INTRODUCTION TO CONTROL SYSTEMS

Kamran Iqbal
University of Arkansas at Little Rock

University of Arkansas at Little Rock

Introduction to Control Systems

Kamran Iqbal

The LibreTexts libraries are Powered by NICE CXOne and are supported by the Department of Education Open Textbook Pilot Project, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

TABLE OF CONTENTS

Licensing

*A detailed breakdown of this resource's licensing can be found in **Back Matter/Detailed Licensing**.*

CHAPTER OVERVIEW

1: Mathematical Models of Physical Systems

Learning Objectives

1. Obtain a mathematical model of a physical system.
2. Obtain system transfer function from its differential equation model.
3. Obtain a physical system model in the state variable form.
4. Linearize a nonlinear dynamic system model about an operating point.

1.0: Prelude to Mathematical Models of Physical Systems

This chapter describes the process of obtaining the mathematical description of a dynamic system, i.e., a system whose behavior changes over time. The system is assumed to be assembled from components. The system model is based on the physical laws that govern the behavior of various system components.

Physical systems of interest to engineers include, for example, electrical, mechanical, electromechanical, thermal, and fluid systems. By using lumped parameter assumption, their behavior is mathematically described in terms of ordinary differential equation (ODE) models. These equations are nonlinear, in general, but can be linearized about an operating point for analysis and design purposes.

Models of interconnected components are assembled from individual component describptions. The components in electrical systems include resistors, capacitors, and inductors. The components used in mechanical systems include inertial masses, springs, and dampers (or friction elements). For thermal systems, these include thermal capacitance and thermal resistance. For hydraulic and fluid systems, these include reservoir capacity and flow resistance.

In certain physical systems, properties (or entities) flow in and out of a system boundary, e.g., a hydraulic reservoir, or thermal chamber. The dynamics of such systems is described by conservation laws and/or balance equations. In particular, let Q represent an accumulated property, q_{in} and q_{out} represent the inflow and outflow rates, then the model is described as:

$$\frac{dQ}{dt} = q_{in} - q_{out} + g - c$$

where g and c denote the internal generation and consumption of that property.

The Laplace transform converts a linear differential equation into an algebraic equation, which can be manipulated to obtain an input-output description described as a transfer function. The transfer function forms the basis of analysis and design of control systems using conventional methods. In contrast, the modern control theory is established on time-domain analysis involving the state equations, that describe system behavior as time derivatives of a set of state variables.

Linearization of nonlinear models is accomplished using Taylor series expansion about a critical point, where the linear behavior is restricted to the neighborhood of the critical point. The linear systems theory is well-estabished and serves as the basic tool for controller design.

1.1: Model Variables and Element Types

Flow and Across Variables

Modeling of a physical system involves two kinds of variables: flow variables that 'flow' through the components, and across variables that are measured across those components.

For example, in electric circuits, potential (voltage) is measured across elements, whereas electrical charge (current) flows through the circuit elements.

In mechanical linkage systems, displacement and velocity are measured across elements, whereas force or effort 'flows' through the linkages.

In thermal and fluid systems, heat and mass serve as the flow variables, whereas temperature and pressure constitute the across variables.

A physical element is characterized by the relationship between flow and across variables. The three basic types are the **resistive, inductive, and capacitive** elements. The terminology, taken from electrical circuits, extends to other types of physical systems as well.

While the resistive element dissipates energy, both the capacitive and inductive elements store energy. For example, a capacitor stores electrical energy and a moving mass stores kinetic energy. The energy storage accords memory to the element that accounts for it dynamic behavior modeled by an ODE.

Let $q(t)$ denote a flow variable and $x(t)$ denote an across variable associated with an element; then, the element type is defined by their mutual relationships, described as follows:

Definition: Resistive Element

A resistive element is described by a proportional relationship:

$$x(t) = k\, q(t)$$

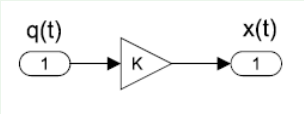


Figure 1.1.1: A resistive element.

For example, the voltage and current relationship through a resister is described by a proportional relationship called **Ohm's law**: $V(t) = R\, i(t)$.

Similarly, the force–velocity relationship though a linear mechanical damper is a proportional one: $v(t) = \frac{1}{b} f(t)$.

Definition: Capacitive Element

A capacitive element is described by the relation:

$$x(t) = k \int q(t)dt + x_0$$

i.e., the across variable varies in proportion to the accumulated amount of the flow variable. Whereas, the flow variable varies proportionally with the rate of change of the across variable:

$$q\,(t) = \frac{1}{k}\frac{dx\,(t)}{dt}$$

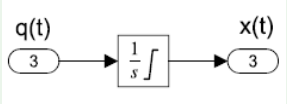


Figure 1.1.2: A capacitive element.

For example, the voltage and current relationship through a capacitor is given as: $V(t) = \frac{1}{C}\int i(t)dt + V_0$. Since the current integral represents the accumulation of electrical charge, we have: $Q = CV$. The inverse relationship is described as: $i\,(t) = C\frac{dV}{dt}$.

Similarly, the force–velocity relationship that governs the movement of an inertial mass is described as: $v(t) = \frac{1}{m}\int f(t)dt + v_0$. Its inverse is the familiar Newton's second law of motion: $f\,(t) = m\frac{dv(t)}{dt}$.

Definition: Inductive Element

An inductive element is described by the relationship:

$$x(t) = k\frac{dq(t)}{dt}$$

i.e., the across variable is obtained by differentiating the flow variable. Alternatively, the flow variable varies in proportion to the accumulation of the across variable as:

$$q(t) = \frac{1}{k}\int x(t)dt + q_0$$

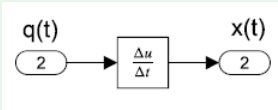


Figure 1.1.3: An inductive element.

For example, the voltage–current relationship through an inductive coil in an electric circuit is given as: $V(t) = L\frac{di(t)}{dt}$. The inverse relationship is described as: $i(t) = \frac{1}{L}\int V(t)dt + i_0$.

Similarly, the force–velocity relationship though a linear spring is given as: $v(t) = \frac{1}{K}\frac{df(t)}{dt}$. The inverse relation is described as: $f\,(t) = K\int v\,(t)\,dt + f_0$.

1.2: First-Order ODE Models

Electrical, mechanical, thermal, and fluid systems that contain a single energy storage element are described by first-order ODE models.

Let $u(t)$ denote a generic input, $y(t)$ denote a generic output, and τ denote the time constant; then, a generic first-order ODE model is expressed as:

$$\tau \frac{dy(t)}{dt} + y(t) = u(t)$$

The time constant, τ, denotes the time when the system response to a constant input rises to 63.2% of its final value. The time constant is measured in $[sec]$.

Examples

Example 1.2.1

A series RC network is connected across a constant voltage source, V_s (Figure 1.2.1). Kirchhoff's voltage law (KVL) is used to model the circuit behavior as: $v_R + v_C = V_s$

In the above capital letters represent constant values and small letters represent time-varying quantities.

Let $v_C = v_0$ define the circuit output; then, $v_R = iR = RC\frac{dv_0}{dt}$, hence

$$RC\frac{dv_0(t)}{dt} + v_0(t) = V_s$$

The time constant of the RC circuit is given as: $\tau = RC$.

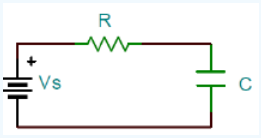


Figure 1.2.1: A series RC circuit driven by a constant voltage source.

Example 1.2.2

A parallel RL network is connected across a constant current source, I_s (Figure 1.2.2). The circuit is modeled by a first-order ODE, where the variable of interest is the inductor current, i_L, and Kirchhoff's current law (KCL) is applied at a node to obtain: $i_R + i_L = I_s$.

By substituting $i_R = \frac{v}{R} = \frac{L}{R}\frac{di_L}{dt}$ we obtain the ODE description of the RL circuit as:

$$\frac{L}{R}\frac{di_L(t)}{dt} + i_L(t) = I_s$$

The time constant of the RL circuit is given as: $\tau = \frac{L}{R}$.

Figure 1.2.2: A parallel RL circuit driven by a constant current source.

Example 1.2.3

The motion of an inertial mass, m, acted by a force, $f(t)$, in the presence of kinetic friction, represented by b, is governed by Newton's second law of motion (Figure 1.2.3). Let $v(t)$ represent the velocity; then, the resultant force on the mass element is: $f - bv$. Hence

$$m\frac{dv(t)}{dt} + bv(t) = f(t)$$

The time constant for the mechanical model is: $\tau = \frac{m}{b}$, which describes the rate at which the velocity builds up in response to a constant force input.

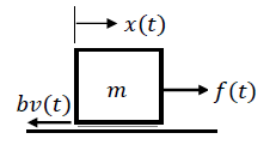


Figure 1.2.3: Motion of an inertial mass with applied force under surface friction.

Example 1.2.4

A model for room heating is developed as follows (Figure 1.2.4): let q_i, denote heat inflow, C_r denote the thermal capacity of the room, θ_r denote the room temperature, θ_a denote the ambient temperature, and R_w denote a thermal resistance representing wall insulation; then, the heat energy balance equation is given as:

$$C_r \frac{d\theta_r}{dt} + \frac{\theta_r - \theta_a}{R_w} = q_i$$

In terms of the temperature differential, $\Delta\theta = \theta_r - \theta_a$, the governing ODE is:

$$R_w C_r \frac{d\Delta\theta}{dt} + \Delta\theta = R_w q_i$$

By comparing with the generic first-order ODE model, $\tau \frac{\mathrm{dy}}{\mathrm{dt}} + y = u$, the thermal time constant is described as: $\tau = R_w C_r$. Further, the temperature is measures in $[^\circ C]$, heat flow is measured in $[W]$, thermal capacitance is measured in $\left[\frac{J}{^\circ C}\right]$, and thermal resistance is measured in $\left[\frac{^\circ C}{W}\right]$.

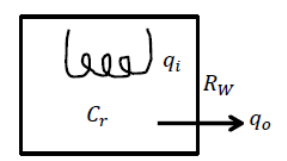


Figure 1.2.4: A model of room heating with heat source and outflow through the walls.

Example 1.2.5

A cylindrical reservoir is filled with an incompressible fluid supplied from an inlet with a controlled exit through a hydraulic valve at the bottom (Figure 1.2.5).

Let P denote the hydraulic pressure, A denote the area of the reservoir, h denote the height, V denote the volume, ρ denote the mass density, R_l denote the valve resistance to the fluid flow, q_{in}, q_{out} denote the volumetric flow rates, and g denote the gravitational constant; then, the base pressure in the reservoir is obtained as:

$$P = P_{atm} + \rho g h = P_{atm} + \frac{\rho g}{A} V$$

The reservoir capacitance is defined as: $C_h = \frac{dV}{dP} = \frac{A}{\rho g}$; then, the governing equation for hydraulic flow through the reservoir is given as:

$$C_h \frac{dP}{dt} = q_{in} - \frac{P - P_{atm}}{R_l}$$

In terms of the pressure difference, the equation is written as:

$$R_l C_h \frac{d\Delta P}{dt} + \Delta P = R_l q_{in}.$$

By comparing with generic first-order ODE model, $\tau \frac{dy}{dt} + y = u$, the hydraulic time constant is described as: $\tau = R_w C_r$. Further, by substituting: $\Delta P = \rho g h$ and $C_h = \frac{A}{\rho g}$, we can equivalently express the ODE in terms of the liquid height, $h\,(t)$, in the reservoir as:

$$A R_l \frac{dh}{dt} + \rho g h = R_l q_{in}$$

In the above, the hydraulic pressure is measures in $\left[\frac{N}{m^2}\right]$, volumetric flow is measured in $\left[\frac{m^3}{s}\right]$, hydraulic capacitance is measured in $\left[\frac{m^5}{N}\right]$, and flow resistance is measured in $\left[\frac{Ns}{m^5}\right]$.

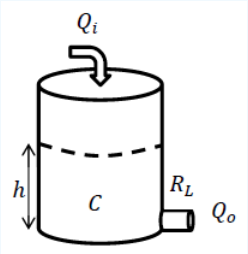


Figure 1.2.5: A hydraulic reservoir with flow in and flow out through bottom valve.

Solving First-Order ODE Model

Consider the response of a first-order ODE model: $\tau\frac{dy(t)}{dt}+y\left(t\right)=u(t)$ to a step forcing function.
By applying the Laplace transform with initial conditions: $y\left(0\right)=y_0$, we obtain an algebraic equation:

$$\tau\left(sy\left(s\right)-y_0\right)+y(s)=u(s)$$

We assume a unit step input $u\left(t\right)$, where $u\left(s\right)=\frac{1}{s}$; then, the output is solved as:

$$y\left(s\right)=\frac{1}{s\left(\tau s+1\right)}+\frac{\tau y_0}{\tau s+1}$$

We use partial fraction expansion (PFE) to express the output as:

$$y\left(s\right)=\frac{1}{s}-\frac{\tau}{\tau s+1}+\frac{\tau y_0}{\tau s+1}$$

By using inverse Laplace transform, we obtain the time-domain solution to the ODE as:

$$y\left(t\right)=1+\left(y_0-1\right)e^{-t/\tau},\ \ t\geq 0$$

Let the steady-state value of the output be denoted as: $y_\infty=lim_{t\to\infty}\,y(t)$; then, the step response of a general first-order ODE model is expressed as:

$$y\left(t\right)=\left[y_\infty+\left(y_0-y_\infty\right)e^{-t/\tau}\right]u\left(t\right)$$

In the above, $u\left(t\right)$ denotes a unit step function, used to show causality, i.e., the response is valid for $t\geq 0$.

Transfer Function

The transfer function description of a dynamic system is obtained from the ODE model by the application of Laplace transform assuming zero initial conditions. The transfer function describes the input-output relationship in the form of a rational function, i.e., a ratio of two polynomials in the Laplace variable s.

A first-order ODE model with input $u(t)$ and output $y(t)$ is described as: $\tau\frac{\mathrm{dy(t)}}{\mathrm{dt}}+y(t)=u(t)$.
For the first-order model, application of Laplace transform with zero initial conditions gives: $(\tau s+1)y(s)=u(s)$.
The resulting input–output transfer function is given as:

$$\frac{y(s)}{u(s)}=\frac{1}{\tau s+1}$$

Output Response

For zero initial conditions, $y_0=0$, the output is expressed as:

$$y\left(t\right)=\left(1-e^{-t/\tau}\right)u\left(t\right)$$

The system response for $\tau=1\ sec$ is plotted in Figure 1.5.
The output values at selected times, $t=k\tau,\ \ k=0,1,\ldots$ are compiled in the following table. By convention, the output is assumed to have reached steady-state when it attains 98% of its final value. Hence, the settling time of the system is expressed as:

$t_s = 4\tau$.

Table 1.1: The step response of a first-order model at selected time instances.

Time	Output value
0	$y(0) = 0$
1τ	$1 - e^{-1} \cong 0.632$
2τ	$1 - e^{-2} \cong 0.865$
3τ	$1 - e^{-3} \cong 0.950$
4τ	$1 - e^{-4} \cong 0.982$
5τ	$1 - e^{-5} \cong 0.993$

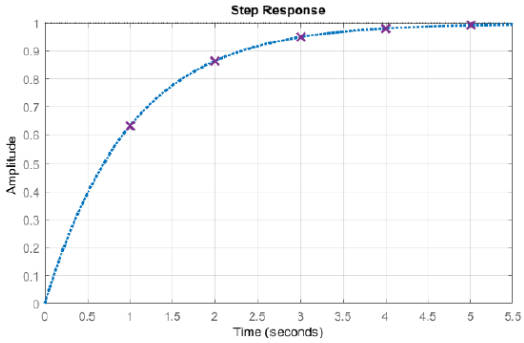


Figure 1.2.6: Response of a first-order ODE model ($\tau = 1sec$).

```
pkg load control
s=tf('s');
Gs=1/(s+1);
step(Gs), hold
plot(1:5,1-exp(-(1:5)),'x')
```

run | restart | restart & run all

1.3: Second-Order ODE Models

A physical system that contains two energy storage elements is described by a second-order ODE. Examples of second-order models are discussed below:

Example \(\PageIndex{1}\)

A series RLC circuit with voltage input \(V_s(t)\) and current output \(i(t)\) has a governing relationship obtained by applying the Kirchoff's voltage law to the mesh (Figure \(\PageIndex{1}\)):

\[L\frac{ di(t)}{ dt} +Ri(t)+\frac{1}{C} \int i(t) dt=V_ s (t) \nonumber \]

The above integro-differential equation can by converted into a second-order ODE by expressing it in terms of the electric charge, \(q(t)\), as:

\[L\frac{ d^{2} q(t)}{ dt^{2} } +R\frac{ dq(t)}{ dt} +\frac{1}{C} q(t)=V_ s (t) \nonumber \]

Alternatively, the series RLC circuit can be described in terms of two first-order ODE's involving natural variables, the current, \(i(t)\), and the capacitor voltage, \(V_c(t)\), as:

\[L\frac{ di(t)}{ dt} +Ri(t)+V_{c} (t)=V_s (t), \ \ \ \ \ C\frac{dV_c}{dt}=i(t) \nonumber \]

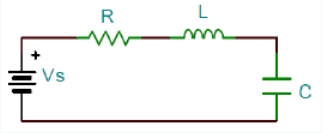


Figure \(\PageIndex{1}\): A series RLC circuit driven by a constant voltage source.

Example \(\PageIndex{2}\)

The motion of a mass element of weight, \(mg\), pulled upward by a force, \(f(t)\), is described using position output, \(y(t)\), by a second-order ODE:

\[m\frac{ d^{2} y(t)}{ dt^{2} } +mg=f(t) \nonumber \]

The second-order ODE expresses the fact that the moving mass has both the kinetic and potential energies (Figure \(\PageIndex{2}\)).

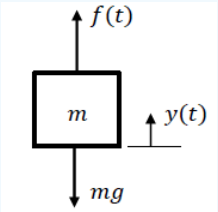


Figure \(\PageIndex{2}\): Motion of an inertial mass under gravitational field.

Example \(\PageIndex{3}\)

A mass–spring–damper system includes a mass affected by an applied force, \(f(t)\); its motion is restrained by a combination of a spring and a damper (Figure 1.8).

Let \(x(t)\) denote the displacement of the mass from a fixed reference; then, the dynamic equation of the system, obtained by using Newton's second law of motion, takes a familiar form:

\[m\frac{ d^{2} x(t)}{ dt^{2} } +b\frac{ dx(t)}{ dt} +kx(t)=f(t) \nonumber \]

In compact notation, we may express the ODE as: \[m\ddot{x}\; +\; b\dot{x}\; +\; kx=f \nonumber \]

where the dots above the variable represent time derivative, i.e., \(\dot{x}\left(t\right)=\frac{dx\left(t\right)}{dt}\), \(\ddot{x}\left(t\right)=\frac{d^2x\left(t\right)}{dt^2}\).

Using position, \(x(t)\), and velocity, \(v(t)\) as variables, the mass-spring-damper system is described by two first-order equations (called state equations) given as: \[\frac{dx}{dt}=v(t), \ \ \ \ \ \frac{dv}{dt}=\frac{1}{m} \left(-kx(t)-bv(t)+f(t) \right) \nonumber \]

In the absence of damping, the dynamic equation of the mass-spring system reduces to:

\[m\frac{d^2x\left(t\right)}{dt^2}+kx\left(t\right)=f(t) \nonumber \]

The above equation describes simple harmonic motion (SHM) with \({\omega }^2_0=k/m\). From elementary physics, the general solution to the equation is given as:

\[x\left(t\right)=A{cos {\omega }_0\ }t+B{sin {\omega }_0t\ } \nonumber \]

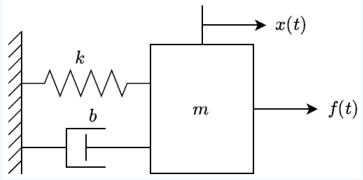


Figure \(\PageIndex{3}\): A mass-spring-damper system model.

Solving Second-Order Models

A second-order ODE model can be solved by applying the Laplace transform to both sides of the differential equation. Let a general second-order ODE model be described as:

\[\ddot{y}\left(t\right)+a_1\dot{y}\left(t\right)+a_2y\left(t\right)=b_1\dot{u}\left(t\right)+b_2u\left(t\right) \nonumber \]

Application of the Laplace transform, assuming the following initial conditions: \(y\left(0\right)=y_0,\ \ \dot{y}\left(0\right)={\dot{y}}_0, \ \ u(0)=0\), gives:

\[\left(s^2+a_1s+a_2\right)y\left(s\right)-\left(s+a_1\right)y_0-{\dot{y}}_0=\left(b_1s+b_2\right)u\left(s\right) \nonumber \]

or,

\[y\left(s\right)=\frac{1}{s^2+a_1s+a_2}\left[\left(s+a_1\right)y_0+{\dot{y}}_0+\left(b_1s+b_2\right)u\left(s\right)\right] \nonumber \]

Transfer Function

By applying Laplace transform assuming no initial conditions, we obtain: \(\left(s^2+a_1s+a_2\right)y\left(s\right)=(b_1s+b_2)u(s)\); the resulting input-output transfer function is given as:

\[\frac{y\left(s\right)}{u\left(s\right)}=\frac{b_1s+b_2}{s^2+a_1s+a_2} \nonumber \]

The characteristic equation of the model is defined as: \(s^{2} +a_{1} s+a_{2} =0\).

For the mass–spring–damper model described by: \(m\ddot{x}\; +\; b\dot{x}\; +\; kx=f\) (Example 1.3.3), the transfer function from force input to displacement output is given as:

\[\frac{x(s)}{f(s)} =\frac{1}{ms^{2} +bs+k} \nonumber \]

Transfer function of a physical systems is a proper fraction, i.e., the degree of the denominator polynomial is greater than the degree of numerator polynomial.

The roots of its denominator polynomial characterize the response of the second-order ODE model. The response is monotonic in the case of real roots, and oscillatory for complex roots.

✔ Example \(\PageIndex{4}\)

Consider the mass-spring-damper model (Example \(\PageIndex{3}\)), where the following parameter values are assumed: \(m=1\), \(k=2\), and \(b=3\). Then, the second-order ODE is given as:

\[\ddot{x}\left(t\right)+3\dot{x}\left(t\right)+2x\left(t\right)=f\left(t\right) \nonumber \]

The application of the Laplace transform assuming zero initial conditions gives: \[(s^2+3s+2)y\left(s\right)=f(s) \nonumber \]

The characteristic equation of the model is given as: \(s^2+3s+2=0\). The equation has real roots at: \(s=-1,\ -2\).

Next, let \(f\left(t\right)=2u\left(t\right),\ f\left(s\right)=\frac{2}{s}\); then, the output is solved as: \[x\left(s\right)=\frac{2}{s\left(s+1\right)\left(s+2\right)} \nonumber \]

We use partial fraction expansion (PFE) to obtain:

\[x(s)=\frac{1}{s}-\frac{2}{s+1}+\frac{1}{s+2} \nonumber \]

By applying the inverse Laplace transform, the output response of the spring-mass-damper system is obtained as (Figure 1.9):

\[x\left(t\right)=\left(1-2e^{-t}+e^{-2t}\right)u(t) \nonumber \]

where \(u\left(t\right)\) represents the unit-step function.

Note: The solution of an ODE generally involves the PFE. We may use the online SimboLab partial fraction calculator for this purpose (https://www.symbolab.com/solver/part...ns-calculator/).

✔ Example \(\PageIndex{5}\)

Consider the mass-spring-damper model (Example \(\PageIndex{3}\)), with following parameter values: \(m=1,\ k=2,\ b=2\). Then, the second-order ODE is given as:

\[\ddot{x}\left(t\right)+2\dot{x}\left(t\right)+2x\left(t\right)=f\left(t\right) \nonumber \]

The application of the Laplace transform assuming zero initial conditions gives:

\[(s^2+2s+2)y\left(s\right)=f(s) \nonumber \]
The characteristic equation of the model is given as: \(s^2+2s+2=0\). The equation has complex roots at: \(s=-1\pm j1\).
Let \(f\left(t\right)=2u\left(t\right),\ f\left(s\right)=2/s\); then, the output is solved as:
\[x\left(s\right)=\frac{2}{s\left(s^2+2s+2\right)} \nonumber \]
Next, a PFE of the output is carried out to obtain:
\[x(s)=\frac{1}{s}-\frac{s+2}{s^2+2s+2} \nonumber \]
The quadratic factor is expressed as: \({\left(s+1\right)}^2+1^2\); the quadratic term is split to obtain:
\[x\left(s\right)=\frac{1}{s}-\frac{s+1}{{\left(s+1\right)}^2+1^2}-\frac{1}{{\left(s+1\right)}^2+1^2} \nonumber \]
By applying the inverse Laplace transform, the output response of the spring-mass-damper system is obtained as (Figure 1.9):
\[x\left(t\right)=\left(1-e^{-t}\cos t -e^{-t}\sin t \right)u\left(t\right) \nonumber \]
where \(u\left(t\right)\) represents the unit-step function.

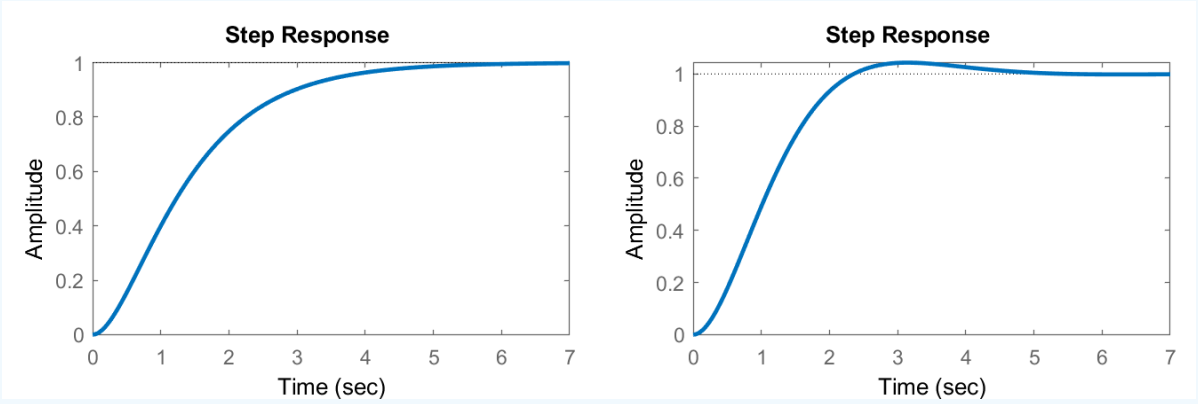


Figure \(\PageIndex{4}\): Time response of second-order system models: characteristic equation with real roots (left); with complex roots (right).

```
pkg load control
s=tf('s');
Gs1=1/(s^2+3*s+2);
Gs2=1/(s^2+2*s+2);
step(Gs1)
step(Gs2)
```

run | restart | restart & run all

```
  Hello world! Hello world!
```

1.4: An Electro-Mechanical System Model

Model of a DC Motor

A electro-mechanical system converts electrical energy into mechanical energy or vice versa. A armature-controlled DC motor (Figure 1.4.1) represents such a system, where the input is the armature voltage, $V_a(t)$, and the output is motor speed, $\omega(t)$, or angular position $\theta(t)$.

In order to develop a model of the DC motor, let $i_a(t)$ denote the armature current, and L and R denote the electrical side inductance and the coil resistance. The mechanical side inertia and friction are denoted as J and b, respectively. Let k_t denote the torque constant and k_b the motor constant; then, the dynamic equations of the DC motor are given as:

$$L\frac{di_a(t)}{dt} + Ri_a(t) + k_b\omega(t) = V_a(t)$$

$$J\frac{d\omega(t)}{dt} + b\omega(t) - k_t i_a(t) = 0$$

By using the Laplace transform, these equations are transformed into algebraic equations as:

$$(Ls+R)i_a(s) + k_b\omega(s) = V_a(s)$$

$$(Js+b)\omega(s) - k_t i_a(s) = 0$$

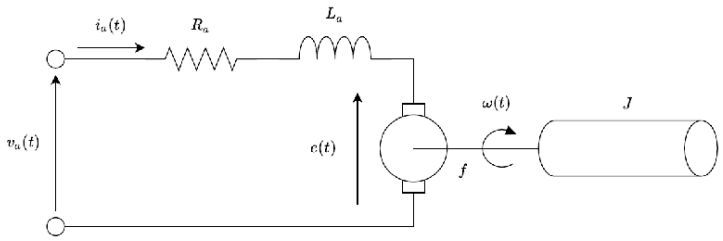


Figure 1.4.1: Schematic of an armature-controlled DC motor.

Motor Transfer Function

In order to obtain an input-output relation for the DC motor, we may solve the first equation for $i_a(s)$ and substitute in the second equation. Alternatively, we multiply the first equation by k_t, the second equation by $(Ls+R)$, and add them together to obtain:

$$(Ls+R)(Js+b)\omega(s) + k_t k_b\omega(s) = k_t V_a(s)$$

Then, the transfer function of the DC motor with voltage input and angular velocity output is derived as:

$$\frac{\omega(s)}{V_a(s)} = \frac{k_t}{(Ls+R)(Js+b)+k_t k_b}$$

The denominator polynomial in the DC motor transfer function typically has real roots, which are reciprocals of the motor time constants $(\tau_e,\ \tau_m)$. In terms of the time constants, the DC motor model is described as:

$$\frac{\omega(s)}{V_a(s)} = \frac{k_t/JL}{(s+1/\tau_e)(s+1/\tau_m)}$$

The electrical constant represents the build up of electrical current in the armature circuit, whreas the mechanical constat represents the build up of motor speed in response to the developed motor torque. Further, the slower mechanical time constant dominates the overall motor response to a change in the armature voltage.

The angular position $\theta(s)$ of the shaft is obtained by integrating the angular velocity $\omega(s)$; the transfer function from $V_a(s)$ to the angular displacement $\theta(s)$ is given as:

$$\frac{\theta(s)}{V_a(s)} = \frac{k_t}{s\left[(Ls+R)(Js+b)+k_t k_b\right]}$$

Example 1.4.1

A small DC motor has the following parameter values: $R=1\Omega,\ L=0.01\,H,\ J=0.01\ kgm^2,\ b=0.1;\frac{N-s}{rad},$ and $k_t=k_b=0.05$; then, the motor transfer function from armature voltage to angular velocity is obtained as:

$$\frac{\omega(s)}{V_a(s)}=\frac{500}{(s+100)(s+10)+25}=\frac{500}{(s+10.28)(s+99.72)}$$

The two motor time constants are given as: $\tau_e\cong 1\ 0ms,\ \tau_m\cong 100\ ms$, where τ_e matches the time constant of an RL circuit ($\tau_e=L/R$) and τ_m matches the time constant of inertial mass in the presence of friction ($\tau_m=J/b$).
Assuming a unit-step input, $u(s)=\frac{1}{s}$, is applied to the motor, the motor speed is obtained as:

$$\omega(s)=\frac{500}{s\,(s+10.28)\,(s+99.72)}=\frac{0.488}{s}-\frac{0.544}{s+10.28}+\frac{0.056}{s+99.72}$$

By applying the inverse Laplace transform, the time-domain output is given as (Figure 13a):

$$\omega(t)=\left[0.488-0.544e^{-10.28t}+0.056e^{-99.72t}\right]u(t)$$

where $u(t)$ denotes a unit-step function. The motor response is plotted in Figure Figure 1.4.2.

Simplified Model of a DC motor

A simplified model of the DC motor is obtained by ignoring the coil inductance ($L\to 0$). Then, the electrical side equation is modified as:

$$Ri_a(s)+k_b\omega(s)=V_a(s)$$

By substituting $i_a(s)$ into the torque equation, the mechanical side equation is given as:

$$R(Js+b)\omega(s)+k_tk_b\omega(s)=k_tV_a(s)$$

The resulting first-order motor transfer function is given as:

$$\frac{\omega(s)}{V_a(s)}=\frac{k_t/R}{Js+b+k_tk_b/R}$$

The first-order model has a single motor time constant $\left(\tau_m=\frac{JR}{bR+k_tk_b}\right)$, and is written as:

$$\frac{\omega(s)}{V_a(s)}=\frac{k_t/JR}{s+1/\tau_m}$$

Example 1.4.2

Using the parameter values for a small DC motor (Example 1.4.1), its reduced first-order transfer function is obtained as:

$$\frac{\omega(s)}{V_a(s)}=\frac{5}{s+10.25}$$

The resulting motor time constant evaluates as: $\tau_m\cong 97.6\ ms$, which approximates the slower mechanical time constant in the second-order model.
Assuming a unit-step input, the motor response is obtained as:

$$\omega(s)=\frac{5}{s\,(s+10.25)}=\frac{0.488}{s}-\frac{0.488}{s+10.25}$$

By applying the inverse Laplace transform, the motor output is given as:

$$\omega(t)=\left[0.488-0.488e^{-10.25t}\right]u(t)$$

The motor response to a unit-step input is plotted in Figure 1.4.2.

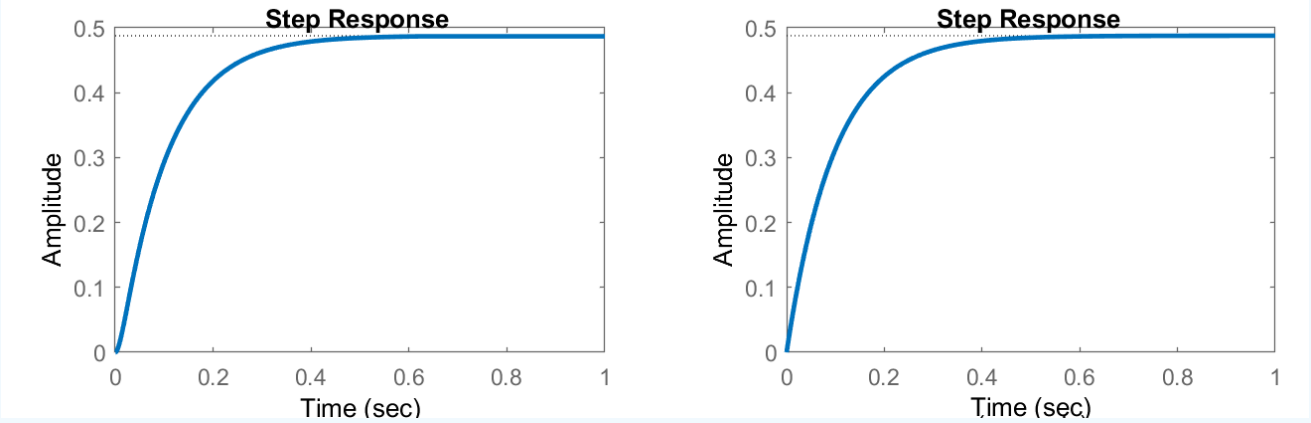


Figure 1.4.2: DC motor response to unit-step input: second-order motor model (left); first-order motor model (right).

```
pkg load control
R=1; L=.01; J=.01; b=.1; kt=.05; kb=.05;
s=tf('s');
Gs=kt/((L*s+R)*(J*s+b)+kt*kb)
step(Gs), hold
L=0;
Gs1=kt/((L*s+R)*(J*s+b)+kt*kb)
step(Gs1)
```

run | restart | restart & run all

1.5: Industrial Process Models

Industrial Process Models

Industrial processes comprise exchange of chemical, electrical or mechanical energy in the manufacturing of industrial products. An industrial process model in its simplified form is represented by a first-order lag with a dead-time that represents the time delay between the application of input and the appearance of the process output.

Let τ represent the time constant associated with an industrial process, τ_d represent the dead-time, and K represent the process dc gain; then, simplified industrial process dynamics are represented by the following delay-differential equation:

$$\tau\frac{dy(t)}{dt}+y(t)=Ku(t-t_d)$$

Application of the Laplace transform produces the following first-order-plus-dead-time (FOPDT) model of an industrial process:

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

where the process parameters $\{K,\ \tau,\ \tau_d\}$, can be identified from the process response to inputs. An rational process model is obtained by using a Taylor series approximation of the delay term, $e^{-\tau_d s}$. Typical such approximations include:

$$e^{-\tau_d s}\simeq 1-\tau_d s,\quad e^{-\tau_d s}=\frac{1}{e^{\tau_d s}}\simeq\frac{1}{1+\tau_d s},\quad e^{-\tau_d s}=\frac{e^{-\tau_d s/2}}{e^{\tau_d s/2}}\simeq\frac{1-\tau_d s/2}{1+\tau_d s/2}$$

The last expression is termed as first-order Pade' approximation and is often preferred. Higher order approximations can also be used.

Example 1.5.1

The process parameters of a stirred-tank bioreactor are given as: $\{K,\ \tau,\ \tau_d\}=\{20,0.5,1\}$. The transfer function model of the process is formed as:

$$G(s)=\frac{20e^{-s}}{0.5s+1}. \tag{1.5.1}$$

By using a first-order Pade' approximation, a rational transfer function model of the industrial process with delay is obtained as:

$$G(s)=\frac{20\,(1-0.5s)}{(0.5s+1)^2}. \tag{1.5.2}$$

The step response of the bioreactor transfer function with Pade' approximation shows an undershoot due to the presence of right half-plane (RHP) zero in the transfer function.

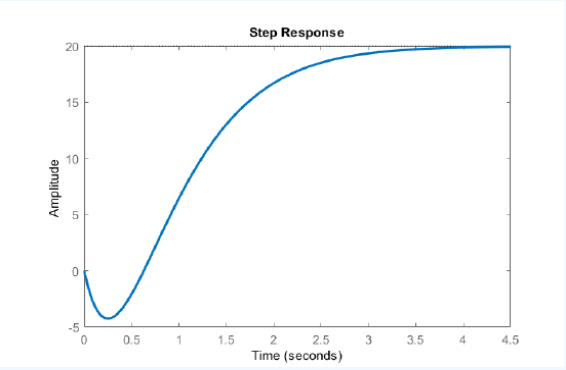


Figure 1.5.1: Step response of the bioreactor model with Pade' approximation.

```
pkg load control
s=tf('s');
Gs=20*(1-.5*s)/(1+.5*s)^2;
```

```
step(Gs)
```

run | restart | restart & run all

1.6: State Variable Models

State Variable Models

State variable models are time-domain models that express system behavior as time derivatives of state variables, i.e., the variables to express the process state. The state variables are typically selected as the natural variables associated with the energy storage elements in the process, but alternate variables can also be used. The state equations of the system describe the time derivatives of the state variables. When the state equations are linear, they are expressed in a vector-matrix form.

In the case of electrical networks, capacitor voltages and inductor currents serve as natural state variables. In the case of mechanical systems, positions and velocities of inertial masses serve as natural state variables. In thermal systems, heat flow is a natural state variable. In hydraulic systems, the head (height of the liquid in the reservoir) is a natural state variable.

The number of state variables determines the system order, however, the choice of state variables for a system model is not unique. For example, in a mechanical system model, position and momentum can serve as state variables in place of position and velocity.

Example 1.6.1

A series RLC circuit driven by a constant voltage source contains two energy storage elements, an inductor and a capacitor. Accordingly, let the inductor current, $i(t)$, and the capacitor voltage, $v_c(t)$, serve as state variables. Then, the circuit behavior is represented by the following equations:

$$C\frac{dv_c}{dt} = i, \quad L\frac{di}{dt} = V_s - v_c - Ri$$

In vector-matrix form, the state equations are represented as:

$$\frac{d}{dt}\begin{bmatrix} v_c \\ i \end{bmatrix} = \begin{bmatrix} 0 & 1/C \\ -1/L & -R/L \end{bmatrix}\begin{bmatrix} v_c \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} V_s$$

Let v_c denote the circuit output; then, the output equation is formed as:

$$v_c = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} v_c \\ i \end{bmatrix}$$

Example 1.6.2

The dynamic equation of the mass–spring–damper system is given as:

$$m\frac{d^2x(t)}{dt^2} + b\frac{dx(t)}{dt} + kx(t) = f(t)$$

Let the position, $x(t)$, and velocity, $v(t) = \dot{x}(t)$ serve as the state variables, and let $x(t)$ represent the output; then, the state and output equations for the model are given as:

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} f$$

$$x = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix}$$

Example 1.6.3

The dynamic equations for the DC motor are given as:

$$L\frac{di_a(t)}{dt} + Ri_a(t) + k_b\omega(t) = V_a(t)$$

$$J\frac{d\omega(t)}{dt} + b\omega(t) - k_t i_a(t) = 0$$

Let $i_a(t),\ \omega(t)$ serve as the state variables, and let $\omega(t)$ represent the output; then, the state variable model of the DC motor is given as:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -R/L & -k_b/L \\ k_t/J & -b/J \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} V_a$$

$$\omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

For a small DC motor, let the following parameter values be assumed: $R = 1\Omega,\ L = 1\ mH,\ J = 0.01\ kg\cdot m^2,\ b = 0.1\ \frac{N\cdot s}{rad},\ k_t = k_b = 0.05$. Then, the state variable model of the motor is given as:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a$$

$$\omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

State variable models are covered in more detail in Chapters 8-10.

1.7: Linearization of Nonlinear Models

Linearization of Nonlinear Functions

The behavior of a nonlinear system, described by $y = f(x)$, in the vicinity of a given operating point, $x = x_0$, can be approximated by plotting a tangent line to the graph of $f(x)$ at that point.

Analytically, linearization of a nonlinear function involves first-order Taylor series expansion about the operative point.

Let $\delta x = x - x_0$ represent the variation from the operating point; then the Taylor series of a function of single variable is written as:

$$f(x_0 + \delta x) = f(x_0) + \frac{\partial f(x_0)}{\partial x}\delta x + \dots$$

The resulting first order model is described by:

$$f(x) - f(x_0) = \frac{\partial f(x_0)}{\partial x}(x - x_0)$$

Example 1.7.1

Consider a vehicle driven in cruise control are represented by the block diagram (Figure 1.7.1). The forces acting on the car include its weight, driving force generated by engine torque applied to the wheels, aerodynamic drag, and tire to surface rolling friction.

The vehicle weighs $1440kg$ and is driven at $20m/s$ (about 45 mph). The vehicle experiences aerodynamic drag: $F_d = \frac{1}{2}\rho v^2 A c_d$. Assuming $\rho = 1.2\ kg/m^3$ (for air), $A = 4m^2$, and $c_d = .25$, results in a nonlinear drag force: $F_d = 0.6v^2 N$.

A first-order Taylor series expansion of the drag force about the cruising speed ($20\ m/s$) is given as:

$$F_d(v) = F_d(20) + F_d'|_{v=20}(v - 20).$$

Let $\delta F_d = F_d - F_d(20)$, $\delta v = v - 20$, denote the variations in the force and speed; then, the linearized model for the drag force is given as:

$$\delta F_d = 24\delta v.$$

Further, the tires generate a friction force: $F_r = 0.015W$, where W is the weight of the car. For $m = 1440kg$ and $g = 9.8m/s^2$, the tire friction is: $F_r = 212N$.

Let $T_e = rF_e$ denote the engine torque, where F_e is the force output and $r = 0.33m$ is the wheel radius; then, the dynamic equation of the vehicle is given as:

$$T_e/r - F_r - F_d = m\frac{dv}{dt}$$

Substituting the above parameter values results in:

$$m\frac{dv}{dt} + 0.6v^2 = 3T_e - 212.$$

Let δT_e denote the variation in the engine torque; then, a linearized model of the vehicle cruising at $20m/s$is given as:

$$1440\frac{d\delta v}{dt} + 24\delta v = 3\delta T_e$$

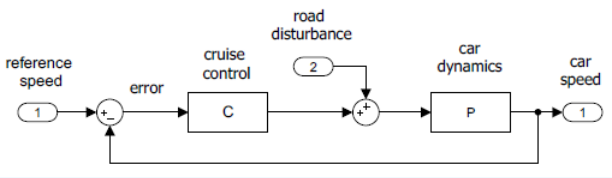


Figure 1.7.1: Block diagram of vehicle cruise control system.

Linearization of State Variable Models

Assume that nonlinear state variable model of a single-input single-output (SISO) system is described by the following equations:

$$\dot{x}(t) = f(x,u)$$

$$y(t) = g(x,u)$$

where x is a vector of state variables, u is a scalar input, y is a scalar output, f is a vector function of the state and input variables, and g is a scalar function of those variables.

A stationary point for the model is defined by: $f(x_e,u_e)=0$. The deviations from the stationary point are expressed as: $x(t)=x_e(t)+\delta x(t);\ \ u(t)=u_e(t)+\delta u(t)$.

In terms of the variations: $\delta x, \delta u$, the linearized model of the system is expressed as:

$$\dot{\delta}\, x(t) = [\partial f_i/\partial x_j]\,|_{(x_e,u_e)}\delta x(t) + [\partial f_i/\partial u]\,|_{(x_e,u_e)}u(t)$$

$$y(t) = [\partial g/\partial x_j]\,|_{(x_e,u_e)}\delta x(t) + [\partial g/\partial u]\,|_{(x_e,u_e)}\delta u(t),$$

where $[\partial f_i/\partial x_j]$ is a **Jacobian** matrix of partial derivatives; $[\partial f_i/\partial u]$, $[\partial g/\partial x_j]$ are vectors of partial derivatives, and $[\partial g/\partial u]$ is a scalar partial derivative; all derivatives are computed at the stationary point.

The linearized model is expressed in its familiar vector-matrix form as:

$$\dot{\delta}\, x(t) = A\delta x(t) + bu(t)$$

$$y(t) = c^T\delta x(t) + d\delta u(t).$$

In the above, A represents an $n\times n$ system matrix, b is a $n\times 1$ column vector of input distributions, c^T is a $1\times n$ row vector of output contributions, and d is a scalar gain.

Example 1.7.2

The model of a simple pendulum is described by the dynamic equation:

$$ml^2\ddot{\theta}(t) + mgl\ \sin\theta(t) = T(t),$$

where $\theta(t)$ is the pendulum angle, $T(t)$ is the applied torque; m, l represent the mass and the length of the pendulum, and g is the gravitational constant.

By using $(\theta,\ \omega)$ as the state variables for the pendulum, the nonlinear model is expressed as:

$$\frac{d}{dt}\begin{pmatrix}\theta\\ \omega\end{pmatrix} = \begin{pmatrix}\omega\\ -\frac{g}{l}\sin\theta\end{pmatrix} + \begin{pmatrix}0\\ T(t)\end{pmatrix}.$$

The Jacobian matrix for the simple pendulum is expressed as:

$$\left[\frac{\partial f}{\partial x}\right] = \begin{pmatrix}0 & 1\\ -\frac{g}{l}\cos\theta & 0\end{pmatrix}.$$

Two equilibrium points in the case of a simple pendulum can be identified: $\theta_e = 0^\circ,\ 180^\circ$. The linearized models defined at the equilibrium points are given as:

$$\theta_e = 0^\circ : \frac{d}{dt}\begin{bmatrix}\theta \\ \omega\end{bmatrix} = \begin{bmatrix}0 & 1 \\ -\frac{g}{l} & 0\end{bmatrix}\begin{bmatrix}\theta \\ \omega\end{bmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix} T$$

$$\theta_e = 180^\circ : \frac{d}{dt}\begin{bmatrix}\theta \\ \omega\end{bmatrix} = \begin{bmatrix}0 & 1 \\ \frac{g}{l} & 0\end{bmatrix}\begin{bmatrix}\theta \\ \omega\end{bmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix} T$$

The output equation in both cases is given as: $\theta(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta \\ \omega\end{bmatrix}$.

Ocatve examples

An NPV example in Octave

```
pkg load financial
r=.03/12;
p=(1:60)*1000;
npv(r,p)
```

run | restart | restart & run all

```
  Hello world! Hello world!
```

Histogram

run | restart | restart & run all

A data smoothing example in Octave

```
pkg load data-smoothing
npts = 20;
 x = rand(npts,1)*2*pi;
 y = sin(x);
 y = y + 1e-1*randn(npts,1);
 xh = linspace(0,2*pi,200)';
 [yh, lambda] = regdatasmooth (x, y, "d", 3, "xhat", xh);
 lambda
 plot(x,y,'o','markersize',10,xh,yh,xh,sin(xh))
 title("y(x)")
 legend("noisy","smoothed","sin(x)","location","northeast");
```

run | restart | restart & run all

```
          hello world
```

A linear algebra example in Octave

```
A=randn(3,5);
[u,s,v]=svd(A)
```

run | restart | restart & run all

R Examples

```
x<-1:10
```

run | restart | restart & run all

Sampling example in R

```
x<-1:7
sample(x,2)
```

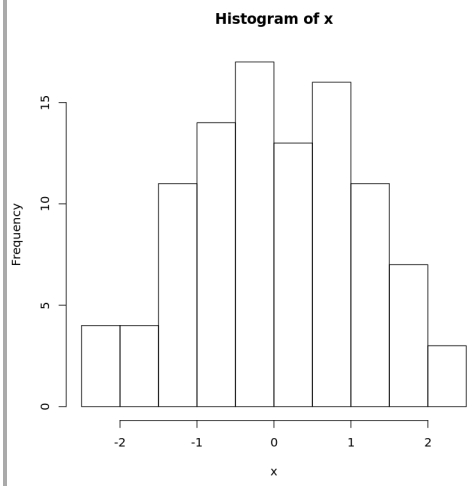
run | restart | restart & run all

```
1 · 5
```

Plot histogram in R

```
x<-rnorm(100)
hist(x)
```

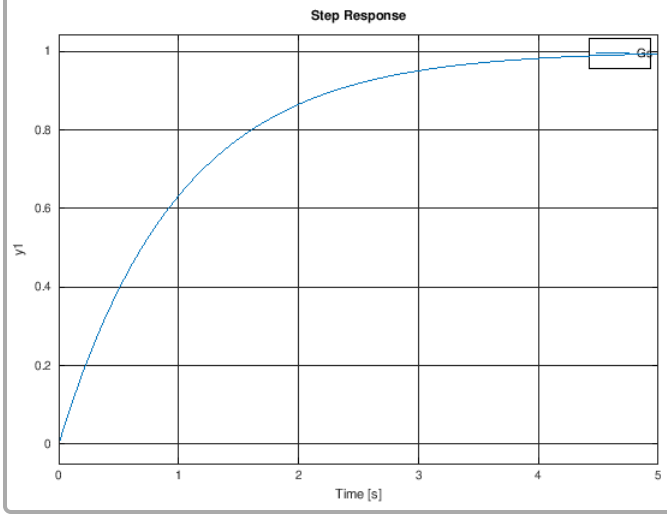
run | restart | restart & run all



Octave examples

```
pkg load control
s=tf('s');
Gs=1/(s+1);
step(Gs)
```

run | restart | restart & run all




```
pkg load control
s=tf('s');
Gs=2/(s^2+2*s+2);
step(Gs)
```

run | restart | restart & run all

```
warning: the 'damp' function belongs to the control package from
Octave Forge but
has not yet been implemented.

Please read <http://www.octave.org/missing.html> to learn how you can
contribute missing functionality.
error: 'damp' undefined near line 1 column 1
```

CHAPTER OVERVIEW

2: Transfer Function Models

Learning Objectives

1. Analyze transfer function models of dynamic systems.
2. Characterize the natural response of the system model.
3. Characterize the stability of the system model.
4. Obtain system response to step, impulse, and sinusoidal inputs.
5. Visualize the frequency response of the system.

2.0: Prelude to Transfer Function Models

This chapter analyzes the transfer function models of physical systems developed in Chapter 1. The transfer function is obtained by the application of Laplace transform to the linear differential equation description of the system. The transfer function, denoted by $G(s)$, is a rational function of a complex frequency variable, s. Given the transfer function and an input, $u(s)$, the response of the system can be computed as:

$$y(s) = G(s)\,u(s).$$

The transfer function is a ratio of two polynomials is s. The zeros of the transfer function, i.e., those frequencies that elicit zero system response, are represented by the roots of numerator polynomial. The poles of the transfer function, i.e., those frequencies where the system response is undefined, are represented by the roots of denominator polynomial.

The system impulse response, i.e., its response to a unit-impulse input, contains the natural modes of system response. The natural response includes terms of the form $e^{p_i t}$, where p_i is a pole of the transfer function. The natural response of a stable system dies out with time.

The system step response, i.e., its response to a unit-step input, comprises both natural and forced responses, where the forced response is a constant value. Once system's natural response dies out, the output reaches a steady-state. The dc gain of the system denotes its gain to a constant input.

System stability refers to the system being well-behaved and predictable under various operating conditions. The bounded-input bounded-output (BIBO) stability refers to the system response staying finite to every finite input, i.e., $|y(t)| < N < \infty$ if $|u(t)| < M < \infty$. The BIBO stability requires that the poles of the system transfer function are located in the open left-half of the complex s-plane.

The frequency response function of a system, obtained by substituting $s = j\omega$ in the transfer function, characterizes its response to sinusoidal inputs in the steady-state, which is a sinusoid at the input frequency. Further, the magnitude of the response is scaled by the gain of the system transfer function evaluated at the input frequency, and it has a phase contribution from the system transfer function. The frequency response function can be visualized on a Bode plot.

2.1: System Poles and Zeros

System Poles and Zeros

The transfer function, $G(s)$, is a rational function in the Laplace transform variable, s. It is expressed as the ratio of the numerator and the denominator polynomials, i.e., $G(s) = \frac{n(s)}{d(s)}$.

Definition: Transfer Function Zeros

The roots of the numerator polynomial, $n(s)$, define system zeros, i.e., those frequencies at which the system response is zero. Thus, z_0 is a zero of the transfer function if $G(z_0) = 0$.

Definition: Transfer Function Poles

The roots of the denominator polynomial, $d(s)$, define system poles, i.e., those frequencies at which the system response is infinite. Thus, p_0 is a pole of the transfer function if $G(p_0) = \infty$.

The poles and zeros of first and second-order system models are described below.

First-Order System

A first-order system has a generic ODE description: $\tau \dot{y}(t) + y(t) = u(t)$, where $u(t)$ and $y(t)$ denote the input and the output, and τ is the system time constant. By applying the Laplace transform, a first-order transfer function is obtained as:

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

The transfer function has no finite zeros and a single pole located at $s = -\frac{1}{\tau}$ in the complex plane.

Example 2.1.1

The reduced-order model of a DC motor with voltage input and angular velocity output (Example 1.4.3) is described by the differential equation: $\tau \dot{\omega}(t) + \omega(t) = V_a(t)$.

The DC motor has a transfer function: $G(s) = \frac{K}{\tau_m s + 1}$ where τ_m is the motor time constant.

For the following parameter values: $R = 1\Omega,\ L = 0.01H,\ J = 0.01\ kgm^2,\ b = 0.1\ \frac{N-s}{rad},\ and\ k_t = k_b = 0.05$, the motor transfer function evaluates as:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{5}{s + 10.25} = \frac{0.49}{0.098s + 1}$$

The transfer function has a single pole located at: $s = -10.25$ with associated time constant of $0.098sec$.

Second-Order System with an Integrator

A first-order system with an integrator is described by the transfer function:

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

The system has no finite zeros and has two poles located at $s = 0$ and $s = -\frac{1}{\tau}$ in the complex plane.

Example 2.1.2

The DC motor modeled in Example 2.1.1 above is used in a position control system where the objective is to maintain a certain shaft angle $\theta(t)$. The motor equation is given as: $\tau \ddot{\theta}(t) + \dot{\theta}(t) = V_a(t)$; its transfer function is given as: $G(s) = \frac{K}{s(\tau s + 1)}$.

Using the above parameter values in the reduced-order DC motor model, the system transfer function is given as:

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{5}{s(s+10.25)} = \frac{0.49}{s(0.098s+1)}$$

The transfer function has no finite zeros and poles are located at: $s = 0, -10.25$.

Second-Order System with Real Poles

A second-order system with poles located at $s = -\sigma_1, \ -\sigma_2$ is described by the transfer function:

$$G(s) = \frac{1}{(s+\sigma_1)(s+\sigma_2)}$$

Example 2.1.3

From Section 1.4, the DC motor transfer function is described as:

$$G(s) = \frac{K}{(s+1/\tau_e)(s+1/\tau_m)}$$

Then, system poles are located at: $s_1 = -\frac{1}{\tau_m}$ and $s_2 = -\frac{1}{\tau_e}$, where τ_e and τ_m represent the electrical and mechanical time constants of the motor.

For the following parameter values: $R = 1\Omega, \ L = 0.01H, \ J = 0.01 \ kgm^2, \ b = 0.1 \ \frac{N-s}{rad}, \ and \ k_t = k_b = 0.05$, the transfer function from armature voltage to angular velocity is given as:

$$\frac{\omega(s)}{V_a(s)} = \frac{500}{(s+100)(s+10)+25} = \frac{500}{(s+10.28)(s+99.72)}$$

The transfer function poles are located at: $s = -10.28, -99.72$

The motor time constants are given as: $\tau_e \cong \frac{L}{R} = 10 \ ms, \ \tau_m \cong \frac{J}{b} = 100 \ ms$.

Second-Order System with Complex Poles

A second-order model with its complex poles located at: $s = -\sigma \pm j\omega$ is described by the transfer function:

$$G(s) = \frac{K}{(s+\sigma)^2 + \omega^2}.$$

Equivalently, the second-order transfer function with complex poles is expressed in terms of the damping ratio, ζ, and the natural frequency, ω_n, of the complex poles as:

$$G(s) = \frac{K}{(s+\zeta\omega_n)^2 + \omega_n^2(1-\zeta^2)}$$

The transfer function poles are located at: $s_{1,2} = -\zeta\omega_n \pm j\omega_d$, where $\omega_d = \omega_n\sqrt{1-\zeta^2}$ (Figure 2.1.1).

As seen from the figure, ω_n equals the magnitude of the complex pole, and $\zeta = \frac{\sigma}{\omega_n} = \cos\theta$, where θ is the angle subtended by the complex pole at the origin.

The damping ratio, ζ, is a dimensionless quantity that characterizes the decay of the oscillations in the system's natural response. The damping ratio is bounded as: $0 < \zeta < 1$.

1. As $\zeta \to 0$, the complex poles are located close to the imaginary axis at: $s \cong \pm j\omega_n$. The resulting impulse response displays persistent oscillations at system's natural frequency, ω_n.
2. As $\zeta \to 1$, the complex poles are located close to the real axis as $s_{1,2} \cong -\zeta\omega_n$. The resulting impulse response has no oscillations and exponentially decays to zero resembling the response of a first-order system.

Example 2.1.4

A spring–mass–damper system has a transfer function:

$$G(s) = \frac{1}{ms^2 + bs + k}.$$

Its characteristic equation is given as: $ms^s + bs + k = 0$, whose roots are characterized by the sign of the discriminant, $\Delta = b^2 - 4mk$.

Specifically,

1. For $\Delta > 0$, the system has real poles, located at:

   $s_{1,2} = -\frac{b}{2m} \pm \sqrt{\left(\frac{b}{2m}\right)^2 - \frac{k}{m}}$.

2. For $\Delta < 0$, the system has complex poles, located at:

   $s_{1,2} = -\frac{b}{2m} \pm j\sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2}$.

3. For $\Delta = 0$, the system has two real and equal poles, located at:

   $s_{1,2} = -\frac{b}{2m}$.

Next, assume that the mass-spring-damper has the following parameter values: $m = 1, b = k = 2$; then, its transfer function is given as:

$$G(s) = \frac{1}{ms^2 + bs + k} = \frac{1}{s^2 + 2s + 2}$$

The transfer function has complex poles located at: $s = -1 \pm j1$.

The complex poles have: $\omega_n = \sqrt{2}\frac{rad}{s}, \zeta = \frac{1}{\sqrt{2}}$.

Further, the complex poles have an angle: $\theta = 45^\circ$, and $\cos 45^\circ = \frac{1}{\sqrt{2}}$.

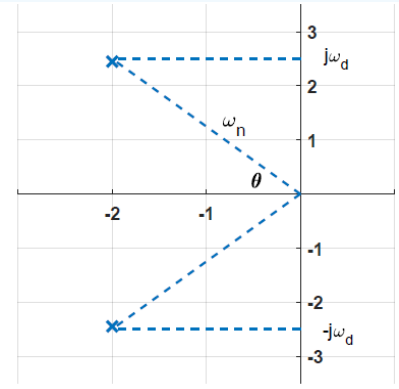


Figure 2.1.1: Second-order transfer function pole locations in the complex plane.

2.2: System Natural Response

System Natural Response

The transfer function of a dynamic linear time-invariant (LTI) system is given as a ratio of polynomials: $G(s)=\frac{n(s)}{d(s)}$. The poles of the transfer function are the roots of the denominator polynomial $d(s)$.

The poles of the transfer function characterize the natural response modes of the system. Thus, if p_i is a pole of the transfer function, $G\left(s\right)$, then $\left\{e^{p_it}\right\}$ constitutes a natural response mode.

- A real pole: $p_i=-\sigma$, contributes a term $e^{-\sigma t}$ to system natural response.
- A pair of complex poles: $p_i=-\sigma \pm j\omega$, contributes oscillatory terms of the form $e^{-\sigma t}e^{j\omega t}=e^{-\sigma t}\left(\cos \omega t +j \sin \omega t\right)$ to the natural response.

The natural response of the system is a weighted sum of the natural response modes, i.e., $y_n(t)=\sum_{i=1}^n C_i e^{p_it}$. Further, the natural response is reflected in the impulse response of a system.

Definition: Impulse Response

The impulse response of a system, represented by $G(s)$, is defined as system response to a unit-impulse input, $\delta \left(t\right)$, when the initial conditions are zero.

Let $u\left(t\right)=\delta \left(t\right),\ \ u\left(s\right)=1$; then, the impulse response is computed as: $y\left(s\right)=G(s)$; in the time-domain, the impulse response is given as: $g(t)={\rm L}^{-1} \left[G(s)\right]$.

To proceed further, let the system transfer function be represented in the factored form as:

$$G\left(s\right)=\frac{n(s)}{\prod^n_{i=1}{\left(s-p_i\right)}} \nonumber$$

where $n(s)$ is the numerator polynomial, and $p_i,\ i=1,\dots n$, are the system poles, assumed to be distinct and may include a single pole at the origin. Using partial fraction expansion (PFE), the impulse response is given as:

$$y_{imp}\left(s\right)=\sum^n_i{\frac{A_i}{s-p_i}} \nonumber$$

Using the inverse Laplace transform, the impulse response of the system is computed as:

$$y_{imp}\left(t\right)=\left(\sum^n_i{A_ie^{p_it}}\right)u(t) \nonumber$$

where $u(t)$ represents the unit-step function, used here to indicate that the expression for $g(t)$ is valid for $t\ge 0$.

Impulse Response of Low Order Systems

First-Order System

Let $G(s)=\frac{1}{\tau s+1}$; the system has a signle response mode given as: $\{e^{-t/\tau }\}$. Thus, the impulse response of the first-order system is computed as: $g(t)=\frac{1}{\tau }e^{-t/\tau } u(t) \nonumber$

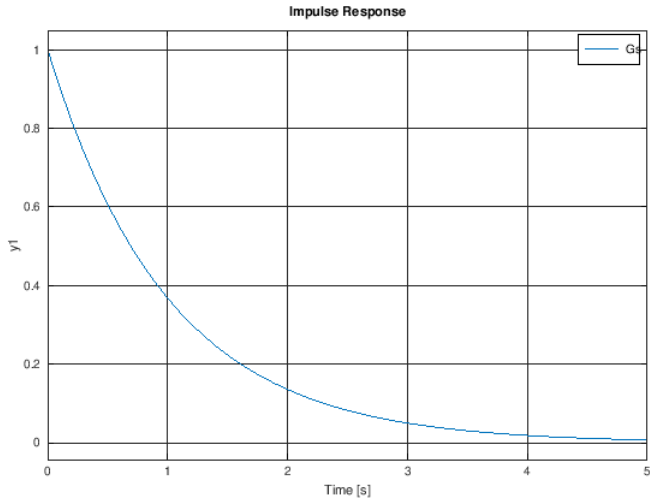
Example $\PageIndex{1}$

The impulse response of $G(s)=\frac{1}{s+1}$ is given as: $g(t)=e^{-t}u(t)$.

The impulse response begins at $g(0)=1$ and asymptotically approaches $g(\infty)=0$.

```
pkg load control
s=tf('s');
Gs=1/(s+1);
impulse(Gs), grid on
```

run | restart | restart & run all



Definition: System Time Constant

The time constant, \(\tau\), describes the time when, starting from unity, the natural response decays to \(e^{-1}\cong 0.37\), or 37% of its initial value. For a first-order system, with a real pole, \(s=-\sigma\), the time constant is given as: \(\tau =\frac{1}{\sigma }\).

Second-Order System with an Integrator

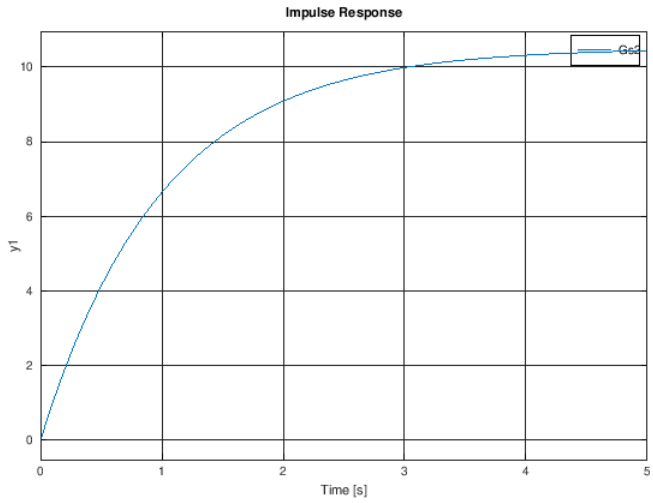
Let: \(G(s)=\frac{1}{s\left(\tau s+1\right)}\); then, the natural response modes are: \(\left\{1,\ e^{-t/\tau }\right\}\). The transfer function is expanded using PFE as: \(G(s)=\frac{1}{s} +\frac{\tau }{\tau s+1}\). The impulse response is expressed as: \[g(t)=\left(1-e^{-t/\tau } \right)\, u(t) \nonumber \]

Example \(\PageIndex{2}\)

The impulse response of \(G(s)=\frac{1}{s(s+1)}=\frac{1}{s}-\frac{1}{s+1}\) is given as: \(g(t)=(1-e^{-t}) u(t)\).
The impulse response begins at \(g(0)=0\) and asymptotically approaches \(g(\infty)=1\).

```
pkg load control
s=tf('s');
Gs=1/s/(s+1);
impulse(Gs), grid on
```

run | restart | restart & run all



Second-Order System with Real Poles

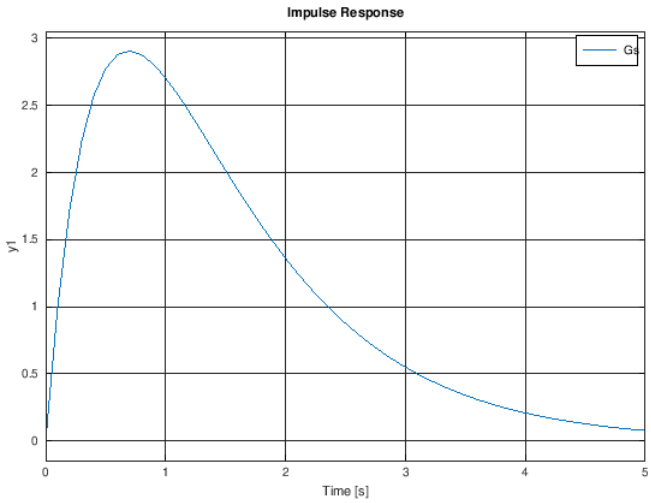
Let $G\left(s\right)=\frac{1}{\left(s+\sigma_1\right)\left(s+\sigma_2\right)}$; then, the system natural response modes are: $\left\{e^{-\sigma_1t},\ e^{-\sigma_2t}\right\}$. Assuming $\sigma_2 \ge \sigma_1$, and using PFE followed by inverse Laplace transform, the system impulse response is expressed as: $$g(t)=\frac{1}{\sigma_2-\sigma_1}\left(e^{-\sigma_1t} -e^{-\sigma_2t} \right)\, u(t) \nonumber$$

Example \(\PageIndex{3}\)

The impulse response of $G(s)=\frac{1}{(s+1)(s+2)}$ is given as: $g(t)=(e^{-t}-e^{-2t}) u(t)$.
The impulse response begins at $g(0)=0$ and asymptotically approaches $g(\infty)=0$.

```
pkg load control
s=tf('s');
Gs=1/(s+1)/(s+2);
impulse(Gs), grid on
```

run | restart | restart & run all



Second-Order System with Complex Poles

Let $G\left(s\right)=\frac{1}{{\left(s+\sigma \right)}^2+{\omega}^2}$; then, by using Euler's identity, its natural response modes are given as: $\left\{e^{-\sigma t}{\cos \omega t\ },\ \ e^{-\sigma t}{\sin \omega t\ }\right\}$. Its impulse response is given as: $$g(t)=\frac{1}{\omega } e^{-\sigma t} \; \sin (\omega t)\, u(t) \nonumber$$

The oscillatory natural response is contained in the envelope defined by: $\pm e^{-\sigma t}$. The effective time constant of a second-order system is given as: ${\tau }_{eff}=\frac{1}{\sigma }$.

The natural response is of the form: $y_n\left(t\right)=(C_1{\cos \omega t\ }+C_2{\sin \omega t\ })e^{-\sigma t}$ can be alternatively expressed as: $$y_n\left(t\right)=Ce^{-\sigma t}{sin \left(\omega t+\phi \right)\ } \nonumber$$ where $C=\sqrt{C^2_1+C^2_2}$ and ${tan \phi \ }=\frac{C_1}{C_2}$.

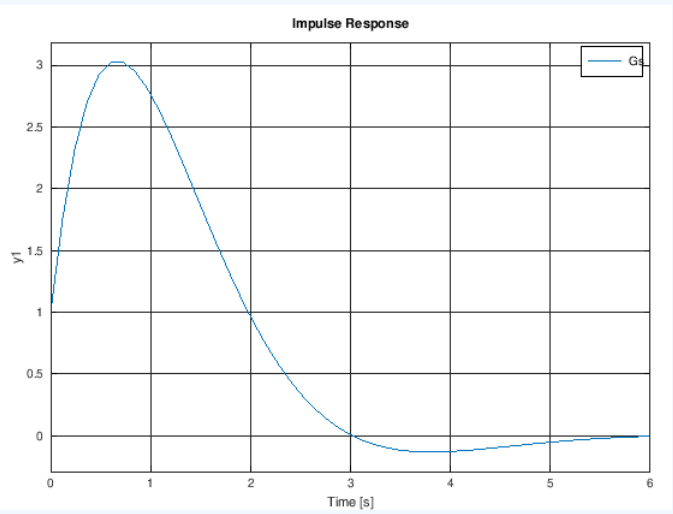
Example \(\PageIndex{4}\)

The impulse response of $G(s)=\frac{s}{s^2+2s+2}=\frac{s}{(s+1)^2+1^2}=\frac{s+1-1}{(s+1)^2+1^2}$ is given as: $$g(t)= (\cos t-\sin t)e^{-t} u(t)=\sqrt{2}\sin (t+135^{\circ}) u(t)=\sqrt{2}\cos (t+45^{\circ})u(t) \nonumber$$

The impulse response begins at $g(0)=\cos(0)=1$ and asymptotically approaches $g(\infty)=0$.

```
pkg load control
s=tf('s');
Gs=s/(s^2+2*s+2);
impulse(Gs), grid on
```

run restart restart & run all



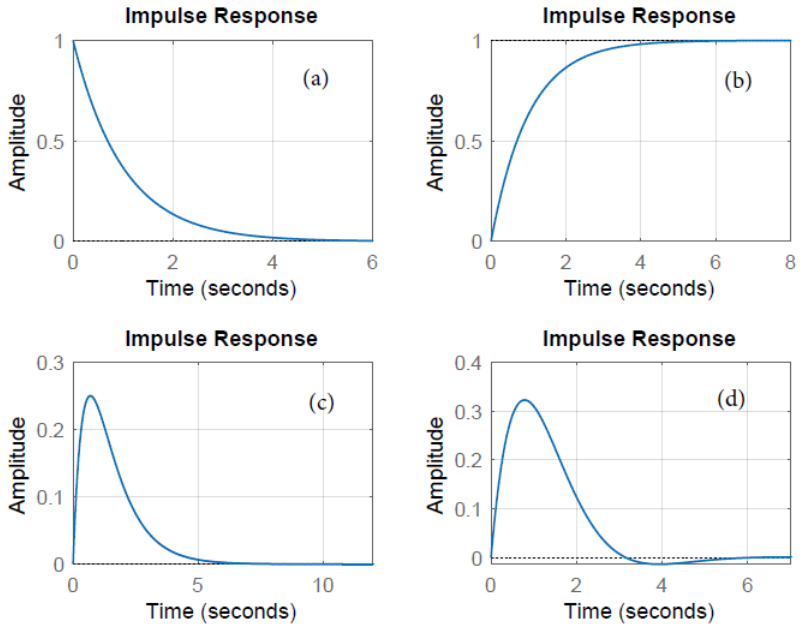


Figure \(\PageIndex{1}\): Impulse response of transfer function models: (a) first-order system; (b) second-order system with a pole at the origin; (c) second-order system with real poles; and, (d) second-order system with complex poles.

From the figure, we may observe that:

1. While the impulse response of a first-order system starts from a value of unity, the impulse response of a second-order system starts from zero.
2. The impulse response of a stable system with poles in the open left-half plane (OLHP), \(Re\left(p_i\right)<0\), asymptotically dies out with time, i.e., \({\mathop{lim}_{t\to \infty } g\left(t\right)=0\ }\).
3. ; the impulse response of a system with a pole at the origin approaches a constant value of unity in the steady-state (which represents the integral of the delta function). Such systems are termed marginally stable.

2.3: System Stability

Stability is a desired characteristic of any dynamic system; it refers to the system being well behaved and in control under various operating conditions. Stability may be categorized in multiple ways, some of which are discussed below.

Bounded-Input Bounded-Output Stability (BIBO)

The bounded-input bounded-output (BIBO) stability implies that for every bounded input, $u(t)\ :\ |u(t)| < M_1 < \infty,$ the output of the system stays bounded, that is, $y(t)\ :\ |y(t)| < M_2 < \infty$.

The output of the system in time-domain is given in terms of the convolution integral:

$$y(t) = \int_0^\infty g(t-\tau)u(\tau)d\tau$$

where $g(t)$ is the impulse response of the system. Hence, a necessary condition for BIBO stability is that the impulse response dies out with time, that is, $\lim_{t\to\infty} g(t) = 0$.

The impulse response contains the modes of system natural response and is given as:

$$g(t) = \sum_{i=1}^{n} A_i e^{p_i t}$$

where p_i is a pole of the system transfer function. Hence, a necessary condition for BIBO stability is: $\mathrm{Re}\,[p_i] < 0$.

Physically, the condition $\mathrm{Re}[p_i] < 0$ implies the presence of damping in the system, where the damping terms in the transfer function indicate dissipation of residual energy with time.

Marginal Stability

The imaginary axis on the complex plane serves as the stability boundary. A system with poles in the open left-half plane (OLHP) is stable.

If the system transfer function has simple poles that are located on the imaginary axis, it is termed as marginally stable. The impulse response of such systems does not go to zero as $t \to \infty$, but stays bounded in the steady-state.

As an example, a simple harmonic oscillator is described by the ODE: $\ddot{y} + \omega_n^2 y = 0$, where ω_n represents the natural frequency and the system has no damping. The oscillator transfer function, $G(s) = \frac{1}{s^2+\omega_n^2}$ has simple poles $(p_{1,2} = \pm j\omega_n)$ on the $j\omega$-axis.

The natural response of the simple harmonic oscillator contains the response mode: $e^{j\omega_n t} = \cos\omega_n t + j\sin\omega_n t$. Its impulse response displays persisting oscillations at the natural frequency.

Internal Stability

The notion of internal stability requires that all signals within a control system remain bounded for every bounded input. It further implies that all relevant transfer functions between input–output pairs in a feedback control system are BIBO stable.

Internal stability is a stronger notion than BIBO stability. It is so because the internal modes of system response may include those modes not be reflected in the input-output transfer function.

In the case of linear system models involving feedback, the internal stability requirements are met if the closed-loop characteristic polynomial is stable and any pole-zero cancelations appearing in the loop gains are restricted to the OLHP.

In particular, for a single-input single-output (SISO) feedback control system, the loop gain includes the product of the plant and the controller transfer functions.

Suppose the plant transfer function is: $G(s) = \frac{1}{s+1}$, and the controller is given as: $K(s) = K\left(\frac{s+1}{s+10}\right)$; then, $K(s)G(s) = \frac{K(s+1)}{(s+1)(s+10)}$, which includes an OLHP pole-zero cancelation. However, the closed-loop characteristic polynomial: $\Delta(s) = s + 10 + K$ has stable roots for $K > -10$. Hence, the closed-loop system is internally stable for $K > -10$.

Asymptotic Stability

For a general nonlinear system model, $\dot{x}(t) = f(x, u)$, stability refers to the stability of an equilibrium point (x_e, u_e) defined by: $f(x_e, u_e) = 0$.

In particular, the equilibrium point is said to be stable if a system trajectory, $x(t)$, that starts in the vicinity of x_e stays close to x_e. The equilibrium point is said to be asymptotically stable if a system trajectory $x(t)$ that starts in the vicinity of x_e converges to x_e.

For linear system models, defined by: $\dot{x} = Ax(t) + Bu(t)$, the origin $x_e = 0$ serves as an equilibrium point. In such cases, the asymptotic stability requires that the poles of the system transfer function (equivalently, the eigenvalues of the system matrix) lie in the open left-half plane, or $\mathrm{Re}[p_i] < 0$, and there are no RHP pole-zero cancelations.

2.4: The Step Response

Step Response

The impulse and step inputs are among prototype inputs used to characterize the response of the systems. The unit-step input is defined as:

$$u(x)=\begin{cases}0, &x<0 \\ 1,& x\ge 0\end{cases}$$

Definition: Step Response

The step response of a system is defined as its response to a unit-step input, $u(t)$, or $u(s)=\frac{1}{s}$.

Let $G(s)$ describe the system transfer function; then, the unit-step response is obtained as: $y(s)=G(s)\frac{1}{s}$.
Its inverse Laplace transform leads to: $y(t)={\mathcal{L}}^{-1}\left[\frac{G(s)}{s}\right]$.
Alternatively, the step response can be obtained by integrating the impulse response: $y(t)=\int_0^t g(t-\tau)d\tau$.
The unit-step response of a stable system starts from some initial value: $y(0)=y_0$, and settles at a steady-state value: $y_{\infty }={\lim_{t\to \infty } y(t)}$.
Further, from the application of the final value theorem (FVT): $y_{\infty }=G{(s)|}_{s=0}$.

Step Response of Low Order Systems

The unit-step response in the case of the first- and second-order systems is described below.

First-Order System

Let $G(s)=\frac{K}{\tau s+1},\; u(s)=\frac{1}{s}$; then, $y(s)=\frac{K}{s(\tau s+1)} =\frac{K}{s} -\frac{K\tau }{\tau s+1}$.
The time-domain response is given as: $y(t)=K(1-e^{-t/\tau })$, $u(t)$.
Assuming arbitrary initial conditions, $y(0)=y_0$, the step response of a first-order system is given as:

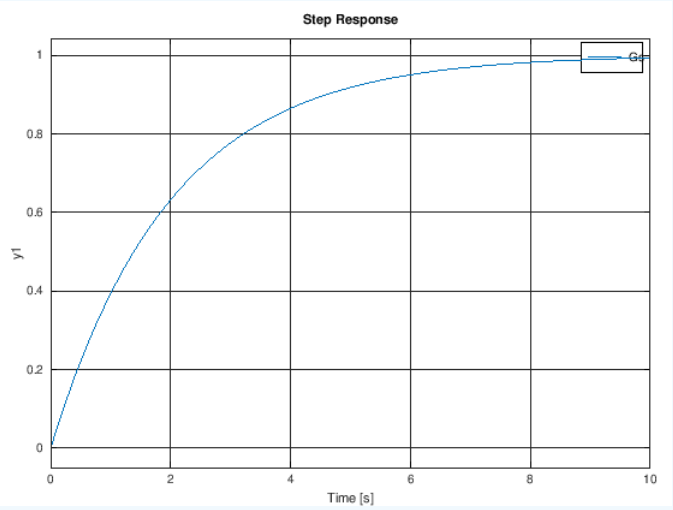
$$y(t)=y_{\infty }+(y_0-y_{\infty })e^{-t/\tau },\ t\ge 0$$

Example \(\PageIndex{1}\)

Let $G(s)=\frac{1}{2s+1}$; then, the unit-step response is obtained as: $y(s)=\frac{1}{s(2s+1)} =\frac{1}{s} -\frac{2}{2s+1}$.
The time-domain response is given as: $y(t)=(1-e^{-t/2 })$, $u(t)$.

```
pkg load control
s=tf('s');
Gs=1/(2*s+1);
step(Gs), grid on
```

run | restart | restart & run all



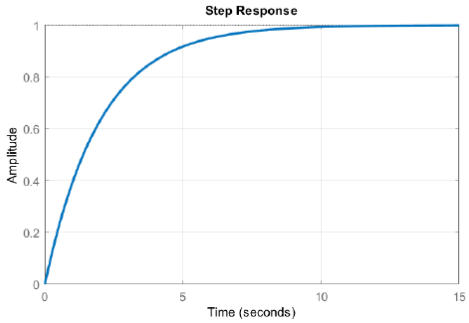


Figure \(\PageIndex{1}\): Step response of first-order DC motor model.

First-Order System with Integrator

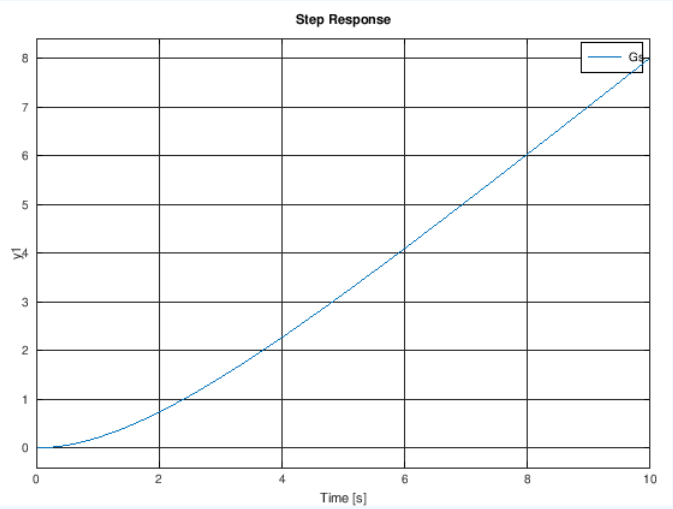
Let \(G(s)=\frac{K}{s\left(\tau s+1\right)}, \;\;u(s)=\frac{1}{s}\); then, \(y(s)=\frac{K}{s^{2} \left(\tau s+1\right)}\).
Using PFE, we obtain: \(y(s)=K\left(\frac{1}{s^{2} } -\frac{\tau }{s} +\frac{\tau ^{2} }{\tau s+1} \right)\). Hence, \[g(t)=K\left(t-\tau \left(1-e^{-t/\tau } \right)\right)\, u(t) \nonumber \] The step response grows out of bound as \(t\to \infty\).

Example \(\PageIndex{2}\)

Let \(G(s)=\frac{1}{s(2s+1)}\); then, the unit-step response is computed as: \(y(s)=\frac{1}{s^2(2s+1)}=\frac{1}{s^2}-\frac{2}{s} +\frac{4}{2s+1}\).
The time-domain response is given as: \(y(t)=(t-2+2e^{-t/2 })\, u(t)\).

```
pkg load control
s=tf('s');
Gs=1/s/(2*s+1);
step(Gs), grid on
```

run | restart | restart & run all



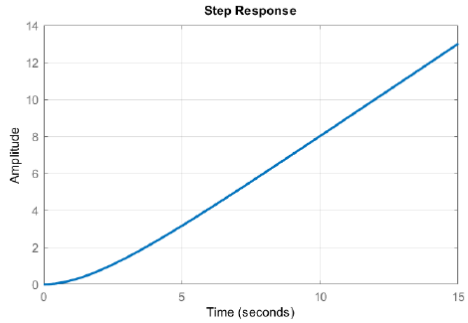


Figure \(\PageIndex{2}\): Step-response of a second-order model with a pole at the origin.

Second-Order System with Real Poles

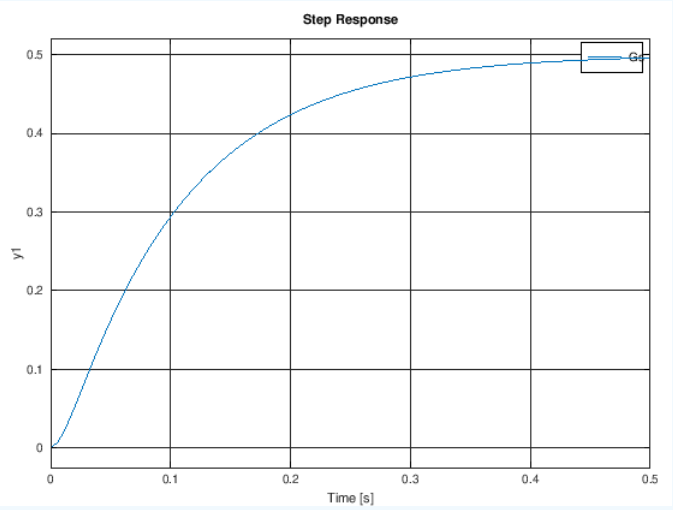
Let \(G(s)=\frac{K}{(\tau _{1} s+1)(\tau _{2} s+1)}\), \({\tau }_1>{\tau }_2\); then, the step response is computed as: \(y(s)=\frac{K}{s(\tau _{1} s+1)(\tau _{2} s+1)} =\frac{A}{s} +\frac{B}{\tau _{1} s+1} +\frac{C}{\tau _{2} s+1}\). Hence, \[y(t)=(A+Be^{-t/\tau _{1} } +Ce^{-t/\tau _{2} })u(t) \nonumber \]
where \(A=K,B=-\frac{K{\tau }^2_1}{{\tau }_1-{\tau }_2},\ C=\frac{K{\tau }^2_2}{{\tau }_2-{\tau }_1}\).

Example \(\PageIndex{3}\)

A small DC motor has the following parameter values: \(R=1\; \Omega ,\; L=10\; {mH},\; J=0.01\; {kg-m}^{2} ,b=0.1\; \frac{N-s}{rad} ,\; k_{t} =k_{b} =0.05\).
The motor transfer function, from armature voltage to motor speed, is approximated as: \(G\left(s\right)=\frac{500}{\left(s+10\right)\left(s+100\right)}\).
The step response of the motor is obtained as: \(y\left(s\right)=\frac{1}{2s}-\frac{0.556}{s+10}+\frac{0.056}{s+100}\)
The time-domain response is given as: \(y\left(t\right)=\left(0.5-0.556e^{-10t}+0.056e^{-100t}\right)u(t)\), which settles at \(y_{\infty }=0.5\).
The natural response modes, \(e^{-10t}\) and \(e^{-100t}\), reflect motor electrical and mechanical time constants: \({\tau }_e\cong 0.01s\) and \({\tau }_m\cong 0.1s\). The slower mechanical time constant dominates the motor step response.

```
pkg load control
s=tf('s');
Gs=500/(s+10)/(s+100);
step(Gs), grid on
```

run | restart | restart & run all



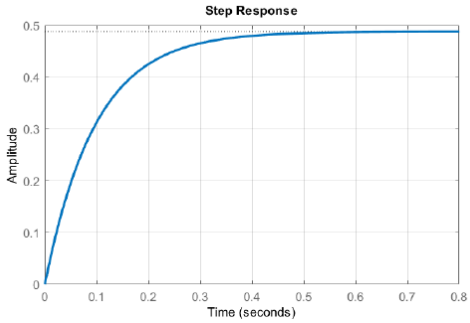


Figure \(\PageIndex{3}\): Step response of DC motor model with real poles.

Second-Order System with Complex Poles

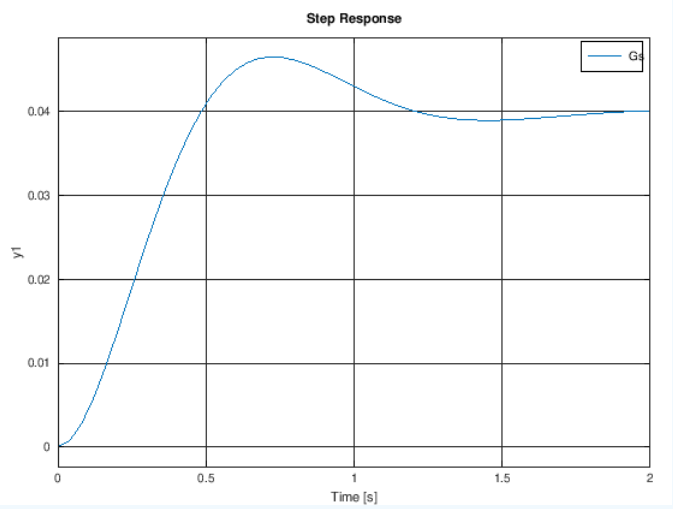
Let \(G(s)=\frac{K}{(s+\sigma)^{2} +\omega _{d}^{2} }\). Then, the unit-step response is computed as: \(y(s)=\frac{A}{s} +\frac{Bs+C}{(s+\sigma)^{2} +\omega _{d}^{2} }\),
where \(A=G(0)=\frac{K}{\sigma ^{2} +\omega _{d}^{2} } ,\, \, B=-A,\, \, C=-2A\sigma\). Hence,
\[y(t)=\frac{K}{\sigma ^{2} +\omega _{d}^{2} } \left[1-e^{-\sigma t} \left(\cos \; \omega _{d} t+\frac{\sigma }{\omega _{d} } \sin \; \omega _{d} t\right)\right]u(t) \nonumber \]
In the phase form, the unit-step response is given as:
\[y(t)=\frac{K}{\sigma ^{2} +\omega _{d}^{2} } \left[1-e^{-\sigma t} \cos \; \left(\omega _{d} t-\phi \right)\right]\, u(t),\, \, \, \, \phi =\tan ^{-1} \frac{\sigma }{\omega _{d} } \nonumber \]

Example \(\PageIndex{4}\)

A mass–spring–damper system has the following parameter values: \(m=1\ kg,\ b=6\frac{Ns}{m},\ k=25\frac{N}{m}\).
Its transfer function is given as: \(G\left(s\right)=\frac{1}{s^2+5s+25}=\frac{1}{{\left(s+3\right)}^2+4^2}\). The system has complex poles located at: \(s=-3\pm j4\).
The unit-step response of the system is computed as: \(y\left(s\right)=\frac{1}{25}\left[\frac{1}{s}-\frac{s+6}{{\left(s+3\right)}^2+4^2}\right]\).
By applying the inverse Laplace transform, the time-domain response is obtained as:
\[y\left(t\right)=\frac{1}{25}\left[1-e^{-3t}\left({cos \left(4t\right)\ }+\frac{3}{4}{sin \left(4t\right)\ }\right)\right]u(t) \nonumber \]

```
pkg load control
s=tf('s');
Gs=1/(s^2+5*s+25);
step(Gs), grid on
```

run | restart | restart & run all



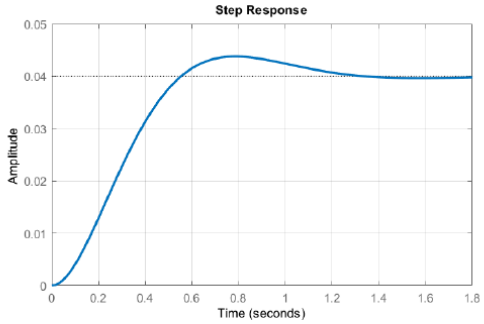


Figure \(\PageIndex{4}\): Step response of mass–spring–damper model with complex poles.

System with Dead-time

The first-order-plus-dead-time (FOPDT) model of an industrial process is given as: \(G(s)=\frac{Ke^{-t_ds} }{\tau s+1}\), where \(\{K,\tau,t_d\}\) represent the process gain, time constant, and dead-time.

The step response of the FOPDT model is computed as: \(G(s)=\frac{Ke^{-t_ds} }{s(\tau s+1)}\), which translated into a time-domain response as: \(g(t)=(1-e^{-(t-t_d)/\tau})u(t-t_d)\).

An approximate process model, obtained by using Pade' approximation, is given as: \(G(s)=\frac{K(1-t_ds/2)}{(1+t_ds/2)(\tau s+1)}\). This model is termed as non-minimum phase due to the additional phase contributed by the right half-plane (RHP) zero.

The unit-step response of the approximate model is computed as: \(y(s)=\frac{K(1-t_ds/2)}{s(1+t_ds/2)(\tau s+1)}\).

Example \(\PageIndex{5}\)

The FOPDT model of a stirred-tank bio-reactor is given as: \(G(s)=\frac{20e^{-s} }{0.5s+1}\).

The step response of the system is computed as: \(y(s)=\frac{20e^{-s} }{s\left(0.5s+1\right)} \).

The time-domain response is obtained as: \(y(t)=20\left(1-e^{-2(t-1)} \right)\, u(t-1)\).

Using a first-order Pade' approximation, an approximate process model is obtained as: \(G_{a} (s)=\frac{20\left(1-0.5s\right)}{\left(0.5s+1\right)^{2} }\).

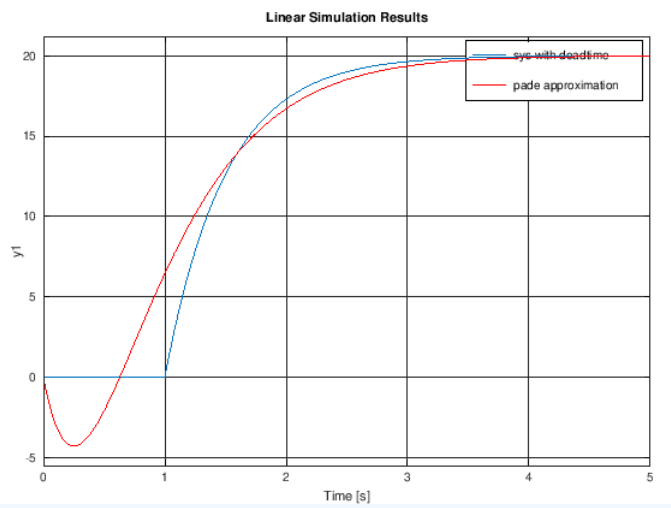
The step response of the approximate model is computed as: \(y(s)=\frac{20\left(1-0.5s\right)}{s\left(0.5s+1\right)^{2} } \), \(y(t)=20\left(1-(1-4t)e^{-2t} \right)\, u(t)\).

The two responses are compared below (Figure 2.4.5). The step response for the FOPDT model starts after the designated delay. The step response for the Pade' approximation starts with an undershoot due to the presence of RHP zero.

```
pkg load control
s=tf('s');
```

```
Gs=20/(.5*s+1);
t=0:.01:5;
u=ones(size(t));
u(1:100)=0;
lsim(Gs,u,t), grid on
hold
Gsa=20*(1-.5*s)/(1+.5*s)^2;
%step(Gsa)
lsim(Gsa,ones(size(t)),t,'r')
legend('sys with deadtime','pade approximation')
```

run | restart | restart & run all



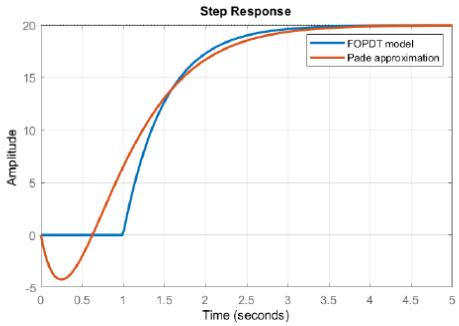


Figure \(\PageIndex{5}\): Step response of an industrial process model with dead-time.

We make the following observations based on the figure:

1. The step response of the process with dead-time starts after 1 s delay (as expected).
2. The step response of Pade' approximation of delay has an undershoot. This behavior is characteristic of transfer function models with zeros located in the right-half plane.

2.5: Sinusoidal Response of a System

Response to Sinusoidal Input

The sinusoidal response of a system refers to its response to a sinusoidal input: $u(t) = \cos\ \omega_0 t$ or $u(t) = \sin\ \omega_0 t$.

To characterize the sinusoidal response, we may assume a complex exponential input of the form: $u\left(t\right) = e^{j\omega_0 t},\ \ u\left(s\right) = \frac{1}{s - j\omega_0}$.

Then, the system output is given as: $y\left(s\right) = \frac{G(s)}{s - j\omega_0}$.

To proceed, let $G\left(s\right) = \frac{n(s)}{\prod_{i=1}^{n}\left(s - p_i\right)}$, where $p_i,\ i = 1, \ldots n$, denote the poles of the transfer function; then, using PFE, the system response is given as:

$$y\left(s\right) = \frac{A_0}{s - j\omega_0} + \sum_{i=1}^{n} \frac{A_i}{s - p_i}$$

where $A_0 = G\left(j\omega_0\right)$. The time response of the system is given as:

$$y\left(t\right) = G\left(j\omega_0\right) e^{j\omega_0 t} + \sum_{i=1}^{n} A_i e^{p_i t}$$

Assuming BIBO stability, i.e., $Re\left[p_i\right] < 0$, the transient response component dies out with time. Then, the steady-state response is described as:

$$y_{ss}\left(t\right) = G\left(j\omega_0\right) e^{j\omega_0 t}$$

Definition: Frequency Response Function

The frequency response function, for a given transfer function $G(s)$ is defined as: $G(j\omega) = G(s)|_{s=j\omega}$.

The frequency response function is described in the magnitude-phase form as: $G(j\omega) = |G(j\omega)|\, e^{\angle G(j\omega)}$.

When computed at a particular frequency, $\omega = \omega_0$, the frequency response function, $G\left(j\omega_0\right)$ is a complex number.

Then, the steady-state response to a complex sinusoid $u(t) = e^{j\omega_0 t}$ is given as:

$$y_{ss}\left(t\right) = |G(j\omega_0)|\, e^{\angle G(j\omega_0)} e^{j\omega_0 t}$$

By Euler's identity, we have: $e^{j\omega_0 t} = \cos(\omega_0 t) + j\ \sin(\omega_0 t)$; therefore, the steady-state response to $u(t) = \cos\ \omega_0 t$ is given as:

$$y_{\text{ss}}(t) = |G(j\omega_0)| \cos(\omega_0 t + \angle G(j\omega_0))$$

Similarly, the steady-state response to $u(t) = \sin\ \omega_0 t$ is given as:

$$y_{\text{ss}}(t) = |G(j\omega_0)|\ \sin(\omega_0 t + \angle G(j\omega_0))$$

Thus, the steady-state response to sinusoid of a certain frequency is a sinusoid at the same frequency, scaled by the magnitude of the frequency response function; the response includes a phase contribution from the frequency response function.

Examples

Example 2.5.1

A small DC motor model is approximated as: $G\left(s\right) = \frac{500}{(s+10)(s+100)}$.

Its frequency response function is given as: $G\left(j\omega\right) = \frac{0.5}{(0.01\omega+1)(0.1\omega+1)}$.

Let $u(t) = \sin 10t$; then, in the steady-state, the sinusoidal response is given as: $y_{ss}\ \left(t\right) = 0.35\ \sin(10t - 50^\circ)$.

The sinusoidal response of the DC motor is plotted in Figure 2.5.1.

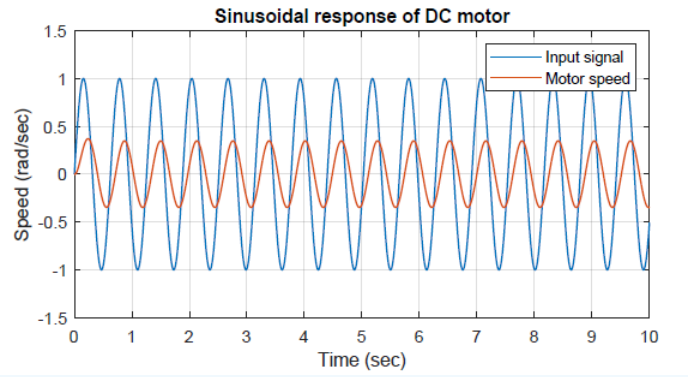


Figure 2.5.1: Sinusoidal response of the DC motor model.

Example 2.5.2

The model of a mass–spring–damper system is given as: $G(s) = \frac{1}{s^2+2s+5} = \frac{1}{(s+1)^2+2^2}$; the complex poles are located at: $s = -1 \pm j2$.

Let $u(t) = \sin\ \pi t$; then, in the steady-state, we have: $y_{ss}\ (t) = 0.126\ \sin(\pi t - 128^\circ)$.

The response is plotted below and includes the transient response, which dies out in 4-5 sec.

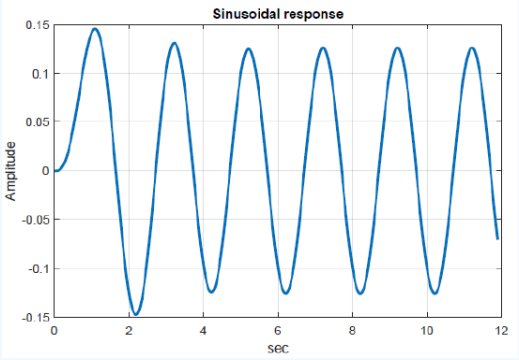


Figure 2.5.2: Sinusoidal response of mass-spring-damper system.

Visualizing the Frequency Response

The frequency response function is given in polar form as: $G(j\omega) = |G(j\omega)|e^{j\phi(\omega)}$.

As ω varies from 0 to ∞, both magnitude and phase may be plotted as functions of ω.

It is customary to plot both magnitude and phase against $\log\ \omega$, which accords a high dynamic range to ω. Further, the magnitude is plotted in decibels as: $|G(j\omega)|_{\text{dB}} = 20\ \log_{10}|G(j\omega)|$. The phase $\phi(\omega)$ is plotted in degrees. The resulting frequency response plots are commonly known as **Bode plots**.

The plotting of the frequency response in the case of first and second-order systems is illustrated below.

First-Order System

Let $G(s) = \frac{K}{\tau s+1}$; then $G\,(j\omega) = \frac{K}{1+j\omega\tau}$; Hence,

$$|G\,(j\omega)|_{\text{dB}} = 20\ \log K\ - 20\ \log|1+j\omega\tau|\ , \quad \angle G\,(j\omega) = -\tan^{-1}\omega\tau$$

Example 2.5.3

The Bode magnitude and phase plots for $G\,(s) = \frac{1}{s+1}$ are plotted (Figure 2.5.3).

The magnitude plot is characterized by: $G\,(j0) = 0\ \text{dB},\ \ G\,(j1) = -3\ \text{dB}$, and a $-20\ \text{dB/decade}$ slope for large ω.

The phase plot is characterized by: $G(j0) = 0°,\ G(j0.1) = -5.7°,\ G(j1) = -45°,\ G(j10) = -84.3°,\ G(j\infty) = -90°$.

First-Order System with Integrator

Let $G(s) = \frac{K}{s(\tau s+1)}$; then $G(j\omega) = \frac{K}{j\omega(1+j\omega\tau)}$; Hence,

$$|G(j\omega)|_{\text{dB}} = 20\log K - 20\log\omega - 20\log|1+j\omega\tau|$$

$$\angle G(j\omega) = -\tan^{-1}\omega\tau - 90°$$

Example 2.5.4

The Bode magnitude and phase plots for $G(s) = \frac{1}{s(s+1)}$ are plotted below.

The magnitude plot is characterized by an initial slope of -20 dB/decade that changes to -40 dB/decade for large ω; at the corner frequency, $G(j1) = -3$ dB .

The phase plot is characterized by: $G(j0) = -90°,\ G(j0.1) = -96°,\ G(j1) = -135°,\ G(j10) = -174.3°,\ G(j\infty) = -180°$.

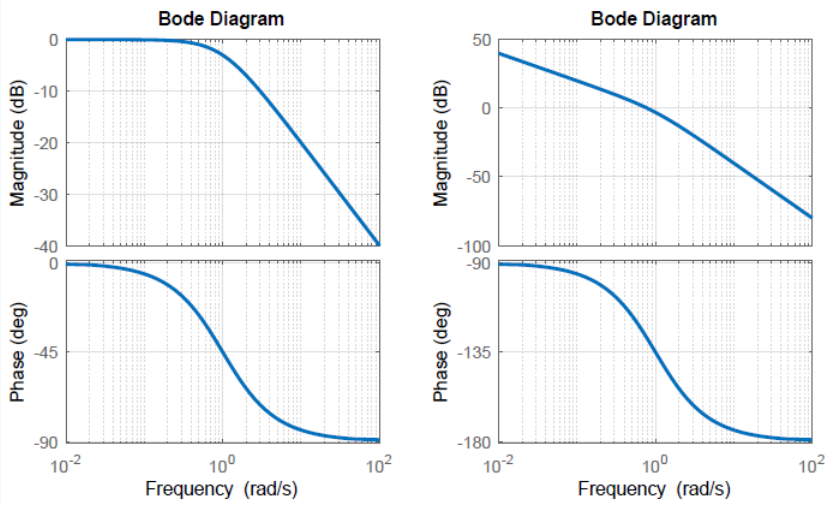


Figure 2.5.3: Bode plot for Examples 2.5.3 (left) and 2.5.4 (right).

Second-Order System with Real Poles

Let $G(s) = \frac{K}{(\tau_1 s+1)(\tau_2 s+1)}$; then $G(j\omega) = \frac{K}{(1+j\omega\tau_1)(1+j\omega\tau_2)}$; Hence,

$$|G(j\omega)|_{\text{dB}} = 20\log K - 20\log|1+j\omega\tau_1| - 20\log|1+j\omega\tau_2|$$

$$\angle G(j\omega) = -\tan^{-1}\omega\tau_1 - \tan^{-1}\omega\tau_2$$

Example 2.5.5

The Bode magnitude and phase plots for $G(s) = \frac{2}{(s+1)(s+2)}$ are plotted in Figure 2.5.4.

The magnitude plot is characterized by: $G(j0) = 0$ dB, $G(j1) = -3$ dB , an intermediate slope of -20 dB/decade, $G(j2) = -14$ dB , and a slope of -40 dB/decade for large ω.

The phase plot is characterized by: $G(j0) = 0°,\ G(j1) = -45°,\ G(j2) = -135°,\ G(j\infty) = -180°$.

Second-Order System with Complex Poles

Let $G(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$; then $G(j\omega) = \frac{\omega_n^2}{\omega_n^2-\omega^2+j2\zeta\omega\omega_n}$, where ω_n and ζ represent the natural frequency and the damping ratio. Hence,

$$|G(j\omega)|_{\text{dB}} = -20\log_{10}\left|1-\left(\frac{\omega}{\omega_n}\right)^2 + j2\zeta\left(\frac{\omega}{\omega_n}\right)\right|$$

$$\phi(\omega) = \tan^{-1}\left(\frac{2\zeta\left(\frac{\omega}{\omega_n}\right)}{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]}\right)$$

Example 2.5.6

The Bode magnitude and phase plots for $G(s) = \frac{10}{s^2+2s+10}$ are plotted below.

The magnitude plot is characterized by: $G(j0) = 0$ dB, $G(j\omega_n) = -20\log(2\zeta)$, and a -40 dB/decade slope for large ω.

The phase plot is characterized by: $\phi(j0) = 0^\circ$, $\phi(j\omega_n) = -90^\circ$, $\phi(j\infty) = -180^\circ$.

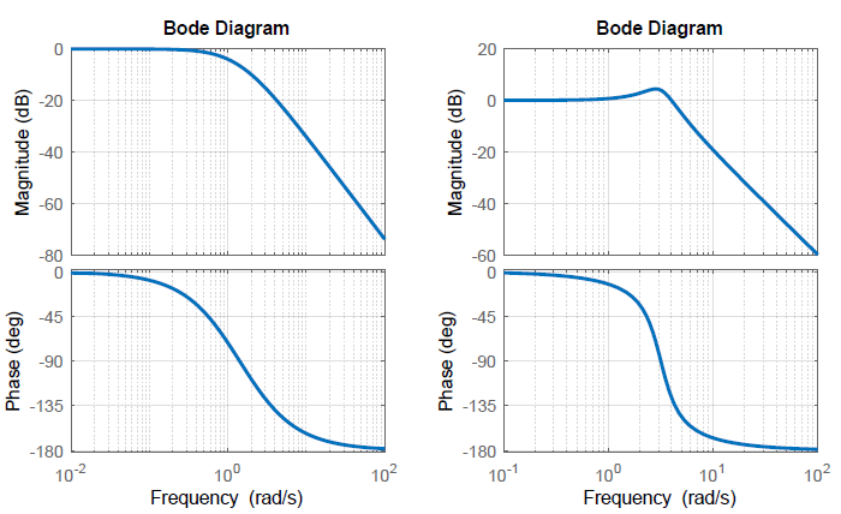


Figure 2.5.4: Bode plot for Examples 2.5.5 (left) and 2.5.6 (right).

Resonance Peak in the Frequency Response

The Bode magnitude plot of a transfer function with complex poles and low damping displays a distinctive peak in the Bode magnitude plot at the resonant frequency, ω_r; the resonant frequency and peak magnitude are computed as:

$$\omega_r = \omega_n\sqrt{1-2\zeta^2}, \quad \zeta < \frac{1}{\sqrt{2}}$$

$$M_{\text{p}\omega} = \frac{1}{2\zeta\sqrt{1-\zeta^2}}, \quad \zeta < \frac{1}{\sqrt{2}}$$

Example 2.5.7

Let $G(s) = \frac{10}{s^2+2s+10}$; then, we have: $\omega_n = \sqrt{10}\ \frac{\text{rad}}{\text{sec}}$, $\zeta = \frac{1}{\sqrt{10}}$, $\omega_r = 8\ \frac{\text{rad}}{\text{sec}}$, and $M_{\text{p}\omega} = 1.67$ or 4.44 dB (Figure 2.5.4).

Relating the Time and Frequency Response

When the system transfer function has poles with a low damping ratio, the Bode magnitude plot displays a resonant peak. Concurrently, the step response of the system displays oscillations.

To illustrate this relationship, we consider a prototype second-order systems: $G(s) = \frac{1}{s^2+2\zeta s+1}$, where $\omega_n = 1$, $\zeta \in \{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9\}$

The frequency response and the time-domain unit-step response for the second-order transfer functions with $\omega_n = 1$ for the various values of $\zeta \in \{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9\}$ are plotted below.

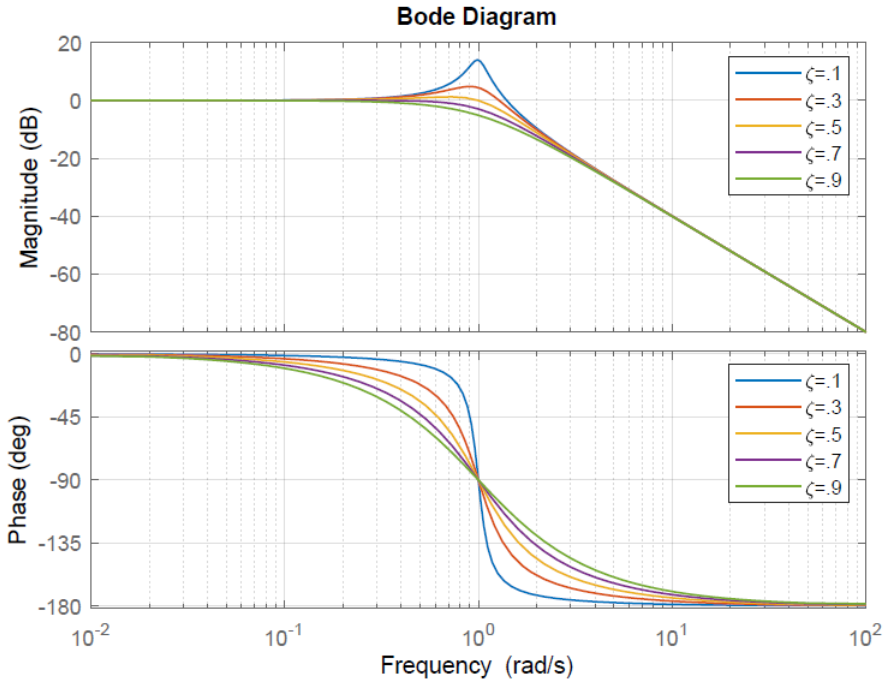


Figure 2.5.5: Bode magnitude and phase plots for selected damping ratios.

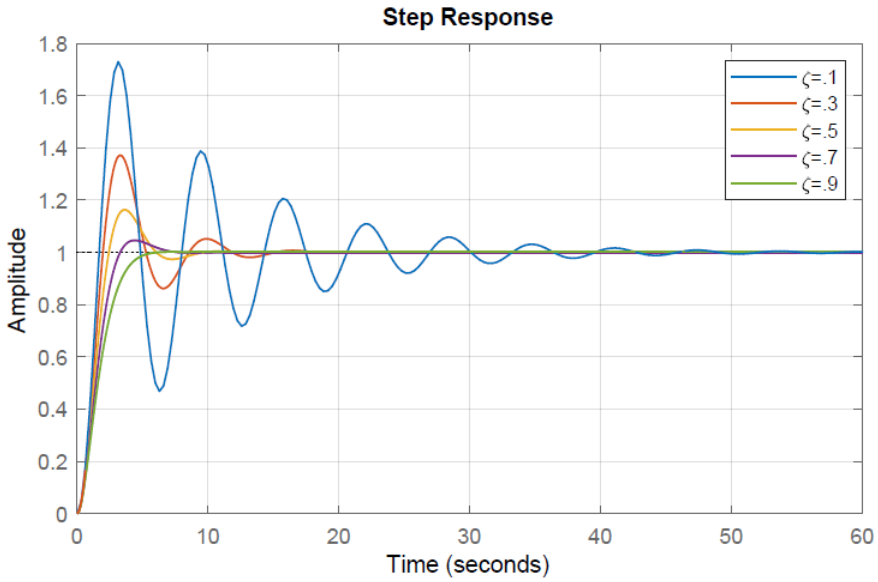


Figure 2.5.6: Step response of the second-order system for selected damping ratios.

CHAPTER OVERVIEW

3: Feedback Control System Models

Learning Objectives

1. Characterize the static controller in a feedback control system.
2. Characterize first-order dynamic controllers for feedback back control systems.
3. Characterize the PID controller in terms of the three basic control modes.
4. Characterize rate feedback controllers for single-input single-output systems.

3.0: Prelude to Feedback Control System Models

Feedback, i.e., using observed outputs as another input to the system, is an important characteristic of automatic control systems. In particular, negative feedback permits precise control of the overall system gain, and can compensate for signal distortions and nonlinearities. Feedback makes the system robust against disturbance inputs and parameter variations.

The design of a single-input single-output (SISO) feedback control system involves the use of a comparator to generate an error signal, $e = r - y$ (Figure 3.0.1). The controller, $K(s)$, suitably conditions the error signal before it is input to the process, $G(s)$. A sensor, $H(s)$, monitors the output and feeds it back to the input. Often a sensor gain $H(s) = 1$ is assumed.

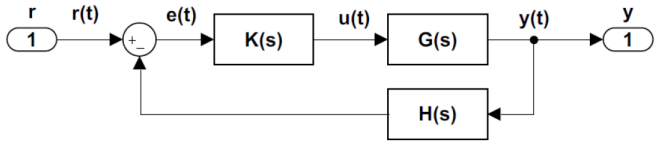


Figure 3.0.1: Feedback control system block diagram.

A static, i.e., proportional gain controller, $u(t) = Ke(t)$, serves as baseline controller in a SISO feedback control system. The static controller is simple to design and is effective in meeting the basic design objectives in output regulation and reference tracking problems.

First-order phase-lead and phase-lag controller are used for transient and steady-state response improvements in feedback control systems. The phase-lead controller is characterized by positive phase contribution to the Bode plot of loop transfer function. The phase-lag controller, used for steady-state error improvement, adds negative phase to the Bode plot.

A proportional-integral-derivative (PID) controller comprises three basic modes of control that include the proportional, the derivate, and the integral modes. The PID controller is popular in industrial process control for its ability to provide robustness in designs involving simple models of complex processes.

The cascade controller design employs output feedback, which has a limited potential to affect the behavior of the closed-loop control system. Pole placement using state feedback is a more powerful controller design method that may be employed with state variable models of dynamic systems.

Rate sensors, such as rate gyros and tachometers, are used in position control applications to enhance the controller design options. A rate feedback controller offers additional design flexibility and may be considered when static controller falls short in meeting design objectives.

In this chapter, we will limit our discussion to the cascade and rate feedback controllers of the static and dynamic types. These controller are described below.

3.1: Static Feedback Controller

Feedback Control System

The standard block diagram of a single-input single-output (SISO) feedback control system includes a plant, $G(s)$, a controller, $K(s)$, and a sensor, $H(s)$, where $H(s)=1$ is often assumed.

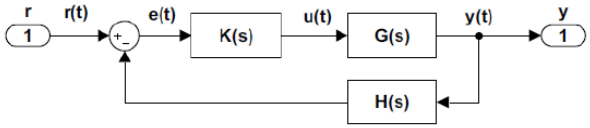


Figure 3.1.1: Feedback control system with plant, $G(s)$, sensor, $H(s)$, and controller, $K(s)$.

The overall system transfer function from input, $r(t)$, to output, $y(t)$, can be obtained by considering the error signal, $e=r-Hy=r-KGHe$. Thus, $(1+KGH)e=r$.

Let $L(s)=KGH(s)$ denote the feedback loop gain; then, $(1+L)e=r$.

The error transfer function from r to e is obtained as:

$$S(s)=\frac{1}{1+L(s)}=\frac{1}{1+KGH(s)}$$

The closed-loop transfer function from r to y is obtained as:

$$T(s)=\frac{KG(s)}{1+KGH(s)}$$

We may note that: $S(s)+T(s)=1$.

Static Loop Controller Design

A static controller denotes the use of an amplifier with a gain, K, to generate input to the plant, $G(s)$. The controller action is represented as: $u=Ke$, where e represents the error signal and u is the plant input.

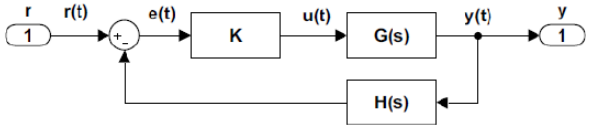


Figure 3.1.2: Feedback control system with static gain controller.

Assuming $H(s)=1$, the closed-loop transfer function is given as:

$$\frac{y(s)}{r(s)}=T(s)=\frac{KG(s)}{1+KG(s)}$$

Let $G(s)=\frac{n(s)}{d(s)}$; then, the closed-loop transfer function is obtained as:

$$\frac{y(s)}{r(s)}=\frac{Kn(s)}{d(s)+Kn(s)}$$

The closed-loop characteristic polynomial is defined as: $\Delta(s,K)=d(s)+Kn(s)$.

From a controller design perspective, the gain K can be selected to achieve desirable root locations for the closed-loop characteristic polynomial. The design can be performed by comparing the coefficients of the characteristic polynomial with a desired characteristic polynomial.

Example 3.1.1

The model for an environmental control system is given as: $G(s) = \frac{1}{20s+1}$.

Assuming a static gain controller, the closed-loop characteristic polynomial is obtained as: $\Delta(s, K) = 20s + 1 + K$.

Suppose a desired characteristic polynomial is selected as: $\Delta_{des}(s) = 4(5s+1)$. Then, by comparing the coefficients, we obtain the static controller as: $K = 3$.

Example 3.1.2

Jun 22, 2020, 2:26 PM

Example 3.2:

The model for a position control system is given as: $G(s) = \frac{1}{s(0.1s+1)}$.

Assuming a static gain controller, the characteristic polynomial is obtained as: $\Delta(s, K) = s(0.1s+1) + K$.

Suppose a desired characteristic polynomial is selected as: $\Delta_{des}(s) = 0.1(s^2 + 10s + 50)$. Then, by comparing the coefficients, we obtain the static controller as: $K = 5$.

3.2: First-Order Dynamic Controllers

Dynamic Loop Controllers

A dynamic controller, $K(s)$, in the feedback loop is a dynamic system that suitably conditions the error signal as $u(s) = K(s)e(s)$ to affect the closed-loop system response. The feedback control system configuration is shown below where $H(s) = 1$ is assumed.

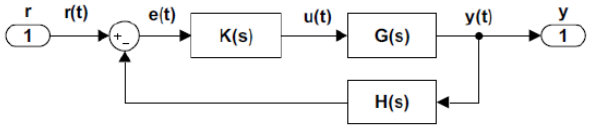


Figure 3.2.1: Feedback control system with plant, $G(s)$, sensor, $H(s)$, and controller, $K(s)$.

To analyze the feedback control system, assume that the plant is described by $G(s) = \frac{n(s)}{d(s)}$, and the dynamic controller is described by $K(s) = \frac{n_c(s)}{d_c(s)}$; then, the closed-loop transfer function from the reference input to the plant output is given as:

$$\frac{y(s)}{r(s)} = \frac{n(s)\,n_c(s)}{d(s)\,d_c(s) + n(s)\,n_c(s)}.$$

The error transfer function from the reference input to the comparator output is given as:

$$\frac{e(s)}{r(s)} = \frac{1}{d(s)\,d_c(s) + n(s)\,n_c(s)}.$$

The closed-loop characteristic polynomial is given as: $\Delta(s) = d(s)\,d_c(s) + n(s)\,n_c(s)$.

First-Order Dynamic Controllers

In particular, we consider a first-order dynamic controller, described by the transfer function:

$$K(s) = K\left(\frac{s + z_c}{s + p_c}\right)$$

The first-order controller adds a zero located at $-z_c$ and a pole located at $-p_c$ to the loop transfer function: $KGH(s)$.

Let $G(s) = \frac{n(s)}{d(s)}$, $H(s) = 1$; then, the closed-loop characteristic polynomial with the controller in the loop is given as: $\Delta(s) = d(s)(s + p_c) + n(s)(s + z_c)$.

Traditionally, first-order controllers are characterized as of phase-lead or phase-lag type, where lead and lag terms refer to the phase contribution from the controller to the Bode phase plot of the loop transfer function.

Let $K(j\omega) = K\left(\frac{j\omega + z_c}{j\omega + p_c}\right)$; then, the phase angle contribution of the controller toward the loop transfer function is given as:

$$\angle K(j\omega) = \angle(j\omega + z_c) - \angle(j\omega + p_c) = \theta_z - \theta_p$$

where θ_z and θ_p denote the angles subtended by the controller pole and zero at a desired closed-loop pole location in the complex plane.

The phase contribution is positive for phase-lead controller and negative for phase-lag controller (see Figures 3.2.2 and 3.2.3 below).

Phase-Lead Controller

A phase-lead controller is characterized by $z_c < p_c$, and contributes a net positive phase to the loop transfer function.

Example 3.2.1

A phase-lead controller is expressed as: $K(s) = \frac{s+1}{s+10}$. The phase contribution from the controller is shown below.

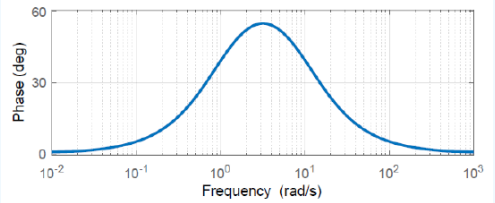


Figure 3.2.2: Phase contribution from a first-order phase-lead controller.

Phase-Lag Controller

A phase-lag controller is characterized by $z_c > p_c$, and contributes a net negative phase to the loop transfer function.

Example 3.2.2

A phase-lead controller is expressed as: $K(s) = \frac{s+0.1}{s+0.01}$. The phase contribution from the controller is shown below.

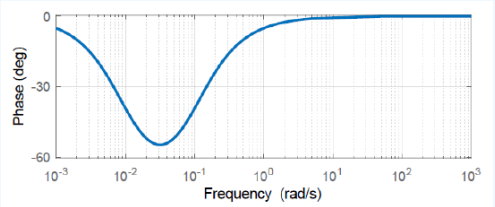


Figure 3.2.3: Phase contribution from a first-order phase-lag controller.

Lead-lag controller

A lead-lag controller is a second-order controller that combines a phase-lead section with a phase-lag section; it thus has the form:

$$K(s) = \frac{K(s+z_1)(s+z_2)}{(s+p_1)(s+p_2)}$$

As an example, a lead-lag controller is given as: $K(s) = \frac{K(s+0.1)(s+10)}{(s+1)^2}$.

We may note that the phase-lag zero is placed at a frequency much lower than the dominant pole frequency, whereas the phase-lead zero is placed in the vicinity of the dominant pole frequency.

Controller Design Example

The controller gain selection for a phase-lead design is explored in the following example.

Example 3.2.3

The transfer function model for a position control system is given as: $G(s) = \frac{1}{s(0.1s+1)}$.

Let a phase-lead controller for the model be defined by: $K(s) = K\left(\frac{0.1s+1}{0.02s+1}\right)$.

Assuming $H(s) = 1$, the loop transfer function is formed as: $KG(s) = \frac{K}{s(0.0.02s+1)}$.

The closed-loop characteristic polynomial is formulated as: $\Delta(s, K) = s(0.02s+1) + K$.

Let a desired characteristic polynomial be given as: $\Delta_{des}(s) = 0.02(s^2 + 50s + 1000)$.

By comparing the coefficients, we obtain the controller gain as: $K = 20$.

Hence, the dynamic controller for the system is given as: $K(s) = 20\left(\frac{0.1s+1}{0.02s+1}\right)$.

We may note that when designing cascade controllers, not all closed-loop pole locations are reachable. The reachable pole locations can be found by the application of root locus technique (see Chapter 5).

3.3: PI, PD, and PID Controllers

The PID Controller

The PID controller is a general-purpose controller that combines the three basic modes of control, i.e., the proportional (P), the derivative (D), and the integral (I) modes.

The PID controller in the time-domain is described by the relation:

$$u(t) = k_p + k_d \frac{d}{dt} e(t) + k_i \int e(t) dt$$

The PID controller has a transfer function:

$$K(s) = k_p + k_d s + \frac{k_i}{s}$$

The controller gains for the three basic modes of control are given as: $\{k_p,\ k_d,\ \ k_i\}$. Of these, the proportional term serves as a static controller, the derivative term helps speed up the system response, and the integral term helps reduce the steady-state error.

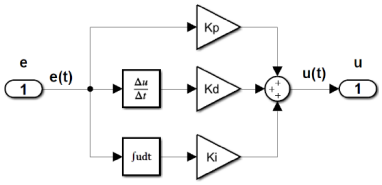


Figure 3.3.1: Three basic control modes of control represented in the PID controller.

Closed-Loop Characteristic Polynomial

Let $G(s) = \frac{n(s)}{d(s)}$; then the closed-loop characteristic polynomial with a PID controller in the loop is given as:

$$\Delta(s) = s\, d(s) + (k_d s^2 + k_p s + k_i)\, n(s)$$

For noise suppression, a first-order filter may be added to the PID controller; the modified controller transfer function is given as:

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{T_f s + 1}$$

The filter additionally makes the controller transfer function proper and hence realizable by a combination of a low-pass and high-pass filters.

The control system design objectives may require using only a subset of the three basic controller modes. The two common choices, the proportional-derivative (PD) controller and the proportional-integral (PI) controller are described below.

PD Controller

A PD controller is described by the transfer function:

$$K(s) = k_p + k_d s = k_d \left(s + \frac{k_p}{k_d} \right)$$

A PD controller thus adds a single zero to the loop transfer function. The closed-loop characteristic polynomial is given as:

The phase contribution of the PD controller increases from 0° at low frequencies to 90° at high frequencies.

For practical reasons, a pole with a short time constant, T_f, may be added to the PD controller. The pole helps limit the loop gain at high frequencies, which is desirable for disturbance rejection. The modified PD controller is described by the transfer function:

$$K(s) = k_p + \frac{k_d s}{T_f s + 1}$$

The modified PD controller is very similar to a first-order phase-lead controller; it is similarly employed to improve the transient response of the system.

Example 3.3.1

An environment control system is modeled as: $G(s) = \frac{1}{25s+1}$. The model has a time constant of $25\ sec.$

In order to speed up the response a PD controller with filter is designed as: $K(s) = 10 + \frac{s}{.05s+1}$.

The closed-loop system has an effective time constant $\tau \cong 2.5\ sec$. The step response of the closed-loop system is plotted below.

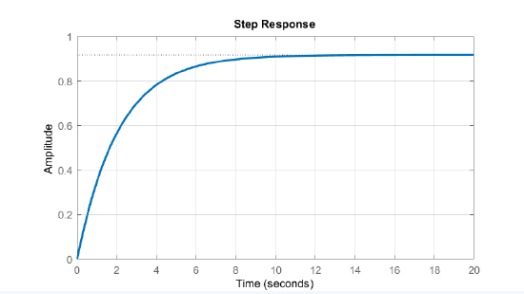


Figure 3.3.2: Step response of the closed-loop environmental control system.

PI Controller

A PI controller is described by the transfer function:

$$K(s) = k_p + \frac{k_i}{s} = \frac{k_p(s + k_i/k_p)}{s}$$

The PI controller thus adds a pole at the origin (an integrator) and a finite zero to the feedback loop. The presence of the integrator in the loop forces the error to a constant input to go to zero in steady-state; hence PI controller is commonly used in designing servomechanisms.

The controller zero is normally placed close to the origin in the complex s-plane. The presence of a pole–zero pair adds a closed-loop system pole with a large time constant. The zero location can be adjusted so that the contribution of the slow mode to the overall system response stays small.

The PI-PD Controller

The PD and PI sections can be combined in a PI-PD controller as:

$$K(s) = \left(k_p + \frac{k_i}{s}\right)(1 + k_d s) \quad \text{or} \quad K(s) = (k_p + k_d s)\left(1 + \frac{k_i}{s}\right)$$

The PI-PD controller adds two zeros and an integrator pole to the loop transfer function. The zero from the PI part may be located close to the origin; the zero from the PD part is placed at a suitable location for desired transient response improvement.

The PI-PD controller is similar to a regular PID controller that is described by the transfer function:

$$K(s) = k_p + k_d s + \frac{k_i}{s} = \frac{k_d s^2 + k_p s + k_i}{s}$$

The PID controller imparts both transient and steady-state response improvements to the system. Further, it delivers stability as well as robustness to the closed-loop system.

PID Controller Tuning

The PID controller tuning refers to the selection of the controller gains: $\{k_p,\ k_d, k_i\}$ to achieve desired performance objectives. Industrial PID controllers are often tuned using empirical rules, such as the Ziegler–Nicholas rules.

In the MATLAB Control System Toolbox, a PID controller object is created using the ‘pid’ command that can then be tuned with the ‘pidtune’ command. The MATLAB PID tuning algorithm aims for a 60° phase margin that results in a moderate 10%

overshoot.

Example 3.3.2

For a small DC motor, the following parameter values are assumed: $R = 1\,\Omega$, $L = 10\,\text{mH}$, $J = 0.01\,kg \cdot m^2$, $b = 0.1\ \frac{N \cdot s}{rad}$, and $k_t = k_b = 0.05$; the transfer function of the DC motor from armature voltage to motor speed is approximated as: $G(s) = \frac{500}{(s+10)(s+100)}$.

The MATLAB tuned PID controller for the motor is given as:

$$K(s) = 3.51 + 0.037s + \frac{56.2}{s}$$

The MATLAB tuned PID controller with filter (PIDF) is given as:

$$K(s) = 3.26 + \frac{0.015s}{0.005s+1} + \frac{54.2}{s}$$

The response of the closed-loop system for the two controllers (PID and PIDF) is shown below (Figure 3.3.3). The response in both cases shows a settling time of 0.3sec with 7-8% overshoot and no steady-state error.

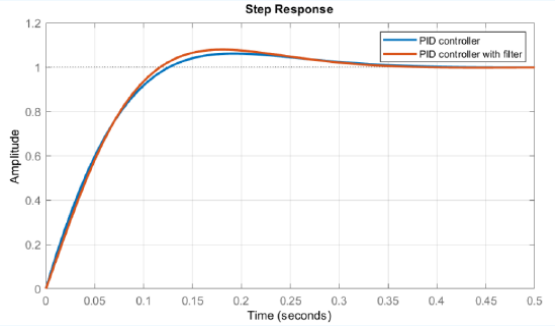


Figure 3.3.3: Unit-step response of the DC motor model with PID and PIDF controllers.

3.4: Rate Feedback Controllers

Rate Feedback Controllers

Rate feedback signal is often available from a tachometer or a rate gyro in position control applications. The signal can be employed to improve the response of the feedback control system.

The basic rate feedback configuration (Figure 3.4.1) includes a rate constant k_f that multiplies the velocity signal together with a static controller, K, in the loop.

The closed-loop transfer function for the rate feedback configuration can be obtained by applying Mason's gain rule, originally defined for signal flow graphs. For simple block diagrams, the gain rule is stated as:

$$\frac{y(s)}{r(s)} = \frac{F(s)}{1 - \sum L_i(s)}$$

where $F(s)$ denotes the forward path gain from an input node to an output node, and $\sum L_i(s)$ denotes sum of the loop gains.

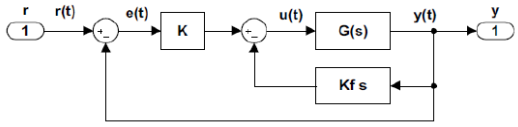


Figure 3.4.1: The rate feedback configuration in feedback control system.

Rate Feedback as PD Controller

The effective loop gain in the case or rate feedback configuration with a static controller is computed as:

$$\sum L_i(s) = -G(s)(k_f s + K)$$

The loop gain includes a plant transfer function in cascade with a PD controller, where the controller zero is located at: $s = -\frac{K}{k_f}$. Hence rate feedback configuration is identical to placing a PD controller in the loop.

Let $G(s) = \frac{n(s)}{d(s)}$; then, the closed-loop transfer function is obtained as:

$$\frac{y(s)}{r(s)} = \frac{Kn(s)}{d(s) + (k_f s + K)n(s)},$$

The resulting closed-loop characteristic polynomial is given as:

$$\Delta(s) = d(s) + (k_f s + K)n(s)$$

By using simple block diagram manipulation, the rate feedback configuration (Figure 3.4.1) can be equivalently represented as a feedback control system with a PD controller in the feedback path (Figure 3.4.2).

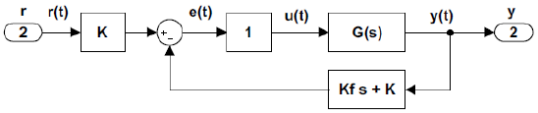


Figure 3.4.2: Alternative structure for rate feedback configuration with static controller.

Example 3.4.1

The transfer function of a mass–spring–damper system is given as: $G(s) = \frac{1}{ms^2 + bs + k}$. The following parameter values are assumed: $m = 1, b = 2, \text{ and } k = 10$. It is desired to improve the transient response using rate feedback controller.

The closed-loop characteristic polynomial for the rate feedback configuration is:

$$\Delta(s) = s^2 + (k_f + 2)s + K + 10$$

We assume that a desired characteristic polynomial is selected as: $\Delta_{des}(s) = \left(s^2 + 6s + 25\right)$.

By comparing the polynomial coefficients, the required controller gains are obtained as: $K = 15,\ k_f = 4$. The unit-step response of the closed-loop system is shown in Figure 3.4.3.

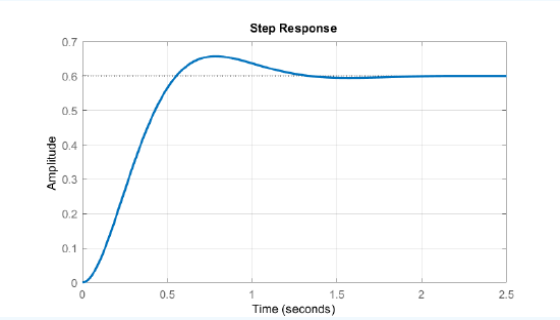


Figure 3.4.3: Step response of mass-spring-damper system with rate feedback controller.

Rate Feedback as PID Controller

The rate feedback can be used in conjunction with a PI controller to effectively implement a PID controller. A rate feedback controller with cascade PID controller is shown in Figure 3.4.4.

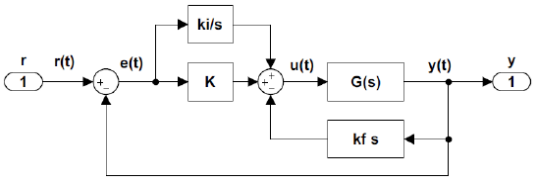


Figure 3.4.4: Rate feedback configuration with cascade PI controller.

Let the cascade PI controller be defined as: $K_{PI}(s) = K + \frac{k_i}{s}$; then, using gain rule (3.4.1), the effective loop gain is obtained as:

$$\sum L_i(s) = -\left(K + \frac{k_i}{s} + k_f s\right) G(s)$$

Hence using a cascade PI controller with rate feedback amounts to placing a PID controller in the feedback loop (Figure 3.4.5).

Assuming a plant transfer function: $G(s) = \frac{n(s)}{d(s)}$, the closed-loop transfer function is obtained as:

$$\frac{y(s)}{r(s)} = \frac{(Ks + k_i)\, n(s)}{sd(s) + (k_f s^2 + Ks + ki)\, n(s)},$$

The resulting closed-loop characteristic polynomial is given as:

$$\Delta(s) = sd(s) + \left(k_f s^2 + Ks + k_i\right) n(s)$$

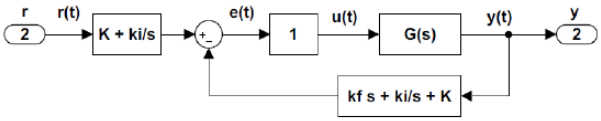


Figure 3.4.5: Alternate structure for rate feedback with cascade PI controller.

Example 3.4.2

The transfer function of a mass–spring–damper system is given as: $G(s) = \frac{1}{ms^2+bs+k}$. The following parameter values are assumed: $m = 1, b = 2,\ and\ k = 10$. It is desired to improve the transient as well as steady-state response using rate feedback with cascade PI controller.

The closed-loop characteristic polynomial for the rate feedback configuration with cascade PI controller is given as:

$$\Delta(s) = s^3 + (k_f + 2)\, s^2 + (K + 10)\, s + k_i.$$

We assume that a desired closed-loop characteristic polynomial is selected as: $\Delta_{des}(s) = \left(s^2 + 10s + 25\right)(s + 5)$.

Then, by comparing the polynomial coefficients, the controller gains are obtained as: $k_f = 13,\ \ K = 65,\ \ and\ k_i = 125$. The unit-step response of the closed-loop system is shown in Figure 3.4.6.

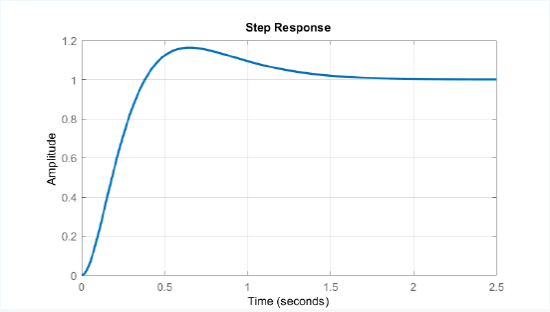


Figure 3.4.6: Step response of mass-spring-damper system with rate feedback and cascade PI controllers.

By comparing the step responses in Examples 3.4.1 and 3.4.2, we observe that:

1. The rate feedback with cascade PI controller has a higher (18%) overshoot compared to rate feedback with static controller (12%).
2. The rate feedback with cascade PI controller has a smaller rise time ($0.3\ sec$) compared to rate feedback with static controller (about $0.5\ sec$).
3. The rate feedback with cascade PI controller has no steady-state error compared to (40%) error in the case of rate feedback with static controller.
4. The step response in both cases settles in about $1.5\ sec$.

CHAPTER OVERVIEW

4: Control System Design Objectives

Learning Objectives

1. Characterize closed-loop stability in a feedback control system model.
2. Characterize performance objectives for transient and steady-state improvement.
3. Characterize the disturbance rejection problem in feedback control systems.
4. Determine the sensitivity of the closed-loop system to parameter variations.

4.0: Prelude to Control System Design Objectives

In a feedback control system, the controller is designed to achieve certain objective. The first and foremost among them is closed-loop system stability. The next is good dynamic stability displayed by an acceptable transient response. In tracking systems, the elimination of tracking error in desired. Further, the controlled system is required to mitigate disturbances entering the system and guard it against parameter variations and unmodeled dynamics. Hence we will discuss the ways to achieve the following objectives in this chapter:

1. Closed-loop stability
2. Acceptable transient response
3. Acceptable steady-state error
4. Sensitivity and robustness

In control system design, the stability of the closed-loop characteristic polynomial can be ascertained by algebraic methods, such as the Routh's array. These method can be applied toward the design of static controller. Stability can also be determined from the Bode plot of the loop transfer function.

The transient response is commonly evaluated in terms of time-domain performance metrics applied to the step response of the closed-loop system. The design specifications specify limits on the settling time of the system response and/or damping ratio of closed-loop roots. A phase-lead or PD/PID controller can be employed to obtain transient response improvements.

The steady-state error to prototype (step and ramp) inputs is measured in terms of position and velocity error constants of the loop transfer function. Generally speaking, high loop gain reduces steady-state error, but also makes the system prone to oscillations. The steady-state error can be eliminated by placing a PI/PID controller in the feedback loop.

Disturbance inputs are unavoidable in the operation of physical systems. Common examples of disturbance inputs include road bumps while driving, turbulence in the airplanes, and machinery vibrations in industrial plants, etc. Effective disturbance rejection requires high loop gain in the frequency range of disturbance.

Aging in the physical system causes changes in the plant transfer function thus reducing the effectiveness of the controller. A well-designed control system is desired to have robustness against unmodeled dynamics as well as low sensitivity to parameter variations.

The control system design objectives involve inherent trade-offs. For example, a static controller cannot simultaneously improve the transient response and reduce steady-state error to a constant input. Similarly, disturbance rejection and reference tracking pose conflicting design objectives. When faced with such situations, it is not always easy to find the right balance in meeting and prioritizing design objectives. Hence, the control system design is more of an art than an exact science, i.e., it relies much on the skills of the designer.

4.1: Stability of the Closed-Loop System

Closed-Loop Stability

Ensuring the stability of the closed-loop is the first and foremost control system design objective. Even though the physical plant, $G(s)$, may be stable, the presence of feedback can cause the closed-loop system to become unstable, as in the case of higher order plant models.

The standard block diagram of a single-input single-output (SISO) feedback control system (Figure 4.1.1) includes a plant, $G(s)$, a controller, $K(s)$, and a sensor, $H(s)$, where $H(s) = 1$ is assumed.

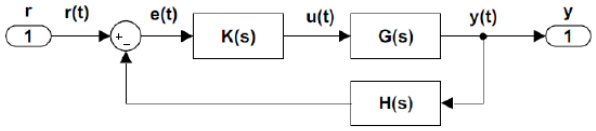


Figure 4.1.1: Feedback control system with plant, $G(s)$, sensor, $H(s)$, and controller, $K(s)$.

The overall system transfer function from input, $r(t)$, to output, $y(t)$, is given as:

$$T(s) = \frac{KG(s)}{1 + KGH(s)}$$

The determination of stability is based on the closed-loop characteristic polynomial:

$$\Delta(s, K) = 1 + KGH(s).$$

In particular, let $G(s) = \frac{n(s)}{d(s)}$; then, assuming a static controller: $K(s) = K$, the closed-loop characteristic polynomial is given as:

$$\Delta(s, K) = d(s) + Kn(s)$$

Alternatively, assuming a dynamic controller, $K(s) = \frac{n_C(s)}{d_C(s)}$, the characteristic polynomial is given as:

$$\Delta(s) = d(s)d_C(s) + n(s)n_C(s)$$

Stability Determination by Algebraic Methods

The stability of the characteristic polynomial is determined by algebraic methods that characterize its root locations based on the coefficients of the polynomial. In the following, we assume that $\Delta(s)$ is an nth order polynomial expressed as:

$$\Delta(s) = s^n + a_1 s^{n-1} + \ldots + a_{n-1}s + a_n$$

Necessary Condition. A necessary condition for stability of the polynomial is that coefficients $a_i,\ i = 1, 2, \cdots, n$, are all nonzero and are positive, i.e., $a_i > 0$.

Sufficient Condition. A sufficient condition for polynomial stability is obtained through the application of the following equivalent criteria:

The Hurwitz Criterion

The nth order polynomial, $s^n + a_1 s^{n-1} + \cdots + a_n$ is stable, i.e., has its roots in the open left-half plane (OLHP), if and only if the following determinants assume positive values:

$$|a_1|, \quad \begin{vmatrix} a_1 & a_3 \\ 1 & a_2 \end{vmatrix}, \quad \begin{vmatrix} a_1 & a_3 & a_5 \\ 1 & a_2 & a_4 \\ 0 & a_1 & a_3 \end{vmatrix}, \ldots$$

The Routh's Criterion

The nth order polynomial, $s^n + a_1 s^{n-1} + \cdots + a_n$ is stable, i.e., has its roots in the open left-half plane (OLHP), if and only if all entries in the Routh's array are positive.

$$\begin{array}{c|lll} s^n & 1 & a_2 & \dots \\ s^{n-1} & a_1 & a_3 & \dots \\ \vdots & b_1 & b_2 & \dots \\ & c_1 & c_2 & \dots \\ s^1 & \dots & & \\ s^0 & \dots & & \end{array}$$

The first two rows in the Routh's array are filled by alternating the coefficients of the polynomial. The entries appearing in the third and subsequent rows are computed as follows:

$b_1 = -\frac{1}{a_1}\begin{vmatrix} 1 & a_3 \\ a_1 & a_2 \end{vmatrix}$, $b_2 = -\frac{1}{a_3}\begin{vmatrix} a_2 & a_4 \\ a_3 & a_5 \end{vmatrix}$, $c_1 = -\frac{1}{b_1}\begin{vmatrix} a_1 & a_3 \\ b_1 & b_2 \end{vmatrix}$, etc.

The Routh's stability criterion states that the number of unstable roots of the polynomial equals the number of sign changes in the first column of the Routh's array.

Stability Determination for Low-Order Polynomials

Low-order polynomials (i.e., polynomials of degree $n = 2,\ 3$) are often encountered in model-based control system design. The Routh or Hurwitz can be simplified when applied to such polynomials.

The stability criteria for second and third-order polynomials are given below:

Second-order polynomial: $(n = 2)$: $a_1 > 0,\ \ a_2 > 0$.

Third-order polynomial: $(n = 3)$: $a_1 > 0, \quad a_2 > 0, \quad a_3 > 0, \quad a_1 a_2 - a_3 > 0.$

Controller Gain Selection

The stability conditions can be used to determine the range of controller gain, K, to ensure that the roots of the closed-loop characteristic polynomial, $\Delta(s, K)$, lie in the open left-half plane (OLHP).

Example 4.1.1

Let $G(s) = \frac{K}{s(s+2)}$, $H(s) = 1$; then, $\Delta(s, K) = s^2 + 2s + K$. By using the above stability criteria, $\Delta(s)$ is stable for $K > 0$.

Example 4.1.2

Let $\Delta(s, K) = s^3 + 3s^2 + 2s + K$. By using the above stability criteria, $\Delta(s)$ is stable if the following conditions are met: $K > 0$ and $6 - K > 0$. Accordingly, the range of K for closed-loop stablity is given as $0 < K < 6$.

Example 4.1.3

The simplified model of a small DC motor is given as: $\frac{\theta(s)}{V_a(s)} = \frac{10}{s(s+6)}$.

A PID controller for the motor model is defined as: $K(s) = k_p + k_d s + \frac{k_p}{s}$. The closed-loop characteristic polynomial is given as:

$$\Delta(s) = s^2(s+6) + 10(k_d s^2 + k_p s + k_i) = s^3 + (6 + 10k_d)s^2 + 10k_p s + 10k_i.$$

The constraints on the PID controller gains to ensure the stability of the third-order polynomial are given as:

$$k_p,\ k_i,\ 6 + 10k_d > 0$$

$$k_p(6 + 10k_d) - k_i > 0$$

We may choose, e.g., $k_p = 1,\ k_i = 1,\ k_d = 1$ to meet the stability requirements.

Stability Determination from Frequency Response Plots

The frequency response function of the loop gain, $KGH(j\omega)$, can be used to determine the stability of the closed-loop system. In particular, the root condition on the closed-loop characteristic polynomial implies: $1+KGH(j\omega)=0$, or $KGH(j\omega)=-1$.

Hence stability of the closed-loop system can be inferred from the frequency response of $KGH(j\omega)$, relative to the $1\angle\pm 180^\circ$ point on the complex plane. The stability determination is performed using the following relative stability criteria:

Definition: Gain Margin

The gain margin (GM) denotes the factor by which the loop gain, $|KGH(j\omega)|$, can be increased without compromising the closed-loop stability.

The GM is computed as: $GM=-|KGH(j\omega_{pc})|_{dB}$, where ω_{pc} denotes the phase crossover frequency defined by $\angle KGH(j\omega)=-180^\circ$.

A positive value of GM denotes closed-loop stability.

Definition: Phase Margin

The phase margin (PM) denotes the additional phase that can be added by the controller to $\angle KGH(j\omega)$ without compromising the closed-loop stability.

The PM is computed as: $PM=\angle KGH(j\omega_{gc})+180^\circ$, where ω_{gc} denotes the gain crossover frequency defined by $|KGH(j\omega_{gc})|_{dB}=0dB$.

A positive value of PM denotes closed-loop stability. Additionally, PM represents a measure of dynamic stability; hence adequate PM is desired to suppress oscillations in the output response.

To proceed further, assume that the loop transfer function, $KGH(s)$, has m zeros and n poles. Then,

1. For low-order models ($n\leq 2$), the loop gain, $KGH(s)$, displays an infinite GM.
2. GM is finite for $n-m>2$ and may be so for $n-m=2$.

The gain crossover occurs when $|KGH(j0)|_{dB}>0dB$; hence PM become relevant only in that case.

Stability Determination on Bode Plot

On the Bode magnitude plot the gain crossover frequency, ω_{gc}, is indicated as the magnitude plot crosses the $0dB$ line. The Bode phase plot at that frequency reveals the phase margin as: $PM=\angle KGH(j\omega_{gc})+180^\circ$.

On the Bode phase plot, phase crossover frequency, ω_{pc}, is indicated as the phase plot crosses the -180° line. The magnitude plot at that frequency reveals the gain margin as: $GM=-|KGH(j\omega_{pc})|_{dB}$.

The relative stability margins can be obtained in the MATLAB Control Systems Toolbox by using the 'margin' command. When invoked the command produces a Bode plot with stability margins indicated.

Example 4.1.1

Let $KG(s)=\frac{K}{s(s+1)(s+2)}$; the Bode magnitude and phase plots for $K=1$ are shown in Figure 4.1.2.

The Bode magnitude plot displays a $15.6\,dB$ gain margin, i.e., the controller gain can be increased by a factor of $6\,(15.6\,dB)$ before losing stability of the closed-loop system.

The Bode phase plot displays a 53.4° phase margin, which indicates closed-loop stability; further, it corresponds to $\zeta\cong 0.55$ for the closed-loop system, which indicates adequate dynamic stability.

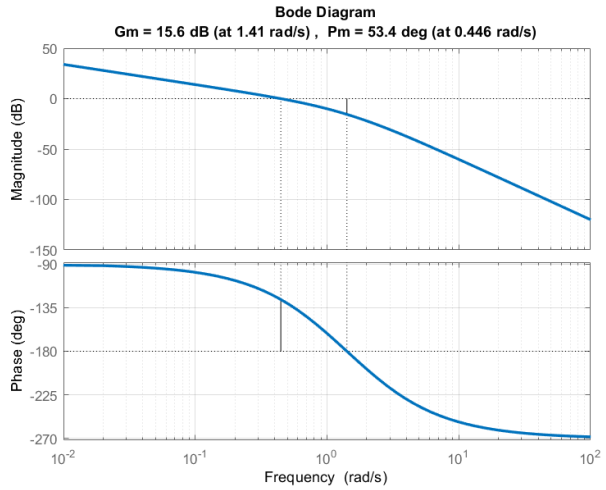


Figure 4.1.2: Bode gain and phase plots showing stability margins.

Stability Determination on Nyquist Plot

On the Nyquist plot, the gain crossover is indicated when the Nyquist plot enters the unit circle, whereas phase crossover is indicated when the Nyquist plot crosses the negative real-axis.

Thus, GM is positive if the real-axis crossing is to the right of $-1+j0$ point. The PM is positive if $KGH\left(j\omega\right)$ magnitude drops below unity in the third quadrant.

Example 4.1.2

Let $KG\left(s\right)=\frac{K}{s(s+1)(s+2)}$; the Nyquist plot for $K=1$ is shown in Figure 4.1.3.

The Nyquist plot crosses the real axis at $s=-0.165$, which corresponds to $GM=6$.

The Nyquist plot enters the unit circle at an angle of 53° from the negative real axis that indicates the phase margin.

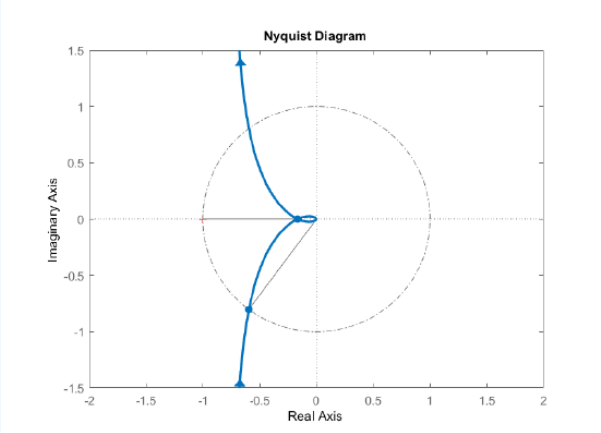


Figure 4.1.3: Nyquist plot showing stability margins.

Example 4.1.3

The model of a small DC motor, including an amplifier with a gain of 3 is given as: $\frac{\omega(s)}{V_a(s)}=\frac{1500}{(s+100)(s+10)+25}$.

The relative stability margins for the DC motor model are given as: $GM=\infty; PM=127^{\circ}$ (Figure 4.1.4).

Assume that the motor is used in a position control application, so that the plant transfer function is given as: $\frac{\theta(s)}{V_a(s)} = \frac{1500}{s[(s+100)(s+10)+25]}$.

The corresponding relative stability margins are given as: $GM = 37.5\ dB;\ PM = 81.1^\circ$ (Figure 4.1.4).

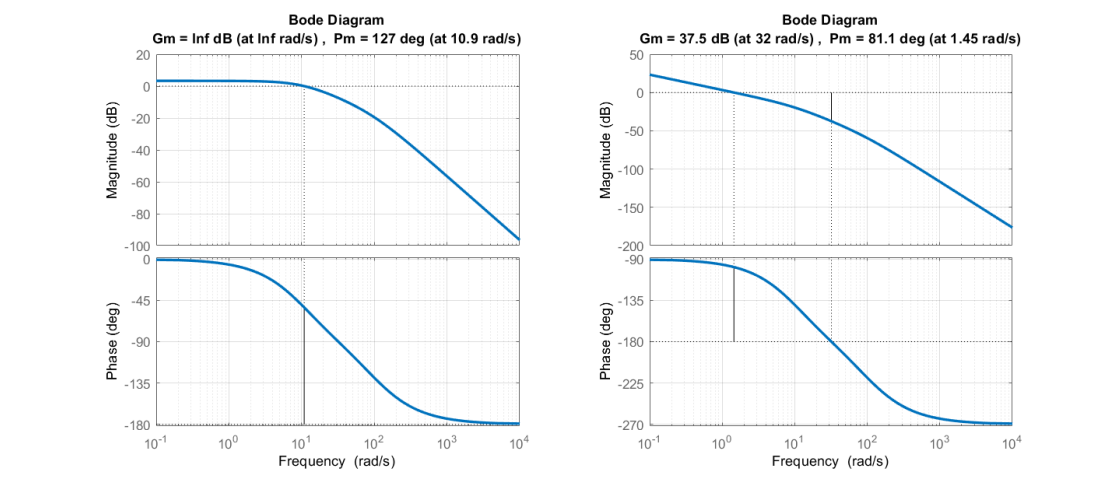


Figure 4.1.4: Stability margins for the DC motor model (left); stability margins for the position control application (right).

4.2: Transient Response Improvement

Transient Response Improvement

We consider the feedback control system (Figure 4.2.1) with input $r(s)$ and output $y(s)$.

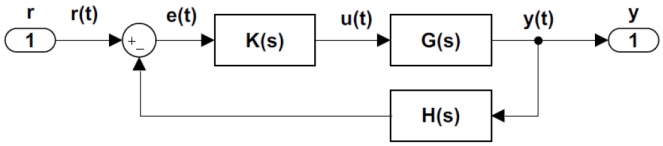


Figure 4.2.1: Copy and Paste Caption here. (Copyright; author via source)

The closed-loop system response is given as: $y(s) = T(s)r(s)$, where $T(s) = \frac{KG(s)}{1+KGH(s)}$.

The system response consists of transient and steady-state components, i.e., $y(t) = y_{tr}(t) + y_{ss}(t)$.

In particular, for a constant input, r_{ss}, the steady-state component of the system response is given as: $y_{ss} = T(0)r_{ss}$.

The transient response is characterized by the roots of the closed-loop characteristic polynomial, given as: $\Delta(s) = 1 + KGH(s)$.

These roots can be real or complex, distinct or repeated. Accordingly, the system natural response modes are characterized as follows:

Real and Distinct Roots. Let $\Delta(s) = (s-p_1)(s-p_2)\ldots(s-p_n)$. ; then, the system natural response modes are given as: $\{e^{p_1 t}, e^{p_2 t}, \ \ldots\ , e^{p_n t}\}$.

Real and Repeated Roots. Let $\Delta_m(s) = (s-s_1)^m$; then, the corresponding natural response modes is given as: $\{e^{s_1 t},\ te^{s_1 t},\ \ldots,\ t^{m-1}e^{s_1 t}\}$.

Complex Roots. Let $\Delta_c(s) = (s+\sigma)^2 + \omega^2$; then, the corresponding natural response modes are given as: $\{e^{-\sigma t}\cos\omega t\ ,\ \ e^{-\sigma t}\sin\omega t\}$.

General Expression. Assume that for an nth order polynomial, the response modes are given as: $\phi_k(t),\ k = 1, \ldots, n$. Then, using arbitrary constants, c_k, the system natural response is given as: $y(t) = \sum_{k=1}^{n} c_k \phi_k(t)$.

Control System Design Specifications

The control system design specifications include desired characteristics for the transient and steady-state components of system response with respect to a prototype input. A step input is used to define the desired transient response characteristics.

The qualitative indicators of the step response include the following:

1. The rise time (t_r)
2. The peak time (t_p)
3. The response peak (M_p) and percentage overshoot (%OS),
4. The settling time (t_s)

Rise Time. For overdamped systems ($\zeta > 1$), the rise time is the time taken by the response to reach from 10% to 90% of its final value. For underdamped systems ($\zeta < 1$), the rise time is the time when the response first reaches its steady-state value.

Peak Time. For underdamped systems, the peak time is the time when the step response reaches its peak.

Peak Overshoot. The peak overshoot is the overshoot above the steady-state value.

Settling Time. The settling time is the time when the step response reaches and stays within 2% of its steady-state value. Alternately, 1% limits can be used.

Prototype Second-Order System

To illustrate the above characteristics, we consider a prototype second-order transfer function, given by the closed-loop transfer function as:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The poles for the prototype system are located at: $s = -\zeta\omega_n \pm j\omega_d$, where $\omega_d = \omega_n\sqrt{1-\zeta^2}$ is the damped natural frequency.

The impulse and the step response of the prototype system are given as:

$$y_{imp}(t) = \frac{\omega_n^2}{\omega_d} e^{-\zeta\omega_n t}\sin(\omega_d t)\, u(t)$$

$$y_{step}(t) = T(0)\left(1 - e^{-\zeta\omega_n t}\sin\left(\omega_d t + \phi\right)\right) u(t), \quad \phi = \tan^{-1}\frac{\omega_d}{\zeta\omega_n}$$

For the prototype system, the rise time, t_r, is indicated by $\omega_d t_r + \phi = \pi$. Thus, $t_r = \frac{\pi-\phi}{\omega_d}$; the rise time is bounded as: $\frac{\pi}{2\omega_d} \le t_r \le \frac{\pi}{\omega_d}$.

The peak time is indicated by $\omega_d t_p = \pi$; thus, $t_p = \frac{\pi}{\omega_d}$.

The peak overshoot is given as: $M_p = 1 + e^{-\zeta\omega_n t_p}$; the percentage overshoot is given as: $\%OS = 100e^{-\zeta\omega_n t_p}$.

The effective time constant of the prototype system is: $\tau = \frac{1}{\zeta\omega_n}$. The step response reaches and stays within 2% of its final value in about 4τ, and within 1% of its final value in about 4.5τ.

These metrics are summarized in the Table below.

Table 2.1: Quality metrics used to measure second-order system response.

Quality indicator	Expression
Rise time	$t_r = \frac{\pi-\phi}{\omega_d}, \quad \phi = tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}$
Peak time	$t_p = \frac{\pi}{\omega_d}$
Peak overshoot	$M_p = 100e^{-\zeta\omega_n t_p}(\%)$
Settling time	$t_s \cong \frac{4.5}{\zeta\omega_n}$

We may note that the above step response quality indicators are all functions of the damping ratio, ζ, of the closed-loop roots.

To clarify this relationship, the variation in the step overshoot, $\%OS$, and the normalized rise time, $\omega_n t_r$, with ζ is tabulated below for selected values of ζ.

Table 2.2: Percentage overshoot and rise time vs. damping ratio

ζ	0.9	0.8	0.7	0.6	0.5
$\%OS$	0.2%	1.5%	4.6%	9.5%	16.3%
$\omega_n t_r$	5.54	4.18	3.3	2.76	2.42

A low ($\le 10\%$) overshoot in the step response is often desired, which translates into $\zeta \ge 0.6$.

Example 4.2.1

For a prototype second-order system, let $T(s) = \frac{25}{s^2+6s+25}$. The closed-loop poles are located at: $p_{1,2} = -3 \pm j4$, hence $\omega_n = 5\frac{rad}{s}$ and $\zeta = 0.6$.

The step response of the system displays: $t_r \cong 0.47\ sec,\ t_p \cong 0.78\ sec,\ \%OS = 9\%$, and $t_s \cong 1.19\ sec$ (2% threshold).

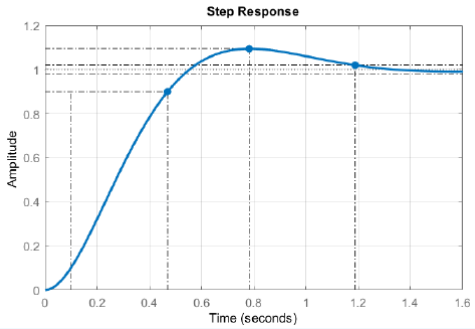


Figure 4.2.1: Step response of a prototype second-order system showing rise time, peak time and settling time.

Desired Closed-Loop Characteristic Polynomial

The system design specifications, expressed in terms of rise time (t_r), settling time (t_s), damping ratio (ζ), and percentage overshoot ($\%OS$), are used to define desired root locations for the closed-loop characteristic polynomial.

In particular, assuming a root root location, $-\sigma \pm j\omega$, the following restrictions are placed on the real and imaginary parts:

1. The settling time constraint places a bound on the real part of the roots: $\sigma \geq \frac{4.5}{t_s}$.
2. The damping ratio constraint requires that: $\theta \leq \pm \cos^{-1} \zeta$, where θ is the angle of the desired root location from the origin of the complex plane.
3. The rising time constraint places a bound on the natural frequency of the closed-loop roots as $\omega_n \geq \frac{2}{t_r}$.

These constraints are summarized below:

$$\sigma \geq \frac{4.5}{t_s}, \quad \omega_n \geq \frac{2}{t_r}, \quad \theta \leq \cos^{-1} \zeta$$

Example 4.2.2

We assume the following design specifications: $0.7 < \zeta < 1$ and $t_s < 2s$. Then, the desired region for the closed-loop root locations is bounded by: $\sigma \geq 2$ and $\theta \leq \pm 45^\circ$.

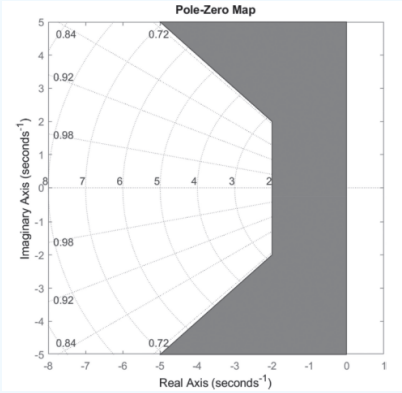


Figure 4.2.2: The desired region for closed-loop pole placement for $0.7 < \zeta < 1$ and $t_s < 2$s.

Example 4.2.3

Assume that the design specifications are specified as: $t_r \leq 0.5s$, $t_s \leq 3.0s$, and % $\%OS \leq 10\%$. These specifications place the following constraints on the closed-loop pole locations: $\sigma \geq 1.5$, $\omega_n \geq 4$, and $\zeta \geq 0.6$.

To meet the constraints, we may choose, e.g., the desired closed-loop characteristic polynomial as: $\Delta_{des}(s) = (s+\sigma)^2 + \omega_d^2 = s^2 + 6s + 20$.

Controller Gain Selection

The closed-loop characteristic polynomial for a unity-gain feedback system includes static controller gain, K, as a parameter. The characteristic polynomial is given as: $\Delta(s,K)=1+KG(s)$.

Given a desired characteristic polynomial, $\Delta_{des}(s)$, we may choose the controller gain by comparing the coefficients of the two polynomials.

Example 4.2.4

The simplified model of a small DC motor is given as: $G(s)=\frac{\theta(s)}{V_a(s)}=\frac{10}{s(s+6)}$. Assuming a static gain controller in unity-gain feedback configuration, the closed-loop characteristic polynomial is given as: $\Delta(s)=s^2+6s+10K$.

Suppose the design specifications are given as: $OS\leq 10\%\ (\zeta\geq 0.6)$, $t_s\leq 1.5sec.$ Then, closed-loop root locations may be selected as: $s=-3\pm j4$.

The desired characteristic polynomial is formed as: $\Delta_{des}(s)=s^2+6s+25$.

By comparing polynomial coefficients, we obtain: $K=2.5$.

The closed-loop system step response shows a rise time $t_r\cong 0.47\ sec$ ($\omega_n t_r\cong 3$), and the settling time $t_s\cong 1.06\ sec$.

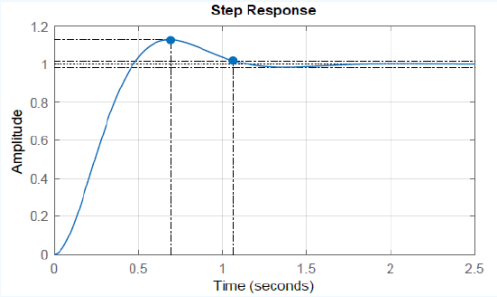


Figure 4.2.3: Step response of the closed-loop simplified DC motor model (Example 4.2.4).

The next example illustrates the design of rate feedback controller.

Example 4.2.5

The characteristic polynomial for rate feedback controller design is given as: $\Delta(s,K,K_1)=s^2+K_1s+K$.

Assume that the design objectives are: $\%\ OS\leq 5\%\ (\zeta=0.7)$ and $t_s\leq 2.5s$.

A desired characteristic polynomial for $\zeta=0.7$ and $\zeta\omega_n=2$ is given as: $\Delta_{des}(s)=(s+2)^2+2^2$. Accordingly, we may choose: $K_1=4, K=8$.

Optimal Performance Indices

An alternate way to characterize the transient response of the closed-loop system is to define a time-domain performance index and choose a characteristic polynomial to minimize that index.

Toward this end, three popular performance indices have been defined:

Integral Absolute Error, IAE : $\int_0^{t_s}|e(t)|\,dt$

Integral Square Error, ISE : $\int_0^{t_s}|e(t)|^2dt$

Integral Time Absolute Error, $ITAE$: $\int_0^{t_s}t\,|e(t)|\,dt$.

The ITAE index, in particular, is commonly used to evaluate system performance in industrial process control.

For a prototype second-order system model, driven by a step input, minimization of the ITAE index results in an optimal design with $\zeta=0.7$ ($\%OS=4.6\%$).

The optimum coefficients of the desired characteristic polynomials based on the ITAE index are given below:

$$s+\omega_n$$

$$s^2+1.4\omega_n s+\omega_n^2$$

$$s^3+1.75\omega_n s^2+2.15\omega_n^2 s+\omega_n^3$$

$$s^4+2.1\omega_n s^3+3.4\omega_n^2 s^2+2.7\omega_n^2 s+\omega_n^3.$$

Example 4.2.6

The first-order-plus-dead-time (FOPDT) model of an industrial process is given as: $G(s)=\frac{e^{-s}}{s+1}$. By using first-order Pade' approximation: $e^{-s}\cong\frac{2-s}{2+s}$, we obtain a rational approximation as: $G(s)\cong\frac{2-s}{(s+1)(s+2)}$.

The closed-loop characteristic polynomial for a unity-gain feedback control system is: $\Delta(s)=(s+1)(s+2)+K(2-s)$.

Using the ITAE index, a desired second-order polynomial is selected as: $\Delta_{des}(s)=s^2+1.4\omega_n s+\omega_n^2$.

By comparing the polynomial coefficients, we obtain two equations that are solved to obtain: $\omega_n=1.76\frac{rad}{s}$ and $K=0.54$.

4.3: Steady-State Error Improvement

Steady-State Tracking Error

A tracking control system is designed to have a low steady-state error in response to a constant (i.e., unit-step) or linearly varying (i.e., unit-ramp) input. The desired error tolerance may be specified as percentage of the reference input. Additionally the system response to a constant disturbance input should stays small.

We consider the feedback control system (Figure 4.3.1) with input $r(s)$ and output $y(s)$.

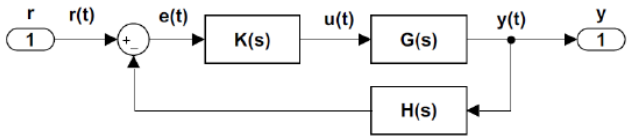


Figure 4.3.1: Copy and Paste Caption here. (Copyright; author via source)

Let $T(s)$ denote the closed-loop transfer function; then $y(s) = T(s)r(s)$. The tracking error, $e(s)$, in response to a reference signal, $r(s)$, is defined as:

$$e(s) = (1 - T(s))r(s)$$

To characterize the tracking error, we consider a unity-gain feedback system ($H(s) = 1$). Then, the tracking error, $e(s)$, is given as:

$$e(s) = \frac{1}{1 + KG(s)} r(s)$$

By using the final-value theorem (FVT), the steady-state tracking error is computed as: $e(\infty) = \lim_{s \to 0} se(s)$

The steady-state error to a step reference input, $\left(r(s) = \frac{1}{s}\right)$, is given as: $e(\infty) = \frac{1}{1+KG(0)}$.

The steady-state error to a ramp reference input, $\left(r(s) = \frac{1}{s^2}\right)$, is given as: $e(\infty) = \frac{1}{sKG(s)|_{s=0}}$.

System Error Constants

The steady-state tracking error in the case of a unity-gain feedback control system is characterized in terms of system position and velocity error constants, which are defined as:

$$K_p = \lim_{s \to 0} KG(s)$$

$$K_v = \lim_{s \to 0} sKG(s)$$

In terms of the error constants, the steady-state tracking error to a step or a ramp input is evaluated as:

$$e(\infty)|_{step} = \frac{1}{1 + K_p}$$

$$e(\infty)|_{ramp} = \frac{1}{K_v}$$

Example 4.3.1

Let $KG(s) = \frac{K}{s(s+2)}$; then we have: $K_p = \infty, \quad K_v = \frac{K}{2}$.

Hence $e(\infty)|_{step} = 0, \; e(\infty)|_{ramp} = \frac{2}{K}$.

We note that the presence of an integrator in the loop forces $K_p \to \infty$, and the steady-state error to a step input to go to zero.

Example 4.3.2

The approximate model of a small DC motor is given as:

$$KG(s) = \frac{500K}{(s+10)(s+100)}$$

Then we have: $K_p = 0.5K$, and $K_v = 0$.

Hence $e(\infty)|_{step} = \frac{2}{K}$, $e(\infty)|_{ramp} = \infty$.

Integral control, $K(s) = \frac{k_i}{s}$, is often employed to reduce the steady-state tracking error. The presence of an integrator in the loop forces $K_p \to \infty$, and the steady-state error to a step input to go to zero.

Steady-State Error to Ramp Input

Assuming that the feedback loop contains an integrator, so that the steady-state error to a step input is zero, the steady-state error to a ramp input is expressed as:

$$e(s) = [1 - T(s)]\, r(s); \quad r(s) = \frac{1}{s^2}.$$

The error is evaluated by the application of FVT: $e(\infty)|_{ramp} = \lim_{s\to 0} \frac{1-T(s)}{s}$.

By using the L' Hospital's rule, we obtain: $e(\infty)|_{ramp} = \lim_{s\to 0} \left(-\frac{dT(s)}{ds}\right)$.

To evaluate the RHS, we use the natural logarithm to write: $\frac{d}{ds}\ln T(s) = \frac{1}{T(s)}\frac{dT(s)}{ds}$.

Since $\lim_{s\to 0} T(s) = 1$ by the integrator assumption, we have: $\lim_{s\to 0} \frac{d}{ds}\ln T(s) = \lim_{s\to 0} \frac{dT(s)}{ds}$.

Next, assume that $T(s)$ is expressed as: $T(s) = \frac{K(s-z_1)\ldots(s-z_m)}{(s-p_1)\ldots(s-p_n)}$.

Then, we have: $\ln T(s) = \ln K + \sum_{ln}(s-z_i) - \sum_{ln}(s-p_i)$, and

$$\frac{d}{ds}\ln T(s) = \ln K + \sum \frac{1}{s-z_i} - \sum \frac{1}{s-p_i}$$

It follows that:

$$e(\infty)|_{ramp} = \sum \frac{1}{p_i} - \sum \frac{1}{z_i}$$

where z_i and p_i denote the zeros and poles of the closed-loop system.

Example 4.3.3

Let $KG(s) = \frac{K}{s(s+2)}$; then, the closed-loop characteristic polynomial is: $\Delta(s) = s^2 + 2s + K$.

Assuming a value of $K = 1$, the closed-loop poles are located at: $s_{1,2} = -1, -1$.

The resulting steady-state error to a ramp input is given as: $e(\infty)|_{ramp} = \sum \frac{1}{p_i} = 2$.

For verification, the closed-loop system response is plotted in Figure 4.3.1.

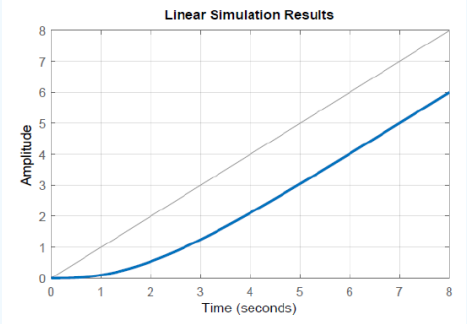


Figure 4.3.1: Steady-state error to a ramp input (Example 4.3.3).

4.4: Disturbance Rejection

System Response to Disturbance Inputs

Disturbances are unwanted signals entering into a feedback control system. A disturbance may act at the input or output of the plant. Here we consider the effect of input disturbance.

To characterize the effect of a disturbance input on the feedback control system, we consider the modified block diagram (Figure 4.4.1) that includes a disturbance input.

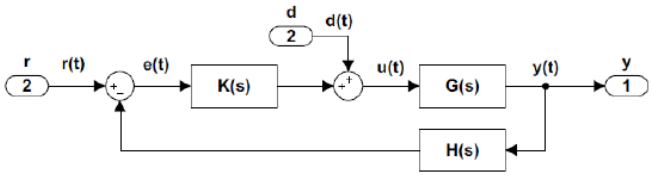


Figure 4.4.1: A feedback control system with reference and disturbance inputs.

Let $r(t)$ denote a reference, and $d(t)$ a disturbance input; then the system output is expressed in the Laplace domain as:

$$y(s)=\frac{KG(s)}{1+KGH(s)}r(s)+\frac{G(s)}{1+KGH(s)}d(s)$$

Assuming unity-gain feedback configuration $(H(s)=1)$, the tracking error, $e(s)$, is computed as:

$$e(s)=\frac{1}{1+KG(s)}r(s)-\frac{G(s)}{1+KG(s)}d(s)$$

By using the FVT, the steady-state error is expressed as:

$$e(\infty)=\frac{1}{1+K_p}r(\infty)-\frac{G(0)}{1+K_p}d(\infty)$$

where K_p is the position error constant.

A large loop gain (large K_p) reduces steady-state error in the presence of both reference and disturbance inputs. A large controller gain, K, can be used to increase K_p, however, a large K would generate a large magnitude input signal to the plant, which may cause saturation in the actuator devices (amplifiers, mechanical actuators, etc.).

Simultaneous Tracking and Disturbance Rejection

To analyze the control requirements for simultaneous tracking and disturbance rejection, we consider a unity-gain feedback control system ($H(s)=1$).

Let $G(s)=\frac{n(s)}{d(s)}$ represent the plant and $K(s)=\frac{n_C(s)}{d_C(s)}$ represent the controller; then, the output in the presence of reference and disturbance inputs is given as:

$$y(s)=\frac{d(s)\,d_c(s)}{n(s)\,n_c(s)+d(s)\,d_c(s)}r(s)-\frac{n(s)\,d_c(s)}{n(s)\,n_c(s)+d(s)\,d_c(s)}d(s)$$

The characteristic polynomial is given as: $\Delta(s)=n(s)n_c(s)+d(s)d_c(s)$.

The requirements for asymptotic tracking and disturbance rejection are stated as follows:

Asymptotic tracking. For asymptotic tracking, $d(s)\,d_c(s)$ should contain any unstable poles of $r(s)$. For example, an integrator in the feedback loop ensures zero steady-state error to a constant reference input.

Disturbance Rejection. For disturbance rejection, $n(s)\,d_c(s)$ should contain any unstable poles of $d(s)$. For example, a notch filter centered at $60\ Hz$ removes power line noise from the measured signal.

Example 4.4.1

A small DC motor has the following component values: $R = 1\ \Omega,\ L = 10\ \text{mH},\ \text{J} = 0.01\ \text{kgm}^2,\ \text{b} = 0.1\ \frac{\text{Ns}}{\text{rad}}, \text{k}_\text{t} = \text{k}_\text{b} = 0.05.$

The DC motor transfer function is given as: $G(s) = \frac{500}{s^2+110s+1025}$.

Since $G(0) \cong 0.5$, with a static controller, the position error constant is: $K_p = 0.5K$. The steady-state error in the presence of reference and disturbance inputs is given as:

$$e(\infty) = \frac{1}{1+0.5K} r(\infty) - \frac{0.5}{1+0.5K} d(\infty).$$

We may choose a large K to reduce the steady-state tracking error as well as improve the disturbance rejection. A large value of K, however, reduces system damping and results in an oscillatory response of the DC motor.

For a static controller, the closed-loop characteristic polynomial is given as: $\Delta(s, K) = s^2 + 110s + 1025 + 500K$. The resulting damping ratio is: $\zeta = \frac{55}{\sqrt{1025+500K}}$.

In order to limit the damping to, say $\zeta \leq 0.6$, the controller gain is limited to: $K \leq 14.75$. We may choose, for example, $K = 14$, which results in the following steady-state error to reference and disturbance inputs: $e(\infty) = \frac{1}{8} r(\infty) - \frac{1}{16} d(\infty)$.

4.5: Sensitivity and Robustness

Sensitivity and Robustness

The robustness refers to the ability of a control system to withstand parameter variations in the plant transfer function, and still maintain the stability and performance goals.

Robustness is characterized in terms of the sensitivity of the closed-loop transfer function $T(s)$ to variation in one or more of the plant parameters.

System Sensitivity Function

The system sensitivity function is defined as the ratio of percentage change in the closed-loop transfer function to percentage change in the plant transfer function, i.e., assuming unity feedback ($H(s) = 1$),

$$S_G^T = \frac{\partial T/T}{\partial G/G} = \frac{\partial T}{\partial G}\frac{G}{T} = \frac{1}{1+KG(s)}$$

Hence, increasing loop gain in a feedback control system by choosing a larger controller gain, K, reduces its sensitivity to parameter variations.

It may be noted that the system sensitivity function, denoted as $S(s)$, is the same as the transfer function from the error, $e(s)$, to output, $y(s)$. The sensitivity function is, thus, a function of frequency. Further, $S(s) + T(s) = 1$, where $T(s)$ denotes the closed-loop transfer function.

Parameter Sensitivity

The sensitivity of the closed-loop transfer function, $T(s)$, to percentage change in a plant parameter, α, is given as:

$$S_\alpha^T = S_G^T S_\alpha^G$$

Example 4.5.1

Let $KG(s) = \frac{K}{s(s+a)}$, $H(s) = 1$; then, we have: $T(s) = \frac{K}{s(s+a)+K}$.

For unity-gain feedback, the system sensitivity function is computed as: $S_G^T = \frac{s(s+a)}{s(s+a)+K}$.

Further, $S_a^G = -\frac{a}{s+a}$; hence $S_a^T = -\frac{as}{s(s+a)+K}$.

Thus, increasing the loop gain (by choosing a large K) reduces the sensitivity of $T(s)$ to variations in a. However, increasing K reduces damping in the system and gives rise to output oscillations.

Example 4.5.2

The model of a small DC motor is approximated as: $G(s) = \frac{k_t}{(Js+b)(Ls+R)+k_t k_b} = \frac{500}{s^2+110s+1025}$.

For unity-gain feedback, the system sensitivity function is computed as:

$$S_G^T = \frac{1}{1+KG(s)} = \frac{s^2+110s+1025}{s^2+110s+500K+1025}$$

For $K = 10$, we have $S_G^T = \frac{s^2+110s+1025}{s^2+110s+6025}$.

Suppose we want to compute the sensitivity of $T(s)$ to motor torque constant, k_t; then

$$S_{k_t}^G = \frac{s^2+110s+1000}{s^2+110s+1000+500k_t}$$

For $k_t = 0.05$, we obtain $S_{k_t}^G = \frac{s^2+110s+1000}{s^2+110s+1025}$, and hence

$$S_{k_t}^T = S_{k_t}^G S_G^T = \frac{s^2+110s+1000}{s^2+110s+6025}$$

A Bode magnitude plot of the sensitivity function is shown in Fig. 4.5.1.

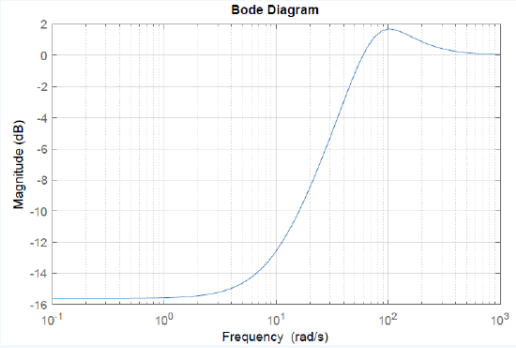


Figure 4.5.1: Bode magnitude plot of the sensitivity function.

CHAPTER OVERVIEW

5: Control System Design with Root Locus

Learning Objectives

1. Sketch the root locus of a transfer function with respect to the controller gain K.
2. Use root locus technique to design a static controller for a transfer function model.
3. Use root locus technique to design first-order phase-lead and phase-lag controllers.
4. Realize the controller design using operational amplifiers and filters.

5.0: Prelude to Control System Design with Root Locus

The typical structure of a feedback control system (Figure 5.0.1) includes the process to be controlled, represented by $G(s)$, a sensor that measures the output, represented by $H(s)$, and a cascade controller, represented by $K(s)$. The plant is assumed to be of single-input single-output (SISO) type.

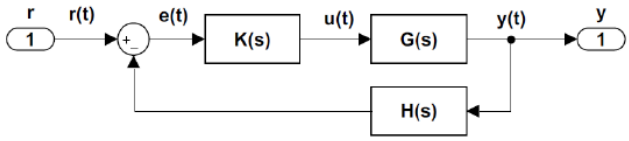


Figure 5.0.1: Copy and Paste Caption here. (Copyright; author via source)

Assuming a static gain controller is selected, the characteristic polynomial includes the controller gain, K, as a parameter. The root locus (RL) refers to the locus of the closed-loop roots as the controller gain, K, varies from $0 \rightarrow \infty$. The root locus technique allows graphical representation of achievable closed-loop pole locations.

The RL plot in the case of low-order plant models can be sketched by hand following a set of rules; a computer generated plot can be obtained from MATLAB. The controller gain needed to achieve a set of design specifications can be selected from the RL plot. In MATLAB, the controller gain can be read by clicking on the RL plot.

The feedback control system is designed for closed-loop stability and the desired transient and/or steady-state response improvements. The transient response improvements are measured in terms of step response parameters (t_r, t_s, $\%OS$, etc.); the steady-state response improvements are measured in terms of relevant error constants (K_p, K_v).

The root locus plot of a given loop transfer function, $KGH(s)$, constitutes n branches in the complex s-plane, where n is the order of the denominator polynomial. These branches commence at the open-loop (OL) poles and proceed towards finite zeros of the loop transfer function, or asymptotically toward infinity. All root locus plots bear common characteristics that are referred as root locus rules. These describe the location of real-axis locus, break points, asymptote directions, etc.

The root locus design technique can be extended to dynamic controllers of phase-lead, phase-lag, lead–lag, PD, PI, and PID types. The dynamic controller introduces poles and zeros that modify the original RL to bring about the desired design improvements. With the availability of a rate sensors (rate gyroscope and tachometer), rate feedback design that includes the design of inner and outer feedback loops may be considered.

The static and dynamic controllers designed for process improvement can be implemented with electronic circuits built with operational amplifiers, resistors, and capacitors. Alternatively, controller realization using microcontrollers can be considered. This topic is addressed in the next chapter after introduction of the sampled-data systems.

5.1: Root Locus Fundamentals

Root Locus

The root locus (RL) constitutes a graph of the closed-loop root locations, with variation in static feedback controller gain, K.

In order to develop the RL concepts, we consider a typical feedback control system (Figure 5.1), where K represents a controller, $G(s)$ is the plant transfer function, and $H(s)$ is the sensor transfer function. Unless noted otherwise, unity-gain feedback ($H(s)=1$) is assumed.

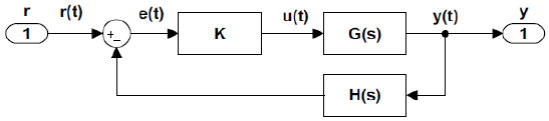


Figure \(\PageIndex{1}\): Feedback control system with plant $G(s)$, sensor $H(s)$, and a static gain controller represented by K.

The loop gain for the feedback loop includes the controller, the plant, and the sensor transfer functions, and is given as: $$L(s)=KG(s)H(s)=KGH(s)$$

The characteristic polynomial of the closed-loop system is defined as: $$\Delta (s,K)=1+KGH(s)$$

The roots of $\Delta (s,K)$ vary with the controller gain, K, and are plotted for different values of K. The loci of all such roots locations, as K varies from $0\to \infty ,$ constitutes the root locus plot of the loop transfer function.

Example \(\PageIndex{1}\)

Let $G(s)=\frac{1}{s+1} ,\; H(s)=1$; the closed-loop characteristic polynomial is given as: $\Delta (s)=s+1+K.$ The polynomial has a single root at $s_{1} =-1-K$.

A short table of closed-loop root for different values of K is given below:

K	0	1	2	5	10
Closed-loop root	-1	-2	-3	-6	-11

The root locus plot, as K varies from 0 to ∞, traces a line that proceeds from $\sigma =-1$ along the negative real-axis to infinity (Figure 5.1.2). The overlaid grid shows the damping ratio of the closed-loop root ($\zeta=1$ in this case).

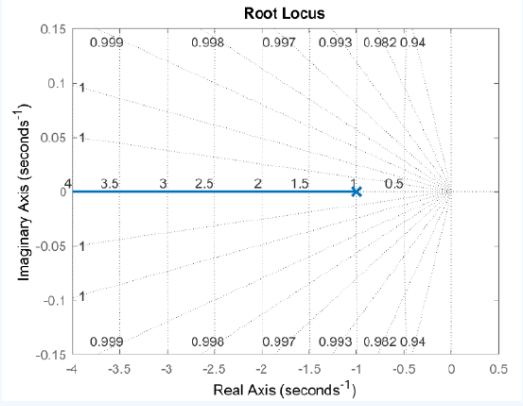


Figure \(\PageIndex{2}\): The root locus plot for $G(s)=\frac{1}{s+1} ,\; H(s)=1$

```
pkg load control
s=tf('s');
Gs=1/(s+1);
rlocus(Gs)
```

run | restart | restart & run all

The MATLAB Control Systems Toolbox provides the ‘rlocus’ command to plot the root locus of the loop transfer function. The ‘rlocus’ command is invoked after defining a dynamic system object using ‘tf’ or ‘zpk’ command.
The ‘grid’ command adds constant damping ratio and constant natural frequency contours to the RL plot. These contours help design a static controller for desired values of those parameters.

Example \(\PageIndex{2}\)

Let \(G(s)=\frac{1}{s(s+2)} ,\; \; H(s)=1\). The closed-loop characteristic polynomial is given as: \(\Delta (s,K)=s^{2} +2s+K.\)
The roots of the characteristic polynomial are given as: \(s_{1,2}=-1\pm \sqrt{1-K}\) for \(K\le 1\), and as \(s_{1,2} =-1\pm j\sqrt{K-1}\) for \(K>1\). The roots for selected values of \(K\) are tabulated below.

\(K\)	\(0\)	\(1\)	\(2\)	\(10\)
Closed-loop roots	\(-1,-2\)	\(-1,-1\)	\(-1\pm j1\)	\(-1\pm j3\)

The loci of these roots, as \(K\) varies from \(0\to \infty\), comprise two branches that commence at the open-loop (OL) poles located at \(\{ 0,\; -2\}\), proceed inward along the real-axis to meet at \(\sigma =-1\), then split and extend along the \(\sigma =-1\) line to \(s=-1\pm j\infty\) (Figure 5.2). These directions are called the RL asymptotes.

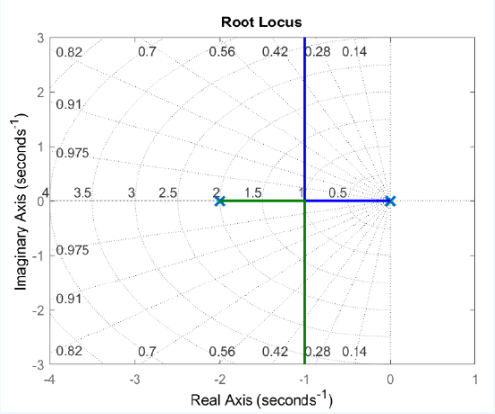


Figure \(\PageIndex{3}\): The root locus plot for \(G(s)=\frac{1}{s(s+2)} ,\; \; H(s)=1\).

```
pkg load control
s=tf('s');
Gs=1/s/(s+1);
rlocus(Gs)
```

run | restart | restart & run all

The RL plot can be extended to negative values of \(K\), i.e., \(K\in (-\infty ,\; \infty)\), and is called generalized root locus.

Root Locus Rules

All RL plots share a few common properties, referred as the root locus rules. The prominent rules are described below.
In the following, it is assumed that the loop transfer function, \(KGH(s)\), is proper, has \(n\) open loop (OL) poles and \(m\le n\) finite OL zeros. The remaining \(n-m\) zeros are assumed to lie at infinity.

1. The RL has \(n\) branches and \(n-m\) asymptotes (assumed RL directions for large \(K\)). The RL branches commence at the OL poles (for \(K=0\)); of these, \(m\) branches terminate at the OL zeros (as \(K\to \infty\)); the remaining \(n-m\) branches follow the RL asymptotes to infinity (as \(K\to \infty\)).
2. The real-axis locus, indicating real roots of the characteristic polynomial, lies to the left of an odd number of poles and zeros of \(KGH(s)\) for \(\left(K>0\right)\), and to the right of an odd number of poles and zeros for \(\left(K<0\right)\).
3. The real-axis locus separating a pair of OL poles contains a break-away point where the two RL branches split; the real-axis locus separating two OL zeros, contains a break-in point where the two branches meet. The break points for both \(K>0\) and \(K<0\) are found among the solutions to the equation: \(\sum \frac{1}{\sigma -p_{i} } -\sum \frac{1}{\sigma -z_{i} } =0\).

4. The RL asymptotes, assume angles: \(\phi _{a} =\frac{2k+1}{n-m} (180^{\circ }),\; k=0,\; 1,\cdots ,n-m-1\), (for\(K>0\)), and angles: \(\phi _{a} =\frac{2k+1}{n-m} (360^{\circ })\), \(k=0,\; 1,\cdots ,n-m-1\) (for \(K<0\)). These asymptotes intersect at a common point on the real-axis, given as: \(\sigma _a =\frac{\sum p_{i} -\sum z_{i} }{n-m}\).
5. The RL plot is symmetric with respect to the real-axis (this is a consequence of the fact that the complex roots of a real polynomial occur in conjugate pairs).

The above rules suffice to sketch the RL plot for low-order plants. Additional rules for refining the RL plot may be added.

Example \(\PageIndex{3}\)

Let \(KGH(s)=\frac{K(s+3)}{s(s+2)}\); hence, \(n=2,\ \ m=1\).The closed-loop characteristic polynomial is formed as: \(\mathit{\Delta}\left(s,K\right)=s^2+\left(K+2\right)s+3K\).
The RL plot is sketched as follows:

1. The RL has two branches that commence at the OL poles at \(0,-2\) (for \(K=0\)); one branch terminates at the finite OL zero at \(s=-3\); the other branch follows the RL asymptote as \(K\to \infty\).
2. The real-axis locus lies in the intervals: \(\sigma \in \left[-2,0\right]\cup (-\infty ,-3]\) (for \(K>0\)).
3. The real-axis break points are defined by: \(\frac{1}{\sigma } +\frac{1}{\sigma +2} =\frac{1}{\sigma +3}\), or \({\sigma }^2+6\sigma +6=0\); the solutions include: \(\sigma =-1.27\) (break-away) and \(-4.73\) (break-in).
4. The asymptote angle is given as: \(\pm 180^{\circ }\) (for \(K>0\)), with intersection at: \(\sigma _{a} =1\).

The resulting RL plot (for \(K>0\)) is shown in Figure 5.1.4.

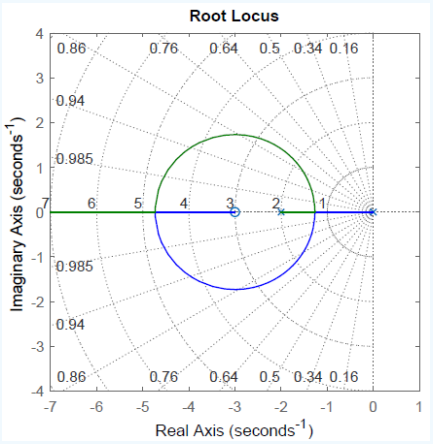


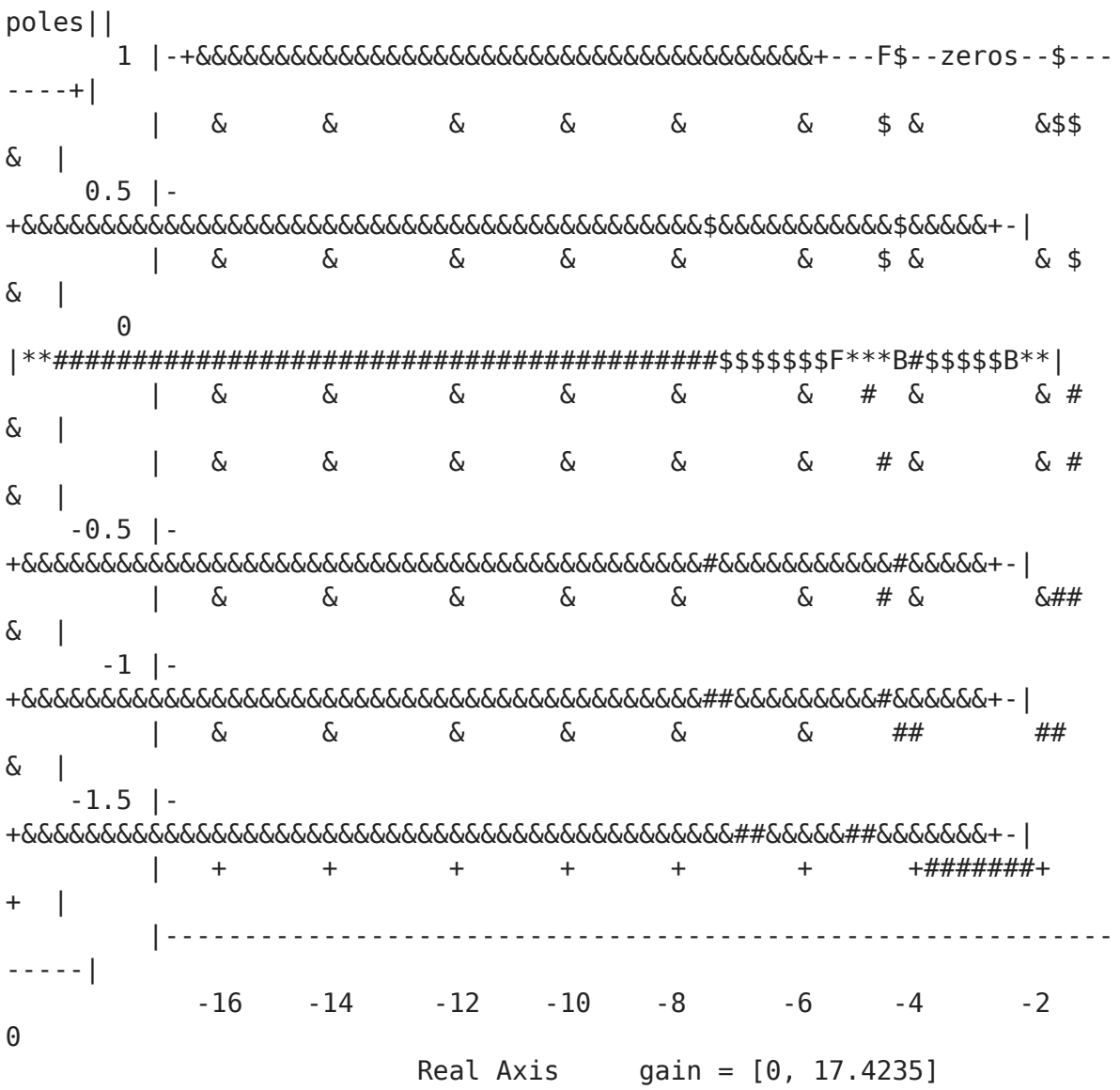
Figure \(\PageIndex{4}\): The root locus plot for \(G(s)=\frac{(s+3)}{s(s+2)}\).

```
pkg load control
s=tf('s');
Gs=(s+3)/s/(s+2);
rlocus(Gs)
```

run | restart | restart & run all

```
                                Root Locus of Gs

          |-------------------------------------------------------------
-----|
          |   +       +       +       +       +       ++------$$$$$$$-----
----+|
     1.5 |-+&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&|###?#$$locus$$
+||
          |   &       &       &       &       &       &|   B$$ open loop
```



Example \(\PageIndex{4}\)

Let \(KGH(s)=\frac{K}{s(s+1)(s+2)}\); then, \(n=3,\ m=0\). The closed-loop characteristic polynomial is formed as: \(\Delta (s)=s^{3} +3s^{2} +2s+K\).

The RL plot is sketched as follows:

1. The RL has three branches that commence at the OL poles at \(s=0,-1,-2\) (for \(K=0\)); the branches follow RL asymptotes as \(K\to \infty\).
2. The real-axis locus lies along \(\sigma \in \left[-1,0\right]\) (for \(K>0\)).
3. The break points are given as the solution to: \(\frac{1}{\sigma } +\frac{1}{\sigma +1} +\frac{1}{\sigma +2} =0\), which reduces to: \(3\sigma ^{2} +6\sigma +2=0\), and has solutions at \(\sigma =-0.38,\; -2.62\). The first solution is a break-away point for \(K>0\); the second one is the break-in point for \(K<0\).
4. The asymptote angles are given as: \(\pm 60^{\circ } ,180^{\circ }\) (for \(K>0\)); their common intersection point is at: \(\sigma _a =-\frac{3}{3} =-1\).

The resulting RL plot (for \(K>0\)) is shown in Figure 5.1.5.

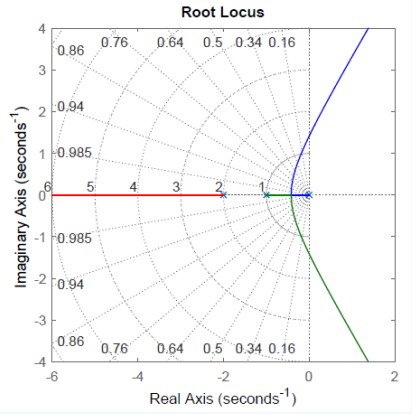


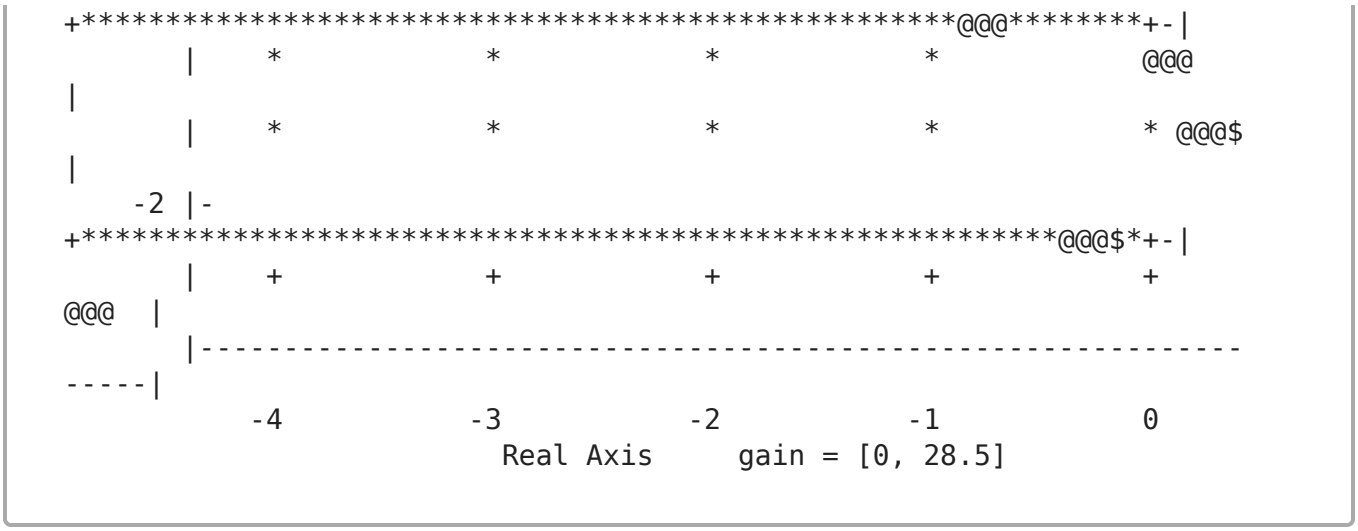
Figure \(\PageIndex{5}\): The root locus plot for \(G(s)=\frac{1}{s(s+1)(s+2)}\).

```
pkg load control
s=tf('s');
Gs=1/s/(s+1)/(s+2);
rlocus(Gs)
```

run | restart | restart & run all

```
                                  Root Locus of Gs

       |-------------------------------------------------------------
-----|
       |    +            +            +            +----------------
**&&&-+|
     2 |-+*****************************************|###?
###asymptote&&&  +||
       |    *            *            *           |%%%?%%%locus* &&&
||
       |    *            *            *           +---B---open loop
poles+|
     1 |-
+*************************************************&&&********+-|
       |    *            *            *           *   ***  &&& *
|
       |    *            *            *           * **    &&   *
|
     0
|##%%%%%%%%%%%%%%%%%%%%%%%%%%B############B@@@@@@@&&&&&B********+-|
       |    *            *            *           *   $$$ @    *
|
       |    *            *            *           *      $@@   *
|
       |    *            *            *           *        @@@ *
|
    -1 |-
```



Analytic RL Constraints

The analytic RL constraints are derived from the closed-loop characteristic polynomial, $\Delta (s,K)=1+KGH(s).$ that is satisfied by the closed-loop roots of the feedback control system.

Thus, for a point s_1 to lie on root locus, it must satisfy: $KGH\left(s_1\right)=-1$. Further, $KGH\left(s_1\right)$ is a complex number hence it must satisfy: $KGH\left(s_1\right)=1\angle 180{}^\circ$.

To proceed further, we assume that the loop transfer function $KGH(s)$, is written in the factored form as:

$$KGH(s)=\frac{K(s-z_{1})\ldots (s-z_{m})}{(s-p_{1})\ldots (s-p_{n})} ;\; n>m. \nonumber$$

Then, the corresponding magnitude and the angle conditions involve distances and angles from the OL poles and zeros to a point on the RL. The conditions are given as:

Definition: Magnitude Condition

For a point to lie on root locus plot, the transfer function magnitude at that point is bounded as:

$$\left|KGH(s_{1})\right|=\frac{K\left|s_{1} -z_{1} \right|\ldots \left|s_{1} -z_{m} \right|}{\left|s_{1} -p_{1} \right|\ldots |s_{1} -p_{n} |} =1 \nonumber$$

Definition: Angle Condition

For a point to lie on root locus plot, the transfer function angle at that point is bounded as:

$$\angle KGH(s_{1})\; =\; \angle (s_{1} -z_{1})+\ldots +\angle (s_{1} -z_{m})-\angle (s_{1} -p_{1})-\ldots -\angle (s_{1} -p_{n})=\pm 180^{\circ } \nonumber$$

The individual terms, $\left|s_1-p_i\right|$ and $\left|s_1-z_i\right|$, in the magnitude condition represent the Euclidean distances from the open-loop poles and zeros to the closed-loop pole location, $s=s_1$.

Similarly, the terms, $\angle(s_1-p_i)$ and $\angle(s_1-z_i)$ denote the angles ot the designated pole location.

Example \PageIndex{5}

Let $KGH(s)=\frac{K}{s(s+1)(s+2)}$; then, assuming that a point, $s_1=j\sqrt{2}$, lies on the root locus, it satisfies the following conditions (Figure 5.1.6):

Magnitude condition: $\frac{K}{|j\sqrt{2}|\,|1+j\sqrt{2}|\,|2+j\sqrt{2}|}=1$

Angle condition: $-\angle(j\sqrt{2})-\angle(1+j\sqrt{2})-\angle(2+j\sqrt{2})=-180^\circ$

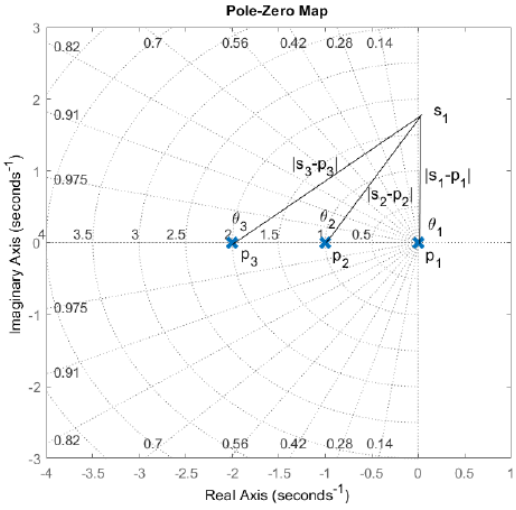


Figure \(\PageIndex{6}\): Displacements and angles from OL poles to a point on the RL.

Controller Design

The magnitude condition, \(\left|KGH(s_{1})\right|=1\), can be solved for the static controller gain, \(K\), to get: \(K=\frac{\left|s_{1} -p_{1} \right|\ldots |s_{1} -p_{n} |}{\left|s_{1} -z_{1} \right|\ldots \left|s_{1} -z_{m} \right|}\).

Further, let \(\theta _{z_{i} } ,\theta _{p_{i} }\) denote the angles from the open-loop zeros and poles to the point \(s_{1}\); then, the angle condition states that: \(\sum \theta _{z_{i} } -\sum \theta _{p_{i} } =\pm 180^{\circ }\) (for \(K>0\)). Alternatively, \(\sum \theta _{z_{i} } -\sum \theta _{p_{i} } =0^{\circ }\) (for \(K<0\)).

The angle condition can be used to add poles and zeros to the feedback loop, i.e., design a dynamic controller that would force an RL branch to pass through a desired closed-loop root location.

5.2: Static Controller Design

Static Controller Design

The RL plot displays achievable root locations for the closed-loop characteristic polynomial, $\Delta (s)=1+KGH(s)$, as the controller gain K varies from $0\to \infty$.

Assuming $G(s)=\frac{n(s)}{d(s)}$, the root locus plot displays root locations for the polynomial given as: $\Delta (s)=d(s)+Kn(s)$.

The static controller design involves the selection of the controller gain, K, that marks a desired closed-loop root location on the RL plot of the loop transfer function, $KGH(s)$.

Assuming that a RL branch passes through a desired closed-loop root location, s_{1}, the associated controller gain K can be obtained from the magnitude condition, or from the MATLAB generated RL plot by clicking on that location.

Example $\PageIndex{1}$

Let $KGH(s)=\frac{K}{s(s+1)(s+2)} \nonumber$; then, the closed-loop characteristic polynomial is formulated as:

$$\Delta (s)=s^{3} +3s^{2} +2s+K. \nonumber$$

Assume that we desire the closed-loop system step response to have: $\zeta =0.7$.

From the RL plot (Figure 5.2.1), we may choose, e.g., $K=0.66$ to satisfy this requirement. For $K=0.66,$ the closed-loop transfer function is given as:

$$T\left(s\right)=\frac{0.66}{\left(s+2.24\right)\left(s^2+0.76s+0.29\right)} \nonumber$$

The characteristic polynomial, $\mathit{\Delta}\left(s\right)=\left(s+2.24\right)\left(s^2+0.76s+0.29\right)$, has dominant closed-loop roots at: $-0.38\pm 0.38\; (\zeta =0.7)$.

The step response of the closed-loop system (Figure 5.2.2) shows a 5% overshoot and a settling time of 11.5 sec.

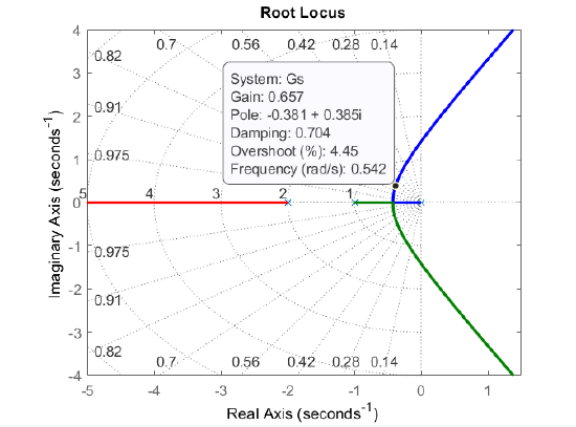


Figure $\PageIndex{1}$: Controller gain selection on RL plot.

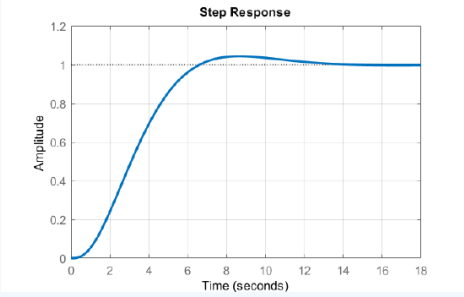
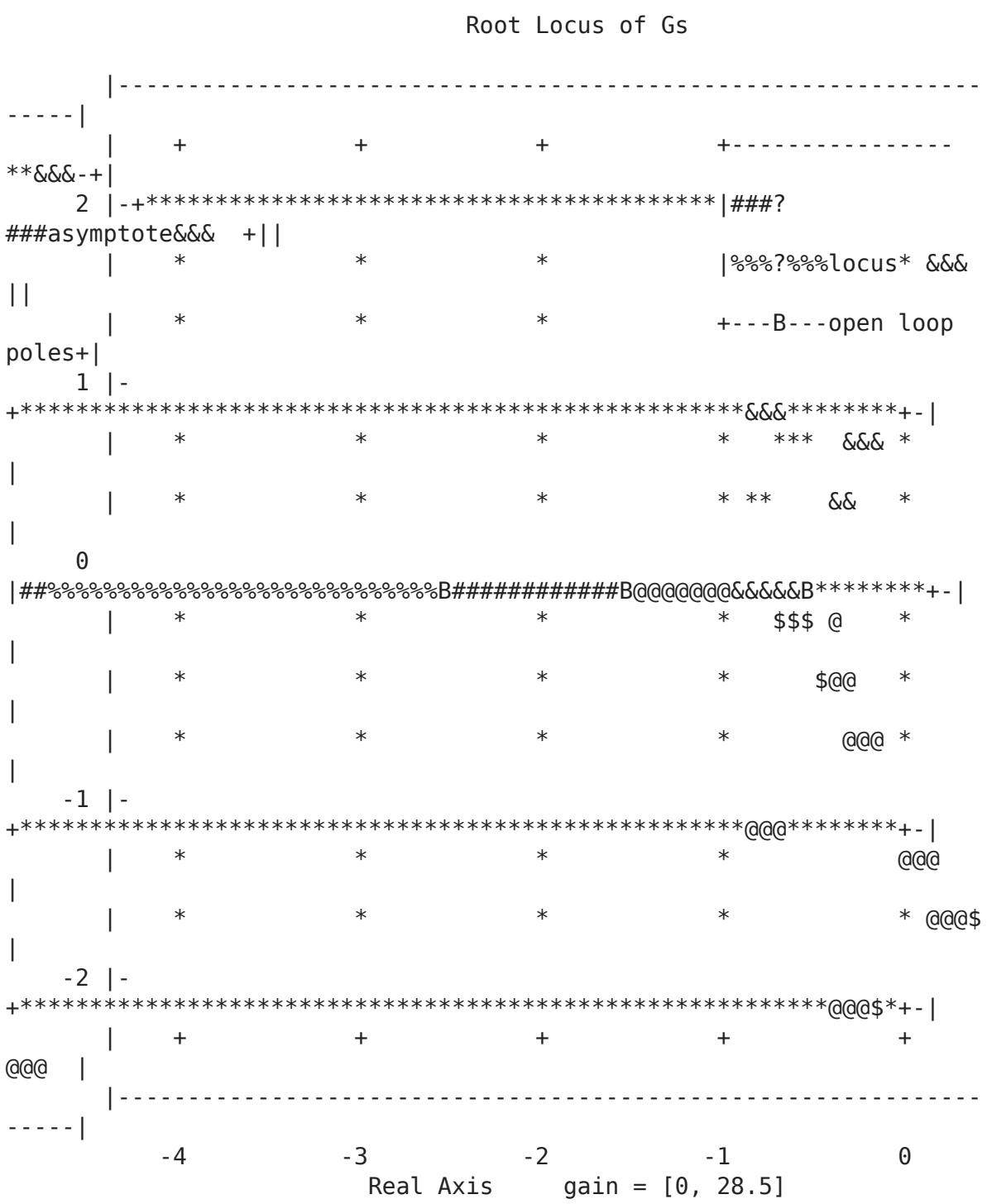


Figure $\PageIndex{2}$: Step response of the closed-loop system.

```
pkg load control
s=tf('s');
Gs=1/s/(s+1)/(s+2);
rlocus(Gs)
```

```
Ts=feedback(0.657*Gs,1);
figure, step(Ts)
```

run | restart | restart & run all



Stability Determination from Root Locus Plot

The range of \(K\) for closed-loop stability can be obtained from any RL branch intersection with the stability boundary, i.e., the \(j\omega\)-axis. The RL branches are likely to cross the stability boundary for \(n-m>2\).

In particular, for \(n-m=3\), the RL asymptotes are along: \({\theta }_a=\pm 60{}^\circ ,\ 180{}^\circ\). Hence, for high enough controller gains, the RL branches will cross the stability boundary leading to instability. The corresponding value of controller gain can be obtained by clicking on the RL plot.

Example \(\PageIndex{2}\)

Let \(KGH(s)=\frac{K}{s(s+1)(s+2)} \nonumber\); then, the closed-loop characteristic polynomial is:

\[\Delta (s)=s^{3} +3s^{2} +2s+K. \nonumber \]

The stability conditions for the third-order polynomial are: \(K>0,\, \, 6-K=0\). Thus, the range of \(K\) for stability is: \(0<K<6\).

Indeed for \(K=6,\) the closed-loop characteristic polynomial, \(\mathit{\Delta}\left(s\right)=s^3+3s^2+2s+6\), has roots at: \(s=-3,\; \; \pm j\sqrt{2}\).

For \(K=6,\) the closed-loop transfer function is given as:

\[T\left(s\right)=\frac{6}{(s+3)(s^2+2)} \nonumber \]

The closed-loop system impulse response shows oscillates at the natural frequency of\(\sqrt{2}\) rad/sec (Figure 5.2.3):

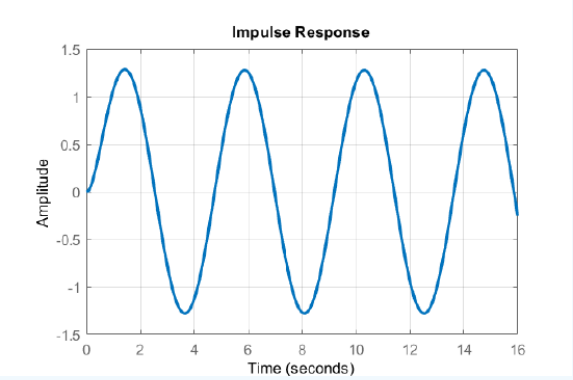
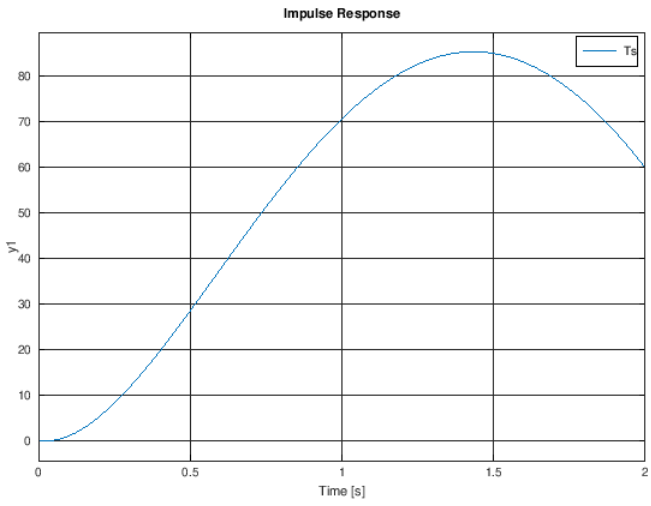


Figure \(\PageIndex{3}\): Impulse response for imaginary closed-loop roots (\(K=6\)).

```
pkg load control
s=tf('s');
Gs=1/s/(s+1)/(s+2);
Ts=feedback(6*Gs,1);
impulse(Ts)
```

run | restart | restart & run all



The static controller possesses only limited ability to a shape the response of the closed-loop system. In particular, the choice of closed-loop roots is restricted to those locations on the available root locus plot.
In the event that the existing root locus does not offer favorable pole locations, we may explore the possibility of adding a dynamic controller to the feedback loop that would modify the existing RL and cause it to pass through a desired location in the complex plane.

5.3: Transient Response Improvement

Root Locus Improvements

A dynamic controller alters the root locus plot by adding poles and zeros to the loop transfer function. In general, addition of a finite zero to the loop transfer function causes the RL branches to bend towards it, whereas, the presence of a closed-loop pole repels the RL branch close to it.

In particular, we explore the effect of adding a first-order dynamic controller, defined by: $K(s)=\frac{K(s+z_c)}{s+p_c}$ to the feedback loop. A first-order controller adds a zero, $-z_c$, and a pole, $-p_c$, to the loop transfer function.

Addition of a first-order phase-lead controller to the feedback loop improves the transient response of the system; addition of a phase-lag or PI controller improves the steady-state tracking response.

Phase-Lead Controller Design

To bring about transient response improvements, a phase-lead or a PD controller may be added to the feedback loop. A phase-lead controller is described by:

$$K\left(s\right)=\frac{K\left(s+z_c\right)}{s+p_c};\ z_c<p_c$$

In the limiting case of $p_c\rightarrow\infty$, the phase-lead controller changes into a PD controller, described by:

$$K\left(s\right)=K\left(s+z_c\right)$$

The PD controller has unlimited gain at high frequencies, which would amplify high-frequency noise entering the system. Thus, for effective noise suppression, a phase-lead design with $p_c\gg z_c$ is preferred to a PD controller.

The locations of the controller pole and zero in the case of phase-lead and PD controllers are selected with the help of RL angle condition. The general design procedure is given as follow:

1. Select a desired closed-loop pole location, s_1 from design specifications.
2. Compute $G\left(s_1\right)$; the MATLAB ‘evalfr’ command can be used for this purpose.
3. Use the angle condition to determine the angle contribution required of the controller:

$$\theta_z-\theta_p=180^\circ-\angle G\left(s_1\right)$$

4. Select controller pole and zero locations to meet the angle contribution from the controller; since the angle condition defines a single constraint among two variables, multiple designs are possible.
5. Use the MATLAB ‘evalfr’ command to verify controller angle contribution. Note that we may only need a 'ballpark' design.
6. Plot the root locus for the compensated system and choose gain K to complete controller design.

Example 5.3.1: Phase-Lead Design

Let $G(s)=\frac{2}{s(s+2)}$; design a phase-lead controller to achieve the following specifications: $\zeta\cong0.7,\ \ t_s\leq2\ s$.

The phase-lead controller design proceeds as follows:

1. Let $s_1=-2.5\pm j2.5$ describe a desired closed-loop root location.
2. By using the MATLAB ‘efalfr’ command to evaluate the plant transfer function, we get: $G(s_1)=0.222\angle123.7^\circ$; hence, the required controller phase contribution to realize the desired pole location is: $K\left(s_1\right)=K\angle56.3^\circ$.
3. Let $z_c=2,\ p_c=5$; then, $K\left(s_1\right)=0.72\angle55^\circ$.
4. Further, $K(s_1)G(s_1)\cong0.16\angle180^\circ$; hence, $K=\frac{1}{0.16}\cong6.3$. Alternatively, from the root locus plot of the compensated system, the controller gain is found as: $K=6.29$; hence, $K\left(s\right)=6.29\left(\frac{s+2}{s+5}\right)$.

The root locus plot for the phase-lead design is shown below (Fig. 5.3.1).

The closed-loop system is formed as:

$$T(s)=\frac{12.58(s+2)}{(s+2)(s^2+5s+12.58)}$$

We note that the closed-loop transfer function $T(s)$ includes a pole-zero cancelation at $s=-2$. Such cancelations are permitted as long as they are restricted to the open left-half plane (OLHP). Moreover, component inaccuracies make exact pole–zero cancelation an unlikely event in practice.

The step response of the closed-loop system is plotted in Fig. 5.3.2. From the figure, the settling time is given as: $t_s=1.7s$.

We also note that alternate phase-lead designs for this example are possible, however, choosing controller zero location $z_c<2$ is likely to increase the settling time of the closed-loop system.

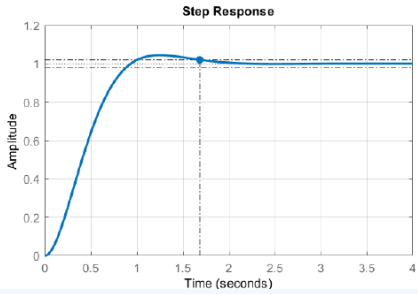


Figure 5.3.2: Step response of closed-loop system.

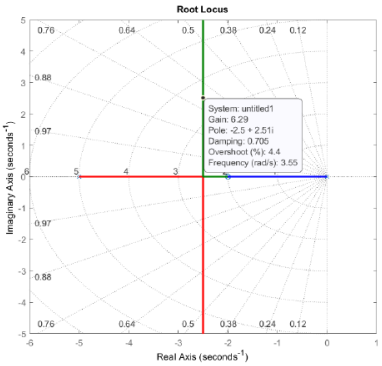


Figure 5.3.1: Root locus for the phase-lead design.

✔ Example 5.3.2: PD Controller Design

Let $G(s)=\frac{2}{s(s+2)}$; design a PD controller to achieve the following specifications are: $\zeta\cong 0.7,\quad t_s\le 2\,s$.

The PD controller design proceeds as follows:

1. Let $s_1=-2.5\pm j2.5$ describe a desired closed-loop root location.
2. By using the MATLAB 'efalfr' command to evaluate the plant transfer function, we get: $G(s_1)=0.222\angle 123.7^\circ$; hence, the required controller phase contribution to realize the desired pole location is: $K(s_1)=K\angle 56.3^\circ$.
3. From trigonometry (Fig. 5.3.3), let $z_c=4.17$; then, for $K(s)=s+z_c$, we have: $K(s_1)\cong 3\angle 56.3^\circ$.
4. Since $K(s_1)G(s_1)\cong 0.66\angle 180^\circ$, we have $K=\frac{1}{0.66}=1.5$. Alternatively, from the RL plot of the compensated system (Fig. 5.3.4), we obtain: $K=1.5$; hence, $K(s)=1.5(s+4.17)$.

For noise suppression considerations, it is preferable to replace the PD controller by a phase-lead design.

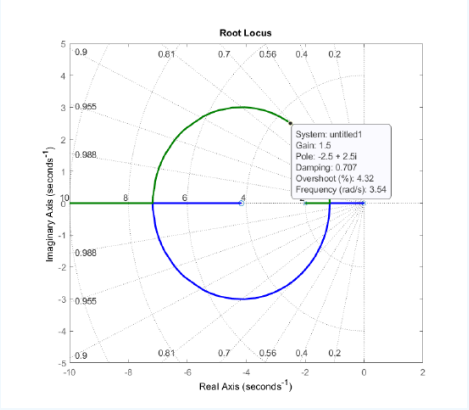


Figure 5.3.4: Root locus plot for the PD controller.

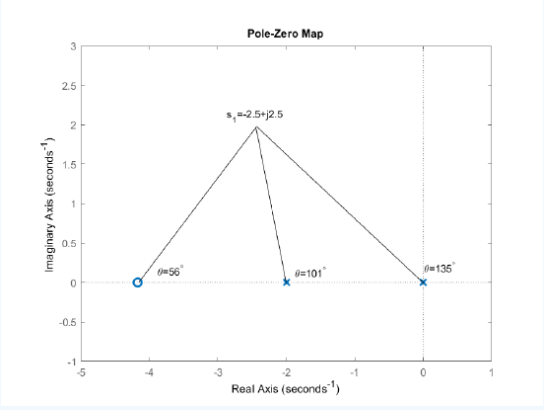


Figure 5.3.3: Trigonometry to compute the required angle from controller zero.

Accordingly, let $K(s)=K\frac{s+4}{s+55}$; since $K(s_1)G(s_1)\cong 0.012\angle 180^\circ$, let $K=81.25$. Then, for $K(s)=81.25\frac{s+4}{s+55}$, we have $K(s_1)G(s_1)=-1$.

The step response of the PD controller and proposed phase-lead design are almost identical (Fig. 5.3.5).

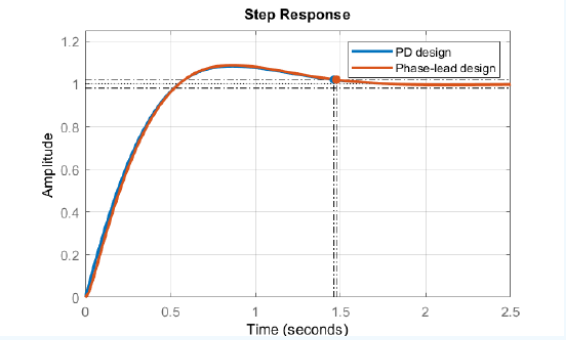


Figure 5.3.5: Step response of the closed-loop systems.

A second-example of phase-lead design is presented below.

Example 5.3.3: Phase-Lead Design

Let $G(s)=\frac{10}{s(s+2)(s+5)}$; assume that the design specifications are given as: $OS \leq 10\%,\ \ t_s \leq 2sec$.

The phase-lead controller design proceeds as follows:

1. Let $s_1=-2.2 \pm j2.4$ describe a desired closed-loop root location.
2. Using MATLAB 'efalfr' command, we get: $G(s_1)=0.35\angle 92^\circ$; the required controller phase contribution is: $K(s_1)=K\angle 88^\circ$.
3. Let $z_c=2,\ p_c=22$; then $K(s_1)=0.11\angle 88^\circ$.
4. From the gain consideration, let $|K(s_1)G(s_1)|=1$; then, by using 'evalfr' command, we obtain $K=24$, hence the desired phase-lead controller is: $K(s)=\frac{24(s+2)}{s+22}$. The compensated system has, $K(s_1)G(s_1) \cong -1$.

The resulting closed-loop system after pole-zero cancelation is given as: $T(s)=\frac{240}{(s+22.6)(s^2+4.4s+10.62)}$.

The step response of the closed-loop system shows a settling time of $1.8s$.

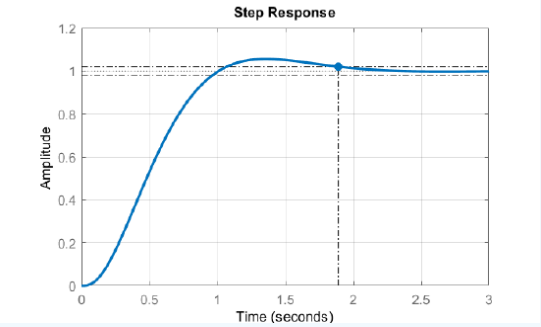


Figure 5.3.6: Step response of the closed-loop system.

Rate Feedback Controller Design

The rate feedback configuration, shown in Figure 5.3.7, involves an inner loop with rate feedback in addition to regular output feedback. In this configuration, the rate feedback gain is k_f, while $K(s)$ denotes the outer loop controller.

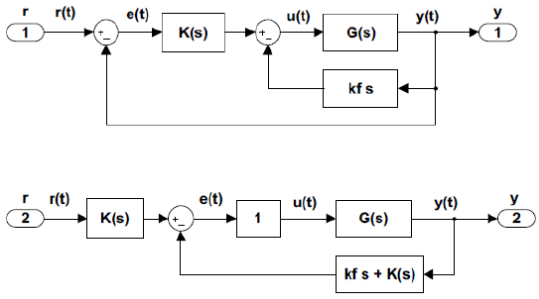


Figure 5.3.7: Rate feedback configuration (top); equivalent diagram (bottom).

The rate feedback design is typically performed in two stages. First, the inner loop is designed for suitable closed-loop root locations. The resulting inner loop transfer function serves as a plant for the outer loop controller design.

Minor Loop Design. The loop transfer function for the minor loop is given as: $G(s)k_f s$. Let $G(s) = \frac{n(s)}{d(s)}$; then, the closed-minor-loop transfer function is: $G_{ml}(s) = \frac{n(s)}{d(s)+n(s)k_f s}$.

Outer Loop Design. The loop transfer function for the outer loop is given as: $K(s)G_{ml}(s)$.

Let $K(s) = K$; then, the closed-outer-loop transfer function is obtained as: $T(s) = \frac{Kn(s)}{d(s)+(k_f s+K)n(s)}$.

Example 5.3.4

Let $G(s) = \frac{2}{s(s+2)}$; assume that the design specifications are given as: $OS \leq 10\%,\ t_s \leq 2s\ (\zeta \geq 0.6)$.

Minor Loop Design. The minor loop gain is: $\frac{2k_f s}{s(s+2)}$; the closed-minor-loop transfer function is: $G_{ml}(s) = \frac{2k_f}{s(s+2+2k_f)}$.

We may select, e.g., $k_f = 1$ to obtain: $G_{ml}(s) = \frac{2}{s(s+4)}$.

Alternatively, from the minor loop root locus (Figure 5.3.8), we select $k_f = 1$ for closed-loop root at $s = -4$.

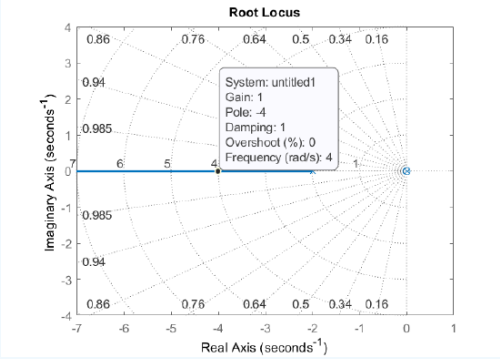


Figure 5.3.8: Minor loop root locus.

The minor loop root locus shows an apparent pole-zero cancelation at the origin. The pole at the origin is, however, retained in the outer loop design.

Outer Loop Design. The outer loop transfer function is given as: $KG_{ml}(s) = \frac{2K}{s(s+4)}$.

From the closed-loop characteristic polynomial, $\Delta(s) = s(s+4)+K$, we may choose, e.g., $K = 4$, for closed-loop roots to be located at: $s = -2 \pm j2$.

Alternatively, we may choose $K = 4$ from the outer loop root locus plot (Fig. 5.3.9).

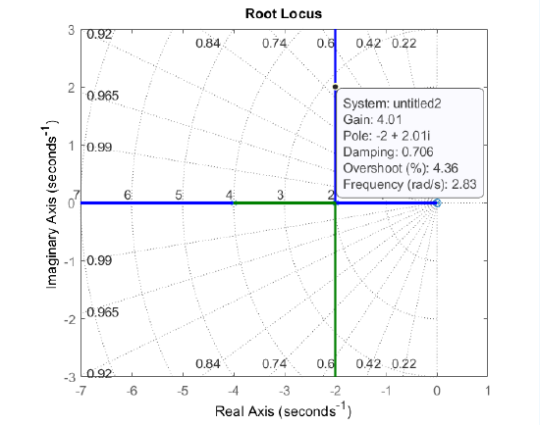


Figure 5.3.9: Outer loop root locus.

The closed-outer-loop transfer function is given as: $T(s) = \frac{8}{s^2+4s+8}$.

The step response of the closed-loop system shows a settling time of $2.1s$.

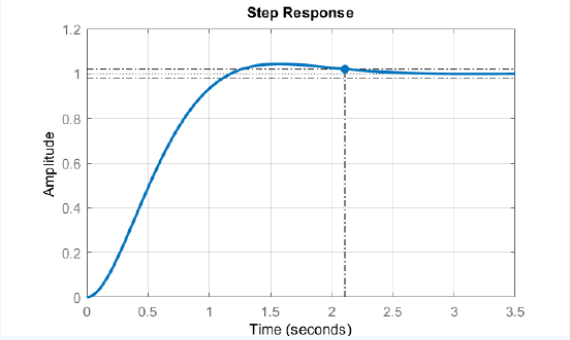


Figure 5.3.10: Step response of the closed-loop system.

Example 5.3.5

Let $G(s) = \frac{10}{s(s+2)(s+5)}$; assume that the step response specifications are: $OS \leq 10\%,\ t_s \leq 2sec\ (\zeta \geq 0.6)$.

Minor Loop Design. The minor loop gain is $k_f s\ G(s) = \frac{10k_f s}{s(s+2)(s+5)}$. From the minor loop root locus (Figure 5.3.11), we may choose, e.g., $k_f = 2$ for the closed-minor-loop roots to be located at: $s = -3.5 \pm j4.2\ (\zeta \cong 0.64)$.

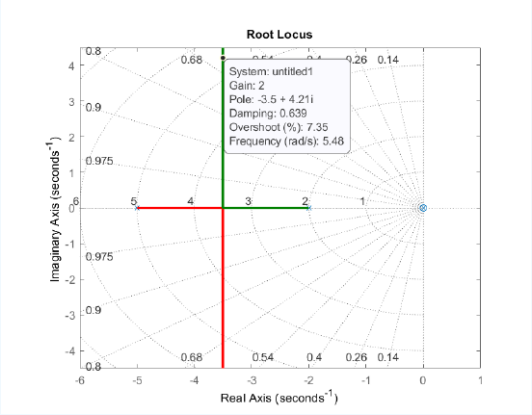


Figure 5.3.11: Minor loop root locus.

Outer Loop Design. The outer loop gain is obtained as: $KG_{ml}(s) = \frac{10K}{s[(s+2)(s+5)+k_f]} = \frac{10K}{s(s^2+7s+30)}$.

From the outer loop root locus, we may choose, e.g., $K = 5.4$ for closed-loop roots located at: $s = -2 \pm j3.75$ $(\zeta \cong 0.47)$. Although ζ appears to be low, the step response of the closed-loop system has a small overshoot due to the presence of a real pole at $s = -3$.

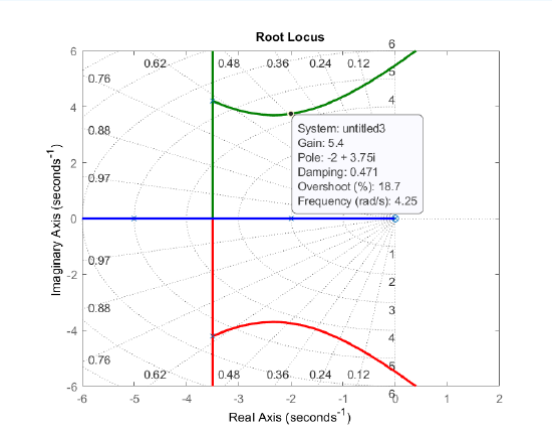


Figure 5.3.12: Outer loop root locus.

The closed-outer-loop transfer function is given as: $T(s) = \frac{54}{(s+3)(s^2+4s+18)}$.

The step response of the closed-loop system shows a settling time of $1.5s$.

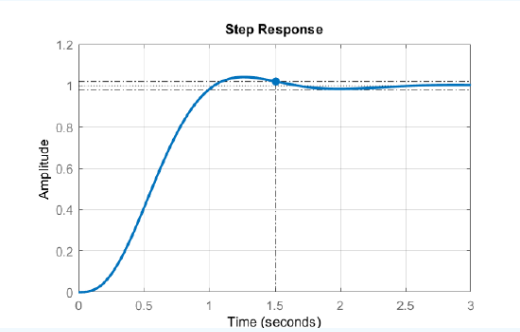


Figure 5.3.13: Step response of the closed-loop system.

5.4: Steady-State Error Improvement

Steady-State Error Improvement

The steady-state tracking error to a step or ramp input is evaluated using the position and velocity error constants of the system. These constants are defined as:

$$K_p = \lim_{s \to 0} KGH(s)$$

$$K_v = \lim_{s \to 0} sKGH(s)$$

In the case of unity-gain feedback configuration ($H(s) = 1$), the steady-state tracking error to step and ramp inputs is computed as:

$$e(\infty)|_{\text{step}} = \frac{1}{1 + K_p}$$

$$e(\infty)|_{\text{ramp}} = \frac{1}{K_v}$$

where the MATLAB 'dcgain' command can be used to compute the relevant error constant.

The addition of a first-order PI or phase-lag controller to the feedback loop improves the relevant error constant. Both controllers add a pole–zero pair to the loop transfer function, and are described by:

$$K(s) = \frac{K(s + z_c)}{(s + p_c)}, \quad p_c < z_c$$

where $p_c = 0$ denotes the PI controller.

Assuming $K = 1$, the position error constant is raised as: $K_p^{'} = (z_c / p_c) K_p > K_p$.

The PI controller, in particular, results in: $K_p^{'} = \infty$, hence $e(\infty)|_{\text{step}} = 0$. By virtue of adding an integrator to the feedback loop, the steady state error to a step input is zero for the PI controller.

Phase-Lag and PI Design

The PI/phase-lag controller design aims to boost the relevant error constant while not appreciably disturbing the existing RL plot.

Let s_1 denote a desired closed-loop pole location on the existing RL plot; then, the controller pole and zero are selected in accordance with: $p_c < z_c \ll Re(s_1)$. This ensures that the adverse angle contribution from the controller stays small, i.e., $\theta_z - \theta_p \approx 0^\circ$.

The actual pole-zero locations for phase-lag and zero location for PI can be arbitrarily selected.

The PI/phase-lag controller adds a pole-zero pair close to the origin to the loop transfer function. Hence a closed-loop pole with a large time constant appears in the closed-loop transfer function. Nevertheless, the contribution of the added slow mode to the overall system response remains small due to the close proximity of the pole to the controller zero.

Example 5.4.1: Phase-Lag Design

Let $G(s) = \frac{10}{s(s+2)(s+5)}$; assume that the design specifications are given as: $\zeta \geq 0.6, \;\; e_{ss}|_{\text{ramp}} \leq 0.1$.

Since transient response improvement is not aimed, we may use a static controller to satisfy the damping ratio requirement.

As an example, let $K = 0.7$; then, the closed-loop system has dominant roots at: $s = -0.8 \pm j0.8 \; (\zeta = 0.7)$. The characteristic polynomial is factored as: $\Delta(s) = (s + 5.38)(s^2 + 1.62s + 1.3)$.

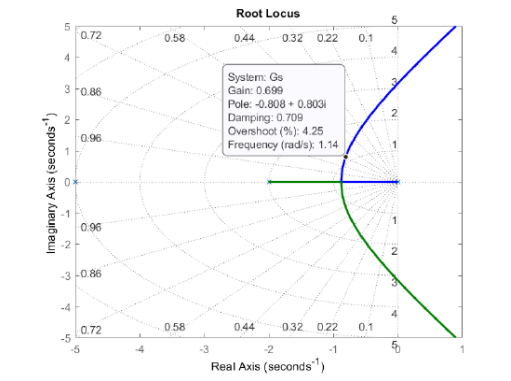


Figure 5.4.1: Root locus plot for phase-lag design.

Next, the velocity error constant is evaluated as: $K_v = 0.7$.

In order to raise $K_v > 10$, we consider a phase-lag controller: $K(s) = \frac{s+0.02}{s+0.0012}$, where $\angle K(s_1) = -0.3^\circ$.

The compensated system has: $K_v = 11.67$; hence $e_{ss}|_{\text{ramp}} \leq 0.1$.

The ramp response of the closed-loop system is plotted to confirm the results.

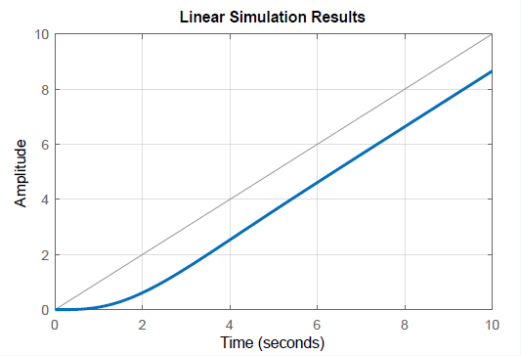


Figure 5.4.2: Unit-ramp response of the closed-loop system.

With the addition of the phase-lag controller, the closed-loop transfer function is given as:

$$T(s) = \frac{7(s+0.02)}{(s+0.0202)(s+5.38)(s^2+1.61s+1.29)}$$

The closed-loop root at $s = -0.0202$ is almost canceled by the presence of the zero at $s = -0.02$. Hence, the slow mode $e^{-0.0202t}$ has a very small coefficient and only minimally affects the system response. The dominant closed-loop roots are also minimally affected.

Example 5.4.2: PI Design

The approximate model of a DC motor is given as: $G(s) = \frac{.5}{(.01s+1)(.1s+1)}$. The design specifications are given as: $\zeta \geq 0.6$, $e_{ss}|_{\text{step}} \leq 0.01$.

We choose a PI controller, $K(s) = \frac{K(s+z_c)}{s}$, with unity-gain feedback $H(s) = 1$. The loop gain is given as: $K(s)G(s) = \frac{.5K(s+z_c)}{s(.01s+1)(.1s+1)}$.

The position error constant is $K_p = \infty$. Hence $e_{ss}|_{\text{step}} = 0$.

The controller zero can be selected to cancel one of the plant poles; hence $z_c = 10$. Next, from the RL plot, we may choose, e.g., $K = 10$. The PI controller design is given as: $K(s) = \frac{10(s+10)}{s}$.

The closed-loop system transfer function is given as: $T(s) = \frac{5000}{s^2+100s+5000}$. The step response of the closed-loop system is plotted below.

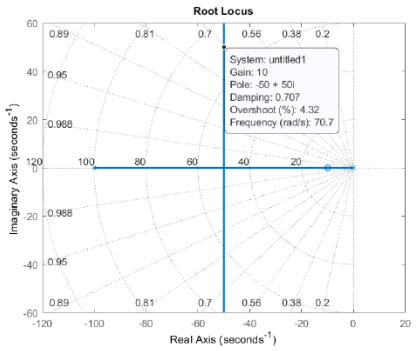


Figure 5.4.3: Root locus plot for the PI controller.

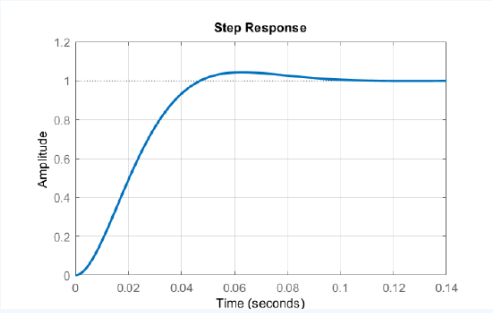


Figure 5.4.4: Step response of the closed-loop system.

5.5: Transient and Steady-State Improvement

Lead–Lag and PID Designs

When control system design specifications require simultaneous improvement to the transient response and the steady-state tracking error, a lead-lag or a PID controller may be considered.

A lead-lag controller combines phase-lead and phase-lag stages; as a rule, the the transient response is improved first, followed by steady-state error improvement. The lead-lag controller is expressed as:

$$K(s) = K\left(\frac{s+z_{c1}}{s+p_{c1}}\right)\left(\frac{s+z_{c2}}{s+p_{c2}}\right), \quad z_{c1} < p_{c1},\ z_{c2} > p_{c2}$$

a PID controller similarly combines PD and PI controllers. It is expressed as:

$$K(s) = K\left(s+z_{c1}\right)\left(\frac{s+z_{c2}}{s}\right)$$

As a rule, the transient response improvement is aimed first. The pole and zero selected for phase-lead or PD design steer the root locus toward the desired closed-loop pole locations. The pole-zero pair in the phase-lag or PI part of the design is located close to the origin in order to minimally disturb the existing root locus.

Lead-Lag Design

Example 5.5.1: Lead-Lag Design

Let $G(s) = \frac{10}{s(s+2)(s+5)}$; assume that the design specifications are: $\%OS \leq 10\%$, $t_s \leq 2\text{s}$, $e_{ss}|_{\text{ramp}} \leq 0.1$. These specifications translate into: $\zeta \geq 0.6,\ \sigma \geq 2,\ K_v \geq 10$.

Phase-Lead Design. The phase-lead controller design (covered in Example 5.3.3) proceeds as follows: Let $s_1 = -2.2 \pm j2.4,\ \zeta = 0.68,$ denote a desired closed-loop pole location; then, $G(s_1) = 0.35\angle 92^\circ$, i.e., the required controller phase contribution is: $K(s_1) = K\angle 88^\circ$.

Let $z_c = 2,\ p_c = 22$; then $K(s_1) = 0.11\angle 88^\circ$. From the RL plot, a controller gain $K = 24$ is selected for closed-loop roots at: $s = -2.2 \pm j2.4$. Hence, the phase-lead section is designed as: $K_{\text{lead}}(s) = 24\left(\frac{s+2}{s+22}\right)$.

Phase-lag Design. For $K_{\text{lead}}(s)G(s)$, the velocity error constant is given as: $K_v = 2.2$. To boost the error constant to $K_v > 10$, a phase-lag controller, $K_{\text{lag}} = \frac{s+0.01}{s+0.002}$ is considered.

The angle contribution of the phase-lag controller is: $\angle K_{\text{lag}}(s_1) = -0.1^\circ$ hence the dominant closed-loop roots are negligibly affected.

The lead-lag controller is formed as: $K(s) = 25\left(\frac{s+2}{s+24}\right)\left(\frac{s+0.01}{s+0.002}\right)$.

The closed-loop transfer function is obtained as: $T(s) = \frac{240(s+0.01)}{(s+0.01004)(s+22.6)(s^2+4.39s+10.58)}$. The dominant closed-loop roots are located at: $s = -2.2 \pm j2.4\ (\zeta = 0.68)$.

The step and ramp responses of the closed-loop system are plotted below. The step response displays a settling time of $1.9sec.$

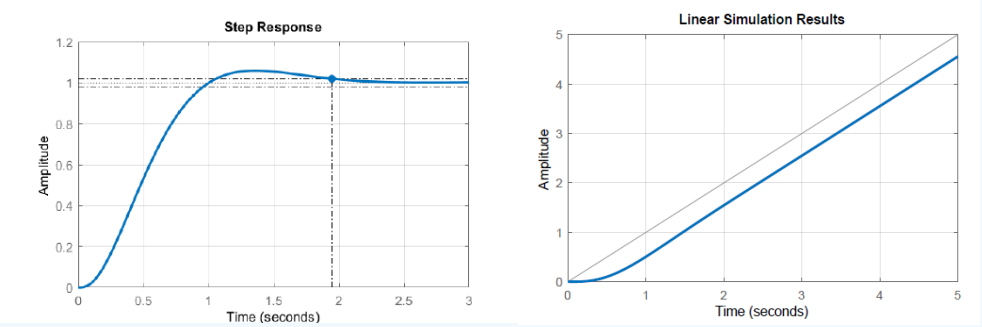


Figure 5.5.1 : Step response (left) and ramp response (right) of the closed-loop system.

PID Design

An alternate PID design is performed in the next example.

Example 5.5.2: PID Design

Let $G(s)=\frac{10}{s(s+2)(s+5)}$; assume that the design specifications are: $\%OS\leq 10\%,\ t_s\leq 2\text{s},\ e_{ss}|_{\text{ramp}}\leq 0.1$. These specifications translate into: $\zeta\geq 0.6,\ \sigma\geq 2,\ K_v\geq 10$.

PD design. The PD controller is given as: $K_{PD}(s)=K(s+z_c)$. We may arbitrarily choose the controller zero at $z_c=-2$, and use the root locus of the compensated system to select $K=1$ with closed-loop roots at: $s_1=-2.5\pm j1.95$.

PI design. The PI controller is given as: $\frac{s+z_c}{s}$, where $z_c\ll s_1$ is desired. We may arbitrarily select a zero location: $z_c=0.01$ to define the PID controller as: $K_{PID}(s)=\frac{(s+0.01)(s+2)}{s}$.

The closed-loop transfer function is given as: $T(s)=\frac{10(s+0.01)}{(s+0.01005)(s+2)(s^2+4.99s+9.95)}$.

The step response of the closed-loop system shows a settling time of $t_s=1.8sec$.

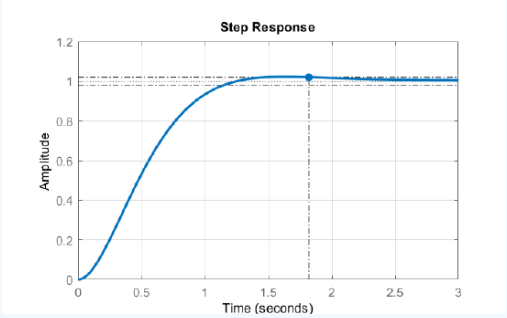


Figure 5.5.3: Step response of the closed-loop system.

Phase-Lead with PI

We can also combine the phase-lead controller design with a PI controller to define the composite controller as shown below.

Example 5.5.3: Phase-Lead with PI

Let $G(s)=\frac{10}{s(s+2)(s+5)}$; assume that the design specifications are: $\%OS\leq 10\%,\ t_s\leq 2\text{s},\ e_{ss}|_{\text{ramp}}\leq 0.1$. These specifications translate into: $\zeta\geq 0.6,\ \sigma\geq 2,\ K_v\geq 10$.

Phase-Lead Design. Let $s_1=-2.2\pm j2.4\ \ \zeta=0.68$; then, we may choose, as in Example 5.5.1 above, $z_c=2,\ p_c=22$ and $K=24$ for closed-loop roots at: $s=-2.2\pm j2.4$. The phase-lead controller is defined as: $K_{\text{lead}}(s)=24\left(\frac{s+2}{s+22}\right)$.

PI design. The PI controller is given as: $\frac{s+z_c}{s}$. In this case, $z_c=0.01$ does not appear to be a good choice, as the slow mode in the step response remains dominant, increasing the settling time.

A better strategy is to select the PI controller zero to cancel the second plant pole at $s=-5$. The composite controller is formed as: $K(s)=24\left(\frac{s+2}{s+22}\right)\left(\frac{s+5}{s}\right)$.

The resulting closed-loop transfer function after pole-zero cancelations is given as: $T(s)=\frac{240}{(s^2+22s+240)}$, which has dominant closed-loop roots located at: $s=-11\pm j11\ (\zeta=0.7)$.

The step response of the closed-loop system shows a short settling time of $t_s\cong 0.38sec$.

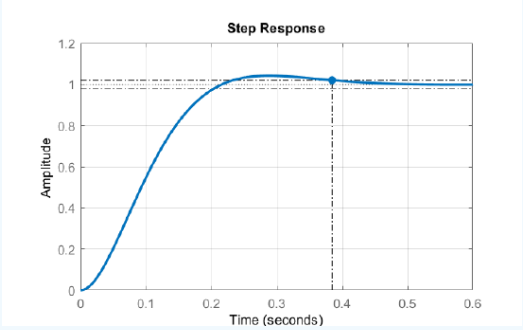


Figure 5.5.4: Step response of the closed-loop system.

MATLAB Tuning of PID Controller

The MATLAB Control Systems Toolbox offers the 'pidtune' command to design an optimal PID controller or a PID controller with a filter to reduce high frequency noise.

Example 5.5.4: MATLAB PID Controller

Let $G(s)=\frac{10}{s(s+2)(s+5)}$; then, the MATLAB tuned PID controller is obtained as:

$$K(s)=\frac{2.01(s+0.314)(s+1.43)}{s}$$

The resulting closed-loop roots are located at: $s=-0.34,\ -1.16,\ -2.73\pm 3.71.$

Alternatively, MATLAB tuned PID controller with filter is obtained as:

$$K(s)=\frac{727.25(s+0.302)(s+1.41)}{s(s+361.4)}$$

The resulting closed-loop roots are located at: $s=-0.34,\ -1.16,\ -2.73\pm 3.71,\ -361.5.$

The closed-loop system responses for the MATLAB designed PID controllers are plotted below. The responses are almost identical with an overshoot of about 10% with a settling time of $t_s\cong 6sec$.

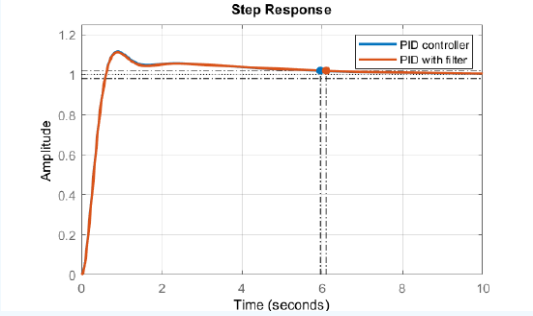


Figure 5.5.5: Step response of MATLAB tuned PID controllers.

5.6: Controller Realization

Frequency Selective Filters

The dynamic controllers, including phase-lead, phase-lag, lead-lag, PD, PI, PID, represent frequency selective filters that may be realized by electronic circuits built with operational amplifiers and resistor-capacitor networks.

A resister-capacitor (RC) circuit connected in series or parallel has the following impedance:

Series RC circuit: $Z_{\text{ser}}(s) = R + \frac{1}{Cs} = \frac{RCs+1}{Cs}$

Parallel RC circuit: $Z_{\text{par}}(s) = \frac{R/Cs}{R+\frac{1}{Cs}} = \frac{R}{RCs+1}$

An operational amplifier (Op-Amp) in the inverting configuration has an input–output transfer function: $\frac{V_0(s)}{V_{\text{i}}(\text{s})} = -\frac{Z_{\text{f}}(\text{s})}{Z_{\text{i}}(\text{s})}$, where $Z_{\text{i}}(\text{s})$ and $Z_{\text{f}}(\text{s})$ denote the input and feedback path impedance.

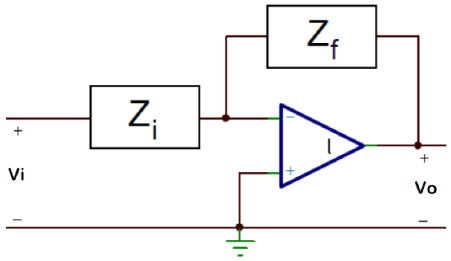


Figure 5.6.1: Operational amplifier in the inverting configuration

Phase-Lead/Phase-Lag Controllers

A first-order phase-lead or phase-lag controller can be realized with parallel RC circuits placed as input and feedback paths. The resulting controller transfer function is given as:

$$K(s) = -\frac{Z_{\text{f}}(\text{s})}{Z_{\text{i}}(\text{s})} = -\frac{R_f}{R_i}\frac{(R_{\text{i}}\text{C}_{\text{i}}\text{s}+1)}{(R_{\text{f}}\text{C}_{\text{f}}\text{s}+1)}.$$

The controller transfer function has a zero located at: $z_{\text{c}} = \frac{1}{\text{R}_{\text{i}}\text{C}_{\text{i}}}$, and a pole at: $p_{\text{c}} = \frac{1}{\text{R}_{\text{f}}\text{C}_{\text{f}}}$.

Hence we may choose $R_{\text{i}}C_{\text{i}} > R_{\text{f}}C_{\text{f}}$ for the phase-lead and $R_{\text{i}}C_{\text{i}} < R_{\text{f}}C_{\text{f}}$ for the phase-lag design.

For static gain and sign correction, a resistive Op-Amp circuit can be employed. The circuit has a gain of: $\frac{V_{\text{o}}}{V_{\text{i}}} = -\frac{R_{\text{f}}}{R_{\text{i}}}$.

Example 5.6.1

Let $K(s) = \frac{5(s+1)}{s+10} = \frac{0.5(s+1)}{0.1s+1}$; then, the transfer function realization involves the following constraints: $R_{\text{i}}C_{\text{i}} = 1,\ R_{\text{f}}C_{\text{f}} = 0.1,\ R_{\text{f}}/R_{\text{i}} = 0.5.$

We may choose, for example, $R_{\text{i}} = 100\text{K}\Omega$. Then, $R_{\text{f}} = 50\text{K}\,\Omega,\ \text{C}_{\text{i}} = 10\mu\,\text{F},\ \text{C}_{\text{f}} = 2\mu\,\text{F}.$

PD, PI, PID Controllers

These controllers can be realized by combining the following impedance:

$$G_{\text{PD}}(s) = -\frac{R_{\text{f}}}{Z_{\text{par}}(s)} = -\frac{R_{\text{f}}}{R_{\text{i}}}(R_{\text{i}}\text{C}_{\text{i}}\text{s}+1)$$

$$G_{\text{PI}}(s) = -\frac{1/C_{\text{f}}\text{s}}{Z_{\text{par}}(s)} = -\frac{1}{R_{\text{i}}\text{C}_{\text{f}}}\frac{(R_{\text{i}}\text{C}_{\text{i}}\text{s}+1)}{s}$$

$$G_{\mathrm{PID}}(s) = -\frac{Z_{\mathrm{f-ser}}(s)}{Z_{\mathrm{i-par}}(s)} = -\frac{1}{R_{\mathrm{i}}\mathrm{C_f}}\frac{(R_{\mathrm{i}}\mathrm{C_i s}+1)(\mathrm{R_f C_f s}+1)}{s}.$$

For the PID controller, the controller gains are solved as functions of component values:

$$\mathrm{k_p} = \frac{\mathrm{R_f}}{\mathrm{R_i}} + \frac{\mathrm{C_i}}{\mathrm{C_f}}, \quad \mathrm{k_i} = \mathrm{R_f C_i}, \quad \mathrm{k_d} = \frac{1}{\mathrm{R_i C_f}}$$

Example 5.6.2

Let $G_{\mathrm{PID}}(s) = \frac{(s+0.1)(s+10)}{s}$; then, to realize the transfer function with RC networks, we may choose, for example:

$$R_{\mathrm{i}} = 100\ \mathrm{K\Omega},\ R_{\mathrm{f}} = 1\mathrm{M\Omega},\ C_{\mathrm{i}} = 1\ \mu\mathrm{F},\ C_{\mathrm{f}} = 10\ \mu\mathrm{F}$$

CHAPTER OVERVIEW

6: Compensator Design with Frequency Response Methods

Learning Objectives

1. Characterize and plot the frequency response of transfer function models.
2. State the performance criteria for controller design in the frequency domain.
3. Design compensators for dynamic system models using frequency response methods.
4. Characterize the closed-loop frequency response of the compensated system.

6.0: Prelude to Compensator Design with Frequency Response Methods

Frequency response methods for designing compensators for feedback control systems predate the root locus and the time-domain (state variable) methods. They have been used effectively for designing amplifiers and filters in the case of electrical networks, and vibration analysis in the case of mechanical systems.

Control systems design using the frequency response method requires knowledge of $KGH(j\omega)$. Strictly speaking, the knowledge of the plant transfer function $G(s)$ is not needed. In the absence of a mathematical model, the plant transfer function can be identified from empirical measurements of the frequency response, $G(j\omega)$.

The frequency response of a system can be graphed in multiple ways. The two most common representations are: [GrindEQ__1_] the Bode plot and [GrindEQ__2_] the Nyquist or Polar plot. The closed-loop frequency response may be visualized on the Nichol's chart.

The frequency response design seeks to impart a certain degree of relative stability, measured by gain and phase margins, to the feedback control loop. The closed-loop system stability is alternatively ascertained by using the celebrated Nyquist criterion.

The peak gain in the closed-loop frequency response is a measure of the relative stability; the higher the peak the lower the relative stability. The crossover frequency on Bode magnitude plot and bandwidth on the closed-loop frequency response define measures of speed of response in the time-domain.

We consider standard feedback control system configuration (Figure 6.1) that includes a plant $G(s)$, a sensor $H(s)$, and a controller $K(s)$. Unless stated otherwise, unity-gain feedback ($H(s) = 1$) is assumed.

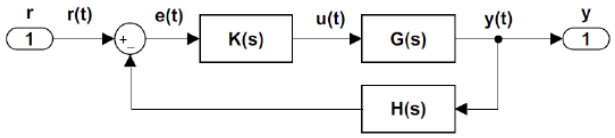


Figure 6.0.1: Copy and Paste Caption here. (Copyright; author via source)

In the following, we first review the plotting of frequency response and the associated performance metrics. Later, we will discuss the frequency response modification through adding phase-lead, phase-lag, lead–lag, PD, PI, and PID compensators to the control loop.

6.1: Frequency Response Plots

Frequency Response Function

The frequency response of the loop transfer function, $KGH(s)$, is represented as: $KGH(j\omega) = KGH(s)|_{s=j\omega}$.

For a particular value of ω, the $KGH(j\omega)$ is a complex number, which is described in terms of its magnitude and phase as $KGH(j\omega) = |KGH(j\omega)|e^{j\phi(\omega)}$.

As ω varies from 0 to ∞, $KGH(j\omega)$ can be plotted in the complex plane (the polar plot). Alternatively, both magnitude and phase can be plotted as functions of ω (the Bode magnitude and phase plots).

Bode Plot

To proceed further, we assume that the loop transfer function is expressed using first and second order factors as:

$$KGH(s) = \frac{K\prod_{i=1}^{m}\left(1+\frac{s}{z_i}\right)}{s^{n_0}\prod_{i=1}^{n_1}\left(1+\frac{s}{p_i}\right)\prod_{i=1}^{n_2}\left(1+2\zeta_i\frac{s}{\omega_{n,i}}+\frac{s^2}{\omega_{n,i}^2}\right)}$$

where m is the number of real zeros, n_0 is the number of poles at the origin, n_1 is the number of real poles, n_2 is the number of complex pole pairs, and K is a scalar gain. Further, z_i and p_i represent the zero and pole frequencies for first-order factors; $\omega_{n,i}$ and ζ_i represent the natural frequency and damping ratio of second-order factors.

The corresponding frequency response function is given as:

$$KGH(j\omega) = \frac{K\prod_{i=1}^{m}\left(1+j\frac{\omega}{z_i}\right)}{(j\omega)^{n_0}\prod_{i=1}^{n_1}\left(1+j\frac{\omega}{p_i}\right)\prod_{i=1}^{n_2}\left(1-\frac{\omega^2}{\omega_{n,i}^2}+j2\zeta_i\frac{\omega}{\omega_{n,i}}\right)}$$

It is customary to plot the magnitude of the frequency response function on the log scale as $|G(j\omega)|_{\text{dB}} = 20\ \log_{10}|G(j\omega)|$. The magnitude of the loop gain is given in dB as:

$$\begin{aligned}|KGH(j\omega)|_{\text{dB}} = 20\ \log K + \textstyle\sum_{i=1}^{m} 20\ \log\left|1+\frac{j\omega}{z_i}\right| - (20n_0)\log\omega \\ - \textstyle\sum_{i=1}^{n_1} 20\ \log\left|1+\frac{j\omega}{p_i}\right| - \sum_{i=1}^{n_2} 20\ \log\left|1-\frac{\omega^2}{\omega_{n,i}^2}+j2\zeta_i\frac{\omega}{\omega_{n,i}}\right|.\end{aligned}$$

At low frequencies, the frequency response magnitude is a constant, i.e., $\lim_{\omega\to 0}|KGH(j\omega)|_{\text{dB}} = 20\ \log K$.

For large ω, the magnitude plot is characterized by a slope: $-20(n-m)$ dB/decade of ω, where $n-m$ represents the pole excess of the loop transfer function.

The phase angle $\phi(\omega)$ of the loop transfer function is computed as:

$$\phi(\omega) = \sum_{i=1}^{m}\angle\left(1+\frac{j\omega}{z_i}\right) - n_0(90^\circ) - \sum_{i=1}^{n_1}\angle\left(1+\frac{j\omega}{p_i}\right) - \sum_{i=1}^{n_1}\angle\left(1-\frac{\omega^2}{\omega_{n,i}^2}+j2\zeta_i\frac{\omega}{\omega_{n,i}}\right).$$

The phase angle for large ω is given as: $\phi(\omega) = -90^\circ(n-m)$.

In the MATLAB Control Systems Toolbox, the Bode plot is obtained by using the 'bode' command, which is invoked after defining the transfer function object.

Nyquist Plot

The frequency response function $KGH(j\omega)$ represents a complex rational function of ω. The function can be plotted in the complex plane. A polar plot describes the graph of $KGH(j\omega)$ ω varies from $0 \to \infty$.

The Nyquist plot is a closed curve that describes a graph of $KGH(j\omega)$ for $\omega \in (-\infty, \infty)$. In the MATLAB Control Systems Toolbox the Nyquist plot is obtained by invoking the ‘nyquist’ command, invoked after defining the transfer function.

The shape of the Nyquist plot depends on the poles and zeros of $KGH(j\omega)$; it can be pictured by estimating magnitude and phase of $KGH(j\omega)$ at low and high frequencies. For example, if $KGH(s)$ has no poles at the origin, then at low frequency, $KGH(j0) \cong K\angle 0^\circ$; while, at high frequency, $|KGH(j\infty)| \to 0$, and$\angle KGH(j\infty) = -90^\circ(n-m)$, where $n-m$ represents the pole excess of $KGH(s)$.

In particular, for $n-m=3$, the Nyquist plot crosses the negative real-axis at the phase crossover frequency, ω_{pc}. Let $G(j\omega_{pc}) = g\angle 180^\circ$; then, the gain margin is given as g^{-1}. Further, for $K = g^{-1}$, the Nyquist plot of $KG(j\omega)$ passes through the $-1+j0$, described as the critical point for stability determination.

Definition: Nyquist Stability Criterion

The number of unstable closed-loop poles of $\Delta(s) = 1+KGH(s)$ equals the number of unstable open-loop poles of $KGH(s)$ plus the number of clock-wise (CW) encirclements of the $-1+j0$ point on the complex plane by the Nyquist plot of $KGH(s)$.

Examples

Example 6.1.1

Let $G(s) = \frac{1}{s+1}$; then, $G(j\omega) = \frac{1}{1+j\omega} = \frac{1}{\sqrt{1+\omega^2}} \angle -\tan^{-1}\omega$.

In particular, at specific points, $G(j0) = 1\ \angle 0^\circ$, $G(j1) = \frac{1}{\sqrt{2}}\angle -45^\circ$, and $G(j\infty) = 0\ \angle -90^\circ$.

The Bode magnitude plot (Figure 6.1.1) starts at $0\ dB$ with an initial slope of zero that gradually changes to $-20\ dB$ per decade at high frequencies. An asymptotic Bode plot consists of two lines joining ar the corner frequency (1 rad/s).

The Bode phase plot varies from 0° to -90° with a phase of -45° at the corner frequency.

The Nyquist plot of $G(s)$ is circle in the right-half plane (RHP). Further, for $K>0$, the Nyquist plot of $KG(j\omega)$ is confined to the RHP; hence, by Nyquist stability criterion, the closed-loop system is stable for all $K>0$.

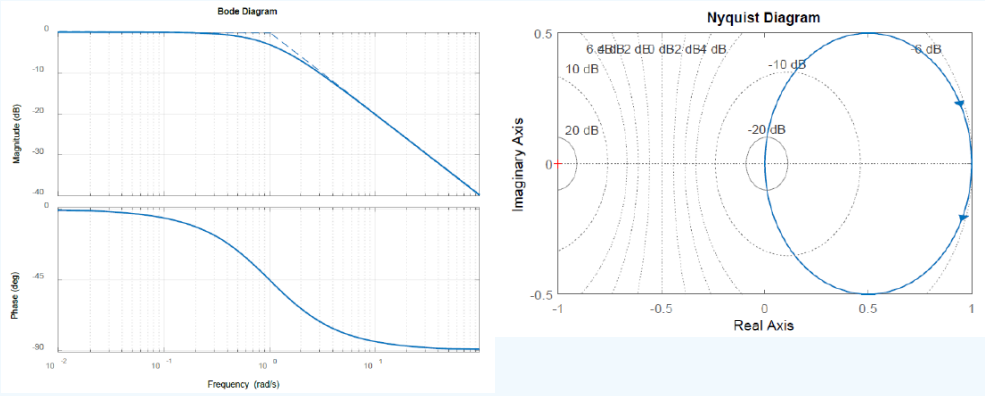


Figure 6.1.1 : Bode and Nyquist plots for $G(s) = \frac{1}{s+1}$.

Example 6.1.2

Let $G(s) = \frac{1}{s(s+1)}$; then, $G(j\omega) = \frac{1}{j\omega(1+j\omega)} = \frac{1}{\omega}\frac{1}{\sqrt{1+\omega^2}}\angle -90^\circ - \tan^{-1}\omega$.

In particular, $G(j0) = \infty\angle -90^\circ$, $G(j1) = \frac{1}{\sqrt{2}}(-3dB)\angle -135^\circ$, $G(j\infty) = 0\angle -180^\circ$.

The Bode magnitude plot (Figure 6.1.2) has an initial slope of $-20\ dB$ per decade that gradually changes to $-40\ dB$ per decade at high frequencies. The phase plot shows a variation from -90° to -180° with a phase of -135° at the corner frequency.

The Nyquist plot is a closed curve that travels along negative $j\omega$-axis for $\omega \in (0,\infty)$, along positive $j\omega$-axis for $\omega \in (-\infty, 0)$, and scribes a semi-circle of a large radius for $\omega \in (0^-, 0^+)$. As the Nyquist plot of $KG(j\omega)$ stays away from the critical point $(-1+j0)$ for positive values of K, the closed-loop system is projected to be stable for all $K>0$.

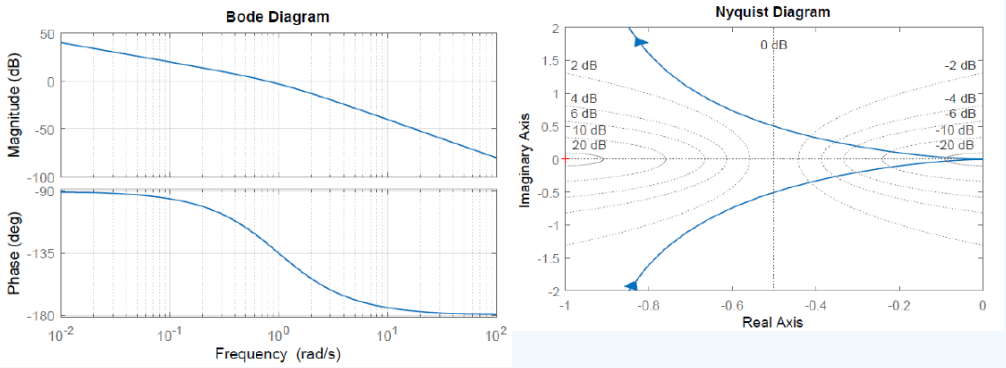


Figure 6.1.2: Bode and Nyquist plots for $G(s) = \frac{1}{s(s+1)}$.

Example 6.1.3

Let $G(s) = \frac{1}{s^2+s+1}$; then, $G(j\omega) = \frac{1}{1-\omega^2+j\omega} = \frac{1}{\sqrt{(1-\omega^2)^2+(\omega)^2}} \angle -tan^{-1}\frac{\omega}{1-\omega^2}$.

In particular, $G(j0) = 1\,(0dB)\angle 0^\circ$, $G(j1) = 1\,(0dB)\angle -90^\circ$, $G(j\infty) = 0\angle -180^\circ$.

The Bode magnitude plot (Figure 6.1.3) starts at $0\ dB$ with an initial slope of 0; the slope changes to $-40\ dB$ per decade for $\omega > 1$. The magnitude plot shows a resonance peak at the corner frequency due to the low value of $\zeta = 0.5$.

The Nyquist plot of $G(j\omega)$ is a closed curve that has no crossing with the negative real-axis. As the Nyquist plot of $KG(j\omega)$, $K > 0$ stays away from the critical point $(-1 + j0)$, the closed-loop system is projected to be stable for all $K > 0$.

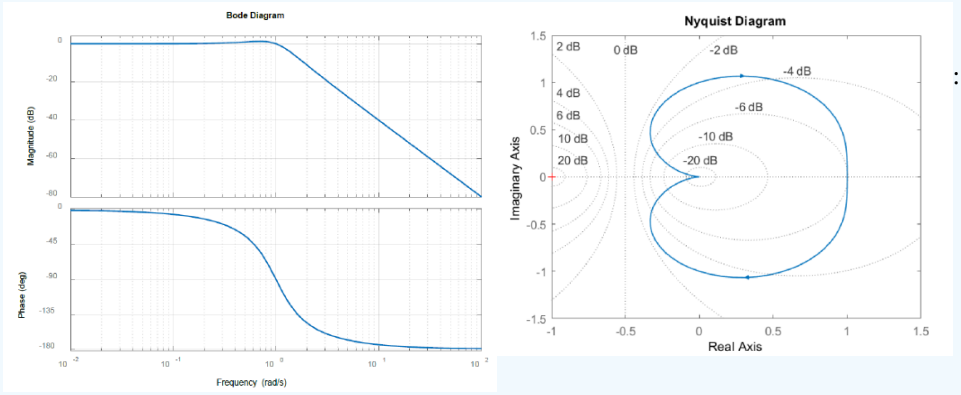


Figure 6.1.3 : Bode and Nyquist plots for $G(s) = \frac{1}{s^2+s+1}$.

Example 6.1.4

Let $G(s) = \frac{2}{s(s+1)(s+2)}$; then, $G(j\omega) = \frac{2}{j\omega(1+j\omega)(2+j\omega)} = \frac{1}{\omega}\frac{1}{\sqrt{1+\omega^2}}\frac{2}{\sqrt{2+\omega^2}} \angle -90^\circ - \tan^{-1}\omega - \tan^{-1}2\omega.$.

In particular, $G(j0) = \infty\angle -90^\circ$, $G(j1) = \frac{2}{\sqrt{10}}(-4dB)\angle -108^\circ$, $G(j1) = \frac{1}{5\sqrt{2}}(-13dB)\angle -139^\circ$, and $G(j\infty) = 0dB\angle -270^\circ$.

The Bode magnitude plot (Figure 6.1.4) has an initial slope of $-20\ dB$ per decade that changes first to $-40\ dB$ per decade and then to $-60\ dB$/decade at high frequencies. The phase plot shows a variation from -90° to -270°.

The polar plot begins with a large magnitude along the negative $j\omega$-axis (for $\omega = 0^+$), crosses the negative real-axis at $0.33\angle 180^\circ$(for $\omega = 3.32$), and approaches the origin from the positive $j\omega$-axis (for $\omega \to \infty$).

The Nyquist plot, obtained by joining a reflection of the polar plot, includes a closed contour that includes a negative real-axis crossing at $0.33\angle 180^\circ$. As the plot expands for $K > 1$, it encloses the critical $(-1 + j0)$ point. The closed-loop system is projected to be stable for $K < 3$.

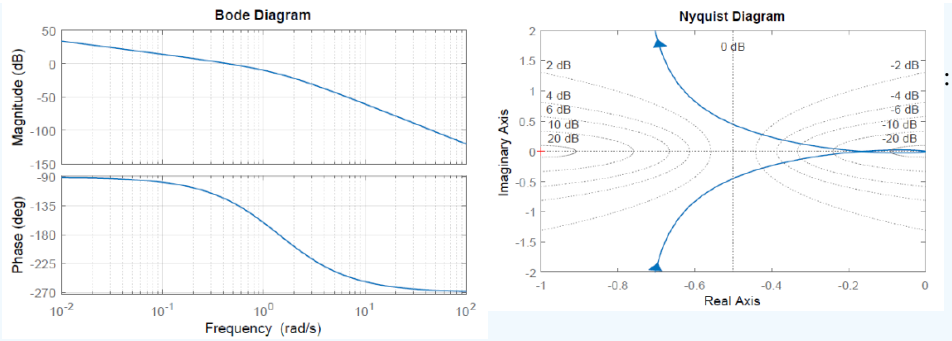


Figure 6.1.4 : Bode and Nyquist plots for $G(s) = \frac{2}{s(s+1)(s+2)}$.

Further, the Nyquist plot for $K = 3$ passes through the critical point.

The Nyquist plot for $K = 4$ crosses the real axis to the left of the critical point as shown in the figure below. In addition, the plot of $KGH(j\omega)$ for $s \in (0^-, 0^+)$ subscribes a large circle in the complex plane (not shown in the figure). Hence the Nyquist plot completes two CW encirclements of the critical point, indicating two unstable closed-loop roots.

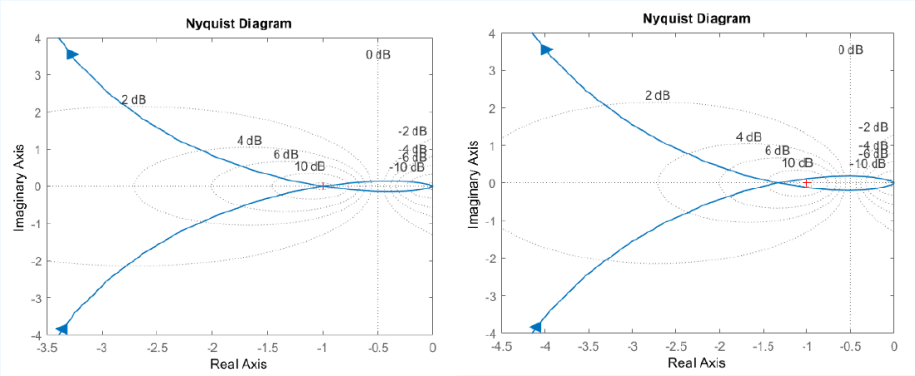


Figure 6.1.5 : Nyquist plots for $K = 3$ and $K = 4$.

6.2: Measures of Performance

In the frequency response design methods, the measures of performance include relative stability, described in terms of gain and phase margins, error constants, sensitivity and complementary sensitivity functions, etc. These are described below.

It is assumed that the loop transfer function $KGH(j\omega)$ is minimum-phase, i.e., its poles and zeros are located in the closed left half-plane (CLHP) including the $j\omega$-axis.

Relative Stability

In frequency domain design, the relative stability of the feedback loop is described in terms of the gain and the phase margins.

Definition: Gian Margin

The gain margin (GM) is the maximum amount of loop gain that can be added to the feedback loop (by the controller) without compromising stability.

On the Bode magnitude plot, the GM is indicated as: $GM = -|KGH(j\omega_{pc})|_{dB}$, where $\omega_{\text{pc}}$is the phase crossover frequency, i.e., the frequency at which $\phi(\omega_{\text{pc}}) = -180^\circ$.

In the MATLAB Control Systems Toolbox, the 'margin' command is used to obtain the GM and PM as well as the gain and phase crossover frequencies on the Bode plot. The command is invoked after defining the loop transfer function using 'tf' or 'zpk' command.

On the Nyquist plot, ω_{pc} is marked by the negative real-axis crossing of $KGH(j\omega)$, i.e., let $G(\omega_{pc}) = g\angle -180^\circ$; then, $GM = g^{-1}$.

Definition: Phase Margin

The phase margin (PM) is the maximum amount of phase that can be added to the feedback loop (by the controller) without compromising stability.

On the Bode phase plot, the PM is indicated as: $\text{PM} = 180^\circ + \phi(\omega_{\text{gc}})$, where ω_{gc} is the gain crossover frequency, i.e., the frequency defined by $|G(j\omega_{\text{gc}})| = 1$.

On the Nyquist plot, the PM is indicated when the plot enters the unit circle, i.e., assume that $KGH(j\omega) = 1\angle -\phi$, then the phase margin is given as: $\text{PM} = 180^\circ - \phi$.

A loop transfer function with pole excess ($n - m \geq 3$) has a finite gain margin, i.e., $0 < \text{GM} < \infty$. The loop transfer function with ($n - m = 2$) may have a finite GM.

Definition: Return Difference

The return difference is described by one minus the loop gain. At the input to the plant, the return difference is 1+KGH(j\omega). The minimum value of the return difference is $\min_\omega |1 + KGH(j\omega)|$.

For minimum-phase plants, the closed-loop stability depends on the Nyquist plot keeping a finite distance from the $-1 + j0$ point. The closest distance of $KGH(j\omega)$ from the $-1 + j0$ point indicates the minimum return difference with respect to $G(s)$. The minimum return difference is a measure of the robustness of the control loop to parameter variations in $G(s)$.

Examples

Example 6.2.1

Let $KG(s) = \frac{1}{s(s+1)}$; then, from the bode plot, we have $\text{GM} = \infty$ and $\text{PM} = 51.8^\circ$.

From Fig. 6.1.1, the minimum return difference is $\min_\omega |1 + KG(j\omega)| = 1$.

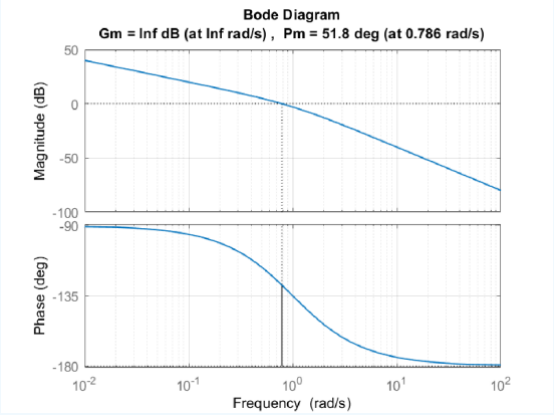


Figure 6.2.1: Relative stability from the Bode plot: $G(s) = \frac{1}{s(s+1)}$.

Example 6.2.2

Let $KG(s) = \frac{1}{s^2+s+1}$; then, we have $\text{GM} = \infty\, dB$ and $\text{PM} = 90^\circ$.

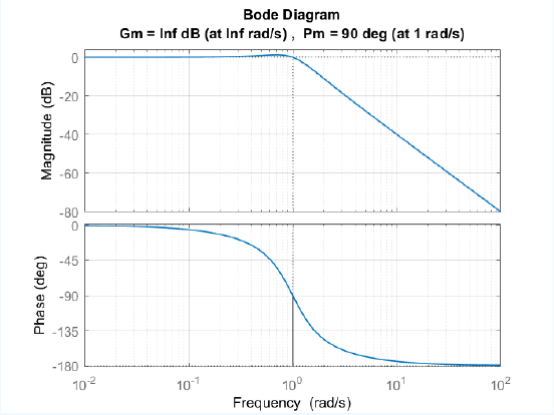


Figure 6.2.2: Relative stability from the Bode plot: $G(s) = \frac{1}{s^2+s+1}$.

Example 6.2.1

Let $KG(s) = \frac{2}{s(s+1)(s+2)}$; then, we have $\text{GM} = 15.6\, dB$ and $\text{PM} = 53.4^\circ$.

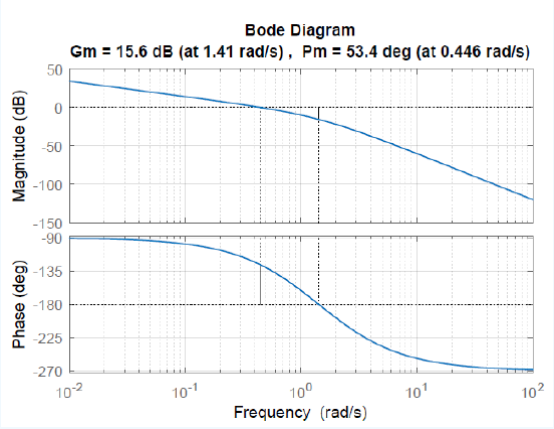


Figure 6.2.3: Relative stability from the Bode plot: $G(s) = \frac{2}{s(s+1)(s+2)}$.

Closed-Loop System Response

Damping ratio. The damping ratio of the dominant poles in the closed-loop transfer function is related to the phase margin.

In particular, for a prototype second-order transfer function: $KGH(j\omega) = \frac{\omega_n^2}{j\omega\,(j\omega+2\zeta\omega_n)}$, the gain crossover occurs at: $\omega_{gc} = k\omega_n$, where $k = \sqrt{\sqrt{4\zeta^4+1} - 2\zeta^2}$, whereas $\phi_m = \tan^{-1}\left(\frac{2\zeta}{k}\right)$.

In the range of $0.5 < \zeta < 0.9$, the approximate relation is given as: $\phi_m \cong tan^{-1}\left(\frac{6\zeta}{3.3-2\zeta}\right)$, $0.5 < \zeta < 0.9$. As an example, in order to have a $\zeta = 0.7$, we may design the system for a $PM \cong 65.6^\circ$.

Settling time. The settling time of the closed-loop system step response is related to the phase margin. For a second order transfer function, the settling time t_s and the phase margin ϕ_{m} are related as: $\tan\phi_m = \frac{2\zeta\omega_n}{\omega_{\mathrm{gc}}} = \frac{8}{t_s\omega_{\mathrm{gc}}}$.

Bandwidth. The closed-loop bandwidth is bounded as: $\omega_n \le \omega_B \le 2\omega_n$.

In particular, for systems with low damping, $\omega_B \cong \omega_n$.

The bandwidth is related to the rise time t_r of the closed-loop system step response as: $\omega_{\mathrm{B}} t_r \approx 1$ or $t_r \cong 1/\omega_{\mathrm{B}}$.

In particular, in the case of second-order systems ($0.6 < \zeta < 0.9$), the rise time can be approximated as: $t_r \cong \frac{3\zeta}{w_n}$.

Example 6.2.4

We consider the model of a small DC motor, given as: $G(s) = \frac{500}{s^2+110s+1025}$.

Assume that a PI compensator for the model is defined as: $K(s) = \frac{K(s+10)}{s}$. Then, for $K = 10$, we have closed-loop roots located at: $s = --50 \pm j50.4$.

The Bode plot of the loop gain with compensator in the loop displays a phase margin of $\phi_{\mathrm{m}} = 65.8^\circ$, which corresponds to a closed-loop damping ratio of $\zeta = 0.7$.

The step response of the compensated system displays a rise time of $t_r = 0.028s$ and a settling time of $t_s = 0.077s$. The predicted value of the settling time is: $t_{\mathrm{s}} = \frac{8}{\omega_{\mathrm{gc}}\tan\phi_{\mathrm{m}}} = 0.078\mathrm{s}$.

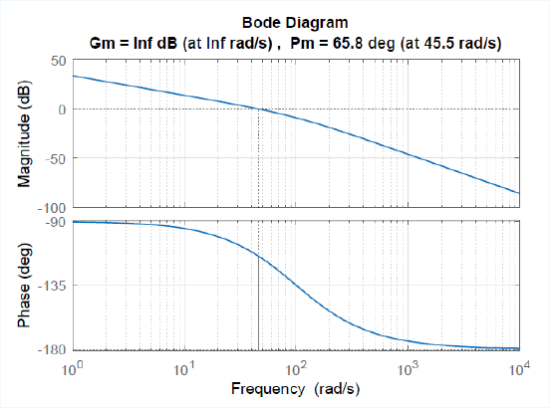


Figure 6.2.4: The Bode plot of the DC motor model with a PI controller.

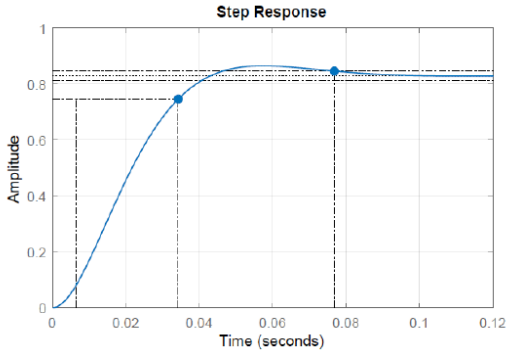


Figure 6.2.5: The step response of the compensated system.

Error Constants and System Type

The position and velocity error constants for a unity-gain feedback control system are defined as:

$$K_{\mathrm{p}} = \lim_{s\to 0} KG(s)\ , \quad K_{\mathrm{v}} = \lim_{s\to 0} sKG(s).$$

These constants can be inferred from the Bode plot at follows: The position error constant is given by the low frequency asymptote on the Bode magnitude plot, that is, $K_p = \lim_{s\to 0} |KG(j\omega)|$.

The velocity error constant is given as the slope of the low frequency asymptote on the Bode magnitude plot, that is, $\lim_{s\to 0} |KG(j\omega)| = \frac{K_v}{\omega}$.

The system type is inferred by the slope of the Bode magnitude plot in the low frequency region. Thus, a slope of $0dB$ implies a type 0 system; a slope of -1, i.e., $-20\ dB$/decade implies a type 1 system, etc.

System Sensitivity

The sensitivity of the closed-loop transfer function, $T(j\omega)$, to variations in the plant transfer function, $G(j\omega)$, is given as:

$$S_G^T(j\omega) = \frac{\partial T/T}{\partial G/G} = \frac{1}{1+KGH(j\omega)}$$

Keeping the sensitivity small over a frequency band requires a high loop gain, $KGH(j\omega)$. In general, increasing the loop gain decreases the stability margins, i.e., a trade-off exists between the sensitivity and relative stability requirements.

In the context of robust control, the closed-loop transfer function $T\,(j\omega)$ is referred as complementary sensitivity function.

The sensitivity and the complementary sensitivity are fundamentally constrained as: $S(j\omega)+T(j\omega)=1$ .The control system designers generally aim for high loop gain at low frequencies to obtain $T(j\omega)\cong 1$. The loop gain is reduced at high frequencies to raise the stability margins.

Loopshaping is a graphical technique that aims to impart a desired loop shape to the plot of $KGH(j\omega)$ by adding compensator poles and zeros to the loop.

6.3: Frequency Response Design

The frequency response design involves adding a compensator to the feedback loop to shape the frequency response function. The design aims to achieve the following:

1. A desired degree of relative stability and indicated by the phase margin.
2. A desired speed of response as indicated by the gain crossover frequency.
3. A mild slope of -1 (or $-20dB/\text{decade}$) at the crossover.

The choice of compensators in the frequency response design method includes the gain compensator, the phase-lag and phase-lead compensators, and the PD, PI, and PID compensators. These are described next.

Gain Compensation

The gain compensation aims to add a static compensator, $K(s) = K$, to the feedback loop.

The gain compensation raises the loop gain by K; the Bode magnitude plot is shifted up by $20\ \log_{10}\ K$. The resulting change in the gain crossover frequency, ω_{gc}, affects the PM.

The phase margin is reduced for ($K > 1$), and increased for ($K < 1$). Further, the system bandwidth increases for ($K > 1$), which improves the transient response by reducing the settling time. This is illustrated in the following example.

Example 6.3.1

Let $G(s) = \frac{2}{s(s+1)(s+2)}$; assume that the performance specifications state a 50°phae margin.

For $K = 1$, the GM, PM and the crossover frequencies are obtained from MATLAB as: $\text{GM} = 3\ (9.54\ \text{dB})$, $\omega_{\text{pc}} = 1.41\ \frac{\text{rad}}{\text{s}}$; $\text{PM} = 32.6^\circ$, $\omega_{\text{gc}} = 0.749\ \frac{\text{rad}}{\text{s}}$.

The gain crossover frequency need to be reduced to increase the phase margin. We observe from the Bode plot that $|G(j0.49)| = 5$ dB and $\phi\,(j0.49) = -130^\circ$.

Thus, reducing loop gain by $5dB$ will affect a crossover at $\omega_{\text{gc}} = 0.49\ \frac{\text{rad}}{\text{s}}$ and achieve a $\text{PM} = 50^\circ$. The required gain compensator is given as: $K = 0.56$.

The design is verified by plotting the frequency response for the loop transfer function: $KG(s) = \frac{1.12}{s(s+1)(s+2)}$, which shows $PM = 50^\circ$ at the new $\omega_{\text{gc}} = 0.49\ \frac{\text{rad}}{\text{s}}$ (Figure 6.3.1).

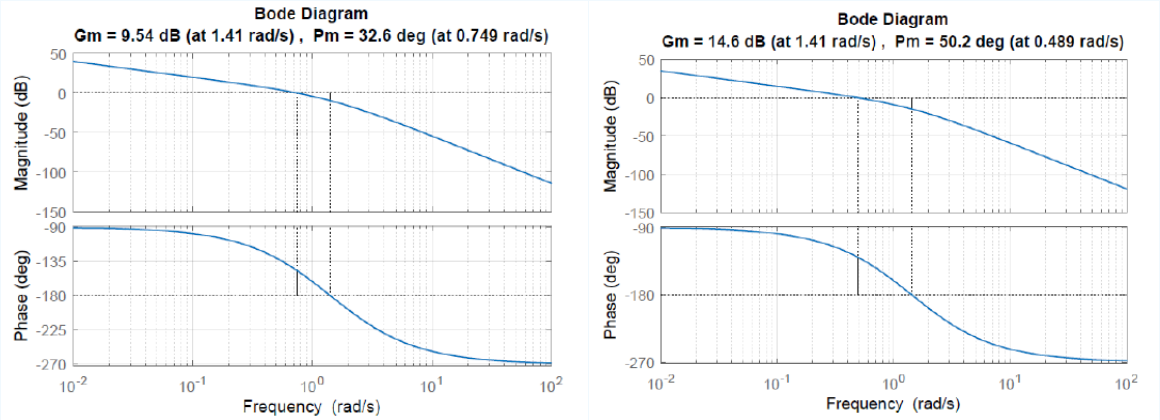


Figure 6.3.1: Gain compensation: uncompensated system (left); compensated system (right).

Phase-Lag Compensation

In the frequency response design, the phase-lag compensator serves a dual purpose: it can improve the phase margin (a measure of transient response), as well the DC gain (a measure of steady-state response).

The phase-lag compensator is described as:

$$K(s)=\frac{K(1+s/\omega_z)}{1+s/\omega_p},\quad K(j\omega)=\frac{K(1+j\omega/\omega_z)}{(1+j\omega/\omega_p)},\quad \omega_z>\omega_p$$

Phase Margin Improvement. For $K=1$, the phase-lag compensator is characterized by: $|KG(j0)|=0dB$ and $|KG(j\infty)|=-20log\left(\frac{\omega_z}{\omega_p}\right)$. Thus, the addition of the compensator to the feedback loop will lower the gain crossover frequency and increase the phase margin.

Error Constant Improvement. Alternatively, let $K=\frac{\omega_z}{\omega_p}$; then, the phase-lag compensator is characterized by: $KG(j0)=20log\left(\frac{\omega_z}{\omega_p}\right)$ and $KG(j\infty)=0dB$. Since the DC gain of the compensator is greater than one, addition of the compensator to the feedback loop will boost the relevant error constant by a factor of $K=\omega_z/\omega_p$.

In order to minimize the adverse phase contribution from the compensator, the compensator pole and zero locations for steady-state error improvement must be selected in accordance with: $\omega_{\mathrm{p}}<\omega_{\mathrm{z}}<0.1\omega_{\mathrm{gc}}$, where ω_{gc} is the gain crossover frequency.

Compensator Design. The phase-lag compensator design is summarized below:

1. Use gain compensation to adjust the dc gain $|KGH(j0)|$ to the desired value to meet the steady-state error requirement.
2. On the Bode plot of $KGH(j\omega)$, select a frequency ω_1 to satisfy $\angle KGH(j\omega_1)=-180^\circ+\phi_{\mathrm{m}}+5^\circ$, where ϕ_{m} is the desired phase margin, ω_1 serves as the new gain crossover frequency, and 5° is a safety margin to compensate for an estimated: $\angle K(j\omega_1)\cong-5^\circ$.
3. Select the compensator pole and zero frequencies as: $\omega_{\mathrm{z}}=0.1\omega_1$, $\omega_{\mathrm{p}}=\omega_{\mathrm{z}}/|\mathrm{KGH}(\mathrm{j}\omega_1)|$.
4. Draw the Bode plot for the compensated system and verify the design.

Example 6.3.2

Let $G(s)=\frac{2}{s(s+1)(s+2)}$; assume that the design specifications are: $PM=50^\circ$, and $e_{ss}|_{\mathrm{ramp}}<0.1(K_v>10)$. Then, the phase-lag compensator proceeds as follows:

1. Choose $K=11$ to meet the error constant requirement.
2. From the Bode plot for $KG(j\omega)$; choose $\omega_1=0.4\ rad/s$ for $\angle KG(j\omega_1)=-123^\circ$; then, $|\mathrm{KG}(j\omega_1)|=24.9\ (28\ \mathrm{dB})$.
3. Choose $\omega_{\mathrm{z}}=0.04,\ \omega_{\mathrm{p}}=0.0016$; then, $K(s)=\frac{11(1+s/0.04)}{1+s/0.0016}$.
4. The Bode plot of the compensated system shows $\omega_{gc}=0.4\ rad/s$ and $\mathrm{PM}=51^\circ$ (Fig. 6.3.2).

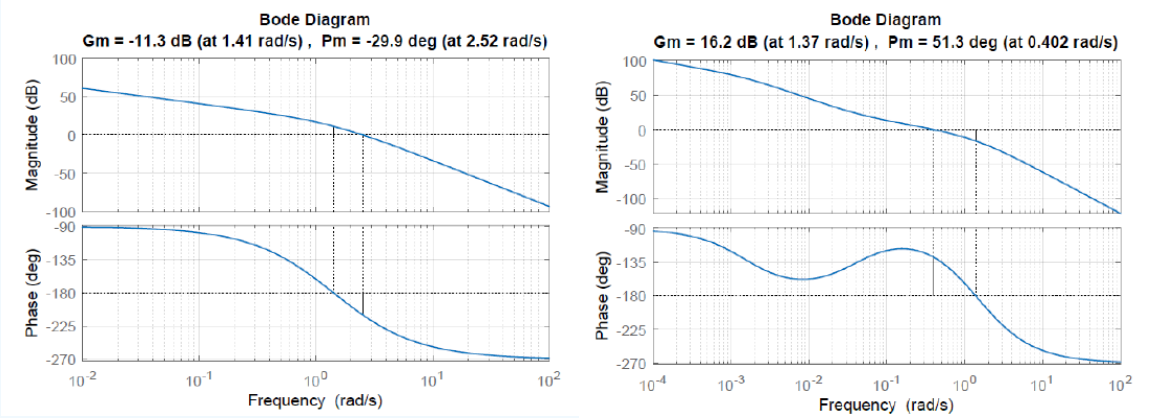


Figure 6.3.2: Phase-lag compensator design: uncompensated system (left); compensated system (right).

Phase-Lead Compensation

In the frequency response design, the phase-lead compensator serves to increase the closed-loop bandwidth, leading to transient response improvements. The phase-lead compensator is described by the transfer function:

$$K(s)=\frac{K(1+s/\omega_z)}{1+s/\omega_p},\quad K(j\omega)=\frac{K(1+j\omega/\omega_z)}{(1+j\omega/\omega_p)},\quad \omega_z<\omega_p$$

Bandwidth Improvement. For $K=1$, phase-lead compensator response is characterized by: $|KG(j0)|=0dB$ and $|KG(j\infty)|=20log\left(\frac{\omega_p}{\omega_z}\right)$.

Since the high frequency gain of the compensator is greater than one, its addition to the feedback loop increases the crossover frequency and the bandwidth.

The compensator contributes maximum phase-lead θ_m at a frequency $\omega_m=\sqrt{\omega_z\omega_p}$.

To proceed further, let $\alpha=\frac{\omega_p}{\omega_z}$; then, $\sin\theta_m=\frac{\alpha-1}{\alpha+1}$, or $\alpha=\frac{1+\sin\theta_m}{1-\sin\theta_m}$. Further, the compensator pole and zero locations are obtained as: $\omega_z=\frac{\omega_m}{\sqrt{\alpha}}$; $\omega_p=\omega_m\sqrt{\alpha}$.

Practically, the maximum achievable value of θ_m is about 70° for $\sqrt{\alpha}\cong 6$.

The compensator transfer function at ω_m is given as: $K(j\omega_m)=K\sqrt{\alpha}$. Then, from the crossover condition: $|KGH(j\omega_{gc})|=1$, we obtain: $K=\frac{\sqrt{\alpha}}{|GH(j\omega_{gc})|}$.

Compensator Design. The phase-lead design involves the following steps:

1. Choose a desired crossover frequency, ω_{gc}, as the greater of $\frac{8}{t_s\tan\phi_m}$ and ω_1 where $\angle KGH(j\omega_1)=-180^\circ+\phi_{\text{m}}+5^\circ$.
2. Compute the required compensator phase angle: $\theta=\phi_m-\angle G(j\omega_{\text{gc}})-180^\circ$; compute $|GH(j\omega_{\text{gc}})|$.
3. Solve for ω_z and ω_p.
4. Inspect the Bode plot of the compensated system and verify the design.

Example 6.3.3

Let $G(s)=\frac{2}{s(s+1)(s+2)}$; $PM=32.6^\circ$. The design specifications are: $\phi_m=50^\circ$, $t_s=4$ s. Then, the phase-lead compensator design steps are:

1. From the settling time requirement, we have, $\omega_{\text{gc}}=\frac{2}{\tan\phi_m}=1.68\ \frac{\text{rad}}{s}$; then, $\theta=50^\circ+9^\circ=59^\circ\cong 60^\circ$. Further, $G(j1.68)=0.23\ \angle -189^\circ$.
2. For $\theta=60^\circ$, we have $\alpha=13.93$; the phase-lead design is given as:

$\omega_z=0.45\frac{\text{rad}}{\text{s}}$, $\omega_p=6.27\frac{\text{rad}}{\text{s}}$, $K=16$; thus $K(s)=16\left(\frac{s+0.45}{s+6.27}\right)$

The Bode plot of the compensated system has $\phi_m=50.4^\circ$ at $\omega_{gc}=1.69\ \frac{\text{rad}}{\text{s}}$ (Figure 6.3.3).

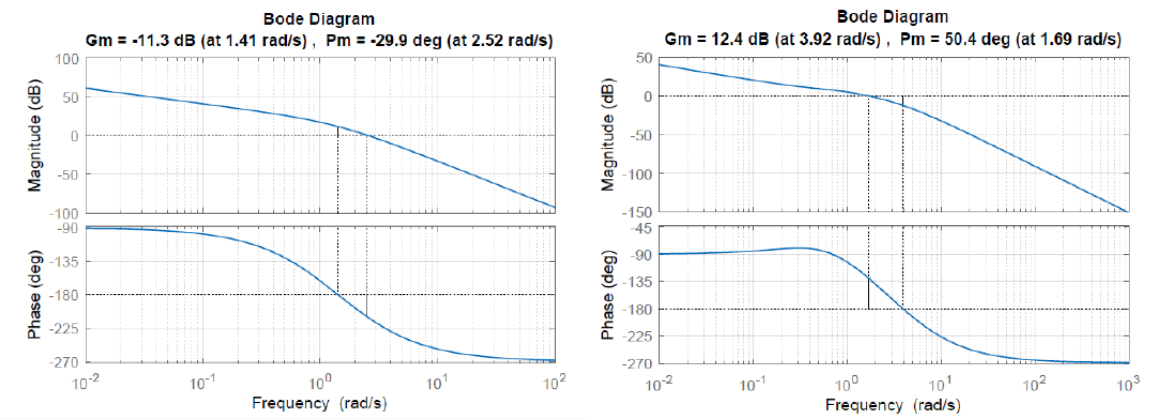


Figure 6.3.3: Phase-lead compensator design: uncompensated system (left); compensated system (right).

Lead-Lag Compensation

A lead-lag compensator combines the phase-lead and phase-lag sections. The phase-lag section improves the phase margin and the dc gain; the phase-lead section improves the bandwidth and the phase margin. Since both lead and lag sections can contribute to the phase margin improvement, the desired PM improvement can be distributed among the two sections.

The lead-lag compensator transfer function is given as: $K(s)=K_{lead}(s)K_{lag}(s)$.

The steps to design a lead-lag compensator are as follows:

1. Choose static gain K to meet the steady-state error requirement.
2. Design the phase-lag section to meet part of the phase margin requirement.
3. Design phase-lead section to meet the bandwidth/settling time requirement.

Example 6.3.4

Let $G(s)=\frac{2}{s(s+1)(s+2)}$; assume that the design specifications are: $\phi_m=50^\circ$, $t_s=4s$, and $e_{\text{ss}}|_{\text{ramp}}<0.1$. The design steps for the lag-lead design are as follows:

1. **DC Gain.** Choose $K=10$ to meet the e_{ss} requirement. Draw the Bode plot for $KG(s)$. The plot has $\omega_{gc}=2.5$ rad/s and $\text{PM}=-30^\circ$.
2. **Phase-Lag Design.** Suppose we aim to raise the phase margin to $\text{PM}\approx 40^\circ$. From the Bode plot, let $\omega_1=0.5$ rad/s, such that $KG\,(j0.5)=18.6\ (25.8\text{ dB})\angle -130^\circ$.
3. Complete the phase-lag design with: $\omega_z=0.05,\ \omega_{\text{p}}=0.003$.
4. **Phase-Lead Design.** From the Bode plot of $K_{lag}G(j\omega)$ choose $\omega_{\text{gc}}=1.7$ rad/s to meet the t_s requirement. Then, $K_{lag}G(j1.7)=0.15\ (-16.5\text{ dB})\angle -191^\circ$.
5. Compute the phase-lead compensator parameters: $\theta=62^\circ$, $\alpha=16,\ K=26.7,\ \omega_z=0.425\ rad/s,\ \omega_p=6.8$ rad/s .
6. The lead-lag compensator is described as: $K(s)=\frac{11(s+0.05)(s+0.24)}{(s+0.003)(s+4.7)}$. The Bode plot for the compensated system has $\text{PM}=50.2^\circ$ at $\omega=1.71$ rad/s (Figure 6.3.4).

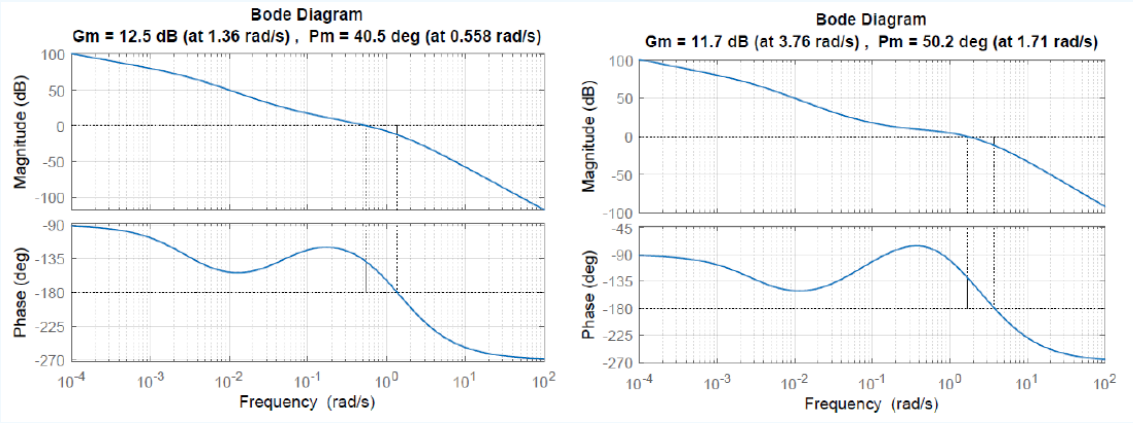


Figure 6.3.4: Lag-lead compensator design: phase lag compensation (left); phase-lead compensation (right).

The PID Compensator

The PID compensator is a combination of the PI and PD sections. The PI compensator adds an integrator to the feedback loop that reduces the steady-state error to zero; the PI compensator can also realize the PM requirement. The PI compensator is defined as:

$$K(s)=k_p+\frac{k_i}{s},\quad K(j\omega)=\frac{k_i(1+j\omega/\omega_z)}{j\omega},\quad \omega_z=\frac{k_i}{k_p}$$

The PI compensator zero is arbitrarily located close to the origin. The gain k_i is selected to achieve a desired PM improvement.

The PD compensator adds a first-order zero to the loop transfer function, which increases the bandwidth and hence the transient response. The PD compensator is defined as:

$$K(s)=k_d s+k_p,\ K(j\omega)=k_p(1+j\omega/\omega_z)$$

The desired gain crossover ω_{gc} is selected as the greater of $\frac{8}{t_s\tan\phi_m}$ and ω_1, where ω_1 is found from the Bode plot for $\angle KGH(j\omega_1)=-180^\circ+\phi_{\text{m}}-90^\circ+5^\circ$; the 90° term in this expression refers to the phase added by the PD compensator, and 5° is a safety margin.

The compensator zero location may be selected as: $\omega_z=0.1\omega_{gc}$; the compensator gain k_p is selected to meet the crossover condition: $|KGH(j\omega_{gc})|=1$.

Example 6.3.5

Let $G(s)=\frac{2}{s(s+1)(s+2)}$; assume that the design specifications are: $\phi_m=50^\circ$, $t_s=4$ s, and $e_{\text{ss}}|_{\text{ramp}}<0.1$.

No gain adjustment is needed as the PI compensator will boost the position error constant $K_{\text{p}}\to\infty$.

1. **PI design.** We aim to achieve $PM=30^\circ$ in the PI section. The Bode plot for $G(j\omega)$ has $G(j0.8)=0.9\angle-150^\circ$. Let $\omega_z=0.05$; then $K_{PI}(s)=\frac{0.9(s+.05)}{s}$.
2. **PD design.** From the settling time requirement, $\omega_{\text{gc}}=\frac{2}{\tan\phi_{\text{m}}}\cong 1.7\ \frac{\text{rad}}{\text{s}}$; from the phase angle requirement, $\omega_1=2.5\ \frac{\text{rad}}{\text{s}}$; hence, $\omega_{\text{gc}}=2.5\ \frac{\text{rad}}{\text{s}}$; the Bode plot of $K_{PI}G(j\omega)$ shows $G(j2.5)=0.1\angle-210^\circ$.
3. The PD compensator is zero selected as: $\omega_z=0.2\ \frac{\text{rad}}{\text{s}}$, $K_{PD}(s)=5(s+0.2)$.
4. The PID compensator is given as: $K(s)=\frac{4.5(s+0.05)(s+0.2)}{s}$. The Bode plot of the compensated system has $\phi_m=53.4^\circ$ at $\omega_{\text{gc}}=2.58\ \frac{\text{rad}}{\text{s}}$ (Figure 6.3.5).

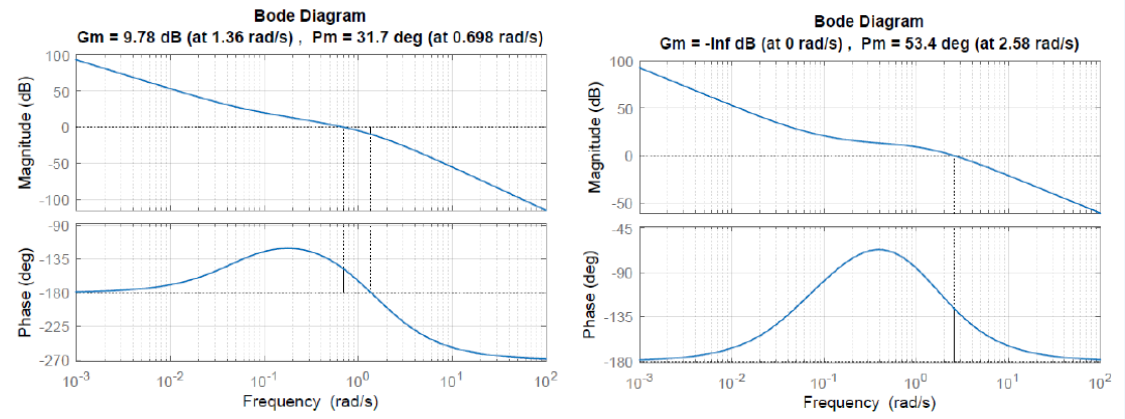


Figure 6.3.5: The PID compensator design: PI compensation (left), PD compensation (right).

6.4: Closed-Loop Frequency Response

Closed-Loop Frequency Response

The closed-loop frequency response reveals important information about the relative stability and the speed of response in the time-domain. For unity-gain feedback configuration $(H(s)=1)$, the closed-loop frequency response is computed as:

$$T(j\omega)=\frac{KG(j\omega)}{1+KG(j\omega)}$$

To proceed further, we use rectangular form expression for the open loop frequency response: $KG(j\omega)=X(j\omega)+jY(j\omega)$; then,

$$T(j\omega)=\frac{X+jY}{(1+X)+jY}=Me^{j\phi}$$

and

$$|T(j\omega)|^2=\frac{X^2+Y^2}{(1+X)^2+Y^2}=M^2.$$

The latter can be rearranged to obtain:

$$\left(X+\frac{M^2}{M^2-1}\right)^2+Y^2=\frac{M^2}{(M^2-1)^2}.$$

The above relation represents the equation of a circle with center at $X=\frac{M^2}{1-M^2}$, $Y=0$, and radius $r=\frac{M}{1-M^2}$; these are described as constant M circles on the frequency response plot. Further, as $M\to 1$, $X=-\frac{1}{2}$, which is a vertical line that separates the circles for $M<1$ from those with $M>1$.

The phase relationship similarly reveals an equation for constant phase circles:

$$\left(X+\frac{1}{2}\right)^2+\left(Y-\frac{1}{2N}\right)^2=\frac{1}{4}+\left(\frac{1}{2N}\right)^2$$

where $N=\tan\phi$. The constant phase circles are centered at: $X=-\frac{1}{2}$, $Y=\frac{1}{2N}$, and have radius: $r=\sqrt{\frac{1}{4}+\left(\frac{1}{2N}\right)^2}$.

Resonance Peak

The resonance peak in the frequency response occurs at $\omega=\omega_r$ for $\zeta<0.7$. In particular, for a prototype second-order system, we have:

Resonant frequency: $\omega_r=\omega_n\sqrt{1-2\zeta^2}$

Resonant peak: $M_r=|T(j\omega_r)|=\frac{1}{2\zeta\sqrt{1-\zeta^2}}$

When the polar plot of the loop gain, $KGH(j\omega)$, is superimposed onto the constant M and constant N contours, it reveals the magnitude peak M_r in $T(j\omega)$. The MATLAB Control System Toolbox 'grid' command adds constant M,N contours on the Nyquist plot.

The resonance peak in the closed-loop frequency response represents a measure of relative stability; the resonant frequency serves as a measure of speed of response in the time-domain. A value of $M_r=1.3(or\ 2.5dB)$ is considered a good compromise between speed and stability.

The Nichol's Chart

The closed-loop frequency response can be alternately visualized on the Nichol's chart, where the magnitude in dB is plotted along the vertical axis, and the phase in degrees is plotted along the horizontal axis.

The MATLAB Control System Toolbox 'grid' command similarly adds constant M,N contours on the Nichol's chart.

Example 6.4.1

Let $KG(s)=\frac{10}{s(s+1.86)}$, $KG(j\omega)=\frac{10}{-\omega^2+j1.86\omega}$;

then, the closed-loop frequency response has a peak $M_r=5$ dB, as can be observed on both the Nyquist plot and Nichol's chart in Figure 6.4.1 (an imaginary 5dB circles touches the $KG(j\omega)$ curve).

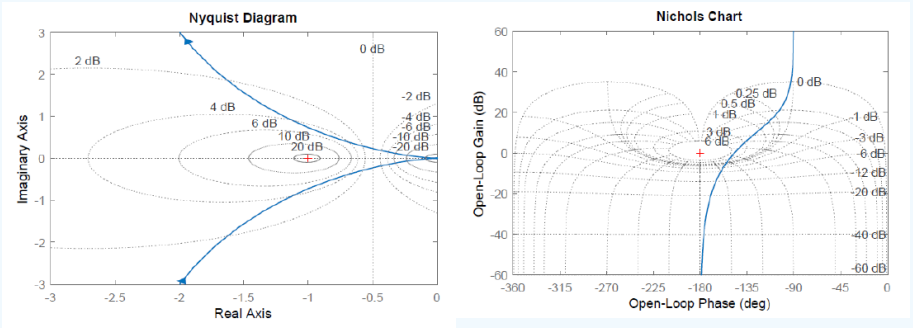


Figure 6.4.1: Closed-loop frequency response evaluation on the Nyquist plot (left) and Nichol's chart (right).

CHAPTER OVERVIEW

7: Design of Sampled-Data Systems

Learning Objectives

1. Understand and analyze the sampled-data systems models.
2. Solve difference equation models of sampled-data systems by iteration.
3. Analyze the stability of closed-loop sampled-data systems.
4. Design/emulate controllers for the sampled-data system models.

7.0: Prelude to Design of Sampled-Data Systems

The sampled-data control systems include clock-driven elements and reflect the current trends in the design of feedback control systems. In the contemporary control systems technology, data acquisition card (DAQ) is commonly used to sense, sample, and process variables of interest. The controller is digitally implemented as a software routine on a programmable logic controller (PLC), microcontroller, or digital signal processor (DSP).

The sampled-data control systems employ software-based controllers that work with discretized time. The discrete-time system models are represented by difference equations, with input and output variables represented by number sequences. The analog-to-digital (ADC) and digital-to-analog (DAC) converters are modeled as sampler and hold devices (Figure 7.0.1).

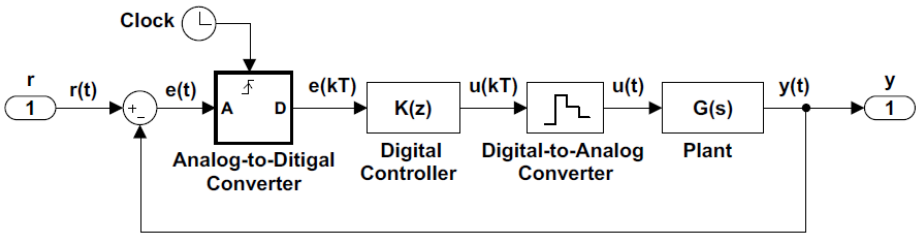


Figure 7.0.1: A sampled-data feedback control system

A continuous-time system model can be converted to a discrete system model by assuming a piece-wise constant input generated by a zero-order hold (ZOH). In the MATLAB, the continuous-to-discrete conversion is handled by the ‘c2d’ function in the Control Systems Toolbox. The function allows a variety of input methods.

The z-transform for discrete-time systems serves as the equivalent of Laplace transform for continuous-time systems. Discrete system models are represented by pulse transfer functions that are valid at sampling instances. The added phase angle due to sampling adversely affects the dynamic stability of the closed-loop system. The discrete system stability is indicated by the roots of the characteristic polynomial being restricted to inside of unit circle.

Analog controllers designed for transfer function models of continuous-time systems can be approximated for their application toward sampled-data systems. Assuming a high enough sampling rate (five to ten times the system bandwidth), the digital controller obtained by emulation gives comparable performance to the analog controller it mimics.

Root locus technique can be similarly used for controller design in the case of discrete systems. The design is performed on the z-plane, keeping in view the stability boundary, i.e., the unit circle. The performance criteria defined in terms of settling time, damping ratio, etc., can be reflected on the z-plane, as conveniently done by using the ‘grid’ command in MATLAB.

In this chapter, we will discuss models of sampled-data system, their properties, stability characterization, and the analysis and controller design for such systems.

7.1: Models of Sampled-Data Systems

7.1.1 Sampled-Data Systems

The sampled-data systems operate on discrete-time, represented as integral multiples of the sampling time (T).

To model the sampled-data systems, we consider an ideal sampler that samples a physical signal $r(t)$ every T seconds and generates a series of impulses with weights $r(kT), \quad k = 0, 1, \ldots$.

Mathematically, the sampler output, $r^*(t)$, represents multiplication of $r(t)$ with an impulse train, given as:

$$r^*(t) = \sum_{k=0}^{\infty} r(kT)\delta(t - kT)$$

By applying the Laplace transform to the sampled signal $r^*(t)$, we obtain:

$$r^*(s) = \sum_{0}^{\infty} r(kT)e^{-skT}$$

Since sampling time, T, is constant, we can change the variable to: $z = e^{sT}$, to represent the sampled signal as:

$$r(z) = \sum_{0}^{\infty} r(kT)z^{-k}$$

where $r(z)$ defines the z-transform of the sampled signal, $r(kT)$.

7.1.2 The z-transform

The z-transform is the Laplace transform equivalent in the case of sampled-data systems. The domain of the z-transform includes real- or complex-valued number sequences.

Assuming the number sequence, $r(kT)$, is obtained by sampling a real-valued signal $r\,(t)$, the time dependence may be suppressed to represent the sequence as: $r\,(k) = \{r\,(0)\,, r\,(1)\,, \ldots\}$. The z-transformed sequence is given as:

$$r(z) = z[r(kT)] = \sum_{0}^{\infty} r(k)z^{-k}$$

The transformed sequence, $r(z)$, represents a rational function of a complex variable $\; z = |z|\, e^{j\theta}$.

Example 7.1.1

The z-transform of a unit-step sequence: $u(k) = \{1,\ 1,\ \ldots\}$ is given as:

$$u(z) = \sum_{0}^{\infty} z^{-k} = \frac{1}{1 - z^{-1}}; \quad |z^{-1}| < 1$$

The convergence of the above geometric series is conditioned on: $|z^{-1}| < 1$, or $|z| > 1$which defines its region of convergence (ROC); the ROC is outside of the unit-circle: $e^{j\theta},\ 0 \leq \theta < 2\pi$ in the complex z-plane.

Example 7.1.2

Let $r(t) = e^{-at}u(t)$; then $r(kT) = \{1,\ e^{-aT}, e^{-2aT}, \ldots\}$. Hence,

$$r(z) = \sum_{0}^{\infty} e^{-akT}z^{-k} = \frac{1}{1 - e^{-aT}z^{-1}}; \ |e^{-aT}z^{-1}| < 1$$

The ROC is outside of the circle of radius e^{-aT} in the complex z-plane.

In digital signal processing (DSP) applications, the index 'n' is commonly used to represent a real-valued sequenc, i.e., the z-transform of a sequence, $r(n) = a^n u(n)$ is given as: $r(z) = \frac{1}{1 - az^{-1}}$.

Example 7.1.3

Let $r(t) = e^{j\omega t}u(t)$; then $r(kT) = \{1, e^{j\omega T}, e^{2j\omega T}, \ldots\}$. Hence,

$$r(z) = \sum_{0}^{\infty} e^{jk\omega T} z^{-k} = \frac{1}{1 - e^{j\omega T} z^{-1}}; \quad |e^{j\omega T} z^{-1}| = |z^{-1}| < 1$$

The ROC is outside of the unit-circle in the complex z-plane.

The z-transforms of the sampled sinusoidal signals are obtained by applying the Euler's identity to the above z-transform of $e^{jk\omega T}$ and separating the real and imaginary parts. Thus

$$\sin(k\omega T) \overset{z}{\leftrightarrow} \frac{\sin(\omega T)\ z}{z^2 - 2\ \cos(\omega T)\ + 1}$$

$$\cos(k\omega T) \overset{z}{\leftrightarrow} \frac{z(z - \cos(\omega T)\ )}{z^2 - 2\ \cos(\omega \mathrm{T})\ + 1}$$

$$e^{-akT}\ \sin(k\omega T) \overset{z}{\leftrightarrow} \frac{e^{-aT}\ \sin(\omega T) z}{z^2 - 2\ \cos(\omega T)\ e^{-aT} + e^{-2aT}}$$

$$e^{-akT}\ \cos(k\omega T) \overset{z}{\leftrightarrow} \frac{z(z - e^{-aT}\ \cos(\omega T))}{z^2 - 2\ \cos(\omega T)\ e^{-aT} + e^{-2aT}}.$$

The z-transforms of other complex signals may be obtained by using its properties of linearity, differentiation, and translation, etc.

Inverse z-Transform

Given the z-transform of a composite signal, the underlying sequence can be recovered by long division. Alternatively, partial fraction expansion (PFE) can be used to obtain first and second-order factors that can be inverse transformed with the help of z-transform tables.

Example 7.1.4

Let $y\,(z) = \frac{0.2z(z+0.9)}{(z-1)(z-0.6)(z-0.9)}$ represent the output of a feedback control system; then, the output sequence can be recovered by the following methods:

1. Expand $\frac{y(z)}{z}$ in partial fractions to obtain: $y\,(z) = \frac{4.75z}{z-1} - \frac{6z}{z-0.9} + \frac{1.25}{z-0.6}$. Hence, $y\,(k) = 4.75 - 6(0.9)^k + 1.25(0.6)^k$.
2. Alternatively, use long division to obtain: $y\,(z) = 0.1z^{-1} + 0.34z^{-2} + 0.65z^{-3} + 0.98z^{-4} + \ldots$. Hence, $y\,(k) = \{0,\ 0.1,\ 0.34,\ 0.65,\ 0.98,\ \ldots\}$

7.2: Pulse Transfer Function

Zero-Order Hold

A zero-order hold (ZOH) reconstructs a piece-wise constant signal from a number sequence and represents a model of the digital-to-analog converter (DAC).

Assuming that the input to the ZOH is a sampled signal, $r(kT) = r(t)|_{t=kT}$, its output is given as:

$$r(t) = r((k-1)T) \quad \text{for} \quad (k-1)T \le t < kT$$

The output of the ZOH to an arbitrary input, $r\,(kT)$, is a staircase reconstruction of the analog signal, $r(t)$.

The impulse response of the ZOH is square pulse (Figure 7.2): $g_{ZOH}\,(t) = 1,\ 0 < t < 1$.

By applying Laplace transform to the ZOH impulse response, its transfer function is obtained as:

$$G_{ZOH}(s) = \frac{1}{s} - \frac{e^{-sT}}{s} = \frac{1 - e^{-sT}}{s}$$

Further, the frequency response function of the ZOH is given as:

$$G_{\text{ZOH}}(\omega) = \frac{1 - e^{-j\omega T}}{j\omega} = T\frac{\sin(\omega T/2)}{\omega T/2} e^{-j\omega T/2}$$

Thus, the frequency response of the ZOH is a $sinc$ function with a phase delay. When placed in the feedback loop, the added phase from the ZOH reduces the available phase margin, adversely impacting closed-loop stability.

Specifically, let ω_{gc} denote the gain crossover frequency; then, the phase margin is reduced by $\omega_{gc}T/2$.

The acceptable degradation in the PM can be used to select the sample time, T. For example, if we want to limit the PM degradation to 5° $(0.087\ rad)$, the sampling time may be selected as: $T \le \frac{0.175}{\omega_{gc}}$.

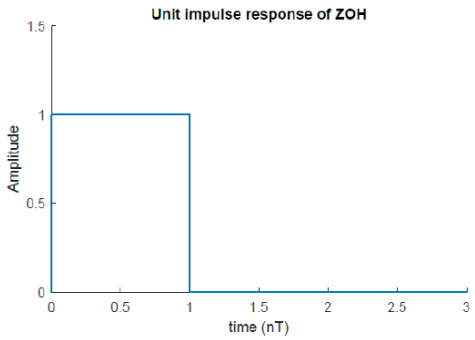


Figure 7.2.1: Impulse response of the ZOH.

The Pulse Transfer Function

We use z-transform to obtain a transfer function description of the plant cascaded with the ZOH. The result is the pulse transfer function, $G(z)$, that is valid at the sampling intervals.

The pulse transfer function of a continuous-time plant, $G(s)$, is obtained as:

$$G(z) = \mathrm{z}\left\{\frac{1 - e^{-sT}}{s} G(s)\right\} = (1 - z^{-1})\mathrm{z}\left\{\frac{G(s)}{s}\right\}$$

where $z\left\{\frac{G(s)}{s}\right\}$ denotes the z-transform of a sequence obtained by sampling $\int g\,(t)\,dt$.

Example 7.2.1

Let $G(s) = \frac{a}{s+a}$; then $G(z) = (1 - z^{-1})\ \mathrm{z}\left\{\frac{a}{s(s+a)}\right\} = (1 - z^{-1})\mathrm{z}\left\{\frac{1}{s} - \frac{1}{s+a}\right\}$.

By using the z-transform table, we obtain: $G(z) = \frac{z-1}{z}\left(\frac{z}{z-1} - \frac{z}{z-e^{-aT}}\right) = \frac{1-e^{-aT}}{z-e^{-aT}}$.

As an example, let $G(s) = \frac{1}{s+1}$, $T = 0.2s$. Then, the pulse transfer function is obtained as: $G(z) = \frac{0.181}{z-0.819}$, or $G(z) = \frac{1-e^{-0.2}}{z-e^{-0.2}}$.

Example 7.2.2

Let $G(s) = \frac{a}{s(s+a)}$; then $G(z) = (1-z^{-1})\ \mathrm{z}\left\{\frac{a}{s^2(s+a)}\right\} = (1-z^{-1})\mathrm{z}\left\{\frac{1}{s^2} - \frac{1/a}{s} + \frac{1/a}{s+a}\right\}$.

By using the z-transform table, we obtain: $G(z) = \frac{z-1}{z}\left[\frac{Tz}{(z-1)^2} - \frac{1}{a}\left(\frac{z}{z-1} - \frac{z}{z-e^{-aT}}\right)\right] = \frac{aT(z-e^{-aT})-(z-1)(1-e^{-aT})}{a(z-1)(z-e^{-aT})}$.

As an example, let $G(s) = \frac{1}{s(s+1)}$, $T = 0.2s$; then, the pulse transfer function is obtained as: $G(z) = \frac{0.0187(z+0.936)}{(z-1)(z-0.819)}$.

From the above examples, we observe that:

1. The order of the pulse transfer function, i.e., the degree of the denominator polynomial in$G(z)$, matches that of the continuous-time transfer function, $G(s)$.
2. The poles of the pulse transfer function are related to the transfer function poles as: $z_i = e^{s_i T}$.
3. The pulse transfer functions of the second and higher order systems additionally includes finite zeros.

In the MATLAB Control Systems Toolbox, the pulse transfer function is obtained by using the “c2d” command and specifying a sampling time (T_s). The command is invoked after defining the continuous-time transfer function model.

The choice of the sampling time is guided by the natural frequency of the system. In particular, we may choose $T_s \ll 1/\omega_n$, where ω_n represents the natural frequency.

7.3: Sampled-Data System Response

The response of a sampled-data system, $G(z)$, to an input sequence, $u(k)$, is the output sequence, $y(k)$. In the z-domain, the system response is given as: $y(z) = G(z)\,u(z)$.

Alternatively, we may use inverse z-transform to obtain an input–output description in the form of a difference equation that can be solved by iteration.

7.3.1 Difference Equation Representation

Let $G(z) = \frac{n(z)}{d(z)}$, where $n(z)$ and $d(z)$ are mth order and nth order polynomials, given as:

$$n(z) = \sum_{i=0}^{m} b_i z^{m-i}, \quad d(z) = z^n + \sum_{i=1}^{n} a_i z^{n-i}.$$

Then $d(z)\,y(z) = n(z)\,u(z)$. By applying the inverse z-transform, we obtain a difference equation, described as:

$$y(k+n) + \sum_{i=1}^{n} a_i y(k+n-i) + \sum_{i=0}^{m} b_i u(k+m-i)$$

As the system is linear and time-invariant, we apply a time shift and rearrange the difference equation into and update rule:

$$y(k) = -\sum_{i=1}^{n} a_i y(k-i) + \sum_{i=0}^{m} b_i u(k-i)$$

The above rule can be programmed on a computer to solve for the response of the sampled-data system to an input sequence, $u(k)$, described as: as $u(k) = u(t)|_{t=kT}$.

Example 7.3.1

Let $G(z) = \frac{y(z)}{u(z)} = \frac{0.181}{z-0.819}$; then, the sampled-data system is described by the following input–output relation:

$$y(k) = 0.819y(k-1) + 0.181u(k-1)$$

Example 7.3.2

Let $G(z) = \frac{0.0187(z+0.936)}{(z-1)(z-0.819)}$; then, the sampled-data system is described by the following input–output relation:

$$y(k) = 1.819y(k-1) - 0.819y(k-2) + 0.0187u(k-1) + 0.0175u(k-2)$$

Unit-Pulse Response

The unit-pulse response of a sampled-data system model, $G(z)$, is its response to a unit-pulse $r(k) = \delta(k)$.

Using $r(z) = 1$, the unit-pulse response is computed as: $g(kT) = z^{-1}[G(z)]$, with an implicit sampling time $T = 1s$.

Unit-pulse response for sample time, T, is obtained by assuming $\delta(z) = \frac{1}{T}$, hence $g(kT) = \frac{1}{T} z^{-1}[G(z)]$.

Alternatively, given the input–output description, the unit-pulse response can be computed by iteration.

The unit-pulse response comprises the natural response modes $G(z) = n(z)/d(z)$. These modes depend on the roots of $d(z)$, and are characterized as follows:

Real Roots. Assume that $d(z)$ has real and distinct roots: z_i, $i = 1, \ldots, n$,then, the natural response modes are given as: $\phi_i(k) = (z_i)^k$. These modes die out with time if $|z_i| < 1$.

Complex Roots. Assume that $d(z)$ has complex roots of the form: $re^{\pm j\theta}$; then, its natural response modes are given as: $\phi_i(k) = \{r^k \cos k\theta,\ r^k \sin k\theta\}$. These modes similarly die out with time if $|r| < 1$.

Example 7.3.3

Let $G(s)=\frac{1}{s+1}$; use $T=0.2$s to obtain: $G(z)=\frac{0.181}{z-0.819}$. The sampled-data system is described by the following input-output relation: $y(k)=0.819y(k-1)+0.181u(k-1)$.

As $T=0.2s$, let $\delta(k)=\{5,\ 0,\ 0,\ldots\}$; then, assuming zero initial conditions, the unit-pulse response is obtained as:

$$y(k)=\{0,0.906,\ 0.742,\ 0.608,\ 0.497,0.407,\ 0.333,\ \ldots\}$$

For comparison, the impulse response of the analog system is obtained as: $g(t)=e^{-t}u(t)$.

The impulse responses of continuous and discrete systems are compared in Figure 7.3.1.

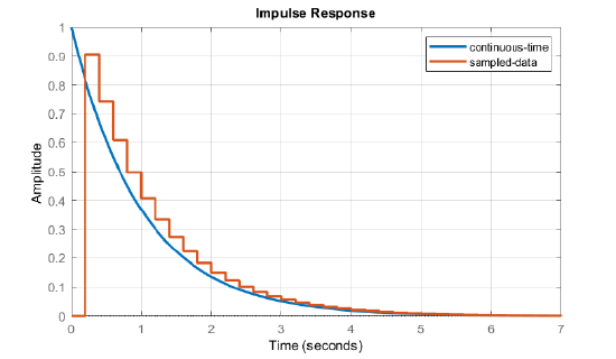


Figure 7.3.1: Unit-pulse responses of continuous-time and sampled-data systems for Example 7.3.3.

Example 7.3.4

Let $G(s)=\frac{1}{s(s+1)}$; use $T=0.2$s to obtain: $G(z)=\frac{0.0187(z+0.936)}{(z-1)(z-0.819)}$. The sampled-data system is described by the following input-output relation:

$$y(k)=1.819y(k-1)-0.819y(k-2)+0.0187u(k-1)+0.0175u(k-2).$$

Assuming $u(k)=\{5,0,0,\ldots\}$, and zero initial conditions, the unit-pulse response is obtained as:

$$y(k)=\{0,0.0937,\ 0.258,\ 0.392,\ 0.502,0.593,\ 0.667,\ \ldots\}$$

For comparison, the impulse response of the original analog system is obtained as: $g(t)=(1-e^{-t})u(t)$.

The impulse responses of continuous and discrete systems are compared in Figure 7.3.2. We note that the presence of integrator in the transfer function causes the impulse response to asymptotically approach unity in the steady-state.

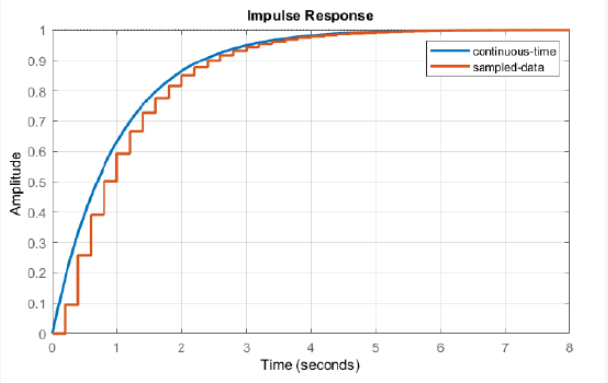


Figure 7.3.2: Unit-pulse responses of continuous-time and sampled-data systems for Example 7.3.4.

In the MATLAB Control Systems Toolbox, the 'impulse' command can be used to obtain or plot the unit-pulse response of a sampled-data system.

Unit-Step Response

The unit-step response of a sampled-data system model, $G(z)$, is its response to the unit-step sequence: $r\{kT\}=\{1,1,1\ldots\}$; $r(z)=\frac{1}{1-z^{-1}}$. In the z-domain, the unit-step response is obtained as: $y(z)=\frac{G(z)}{1-z^{-1}}$.

Alternatively, using the input-output system description, the unit-step response can be obtained by iteration.

Example 7.3.5

Let $G(s)=\frac{1}{s+1}$; use $T=0.2$s to obtain: $G(z)=\frac{0.181}{z-0.819}$. The unit-step response is computed as:$y(z)=\frac{0.181z}{(z-1)(z-0.819)}$.

We use PFE of $y(z)/z$ to obtain: $y(z)=\left(\frac{z}{z-1}-\frac{z}{z-0.819}\right)$. Using the inverse z-transform, the step response sequence is obtained as: $y(kT)=1-(0.819)^k,\quad k=0,1,\ldots$

Alternatively, we use the input-output system description with zero initial conditions: $y(k)=0.819y(k-1)+0.181u(k-1)$. The resulting unit-step sequence is given as: $y(k)=\{0,\ 0.181,\ 0.329,\ 0.451,\ 0.55,\ldots\}$

For comparison, the unit-step response of continuous-time system is given as:$y(t)=1-e^{-t},\quad t>0$. The step responses of the continuous and discrete systems are plotted alongside (Figure 7.3.3).

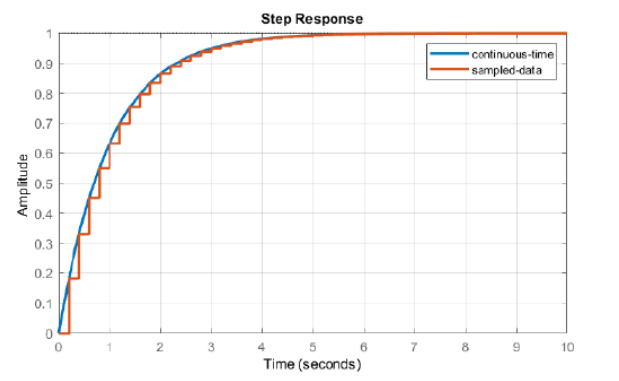


Figure 7.3.3: Unit-step responses of continuous-time and sampled-data systems for Example 7.3.5.

Example 7.3.6

Let $G(s)=\frac{1}{s(s+1)}$; use $T=0.2$s to obtain: $G(z)=\frac{0.0187z+0.0175}{z^2-1.819z+0.819}$. Using the time-domain relation (Example 6.8), the unit-step response is given as: $\{0,\ \ 0.0187,\ 0.0702,\ 0.149,\ 0.249,\ 0.368,\ 0.501, 0.647, 0.802, 0.965,\ldots\}$

Analytically, let $r(z)=\frac{z}{z-1}$ (unit-step); then, the system output is given as: $y(z)=\frac{z(0.0187z+0.0175)}{(z-1)(z^2-1.819z+0.819)}$. By using long division, we can express the quotient as:

$$y(z)=0.0187z^{-1}+0.0703z^{-2}+0.149z^{-3}+0.249z^{-4}+0.501z^{-5}+\ldots$$

The resulting output sequence matches the one obtained by iteration.

For comparison, continuous-time system step response is given as: $y(t)=(t-1+e^{-t})u(t)$.

The step responses of the continuous and discrete systems are plotted alongside (Figure 7.3.4).

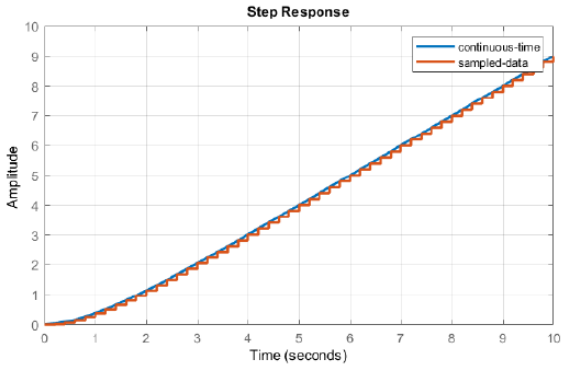


Figure 7.3.4: Unit-step responses of continuous-time and sampled-data systems for Example 7.3.6.

In the MATLAB Control Systems Toolbox, the 'step' command can be used to obtain or plot the unit-step response of a sampled-data system.

Response to Arbitrary Inputs

Given the unit-pulse response of the system, its response to an arbitrary input sequence is obtained by convolution. Accordingly, let the unit-pulse response sequence be given as:

$$\{g(k)\} = \{g(1),\ g(2),\ \ldots\}$$

Then, using an arbitrary input sequence: $\{u(k)\} = \{u(1),\ u(2),\ \ldots\}$, the output sequence is given by the convolution sum:

$$y(k) = g(k) * u(k) = \sum_{i=0}^{\infty} g(k-i)\, u(i)$$

As an example, we obtain the output for a sinusoidal input below.

Example 7.3.7

Let $G(z) = \frac{0.368z+0.264}{(z-1)(z-0.368)}$; we assume a sinusoidal input sequence, $u(kT) = sin(\pi kT)$. Then, the output response is computed using the convolution sum is shown (Figure 7.3.5).

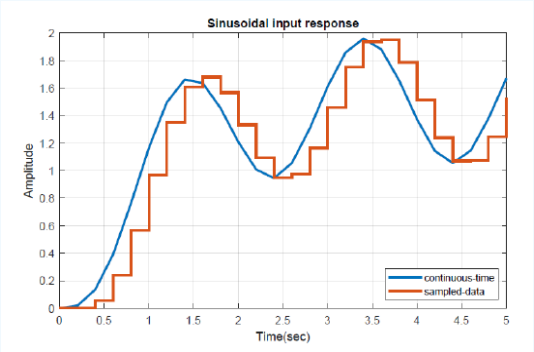


Figure 7.3.5: Sinusoidal response of a second-order sampled-data system preceded by a ZOH.

7.4: Stability of Sampled-Data Systems

Stability Region in the Complex Plane

In continuous-time control system design, $s = j\omega$ defines the stability boundary.

The z-plane stability boundary is obtained from the transform: $z = e^{j\omega T} = 1\angle\omega T$; it maps the $j\omega$-axis to the unit circle and the open left-half plane to the inside of the unit circle: $|z| < 1$.

Thus, the sampled-data system is stable if and only if the pulse characteristic polynomial $\Delta(z)$ has its roots inside the unit circle, i.e., $|z_i| < 1$.where z_i is the root of $\Delta(z)$.

The analytical conditions for a z-domain polynomial, $A(z)$, to have its roots inside the unit circle are given by the Schur–Cohn stability test. When applied to real polynomials, the Schur–Cohn test results in a criterion similar to the Ruth's test, and is known as Jury's stability test.

Jury's Stability Test

Assume that the nth order z-domain polynomial to be investigated is given as:

$$A(z) = a_0 z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n; a_0 > 0.$$

For stability determination, the Jury's table is built as follows:

$$\left|\begin{array}{cccc} a_n & a_{n-1} & \ldots & a_0 \\ a_0 & a_1 & \ldots & a_n \\ b_{n-1} & b_{n-2} & \ldots & b_0 \\ b_0 & b_1 & \ldots & b_{n-1} \\ c_{n-2} & c_{n-1} & \ldots & c_0 \\ c_0 & c_1 & \ldots & c_{n-2} \\ & & \vdots & \end{array}\right.$$

where the first two rows reflect the coefficients of the polynomial; the coefficients of the third and subsequent rows are computed as:

$$b_k = \begin{vmatrix} a_n & a_{n-1-k} \\ a_0 & a_{k+1} \end{vmatrix}, \quad k = 0, \ldots, n-1$$

$$c_k = \begin{vmatrix} b_{n-1} & b_{n-2-k} \\ b_0 & b_{k+1} \end{vmatrix}, \quad k = 0, \ldots, n-2$$

and so on. The necessary conditions for polynomial stability are:

$$A(1) > 0, (-1)^n A(-1) > 0.$$

The sufficient conditions for stability, given by the Jury's test, are:

$$a_0 > |a_n|, \ |b_{n-1}| > |b_0|, \ |c_{n-2}| > |c_0|, \ldots (\mathrm{n}-1)\mathrm{constraints}.$$

Second-Order Polynomial

Let $A(z) = z^2 + a_1 z + a_2$; then, the Jury's table is given as:

$$\left|\begin{array}{ccc} a_2 & a_1 & 1 \\ 1 & a_1 & a_2 \\ b_1 & b_0 & \\ b_0 & b_1 & \end{array}\right.$$

The resulting necessary conditions are: $1 + a_1 + a_2 > 0, \quad 1 - a_1 + a_2 > 0.$

The sufficient conditions are: $|a_2| < 1, |1 + a_2| > |a_1|$.

Stability Determination through Bilinear Transform

The bilinear transform (BLT) defines a linear map between s-domain and z-domain.

The BLT is based on the first-order Pade' approximation of $z = e^{sT}$, given as:

$$z = \frac{e^{sT/2}}{e^{-sT/2}} \cong \frac{1+sT/2}{1-sT/2}, \quad s = \frac{2}{T}\frac{z-1}{z+1}$$

Since T has no impact on stability determination, we may use $T = 2$ for simplicity. Further, to differentiate from continuous-time systems, a new complex variable, w, is introduced. Thus $z = \frac{1+w}{1-w}$, $w = \frac{z-1}{z+1}$

Let $\Delta(z)$ represent the polynomial to be investigated; application of BLT to $\Delta(z)$ returns a polynomial, $\Delta(w)$, whose stability is determined through the application of Hurwitz criterion (Sec. 2.5).

Second-Order Polynomial

Let $\Delta(z) = z^2 + a_1 z + a_2$; then, application of BLT, ignoring the denominator term, results in:

$$\Delta(w) = \Delta(z)|_{z=\frac{1+w}{1-w}} = (1 - a_1 + a_2)w^2 + 2(1 - a_2)w + 1 + a_1 + a_2$$

Application of the Hurwitz criterion results in the following stability conditions for $\Delta(z)$:

$$a_2 + a_1 + 1 > 0, a_2 - a_1 + 1 > 0, 1 - a_2 > 0$$

The above conditions are similar those obtained from the application of Jury's stability test.

Stability of the Closed-Loop System

Let the pulse transfer function be given as: $G(z) = \frac{n(z)}{d(z)}$; then, for a static controller, the closed-loop pulse characteristic polynomial is given as: $\Delta(z) = d(z) + Kn(z)$.

The stability of the closed-loop characteristic polynomial can be determined by applying Jury's stability test. Alternatively, BLT can be used to determine stability through the application of Routh's test to the transformed polynomial, $\Delta(w)$.

Example 7.4.1

Let $G(s) = \frac{1}{s+1}$, $T = 0.2$s; then, the pulse transfer function is given as: $G(z) = \frac{0.181}{z-0.819}$.

The closed-loop characteristic polynomial is: $\Delta(z) = z - 0.819 + 0.181K$.

The polynomial is stable for $|0.819 + 0.181K| < 1$, or $-1 < K < 10$ for stability.

Example 7.4.2

Let $\Delta(z) = z^2 + z + K$; then, $\Delta(w) = \Delta(z)|_{z=\frac{1+w}{1-w}} = Kw^2 + 2(1-K)w + 1 + K$.

The application of the Hurwitz criteria to $\Delta(w)$ reveals $0 < K < 1$ for stability.

Example 7.4.3

Let $G(s) = \frac{1}{s(s+1)}$, $T = 0.2s$; then, the pulse transfer function is given as: $G(z) = \frac{0.0187z+0.0175}{(z-1)(z-0.181)}$.

The characteristic polynomial is:

$$\Delta(z) = z^2 + (0.0187K - 1.819)z + 0.0175K + 0.819.$$

The w-polynomial obtained through BLT is given as:

$$\Delta(w) = (3.637 - 0.0012K)w^2 + (0.363 - 0.035K)w + 0.036K$$

The application of the Hurwitz criteria to $\Delta(w)$ gives $0 < K < 10.34$ for stability.

7.5: Closed-Loop System Response

Closed-Loop System Step Response

We consider a unity-gain feedback sampled-data control system (Figure 7.1), where an analog plant is driven by a digital controller through a ZOH.

Let the pulse transfer function be given as: $G(z)=\frac{n(z)}{d(z)}$

Then, for a static controller, the closed-loop pulse transfer function in unity-feedback configuration is defined as:

$$T(z)=\frac{Kn(z)}{d(z)+Kn(z)}.$$

Given the closed-loop pulse transfer function, $T(z)$, we compute its response to a unit-step sequence, $r(kT)=\{1,1,\ldots\}$.

The response can be obtained by iteration, or analytically from the expression: $y(z)=T(z)r(z)$.

Example 7.5.1

Let $G(s)=\frac{1}{s+1}$, $T=0.2$ s; then, we have $G(z)=\frac{0.181}{z-0.819}$

The closed-loop pulse transfer function is given as: $T(z)=\frac{0.6181K}{z-0.819+0.181K}$.

Let $K=1$, and assume $r(kT)=1,\quad r(z)=\frac{1}{1-z^{-1}}$; then, system response is obtained as: $y(z)=\frac{0.181z}{(z-1)(z-0.638)}$.

Using PFE, $y(z)=0.5\left(\frac{z}{z-1}-\frac{z}{z-0.638}\right)$.

The resulting output sequence is given as: $y(kT)=0.5(1-0.638^k),\quad k=0,1,\ldots$

or $y(k)=\{0,\ \ 0.632,\ 0.465,\ 0.509,\ 0.498,\ 0.501,\ 0.5,\ 0.5\ldots\}$.

For comparison, for $K=1$, the analog system response is given as: $y(t)=\frac{1}{2}(1-e^{-2t}),\quad t>0$.

The step responses are compared in Figure 7.5.1.

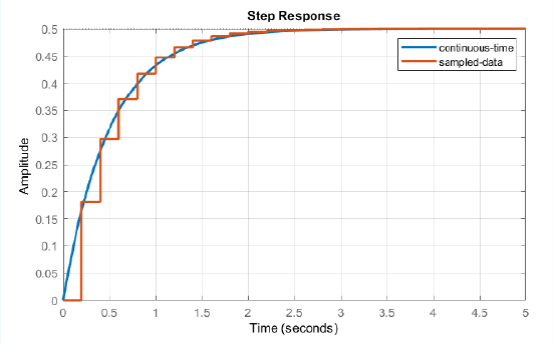


Figure 7.5.1: Step responses of the continuous-time and sampled-data systems for Example7.5.1.

Example 7.5.2

Let $G(s)=\frac{1}{s(s+1)}$, $T=0.2$s; then, we obtain: $G(z)=\frac{0.0187z+0.0175}{(z-1)(z-0.819)}$.

The closed-loop pulse transfer function is: $T(z)=\frac{(0.0187z+0.0175)K}{(z-1)(z-0.819)+(0.0187z+0.0175)K}$

Let $K=1;$ then, the closed-loop pulse transfer function is: $T(z)=\frac{0.0187z+0.0175}{z^2-1.8z+0.836}$.

Let $r(z)=\frac{z}{z-1}$ (unit step); then, the closed-loop system response is given as: $y(z)=\frac{z(0.0187z+0.0175)}{(z-1)(z^2-1.8z+0.836)}$.

By using long division, we obtain: $y(z)=0.368z^{-1}+z^{-2}+1.4z^{-3}+1.4z^{-4}+1.15z^{-5}+\ldots$

The step response sequence is given as: $y(kT)=\{0,\ 0.368,\ 1,\ 1.4,\ 1.4,\ 1.15,\ \ldots\}$

For comparison, the analog system has a closed-loop transfer function: $T(s)=\frac{1}{s^2+s+1}$.

Its unit-step response is obtained as: $y(t)=(1-1.15e^{-0.5t}\sin 0.866\,t)u(t)$.

The step responses are compared in Figure 7.5.2.

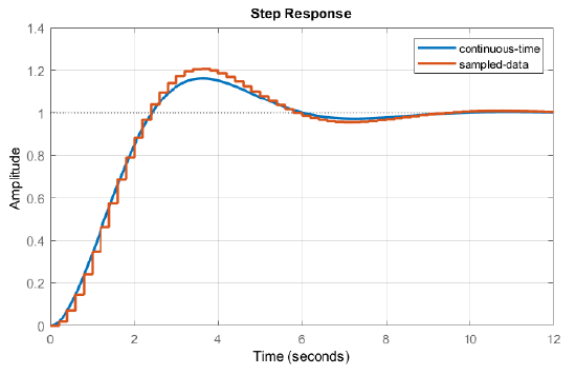


Figure 7.5.2: Step responses of the continuous-time and sampled-data systems.

From the comparison of step responses, we observe that the analog system response has a 16.3% overshoot, whereas the discrete system response has a higher (18%) overshoot. The discrete system also has a higher settling time compared to the analog system.

The higher overshoot in the case of sampled-data system occurs due to the negative phase contributed by the ZOH that reduces the available phase margin by an amount: $\Delta\phi_{\mathrm{m}}=\frac{\omega_{\mathrm{gc}}\mathrm{T}}{2}$.

In particular, from the MATLAB 'margin' command, the continuous-time system has a PM of 52° at $\omega_{\mathrm{gc}}=0.786\ \frac{\mathrm{rad}}{\mathrm{s}}$. Since $T=0.2$ sec, the reduction is phase margin is: $\Delta\phi_{\mathrm{m}}=4.5^\circ$.

Steady-State Tracking Error

The tracking error, $e(z)$, in response to a given reference input $r(z)$, in the case of a unity-gain feedback sampled-data system is computed as:

$$e(z)=r(z)(1-T(z)).$$

The final value theorem (FVT) in the z-domain is stated as:

$$\lim_{k\to\infty}e(k)\ =\lim_{z\to 1}(z-1)e(z).$$

The position and velocity error constants in the case of sampled-data system are defined as:

$$K_{\mathrm{p}}=\lim_{\mathrm{z}\to 1}\mathrm{G(z)}$$

$$K_{\mathrm{v}}=\lim_{z\to 1}\frac{(z-1)}{T}G(z)$$

Using the error constants, the steady-state errors to step and ramp inputs are computed as:

$$e_{\mathrm{ss}}|_{\mathrm{step}}=\frac{1}{1+K_{\mathrm{p}}}$$

$$e_{\mathrm{ss}}|_{\mathrm{ramp}}=\frac{1}{K_{\mathrm{v}}}.$$

Example 7.5.3

Let $G(z)=\frac{0.0187z+0.0175}{(z-1)(z-0.819)}$ $(T=0.2s)$; then, the error constants are given as $K_{\mathrm{p}}=\infty$ and $K_{\mathrm{v}}=1$.

Accordingly, $e_{\mathrm{ss}}|_{\mathrm{step}}=0;\ e_{\mathrm{ss}}|_{\mathrm{ramp}}=0.2$.

Example 7.5.4

Let $G(z)=\frac{0.368z+0.264}{(z-1)(z-0.368)}$ with $T=1\text{s}$; then, we have $K_\text{p}=\infty,\ \ \text{K}_\text{v}=1$.

Accordingly, $e_\text{ss}|_\text{step}=0;\ e_\text{ss}|_\text{ramp}=1$.

7.6: Digital Controller Design by Emulation

Emulation of Analog Controllers

A controller for a sampled-data control system can be obtained by emulating (i.e., approximating) an available analog controller design. Controller emulation aims to obtain an approximate digital controller, $K(z)$, whose response matches that of the analog controller, $K(s)$ as measured by a selected metric.

Popular controller emulation methods include:

Impulse Invariance. The impulse invariance method of digital controller design aims to match the impulse response of the analog controller. Assume that the analog controller is of the form: $K(s)=\sum_{k=1}^{n}\frac{A_k}{s-s_k}$. Then, a corresponding digital controller is obtained as: $K(z)=T\sum_{k=1}^{n}\frac{A_k}{1-p_k z^{-1}}, \quad p_k=e^{s_k T},$ that is, the controller poles are mapped to their respective locations in the z-plane.

Pole–Zero Matching. In pole–zero matching, both the plant poles and zeros are mapped to their equivalent locations in the z-plane using: $z_i=e^{s_i T}$. If the analog controller transfer function has n poles and m finite zeros, to be matched, then another $n-m$ zeros are added at $z=1$. Additionally, the dc gain of the digital controller is matched to that of the analog controller.

Zero-Order Hold (ZOH). The ZOH method assumes the presence of a ZOH at the input to the controller. The method is ineffective if the analog controller has poles at the origin (as in the case of PI and PID controllers).

First-Order Hold (FOH). The FOH method assumes a piece-wise linear input to the controller.

Bilinear Transform (BLT). The BLT method uses Tustin's approximation $\left(z\cong\frac{1+Ts/2}{1-Ts/2}\right)$ to emulate an analog controller. The BLT method is quite effective when used with a high sampling frequency. Using the BLT, an equivalent digital controller is obtained as: $K(z)=K(s)|_{s=\frac{2}{T}\frac{z-1}{z+1}}$.

The 'c2d' command in the MATLAB Control Systems Toolbox allows controller emulation method to be specified. The default method is ZOH.

Example 7.6.1

Let $G(s)=\frac{10}{s(s+2)(s+5)}$, $T=0.2s$; then, the pulse transfer function is obtained as: $G(z)=\frac{0.0095(z+0.18)(z+2.68)}{(z-1)(z-0.67)(z-0.37)}$

A lead-lag controller for this example was designed earlier: $K(s)=25\left(\frac{s+2}{s+24}\right)\left(\frac{s+0.05}{s+0.004}\right)$.

We use MATLAB Control System Toolbox 'c2d' commands to obtain discrete approximations of the controller using the following methods:

ZOH: $K(z)=\frac{25(z-0.99)(z-0.925)}{(z-0.999)(z-0.008)}$

FOH: $K(z)=\frac{6.86(z-0.99)(z-0.7)}{(z-0.999)(z-0.008)}$

BLT: $K(z)=\frac{8.86(z-0.99)(z-0.667)}{(z-0.999)(z-0.412)}$

Impulse-invariance: $K(z)=\frac{-109.77z(z-0.999)}{(z-0.999)(z-0.008)}$

Pole-zero matching: $K(z)=\frac{6.3(z-0.99)(z-0.67)}{(z-0.999)(z-0.008)}$

Least-squares: $K(z)=\frac{5.74(z-0.99)(z-0.542)}{(z-0.999)(z-0.376)}$

An analysis of the closed-loop systems shows that all except the impulse invariance approximation are stable with damping in the range of $\zeta\in[0.72,0.81]$ The closed-loop responses are compared in Figure 7.6.1.

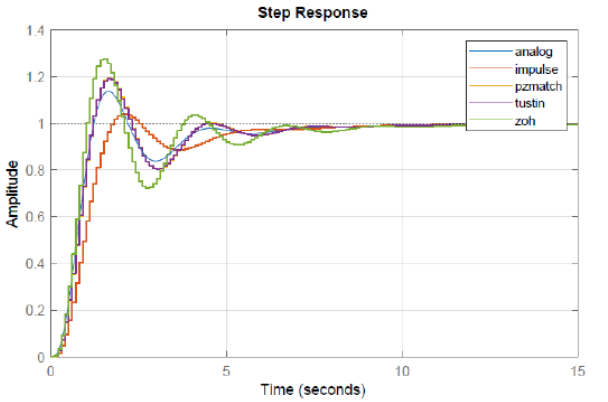


Figure 7.6.1: Step response comparison for analog controller emulation methods.

As seen from the figure and otherwise, the BLT method generally provides the best approximation of an analog controller.

Emulation of Analog PID Controller

An analog PID controller with input $e(t)$ and output $u(t)$ is described as:

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} + k_i \int e(t)dt.$$

An equivalent digital PID controller can be obtained by employing a forward Euler approximation of the derivative term: $\frac{d}{dt}e\left(t\right) \approx \frac{1}{T}\left(e_{k+1} - e_k\right)$.

The resulting z-domain transfer functions for the PID controller are given as:

$$K\left(z\right) = k_p + k_d\left(\frac{z-1}{T}\right) + k_i\left(\frac{T}{z-1}\right)$$

Further, let v_k denote the integrator output; then, the digital PID controller is implemented using the following update rules:

$$u_k = k_p e_{k-1} + \frac{1}{T}\left(e_k - e_{k-1}\right) + k_i v_k$$

$$v_k = v_{k-1} + T e_{k-1}$$

Example 7.6.2

Let $G\left(s\right) = \frac{10}{s(s+2)(s+5)}$, $T = 0.2s$; then, the pulse transfer function is obtained as: $G\left(z\right) = \frac{0.0095(z+0.18)(z+2.68)}{(z-1)(z-0.67)(z-0.37)}$

A PID controller for this example was designed earlier as: $K_{PID}\left(s\right) = \frac{1.2(s+0.05)(s+2)}{s}$.

An approximate digital controller is obtained as: $K_{PID}\left(z\right) = 2.46 + 1.2\left(\frac{z-1}{T}\right) + 0.12\left(\frac{T}{z-1}\right)$.

The step response of the closed-loop system for analog and digital PID controllers is shown in Figure 7.6.2. The discrete PID controller with forward Euler approximation of derivative emulates the analog controller well. However, pole-zero matching is seen to give the best result in this example.

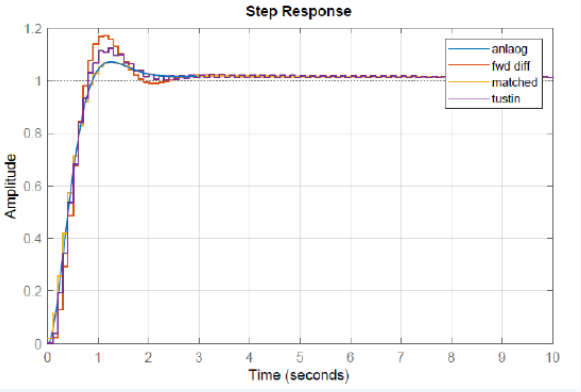


Figure 7.6.2: A comparison of the step response for analog and digital PID controllers.

7.7: Root Locus Design of Digital Controllers

Digital Controller Design

The root locus technique (Chapter 5) employs the open-loop transfer function, $KGH(s)$, to describe the locus of the roots of the closed-loop characteristic polynomial: $\Delta(s) = 1 + KGH(s)$, with variation in the controller gain, K.

The z-plane root locus similarly describes the locus of the roots of closed-loop pulse characteristic polynomial, $\Delta(z) = 1 + KG(z)$, as controller gain K is varied. A suitable value of K can then be selected form the RL plot.

To ensure closed-loop stability, the closed-loop roots should be confined to inside the unit circle.

In the MATLAB Control Systems Toolbox, 'rlocus' command is used to plot both the s-plane and z-plane root loci. The RL design of digital controller is described below.

Design for a Desired Damping Ratio

Assuming that the controller design specifications include a desired damping ratio, ζ, of the closed-loop poles, we consider the z-plane root locus design based on the ζ requirement.

For a prototype second-order transfer function: $T(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$, the constant ζ lines are defined by: $\frac{\sigma}{\omega} = \mp\frac{\zeta}{\sqrt{1-\zeta^2}}$. The constant ζ lines in the z-plane are obtained by: $z = e^{Ts} = e^{\sigma T}e^{\pm j\omega T}$, where $\sigma = \mp\frac{\zeta}{\sqrt{1-\zeta^2}}\omega$ and ωT ranges from 0 to π.

In the MATLAB Control Systems Toolbox, the 'grid' command displays the constant ζ lines in the complex z-plane. The grid command additionally displays constant frequency (ω_n) contours that are helpful in selecting a suitable sampling time (T) given the natural frequency of the model. The desired region in the z-plane for closed-loop root locations may be specified as: $0.1\pi \le \omega_n T \le 0.5\pi, \quad \zeta \ge 0.6$, which assumes a sampling frequency between $2\ and 10$ times the natural frequency of the analog system.

Example 7.7.1

Let $G(s) = \frac{1}{s(s+1)}$, $T = 0.2$s; then, we have: $G(z) = \frac{0.0187z+0.0175}{(z-1)(z-0.819)}$. Assume that the design specifications call for $\zeta = 0.7$.

The closed-loop pulse characteristic polynomial is obtained as: $\Delta(z) = z^2 + (0.0187\,K - 1.819)z + 0.0175\,K + 0.819$.

The z-plane root locus is plotted using MATLAB 'rlocus' command and is shown in Fig. 7.7.1. From the z-plane RL plot (Figure 7.7.1), we can choose, e.g., $K = 0.46$, to have $\zeta = 0.7$ for the closed-loop roots.

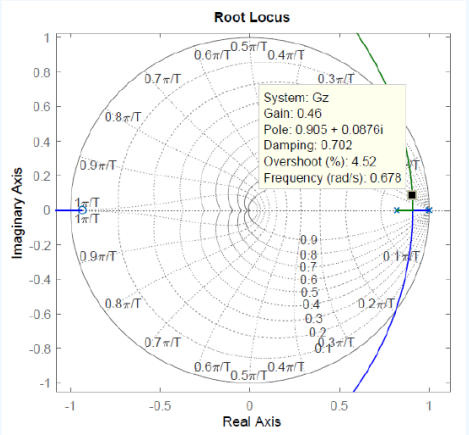


Figure 7.7.1: Root locus design in the z-plane.

The resulting pulse characteristic polynomial is: $\Delta(z) = z^2 - 1.81z + 0.827,$ with closed-loop roots located at: $z = 0.9 \pm j0.09$. The use of MALTAB 'damp' command shows a damping ratio of $\zeta = 0.7$ with a natural frequency $\omega_n = 0.68$ rad/s.

For comparison, the continuous-time system has a characteristic polynomial: $\Delta(s) = s^2 + s + K,$; for $K = 0.46$, the closed-loop roots are located at $s = 0.5 \pm j0.46$ with $\zeta = 0.74$ with $\omega_n = 0.68$ rad/s.

The step responses of analog and discrete systems are compared in Figure 7.7.2.

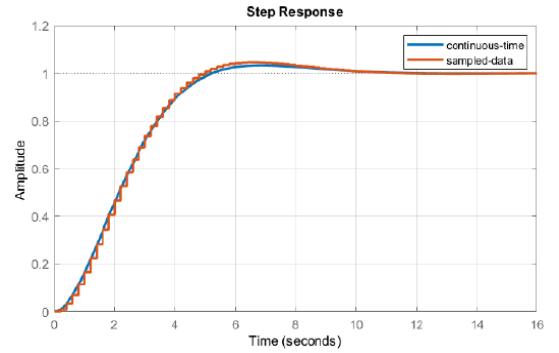


Figure 7.7.2: Step response of the closed-loop continuous-time and sampled-data systems.

Design for Settling Time and Damping Ratio

The settling time and the damping ratio of the dominant closed-loop roots in the z-plane are related as: $re^{\pm j\theta} = e^{-\zeta\omega_n T} e^{\pm j\omega_d T}$, $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

By separating into real and imaginary part gives two equations: $\ln r = -\zeta\omega_n T$ and $\theta = \omega_d T$, which can be solved to obtain:

$$\zeta = -\ln r / \sqrt{\ln^2 r + \theta^2}$$

$$\omega_n = \sqrt{\ln^2 r + \theta^2}/T$$

$$\tau = \frac{1}{\zeta\omega_n} = -\frac{T}{\ln r}.$$

Example 7.7.2

Let $G(z) = \frac{0.0187z+0.0175}{(z-1)(z-0.819)}$. The closed-loop pulse characteristic polynomial is obtained as: $\Delta(z) = z^2 + (0.0187\,K - 1.819)z + 0.0175\,K + 0.819.$

For $\zeta = 0.7$, the z-plane closed-loop roots are located at: $z = 0.905 \pm j0.0875 = 0.91\ e^{\pm j0.096}$. Then, from the above relations, we obtain: $\zeta = 0.7,\ \omega_n = 0.74,\ \tau = 2.1$ s, $t_s = 4.6\,\tau = 9.5$ s.

A Comparison of Analog and Digital Controllers

The following example compares the root locus design of analog and digital controllers in the case of a DC motor model.

Example 7.7.3

A DC motor model is described as: $G(s) = \frac{500}{s^2+110s+1025}$; the design specifications are for the motor step response to have: $OS \le 10\%\ (\zeta \ge 0.59),\ t_s \le 100ms,\ e(\infty)\big|_{step} = 0.$

The sampling time is selected as: $T = 0.01s$; then, from the MATLAB 'c2d' command, the motor pulse transfer function is obtained as: $G(z) = \frac{0.0178z+0.0123}{z^2-1.27z+0.333}$.

In order to obtain zero steady-state error, we may choose a PI controller, where the controller zero approximately cancels a plant pole, i.e., let $K(s) = \frac{K(s+10)}{s}$.

A comparable PI controller for the sampled-data system is obtained by using the transformation: $z = e^{Ts}$, and is given as: $K(z) = \frac{K(z-0.905)}{z-1}$.

The s-plane and z-plane root locus plots are shown in Figure 7.7.3. From the plots, we may choose, e.g., $K = 8$ for the closed-loop system to achieve a time constant: $\tau \cong 25ms$.

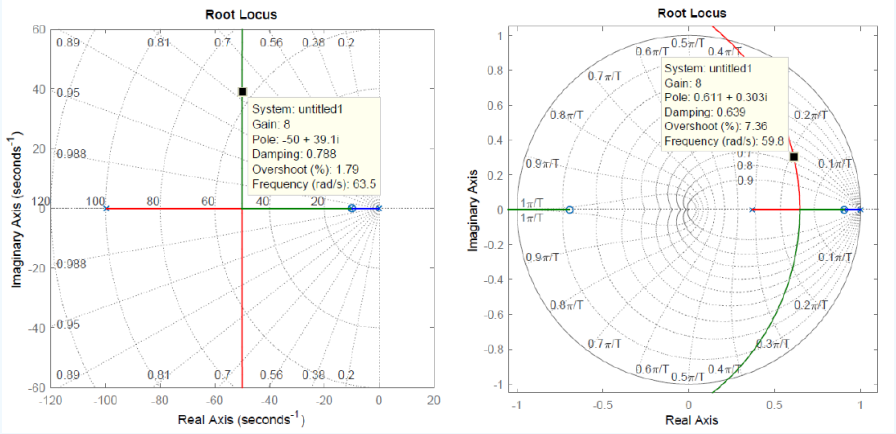


Figure 7.7.3: Root locus plot for the DC motor model: analog system design (left); sampled-data system design (right).

For $K = 8$, the resulting closed-loop transfer functions for analog and discrete system are obtained as:

Analog: $T(s) = \frac{4000(s+10)}{s^3+110s^2+5025s+40{,}000}$

Discrete: $T(z) = \frac{0.14z^2-0.03z-0.089}{z^3-2.13z^2+1.57z-0.42}$

The use of the MATLAB 'damp' command shows a damping of $\zeta = 0.79$ for the analog system and a damping of $\zeta = 0.68$ for the digital system.

The step responses of the analog and discrete systems are compared in Figure 7.7.4. As seen from the figure, both analog and discrete systems meet the settling time and overshoot requirements.

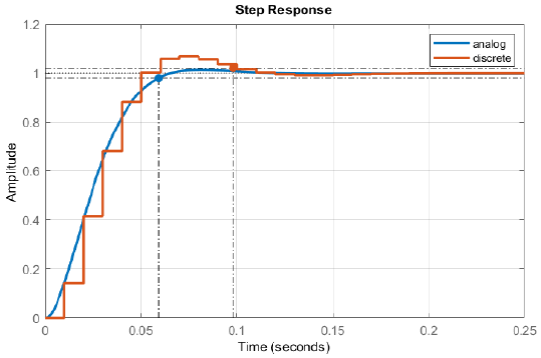


Figure 7.7.4: Step response of the DC motor for the analog and discrete PI controller design.

MATLAB Tuned PID Controller Design

The MATLAB Control Systems Toolbox provide 'pidtune' command that can be used with both analog and discrete system models. The tuned PID controller design in the case of a DC motor is explored in the next example.

Example 7.7.4

Let $G(s) = \frac{500}{s^2+110s+1025}$; the design specifications are for the motor step response to have: $OS \le 10\%\ (\zeta \ge 0.59),\ t_s \le 100ms,\ e(\infty)|_{step} = 0.$

Let $T = 0.01s$; from the MATLAB 'c2d' command, the motor pulse transfer function is obtained as: $G(z) = \frac{0.0178z+0.0123}{z^2-1.27z+0.333}$.

By using the MATLAB 'pidtune' command, an analog PID controller for the DC motor model with a crossover frequency of $100rad/s$ is obtained as: $K(s) = 25.1 + 0.189s + \frac{567}{s}$.

The 'pidtune' command is used to design a similar PID controller for the discrete model. The result is: $K(z) = 23.4 + 0.368\left(\frac{z-1}{T}\right) + 202\left(\frac{T}{z-1}\right)$.

The unit-step responses of the closed-loop systems for the analog and discrete models are compared in Figure 7.7.5. Both controller achieve a settling time $t_s < 0.1s$ with no steady-state error. However, in this case, the tuned digital PID controller has a lower settling time.

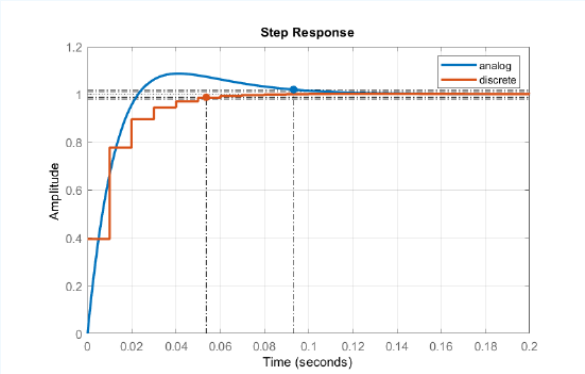


Figure 7.7.5: Step response of the DC motor for the analog and discrete PID controller design.

CHAPTER OVERVIEW

8: State Variable Models

Learning Objectives

1. Analyze the state variable models of single-input single-output dynamic systems.
2. Solve state equations in time-domain using the state-transition matrix.
3. Obtain the modal matrix and determine the stability of the model.
4. Obtain state-space realization of a transfer function model in alternate forms.

8.0: Prelude to State Variable Models

This chapter discusses algebraic methods to analyze the state variable models of dynamic systems. These models comprise a set of first-order linear differential equations that describe the system behavior using system state variables. A common choice includes the natural variables associated with the energy storage elements present in the system. Alternate variables in equal number can also be selected.

The state equations include a set of coupled first-order ODE's that describe the behavior of the system. The state equations can be collectively integrated using a matrix exponential, i.e., the state-transition matrix, as an integrating factor. The state-transition matrix contains natural modes of system response and plays a fundamental role in the evolution of system trajectories. The general solution to the homogeneous state equations is a weighted sum of the columns of the state-transition matrix.

Given a state variable model of the system, a transfer function representation can be obtained by applying the Laplace transform. Generally, the degree of the denominator polynomial in the transfer function model equals the number of state variables used to represent it. A pole-zero cancellation in the transfer function would, however, cause some of the response modes to be absent from the input-output system description.

A given transfer function model can be realized into a state variable model in multiple ways. Different choices of state variables accord different structures to the system matrix. Specific realization structures may be preferred for ease of computing the system response or determining its stability characteristics.

The modal realization reveals the natural modes of system response. The controller realization facilitates the controller design using state variable methods. The diagonal representation decouples the system variables into a set of independent first-order ordinary differential equations (ODEs) that can be easily integrated. The state variable vectors for these alternate models are related through bilinear transformations.

This chapter describes the analysis techniques for state variable models of the continuous-time systems. The discrete-time system models are covered later in Chapter 8. The analysis is restricted to the single-input single-output (SISO) systems, that exhibit a rational transfer function, $G(s)$. The methods, however, can be generalized to include multi-input multi-output (MIMO) systems.

8.1: State Variable Models

The State Equations

The state variable model of a dynamic system comprises first-order ODEs that describe time derivatives of a set of state variables. The number of state variables represents the order of the system, which is assumed to match the degree of the denominator polynomial in its transfer function description.

The natural variables associated with the energy storing elements are commonly used as state variables, though alternate variables can be selected. The natural variables include, for example, capacitor voltage and inductor current in the electrical networks, and position and velocity of the inertial mass in the mechanical systems.

Let $\mathbf{x}(t)$ describe a vector of state variables, $u(t)$ describe a scalar input, and $y(t)$ describe a scalar output; then the state variable model of a linear time-invariant (LTI) single-input single-output (SISO) system is written in its generic form as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t)$$

$$y(t) = \mathbf{c}^T\mathbf{x}(t) + du(t)$$

In the above, $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{b}$ is a column vector that distributes the inputs, $\mathbf{c}^T$ is a row vector that combines the state variables to form the output, and d is a scalar feedforward term contributing to the output.

The state variable model of a multi-input multi-output (MIMO) system with m inputs and p outputs is described by the state and output equations given as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}^T\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

In the above, the variable dimensions are: $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{u} \in \mathbf{R}^m$, and $\mathbf{y} \in \mathbf{R}^p$; the matrices have the following dimensions: $\mathbf{A} \in \mathbf{R}^{n\times n}$, $\mathbf{B} \in \mathbf{R}^{n\times m}$, $\mathbf{C} \in \mathbf{R}^{p\times n}$, and $\mathbf{D} \in \mathbf{R}^{p\times m}$.

In the following, we restrict our attention to models of SISO systems. Unless noted otherwise, $d = 0$ is assumed, which corresponds to a strictly proper transfer function description of the system.

Solution to the State Equations

In order to explore a time-domain solution to the state equations, we begin with a scalar differential equation:

$$\dot{x}(t) = ax(t) + bu(t).$$

Using an integrating factor, e^{-at}, the equation is written as total differential:

$$\frac{d}{dt}(e^{-at}x(t)) = e^{-at}bu(t)$$

Next, assuming an initial conditions: $x\,(0) = x_0$, we integrate from $\tau = 0$ to $\tau = t$ to obtain:

$$e^{-at}x(t) - x_0 = \int_0^\tau e^{-a\tau}bu(\tau)d\tau$$

Hence, the solution to the scalar ODE in terms of $x(t)$ is obtained as:

$$x(t) = e^{at}x_0 + \int_0^\tau e^{a(t-\tau)}bu(\tau)d\tau$$

Further, the solution to homogeneous equation: $\dot{x}\,(t) = ax\,(t)$ is given as: $x\,(t) = e^{at}x_0$.

Next, we explore the possibility to generalize this solution to the matrix case. For this purpose, we define a matrix exponential function as:

$$e^{\mathbf{A}t} = \sum_{i=0}^{\infty}\frac{\mathbf{A}^i t^i}{i!} = \mathbf{I} + \mathbf{A}t + \ldots$$

The infinite series converges in the case of systems that are well-behaved. Further, the matrix exponential obeys the matrix differential equation:

$$\frac{d}{dt}\left(e^{\mathbf{A}t}\right) = \mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A}$$

Using the matrix exponential, the solution to the matrix differential equation, $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t)$, is written as:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^\tau e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau)d\tau$$

The above solution has two parts: the first term describes the system response to initial conditions $\mathbf{x}_0$, while the second term describes the system response to an input, $u(t)$.

Laplace Transform Solution

Consider the state equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \mathbf{x}\left(0\right) = \mathbf{x}_0;$$

Apply the Laplace transform to obtain:

$$s\mathbf{x}(s) - \mathbf{x}_0 = \mathbf{A}\mathbf{x}(s) + \mathbf{b}u(s).$$

The state variable vector is solved as:

$$\mathbf{x}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}_0 + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}u(s),$$

where $\mathbf{I}$ denote an $n \times n$ identity matrix.

The $n \times n$ matrix $(s\mathbf{I} - \mathbf{A})$ is called the **characteristic matrix** of $\mathbf{A}$.

Its determinant defines the **characteristic polynomial** of $\mathbf{A}$, i.e., $\Delta(s) = |s\mathbf{I} - \mathbf{A}|$.

The roots of the characteristic polynomial are the eigenvalues of $\mathbf{A}$.

The $n \times n$ matrix $(s\mathbf{I} - \mathbf{A})^{-1}$ is called the **resolvent matrix** of $\mathbf{A}$. The resolvent of $\mathbf{A}$ can be computed as an infinite series:

$$(s\mathbf{I} - \mathbf{A})^{-1} = s^{-1}(\mathbf{I} + s^{-1}\mathbf{A} + \ldots)$$

By comparing the Laplace transform solution with the time-domain solution, we arrive at the following relations:

$$\mathcal{L}[e^{\mathbf{A}t}] = (s\mathbf{I} - \mathbf{A})^{-1}$$

$$\mathcal{L}\left[\int_0^\tau e^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau)d\tau\right] = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}u(s)$$

The first equation above can be used to compute the matrix exponential:

$$e^{\mathbf{A}t} = \mathrm{L}^{-1}\left[(s\mathbf{I} - \mathbf{A})^{-1}\right]$$

The second equation is used to solve for the output response when initial conditions are zero:

$$y(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}u(s)$$

The State-Transition Matrix

Consider the homogenous system of equations: $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ and $\mathbf{x}(0) = \mathbf{x}_0$.

The solution to the homogenous state equation is given as: $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$.

The matrix exponential $e^{\mathbf{A}t}$ that relates $\mathbf{x}_0$ to $\mathbf{x}(t)$ is termed as the state-transition matrix. The state-transition matrix contains the natural modes of system response.

Transfer Function

The output response of the state variable model defines the transfer function: $y(s) = G(s)u(s)$, where

$$G(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} = \mathbf{c}^T \frac{\text{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})}\mathbf{b}$$

In the case of a MIMO system, $G(s)$ represents a $p \times m$ transfer matrix, given as:

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

Alternatively, the transfer matrix of a MIMO system can be obtained as:

$$G(s) = \frac{\det \begin{bmatrix} s\mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}}{\det(s\mathbf{I} - \mathbf{A})}$$

Assuming no pole-zero cancelations in $G(s)$, the characteristic polynomial matches the denominator polynomial.In the event of pole-zero cancelations, the order of the denominator polynomial, $d(s)$, is less than n, i.e., the zeros of $d(s)$ form a subset of the eigenvalues of $\mathbf{A}$.

Example 8.1.1

Consider the mass–spring–damper model (Example 1.9.2), where mass position, $x(t)$, and velocity, $v(t) = \dot{x}(t)$ are selected as state variables; then, the state output equations are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f, \quad x = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}$$

The characteristic matrix of the model is given as: $s\mathbf{I} - \mathbf{A} = \begin{bmatrix} s & -1 \\ \frac{k}{m} & s + \frac{b}{m} \end{bmatrix}$.

The resolvent of $\mathbf{A}$ is computed as: $(s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{\Delta(s)} \begin{bmatrix} s + b/m & 1 \\ -k/m & s \end{bmatrix}$; $\Delta(s) = s^2 + \frac{b}{m}s + \frac{k}{m}$.

The transfer function of the model is computed as:

$$G(s) = \frac{1}{\Delta(s)} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} s + b/m & 1 \\ -k/m & s \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{m} \end{bmatrix} = \frac{1}{ms^2 + bs + k}$$

Example 8.1.2

Consider the model of a DC motor (Example 1.9.3), where armature current, $i_a(t)$, and motor speed, $\omega(t)$, are selected as state variables; the state and output equations of the DC motor are given as:

$$\frac{d}{dt} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -R/L & -k_b/L \\ k_t/J & -b/J \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

Assume that the following parameter values be assumed: $R = 1\Omega$, $L = 1\ mH$, $J = 0.01\ kg \cdot m^2$, $b = 0.1\ \frac{N \cdot s}{rad}$, $k_t = k_b = 0.05$. Then, the state variable model of the motor is given as:

$$\frac{d}{dt} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

The resolvent matrix and the motor transfer function is obtained as follows:

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{\Delta(s)} \begin{bmatrix} s + 10 & -5 \\ 5 & s + 100 \end{bmatrix}; \quad \Delta(s) = s^2 + 110s + 1025$$

$$G(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} = \frac{500}{s^2 + 110s + 1025}$$

Impulse Response

Consider the state equation:

$$\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{b}u(t),\quad y(t)=\mathbf{c}^T\mathbf{x}$$

For $\mathbf{x}_0=\mathbf{0}$, the input response of the system is obtained as:

$$y(s)=G(s)u(s)=\mathbf{c}^T(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}u(s)$$

Let $u(t)=\delta(t)$, $u(s)=1$; then, the system response is given as:

$$y_{imp}(s)=G(s)=\mathbf{c}^T(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}$$

The impulse response in the time-domain is obtained as:

$$g(t)=\mathcal{L}^{-1}[G(s)]=\mathbf{c}^Te^{\mathbf{A}t}\mathbf{b}$$

In terms of the impulse response, the system output is given by the convolution integral:

$$y(t)=\int_0^t g(t-\tau)u(\tau)\,d\tau$$

Step Response

Let $u(t)=1(t)$, $u(s)=\frac{1}{s}$. Then, we have:

$$s\mathbf{x}(s)=\mathbf{c}^T(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}$$

The unit-step response in time-domain is obtained as:

$$y_{step}(t)=\mathbf{c}^T\left(\int_0^t e^{\mathbf{A}(t-\tau)}d\tau\right)\mathbf{b}$$

Alternatively, the step response represents the integral of the impulse response:

$$y_{step}(t)=\int_0^t g(t-\tau)\,d\tau$$

? Exercise 8.1.3

The state and output equations for the model of a dc motor are:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix}i_a\\ \omega\end{bmatrix}=\begin{bmatrix}-100 & -5\\ 5 & -10\end{bmatrix}\begin{bmatrix}i_a\\ \omega\end{bmatrix}+\begin{bmatrix}100\\ 0\end{bmatrix}V_a,\omega=\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}i_a\\ \omega\end{bmatrix}$$

The state-transition matrix for the dc motor model is obtained as:

$$e^{\mathbf{A}t}=\begin{bmatrix}1.003 & 0.056\\ -0.056 & -0.003\end{bmatrix}e^{-99.72t}+\begin{bmatrix}-0.003 & -0.056\\ 0.056 & 1.0003\end{bmatrix}e^{-10.28t}$$

The impulse response of the DC motor is computed as:

$$g(t)=\mathbf{c}^Te^{\mathbf{A}t}\mathbf{b}=5.6\left(e^{-99.7t}-e^{-10.3t}\right)$$

where $\left\{e^{-99.72t},\ e^{-10.28t}\right\}$ represent the system's natural response modes that correspond to the electrical and mechanical time constants of the DC motor: $\tau_{\mathrm{e}}\cong 0.01s,\ \tau_{\mathrm{m}}\cong 0.1\mathrm{s}$.

Assuming $V_a(t)=u(t)$, the step response of the DC motor is computed as:

$$y(t)=\int_0^t g(t-\tau)\,d\tau=0.488+0.056e^{-99.7t}-0.544e^{-10.3t}$$

The impulse and the step responses are plotted below (Figures 8.1.1).

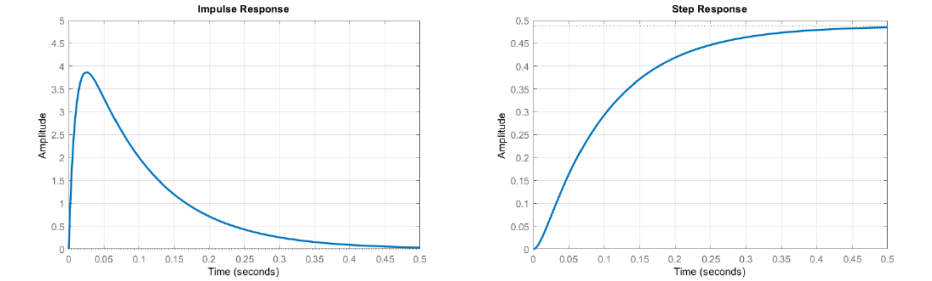


Figure 8.1.1: The DC motor model: impulse response (left); step response (right).

8.2: State-Transition Matrix and Asymptotic Stability

The State-Transition Matrix

Consider the homogenous state equation: $\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t),\ \ \mathbf{x}(0)=\mathbf{x}_0$.

The solution to the homogenous equation is given as: $\mathbf{x}(t)=e^{\mathbf{A}t}\mathbf{x}_0$, where the state-transition matrix, $e^{\mathbf{A}t}$, describes the evolution of the state vector, $x\,(t)$.

The state-transition matrix of a linear time-invariant (LTI) system can be computed in the multiple ways including the following:

1. $e^{\mathbf{A}t}=\mathcal{L}^{-1}[(s\mathbf{I}-\mathbf{A})^{-1}]$
2. $e^{\mathbf{A}t}=\sum_0^\infty \frac{\mathbf{A}^i t^i}{i\,!}$
3. By using the modal matrix (see below)
4. By using the fundamental matrix
5. By using the Cayley–Hamilton theorem

Characteristic Polynomial of A

The characteristic polynomial of $\mathbf{A}$ is an nth order polynomial obtained as the determinant of $(s\mathbf{I}-\mathbf{A})$, i.e., $\Delta(s)=|s\mathbf{I}-\mathbf{A}|$. The roots of the characteristic polynomial are the eigenvalues of $\mathbf{A}$.

The transfer function, $G\,(s)$, is expressed as:

$$G\,(s)=\mathbf{c}^T\frac{adj\,(s\mathbf{I}-\mathbf{A})}{|s\mathbf{I}-\mathbf{A}|}\mathbf{b}=\frac{n\,(s)}{d\,(s)}$$

where $adj\,(s\mathbf{I}-\mathbf{A})$ represents the adjoint matrix of $s\mathbf{I}-\mathbf{A}$.

Assuming no pole-zero cancelations in $G\,(s)$, the characteristic polynomial matches the denominator polynomial.In the event of pole-zero cancelations, the order of the denominator polynomial, $d\,(s)$, is less than n, i.e., the zeros of $d\,(s)$ form a subset of the eigenvalues of $\mathbf{A}$.

State-Transition Matrix in MATLAB

The state-transition matrix may be obtained by using the symbolic variable 't' defined by using the 'syms' command from the MATLAB Symbolic Math Toolbox. Assuming that a symbolic variable 't' and an nxn numeric matrix $\mathbf{A}$ have been defined, the state-transition matrix can be obtained by issuing the matrix exponential command as: $\mathrm{expm}(\mathrm{t}*\mathbf{A})$.

The Modal Matrix

Consider the homogenous state equation: $\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t),\ \ \mathbf{x}(0)=\mathbf{x}_0$.

Assume that $\mathbf{A}$ has a full set of eigenvectors, that obey $(\lambda_i\mathbf{I}-\mathbf{A})\,\mathbf{v}_i=0,\ \ i=1,\ldots,n$, where λ_i denotes an eigenvalue of $\mathbf{A}$.

Let $\mathbf{M}=[\mathbf{v_1},\,\mathbf{v_2},\cdots,\mathbf{v_n}]$ denote the modal matrix containing eigenvectors of $\mathbf{A}$, $e^{\mathbf{\Lambda}t}=diag\left(\left[e^{\lambda_1 t},e^{\lambda_2 t},\cdots,e^{\lambda_n t}\right]\right)$ define a diagonal matrix of the natural response modes of $\mathbf{A}$, and $\mathbf{c}$ denote a constant vector; then, the general solution to the homogenous state equation is given as:

$$\mathbf{x}_h(t)=\mathbf{M}e^{\mathbf{\Lambda}t}\mathbf{c}$$

Assuming a set of initial conditions: $\mathbf{x}_h(0)=\mathbf{x}_0=\mathbf{M}\mathbf{c}$; we have, $\mathbf{c}=\mathbf{M}^{-1}\mathbf{x}_0$. Then, a particular solution to the homogenous system is given as:

$$\mathbf{x}_h(t)=\mathbf{M}e^{\mathbf{\Lambda}t}\mathbf{M}^{-1}\mathbf{x}_0$$

The state-transition matrix can be computed from the modal matrix as:

$$e^{\mathbf{A}t}=\mathbf{M}e^{\mathbf{\Lambda}t}\mathbf{M}^{-1}$$

Modal Matrix in MATLAB

The modal matrix is obtained by using the ‘eig’ command in MATLAB. The eigenvectors of $\mathbf{A}$ obtained from MATLAB are normalized to unity. The 'eig' command also provides a diagonal matrix of eigenvalues of $\mathbf{A}$. Given the modal matrix $\mathbf{M}$ of

eigenvectors and the diagonal matrix $\mathbf{D}$ of eigenvalues, the state-transition matrix is obtained as $\mathrm{M} * \mathrm{expm}(\mathrm{t} * \mathrm{D}) / \mathrm{M}$.

Asymptotic Stability

The asymptotic stability refers to the long-term behavior of the natural response modes of the system. These modes are also reflected in the state-transition matrix, $e^{\mathbf{A}t}$.

Consider the homogenous state equation: $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0$.

Asymptotic Stability

The homogenous state equation is said to be asymptotically stable if $\lim_{t\to\infty} \mathbf{x}(t) = 0$.

Since $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$, the homogenous state equation is asymptotically stable if $\lim_{t\to\infty} e^{\mathbf{A}t} = 0$.

Further, using modal decomposition, $e^{\mathbf{A}t} = \mathbf{M}e^{\mathbf{\Lambda}t}\mathbf{M}^{-1}$, the homogenous system is asymptotically stable if $\lim_{t\to\infty} e^{\mathbf{A}t} = 0$.

Since $e^{\mathbf{\Lambda}t} = diag\left(\left[e^{\lambda_1 t}, e^{\lambda_2 t}, \cdots, e^{\lambda_n t}\right]\right)$, the above condition implies that $Re\left[\lambda_i\right] < 0,\ i = 1, \ldots, n$, where λ_i represents a root of the characteristic polynomial: $\Delta(s) = |s\mathbf{I} - \mathbf{A}|$.

The computation of modal and state-transition matrices is illustrated separately when the characteristic polynomial, $\Delta(s)$, has real or complex roots.

Example 8.2.1: Polynomial with Real Roots

For the mass-spring-damper model, let $m = 1,\ k = 2,$ and $b = 3$; then, the characteristic polynomial is: $\Delta(s) = s^2 + 3s + 2$, which has real roots: $s_1, s_2 = -1, -2$.

The natural response modes are: $\{e^{-t}, e^{-2t}\}$.

The modal matrix of eigenvectors is obtained as: $\mathbf{M} = \begin{bmatrix} -1 & -1 \\ 1 & 2 \end{bmatrix}$. The diagonal matrix of eigenvalues is: $\mathbf{\Lambda} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}$.

Since $\mathbf{A} = \mathbf{M}\mathbf{\Lambda}\mathbf{M}^{-1}$, we have: $e^{\mathbf{A}t} = \mathbf{M}e^{\mathbf{\Lambda}t}\mathbf{M}^{-1}$, which computes as: $e^{\mathbf{A}t} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ 2e^{-2t} - 2e^{-t} & 2e^{-2t} - e^{-t} \end{bmatrix}$.

Further, since $\lim_{t\to\infty} e^{\mathbf{A}t} = 0$, the homogenous state equation is asymptotically stable.

Example 8.2.2: Polynomial with Complex Roots

For the mass-spring-damper model, , let $m = 1,\ k = 2,$ and $b = 2$; then, the characteristic polynomial is: $\Delta(s) = s^2 + 2s + 2$, which has complex roots: $s_1, s_2 = -1 \pm j1$. The natural response modes are: $\{e^{-t}\sin t, e^{-t}\cos t\}$.

The complex eigenvectors are given as: $\begin{bmatrix} 1 \\ 1 \pm j1 \end{bmatrix}$. Let $\mathbf{V} = \begin{bmatrix} 1 & 1 \\ -1 + j & -1 - j \end{bmatrix}$; then, $\mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \mathrm{diag}(s_1, s_2)$.

In this case, to avoid the complex algebra, we construct a modal matrix from the real and imaginary parts of the eigenvectors. Accordingly, let $\mathbf{M} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$.

Define $\mathbf{M}^{-1}\mathbf{A}\mathbf{M} = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} = \mathbf{\Gamma}$; then, $\mathbf{A} = \mathbf{M}\mathbf{\Gamma}\mathbf{M}^{-1}$ and $e^{\mathbf{A}t} = \mathbf{M}e^{\mathbf{\Gamma}t}\mathbf{M}^{-1}$, which computes as: $e^{\mathbf{A}t} = \begin{bmatrix} e^{-t}(\cos t + \sin t) & e^{-t}\sin t \\ -2e^{-t}\sin t & e^{-t}(\cos t - \sin t) \end{bmatrix}$.

Further, $\lim_{t\to\infty} e^{\mathbf{A}t} = 0$; hence, the homogenous state equation is asymptotically stable.

8.3: State-Space Realization of Transfer Function Models

Simulation Diagrams

We consider the problem of realizing a given transfer function model as a state variable model. The realization process is facilitated by first drawing a simulation diagram for the system.

A simulation diagram realizes an ODE model into a block diagram representation using scalar gains, integrators, summing nodes, and feedback loops. Historically, such diagrams were used to simulate dynamic system models on analog computers.

Given a transfer function model, its two common realizations are described below.

Serial Realization

A serial chain of integrators can be used to realize the output of an ODE. To illustrate this idea, we consider a general second-order system model described as:

$$\frac{y(s)}{u(s)} = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The transfer function is converted into an ODE representation by cross multiplying followed by inverse Laplace transform to obtain:

$$\ddot{y}(t) + 2\zeta\omega_n \dot{y}(t) + \omega_n^2 y(t) = Ku(t)$$

The above equation is rearranged to form the highest derivative as:

$$\ddot{y}(t) = -2\zeta\omega_n \dot{y}(t) - \omega_n^2 y(t) + Ku(t)$$

The simulation diagram first realizes the highest derivative using a summing node and then uses a series of integrators to obtain the lower order derivatives up to the output (Figure 8.3.1).

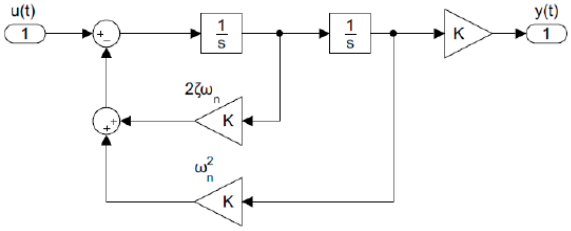


Figure 8.3.1: Serial realization of a second-order transfer function.

Parallel Realization

Consider a second-order transfer function with real roots expressed as:

$$\frac{y(s)}{u(s)} = \frac{K}{s^2 + (\sigma_1 + \sigma_2)s + \sigma_1\sigma_2}$$

The transfer function can be expanded using partial fractions expansion (PFE) to obtain:

$$y(s) = \frac{K_1}{s+\sigma_1}u(s) + \frac{K_2}{s+\sigma_2}u(s)$$

In parallel realization, the system output is expressed as: $y(s) = y_1(s) + y_2(s)$; further, the two output components are formed by ODEs that share a common input, $u(t)$:

$$\sigma_i \dot{y}_i(t) + y_i(t) = K_i u(t), \quad i = 1, 2$$

These ODEs are realized into a parallel simulation diagram (Figure 8.3.2).

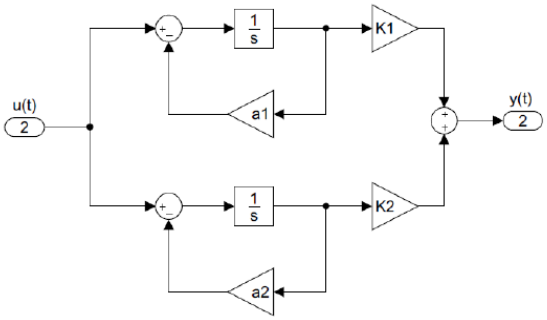


Figure 8.3.2: Parallel realization of a second-order transfer function.

Having drawn a simulation diagram, we designate the outputs of the integrators as state variables and express integrator inputs as first-order differential equations, referred as the state equations. A serial realization acquires a controller form structure; a parallel realization acquires a modal structure. These are discussed below.

Controller Form Realization

A controller form state variable structure results from a serial realization of the transfer function model. This is illustrated using a third-order transfer function model:

$$G(s) = \frac{b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3}$$

The above transfer function corresponds to the following ODE:

$$\dddot{y}(t) + a_1 \ddot{y}(t) + a_2 \dot{y}(t) + a_3 y(t) = b_1 \ddot{u}(t) + b_2 \dot{u}(t) + b_3 u(t)$$

To remove the derivatives of the input from state variables, an auxiliary variable, $v(t)$, is introduced to split the ODE as:

$$\dddot{v}(t) + a_1 \ddot{v}(t) + a_2 \dot{v}(t) + a_3 v(t) = u(t)$$

$$y(t) = b_1 \ddot{v}(t) + b_2 \dot{v}(t) + b_3 v(t)$$

The highest derivative term in the first ODE is realized using a summing node:

$$\dddot{v}(t) = -a_1 \ddot{v}(t) - a_2 \dot{v}(t) - a_3 v(t) + u(t)$$

The lower order derivative terms, $\ddot{v}(t)$ and $\dot{v}(t)$, and the output, $v(t)$, are realized using integrators.

The second ODE can be realized by summing the outputs of the integrators using coefficients as weights (Figure 8.3.3).

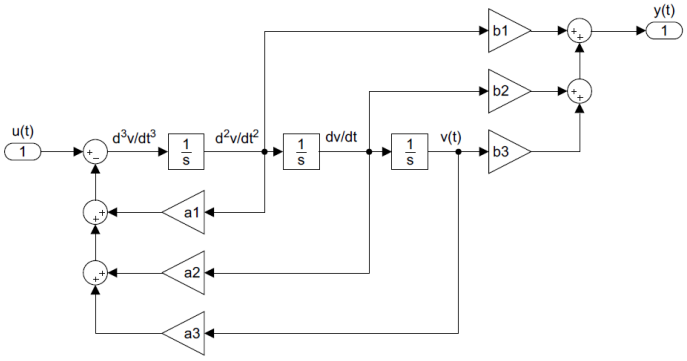


Figure 8.3.3: Simulation diagram for controller form realization of the transfer function model.

Next, the state variables are designated as: $x_1(t) = v(t),\ x_2(t) = \dot{v}(t),\ x_3(t) = \ddot{v}(t)$.

The resulting state equations are:

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = x_3,$$
$$\dot{x}_3 = -a_3 x_1 - a_2 x_2 - a_3 x_3 + u$$

The output equation is given as:

$$y = b_1 x_1 + b_2 x_2 + b_3 x_3$$

In vector–matrix form, the state variable model is given as:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Generalization to n Variables

The controller form realization is generalized to n variables as follows: let $G(s) = \frac{n(s)}{d(s)}$, where $d(s) = s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n$ represents the denominator polynomial.

By selecting the output variable and its derivatives: $y(t), \dot{y}(t), \ldots, y^{(n-1)}(t)$, as state variables, we obtain the following controller form structure:

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots \\ 0 & 0 & 1 & \ldots \\ \vdots & \vdots & \ddots & 1 \\ -a_n & -a_{n-1} & \ldots & -a_1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Let $n(s) = b_0 s^m + b_1 s^{m-1} + \cdots + b_m$ represent the numerator polynomial, where $m = n - 1$ is assumed; then, the output matrix is formed as:

$$c^T = \begin{bmatrix} b_0 & b_1 & \ldots & b_m \end{bmatrix}$$

In the above controller form structure, the coefficients of the characteristic polynomial appear in reverse order in the last row of the system matrix, whereas the output matrix contains the coefficients of the numerator polynomial.

Hence, given a proper transfer function, $G(s)$, the controller form can be written by inspection as shown in the following example.

Example 8.3.1

A mass–spring–damper system is described by transfer function model: $G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2+2s+2}$.

Its controller form state variable model is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Alternate Controller Form Structure

The integrator outputs in the simulation diagram can be alternatively numbered left to right; this reorders the state variables whereby the coefficients of the characteristic polynomial appear in the first row of the matrix. For the third-order model presented in Fig. 8.3.3, the resulting equations are given as:

$$\begin{aligned} \dot{x}_3 &= -a_1 x_1 - a_2 x_2 - a_3 x_3 + u, \\ \dot{x}_2 &= x_1 \\ \dot{x}_3 &= x_2 \end{aligned}$$

The alternate controller form is obtained by using 'ss' command after defining a transfer function in the MATLAB Control Systems Toolbox.

Example 8.3.2

A mass–spring–damper system is described by the following transfer function: $G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2+2s+2}$. The controller form state variable model is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The state variables are numbered in reverse order. Hence, x_1 represents the velocity and x_2 the position of the inertial mass.

Dual (Observer Form) Realization

Assume that a state variable realization of the transfer function model is given as:

$$\dot{x}(t) = Ax(t) + bu(t), \quad y(t) = c^T x(t)$$

Its dual realization is defined by the following state variable model:

$$\dot{z}(t) = A^T z(t) + cu(t), \quad y(t) = b^T z(t)$$

Both the original and the dual models share the same scalar transfer function, i.e.,

$$G(s) = c^T(sI - A)^{-1}b = b^T\left(sI - A^T\right)^{-1}c = G^T(s)$$

When the original state variable model is in controller form, its dual appears in the 'observer form,' named so because of its use in the design of state observers.

In the MATLAB Control Systems Toolbox, the 'canon' command with 'companion' option is used to obtain the observer form realization.

Example 8.3.3

A mass–spring–damper system is described by the following transfer function: $G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2+2s+2}$. Its controller form model was derived in Example 8.3.1. The alternate observer form state variable model is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

From the output equation, the state variable x_2 represent the position of the inertial mass. From the first state equation, $\dot{x}_1 = -2x_2 + u$, the variable x_1 includes an integral of position.

Modal Realization

A modal realization has a block diagonal structure consisting of 1×1 and 2×2 blocks that contain real and complex eigenvalues. A PFE of the transfer function is used to obtain first and second-order factors in the transfer function model.

The denominator of a second-order factor, expressed as $(s+\sigma)^2 + \omega^2$, can be realized as a 2×2 block containing the real and imaginary parts of the eigenvalue as: $A_i = \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix}$. Alternately, a second-order factor can be realized in the serial form, as illustrated in the following example.

In the MATLAB Control Systems Toolbox, a modal realization is obtained by using the 'canon' command with the 'modal' option.

Example 8.3.4

Consider the following third-order model with one real and two complex poles:

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

Its modal realization contains a block diagonal matrix with 1×1 and 2×2 blocks.

A serial realization of the second-order term results in the following modal realization:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 3 & -2 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Alternately, the modal realization includes the real and imaginary parts of the complex eigenvalue. The resulting state variable model is given as:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 3 & -1 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

We can verify that the two modal realizations share the same transfer function.

Diagonalization and Decoupling

When the denominator polynomial in the transfer function model has real and distinct roots, its modal matrix is a diagonal matrix with eigenvalues on the main diagonal. The resulting state equations describe a set of decoupled first-order ODEs that can be easily integrated.

Example 8.3.5

The transfer function model of a small DC motor, given as: $G(s) = \frac{500}{s^2+10s+1025}$.

A PFE of the transfer function gives: $G(s) = 5.59\left[\frac{1}{s+10.28} - \frac{1}{s+99.72}\right]$.

A modal realization of the transfer function model is obtained as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -99.72 & 0 \\ 0 & -10.28 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} V_{\mathrm{a}}, \quad \omega = \begin{bmatrix} -5.59 & 5.59 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The decoupled state and output equations representing the DC motor model are given as:

$$\dot{x}_1(t) = -99.72x_1(t) + V_a$$

$$\dot{x}_2(t) = -10.28x_2(t) + V_a$$

$$\omega(t) = 5.59\left(x_2(t) - x_1(t)\right)$$

We may note that the state variables in diagonal realization represent linear combinations of the original state variables. This topic is discussed in the next section.

8.4: Linear Transformation of State Variables

Linear State Transformation

This section describes a general procedure to transform a state variable model into an alternate model using state variables that are linear combinations of the original variables.

Consider the general state variable model of a SISO system, described as:

$$\dot{x}(t) = Ax(t) + bu(t)$$

$$y(t) = c^T x(t)$$

Define a bilinear transformation of the state variable vector, $x(t)$, by multiplying with a constant invertible matrix P, resulting in a new set of state variables, $z(t)$:

$$z = Px, \quad x = P^{-1}z$$

Substitute the above relations in the state and output equations:

$$P^{-1}\dot{z} = AP^{-1}z + bu, \quad y = c^T P^{-1}z$$

Multiplying on the left by P results in a new state variable model:

$$\dot{z} = PAP^{-1}z + Pbu, \quad y = c^T P^{-1}z$$

The two models share the same transfer function, i.e.,

$$G(s) = c^T P^{-1}(sI - PAP^{-1})^{-1}Pb = c^T(sI - A)^{-1}b$$

Next, we explore the possibility to impart a desired structure to the system matrix through linear transformation of the state variables.

First, define $\bar{A} = PAP^{-1}$, $\bar{b} = Pb$; the first equation is alternatively expressed as: $\bar{A}P = PA$.

Then, given A, b, and the desired structure of $\bar{A}$, $\bar{b}$, the above relations may be used to define the transformation matrix, P. In particular, the transformation to the controller and modal forms is explored below.

Transformation into Controller Form

A necessary condition to obtain a linear transformation matrix P to convert a state variable model into controller form is that the following $n \times n$ controllability matrix has full rank:

$$M_C = [b, \ Ab, \ldots, \ A^{n-1}b]$$

The controllability matrix is guaranteed to be of full rank if the transfer function description of the model has the same order, n, as the number of state variables used to describe the system, i.e., there are no pole-zero cancelations in the transfer function, $G(s) = \mathbf{c}^T(sI - A)^{-1}\mathbf{b}$.

Next, assume that the controllability matrix, M_C, of pair (A, b) is of full rank. Let M_{CF} denote the controllability matrix of the controller form representation $(\bar{A}, \bar{b})$; then, M_{CF} contains the coefficients of the characteristic polynomial and can be written by inspection.

The required transformation matrix is defined by:

$$Q = P^{-1} = M_C M_{CF}^{-1}$$

Example 8.4.1

The state and output equations for a small DC motor model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

The controllability matrix for the model is formed as: $M_C = [b,\ Ab] = \begin{bmatrix} 100 & -10^4 \\ 0 & 500 \end{bmatrix}$.

As the controllability matrix is of full rank, the dc motor model is controllable.

From the transfer function description, $G(s) = \frac{500}{s^2+110s+1025}$, the controller form realization is obtained as:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1025 & -110 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} V_a,\ \ \omega = [\,500 \quad 0\,]\, x$$

The controllability matrix for the controller form realization is given as: $M_{CF} = \begin{bmatrix} 0 & 1 \\ 1 & -110 \end{bmatrix}$.

The state transformation matrix for the model is obtained as:

$$Q = P^{-1} = \begin{bmatrix} 1000 & 100 \\ 500 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0.002 \\ 0.01 & -0.02 \end{bmatrix}$$

Indeed, $\overline{A} = PAQ$ matches the system matrix in the controller form.

Transformation into Modal Form

A matrix that has a full set of eigenvectors is diagonalizable by a linear transformation matrix when the eigenvectors of A are selected as the columns of P^{-1}.

In the event when A has complex eigenvalues, its eigenvectors are also complex. By placing the real and imaginary parts of the complex eigenvector into separate columns of P^{-1}, the resulting modal form matrix, $\bar{A} = PAP^{-1},\ \bar{b} = Pb$, has real entries.

Example 8.4.2

The state and output equations for a mass–spring–damper model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,\ \ y = [\,1 \quad 0\,] \begin{bmatrix} x \\ v \end{bmatrix}.$$

The eigenvalues of the system matrix are: $-1 \pm j1$; the complex eigenvectors are: $\begin{bmatrix} 1 \\ 1 \pm j1 \end{bmatrix}$.

By choosing $P^{-1} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$, the modal form system matrix is obtained as: $\bar{A} = PAP^{-1} = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}$.

Example 8.4.3

The state variable model of a small DC motor is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a,\ \ \omega = [\,0 \quad 1\,] \begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

We use MATLAB 'eig' command to obtain: $V = P^{-1} = \begin{bmatrix} -0.9985 & 0.0556 \\ 0.0556 & -0.9985 \end{bmatrix}$.

The transformed state variable model has a diagonal structure.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -99.72 & 0 \\ 0 & -10.28 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -100.47 \\ -5.6 \end{bmatrix} V_a,\ \ \omega = [\,0.056 \quad -0.998\,] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The decoupled system of equations can be easily integrated. Assuming a unit-step input, the state variables are solved as:

$$x_1(t) = 1.0075 + (x_{10} - 1.0075)e^{-t/99.72}$$

$$x_2(t) = 0.545 + (x_{20} - 0.545)e^{-t/10.28}$$

where $x_{10},\ x_{20}$ represent the initial conditions on state variables. The output is computed as:

$$\omega(t) = 0.056x_1(t) - 0.998x_2(t)$$

The original state variables are recovered as: $\begin{bmatrix} i_a \\ \omega \end{bmatrix} = P^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, i.e.,

$$i_a\,(t) = -0.9985x_1\,(t) + 0.0556x_2\,(t)$$

$$\omega\,(t) = 0.0556x_1\,(t) - 0.9985x_2\,(t)$$

CHAPTER OVERVIEW

9: Controllers for State Variable Models

Learning Objectives

1. Perform controller design for state variable models.
2. Perform pole placement design using a variety of techniques.
3. Design reference tracking controllers for state variable models.

9.0: Prelude to Controllers for State Variable Models

The state variable models of dynamic systems comprises first-order differential equations that express the time derivatives of a set of state variables. For linear time-invariant (LTI) systems, these equations are commonly expressed in vector-matrix form.

The controller design for the state variable models involves feeding back all the state variables using appropriate weights to generate the error signal. State feedback allows arbitrary placement of roots of the closed-loop characteristic polynomial. It is more powerful and offers greater flexibility than the output feedback that allows only selective placement of closed-loop roots. State feedback assumes that the complete set of state variables are availabe for feedback.

The pole placement design refers to the selection of feedback gains for placing the roots of the closed-loop characteristic polynomial at the desired locations in the complex plane. The pole placement design is performed with ease when the state variable model is in the controller form. Alternately, Ackermann's and Bass-Gura formulas, or the Sylvester's equation can be used for this purpose.

Output regulation refers to finding a control law to asymptotically track prescribed refernce signals and/or asymptotically reject undesired disturbance inputs. It includes imparting desired degree of dynamic stability to the system through arbitrary selection of the closed-loop characteristic polynomial.

The tracking system design involves reducing the steady-state error to a given reference input to zero. Though reference signal can be used to cancel the tracking error, the design is not robust against parameter variations. A more robust design involves integrating the error signal inside the feeback loop to form an augmented system model, which can used for pole placement. The augmented model includes the differential equation describing the out put of the integrator.

The state-space design methods primarily cater to the design of multi-input multi-output (MIMO) systems. This chapter, however, introduces the state-space design methods using examples from single-input single-output (SISO) systems.

9.1: Controller Design in Sate-Space

State Feedback Controller Design

The state feedback controller design refers to the selection of individual feedback gains for the complete set of state variables. It is assumed that all the state variables are available for observation. The design goal is to improve the transient response of the system and reduce the steady-state tracking error.

Let $\mathbf{x}(t)$ denote a vector of n state variables, $u(t)$ denote a scalar input, and $y(t)$ denote a scalar output; then the state variable model of a SISO system is given as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \quad y(t) = \mathbf{c}^T\mathbf{x}(t)$$

where $\mathbf{A}$ is a $n \times n$ system matrix, $\mathbf{b}$ is a $n \times 1$ column vector, and $\mathbf{c}^T$ is a $1 \times n$ row vector. The above state variable model represents a strictly proper input-output transfer function.

Controller design in state-space involves selection of suitable feedback gain vector, $\mathbf{k}^T$, that imparts desired stability and performance characteristics to the closed-loop system.

Pole Placement Design

Pole placement design refers to the selection of the feedback gain vector that places the poles of the characteristic polynomial at the desired locations. The control law for the pole placement design is expressed as:

$$u(t) = -\mathbf{k}^T\mathbf{x}(t) + r(t),$$

where $\mathbf{k}^T = [k_1, \quad k_2, \ \ldots, k_n]$ is a vector of n feedback gains, one for each of the n state variables, and $r(t)$ is a reference input.

A necessary condition for pole placement using state feedback is that pair $(\mathbf{A}, \mathbf{b})$ is controllable, i.e., its controllability matrix, $\mathbf{M}_\mathrm{C}$, is of full rank, where $\mathbf{M}_\mathrm{C} = [\mathbf{b}, \ \mathbf{A}\mathbf{b}, \ \ldots, \ \mathbf{A}^{n-1}\mathbf{b}]$.

By including control law in the state equations, the closed-loop system is given as:

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{b}\mathbf{k}^T)\mathbf{x}(t) + \mathbf{b}r(t)$$

The design problem, then, is to select the feedback gain vector $\mathbf{k}^T$ that multiplies the state vector $\mathbf{x}(t)$, such that the closed-loop system matrix $\mathbf{A} - \mathbf{b}\mathbf{k}^T$ has characteristic polynomial that aligns with a desired polynomial, i.e.,

$$\left|s\mathbf{I} - \mathbf{A} + \mathbf{b}\mathbf{k}^T\right| = \Delta_{\text{des}}(s)$$

The above equation relates two nth order polynomials; by equating the polynomial coefficients on both sides of the equation, we obtain n linear equations that can be solved for the n feedback gains: $k_i, \ i = 1, \ldots, n$.

The desired characteristic polynomial for pole placement design is selected with root locations that meet the time-domain performance criteria.

Closed-loop System

The closed-loop system model is given as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_\text{cl}\mathbf{x}(t) + \mathbf{b}r(t)$$

where $\mathbf{A}_\text{cl} = (\mathbf{A} - \mathbf{b}\mathbf{k}^T)$. Assuming closed-loop stability, for a constant input $r(t) = r_\text{ss}$, the steady-state response, $\mathbf{x}_\text{ss}$, obeys:

$$0 = \mathbf{A}_\text{cl}\mathbf{x}_{ss} + \mathbf{b}r_{ss}, \quad y_\text{ss} = \mathbf{c}^T\mathbf{x}_{ss}$$

Hence, $y_\text{ss} = \mathbf{c}^T \ \mathbf{A}_\text{cl}^{-1} \ \mathbf{b} \ r_\text{ss}$.

Example 9.1.1

The state and output equations for a DC motor model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}.$$

The motor model has a denominator polynomial $\Delta(s) = s^2 + 110s + 1025$; its roots are located at $s_{1,2} = -10.28,\ -99.72$ that are reciprocals of the motor time constants ($\tau_e \cong 10$ msec, $\tau_m \cong 100$ msec).

Assume that the design specifications require the closed-loop response to have a time constant of $20ms$.

To reduce a time constants to 20 msec, we may choose to place the closed-loop poles at $-50, -100$. Then, the desired characteristic polynomial is given as: $\Delta_{des}(s) = s^2 + 150s + 5000$.

We may assume $r = 0$, so the control law is given as: $V_a = -k^T x$, where $k^T = [k_1,\ k_2]$.

With the inclusion of state feedback, the closed-loop system matrix is given as:

$$A - bk^T = \begin{bmatrix} -100(1+k_1) & -5(1+20k_2) \\ 5 & -10 \end{bmatrix}$$

The closed-loop characteristic polynomial is given as:

$$\left|sI - A + bk^T\right| = (s+10)\left[s + 100(1+k_1)\right] + 25(1+20k_2).$$

By comparing the coefficients of the two polynomials, we obtain the feedback gains: $k_1 = 0.4, k_2 = 7.15$. The resulting control law is given as:

$$V_a(t) = -0.4\, i_a(t) - 7.15\, \omega(t)$$

The closed-loop system system matrix is given as: $\begin{bmatrix} -140 & -20 \\ 5 & -10 \end{bmatrix}$.

The steady-state system response is given as: $y_{ss} = 0.1 r_{ss}$.

The step response of the closed-loop system is plotted in Figure 9.1.1.

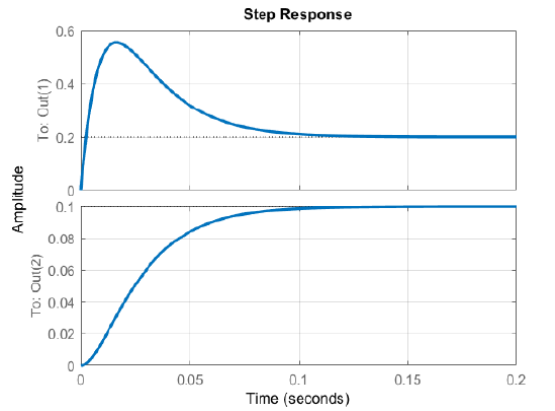


Figure 9.1.1: The step response of the DC motor with pole placement controller: armature current (top); motor speed (bottom).

Pole Placement in Controller Form

The pole placement design is facilitated if the system model is in the controller form (Section 8.3.1). In the controller form structure, the coefficients of the characteristic polynomial appear in reverse order in the last row of A matrix.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \ddots & 1 \\ -a_n & -a_{n-1} & \dots & -a_1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Then, using full state feedback through $u = -k^T x + r$, the closed-loop system matrix is given as:

$$A - bk^T = \begin{bmatrix} 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \ddots & 1 \\ -a_n - k_1 & -a_{n-1} - k_{n-1} & \dots & -a_1 - k_n \end{bmatrix}.$$

The closed-loop characteristic polynomial can be written by inspection, and is given as:

$$\Delta(s) = s^n + (a_1 + k_n)s^{n-1} + \ldots + a_n + k_1$$

Let the desired characteristic polynomial be defined as:

$$\Delta_{\text{des}}(s) = s^n + \bar{a}_1 s^{n-1} + \ldots + \bar{a}_{n-1}s + \bar{a}_n.$$

By comparing the coefficients, the desired feedback gains are computed as:

$$k_1 = \bar{a}_n - a_n, \quad k_2 = \bar{a}_{n-1} - a_{n-1}, \ \ldots \ , \quad k_n = \bar{a}_1 - a_1.$$

Since the state variables in the controller form include system output and its derivatives, pole placement using state feedback is a generalization of proportional-derivative (PD) and rate feedback controllers.

Example 9.1.2

The state variable model of a mass–spring–damper is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & -1 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f, \quad x = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix}.$$

Assume that the design specifications require raising the damping to $\zeta = 0.6$.

Since the system model is in controller form, its characteristic polynomial can be written by inspection, and is given as: $\Delta(s) = s^2 + s + 10$.

In order to improve the damping, a desired characteristic polynomial is selected as: $\Delta_{\text{des}}(s) = s^2 + 4s + 10$.

The desired feedback gains can be written by inspection, and are given as: $k^T = [0 \ \ 3]$.

Pole Placement using Bass-Gura Formula

The Bass-Gura formula describes a simple expression to compute the state feedback controller gains from the coefficients of the available and desired closed-loop characteristic polynomials. To proceed, let the state variable model be given as:

$$\dot{x}(t) = Ax(t) + bu(t), \quad y(t) = c^T x(t)$$

Assuming that pair (A, b) is controllable, its controllability matrix is given as: $M_{\text{C}} = [b, \ Ab, \ \ldots, \ A^{n-1}b]$.

The model is transformed into controller form by a linear transformation, $z = Px$; the transformed model is described as:

$$\dot{z}(t) = A_{\text{CF}}z(t) + b_{\text{CF}}u(t), \quad y(t) = c_{\text{CF}}^T z(t)$$

The controllability matrix of the controller form representation is given as: $M_{\text{CF}} = [b_{\text{CF}}, \ A_{\text{CF}}b_{\text{CF}}, \ldots, \ {A_{\text{CF}}}^{n-1}b_{\text{CF}}]$. The matrix comprises the coefficients of the characteristic polynomial, $\Delta(s) = |sI - A|$, and can be written by inspection.

The required linear transformation matrix is obtained from the following relation:

$$P^{-1} = M_{\text{C}}M_{\text{CF}}^{-1}, \quad P = M_{\text{CF}}M_{\text{C}}^{-1}.$$

The state feedback control law in the controller form is defined as:

$$u = -{k_{\text{CF}}}^T z(t) = -{k_{\text{CF}}}^T Px(t).$$

In terms of polynomial coefficients, the controller gains are given as: ${k_{\text{CF}}}^T = [\, \bar{a}_n - a_n \quad \bar{a}_{n-1} - a_{n-1} \quad \cdots \quad \bar{a}_1 - a_1 \,]$.

The controller gains for the original state variable model are obtained as: $k^T = {k_{\text{CF}}}^T P$.

Hence, the Bass-Gura formula is obtained as:

$$k^T = [\ \bar{a}_n - a_n \quad \bar{a}_{n-1} - a_{n-1} \quad \cdots \quad \bar{a}_1 - a_1\ ]\ M_{\text{CF}} M_{\text{C}}^{-1}$$

Example 9.1.3

The state equations for a DC motor model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a$$

Assume that the design specifications require a time constant of $20ms$.

The characteristic polynomial of the model is: $|sI - A| = s^2 + 110s + 1025$.

The controllability matrix of the model is formed as: $M_{\text{C}} = [\text{b},\ \text{Ab}] = \begin{bmatrix} 100 & -10^4 \\ 0 & 500 \end{bmatrix}$.

The controller form representation for the model is developed as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1025 & -110 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} V_a$$

The controllability matrix for the controller form representation is: $M_{\text{CF}} = \begin{bmatrix} 0 & 1 \\ 1 & -110 \end{bmatrix}$.

Assume that the desired characteristic polynomial is selected as: $\Delta_{\text{des}}(s) = s^2 + 150s + 5000$.

The feedback gains for the controller form representation are obtained as: ${k_{\text{CF}}}^T = [\,3975 \quad 40\,]$.

Using the Bass-Gura formula, the state feedback controller gains for the DC motor model are computed as: $k^T = {k_{\text{CF}}}^T M_{\text{CF}} M_{\text{C}}^{-1} = [\,0.4 \quad 7.15\,]$.

The resulting state feedback control law is given as: $V_a = -0.4 i_a - 7.15\omega$.

Pole Placement using Ackermann's Formula

The Ackermann's formula is, likewise, a simple expression to compute the state feedback controller gains for pole placement. To develop the formula, let an n-dimensional state variable model be given as:

$$\dot{x}(t) = Ax(t) + bu(t)$$

The controllability matrix of the model is formed as: $M_{\text{C}} = [b,\ Ab,\ \ldots,\ A^{n-1}b]$.

Assume that a desired characteristic polynomial is given as: $\Delta_{\text{des}}(s) = s^n + \bar{a}_1 s^{n-1} + \ldots + \bar{a}_{n-1}s + \bar{a}_n$.

Next, we evaluate the following matrix polynomial that involves the system matrix:

$$\Delta_{\text{des}}(A) = A^n + \bar{a}_1 A^{n-1} + \ldots + \bar{a}_{n-1}A + \bar{a}_n I.$$

where I denotes an $n \times n$ identity matrix. The state feedback controller gains are obtained from the following Ackermann's formula:

$$k^T = [\,0 \quad \cdots \quad 0 \quad 1\,]\ {M_{\text{C}}}^{-1} \Delta_{\text{des}}(A)$$

Example 9.1.4

The state equations for a DC motor model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a$$

Let the desired characteristic polynomial be given as: $\Delta_{\text{des}}(s) = s^2 + 150s + 5000$.

The controllability matrix for the model is computed as: $M_{\text{C}} = [b,\ Ab] = \begin{bmatrix} 100 & -10^4 \\ 0 & 500 \end{bmatrix}$.

The matrix polynomial used in the formula is evaluated as: $\Delta_{\text{des}}(A) = \begin{bmatrix} -25 & -200 \\ 200 & 3575 \end{bmatrix}$.

Using the Ackermann's formula, the pole placement controller is computed as: $k^T = [0.4 \quad 7.15]$.

The resulting state feedback control law is given as: $V_a = -0.4 i_a - 7.15\omega$.

The MATLAB Control System Toolbox includes the 'place' command that uses Ackermann's formula for pole placement design. The command is invoked by entering the system matrix, the input vector, and a vector of desired roots of the characteristic polynomial. The function returns the feedback gain vector that places the eigenvalues of $A - bk^T$, at the desired root locations.

Pole Placement using Sylvester's Equation

The Sylvester equation in linear algebra is a matrix equation, given as: $AX + XB = C$, where A and B are square matrices of not necessarily equal dimensions, and C is a matrix of appropriate dimensions. The solution matrix X is of the same dimension as C.

In order to apply the Sylvester equation for pole placement design, let an n-dimensional state variable model be given as:

$$\dot{x}(t) = Ax(t) + bu(t)$$

The pair (A, b) is assumed to be controllable.

Let $A - bk^T$ represent the closed-loop system matrix; assume that a diagonalizing similarity transform is defined as:

$$X^{-1}\left(A - bk^T\right)X = \Lambda$$

The matrix Λ is diagonal and contains the desired eigenvalues; it may be assumed in modal form for complex eigenvalues.

Let $k^T X = g^T$; then, the Sylvester eqution is formulated as:

$$AX - X\Lambda = bg^T$$

A unique solution X to the Sylvester's equation exists if the eigenvalues of A and Λ are distinct, i.e., $\lambda_i(A) - \lambda_j(\Lambda) \neq 0$.

The design procedure for pole placement using Sylvester's equation is given as follows:

1. Choose a matrix Λ of desired eigenvalues in modal form. Choose any g^T and solve $AX - X\Lambda = bg^T$ for X.
2. Recover the feedback gain vector from $k^T = g^T X^{-1}$. If X is not invertible, choose a different g^T and repeat the procedure.

Example 9.1.5

The state equation for the mass–spring–damper model is described as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & -1 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f$$

Let a desired characteristic polynomial be selected as: $\Delta_{\text{des}}(s) = s^2 + 4s + 10$; then, the modal matrix of the desired eigenvalues is: $\Lambda = \begin{bmatrix} -2 & \sqrt{6} \\ -\sqrt{6} & -2 \end{bmatrix}$.

Let $g^T = [1 \quad 0]$; then, $bg^T = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$. The solution to Sylvester's equation is obtained as: $X = \begin{bmatrix} -0.067 & -0.082 \\ 0.333 & 0.0 \end{bmatrix}$.

The feedback controller gains are computed as: $k^T = [0 \quad 3]$.

9.2: Tracking with Feedforward Controller

Tracking Design with Feedforward Controller

A tracking system is designed to asymptotically reach and maintain zero steady-state error with respect to a reference input, $r(t)$. This can be achieved by using $r(t)$ to cancel $e(t)$ in the steady state.

The reference signal can be used with a feed forward gain to cancel the error signal. Toward this end, let the state variable model be given as:

$$\dot{x}(t) = Ax(t) + bu(t), \quad y(t) = c^T x(t)$$

The asymptotic tracking control law is defined as:

$$u(t) = -k^T x(t) + k_r r(t)$$

where k_r is a scalar gain. The closed-loop system is formed as:

$$\dot{x}(t) = \left(A - bk^T\right) x(t) + bk_r r(t), y(t) = c^T x(t)$$

Assuming the closed-loop system is stable, let $\dot{x}(t) = 0$; the steady-state values of the state variables are obtained as:

$$x_{ss} = -\left(A - bk^T\right)^{-1} bk_r r_{ss}.$$

The steady-state output is given as:

$$y_{ss} = -c^T \left(A - bk^T\right)^{-1} b\, k_r r_{ss}$$

Define the closed-loop transfer function:

$$T(s) = c^T \left(sI - A + bk^T\right)^{-1} b$$

Then, the steady-state output is: $y_{ss} = T(0) k_r r_{ss}$.

In order to ensure $y_{ss} = r_{ss}$, we may choose, $k_r = T(0)^{-1}$.

Further, if $y = x_1$, then k_r may be selected as $k_r = k_1$, to ensure $r - y = 0$ in the steady-state.

Example 9.2.1

The state and output equations for a DC motor model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix}.$$

A pole placement controller for the DC motor model was previously designed as: $V_a = -0.4 i_a - 7.15\omega$. The control law is revised to include a feedforward gain: $V_a = -0.4 i_a - 7.15\omega + k_r r$, where we want to choose k_r for zero steady-state error.

By including the controller, the closed-loop system model is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -140 & -720 \\ 5 & -10 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} k_r r$$

The state variables assume the following values in steady-state: $\begin{bmatrix} i_{a,ss} \\ \omega_{ss} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix} k_r r_{ss}$.

The steady-state value of the output is given as: $y_{ss} = 0.1 k_r r_{ss}$. Then, we may choose $k_r = 10$ for zero steady-state error.

The resulting conrol law is given as: $V_a = -0.4 i_a - 7.15\omega + 10r$.

A plot of the motor response with feedforward compensation is shown in Fig. 9.2.1. As expected, the motor speed asymptotically approaches unity in the steady-state.

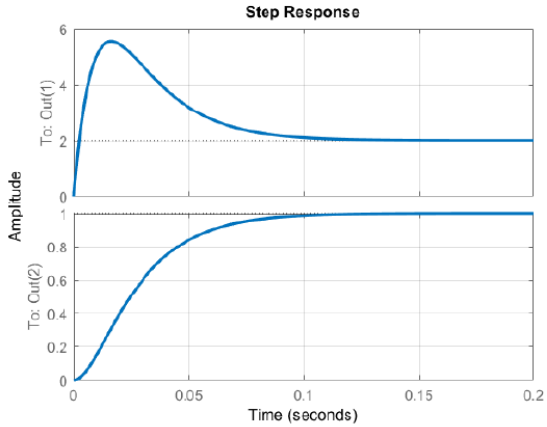


Figure 9.2.1: The step response of the DC motor with pole placement controller and feedforward compensation: armature current (top); motor speed (bottom).

Example 9.2.2

The state variable model of a mass–spring–damper is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & -1 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f, \quad x = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix}$$

A pole placement controller for the model was previously designed as: $f = -3v$.

The control law is modified to include a feedforward gain as: $f = -3v + k_r r$.

The closed-loop system model is given as: $\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & -4 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + k_r r$.

The steady-state output is: $y_{ss} = 0.1 k_r r_{ss}$. Hence, we may choose $k_r = 10$ for error-free tracking. The resulting control law is given as: $f = -3v + 10\, r$.

9.3: Tracking PI Controller Design

Tracking PI Controller Design

The tracking system design by using a feedforward gain to cancel the tracking error is not robust to changes in plant parameters. A robust tracking system design requires addition of an integrator to the feedback loop (Figure 9.3.1).

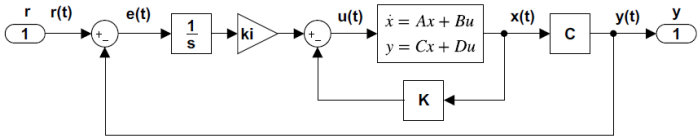


Figure 9.3.1: State feedback with integrator in the loop for tracking a reference input.

A proportional-integral (PI) type control law using state feedback is defined as:

$$u(t) = -\mathbf{k}^T\mathbf{x}(t) + k_i \int (r-y)\mathrm{dt}$$

where k_i represents the integral gain. The integrator operates on the error signal: $e = r - y$. The integral controller forces the error signal to zero in the steady-state.

Let $x_a(t)$ denote the integrator output; then, the integrator state equation is given as:

$$\dot{x}_a = r - y = r - \mathbf{c}^T\mathbf{x}$$

An augmented system model is formed by adding the integrator output to the set of state variables; the resulting model with $n+1$ variables is described as:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_a \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{c}^T & 0 \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix} + \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix} u + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r$$

A state feedback control law for the augmented system model is defined as:

$$u = [-\mathbf{k}^T \quad k_\mathrm{i}] \begin{bmatrix} \mathbf{x} \\ x_a \end{bmatrix},$$

Substitution of the control law in the augmented system model defines the closed-loop system:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_a \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{b}\mathbf{k}^T & \mathbf{b}k_i \\ -\mathbf{c}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_a \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r$$

The closed-loop characteristic polynomial is formed as: $\Delta_a(s) = \begin{vmatrix} s\mathbf{I} - \mathbf{A} + \mathbf{b}\mathbf{k}^T & -\mathbf{b}k_i \\ \mathbf{c}^T & s \end{vmatrix}$, where $\mathbf{I}$ denotes an identity matrix of order n.

Next, we choose a $(n+1)$ order desired characteristic polynomial, $\Delta_{des}(s)$, and perform the pole placement design on the augmented system. The location of the integrator pole may be adjusted by trial and error keeping in view the desired settling time of the system.

Examples

Example 9.3.1

The state variable model of a mass-spring-damper system is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & -1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f, \quad x = [1 \quad 0] \begin{bmatrix} x \\ v \end{bmatrix}$$

It is desired to have less than 10% overshoot and zero steady-state error to a step input.

In order to perform the integral control design, an augmented state variable model is formed as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x \\ v \\ x_a \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -10 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ v \\ x_a \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r.$$

The control law for the augmented system is given as: $u = -k_1 x - k_2 v + k_i \int (r - x)\mathrm{dt}$.

The closed-loop system characteristic polynomial is formed as: $\Delta(s) = s^3 + (k_2 + 1)s^2 + (k_1 + 10)s + k_i$. Let a third-order desired characteristic polynomial be selected as: $\Delta_{\text{des}}(s) = (s+2)(s^2 + 4s + 10)$.

By comparing the polynomial coefficients, the controller gains are obtained as: $k_1 = 8, k_2 = 5,\ k_i = 20$. The resulting control law is given as: $f = -8x - 5v + 20 \int (r - y)\, dt$.

The step response of the closed-loop system is shown in Fig. 9.3.1 for three different values of integral gain.

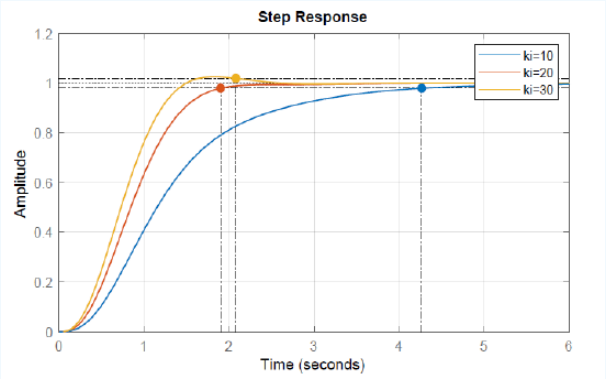


Figure 9.3.1 : tep response of the mass-spring-damper system with integrator in the loop.

Example 9.3.2

The state variable model of a DC motor is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

Assume that design specifications call for a low overshoot to step response and a settling time, $t_s \cong 0.1\text{s}$.

The control law for the tracking PI controller is given as: $u = -k_1 i_a - k_2 \omega + k_i \int (r - \omega)\mathrm{dt}$.

The augmented system model for pole placement using integral control is obtained as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \\ x_a \end{bmatrix} = \begin{bmatrix} -100 & -5 & 0 \\ 5 & -10 & 0 \\ 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \\ x_a \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

The resulting characteristic polynomial is given as:

$$\Delta(s) = s^3 + (100k_1 + 100)\, s^2 + (1000k_1 + 500k_2 + 1025)\, s - 500k_i.$$

We choose a desired characteristic polynomial as: $\Delta_{des}(s) = (s + 50)\left(s^2 + 100s + 5,000\right)$. The closed-loop roots are located at: $s = -50,\ -50 \pm j50$.

The tracking controller is defined as: $V_a(s) = -0.4\, i_a(t) - 17.2\, \omega(t) + 500 \int (r - \omega) dt$.

For comparison a cascade PI controller for the transfer function model of the DC motor is designed for $\zeta = 0.7$. The PI controller is given as: $K(s) = \frac{10(s+10)}{s}$. The closed-loop roots are placed at: $s = -9.94, -50 \pm j50.3$.

The step response of the closed-loop system for the tracking and cascade PI controllers is shown in Fig. 9.3.2. Both controllers achieve the desired settling time.

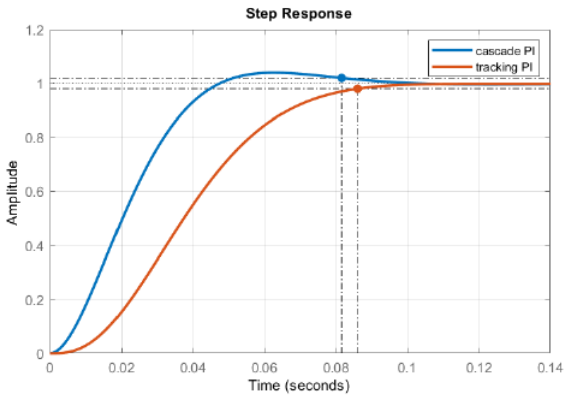


Figure 9.3.2: The step response of the DC motor model for cascade and tracking PI controllers.

CHAPTER OVERVIEW

10: Controllers for Discrete State Variable Models

Learning Objectives

1. Analyze state variable models of sampled-data systems.
2. Solve discrete-time state equations by iteration.
3. Perform pole placement controller design for sampled-data systems.
4. Perform tracking controller desing for sampled-data systems.

10.0: Prelude to Controllers for Discrete State Variable Models

The state variable model of a dynamic system comprises a set of first-order differential equations that express the system behavior as time derivatives of the selected variables. The controller design for state state variable model involves selecting feedback gains for each (or a selection) of the state variables. The state feedback controller offers greater flexibility in system design compared to the output feedback controllers designed with transfer function models.

The continuous-time state variable model of a linear time-invariant (LTI) system can be converted to a discrtete-time LTI model by assuming a piece-wise constant input to the system. The conversion is easily done in MATLAB by using the 'c2d' command. The assumed sampling frequency should be five to ten times higher than the system bandwidth.

The control system design methods used with continuous-time systems can be extended to digital controller design of sampled-data systems that include sample and hold elements. The time-domain description of a sampled-data system comprisess a set of difference equations that can be easily solved by iteration.

The pole placement design using full state feedback can be similarly performed on discrete state variable models. The desired characteristic polynomial in the discrete case should have its roots inside the unit circle to ensure the stability of the closed-loop system. A discrete system model addionally allow the design of a deadbeat controller that places all roots of the closed-loop characteristic polynomial at the origin. The deadbeat controller ensures that the system response reaches steady state in exactly n iterations.

Compared to the root locus design that allows selective placement of the closed-loop poles of the characteristic polynomial, arbitrary pole placement is made possible through the use of state feedback design in the case of state variable models. In the following, we assume that the system to be controlled is of single-input single-output (SISO) type.

10.1: State Variable Models of Sampled-Data Systems

Discretizing the State Variable Model

System models described with state variables can be converted to their discrete-time equivalents by considering a zero-order hold (ZOH) at the input of the model. The ZOH device converts the output of a digital controller (a number sequence) into a piece-wise constant continuous-time signal by holding its output constant over successive time periods.

To develop this approach, let the continuous-time state variable model be given as:

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

$$y(t) = Cx(t)$$

where A is the system matrix, B is the input matrix, and C is the output matrix.

In order to discretize the continuous-time state equations, we consider the time-domain solution to the state equation (Chapter 8) given in terms of a convolution integral:

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{\tau} e^{A(t-\tau)}\,B\,u(\tau)\,\mathrm{d}\tau$$

It is assumed that the system state is available at $t_0 = (k-1)T$; then, assuming a constant input, u_k, the state at $t = kT$ is given as:

$$x_k = e^{At}x_{k-1} + \int_{(k-1)T}^{kT} e^{AT}\,B\,u_k\,\mathrm{d}\tau$$

A simple change of variables results in the following expression:

$$x_k = e^{At}x_{k-1} \; + \int_{0}^{T} e^{AT}d\tau\,B\,u_k$$

Let the system and input matrices appearing in the discrete-time model be defined as:

$$A_\mathrm{d} = e^{AT}, \;\; B_\mathrm{d} = \int_0^T e^{A\tau}d\tau B$$

Then, after a time shift, the discrete-time state variable model is expressed as:

$$x_{k+1} = A_\mathrm{d}x_k + B_\mathrm{d}u_k, \quad y_k = C_\mathrm{d}x_k.$$

where $C_\mathrm{d} = C$.

Assuming that system matrix A is invertible, the expression for B_d can be simplified as:

$$B_\mathrm{d} = \left[\int_0^T \left(I + A\tau + \frac{A^2\tau^2}{2!} + \ldots\right) d\tau\right] = \left(IT + \frac{AT^2}{2!} + \ldots\right) B = A^{-1}(e^{AT} - I)B$$

The system and input matrices $(A_\mathrm{d}, B_\mathrm{d})$ appearing in the discrete-time model are parameterized by the sample time, T. Hence, changing T will change the discrete-time model.

In the MATLAB Control Systems Toolbox, ‘c2d’ command is used for the discretizing a state variable model; the presence of a ZOH at the input is assumed, but other options are available.

Example 10.1.1

The state variable model of a DC motor is given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \;\; \omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

Let $T = 0.02$ s, a value five times faster than the dominant motor time constant: $\tau_{\mathrm{m}} \cong 0.1$ s.; then, the system and input matrices for the discrete model are computed as:

$$A_{\mathrm{d}} = e^{At} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix}, B_{\mathrm{d}} = A^{-1}\left(e^{At} - I\right)B = \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix}$$

The resulting discrete state variable model is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix} V_k, \quad y_k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix}$$

An alternate state variable model of the DC motor is obtained by using the 'ss' command in the MATLAB Control Systems Toolbox to realize the motor transfer function; the model is given as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -110 & -32.03 \\ 32 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 3.906 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

By using the MATLAB 'c2d' command, the corresponding discrete-time model is obtained as:

$$\begin{bmatrix} x_{1,\,k+1} \\ x_{2,\,k+1} \end{bmatrix} = \begin{bmatrix} 0.058 & -0.243 \\ 0.243 & 0.892 \end{bmatrix} \begin{bmatrix} x_{1,\,k} \\ x_{2,\,k} \end{bmatrix} + \begin{bmatrix} 0.030 \\ 0.013 \end{bmatrix} u_k, \quad y_k = \begin{bmatrix} 0 & 3.91 \end{bmatrix} \begin{bmatrix} x_{1,\,k} \\ x_{2,\,k} \end{bmatrix}$$

The two discrete-time state variables of the DC motor are equivalent, i.e., they both share the same z-plane eigenvalues: $z_{1,2} = 0.814,\ 0.136$; these values are related to the analog system eigenvalues: $s_{1,2} = -99.7,\ -10.28$ by the relation: $z = e^{Ts}$.

Iterative Solution to Discrete State Equations

The discrete-time state equations constitute a set of first-order difference equations that can be easily solved by iteration. Toward this end, let the discrete state equation be given as:

$$x_{k+1} = A_{\mathrm{d}} x_k + B_{\mathrm{d}} u_k, \quad y_k = C_{\mathrm{d}} x_k$$

Starting from an initial vector, x_0, and given an input sequence $u\{k\}$, an iterative solution to the discrete state equation is developed as follows:

$$x_1 = A_{\mathrm{d}} x_0 + B_{\mathrm{d}} u_0$$

$$x_2 = A_{\mathrm{d}}^2 x_0 + A_{\mathrm{d}} B_{\mathrm{d}} u_0 + B_{\mathrm{d}} u_1$$

$$\vdots$$

$$x_n = A_{\mathrm{d}}^n x_0 + \sum_{k=0}^{n-1} A_{\mathrm{d}}^{n-1-k} B_{\mathrm{d}} u_k$$

Let $\Phi(k) = A_{\mathrm{d}}^k$ define the discrete state-transition matrix; then, the solution is given as:

$$x_n = \Phi(n) x_0 + \sum_{k=0}^{n-1} \Phi(n-1-k) B_{\mathrm{d}} u_k$$

Example 10.1.2

The discrete state variable model of a dc motor ($T = 0.02s$) is given as:

$$A_{\mathrm{d}} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix}, \quad B_{\mathrm{d}} = \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix}, \quad C_{\mathrm{d}} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Assuming zero initial conditions and a unit-input sequence: $u_k = \{1,\ 1, \ldots\}$; the output sequence is iteratively computed as:

$$y\{k\} = \{0,\ 0.053,\ 0.128,\ 0.194,\ 0.249,\ 0.293,\ 0.329,\ 0.359,\ 0.383,\ 0.402,\ 0.418, \ldots\}$$

The Pulse Transfer Function

The pulse transfer function, $G(z)$, of a sampled-data system can be obtained from discrete state equations by the application of z-transform:

$$zx(z) - zx_0 = A_{\mathrm{d}}x(z) + B_{\mathrm{d}}u(z)$$

The above equation is solved assuming zero initial conditions to obtain:

$$x(z) = (zI - A_{\mathrm{d}})^{-1}B_{\mathrm{d}}u(z)$$

The output equation is defined as:

$$y(z) = C_{\mathrm{d}}(zI - A_{\mathrm{d}})^{-1}B_{\mathrm{d}}u(z) = G(z)u(z)$$

Given the discrete state equations, the pulse transfer function is obtained as:

$$G(z) = C_{\mathrm{d}}(zI - A_{\mathrm{d}})^{-1}B_{\mathrm{d}}$$

The discrete state-transition matrix is obtained by taking the inverse z-transform of $(zI - A_{\mathrm{d}})^{-1}$ as:

$$\phi(k) = z^{-1}\left\{(zI - A_{\mathrm{d}})^{-1}\right\} = A_{\mathrm{d}}^{k}$$

In terms of the state transition matrix, the unit-pulse response the sampled-data system is obtained as:

$$g_k = C_{\mathrm{d}}A_{\mathrm{d}}^{k-1}B, \quad k \geq 0$$

In the MATLAB Control System Toolbox, the pulse transfer function for a given discrete state variable model can be obtained by invoking the ‘tf’ command.

Example 10.1.3

The discrete state variable model of a dc motor ($T = 0.02s$) is given as:

$$A_{\mathrm{d}} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix}, \quad B_{\mathrm{d}} = \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix}, \quad C_{\mathrm{d}} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

By using the 'tf' command in MATLAB, the motor pulse transfer function is obtained as:

$$G(z) = \frac{0.053z + 0.0257}{z^2 - 0.95z + 0.111}.$$

10.2: Controllers for Discrete State Variable Models

Emulating an Analog Controller

The pole placement controller designed for a continuous-time state variable model can be used with derived sampled-data system model. Successful controller emulation requires a high enough sampling rate that is at least ten times the frequency of the dominant closed-loop poles of the system.

In the following we illustrate the emulation of pole placement controller designed for the DC motor model (Example 8.3.4) for controlling the discrete-time model of the DC motor. The DC motor model is discretized at two different sampling rates for comparison, assuming ZOH at the plant input.

Example 10.2.1

The state and output equations for a DC motor model are given as:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -100 & -5 \\ 5 & -10 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega \end{bmatrix}.$$

The motor model is discretized at two different sampling rates in MATLAB. The results are:

$$T = 0.01s : A_{\mathrm{d}} = \begin{bmatrix} 0.367 & -0.030 \\ 0.030 & 0.904 \end{bmatrix}, \quad B_{\mathrm{d}} = \begin{bmatrix} 0.632 \\ 0.018 \end{bmatrix}, \quad C_{\mathrm{d}} = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

$$T = 0.02s : A_{\mathrm{d}} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix}, \quad B_{\mathrm{d}} = \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix}, \quad C_{\mathrm{d}} = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

For a desired characteristic polynomial: $\Delta_{\mathrm{des}}(s) = s^2 + 150\,s + 5000$, a state feedback controller for the continous-time state variable model was obtained as (Example 9.1.1): $k^T = \begin{bmatrix} 0.4 & 7.15 \end{bmatrix}$.

We can use the same controller to control the corresponding sample-data system models.

The unit-step response of the closed-loop system is simulated in Figure 10.2.1, where both state variables, $i_a\,(t)$ and $\omega\,(t)$, are plotted.

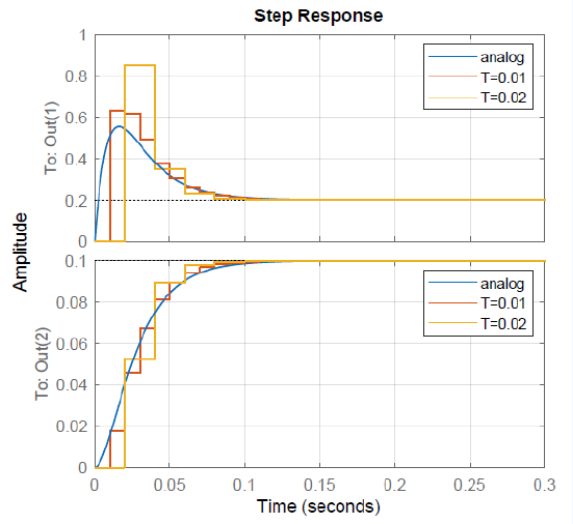


Figure 10.2.1: The step response of the DC motor model with analog controller emulation: armature current (top); motor speed (bottom)

We observe from the figure that the armature current has a higher overshoot at the lower sampling rate, though both models display similar settling time of about 100 msec.

Pole Placement Design of Digital Controller

Given a discrete state variable model $\{A_{\mathrm{d}},\ B_{\mathrm{d}}\}$, and a desired pulse characteristic polynomial $\Delta_{\mathrm{des}}(z)$, a state feedback controller for the system can be designed using pole placement similar to that of the continuous-time system (Sec. 9.1.1).

Let the discrete-time model of a SISO system be given as:

$$\mathbf{x}_{k+1} = \mathbf{A}_{\mathrm{d}}\mathbf{x}_k + \mathbf{b}_{\mathrm{d}}u_k,\ \ y_k = \mathbf{c}^T\mathbf{x}_k$$

A state feedback controller for the discrete state variable model is defined as:

$$u_k = -\mathbf{k}^T\mathbf{x}_k + r_k$$

where $\mathbf{k}^T$ represents a row vector of constant feedback gains and r_k is a reference input sequence. The controller gains can be obtained by equating the coefficients of the characteristic polynomial with those of a desired polynomial:

$$\Delta(z) = |z\mathbf{I} - \mathbf{A}_{\mathrm{d}}| = \Delta_{\mathrm{des}}(z)$$

The $\Delta_{\mathrm{des}}(z)$ above is a Hurwitz polynomial (in z), with roots inside the unit circle that meet given performance (damping ratio and/or settling time) requirements. Assuming that desired s-plane root locations are known, the corresponding z-plane root locations can be obtained from the equivalence: $z = e^{Ts}$.

Closed-loop System

The closed-loop system model is given as:

$$\mathbf{x}_{k+1} = \mathbf{A}_{\mathrm{cl}}\mathbf{x}_k + \mathbf{b}_{\mathrm{d}}r_k,\ \ y_k = \mathbf{c}^T\mathbf{x}_k$$

where $\mathbf{A}_{\mathrm{cl}} = (\mathbf{A}_{\mathrm{d}} - \mathbf{b}_{\mathrm{d}}\mathbf{k}^T)$.

Assuming closed-loop stability, for a constant input $r_k = r_{\mathrm{ss}}$, the steady-state response, $\mathbf{x}_{\mathrm{ss}}$, of the system obeys:

$$\mathbf{x}_{ss} = \mathbf{A}_{\mathrm{cl}}\mathbf{x}_{ss} + \mathbf{b}_{\mathrm{d}}r_{ss},\ \ y_{\mathrm{ss}} = \mathbf{c}^T\mathbf{x}_{ss}$$

Hence, $y_{\mathrm{ss}} = \mathbf{c}^T\left(\mathbf{A}_{\mathrm{cl}} - \mathbf{I}\right)^{-1}\mathbf{b}_{\mathrm{d}}\, r_{\mathrm{ss}}$.

Example 10.2.2

The discrete state variable model of a DC motor ($T = 0.02$s) is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix}\begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix} V_k,\ \ y_k = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_k \\ \omega_k \end{bmatrix}$$

The desired s-plane root locations for the model are given as: $s = -50,\ -100$.

The corresponding z-plane roots ($T = 0.02s$) are obtained as: $z = e^{-1},\ e^{-2}$.

The desired characteristic polynomial is given as: $\Delta_{\mathrm{des}}(z) = z^2 - 0.95z + 0.05$.

The feedback gains $k^T = [k_1,\ k_2]$, computed using the MATLAB 'place' command, are given as: $k_1 = 0.247,\ k_2 = 4.435$.

The closed-loop system matrix is given as: $A_{\mathrm{d}}) = \begin{bmatrix} -0.080 & -3.867 \\ 0.025 & 0.583 \end{bmatrix}$.

An update rule for implementation of the controller on computer is obtained as: $u_k = -0.247\, i_k - 4.435\, \omega_k$.

The closed-loop response has steady-state value of $\omega_{\mathrm{ss}} = 0.143\ \mathrm{rad/s}$.

The step response of the closed-loop system is plotted in Figure 10.2.2, where the discrete system response was scaled to match the analog system response. The step response of the continuous-time system and that for the emulated controller gains are plotted alongside.

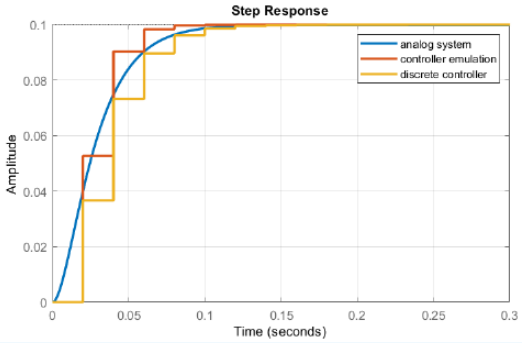


Figure 10.2.2: Unit-step response of the DC motor model for the following choices: analog system; controller emulation; digital controller.

Deadbeat Controller Design

A discrete-time system is called deadbeat if all closed-loop poles are placed at the origin $(z = 0)$.

A deadbeat system has the remarkable property that its response reaches steady-state in n-steps, where n represents the model dimension.

The desired closed-loop pulse characteristic polynomial is selected as $\Delta_{\text{des}}(z) = z^n$.

To design a deadbeat controller, let the closed-loop pulse transfer function be defined as:

$$T(z) = \frac{K(z)G(z)}{1 + K(z)G(z)}$$

The above equation is solved for $K(z)$ to obtain:

$$K(z) = \frac{1}{G(z)} \frac{T(z)}{1 - T(z)}$$

Let the desired $T(z) = z^{-n}$; then, the deadbeat controller is given as:

$$K(z) = \frac{1}{G(z)(z^n - 1)}$$

Example 10.2.3

Let $G(s) = \frac{1}{s+1}$; then $G(z) = \frac{1-e^{-T}}{z-e^{-T}}$.

A deadbeat controller for the model is obtained as: $K(z) = \frac{z-e^{-T}}{(1-e^{-T})(z-1)}$.

Example 10.2.4

The discrete state variable model of a DC motor for $T = 0.02$ s is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix} V_k, \quad y_k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix}$$

The state feedback controller is given as: $u_k = -\begin{bmatrix} k_1, & k_2 \end{bmatrix} x_k$.

The closed-loop characteristic polynomial is obtained as:

$$\Delta(z) = z^2 + (0.863k_1 + 0.053k_2 - 0.95)z - 0.707k_1 + 0.026k_2 + 0.111$$

For pole placement design, let $\Delta_{\text{des}}(z) = z^2$. By equating the polynomial coefficients, the deadbeat controller gains are obtained as: $k_1 = 0.501, \ k_2 = 9.702$.

The update rule for controller implementation is given as:

$$u_k = 0.501\ i_k + 9.702\ \omega_k$$

The step response of the deadbeat controller (Figure 10.2.3) settles in two time periods. The response was scaled to match that of the continuous-time system.

An approximate deadbeat design can be performed by choosing distinct closed-loop eigenvalues close to the origin, e.g., $z = \pm 10^{-5}$, and using the 'place' command from the MATLAB Control Systems Toolbox.

The feedback gains for the approximate design are obtained as: $k_1 = 0.509,\ k_2 = 9.702$. The resulting closed-loop system response is still deadbeat.

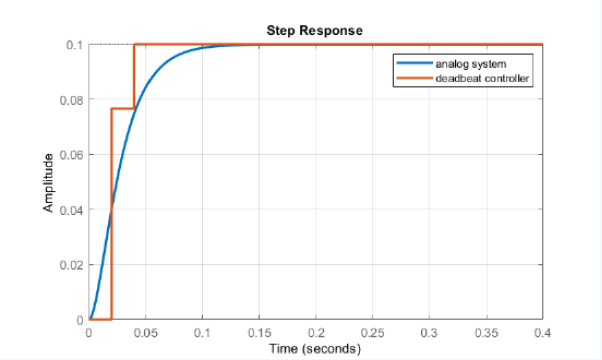


Figure 10.2.3: The step response of the DC motor model with deadbeat controller.

Feedforward Tracking System Design

A tracking system was previously designed by using feedforward cancelation of the error signal (Section 9.2.1). A similar design can be performed in the case of discrete systems.

Towards this end, let the discrete state variable model be given as:

$$\mathbf{x}_{k+1} = \mathbf{A}_\mathrm{d}\mathbf{x}_k + \mathbf{b}_\mathrm{d}u_k,\quad y_k = \mathbf{c}^T\mathbf{x}_k$$

A tracking controller for the model is defined as:

$$u_k = -\mathbf{k}^T\mathbf{x}_k + k_r r_k$$

where $\mathbf{k}^T$ represents a row vector of feedback gains, k_r is a feedforward gain, and r_k is a reference input sequence.

Assuming that a pole placement controller for the discrete system has been designed, the closed-loop system is given as:

$$\mathbf{x}_{k+1} = \left(\mathbf{A}_\mathrm{d} - \mathbf{b}_\mathrm{d}\mathbf{k}^T\right)\mathbf{x}_k + \mathbf{b}_\mathrm{d}k_r r_k$$

The closed-loop pulse transfer function is obtained as:

$$T(z) = \mathbf{c}_\mathrm{d}^T\left(z\mathbf{I} - \mathbf{A}_\mathrm{d} + \mathbf{b}_\mathrm{d}\mathbf{k}^T\right)^{-1}\mathbf{b}_\mathrm{d}k_r$$

where $\mathbf{I}$ denotes an identity matrix. The condition for asymptotic tracking is given as:

$$T(1) = \mathbf{c}_\mathrm{d}^T\left(\mathbf{I} - \mathbf{A}_\mathrm{d} + \mathbf{b}_\mathrm{d}\mathbf{k}^T\right)^{-1}\mathbf{b}_\mathrm{d}k_r = 1$$

The feedforward gain for error cancelation is obtained as: $k_r = \frac{1}{T(1)}$.

Example 10.2.5

The discrete state variable model of a DC motor ($T = 0.02$s) is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix}\begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix} V_k,\quad y_k = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_k \\ \omega_k \end{bmatrix}$$

A state feedback controller for the motor model was previously designed as: $k^T = [k_1,\ k_2]$, where $k_1 = 0.247,\ k_2 = 4.435$.

The closed-loop system is defined as:

$$T(z) = \frac{0.367z + 0.179}{z^2 - 0.503z + 0.05} k_r$$

From the asymptotic condition, the feedforward gain is solved as: $k_r = 6.98$.

The step response of the closed-loop system is shown in Figure 10.2.4.

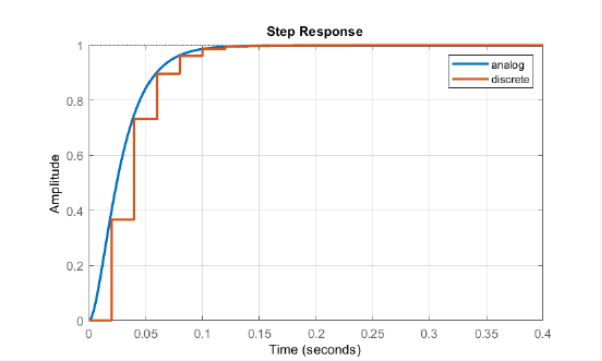


Figure 10.2.4: The step response of the tracking controllers for the DC motor model.

Example 10.2.6

The discrete state variable model of a DC motor ($T = 0.02$s) is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix} V_k, \quad y_k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix}$$

A dead-beat controller for the motor model was designed as: $k^T = [k_1,\ k_2]$, where $k_1 = 0.501,\ k_2 = 9.702$.

The closed-loop system is defined as:

$$T(z) = \frac{0.672z + 0.328}{z^2} k_r$$

From the asymptotic condition, the feedforward gain is solved as: $k_r = 12.77$.

The step response of the closed-loop system is shown in Figure 10.2.5.

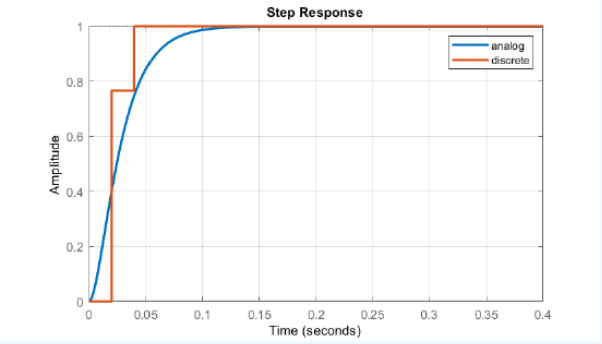


Figure 10.2.5: The step response of the deadbeat tracking controller for the DC motor model.

Tracking PI Controller Design

A tracking PI controller for the discrete state variable model is designed similar to the design of continuous-time system (Figure 9.3.1). The tracking PI controller places an integrator in the feedback loop, thus ensuring that the tracking error goes to zero in the steady-state.

In the case of continuous-time system, the tracking PI controller was defined as: $u = -\mathbf{k}^T\mathbf{x} + k_i \int (r - y)\mathrm{dt}$.

Using the forward difference approximation to the integrator, given as: $v_k = v_{k-1} + Te_k$, an augmented discrete-time system model including the integrator state variable is formed as:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_\mathrm{d} & \mathbf{0} \\ -\mathbf{c}^T T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \mathbf{b}_\mathrm{d} \\ 0 \end{bmatrix} u + \begin{bmatrix} \mathbf{0} \\ T \end{bmatrix} r$$

The state feedback controller for the augmented system is defined as:

$$u(k) = \begin{bmatrix} -\mathbf{k}^T & k_i \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ v(k) \end{bmatrix}$$

where k_i represents the integral gain. With the addition of the above controller, the closed-loop system is described as:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_\mathrm{d} - \mathbf{b}_\mathrm{d} k^T & \mathbf{b}_\mathrm{d} k_i \\ -\mathbf{c}^T T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ T \end{bmatrix} r(k)$$

The closed-loop characteristic polynomial of the augmented system is formed as:

$$\Delta_a(z) = \begin{vmatrix} z\mathbf{I} - \mathbf{A}_\mathrm{d} + \mathbf{b}_\mathrm{d} k^T & -\mathbf{b}_\mathrm{d} k_i \\ -\mathbf{c}_\mathrm{d}^T T & z-1 \end{vmatrix}$$

where $\mathbf{I}$ denotes an identity matrix of order n.

Next, we choose a desired characteristic polynomial of $(n+1)$ order, and perform pole placement design for the augmented system. The location of the integrator pole in the z-plane may be selected keeping in view the desired peformance criteria for the closed-loop system.

Example 10.2.6

The discrete state variable model of a DC motor ($T = 0.02$s) is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 \\ 0.038 & 0.816 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \end{bmatrix} V_k, \quad \omega_k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \end{bmatrix}$$

The control law for the tracking PI controller is defined as:

$$u_k = -k_1 i_k - k_2 \omega_k + k_i v_k$$

where $v_k = v_{k-1} + T(r_k - \omega_k)$ describes the output of the integrator. The augmented system model for the pole placement design using integral control is given as:

$$\begin{bmatrix} i_{k+1} \\ \omega_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 0.134 & -0.038 & 0 \\ 0.038 & 0.816 & 0 \\ 0 & -0.02 & 1 \end{bmatrix} \begin{bmatrix} i_k \\ \omega_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0.863 \\ 0.053 \\ 0 \end{bmatrix} V_k + \begin{bmatrix} 0 \\ 0 \\ 0.02 \end{bmatrix} r_k$$

The desired z-plane pole locations for a desired $\zeta = 0.7$ are selected as: $z = e^{-1},\ e^{-1 \pm j1}$.

The controller gains, obtained using the MATLAB 'place' command, are given as: $k_1 = 0.43, k_2 = 15.44,\ k_i = -297.79.$

An update rule for controller implementation on computer is given as:

$$u_k = -0.43 i_k - 15.44 \omega_k + 297.8 v_k$$

$$v_k = v_{k-1} + 0.02\,(r_k - \omega_k)$$

The step response of the closed-loop system is plotted in Figure 10.2.6. The step response of the continuous-time system (Example 9.1.1) is plotted alongside. The output in both cases attains steady-state value of unity in about 0.12sec.

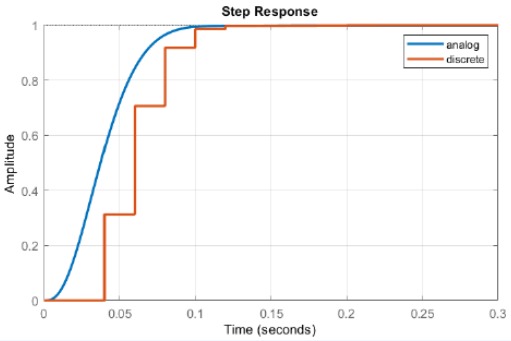


Figure 10.2.6: Tracking PI control of the DC motor model: closed-loop response for the analog and discrete systems.

Index

Detailed Licensing

Overview

Title: Introduction to Control Systems (Iqbal)

Webpages: 80

Applicable Restrictions: Noncommercial

All licenses found:

- CC BY-NC-SA 4.0: 96.3% (77 pages)
- Undeclared: 3.8% (3 pages)

By Page