Academic year 2025–2026

Echahid Cheikh Larbi Tebessi University- Tebessa
Faculty of Science and Technology
Department of Electrical Engineering

Lecture Notes, Solved Problems, and MATLAB Scripts

Control of
Linear Systems

PID Control – State-Space Modeling – Observers – Kalman Filter

Prepared by Dr. Djamel OUNNAS

Summary

These lecture notes present essential methods for the analysis and control of linear systems, including frequency-domain design of PID controllers, state-space modeling, observers, and the Kalman filter. They combine theoretical foundations, solved problems, and MATLAB implementations to provide a comprehensive learning resource for students and practitioners in control engineering. They also serve as a consolidation of previously acquired knowledge, with a focus on mastering the representation of dynamic systems in state space, understanding their fundamental properties, and acquiring key methods for analysis and control synthesis. This course is intended primarily for third-year undergraduate students in Automation and Electrical Engineering. It is also suitable for learners who wish to acquire the fundamental concepts of **linear systems control** in general.

Contents

Introduction

These lecture notes provide a structured introduction to the analysis and design of linear control systems. The focus is on two complementary perspectives:

- **Frequency-domain methods**, using transfer functions to study stability and design classical PID controllers.
- **State-space methods**, offering a modern framework for system modeling, feedback control, and state estimation.

The material is intended for undergraduate and graduate students in automatic control, electrical engineering, and related fields. It combines theoretical concepts with practical techniques, aiming to build both intuition and technical competence.

The notes are organized into two parts:

- **Part I** (two chapters): stability analysis and PID controller design in the Laplace domain.
- **Part II** (four chapters): state-space modeling, system response, feedback control, observers, and Kalman filtering.

Throughout the course, emphasis is placed on bridging classical and modern approaches, highlighting how transfer function methods and state-space techniques complement one another. Worked examples and MATLAB scripts are provided to support learning and practical application.

Overall, these notes serve as both a reference and a companion for students, guiding them from fundamental principles to advanced control strategies. They are intended to support self-study, foster deeper understanding, and prepare students for advanced courses in modern control theory and applications.

Chapter 1

A Review of Stability Concepts in Linear Systems

1 Introduction

Stability is the primary performance requirement for any controlled system. An unstable system is characterized either by output oscillations with increasing amplitude, or by an unbounded positive or negative growth of the output signal.

In fact, ensuring the stability of the controlled system is the first essential objective for the proper functioning of any control system. Other objectives, such as accuracy and speed of response, cannot be achieved if the system itself is unstable.

This chapter introduces methods for assessing the stability of continuous linear systems, based either on the analytical expression of their transfer function or on graphical representations of their frequency response (such as Bode or Nyquist plots).

Definition 1 *A linear system is said to be stable if its output response remains bounded for all bounded inputs. Otherwise, it is considered unstable.*

Example 1 *Consider the following two first-order systems whose open-loop transfer functions are given by equations 1.1 and 1.2.*

$$G_1(s) = \frac{S_1(s)}{E(s)} = \frac{5}{10s+1} \tag{1.1}$$

$$G_2(s) = \frac{S_2(s)}{E(s)} = \frac{5}{2s-1} \tag{1.2}$$

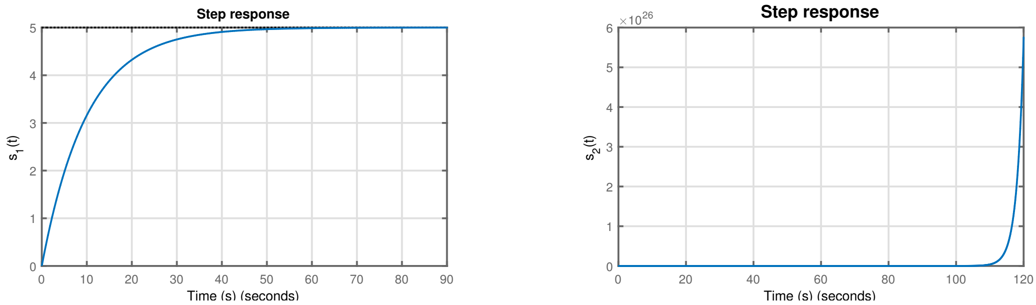
The time-domain responses of these systems to a unit step input ($E(s) = 1/s$) are provided in equations 1.3 and 1.4.

$$s_1(t) = 5(1 - e^{-\frac{1}{10}t}) \tag{1.3}$$

$$s_2(t) = 5(1 - e^{\frac{1}{2}t}) \tag{1.4}$$

The step response plots are presented in Figures 1.1(a) and 1.1(b). From these results, it can be observed that the first system is stable, as its output remains bounded over time, whereas the second system is unstable due to its output diverging without bound.

To reproduce these results, the Matlab script [Script 1] can be utilized.



(a) Unit step response of a stable system (b) Unit step response of an unstable system

Figure 1.1: Time-domain responses of the two systems under unit step excitation

```
%% Script: Example 1
clc, close all, clear all;
%% Declaration of system 1
Num1=5;                    % Numerator
Den1=[10 1];               % Denominator
G1=tf(Num1,Den1);          % Create transfer function G1
%% Declaration of system 2
Num2=5;                    % Numerator
Den2=[2 -1];               % Denominator
G2=tf(Num2,Den2);          % Create transfer function G2
%% Step response: System 1
figure(1)                  % Open figure window number 1
step(G1)                   % Plot the step response of system 1
grid on                    % Enable the grid on the plot
xlabel('Time (s)')         % Label for the x-axis
ylabel('s_1(t)')           % Label for the y-axis
title('Step response')     % Title of the plot
%% Step response: System 2
figure(2)                  % Open figure window number 2
step(G2)                   % Plot the step response of system 2
grid on                    % Enable the grid on the plot
xlabel('Time (s)')         % Label for the x-axis
ylabel('s_2(t)')           % Label for the y-axis
title('Step response')     % Title of the plot
```

In general, to assess the stability of a continuous-time linear system, several criteria can be used, which can be grouped into three main categories: mathematical criteria based on the system's poles, algebraic criteria relying on the analysis of the system's characteristic equation, and graphical criteria using various plots (such as Bode, Nyquist, and Black diagrams, etc.). In the following sections, we present the essential aspects of these criteria.

2 Mathematical Criterion

We consider the closed-loop system represented by the block diagram in Figure 1.2.

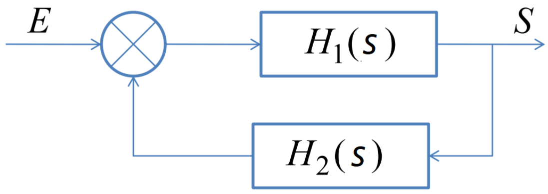


Figure 1.2: Functional block diagram of a closed-loop system

The closed-loop transfer function (CLTF) of the system is given by equation 1.5.

$$H(s) = \frac{S(s)}{E(s)} = \frac{H_1(s)}{1 + H_1(s)H_2(s)} \tag{1.5}$$

The closed-loop transfer function (CLTF) of the system can be rewritten in polynomial form (with respect to the Laplace variable s) as shown in equation 1.6.

$$H(s) = \frac{N(s)}{D(s)} = \frac{b_n s^n + b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} \tag{1.6}$$

The equation $D(s) = a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0 = 0$ is known as the *characteristic equation* of the system. The closed-loop transfer function of the system can also be rewritten in the pole-zero form given in equation 1.7:

$$H(s) = \frac{(s - z_1)(s - z_2)\ldots(s - z_n)}{(s - s_1)(s - s_2)\ldots(s - s_n)} \tag{1.7}$$

Where $z_1, \ldots, z_n$ are the zeros of the system (i.e., the roots of the numerator $N(s)$), and $s_1, \ldots, s_n$ are the poles of the system (i.e., the roots of the denominator $D(s)$).

Therefore, the stability condition according to the mathematical criterion can be stated as follows:

A closed-loop system is stable if and only if all the poles of its closed-loop transfer function have strictly negative real parts.

Example 2 *Let us consider again the transfer functions (TF) given in 1.1 and 1.2. The pole of the first TF is negative ($s_1 = -1/10$, stable system), whereas the pole of the second TF is positive ($s_1 = 1/2$, unstable system).*

Example 3 *Consider an open-loop transfer function $G(s)$ defined by:*

$$G(s) = \frac{8}{(s+1)(s+4)} \tag{1.8}$$

The closed-loop transfer function of the system 1.8 is given by:

$$G(s) = \frac{8}{s^2 + 5s + 12} \tag{1.9}$$

The poles of the system are: $s_1 = -2.5 + 2.39j$, $s_2 = -2.5 - 2.39j$. This system is stable because both poles have negative real parts.

The Matlab script [Script 2] can be used to check the stability of a system by analyzing its closed-loop transfer function.

```
%% Scripts: Example 2
clc, close all, clear all;              % Clear the command ...
    window, close all figures, and remove all variables from ...
    workspace

%% System declaration
NUM=8;                                  % Numerator of the ...
    transfer function
DEN=conv([1 1],[1 4]);                  % Denominator obtained by ...
    convolving (s+1) and (s+4)
FTBO=tf(NUM,DEN);                       % Open-loop transfer function
FTBF=feedback(FTBO,1);                  % Closed-loop transfer ...
    function with unity feedback
%% Extraction of numerator and denominator of FTBF
[num,den]=tfdata(FTBF,'v');             % Get numerator and ...
    denominator vectors of FTBF
%% Calculation of poles
p=roots(den);                           % Compute the poles by ...
    finding roots of the denominator
%% Checking the sign of the poles
if (real(p(1))<0) & (real(p(2))<0)      % If real parts of both ...
    poles are negative
    disp('System is stable')            % Display message: system ...
        is stable
else
    disp('System is unstable')          % Display message: system ...
        is unstable
end
%% Step response of the system
step(FTBF)                              % Plot the step response ...
    of the closed-loop system
grid on                                 % Enable the grid on the plot
xlabel('Time (s)')                      % Label for the x-axis
ylabel('s(t)')                          % Label for the y-axis
title('Step response')                  % Title of the plot
```

The step response of the system 1.9 is illustrated in Figure 1.3.

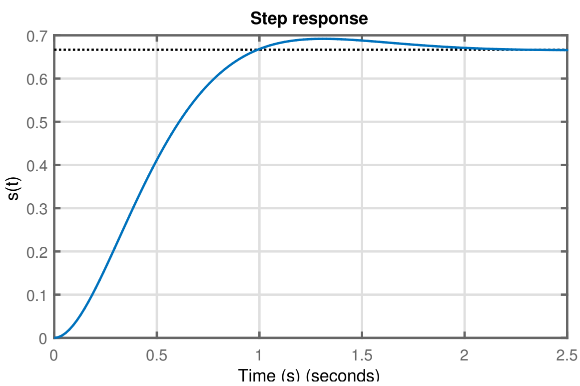


Figure 1.3: Step response of the system 1.9

3 Algebraic Criterion

The mathematical criterion is simple and easy to apply to assess the stability of a continuous linear system. However, this criterion cannot be used to judge the stability of systems with higher-order transfer functions, since it becomes impossible to find the roots of the characteristic equation analytically. To overcome this limitation, another well-known method called the *Routh Criterion* is used.

The Routh criterion is an analytical technique that determines whether the characteristic equation has roots with positive real parts. This method can only be applied when the characteristic equation is a polynomial in s. Therefore, the Routh criterion cannot be directly applied to systems containing delays (terms like $e^{-\theta s}$).

The Routh criterion is based on the characteristic equation 1.10:

$$D(s) = a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + a_{n-3} s^{n-3} + \ldots + a_1 s + a_0 = 0 \tag{1.10}$$

We arbitrarily assume that $a_n > 0$. If $a_n < 0$, the equation 1.10 is multiplied by -1 to ensure this condition.

- *First test*: A necessary (but not sufficient) condition for stability is that all coefficients $(a_n, a_{n-1}, \ldots, a_0)$ of the characteristic equation must be positive. If any coefficient is negative or zero, then at least one root has a positive real part, and the system is unstable.

- *Second test*: If all coefficients are positive, the Routh table can be constructed as follows:

a_n	a_{n-2}	a_{n-4}	a_{n-6}	$\ldots$
a_{n-1}	a_{n-3}	a_{n-5}	a_{n-7}	$\ldots$
b_1	b_2	b_3	b_4	$\ldots$
c_1	c_2	c_3	c_4	$\ldots$

Where:

$$b_1 = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}, \quad b_2 = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}, \ldots$$

$$c_1 = \frac{b_1 a_{n-3} - a_{n-1} b_2}{b_1}, \quad c_2 = \frac{b_1 a_{n-5} - a_{n-1} b_3}{b_1}, \ldots$$

The stability condition according to the Routh criterion is stated as follows:

A necessary and sufficient condition for all roots of the characteristic equation to have negative real parts is that all elements of the first column of the Routh table must be positive.

Remark: The number of sign changes in the first column of the Routh table equals the number of poles with positive real parts (unstable poles).

Example 4 *Study the stability of the system whose closed-loop transfer function $H(s)$ is given by:*

$$H(s) = \frac{10}{s^4 + 5s^3 + s^2 + 1} \tag{1.11}$$

Solution: The characteristic polynomial is $D(s) = s^4 + 5s^3 + s^2 + 0s + 1$. This system is unstable because the coefficient of s is zero (fails the first test).

Example 5 *Consider a system with transfer function $G(s)$. Determine the conditions on the value of K so that the system remains stable:*

$$H(s) = \frac{15}{10s^3 + 17s^2 + 8s + 1 + K} \tag{1.12}$$

Solution: The characteristic polynomial is $D(s) = 10s^3 + 17s^2 + 8s + 1 + K$. All coefficients are positive and non-zero, satisfying the first test. We construct the Routh table:

10	8
17	$K+1$
$\frac{17\cdot 8 - 10\cdot(K+1)}{17}$	0
$K+1$	

The system is stable if:

$$\frac{136 - 10(K+1)}{17} > 0 \quad \Rightarrow \quad 0 < K < 12.6$$

4 Graphical Criterion

Consider a system represented by the block diagram in Figure 1.2. The closed-loop transfer function (CLTF) is:

$$H(p) = \frac{S(p)}{E(p)} = \frac{H_1(p)}{1 + H_1(p)H_2(p)} = \frac{H_1(p)}{1 + G(p)} \tag{1.13}$$

Where $G(p) = H_1(p)H_2(p)$ is the open-loop transfer function (OLTF). The characteristic equation becomes:

$$1 + G(p) = 0 \quad \Rightarrow \quad G(p) = -1 \tag{1.14}$$

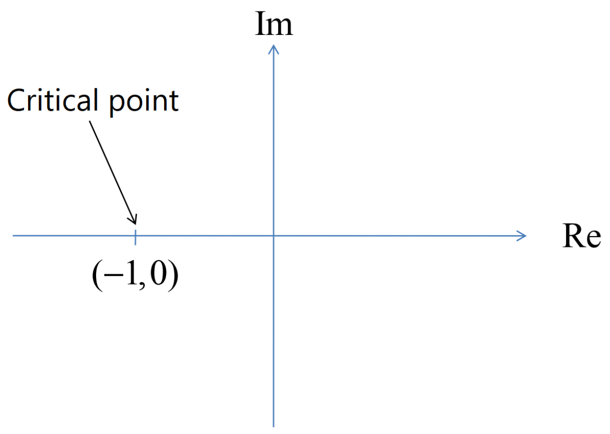
The location of the OLTF with respect to the critical point -1 tells us about the stability of the system. In the complex plane, the critical point is $(-1, 0)$ (Figure 1.4(a)); in the Black (gain-phase) diagram, it corresponds to $(-180°, 0)$ (Figure 1.4(b)).

Graphical criteria judge closed-loop stability based on plots (Nyquist, Bode, Black) of the OLTF.

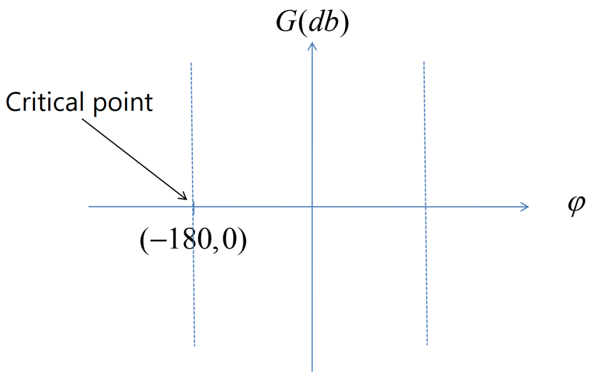
Revers Criterion: The Revers criterion is based on the Nyquist plot of the OLTF, which must not have poles with positive real parts. Under these conditions, the criterion can be stated as:

- **In the Nyquist plot:** *A linear feedback system is stable if, when drawing the Nyquist plot of the OLTF, the critical point is kept to the left. Otherwise, the system is unstable.*
- **In the Black plot:** *A linear feedback system is stable if, when drawing the Black plot of the OLTF, the critical point is kept to the right. Otherwise, the system is unstable.*
- **In the Bode plot:** The Revers criterion reduces to checking at the crossover frequency ω_{c0} (frequency where gain is 0 dB) that:

$$|G(j\omega_{c0})| = 1 \quad \text{and} \quad \angle G(j\omega_{c0}) > -180° \tag{1.15}$$



(a)



(b)

Figure 1.4: Critical point: (a) in the Nyquist plot, (b) in the Black plot

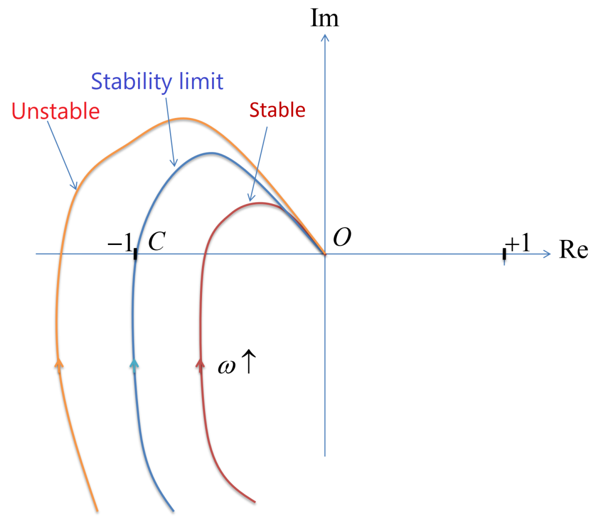


Figure 1.5: Revers criterion in the Nyquist plot

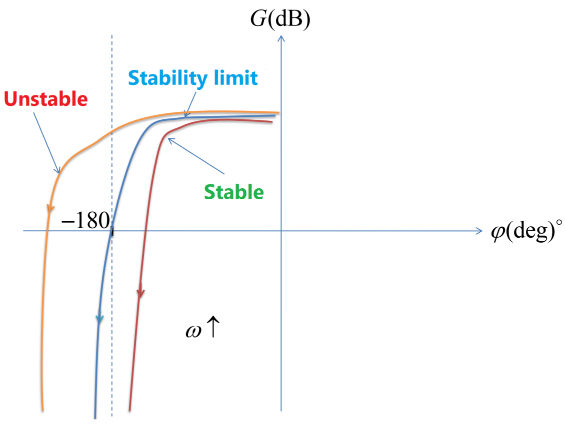


Figure 1.6: Revers criterion in the Black plot

5 Stability Margins

The criteria presented above are absolute stability criteria; however, they do not generally allow for optimal system tuning. Referring again to example 5, we found that for the system to remain stable, the gain K must be selected between 0 and 12.6. This raises the question: what is the most appropriate value of K to choose?

To address this, we introduce the concept of stability margins, which represent a safety distance to be maintained between the critical point and the open-loop transfer function locus. In this context, we define the *gain margin* and the *phase margin*. Common recommended values are a gain margin of approximately 6 dB and a phase margin of 45°.

In what follows, we define these margins for a system whose open-loop transfer function is denoted by $G(s)$, characterized by its magnitude $G(\omega)$ and phase $\varphi(\omega)$, and assumed to be stable.

5.1 Gain Margin

Consider the Nyquist diagram of the system, shown in Figure 1.7(a). The gain margin corresponds to the distance OA between the origin O and the point A, defined as the point where the curve intersects the negative real axis.

This distance OA is defined by the following coordinates:

$$G(\omega_\pi) = OA \quad \text{and} \quad \varphi(\omega_\pi) = -\pi \tag{1.16}$$

The gain margin, usually expressed in decibels, is given by:

$$M_G = -20 \log G(\omega_\pi) \quad \text{with} \quad \varphi(\omega_\pi) = -\pi \tag{1.17}$$

Example 6 *Calculate the gain margin of the following system:*

$$G(s) = \frac{2 \times 10^6}{(s+100)^3} \tag{1.18}$$

Solution: The magnitude and phase expressions can be written as:

$$G(s) = \frac{2 \times 10^6}{(s+100)^3} \Rightarrow \begin{cases} G(\omega) = \dfrac{2 \times 10^6}{(\sqrt{\omega^2 + 100^2})^3} \\ \varphi(\omega) = -3 \arctan\left(\frac{\omega}{100}\right) \end{cases}$$

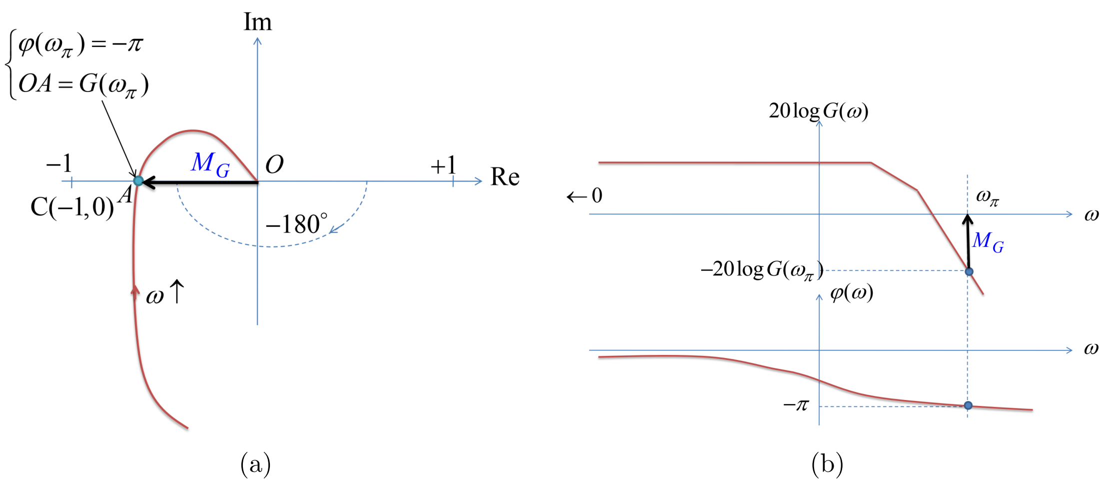


Figure 1.7: Gain margin: (a) in the Nyquist plane, (b) in the Bode diagram

We first determine the frequency ω_π such that $\varphi(\omega_\pi) = -\pi$:

$$-3\arctan\left(\frac{\omega_\pi}{100}\right) = -\pi \Rightarrow \arctan\left(\frac{\omega_\pi}{100}\right) = \frac{\pi}{3} \Rightarrow \omega_\pi = 100\sqrt{3}$$

Then, the gain margin is computed as follows:

$$M_G = -20\log\left(\frac{2\times 10^6}{\left(\sqrt{\omega_\pi^2+100^2}\right)^3}\right) \Rightarrow M_G = 12\,\text{dB}$$

5.2 Phase Margin

The concept of phase margin quantifies the angular distance between the critical point and the Nyquist plot of the system. Consider the Nyquist plot shown in Figure 1.8(a) for a system assumed to be stable. The point B (located at the intersection of the Nyquist plot and the unit circle centered at O) corresponds to the gain crossover frequency at 0 dB (by definition, $G(\omega_{c0}) = 1$). The phase margin is then the angular separation between point B and the critical point C. Consequently, the phase margin is defined as follows:

$$M_\varphi = \pi + \varphi(\omega_{c0}) \quad \text{with} \quad G(\omega_{c0}) = 1 \tag{1.19}$$

Example 7 *Compute the phase margin of the following system:*

$$G(s) = \frac{2\times 10^6}{(s+100)^3} \tag{1.20}$$

Solution: *The magnitude and phase expressions can be derived from the open-loop transfer function as follows:*

$$G(j\omega) = \frac{2\times 10^6}{(j\omega+100)^3} \Rightarrow \begin{cases} G(\omega) = \frac{2\times 10^6}{(\sqrt{\omega^2+100^2})^3} \\ \varphi(\omega) = -3\arctan\left(\frac{\omega}{100}\right)\end{cases}$$

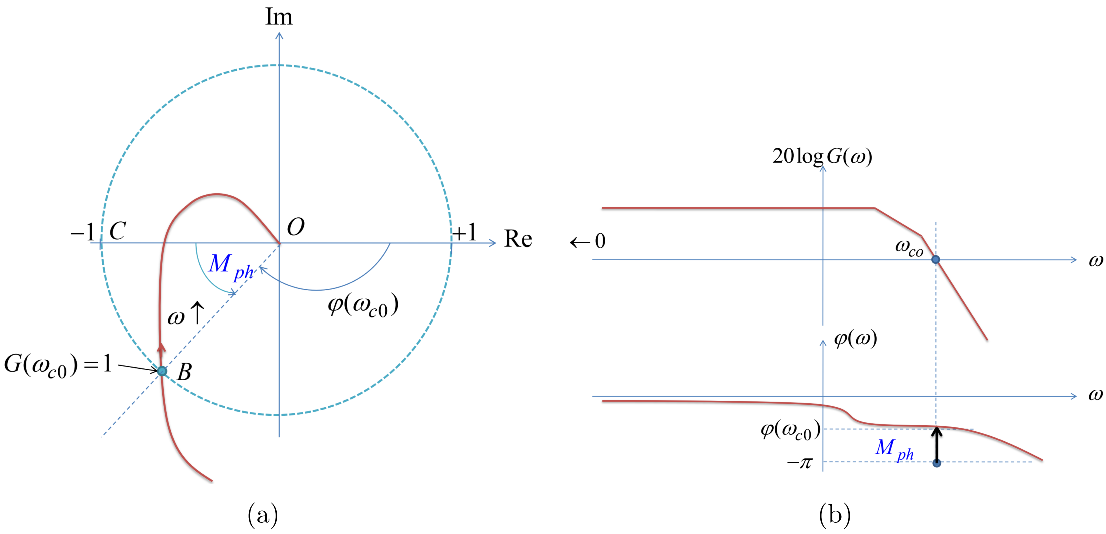


Figure 1.8: Phase margin: (a) in the Nyquist plane, (b) in the Bode plot

First, we calculate the gain crossover frequency ω_{c0} and then deduce the phase margin M_φ:

$$G(\omega_{c0}) = 1 \Rightarrow \frac{2\times 10^6}{(\sqrt{\omega_{c0}^2+100^2})^3} = 1 \Rightarrow \omega_{c0} = \sqrt{(\sqrt[3]{2\times 10^6})^2 - 100^2} = 76.6\,\mathrm{rad/s}$$

Thus, we have:

$$M_\varphi = \pi + \varphi(\omega_{c0}) = \pi - 3\arctan\left(\frac{76.6}{100}\right)$$

$$\Rightarrow M_\varphi = 67.6^\circ$$

The Matlab script [Script 3] can be used to compute the gain margin, phase margin, and to display them on the Bode diagram.

```
%% Script: Examples 5 and 6
clc, close all, clear all;%
%% System declaration
NUM=2e6;                      % Numerator
DEN1=conv([1 100],[1 100]); %
DEN=conv(DEN1,[1 100]);     %
G=tf(NUM,DEN);              %
%% Calculation of gain margin MG and phase margin Mph
[MG,Mph,Wg,Wp]=margin(G);%
%% Bode diagram with margins
margin(G);%
```

The Bode plots of system 1.20, analyzed in Examples 6 and 7, are shown in Figure 1.9.

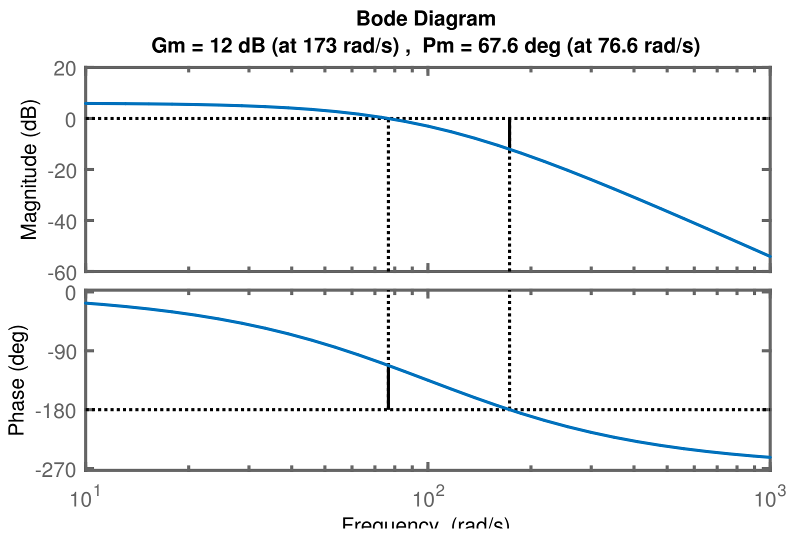


Figure 1.9: Bode plots of system 1.20

6 Exercises

6.1 Exercise 1

Consider an open-loop transfer function $G(s)$ given by:

$$G(s) = \frac{K}{s(s+2)(s+3)}$$

when the system is placed in a unity-feedback closed-loop configuration.

- *Study the stability of the system.*

6.2 Exercise 2

Consider an open-loop transfer function $G(s)$ given by:

$$G(s) = \frac{100K}{s^2+10s+11}$$

when the system is placed in a unity-feedback closed-loop configuration.

1- Study the stability of the system.

2- Compute the value of K that ensures a gain margin of $6\,\mathrm{dB}$.

6.3 Exercise 3

Consider an open-loop transfer function $G(s)$ given by:

$$G(s) = \frac{K}{s(s+10)^2}$$

Determine the value of K that gives a crossover frequency equal to $2\,\mathrm{rad/s}$ *and deduce the corresponding phase margin.*

6.4 Exercise 4

Consider an open-loop transfer function $G(s)$ given by:

$$G(s) = \frac{K}{(s+1)^3}$$

Compute the value of K that ensures a phase margin of $45°$.

7 Solutions

7.1 Solution to Exercise 1

The open-loop transfer function is:

$$G(s) = \frac{K}{s(s+2)(s+3)}$$

The closed-loop transfer function is:

$$H(s) = \frac{G(s)}{1+G(s)} = \frac{K}{s^3+5s^2+6s+K}$$

The characteristic equation is:

$$s^3+5s^2+6s+K=0$$

All coefficients are positive and non-zero, so we can build the Routh-Hurwitz table:

1	6
5	K
$\frac{30-K}{5}$	0
K	

The system is stable if:

$$\frac{30-K}{5} > 0 \quad \Rightarrow \quad 0 < K < 30.$$

7.2 Solution to Exercise 2

The open-loop transfer function is:

$$G(s) = \frac{2K}{(5s+1)^3}$$

1- Stability analysis:
The closed-loop transfer function is:

$$H(s) = \frac{2K}{125s^3 + 75s^2 + 15s + 2K}$$

Characteristic equation:

$$125s^3 + 75s^2 + 15s + 2K = 0$$

The Routh-Hurwitz table is:

125	15
75	$2K$
$\frac{1000-30K}{75}$	0
$2K$	

The system is stable if:

$$0 < K < \frac{1000}{30} \approx 33.33.$$

2- Compute K for a gain margin of $6\,\mathrm{dB}$:
The magnitude and phase of $G(j\omega)$ are:

$$|G(j\omega)| = \frac{2K}{(\sqrt{25\omega^2+1})^3}, \quad \varphi(\omega) = -3\arctan(5\omega)$$

At phase crossover ($\varphi(\omega_\pi) = -\pi$):

$$-3\arctan(5\omega_\pi) = -\pi \quad \Rightarrow \quad \arctan(5\omega_\pi) = \frac{\pi}{3}$$

$$5\omega_\pi = \tan\frac{\pi}{3} = \sqrt{3} \quad \Rightarrow \quad \omega_\pi = \frac{\sqrt{3}}{5}$$

Compute the gain margin:

$$GM = -20\log_{10}\left(\frac{2K}{(\sqrt{25\omega_\pi^2+1})^3}\right) = 6\,\mathrm{dB}$$

From this, solve for K.

7.3 Solution to Exercise 3

1- Find K for crossover frequency $\omega_{c0} = 2\,\text{rad/s}$:

$$|G(j\omega_{c0})| = 1 \quad \Rightarrow \quad \frac{K}{\omega_{c0}(\omega_{c0}^2 + 100)} = 1$$

$$K = \omega_{c0}(\omega_{c0}^2 + 100) = 2(4 + 100) = 208.$$

2- Compute phase margin:

$$\varphi(\omega_{c0}) = -90^\circ - 2\arctan\left(\frac{\omega_{c0}}{10}\right)$$

$$M_\varphi = 180^\circ + \varphi(\omega_{c0})$$

$$= 180^\circ - 90^\circ - 2 \cdot 11.31^\circ = 90^\circ - 22.62^\circ = 67.38^\circ.$$

7.4 Solution to Exercise 4

Find K for a phase margin of 45°:

Phase at crossover frequency ω_{c0}:

$$M_\varphi = 180^\circ + \varphi(\omega_{c0}) = 45^\circ \quad \Rightarrow \quad \varphi(\omega_{c0}) = -135^\circ$$

$$-3\arctan(\omega_{c0}) = -135^\circ \quad \Rightarrow \quad \arctan(\omega_{c0}) = 45^\circ$$

$$\omega_{c0} = \tan(45^\circ) = 1.$$

Magnitude condition:

$$|G(j\omega_{c0})| = 1 \quad \Rightarrow \quad \frac{K}{(\sqrt{\omega_{c0}^2 + 1})^3} = 1$$

$$\omega_{c0} = 1 \quad \Rightarrow \sqrt{2}^3 = 2\sqrt{2} \quad \Rightarrow K = 2\sqrt{2}.$$

Chapter 2

Design of PID Controllers

1 Introduction

The objective of controlling linear or nonlinear systems is to modify their performance so that they follow a desired reference and meet predetermined specifications. To achieve this, it is necessary to include in the feedback loop a control element, commonly called controller, regulator, or compensator, which is fundamentally based on the concept of feedback.

The controller computes the error value, corresponding to the difference between the system output and the reference input, and then attempts to minimize this error by increasing or decreasing the control action so that the output approaches the setpoint.

To better illustrate this concept, consider the diagram in Figure 2.1, which shows an example of furnace control implemented on a programmable logic controller (PLC). The furnace temperature is regulated by adjusting a gas valve. The operator sets the desired temperature as the setpoint. A temperature sensor measures the actual temperature and sends it to the controller. The feedback signal is compared to the setpoint, and an error value is computed. The controller then determines the appropriate valve position to reduce the error.

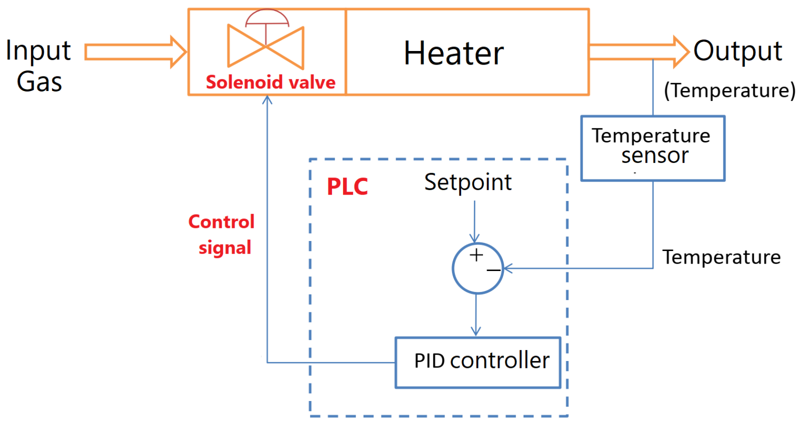


Figure 2.1: Furnace control

The example in Figure 2.1 can also be represented by the functional block diagram in Figure 2.2.

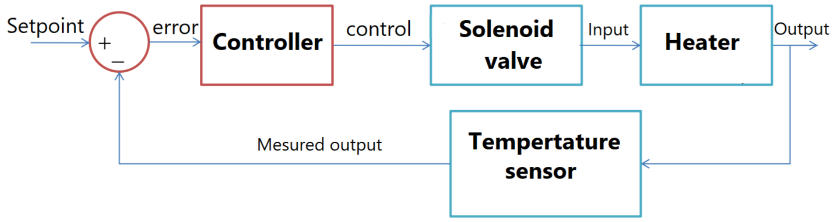


Figure 2.2: Functional diagram of a feedback control system

In automatic control, each element of the feedback loop is usually modeled by a transfer function, as shown in Figure 2.3. Here, $G(s)$, $A(s)$, $B(s)$, and $C(s)$ denote respectively the transfer functions of the process to be controlled, the actuator, the sensor, and the controller (compensator). The signals W, ε, U, E, and S represent respectively the reference input, the error, the control signal, the system input, and the system output.

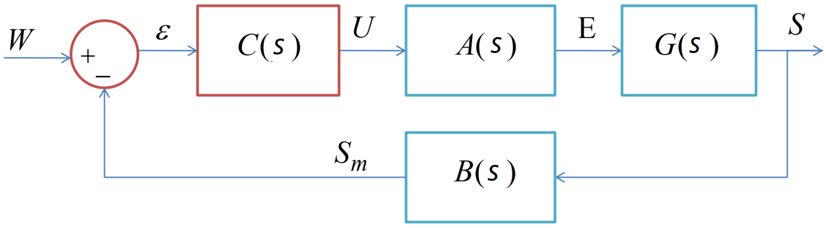


Figure 2.3: Block diagram of a feedback control system

In the control of linear systems, the Proportional-Integral-Derivative (PID) controller is widely used in industrial applications. The PID controller combines three corrective actions with their own parameters, which can be interpreted in time domain as follows: the proportional (P) action depends on the present error, the integral (I) action accounts for the accumulation of past errors, and the derivative (D) action predicts future errors based on the current rate of change. The weighted sum of these three actions is used to produce the control signal applied to the system.

2 Control system specifications

In general, a feedback control system must satisfy a set of specifications that impose, in closed loop, the following static and dynamic performance criteria:

- *Accuracy*, characterized by the steady-state error, defined as the permanent deviation between the measured output and the setpoint. Usually, the position error ε_p (error for a step input) is used to quantify accuracy:

$$\varepsilon_p = \lim_{t\to\infty} \varepsilon(t) = \lim_{s\to 0} s\,\varepsilon(s) = \lim_{s\to 0}\big(1 - FTBF(s)\big) \tag{2.1}$$

- *Speed of response*, often measured by the rise time, defined as the time required for the system output to rise from 10% to 90%, 5% to 95%, or 0% to 100% of its final value. A common approximate relation is:

$$t_m = \frac{3}{\omega_{nBF}} = \frac{3}{\omega_{c0}} \tag{2.2}$$

- *Stability margin*, generally quantified by the phase margin. The closed-loop damping ratio can be approximately related to the phase margin by:

$$\xi_{BF} = \frac{M_\varphi^\circ}{100} \tag{2.3}$$

- *Overshoot limitation*, defined as the maximum peak of the response relative to the final steady-state value. If the final value differs from unity, the maximum overshoot expressed in percentage is used:

$$d\% = 100 \cdot \frac{s(t_p) - s(\infty)}{s(\infty)} \tag{2.4}$$

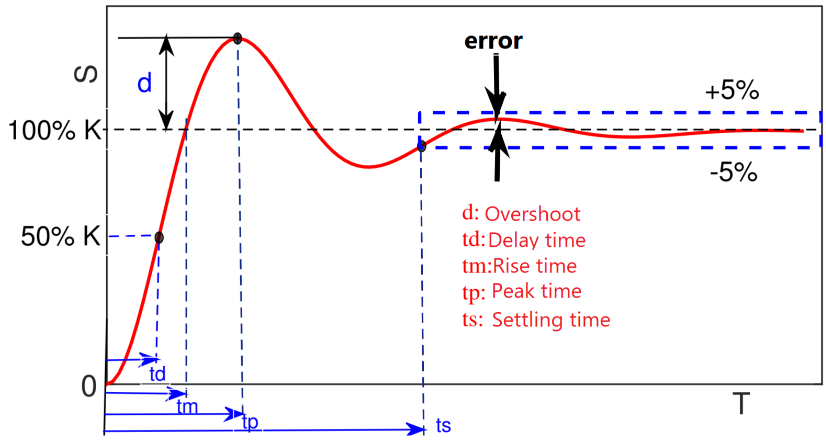


Figure 2.4: Typical performance metrics of a feedback control system

3 Elementary corrective actions

Consider the general diagram of a corrected control loop (Figure 2.5), where $C(p)$ is the transfer function of the controller, $G(p)$ is the transfer function of the plant, and $B(p)$ is the transfer function of the sensor. The controller contains three elementary corrective actions (P, I, or D), which individually help improve specific performances. They are relatively simple to implement but generally degrade other performances. These actions are suitable when the specifications are not very demanding. Otherwise, it is necessary to combine these different actions within a more complex controller.

3.1 Proportional controller

The controller is a simple adjustable gain amplifier $C(p) = K$, whose purpose is to modify the system's initial static gain. The influence of the static gain on performance can be deduced from the Bode diagrams shown in Figure 2.6.

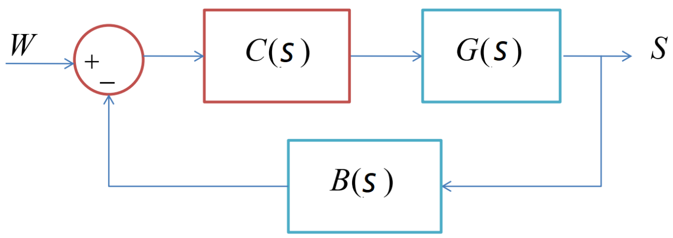


Figure 2.5: Functional block diagram of a control loop

- If $K > 1 \Rightarrow$ upward shift of the gain diagram $\Rightarrow$ increase of the crossover frequency ω_{c0}, which improves the speed ($t_m = 3/\omega_{c0}$) and precision. However, increasing ω_{c0} decreases the phase margin, degrading stability and increasing overshoot.
- If $K < 1 \Rightarrow$ downward shift of the gain diagram $\Rightarrow$ decrease of ω_{c0}, which degrades speed and precision. Conversely, decreasing ω_{c0} increases the phase margin, improving stability but at the cost of higher overshoot.

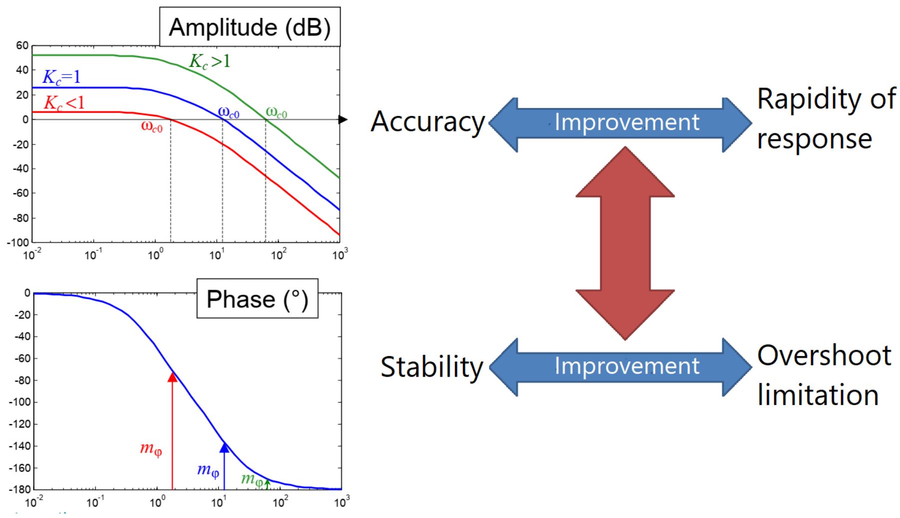


Figure 2.6: Influence of static gain on performance

Electronic implementation

Inverting amplifier This is the most commonly used operational amplifier circuit, shown in Figure 2.7.

The operational amplifier operates in linear mode:

$$\begin{cases} V_d = 0 \\ i^+ = i^- = 0 \Rightarrow i_1 = i_2 \end{cases} \tag{2.5}$$

From loop (1):

$$V_e - R_1 i_1 + V_d = 0 \Rightarrow i_1 = \frac{V_e}{R_1} \tag{2.6}$$

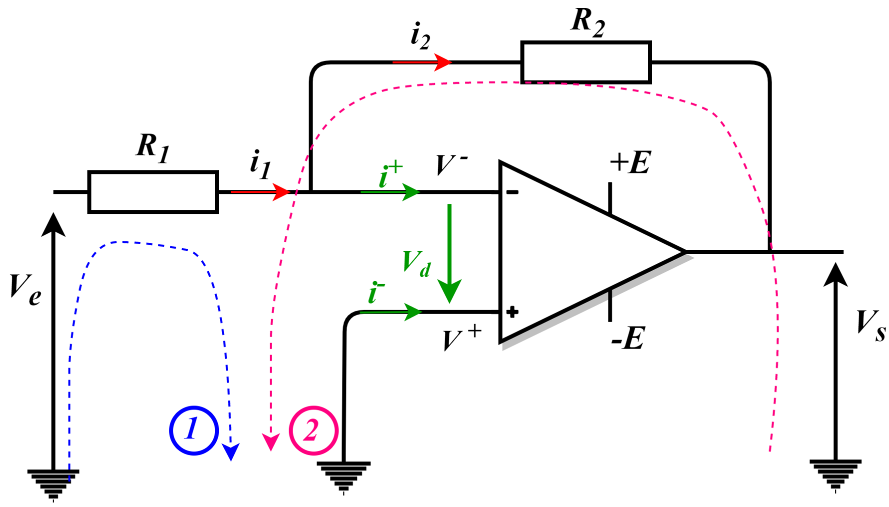


Figure 2.7: Inverting amplifier

From loop (2):

$$V_s + R_2 i_2 + V_d = 0 \Rightarrow i_2 = -\frac{V_s}{R_2} \tag{2.7}$$

Thus:

$$A_v = \frac{V_s}{V_e} = -\frac{R_2}{R_1} \tag{2.8}$$

If $R_1 = R_2$:

$$A_v = -1 \tag{2.9}$$

The amplifier is called inverting because the voltage gain is negative: input and output voltages are in opposite phase.

Non-inverting amplifier Now consider the circuit in Figure 2.8.

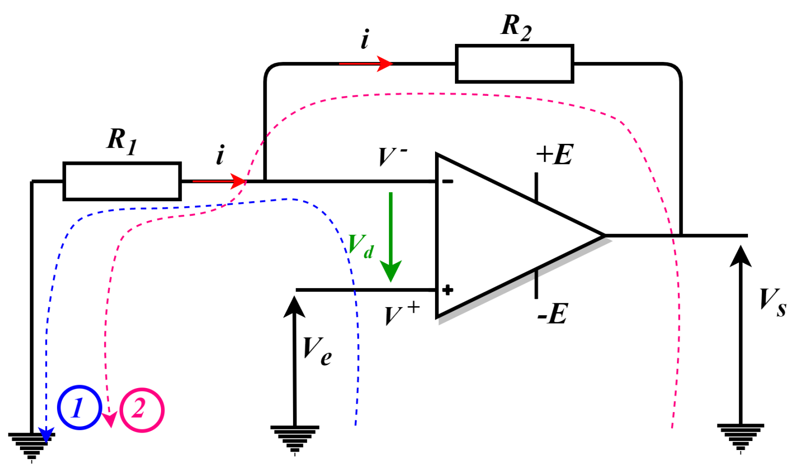


Figure 2.8: Non-inverting amplifier

From loop (1):

$$V_e - V_d - R_1 i = 0 \Rightarrow i = -\frac{V_e}{R_1} \tag{2.10}$$

From loop (2):

$$V_s + (R_1 + R_2)i = 0 \Rightarrow i = -\frac{V_s}{R_1 + R_2} \tag{2.11}$$

Therefore:

$$A_v = \frac{V_s}{V_e} = 1 + \frac{R_2}{R_1} \tag{2.12}$$

The gain is positive and always greater than one.

3.2 Integral controller

The controller is a pure integrator with transfer function:

$$C(p) = \frac{1}{p} \tag{2.13}$$

According to the Bode diagrams in Figure 2.9, adding a pure integrator improves precision (eliminates the steady-state error and reduces the speed error if the original system is class 0) and ensures asymptotic rejection of constant disturbances. However, it reduces the crossover frequency ω_{c0}, lowering the speed. It also decreases the phase margin, which degrades stability and can even lead to instability.

In summary, integral action improves only precision, while other performances are degraded.

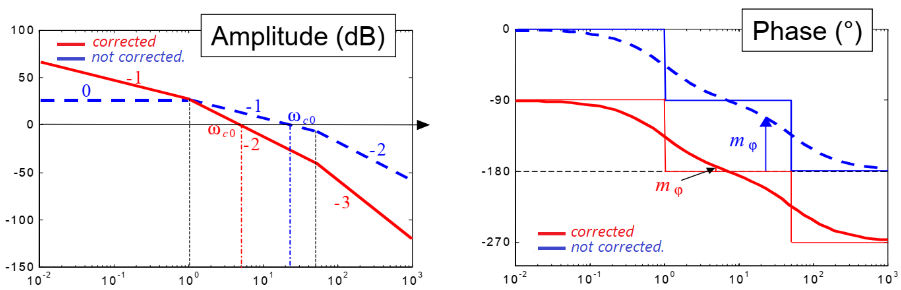


Figure 2.9: Influence of an integrator on performance

Electronic implementation

By replacing resistor R_2 in the inverting amplifier circuit with a capacitor, we obtain an integrator circuit.

$$V_e(t) = Ri(t) \tag{2.14}$$

$$V_C(t) = -V_s(t) \tag{2.15}$$

Since:

$$i_c(t) = C\frac{dV_C(t)}{dt} = C\frac{d(-V_s(t))}{dt} \tag{2.16}$$

We get:

$$V_s(t) = -\frac{1}{RC}\int V_e(t)dt \tag{2.17}$$

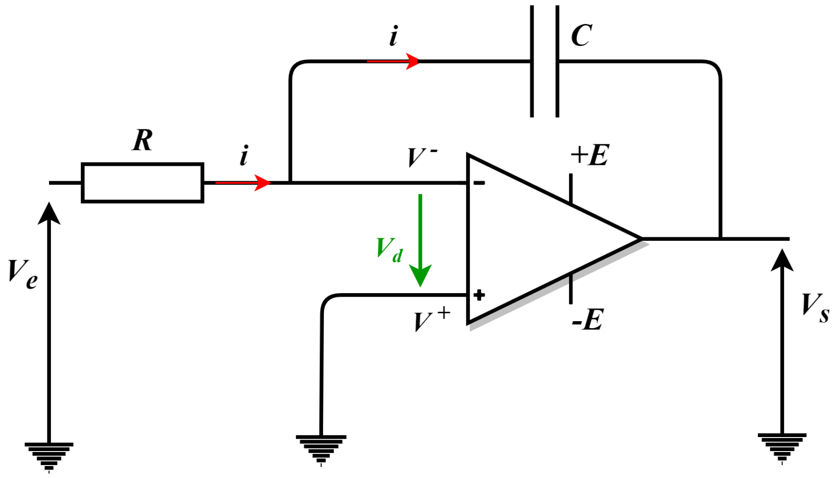


Figure 2.10: Integrator circuit

3.3 Derivative controller

The controller is a pure differentiator with transfer function:

$$C(p) = p \tag{2.18}$$

Its purpose is to add a zero at the origin to the open-loop transfer function. The derivative action improves only speed; other performances are degraded.

Electronic implementation

The differentiator circuit is shown in Figure 2.11. It resembles the inverting amplifier, but with the resistor and capacitor positions swapped.

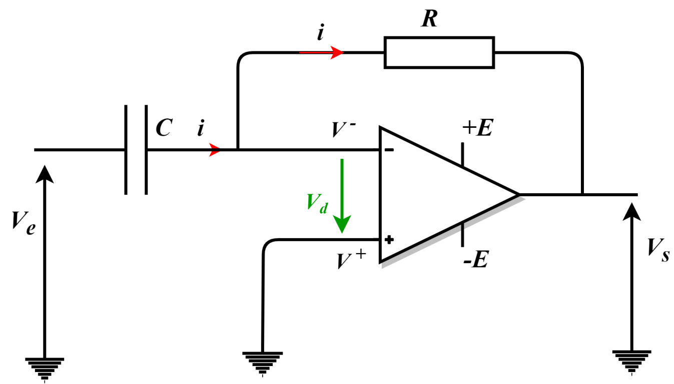


Figure 2.11: Differentiator circuit

$$V_s(t) = -Ri(t) \tag{2.19}$$

$$V_C(t) = V_e(t) \tag{2.20}$$

$$i_c(t) = C\frac{dV_e(t)}{dt} \tag{2.21}$$

Therefore:

$$V_s(t) = -RC\frac{dV_e(t)}{dt} \tag{2.22}$$

At high frequencies, this circuit becomes unstable, leading to oscillations. To mitigate this, a resistor is added in series with the capacitor.

4 Phase Lag Compensator

The phase lag compensator is an approximate form of the PI controller. It performs an integral-like action (increasing the low-frequency gain) without introducing an integrator. Its transfer function is:

$$C(s) = \frac{a(1+Ts)}{1+aTs} \quad \text{with} \quad a > 1 \tag{2.23}$$

It is generally used to impose a specified steady-state error, phase margin, or speed of response. To better understand this compensator, let us consider its Bode diagram (Fig. 2.12). Examining the Bode diagram shown in Fig. 2.12 helps predict the effect of this compensator. When placed in cascade with the system to be corrected, the two Bode diagrams add together. The static gain is thus increased by $20\log(a)$, which improves accuracy. By setting the parameter T to a sufficiently low value, this correction only affects low frequencies; the high-frequency gain is practically unchanged. The additional negative phase shift introduced by the compensator also occurs at low frequencies. Therefore, it does not affect the stability margin, since the crossover frequency at 0 dB is usually located in higher frequency ranges.

In any case, to tune the phase lag compensator, one chooses the value of a that ensures the desired resulting static gain, then chooses T so that $1/T \ll \omega_{c0}$.

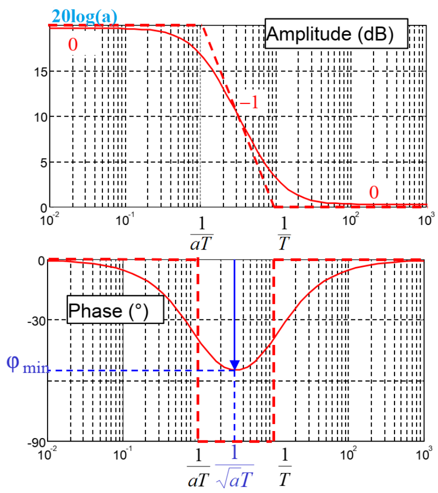


Figure 2.12: Bode diagram of a phase lag compensator

Example 8 *Consider the system with transfer function $G(s)$ in a unity-feedback loop, given by:*

$$G(s) = \frac{1}{\left(1+\frac{s}{10}\right)^3} \tag{2.24}$$

Propose a compensator that meets the following specifications: phase margin $M_\varphi = 45°$ and position error.

Solution: First, we look for a proportional compensator that ensures a closed-loop system phase margin of $M_\varphi = 45°$. The open-loop transfer function with the compensator becomes:

$$G(s) = \frac{K}{\left(1+\frac{s}{10}\right)^3} \quad \Rightarrow \quad \begin{cases} G(\omega) = \frac{K}{\left(\sqrt{\frac{\omega^2}{100}+1}\right)^3} \\ \varphi(\omega) = -3\arctan\left(\frac{\omega}{10}\right) \end{cases} \tag{2.25}$$

We have $M_\varphi = \pi + \varphi(\omega_{c0})$ with $G(\omega_{c0}) = 1$.

$$M_\varphi = \pi - 3\arctan\left(\frac{\omega_{c0}}{10}\right) = \frac{\pi}{4} \Rightarrow \omega_{c0} = 10\,\mathrm{rad/s} \tag{2.26}$$

Then,

$$G(10) = \frac{K}{\left(\sqrt{\frac{100}{100}+1}\right)^3} = 1 \Rightarrow K = 2.8 \tag{2.27}$$

The new open-loop transfer function with the phase lag compensator is:

$$G(s) = \frac{(1+Ts)}{1+aTs} \cdot \frac{2.8a}{\left(1+\frac{s}{10}\right)^3} \tag{2.28}$$

We calculate the steady-state position error in closed loop as follows:

$$\varepsilon_p = \lim_{s\to 0}(1 - \mathrm{FTBF}(s)) = 1 - \frac{2.8a}{2.8a+1} \tag{2.29}$$

To achieve a position error of 5%, it is necessary to set:

$$\varepsilon_p = 1 - \frac{2.8a}{2.8a+1} = 0.05 \Rightarrow a = 6.8 \tag{2.30}$$

Finally, simply choose T so that $1/T$ is much lower than the crossover frequency at 0 dB. For example, we can take $T = 10\,\mathrm{s}$.

Thus, we finally obtain:

$$G(s) = 6.8\frac{(1+10s)}{1+68s} \tag{2.31}$$

The Matlab code [Script 4] gives the same results as found above. The Bode diagram of the compensated system is shown in Fig. 2.13.

```
%% Script: Example 7
clc, close all, clear all;
%% Transfer function of the system
NUM = 2.8;
DEN1 = conv([1/10 1], [1/10 1]); %
DEN = conv(DEN1, [1/10 1]);      %
G = tf(NUM, DEN);                % Define the transfer function ...
    G(s) of the plant
%% Transfer function of the controller
C = tf([68 6.8], [68 1]);%
%% Transfer function of the system with controller
Gc = series(G, C);%
%% Bode diagram of the compensated system
margin(Gc);%
```

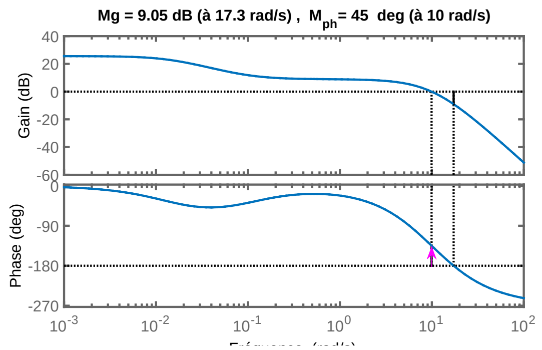


Figure 2.13: Bode diagram of the compensated system 2.24

Electronic Circuit

This circuit is an electrical implementation of a phase lag compensator designed using control theory methods.

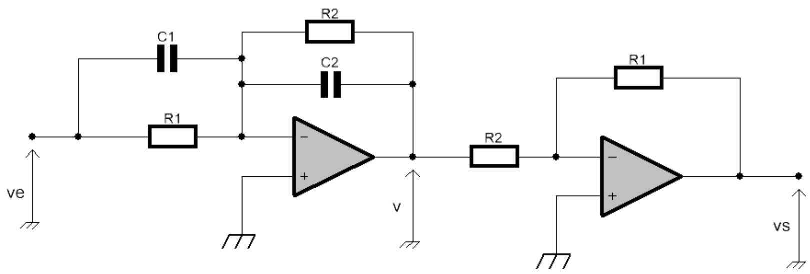


Figure 2.14: Phase lag compensator circuit.

Let us determine the voltage transfer function of this circuit, i.e., the transfer function between output and input. Both stages are inverting amplifiers:

$$\frac{V_s}{V_e} = \frac{1 + R_1 C_1 s}{1 + R_2 C_2 s} \text{ with } R_2 C_2 > R_1 C_1 \tag{2.32}$$

5 Phase-Lead Compensator

The phase-lead compensator, as its name suggests, is a compensator designed to increase the phase margin of a system. It is used to compensate for insufficient phase shift around the crossover frequency at 0 dB. Its transfer function is:

$$C(s) = \frac{1 + aTs}{1 + Ts} \quad \text{with} \quad a > 1 \tag{2.33}$$

The advantage of this compensator becomes evident on the Bode diagram (2.15), where at the frequency ω_{max}, the phase shift reaches a maximum value φ_{max}, given by:

$$\omega_{max} = \frac{1}{T\sqrt{a}} \quad \text{and} \quad \varphi_{max} = \arcsin\frac{a-1}{a+1} \tag{2.34}$$

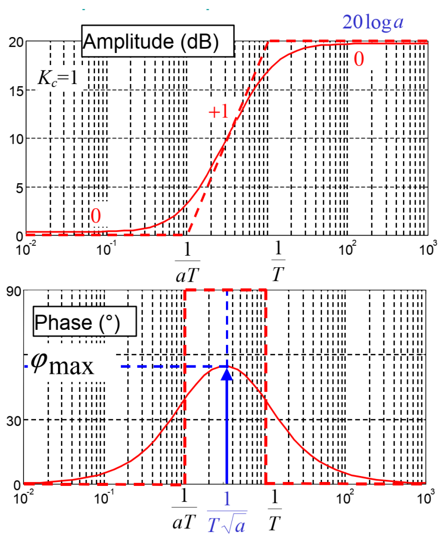


Figure 2.15: Bode diagram of a phase-lead compensator

The idea of this compensator design is to align ω_{max} with the crossover frequency ω_{co} of the system to be compensated and to set φ_{max}, known as the phase boost, in order to achieve the desired phase margin.

Example 9 *Consider a system whose open-loop transfer function (OLTF) is defined by:*

$$G(s) = \frac{100}{(1+s)^2} \tag{2.35}$$

Calculate the parameters of a phase-lead compensator that provides a phase margin of $M_\varphi = 45°$.

Solution: We first calculate the phase margin without the compensator. The OLTF is given by:

$$G(s) = \frac{100}{(1+s)^2} \quad \Rightarrow \quad \begin{cases} G(\omega) = \frac{100}{\omega^2+1} \\ \varphi(\omega) = -2\arctan(\omega) \end{cases} \tag{2.36}$$

We have $M_\varphi = \pi + \varphi(\omega_{c0})$ with $G(\omega_{c0}) = 1$:

$$G(\omega_{c0}) = \frac{100}{\omega_{c0}^2+1} = 1 \Rightarrow \omega_{c0} = 9.9\,\mathrm{rad/s} \tag{2.37}$$

Thus:

$$M_\varphi = \pi - 2\arctan(9.9) \Rightarrow M_\varphi = 11° \tag{2.38}$$

To obtain a phase margin of 45°, we need a phase boost of 34° at the frequency ω_{c0}. We introduce a phase-lead compensator designed so that:

$$\varphi_{max} = 45° - 11° = 34° = \arcsin\frac{a-1}{a+1} \Rightarrow a = \frac{1+\sin 34°}{1-\sin 34°} = 3.54 \tag{2.39}$$

We also have:

$$\frac{1}{T\sqrt{a}} = \omega_{c0} \Rightarrow \frac{1}{T\sqrt{3.54}} = 9.9 \Rightarrow T = 0.053\,\mathrm{s} \tag{2.40}$$

Finally:

$$C(s) = \frac{1 + 0.19s}{1 + 0.053s} \tag{2.41}$$

The new open-loop transfer function of the compensated system becomes:

$$FTBO_c = G(s)C(s) = \frac{100}{(1+s)^2} \cdot \frac{1 + 0.19s}{1 + 0.053s} \tag{2.42}$$

The Matlab code [Script 5] can be used to reproduce these results.

```
%% Script: Example 8
clc, close all, clear all;
%% Transfer function of the system
NUM = 100;                  % Numerator
DEN = conv([1 1], [1 1]); % Denominator
G = tf(NUM, DEN);

%% Transfer function of the controller
C = tf([68 6.8], [68 1]);
%% Transfer function of the compensated system
Gc = series(G, C);
%% Bode diagram
margin(Gc);
```

Electronic Circuit

This circuit is identical to that of a phase-lag compensator, but with different conditions on R_1, R_2, C_1, and C_2. In this case, the transfer function is:

$$\frac{V_s}{V_e} = \frac{1 + R_1C_1s}{1 + R_2C_2s} \text{ with } R_1C_1 > R_2C_2 \tag{2.43}$$

This circuit in Figure 2.16 is identical to the phase-lead circuit, but with different

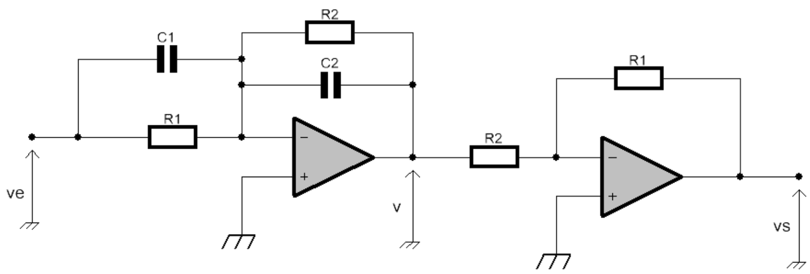


Figure 2.16: Phase lag compensator circuit.

conditions on R_1, R_2, C_1, and C_2.

6 PID Controller Structures

Three typical structures can be identified for PID controllers:

6.1 Parallel Structure

The time-domain expression relating the control signal $u(t)$ to the error signal $\varepsilon(t)$ in this structure is given by:

$$u(t) = K_p\varepsilon(t) + T_d\frac{d\varepsilon(t)}{dt} + \frac{1}{T_i}\int \varepsilon(t)dt \tag{2.44}$$

The transfer function of the PID controller can thus be written as:

$$C(s) = \frac{U(s)}{\varepsilon(s)} = K_p + T_d s + \frac{1}{T_i s} \tag{2.45}$$

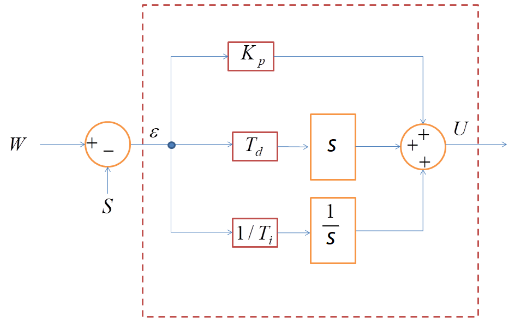


Figure 2.17: Parallel structure of the PID controller

6.2 Series Structure

The transfer function of the series PID controller can be expressed as:

$$C(s) = \frac{U(s)}{\varepsilon(s)} = K_p(1 + T_d s)\big(1 + \frac{1}{T_i s}\big) \tag{2.46}$$

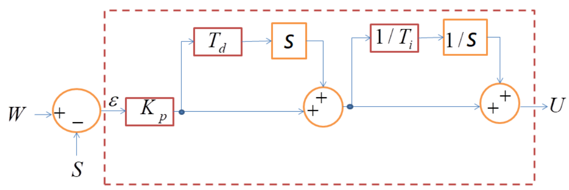


Figure 2.18: Series structure of the PID controller

6.3 Mixed Structure

The transfer function of the mixed PID controller is given by:

$$C(s) = \frac{U(s)}{\varepsilon(s)} = K_p\big(1 + T_d s + \frac{1}{T_i s}\big) \tag{2.47}$$

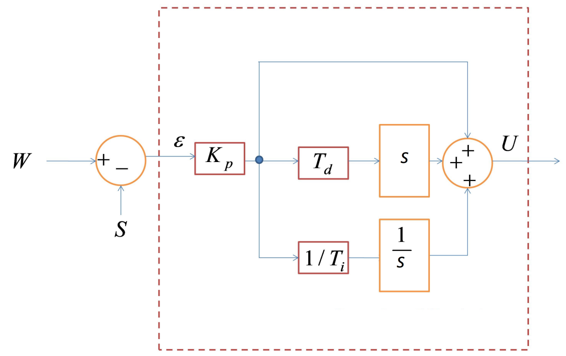


Figure 2.19: Mixed structure of the PID controller

7 Design Methods

The problem of determining the controller parameters is known as controller Design. There is a wide range of Design methods, and a rigorous classification is not always straightforward. However, within the scope of this section, two main types of methods can be distinguished:

- *Empirical methods* do not require perfect knowledge of the process model. The controller parameters are calculated based on experimental observations of the process (such as the step response). The main advantage of these methods lies in their simplicity. They are widely used in industry and are often sufficient, although they do not allow for fine-tuning.
- Methods based on *knowledge of the system model*, for example given in the form of a transfer function. The controller parameters are then computed in order to achieve the desired transfer function in open-loop or closed-loop configurations.

7.1 Empirical Methods

In 1942, Ziegler and Nichols proposed two heuristic approaches based on their experience and some simulations to quickly tune the parameters of P, PI, and PID controllers. The first method requires recording the open-loop step response, whereas the second involves bringing the closed-loop system to the verge of instability.

Ziegler–Nichols Open-Loop Method

This method applies primarily to systems with a time delay whose behavior resembles that of a first-order system. Such response types are often encountered in chemical and thermal processes.

$$G(s) = \frac{Ke^{-\tau s}}{(T_s + 1)} \tag{2.48}$$

To determine the PID controller parameters, one records the step response of the system to an input of amplitude E_0 alone (without controller), then draws the tangent at the inflection point of the curve. The apparent delay is measured as

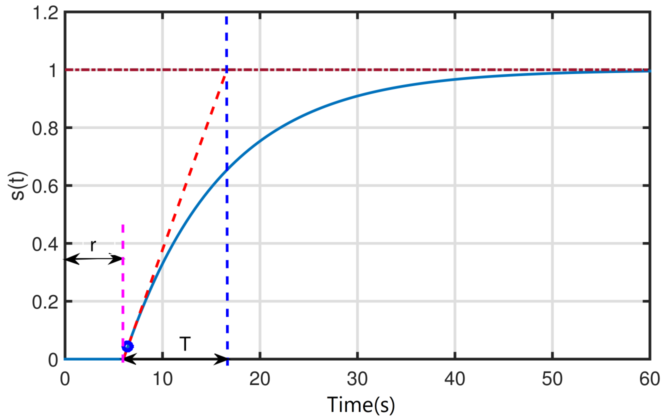


Figure 2.20: Step response of an open-loop system

the intersection of this tangent with the time axis, and the final value is noted, as illustrated in Figure (2.20).

We seek the maximum slope $R = K/\tau$ (with $\tau = T$). Then, the parameters of the chosen controller are calculated using Table 2.1.

Table 2.1: PID parameters: Ziegler–Nichols open-loop method

Controller type	K_p	T_i	T_d
P: $C(s) = K_p$	$\frac{E_0}{rR}$	–	–
PI: $C(s) = K_p\big(1 + \frac{1}{T_i s}\big)$	$0.9\frac{E_0}{rR}$	$0.33\frac{E_0}{rR^2}$	–
PID: $C(s) = K_p\big(1 + \frac{1}{T_i s} + T_d s\big)$	$1.2\frac{E_0}{rR}$	$0.6\frac{E_0}{rR^2}$	$0.6\frac{E_0}{R}$

Example 10 *Consider the system whose open-loop transfer function is defined by:*

$$G(s) = \frac{e^{-4s}}{(1+s)^3} \tag{2.49}$$

Compute the parameters of a PID controller using the Ziegler–Nichols open-loop method, given that its step response is shown in Figure (2.21).

Solution: We have $R = K/\tau = 1/4.0125 = 0.2492$ with $r = 3.2972$. Therefore, the PID controller parameters derived from Table 2.1 are:

$$\begin{cases} K_p = 1.2\frac{E_0}{rR} = \frac{1.2}{(3.2972)(0.2492)} = 1.4605 \\ T_i = 0.6\frac{E_0}{rR^2} = \frac{0.6}{(3.2972)(0.2492)^2} = 2.9303 \\ T_d = 0.6\frac{E_0}{R} = \frac{0.6}{0.2492} = 2.4077 \end{cases} \tag{2.50}$$

Finally:

$$C(s) = 1.4605\Big(1 + \frac{1}{2.9303s} + 2.4077s\Big) \tag{2.51}$$

The Matlab code [Script 6] can be used to obtain the above results.

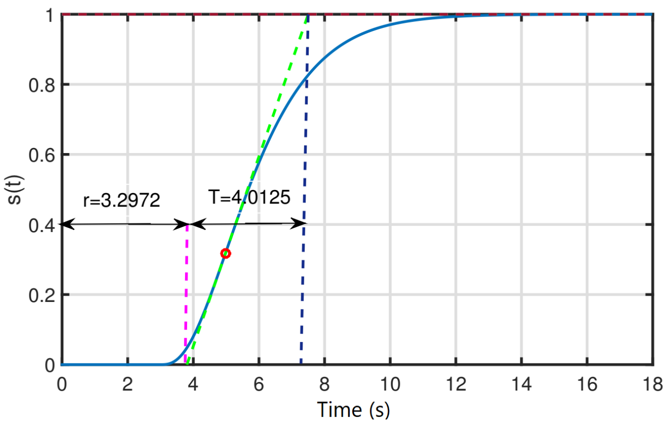


Figure 2.21: Step response of the system in (2.21)

```
clc; close all;  clear all;
K = 1;   % System gain
Gs1 = tf([K], conv([1 2 1], [1 1]), 'InputDelay', 3);
% Simulate the step response of the system
[y, t] = step(Gs1);
% Estimate the sample time based on the time vector
h = mean(diff(t));
% Compute the numerical derivative of the step response
dy = gradient(y, h);
% Find the index where the derivative reaches its maximum ...
    (inflection point)
[¬, idx] = max(dy);
% Fit a straight line (tangent) using two points around the ...
    inflection point
% Solving b(1)*t + b(2) = y for the two chosen points
b = [t([idx-1, idx+1]) ones(2,1)] \ y([idx-1, idx+1]);
% Calculate the intercepts of the tangent line with y=0 and y=1
tv = [-b(2)/b(1); (1-b(2))/b(1)];
% Evaluate the tangent line at these time values
f = [tv ones(2,1)] * b;
% Plot the results
figure
plot(t, y)                                      % Plot the ...
    original step response
hold on
plot(tv, f, '--g')                              % Plot the ...
    tangent line in dashed green
plot(t(idx), y(idx), 'or')                      % Mark the ...
    inflection point with a red circle
plot(t, K*ones(length(t)), '--')                 % Plot the ...
    final steady-state value as a dashed horizontal line
hold off
grid on
```

Ziegler–Nichols Closed-Loop Method

This method requires closing the loop with a simple proportional controller whose gain is increased until the system exhibits sustained oscillations, as shown in Figure (2.22).

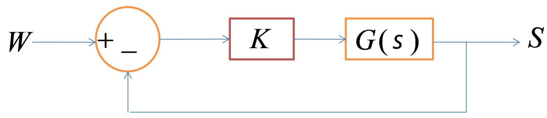


Figure 2.22: System under study: Ziegler–Nichols

At this point, the system is on the verge of instability. Once the critical gain K_{cr} and the oscillation period T_{cr} are measured from the response, the controller parameters are obtained using Table 2.2.

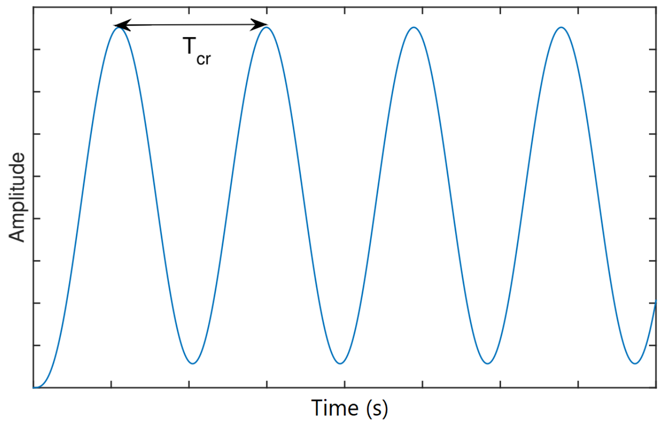


Figure 2.23: Step response of a system

Table 2.2: Ziegler–Nichols closed-loop method

Controller type	K_p	T_i	T_d
P: $C(s) = K_p$	$0.5K_{cr}$	–	–
PI: $C(s) = K_p\left(1 + \frac{1}{T_i s}\right)$	$0.45K_{cr}$	$0.83T_{cr}$	–
PID: $C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$	$0.6K_{cr}$	$0.5T_{cr}$	$0.125T_{cr}$

Example 11 *Consider the system whose open-loop transfer function is defined by:*

$$G(s) = \frac{3}{(s+1)(s+2)(s+3)} \tag{2.52}$$

Compute the parameters of a PID controller using the Ziegler–Nichols closed-loop method.

Solution: We must find the critical gain K_{cr} and the critical period T_u. For a simple system like this, it suffices to use the Routh table to determine the critical gain, then simulate to measure the period.

The Routh table is as follows:

1	11
6	$6+3K$
$\frac{66-(6+3K)}{6}$	0
$6+3K$	0

For stability, $66-(6+3K) > 0 \Rightarrow K < 20$. Thus, the critical gain is $K_{cr} = 20$. The system's step response shows that the critical period is $T_u = 1.9\,\text{s}$.

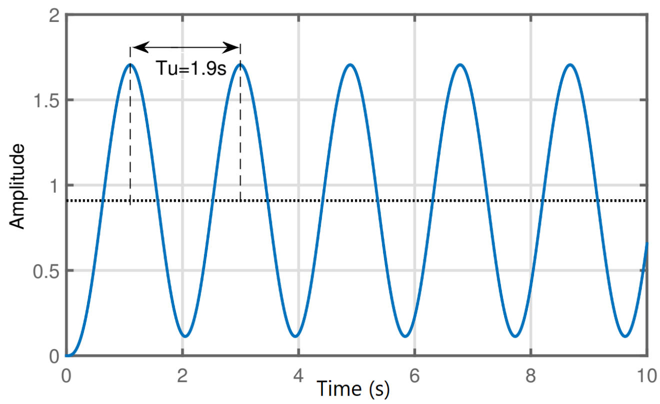


Figure 2.24: Step response of the system

Alternatively, the critical period can be determined using the oscillation frequency ω_c of the system with the critical gain. One uses the second row of the Routh table and isolates ω_c by substituting $s = j\omega_c$:

$$6s^2 + 6 + 3K = 6s^2 + 6 + 60 = 6s^2 + 66 \tag{2.53}$$

Substituting $s = j\omega_c$ gives:

$$6(-\omega_c^2) + 66 = 0 \implies \omega_c^2 = 11 \implies \omega_c = \sqrt{11}$$

The critical period is then:

$$T_u = \frac{2\pi}{\omega_c} = \frac{2\pi}{\sqrt{11}} \approx 1.89\,\text{s}$$

Finally, using Table 2.2:

$$\begin{cases} K_p = 0.6K_{cr} = 0.6 \times 20 = 12 \\ T_i = 0.5T_u = 0.5 \times 1.9 = 0.95 \\ T_d = 0.125T_u = 0.125 \times 1.9 = 0.2375 \end{cases}$$

$$C(s) = 12\Big(1 + \frac{1}{0.95s} + 0.2375s\Big) \tag{2.54}$$

7.2 Algebraic Methods

Algebraic or theoretical methods are numerous and rely on the knowledge of an accurate model of the system to be controlled. The actual performance achieved depends on the quality of this model and its ability to accurately represent the

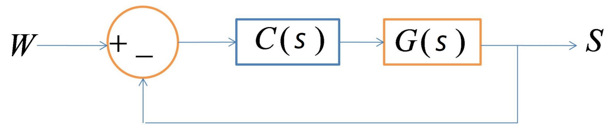


Figure 2.25: Control loop of a system

system. Given this model, it becomes possible to define the characteristics of the controller that will allow the system to be controlled as closely as possible using one of the algebraic design methods.

Consider the control loop illustrated in Figure 2.25:

The closed-loop transfer function is:

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{2.55}$$

The idea of this method is based on the prior definition of a desired closed-loop behavior $H_d(s)$ that meets a well-specified set of requirements. That is, we set $H(s) = H_d(s)$. In this case, the controller $C(s)$ can be derived as follows:

$$C(s) = \frac{H_d(s)}{G(s)(1 - H_d(s))} \tag{2.56}$$

Usually, the desired closed-loop behavior is that of a first-order or second-order system with a unit static gain, which ensures perfect steady-state accuracy.

Example 12 *Consider the system whose open-loop transfer function $G(s)$ is defined by:*

$$G(s) = \frac{10}{(1 + 25s)(1 + 10s)} \tag{2.57}$$

Determine the PID controller that meets the following requirements: a first-order closed-loop behavior with a 5% settling time $t_r = 0.6\,\mathrm{s}$ and perfect steady-state accuracy.

Solution: The general form of a first-order system is given by:

$$H_d(s) = \frac{K}{1 + Ts} \tag{2.58}$$

According to the specifications, $t_r = 3T \Rightarrow T = t_r/3 = 0.6/3 = 0.2\,\mathrm{s}$, and perfect steady-state accuracy implies $K = 1$. Thus, the desired closed-loop behavior becomes:

$$H_d(s) = \frac{1}{1 + 0.2s} \tag{2.59}$$

From 2.56, the transfer function of the controller can be deduced as follows:

$$C(s) = \frac{\frac{1}{1+0.2s}}{\frac{10}{(1+25s)(1+10s)}\left(1 - \frac{1}{1+0.2s}\right)} = \frac{250s^2 + 35s + 1}{2s} \tag{2.60}$$

The parallel PID form can be extracted from 2.60:

$$C(s) = \frac{35}{2} + \frac{1}{2s} + 125s \tag{2.61}$$

The response of the controlled system to a unit step input is shown in Figure 2.26. The results of Example 12 can be obtained using the Matlab script [Script 7].

```matlab
% Script: Example 11
clc; close all; clear all;
den = conv([25 1], [10 1]);
num = 10;
G = tf(num, den);
Nc = [250 35 1];
Dc = [2 0];
C = tf(Nc, Dc);
% Compute the open-loop system
Gs = series(G, C);
% Compute the closed-loop transfer function with unity feedback
H = feedback(Gs, 1);
% Simulate the step response of the closed-loop system
[y, t] = step(H);
% Plot the step response
figure
plot(t, y, 'b')                              % Plot the system ...
    response in blue
hold on
plot(t, ones(length(t)), '--r')              % Plot the ...
    reference (setpoint) as a red dashed line
% Add plot annotations
xlabel('Time (s)')
ylabel('Amplitude')
title('Step Response of the Closed-Loop System')
legend('System Response', 'Reference')
grid on
hold off
```

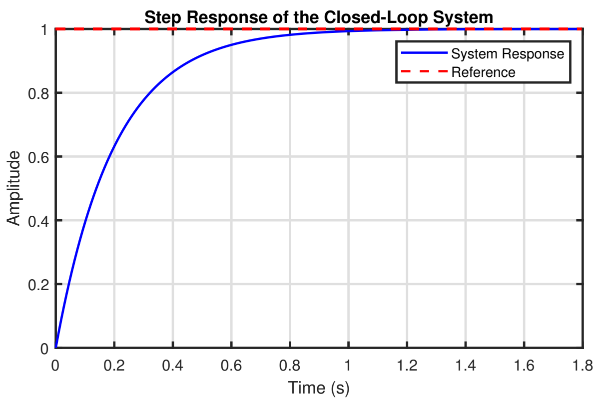


Figure 2.26: Closed-loop response of the system

PI and IP controller design for first-order systems

In this section, we present several methods for designing PI and IP (Integral–Proportional) controllers for first-order systems.

PI controller design

Consider the control loop illustrated in Figure 2.27.

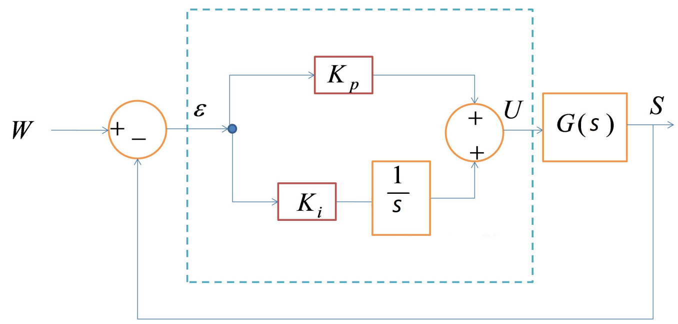


Figure 2.27: Control loop of a system

Let $G(s)$ be a first-order system represented by the transfer function:

$$G(s) = \frac{K}{1 + Ts} \tag{2.62}$$

where K is the static gain and T is the time constant. The transfer function of the PI controller is:

$$C(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \tag{2.63}$$

There are several algebraic methods to determine the PI controller parameters. In this course, we present the following two methods:

Pole placement method

From the control loop in Figure 2.27, the closed-loop transfer function can be written as:

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} = \frac{K(K_p s + K_i)}{Ts^2 + (KK_p + 1)s + KK_i} \tag{2.64}$$

This can be rewritten as:

$$H(s) = \frac{K(K_p s + K_i)/T}{s^2 + \frac{KK_p+1}{T}s + \frac{KK_i}{T}} \tag{2.65}$$

The general form of a second-order system is given by:

$$H_d(s) = \frac{K_s \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{2.66}$$

where K_s is the static gain, ω_n is the natural frequency, and ξ is the damping ratio. To place the closed-loop poles so that the system is underdamped ($0.7 \leq \xi < 1$), by identifying coefficients with 2.65, we obtain:

$$\begin{cases} \frac{KK_p+1}{T} = 2\omega_n\xi \quad \Rightarrow \quad K_p = \frac{2\omega_n\xi T - 1}{K} \\ \frac{KK_i}{T} = \omega_n^2 \quad \Rightarrow \quad K_i = \frac{\omega_n^2 T}{K} \end{cases} \tag{2.67}$$

Pole-zero cancellation method

In the pole placement method, the closed-loop transfer function contains a zero that can affect the transient response. To avoid this, the pole-zero cancellation method sets the controller zero equal to a system pole and selects a time constant to meet design objectives.

The closed-loop transfer function becomes:

$$H(s) = \frac{K(K_p s + K_i)}{s(1+Ts) + (K_p s + K_i)} = \frac{KK_i\left(\frac{K_p}{K_i}s + 1\right)}{s(1+Ts) + (K_p s + K_i)} \tag{2.68}$$

We set:

$$\frac{K_p}{K_i}s + 1 = 1 + Ts \tag{2.69}$$

Then:

$$H(s) = \frac{KK_i}{s + KK_i} = \frac{1}{\frac{1}{KK_i}s + 1} \tag{2.70}$$

In this case, we impose a first-order closed-loop behavior characterized by a static gain K_s and a time constant T_s:

$$H_d(s) = \frac{K_s}{1 + T_s s} \tag{2.71}$$

From the condition...

IP Controller design

The control of first-order systems using Integral-Proportional (PI) controllers, designed by pole placement or pole compensation methods, presents the following drawbacks:

- The pole placement method allows imposing any closed-loop dynamics; however, it has the disadvantage of introducing an (uncontrollable) zero that can alter the imposed dynamics.
- The pole compensation method overcomes this issue, but it can only impose first-order dynamics. Moreover, the regulation behavior with respect to disturbances is poor, since the disturbance rejection dynamics are not imposed by the controller. The IP controller structure addresses all these limitations.

The structure of the IP controller is illustrated in Figure (2.28). It consists of an inner loop with a proportional controller and an outer loop with an integral controller.

From the regulation loop shown in Figure (2.28), the closed-loop transfer function can be written as:

$$H(s) = \frac{KK_pK_i}{Ts^2 + (1 + KK_p)s + KK_pK_i} = \frac{\frac{KK_pK_i}{T}}{s^2 + \frac{1+KK_p}{T}s + \frac{KK_pK_i}{T}} \tag{2.72}$$

By comparing this with the general form of a second-order system (2.67), we can write:

$$\begin{cases} \frac{1+KK_p}{T} = 2\omega_n\xi \Rightarrow K_p = \frac{2\omega_n\xi T - 1}{K} \\ \frac{KK_pK_i}{T} = \omega_n^2 \Rightarrow K_i = \frac{T\omega_n^2}{2\omega_n\xi T - 1} \end{cases} \tag{2.73}$$

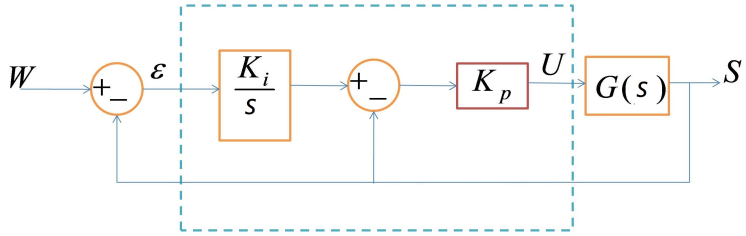


Figure 2.28: Control using an IP controller

Example of controlling a first-order system

In this section, we address the problem of closed-loop control of a first-order system using a PI controller. We will design the parameters K_p and K_i by pole placement and pole compensation methods. Then, we will control this system using an IP controller.

Example 13 *Consider the first-order system with the open-loop transfer function defined by:*

$$G(s) = \frac{10}{(0.5s+1)} \tag{2.74}$$

Calculate the parameters of the PI and IP controllers that guarantee a response time $t_r \leq 0.5\,\mathrm{s}$ *and an overshoot* $D\% \leq 10\%$*.*

To solve this problem, we directly use the expressions obtained above (2.67), (**??**), and (2.74). The Matlab script [Script 5] provides the complete solution to this exercise.

```matlab
% Script: Example 12
clc;
close all;
clear all;

% System open-loop transfer function (plant)
T = 0.5;                 % Time constant of the plant
K = 10;                  % Static gain of the plant
G = tf(K, [T 1]);        % Transfer function G(s) = 10 / (0.5s + 1)

% Desired closed-loop behavior specifications
wn = 100;                % Desired natural frequency (rad/s)
Ksi = 0.6;               % Desired damping ratio

% Parameters for pole placement design of PI controller
% Compute proportional gain Kp_p and integral gain Ki_p
Kp_p = (2*wn*Ksi*T - 1) / K;
Ki_p = (wn^2*T) / (K*Kp_p);

% Parameters for pole compensation design of PI controller
tr = 0.5;                % Desired rise time (s)
T_s = tr/3;              % Slope calculation constant
Kp_c = T / (K*T_s);
K_i_c = 1 / (K*T_s);

% Parameters for IP controller
Kp_ip = (2*Ksi*wn*T - 1) / K;
Ki_ip = (T*wn^2) / (K*Kp_ip);

% Define PI controllers
Cpip = tf([Kp_p Ki_p], [1 0]);     % PI controller using pole ...
    placement
Cpic = tf([Kp_c K_i_c], [1 0]);    % PI controller using pole ...
    compensation

% Compute new open-loop transfer functions (plant + controller)
Gsp = series(G, Cpip);     % Plant with PI controller (pole ...
    placement)
Gsc = series(G, Cpic);     % Plant with PI controller (pole ...
    compensation)

% Compute closed-loop transfer functions
Hp = feedback(Gsp, 1);     % Closed-loop system with PI ...
    controller (pole placement)
Hc = feedback(Gsc, 1);     % Closed-loop system with PI ...
    controller (pole compensation)
Hip = tf(K*Kp_ip*Ki_ip, [T (1+K*Kp_ip) K*Kp_ip*Ki_ip]);  % ...
    Closed-loop with IP controller

% Define reference input (step input of amplitude 5)
t = 0:0.2:5;               % Time vector from 0 to 5 seconds
u = 5*ones(size(t));       % Step input u(t) = 5

% Simulate and plot system responses to step input
figure
lsim(Hp, u, t);                % Response with PI (pole placement)
hold on
```

```
lsim(Hc, u, t, '--g');          % Response with PI (pole compensation)
hold on
lsim(Hip, u, t, '.--');         % Response with IP controller
plot(t, u, '--r');              % Desired reference input

% Plot formatting
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Response Comparison');
legend('PI with Pole Placement','PI with Pole Compensation','IP ...
    Controller','Reference Input');
grid on;
```

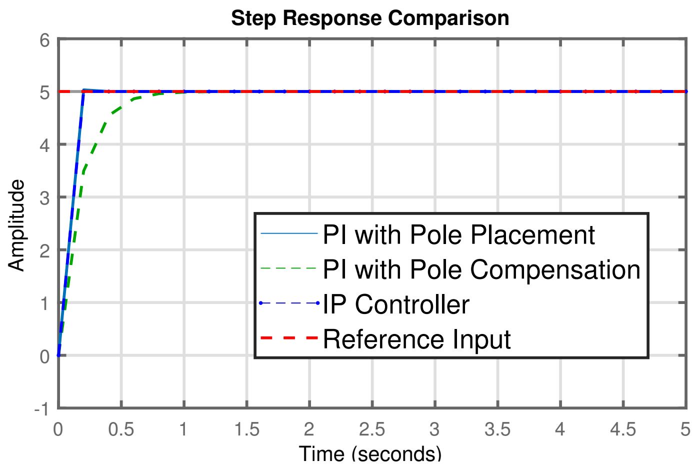


Figure 2.29: Control of a first-order system using PI and IP controllers

Figure (2.29) presents the simulation results of the three methods. It can be clearly observed that the IP controller offers the best performance.

8 Exercises

8.1 Exercise 1

We wish to control a system whose open-loop transfer function is:

$$G(s) = \frac{8}{s^2 + 5s + 6}$$

The system is placed in the direct chain of a regulation loop, in cascade with a proportional controller of gain K. The feedback loop is ensured by a system with transfer function $B(s) = 3$.

- *Determine the necessary condition on* K *so that the system has a phase margin greater than* $45°$.

- *Determine the expression of the new controller* $C(s)$ *that ensures both a phase margin of* $45°$ *and a position error less than* 0.2.

8.2 Exercise 2

We wish to control a system whose open-loop transfer function is:

$$G(s) = \frac{1000}{s(s+10)^2}$$

This system is placed in a unit-feedback loop with a proportional controller of gain K.

- *Determine the value of K that ensures the system a phase margin greater than $45°$.*
- *Calculate the position error and velocity error.*

We now wish to achieve control satisfying the following conditions: zero position error and a velocity error equal to 2%. To do this, we add a phase lag compensator.

Calculate the parameters of the new controller.

8.3 Exercise 3

We wish to control a system whose transfer function is:

$$G(s) = \frac{1000}{s(s+10)^2}$$

The control should meet the following requirements:

- Closed-loop oscillatory response.
- Closed-loop overshoot equal to 5%.
- Closed-loop rise time equal to 0.6 s.
- Zero steady-state error.

- *Propose a controller, knowing that the open-loop system overshoot is 60%.*
- *From the specifications, determine the phase margin required in closed loop and the corresponding gain crossover frequency at 0 dB.*
- *Determine the phase margin the uncompensated system would have at this frequency.*
- *Deduce the controller parameters.*

8.4 Exercise 4

We wish to control a system whose transfer function is:

$$G(s) = \frac{10}{(3s+1)}$$

- *Calculate the parameters of a PI controller by pole placement method, meeting the following conditions: closed-loop oscillatory response, closed-loop overshoot equal to 5%, and closed-loop rise time equal to 0.2 s.*

8.5 Exercise 5

We wish to control a system whose transfer function is:

$$G(s) = \frac{2}{(10s+1)}$$

- *Calculate the parameters of a PI controller by pole compensation method, ensuring a closed-loop rise time equal to* $0.3\,\mathrm{s}$.

8.6 Exercise 6

We wish to control a system whose transfer function is:

$$G(s) = \frac{6}{(50s+1)}$$

- *Calculate the parameters of an IP controller meeting the following conditions: closed-loop oscillatory response, closed-loop overshoot equal to* 5%, *and closed-loop rise time equal to* $0.8\,\mathrm{s}$.

8.7 Exercise 7

We wish to control a system whose transfer function is:

$$G(s) = \frac{25}{(0.1s+1)(0.5s+1)}$$

- *Determine the transfer function of a PID controller that satisfies the following specifications: achieve first-order closed-loop behavior with a* 5% *response time equal to* $0.1\,\mathrm{s}$ *and perfect steady-state accuracy.*

9 Solutions

9.1 Solution to Exercise 1

The open-loop transfer function is:

$$G_o(s) = KG(s)B(s) = \frac{24K}{s^2+5s+6}$$

- *Calculation of* K *ensuring a phase margin equal to* $45°$.
We have:

$$M_\varphi = \pi + \varphi(\omega_{c0}) = \frac{\pi}{4} \Rightarrow \pi - \arctan\left(\frac{6-\omega_{c0}^2}{5\omega_{c0}}\right) = \frac{\pi}{4} \Rightarrow \frac{6-\omega_{c0}^2}{5\omega_{c0}} = \tan\frac{3\pi}{4}.$$

$$\Rightarrow \frac{6-\omega_{c0}^2}{5\omega_{c0}} = -1 \Rightarrow \omega_{c0}^2 - 5\omega_{c0} - 6 = 0 \Rightarrow \begin{cases} \omega_{c0_1} = 6 \\ \omega_{c0_2} = -1 \end{cases}$$

On the other hand, we have:

$$G_o(j\omega_{c0}) = 1 \Rightarrow \frac{24K}{\sqrt{(6-\omega_{c0}^2)^2 + 25\omega_{c0}^2}} = 1$$

$$\Rightarrow K = \frac{\sqrt{(6-\omega_{c0}^2)^2 + 25\omega_{c0}^2}}{24} = 1.76$$

- *Design of a compensator ensuring a phase margin equal to* 45° *and a position error of* 2%*.*

To meet both design requirements, we select a lag compensator. The new open-loop transfer function including the compensator is:

$$G_{oc}(s) = \frac{1+Ts}{1+aTs} \cdot \frac{42.2a}{s^2+5s+6}$$

The steady-state position error in closed-loop can be computed as:

$$\varepsilon_p = \lim_{s\to 0}\big(1 - \mathrm{FTBF}(s)\big) = 1 - \frac{42.2a}{42.2a+1}$$

To obtain a position error of 2%, it is necessary to set:

$$1 - \frac{42.2a}{42.2a+1} = 0.02 \Rightarrow a = 1.16$$

Finally, we choose T so that $1/T$ is much smaller than the gain crossover frequency at 0 dB. For example, we can take $T = 10\,\mathrm{s}$.

We then have:

$$G(s) = 1.16 \cdot \frac{1+10s}{1+11.6s}$$

9.2 Solution to Exercise 2

The open-loop transfer function is:

$$G(s) = \frac{1000K}{s(s+10)^2}$$

- *Calculation of the value of K that ensures a phase margin greater than* 45°*.*

We have:

$$M_\varphi = \pi + \varphi(\omega_{c0}) = \frac{\pi}{4} \Rightarrow \pi - \frac{\pi}{2} - 2\arctan\frac{\omega_{c0}}{10} = \frac{\pi}{4} \Rightarrow \omega_{c0} = 10\tan\frac{\pi}{8} = 4.1421\ \mathrm{rad/s}$$

On the other hand, we have:

$$G(\omega_{c0}) = 1 \Rightarrow \frac{1000K}{\omega_{c0}(\omega_{c0}^2+10^2)} = 1 \Rightarrow K = \frac{\omega_{c0}(\omega_{c0}^2+10^2)}{1000} = 0.4853$$

- *Calculation of the position error and the velocity error.*

The closed-loop transfer function is:

$$H(s) = \frac{G(s)}{1+G(s)} = \frac{1000K}{s(s+10)^2+1000K}$$

The position error in closed loop is:

$$\varepsilon_p = \lim_{s\to 0}\big(1 - H(s)\big) = \lim_{s\to 0}\left(1 - \frac{1000K}{s(s+10)^2+1000K}\right) = 1 - 1 = 0$$

The velocity error in closed loop is:

$$\varepsilon_v = \lim_{s\to 0} \frac{1 - H(s)}{s} = \lim_{s\to 0} \frac{1}{s}\left(\frac{s(s+10)^2}{s(s+10)^2 + 1000K}\right) = \frac{1}{10K} = 0.0485 = 4.85\%$$

To satisfy both specifications, we must choose a phase lag compensator. The new open-loop transfer function with compensator is given by:

$$G_{oc}(s) = a\frac{(1+Ts)}{(1+aTs)}\frac{1000K}{s(s+10)^2}$$

We calculate the velocity error in closed loop as follows:

$$\varepsilon_v = \lim_{s\to 0} \frac{1 - H(s)}{s} = \frac{1}{10Ka}$$

To obtain a velocity error of 2%, we must set:

$$\varepsilon_v = \frac{1}{10Ka} = 0.02 \Rightarrow a = \frac{1}{10 \times 0.02 \times K} = \frac{1}{10 \times 0.02 \times 0.4853} = 10.3$$

Finally, it is sufficient to choose T such that $1/T$ is much smaller than the cutoff frequency at 0 dB. We can take, for example, $T = 10\,\mathrm{s}$.

We finally obtain:

$$G(s) = 10.3\frac{(1+10s)}{1+103s}$$

9.3 Solution to Exercise 3

The transfer function of the system is:

$$G(s) = \frac{1000}{s(s+10)^2}$$

- *Proposal of a compensator given that the open-loop overshoot is* 60%*.*

The overshoot is expressed as:

$$D\% = 100e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} = 60$$

which gives:

$$\xi = \sqrt{\frac{\ln(0.6)^2}{\ln(0.6)^2 + \pi^2}} = 0.1605$$

Thus:

$$M_\varphi^\circ = 100\,\xi = 100 \times 0.1605 = 16^\circ$$

To increase the phase margin, it is necessary to use a lead compensator of the form:

$$C(s) = \frac{1+aTs}{1+Ts}$$

- *Determination of the closed-loop phase margin to be achieved and the corresponding crossover frequency at* 0 dB*.*

The closed-loop settling time is given as $t_m = 0.6\,\mathrm{s}$. Hence, the crossover frequency is:

$$t_m = \frac{3}{\omega_{c0}} \quad \Rightarrow \quad \omega_{c0} = \frac{3}{t_m} = 5\,\mathrm{rad/s}$$

The closed-loop overshoot is 5%, which allows us to compute the new damping ratio:

$$D\% = 100e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} = 5$$

$$\xi = \sqrt{\frac{\ln(0.05)^2}{\ln(0.05)^2 + \pi^2}} = 0.7$$

- Determination of the phase margin of the uncompensated system at this frequency.

We obtain:

$$M_\varphi^\circ = 100\,\xi = 100 \times 0.7 = 70^\circ$$

- Computation of the lead compensator parameters.

To achieve the desired phase margin, the required phase lead is:

$$\varphi_{\max} = 70^\circ - 16^\circ = 54^\circ$$

The relationship between the coefficient a and this maximum phase lead is:

$$\varphi_{\max} = \arcsin\left(\frac{a-1}{a+1}\right) \quad \Rightarrow \quad a = \frac{1+\sin 54^\circ}{1-\sin 54^\circ} = 9.4729$$

The compensator time constant T is then obtained as:

$$\frac{1}{T\sqrt{a}} = \omega_{c0} \quad \Rightarrow \quad T = \frac{1}{\omega_{c0}\sqrt{a}} = \frac{1}{5 \times \sqrt{9.4729}} = 0.065\,\mathrm{s}$$

Finally, the proposed compensator is:

$$C(s) = \frac{1+0.6156s}{1+0.065s}$$

9.4 Solution to Exercise 4

The transfer function of the system is:

$$G(s) = \frac{10}{3s+1}$$

- Calculation of the PI controller parameters

The closed-loop transfer function (CLTF) can be expressed as follows:

$$H(s) = \frac{G(s)C(s)}{1+G(s)C(s)} = \frac{\left(\frac{10}{1+3s}\right)\left(\frac{K_p s + K_i}{s}\right)}{1+\left(\frac{10}{1+3s}\right)\left(\frac{K_p s + K_i}{s}\right)} = \frac{10(K_p s + K_i)}{3s^2 + (10K_p+1)s + 10K_i}$$

$$H(s) = \frac{\frac{10(K_p s + K_i)}{3}}{s^2 + \frac{10K_p+1}{3}s + \frac{10K_i}{3}}$$

According to the specifications, we can deduce:
- Oscillatory closed-loop response ⇒ the characteristic polynomial of a second-order system is:

$$D(s) = s^2 + 2\xi\omega_n s + \omega_n^2$$

The closed-loop overshoot is 5%:

$$D\% = 100e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} = 5 \Rightarrow \xi = \sqrt{\frac{\ln(0.05)^2}{\ln(0.05)^2 + \pi^2}} = 0.7$$

The closed-loop rise time is 0.2 s:

$$t_m = \frac{3}{\omega_n} \Rightarrow \omega_n = \frac{3}{t_m} = 15\,\mathrm{rad/s}$$

Thus, the characteristic polynomial of the second-order system is:

$$D(s) = s^2 + 21s + 225$$

By identification, we obtain:

$$\begin{cases} \dfrac{10K_p + 1}{3} = 21 \\ \dfrac{10K_i}{3} = 225 \end{cases} \Rightarrow \begin{cases} K_p = 6.2 \\ K_i = 67.5 \end{cases}$$

9.5 Solution to Exercise 5

The transfer function of the system is:

$$G(s) = \frac{2}{10s + 1}$$

- *Calculation of the PI controller parameters using the compensation method*
The closed-loop transfer function (CLTF) is:

$$H(s) = \frac{2(K_p s + K_i)}{s(1 + 10s) + (K_p s + K_i)} = \frac{2K_i\big(\frac{K_p}{K_i}s + 1\big)}{s(1 + 10s) + (K_p s + K_i)}$$

Let us set:

$$\frac{K_p}{K_i}s + 1 = 10s + 1$$

Then:

$$H(s) = \frac{2K_i(\frac{K_p}{K_i}s + 1)}{(1 + 10s)\big(s + 2K_i\frac{\frac{K_p}{K_i}s+1}{1+10s}\big)} = \frac{2K_i}{s + 2K_i} = \frac{1}{\frac{1}{2K_i}s + 1}$$

A first-order system is characterized by a static gain K_s and a time constant T_s, as follows:

$$H_d(s) = \frac{K_s}{1 + T_s s}$$

From the previous expression, we get:

$$\frac{K_p}{K_i} = 10$$

By identification, we also obtain:

$$\frac{1}{2K_i} = T_s$$

The response time is:

$$t_r = 3T_s \quad \Rightarrow \quad T_s = \frac{t_r}{3} = \frac{0.3}{3} = 0.1$$

Finally:

$$\begin{cases} K_p = \frac{T}{KT_s} = \frac{10}{2\times0.1} = 50 \\ K_i = \frac{1}{KT_s} = \frac{1}{2\times0.1} = 5 \end{cases}$$

9.6 Solution to Exercise 6

The transfer function of the system is given by:

$$G(s) = \frac{6}{50s+1}$$

- *Calculation of IP controller parameters*

The closed-loop transfer function can be expressed as follows:

$$H(s) = \frac{6K_pK_i}{50s^2 + (1+6K_p)s + 6K_pK_i} = \frac{\frac{6K_pK_i}{50}}{s^2 + \frac{1+6K_p}{50}s + \frac{6K_pK_i}{50}}$$

According to the design specifications, we can deduce:
- Oscillatory closed-loop response $\Rightarrow$ the characteristic polynomial of a second-order system is:

$$D(s) = s^2 + 2\omega_n \xi s + \omega_n^2$$

Closed-loop overshoot equal to 5%:

$$D\% = 100e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} = 5 \Rightarrow \xi = \sqrt{\frac{\ln(0.05)^2}{\ln(0.05)^2 + \pi^2}} = 0.7$$

Closed-loop rise time equal to $0.8\,s$:

$$t_r = \frac{3}{\omega_n} \Rightarrow \omega_n = \frac{3}{t_r} = 3.75\,\text{rad/s}$$

Therefore, the characteristic polynomial of the second-order system can be written as:

$$D(s) = s^2 + 5.25s + 14$$

By identifying with the general form, we obtain:

$$\begin{cases} \frac{1+6K_p}{50} = 5.25 \Rightarrow K_p = \frac{5.25\times50-1}{6} \\ \frac{6K_pK_i}{50} = 14 \Rightarrow K_i = \frac{50\times14}{6K_p} \end{cases} \Rightarrow \begin{cases} K_p = 43.5833 \\ K_i = 0.0535 \end{cases}$$

9.7 Solution to Exercise 7

The transfer function of the system is:

$$G(s) = \frac{25}{(0.1s+1)(0.5s+1)}$$

- *Determination of the transfer function of a PID controller*
The closed-loop transfer function is:

$$H(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$$

Let $H(s) = H_d(s)$. In this case, the controller $C(s)$ can be deduced as:

$$C(s) = \frac{H_d(s)}{G(s)\big(1-H_d(s)\big)}$$

The general form of a first-order system is given by:

$$H_d(s) = \frac{K}{1+Ts}$$

According to the specifications, $t_r = 3T \Rightarrow T = t_r/3 \Rightarrow T = 0.1/3 = 0.0333\,\mathrm{s}$, and perfect steady-state accuracy $\Rightarrow K = 1$. This allows us to write the desired behavior as:

$$H_d(s) = \frac{1}{1+0.0333s}$$

We can then deduce the transfer function:

$$C(s) = \frac{\frac{1}{1+0.0333s}}{\frac{25}{(1+0.1s)(1+0.5s)}\big(1-\frac{1}{1+0.0333s}\big)} = \frac{0.05s^2+0.6s+1}{25s}$$

The transfer function of the PID controller is thus:

$$C(s) = \frac{0.6}{25} + \frac{1}{25s} + \frac{0.05}{25}s$$

Chapter 3

State-Space Representation of Linear Systems

1 Introduction

The transfer function has the advantage of being simple to use; however, this simplicity is lost in the case of multivariable transfer functions. Moreover, initial conditions are not easily taken into account, and only the observable and controllable parts of the system are represented. Nevertheless, frequency-domain representations, which form the basis of these models, provide an irreplaceable perspective on the external behavior of systems.

In this second part, we present the modeling of linear systems through an alternative approach known as the *state-space representation*, which models a dynamic system using *state variables*. This framework makes it possible to determine the system's state at any future instant, provided that its initial state and the behavior of exogenous variables influencing the system are known. The state-space representation thus provides insight into the system's *internal* dynamics, rather than only its *external* behavior, as is the case with the transfer function.

2 State Equations

Many physical processes can be described by differential and algebraic equations. The state-space representation of a linear time-invariant system takes the following general form:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \qquad x(t_0) = x_0 \tag{3.1}$$

where $x(t) = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_n(t) \end{bmatrix}$ is the n-dimensional *state vector*. The components $x_1, x_2, \dots, x_n$ are referred to as the *state variables*. $u(t) = \begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix}$ is the m-dimensional *input or control vector*. The elements $u_1, u_2, \dots, u_m$ are called the *inputs*. $y(t) = \begin{bmatrix} y_1 & y_2 & \dots & y_p \end{bmatrix}$ is the p-dimensional *output or observation vector*. The components $y_1, y_2, \dots, y_p$ are called the *outputs*.

A, B, C, and D are constant matrices. The $n \times n$ matrix A is called the *state matrix* (or system matrix). The $n \times m$ matrix B is called the *input matrix* (or control matrix). The $p \times n$ matrix C is called the *output matrix*, while the $p \times m$ matrix D is called the *feedthrough matrix* (often $D = 0$).

The first-order differential equation $dx(t)/dt = Ax(t) + Bu(t)$ is referred to as the *state equation*. The equation $y(t) = Cx(t) + Du(t)$ is referred to as the *output equation*.

Example 14 ***DC motor with a flexible coupling***: *Figure (3.1) shows a DC motor connected to a flexible coupling. R and L denote the armature resistance and inductance, respectively. k and b represent the spring constant and damping coefficient of the flexible coupling. J_m and J_l denote the moments of inertia of the motor and load, respectively. u is the supply voltage of the DC motor, which serves as the system input, and θ_l is the load angular position, which represents the system output.*

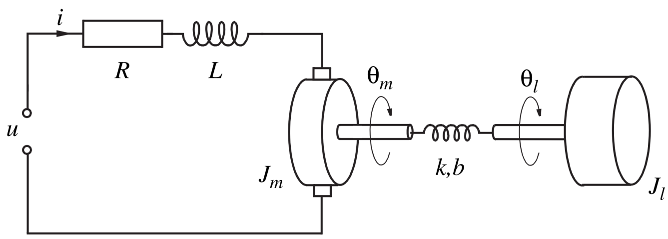


Figure 3.1: DC motor coupled with a flexible shaft

If the two inertias are separated and the appropriate torques (T_k; T_b; T_a) are applied as shown in Figure (3.2), Newton's second law can be written as:

$$J_m\ddot{\theta}_m(t) = K_a i(t) + K(\theta_l(t) - \theta_m(t)) + b(\dot{\theta}_l(t) - \dot{\theta}_m(t)) - b_m\dot{\theta}_m(t), \tag{3.2}$$

$$J_l\ddot{\theta}_l(t) = -k(\dot{\theta}_l(t) - \dot{\theta}_m(t)) - b(\dot{\theta}_l(t) - \dot{\theta}_m(t)) - b_l\dot{\theta}_l(t). \tag{3.3}$$

where K_a is the torque constant, and b_m and b_l denote the viscous friction coefficients of the motor bearings and load shaft, respectively.

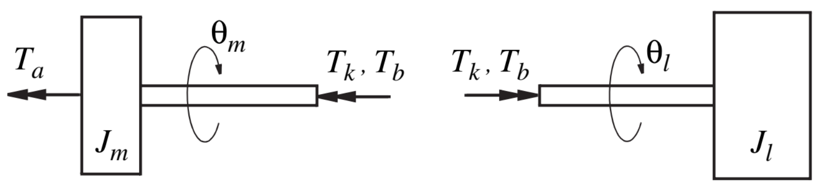


Figure 3.2: Moments of inertia in the system

Applying Ohm's law and Kirchhoff's law to the electrical circuit yields:

$$\frac{di}{dt} = -\frac{R}{L}i(t) - \frac{k_e}{L}\dot{\theta}_m(t) + \frac{1}{L}u(t). \tag{3.4}$$

where k_e is the back EMF constant of the motor's armature windings.

If the state variables are chosen as $x_1 = i$, $x_2 = \theta_m$, $x_3 = \dot{\theta}_m$, $x_4 = \theta_l$, and $x_5 = \dot{\theta}_l$, the following equations can be written:

$$\begin{cases} \dot{x}_1 = \dot{i} = -\frac{R}{L}i(t) - \frac{k_e}{L}\dot{\theta}_m(t) + \frac{1}{L}u \\ \dot{x}_2 = \dot{\theta}_m = x_3 \\ \dot{x}_3 = \ddot{\theta}_m = \frac{K_a}{J_m}i - \frac{k}{J_m}\theta_m - \frac{b+b_m}{J_m}\dot{\theta}_m + \frac{k}{J_m}\theta_l + \frac{b}{J_m}\dot{\theta}_l \\ \dot{x}_4 = \dot{\theta}_l = x_5 \\ \dot{x}_5 = \ddot{\theta}_l = \frac{k}{J_l}\theta_m + \frac{b}{J_l}\dot{\theta}_m - \frac{k}{J_l}\theta_l - \frac{b+b_l}{J_l}\dot{\theta}_l \end{cases} \tag{3.5}$$

Equations 3.5 can be rewritten as:

$$\begin{cases} \dot{x}_1 = -\frac{R}{L}x_1 + 0x_2 - \frac{k_e}{L}x_3 + 0x_4 + 0x_5 + \frac{1}{L}u \\ \dot{x}_2 = 0x_1 + 0x_2 + 1x_3 + 0x_4 + 0x_5 + 0u(t) \\ \dot{x}_3 = \frac{K_a}{J_m}x_1 - \frac{k}{J_m}x_2 - \frac{b+b_m}{J_m}x_3 + \frac{k}{J_m}x_4 + \frac{b}{J_m}x_5 + 0u \\ \dot{x}_4 = 0x_1 + 0x_2 + 1x_3 + 0x_4 + 1x_5 + 0u(t) \\ \dot{x}_5 = 0x_1 + \frac{k}{J_l}x_2 + \frac{b}{J_l}x_3 - \frac{k}{J_l}x_4 - \frac{b+b_l}{J_l}x_5 + 0u \end{cases} \tag{3.6}$$

Equations 3.6 can be expressed in matrix form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 & -\frac{k_e}{L} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{K_a}{J_m} & -\frac{k}{J_m} & -\frac{b+b_m}{J_m} & \frac{k}{J_m} & \frac{b}{J_m} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{k}{J_l} & \frac{b}{J_l} & -\frac{k}{J_l} & -\frac{b+b_l}{J_l} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} -\frac{1}{L} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{3.7}$$

If θ_m and θ_l are chosen as system outputs, the output equation can be written as:

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + 0u \tag{3.8}$$

Example 15 *Consider the electrical circuit shown below, where $V_e(t)$ represents the input voltage and $V_s(t)$ the output voltage. The goal is to derive a state-space representation for this circuit.*

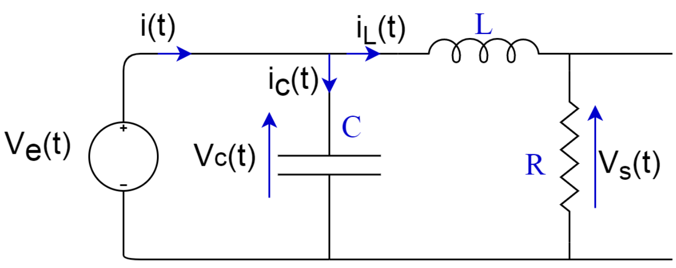


Figure 3.3: Electrical circuit

Applying Ohm's law and Kirchhoff's law yields:

$$V_e(t) = V_c(t) \tag{3.9}$$
$$V_s(t) = Ri_L(t) \tag{3.10}$$
$$V_e(t) = L\frac{di_L}{dt} + Ri_L(t) \tag{3.11}$$
$$V_c(t) = \frac{1}{c}\int i_c dt \tag{3.12}$$
$$i(t) = i_c(t) + i_L(t) \tag{3.13}$$

From equation 3.9, we obtain:

$$\frac{di_L}{dt} = -\frac{R}{L}i_L(t) + \frac{1}{L}V_e(t) \tag{3.14}$$

From equations 3.11 and 3.12, we can write:

$$\frac{dV_c}{dt} = \frac{1}{c}i_c = \frac{1}{c}(i(t) - i_L(t)) = -\frac{1}{c}i_L(t) + \frac{1}{c}i(t) \tag{3.15}$$

If we choose the state variables as $x_1 = i_L$, $x_2 = V_c$, and consider V_e as the known system input, $i(t)$ as an unknown input, and V_s as the system output, then the state-space model is given by:

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 \\ -\frac{1}{c} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_e + + \begin{bmatrix} 0 \\ \frac{1}{c} \end{bmatrix} i \\ y = \begin{bmatrix} R & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases} \tag{3.16}$$

The Matlab code [Script 8] can be used to declare the state-space representation of the system presented in exemple 15:

```
% ==============================
% Script : example15.m
% Purpose: State-space modeling
% of an RLC electrical circuit
% ================================
clc; clear all; close all;
% --- System parameters
%(to be defined before execution)
% R = Resistance (Ohms)
% L = Inductance (Henrys)
% C = Capacitance (Farads)
% --- State-space matrices ---
% State vector: x = [i_L ; v_C]
% where i_L = current through the inductor,
% v_C = capacitor voltage
A = [-R/L,    0;
      1/C,    0];
% System dynamics matrix
B = [-R/L;
       0];
```

```
% Input matrix (input: applied voltage)
C = [R  0];
% Output matrix (output: voltage across the resistor)
D = 0;                  % Direct transmission term
% --- Build state-space model ---
Sys = ss(A, B, C, D);    % Continuous-time state-space system
```

3 Deriving the State-Space Representation

In this subsection, we present methods for obtaining the state-space representation of a linear system from its transfer function.

Consider a system with transfer function $G(s)$ defined as:

$$G(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_1 s + a_0} \tag{3.17}$$

3.1 Diagonal or Quasi-Diagonal State-Space Representation

Case of Distinct Poles

Suppose the transfer function of the system takes the following form:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\alpha_1}{s-\lambda_1} + \frac{\alpha_2}{s-\lambda_2} + \ldots + \frac{\alpha_n}{s-\lambda_n} = \sum_{i=1}^{n} \frac{\alpha_i}{s-\lambda_i} \tag{3.18}$$

Define:

$$X_i(s) = \frac{\alpha_i}{s-\lambda_i} U(s) \tag{3.19}$$

Thus, the output $Y(s)$ can be written as:

$$Y(s) = \sum_{i=1}^{n} \frac{\alpha_i}{s-\lambda_i} U(s) = \sum_{i=1}^{n} X_i(s) \tag{3.20}$$

From equation 3.19, we obtain:

$$sX_i(s) = \alpha_i X_i(s) + \lambda_i U(s) \tag{3.21}$$

Applying the inverse Laplace transform to equation 3.21, we find:

$$\dot{x}_i(t) = \alpha_i x_i(t) + \lambda_i u(t) \tag{3.22}$$

From equation 3.22, we can write:

$$\begin{cases} i = 1 : \dot{x}_1(t) = \alpha_1 x_1(t) + \lambda_1 u(t) \\ i = 2 : \dot{x}_2(t) = \alpha_2 x_2(t) + \lambda_2 u(t) \\ \vdots \\ i = n : \dot{x}_n(t) = \alpha_n x_n(t) + \lambda_n u(t) \end{cases} \tag{3.23}$$

Equation 3.23 can be transformed into the following matrix form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{3.24}$$

Applying the inverse Laplace transform to equation 3.20, we obtain:

$$y(t) = x_1(t) + x_2(t) + \dots + x_n(t) \tag{3.25}$$

Thus, we can write the following matrix form:

$$y(t) = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{3.26}$$

Finally, the state-space representation of a system described by a transfer function of the form 3.18 can be given by:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} x(t) + \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} x(t) \end{cases} \tag{3.27}$$

Example 16 *Consider a system with the following transfer function:*

$$G(s) = \frac{5}{s+1} + \frac{-3}{s-2} + \frac{4}{s-7} \tag{3.28}$$

We can directly deduce the state-space representation as follows:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 7 \end{bmatrix} x(t) + \begin{bmatrix} 5 \\ -3 \\ 4 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} x(t) \end{cases} \tag{3.29}$$

Case of Multiple Poles

Suppose the system transfer function takes the following form:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\alpha_1}{(s-\lambda)^1} + \frac{\alpha_2}{(s-\lambda)^2} + \dots + \frac{\alpha_n}{(s-\lambda)^n} = \sum_{i=1}^{n} \frac{\alpha_i}{(s-\lambda)^i} \tag{3.30}$$

The output $Y(s)$ can be written as:

$$Y(s) = \sum_{i=1}^{n} \alpha_i \frac{U(s)}{(s-\lambda)^i} \tag{3.31}$$

Define:

$$X_{n+1-i}(s) = \frac{U(s)}{(s-\lambda)^i} \tag{3.32}$$

From equation 3.32, we deduce:

$$\begin{cases} i=1: X_n(s) = \frac{U(s)}{(s-\lambda)} \Rightarrow sX_n = \lambda X_n + U(s) \\ i=2: X_{n-1}(s) = \frac{U(s)}{(s-\lambda)^2} = \frac{X_n}{(s-\lambda)} \Rightarrow sX_{n-1} = X_n + \lambda X_{n-1} \\ i=3: X_{n-2}(s) = \frac{X_{n-1}}{(s-\lambda)} \Rightarrow sX_{n-2} = X_{n-1} + \lambda X_{n-2} \\ \vdots \\ i=n: X_1(s) = \frac{X_2}{(s-\lambda)} \Rightarrow sX_1 = X_2 + \lambda X_1 \end{cases} \tag{3.33}$$

Applying the inverse Laplace transform to equations 3.33, we obtain:

$$\begin{cases} i=1: \dot{x}_n(t) = \lambda x_n(t) + u(t) \\ i=2: \dot{x}_{n-1}(t) = x_n(t) + \lambda x_{n-1}(t) \\ i=3: \dot{x}_{n-2}(t) = x_{n-1}(t) + \lambda x_{n-2}(t) \\ \vdots \\ i=n: \dot{x}_1(t) = x_2(t) + \lambda x_1(t) \end{cases} \tag{3.34}$$

From equation 3.34, we can write the following matrix form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \lambda & 1 & \dots & 0 \\ 0 & \lambda & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 \\ 0 & 0 & \dots & \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(t) \tag{3.35}$$

Using equation 3.32, equation 3.31 can be rewritten as:

$$Y(s) = \sum_{i=1}^{n} \alpha_i X_{n+1-i} = \alpha_1 X_n + \alpha_2 X_{n-1} + \ldots + \alpha_n X_1 \tag{3.36}$$

Applying the inverse Laplace transform to equation 3.36 gives:

$$y(t) = \alpha_1 x_n(t) + \alpha_2 x_{n-1}(t) + \ldots + \alpha_n x_1(t) = \begin{bmatrix} \alpha_n & \alpha_{n-1} & \ldots & \alpha_1 \end{bmatrix} x(t) \tag{3.37}$$

Finally, the state-space representation of a system described by a transfer function of the form 3.30 can be given by:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} \lambda & 1 & \dots & 0 \\ 0 & \lambda & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 \\ 0 & 0 & \dots & \lambda \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} \alpha_n & \alpha_{n-1} & \ldots & \alpha_1 \end{bmatrix} x(t) \end{cases} \tag{3.38}$$

Example 17 *Consider a system with the following transfer function:*

$$G(s) = \frac{1}{s+3} + \frac{-3}{(s+3)^2} + \frac{2}{(s+3)^3} \tag{3.39}$$

We can directly deduce the state-space representation as follows:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 2 & -3 & 1 \end{bmatrix} x(t) \end{cases} \tag{3.40}$$

3.2 Canonical State-Space Representation (Companion Form)

There exist several companion-form state-space representations that can be easily obtained from the transfer function. This subsection is restricted to the most common companion forms. The two companion forms most frequently encountered are: - the horizontal companion form (controller canonical form), and - the vertical companion form (observer canonical form).

Horizontal Companion Form

Consider a system with transfer function $G(s)$ defined as:

$$G(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_1 s + a_0} \tag{3.41}$$

Define:

$$\begin{cases} x_1(t) = y(t) \\ x_2(t) = \dot{y}(t) \\ \vdots \\ x_n(t) = \dfrac{d^{n-1}y}{dt^{n-1}} \end{cases} \tag{3.42}$$

We then obtain the following horizontal companion form:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 & \ldots & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & \ldots & -a_{n-1} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} b_0 & b_1 & \ldots & b_n & 0\ldots 0 \end{bmatrix} x(t) \end{cases} \tag{3.43}$$

Example 18 *Consider a system with the following transfer function:*

$$G(s) = \frac{3s^2 + 5s + 2}{s^3 + 7s^2 + 6s + 2} \tag{3.44}$$

Using equation 3.43, the state-space representation in the horizontal companion form can be derived as follows:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -6 & -7 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 2 & 5 & 3 \end{bmatrix} x(t) \end{cases} \tag{3.45}$$

Vertical Companion Form

Consider a system with transfer function $G(s)$ given by:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + b_{m-2} s^{m-2} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \ldots + a_1 s + a_0} \tag{3.46}$$

Define:

$$\begin{cases} x_1(t) = y(t) \\ x_2(t) = \dfrac{d^{n-1}y}{dt^{n-1}} + \dot{y}(t) \\ \vdots \\ x_{n-1}(t) = a_2 y + a_3 \dot{y} + \ldots + \dfrac{d^{n-2}y}{dt^{n-2}} - \left(b_2 u + \ldots + b_m \dfrac{d^{m-2}u}{dt^{m-2}}\right) \\ x_n(t) = a_1 y + a_2 \dot{y} + \ldots + \dfrac{d^{n-1}y}{dt^{n-1}} - \left(b_1 u + \ldots + b_m \dfrac{d^{m-1}u}{dt^{m-1}}\right) \end{cases} \tag{3.47}$$

We then obtain the following vertical companion form:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} a_{n-1} & 1 & \ldots & 0 \\ a_{n-2} & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_1 & 0 & \ldots & 1 \\ -a_0 & 0 & \ldots & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_m \\ \vdots \\ b_0 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} x(t) \end{cases} \tag{3.48}$$

Example 19 *Consider a system with the following transfer function:*

$$G(s) = \frac{3s^2 + 5s + 2}{s^3 + 7s^2 + 6s + 2} \tag{3.49}$$

Using equation 3.48, the state-space representation in the vertical companion form can be derived as follows:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -7 & 1 & 0 \\ -6 & 0 & 1 \\ -2 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 3 \\ 5 \\ 2 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) \end{cases} \tag{3.50}$$

4 From the Transfer Function to the State-Space Representation

Consider a system described by the following state-space model:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{3.51}$$

Applying the Laplace transform to system 3.51, we obtain:

$$\begin{cases} sX(s) = AX(s) + BU(s) \\ Y(s) = CX(s) + DU(s) \end{cases} \tag{3.52}$$

From the first equation of 3.52, we can write:

$$X(s) = (sI - A)^{-1}BU(s) \tag{3.53}$$

Substituting $X(s)$ into the second equation of 3.52, we obtain:

$$Y(s) = \left(C(sI - A)^{-1} + D\right) U(s) \tag{3.54}$$

Example 20 *Consider a system described by the following state-space representation:*

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -6 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t) \end{cases} \tag{3.55}$$

We first compute the matrix $(sI - A)^{-1}$:

$$(sI - A) = \begin{bmatrix} s & -1 \\ 6 & s+2 \end{bmatrix} \Rightarrow (sI - A)^{-1} = \frac{1}{(s^2 + 2s + 6)} \begin{bmatrix} s+2 & 1 \\ -6 & s \end{bmatrix} \tag{3.56}$$

Thus, the transfer function can be computed as:

$$Y(s) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{s+2}{(s^2+2s+6)} & \frac{1}{(s^2+2s+6)} \\ \frac{-6}{(s^2+2s+6)} & \frac{s}{(s^2+2s+6)} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{2s - 3}{s^2 + 2s + 6} \tag{3.57}$$

The Matlab code [Script 9] can be used to compute the transfer function of the system presented in Example 20:

```
% Script pour l'exemple 19
clc, clear all, close all;

A=[0,1;-6,-2]
B=[1;1]
C=[1 1]
D=0
[N,D]=ss2tf(A,B,C,D)
Fonction_de_transfert=tf(N,D)
```

5 Exercises

5.1 Exercise 1

Consider a direct current (DC) motor governed by the following electrical and mechanical equations:

$$\begin{cases} u(t) = Ri(t) + L\frac{di(t)}{dt} + e(t) \\ J\frac{d\omega(t)}{dt} = C_m(t) - f\omega(t) - r\theta(t) - C_r(t) \\ e(t) = K\omega(t) \\ C_m(t) = Ki(t) \end{cases}$$

Here, C_m is the electromagnetic torque, and C_r is the load torque (disturbance). The parameter f is the viscous friction coefficient, r is the restoring constant, and J is the moment of inertia of the rotor shaft. The constant K represents both the torque and speed constants. $u(t)$ is the input voltage applied to the motor, $e(t)$ is the back electromotive force (emf), $i(t)$ is the armature current, $\omega(t)$ is the angular velocity of the rotor, and $\theta(t)$ is the angular position of the rotor.

The observed output y is the angular position, while C_r is considered an additional input.

Derive the state-space equations of the system.

5.2 Exercise 2

Consider an electromechanical system represented by the block diagram below.

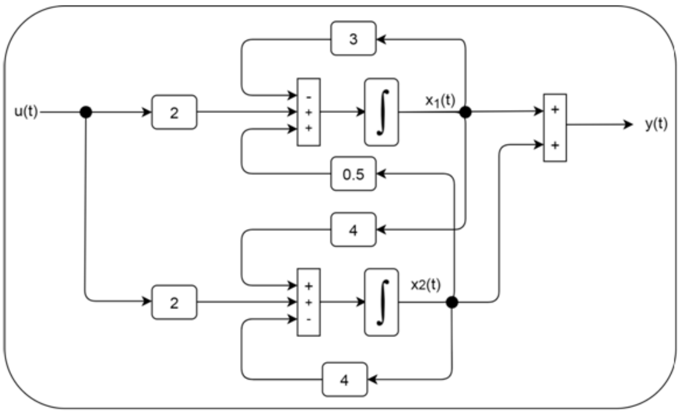


Figure 3.4: Functional diagram of an electromechanical system

Determine the state-space model of the system.

5.3 Exercise 3

The dynamics of a helicopter can be described by the following differential equations:

$$\begin{cases} \frac{d^2\rho(t)}{dt^2} = -0.65\frac{d\rho(t)}{dt} - 0.02\frac{dx(t)}{dt} + 5.4\alpha(t) \\ \frac{d^2x(t)}{dt^2} = -1.57\frac{d\rho(t)}{dt} - 0.03\frac{dx(t)}{dt} + 9.8(\rho(t) + \alpha(t)) \end{cases}$$

Here, ρ is the pitch angle, which can be controlled by the input angle α, and x represents the horizontal position. The observed outputs are the pitch angle and the horizontal position.

Determine a state-space representation for this system.

5.4 Exercise 4

Consider a system with the following transfer function $G(s)$ defined as:

$$G(s) = \frac{2s+1}{s^3+7s^2+5s+6}$$

Provide the state-space representations of the system in both the horizontal companion form and the vertical companion form.

6 Solutions

6.1 Solution to Exercise 1

The system equations are given by:

$$\begin{cases} u(t) = Ri(t) + L\frac{di(t)}{dt} + e(t) \\ J\frac{d\omega(t)}{dt} = C_m(t) - f\omega(t) - r\theta(t) - C_r(t) \\ e(t) = K\omega(t) \\ C_m(t) = Ki(t) \end{cases}$$

From these equations, we can write:

$$\begin{cases} \frac{di(t)}{dt} = -\frac{R}{L}i(t) - \frac{K}{L}\omega(t) + \frac{1}{L}u(t) \\ \frac{d\omega(t)}{dt} = \frac{K}{J}i(t) - \frac{f}{J}\omega(t) - \frac{r}{J}\theta(t) - \frac{1}{J}C_r(t) \\ \frac{d\theta(t)}{dt} = \omega(t) \end{cases}$$

If we choose $x = \begin{bmatrix} i & \omega & \theta \end{bmatrix}^T$, then we obtain the following state-space representation:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -\frac{R}{L} & -\frac{K}{L} & 0 \\ \frac{K}{J} & -\frac{f}{J} & -\frac{r}{J} \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix} C_r(t) \\ y(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(t) \end{cases}$$

6.2 Solution to Exercise 2

From the block diagram in (3.4), we obtain the following representation:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -3 & 0.5 \\ 4 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t) \end{cases}$$

6.3 Solution to Exercise 3

The dynamics of the helicopter are described by the following differential equations:

$$\begin{cases} \frac{d^2\rho(t)}{dt^2} = -0.65\frac{d\rho(t)}{dt} - 0.02\frac{dx(t)}{dt} + 5.4\alpha(t) \\ \frac{d^2x(t)}{dt^2} = -1.57\frac{d\rho(t)}{dt} - 0.03\frac{dx(t)}{dt} + 9.8(\rho(t) + \alpha(t)) \end{cases}$$

Let us define:

$$\begin{cases} x_1 = \dot{\rho} \\ x_2 = \rho \\ x_3 = \dot{x} \\ x_4 = x \end{cases} \Rightarrow \begin{cases} \dot{x}_1 = -0.65x_1 - 0.02x_3 + 5.4\alpha(t) \\ \dot{x}_2 = x_1 \\ \dot{x}_3 = 1.57x_1 + 9.8x_2 - 0.03x_3 + 9.8\alpha(t) \\ \dot{x}_4 = x_3 \end{cases}$$

Thus, the state-space representation is:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -0.65 & 0 & -0.02 & 0 \\ 1 & 0 & 0 & 0 \\ 1.57 & 9.8 & -0.03 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 5.4 \\ 0 \\ 9.8 \\ 0 \end{bmatrix} \alpha(t) \\ y(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(t) \end{cases}$$

6.4 Solution to Exercise 4

The transfer function is defined as:

$$G(s) = \frac{2s + 1}{s^3 + 7s^2 + 5s + 6}$$

From equations 3.43, we can deduce the following horizontal companion form:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -5 & -7 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 2 & 0 \end{bmatrix} x(t) \end{cases}$$

From equations 3.48, we can deduce the following vertical companion form:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -7 & 1 & 0 \\ -5 & 0 & 1 \\ -6 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) \end{cases}$$

Chapter 4

Response of a State-Space Model

1 Introduction

The response of a system consists in determining its output $y(t)$ produced under the effect of an input $u(t)$ (for example, an impulse, a step, or a sinusoidal signal), while using the state-space model. The solution to this problem involves solving the state equations in order to determine the expression of the state $x(t)$, followed by the expression of the output $y(t)$.

2 Solution of the State Equation

Consider the complete state equation given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.1}$$

The solution of such a differential equation is well known and can be written as:

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \tag{4.2}$$

In this formulation, e^{At} represents a matrix exponential, usually denoted as $\Phi(t)$, and referred to as the *state transition matrix* of the system.

3 Computation of the State Transition Matrix

The resolution of the state equations relies on the computation of the state transition matrix $\Phi(t)$. Several methods exist for this purpose. In this section, we present a method based on the use of the Laplace transform.

Consider again the state equation:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.3}$$

Applying the Laplace transform to equation 4.3, we obtain:

$$sX(s) - x(0) = AX(s) + BU(s) \quad \Rightarrow \quad X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s) \tag{4.4}$$

By comparison with equation 4.2, it follows that:

$$\Phi(t) = e^{At} = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right] \tag{4.5}$$

where $\mathcal{L}^{-1}$ denotes the inverse Laplace transform, and I is the identity matrix of dimension n.

Example 21 *Consider a system described by a state-space representation with the following state matrix:*

$$A = \begin{bmatrix} 0 & 1 \\ -6 & -2 \end{bmatrix} \tag{4.6}$$

First, we compute the matrix $(sI - A)^{-1}$:

$$(sI - A) = \begin{bmatrix} s & -1 \\ 6 & s+2 \end{bmatrix} \quad \Rightarrow \quad (sI - A)^{-1} = \frac{1}{s^2 + 2s + 6}\begin{bmatrix} s+2 & 1 \\ -6 & s \end{bmatrix} \tag{4.7}$$

Now, we calculate the inverse Laplace transform of $(sI - A)^{-1}$:

$$e^{At} = \mathcal{L}^{-1}\begin{bmatrix} \dfrac{1}{s+3} & \dfrac{1}{(s+3)(s+2)} \\ \dfrac{-6}{(s+3)(s+2)} & \dfrac{s}{(s+3)(s+2)} \end{bmatrix} = \mathcal{L}^{-1}\begin{bmatrix} \dfrac{1}{s+3} & \dfrac{1}{s+2} - \dfrac{1}{s+3} \\ \dfrac{6}{s+3} - \dfrac{6}{s+2} & \dfrac{3}{s+3} - \dfrac{2}{s+2} \end{bmatrix} \tag{4.8}$$

After computing the inverse Laplace transform, the state transition matrix is obtained as:

$$\Phi(t) = e^{At} = \begin{bmatrix} e^{-3t} & e^{-2t} - e^{-3t} \\ 6e^{-3t} - 6e^{-2t} & 3e^{-3t} - 2e^{-2t} \end{bmatrix} \tag{4.9}$$

4 System Response

Knowing the state transition matrix makes it possible to compute the system response directly for a given input. Substituting the solution (4.2) into the output equation yields:

$$y(t) = Ce^{At}x(0) + C\left(\int_0^t e^{A(t-\tau)}B\,u(\tau)\,d\tau\right) + D\,u(t). \tag{4.10}$$

Example 22 *Consider a system described by the following state-space model:*

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -6 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t), \end{cases} \qquad x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4.11}$$

We seek the response of the system to a unit-step input.

The state transition matrix is:

$$e^{At} = \begin{bmatrix} e^{-3t} & e^{-2t} - e^{-3t} \\ 6e^{-3t} - 6e^{-2t} & 3e^{-3t} - 2e^{-2t} \end{bmatrix}. \tag{4.12}$$

Hence, the system response is

$$y(t) = \underbrace{\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix} e^{-3t} & e^{-2t} - e^{-3t} \\ 6e^{-3t} - 6e^{-2t} & 3e^{-3t} - 2e^{-2t}\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}}_{\text{homogeneous response}}$$

$$+ \underbrace{\begin{bmatrix}1 & 0\end{bmatrix}\int_0^t \begin{bmatrix} e^{-3(t-\tau)} & e^{-2(t-\tau)} - e^{-3(t-\tau)} \\ 6e^{-3(t-\tau)} - 6e^{-2(t-\tau)} & 3e^{-3(t-\tau)} - 2e^{-2(t-\tau)}\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} d\tau}_{\text{forced response}}$$

$$= e^{-2t} + \tfrac{1}{3}\left(1 - e^{-3t}\right).$$

The Matlab code [Script 10] can be used to compute the step response $y(t)$ for Example 22:

```
%% Script : Exemple 21
clc, clear all, close all;
A=[0,1;-6,-2]
B=[0;1]
C=[1 0]
D=0
Sys=ss(A,B,C,D);
[y,t]=step(Sys);
plot(t,y)
xlabel('Temps (s)')
ylabel('y(t)')
```

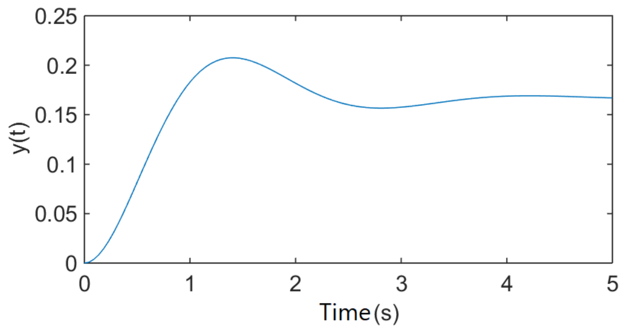


Figure 4.1: Step response $y(t)$ for Example 22

5 Exercises

5.1 Exercise 1

Consider a system described by the state-space model:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix}-1 & -2 \\ 0 & -4\end{bmatrix} x(t) + \begin{bmatrix}1\\0\end{bmatrix} u(t), \\ y(t) = \begin{bmatrix}1 & 0\end{bmatrix} x(t). \end{cases}$$

- *What can be said about the stability of the system?*
- *Determine the explicit expression of the state transition matrix for this system.*

5.2 Exercise 2

Consider a system described by the state-space model:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -1 & 0 \\ -1 & -3 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 0.5 & 0 \end{bmatrix} x(t). \end{cases}$$

Subjected to a unit-step input, *determine the expression of $y(t)$ for all t.*

5.3 Exercise 3

Determine the transfer function of the system defined by the following state-space model:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -2 & -4 \\ -2 & -9 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \end{cases}$$

- *Compute the poles of the transfer function and deduce the step response $y(t)$.*
- *Verify this result by directly solving the state equations, assuming the initial state vector is zero.*

6 Solutions

6.1 Solution to Exercise 1

The system is described by the following state-space representation:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -1 & -2 \\ 0 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \end{cases}$$

1. Stability analysis. To assess the stability of the system, we compute the poles (eigenvalues of the matrix A):

$$\det(sI - A) = 0 \;\Rightarrow\; \det\left(s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & -2 \\ 0 & -4 \end{bmatrix} \right) = \det\left(\begin{bmatrix} s+1 & 2 \\ 0 & s+4 \end{bmatrix} \right) = 0.$$

Thus, the characteristic polynomial is:

$$s^2 + 5s + 4 = 0,$$

whose roots are:

$$\begin{cases} s_1 = -1, \\ s_2 = -4. \end{cases}$$

Since both poles are negative, the system is stable.

2. Computation of the state transition matrix. We first compute $(sI - A)^{-1}$:

$$(sI - A) = \begin{bmatrix} s+1 & 2 \\ 0 & s+4 \end{bmatrix} \Rightarrow (sI - A)^{-1} = \frac{1}{(s+1)(s+4)} \begin{bmatrix} s+4 & -2 \\ 0 & s+1 \end{bmatrix}.$$

Hence, the state transition matrix is:

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} = \mathcal{L}^{-1}\left(\begin{bmatrix} \frac{1}{s+1} & \frac{-2}{(s+1)(s+4)} \\ 0 & \frac{1}{s+4} \end{bmatrix}\right).$$

After computing the inverse Laplace transform, we obtain:

$$e^{At} = \begin{bmatrix} e^{-t} & \frac{2}{3}e^{-t} - \frac{2}{3}e^{-4t} \\ 0 & e^{-4t} \end{bmatrix}.$$

```
clear; clc;

A = [-1 0; -1 -3];
B = [1; 1];
C = [0.5 0];
D = 0;
sys = ss(A,B,C,D);

% Analytical closed-form
t = linspace(0,10,501);
y_anal = 0.5*(1 - exp(-t));

% Simulation with Control System Toolbox
[y_step, t_step] = step(sys, t(end));
% step() may choose its own grid; re-simulate precisely on t:
u = ones(size(t));
y_lsim = lsim(sys, u, t);

% Plots
figure;
plot(t, y_anal, 'LineWidth',1.8); hold on;
plot(t, y_lsim, '--', 'LineWidth',1.4);
grid on; legend('Analytical','lsim ...
    (numeric)','Location','southeast');
xlabel('Time (s)'); ylabel('y(t)');


% Numeric check
err = max(abs(y_anal - y_lsim(:)'));
fprintf('Max |y_{anal} - y_{lsim}| = %.3e\n', err);
```

6.2 Solution to Exercise 2

The system is governed by:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -1 & 0 \\ -1 & -3 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 0.5 & 0 \end{bmatrix} x(t), \qquad x(0) = 0. \end{cases}$$

The system output is given by:

$$y(t) = Ce^{At}x(0) + C \int_0^t e^{A(t-\tau)} B\, u(\tau)\, d\tau + Du(t).$$

We first compute $(sI - A)^{-1}$:

$$(sI - A) = \begin{bmatrix} s+1 & 0 \\ 1 & s+3 \end{bmatrix} \Rightarrow (sI - A)^{-1} = \frac{1}{(s+1)(s+3)} \begin{bmatrix} s+3 & 0 \\ -1 & s+1 \end{bmatrix}.$$

Thus, the state transition matrix is:

$$e^{At} = \mathcal{L}^{-1}\left(\begin{bmatrix} \dfrac{1}{s+1} & 0 \\ \dfrac{-1}{(s+1)(s+3)} & \dfrac{1}{s+3} \end{bmatrix}\right).$$

Computing the inverse Laplace transform yields:

$$e^{At} = \begin{bmatrix} e^{-t} & 0 \\ \dfrac{1}{2}e^{-t} - \dfrac{1}{2}e^{-3t} & e^{-3t} \end{bmatrix}.$$

The convolution integral evaluates to:

$$\int_0^t e^{A(t-\tau)} B\, d\tau = \begin{bmatrix} 1 - e^{-t} \\ \dfrac{1}{2}(1 - e^{-t}) + \dfrac{1}{6}(1 - e^{-3t}) \end{bmatrix}.$$

Finally, the system output is:

$$y(t) = \begin{bmatrix} 0.5 & 0 \end{bmatrix} \begin{bmatrix} 1 - e^{-t} \\ \dfrac{1}{2}(1 - e^{-t}) + \dfrac{1}{6}(1 - e^{-3t}) \end{bmatrix} = \tfrac{1}{2}(1 - e^{-t}).$$

```
clear; clc;
A = [-1 0; -1 -3];B = [1; 1];C = [0.5 0];D = 0;
sys = ss(A,B,C,D);
% Analytical closed-form
t = linspace(0,10,501);
y_anal = 0.5*(1 - exp(-t));
% Simulation with Control System Toolbox
[y_step, t_step] = step(sys, t(end));
% step() may choose its own grid; re-simulate precisely on t:
u = ones(size(t));
```

```
y_lsim = lsim(sys, u, t);
% Plots
figure;
plot(t, y_anal, 'LineWidth',1.8); hold on;
plot(t, y_lsim, '--', 'LineWidth',1.4);
grid on; legend('Analytical','lsim ...
    (numeric)','Location','southeast');
xlabel('Time (s)'); ylabel('y(t)');
% Numeric check
err = max(abs(y_anal - y_lsim(:)'));
fprintf('Max |y_{anal} - y_{lsim}| = %.3e\n', err);
```

6.3 Solution to Exercise 3

The system is defined by the following state-space model:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -2 & -4 \\ -2 & -9 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \end{cases}$$

1. *Transfer function of the system.*

$$G(s) = C(sI - A)^{-1}B.$$

We compute:

$$(sI - A) = \begin{bmatrix} s+2 & 4 \\ 2 & s+9 \end{bmatrix} \quad \Rightarrow \quad (sI - A)^{-1} = \frac{1}{s^2 + 11s + 10} \begin{bmatrix} s+9 & -4 \\ -2 & s+2 \end{bmatrix}.$$

Hence:

$$G(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{s+9}{s^2+11s+10} & \frac{-4}{s^2+11s+10} \\ \frac{-2}{s^2+11s+10} & \frac{s+2}{s^2+11s+10} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{-4}{s^2+11s+10}.$$

2. *Poles of the transfer function.*

$$s^2 + 11s + 10 = 0 \quad \Rightarrow \quad s_1 = -1, \quad s_2 = -10.$$

3. *Step response.* For a unit-step input:

$$Y(s) = G(s)\frac{1}{s} = \frac{-4}{s(s+1)(s+10)}.$$

Partial fraction expansion gives:

$$Y(s) = -\frac{4}{10} \cdot \frac{1}{s} + \frac{4}{9} \cdot \frac{1}{s+1} - \frac{4}{90} \cdot \frac{1}{s+10}.$$

Taking the inverse Laplace transform:

$$y(t) = -\tfrac{4}{10} + \tfrac{4}{9}e^{-t} - \tfrac{4}{90}e^{-10t}.$$

4. Verification using the state equations. The same result is obtained by solving the state equations with $x(0) = 0$ and computing:

$$y(t) = Ce^{At} \int_0^t e^{A(t-\tau)} Bu(\tau)\, d\tau.$$

After performing the calculation, we recover:

$$y(t) = -\tfrac{4}{10} + \tfrac{4}{9}e^{-t} - \tfrac{4}{90}e^{-10t}.$$

```
clear; clc;
A = [-2 -4; -2 -9];B = [0; 1];C = [1 0];D = 0;
% State-space and transfer function
sys = ss(A,B,C,D);
G   = tf(sys);             % should be -4 / (s^2 + 11 s + 10)
disp('G(s) ='); G
% Poles
p = pole(G);
disp('Poles of G(s):'), disp(p.')
% Unit step response :  analytical vs numeric
% Analytical: y(t) = -4/10 + 4/9 e^{-t} - 4/90 e^{-10 t}
t = linspace(0,10,1001);
y_anal = -4/10 + (4/9)*exp(-t) - (4/90)*exp(-10*t);
% Numeric (lsim or step)
u = ones(size(t));
y_lsim = lsim(sys,u,t);
% Plot comparison
figure;
plot(t, y_anal, 'LineWidth',1.8); hold on;
plot(t, y_lsim, '--', 'LineWidth',1.4);
grid on; legend('Analytical','lsim ...
    (numeric)','Location','southeast');
xlabel('Time (s)'); ylabel('y(t)');
% Error check
err = max(abs(y_anal - y_lsim(:)'));
fprintf('Max |y_{anal} - y_{lsim}| = %.3e\n', err);
% Extra: verify G(s) expression explicitly
s = tf('s');
G_expected = -4 / (s^2 + 11*s + 10);
disp('G_expected(s) ='); G_expected
fprintf('Are G and G_expected equal?  %d (1=yes)\n', ...
    isequal(zpk(G), zpk(G_expected)));
```

Chapter 5

State-Feedback and Output-Feedback Control

1 State-Feedback Control

State-feedback control aims to modify the closed-loop behavior of a system described by its state-space representation in such a way that the closed-loop poles are placed at desired locations. Since the poles determine the system dynamics, their placement allows the designer to achieve specific performance objectives.

In this section, we first introduce the concept of controllability, and then we present the structure of state-feedback control along with the algorithm that enables the computation of the feedback gain vector through the so-called *pole placement* method.

1.1 Controllability of a System

The control problem consists of steering a system from an observed initial state to a desired final state. In state-space representation, this amounts to determining the control signal $u(t)$ over a given interval $[t_1, t_2]$ such that the system evolves from $x(t_1)$ to the desired final state $x(t_2)$.

Controllability to the Origin

A system is said to be **controllable** at time t_1 if it is possible to determine an input signal $u(t)$ defined over $[t_1, t_2]$ that drives the system from $x(t_1) = x_1$ to $x(t_2) = 0$. If a system is controllable for any initial instant t_1, it is said to be *completely controllable*.

Reachability

A system is said to be reachable to a state x_2 if it is possible to determine an input signal $u(t)$ over the interval $[t_1, t_2]$ that drives the system from $x(t_1) = x_1$ to $x(t_2) = x_2$.

Complete Controllability

A system is said to be *completely controllable* if, for any time interval $[t_1, t_2]$ and for any states x_1 and x_2, there exists a control signal $u(t)$ defined on $[t_1, t_2]$ that drives the system from $x(t_1) = x_1$ to $x(t_2) = x_2$.

1.2 Kalman's Criterion

Several criteria exist to determine whether a system is controllable or uncontrollable. One of the most widely used is Kalman's criterion.

A system is completely reachable and completely controllable if and only if the controllability matrix defined in (5.1) is full rank, i.e., if its determinant is non-zero.

$$Q_c = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{5.1}$$

1.3 Pole Placement Control

Consider a system described by the following state-space representation:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t). \end{cases} \tag{5.2}$$

State-feedback control is one of the most classical approaches in controlling a system modeled in the state-space framework. It assumes that all components x_i of the state vector x are measurable. A possible control law is:

$$u(t) = Hy_c(t) + Kx(t) \tag{5.3}$$

where K is a row vector of n components, commonly referred to as the *state-feedback gain vector*, H is a scalar known as the *feedforward gain* or *precompensator*, and y_c denotes the reference input of the closed-loop system.

Careful inspection of (5.3) reveals that this type of control law does not correspond to the classical feedback configuration commonly encountered in frequency-domain design, but instead to a new control scheme, as illustrated in Figure 5.1.

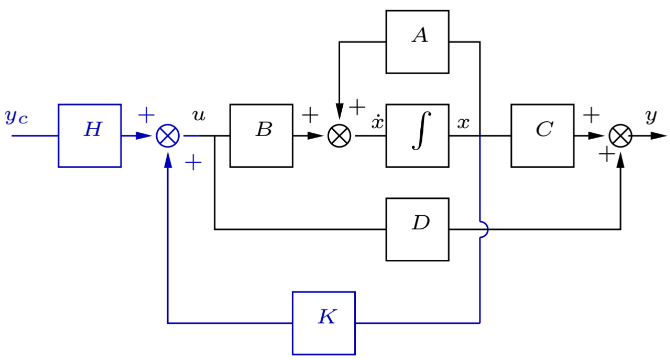


Figure 5.1: Principle of state-feedback control

Pole placement consists in determining the state-feedback gain vector K such that the closed-loop system has the desired poles — more precisely, such that the closed-loop state matrix admits the specified eigenvalues. This approach provides a direct way to shape the transient response of the system, including rise time, overshoot, and oscillatory behavior.

By substituting (5.2) into (5.3), we obtain:

$$\begin{cases} \dot{x}(t) = (A + BK)x(t) + BHy_c(t), \\ y(t) = (C + DK)x(t) + DHy_c(t). \end{cases} \tag{5.4}$$

Thus, the closed-loop state matrix is $A_{cl} = A + BK$. The pole placement problem can therefore be formulated as follows:

Pole Placement Problem: Given a matrix A and a vector B, determine the vector K such that the spectrum of $A + BK$ matches a desired set of eigenvalues.

1.4 Computation of the State-Feedback Gain Vector

To compute the state-feedback gain vector K, the following steps are followed:

- **Verification of controllability.** If the pair (A, B) is not controllable, pole placement is, in general, impossible.

$$\det\left(\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}\right) \neq 0 \tag{5.5}$$

- **Specification of the desired characteristic polynomial:**

$$D_d(s) = \prod_{i=1}^{n}(s - \lambda_i) = s^n + \alpha_{n-1}s^{n-1} + \ldots + \alpha_2 s^2 + \alpha_1 s + \alpha_0 \tag{5.6}$$

- **Determination of the open-loop characteristic polynomial:**

$$D(s) = \det(sI - A) = s^n + a_{n-1}s^{n-1} + \ldots + a_2 s^2 + a_1 s + a_0 \tag{5.7}$$

- **Computation of the feedback vector $\hat{K}$ in the canonical basis:**

$$\hat{K} = \begin{bmatrix} k_1 & k_2 & \dots & k_n \end{bmatrix} = \begin{bmatrix} a_0 - \alpha_0 & a_1 - \alpha_1 & \dots & a_{n-1} - \alpha_{n-1} \end{bmatrix} \tag{5.8}$$

- **Computation of the transformation matrix M:**

$$M = \begin{bmatrix} m_1 & m_2 & \dots & m_n \end{bmatrix} \quad \begin{cases} m_n = B, \\ m_{n-1} = (A + a_{n-1}I)B, \\ m_{n-2} = (A^2 + a_{n-1}A + a_{n-2}I)B, \\ \vdots \\ m_1 = (A^{n-1} + a_{n-1}A^{n-2} + \ldots + a_1 I)B. \end{cases} \tag{5.9}$$

- **Computation of the state-feedback vector in the original basis:**

$$K = \hat{K}M^{-1} \tag{5.10}$$

- **Computation of the feedforward gain:**

$$H = \left[D - (C + DK)(A + BK)^{-1}B\right]^{-1} \tag{5.11}$$

Example 23 *Consider a system described by the following state-space representation:*

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 3 \\ -1 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \end{cases} \tag{5.12}$$

The desired closed-loop poles of the system are $\lambda_1 = \lambda_2 = -1$.

Step 1. The controllability matrix is:

$$Q_c = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 1 & 1.5 \\ 0.5 & -3 \end{bmatrix}, \quad \det(Q_c) = -3.75 \tag{5.13}$$

Since $\det(Q_c) \neq 0$, the system is controllable.

Step 2. The desired closed-loop characteristic polynomial is:

$$D_d(s) = (s+1)(s+1) = s^2 + 2s + 1 \quad \Rightarrow \quad \alpha_0 = 1, \ \alpha_1 = 2 \tag{5.14}$$

Step 3. The open-loop characteristic polynomial is:

$$D(s) = \det(sI - A) = s^2 + s + 0.25 \quad \Rightarrow \quad a_0 = 3, \ a_1 = 4 \tag{5.15}$$

Step 4. The state-feedback vector in the canonical basis is:

$$\hat{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} a_0 - \alpha_0 & a_1 - \alpha_1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \end{bmatrix} \tag{5.16}$$

Step 5. The transformation matrix to the canonical basis is:

$$M = \begin{bmatrix} m_1 & m_2 \end{bmatrix}, \quad \begin{cases} m_2 = B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \\ m_1 = (A + a_1 I)B = \left(\begin{bmatrix} 0 & 3 \\ -1 & -4 \end{bmatrix} + 4I \right) \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}. \end{cases} \tag{5.17}$$

The transformation matrix is obtained as:

$$M = \begin{bmatrix} 1 & 5.5 \\ 0.5 & -1 \end{bmatrix}, \quad M^{-1} = \begin{bmatrix} 0.2667 & 1.4667 \\ 0.1333 & -0.2667 \end{bmatrix} \tag{5.18}$$

Step 6. The state-feedback vector in the original basis is:

$$K = \hat{K}M^{-1} = \begin{bmatrix} 2 & 2 \end{bmatrix} \begin{bmatrix} 0.2667 & 1.4667 \\ 0.1333 & -0.2667 \end{bmatrix} = \begin{bmatrix} 0.8 & 2.4 \end{bmatrix} \tag{5.19}$$

Step 7. The feedforward gain is:

$$H = \left[D - (C + DK)(A + BK)^{-1}B\right]^{-1} = 0.1818 \tag{5.20}$$

The MATLAB code [Script 11] can be used to compute the state-feedback gain for Example 23:

```
%% Script: Example 22
% Purpose: State feedback design with pole placement
%          + pre-compensation gain calculation
% ================================
clc; clear all; close all;
% --- State-space model ---
A = [0  3;
    -1 -4];      % System dynamics matrix
B = [1; 0.5];    % Input matrix
C = [1 0];       % Output matrix
D = 0;           % Direct transmission term
% ===================================
% Step 1: Controllability check
% ====================================
Qc = [B  A*B]; % Controllability matrix
determinant = det(Qc);
if determinant == 0
    disp('The system is NOT controllable');
else
    disp('The system is controllable');
    % ==================================
    % Step 2: Desired closed-loop poles
    % ===================================
    Lambda = [-1 -1];% Desired poles (double pole at -1)
    Dd = poly(Lambda);% Desired characteristic polynomial
    Alpha1 = Dd(2);% Desired coefficient of s
    Alpha0 = Dd(3);% Desired constant term
        % ===================================
    % Step 3: Open-loop characteristic polynomial
    % ========================================
    EigenValues = eig(A);        % Open-loop eigenvalues
    D_poly = poly(EigenValues); % Open-loop characteristic ...
       polynomial
    a1 = D_poly(2);
    a0 = D_poly(3);
        % ===================================
    % Step 4: State feedback in canonical form
    % ======================================
    K_tilde = [a0 - Alpha0, a1 - Alpha1];
        % ==============================
    % Step 5: Transformation matrix
    % ==================================
    M = [B  (A + a1*eye(2))*B]; % Basis transformation matrix
    Inverse_M = inv(M);
        % ==================================
    % Step 6: State feedback in original basis
    % ==========================================
    K = K_tilde * Inverse_M
        % ==================================
    % Step 7: Pre-compensation gain
    % =====================================
    H = inv(-C * inv(A + B*K) * B)
end
```

The Simulink model illustrated in Figure 5.2 can be used to validate the state-feedback control.

The closed-loop response to a unit step reference input is shown in Figure 5.3.

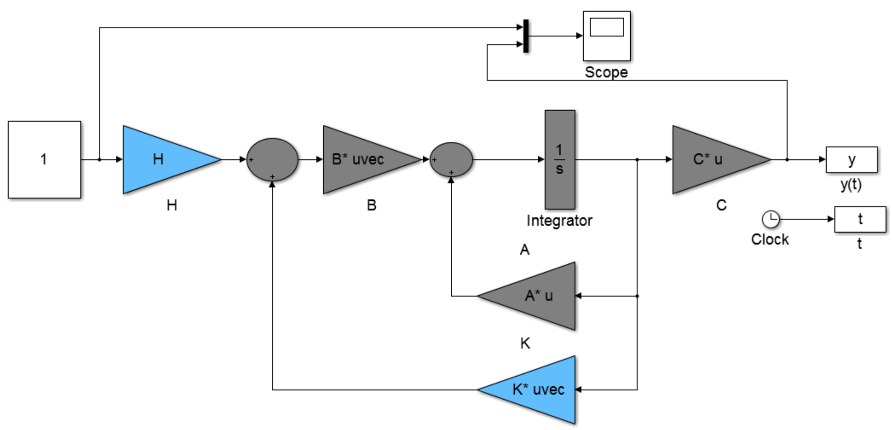


Figure 5.2: Simulink model for state-feedback control

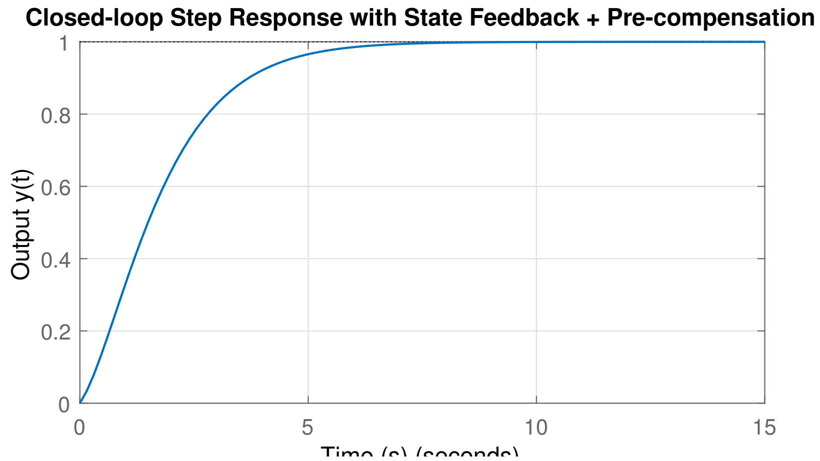


Figure 5.3: System output response for Example 23

2 Output-Feedback Control

In this second subsection, we assume that the components of the state vector x are not necessarily accessible. Instead, we only use the information available at the system output $y(t)$ to design a control law. This reduction in the amount of information available in the feedback loop naturally complicates the control problem. When only the output $y(t)$ is used in the feedback, the approach is referred to as *output feedback*.

2.1 Principle of Observation

The principle of observation is to use $u(t)$ and $y(t)$ to reconstruct an estimated vector $\hat{x}(t)$ that is as close as possible to the true state $x(t)$, so that state feedback can subsequently be implemented, as illustrated in Figure 5.4.

As shown in this figure, the combination of the observer (also called the *state reconstructor*) and the state-feedback gain K constitutes a *dynamic feedback*, which is similar to the state-feedback structure but relies on both $u(t)$ and $y(t)$.

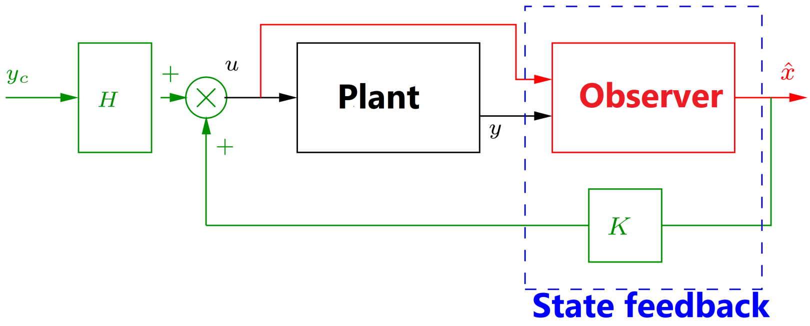


Figure 5.4: Principle of the observer

2.2 System Observability

A system is said to be *observable* at time t_1 if knowledge of the input signal and the output signal over a time interval $[t_1, t_2]$ makes it possible to compute the system state at t_1. If a system is observable for any initial time t_1, it is said to be *completely observable.*

2.3 Observability Criterion

A system is completely observable if and only if the observability matrix defined in (5.21) is full rank, i.e., if its determinant is non-zero.

$$Q_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{5.21}$$

2.4 Observer Equations

An observer can be defined by the following equations:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y - \hat{y}), \\ \hat{y}(t) = C\hat{x}(t). \end{cases} \tag{5.22}$$

The design of the observer consists in determining the gain L.

2.5 Design Procedure

To compute the observer gain L, the following steps are followed:

- **Verification of observability.** If the pair (A, C) is not observable, observer design is impossible.

$$\det\left(\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}\right) \neq 0 \tag{5.23}$$

- **Specification of the desired characteristic polynomial for the observer:**

$$\hat{D}_d(s) = \prod_{i=1}^{n}(s-\tau_i) = s^n + \beta_{n-1}s^{n-1} + \ldots + \beta_2 s^2 + \beta_1 s + \beta_0 \tag{5.24}$$

- **Determination of the open-loop characteristic polynomial:**

$$D(s) = \det(sI - A) = s^n + a_{n-1}s^{n-1} + \ldots + a_2 s^2 + a_1 s + a_0 \tag{5.25}$$

- **Computation of the observer gain $\tilde{L}$ in the canonical basis:**

$$\tilde{L} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_n \end{bmatrix} = \begin{bmatrix} a_0 - \beta_0 \\ a_1 - \beta_1 \\ \vdots \\ a_{n-1} - \beta_{n-1} \end{bmatrix} \tag{5.26}$$

- **Computation of the transformation matrix N:**

$$(N')^{-1} = \begin{bmatrix} n_1 & n_2 & \ldots & n_n \end{bmatrix} \quad \begin{cases} n_1 = C', \\ n_2 = (A' + a_{n-1}I)C', \\ n_3 = ((A')^2 + a_{n-1}A' + a_{n-2}I)C', \\ \vdots \\ n_n = ((A')^{n-1} + a_{n-1}(A')^{n-2} + \ldots + a_1 I)C'. \end{cases} \tag{5.27}$$

- **Computation of the observer gain in the original basis:**

$$L = N\tilde{L} \tag{5.28}$$

2.6 State-Feedback Control with State Estimation

The reconstruction of the state vector is here intended to implement a state-feedback control law when the state vector cannot be directly measured. By combining the state observer and the state-feedback law, we obtain the structure illustrated in Figure 5.5.

The control law in this case is given by:

$$u(t) = Hy_c(t) + K\hat{x}(t) \tag{5.29}$$

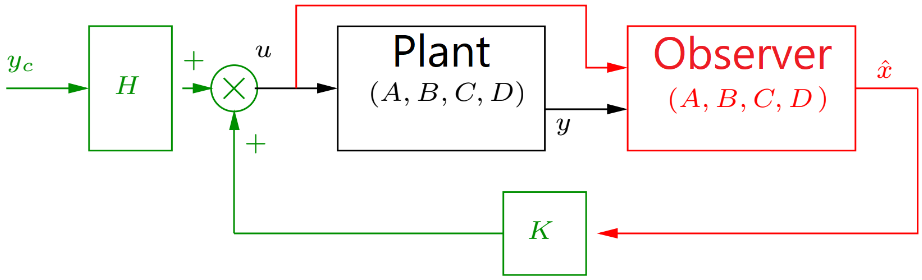


Figure 5.5: Principle of output-feedback control

Example 24 *Consider a system described by the following state-space representation:*

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 3 \\ -1 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases} \tag{5.30}$$

The desired closed-loop poles of the system are $\tau_1 = \tau_2 = -0.5$.

Step 1: The observability matrix is

$$Q_o = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, \quad \det(Q_o) = 3 \tag{5.31}$$

Since $\det(Q_o) \neq 0$, the system is observable.

Step 2: The desired closed-loop characteristic polynomial is

$$D_d(s) = (s + 0.5)(s + 0.5) = s^2 + s + 0.25 \quad \Rightarrow \quad \beta_0 = 0.25,\ \beta_1 = 1 \tag{5.32}$$

Step 3: The open-loop characteristic polynomial is

$$D(s) = \det(sI - A) = s^2 + s + 0.25 \quad \Rightarrow \quad a_0 = 3,\ a_1 = 4 \tag{5.33}$$

Step 4: The observer gain in the canonical basis is

$$\tilde{L} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = \begin{bmatrix} a_0 - \beta_0 \\ a_1 - \beta_1 \end{bmatrix} = \begin{bmatrix} 2.75 \\ 3 \end{bmatrix} \tag{5.34}$$

Step 5: The transformation matrix to the canonical basis is:

$$(N')^{-1} = \begin{bmatrix} n_1 & n_2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 0 & 3 \end{bmatrix}, \quad \begin{cases} n_1 = C' \\ n_2 = (A' + a_1 I)C' \end{cases} \tag{5.35}$$

From this, the matrix N can be obtained as:

$$N = \begin{bmatrix} 1 & 0 \\ -1.3333 & 0.3333 \end{bmatrix} \tag{5.36}$$

Step 6: The observer gain in the original basis is:

$$L = \begin{bmatrix} 2.7500 \\ -2.6667 \end{bmatrix} \tag{5.37}$$

The Matlab script [Script 12] can be used to compute the observer gain for Example 24. The Simulink model shown in Figure 5.6 is used to validate the output-feedback control.

```
%% Script: Example 23 - Observer Design
A=[0  3; -1 -4];B=[1; 0.5];C=[1 0];D = 0;
% ==============================
% Step 1: Observability test
% ===============================
Q_o = [C; C*A];% Observability matrix
determinant = det(Q_o);
if determinant == 0
    disp('The system is NOT observable');
    return
else
    disp('The system is observable');
end
% ============================================
% Step 2: Desired closed-loop poles for the observer
Lambda = [-0.5 -0.5];% Desired observer poles
Dd = poly(Lambda);% Desired characteristic polynomial
Beta1 = Dd(2);
Beta0 = Dd(3);
% Step 3: Open-loop characteristic polynomial
eigVals = eig(A);        % Eigenvalues of A
D_poly = poly(eigVals);  % Characteristic polynomial
a1 = D_poly(2);
a0 = D_poly(3);
% Step 4: Observer gain in canonical form
L_tilde = [a0 - Beta0;  a1 - Beta1];
% Step 5: Transformation matrix N
n1 = C';% First column
n2 = (A' + a1*eye(2)) * C';% Second column
n  = [n1 n2];
N  = (inv(n))';
% Step 6: Observer gain in original basis
L = N * L_tilde
```

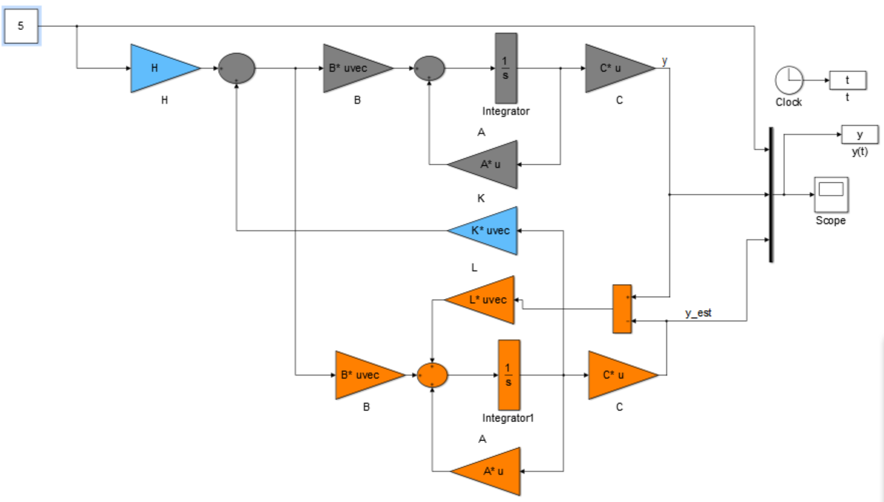


Figure 5.6: Simulink model for output-feedback control

The closed-loop control result for a reference input of 5 is shown in Figure 5.7.

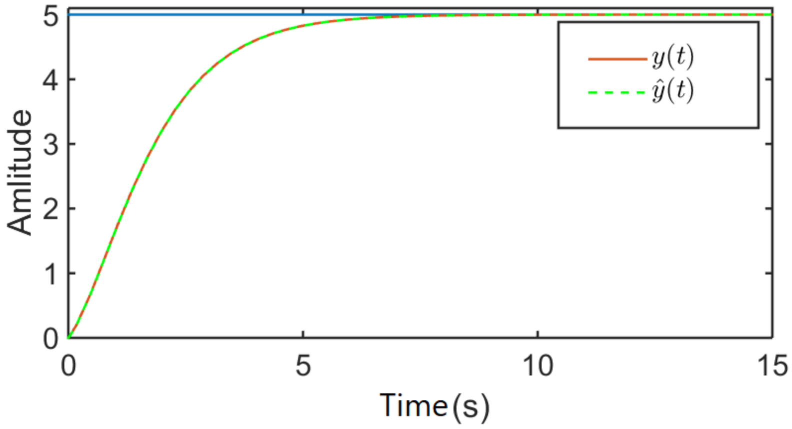


Figure 5.7: System output response for Example 24

3 Exercises

3.1 Exercise 1

Consider a system described by the following state-space representation:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 1 & 2 \\ \alpha & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t) \end{cases}$$

Determine the condition on the parameter α for this system to be completely controllable.

3.2 Exercise 2

Let a system be given by the following state-space representation:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t) \end{cases}$$

Compute the state-feedback vector that places the closed-loop poles at $p_1 = -2 + 3j$ and $p_2 = -2 - 3j$, and then deduce the corresponding pre-compensator matrix.

3.3 Exercise 3

The system is described by the following state-space model:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -2 & 1 \\ -1 & 6 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases}$$

Calculate the observer gain that ensures a closed-loop dynamics characterized by the poles $p_1 = -1 + 0.5j$ and $p_2 = -1 - 0.5j$.

4 Solutions

4.1 Solution to Exercise 1

The system is described by the following state-space representation:

$$\begin{cases} \dot{x} = \begin{bmatrix} 1 & 2 \\ \alpha & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t) \end{cases}$$

To evaluate controllability, we compute the controllability matrix:

$$Q_c = \begin{bmatrix} B \; AB \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & \alpha + 1 \end{bmatrix}, \quad \det(Q_c) = \alpha - 2$$

Thus, the system is controllable if and only if

$$\det(Q_c) \neq 0 \quad \Rightarrow \quad \alpha \neq 2.$$

—

4.2 Solution to Exercise 2

The system is described by:

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t) \end{cases}$$

Step 1: Controllability matrix

$$Q_c = \begin{bmatrix} B \; AB \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ 0.5 & -2 \end{bmatrix}, \quad \det(Q_c) = -1 \neq 0$$

Therefore, the system is controllable.

Step 2: Desired closed-loop characteristic polynomial

$$D_d(s) = (s + 2 + 3j)(s + 2 - 3j) = s^2 + 4s + 13 \quad \Rightarrow \quad \alpha_0 = 13, \; \alpha_1 = 4$$

Step 3: Open-loop characteristic polynomial

$$D(s) = \det(sI - A) = s^2 + 2s + 3 \quad \Rightarrow \quad a_0 = 3, \; a_1 = 2$$

Step 4: Feedback gain in the canonical basis

$$\hat{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} a_0 - \alpha_0 & a_1 - \alpha_1 \end{bmatrix} = \begin{bmatrix} -10 & -2 \end{bmatrix}$$

Step 5: Transformation matrix

$$M = \begin{bmatrix} m_1 \; m_2 \end{bmatrix}, \quad \begin{cases} m_2 = B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \\ m_1 = (A + a_1 I)B \end{cases}$$

After computation:

$$M = \begin{bmatrix} 1 & 2 \\ 0.5 & -1 \end{bmatrix}, \quad M^{-1} = \begin{bmatrix} -0.5 & -1 \\ -0.25 & 0.5 \end{bmatrix}$$

Step 6: Feedback gain in the original basis

$$K = \hat{K}M^{-1} = \begin{bmatrix} -10 & -2 \end{bmatrix} \begin{bmatrix} -0.5 & -1 \\ -0.25 & 0.5 \end{bmatrix} = \begin{bmatrix} 5 & 9 \end{bmatrix}$$

Step 7: Feedforward gain

$$H = \Big(D - (C + DK)(A + BK)^{-1}B\Big)^{-1} \quad \Rightarrow \quad H = \tfrac{1}{3}$$

—

4.3 Solution to Exercise 3

The system is given by:

$$\begin{cases} \dot{x} = \begin{bmatrix} -2 & 1 \\ -1 & 6 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases}$$

Step 1: Observability matrix

$$Q_o = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}, \quad \det(Q_o) = 1 \neq 0$$

Thus, the system is observable.

Step 2: Desired closed-loop polynomial

$$D_d(s) = (s + 1 - 0.5j)(s + 1 + 0.5j) = s^2 + 2s + 1.25 \quad \Rightarrow \quad \beta_0 = 1.25,\ \beta_1 = 2$$

Step 3: Open-loop polynomial

$$D(s) = \det(sI - A) = s^2 - 4s - 11 \quad \Rightarrow \quad a_0 = -11,\ a_1 = -4$$

Step 4: Observer gain in the canonical basis

$$\tilde{L} = \begin{bmatrix} a_0 - \beta_0 \\ a_1 - \beta_1 \end{bmatrix} = \begin{bmatrix} -12.25 \\ -6 \end{bmatrix}$$

Step 5: Transformation matrix

$$(N')^{-1} = \begin{bmatrix} n_1 & n_2 \end{bmatrix} = \begin{bmatrix} 1 & -6 \\ 0 & 1 \end{bmatrix}, \quad \Rightarrow \quad N = \begin{bmatrix} 1 & 0 \\ 6 & 1 \end{bmatrix}$$

Step 6: Observer gain in the original basis

$$L = N\tilde{L} = \begin{bmatrix} -12.25 \\ -79.5 \end{bmatrix}$$

Chapter 6

Kalman Filter

1 Introduction

In general, the filtering function aims to estimate a useful piece of information (signal) that is corrupted by noise. The Kalman filter provides an estimate of the system state based on prior information regarding the state evolution and on actual measurements. It is a statistical approach to data assimilation, whose principle is to correct the model trajectory by combining observations with the information provided by the model, so as to minimize the error between the true state and the filtered state.

In this chapter, we introduce the fundamental notions, concepts, definitions, and general principles of the Kalman filter.

2 Kalman Observer

Consider a disturbed system described by the following state-space model, known as the *Kalman model*:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Mw(t), \\ y(t) = Cx(t) + Du(t) + v(t), \end{cases} \tag{6.1}$$

where:

- $x(t)$ is the n-dimensional *state vector*,
- $u(t)$ is the m-dimensional *input or control vector*,
- $y(t)$ is the p-dimensional *output or observation vector*,
- $w(t)$ is a q-dimensional random input (process noise),
- $v(t)$ is a p-dimensional random input (measurement noise).

We make the following assumptions:

- The pair (A, C) is detectable, i.e., there is no unstable and unobservable mode in the system.

- The signals $w(t)$ and $v(t)$ are zero-mean Gaussian white noises with power spectral densities (PSD) W and V, respectively:

$$\begin{cases} E[w(t)w^T(t+\tau)] = W\delta(t), \\ E[v(t)v^T(t+\tau)] = V\delta(t), \\ E[w(t)v^T(t+\tau)] = 0, \end{cases} \tag{6.2}$$

  where the last relation in (6.2) expresses the stochastic independence of the noises $w(t)$ and $v(t)$.

- V is invertible (there are as many independent white noise sources as there are measurements in the output equation).

All deterministic information that can be known about the system is grouped in the model (i.e., $\dot{x}(t) = Ax(t)+Bu(t)$), while all stochastic information is represented by the noises $w(t)$ and $v(t)$. The state noise term $w_x = Mw$ represents both external disturbances and modeling errors.

3 Structure of an Unbiased Estimator

A Kalman filter is a dynamic system with two (vector) inputs: the deterministic control $u(t)$ and the measurement $y(t)$. The state (or output) of this filter is an estimate of the system state $x(t)$. Let:

$$\dot{\hat{x}}(t) = A_f\hat{x}(t) + B_f u(t) + K_f y(t), \tag{6.3}$$

be the state-space representation of the filter. Naturally, the filter must be initialized with $\hat{x}(t_0)$, the estimated state of the system at the initial instant t_0.

We denote the estimation error of the system state as $\epsilon(t) = x(t) - \hat{x}(t)$, and the initialization error as $\epsilon(t_0) = x(t_0) - \hat{x}(t_0)$.

By subtracting equation (6.3) from the system's state equation and using the output equation y, we obtain:

$$\dot{\epsilon}(t) = Ax + Bu(t) + Mw - A_f\hat{x} - B_f u + K_f(Cx + Du + v(t)), \tag{6.4}$$

$$\dot{\epsilon}(t) = (A - K_f C)\epsilon - (A - K_f C - A_f)\hat{x} + (B - K_f D - B_f)u + Mw(t) + K_f v(t). \tag{6.5}$$

Since the noises w and v are Gaussian and the system is linear, $\epsilon(t)$ is also a Gaussian random variable. We now focus on the expected value (mean) of $\epsilon(t)$.

Unbiased estimator: We require the estimator to be unbiased, i.e.:

- for any control profile $u(\tau)$ applied over the horizon $\tau \in [t_0, t]$,
- and for any initialization $\hat{x}(t_0)$,

we want the mean of the estimation error to converge to zero as $t \to \infty$.

Because the noises w and v are zero-mean, we can write:

$$E[\dot{\epsilon}(t)] = (A - K_f C)E[\epsilon(t)] - (A - K_f C - A_f)E[\hat{x}(t)] + (B - K_f D - B_f)u(t),$$

and $\lim_{t\to\infty} E[\epsilon(t)] = 0$, $\forall u(t)$, $\forall E[\hat{x}]$ if and only if:

$$A_f = A - K_f C, \quad B_f = B - K_f D, \tag{6.6}$$

and

$$A - K_f C \ \text{ is stable.} \tag{6.7}$$

By substituting (6.7) into (6.3), the Kalman filter equation becomes:

$$\dot{\hat{x}}(t) = (A\hat{x}(t) + Bu(t)) + K_f(y(t) - C\hat{x}(t) - Du(t)). \tag{6.8}$$

In this expression, the first term, $(A\hat{x} + Bu)$, corresponds to the *prediction* of the system state evolution based on the current estimate $\hat{x}$. This prediction is essentially an online simulation of the system model. Since the model is imperfect, the prediction is corrected using the discrepancy between the actual measurement y and the predicted measurement $\hat{y} = C\hat{x} + Du$, weighted by the filter gain K_f.

The error signal $y - \hat{y}$ is also called the *innovation*. The corresponding block diagram (for the case $D = 0$) is shown in Figure 6.1.

This structure guarantees that the estimator is unbiased for any system matrices A, B, C, D and any filter gain K_f such that $A - K_f C$ is stable (which justifies assumption H1: the presence of an unstable and unobservable mode prevents the existence of a stabilizing gain K_f, and thus the construction of an unbiased estimator).

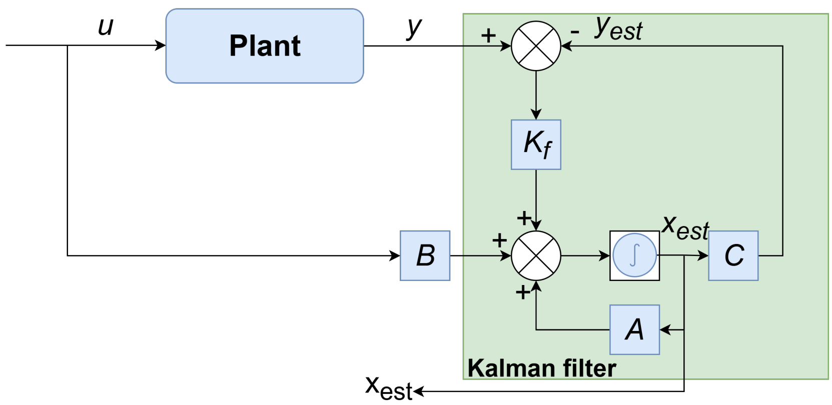


Figure 6.1: Functional block diagram of the Kalman filter (case $D = 0$).

4 Minimum-Variance Estimator

The gain K_f is determined based on the confidence in the model (expressed by the spectral density W) relative to the confidence in the measurement (expressed by the spectral density V). If the model is very accurate (W is very *small*) and the measurement is very noisy (V is very *large*), then the gain K_f must be small.

Among all possible gains K_f satisfying the constraint (6.7), we select the one that minimizes the variance of the state estimation error $\epsilon(t)$ for all t. Recall that $\epsilon(t) = x(t) - \hat{x}(t)$ is a centered (unbiased) Gaussian random vector with n components. The Gaussian nature of this variable guarantees that if the variance of the estimation error is minimized, then $\hat{x}(t)$ is indeed the best possible estimate of $x(t)$.

4.1 General Solution

We therefore seek K_f that minimizes:

$$J(t) = \sum_{i=0}^{n} E[\epsilon_i(t)^2] = E[\epsilon(t)^T \epsilon(t)] = \text{trace}[\epsilon(t)^T \epsilon(t)] = \text{trace}[P(t)], \tag{6.9}$$

where

$$P(t) = E[(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T]$$

is the covariance matrix of the estimation error.

Substituting (6.6) into (6.5), the evolution of $\epsilon(t)$ is described by the state equation:

$$\dot{\epsilon}(t) = (A - K_f C)\epsilon(t) + \begin{bmatrix} M & -K_f \end{bmatrix} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix}. \tag{6.10}$$

with

$$E\left[\begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \begin{bmatrix} w^T(t) & v^T(t) \end{bmatrix}\right] = \begin{bmatrix} W_{q\times q} & 0_{q\times p} \\ 0_{p\times q} & V_{p\times p} \end{bmatrix} \delta(t).$$

By applying the theorem for linear stochastic systems, we conclude that the covariance of the estimation error $P(t)$ satisfies the Riccati differential equation:

$$\dot{P}(t) = AP(t) + P(t)A^T - P(t)C^T V^{-1} C P(t) + MWM^T, \tag{6.11}$$

with

$$K_f = P(t)C^T V^{-1}. \tag{6.12}$$

This Riccati differential equation must be integrated with the initial condition $P(t_0)$, which reflects the confidence in the initial filter state $\hat{x}(t_0)$:

$$P(t_0) = E[(x(t_0) - \hat{x}(t_0))(x(t_0) - \hat{x}(t_0))^T].$$

The gain $K_f(t)$ is then obtained from $P(t)$ using (6.12). The Kalman filter is therefore non-stationary.

Equations (6.9), (6.11), and (6.12) form the continuous-time Kalman filter, which must be integrated starting from $\hat{x}(t_0)$ and $P(t_0)$. The integration of (6.11) and computation of $K_f(t)$ can be carried out either online or offline. In the latter case, the computed law $K_f(t)$ must be stored in the processor.

In practice, implementation of the Kalman filter is usually performed on a digital computer, and thus in discrete time. One may either discretize the continuous-time equations or design a Kalman filter directly in discrete form. In both cases, the filter equations are entirely defined by the problem data, namely the matrices A, B, M, C, D, W, and V.

4.2 Steady-State Kalman Filter

In steady state, once the transient regime due to initialization errors has dissipated, the estimation error becomes a stationary random process. Thus, we have $\dot{P}(t) = 0$.

The constant positive-definite covariance matrix P, which represents the estimation error covariance at steady state, is the positive solution of the algebraic Riccati equation:

$$AP + PA^T - PC^T V^{-1} CP + MWM^T = 0. \tag{6.13}$$

Accordingly, the filter gain becomes constant:

$$K_f = PC^T V^{-1}. \tag{6.14}$$

It can be verified that the positive definiteness of P guarantees the stability of the filter, meaning that all eigenvalues of the matrix $A - K_f C$ have negative real parts.

5 Exercises

5.1 Exercise 1

Consider the continuous-time system:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) + v(t), \end{cases} \tag{6.15}$$

with Gaussian white noises $w(t)$ and $v(t)$ of spectral densities $W = 1$, $V = 0.5$.

- Verify whether the pair (A, C) is observable.
- Check if the assumptions of the Kalman filter are satisfied.
- Derive the Riccati equation for the error covariance matrix $P(t)$.

5.2 Exercise 2

Consider again the system of Exercise 1.

- In steady-state, write the algebraic Riccati equation for P.
- Solve this equation and compute P.
- Deduce the steady-state Kalman gain K_f.

5.3 Exercise 3

Consider the scalar system:

$$\dot{x}(t) = -x(t) + w(t), \qquad y(t) = x(t) + v(t), \tag{6.16}$$

with $W = 0.1$ and $V = 1$.

- Derive the Riccati equation for $P(t)$.
- Solve or sketch the evolution of $P(t)$ over time.
- Discuss the influence of W and V on the Kalman gain $K_f(t)$.

6 Solutions to Exercises

6.1 Solutions to exercise 1

We consider the system:

$$\dot{x}(t) = Ax(t) + Bu(t) + Gw(t), \quad y(t) = Cx(t) + v(t), \tag{6.17}$$

with

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad W = 1, \quad V = 0.5.$$

(i) Observability

The observability matrix is

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since $\det(\mathcal{O}) = 1 \neq 0$, the pair (A, C) is observable.

(ii) Kalman Assumptions

The assumptions are satisfied: $W \succeq 0$, $V > 0$, (A, C) is detectable, and (A, G) is stabilizable.

(iii) Riccati Equation

The Riccati differential equation is

$$\dot{P} = AP + PA^\top + Q - PC^\top V^{-1} CP,$$

with

$$Q = GWG^\top = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

```
%% Exercise 1
clc; clear; close all;

A = [0 1; -2 -3];
B = [0; 1];
C = [1 0];
W = 1;     % process noise covariance
V = 0.5;   % measurement noise covariance

% 1. Observability
Q_o = obsv(A,C);
disp('Rank of observability matrix = '), disp(rank(Q_o));

if rank(Q_o) == size(A,1)
    disp('=> The system is observable');
else
    disp('=> The system is NOT observable');
end
```

```

% 2. Kalman filter assumptions
disp('W > 0 and V > 0 => assumptions satisfied');

% 3. Riccati differential equation
syms P11 P12 P21 P22 real
P = [P11 P12; P21 P22];
Ric = A*P + P*A' + W*eye(2) - P*C'*(1/V)*C*P;
disp('Riccati equation:');
disp(Ric);
```

6.2 Solutions to exercise 2

(i) Algebraic Riccati Equation

In steady-state:

$$0 = AP + PA^\top + Q - PC^\top V^{-1} CP.$$

(ii) Solution

The stabilizing solution is approximately

$$P^\star = \begin{bmatrix} 0.0731 & 0.0053 \\ 0.0053 & 0.1631 \end{bmatrix}.$$

(iii) Steady-State Kalman Gain

$$K_f^\star = P^\star C^\top V^{-1} = \frac{1}{0.5} \begin{bmatrix} p_{11} \\ p_{12} \end{bmatrix} \approx \begin{bmatrix} 0.1463 \\ 0.0107 \end{bmatrix}.$$

```
%% Exercise 2
clc; clear; close all;

A = [0 1; -2 -3];
C = [1 0];
W = eye(2);
V = 0.5;

% Solve Algebraic Riccati Equation
[P,¬,Kf] = care(A',C',W,V);
disp('Steady-state P ='); disp(P);
disp('Kalman Gain Kf ='); disp(Kf');
```

6.3 Solutions to exercise 3

System:

$$\dot{x}(t) = -x(t) + w(t), \qquad y(t) = x(t) + v(t),$$

with $W = 0.1$, $V = 1$.

(i) Riccati Equation

$$\dot{P} = -2P + 0.1 - P^2.$$

(ii) Steady-State Solution

Solving $-2P + 0.1 - P^2 = 0$ gives

$$P^\star = -1 + \sqrt{1.1} \approx 0.0488.$$

(iii) Influence of W and V

The Kalman gain is $K_f(t) = P(t)/V$. Increasing W increases K_f, while increasing V decreases it.

```
%% Exercise 3
clc; clear; close all;

A = -1;
C = 1;
W = 0.1;
V = 1;

% Riccati ODE: dP/dt = 2*A*P + W - (P*C')*(1/V)*(C*P)
f = @(t,P) 2*A*P + W - (P^2)/V;

% Solve numerically
[t,P] = ode45(f,[0 10],1); % initial covariance = 1

figure;
plot(t,P,'b-','LineWidth',2);
xlabel('Time'); ylabel('P(t)');
title('Evolution of covariance P(t)');
grid on;

% Kalman gain
K = P./V;
figure;
plot(t,K,'r-','LineWidth',2);
xlabel('Time'); ylabel('K(t)');
title('Kalman Gain over time');
grid on;
```

General Conclusion

This course has provided a comprehensive introduction to the fundamental concepts and techniques of linear control systems. Starting from the classical frequency-domain approach and progressing to the modern state-space framework, the goal has been to build both intuition and analytical proficiency in modeling, analyzing, and designing feedback systems.

Students have been introduced to the mathematical foundations of system dynamics, the Laplace transform, and transfer function representations, as well as to the principles of stability and performance assessment. Through worked examples and MATLAB simulations, the theoretical results have been connected to practical engineering applications.

A strong emphasis has been placed on understanding the interplay between time-domain and frequency-domain viewpoints, showing how classical control remains relevant and complementary to modern methods. This dual perspective equips students with the conceptual tools needed to analyze, interpret, and design robust control systems in real-world contexts.

Recommendations to Students:

- Revise the fundamental properties of the Laplace transform and system response.
- Practice MATLAB exercises regularly — focus on plotting step, impulse, and frequency responses.
- Develop the habit of checking system stability (poles, zeros, and gain margins) before simulation.
- Complement theory with experimentation — simulate, modify parameters, and observe their effects.
- Always seek to understand the physical meaning behind mathematical results.

Ultimately, the mastery of control theory requires both analytical understanding and practical curiosity. Continue exploring, testing, and designing — this is the essence of becoming a control engineer.

Bibliography

[1] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2nd ed., Princeton University Press, 2020.

[2] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 15th ed., Pearson, 2021.

[3] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 8th ed., Pearson, 2019.

[4] N. S. Nise, *Control Systems Engineering*, 8th ed., Wiley, 2019.

[5] MathWorks, "`margin`: Gain margin, phase margin, and crossover frequencies," Documentation, 2024–2025.

[6] MathWorks, "Stability Margins in Control System Tuning," User's Guide, 2024–2025.

[7] Y. Zheng, "Nyquist Plot and Criterion (ECE 171A, Lecture 17)," lecture slides, UC San Diego, 2025.

[8] School of Engineering and Computer Science, Victoria University of Wellington, "Analysis with Bode Plots," course notes, 2025.

[9] University of Sheffield, "Gain and Phase Margins," online tutorial, 2024.

[10] A. Hastir and R. Muolo, "A generalized Routh–Hurwitz criterion for the stability analysis of polynomials with complex coefficients: application to the PI-control of vibrating structures," *arXiv*, 2023.

[11] J. Yang, X. Hou, and Y. Li, "A Generalization of Routh–Hurwitz Stability Criterion for Fractional-Order Systems with Order $\alpha \in (1, 2)$," *Fractal and Fractional*, vol. 6, no. 10, p. 557, 2022.

[12] P. de Larminat, *Automatique : Commande des systèmes linéaires*, Hermès Lavoisier, 1996.

[13] H. Egon, *Asservissements linéaires échantillonnés et représentation d'état: Méthodes*, 2001.

[14] L. Jaulin, *Représentation d'état pour la modélisation et la commande des systèmes*, Lavoisier, 2005.

[15] R. L. Williams and D. A. Lawrence, *Linear State-Space Control Systems*, John Wiley & Sons, 2007.