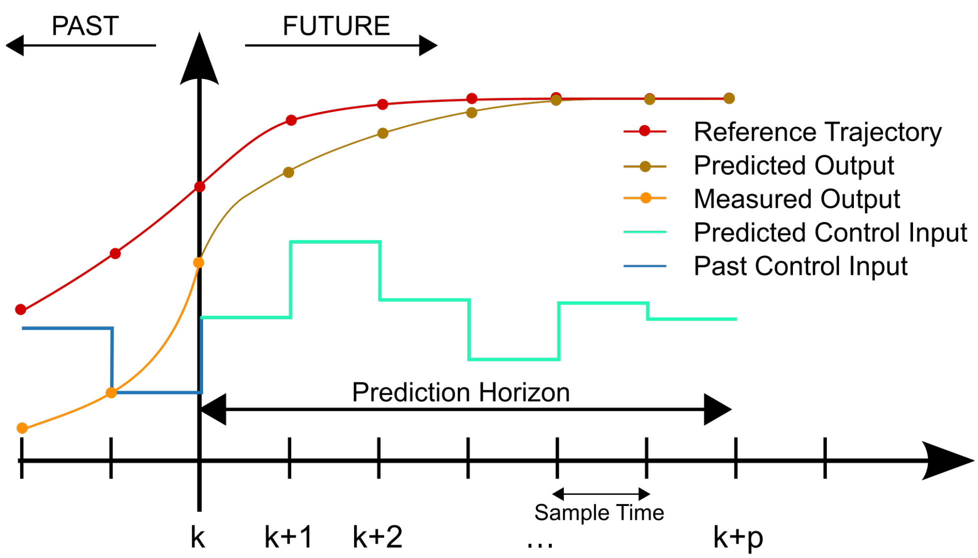
Instruction 6

5LMB0 - Model Predictive Control



1 Introduction

In this exercise set students will practice designing and compare nonlinear MPC controllers for a nonlinear dynamic system. Please, read the problem description carefully, the system is different from the previous instructions.

2 Model

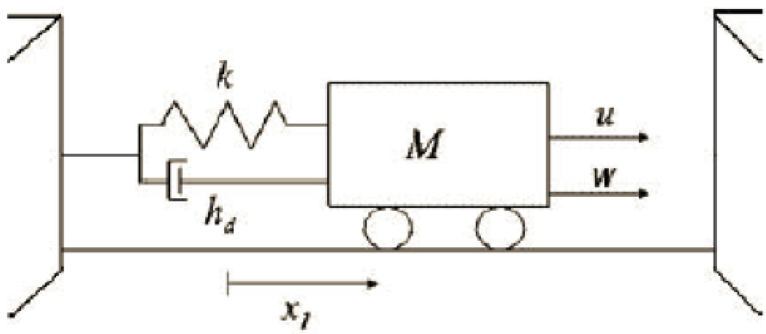


Figure 1: The physical system.

Assume a nonlinear second-order mass-spring-damper system that moves in longitudinal direction only from lecture slides 6 on nonlinear MPC (see Fig.1). The nonlinear model of the form $\dot{x}(t) = f(x(t), u(t))$ is given by:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} f_1(x(t), u(t)) \\ f_2(x(t), u(t)) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -\frac{k_0}{M} e^{-x_1(t)} x_1(t) - \frac{h_d}{M} x_2(t) + \frac{1}{M} u(t) \end{bmatrix}, \quad t \in \mathbb{R}_+ \tag{1}$$

where the two-dimensional state $x(t) := [x_1(t) \;\; x_2(t)]^T$ is the position x_1 and velocity x_2, input $u(t)$ is the force applied to the mass. The rest of the parameters are constants values, given in the Table 1. The goal is to control the position $x_1(t)$ and velocity $x_2(t)$ of the mass by applying force $u(t)$ in x-direction. Assume the discretization sampling period $T_s = 0.4$ s. The constraints are:

$$\begin{gathered} \begin{bmatrix} -2.65 \\ -2 \end{bmatrix} \leq x(t) \leq \begin{bmatrix} 2.65 \\ 2 \end{bmatrix} \\ -4.5 \leq u(t) \leq 4.5. \end{gathered} \tag{2}$$

Table 1: Model parameters

Parameter	Value
Mass m, [kg]	1
Spring constant k_0, [N/m]	0.33
Damping factor h_d, [Ns/m]	1.1

3 Tasks

For the system described above, design a linear constrained MPC, NMPC, and iterative QP NMPC controllers that bring the mass to equilibrium at the origin. When running your simulations, always use the nonlinear model (1) to simulate the dynamics. When testing, try the following initial conditions: $x(0) = [-1 \ \ 1]^T$ and $x(0) = [-2 \ \ 2]^T$. Use prediction horizon $N = 5$.

(a) Discretize the system using forward Euler discretization method for a sampling period T_s.

Solution: See slides page 8-9 in Lecture 6. A T_s is missed in the first term of the second equation:

$$x^2\left((k+1)T_s\right) = -\frac{k_0 T_s}{M} e^{-x^1(kT_s)} x^1\left(kT_s\right) - \frac{h_d T_s}{M} x^2\left(kT_s\right) + \frac{T_s}{M} u\left(kT_s\right) + x^2\left(kT_s\right)$$

(b) Linearize the discrete nonlinear system around the equilibrium point $x_{eq}(t) = [0 \ \ 0]^T$ and construct a linear constrained MPC controller.

Solution: The linearized discrete state-space model at the origin $[0 \ \ 0]^T$ is:

$$x(k+1) = \begin{pmatrix} 1 & T_s \\ -T_s \frac{k_0}{M} e^0 & 1 - T_s \frac{h_d}{M} \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ \frac{T_s}{M} \end{pmatrix} u(k)$$

The linear constrained MPC controller can be constructed using the same method as in the previous seminars. Otherwise, the provided MATLAB code implements the MPC controller using the YALMIP toolbox.

(c) Implement a nonlinear MPC using a nonlinear programming solver. *Note: you may use* MATLAB *constrained optimization function* **`fmincon.f`** *or* **`optimizer.f`** *from YALMIP toolbox. For the latter, check implementation of the implicit MPC on* `yalmip.github.io/example/standardmpc`.

Solution: Check the provided MATLAB code.

(d) Reformulate the discrete system in a quasi-LPV representation and implement the iterative QP nonlinear MPC.

Solution: Please note that the MATLAB code for quasi-LPV will not be provided. Instead, you will be required to construct it yourself in the graded assignment.

(e) Compare the CPU time and the state trajectories for all three methods. What can you conclude?

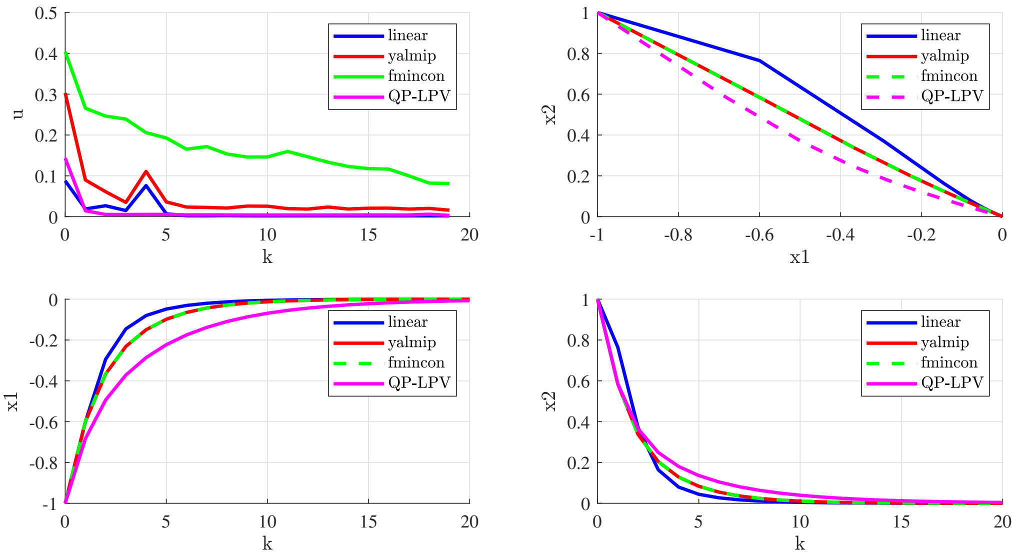


Figure 2: CPU time and state trajectories for the MPC controllers with initial condition $x(0) = [-1 \;\; 1]^T$.

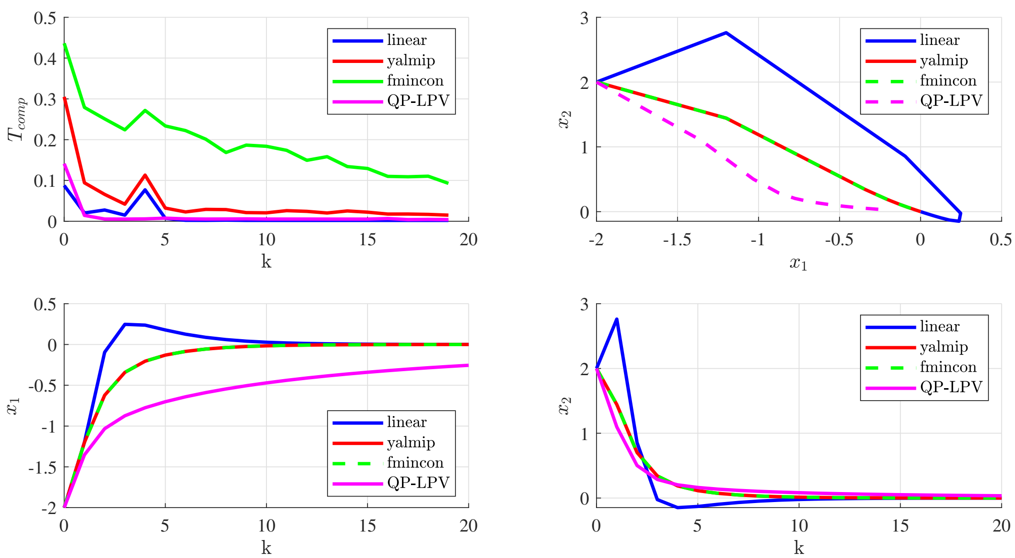


Figure 3: CPU time and state trajectories for the MPC controllers with initial condition $x(0) = [-2 \;\; 2]^T$.