SSY281 Model Predictive Control 2023
PSS 1 - Receding Horizon Control (RHC)

Rémi Lacombe

January 31st, 2023

Course reminder

Consider the quadratic cost function:

$$V_N(\boldsymbol{x}, \boldsymbol{u}) = x(N)^\top P_f x(N) + \sum_{k=0}^{N-1} x(k)^\top Q x(k) + u(k)^\top R u(k)$$

• The unconstrained RHC, or linear-quadratic (LQ) control problem, can be written:

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{u}} \quad & V_N(\boldsymbol{x}, \boldsymbol{u}) \\ \text{s.t.} \quad & x(k+1) = Ax(k) + Bu(k), \qquad k = 0, ..., N-1. \end{aligned}$$

Recall that an *explicit* control law can be obtained for the LQ problem (DP or batch approach, cf Assignment 1).

• The RHC with constraints (or LQ MPC) can be written as:

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{u}} \quad & V_N(\boldsymbol{x}, \boldsymbol{u}) \\ \text{s.t.} \quad & x(k+1) = Ax(k) + Bu(k), \qquad k = 0, ..., N-1 \\ & F\boldsymbol{x} + G\boldsymbol{u} \leq h. \end{aligned}$$

Finding an explicit control law for this class of problems can quickly become very involved, as will be explored later in the course when introducing the concept of explicit MPC. Instead, an *implicit* control law can be obtained by *solving* an optimization problem. This is the focus of today's session.

• Receding horizon idea: once the sequence of optimal control inputs $(u^*(0), ..., u^*(N-1))^\top$ has been computed, only the first input $u^*(0)$ is applied (making the system transition to: $x(1) = Ax(0) + Bu^*(0)$). The horizon is then shifted one step forward and the procedure is repeated.

Tasks

The goal of this session is to design a RHC controller with an *implicit* control law for both the unconstrained and constrained cases of the test system:

$$
\begin{aligned}
x_1(k+1) &= 1.0025x_1(k) + 0.1001x_2(k) + 0.005u(k), \\
x_2(k+1) &= 0.05x_1(k) + 1.0025x_2(k) + 0.1001u(k),
\end{aligned}
$$

with $Q = P_f = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$ and $R = 0.5$.

To do so, we will use the `quadprog` function in Matlab, which can be used to solve quadratic programs (QP). The general syntax of this function is: x = `quadprog`$(H, f, A_{\text{in}}, b_{\text{in}}, A_{\text{eq}}, b_{\text{eq}})$ to solve problems of the form:

$$
\begin{aligned}
\min_x \quad & \frac{1}{2}x^\top Hx + f^\top x, \\
\text{s.t.} \quad & A_{\text{eq}}x = b_{\text{eq}}, \\
& A_{\text{in}}x \leq b_{\text{in}}.
\end{aligned}
$$

More information at: `https://se.mathworks.com/help/optim/ug/quadprog.html`.

Question 1: Use `quadprog` to compute an implicit control law for the unconstrained RHC of the test system. Some Matlab files can be found on Canvas to get you started. Fill in the file `URHC.m` for this question.

Question 2: Let's now consider box constraints on the states and control inputs. We assume that:

$$
|x_2(k)| \leq 0.5, \text{ and } |u(k)| \leq 0.7, \quad \forall k.
$$

Fill in the file `CRHC.m` to compute the control law for the constrained RHC of the test system.

Question 3: Implement time-domain simulations for either the constrained or unconstrained RHC controller. Try to evaluate how the trajectories change for different values of the model parameters. In particular, try playing around with the values of: N, $x(0)$, R, Q, and the bounds on $x_2(k)$ and $u(k)$.

Can you explain what happens when $N \leq 3$?