DISC Course on Modeling and Control of Hybrid Systems

Homework for Chapters 6 and 7

This homework set contains two parts: the first one contains a number of exercises that should help you better grasp some of the theory laid out during the lectures; the second part is a mini-project that should help you gain hands-on experience with some of the tools available in the scientific community.

The total number of points on which the assignment is evaluated is 20, however Part I and Part II add up to a total of 30 points, which means that you don't necessarily need to solve the whole homework to obtain a full grade. You may thus choose to solve a simpler mini-project or skip some of the theoretical exercises.

Part I: Theory

Question 1: Optimization (8p)

Consider the following PWA system:

$$x(k) = \begin{cases} x(k-1) + 2u(k) + 1 & \text{if } \alpha x(k-1) + \beta u(k) \leqslant \gamma \\ u(k) + 3 & \text{if } \alpha x(k-1) + \beta u(k) > \gamma \end{cases}$$

$$y(k) = x(k) + u(k)$$

with $|u(k)| \leq 2$ for all k.

1. **(1p)** Compute the right values of α, β, and γ to make the system a continuous PWA system.
2. **(2p)** Rewrite the system in the form of an MMPS model.
3. **(2p)** Formulate the optimization problem to be solved in order to minimize the norm of the output $|y(k)|$, while keeping the control input variation bounded: $|u(k)-u(k-1)| \leq 1$, $\forall k$. Select for the reformulation some finite horizon.
4. **(3p)** Construct the set of LP problems one could employ at each iteration of an MPC solution to the problem at hand.

Question 2: (bi)-Simulation relations (8p)

Consider the system Σ depicted in figure , ignoring the dashed transitions, and assume that $d(H(q_i), H(q_j)) \leq \epsilon$ if and only if $|i-j| \leq 1$.

1. **(1p)** Determine the set of initial states S_0 and the set of *non*-reachable states of Σ.
2. **(1p)** Provide a discrete abstraction Σ_a ϵ-approximately simulating Σ and containing no more than 3 states.

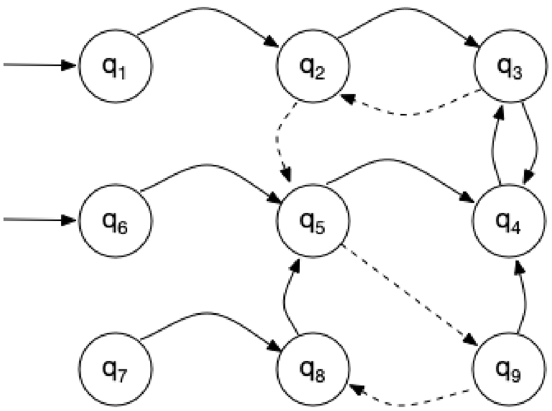

Figure 1: Discrete-state systems.

3. **(1p)** Are the two systems also ϵ-approximately *alternatingly* bisimilar?

4. **(2p)** If additionally we know that $d(H(q_2), H(q_5)) \leq \epsilon$, can you construct an abstraction with no more than 3 states that is ϵ-approximately alternatingly bisimilar to Σ?

Assume now that $d(H(q_i), H(q_j)) \leq 0$ if and only if $i \leq j \leq 3$, and include in the system description the dashed transitions.

5. **(1p)** Can you find now an abstraction exactly simulating, i.e. $\epsilon = 0$, the system Σ containing no more than 6 states? Provide such an abstraction or properly justify why it is not possible to find one satisfying the requirements.

6. **(2p)** Can you find now an abstraction exactly *bisimulating*, i.e. $\epsilon = 0$, the system Σ? Provide the smallest such an abstraction or properly justify why it is not possible to find one satisfying the requirements.

Question 3: Exact abstractions (4p).

Consider the autonomous discrete-time system:

$$x(k+1) = \begin{bmatrix} \min\{4, x_1 + 1\} \\ \min\{4, x_2 + 1\} \end{bmatrix},$$

with x restricted to live in the set $[0,\ 4] \times [0,\ 4]$. The output of this system $H : [0,\ 4] \times [0,\ 4] \rightarrow \{A, B\}$ is given by:

$$y(k) = \begin{cases} A & \text{if } x_1 \geq 2 \wedge x_2 \geq 2 \\ B & \text{else} \end{cases} \tag{1}$$

1. **(3p)** Construct an exact bisimilar abstraction of this system.

2. **(1p)** What is the size of the minimum discrete exact abstraction for this system?

Part II: Practice

(10p) In this part you'll revise the hybrid system you described in the first homework. The main idea is that you try to use at least one of the software toolboxes available for verification and control synthesis of hybrid systems. You are free to select the exact problem you want to solve. Most of the tools have some limitations on the size of problems they can address, thus you may have to simplify your original problem so that a simpler version can actually be solved. The following steps below can be used as a guideline of what you should report and how your project will be graded. The complexity of the problem solved will be taken into account in the grading.

1. **(1p)** Formulate a verification or control problem for the system at hand.
2. **(1p)** Simplify your model so that it results in continuous dynamics with no more than 3 states ($x \in \mathbb{R}^3$).
3. **(1p)** Look at the list of software provided in the lecture slides, justify the selection of a tool from the list to solve the problem you just formulated.
4. **(5p)** Solve the verification/control problem you specified earlier.
5. **(2p)** Provide some simulations illustrating your design or verification (it maybe a falsification of a property too).

As a simple guideline to choose an appropriate software tool is given below:

- UPPAAL - for simple verification problems on timed automata
- UPPAAL-TIGA - for synthesis of controllers for timed automata
- MPT or Hybrid MPC Toolbox - to synthesizex MPC controllers for PWA systems
- SCOTS - to solve control synthesis problems for linear quantized systems. You can also solve problems with non-linear dynamics but this would require more manual work[1].

These are the tools with which we have some (limited) experience and can provide some support, but you may employ any other tool available. For a thorough list of possible tools look at: `http://hybrid-systems.ieeecss.org/tc-hybrid/tools-hybrid-systems`

---

[1] Contact Manuel Mazo for help if you want to follow this route