SC42101 Networked and Distributed Control Systems

Assignment 1 - Lectures 1 and 2 (and beginning of 3)

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 to solve the assignment. Many questions require you clearly to do so.
- Provide detailed answers, describing the steps you followed.
- Provide clear reports, typed, preferably using LaTex.
- Submit the reports digitally on Brightspace as indicated on the lectures.
- Include your code, for reproducibility of your results, in your Brightspace submission on a zip file, or 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.

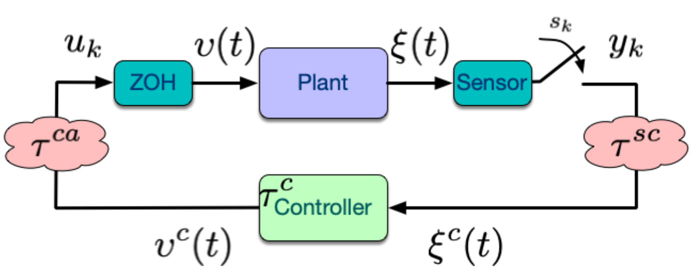

Figure 1: Schematic of a NCS including delays.

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

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

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

The system matrices are given by:

$$A = \begin{bmatrix} 0.3 + a - b & 0.5 - c \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where a is given by the first digit, b by the third digit, and c by the last digit of your student ID number.

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 $\tau = \tau^{sc} + \tau^{c} + \tau^{ca} = 0$.

Important note: For all the questions that follow, you do **NOT** necessarily need to provide analytic expressions, unless explicitly stated. However, you need to describe the steps you took to produce the resulting plots, and design process you followed.

Question 1: (2p)

1. (2p) Design a linear continuous time controller $v(t) = -\bar{K}\xi(t)$ placing the poles of the continuous closed-loop at $-1 + 2j$ and $-1 - 2j$. Construct the exact discrete-time model $x_{k+1} = F(h)x_k + G(h)u_k$, and the closed-loop system resulting from applying the controller in sample-data (piece-wise constant) fashion: $u_k = -\bar{K}x_k$, $x_k := \xi(s_k)$. Provide ranges of sampling times for which the sampled-data system is stable. Comment on what you observe.

Next we consider the case when delays are present in the networked system. Assume that the system is affected by a **constant** small delay $\tau \in [0, h)$, and it is controlled with the *static* controller $\bar{K}$.

Question 2: (4p)

1. (2p) Construct the exact discrete-time model for the NCS with delays and study the combinations of sampling intervals and system delays that result in an asymptotically stable closed-loop. Provide plots illustrating the combinations of (h, τ) retaining stability.

2. (2p) Select a sampling interval h that guarantees stability under no delays. Redesign the controller to improve the robustness against delays, that is, try to design a controller that increases the range of tolerable delays for the selected h. Think about designing now a *dynamic* controller instead.

Next we consider a different approach to deal with delays $\tau \in [0, h)$. Consider a controller that at some time $t = s_{k-1} + \tau^{sc} \in [s_{k-1}, s_k)$ computes $u_{k-1} = \bar{K}x(s_{k-1})$ and $u_{k-2} = \bar{K}x(s_{k-2})$, and both of these inputs are received at a smart actuator at some time $t = s_{k-1} + \tau \in [s_{k-1}, s_k]$. This smart actuator replaces the ZOH block in Figure 1. Consider now the following implementation of the the actuation signal at the smart actuator, a form of *first order hold*:

- The actuator updates the input exactly at time s_k, in a time-driven fashion (not event-based), as in the next bullet.
- At time s_k, $\upsilon(s_k) = u_{k-2}$, and at s_{k+1}, $\upsilon(s_{k+1}^-) = u_{k-1}$. At all times in between, i.e. $t \in [s_k, s_{k+1})$, the input $\upsilon(t)$ is *linearly* interpolated between those two values.

Question 3: (8p)

1. (4p) Derive the exact discrete-time model $x_{k+1}^e = F_1(h)x_k^e + G_1(h)u_k^e$ capturing the dynamics of the described system at sampling instants. Note that you may need to consider extended states x^e and inputs u^e. Provide analytical expressions for $F_1(h)$ and $G_1(h)$, as a function of A, B, and h.
   Hint: You may need to integrate $\int se^{As}ds$, this is easily achievable integrating by parts.
2. (2p) Employing the controller $\bar{K}$ designed in question 1, find the range of values of h for which the system is stable. Compare this controller implementation with the sampled-data (zero order hold) one from Question 2.
3. (2p) Try to design a controller $\tilde{K}$ (possibly dynamic) that extends the range of sampling intervals h for which the system remains stable.

Finally, consider the system is implemented with a varying sampling interval taking two possible values h_1 and h_2. When the sampling interval is h_1 the controller is applied in sample-data fashion (zero order hold, constant input), and when the interval is h_2 the control input is generated as in Question 3.

Question 4: (6p)

1. (2p) Without assuming knowledge of the sequence of sampling intervals employed. Describe the set of LMIs that need to be solved to determine if such a controller guarantees stability for this NCS.
2. (2p) Assume now that the possible sequences of inter-sample intervals is restricted to be either[1]: $(h_1h_2)^\omega$ or $(h_1h_2h_2)^\omega$. Can you simplify the check required to test stability of a given sampled-data controller in this case? e.g. can you reduce the size of the LMIs, if at all needed?

[1]The notation $(a)^\omega$ denotes the infinite repetition of that symbol, in the two cases in the question that means that the sequences end-up repeating forever the last inter-sample time

3. (2p) Under the same scenario as in the previous question. Find values of h_1 and h_2 guaranteeing the stability of the closed loop system. Verify the result through simulations.

We consider now the system from the previous questions, denoted from here on *System 1*, with a constant sampling interval $h = s_{k+1} - s_k$ for all k, and assume no delay is present $\tau = \tau^{sc} + \tau^{c} + \tau^{ca} = 0$. For the following question, we consider a scenario where another control system, *System 2*, shares a network with System 1. System 2 has as system matrices $A_2 = 1/3A$ and $B_2 = B$, and it is implemented with a sampling interval three times the sampling interval of System 1. This means that whenever System 2 communicates it may induce packet dropouts in System 1 or System 2. This situation is depicted in Figure 2.

Employ for System 1 the stabilizing feedback matrix $\bar{K}$ from Question 1, and design a stabilizing feeback matrix $\bar{K}_2$ for System 2. You have to analyze both the **to-hold** and **to-zero** approaches.

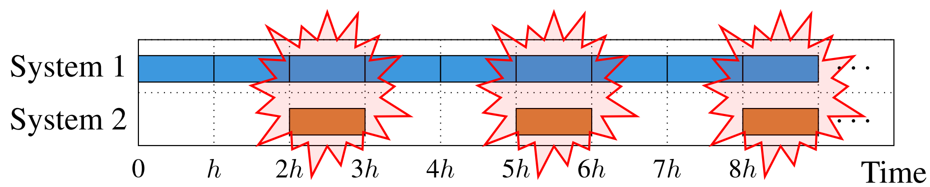

Figure 2: Gantt chart depicting the network time slots used by Systems 1 and 2 for Question 5. The red starbursts indicate packet collisions, potentially causing a loss in the transmission of System 1 and/or System 2.

In the first model, we assume that whenever a collision occurs System 1 and System 2 alternate which one looses a packet. That is, the first time System 2 transmits, System 1 looses a packet, the second time System 2 looses a packet, and so on so forth.

Question 5: (5p)

1. (1p) Model mathematically the described situation.

2. (2p) Show one criterion you can use to analyze stability of Systems 1 and 2 in this scenario.

3. (2p) For what range of sampling intervals h are both systems stable in this scenario? What is best, to zero or to hold?