DISC Course on Modeling and Control of Hybrid Systems 2017

Homework for Chapters 6 and 7

Exercise 1

Consider the PWA system (2.30)–(2.34) of page 32 of the lecture notes and the equivalent MLD system given by equations (2.35)–(2.36) on page 33 of the lecture notes.

The aim of this exercise is to design an optimal controller and an MPC controller for this system that aims to steer the system from the initial state $x_0 = [-3 \ \ 2]^T$ towards the equilibrium state $x_{\text{eq}} = [0 \ 0]^T$ in 12 time steps.
For both controllers we do **not** use the end-point condition (6.21) given on page 111 of the lecture notes.
For the MPC controller we do use the control horizon constraint of page 111 of the lecture notes.

Tasks:

1. Determine equilibrium values for u, y, δ, and z that are consistent with $x_{\text{eq}} = [0 \ 0]^T$. Are these values unique?

2. First we consider the optimal control problem over the period [0,12] (i.e., the sample step counter k takes its values in the set $\{0, 1, \ldots, 12\}$). We use the objective function given on page 111 of the lecture notes — with the 1-norm as norm if you do not have an MIQP solver available or if you cannot install the demo version of the TOMLAB cplex toolbox (see below), and the 2-norm otherwise — with $Q_x = I$, $Q_u = 5$, $Q_y = 0$, $Q_\delta = 0.03$, $Q_z = 0.02I$ and with u_{eq}, y_{eq}, δ_{eq}, and z_{eq} determined in the previous step (Task 1 above). Note that if we apply MPC with $N_{\text{p}} = N_{\text{c}} = 12$ we are in fact solving the optimal control problem.
Write a matlab program to compute the optimal control inputs and make a plot of the evolution of the (open-loop) controlled system in the phase plane (x_1–x_2) and of the evolution of x, u, z, and δ over time.

   In order to solve this problem you will need an MILP or MIQP solver. If you do not have the Matlab NAG toolbox or another MIQP solver such as Gurobi, CPLEX, XPRESS, MOSEK, . . . , you could use one of the following options:

   - use the `cplex` command of the demo version of the TOMLAB cplex toolbox (see `http://tomopt.com/tomlab/`, item Download). With this command you can solve MIQP problems (see also the `miqp_solve_tomlab_cplex` script on the course website).
   - use the `miqp` command of the Hybrid Toolbox for Matlab (see `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox/`)
   - use the solvers provided by the MPT toolbox (see `http://control.ee.ethz.ch/~mpt/`).

3. Now we consider an MPC controller for the above system. Write a matlab program to compute the MPC controller in *closed-loop* (i.e., use the receding horizon approach in which at each step the optimal MPC control input is recomputed).
   Select two different combinations $(N_{\mathrm{p},1}, N_{\mathrm{c},1})$ and $(N_{\mathrm{p},2}, N_{\mathrm{c},2})$ of N_{p} and N_{c} with $N_{\mathrm{p},i} \in \{4,5,6\}$ and $1 < N_{\mathrm{c},i} < N_{\mathrm{p},i}$ and for each combination run your program and make a plot of the evolution of the controlled closed-loop system in the phase plane (x_1–x_2) and of the evolution of x, u, z, and δ over time.
   Compare the two obtained trajectories with the optimal control trajectory (cf. Task 1 above) and discuss the differences, if any.

Notes & hints:

- As we want you to get some insight into the hybrid MPC method and the relation with integer programming, you are *not* allowed to solve the entire exercise using the `mpt_control` or `mpt_optControl` commands (or related commands) of the MPT toolbox or the Hybrid Toolbox. However, feel free to compare the results obtained with your own programs to those obtained with the MPT toolbox or the Hybrid Toolbox , and to discuss the differences, if any.

Exercise 2

One of the steps used in transforming the MPC-MMPS problem into solving a set of linear programming problems at each sample step consists in rewriting MMPS functions in their canonical form and removing redundant terms. Hence, given two MMPS functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ we have to be able to check that f and g are identical or not.

Task:

Devise an efficient method to verify whether two MMPS functions f and g are identical, i.e., $f(x) = g(x)$ for all $x \in \mathbb{R}^n$.
Note that we do not assume that f or g are in canonical form. In addition, they may have several levels of min-max nestings.

Hint:

You may want to use the equivalence between MMPS and continuous PWA functions as part of your method.

Exercise 3

Construct a forward reachility algorithm for transition systems and prove that for finite-state machines the algorithm terminates in a finite number of steps.

Exercise 4

Consider the finite transition system T with set of states $S = \{q_1, q_2, \ldots, q_{10}\}$ represented in Figure 1 with the set of final states not yet indicated.

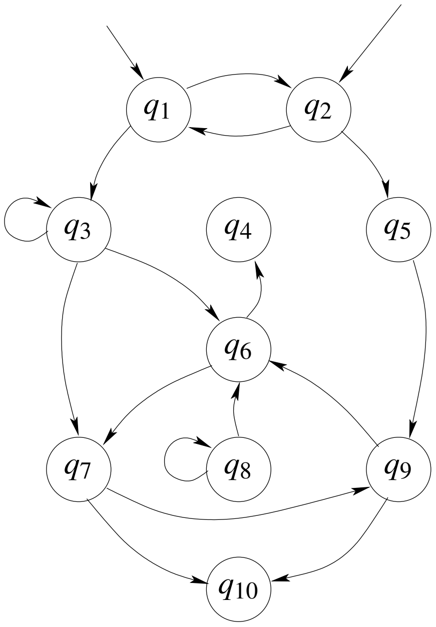

Figure 1: Finite transition system (with set of final states not yet indicated.

Questions (***motivate/prove each answer!***)

a) Give a set of final states that is *not* reachable.

b) Give a set of final states that is reachable.

c) Give a collection of subsets of S that is *not* a partition.

d) Give a collection of subsets of S that is a partition.

Now assume that the set S_F of final states of the transition system is given by $S_F = \{q_4, q_{10}\}$.

e) Is the partition you have given in item e) a bisimulation? Why (not)?

f) Give a bisimulation of T with the minimum possible number of sets.

g) Give a bisimulation of T with the maximum possible number of sets.

Upload your answers for this homework as a pdf file to the DISC website.

Deadline: March 15, 2017