Networked and Distributed Control: Assignment 2

Niels Stienen, 5595738

I. Question 1

> Taking the case from Assignment 1, we assume now that whenever there is a collision, system 1's packet may or may not be lost and system 2 always looses every other packet. Create a ω-automaton that models this behaviour.

To create a ω-automaton that combines both systems into 1 model the difference in sampling interval first needs to be accounted for. In the automaton provided in Figure 1 this difference in sampling interval has been shown by using a empty transition for system 2 in intermediate steps. A possible alternative would be to imply that those packets are all dropped as this would result in the same system evolution.

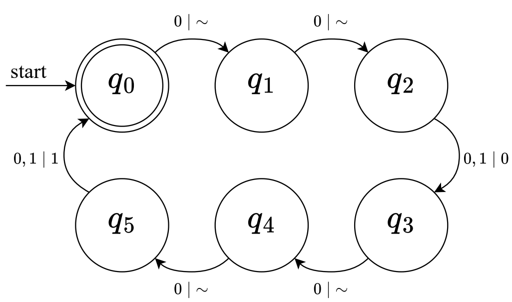


Fig. 1. ω-automaton of the combined systems. The notation is a 0 for no packet dropped, 1 for a dropped packet, and $\sim$ for no packet sent. With $sys_1|sys_2$ denoting which symbol belongs to which system at each transition.

> Write down equations of the switched system describing the dynamics of the received samples using this network model.

To simplify the notation used the following symbols are introduced for the closed-loop dynamics: F_0^i, F_1^i which represent no packet being dropped and a packet being dropped respectively. The superscript i indicates whether the symbol represents system 1 or 2. Below are the equations describing the dynamics where q_k is the automaton state keeping track of when system 2 should receive a packet transmission and ensures that whether the packet is dropped is alternated.

$$x_{k+1}^1 = F_0^1(h)x_k^1 \quad \text{if} \quad q_k \in \{1,2,4,5\} \tag{1}$$

$$x_{k+1}^1 = \begin{cases} F_0^1(h)x_k^1 & \text{if} \quad q_k \in \{3,6\} \\ F_1^1(h)x_k^1 & \text{if} \quad q_k \in \{3,6\} \end{cases} \tag{2}$$

$$x_{k+1}^2 = \begin{cases} F_0^2(3h)x_k^2 & \text{if} \quad q_k = 3 \\ F_1^2(3h)x_k^2 & \text{if} \quad q_k = 6 \end{cases} \tag{3}$$

$$q_{k+1} = (q_k \bmod 6) + 1 \tag{4}$$

> Provide LMI's to verify stability of your closed-loop system.

The LMI's for verifying system 1's closed-loop stability are:

$$P \succ 0 \tag{5}$$

$$F_0^1(h)^T F_0^1(h)^T F_0^1(h)^T P F_0^1(h) F_0^1(h) F_0^1(h) - P \prec 0 \tag{6}$$

$$F_0^1(h)^T F_0^1(h)^T F_1^1(h)^T P F_1^1(h) F_0^1(h) F_0^1(h) - P \prec 0 \tag{7}$$

For system 2 the LMI's needed are equivalent to checking the spectral radius:

$$P \succ 0 \tag{8}$$

$$F_0^2(3h)^T F_1^2(3h)^T P F_1^2(3h) F_0^2(3h) - P \prec 0 \tag{9}$$

$$\rho(F_1^2(3h)F_0^2(3h)) < 1 \tag{10}$$

To describe the dynamics of the combined system as one set of equations the state vector is extended and an additional state p_k is used to keep track of the sampling interval:

$$x_k^e = \begin{bmatrix} x_k^1 & x_k^2 \end{bmatrix}^T \tag{11}$$

with the closed-loop relation between x_{k+1}^e and x_k^e defined as:

$$x_{k+1}^e = A_{i,j}(h, p_k)x_k^e \quad \text{for } i \in \{0,1\}, j \in \{0,1\} \tag{12}$$

$$p_{k+1} = (p_k \bmod 3) + 1 \tag{13}$$

where F is defined as in Question 1 and

$$A_{i,j}(h, p_k) = \begin{cases} \begin{bmatrix} F_i^1(h) & 0 \\ 0 & F_j^2(3h) \end{bmatrix} & \text{if} \quad p_k = 3 \\ \begin{bmatrix} F_i^1(h) & 0 \\ 0 & I \end{bmatrix} & \text{otherwise} \end{cases} \tag{14}$$

To keep the notation compact the possible sub-sequences is defined

$$seq_{(i,j)} = A_{i,j}(h,3)A_{0,0}(h,2)A_{0,0}(h,1) \tag{15}$$

The LMI's needed for verifying stability of the combined system then are

$$P \succ 0 \tag{16}$$

$$seq_{(0,0)}^T seq_{(0,1)}^T P seq_{(0,1)} seq_{(0,0)} - P \prec 0 \tag{17}$$

$$seq_{(0,0)}^T seq_{(1,1)}^T P seq_{(1,1)} seq_{(0,0)} - P \prec 0 \tag{18}$$

$$seq_{(1,0)}^T seq_{(0,1)}^T P seq_{(0,1)} seq_{(1,0)} - P \prec 0 \tag{19}$$

$$seq_{(1,0)}^T seq_{(1,1)}^T P seq_{(1,1)} seq_{(1,0)} - P \prec 0 \tag{20}$$

> For what range of sampling intervals h is System 1 stable in this scenario? Which method is better, to zero or to hold?

While the question asks for the range of System 1 only, looking at the context of the other questions it makes more sense to look at the range for which both systems are stable. Therefore, both of the above are provided in this section.

The range of sampling intervals for which system 1 is stable is decreased compared to that of question 5 in assignment 1. For the to-hold method, the range was found to be $h \in (0, 0.3131]$ and for the to-zero method the range was found to be $h \in (0, 0.3636]$ as illustrated in Figure 2. The better method is still to-zero just like in the previous question.

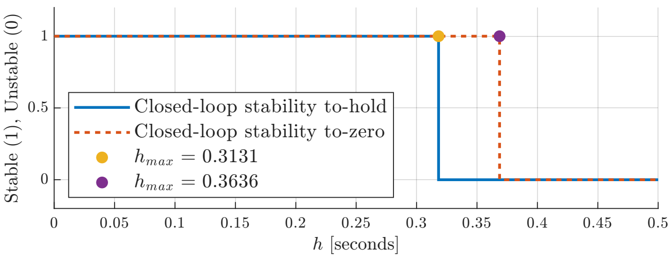


Fig. 2. Inner approximation of sampling intervals for which the cases to-hold and to-zero are stable for system 1.

For the combined system the range of sampling intervals for which both systems are stable is slightly decreased compared to assignment 1 for both the to-zero and to-hold cases. For the to-hold method, the range was found to be $h \in (0, 0.1364]$ and for the to-zero method the range was found to be $h \in (0, 0.2323]$ as illustrated in Figure 3. To-zero is clearly the better method for the combined system.

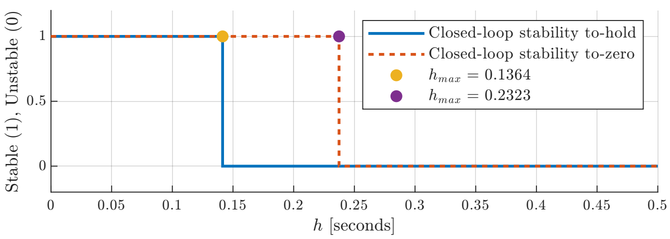


Fig. 3. Inner approximation of sampling intervals for which the cases to-hold and to-zero are stable for the combined systems

Is the range you have obtained larger or smaller than the one in Question 5 from assignment 1?

As already noted the range is smaller than that from the previous assignment for both system 1 and the combined system. This is to be expected as the sequence in assignment 1 only lost a packet every 6th sampling instance while now we need to account for the possibility of losing a packet every 3rd sampling instance which is a stricter scenario. Since the situation in this question includes that of the previous assignment in the possibilities stability for this situation implies stability for that of the situation in assignment 1 as well.
When only looking at system 1 the decrease of sampling intervals for which the system is stable is pretty significant, from 0.6096 to 0.3636 in the to-zero case. However, for the combined system the decrease is not as significant, from 0.2362 to 0.2323 in the to-zero case. In assignment 1 the stable range was already limited by system 2 quite heavily, so the change in behaviour of system 1 does not have as much of an effect.

II. Question 2

In this model, compared to those in Assignment 1, we add probabilities. Assume that whenever there is no collision, the probability of each system losing a packet is independent and equal to 0.01 (1%); while when there is a collision, the probability of only System 1 losing a packet is 0.49 (49%) and the same holds for System 2, and there is a 0.02 probability of both of them loosing their packet. For this question consider only the to-hold case. Create a Markov Chain that models this behaviour.

The Markov chain is shown in Figure 4, note that to reduce the clutter of arrows the collision step is placed in the middle. This has no effect other than making the Figure more readable.

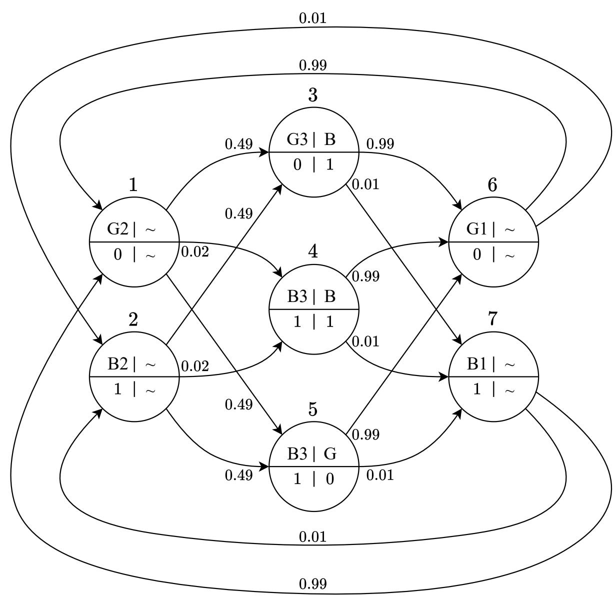


Fig. 4. Markov chain of the combined system

Write down the equations of the Markovian Jump Linear System describing the dynamics of the received samples using this network model.

To describe the dynamics of the combined system as one set of equations some changes are made compared to the extended system defined in Question 1.

$$x_k^e = \begin{bmatrix} x_k^1 & x_k^2 \end{bmatrix}^T \tag{21}$$

with the closed-loop relation between x_{k+1}^e and x_k^e defined as:

$$x_{k+1}^e = A_{i,j}(h, p_k)x_k^e \quad \text{for } i \in \{0,1\}, j \in \{0,1\} \tag{22}$$
$$q_{k+1} = (q_k \bmod 3) + 1 \tag{23}$$

where F is defined as in Question 1 and

$$A_{i,j}(h,q_k) = \begin{cases} \begin{bmatrix} F_i^1(h) & 0 \\ 0 & F_j^2(3h) \end{bmatrix} & \text{if} \quad q_k = 3 \\ \begin{bmatrix} F_i^1(h) & 0 \\ 0 & I \end{bmatrix} & \text{otherwise} \end{cases} \tag{24}$$

From here the Markovian Jump Linear System can easily be derived from the Markov chain as:

$$\begin{aligned}
x_{k+1}^e &= A_{0,j}(h,1)x_k^e,\ \mathbb{P}(q_k = 1|q_{k-1} \in \{6,7\}) = 0.99 \\
x_{k+1}^e &= A_{1,j}(h,1)x_k^e,\ \mathbb{P}(q_k = 2|q_{k-1} \in \{6,7\}) = 0.01 \\
x_{k+1}^e &= A_{0,1}(h,2)x_k^e,\ \mathbb{P}(q_k = 3|q_{k-1} \in \{1,2\}) = 0.49 \\
x_{k+1}^e &= A_{1,1}(h,2)x_k^e,\ \mathbb{P}(q_k = 4|q_{k-1} \in \{1,2\}) = 0.02 \\
x_{k+1}^e &= A_{1,0}(h,2)x_k^e,\ \mathbb{P}(q_k = 5|q_{k-1} \in \{1,2\}) = 0.49 \\
x_{k+1}^e &= A_{0,j}(h,3)x_k^e,\ \mathbb{P}(q_k = 6|q_{k-1} \in \{3,4,5\}) = 0.99 \\
x_{k+1}^e &= A_{1,j}(h,3)x_k^e,\ \mathbb{P}(q_k = 7|q_{k-1} \in \{3,4,5\}) = 0.01
\end{aligned}$$

Choose a stochastic stability notion and provide LMI's to verify stability of your closed-loop system.

The stochastic stability notion chosen is the Mean Square Stability (MSS). A Markovian Jump Linear System is Mean Square Stable if there exist $P_i \succ 0$ for $i = 1,2,...,N$ that satisfies the condition:

$$P_i - \sum_{j=1}^{7} p_{ij} A_j^T P_j A_j \succ 0,\ \forall i = \{1,2,...,7\} \tag{25}$$

with the matrix p_{ij} derived from the MJLS above as:

$$p_{ij} = \begin{bmatrix} 0 & 0 & 0.49 & 0.02 & 0.49 & 0 & 0 \\ 0 & 0 & 0.49 & 0.02 & 0.49 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.99 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0.99 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0.99 & 0.01 \\ 0.99 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0.99 & 0.01 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

The LMI's then become $P_i \succ 0\ \forall i = \{1,2,...,7\}$ and

$$\begin{aligned}
P_1 - \left(p_{13}A_{01}^T P_3 A_{01} + p_{14}A_{11}^T P_4 A_{11} + p_{15}A_{10}^T P_5 A_{10}\right) \succ 0 \\
P_2 - \left(p_{23}A_{01}^T P_3 A_{01} + p_{24}A_{11}^T P_4 A_{11} + p_{25}A_{10}^T P_5 A_{10}\right) \succ 0 \\
P_3 - \left(p_{36}A_{0j}^T P_6 A_{0j} + p_{37}A_{1j}^T P_7 A_{1j}\right) \succ 0 \\
P_4 - \left(p_{46}A_{0j}^T P_6 A_{0j} + p_{47}A_{1j}^T P_7 A_{1j}\right) \succ 0 \\
P_5 - \left(p_{56}A_{0j}^T P_6 A_{0j} + p_{57}A_{1j}^T P_7 A_{1j}\right) \succ 0 \\
P_6 - \left(p_{61}A_{0j}^T P_1 A_{0j} + p_{62}A_{1j}^T P_2 A_{1j}\right) \succ 0 \\
P_7 - \left(p_{71}A_{0j}^T P_1 A_{0j} + p_{72}A_{1j}^T P_2 A_{1j}\right) \succ 0
\end{aligned}$$

Solving this set of LMI's results in a stable range of sampling intervals $h \in (0, 0.0707]$ as can be seen in Figure 5.

Comment on the differences between the range of stabilizing sampling intervals h that you obtained now with respect to the ones in Question 1 of this assignment and Question 5 of assignment 1.

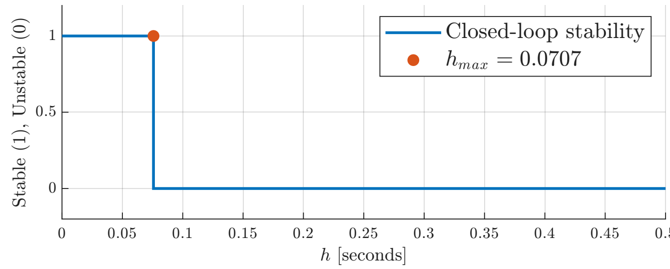


Fig. 5. Inner approximation of sampling intervals for which the MJLS is stable for the combined systems

The range of stabilizing sampling intervals obtained in this question is smaller than those obtained in previous questions. The table below shows the ranges for each question's combined systems in the to-hold case.

Question	Approach	Range of stable intervals
A1, Q5	to-hold	$h \in (0, 0.1401]$
A2, Q1	to-hold	$h \in (0, 0.1364]$
A2, Q2	to-hold	$h \in (0, 0.0707]$

As to why these differences are present is difficult to compare as the system in this question is different from those before. The LMI's used here do not contain the LMI's described in the previous questions nor do those of the previous questions contain the LMI's described here.
However comparing the systems themselves it could be expected that the MJLS would have a smaller range of stabilizing sampling intervals. For the previous questions, both systems were guaranteed to never lose 2 packets in a row. For the MJLS this is theoretically possible for both system 1 and system 2. Although the chances are small this would impact the range of stabilizing sampling intervals. An even more significant change is that system 2 has switched from always losing every other packet, to having a 51% chance of losing a packet at every sampling instant.

III. Question 3

Next we ignore System 2 and consider the case when delays are present in the networked system, but System 1 is no longer subject to packet drop-outs. 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.
Employ the Jordan form approach 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?

The system dynamics for a model affected by a constant delay $\tau \in [0, h)$ is given as:

$$x_{k+1}^e = F(\tau_k)x_k^e + G(\tau_k)u_k \tag{27}$$

$$x_{k+1}^e = \begin{bmatrix} e^{Ah} & \int_{h-\tau_k}^{h} e^{As}ds B \\ 0 & 0 \end{bmatrix} x_k^e + \begin{bmatrix} \int_0^{h-\tau} e^{As}ds B \\ I \end{bmatrix} u_k \tag{28}$$

with $x_k^e = \begin{bmatrix} x_k & u_{k-1} \end{bmatrix}^T$. Applying the Jordan form the matrix A is expressed as $Q^{-1}JQ$ where Q, J are:

$$Q = \begin{bmatrix} 1 & \frac{75}{47} \\ 0 & 1 \end{bmatrix}, \qquad J = \begin{bmatrix} -3.7 & 0 \\ 0 & 1 \end{bmatrix} \tag{29}$$

As can be seen in the Jordan matrix both eigenvalues are real with multiplicity 1, thus the equation from Lecture 4, slide 29 can be used:

$$e^{As} = Q^{-1}\left(\sum_{i=1}^{p}\sum_{j=0}^{q_i-1}\frac{s^j}{j!}e^{\lambda_i s}S_{i,j}\right)Q \tag{30}$$

For the given system $p = 2$ and $q_i = 1$ which results in the expression:

$$e^{As} = Q^{-1}\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} e^{-3.7s} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} e^{s}\right)Q \tag{31}$$

Combining the equations above the matrices $F(\tau_k)$ and $G(\tau_k)$ can be written as

$$F(\tau_k) = F_0 + \alpha_1(\tau_k)F_1 + \alpha_2(\tau_k)F_2 \tag{32}$$

$$G(\tau_k) = G_0 + \alpha_1(\tau_k)G_1 + \alpha_2(\tau_k)G_2 \tag{33}$$

where

$$\alpha_1(\tau_k) = e^{-3.7(h-\tau_k)}, \quad \alpha_2(\tau_k) = e^{(h-\tau_k)} \tag{34}$$

$$F_0 = \begin{bmatrix} e^{-3.7h} & \frac{75}{47}\left(e^{-3.7h} - e^{h}\right) & -\frac{750}{1739}e^{-3.7h} - \frac{75}{47}e^{h} \\ 0 & e^{h} & e^{h} \\ 0 & 0 & 0 \end{bmatrix} \tag{35}$$

$$F_1 = \begin{bmatrix} 0 & 0 & \frac{750}{1739} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0 & 0 & \frac{75}{47} \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{36}$$

$$G_0 = \begin{bmatrix} \frac{75}{37} \\ -1 \\ 1 \end{bmatrix}, \quad G_1 = \begin{bmatrix} \frac{-750}{1739} \\ 0 \\ 0 \end{bmatrix}, \quad G_2 = \begin{bmatrix} \frac{-75}{47} \\ 1 \\ 0 \end{bmatrix} \tag{37}$$

Since the only uncertain entries are the terms α_1 and α_2 it is enough to inspect the plot in Figure 6. Noticing that the curve is convex, it is sufficient to use three vertices to obtain the over-approximation polytope.

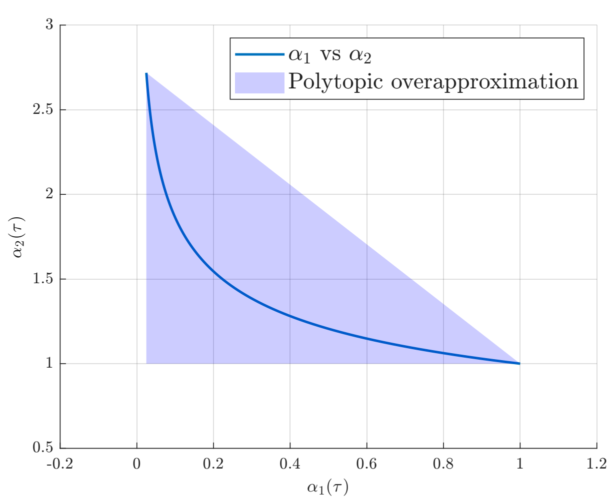


Fig. 6. Plot comparing α_1 vs α_2

Perform stability analyses, solving the relevant LMI's resulting from the polytopic over-approximation. Produce a plot illustrating combinations of (h, τ) as in Question 2 of assignment 1.

The LMI's needed for the stability analysis are derived from the over-approximations denoted by

$$\bar{\mathcal{F}} = \left\{ F_0 + \sum_{i=1}^{r}\delta_i F_i \middle| \delta_i \in [\underline{\alpha}_i, \bar{\alpha}_i], i = 1, 2, .., r \right\} \tag{38}$$

$$\bar{\mathcal{G}} = \left\{ G_0 + \sum_{i=1}^{r}\delta_i G_i \middle| \delta_i \in [\underline{\alpha}_i, \bar{\alpha}_i], i = 1, 2, .., r \right\} \tag{39}$$

This results in $(2^r + 1)$ LMI's described by

$$P \succ 0 \tag{40}$$

$$A_{cl}^T P A_{cl} - P \preceq -\gamma P \ \ \forall\, \underline{\alpha}_i, \bar{\alpha}_j,\ i, j \in \{1, 2\} \tag{41}$$

The resulting range of sampling intervals for which the system is stable is $h \in [0, 0.1616]$.

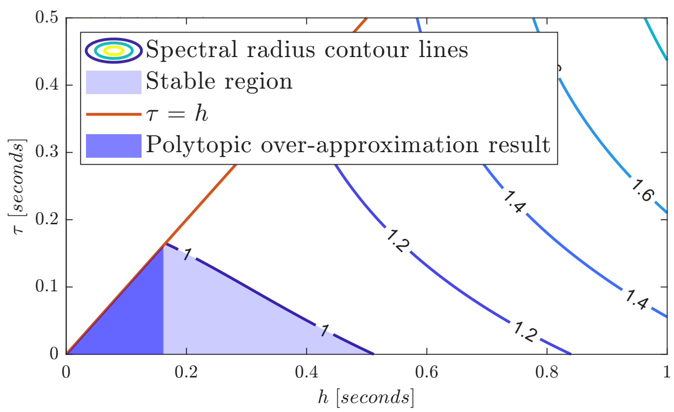


Fig. 7. Stable region found through solving the LMI's resulting from the polytopic over-approximation overlaid on the plot from Question 2, assignment 1.

Refine the plot of the previous question, by employing a smaller polytope that over-approximates the uncertain system, but with more vertices.

To reduce the size of the polytope the shape is changed from a triangle to a trapezoid. This is done by finding the slope between the vertices of the triangle and finding for which point this slope is tangent. Then the points on the intersection of this line and the edges of the polytope are used as the vertices for the smaller polytope. Figure 8 shows the new polytope.

The range of sampling intervals for which the smaller polytope is stable is $h \in [0, 0.1641]$. This is a slightly better result although much improvement was not expected as the initial polytope was already approaching the limit found in assignment 1.

Compare the resulting plot with the one from assignment 1 and discuss any possible differences and the possible causes thereof.

Looking at Figure 7 one can see that the region for which the over-approximation method is stable is slightly smaller than

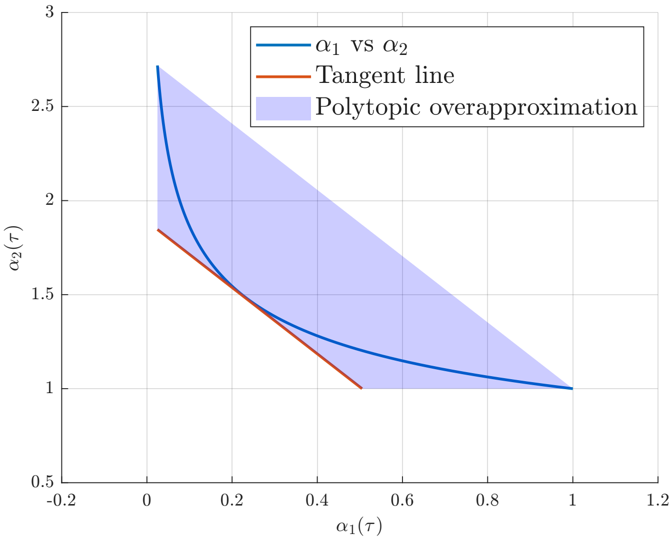


Fig. 8. Plot of the smaller polytope that over-approximates the uncertain system

using the spectral radius method from assignment 1. This is to be expected as the over-approximation is a more conservative approach to proving stability. Additionally, in assignment 1 we also considered how big the delay τ could be for any given sampling interval h while here we only consider if the system is stable for all $\tau \in (0, h)$ for any given h.

IV. Question 4

Finally we go back to the case with no delays to design an event-triggered controller.
Consider your system being controlled by the first static controller you designed for your continuous-time system. Design a quadratic event-triggered condition $\phi(\xi(t), \xi(s_k)) \le 0$ for the system to guarantee global exponential stability of the closed-loop.

To design a event-triggered controller we first find a $P = P^T \succ 0$ that satisfies

$$(A - BK)^T P + P(A - BK) = -Q \tag{42}$$

for some $Q \succ 0$. Since the closed-loop $A - BK$ was designed such that it is globally exponentially stable there always exists such a P.

Letting the desired performance be:

$$\frac{\mathrm{d}}{\mathrm{d}t} V(\xi(t)) \le -\sigma \xi^T(t) Q \xi(t) \tag{43}$$

The performance equation can be rewritten giving the quadratic event-triggered condition as:

$$\phi(\xi(t), \xi(s_k)) := \begin{bmatrix} \xi(t) \\ \xi(s_k) \end{bmatrix}^T \begin{bmatrix} A^T P + PA + \sigma Q & -PBK \\ -K^T B^T P & 0 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \xi(s_k) \end{bmatrix} \tag{44}$$

For implementation in matlab this condition is rewritten by introducing the error $\epsilon(t) = \xi(s_k) - \xi(t)$ resulting in the quadratic event-triggered condition as:

$$\phi(\xi(t), \epsilon(s_k)) := \begin{bmatrix} \xi(t) \\ \epsilon(s_k) \end{bmatrix}^T \begin{bmatrix} (1-\sigma)Q & PBK \\ K^T B^T P & 0 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \epsilon(s_k) \end{bmatrix} \tag{45}$$

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 systems (different σ values) on a predefined fixed time-length of the simulations.

The above event-triggered condition has been implemented in matlab making use of the event option in the ode45 solver. Making use of the parallel processing toolbox multiple simulations were run simultaneously to get good estimates for different values of σ.

First, simulations were run for 10 different initial conditions over the entire range of $\sigma \in (0, 1)$ (Figure 9). It became obvious that high values of σ were not optimal as on average there were 1000+ communications per simulation. To get a better estimate the simulations were run again for the range of $\sigma \in (0, 0.3)$ (Figure 10) as this was clearly where the best σ would be found. For this reduced range 500 simulations were done for each value of σ resulting in the lowest amount of communications per sampling being found for $\sigma = 0.039$. At the end of this report the state trajectories of some of the initial conditions tested can be found. For all simulations tested the state trajectories showed that the feedback still stabilizes the system. On page 7 various state trajectories are shown for different values of σ (left column) and varying initial conditions (right column). For each of the state trajectories at the end of this report, it was intentionally chosen only to vary 1 type of parameter to make for better comparisons between trajectories. From this, it can be observed that for larger values of σ, the system is sampled more frequently and the error state is bound to be smaller. The different initial conditions show that the event triggers mainly occur when one or both states are near zero equilibrium.

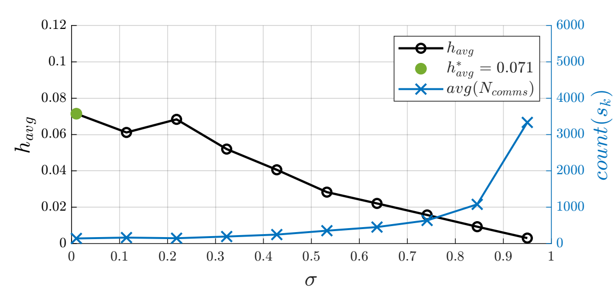


Fig. 9. Plot of the smaller polytope that over-approximates the uncertain system

Let h^*_{avg} be the average inter-sample time for the value of σ in the previous part that resulted in the lowest amount of communications/sampling. Does periodic sampling with $h = h^*_{avg}$ stabilize also your closed-loop system?

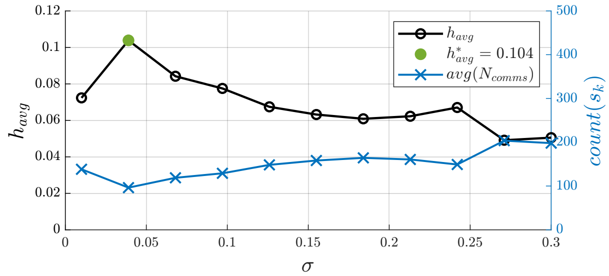


Fig. 10. Plot of the smaller polytope that over-approximates the uncertain system

The average inter-sample time for $\sigma = 0.039$ samples the system on average every 0.104 seconds. To verify whether periodic sampling with this interval stabilizes the closed-loop system the spectral radius can be used.

$$\rho(F(h^*_{avg}) - G(h^*_{avg})\bar{K}) < 1 \tag{46}$$

The spectral radius for the closed-loop system is $\rho(A_{cl}(h^*_{avg}) = 0.9265 < 1$ thus the periodic sampling with $h = h^*_{avg}$ also stabilizes the closed-loop system

While it is not sufficient to prove stability it is interesting to run a significant number of simulations to see if the result still holds. After running a 1000 simulations in matlab the result was that they all stabilized which is to be expected as the spectral radius is less than 1.

Consider now that the system is affected by a bounded disturbance d, i.e. $\dot{\xi}(t) = A\xi(t) + Bv(t) + d(t), \|d(t)\| < D \ \ \forall t$. Simulate again the closed-loop ETC system, now affected by the disturbance, for a sufficiently long time. Simulate for at least three different realizations of the disturbance. Discuss what you observe now happening with the system trajectory and the inter-sample times.

The three different disturbances are chosen to be:

$$d_1 = 0.1 \begin{bmatrix} sin(t) & cos(t) \end{bmatrix}^T \tag{47}$$

$$d_2 = 0.01 \begin{bmatrix} rand() & rand() \end{bmatrix}^T \tag{48}$$

$$d_3 = 0.1 \begin{bmatrix} sin(t) & cos(t) \end{bmatrix}^T \tag{49}$$

The inter-sample times change depending on the type of disturbance, for the disturbance d_1 the average inter-sample time increased while for the other disturbances, it decreased over the range of $\sigma \in (0, 1)$ as can be seen in Figure 11.

For the plots used to compare state trajectories, the reader is referred to Page 8. The state trajectory and error evolution of both no disturbance and the random noise disturbance are similar and converge to the equilibrium relatively quickly. For the other types of disturbances, the transient disappears within the first approximately 5 seconds after which both the state evolution and event triggering occur periodically.

Modify the triggering condition to take the form $\phi(\xi(t), \xi(s_k)) \leq \epsilon$, for some $\epsilon \in \mathbb{R}^+$ of your choice. Simulate again for the same initial conditions as in the previous question. How do the trajectories and inter-sample times differ now from the previous case when ϵ was zero?

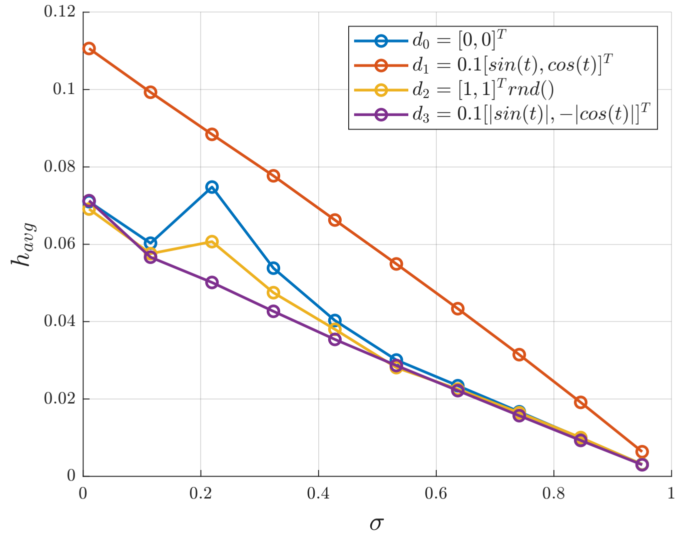


Fig. 11. Average inter-sample time for different disturbances compared to the case of no disturbance

Changing the value of ϵ to be larger than zero resulted in the system being triggered less often while retaining stability for values of $\epsilon \ll 1$. Increasing the value of ϵ eventually caused simulations to not converge within the time frame of the simulations. On Page 9 several plots are shown for different ϵ values. Even for ϵ values as small as 0.01 the system is not converging to the equilibrium in a similar timespan as before.

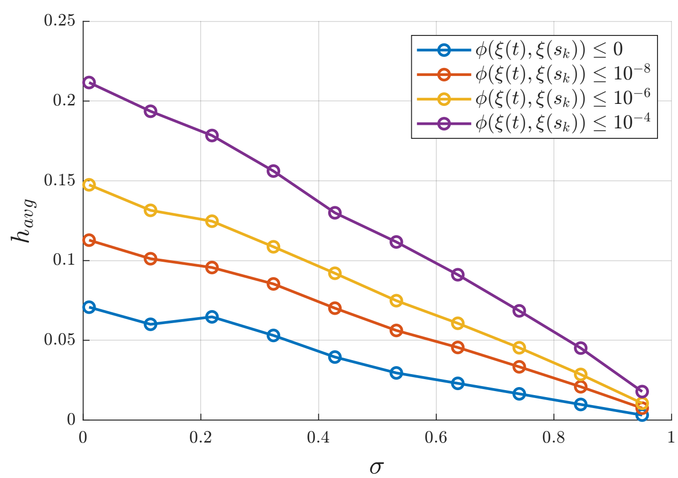
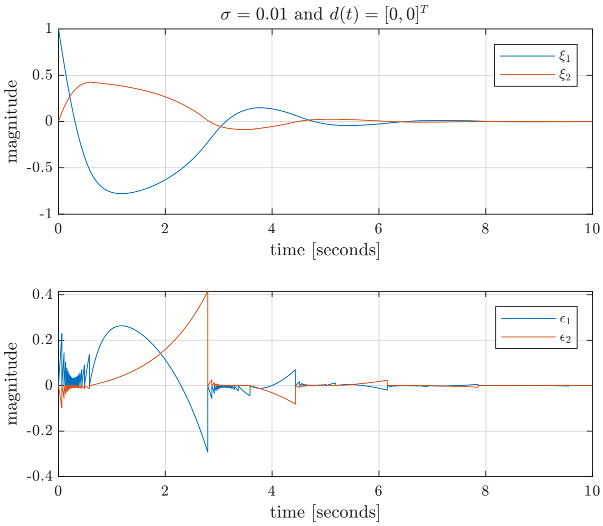
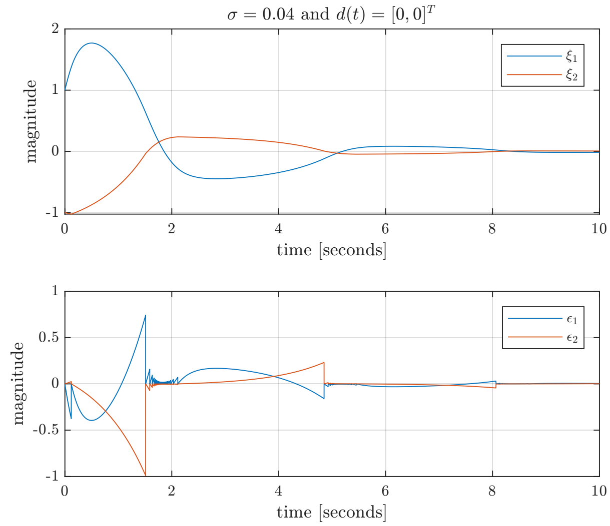
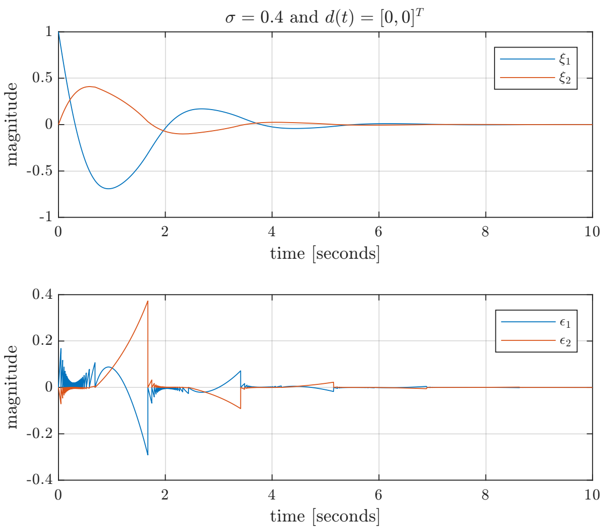
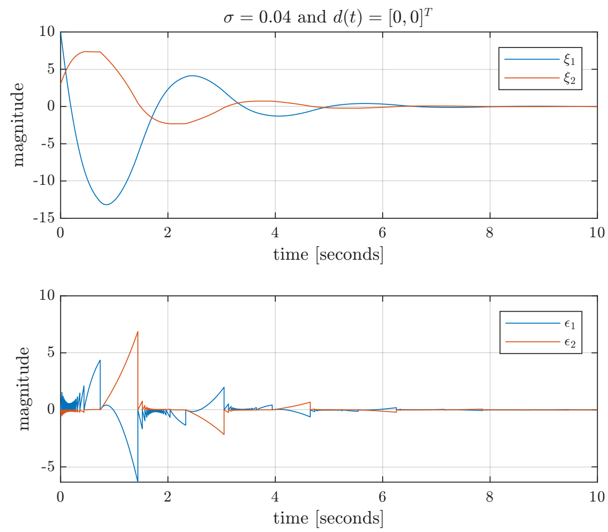


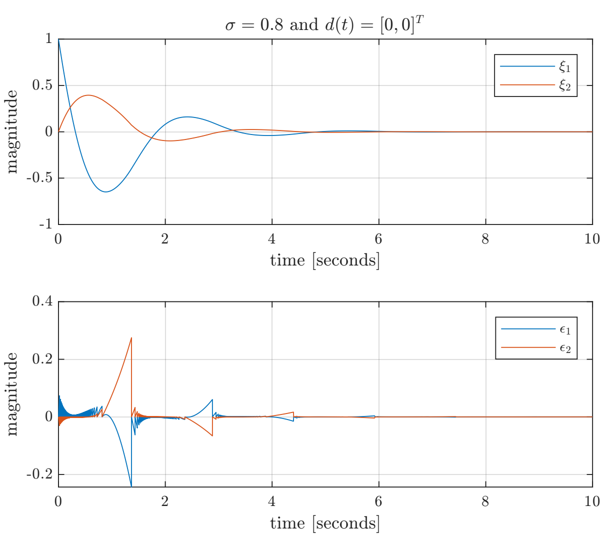
Fig. 12. Comparing different threshold values for the event triggering condition

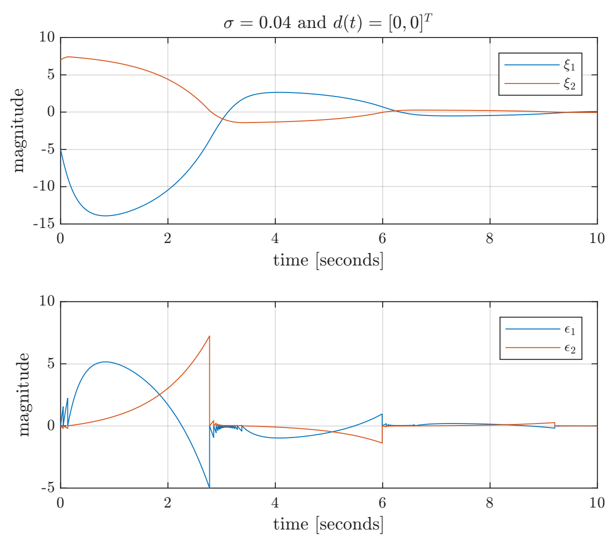

$\sigma = 0.01$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
magnitude
time [seconds]
ϵ_1
ϵ_2

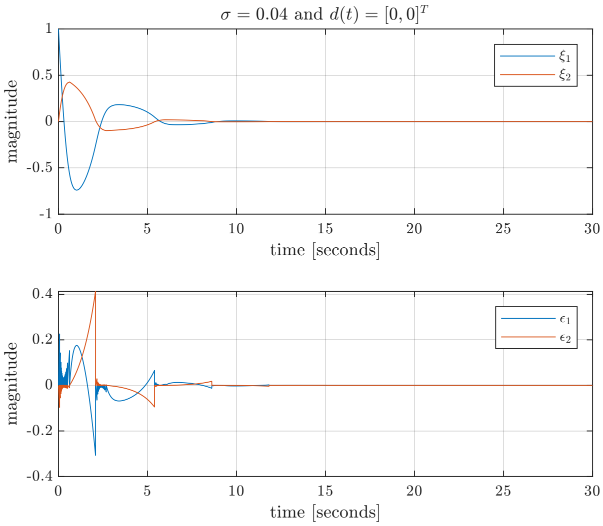

$\sigma = 0.04$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
magnitude
time [seconds]
ϵ_1
ϵ_2


$\sigma = 0.4$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
magnitude
time [seconds]
ϵ_1
ϵ_2

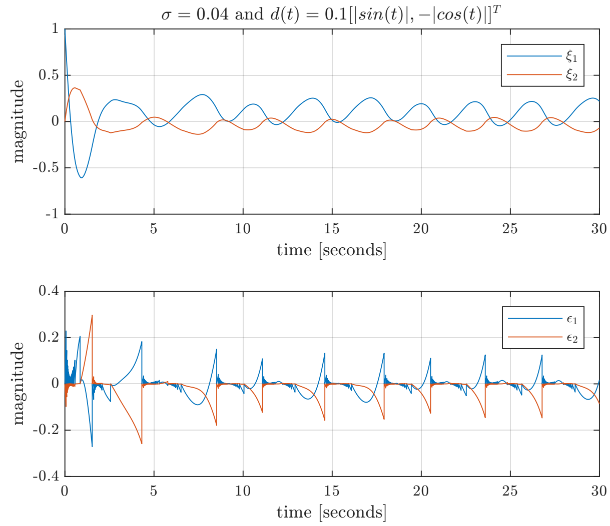

$\sigma = 0.04$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
magnitude
time [seconds]
ϵ_1
ϵ_2


$\sigma = 0.8$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
magnitude
time [seconds]
ϵ_1
ϵ_2

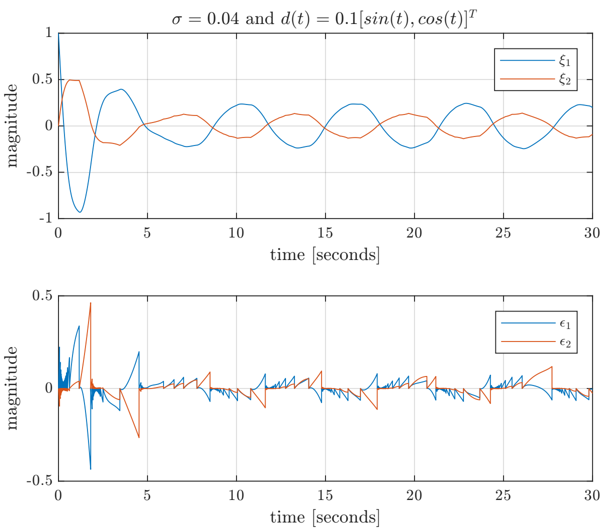

$\sigma = 0.04$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
magnitude
time [seconds]
ϵ_1
ϵ_2



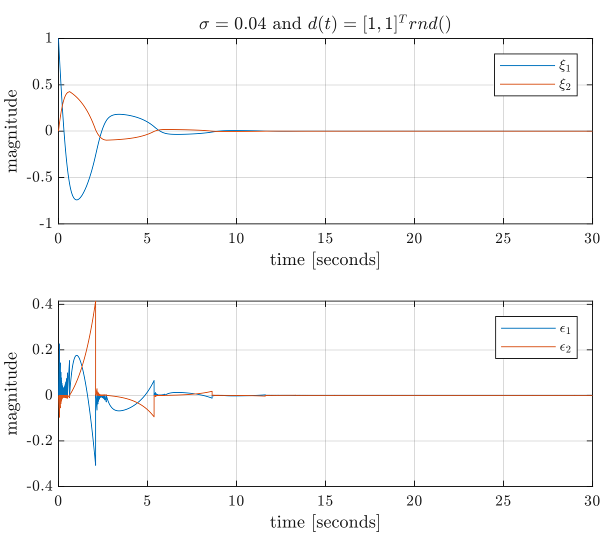
$\sigma = 0.04$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2



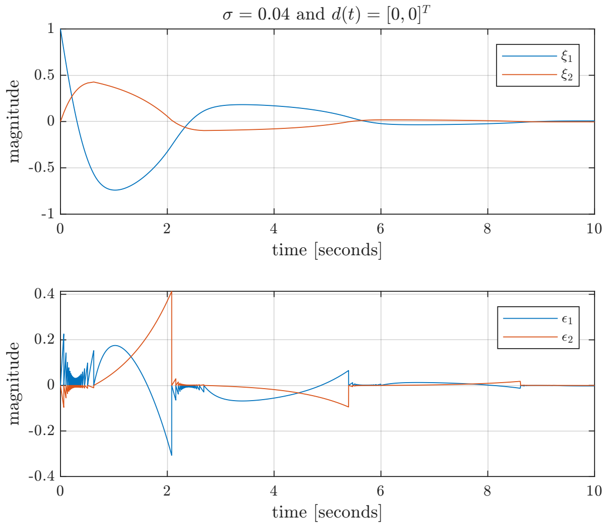
$\sigma = 0.04$ and $d(t) = 0.1[|sin(t)|, -|cos(t)|]^T$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2

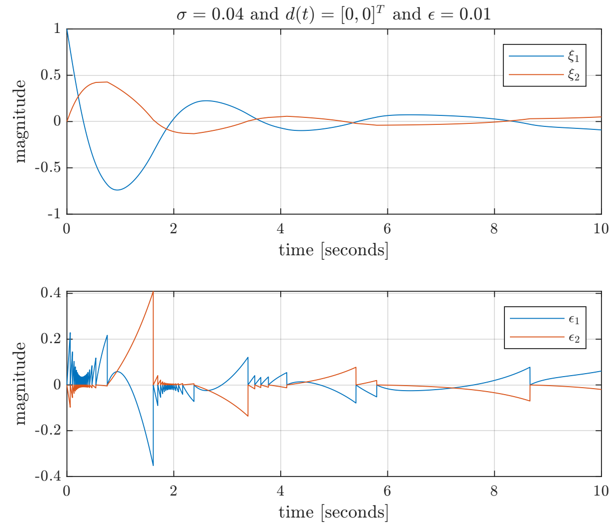


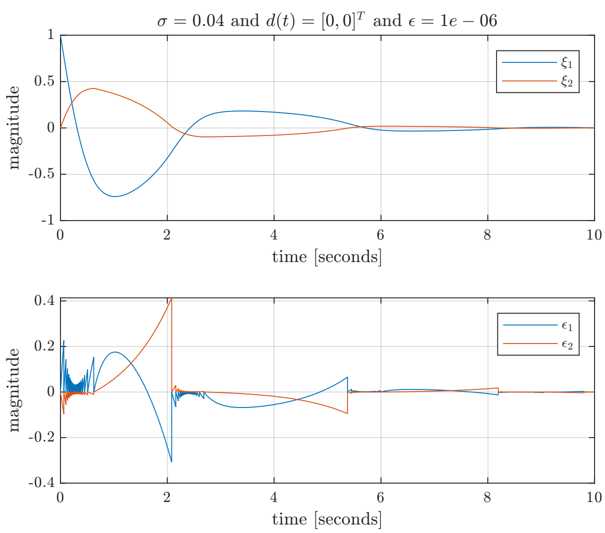
$\sigma = 0.04$ and $d(t) = 0.1[sin(t), cos(t)]^T$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2

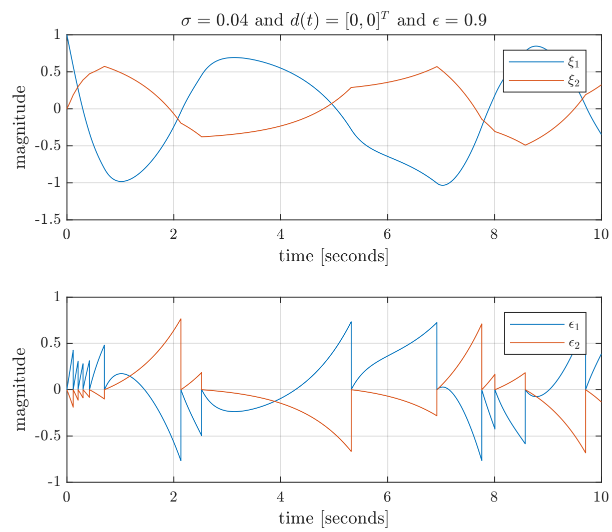


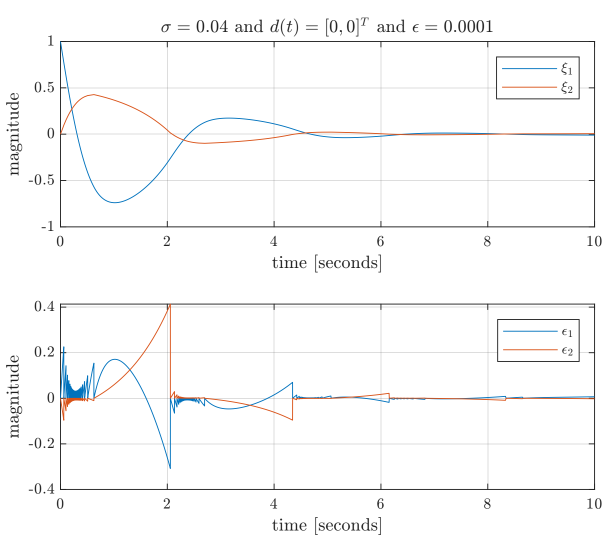
$\sigma = 0.04$ and $d(t) = [1, 1]^T rnd()$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2


$\sigma = 0.04$ and $d(t) = [0, 0]^T$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2


$\sigma = 0.04$ and $d(t) = [0, 0]^T$ and $\epsilon = 0.01$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2


$\sigma = 0.04$ and $d(t) = [0, 0]^T$ and $\epsilon = 1e - 06$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2


$\sigma = 0.04$ and $d(t) = [0, 0]^T$ and $\epsilon = 0.9$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2


$\sigma = 0.04$ and $d(t) = [0, 0]^T$ and $\epsilon = 0.0001$
magnitude
time [seconds]
ξ_1
ξ_2
ϵ_1
ϵ_2