SC42100 Networked and Distributed Control Systems

Assignment 2 - Lectures 3 and 4

Instructions:

- The assignments are individual. You may consult and discuss with your colleagues, but you need to provide independent answers.
- You may use Matlab/Python/Julia to solve the assignment. Many questions require you clearly to do so (all practical problems in this set).
- Provide detailed answers, describing the steps you followed.
- Provide clear reports, typed, preferably using LaTex.
- Submit the reports digitally on Brightspace/Peer as indicated on the lectures.
- Include your code, for reproducibility of your results, in your Brightspace submission, and possibly also through a link on your report.
- The code will *only* be used to verify reproducibility in case of doubts. The grading will be performed based on the results described in the report.

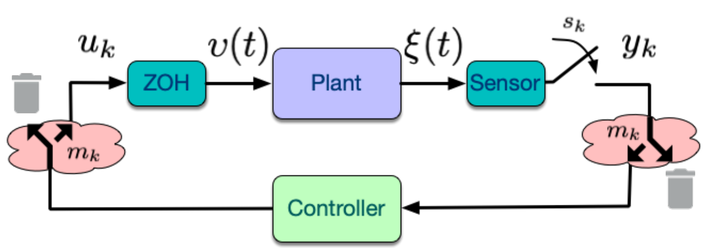

Figure 1: Schematic of a NCS including delays and packet-losses.

Consider a sample-data networked control system (NCS), as depicted in Figure 1. Assume that the plant dynamics are linear time-invariant given by:

$$\dot{\xi}(t) = A\xi(t) + Bv(t),$$

where the control signal v is a piece-wise constant signal resulting from the application of a controller in sampled-and-hold fashion, i.e. $v(t) = u_k,\ t \in [s_k,\ s_{k+1})$.

The system matrices are given by:

$$A = \begin{bmatrix} a-b & 0.5-c \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where a is given by the first digit, b by the 3rd digit, and c by the last digit of your student ID number. **This is the same plant from Assignment 1.** We shall call the control system controlling this plant *System 1*.

We start considering the system with a constant sampling interval $h = s_{k+1} - s_k$ for all k, and assuming no delay is present $\tau = \tau^{sc} + \tau^{c} + \tau^{ca} = 0$. For Questions 1–3, we consider a scenario where another control system, *System 2*, shares the network with System 1. System 2 has three times the sampling interval of System 1, but it has higher priority, which means that whenever it communicates, it may induce packet dropouts in the System 1. This situation is depicted in Figure 2. Pick your preferred stabilizing feedback matrix $\bar{K}$ from Assignment 1. You have to analyze both the **to-hold** and **to-zero** approaches. Questions 1–3 assume different models of this situation, each requiring different analysis approaches.

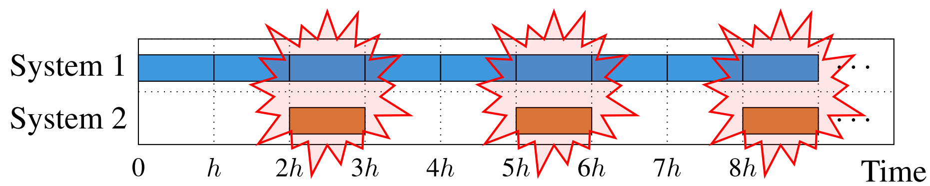

Figure 2: Gantt chart depicting the network time slots used by Systems 1 and 2 for Questions 1–3. The red starbursts indicate packet collisions, potentially causing a loss in the transmission of System 1.

In the first model, we assume that whenever there is a collision, System 1 loses a packet. I.e., the dropout sequence $\mathbf{m}$ is $(001)^\omega$.

Question 1: (4p)

1. (2p) Show one criterion you can use to analyze stability of System 1 in this scenario.
2. (2p) For what range of sampling intervals h is System 1 stable in this scenario? What is best, to zero or to hold?

In the second model, we assume that whenever there is a collision, System 1's packet **may or may not be lost.**

Question 2: (12p)

1. (3p) Create an ω-automaton that models this behavior.
2. (2p) Write down the equations of the switched system describing the dynamics of the received samples using this network model.
3. (2p) Provide LMIs to verify stability of your closed-loop system.
4. (2p) For what range of sampling intervals h is System 1 stable in this scenario? What is best, to zero or to hold?
5. (3p) Is the range you have obtained larger or smaller than the one in Question 1? Comment on the differences: whether they are expected or not, and why.

In the third model, we add probabilities. Assume that whenever there is no collision, the probability of a packet loss is 0.01 (1%); while when there is a collision, the probability of a loss is 0.51 (51%).

Question 3: (12p)

1. (3p) Create a Markov Chain that models this behavior.
2. (3p) Write down the equations of the Markovian Jump Linear System describing the dynamics of the received samples using this network model
3. (2p) Choose a stochastic stability notion and provide LMIs to verify stability of your closed-loop system.
4. (2p) For what range of sampling intervals h is System 1 stable in this scenario? What is best, to zero or to hold?
5. (2p) Comment on the differences between the range you obtained now with the ones in Questions 1 and 2.

Next we consider the case when delays are present in the networked system, but System 1 is no longer subject to packet dropouts. Assume that the system is affected by a **constant** small delay $\tau \in [0, h)$, and it is controlled with the same static controller you designed.

Question 4: (10p)

1. (4p) Employ the so-called Jordan form approach (as in Lecture 4) to construct a polytopic discrete-time model over-approximating the uncertain exact discrete-time closed-loop NCS dynamics. By graphically inspecting the uncertain entries of the obtained matrices, what is the smallest number of vertices the polytope needs? **Note**: This approach would allow you to also deal with time-varying delays.

2. (2p) Perform stability analyses, solving the relevant LMIs resulting from the previous question when the polytopic over-approximation of item 1. Employing this analysis, produce a plot illustrating combinations of (h, τ) as in Question 2 of Assignment 1.

3. (2p) Refine the plot of the previous question, by employing a smaller polytope that over-approximattes the uncertain system, but with more vertices (thus requiring the solution of more LMIs).

4. (2p) Compare the resulting plot with the one from Assignment 1 and discuss any possible differences and the possible causes for such differences.

Finally we go back to the case with no delays to design an event-triggered controller.

Question 5: (12p)

1. (1p) Consider your system being controlled by the first static controller you designed for your continuous-time system. Design a quadratic event-triggered condition for the system guaranteeing global exponential stability of the closed-loop.

2. (4p) Simulate the resulting closed loop for various values of the σ parameter controlling the guaranteed performance of the closed-loop. Simulating for multiple initial conditions for each value of σ, compare the average amount of communications/sampling produced by each of the system on a predefined fixed time-length of the simulations.

3. (1p) Let h^*_{avg} be the largest average inter-sample time you obtained in Item 2. Does periodic sampling with $h = h^*_{\mathrm{avg}}$ stabilize your closed-loop system?

4. (4p) Now consider a Periodic Event-Triggered Control (PETC) implementation: instead of continuously checking the triggering condition, you only check it every $h_{\mathrm{PETC}} := h^*_{\mathrm{avg}}/2$ time units (half the best average inter-sample time obtained in Item 2). Like in Item 2, simulate the resulting closed loop for various values of the σ parameter and initial conditions. Compare the average amount of communications/sampling produced with what was obtained in Item 2.

5. (2p) Provide plots illustrating the closed-loop trajectories of the system's state of your most sample-efficient Continuous ETC (Item 2) and PETC (Item 4) implementations starting from the same initial state. What differences in control performance do you observe? **Note:** Do not forget to include inter-sample behavior.