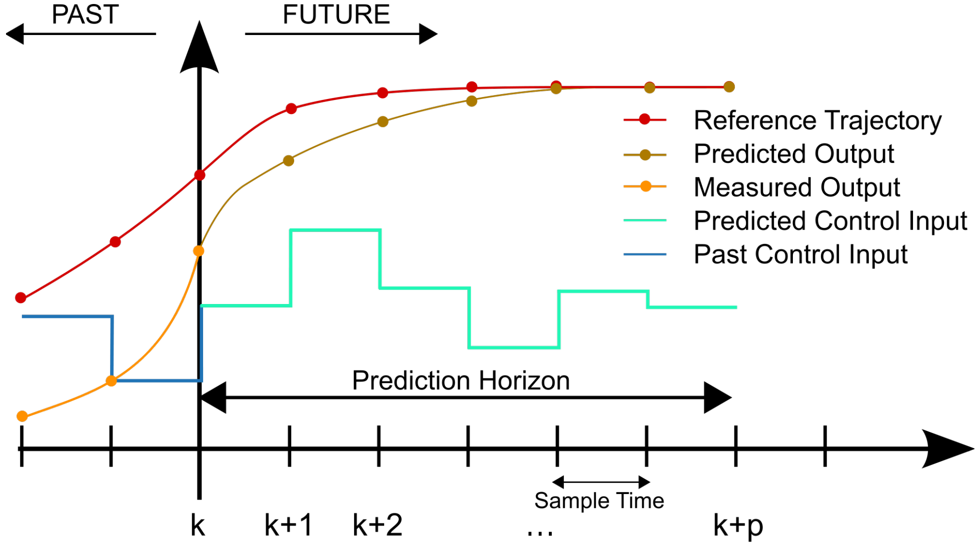
Instruction 2

5LMB0 - Model Predictive Control



1 Introduction

In this exercise set we use the first- and second-order systems introduced in the Instruction 1. Please, read the exercise definition carefully: some of the system parameters may be different from the previous exercise set.

In this exercise set students will practice to solve a quadratic programming optimization problem for optimality and implement a constrained MPC controller using the MATLAB function 'quadprog.m'. (https://www.mathworks.com/help/optim/ug/quadprog.html)

2 Tasks

2.1 Quadratic Programming

(a) Consider the first order model. Calculate U_0 that minimizes the cost function

$$J(x(0), U_0) = \frac{1}{2}\left(\sum_{i=0}^{N-1}(x_{i|0}^2 + u_{i|0}^2) + x_{N|0}^2\right),$$

with the initial condition $x_{0|0} = x(0) = 5$, sampling frequency $T_s = 1s$ **only** in this subsection for simplicity (in Instruction 1, it was $T_s = 0.1s$), $N = 1$ and the input constraints $-1 \le u(k) \le 1$, for all k. Solve the problem using the MATLAB function '`quadprog.m`'. We also demonstrate how to solve it using the KKT conditions in the solution.

Solution: Use $x_{i+1|k} = Ax_{i|k} + Bu_{i|k} = x_{i|k} + u_{i|k}$ to get

$$J(x(0), U_0) = \frac{1}{2}[x_{0|0}^2 + x_{1|0}^2 + u_{0|0}^2]$$
$$J(x(0), U_0) = \frac{1}{2}[2x_{0|0}^2 + 2x_{0|0}u_{0|0} + 2u_{0|0}^2]$$

Use $x(0) = 5$ to get

$$J(x(0), U_0) = \frac{1}{2}[50 + 10u_{0|0} + 2u_{0|0}^2]$$

Use the constraints

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u(k) \le \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Since we want to minimize the cost $J(x(0), U_0)$, it is clear that we are operating on the constraint $u_{0|0} \ge -1$. Use slide 13 of lecture 3 for the KKT conditions. Based on the given input constraints, assign $A_{p_1} = 1$, $A_{p_2} = -1$, $b_{p_1} = 1$ and $b_{p_2} = 1$.

$$5 + 2u_{0|0} + \lambda_2(-1) = 0$$

$$
\begin{aligned}
-u_{0|0} &= 1 \\
u_{0|0} - 1 &< 0 \\
\lambda_2 &\geq 0 \\
\lambda_1 &= 0
\end{aligned}
$$

It holds for $\lambda_2 = 3$ and $\lambda_1 = 0$.

(b) Consider the second order model. Calculate U_0 that minimizes the cost function

$$
J(x(0), U_0) = \frac{1}{2}\left(\sum_{i=0}^{N-1}(x_{i|0}^T I_2 x_{i|0} + u_{i|0}^2) + x_{N|0}^T I_2 x_{N|0}\right), \tag{1}
$$

with initial condition $x_{0|0} = x(0) = [-1, -2]^T$, sampling frequency $T_s = 1s$ **only** in this subsection for simplicity (in Instruction 1, it was $T_s = 0.1s$), $N = 1$ and the input constraints $-1 \leq u(k) \leq 1$ for all k. Solve the problem using the MATLAB function '`quadprog.m`'. We also demonstrate how to solve it using the KKT conditions in the solution.

Solution:

$$
J(x(0), U_0) = \frac{1}{2}[x_{1,1|0}^2 + x_{2,1|0}^2 + x_{1,0|0}^2 + x_{2,0|0}^2 + u_{0|0}^2].
$$

Use $x_{i+1|k} = Ax_{i|k} + Bu_{i|k}$ with

$$
A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -0.5 \\ -1 \end{bmatrix}
$$

and substitute $x(0) = [-1, -2]^T$ to get

$$
x_{1|0} = \begin{bmatrix} -3 - 0.5u_{0|0} \\ -2 - u_{0|0}. \end{bmatrix}
$$

This results in

$$
\begin{aligned}
x_{1,0|0} &= -1, \\
x_{2,0|0} &= -2, \\
x_{1,1|0} &= -3 - \frac{1}{2}u_{0|0}, \\
x_{2,1|0} &= -2 - u_{0|0}.
\end{aligned}
$$

Filling this in the cost function we get

$$
J(x(0), U_0) = \frac{1}{2}[\frac{9}{4}u_{0|0}^2 + 7u_{0|0} + 18]
$$

Use the constraints

$$
\begin{bmatrix} 1 \\ -1 \end{bmatrix} u(k) \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}
$$

Similarly to the first-order model, the cost will be minimized with a negative input $u_{0|0} < 0$. Use slide 13 of lecture 3 for the KKT conditions.

$$\begin{aligned} \frac{9}{4}u_{0|0} + \frac{7}{2} + \lambda_2(-1) &= 0 \\ -u_{0|0} &= 1 \\ u_{0|0} - 1 &< 0 \\ \lambda_2 &\geq 0 \\ \lambda_1 &= 0 \end{aligned}$$

It holds for $\lambda_2 = \frac{5}{4}$ and $\lambda_1 = 0$.

2.2 Constrained MPC (MPC to QP)

1. The constraints for the first order model are defined as

$$-0.3 \leq u(k) \leq 0.3. \tag{2}$$

2. The constraints for the second order model are defined as

$$-0.1 \leq u(k) \leq 0.1 \quad \text{and} \quad \begin{pmatrix} -10 \\ -10 \end{pmatrix} \leq x(k) \leq \begin{pmatrix} 10 \\ 10 \end{pmatrix}. \tag{3}$$

(a) Derive the matrices $\mathcal{D}$, $\mathcal{M}$, $\mathcal{E}$ and the vector c for each corresponding model and constraints. (Hint: Assign an enough large boundary if no constraint exists)

Solution: See slides 61-65 of lecture slides week 1-2.

(b) Construct constraint matrices W, L and the vector c (create a MATLAB function that generates these matrices for any prediction horizon N).

Solution: See slides 61-65 of lecture slides week 1-2.

(c) Simulate the constrained MPC in closed-loop with the model for several initial conditions and plot trajectories.

Solution: See figure 1 and 2.

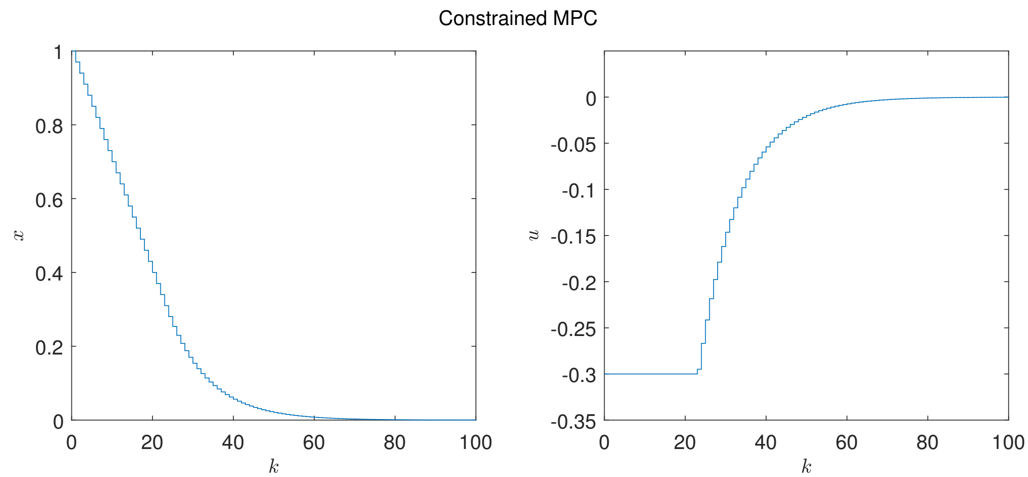


Figure 1: Closed-loop trajecotry, 1D system, $N = 100$.

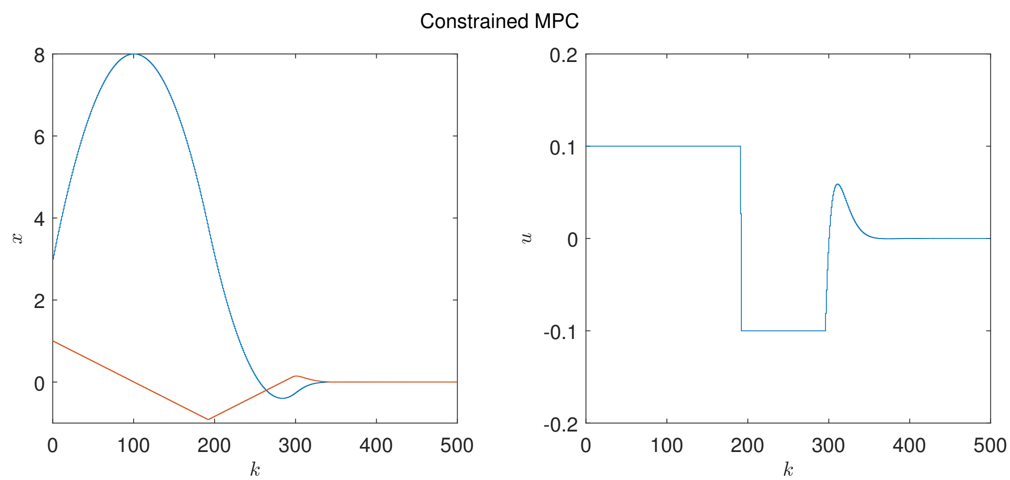


Figure 2: Closed-loop trajecotry, 2D system, $N = 100$.

(d) Plot both the constrained MPC control input $u(k)$ trajectory and the open loop predicted trajectory $u_{0|k}, ..., u_{N-1|k}$ at $k = 0$. What do you notice? Repeat this for other values of k.

Solution: See figure 3 and 4. In the 1D case, no significant difference can be seen. In the 2D system, however, it can be observed a difference between the open-loop and closed-loop trajectories. This happens due to the finite prediction horizon: at earlier samples the optimal input sequence is to apply maximum input for larger period of time.

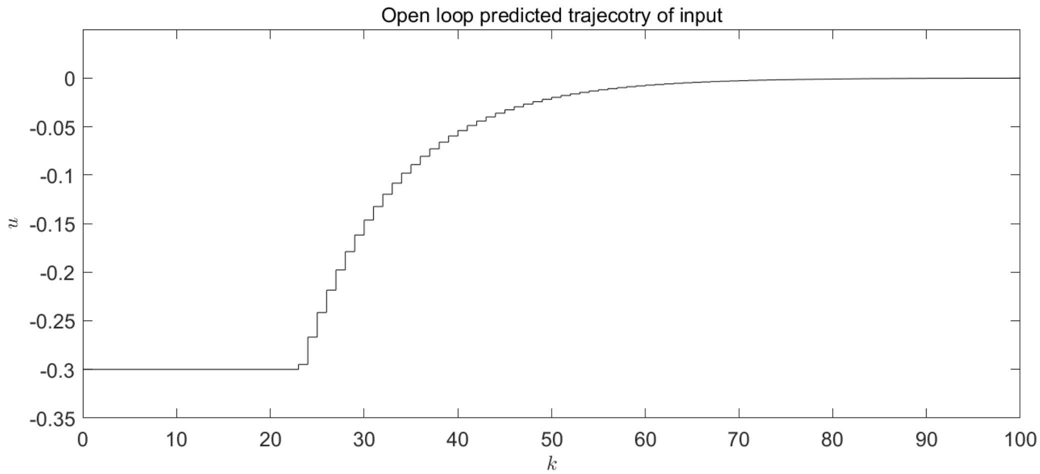


Figure 3: Comparison with open loop predicted trajecotry, 1D system, $N = 100$.

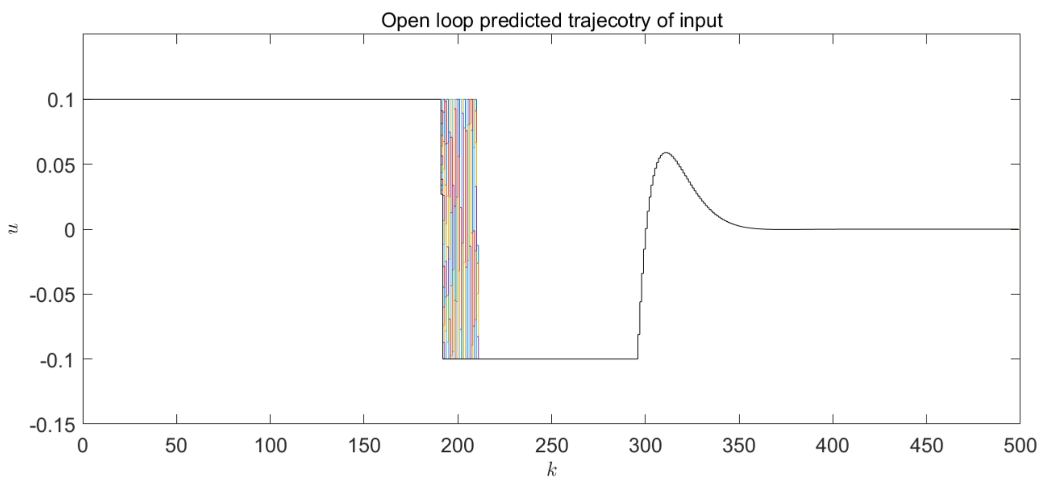


Figure 4: Comparison with open loop predicted trajecotry, 2D system, $N = 100$.

(e) Compare the constrained MPC with a saturated unconstrained MPC. What do you notice? Is there a difference between a short and a long prediction horizon?

Solution: See figure 5 and 6. For the 1D case we notice that the trajectories look very similar for both short and long prediction horizons. In the 2D system, however, we can see a large difference where the constrained MPC converges significantly faster for a sufficiently long prediction horizon.

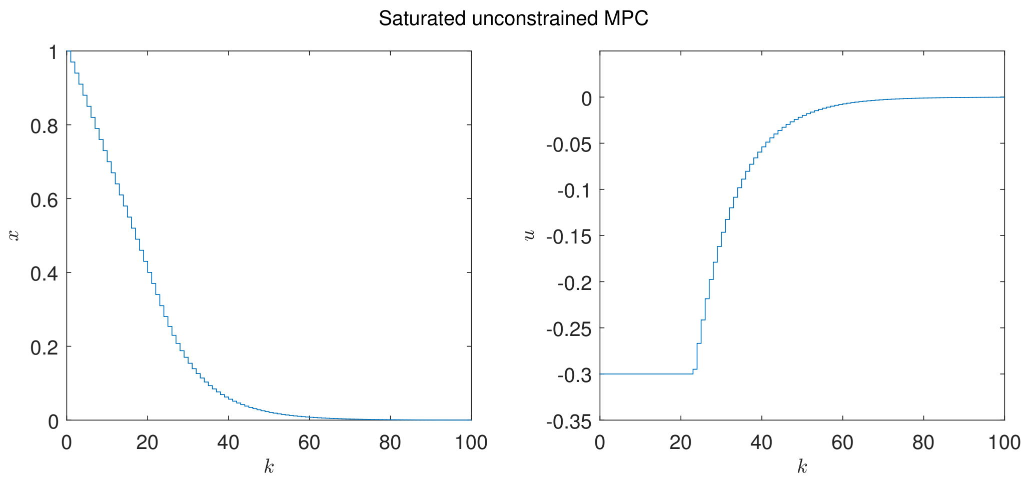


Figure 5: Comparison with saturated unconstrained MPC, 1D system, $N = 100$.

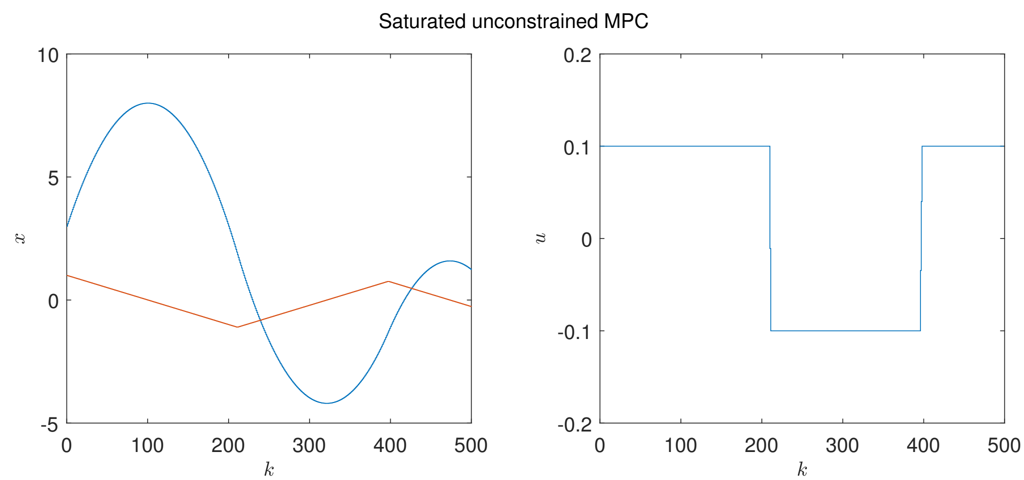


Figure 6: Comparison with saturated unconstrained MPC, 2D system, $N = 100$.