Dutch Institute for Systems and Control (DISC) summer school 2022

Tutorial 1, v1.1
Security in Control Systems,
27 June 2022

André Teixeira and Michelle Chong

1 Preliminary Information

The assignment is divided into two parts:

1. Theory-related and conceptual questions;
2. Computer lab-related questions.

A brief description of the Matlab environment needed for the second part is given in Appendix A.

2 Tutorial Assignments

2.1 Conceptual questions - Security Risk Management

In the first part of the assignment, you will discuss concepts related to security and risk management

Task 1: In the lectures, we have discussed briefly the three basic security properties: confidentiality, integrity, and availability. Give a brief definition of each property and provide one example related to control systems where one or more of these properties are violated by an adversary.

Task 2: Discuss whether a natural fault can also violate one or more of the properties in the previous question. Motivate your answer, possibly with an example.

Task 3: What are the main elements of the concept "Risk"? Give a succinct explanation / interpretation for each element of the "Risk" tuple.

Task 4: What are the main key stages of the "Risk Management" process?

2.2 Theoretical questions - Attacks in Control Systems

In this part of the assignment, you will work with one of the attack scenarios discussed in the lectures.
Consider the discrete-time (DT) system under data injection attacks, which is described by in the following state-space form:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + B_a a[k], \\ y[k] &= Cx[k] + D_a a[k]. \end{aligned} \tag{1}$$

Unless otherwise stated in the questions, assume $u[k] = 0$.
The elements of A are known constants and the matrices A, B, C are given below as:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Task 5: Suppose that the adversary can attack sensor 1 and actuator 2, injecting false data into each of these channels. Model the attack vector $a[k]$ and the matrices B_a and D_a to reflect this scenario.

Task 6: Recall the definition of the output function, $y[k] = \Phi(x_0, a, k)$ and the definition of undetectable attacks. Prove or disprove the statement "There exists an undetectable attack for the system (1) in which actuator 2 is used." (*Hint:* note that the system has only one output, and that the adversary can directly corrupt this measurement signal while also affecting actuator 2. Maybe something can be canceled / hidden?)

Task 7: Suppose now that the adversary can only corrupt sensor 1 (i.e., all sensors can be corrupted). What is the policy that generates undetectable attack signals $a[k]$ (i.e., generates non-zeros attack signal that cannot be detected for all time $k \geq 0$)? How does the impact of the attack depend on the parameters a_{11}, a_{12}, a_{22}?

2.3 Theoretical questions - State estimation under sensor attacks

Consider again the DT system with multiple sensors under attack from Section 2.2 as follows

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k], \\ y_i[k] &= C_i x[k] + a_i[k], \qquad i \in \{1, \ldots, N\}, \end{aligned} \tag{2}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $a_{11} \neq 0$, $a_{12} \neq 0$ and $a_{22} \neq 0$. You will complete the following tasks by considering different matrices C_i and number of sensors N.

Task 8: Suppose system (2) has $N = 1$ output with $C_i = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Pick values of a_{11}, a_{12}, $a_{22} \in \mathbb{R}$ for which the system is observable.

Task 9: Suppose sensor 1 is under attack, i.e., $a_1[k] \neq 0$ for all $k \geq 0$. Can the states of system (2) still be estimated? If so, how? Hint: add redundant sensors.

Task 10: Consider the same system (2) with $N = 6$ sensors, modeled by

$$C_1 = C_2 = C_3 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad C_4 = C_5 = C_6 = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

What is the maximal number of attacked sensors M ?

2.4 Computer Exercises

In this computer exercise, you have received MATLAB & Simulink files that instantiate a nonlinear model of the quadruple tank system [1], and a linear LQG controller with integral action. The physical process, the controller, and the Matlab scripts are briefly described in Appendix A at the end of this document.

The purpose of this exercise is to try out and simulate an attack scenario in the simulated environment.

2.4.1 Performance under attacks

In the tasks that follow, you will simulate the closed-loop system for **one** attack scenario of your choice. The attack scenarios are controlled through the new script `set_attacks.m` but they are not fully implemented, so you will have to complete the implementation of the attack you choose. Read the list of scenarios carefully before moving forward with the implementation.
The possible scenarios are listed below and correspond to attack scenarios discussed in the lecture slides. Recall that you only need to choose and implement **one** scenario.

1. *Scenario 1: Undetectable attack on the actuators.* The attack corresponds to an attack on both actuators. The aim is to construct your attack in the form $a[k] = \alpha\nu_a^k g_a$, where $\alpha > 0$ is a free parameter.

   The variable $\nu_a \in \mathbb{C}$ is the zero of the plant under attack (A, B_a, C, D_a), which can be obtained by thought the MATLAB function `tzero`: $\nu = \texttt{tzero}(ss(A, B_a, C, D_a, T_s))$. (Note: if the function outputs a vector, it means there are more than 1 zero in the system. If this is the case, select the zero that is expected to have the largest impact.)

   Given the zero ν obtained as discussed above, then the vector $g_a \in \mathbb{C}^{n_u}$ can be obtained by taking the following steps:

   (a) construct the matrix
   $$P_\nu = \begin{bmatrix} \nu I_{n_x} - A & -B_a \\ C & D_a \end{bmatrix};$$

   (b) compute a basis matrix $M \in \mathbb{C}^{(n_x+n_u)\times n_\nu}$ for the null-space of P_ν, i.e., $M = \texttt{null}(P_\nu)$;

   (c) select a vector $v \in \mathbb{C}^{n_x+n_u}$ in the image space of M. For instance, v can be taken as any column of M;

   (d) partition $v \in \mathbb{C}^{n_x+n_u}$ as
   $$v = \begin{bmatrix} x_0^a \\ g \end{bmatrix};$$

   (e) set $g_a = \dfrac{g}{\|g\|_2}$.

   The attack is to be controlled in the Matlab script through the variables `flag_actuator_attack` and `mag_actuator_attack`, where `mag_actuator_attack` corresponds to the variable $\alpha > 0$.

Finally, you can use a for loop to construct the attack signal

$$a[k] = \alpha \nu_a^k g_a,$$

which you can save as a vector or as a time series variable.

To implement the attack in Simulink, you can use a "From Workspace" block to read the attack signal from the workspace into Simulink, and add this to the actuator channels.

2. *Scenario 2: Attack on actuator 1.* The attack corresponds to an attack on actuator 1. The aim is to construct your attack in the form $a[k] = \alpha \sin(\omega_a k T_s)$, where $\alpha > 0$ and $\omega_a > 0$ are free parameters, $T_s = 2\,s$ is the sampling period.

   To determine the frequency ω_a, you are required to look at the ratio between the largest singular value from $a[k]$ to $y_p[k]$ and the smallest singular value from $a[k]$ to $y_r[k]$. Choose ω_a that gives you the highest impact and lowest detectability.
   Hint: use the function `sigma(sys)` *to compute the singular values of the systems* Σ_p *and* Σ_r*. Check the lecture on security metrics for how* ω_a *can be related to these singular values.*

   The magnitude of the attack, determined by the parameter α, is to be selected as large as possible, but so that the instantaneous norm of the residual, $\|y_r[k]\|_2$, does not cross the detection threshold.

Task 9: Choose an attack scenario, implement it, and simulate it in the Simulink environment. Discuss the impact of the attack scenario, as well as whether the attack is detected or not by the residual vectors, as you change the "magnitude parameter" of each attack (the scaling α of the attack). If you have selected Scenario 2, you may also modify the frequency parameter.
In both scenarios, you can also simulate the attack in the linearized system by using `lsim` with the attack signal as input.

Task 10 : Consider the case where only sensor y_1 is under attack. Redesign the system by adding redundant sensors such that the water levels of the QTP can be estimated. Implement any of the secure state estimation algorithms mentioned in the lecture.

References

[1] K. Johansson, "The quadruple-tank process: a multivariable laboratory process with an adjustable zero," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 456–465, May 2000.

A Computer Lab Environment

In the following sections, we describe the nonlinear quadruple tank process in more detail.

A.1 Quadruple Tank process

Our testbed consists of a Quadruple-Tank Process (QTP) [1], as shown in Figure 1.

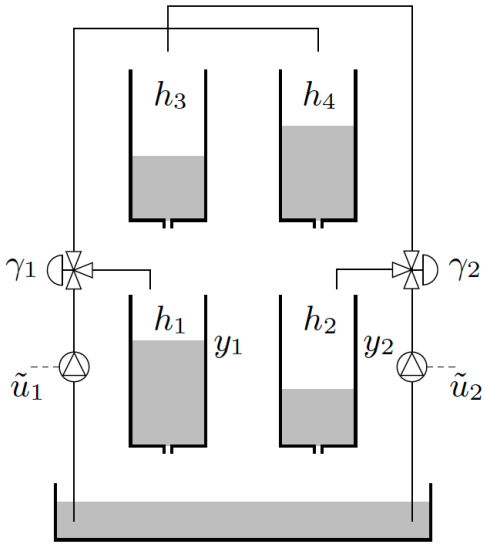

Figure 1: Schematic diagram of the quadruple-tank process.

The nonlinear plant model can be found in [1] and it is reproduced below:

$$\begin{aligned}
\dot{h}_1(t) &= -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{a_3}{A_1}\sqrt{2gh_3(t)} + \frac{\gamma_1 k_1}{A_1}u_1(t), \\
\dot{h}_2(t) &= -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{a_4}{A_2}\sqrt{2gh_4(t)} + \frac{\gamma_2 k_2}{A_2}u_2(t), \\
\dot{h}_3(t) &= -\frac{a_3}{A_3}\sqrt{2gh_3(t)} + \frac{(1-\gamma_2)k_2}{A_3}u_2(t), \\
\dot{h}_4(t) &= -\frac{a_4}{A_4}\sqrt{2gh_4(t)} + \frac{(1-\gamma_1)k_1}{A_4}u_1(t),
\end{aligned} \tag{3}$$

where $h_i(t) \in [0,\ 25]$ are the heights of water in each tank (here and further on $i = 1, 2, 3, 4$), A_i is the cross-section area, a_i is the cross-section area of the outlet hole, k_i are the pump constants, γ_i are the flow ratios determined by the valves, and g is the gravity acceleration. The water levels at the lower tanks ($h_1(t)$ and $h_2(t)$) are measured by two sensors, $y_1(t) = h_1(t)$ and $y_2(t) = h_2(t)$.
Defining the state of the system $x(t)$ as the collection of water levels,

$$x(t) = \begin{bmatrix} h_1(t) \\ h_2(t) \\ h_3(t) \\ h_4(t) \end{bmatrix},$$

the process dynamics can be described as

$$\begin{aligned} \dot{x}(t) &= f\left(x(t)\right) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}, \quad f\left(x(t)\right) = \begin{bmatrix} -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{a_3}{A_1}\sqrt{2gh_3(t)} \\ -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{a_4}{A_2}\sqrt{2gh_4(t)} \\ -\frac{a_3}{A_3}\sqrt{2gh_3(t)} \\ -\frac{a_4}{A_4}\sqrt{2gh_4(t)} \end{bmatrix} \tag{4}$$

The QTP is controlled using a centralized LQG controller with integral action, which ensures an accurate tracking of constant reference signals. To design the controller, the nonlinear plant model is linearized for a given operating point. A Kalman-filter-based anomaly detector is also used to generate residuals.

A.2 Simulink implementation

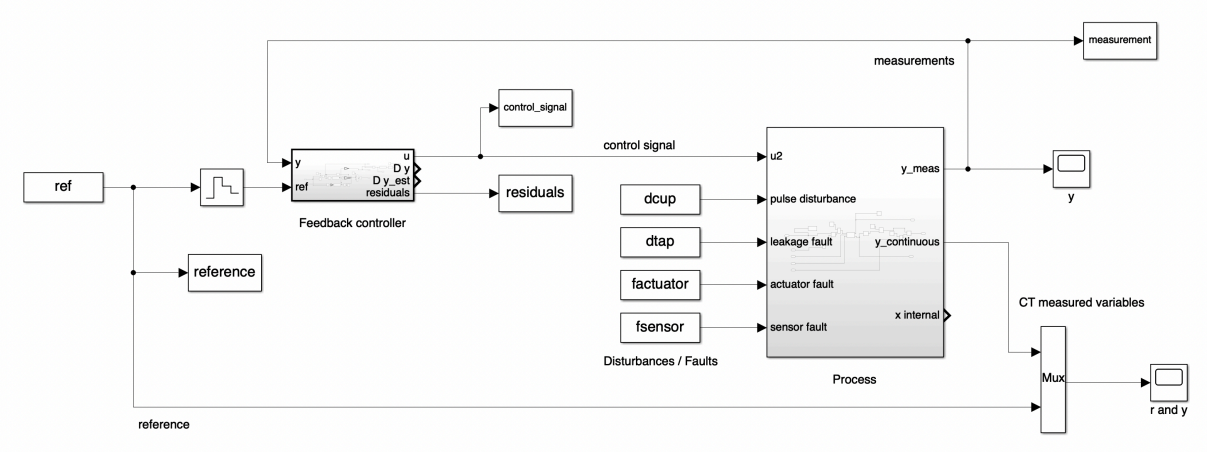

Figure 2: Simulink model / Block diagram of the closed-loop system.

A.3 Matlab Scripts to control the simulations

The main Matlab script that you will use is `run_experiment.m`. This script loads the necessary parameters of the process, including setpoints, after which the process is linearized and discretized with a sampling time of $T_s = 2\,s$. A linear feedback controller with integral action is also designed. All these initialization steps are performed within the script `initialize_closed_loop_system.m`.

After initializing the system and controller, you can extract the matrices of the DT system, based on which one can analyze the system properties. This analysis is not required for the present assignment.

Another part of the script defines variables that control the scenario being simulated. As documented in the script, as comments, the variables define whether or not a reference changes, or the presence and magnitude of disturbance and faults.

The script `set_attacks.m` defines flags that control the attack scenario. Additionally, variables encoding the attack policies and attack signals are also defined and constructed in this script. In the assignment, **you will need to change some of these variables**, to implement your chosen attack scenario.

The variables are then used internally by the script generate_signals.m to generate the corresponding signals.

The Simulink environment is then ready to be called an run through the command `simOut = sim(' ')`, after which the relevant measured signals are saved in the workspace and plotted.

In the assignment, looking into the Simulink model is not needed, it should suffice for you to modify the variables in `run_experiment.m` and run this script to obtain the data. You

can add plotting and simple operations at the end to obtain the relevant plots for your answers.