.0710 & 0.0748 & 0.0787 \end{pmatrix}$$

Illustrative Example

- **Reformulation of the Cost Function**

$$\boldsymbol{\Omega} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \quad \boldsymbol{\Psi} = \begin{pmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}$$

$$\boldsymbol{H} = \begin{pmatrix} 0.5417 & 0.2448 & 0.0314 & -0.0676 \\ 0.2448 & 0.1727 & 0.0286 & -0.0396 \\ 0.0314 & 0.0286 & 0.0460 & -0.0130 \\ -0.0676 & -0.0396 & -0.0130 & 0.0324 \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} 1.9544 & 9.8505 \\ 0.7664 & 4.3479 \\ 0.0325 & 0.4378 \\ -0.2304 & -1.2351 \end{pmatrix}$$

Finite Horizon Control

Illustrative Example

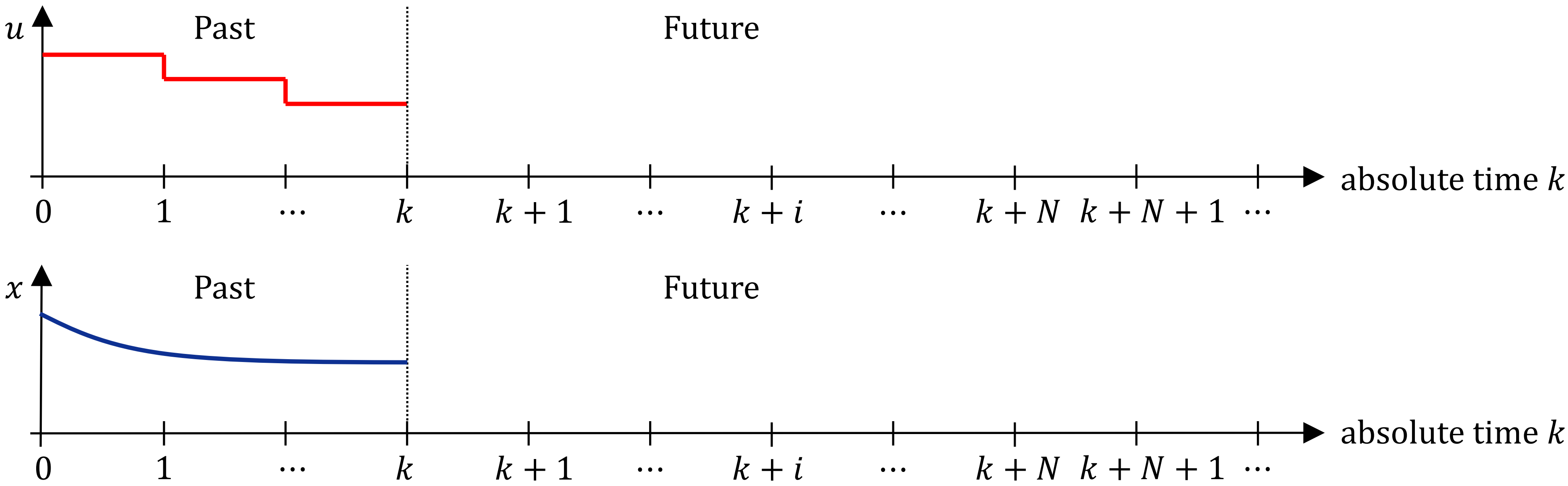
- **Optimal State Feedback Control Law**

$$\boldsymbol{U}^*(k) = -\boldsymbol{H}^{-1}\boldsymbol{F}\boldsymbol{x}(k) = -\begin{pmatrix} 4.3563 & 18.6889 \\ -1.6383 & -1.2379 \\ -1.4141 & -2.9767 \\ -0.5935 & -1.8326 \end{pmatrix}\boldsymbol{x}(k)$$

Conclusions

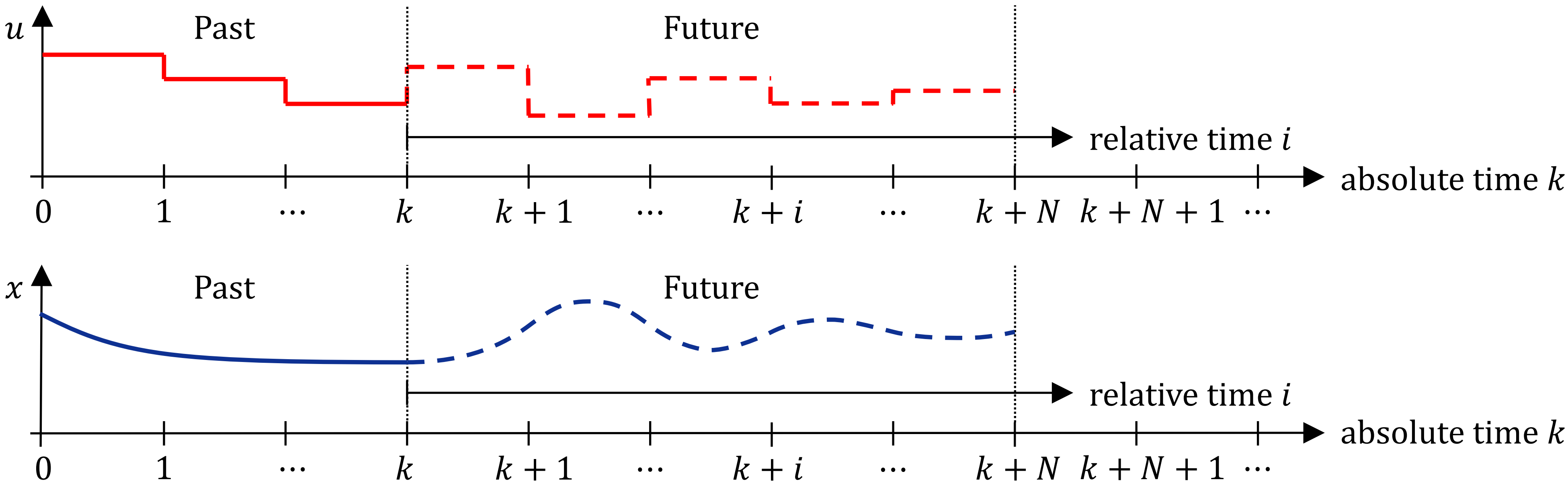
- **Finite Horizon Control**
  - Appropriate for control problems with finite time (e.g. many motion control problems)
  - Inappropriate for control problems with infinite time (e.g. temperature control problems)
- **Infinite Horizon Control**
  - Feasible for LTI systems without constraints (cf. Slide 4-28f)
  - Infeasible for LTI systems with constraints, uncertain systems, hybrid systems, nonlinear systems, …
    Note that there are some exceptions, see e.g. [BMD+02] and [BBM15, Section 12.3]!

Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k)$

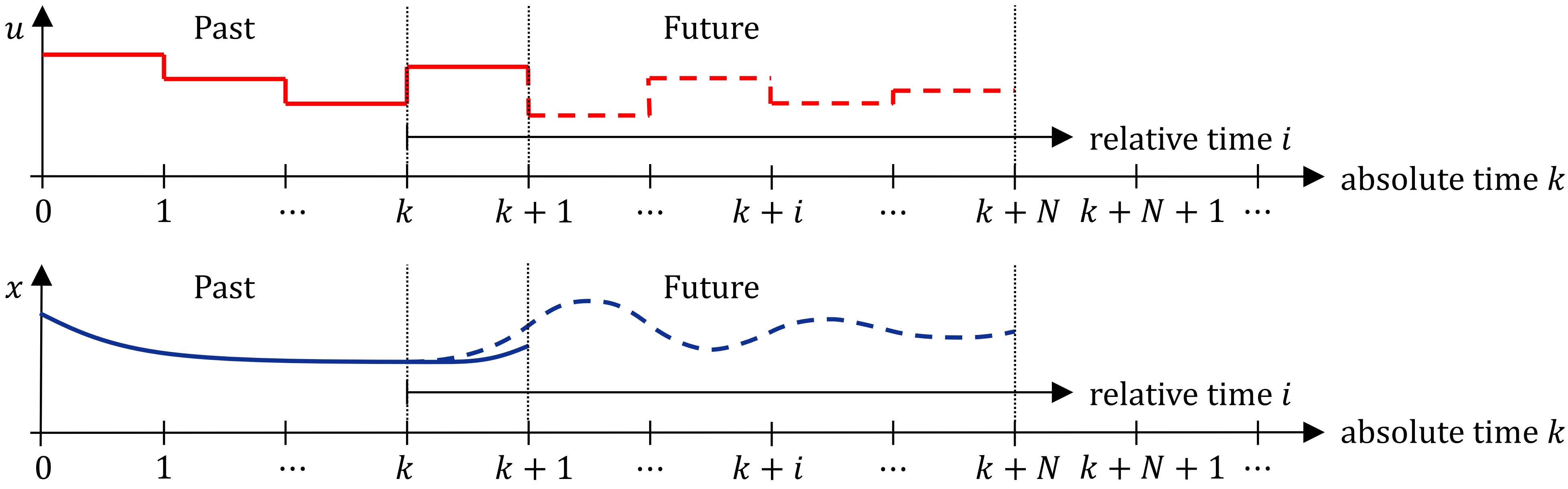
Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k)$
2. Solve Problem 4.1 to determine the optimal input sequence $\boldsymbol{U}^*(k)$

Receding Horizon Control

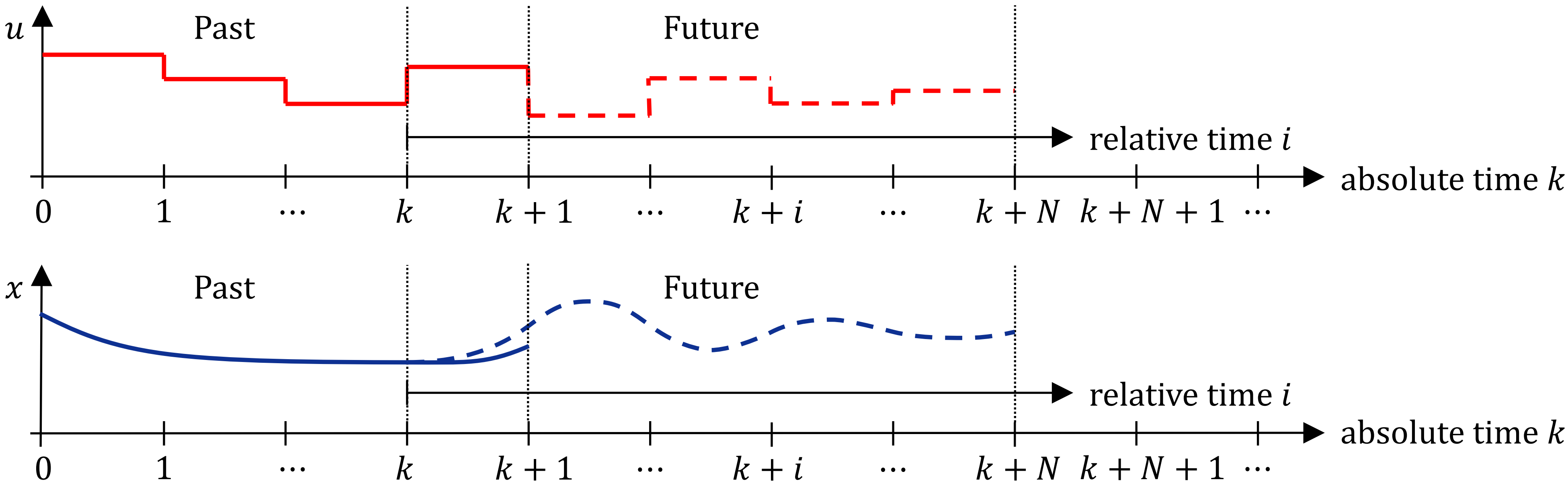
Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k)$
2. Solve Problem 4.1 to determine the optimal input sequence $\boldsymbol{U}^*(k)$
3. Implement the first element of input sequence $\boldsymbol{u}^*(k) = \underbrace{(\boldsymbol{I}_{m\times m} \quad \boldsymbol{0}_{m\times m} \quad \cdots \quad \boldsymbol{0}_{m\times m})}_{\text{“masking” matrix}}\boldsymbol{U}^*(k)$

Receding Horizon Control

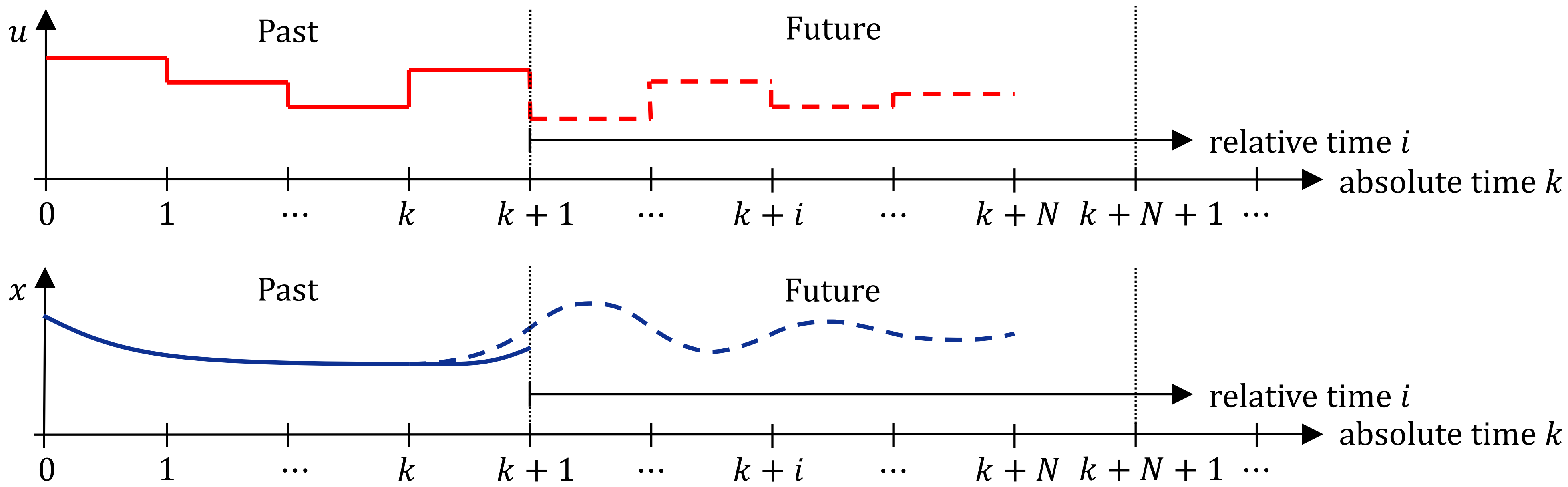
Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k)$
2. Solve Problem 4.1 to determine the optimal input sequence $\boldsymbol{U}^*(k)$
3. Implement the first element of input sequence $\boldsymbol{u}^*(k) = \underbrace{(\boldsymbol{I}_{m\times m} \quad \boldsymbol{0}_{m\times m} \quad \cdots \quad \boldsymbol{0}_{m\times m})}_{\text{“masking” matrix}}\boldsymbol{U}^*(k)$
4. Increment the time instant $k \coloneqq k+1$ and go to 1.

Receding Horizon Control

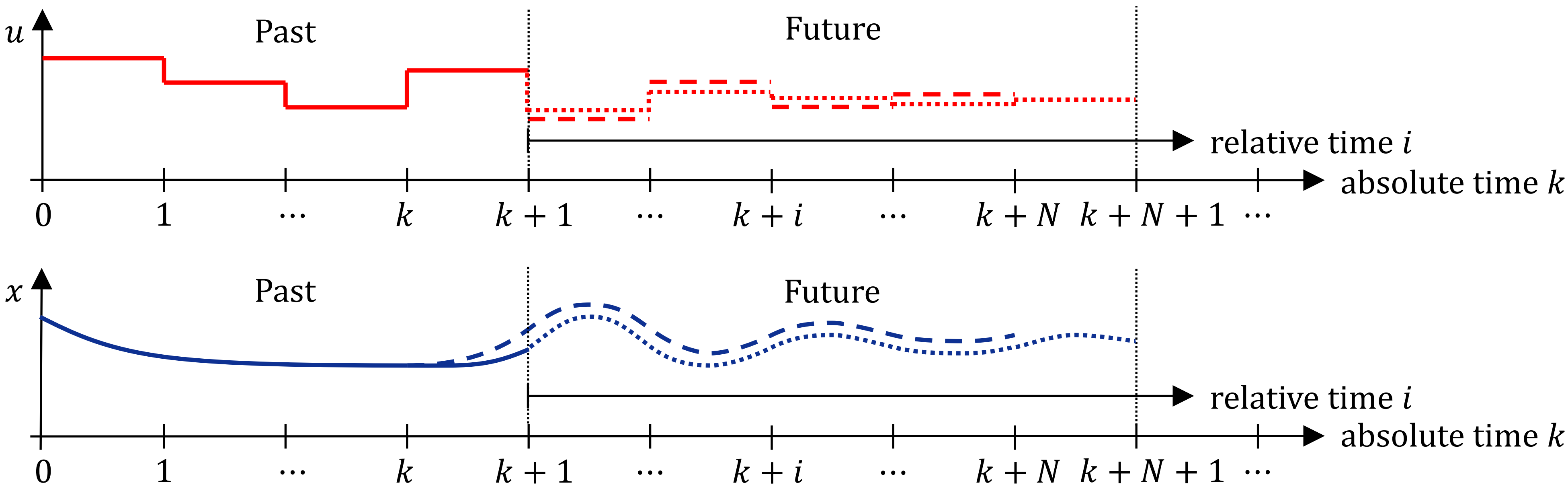
Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k+1)$

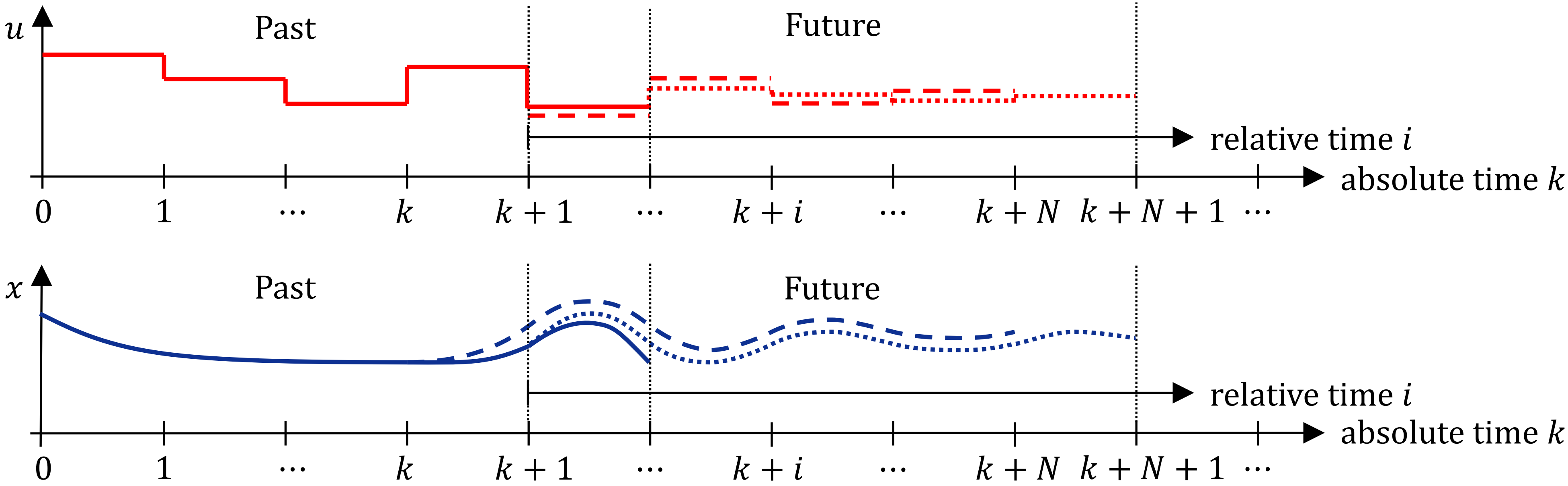
Receding Horizon Control

Receding Horizon Principle



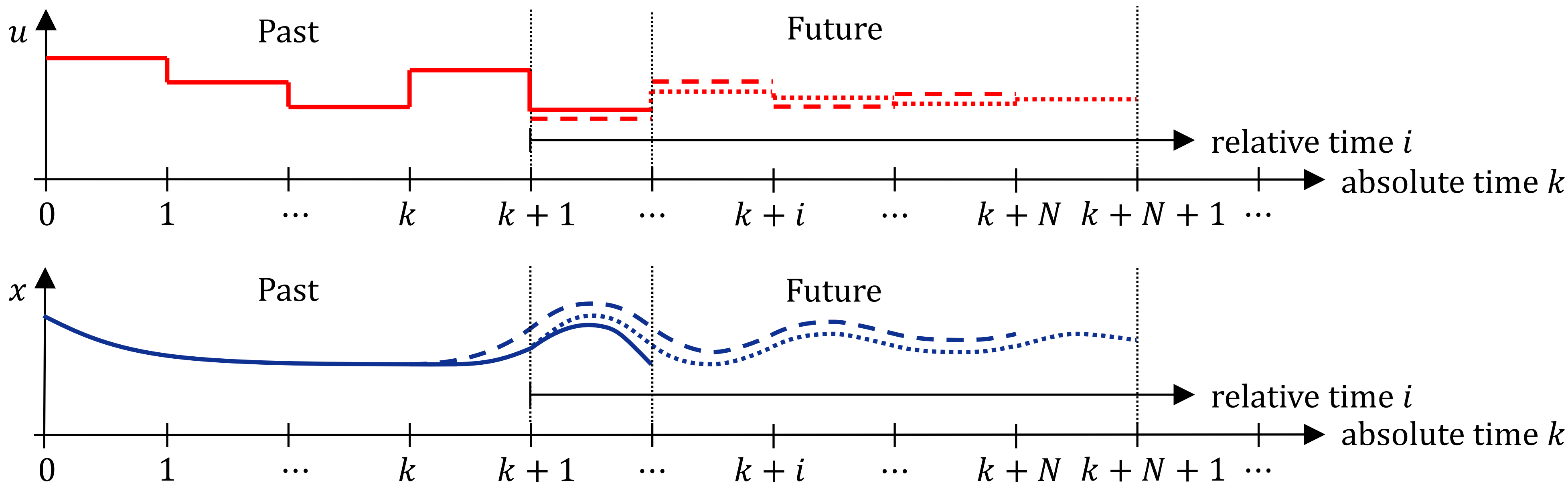
1. Measure the current state $\boldsymbol{x}(k+1)$
2. Solve Problem 4.1 to determine the optimal input sequence $\boldsymbol{U}^*(k+1)$

Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k+1)$
2. Solve Problem 4.1 to determine the optimal input sequence $\boldsymbol{U}^*(k+1)$
3. Implement the first element of input sequence $\boldsymbol{u}^*(k+1) = \underbrace{(\boldsymbol{I}_{m\times m} \quad \boldsymbol{0}_{m\times m} \quad \cdots \quad \boldsymbol{0}_{m\times m})}_{\text{“masking” matrix}} \boldsymbol{U}^*(k+1)$

Receding Horizon Principle



1. Measure the current state $\boldsymbol{x}(k+1)$
2. Solve Problem 4.1 to determine the optimal input sequence $\boldsymbol{U}^*(k+1)$
3. Implement the first element of input sequence $\boldsymbol{u}^*(k+1) = \underbrace{(\boldsymbol{I}_{m\times m} \quad \boldsymbol{0}_{m\times m} \quad \cdots \quad \boldsymbol{0}_{m\times m})}_{\text{“masking” matrix}} \boldsymbol{U}^*(k+1)$
4. Increment the time instant $k+1 \coloneqq k+2$ and go to 1.

Receding Horizon Controller

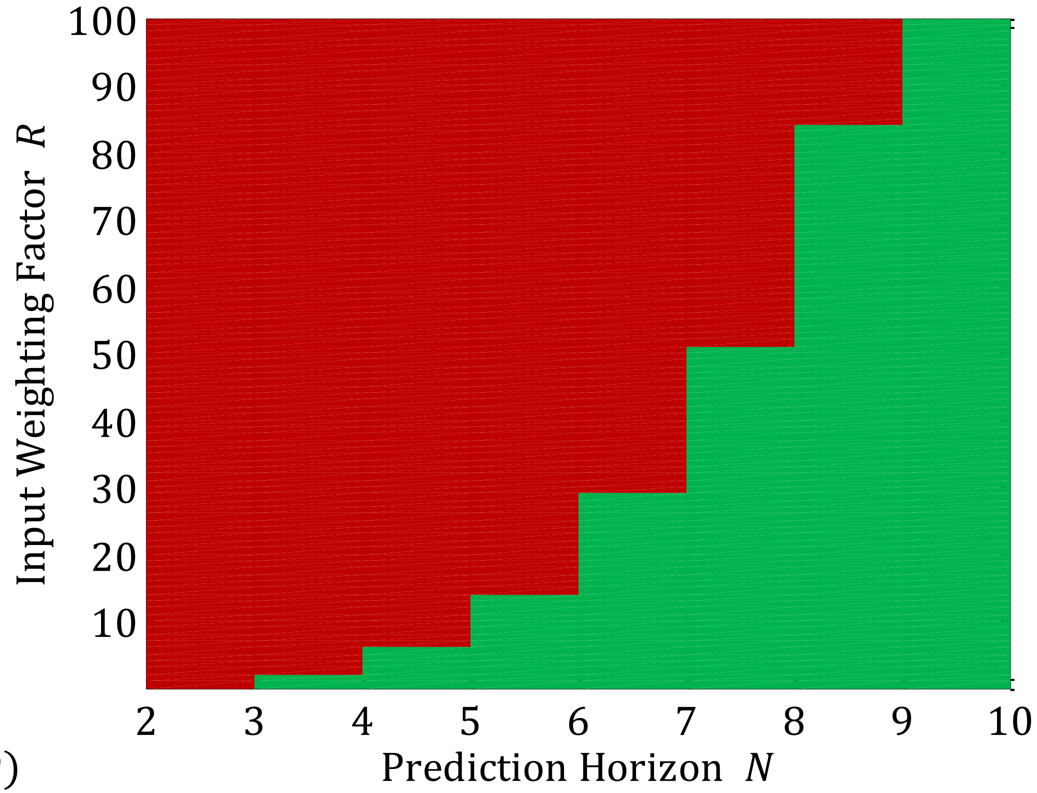
- **Observations**
  - Problem 4.1 only depends on the current state $\boldsymbol{x}(k)$ but not on the time instant k
  - Problem 4.1 is therefore time-invariant
  - The matrices $\boldsymbol{H}$ and $\boldsymbol{F}$ characterizing the solution of Problem 4.1 are therefore also time-invariant
- **Optimal State Feedback Control Law**

$$\begin{aligned}\boldsymbol{u}^*(k) &= (\boldsymbol{I}_{m\times m} \quad \boldsymbol{0}_{m\times m} \quad \cdots \quad \boldsymbol{0}_{m\times m})\boldsymbol{U}^*(k) \\ &= \underbrace{-(\boldsymbol{I}_{m\times m} \quad \boldsymbol{0}_{m\times m} \quad \cdots \quad \boldsymbol{0}_{m\times m})\boldsymbol{H}^{-1}\boldsymbol{F}}_{\boldsymbol{K}_{\mathrm{RHC}}}\boldsymbol{x}(k)\end{aligned}$$

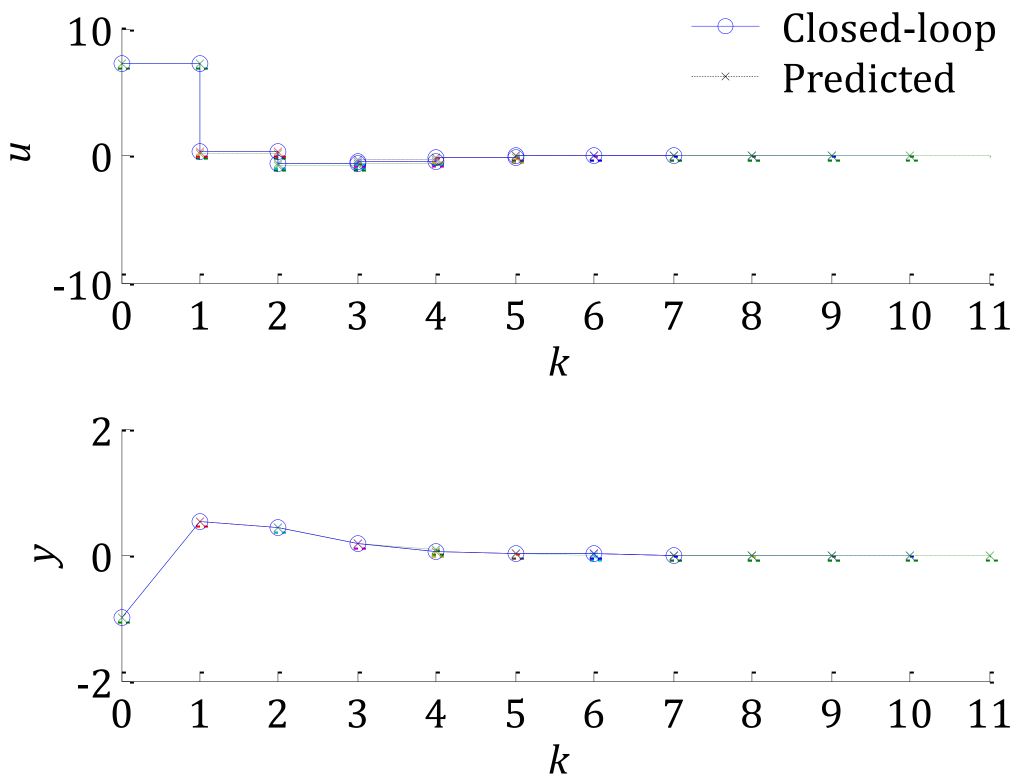
- **Remarks**
  - A receding horizon controller is an LTI state feedback controller in the unconstrained case
  - The feedback matrix $\boldsymbol{K}_{\mathrm{RHC}}$ can be calculated offline in the unconstrained case
  - The closed-loop system is globally asymptotically stable iff $\rho(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_{\mathrm{RHC}}) < 1$ (cf. Theorem 2.3)

Illustrative Example (Cont‘d)

- **Stability Analysis**
  - Vary $N = 2,3,\dots,10$
  - Vary $R = 0.01,1,2,\dots,100$
  - Check $\rho(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_{\mathrm{RHC}}) \overset{?}{<} 1$
  - Green ≙ stable
  - Red ≙ unstable
- **Observations**
  - The larger N, the more likely stability
  - The smaller R, the more likely stability
- **Conclusions**
  - Stability and performance are affected by the parameters N and $\boldsymbol{R}$ (and $\boldsymbol{P}$ and $\boldsymbol{Q}$)



Illustrative Example (Cont‘d)



Prediction horizon $N = 4$

$\boldsymbol{x}(0) = (0.5 \quad -0.5)^T$

$y(k) = (-1 \quad 1)\boldsymbol{x}(k)$

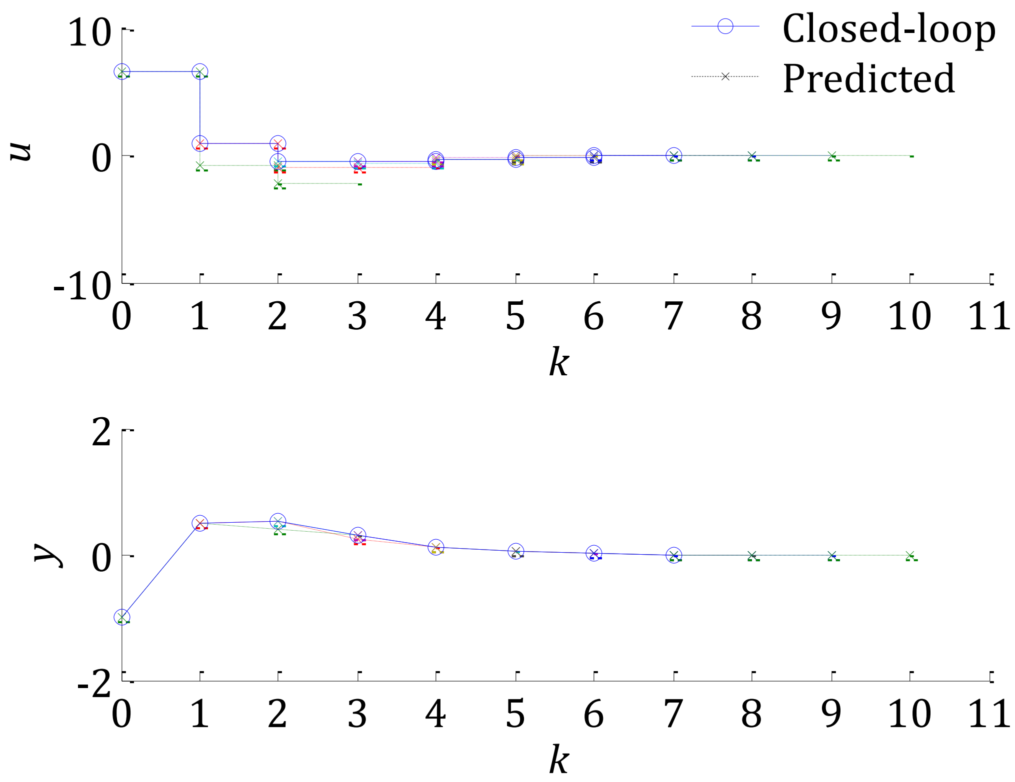
$R = 0.01$

Closed-loop system stable

Good prediction accuracy

Good performance

Illustrative Example (Cont'd)



Prediction horizon $N = 3$

$\boldsymbol{x}(0) = (0.5 \quad -0.5)^T$

$y(k) = (-1 \quad 1)\boldsymbol{x}(k)$

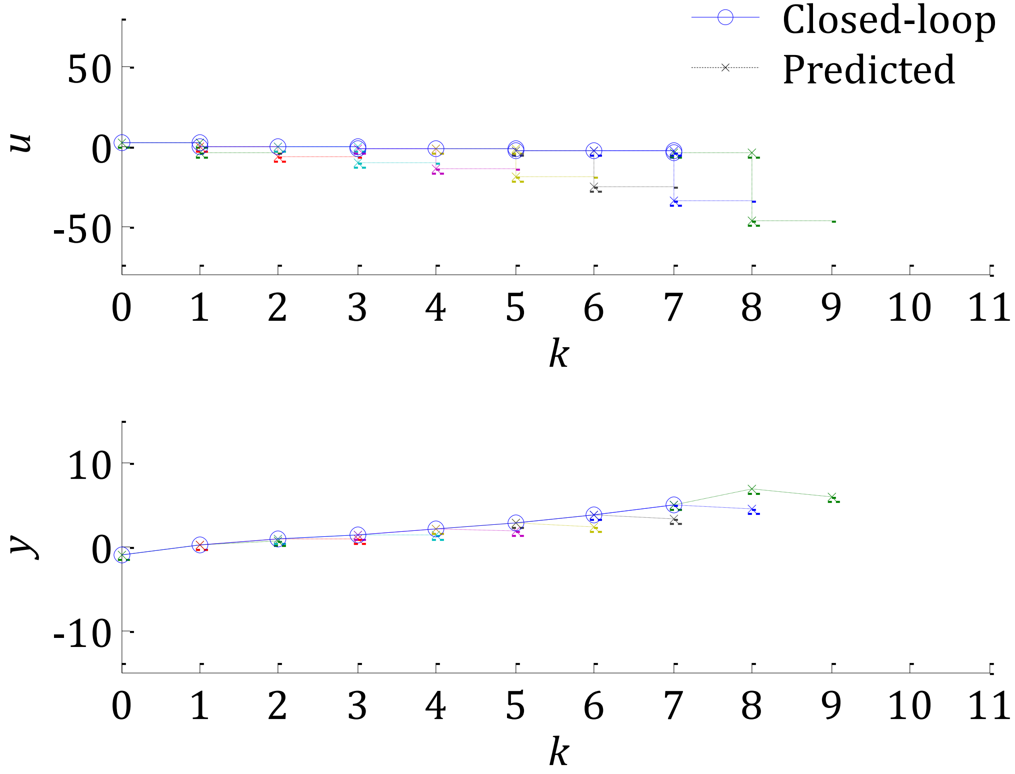
$R = 0.01$

Closed-loop system stable

Poor prediction accuracy

Poor performance

Illustrative Example (Cont'd)



Prediction horizon $N = 2$

$\boldsymbol{x}(0) = (0.5 \quad -0.5)^T$

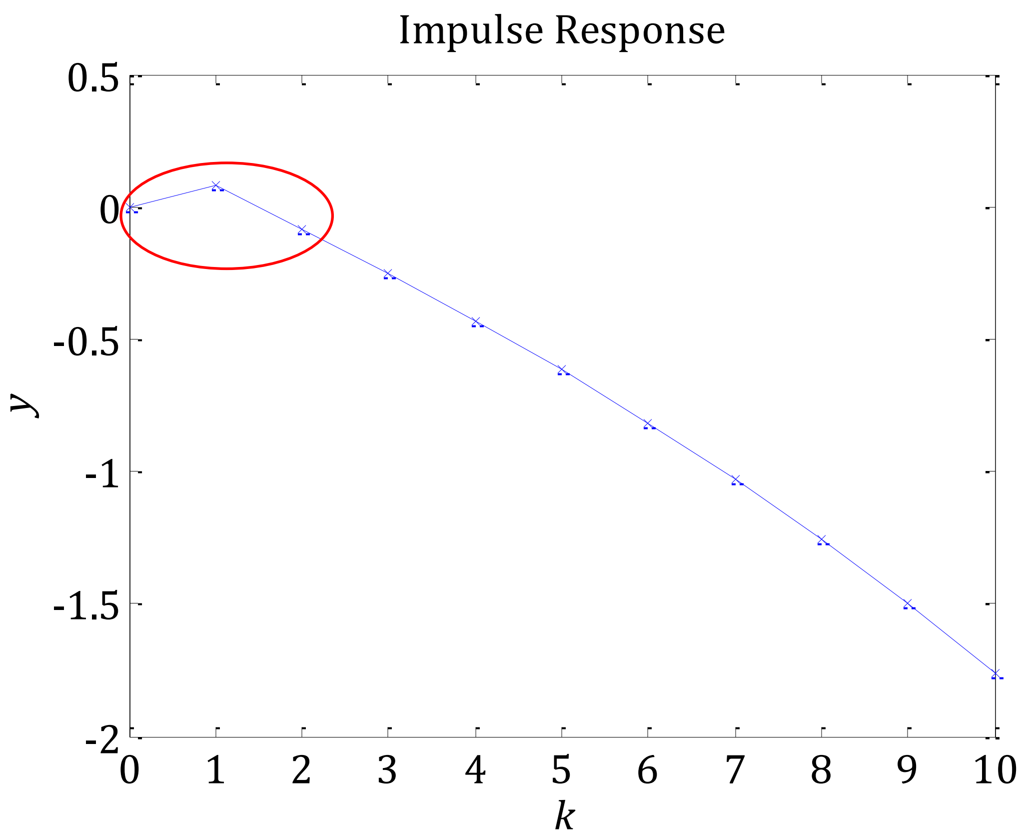
$y(k) = (-1 \quad 1)\boldsymbol{x}(k)$

$R = 0.01$

Closed-loop system unstable

Very poor prediction accuracy

Illustrative Example (Cont‘d)



Interpretation

Non-minimum phase system (due to zero at $z = 3.1$)

The control demand is under-estimated for a small N

Conclusions

N must be sufficiently large to capture the relevant dynamics

N should ideally approach ∞

Can $N \to \infty$ be realized?

Optimization Problem

Tutorial

Problem 4.2 For the discrete-time linear time-invariant system (4.1) and the current state $\boldsymbol{x}(k)$ find an input sequence $\boldsymbol{U}^*(k)$ such that the discrete-time quadratic cost function (4.3) for $N \rightarrow \infty$ is minimized, i.e.

$$\min_{\boldsymbol{U}(k)} V_\infty(\boldsymbol{x}(k), \boldsymbol{U}(k))$$

$$\text{subject to } \boldsymbol{x}(k+i+1) = \boldsymbol{A}\boldsymbol{x}(k+i) + \boldsymbol{B}\boldsymbol{u}(k+i), i = 0,1, \dots$$

- **Remarks**
  - Problem 4.2 can be solved based on linear-quadratic control theory
  - The resulting controller is denoted as linear-quadratic regulator (LQR)
  - For a detailed discussion on linear-quadratic control theory see Optimal Control
- **Assumptions**
  - $(\boldsymbol{Q}^{1/2}, \boldsymbol{A})$ is detectable → state vector must be “detectable” through the cost function
  - $\boldsymbol{R} \succ \boldsymbol{0}$ → to ensure invertibility later on

Solution based on Linear-Quadratic Control Theory

Tutorial

- **Algebraic Riccati Equation (ARE)**

$$(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_{\mathrm{LQR}})^T\boldsymbol{P}_{\mathrm{LQR}}(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_{\mathrm{LQR}})-\boldsymbol{P}_{\mathrm{LQR}}+\boldsymbol{Q}+\boldsymbol{K}_{\mathrm{LQR}}^T\boldsymbol{R}\boldsymbol{K}_{\mathrm{LQR}}=\boldsymbol{0} \tag{4.7}$$

- **Optimal State Feedback Control Law**

$$\begin{aligned}\boldsymbol{u}^*(k) &= \underbrace{(\boldsymbol{R}+\boldsymbol{B}^T\boldsymbol{P}_{\mathrm{LQR}}\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{P}_{\mathrm{LQR}}}_{\boldsymbol{K}_{\mathrm{LQR}}}\boldsymbol{x}(k)\\ &= \boldsymbol{K}_{\mathrm{LQR}}\boldsymbol{x}(k)\end{aligned}$$

where $\boldsymbol{P}_{\mathrm{LQR}}$ is the pos. semidefinite solution of the ARE

- **Minimum Cost**

$$V_\infty^*(\boldsymbol{x}(k)) = \boldsymbol{x}^T(k)\boldsymbol{P}_{\mathrm{LQR}}\boldsymbol{x}(k)$$

where $\boldsymbol{P}_{\mathrm{LQR}}$ is the pos. semidefinite solution of the ARE

- **Remarks**
  - A linear-quadratic regulator is an LTI state feedback controller
  - The feedback matrix $\boldsymbol{K}_{\mathrm{LQR}}$ can be calculated offline (MATLAB `[`$\boldsymbol{K}_{\mathrm{LQR}}$`,`$\boldsymbol{P}_{\mathrm{LQR}}$`,~]=dlqr(`$\boldsymbol{A}$`,`$\boldsymbol{B}$`,`$\boldsymbol{Q}$`,`$\boldsymbol{R}$`)`)
  - The closed-loop system is always globally asymptotically stable

Relationship between RHC and LQR

- **Motivation**
  - An infinite horizon is desirable to ensure stability and improve performance of RHC
  - A solution in a „batch“ way usually used for RHC is only possible for a finite horizon
- **Approach**
  - Consider an infinite prediction horizon but only a finite input sequence subject to optimization
  - Use a dual mode control law for this purpose

$$\boldsymbol{u}(k+i) = \begin{cases} \boldsymbol{u}^*(k+i) \quad \text{for } \; i = 0,1,\dots,N-1 & \text{mode 1 (optimal control law)} \\ \tilde{\boldsymbol{K}}\boldsymbol{x}(k+i) \quad \text{for } \; i = N, N+1, \dots & \text{mode 2 (stabilizing control law)} \end{cases}$$

  - Partition the cost function

$$V_\infty\big(\boldsymbol{x}(k)\big) = \sum_{i=0}^{N-1} [\boldsymbol{x}^T(k+i)\boldsymbol{Q}\boldsymbol{x}(k+i) + \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i)] + V_\infty\big(\boldsymbol{x}(k+N)\big)$$

How to determine?
Lyapunov equation!

Relationship between RHC and LQR

Tutorial

Theorem 4.1 For the discrete-time linear time-invariant system (4.1) under the stabilizing control law $\boldsymbol{u}(k+i) = \widetilde{\boldsymbol{K}}\boldsymbol{x}(k+i)$ the discrete-time quadratic cost (4.3) for $N \to \infty$ and current state $\boldsymbol{x}(k)$ is given by

$$V_\infty\big(\boldsymbol{x}(k)\big) = \boldsymbol{x}^T(k)\widetilde{\boldsymbol{P}}\boldsymbol{x}(k)$$

where $\widetilde{\boldsymbol{P}}$ is the positive definite solution of the discrete-time Lyapunov equation (DLE)

$$\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{P}}\widetilde{\boldsymbol{A}} - \widetilde{\boldsymbol{P}} = -\widetilde{\boldsymbol{Q}} \qquad \text{with } \widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{B}\widetilde{\boldsymbol{K}} \text{ and } \widetilde{\boldsymbol{Q}} = \boldsymbol{Q} + \widetilde{\boldsymbol{K}}^T\boldsymbol{R}\widetilde{\boldsymbol{K}} \tag{4.8}$$

- **Proof**
  - Pre- and post-multiplying (4.8) by $\boldsymbol{x}^T(k+i)$ and $\boldsymbol{x}(k+i)$ leads to

    $$\boldsymbol{x}^T(k+i)\big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big)^T\widetilde{\boldsymbol{P}}\big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big)\boldsymbol{x}(k+i) - \boldsymbol{x}^T(k+i)\widetilde{\boldsymbol{P}}\boldsymbol{x}(k+i) = -\boldsymbol{x}^T(k+i)\boldsymbol{Q}x(k+i) - \boldsymbol{x}^T(k+i)\widetilde{\boldsymbol{K}}^T\boldsymbol{R}\widetilde{\boldsymbol{K}}\boldsymbol{x}(k+i)$$

  - Defining $V_\infty\big(\boldsymbol{x}(k+i)\big) = \boldsymbol{x}^T(k+i)\widetilde{\boldsymbol{P}}\boldsymbol{x}(k+i)$ and utilizing $\boldsymbol{u}(k+i) = \widetilde{\boldsymbol{K}}\boldsymbol{x}(k+i)$ and $\boldsymbol{x}(k+i+1) = \big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big)\boldsymbol{x}(k+i)$ yields

    $$V_\infty\big(\boldsymbol{x}(k+i+1)\big) - V_\infty\big(\boldsymbol{x}(k+i)\big) = -\boldsymbol{x}^T(k+i)\boldsymbol{Q}x(k+i) - \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i)$$

Relationship between RHC and LQR

Tutorial

- **Proof**
  - Summing over $i = 0$ to $i = \infty$ results in

$$\cancel{V_\infty\big(\boldsymbol{x}(k+1)\big)} - V_\infty\big(\boldsymbol{x}(k)\big) + \cancel{V_\infty\big(\boldsymbol{x}(k+2)\big)} - \cancel{V_\infty\big(\boldsymbol{x}(k+1)\big)} + \cdots = -\sum_{i=0}^{\infty} \boldsymbol{x}^T(k+i)\boldsymbol{Q}\boldsymbol{x}(k+i) + \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i)$$

  - Using that $V_\infty\big(\boldsymbol{x}(k+i)\big) = \boldsymbol{x}^T(k)\big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big)^{T^i}\widetilde{\boldsymbol{P}}\big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big)^i\boldsymbol{x}(k) \to 0$ for $i \to \infty$ due to the assumption of a stabilizing control law (i.e. $\rho\big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big) < 1$) finally leads to

$$V_\infty\big(\boldsymbol{x}(k)\big) = \sum_{i=0}^{\infty}[\boldsymbol{x}^T(k+i)\boldsymbol{Q}\boldsymbol{x}(k+i) + \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i)]$$

- **Remarks**
  - The discrete-time Lyapunov equation (4.8) has a unique solution $\widetilde{\boldsymbol{P}}$ iff $\rho\big(\boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big) < 1$
  - $\widetilde{\boldsymbol{P}} \succ \boldsymbol{0}$ if either $\boldsymbol{Q} + \widetilde{\boldsymbol{K}}^T\boldsymbol{R}\widetilde{\boldsymbol{K}} \succ \boldsymbol{0}$ or $\boldsymbol{Q}$ is chosen such that $\big(\boldsymbol{Q}^{1/2}, \boldsymbol{A}+\boldsymbol{B}\widetilde{\boldsymbol{K}}\big)$ is observable

Relationship between RHC and LQR

- **Approach (Cont'd)**
  - Rewrite the cost function using $V_\infty\big(\boldsymbol{x}(k+N)\big) = \boldsymbol{x}^T(k+N)\widetilde{\boldsymbol{P}}\boldsymbol{x}(k+N)$ as

$$V_\infty\big(\boldsymbol{x}(k)\big) = \sum_{i=0}^{N-1} [\boldsymbol{x}^T(k+i)\boldsymbol{Q}\boldsymbol{x}(k+i) + \boldsymbol{u}^T(k+i)\boldsymbol{R}\boldsymbol{u}(k+i)] + \boldsymbol{x}^T(k+N)\widetilde{\boldsymbol{P}}\boldsymbol{x}(k+N)$$

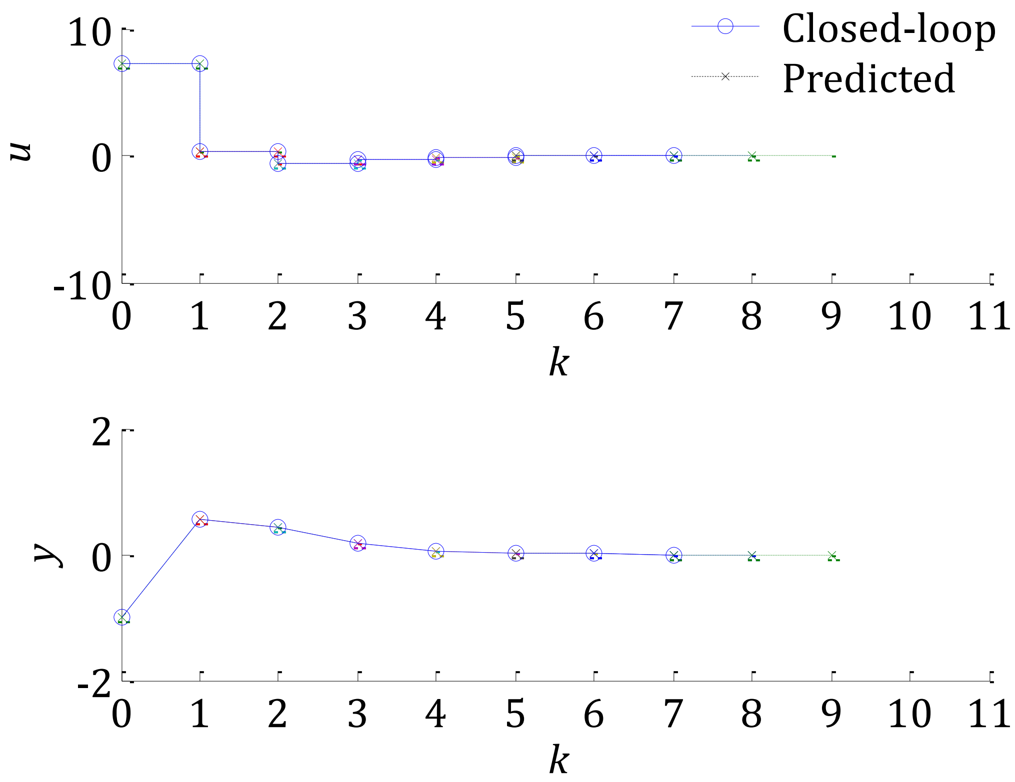
  - Solve Problem 4.1 for a finite prediction horizon N with the terminal weighting matrix $\boldsymbol{P} = \widetilde{\boldsymbol{P}}$
- **Conclusion**
  - An infinite prediction horizon can be „emulated“ by selecting the terminal weighting matrix $\boldsymbol{P}$ as the solution $\widetilde{\boldsymbol{P}}$ of the Lyapunov equation (4.8)
- **Observation**
  - The Lyapunov equation (4.8) corresponds to the Riccati equation (4.7) for $\widetilde{\boldsymbol{K}} = \boldsymbol{K}_{\mathrm{LQR}}$
  - Then also $\widetilde{\boldsymbol{P}} = \boldsymbol{P}_{\mathrm{LQR}}$ holds

Relationship between RHC and LQR

Theorem 4.2 If $\boldsymbol{P} = \boldsymbol{P}_{\mathrm{LQR}}$ in Problem 4.1, then $\boldsymbol{K}_{\mathrm{RHC}} = \boldsymbol{K}_{\mathrm{LQR}}$.

- **Proof**
  - The proof follows immediately from the discussion on the previous slides
  - Optimality is given for both mode 1 and mode 2 if $\boldsymbol{P} = \boldsymbol{P}_{\mathrm{LQR}}$
  - Optimality then overall follows from Bellman‘s principle of optimality
- **Remarks**
  - The closed-loop and predicted state and input sequences are identical for $\boldsymbol{P} = \boldsymbol{P}_{\mathrm{LQR}}$ and arbitrary N
  - RHC for $\boldsymbol{P} = \boldsymbol{P}_{\mathrm{LQR}}$ essentially provides a method for determining an LQR in a „batch“ way
- **Conclusion**
  - For LTI systems without constraints an LQR is the method of choice

Illustrative Example (Cont‘d)



Prediction horizon $N = 2$

Terminal weight $P = P_{\mathrm{LQR}}$

$\boldsymbol{x}(0) = (0.5 \quad -0.5)^T$

$y(k) = (-1 \quad 1)\boldsymbol{x}(k)$

$R = 0.01$

Closed-loop system stable

Perfect prediction accuracy

Optimal performance

Literature

Miscellaneous

[BMD+02] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3-20, 2002.