Assignment 2

SC42101: Networked and Distributed Control Systems

Richard Adam [redacted]

Delft University of Technology

TU Delft

Introduction

The purpose of this assignment is to analyze and control a networked control system(NCS) with uncertain delays τ and changing sampling time h, as well as design an Event Triggered Controller. The schematic for the NCS can be seen in Figure 1:

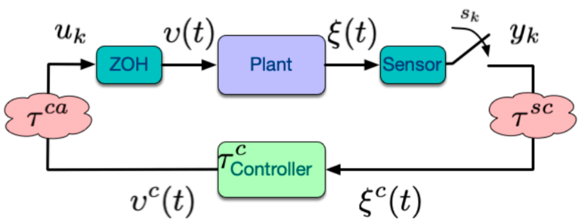


Figure 1: Diagram of a Networked Control System

The plant dynamics represent a linear time-invariant system of the form:

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

where ν is a piecewise constant signal applied in a sample-and-hold fashion. A and B are given by:

$$A = \begin{bmatrix} 0 & -1.5 \\ 2.2 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Assuming no time delays, the dynamics of the discrete-time sampled-data system can be represented by the equation:

$$x_{k+1} = e^{Ah}x_k + (\int_0^h e^{As}Bds)u_k \quad (1)$$

which can be expressed as:

$$x_{k+1} = F(h)x_k + G(h)u_k \quad (2)$$

Where:

$$F(h) = e^{Ah}, \quad G(h) = \int_0^h e^{As}Bds$$

For the initial problems, it is assumed that the system has two possible controllers, both following the form: $\nu(t) = -K\xi(s_k)$ for t ϵ $[s_k, s_{k+1})$, which are:

$$\bar{K} = \begin{bmatrix} 3.0000 & -0.5909 \end{bmatrix}, \quad \bar{K_2} = \begin{bmatrix} 0.4000 & 0.6000 \end{bmatrix}$$

Question 1: Deterministic Modeling of Sampling Time Switching

It is assumed that the system switches between two sampling times h_1 and h_2 following the sampling rule $(h_1h_2)^\omega$. The sampling times are, respectively, $h_1 = h$ and $h_2 = 2h$, and when the sampling interval is h_1, either the controller $\bar{K}$ or $\bar{K_2}$ is applied indistinctly. When the sampling interval is h_2, either the controller $\bar{K_2}$ or the zero input are applied indistinctly.

1.1: ω-Automaton

The ω-automaton that models this indiscriminate switching can be seen in Figure 2:

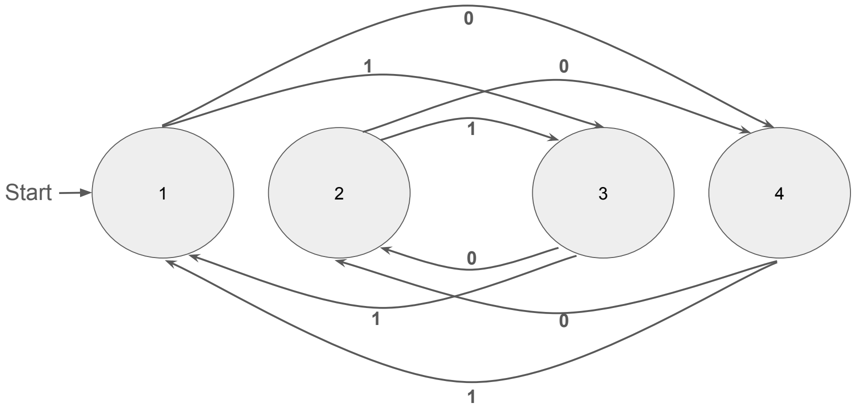


Figure 2: Graphical Representation of the ω-Automaton for a Switching Sampling Rate System

Where the gray circles are states, with the four states defined as:

- State 1: Sampling Rate h_1, K=$\bar{K}$
- State 2: Sampling Rate h_1, K=$\bar{K}_2$
- State 3: Sampling Rate h_2, K=$\bar{K}_2$
- State 4: Sampling Rate h_2, K=0

And the arrows represent state transitions, where each state can only transition to two other states via a transition 0 or transition 1 respectively.

1.2: Received Sample Dynamics

There are several ways to represent the dynamics of the received samples in the switched system, whether that be providing an expression for each time period or after a certain number of timesteps. The dynamics provided incorporates the ω-automaton states, where each state has its own dynamics equation, given by the following:

$$F_1^{cl}(h) = (F(h) - G(h)\bar{K}) \tag{3}$$

$$F_2^{cl}(h) = (F(h) - G(h)\bar{K}_2) \tag{4}$$

$$F_3^{cl}(h) = (F(2h) - G(2h)\bar{K}_2) \tag{5}$$

$$F_4^{cl}(h) = F(2h) \tag{6}$$

Combining these, the overall dynamics equation with respect to the ω-automaton states can be represented as:

$$x_{k+1} = \begin{cases} F_1^{cl}(h)x_k, & \text{if } q_k = 1 \\ F_2^{cl}(h)x_k, & \text{if } q_k = 2 \\ F_3^{cl}(h)x_k, & \text{if } q_k = 3 \\ F_4^{cl}(h)x_k, & \text{if } q_k = 4 \end{cases} \tag{7}$$

1.3: LMIs for Stability Verification

Because switching is indiscriminate, there are several different ways the system can evolve over time. To guarantee global exponential stability, one therefore must confirm that all possible modes starting at state 1 and returning to state 1 is stable. That is, all combination of subsequent states that eventually lead back to state 1, the assumed starting point, or any possible infinite loops between. Each intermediate state has a defined dynamics as given by Equation 7, so dynamics can be derived for all of the combinations. The combinations are given as the series of transitions(0,1), as well as the dynamic equation that must be checked for stability, can be seen in Tables 1 and 2. In those tables, a superscript * represents a finite repetition, while a superscript $^\omega$ represents an infinite repetition.

Path	Dynamics
$(11)^\omega$	$x_{k+2} = F_1^{cl}(h)F_3^{cl}(h)x_k$
$(01)^\omega$	$x_{k+2} = F_1^{cl}(h)F_4^{cl}(h)x_k$
$(1(01)^*1)^\omega$	$x_{k+4} = F_1^{cl}(h)F_3^{cl}(h)F_2^{cl}(h)F_3^{cl}(h)x_k$
$(0(01)^*1)^\omega$	$x_{k+4} = F_1^{cl}(h)F_4^{cl}(h)F_2^{cl}(h)F_4^{cl}(h)x_k$
$((10)^*(00)^*1)^\omega$	$x_{k+4} = F_1^{cl}(h)F_3^{cl}(h)F_2^{cl}(h)F_4^{cl}(h)x_k$
$(0(00)^*1)^\omega$	$x_{k+4} = F_1^{cl}(h)F_4^{cl}(h)F_2^{cl}(h)F_3^{cl}(h)x_k$

Table 1: All paths from State q_1 back to State q_1.

Loop	Dynamics
$(q_3 \to q_2 \to q_3)^\omega$	$x_{k+2} = F_3^{cl}(h)F_2^{cl}(h)x_k$
$(q_4 \to q_2 \to q_4)^\omega$	$x_{k+2} = F_4^{cl}(h)F_2^{cl}(h)x_k$
$(q_3 \to q_2 \to q_4 \to q_2)^\omega$	$x_{k+3} = F_3^{cl}(h)F_2^{cl}(h)F_4^{cl}(h)x_k$
$(q_4 \to q_2 \to q_3 \to q_2)^\omega$	$x_{k+3} = F_4^{cl}(h)F_2^{cl}(h)F_3^{cl}(h)x_k$

Table 2: All Possible Infinite Loops not Involving q_1

If the dynamics of all ten conditions above are considered to be in the form of $x_{k+j} = F_i^{cl}(h)x_k$, then there will be thirty total LMIs in the form of:

There exists some $P_i = P_i^T \succ 0$ such that: $(F_i^{cl})^T P_i F_i^{cl} - P_i \preceq Q_i$ for some $Q_i = Q_i^T \succ 0 \; \forall \; i \in \; [1, 10]$

If all these LMIs can be met, the system is stable no matter the series of state transitions.

1.4: Evaluating Stable Sampling Ranges

Alternatively, the spectral radius $F_i^{cl}(h)$ can be evaluated for $i \in [1, 10]$, and all spectral radii $\rho(F_i^{cl}(h))$ are less than 1, the system is globally exponentially stable at a given h. The max spectral radius between all $F_i^{cl}(h)$ was plotted over a range of hs, and can be seen in Figure 3:

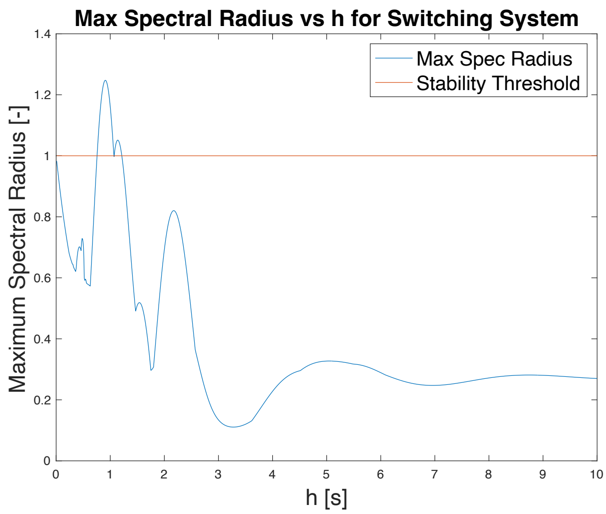


Figure 3: Max Spectral Radius of $F_i^{cl}(h)$ vs Sampling Time h for $h_1 = h$ and $h_2 = 2h$

Because the settling time of the system is approximately 9s, the system is plotted from h=0.01s to h=10s, making sure new samples span the entire transient response. The system is clearly stable for all h except that between h=0.75s to h=1.25s.

Question 2: Stochastic Modeling of Sampling Time Switching

Now, the switching between controllers is no longer indiscriminate. If the sampling time is h_1, there is a probability of p that the controller will be $\bar{K}$, otherwise the controller will be $\bar{K_2}$. If the sampling time is h_2, there is a probability of q that the controller is $\bar{K_2}$, otherwise the controller will be a zero input.

2.1: Markov Chain

This stochastic switching structure can be modeled by a Markov Chain. The diagram of the Markov chain can be seen in Figure 4:

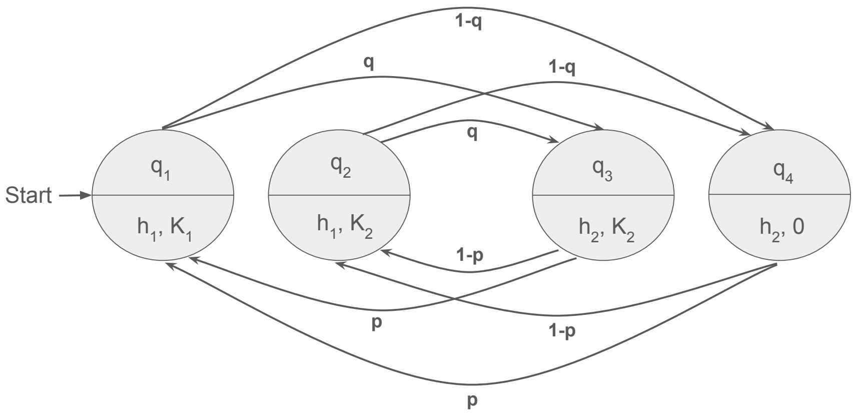


Figure 4: Diagram of the Markov Chain in the Case of Stochastic Switching

Where the gray circles are states, with the sampling time and controller associated with the state below the state name.

2.2: Received Sample Dynamics

Similar to the deterministic case, the stochastic dynamics can be incorporated into the state structure. With the same $F_1^{cl}(h)$, $F_2^{cl}(h)$, $F_3^{cl}(h)$, and $F_4^{cl}(h)$, the sampled data dynamics can be represented as:

$$x_{k+1} = \begin{cases} F_1^{cl}(h)x_k, & \text{if } q_k = 1 \\ F_2^{cl}(h)x_k, & \text{if } q_k = 2 \\ F_3^{cl}(h)x_k, & \text{if } q_k = 3 \\ F_4^{cl}(h)x_k, & \text{if } q_k = 4 \end{cases} \tag{8}$$

The difference now is that the transitions are probabilistic, and the probabilities of switching into one state from another can be seen in the following matrix:

		To			
		q_1	q_2	q_3	q_4
From	q_1	0	0	q	1-q
	q_2	0	0	q	1-q
	q_3	p	1-p	0	0
	q_4	p	1-p	0	0

Table 3: Probabilities of Switching from Each State to Each Other State

2.3: LMIs to Verify Mean-Squared Stability

There are several notions of stochastic stability that could be used to check for the given system. The chosen method is Mean Squared Stability, which has four equivalent stability conditions, of which the one examined is:

$$P_i - \sum_{j=0}^{N} p_{ij} A_j^T P_j A_j \succ 0 \qquad \forall\ i = 0, 1, 2,\ ...N$$

Where N is the number of states, and $p_{i,j}$ is the probability of transitioning from state i to state j.

Plugging in the values for A_i and $p_{i,j}$, the LMI system is as follows:

1. $P_1 = P_1^T \succ 0$
2. $P_2 = P_2^T \succ 0$
3. $P_3 = P_3^T \succ 0$
4. $P_4 = P_4^T \succ 0$
5. $P_1 - q(F_3^{cl})^T P_3 F_3^{cl} - (1-q)(F_4^{cl})^T P_4 F_4^{cl} \succ 0$
6. $P_2 - q(F_3^{cl})^T P_3 F_3^{cl} - (1-q)(F_4^{cl})^T P_4 F_4^{cl} \succ 0$
7. $P_3 - p(F_1^{cl})^T P_1 F_1^{cl} - (1-p)(F_2^{cl})^T P_2 F_2^{cl} \succ 0$
8. $P_4 - p(F_1^{cl})^T P_1 F_1^{cl} - (1-p)(F_2^{cl})^T P_2 F_2^{cl} \succ 0$

2.4: Evaluating Stable Sampling Times

In order to check the stability for various h values, the LMIs were solved using cvxpy and Python.

For each sampling time, the LMI is either solvable(stable) or unsolvable(unstable), and the results can be seen in Figure 5:

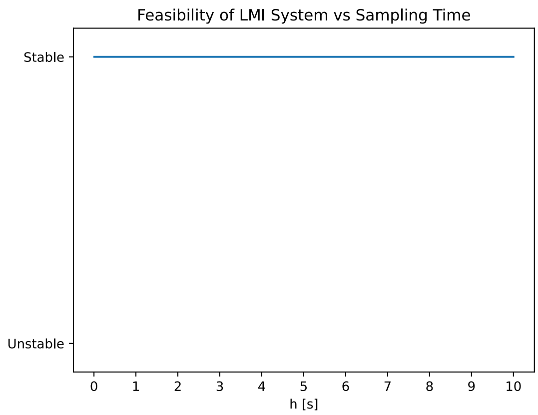


Figure 5: Graph of Stability of Stochastic Sampling Case

The graph clearly shows that the system is stable over the entire range of sampling times from h=0.01s to h=10s. This is somewhat expected, as checking one particular stochastic case should have a larger range of stable sampling frequencies than the deterministic model seen in Figure 3, which checks all possible combinations of state transitions with no regard to probability.

Question 3: Incorporating Uncertain Delays

Now, it is assumed that the varying sampling times and controllers are swapped for a constant sampling time and constant K, but there is some time delay $\tau \in\ [0, h)$.

In addition, there are new state matrices:

$$A = \begin{bmatrix} 2.3 & 0 \\ 1 & 2.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

To avoid complex eigenvalues.

3.1: Controller Design and Discrete-Time Modeling

First, a controller was designed to put the poles of the continuous-time sytem at $-2 \pm i$. This was done by designing a controller $K = [K_1,\ K_2]$ such that the closed-loop system $(A - BK)$ has a characteristic equation with poles at $-2 \pm i$. The specific steps can be seen in Appendix B.

The final controller is $K = [8.8000,\ 21.2500]$. Since the delay is less than the sampling time, the delayed system can be represented by the equation:

$$x_{k+1} = e^{Ah} x_k + \left(\int_{h-\tau}^{h} e^{As} B ds\right) u_{k-1} + \left(\int_{0}^{h-\tau} e^{As} B ds\right) u_k \tag{9}$$

This can be stated as:

$$x_{k+1} = F_x(h) x_k + F_u(h,\tau) u_{k-1} + G_1(h,\tau) u_k \tag{10}$$

where:

$$F_x(h) = e^{Ah}, \quad F_u(h) = \int_{h-\tau}^{h} e^{As} B ds, \quad G_1(h) = \int_{0}^{h-\tau} e^{As} B ds$$

This can be simplified by using the extended state $x_k^e = [x_k^T,\ u_{k-1}]^T$, which results in:

$$x_{k+1}^e = F(h,\tau) x_k^e + G(h,\tau) u_k \tag{11}$$

where

$$F = \begin{bmatrix} F_x(h) & F_u(h,\tau) \\ 0 & 0 \end{bmatrix} \text{ and } G = \begin{bmatrix} G_1(h,\tau) \\ 1 \end{bmatrix}$$

Solving for the following with A and B, which can be seen in Appendix B, one obtains the equations:

$$F(h) = \begin{bmatrix} e^{2.3h} & 0 & \frac{1}{2.3}(e^{2.3h} - e^{2.3(h-\tau)}) \\ 5e^{2.5h} - 5e^{2.3h} & e^{2.5h} & 2(e^{2.5h} - e^{2.5(h-\tau)}) + \frac{5}{2.3}(e^{2.3h} - e^{2.3(h-\tau)}) \\ 0 & 0 & 0 \end{bmatrix} \tag{12}$$

$$G(h) = \begin{bmatrix} 0.435e^{2.3(h-\tau)} - 0.4389 \\ 2e^{2.5(h-\tau)} - 2.174e^{2.3(h-\tau)} + 0.174 \\ 1 \end{bmatrix} \tag{13}$$

To check the stability of this system, one can iterate through a range of h values and τ values, checking the spectral radius of the closed-loop system for each. This is plotted on a heatmap, with all unstable systems(i.e. spectral radius greater than one) shown in black.

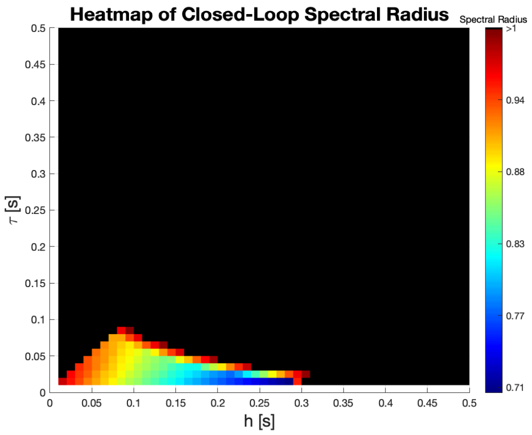


Figure 6: Heatmap of Spectral Radius for Delayed System. Unstable Regions in Black.

The limit of 0.5s was chosen since all sampling times and delays above 0.5s resulted in unstable systems. The plot shows that for sampling times up to 0.09s, the system is stable for any delays. For sampling times between 0.09s and 0.3s, small delays maintain system stability, but any delays above 0.1s will always destabilize the system.

3.2: Jordan Form Approach with Polytopic Over-Approximation

The Jordan form/polytopic over-approximation approach differs slightly in that it examines stability for a range of τ values instead of specific τ values. The concept of the Jordan form approach hinges around putting your system in the form of:

$$x^e_{k+1} = (F_0 + \sum_{i=1}^{r} \alpha(\tau_k)F_i)x^e_k + ((G_0 + \sum_{i=1}^{r} \alpha(\tau_k)G_i)u_k$$

where r is the number of distinct and real eigenvalues a system has. α takes the form, derived from the definition of a Jordan form of e^{At},

$$\alpha_i(\tau_k) = \frac{(h-\tau_k)^j}{j} e^{\lambda_i(h-\tau_k)}$$

Where j ranges from 0 to the algebraic multiplicity of the eigenvalue minus one. It is important to note that this is specific for the case where only delays are uncertain, as more generally α_i can depend on any uncertain quantity of the system. One can rewrite Equations 12 and 13, assuming some constant h, to be:

$$F(\tau_k) = F_0 + \alpha_1(\tau_k)F_1 + \alpha_2(\tau_k)F_2 \tag{14}$$

$$G(\tau_k) = G_0 + \alpha_1(\tau_k)G_1 + \alpha_2(\tau_k)G_2 \tag{15}$$

$$F_0 = \begin{bmatrix} e^{2.3h} & 0 & \frac{1}{2.3}e^{2.3h} \\ 5e^{2.5h} - 5e^{2.3h} & e^{2.5h} & 2e^{2.5h} - \frac{5}{2.3}e^{2.3h} \\ 0 & 0 & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 & 0 & \frac{-1}{2.3} \\ 0 & 0 & \frac{5}{2.3} \\ 0 & 0 & 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$G_0 = \begin{bmatrix} -0.4348 \\ 0.174 \\ 1 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0.435 \\ -2.174 \\ 0 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

$$\alpha_1(\tau_k) = e^{2.3(h-\tau)}, \quad \alpha_2(\tau_k) = e^{2.5(h-\tau)}$$

The α_i expressions have no coefficients because each eigenvalue has an algebraic multiplicity of 1, so j=0. To form the polytopic over-approximation, one must find the minimum and maximum values of $\alpha_i(\tau_k)$, noted as $\underline{\alpha}_i$ and $\bar{\alpha}_i$ respectively. Clearly, considering the expressions for $\alpha_i(\tau_k)$, $\underline{\alpha}_i$ occurs at $\tau_{k,max}$, while $\bar{\alpha}_i$ occurs at $\tau_{k,min}$.

With these, the polytopic over-approximation can be expressed as:

$$\bar{F} = \{F_0 + \sum_{i=1}^{2} \delta_i F_i \mid \delta_i \ \in [\underline{\alpha}_i,\ \bar{\alpha}_i],\ \ i = 1, 2\} \tag{16}$$

$$\bar{G} = \{G_0 + \sum_{i=1}^{2} \delta_i G_i \mid \delta_i \ \in [\underline{\alpha}_i,\ \bar{\alpha}_i],\ \ i = 1, 2\} \tag{17}$$

The matrix vertices of this set can be obtained by evaluating these equations for all combinations of extremes of δ_i. For this set, those combinations would be:

1. $\delta_1 = \underline{\alpha}_1 = \alpha_1(\tau_{k,max}),\ \delta_2 = \underline{\alpha}_2 = \alpha_2(\tau_{k,max})$
2. $\delta_1 = \underline{\alpha}_1 = \alpha_1(\tau_{k,max}),\ \delta_2 = \bar{\alpha}_2 = \alpha_2(\tau_{k,min})$
3. $\delta_1 = \bar{\alpha}_1 = \alpha_1(\tau_{k,min}),\ \delta_2 = \underline{\alpha}_2 = \alpha_2(\tau_{k,max})$
4. $\delta_1 = \bar{\alpha}_1 = \alpha_1(\tau_{k,min}),\ \delta_2 = \bar{\alpha}_2 = \alpha_2(\tau_{k,min})$

Evaluating equations 16 and 17 for each of these conditions, we can get the expressions for $H_{F,i}$ and $H_{G,i}$:

1. $H_{F,1} = F_0 + \alpha_1(\tau_{k,max})F_1 + \alpha_2(\tau_{k,max})F_2$
2. $H_{F,2} = F_0 + \alpha_1(\tau_{k,max})F_1 + \alpha_2(\tau_{k,min})F_2$
3. $H_{F,3} = F_0 + \alpha_1(\tau_{k,min})F_1 + \alpha_2(\tau_{k,max})F_2$
4. $H_{F,4} = F_0 + \alpha_1(\tau_{k,min})F_1 + \alpha_2(\tau_{k,min})F_2$
5. $H_{G,1} = G_0 + \alpha_1(\tau_{k,max})G_1 + \alpha_2(\tau_{k,max})G_2$
6. $H_{G,2} = G_0 + \alpha_1(\tau_{k,max})G_1 + \alpha_2(\tau_{k,min})G_2$
7. $H_{G,3} = G_0 + \alpha_1(\tau_{k,min})G_1 + \alpha_2(\tau_{k,max})G_2$
8. $H_{G,4} = G_0 + \alpha_1(\tau_{k,min})G_1 + \alpha_2(\tau_{k,min})G_2$

The set of LMIs to verify that the system will be stable over the entire range of $\tau \ \in \ [0, h)$ is:

1. There exists some $P = P^T \succ 0$ such that:
2. $(H_{F,1} - H_{G,1}K)^T P(H_{F,1} - H_{G,1}K) - P \preceq -\gamma P$
3. $(H_{F,2} - H_{G,2}K)^T P(H_{F,2} - H_{G,2}K) - P \preceq -\gamma P$
4. $(H_{F,3} - H_{G,3}K)^T P(H_{F,3} - H_{G,3}K) - P \preceq -\gamma P$
5. $(H_{F,4} - H_{G,4}K)^T P(H_{F,4} - H_{G,4}K) - P \preceq -\gamma P$

For some $\gamma \in (0, 1)$.

3.3: Stability Analysis via Polytopic Overapproximation

The LMIs in the previous are solved in Python with cvx to find all sampling times h which are stable for the entirety of $\tau \in [0, h)$. The graph of stable h and τ values can be seen in Figure 7:

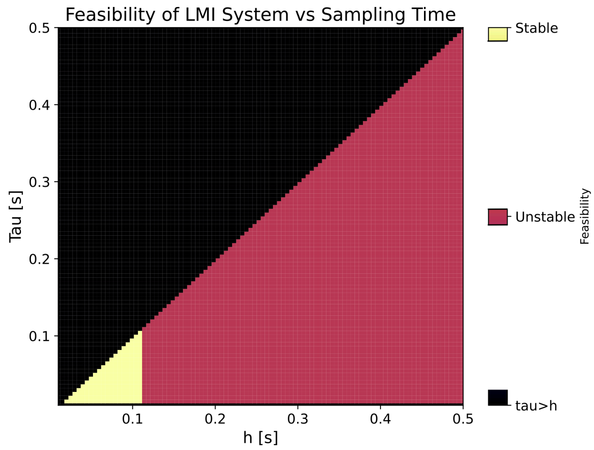


Figure 7: Graph of Stability using Polytopic Over-Approximation.

3.4: Stability Comparison

Comparing with Figure 6, it is clear that the range of stable sampling times is smaller in the case of polytopic over-approximation. This is expected, since the method in 3.1 of evaluating stability at individual values of h and τ has the ability to see where the system may be stable at a particular sampling time only for *some* delay values between 0 and h. Importantly, the graphs almost entirely agree up to $h = 0.1s$. This is because the polytopic over-approximation method shows stability only where the system is stable for the entire range of $\tau \in [0, h)$, and Figure 6 shows this region stable over all τ as well. It should be noted that Figure 7 shows this section extending slightly above $h = 0.1s$, where Figure 6 shows it ending slightly below $h = 0.1s$. This may be due to rounding errors while defining equations in Python and MATLAB.

Question 4: Event-Triggered Controller Design

For the ETC design, it is assumed that the system no longer has any delays acting on it. In addition, the A, and B matrices are set back to their values in Questions 1 and 2, and K is set to $\bar{K}$ from Questions 1 and 2.

4.1: Designing a Triggering Condition

The basic goal of the controller is to retain global exponential stability in the system while being able to specify some level of minimum performance. To retain stability, we can start with the Lyapunov stability condition, saying that there exists a function $V(t) = x^T P x$ with a matrix $P = P^T \succ 0$ such that

$$(A - BK)^T P + P(A - BK) = Q$$

for some matrix $Q \succ 0$. For Q we can select any positive definite matrix, so the identity matrix is selected. Knowing $(A - BK)$ and Q, one can use MATLAB's `lyap()` function to find a corresponding P.

To specify some performance of the system, one can use the property $\frac{d}{dt}V(\xi(t)) = -x^T Q x$ to represent the convergence rate of the system. One can then specify a particular performance by setting $\frac{d}{dt}V(\xi(t)) \leq -\sigma \xi(t)^T Q \xi(t)$ where $\sigma \in (0, 1)]$. Recalling the sampled-data system dynamics:

$$\dot{\xi}(t) = A\xi(t) - BK\xi(s_k)$$

we can rewrite the performance condition as:

$$\xi(t)^T (A^T P - PA)\xi(t) - \xi(t)^T PBK\xi(t) - \xi(t)^T (BK)^T P\xi(t) \leq -\sigma \xi(t)^T Q\xi(t)$$

Moving all variables to one side and putting this into matrix form, one can obtain the condition for triggering the new sample to be gathered:

$$\phi_{ETC}(t) = \begin{bmatrix} \xi(t)^T & \xi(s_k)^T \end{bmatrix} \begin{bmatrix} A^T P + PA + \sigma Q & -PBK \\ -(BK)^T P & 0 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \xi(s_k) \end{bmatrix} \leq 0$$

This means that if the function $\phi_{ETC}(t)$ goes above 0, a new sample has to be taken to ensure a specific conversion rate.

4.2: Simulating ETC and Sampling Analysis

Simulating Event-Triggered Controllers is simple in MATLAB using the `ODE45()` function, which has options to allow for event handling. The system with the event-triggered controller was simulated over five seconds for σ values of 0.1, 0.3, 0.5, 0.7, and 0.9. Each of these σ values was tested for four intitial conditions: [1, 0], [0, 1], [2, 1], and [-10, -5]. At each σ value, the average time between samples was calculated across all initial conditions along with the standard deviation of time between samples across all initial conditions. Table 4 shows the statistical results:

σ	Mean	Standard Deviation
0.1	0.75027	0.38453
0.3	0.72129	0.2863
0.5	0.65559	0.21476
0.7	0.52275	0.13918
0.9	0.30503	0.035249

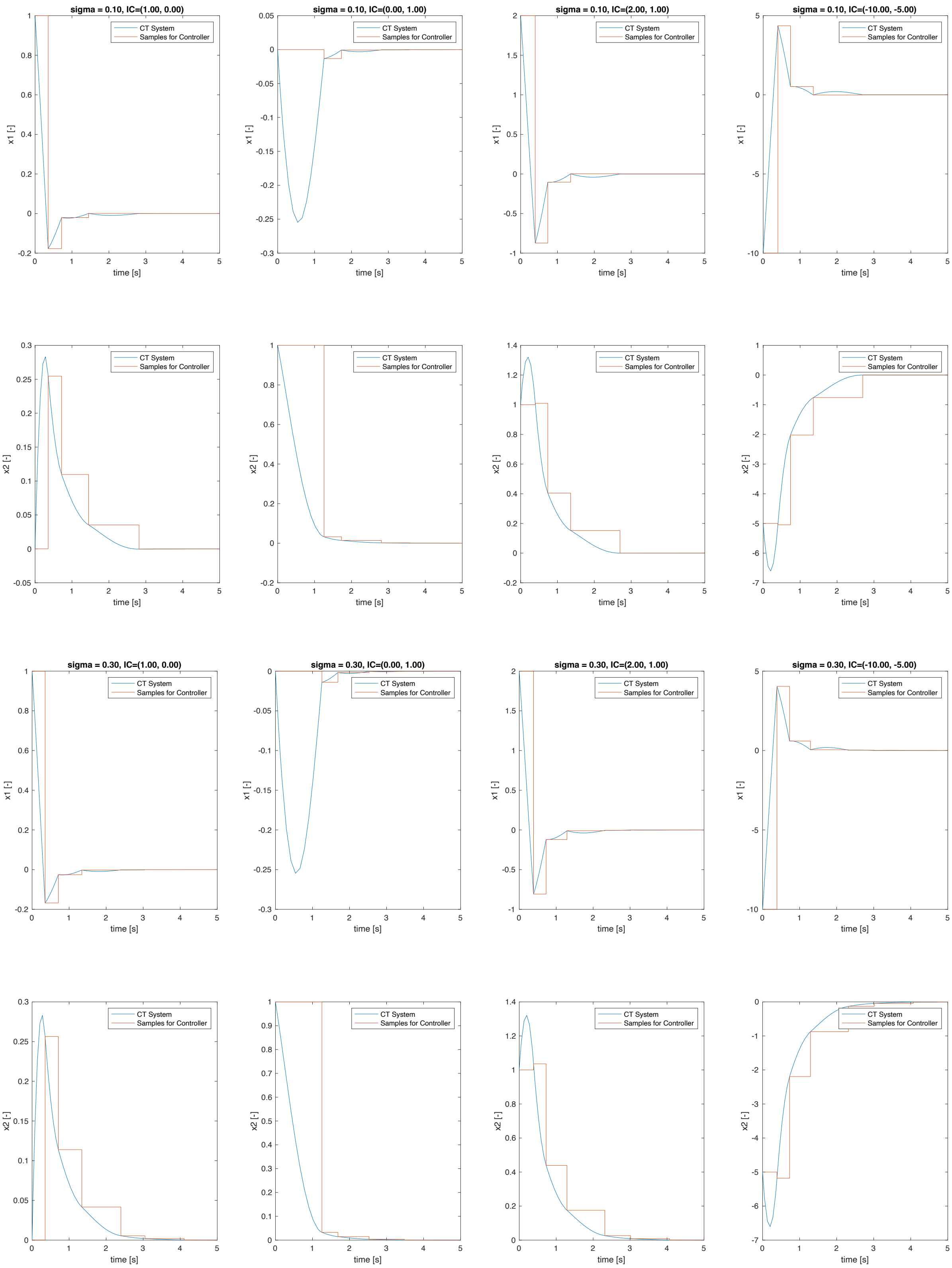
Table 4: Mean and Standard Deviation of Time Between Samples for Event Triggered Controller

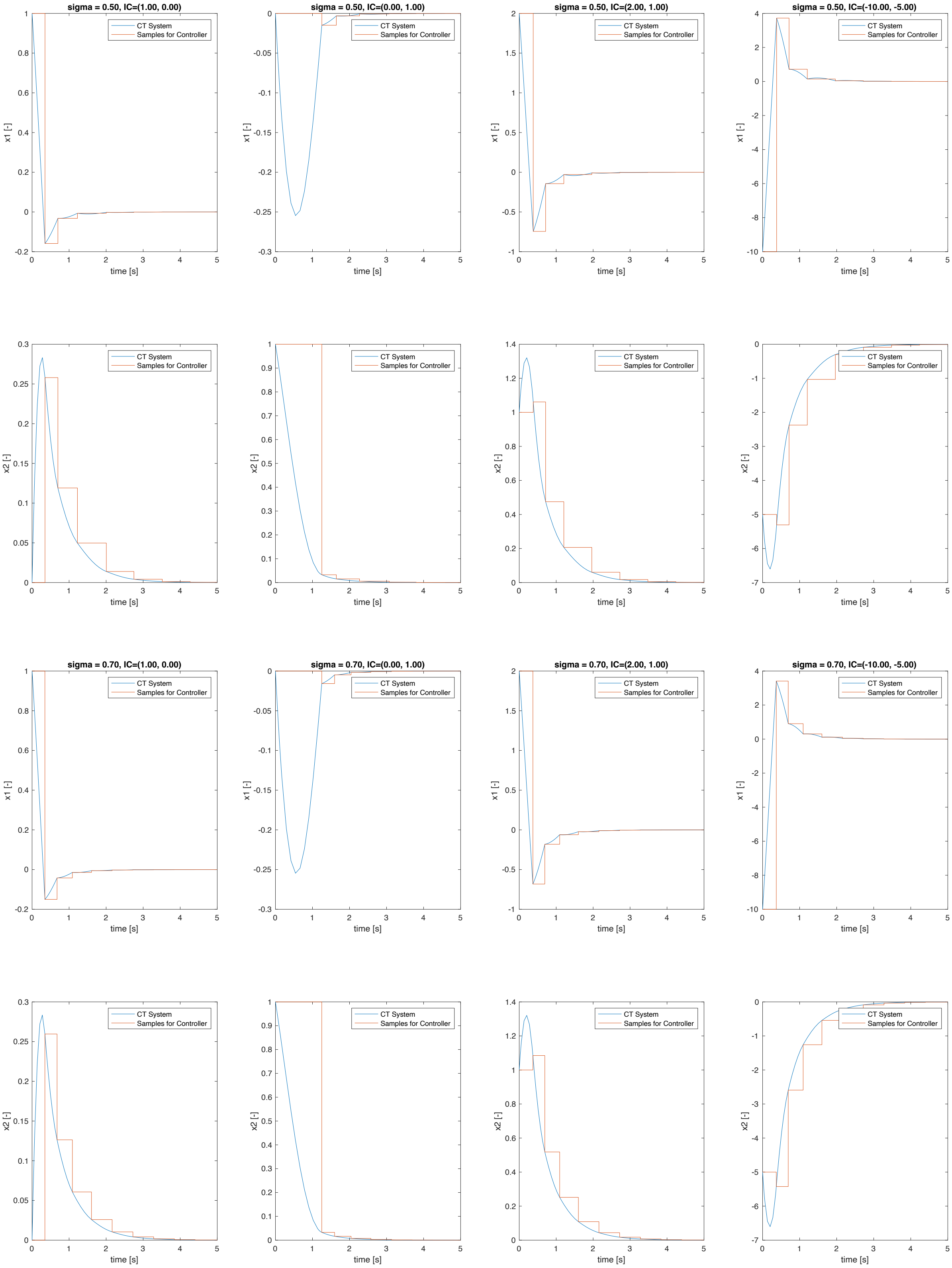
As expected, the more conservative controllers with lower σ values have a higher mean time between samples, as low-σ controllers sacrifice some performance(i.e. lower convergence rate) to have a lower frequency of sampling, while controllers with high σ values prioritize high convergence rates by sampling more often. In addition, the standard deviation of the low-σ controllers is significantly higher than the high-σ controllers. To some extent this is also unsurprising, since a controllers which is more lenient with its triggering condition can accommodate more fluctuation in the sampling time. The full plots of the simulation with respect to time across all σ and all initial conditions, in addition to the the $\xi(s_k)$ value, can be seen in Appendix A.

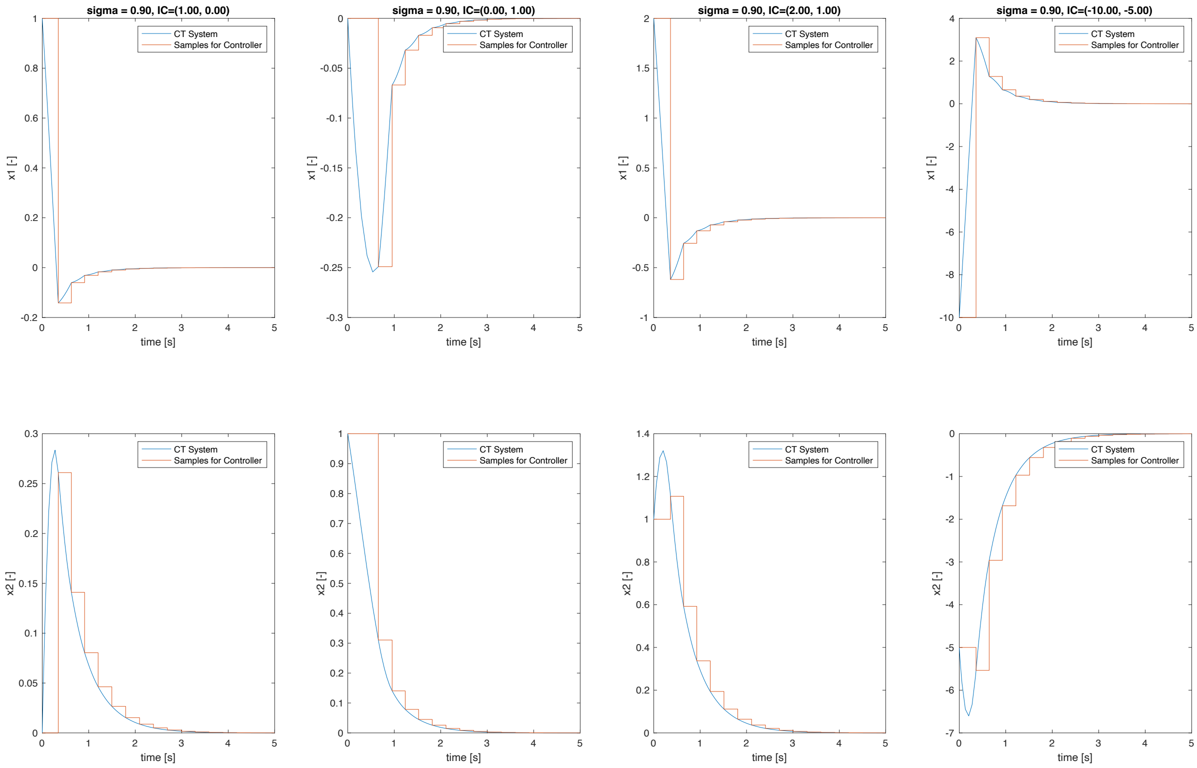
4.3: Using Average Sampling time for Periodic Sampling

Table 4 shows that the highest average time between samples, noted as h^*, is 0.7027s. Using this average sampling time, one can check if periodically sampling at this value would also maintain system stability. Since this controller uses a constant sampling time, MATLAB's `c2d()` function can be used to derive the F and G matrices of the discrete-time system from the A and B matrices of the continuous-time system. Then, one can check the spectral radius of the matrix (F-GK) to determine if the closed-loop discrete system is stable. Doing this with h^*, the closed-loop spectral radius is 1.4095, which means that the discrete-time system is unstable if sampled periodically at h^*. This confirms that the Event-triggered controller can achieve larger average sampling times while retaining stability and performance than a controller with a periodic sampling time.

Appendix A: Simulations of Event-Based Controllers




sigma = 0.50, IC=(1.00, 0.00)
sigma = 0.50, IC=(0.00, 1.00)
sigma = 0.50, IC=(2.00, 1.00)
sigma = 0.50, IC=(-10.00, -5.00)
sigma = 0.70, IC=(1.00, 0.00)
sigma = 0.70, IC=(0.00, 1.00)
sigma = 0.70, IC=(2.00, 1.00)
sigma = 0.70, IC=(-10.00, -5.00)
CT System
Samples for Controller
x1 [-]
x2 [-]
time [s]



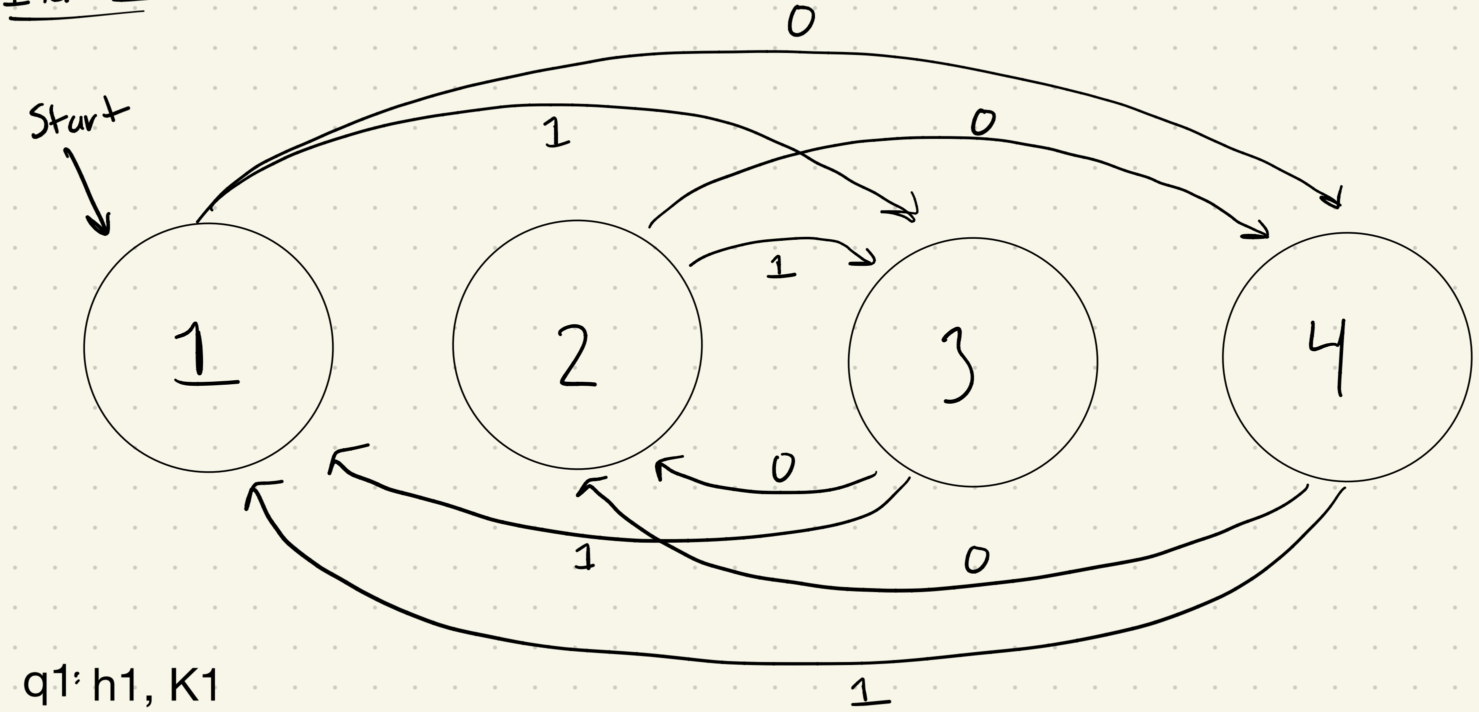
Appendix B: Handwritten Notes and Math

Networked & Distributed Control Systems Assignment 2

Q1 Omega - Automaton $\quad h_1 = h \quad h_2 = 2h$

- If h_1, Controller is either K_1 OR K_2
- If h_2, controller is either K_2 OR 0

Item 1



q1: h1, K1

q2: h1, K2

q3: h2, K2

q4: h2, 0

Item 2: Equations of switched System:

Integrating into state model (As in hw 2 solutions)

Say: $F_1(h) = (F(h) - G(h)K_1)$

$F_2(h) = (F(h) - G(h)K_2)$

$F_0 = F(h)$

$$x_{k+1} = \begin{cases} F_1(h_1)x_k & \text{if } q_k = 1 \\ F_2(h_1)x_k & \text{if } q_k = 2 \\ F_2(h_2)x_k & \text{if } q_k = 3 \\ F_0(h_2)x_k & \text{if } q_k = 4 \end{cases}$$

All ways to get from $q_1 \to q_1$:

$(1\,1)^\omega$, $(0\,1)^\omega$, $(1\,(01)^*1)^\omega$, $(0\,(01)^*1)$, $((10)^*(00)^*1)$, $(0(00)^*\omega)$,

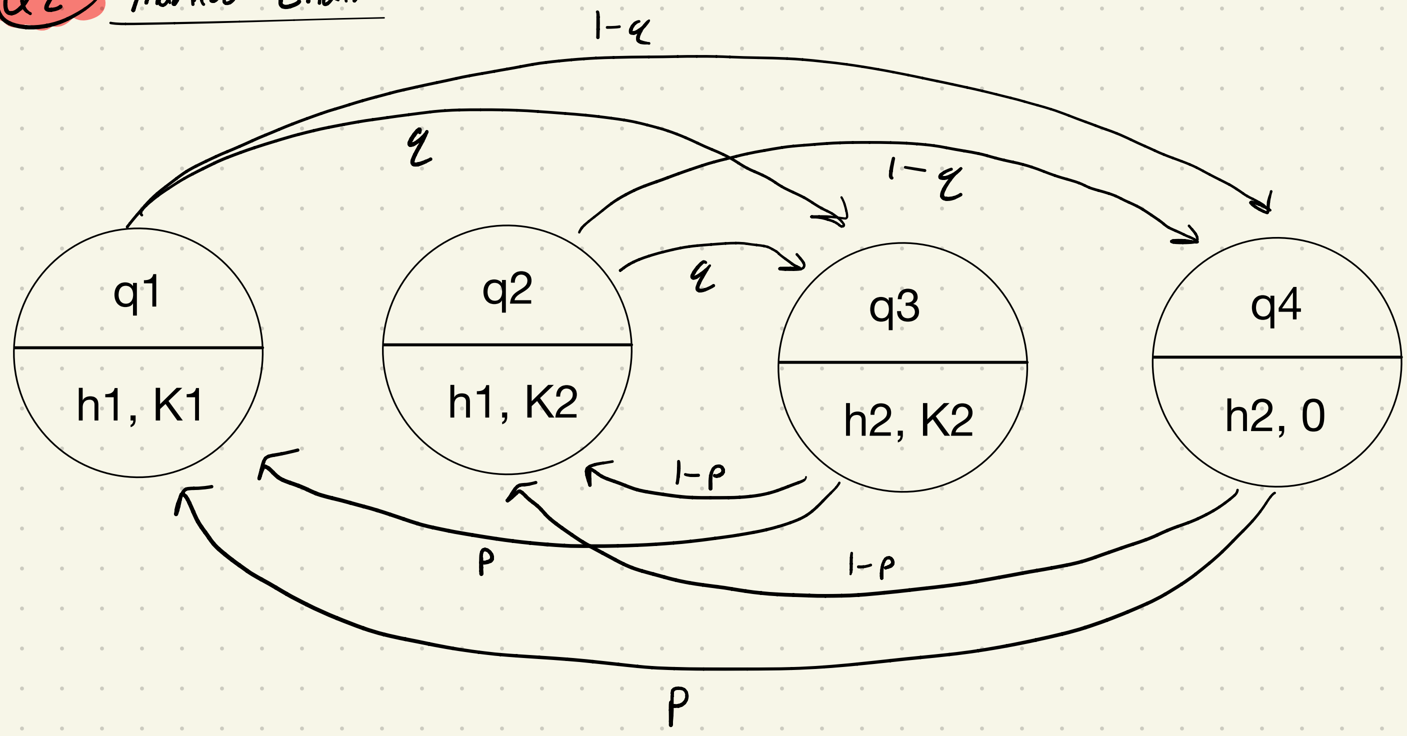
"The CMIs are created by considering the possible modes between visits to an accepting state"

All combos that get from state 1 back to state 1

①

State Transition	ω-notation	Equation to verify
$(1\to 3\to 1)^\omega$	$(1\,1)^\omega$	$x_{n+2} = F_1(h_1)\,F_2(h_2)\,x_n$
$(1\to 4\to 1)^\omega$	$(0\,1)^\omega$	$x_{n+2} = F_1(h_1)\,F_0(h_2)$
$(1\to 3\to 2\to 3\to 1)^\omega$	$(1\,(01)^*1)^\omega$	$x_{n+4} = F_1(h_1)\,F_2(h_2)\,F_1(h_2)\,F_2(h_2)$
$(1\to 4\to 2\to 3\to 1)^\omega$	$(0\,(01)^*1)^\omega$	$x_{n+4} = F_1(h_1)\,F_0(h_2)\,F_1(h_2)\,F_0(h_2)$
$(1\to 3\to 2\to 4\to 1)^\omega$	$((10)^*(00)^*1)^\omega$	$x_{n+4} = F_1(h_1)\,F_2(h_2)\,F_1(h_2)\,F_0(h_2)$
$(1\to 4\to 2\to 3\to 1)^\omega$	$(0(00)^*1)^\omega$	$x_{n+4} = F_1(h_1)\,F_0(h_2)\,F_1(h_2)\,F_2(h_2)$

Q2 Markov Chain



Item 2: Write Markov Jump Linear System

In the form of: $x_{k+1} = A_k x_k$ $\quad \mathbb{P}(q_k = j \mid q_{k-1} = i) = p_{ij}$

Say $A_1^{cl} = (F(h_1) - G(h_1)K)$ $\quad A_2^{cl} = (F(h_1) - G(h_1)K_2)$

$A_3^{cl} = (F(h_2) - G(h_2)K_2)$ $\quad A_4^{cl} = F(h_2)$

Integrating These Into Markov System:

$$x_{k+1} = \begin{cases} A_1^{cl} x_k & \text{if } q_k = 1 \\ A_2^{cl} x_k & \text{if } q_k = 2 \\ A_3^{cl} x_k & \text{if } q_k = 3 \\ A_4^{cl} x_k & \text{if } q_k = 4 \end{cases}$$

Probabilities

$p_{12} = 0$ $\quad p_{31} = p$

$p_{13} = q$ $\quad p_{32} = 1-p$

$p_{14} = 1-q$ $\quad p_{34} = 0$

$p_{21} = 0$ $\quad p_{41} = p$

$p_{23} = q$ $\quad p_{42} = 1-p$

$p_{24} = 1-q$ $\quad p_{43} = 0$

Item 3: Proving Stability of System

- Mean-Square Stability

3rd Condition For MSS:

$$P_i - \sum_{j=0}^{N} p_{ij} A_j^T P_j A_j \succ 0 \quad \forall i = 0 \dots N$$

for $i=1$: $P_1 - p_{13} A_3^T P_3 A_3 - p_{14} A_4^T P_4 A_4 > 0$

for $i=2$: $P_2 - p_{23} A_3 P_3 A_3 - p_{24} A_4^T P_4 A_4 \succ 0$

for $i=3$: $P_3 - p_{31} A_1^T P_1 A_1 - p_{32} A_2^T P_2 A_2 \succ 0$

for $i=4$: $P_4 - p_{41} A_1^T P_1 A_1 - p_{42} A_2^T P_2 A_2 \succ 0$

There exist some $P_1, P_2, P_3, P_4 \succ 0$ such that:

① $P_1 - q A_3^T P_3 A_3 - (1-q) A_4^T P_4 A_4 \succ 0$

② $P_2 - q A_3 P_3 A_3 - (1-q) A_4^T P_4 A_4 \succ 0$

③ $P_3 - \rho A_1^T P_1 A_1 - (1-\rho) A_2^T P_2 A_2 \succ 0$

④ $P_4 - \rho A_1^T P_1 A_1 - (1-\rho) A_2^T P_2 A_2 \succ 0$

All A_is are function of h!!!

Item 4: Integrate these Ins via CVX/~~MATLAB~~ Python ☺

Q3: Jordan Form

Item 1 pole-placement

$$A = \begin{bmatrix} 2.3 & 0 \\ 1 & 2.5 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad K = [K_1, K_2]$$

$$A_{cl} = \begin{bmatrix} 2.3 - K_1 & -K_2 \\ 1 & 2.5 \end{bmatrix} \qquad A_{cl} - \lambda I = \begin{bmatrix} 2.3 - K_1 - \lambda & -K_2 \\ 1 & 2.5 - \lambda \end{bmatrix}$$

$$\det(A_{cl} - \lambda I) = (2.3 - K_1 - \lambda)(2.5 - \lambda) + K_2$$
$$= \lambda^2 - 2.5\lambda - (2.3 - K_1)\lambda + 2.5(2.3 - K_1) + K_2$$
$$= \lambda^2 - 4.8\lambda + K_1\lambda + K_2 + 5.75 - K_1$$

For $\lambda = -2 \pm i$ $\qquad (\lambda + 2)^2 + 1^2 = \lambda^2 + 4\lambda + 4 + 1 = \lambda^2 + 4\lambda + 5$

$$K_1 - 4.8 = 4 \quad K_1 = 8.8$$
$$K_2 + 5.75 - 2.5K_1 = 5$$
$$K_2 = 5 + 2.5K_1 - 5.75 = 21.25$$

Now Redoing the part where you have to solve for the exact discrete-time stuff. :')

Recall:

$$x_{k+1} = e^{Ah} x_k + \left(\int_{h-\tau}^{h} e^{As} B\,ds\right) u_{k-1} + \left(\int_{0}^{h-\tau} e^{As} B\,ds\right) u_k$$

$$F_x(h) = e^{Ah} \qquad F_u(h,\tau) = \int_{h-\tau}^{h} e^{As} B\,ds \qquad G_1(h,\tau) = \int_0^{h-\tau} e^{As} B\,ds$$

$$F_x(h) = e^{Ah} = \begin{bmatrix} e^{2.3h} & 0 \\ 5e^{2.5h} - 5e^{2.3h} & e^{2.5h} \end{bmatrix}$$

$$e^{Ah}B = \begin{bmatrix} e^{2.3h} \\ 5e^{2.5h} - 5e^{2.3h} \end{bmatrix} \qquad \int e^{As}B\,ds = \begin{bmatrix} \frac{1}{2.3}e^{2.3s} \\ \frac{5}{2.5}e^{2.5s} - \frac{5}{2.3}e^{2.3s} \end{bmatrix}$$

$$F_u(h,\tau) = \int_{h-\tau}^{h} e^{As}B\,ds = \left.\begin{bmatrix} \frac{1}{2.3}e^{2.3s} \\ \frac{5}{2.5}e^{2.5s} - \frac{5}{2.3}e^{2.3s} \end{bmatrix}\right|_{h-\tau}^{h}$$

$$= \begin{bmatrix} \frac{1}{2.3}\left(e^{2.3h} - e^{2.3(h-\tau)}\right) \\ 2e^{2.5h} - 2e^{2.5(h-\tau)} - \frac{5}{2.3}e^{2.3h} + \frac{5}{2.3}e^{2.3(h-\tau)} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2.3}\left(e^{2.3h} - e^{2.3(h-\tau)}\right) \\ 2\left(e^{2.5h} - e^{2.5(h-\tau)}\right) + \frac{5}{2.3}\left(e^{2.3(h-\tau)} - e^{2.3h}\right) \end{bmatrix}$$

$$G_1(h,\tau) = \int_0^{h-\tau} e^{As}B\,ds = \left.\begin{bmatrix} \frac{1}{2.3}e^{2.3s} \\ \frac{5}{2.5}e^{2.5s} - \frac{5}{2.3}e^{2.3s} \end{bmatrix}\right|_{0}^{h-\tau}$$

$$= \begin{bmatrix} \frac{1}{2.3} e^{2.3(h-\tau)} - \frac{1}{2.3} \\ \frac{5}{2.5} e^{2.5(h-\tau)} - \frac{5}{2.3} e^{2.3(h-\tau)} - \left(\frac{5}{2.5} - \frac{5}{2.3}\right) \end{bmatrix}$$

$$= \begin{bmatrix} 0.435 e^{2.3(h-\tau)} - 0.4348 \\ 2 e^{2.5(h-\tau)} - 2.174 e^{2.3(h-\tau)} + 0.174 \end{bmatrix}$$

Something seems wrong: lets try one by hand:

If state is: x_1, x_2, v_{k-1}

Problem is controller!!!

Item 2: Polytropic Over-Estimation w/ Jordan Form Approach

Generally:

$$x^e_{k+1} = H(\theta_k) x^e_k$$

system dynamics

Uncertain parameters: τ, h

θ_k is set of parameters (infinite)

We have:

$$x^e_{k+1} = \begin{bmatrix} e^{Ah} & \int_{h-\tau_k}^{h} e^{As} ds B \\ 0 & 0 \end{bmatrix} x_k + \begin{pmatrix} \int_0^h e^{As} ds B \\ 1 \end{pmatrix} u_k$$

recall: $$e^{As} = Q^{-1} \left(\sum_{i=1}^{p} \sum_{j=0}^{q-1} \frac{s^j}{j!} e^{\lambda_i s} S_{ij} \right) Q$$

p is # of eigenvalues (i represents some eigenvalue)

q: is # of uncertain variable (j represent some uncertain variable)

We want to deconstruct system into:

$$x^e_{k+1} = \left(F_0 + \sum_{i=1}^{\nu} \alpha_i(\tau_k) F_i \right) x^e_k + \left(G_0 + \sum_{i=1}^{\nu} \alpha_i(\tau_k) G_i \right) u_k$$

Looking @

$$F = \begin{bmatrix} e^{2.3h} & 0 & \frac{1}{2.3}\left(e^{2.3h} - e^{2.3(h-\tau)}\right) \\ 5e^{2.5h} - 5e^{2.3h} & e^{2.5h} & 2\left(e^{2.5h} - e^{2.5(h-\tau)}\right) + \frac{5}{2.3}\left(e^{2.3(h-\tau)} - e^{2.3h}\right) \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} e^{2.3h} & 0 & \frac{1}{2.3}e^{2.3h} - \frac{1}{2.3}e^{2.3(h-\tau)} \\ 5e^{2.5h} - 5e^{2.3h} & e^{2.5h} & \left(2e^{2.5h} - \frac{5}{2.3}e^{2.3h}\right) + \left(\frac{5}{2.3}e^{2.3(h-\tau)} - 2e^{2.5(h-\tau)}\right) \\ 0 & 0 & 0 \end{bmatrix}$$

Items in red are functions of τ & therefore can be fit into the form: $\alpha_i(\tau_n) = \frac{(h-\tau_n)^j}{j!} e^{\lambda_i(h-\tau_n)}$

- note here that $j = 0$ in this case

$\alpha_i(\tau_n) = e^{\lambda_i(h-\tau_n)}$

say $\lambda_1 = 2.3$, $\lambda_2 = 2.5$

$\alpha_1(\tau_n) = e^{2.3(h-\tau)}$

$\alpha_2(\tau_n) = e^{2.5(h-\tau)}$

$$F_0(h) = \begin{bmatrix} e^{2.3h} & 0 & \frac{1}{2.3}e^{2.3h} \\ 5e^{2.5h} - 5e^{2.3h} & e^{2.5h} & \left(2e^{2.5h} - \frac{5}{2.3}e^{2.3h}\right) \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_1 = \begin{bmatrix} 0 & 0 & \left(-\frac{1}{2.3}\right) \\ 0 & 0 & \left(\frac{5}{2.3}\right) \\ 0 & 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F = F_0 + \alpha_1(\tau_h) F_1 + \alpha_2(\tau_h) F_2$$

Now For G:

$$G = \begin{bmatrix} 0.435e^{2.3(h-\tau)} - 0.4348 \\ 2e^{2.5(h-\tau)} - 2.174e^{2.3(h-\tau)} + 0.174 \end{bmatrix}$$

$$G_0 = \begin{bmatrix} -0.4348 \\ 0.174 \\ 1 \end{bmatrix} \quad G_1 = \begin{bmatrix} 0.435 \\ -2.174 \\ 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

$$G = G_0 + G_1\alpha_1(\tau_h) + G_2\alpha_2(\tau_h)$$

Now we have:

$$\alpha_1(\tau_h) = e^{2.3(h-\tau_h)} \qquad \underline{\alpha}_1 = e^{2.3(h-\tau_{max})}$$

$$\alpha_2(\tau_h) = e^{2.5(h-\tau_h)} \qquad \overline{\alpha}_1 = e^{2.3(h-\tau_{min})}$$

$$\underline{\alpha}_2 = e^{2.5(h-\tau_{max})}$$

$$\overline{\alpha}_2 = e^{2.5(h-\tau_{min})}$$

Vertices of the Polytope (AKA LMIs to check) 2 eigenvalues, so $\underline{5}$ LMIs to check

$$F = F_0 + \sum_{i=1}^{r} \alpha_i(t) F_i \mid \tau \in [\tau_{min}, \tau_{max}]$$

$$G = G_0 + \sum_{i=1}^{r} \alpha_i(\tau) G_i \mid \tau \in [\tau_{min}, \tau_{max}]$$

Get Over-Approximation via:

$$\overline{F} = F_0 + \sum_{i=1}^{r} \delta_i F_i \mid \delta_i \in [\underline{\alpha}_i, \overline{\alpha}_i] \qquad \overline{F} = F_0 + \delta_1 F_1 + \delta_2 F_2 \quad \begin{matrix} \delta_1 \in [\underline{\alpha}_1, \overline{\alpha}_1] \\ \delta_2 \in [\underline{\alpha}_2, \overline{\alpha}_2] \end{matrix}$$

$$\overline{G} = G_0 + \sum_{i=1}^{r} \delta_i G_i \mid \delta_i \in [\underline{\alpha}_i, \overline{\alpha}_i] \qquad \overline{G} = G_0 + \delta_1 G_1 + \delta_2 G_2 \quad \begin{matrix} \delta_1 \in [\underline{\alpha}_1, \overline{\alpha}_1] \\ \delta_2 \in [\underline{\alpha}_2, \overline{\alpha}_2] \end{matrix}$$

Combinations

$\underline{\alpha}_1, \underline{\alpha}_2$

$\underline{\alpha}_1, \overline{\alpha}_2$

$\overline{\alpha}_1, \underline{\alpha}_2$

$\overline{\alpha}_1, \overline{\alpha}_2$

$$H_{F,1} = \overline{F}(\delta_1 = \underline{\alpha}_1, \delta_2 = \underline{\alpha}_2)$$

$$H_{G,1} = \overline{G}(\delta_1 = \underline{\alpha}_1, \delta_2 = \underline{\alpha}_2)$$

$$① \quad (H_{F,1} - H_{G,1}K)^T P (H_{F,1} - H_{G,1}K) - P \preceq -\gamma P$$

$$H_{F,2} = \overline{F}(\delta_1 = \underline{\alpha}_1, \delta_2 = \overline{\alpha}_2)$$

$$H_{G,2} = \overline{G}(\delta_1 = \underline{\alpha}_1, \delta_2 = \overline{\alpha}_2)$$

$$② \quad (H_{F,2} - H_{G,2}K)^T P (H_{F,2} - H_{G,2}K) - P \preceq -\gamma P$$

$$H_{F,3} = \bar{F}(\delta_1 = \bar{\alpha}_1,\ \delta_2 = \underline{\alpha}_2)$$

$$H_{G,3} = \bar{G}(\delta_1 = \bar{\alpha}_1,\ \delta_2 = \underline{\alpha}_2)$$

③ $(H_{F,3} - H_{G,3}h)^T P (H_{F,3} - H_{G,3}h) - P \leq -\gamma P$

$$H_{F,4} = \bar{F}(\delta_1 = \bar{\alpha}_1,\ \delta_2 = \bar{\alpha}_2)$$

$$H_{G,4} = \bar{G}(\delta_1 = \bar{\alpha}_1,\ \delta_2 = \bar{\alpha}_2)$$

④ $(H_{F,4} - H_{G,4}h)^T P (H_{F,4} - H_{G,4}h) - P \leq -\gamma P$

⑤ $P = P^T \succ 0$

<u>Summarizing LMIs</u>

① $(H_{F,1} - H_{G,1}h)^T P (H_{F,1} - H_{G,1}h) - P \leq -\gamma P$

② $(H_{F,2} - H_{G,2}h)^T P (H_{F,2} - H_{G,2}h) - P \leq -\gamma P$

③ $(H_{F,3} - H_{G,3}h)^T P (H_{F,3} - H_{G,3}h) - P \leq -\gamma P$

④ $(H_{F,4} - H_{G,4}h)^T P (H_{F,4} - H_{G,4}h) - P \leq -\gamma P$

⑤ $P = P^T \succ 0$

Q: is γ a variable here or some arbitrarily small value?

(Q4) Back to original A, B, K:

Designing an Event-Driven Controller:

trying to retain GES!

Remember that System $(A - BK)$ is GES if

$\exists\ V(x) = x^T P x$ w/ $P \succ 0$ such that

$$(A - BK)^T P (A - BK) - P = -Q$$

how do we find a triggering condition Φ_{ETC}?

To get desired performance:

$$\frac{d}{dt} V(\xi(t)) \leq -\sigma\, \xi(t)^T Q\, \xi(t) \quad \sigma \in (0,1)$$

but w/ $\dot{\xi}(t) = A\xi(t) - BK\xi(s_k)$,