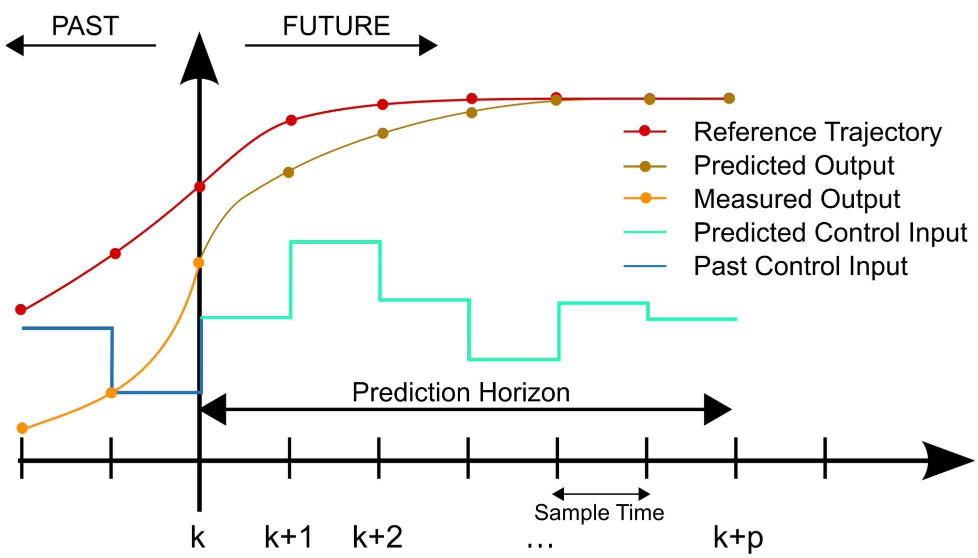
Instruction 3

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 compute the feasible sets of input and state and implement a constrained and explicit MPC controller using a MPT3 toolbox. The MPT3 toolbox is necessary for this course. (https://www.mpt3.org/Main/Installation)

2 Tasks

2.1 Feasible set of inputs and states

1. Consider the first-order model with $N = 2$ and the constraints

$$-1 \leq x(k) \leq 1 \quad \text{and} \quad -1 \leq u(k) \leq 1 \tag{1}$$

   (a) Plot the feasible set of inputs. Consider several initial states $x(0)$. Take, for example, $x(0) = 0.9$. What do you notice? (See slide 16 from the lecture on constrained MPC). (https://www.mpt3.org/Geometry/Sets#Polyhedron)

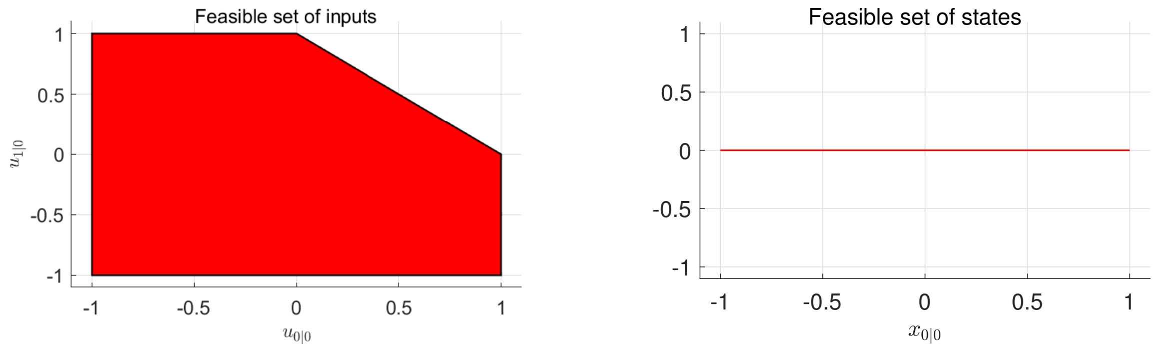
   Try the input constraints $0.2 \leq u(k) \leq 1$.

   Solution: Result can be found in Fig. 1a.
   $LU_k \leq c + Wx_0$
   `Feasibleset_U = Polyhedron('A',L,'B',c+W*x0);`
   `plot(Feasibleset_U).`

   (b) Plot the feasible set of states. Use $N = 1, 2, 3$. Explain the results. (Hint: define a polytope in the augmented space $[x; U]$ using the MPC constraints in compact form. Then, use projection on the x space). (http://people.ee.ethz.ch/~mpt/2/docs/refguide/mpt/@polytope/projection.html)

   Try the input constraints $0.2 \leq u(k) \leq 1$.

   Solution: The results are all the same (Fig. 1b). The system only has 1 state, thus the set for feasible states is only in one dimension, limited by the state constraints.
   $[-WL][x_0; U_k] \leq c$
   `Feasibleset_x0_U = Polyhedron('A',[-W L],'B',c);`
   `Feasibleset_x0 = projection(Feasibleset_x0_U,1);`
   `plot(Feasibleset_x0).`



(a) Input (b) State

Figure 1: Feasible set of inputs and states, 1D system.

2. Consider the second-order model with $N = 2$ and the constraints

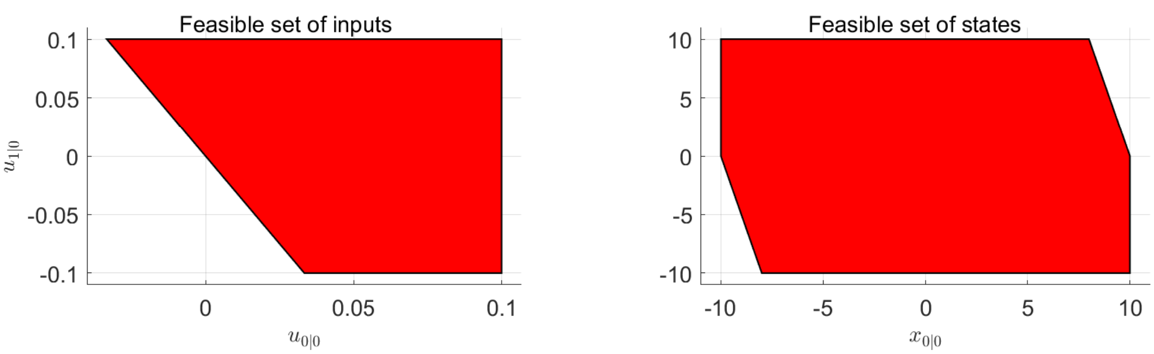
$$\begin{pmatrix} -10 \\ -10 \end{pmatrix} \leq x(k) \leq \begin{pmatrix} 10 \\ 10 \end{pmatrix} \quad \text{and} \quad -1 \leq u(k) \leq 1. \tag{2}$$

(c) Plot the feasible set of inputs. Consider several initial states $x(0)$. Take, for example, $x(0) = [9, 5]^T$. What do you notice?

Solution: Result can be found in Fig. 2a.

(d) Plot the feasible set of states. Use $N = 1, 2, ..., 5$. Explain the results.

Solution: The results for $N = 2$ can be found in figure 2b. The system only has 2 states, thus, the set for feasible states is 2-dimensional, x_1 and x_2. Notice that when you increase the prediction horizon N, the feasible state space shrinks.



(a) Input (b) State

Figure 2: Feasible set of inputs and states, 2D system.

2.2 Constrained MPC

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

$$-0.3 \leq u(k) \leq 0.3. \tag{3}$$

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

$$-0.1 \leq u(k) \leq 0.1 \quad \text{and} \quad \begin{pmatrix} -10 \\ -10 \end{pmatrix} \leq x(k) \leq \begin{pmatrix} 10 \\ 10 \end{pmatrix}. \tag{4}$$

(a) Simulate the constrained MPC for the second order model with initial condition $x(0) = [2, 2]^T$. Explain the results.

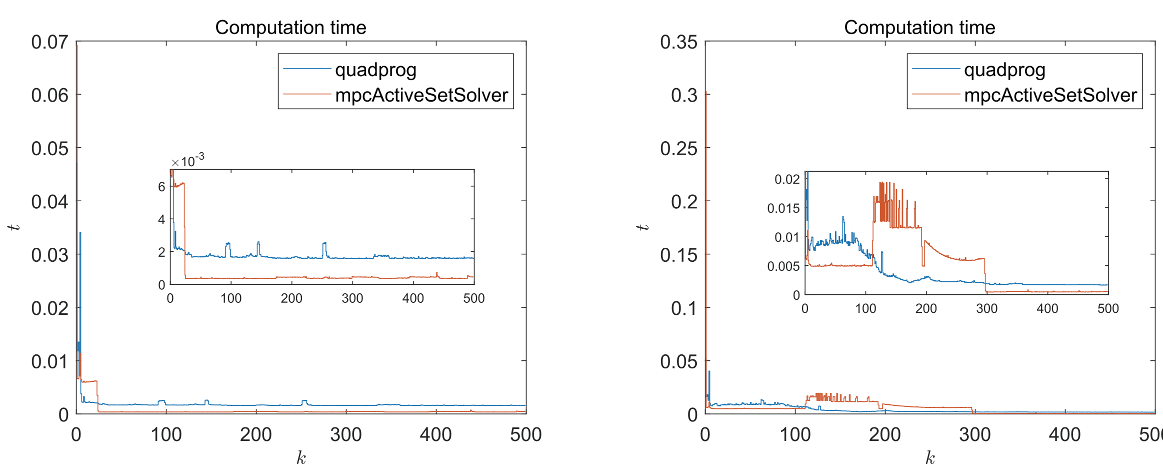
Solution: At a certain point the optimization problem becomes infeasible because the trajectory goes out of the feasible set. Namely, the last future predicted states does not satisfy the state constraints no matter what input is applied. A terminal constraint set can be used to prevent this. It will be discussed later in this course (week 5).

(b) Measure the computation time per iteration of the `quadprog.m` using the MATLAB functions `tic.m` and `toc.m`. Plot the measured computation time over k.

Solution: Results can be found in figure 3. Use a warm start to reduce the computation time of `quadprog.m` by using the shifted solution U_k^* of the previous computation and add a zero at the end.

(c) Use the Matlab function `mpcActiveSetSolver.m` instead of `quadprog.m`. Measure the computation time. What do you notice? (https://www.mathworks.com/help/mpc/ref/mpcactivesetsolver.html) Note: Model Predictive Control Matlab toolbox is required.

Solution: Results can be found in figure 3. Check the trajectories to see if the results are the same as with `quadprog.m`.
The `mpcActiveSetSolver.m` uses $iA0$ which indicates which constraints are active. These can be used in the next iteration for a faster start. Check the results with and without an updated $iA0$.



(a) 1D (b) 2D

Figure 3: Computation time per iteration.

2.3 Explicit MPC

Use the second-order model with the following constraints:

$$\begin{pmatrix} -10 \\ -10 \end{pmatrix} \leq x(k) \leq \begin{pmatrix} 10 \\ 10 \end{pmatrix} \quad \text{and} \quad -1 \leq u(k) \leq 1 \tag{5}$$

(a) Define the MPC controller using the MPT3 toolbox and compute the explicit solution. Plot the partitions. (Hint: study "Set constraint" section on https://www.mpt3.org/UI/Filters)

Solution: Use the "Set constraint" section from https://www.mpt3.org/UI/Filters Be sure to use the discrete time A and B matrices for the LTI system. The results can be found in Fig. 4. The outline of the partition plot is equal to the feasible state set.

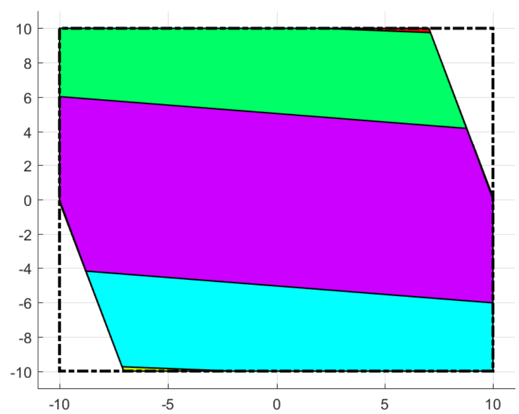


Figure 4: Partitions for $N = 3$

(b) Simulate the explicit MPC in closed-loop with the model and plot trajectories.

Solution: See figure 5.

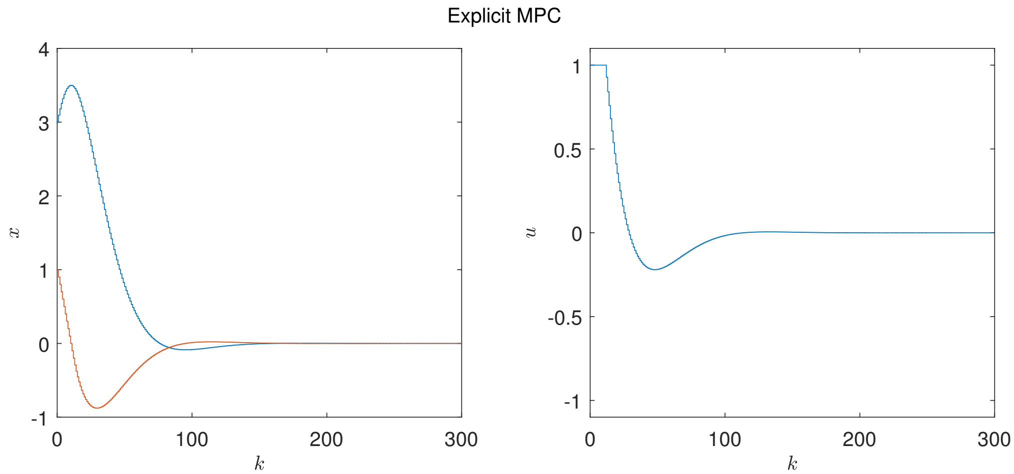


Figure 5: Closed-loop trajecotry, explicit MPC, $N = 10$.

(c) Compare the results with the on-line constrained MPC.

Solution: See figure 6.

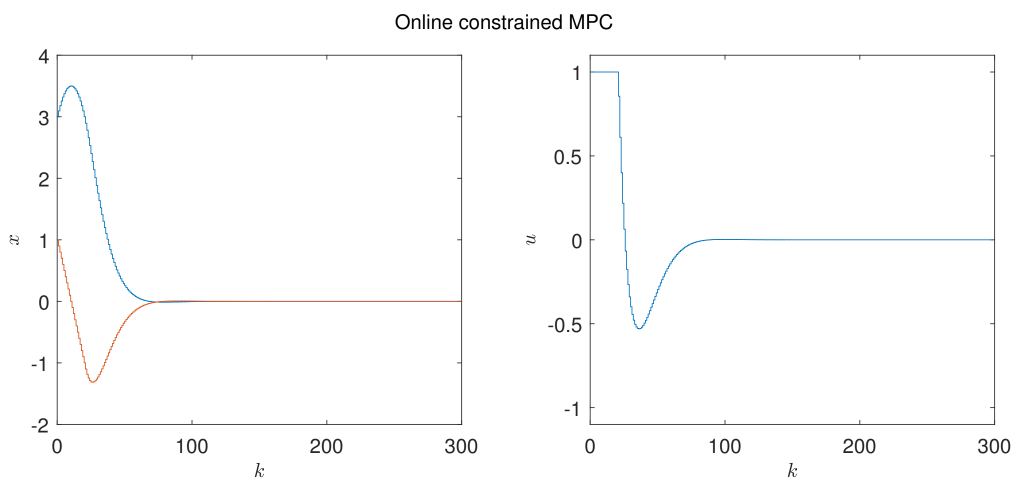


Figure 6: Closed-loop trajecotry, online constrained MPC, $N = 10$.

(d) Compare the computation time of the explicit MPC with the `quadprog.m` or the `mpcActiveSetSolver.m` used for on-line constrained MPC (choose the fastest of the two).

Solution: See figure 7.

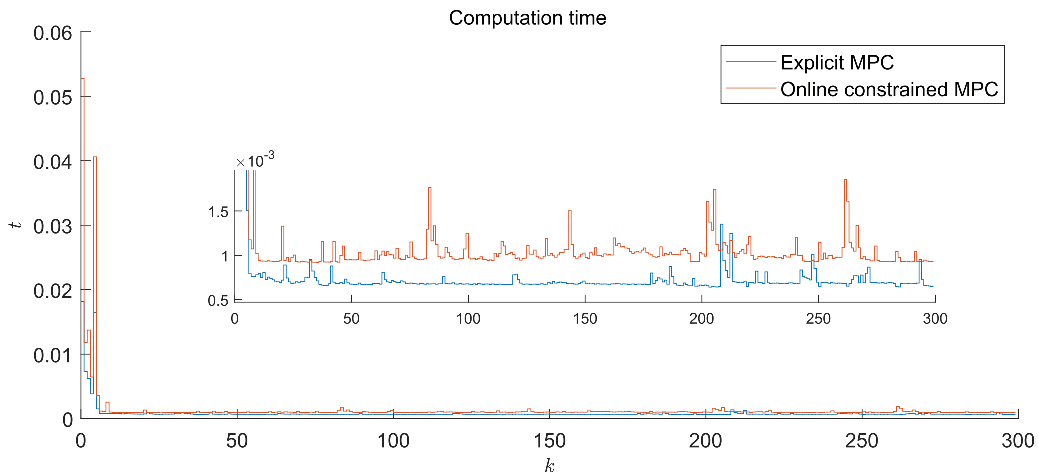


Figure 7: Computation time comparison, $N = 10$

(e) Find the value of N for which on-line MPC is faster or explicit MPC fails to return a solution in the off-line calculation of the explicit solution.

Can you think of a reason why on-line MPC would be preferred to explicit MPC, even when explicit MPC is feasible and faster?

Solution: On-line MPC is faster for large prediction horizon N. The explicit MPC is more sensitive to dimension of the system and prediction horizon. The break-even point in computation time is $N = 25$, but this may differ per implementation and computer. See figure 8.

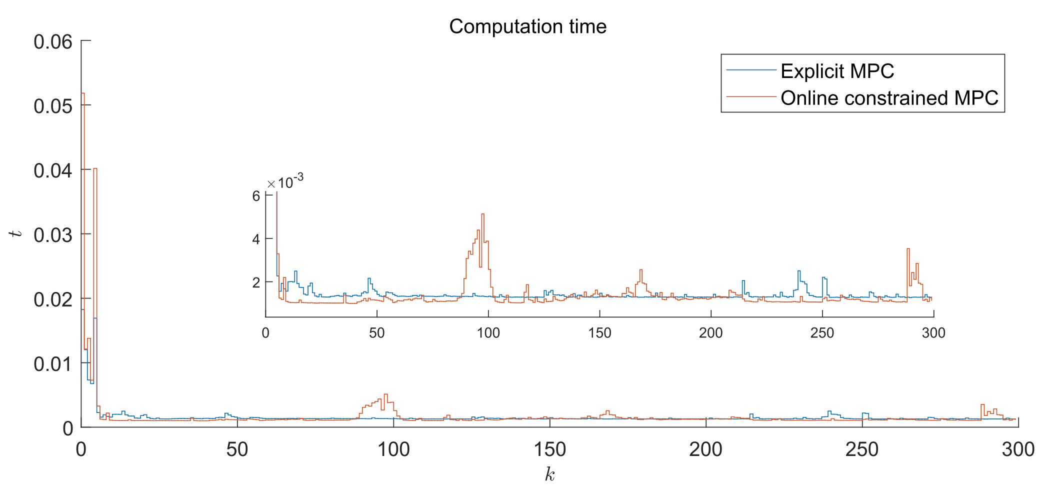


Figure 8: Computation time comparison, $N = 25$

(f) For the same value of N found at the previous point, think how you can redesign the MPC algorithm such that the MPC QP problem can be solved faster.

Solution: As an option, there is move blocking. It reduces the number of optimization variables by reducing the control horizon. It is implemented by introducing equality constraints on input.

2.4 (Optional) MPC formulation using YALMIP

Try to get familiar and formulate the constrained MPC using YALMIP (`https://yalmip.github.io/example/standardmpc`). We will use YALMIP to solve the nonlinear MPC problem in Lecture 6. YALMIP is not allowed for homework assignment to solve the linear MPC problem.