Assignment 1

Networked and Distributed Control - SC42101

Joshua C.S. Amato (4688368)

Q4 2023/2024

Introduction

We consider a sample-data networked control system (NCS) as depicted in figure 1.

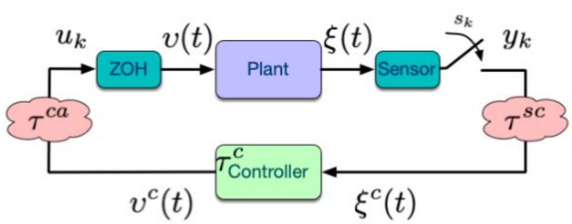

Figure 1: Schematic figure of an NCS including delays

The continuous time LTI plant dynamics are given by:

$$\begin{aligned} \dot{\zeta}(t) &= A\zeta(t) + Bu(t) \\ A &= \begin{bmatrix} -3.7 & -7.5 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{1}$$

Then we can verify that $rank(conmat(A, B)) = 2$, hence the continuous-time system is controllable.

The computations and algorithms that follow in this report are performed and implemented respectively using MATLAB.

Q1: Stability of inter-sampling times

We firstly consider the design of continuous time static controller with linear feedback law $u = -\bar{K}\zeta(t)$. By using the MATLAB command `place()` with desired pole locations $-1 + 2j$ and $-1 - 2j$ the resulting controller amounts to:

$$\bar{K} = \begin{bmatrix} -1.5053 & -0.7000 \end{bmatrix}$$

Subsequently, we can define the discrete-time equations for a constant inter-sampling time without delays:

$$\begin{aligned} x_{k+1} &= e^{Ah}x_k + \int_0^h e^{As}Bu_k\,ds \\ &= F(h) + G(h)u_k \end{aligned} \tag{2}$$

where h is the constant inter-sampling time and u_k is a piecewise constant control signal such that $u(t) = u(k)$ for $t \in [s_k,\ s_{k+1}]$ where $s_k = hk$ and $k \in \mathbb{N}$ is the sampling label. Since A is invertible we can compute:

$$G(h) = (e^{Ah} - I)A^{-1}B \tag{3}$$

It can be easily verified that $rank(conmat(F(h), G(h))) = 2$ hence the discrete-time system is controllable.

Here, we consider the implementation of the continuous-time controller using feedback law $u_k = -\bar{K}x_k$ such that we obtain the discrete-time dynamics:

$$\begin{aligned} x_{k+1} &= (F(h) - G(h)\bar{K})x_k \\ &= F_K(h)x_k \end{aligned} \tag{4}$$

To study for which inter-sampling time the NCS is stable using $\bar{K}$, algorithm 1 has been implemented. Note that $\rho(\cdot)$ denotes the spectral radius of a matrix.

Algorithm 1 Stability for various values of h

> **for** $h \leftarrow h_{min}$ to h_{max} **do**
>   **if** $\rho(F_K) < 1$ **then**
> $F_K(h)$ `is stable`
>   **else**
> $F_K(h)$ `is unstable`
>   `save first unstable` h
>   **end if**
> **end for**

It must be noted that merely saving the first h for which the NCS is unstable does not account for cases where there are multiple intervals of unstable inter-sampling times (i.e., where in between these intervals the closed-loop is stable). Hence, I have firstly checked the spectral radius for a large interval of h (e.g. up to $h = 20$) with relatively large steps and noticed that it increases exponentially for increasing h. Subsequently, I studied stability for a smaller interval with smaller steps to obtain a more precise bound. The latter is shown visually in figure 2. We can conclude by this analysis that the NCS is stable using linear feedback law $u_k = -\bar{K}x_k$ for a constant inter-sampling time $h \in [5 \times 10^{-3},\ 0.5116]$. Notably the stability property is not affected using small inter-sampling times. This is intuitive since the behaviour of the discrete-time solution 'resembles' the continuous-time system using small values of h. One could test for even smaller values of h with the risk of encountering numerical issues.

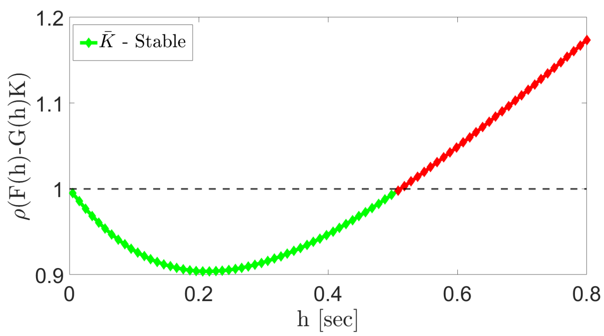

Figure 2: Stable closed loop dynamics (green) and unstable closed loop dynamics (red) for various inter-sampling times using feedback gain $\bar{K}$

Q2.1: h τ combinations

In the following analysis we consider that the NCS is affected by a constant small delay $\tau \in [0, h)$ and is sampled at a constant rate. We define the following matrices:

$$\begin{aligned} F_x(h) &= e^{Ah} \\ F_u(h,\tau) &= \int_{h-\tau}^{h} e^{As}B\,ds = (e^{Ah} - e^{A(h-\tau)})A^{-1}B \\ G_1(h,\tau) &= \int_{0}^{h-\tau} e^{As}B\,ds = (e^{A(h-\tau)} - I)A^{-1}B \end{aligned} \tag{5}$$

such that we can consider the extended state vector $x_k^e = \begin{bmatrix} x_k^T & u_{k-1} \end{bmatrix}^T$, and define:

$$\begin{aligned} F(h,\tau) &= \begin{bmatrix} F_x(h) & F_u(h,\tau) \\ 0 & 0 \end{bmatrix} \\ G(h,\tau) &= \begin{bmatrix} G_1(h,\tau) \\ I \end{bmatrix} \end{aligned} \tag{6}$$

then the extended state-dynamics become:

$$x_{k+1}^e = F(h,\tau)x_k^e + G(h,\tau)u_k \tag{7}$$

As a consequence, it is necessary to augment the feedback gain and define $K = \begin{bmatrix} \bar{K} & K_u \end{bmatrix}$.

At first, I studied combinations of h and τ that retain stability using a static feedback gain. That is $K = \begin{bmatrix} \bar{K} & 0 \end{bmatrix}$. I have implemented an algorithm which is equivalent to 1, albeit it with an additional inner-loop to iterate over a set of delays up to $h - 0.01$. Figure 3 shows the stable and unstable combinations of h and τ.

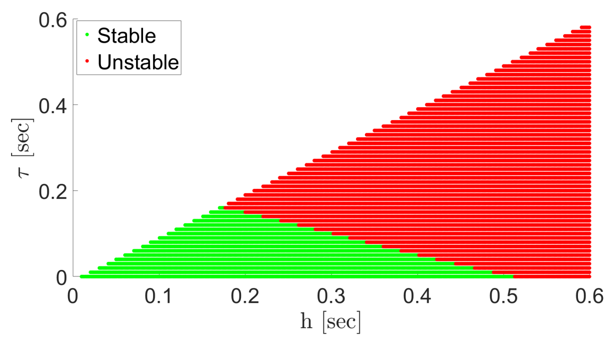

Figure 3: Combinations of h and τ that are stable (green) or unstable (red) using a static feedback gain

The results validate the analysis visualized in figure 2 in the sense that the range of stable inter-sampling times without delays is equivalent. Furthermore, it is shown that up to approximately $h = 0.19$ the NCS is robust to small delays. Thereafter the magnitude of acceptable delay descends linearly. This is an expected result since the system has a decreasing margin to instability with increasing inter-sampling time, hence with a small delay the system destabilizes easily.

Q2.2: Dynamic LQR controller

In the following analysis we study stability of the NCS with a constant inter-sampling time and a constant delay $\tau \in [0, h)$ and a dynamic controller with $K_u \neq 0$. That is, consider the linear state feedback law:

$$u_k = -Kx_k^e = -\bar{K}x_k - K_u u_{k-1}$$

then one can see that the feedback law is in fact a dynamic system as it has inherent "memory".

I designed a feedback gain for $h_d = 0.4$, as the system is not so robust to delays using this sampling rate, with a corresponding delay τ_d. The approach involves the design of a linear quadratic regulator (LQR) by computing the solution to the discrete-time algebraic Riccati equation (DARE):

$$
\begin{aligned}
&F^T P F - E^T P E \\
&- (F^T P G)(G^T P G + R)^{-1}(F^T P G)^T + Q = 0
\end{aligned}
\tag{8}
$$

where for notational brevity $F(h_d, \tau_d) = F$ and $G(h_d, \tau_d) = G$. Here, h_d and τ_d are essentially design parameters as you solve the discrete-time algebraic Riccati equation for the $F(h_d, \tau_d)$ and $G(h, \tau_d)$ matrices. Using `idare()` the feedback gain is computed:

$$
K_{LQR}(\tau_d) = (G^T P G + R)^{-1} G^T P F \tag{9}
$$

The idea is thus to design a possibly dynamic feedback law that stabilizes the system under a constant inter-sampling time $h_d = 0.4$ and a constant delay τ_d, yet to be chosen. The DARE matrices have been chosen as $Q = E = I$ and $R = 1$.

In a first approach, I designed a controller using $\tau_d = 0.1$. The resulting feedback gain is $K_{LQR} = \begin{bmatrix} -0.0178 & 2.2947 & 0.2176 \end{bmatrix}$. Figure 4 shows the resulting stable and unstable combinations. It is clear that indeed the range of tolerable delays using $h = 0.4$ has increased significantly. In addition, the controller is much more robust to delays within the entirety of selected inter-sampling times, which is a remarkable result.

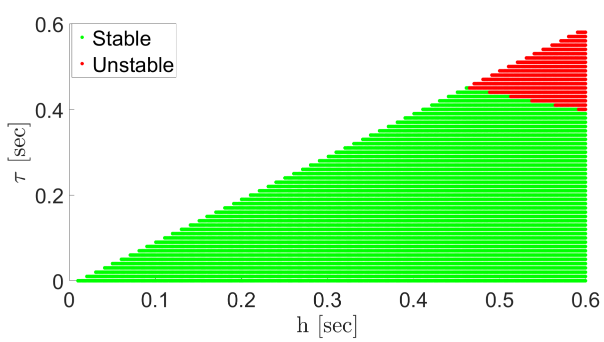

Figure 4: Combinations of h and τ that are stable (green) or unstable (red) using a dynamic LQR feedback gain designed for $h_d = 0.4$ and $\tau_d = 0.1$

One might note that, when computing a gain with the discrete-time equations instead of the continuous-time equations, the result is unavoidably better. Hence, in figure 5 the additional stable region due to $K_u \neq 0$ is shown in blue, i.e. the dynamic part of the controller. This confirms that a dynamic controller performs better in these circumstances both for $h_d = 0.4$ and for other h-τ combinations.

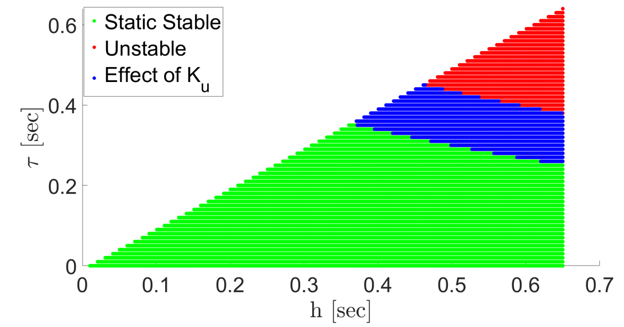

Figure 5: Combinations of h and τ that are stable (green) or unstable (red) using a dynamic LQR feedback gain designed for $h_d = 0.4$ and $\tau_d = 0.1$ with the effect of K_u shown in blue

In a second approach, arguably a less conservative one, I designed a controller using $\tau_d = 0.39$ which is close to the defined bound for "small" delays. The resulting feedback gain is $K_{LQR} = \begin{bmatrix} -0.0057 & 3.0873 & 0.9920 \end{bmatrix}$. Compared to the previous design the feedback law is "more dynamic" in the sense that the effect of u_{k-1} is larger. Figure 6 shows a very interesting result. The system is stable only for relatively large delays, both for $h = 0.4$ and other inter-sampling times in the analyzed range.

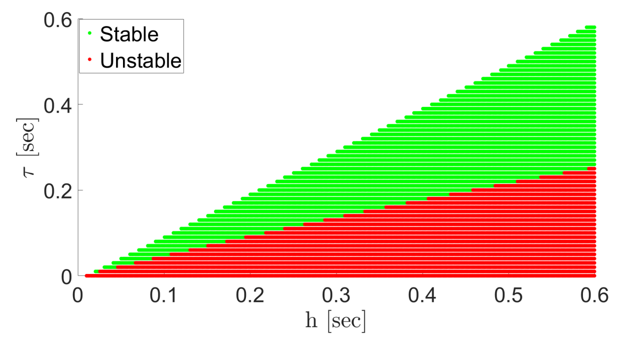

Figure 6: Combinations of h and τ that are stable (green) or unstable (red) using a dynamic LQR feedback gain designed for $h_d = 0.4$ and $\tau_d = 0.39$

In conclusion, it is shown that when redesigning a controller using the discrete-time equations, the performance is greatly improved. In addition, the use of a dynamic controller boosts the performance. However, when designing a dynamic controller for a specific h and τ it is risky to design the controller for a $\tau \to h$ as it can make the system unstable for $\tau \to 0$.

Q3: 1^{st}-ORDER HOLD

Here, we consider a different approach to deal with "small" delays $\tau \in [0, h)$. We employ a controller that at some time $t = s_{k-1} + \tau^{sc}$ computes $u_{k-1} = -\bar{K}x(s_{k-1})$ and $u_{k-2} = -\bar{K}x(s_{k-2})$. Both of these inputs are received at some smart actuator at time $t = s_{k-1} + \tau \in [s_{k-1}, s_k]$. The smart actuator, which replaces the ZOH block in figure 1, updates the input in a time-driven fashion by setting $u(s_k) = u_{k-2}$ and $u(s_{k+1}) = u_{k-1}$, and linearly interpolating at all values $t \in [s_k, s_{k+1})$.

Firstly, let us define an equation for the linearly interpolated input. It can be seen that for each $t \in [s_k, s_{k+1})$:

$$\begin{aligned} u(t) &= \frac{u_{k-1} - u_{k-2}}{h}(t - s_k) + u_{k-2} \\ &= \frac{\Delta u}{h}(t - s_k) + u_{k-2} \end{aligned} \tag{10}$$

therefore, the solution becomes:

$$\begin{aligned} x_{k+1} &= e^{Ah}x_k \\ &\quad + \int_{s_k}^{s_{k+1}} e^{A(s_{k+1}-s)}(\frac{\Delta u}{h}(s - s_k) + u_{k-2})B\,ds \end{aligned}$$

now let $s' = s_{k+1} - s$, then:

$$x_{k+1} = F_x(h)x_k - \int_h^0 e^{As'}(\frac{\Delta u}{h}(h - s') + u_{k-2})B\,ds'$$

redefine $s = s'$ and flip the integration limits such that we obtain:

$$\begin{aligned} x_{k+1} &= F_x(h)x_k + \int_0^h e^{As}\Delta u B\,ds \\ &\quad - \int_0^h \frac{1}{h}se^{As}B\Delta u\,ds + \int_0^h e^{A}u_{k-2}B\,ds \end{aligned}$$

$$\begin{aligned} x_{k+1} &= F_x(h)x_k + G(h)(u_{k-1} - u_{k-2}) \\ &\quad - G_u(h)(u_{k-1} - u_{k-2}) + G(h)u_{k-2} \end{aligned}$$

$$x_{k+1} = F_x(h)x_k + (G(h) - G_u(h))u_{k-1} + G_u(h)u_{k-2}$$

$$x_{k+1} = F_x(h)x_k + \tilde{F}_u(h)u_{k-1} + G_u(h)u_{k-2}$$

Subsequently, it is necessary to find an expression for the matrices defined above. Recalling that A is invertible, we can solve $G_u(h)$ symbolically through integration by parts. That is, in general $\int_a^b u\,dv = uv|_b^a - \int_a^b v\,du$ where we choose $u = s$ and $dv = e^{As}$, hence we obtain:

$$\begin{aligned} G_u(h) &= \frac{1}{h}(se^{As}A^{-1}|_0^h - \int_0^h e^{As}A^{-1}\,ds)B \\ &= e^{Ah}A^{-1}B + \frac{1}{h}(I - e^{Ah})A^{-2}B \end{aligned}$$

and $G(h)$ is defined in equation 3. Furthermore, it can be seen that:

$$G(h) - G_u(h) = \frac{1}{h}(e^{Ah} - I)A^{-2}B - A^{-1}B := \tilde{F}_u(h)$$

By extending the state vector as $x^e = \begin{bmatrix} x_k^T & u_{k-1} & u_{k-2} \end{bmatrix}^T$, we can finally define the exact discrete-time model:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ u_k \\ u_{k-1} \end{bmatrix} &= \begin{bmatrix} F_x & \tilde{F}_u & G_u \\ 0 & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} + \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} u_k \\ x_{k+1}^e &= \tilde{F}_1(h)x_k^e + \tilde{G}_1(h)u_k \end{aligned} \tag{11}$$

The first order-hold (FOH) controller will be implemented using the gain $\bar{K}$ designed in question 1 in a static fashion, as $K_{S,FOH} = \begin{bmatrix} \bar{K} & 0 \end{bmatrix}$. An analysis will follow comparing the following controllers:

- The ZOH controller $\bar{K}$ with FOH controller$K_{S,FOH}$
- A ZOH LQR controller K_{LQR} with the dynamic FOH LQR controller $\tilde{K}_{LQR}$

The latter is designed using the same weights as in Q2.2 and follows the dynamic feedback law:

$$\begin{aligned} u_k &= -\begin{bmatrix} K & K_u \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} \\ &= -\tilde{K}_{LQR}x_k^e \end{aligned}$$

Let us firstly observe the comparison of the ZOH controller using feedback gain $\bar{K}$ and the static FOH controller using $K_{S,FOH} = \begin{bmatrix} \bar{K} & 0 \end{bmatrix}$. It is clear from figure 7 that the FOH controller is stable for a smaller range of inter-sampling times. More precisely, the FOH controller is stable up to approximately a fifth of the stable bound for the ZOH controller.

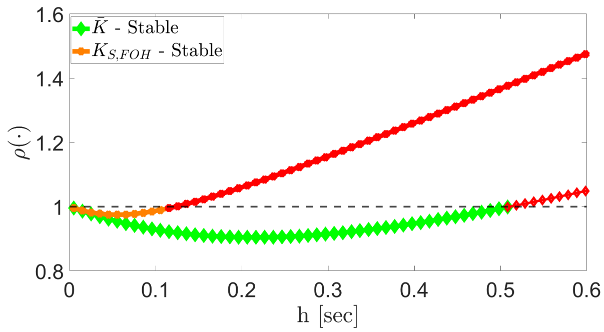

Figure 7: Stability of ZOH controller with gain $\bar{K}$ and FOH controller with gain $[\bar{K}\ 0]$ over a range of inter-sampling times

Figure 8 compares a K_{LQR} (ZOH) designed on $h_d = 0.4$ to a $\bar{K}_{LQR}$ (FOH) designed on $h_d = 0.4$ using the same weights. Similarly to the previous analysis, the FOH controller performs worse than the ZOH controller. More precisely, the difference between the highest stable h is approximately 0.3 seconds. However, the performance of both controllers has been greatly improved using an LQR designed using the discrete-time equations for $h_d = 0.4$.

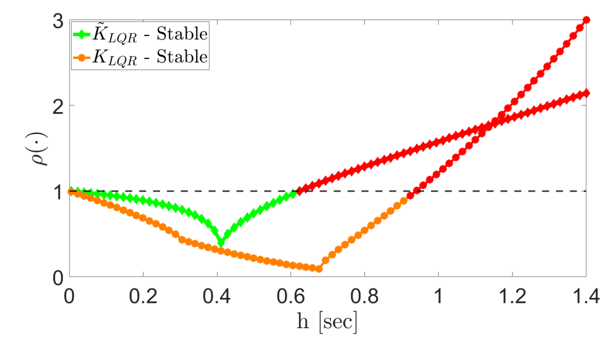

Figure 8: Stability of ZOH controller with LQR gain designed on $h_d = 0.4$ and **dynamic** FOH LQR controller (diamonds) with gain $\tilde{K}$ designed on $h_d = 0.4$ over a range of sampling times

In conclusion, the FOH controllers seem to perform worse than the ZOH controller. This holds true both for a static gain, designed using continuous-time equations, as for LQR controllers, designed using discrete-time equations.

Q4: Varying inter-sampling time

Let us consider a system with varying inter-sampling time $h \in \mathcal{H}_{\in} := \{h_1, h_2\}$ where the NCS uses a ZOH implementation when $h = h_1$ and a FOH implementation (i.e., from Q3) when $h = h_2$. In this situation an eigenvalue check is nor necessary nor sufficient, instead we must find a common Lyapunov function to conclude closed-loop stability for which we can use theorem 1.

Theorem 1 *The origin of the system $x_{k+1} = F(h)x_k + G(h)u_k$ with feedback law $u_k = -Kx_k$ is globally exponentially stable if there exists a matrix $P = P^T \succ 0$ such that the following LMIs hold for all $h \in \mathcal{H}$:*

$$(F(h) - G(h)K)^T P(F(h) - G(h)K) - P \preceq -Q$$

for some $Q = Q^T \succ 0$.

Therefore, in the problem at hand it is necessary to solve two linear matrix inequalities (LMIs). Since P is the same for both equations, the discrete-time solution for the ZOH system must be augmented to match the size of the FOH equations. Hence we have that:

$$\begin{bmatrix} x_{k+1} \\ u_k \\ u_{k-1} \end{bmatrix} = \begin{bmatrix} F_x & F_u & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \\ u_{k-2} \end{bmatrix} + \begin{bmatrix} G_1(h) \\ I \\ 0 \end{bmatrix} u_k \tag{12}$$

and denote its closed-loop form as $x^e_{k+1} = F_K(h_1)x^e_k$. Furthermore, denote the FOH closed loop form as $x^e_{k+1} = \tilde{F}_K(h_1)x^e_k$, then we can solve the following set of LMIs:

$$\begin{aligned} &F_K^T(h_1)PF_K(h_1) - P \preceq -Q \\ &\tilde{F}_K^T(h_2)P\tilde{F}_K(h_2) - P \preceq -Q \\ &P = P^T \succ 0 \end{aligned} \tag{13}$$

which can be simplified to:

$$\begin{aligned} &\begin{bmatrix} F_K^T(h_1)PF_K(h_1) - P & 0 \\ 0 & \tilde{F}_K^T(h_2)P\tilde{F}_K(h_2) - P \end{bmatrix} \preceq -Q \\ &P = P^T \succ 0 \end{aligned} \tag{14}$$

for some known $Q = Q^T > 0$.

Next, we consider the case where the sequence of inter-sampling times is known to be $(h_1, h_2)^\omega$ for a given **sampled-data** controller (i.e., not necessarily the one described above). Denote the closed-loop form as $x_{k+1} = A_K(h)x_k$. As the switching is constant, known, and infinite, we can denote the sequence of two inter-sampling times as a new dynamical system, that is:

$$x_{k+2} = A_K(h_1)A_K(h_2)x_k \tag{15}$$

Therefore, we can simply check that $\rho(A_K(h_1)A_K(h_2)) < 1$ to check whether the closed-loop is stable. Equivalently for the sequence $(h_1h_2h_2)^\omega$ we obtain the dynamical system:

$$x_{k+3} = A_K(h_1)A_K^2(h_2)x_k \tag{16}$$

Although superfluous, since an eigenvalue check is sufficient, we can equivalently solve the LMI:

$$\begin{aligned}&(A_K(h_1)A_K^2(h_2))^T P(A_K(h_1)A_K^2(h_2)) - P \preceq -Q\\ &P = P^T \succ 0\end{aligned} \tag{17}$$

Following an equivalent approach for the system switching between a ZOH and a FOH controller we can define the following dynamical systems:

$$\begin{aligned}x^e_{k+2} &= F_K(h_1)\tilde{F}_K(h_2)x^e_k, &&\text{for} \quad (h_1h_2)^\omega\\ x^e_{k+3} &= F_K(h_1)\tilde{F}^2_K(h_2)x^e_k, &&\text{for} \quad (h_1h_2h_2)^\omega\end{aligned} \tag{18}$$

Subsequently, we can check the spectral radius iteratively for combinations of h_1 and h_2 in the fashion of algorithm 1. Figures 9 and 10 show the stable combinations of h_1 and h_2 for the sequences $(h_1h_2)^\omega$ and $(h_1h_2h_2)^\omega$ respectively. From this analysis, we deduce 3 main results.

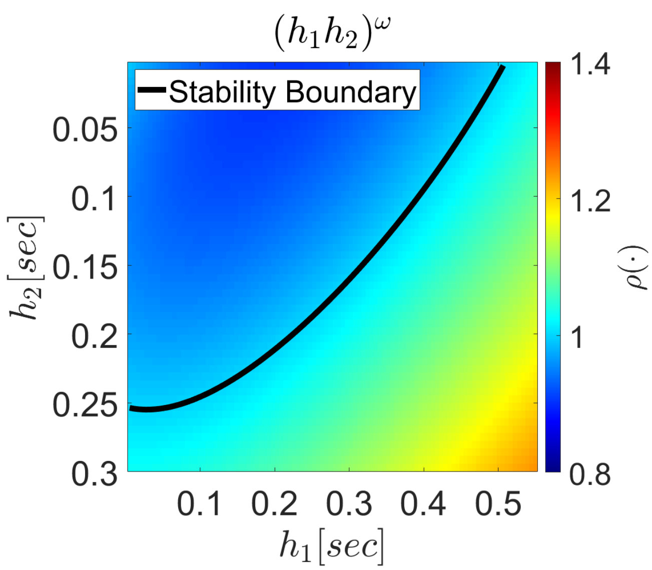

Figure 9: Stability of the switched closed-loop system $x^e_{k+2} = F_K(h_1)\tilde{F}_K(h_2)x^e_k$, using $K = [\bar{K} \quad 0 \quad 0]$ for a range of h_1 and h_2 values.

Firstly, it is shown that when sampling "slow" with one system we want to sample "fast" with the other system. "Slow" and "fast" are obviously relative terms. Secondly, the stability bound has a positive exponential relation between h_1 and h_2. That is, when decreasing the sampling rate on h_2 we want to increase the sampling rate on h_1 exponentially. Finally, the system with switching sequence $(h_1h_2h_2)^\omega$ has a smaller range of stable inter-sampling times combinations, mostly on the range of h_2. In addition, the gradient in the anti-diagonal direction is higher, which may indicate a less robust system.

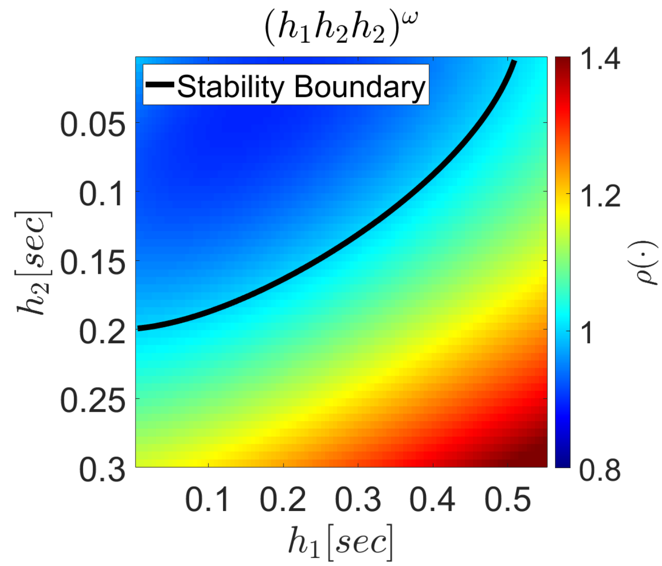

Figure 10: Stability of the switched closed-loop system $x^e_{k+2} = F_K(h_1)\tilde{F}^2_K(h_2)x^e_k$, using $K = [\bar{K} \quad 0 \quad 0]$ for a range of h_1 and h_2 values.

Q5: Interfering systems

The situation described in the assignment has a deterministic nature, since it is known when packets are lost. Let us firstly define the variable m_k such that we can model the loss of packet at sampling instant k:

$$\begin{cases} m_k = 1, & \text{when a packet is lost}\\ m_k = 0, & \text{when the packet is not lost}\end{cases} \tag{19}$$

In the to-zero strategy, for **system 1** the solution becomes:

$$x_{k+1} = \begin{cases} F_0^z x_k := (F(h_1) - G(h_1)\bar{K})x_k, & m_k = 0 \\ F_1^z x_k := F(h_1)x_k, & m_k = 1 \end{cases} \tag{20}$$

whilst for **system 2**:

$$x_{k+1} = \begin{cases} \mathcal{F}_0^z x_k := (F_2(h_2) - G(h)\bar{K}_2)x_k, & m_k = 0 \\ \mathcal{F}_1^z x_k := F_2(h_2)x_k, & m_k = 1 \end{cases} \tag{21}$$

where $h_2 = 3h_1$ and $F_2(h_2) = e^{\frac{1}{3}Ah_2}$.

In the **to hold** strategy, it is necessary to augment the state-space such that we can use the previous input (i.e., hold it), the extended state then becomes $x_k^e = \begin{bmatrix} x_k & u_{k-1} \end{bmatrix}$. Therefore, for **system 1** we become:

$$x_{k+1}^e = \begin{cases} F_0^h x_k := \begin{bmatrix} (F(h_1) - G(h_1)\bar{K}) & 0 \\ -\bar{K} & 0 \end{bmatrix} x_k^e, \; m_k = 0 \\ F_1^h x_k := \begin{bmatrix} F(h_1) & G(h_1) \\ 0 & I \end{bmatrix} x_k^e, \; m_k = 1 \end{cases} \tag{22}$$

whilst for **system 2** in the to hold strategy we become:

$$x_{k+1}^e = \begin{cases} \mathcal{F}_0^h x_k := \begin{bmatrix} (F_2(h_2) - G(h_2)\bar{K}_2) & 0 \\ -\bar{K}_2 & 0 \end{bmatrix} x_k^e, m_k = 0 \\ \mathcal{F}_1^h x_k := \begin{bmatrix} F_2(h_2) & G(h_2) \\ 0 & I \end{bmatrix} x_k^e, \; m_k = 1 \end{cases} \tag{23}$$

note that the superscript h denotes to-hold and the superscript z denotes the to-zero strategy.

It is known that (only) during simultaneous communication over the network System 2 and 1 alternate who loses a packet, in that order. Hence, the sequence is deterministic and known for both systems. We can therefore denote the sequences of both systems:

$$\begin{aligned} m^1 &\models (001000)^\omega, && \text{for system 1} \\ m^2 &\models (01)^\omega, && \text{for system 2} \end{aligned} \tag{24}$$

Due to the periodic nature of the sequence, it can be seen that for **system one**:

$$x_{k+6}^{(e)} = (F_0^i)^2 F_1^i (F_0^i)^3 x_k^{(e)}, \; k = 0, 6 \ldots, \; i \in \{h, z\} \tag{25}$$

whilst for **system 2**

$$x_{k+2}^{(e)} = \mathcal{F}_0^i \mathcal{F}_1^i x_k^{(e)}, \; k = 0, 2 \cdots, \; i \in \{h, z\} \tag{26}$$

which are both discrete-time LTI systems. Therefore, we can guarantee stability of both closed-loop system by verification of proposition 1.

Proposition 1 *The origins of System 1 and System 2 as described in Assignment 1.5, with their respective sequences m^1 and m^2 defined in equation 24, are both exponentially stable if and only if:*

$$i \in \{h, z\} \begin{cases} \rho((F_0^i)^2 F_1^i (F_0^i)^3) < 1 \\ \rho(\mathcal{F}_0^i \mathcal{F}_1^i) < 1 \end{cases}$$

for a pre-specified h_1 and h_2 respectively.

Let us verify the proposition for a set of inter-sampling times h_1 with corresponding h_2 and by combining both strategies. The stability has been verified in an iterative fashion by checking the maximum spectral radius of both systems using combinations for $h_1 \in [0.005, 0.7$. Figure 11 shows combinations of strategies and their stability as a function of h_1.

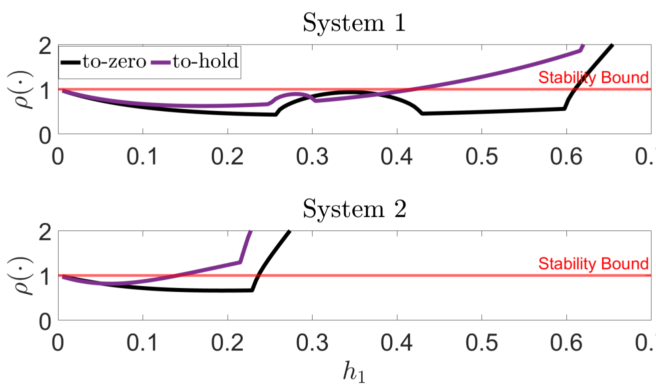

Figure 11: Stability of two packet loss strategies for a range of h_1 where $h_2 = 3h_1$

It is shown that the best strategy to employ is to-zero on both systems. Furthermore, it is clear that system 2limits the sampling time for stability.