CHALMERS
UNIVERSITY OF TECHNOLOGY

SSY281 - MODEL PREDICTIVE CONTROL

NIKOLCE MURGOVSKI

Division of Systems and Control
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden

2021-03-04

Lecture 14: Beyond linear MPC

Goals for today:

- To understand some of the ideas used to extend MPC to the nonlinear case
- To understand what is meant by robust MPC

Lecture 14: Beyond linear MPC

Goals for today:

- To understand some of the ideas used to extend MPC to the nonlinear case
- To understand what is meant by robust MPC

Learning objectives:

- Understand and explain the basic principles of model predictive control, its pros and cons, and the challenges met in implementation and applications

Linear time-invariant MPC

At every time instant k, solve the constrained optimisation problem

$$\underset{\delta u_k, \delta x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix}^\top W \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix} \tag{117a}$$

$$\text{subject to} \quad \delta x_k(0) = \hat{x}(k) - x_k^r(0) \tag{117b}$$

$$\delta x_k(i+1) = A\,\delta x_k(i) + B\,\delta u_k(i); \quad i = 0, \dots, N-1 \tag{117c}$$

$$F\,\delta u_k(i) + G\,\delta x_k(i) \leq h; \quad i = 0, \dots, N-1. \tag{117d}$$

Linear time-invariant MPC

At every time instant k, solve the constrained optimisation problem

$$\underset{\delta u_k, \delta x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix}^\top W \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix} \tag{117a}$$

$$\text{subject to} \quad \delta x_k(0) = \hat{x}(k) - x_k^r(0) \tag{117b}$$

$$\delta x_k(i+1) = A\,\delta x_k(i) + B\,\delta u_k(i); \quad i = 0, \dots, N-1 \tag{117c}$$

$$F\,\delta u_k(i) + G\,\delta x_k(i) \leq h; \quad i = 0, \dots, N-1. \tag{117d}$$

The solution is written as

$$(\boldsymbol{\delta u}_k, \boldsymbol{\delta x}_k) = \mathsf{QP}_{\mathsf{MPC}}(\hat{x}(k), \boldsymbol{u}_k^r, \boldsymbol{x}_k^r). \tag{117e}$$

Linear time-varying MPC

At every time instant k, solve the constrained optimisation problem

$$\underset{\delta u_k, \delta x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix}^\top W_{k,i} \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix} \tag{118a}$$

$$\text{subject to} \quad \delta x_k(0) = \hat{x}(k) - x_k^r(0) \tag{118b}$$

$$\delta x_k(i+1) = A_{k,i}\, \delta x_k(i) + B_{k,i}\, \delta u_k(i) + r_k(i); \quad i = 0, \dots, N-1 \tag{118c}$$

$$F_{k,i}\, \delta u_k(i) + G_{k,i}\, \delta x_k(i) \le h_{k,i}; \quad i = 0, \dots, N-1. \tag{118d}$$

Linear time-varying MPC

At every time instant k, solve the constrained optimisation problem

$$\underset{\delta u_k, \delta x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix}^\top W_{k,i} \begin{bmatrix} \delta x_k(i) \\ \delta u_k(i) \end{bmatrix} \tag{118a}$$

$$\text{subject to} \quad \delta x_k(0) = \hat{x}(k) - x_k^r(0) \tag{118b}$$

$$\delta x_k(i+1) = A_{k,i}\, \delta x_k(i) + B_{k,i}\, \delta u_k(i) + r_k(i); \quad i = 0, \ldots, N-1 \tag{118c}$$

$$F_{k,i}\, \delta u_k(i) + G_{k,i}\, \delta x_k(i) \leq h_{k,i}; \quad i = 0, \ldots, N-1. \tag{118d}$$

The solution is written as

$$(\boldsymbol{\delta u}_k, \boldsymbol{\delta x}_k) = \mathsf{QP}_{\mathsf{MPC}}(\hat{x}(k), \boldsymbol{u}_k^r, \boldsymbol{x}_k^r). \tag{118e}$$

The control output is given by

$$u(k) = u_k^r(0) + \delta u_k(0). \tag{118f}$$

Nonlinear MPC

At every time instant k, solve the constrained optimisation problem

$$\begin{aligned}
\underset{u_k, x_k}{\text{minimize}} \quad & \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} x_k(i) - x_k^r(i) \\ u_k(i) - u_k^r(i) \end{bmatrix}^\top W_{k,i} \begin{bmatrix} x_k(i) - x_k^r(i) \\ u_k(i) - u_k^r(i) \end{bmatrix} && \text{(119a)} \\
\text{subject to} \quad & x_k(0) = \hat{x}(k) && \text{(119b)} \\
& x_k(i+1) = f(x_k(i), u_k(i)); \quad i = 0, \ldots, N-1 && \text{(119c)} \\
& h(x_k(i), u_k(i)) \leq 0; \quad i = 0, \ldots, N-1. && \text{(119d)}
\end{aligned}$$

Nonlinear MPC

At every time instant k, solve the constrained optimisation problem

$$\underset{u_k, x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} x_k(i) - x_k^r(i) \\ u_k(i) - u_k^r(i) \end{bmatrix}^\top W_{k,i} \begin{bmatrix} x_k(i) - x_k^r(i) \\ u_k(i) - u_k^r(i) \end{bmatrix} \tag{119a}$$

$$\text{subject to} \quad x_k(0) = \hat{x}(k) \tag{119b}$$

$$x_k(i+1) = f(x_k(i), u_k(i)); \quad i = 0, \dots, N-1 \tag{119c}$$

$$h(x_k(i), u_k(i)) \le 0; \quad i = 0, \dots, N-1. \tag{119d}$$

The solution is written as

$$(\boldsymbol{u}_k, \boldsymbol{x}_k) = \text{NLP}(\hat{x}(k), \boldsymbol{u}_k^r, \boldsymbol{x}_k^r). \tag{119e}$$

The control output is given by

$$u(k) = u_k(0). \tag{119f}$$

SQP for NMPC

At every time instant k, the SQP optimisation variables $(\boldsymbol{u}_k, \boldsymbol{x}_k)$ are initialized as

$$(\boldsymbol{u}_k, \boldsymbol{x}_k) = (\boldsymbol{u}_k^{\text{guess}}, \boldsymbol{x}_k^{\text{guess}}). \tag{120a}$$

SQP for NMPC

At every time instant k, the SQP optimisation variables $(\boldsymbol{u}_k, \boldsymbol{x}_k)$ are initialized as

$$(\boldsymbol{u}_k, \boldsymbol{x}_k) = (\boldsymbol{u}_k^{\text{guess}}, \boldsymbol{x}_k^{\text{guess}}). \tag{120a}$$

Then the following QP is solved repeatedly at the current SQP iterate $(\boldsymbol{u}_k, \boldsymbol{x}_k)$ for the Newton correction $(\Delta\boldsymbol{u}_k, \Delta\boldsymbol{x}_k)$, which gives the next iterate by taking a (reduced) Newton step $(\boldsymbol{u}_k, \boldsymbol{x}_k) \leftarrow (\boldsymbol{u}_k, \boldsymbol{x}_k) + t(\Delta\boldsymbol{u}_k, \Delta\boldsymbol{x}_k)$:

$$\underset{\Delta u_k, \Delta x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta x_k(i) \\ \Delta u_k(i) \end{bmatrix}^\top H_{k,i} \begin{bmatrix} \Delta x_k(i) \\ \Delta u_k(i) \end{bmatrix} + J_{k,i}^\top \begin{bmatrix} \Delta x_k(i) \\ \Delta u_k(i) \end{bmatrix} \tag{120b}$$

$$\text{subject to} \quad \Delta x_k(0) = \hat{x}(k) - x_k(0) \tag{120c}$$

$$\Delta x_k(i+1) = A_{k,i}\,\Delta x_k(i) + B_{k,i}\,\Delta u_k(i) + r_k(i); \tag{120d}$$

$$F_{k,i}\,\Delta u_k(i) + G_{k,i}\,\Delta x_k(i) + h_{k,i} \le 0; \quad i = 0, \dots, N-1. \tag{120e}$$

SQP for NMPC

At every time instant k, the SQP optimisation variables $(\boldsymbol{u}_k, \boldsymbol{x}_k)$ are initialized as

$$(\boldsymbol{u}_k, \boldsymbol{x}_k) = (\boldsymbol{u}_k^{\text{guess}}, \boldsymbol{x}_k^{\text{guess}}). \tag{120a}$$

Then the following QP is solved repeatedly at the current SQP iterate $(\boldsymbol{u}_k, \boldsymbol{x}_k)$ for the Newton correction $(\Delta\boldsymbol{u}_k, \Delta\boldsymbol{x}_k)$, which gives the next iterate by taking a (reduced) Newton step $(\boldsymbol{u}_k, \boldsymbol{x}_k) \leftarrow (\boldsymbol{u}_k, \boldsymbol{x}_k) + t(\Delta\boldsymbol{u}_k, \Delta\boldsymbol{x}_k)$:

$$\underset{\Delta u_k, \Delta x_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta x_k(i) \\ \Delta u_k(i) \end{bmatrix}^\top H_{k,i} \begin{bmatrix} \Delta x_k(i) \\ \Delta u_k(i) \end{bmatrix} + J_{k,i}^\top \begin{bmatrix} \Delta x_k(i) \\ \Delta u_k(i) \end{bmatrix} \tag{120b}$$

$$\text{subject to} \quad \Delta x_k(0) = \hat{x}(k) - x_k(0) \tag{120c}$$

$$\Delta x_k(i+1) = A_{k,i}\, \Delta x_k(i) + B_{k,i}\, \Delta u_k(i) + r_k(i); \tag{120d}$$

$$F_{k,i}\, \Delta u_k(i) + G_{k,i}\, \Delta x_k(i) + h_{k,i} \leq 0; \quad i = 0, \ldots, N-1. \tag{120e}$$

The solution of the SQP is written as

$$(\boldsymbol{u}_k, \boldsymbol{x}_k) = \text{SQP}(\hat{x}(k), \boldsymbol{u}_k^{\text{guess}}, \boldsymbol{x}_k^{\text{guess}}, u_k^r, x_k^r). \tag{120f}$$

The control output is given by

$$u(k) = u_k(0). \tag{120g}$$

Comparison between linear and nonlinear MPC

Consider the following simple problem

$$
\begin{aligned}
\underset{\boldsymbol{u}_k,\boldsymbol{x}_k}{\text{minimize}} \quad & \sum_{i=0}^{N} \|x_k(i)\|^2 + 20 \sum_{i=0}^{N-1} \|u_k(i)\|^2 && \text{(121a)} \\
\text{subject to} \quad & x_k(0) = \hat{x}(k) && \text{(121b)} \\
& x_k(i+1) = 0.9 x_k(i) + \begin{bmatrix} \sin(\begin{bmatrix} 0 & 1 \end{bmatrix} x_k(i)) \\ u_k(i) + u_k(i)^3 \end{bmatrix}, \quad i = 0, \ldots, N-1 && \text{(121c)} \\
& |u_k(i)| \leq 0.5, \quad i = 0, \ldots, N-1. && \text{(121d)}
\end{aligned}
$$

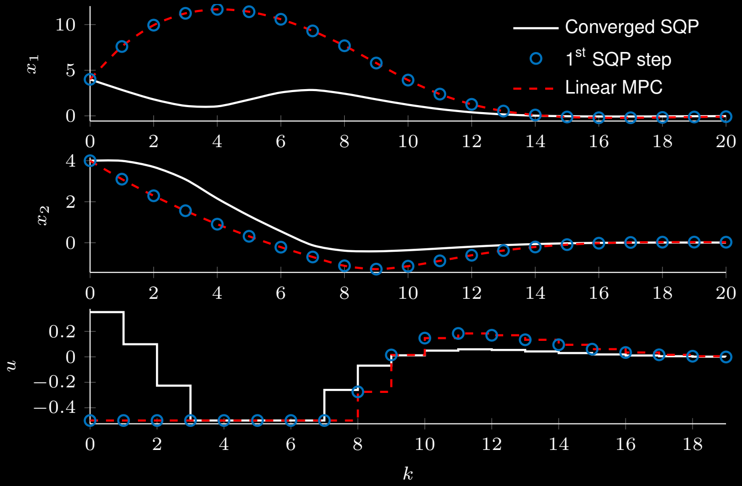

Figure 46: Comparison between receding horizon control solution from SQP after full convergence, SQP after one step and linear MPC.

Real-time iteration (RTI) scheme

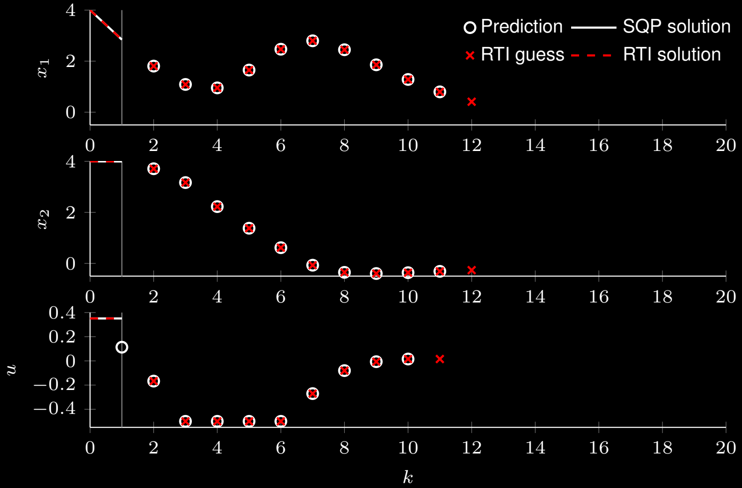

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

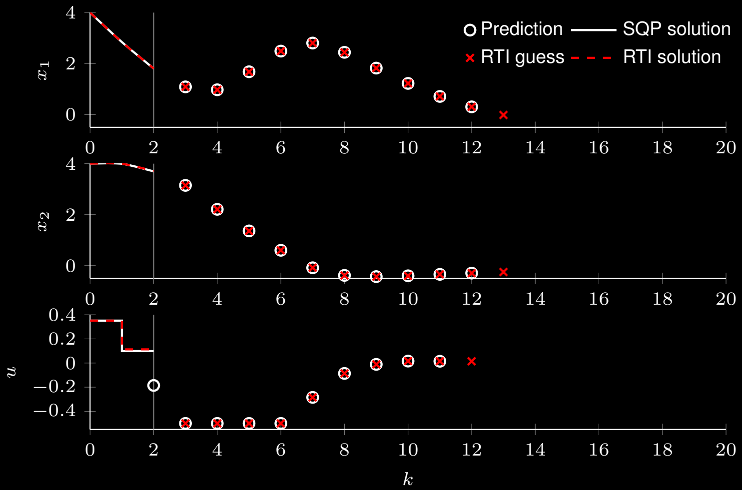

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

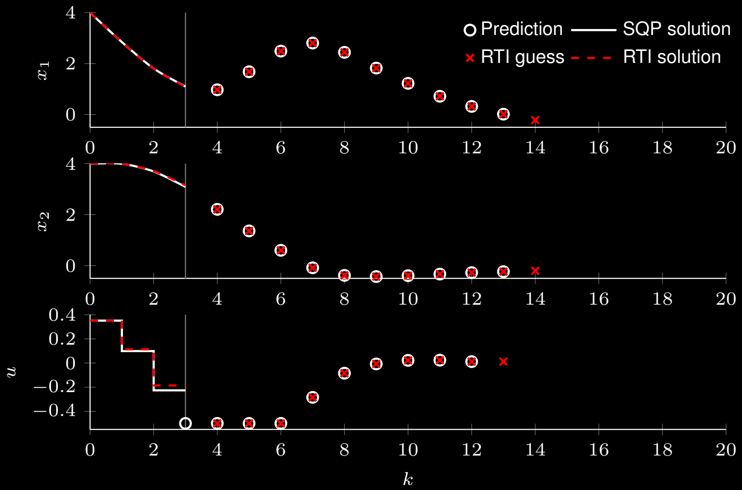

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

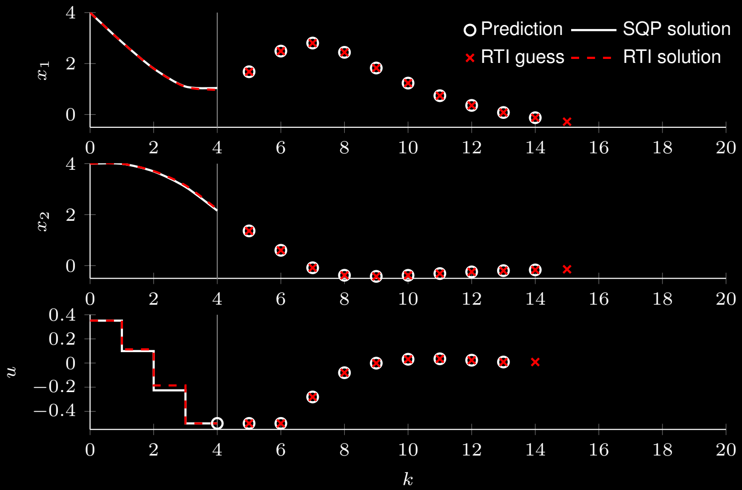

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

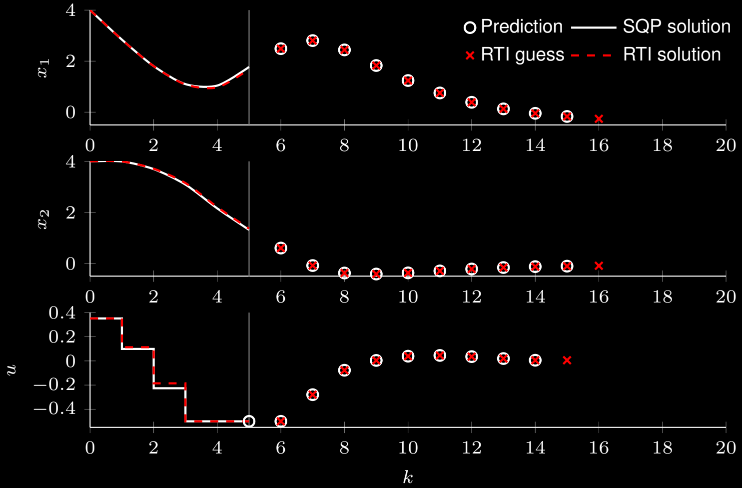

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

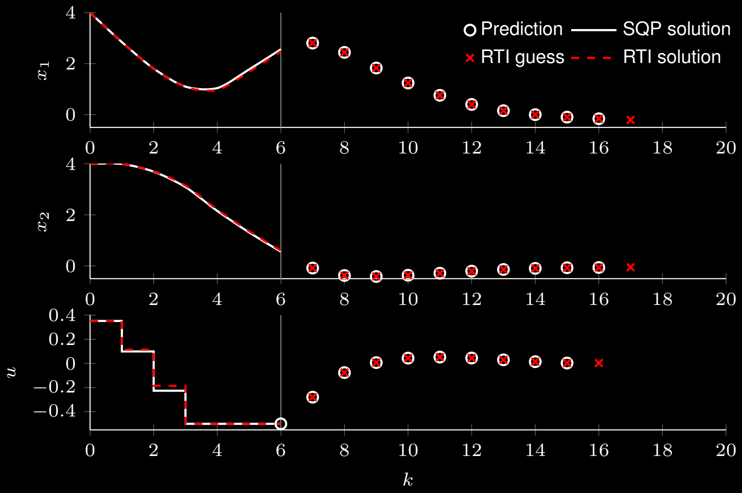

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

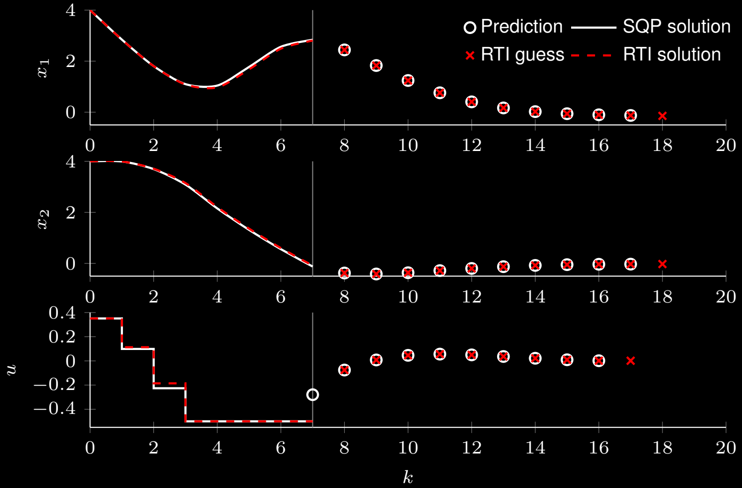

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

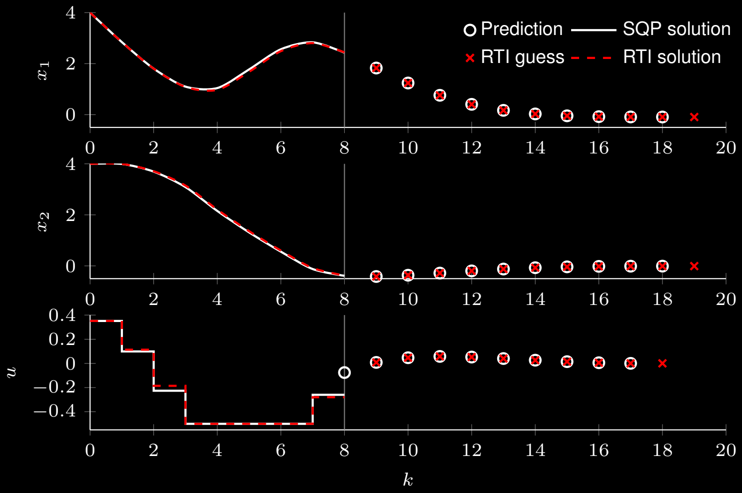

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration (RTI) scheme

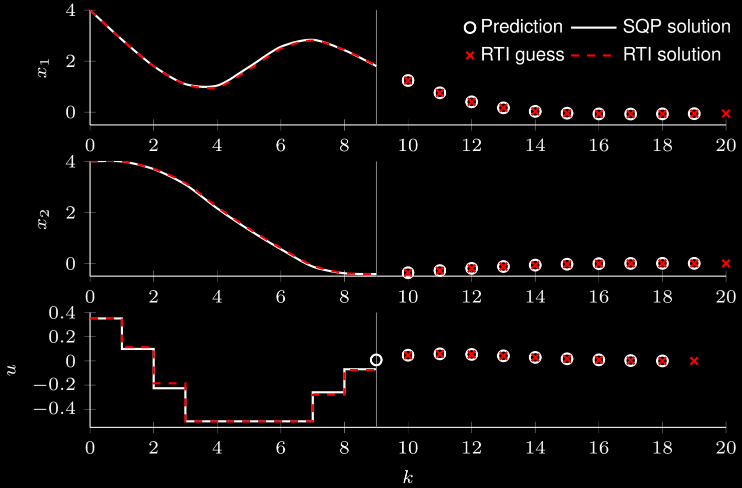

Figure 47: Real-time iteration scheme (RTI). The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI.

Real-time iteration in the presence of disturbances

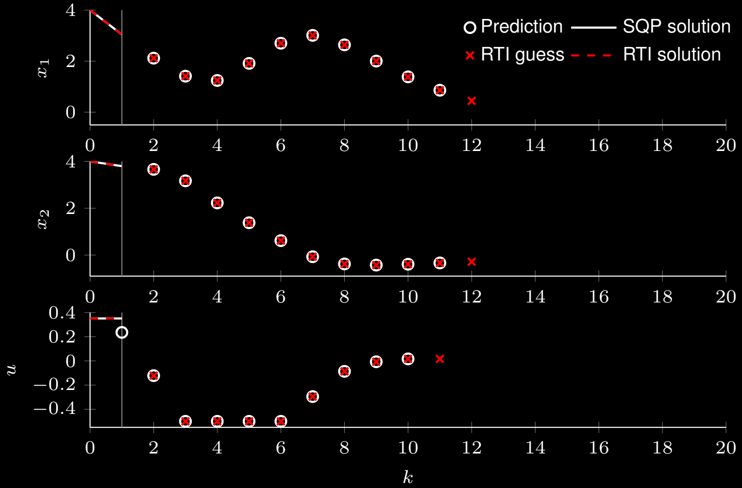

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

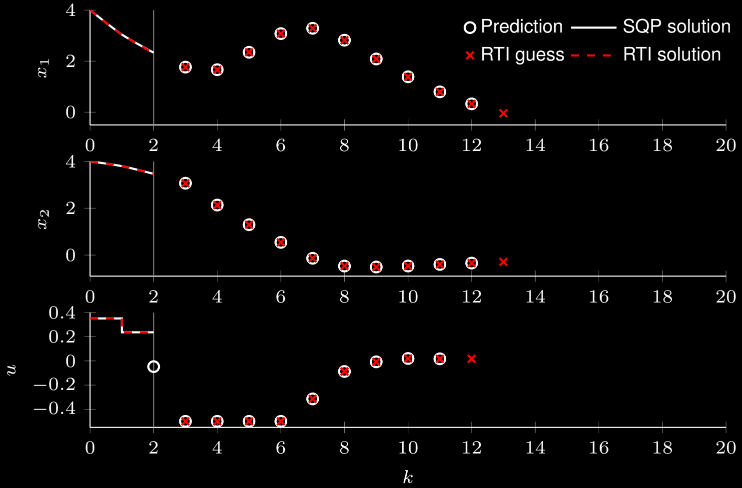

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

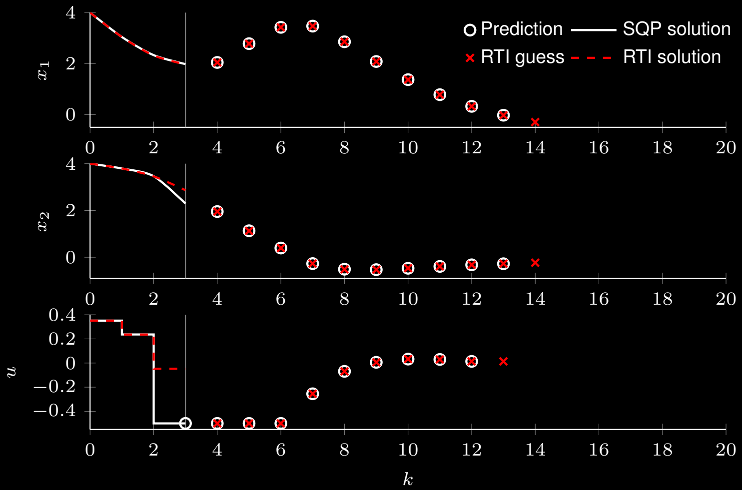

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

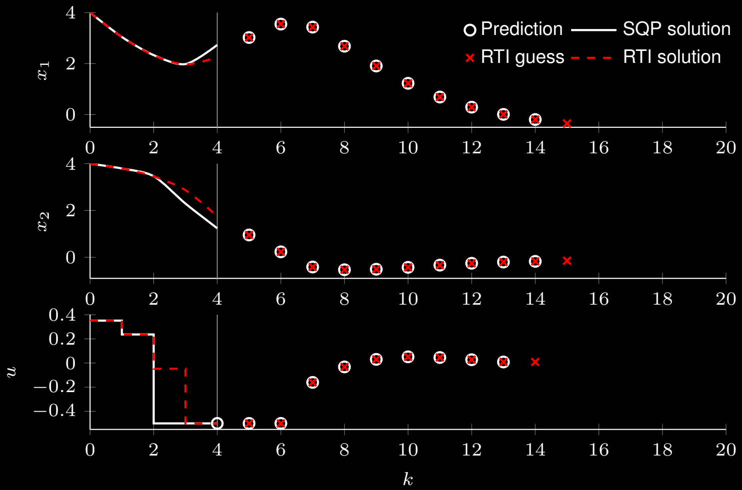

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

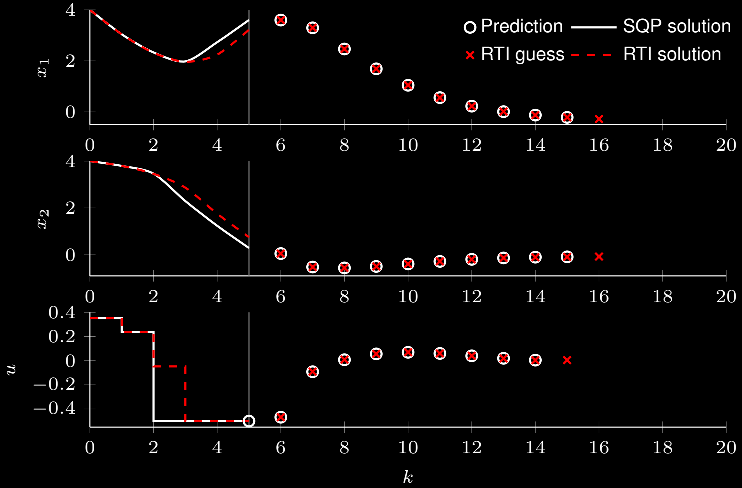

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

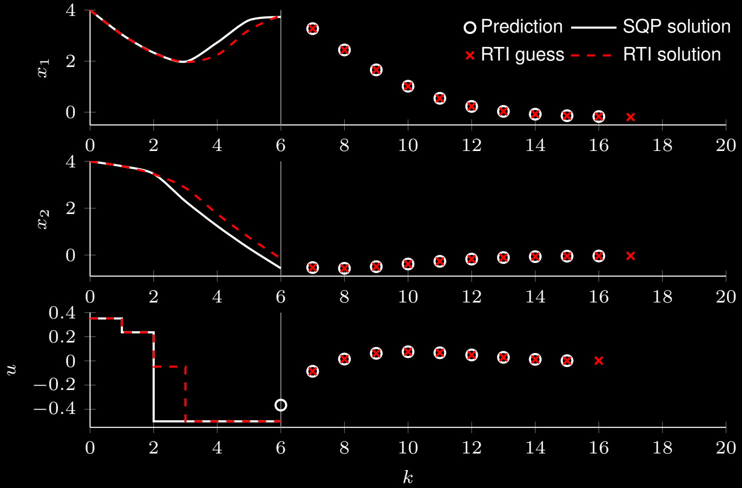

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

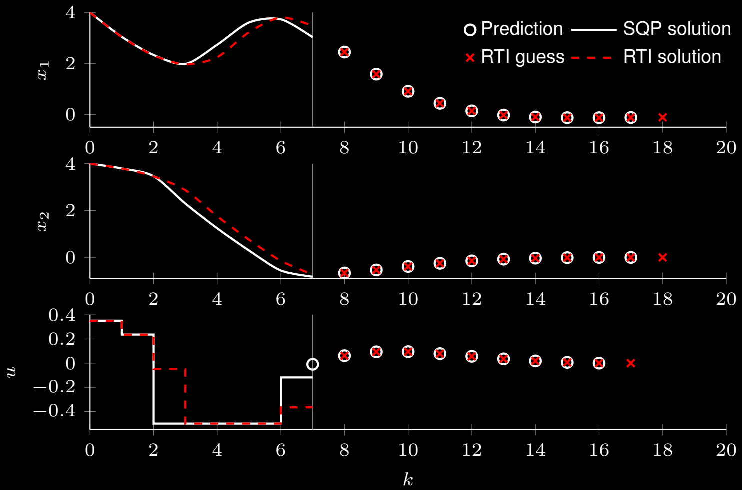

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

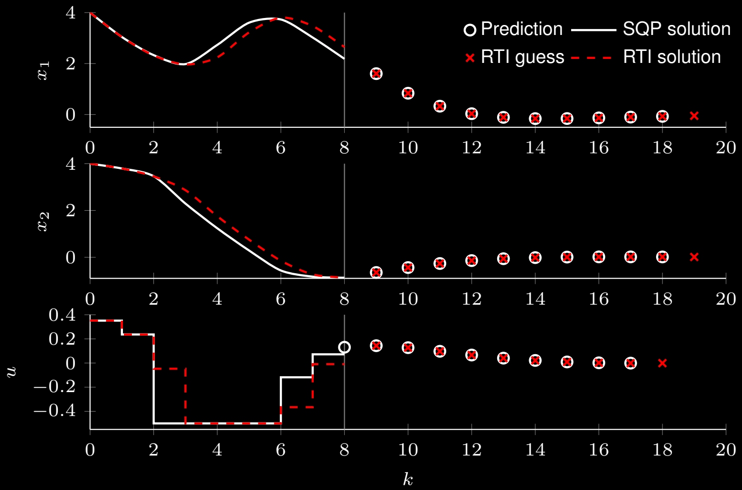

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

Real-time iteration in the presence of disturbances

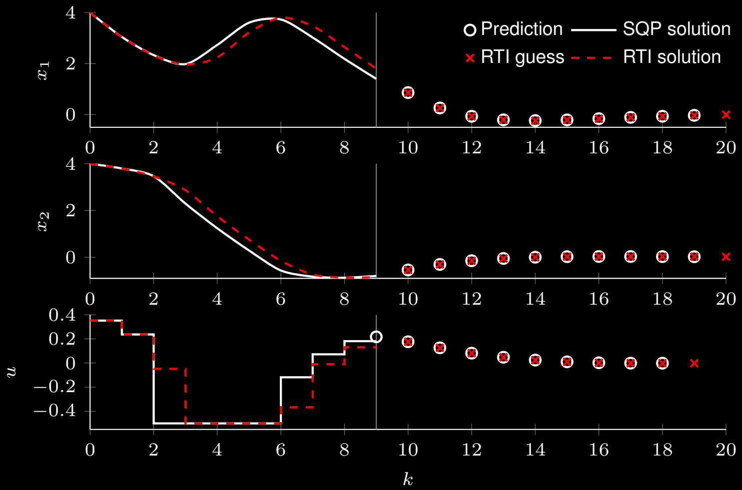

Figure 48: Real-time iteration scheme (RTI) in the presence of disturbances. The circles indicate predictions made by the MPC update. The crosses indicate the shifted predictions as an initial guess in RTI. The solid line shows the optimal solution for the

References

[1] J.B. Rawlings, D.Q. Mayne, and M.M. Diehl. *Model Predictive Control: Theory, Computation, and Design, 2nd edition*. Nob Hill Publishing 2017.
Available online at `https://sites.engineering.ucsb.edu/~jbraw/mpc`

[2] G. Goodwin, M.M. Seron, and J.A. De Don. *Constrained Control and Estimation*. Springer 2004.
Available online via Chalmers Library.

[3] F Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*.
Available online at http://www.mpc.berkeley.edu/mpc-course-material

[4] J. Maciejowski. *Predictive Control with Constraints*. Prentice Hall 2002.

[5] S. Boyd and L. Vandenberghe. *Convex optimisation*. Cambridge University Press 2004.

[6] Diehl, M. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, University of Heidelberg, 2001.

CHALMERS
UNIVERSITY OF TECHNOLOGY