Exercises for Course on State-Space Control Systems (SSC)
Albert-Ludwigs-Universität Freiburg – Summer Term 2019

Exercise 11: Model Predictive Control

Prof. Dr. Moritz Diehl, Dr. Dang Doan, Benjamin Stickan, Katrin Baumgärtner

MATLAB Consider again the inverted pendulum (Example 4.4), with nonlinear dynamics

$$\frac{dx}{dt} = f(x,u) = \begin{bmatrix} x_2 \\ \sin x_1 - cx_2 + u\cos x_1 \end{bmatrix}.$$

The states are defined as $x_1 = \theta$ and $x_2 = \dot{\theta}$.

1. In this task, we will linearize the system and compare the trajectories of the nonlinear and linearized dynamics.

   (a) Write a MATLAB function `ode(t,x,u)` that defines the ODE model. Choose $c = 0.2$.

   (b) Linearize the system around $\bar{x} = [0,0]^\top$ and $\bar{u} = 0$ to get the approximate system

   $$\frac{dx}{dt} \approx f(\bar{x},\bar{u}) + \underbrace{\frac{\partial f}{\partial x}(\bar{x},\bar{u})}_{A}(x-\bar{x}) + \underbrace{\frac{\partial f}{\partial u}(\bar{x},\bar{u})}_{B}(u-\bar{u}).$$

   (c) Write a MATLAB function `ode_lin(t,x,u)` that defines the linearized ODE model

   (d) Simulate and compare both systems on the time inverval $T = [0,2]\,s$ using the MATLAB function `ode45`. Set the initial state to $x_0 = [\frac{\pi}{3}, 0]^\top$.

2. Now, we are going to design a discrete-time LQR controller for the linearized system.

   (a) Set up the linearized state-space system of the inverted pendulum using MATLAB's `ss` function and discretize it with `c2d`. Use a sampling time of $T_\text{s} = 0.1\,\text{s}$.

   (b) Design a discrete time LQR controller with the `dlqr` function. Use

   $$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad \text{and} \quad R = 0.01$$

   (c) Simulate and plot the discretized closed-loop system for $N = 20$ steps. The initial state is $x_0 = [\frac{\pi}{3}, 0]^\top$.
   Hint: Use the linearized system and input matrices.

3. We will construct an unconstrained MPC problem for the linearized inverted pendulum with a prediction horizon of $N_\text{h} = 20$. The initial state is $x_0 = [\frac{\pi}{3}, 0]^\top$.

   (a) Implement a cost function `cost_function(Z,Q,R,P,N)`, which defines the total cost over the prediction horizon. All optimization variables are collected in a single vector

   $$Z = \begin{bmatrix} x_0 & u_0 & x_1 & u_1 & \dots & x_{N-1} & u_{N-1} & x_N \end{bmatrix}^\top.$$

   Use the infinite horizon LQR cost as final state penalty P.
   Hint: Use the `dlqr` command to obtain P.

   (b) Implement the equality constraints matrix Aeq and vector beq arising from the system dynamics.

   (c) Solve the problem with `fmincon` and simulate the state trajectories for the optimal control inputs applied to the linearized system.

4. Add inequality constraints `lb` and `ub`, such that $|x_2| \leq 0.8$. Solve the problem with `fmincon` and simulate the state trajectories for the optimal controls applied to the linearized system.

5. Simulate the **nonlinear** system with the optimal control trajectory from exercise (4). Start at $x_0 = [\frac{\pi}{3}, 0]^\top$.

6. (a) Write a function `[c, ceq] = nonl_constr(Z, N, Ts, nx, nu, x0)`, which implements the nonlinear system dynamics. Here, $nx = 2$ denotes the number of states and $nu = 1$ the number of controls.
   Hint: You can use the `ode45` function inside this function to implement

   $$x(k+1) - f_d\left(x(k), u(k)\right) = 0,$$

   where $f_d\left(x(k), u(k)\right)$ denotes a discrete time integration of the nonlinear system.

   (b) Solve the **nonlinear** optimization problem with `fmincon` and simulate the state trajectories for the optimal controls applied to the nonlinear system.