Networked and Distributed Control: Assignment 1

Niels Stienen, 5595738

I. Question 1

> Design a linear continuous time controller $v(t) = -\bar{K}\xi(t)$ with the continuous closed-loop poles placed at $-1 \pm 2j$. 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.

The LTI plant dynamics and control signal are given by:

$$\dot{\xi}(t) = A\xi(t) + Bv(t) \tag{1}$$
$$v(t) = u_k, t \in [s_k, s_{k+1}) \tag{2}$$

The system matrices with the student ID digits filled in are:

$$A = \begin{bmatrix} -3.7 & -7.5 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3}$$

For now the sampling interval is assumed constant with $h = s_{k+1} - s_k$ for all k, and assuming no delay is present: $\tau = \tau^{sc} + \tau^{c} + \tau^{ca} = 0$.

The controller is designed using the `place` command in MatLab resulting in a control gain $K = \begin{bmatrix} -1.505 & -0.700 \end{bmatrix}$.

The exact discrete-time model is constructed with the equation:

$$x_{k+1} = e^{Ah}x_k + \int_0^h e^{As}Bds u_k \tag{4}$$
$$x_{k+1} = F(h)x_k + G(h)u_k \tag{5}$$

Since the A matrix is invertible, $G(h)$ can be calculated using the identity:

$$G(h) = \int_0^h e^{As}Bds = \left(e^{Ah} - I\right)A^{-1} \tag{6}$$

To find the range of sampling intervals for which the sampled-data system is stable the spectral radius is used (eq. 7). Experimentally, the largest sampling interval for which the system is stable is found to be $h = 0.5118$. Due to the nature of checking a range of sampling intervals, this gives us a lower approximation of the largest sampling time for which the system is stable.

$$\rho(F(h) - G(h)\bar{K}) < 1 \tag{7}$$

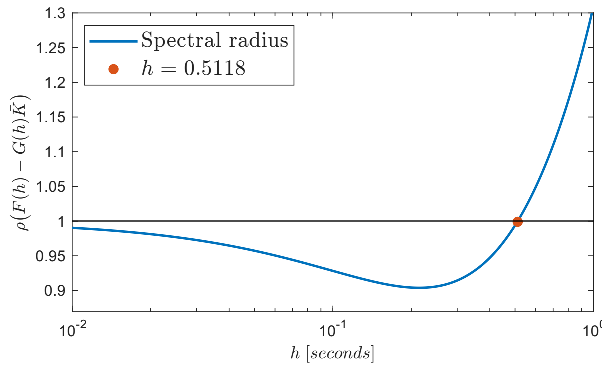


Fig. 1. Plot of the spectral radius for different sampling times

II. Question 2

> Introducing a small constant delay $\tau \in [0, h)$, construct the exact discrete-time model for the NCS and study the combinations of sampling intervals and system delays that result in an asymptotically stable closed-loop.

With the introduction of a constant delay in the range $0 \leq \tau < h$ equation 4 is changed to include this delay which introduces an extra term:

$$x_{k+1} = e^{Ah}x_k + \int_{h-\tau}^{h} e^{As}Bds u_{k-1} + \int_0^{h-\tau} e^{As}Bds u_k \tag{8}$$
$$x_{k+1} = F_x(h)x_k + F_u(h,\tau)u_{k-1} + G_1(h,\tau)u_k \tag{9}$$

Introducing the extended state vector $x_k^e = \begin{bmatrix} x_k^T & u_{k-1}^T \end{bmatrix}$ the system can again be described with a discrete matrix $F(h,\tau)$ and $G(h,\tau)$:

$$F(h,\tau) := \begin{bmatrix} F_x(h) & F_u(h,\tau) \\ \mathbf{0} & 0 \end{bmatrix} \tag{10}$$
$$G(h,\tau) := \begin{bmatrix} G_1(h,\tau) \\ 1 \end{bmatrix} \tag{11}$$

with the new system dynamics:

$$x_{k+1}^e = F(h,\tau)x_k^e + G(h,\tau)u_k \tag{12}$$

Since the delay is constant the spectral radius can be used again giving an area in Figure 2 described by the intersection of $\rho < 1$ and $\tau < h$.

> Select a sampling interval h that guarantees stability under no delays. Redesign the controller to improve robustness against delays.

The sampling interval is chosen to be $h = 0.4s$ as it is towards the upper limit of allowable sampling intervals for the

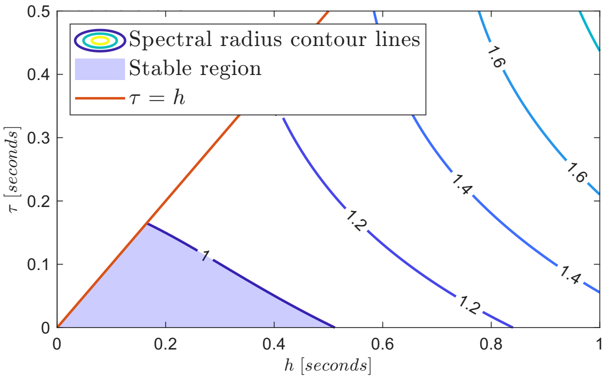


Fig. 2. Contour plot of the spectral radius for different combinations of h and τ, the region above the line $\tau = h$ is empty as the delay is constrained to exclude this region.

system. From here there is significant room for improvement as currently the maximum allowable delay is around 15% of the chosen sampling interval. To increase the range of tolerable delays for the selected h the controller is changed to a dynamic controller.

$$u_k = -\bar{K}x_k \rightarrow u_k = -\bar{K}x_k - K_u u_{k-1} \tag{13}$$

To make the controllers more comparable the same static gain $\bar{K}$ is used for both controllers, the additional term K_u was then found by iterating through values that resulted in significantly more tolerable delays for the selected sampling time. The new controller has improved the maximum allowable delay from around 15% of the sampling interval to the entire range $\tau \in [0, h)$. Interesting to note is that the maximum absolute eigenvalue switches at 3 points caused by the convergence of a complex eigenvalue pair.

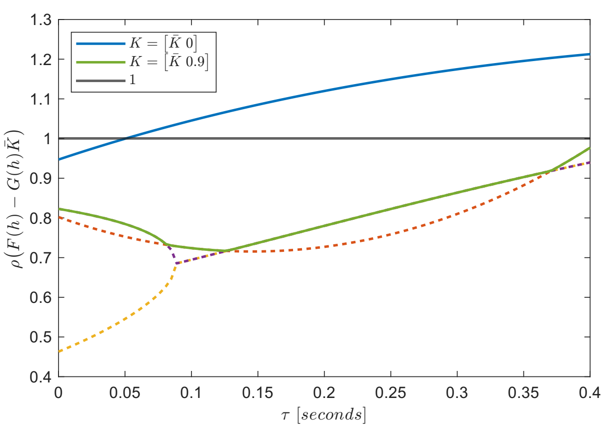


Fig. 3. Comparison of a truly static controller and a controller with a dynamic term

III. Question 3

> Next we consider a form of first order hold to deal with delays $\tau \in [0, h)$. Derive the exact discrete-time model $x^e_{k+1} = F_1(h)x^e_k + G_1(h)u^e_k$ capturing the dynamics of the described system at sampling instants. Provide analytical expressions for $F_1(h)$ and $G_1(h)$ as a function of A, B, h.

An interpolation function is needed to create a first-order hold that interpolates from u_{k-2} to u_{k-1}. The function used is of the form $ax + b$ where a is the slope and b the intercept resulting in:

$$v(t) = u_{k-2} + \frac{(t - s_k)(u_{k-1} - u_{k-2})}{(s_{k+1} - s_k)} \tag{14}$$

Filling into the equation:

$$x_{k+1} = e^{Ah}x_k + \int_{s_k}^{s_{k+1}} e^{A(s_{k+1}-s)}Bv(s)ds \tag{15}$$

After applying a change in variables ($s' = s_{k+1} - s$) and simplifying this results in the equation (Note, from here on out s' is denoted by s again):

$$\begin{aligned} x_{k+1} = e^{Ah}x_k + \int_0^h e^{As}Bds u_{k-1} \\ + \int_0^h se^{As}Bds\frac{u_{k-2} - u_{k-1}}{h} \end{aligned} \tag{16}$$

The second term has an extra s term which can be accounted for by applying integration by parts:

$$\int_0^h se^{As}ds = sA^{-1}e^{As}\Big|_0^h - \int_0^h A^{-1}e^{As}ds \tag{17}$$

Filling the result from integration by parts in equation 16 and simplifying further results in the final equation from which the terms of the extended system are derived:

$$x_{k+1} = F_x(h) + F_{u_1}(h)u_{k-1} + F_{u_2}(h)u_{k-2} \tag{18}$$

$$x^e_k = \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} \tag{19}$$

$$x^e_{k+1} = \underbrace{\begin{bmatrix} F_x(h) & F_{u_1}(h) & F_{u_2}(h) \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix}}_{F_1(h)} x^e_k + \underbrace{\begin{bmatrix} \mathbf{0} \\ 1 \\ 0 \end{bmatrix}}_{G_1(h)} u^e_k \tag{20}$$

with

$$F_x(h) = e^{Ah} \tag{21}$$

$$F_{u_1}(h) = -A^{-1}B + \frac{A^{-1}\left(e^{Ah} - I\right)A^{-1}B}{h} \tag{22}$$

$$F_{u_2}(h) = A^{-1}e^{Ah}B - \frac{A^{-1}\left(e^{Ah} - I\right)A^{-1}B}{h} \tag{23}$$

> 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 one from Question 2.

The range of values of h for which the system is stable is smaller than the range found in Question 2. This is to be expected as this controller implementation is delayed by a step of size h per definition. Unlike the previous implementation, the size of the delay does not matter as long as it is less than the time step h.

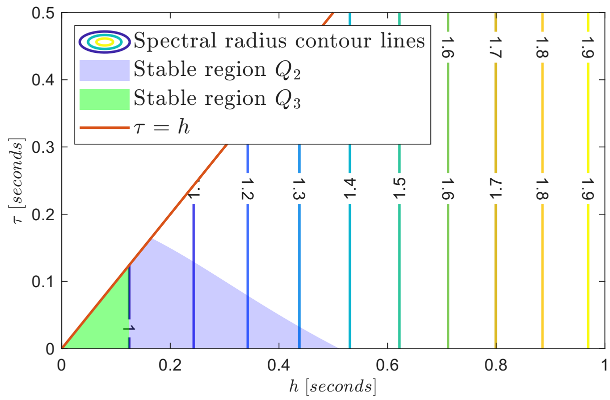


Fig. 4. Comparison of the ranges of h for which the closed loop is stable. Note that the stable region from Q_2 includes the region depicted by Q_3

Try to design a controller $\tilde{K}$ that extends the range of sampling intervals h for which the system remains stable.

A new controller was designed by keeping the influence of older input terms in reducing order. This greatly improved the range of h for which the system is stable.

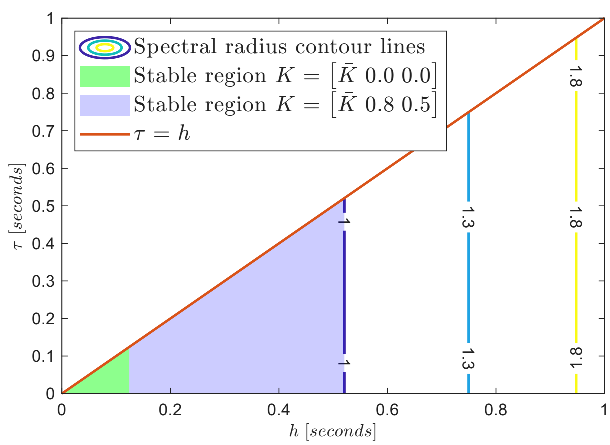


Fig. 5. Comparison of a truly static controller and a controller with a dynamic term

IV. Question 4

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, and when the interval is h_2 the control input is generated as in Question 3.
Without assuming knowledge of the sequence of sampling intervals employed. Describe the set of LMI's that need to be solved to determine if such a controller guarantees stability for this NCS.

We have two different situations that we need to consider with the LMI's. The first being the system from the zero order hold method and the second from the first order hold method. In both cases, the delay is assumed to be a known constant. This assumption was checked with a TA as without this assumption we have an infinite set of LMI's which is covered in lecture 4 and thus not the goal for this assignment yet.

To describe the set of LMI's needed both systems must have the same state space. The zero order hold model is an $n+m$ dimensional model:

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} F_x(h) & F_u(h,\tau) \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} G_1(h,\tau) \\ 1 \end{bmatrix} u_k \quad (24)$$

while the other case has $n+2m$ states:

$$\begin{bmatrix} x_{k+1} \\ u_k \\ u_{k-1} \end{bmatrix} = \begin{bmatrix} F_x(h) & F_{u_1}(h) & F_{u_2}(h) \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \\ 0 \end{bmatrix} u_k \quad (25)$$

In the zero order hold case the input from two steps before u_{k-2} does not affect the state x. However, adding it to the matrix it does enable us to add an extra state such that both systems have the same state space:

$$\begin{bmatrix} x_{k+1} \\ u_k \\ u_{k-1} \end{bmatrix} = \begin{bmatrix} F_x(h) & F_u(h,\tau) & \mathbf{0} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} + \begin{bmatrix} G_1(h,\tau) \\ 1 \\ 0 \end{bmatrix} u_k \quad (26)$$

The combined system is then defined as:

$$x^e_{k+1} = F(h,\tau)x^e_k + G(h,\tau)u_k \quad (27)$$

where

$$x^e_k = \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} \quad (28)$$

$$F(h,\tau) = \begin{cases} \begin{bmatrix} F_x(h) & F_u(h,\tau) & \mathbf{0} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix}, & h = h_1 \\ \begin{bmatrix} F_x(h) & F_{u_1}(h) & F_{u_2}(h) \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix}, & h = h_2 \end{cases} \quad (29)$$

$$G(h,\tau) = \begin{cases} \begin{bmatrix} G_1(h,\tau) \\ 1 \\ 0 \end{bmatrix}, & h = h_1 \\ \begin{bmatrix} \mathbf{0} \\ 1 \\ 0 \end{bmatrix}, & h = h_2 \end{cases} \quad (30)$$

To verify stability a common Lyapunov function needs to be found. For a given controller K, the closed-loop system is defined as

$$A_{cl}(h,\tau) := F(h,\tau) - G(h,\tau)K \quad (31)$$

The LMI set to verify stability is then

$$P, Q \succ 0 \tag{32}$$
$$A_{cl}(h_1,\tau)^T P A_{cl}(h_1,\tau) - P \preceq 0, \text{ for } \tau = C \tag{33}$$
$$A_{cl}(h_2)^T P A_{cl}(h_2) - P \preceq 0 \tag{34}$$

Assume now that the possible sequences of inter-sample intervals is restricted to be either: $(h_1h_2)^\omega$ or $(h_1h_2h_2)^\omega$. Can you simplify the check required to test stability of a given sampled-data controller in this case?

Since the sequence is known the set of LMI's needed can be reduced significantly. We can analyse the following discrete-time system for a periodic sequence with A_{cl} defined in 31.

$$x^e_{k+2} = A_{cl}(h_2)A_{cl}(h_1,\tau)x^e_k \tag{35}$$

Defining $A_{h_{12}} = A_{cl}(h_2)A_{cl}(h_1,\tau)$ gives a simple LMI system which is equivalent to the spectral radius condition:

$$P \succ 0,\ A^T_{h_{12}} P A_{h_{12}} - P \prec 0 \tag{36}$$
$$\rho(A_{h_{12}}) < 1 \tag{37}$$

For the other sequence we can follow these exact same steps resulting in checking the discrete-time system:

$$x^e_{k+3} = A_{cl}(h_2)A_{cl}(h_2)A_{cl}(h_1,\tau)x^e_k \tag{38}$$

Defining $A_{h_{122}} = A_{cl}(h_2)A_{cl}(h_2)A_{cl}(h_1,\tau)$ gives a simple LMI system which is equivalent to the spectral radius condition:

$$P \succ 0,\ A^T_{h_{122}} P A_{h_{122}} - P \prec 0 \tag{39}$$
$$\rho(A_{h_{122}}) < 1 \tag{40}$$

If the spectral radius condition holds for both of these sequences we can guarantee stability of the system in this situation.

Under the same scenario as the previous question. Find values of h_1 and h_2 guaranteeing stability of the closed loop system.

To find combinations of h_1 and h_2 that guarantee stability of the closed loop system a grid search was used that checked the spectral radius of both closed loop sequences defined in 4.2. Figure 6 shows the region where both sequences had a spectral radius less than 1 for a certain combination of sampling intervals. From this, any tuple of sampling intervals can be chosen to use and verify the stability of the closed loop system.

To verify the stability of the closed loop system several combinations of (h_1, h_2) were simulated. Figure 7 shows one such simulation. If one wants to see simulations of other combinations they can be found by opening the file in the linked github repository and changing the value of h in the section "question 4.3" [1].

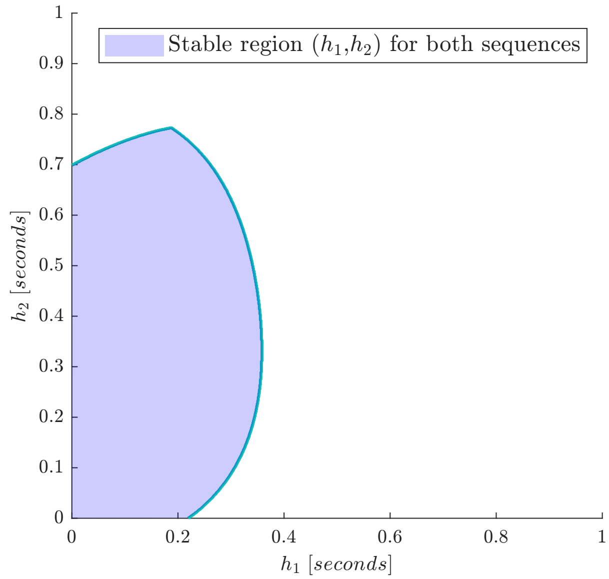
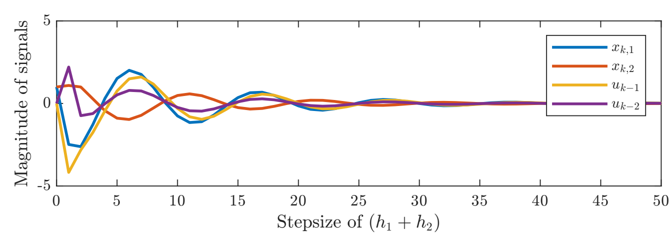


Fig. 6. Inner approximation of all tuples (h_1,h_2) that guarantee stability for the previously described system.



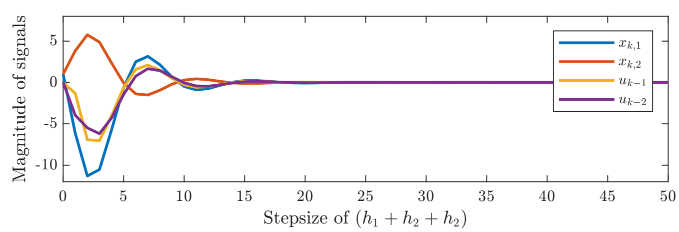


Fig. 7. Simulations of (1) the sequence $(h_1h_2)^\omega$, (2) the sequence $(h_1h_2h_2)^\omega$ both showing stability of the closed-loop system. $(h_1, h_2) = (0.2, 0.5)$

V. Question 5

We now consider the system (System 1) with a constant sampling interval and no delay present. 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 $A_2 = \frac{1}{3}A$ and $B_2 = B$, and it is implemented with a sampling interval three times the sampling interval of System 1. Employ for System 1 the stabilizing feedback matrix $\bar{K}$ from Question 1, and design a stabilizing feedback matrix $\bar{K}_2$ for System 2. You have to analyze both the to-hold and to-zero approaches.
Model mathematically the described situation.

There is no delay present and both systems can be described as a sampled-data system with periodic sampling.

$$x_{k+1} = F(h)x_K + G(h)u_k \tag{41}$$

The matrices $F(h)$ and $G(h)$ depend on whether there is a packet loss and whether to-zero or to-hold is used. The to-zero mechanism sets $u_k = 0$ if a packet is dropped and $u_k = -Kx_k$ when the transmission is successful. This results in the closed loop dynamics:

$$x_{k+1} = \begin{cases} (F(h) - G(h)K)x_k, & \text{if } p_k = 1 \\ F(h)x_k, & \text{if } p_k = 0 \end{cases} \tag{42}$$

where p_k indicates whether the packet arrived ($p_k = 1$) or is dropped ($p_k = 0$).
The to-hold mechanism meanwhile repeats the previous control input if a packet is dropped, $u_k = u_{k-1}$, and $u_k = -Kx_k$ when the transmission is successful. Due to the possibility of needing a previous control input the state is augmented as in previous questions resulting in the closed loop dynamics:

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{cases} \begin{bmatrix} F(h) - G(h)K & 0 \\ -K & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}, & \text{if } p_k = 1 \\ \begin{bmatrix} F(h) & G(h) \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}, & \text{if } p_k = 0 \end{cases} \tag{43}$$

Note that the above $F(h)$, $G(h)$, and K are just generic placeholders that change depending on which system is being analyzed. To simplify future notations the following short-hand symbols are defined for the closed loop dynamics, $A_p^z, A_{np}^z, A_p^h, A_{np}^h$ meaning to-zero with packet, to-zero with no packet, to-hold with packet, and to-hold with no packet respectively. In both the to-zero and the to-hold case the packet drop sequence can be modeled as a chained system. For system 1 every 6th packet is dropped while for system 2 every 2nd packet is dropped:

$$x_{k+6} = A_p^i A_p^i A_p^i A_{np}^i A_p^i A_p^i x_k,\ i \in \{h, z\} \tag{44}$$
$$x_{k+2} = A_{np}^i A_p^i x_k,\ i \in \{h, z\} \tag{45}$$
$$\tag{46}$$

The controller K_2 has been designed with pole placement just as in Question 1 with the same desired poles. Thus $K_1 = \begin{bmatrix} -1.505 & -0.700 \end{bmatrix}$ and $K_2 = \begin{bmatrix} -1.622 & 1.100 \end{bmatrix}$.

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

Since both systems drop packets in a periodic nature it is sufficient to check whether the spectral radius over the sequence is smaller than 1 for stability. Thus the following situations will be checked for a range of sampling intervals h:

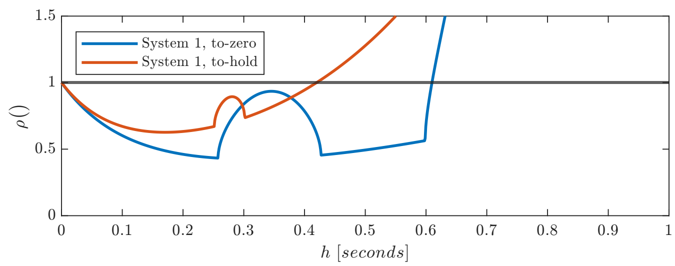
$$\rho(A_p^z A_p^z A_p^z A_{np}^z A_p^z A_p^z) < 1 \tag{47}$$
$$\rho(A_{np}^z A_p^z) < 1 \tag{48}$$
$$\rho(A_p^h A_p^h A_p^h A_{np}^h A_p^h A_p^h) < 1 \tag{49}$$
$$\rho(A_{np}^h A_p^h) < 1 \tag{50}$$

For what range of sampling intervals h are both systems stable in this scenario? What is best, to zero or to hold?

Checking the spectral radius of both systems with both approaches for a range of sampling intervals h results in the following ranges for stability:

System:	Approach:	Range of stable intervals:
System 1	to-zero	$h \in (0, 0.6096)$
System 1	to-hold	$h \in (0, 0.4194)$
System 2	to-zero	$h \in (0, 0.7087)$
System 2	to-hold	$h \in (0, 0.4204)$
Combined	to-zero	$h \in (0, 0.2362)$
Combined	to-hold	$h \in (0, 0.1401)$

From the above table it is obvious that in this case the to-zero approach is better as it has a greater range of sampling intervals for which not only the combined system but also both individual systems are stable.



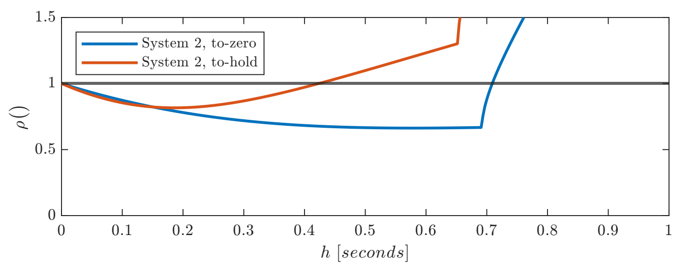


Fig. 8. Spectral radius of System 1 and System 2 for different sampling interval sizes h for both the to-zero and to-hold approach.

REFERENCES

[1] N. Stienen, "Networked and distributed control assignments," 2024. [Online]. Available: https://github.com/StienenNiels/NetworkedControl