Chapter 1

Introduction

The objective of this text is to introduce methods for design and analysis of classical feedback control systems. The basic principle of feedback is to:

- Use a sensor to measure the system behavior
- Compare the measured behavior with desired behavior, and
- Take an action based on this comparison.

Section 1.1 motivates the use of feedback control through a variety of applications. Next, Section 1.2 introduces the use of block diagrams to represent feedback control systems consisting of many components. This section also introduces the basic terminology used to describe the key aspects of a control system.

1.1 Applications

Summary: This section introduces a variety of applications to motivate the use of feedback control.

See slides.

1.2 Terminology and Block Diagrams

Summary: This section introduces the use of block diagrams to represent feedback control systems consisting of many components. In addition, basic terminology is introduced to describe the key aspects of a control system.

See slides.

Chapter 2

Modeling

The methods introduced in these notes to analyze and design a control system are primarily for systems modeled by linear ordinary differential equations (ODEs). Section 2.1 first reviews models using both linear and nonlinear ODE. Next, Section 2.2 describes a method to approximate a nonlinear ODE by a related linear ODE. Finally, Section 2.3 summarizes alternative model representations including transfer functions and state-space models.

2.1 Modeling with Ordinary Differential Equations

Summary: Many systems can be modeled by nonlinear ordinary differential equations (ODEs). However, the control design and analysis tools introduced in these notes are primarily for systems modeled by linear ODEs. Linear ODEs are easier to analysis because they satisfy the important principle of superposition.

2.1.1 Linear and Nonlinear ODE Models

Models of physical systems are used for a variety of tasks in the design and analysis of control systems. The models used for control design are typically given in the form of ODEs that describe the effect of an input u on an output y. Systems can have many inputs and outputs in which case u and/or y are vectors. These notes will mainly focus on the case where both u and y are scalars. In this case, the system is called single-input, single-output (SISO). An n^{th} order *linear ODE* model takes the following form:

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^{[m]}(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t) \tag{2.1}$$

Here $y^{[k]}$ denotes the k^{th} derivative with respect to time, i.e. $y^{[k]} := \frac{d^k y}{dt^k}$. To simplify notation, derivatives with respect to time are also occasionally represented using dots: $\dot{y} := y^{[1]}$, $\ddot{y} := y^{[2]}$, etc. The coefficients $\{a_0, \ldots, a_n, b_0, \ldots, b_m\}$ are constants that are selected to model the dynamics of a given system. We can assume without loss of generality that both $a_n \neq 0$ and $b_m \neq 0$. Moreover, we typically assume the system is *proper* in the sense that $m \leq n$. The system is called *strictly proper* if $m < n$.

An n^{th} order ODE requires n initial conditions (ICs) to complete the model. These are typically specified by the initial value of y and its first $(n-1)$ derivatives:

$$y(0) = y_0;\ \dot{y}(0) = \dot{y}_0;\ \ldots;\ y^{[n-1]}(0) = y_0^{[n-1]} \tag{2.2}$$

If the input u and initial conditions are given then the ODE can be solved for the output y.[*]

Equation 2.1 is an *ordinary differential equation* because it involves functions (and their derivatives) that depend on a single variable, time t. This is in contrast to partial differential equations (PDEs) which involve functions of more than one variable. For example, Maxwell's equations and the Navier-Stokes equations describe electromagnetic fields and fluid motion, respectively. These two PDEs involve functions (and their partial derivatives) that depend on space and time. These notes will focus on ODEs and systems modeled by PDEs will not be considered here.

Equation 2.1 is also *linear* as it only involves linear combinations of y, u, $\dot{y}$, etc. In other words, nonlinear terms like $u^2(t)$ or $\sin(y(t))$ do not appear. Many systems are more accurately described by n^{th} order *nonlinear ODEs* of the following form:

$$y^{[n]}(t) = f(y(t), \dot{y}(t), \ldots, y^{[n-1]}(t), u(t), \dot{u}(t), \ldots, u^{[m]}(t)) \tag{2.3}$$

[*] In general, certain technical conditions must be satisfied for the solution to exist and be unique. These conditions can be found in standard textbooks on ODEs [22]. We assume in these notes that the conditions for existence and uniqueness are satisfied.

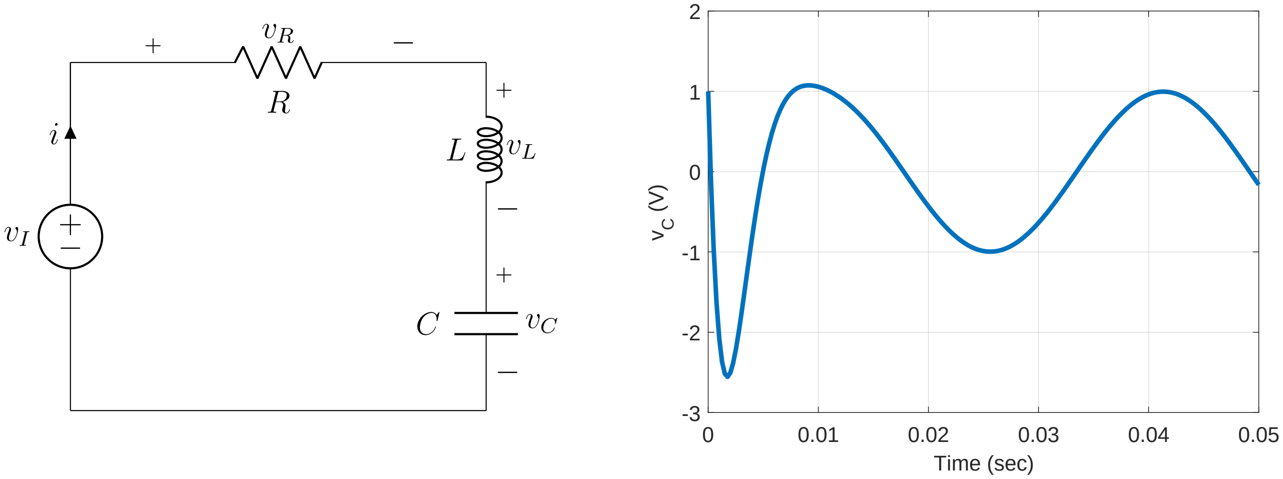


Figure 2.1: RLC circuit (left) and capacitor voltage response (right)

The function $f : \mathbb{R}^{n+m+1} \to \mathbb{R}$ is selected to model the (nonlinear) dynamics of a given system. Nonlinear ODEs can be specified in more general forms, e.g. they can be nonlinear in $y^{[n]}$, but Equation 2.3 will be sufficient for our purposes. Nonlinear ODEs are typically higher fidelity (more accurate) models used for simulation. However, the control design and analysis tools presented in these notes will be almost exclusively for models described by linear ODEs. Thus we'll need a method to approximate a nonlinear ODE by a related linear ODE. This will be covered in Section 2.2.

Examples of linear and nonlinear ODE models are provided next for a few simple systems. Note that modeling is a domain-specific endeavor. In other words, each field of engineering applies certain physical laws to model specific classes of systems, e.g. Kirchoff's laws for electrical circuits and Newton's laws for the motion of mechanical systems. Constructing models using basic laws of physics is known as *first-principles* modeling. Alternatively, the field of *system identification* [12] considers techniques to construct models from experimental data. The term *black-box modeling* refers to models constructed only using experimental input-output data from the system. This will be discussed further in Section 5.7. In *grey-box modeling* the form of the model is given by first-principles but the model parameters are determined using experimental data. The important point is that many systems can ultimately be modeled (for control design) using ordinary differential equations. As a result, the tools described in these notes can be applied to design control systems for a wide variety of engineering applications.

Example 2.1. Consider the RLC circuit shown in Figure 2.1 (left). The resistor, capacitor, and inductor values are $R = 10\Omega$, $C = 2 \times 10^{-4}F$, and $L = 0.01H$. Assume the components are ideal so that $v_R = iR$, $\dot{v}_C = \frac{i}{C}$, and $v_L = L\dot{i}$ where the voltages are in units of Volts (V) and the current is in Amperes (A). The voltages around the circuit sum to zero by Kirchoff's voltage law. Hence using the ideal component relationships yields:

$$LC\,\ddot{v}_C(t) + RC\,\dot{v}_C(t) + v_C(t) = v_I(t) \tag{2.4}$$

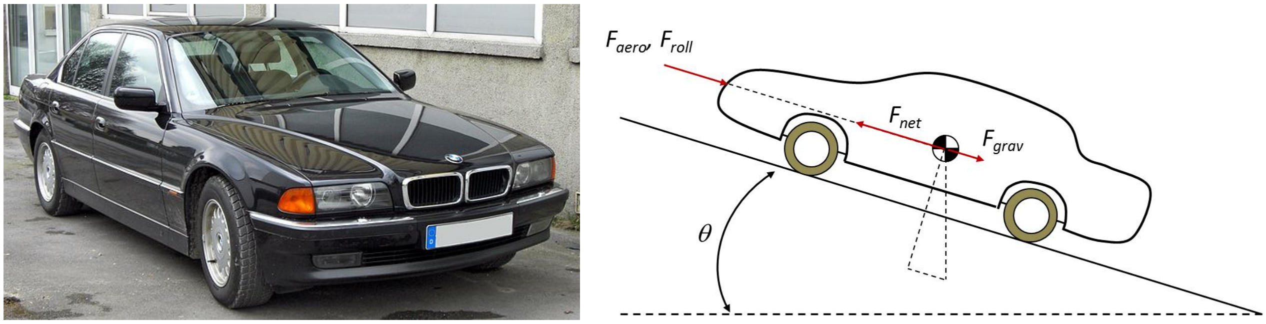


Figure 2.2: BMW 750iL [13] (left) and free body diagram (right)

This is a second order linear ODE that models the dynamics with the voltage source as input ($u := v_I$) and voltage across the capacitor as the output ($y := v_C$). Using notation from Equation 2.1, the coefficients of this linear ODE are $a_2 = LC = 2 \times 10^{-6} sec^2$, $a_1 = RC = 2 \times 10^{-3} sec$, $a_0 = 1$, and $b_0 = 1$. This is a first-principles model since it is based only on Kirchoff's voltage law and the ODE coefficients (a_2, a_1, a_0, and b_0) can be determined directly from the selected components. To complete the model we must specify initial conditions. If $v_{C,0} = 1V$ and $i_0 = -1A$ are the initial capacitor voltage and circuit current then $v_C(0) = 1V$ and $\dot{v}_C(0) = \frac{i_0}{C} = -5000 \frac{V}{sec}$. Figure 2.1 (right) shows the response of Equation 2.4 with these initial conditions and input voltage $v_I(t) = \sin(200t)$. △

Example 2.2. Again consider the RLC circuit shown in Figure 2.1 with the voltage source as input ($u := v_I$). However, now let the output be given by the voltage across the resistor ($y := v_R$). A second-order linear ODE also models these input/output dynamics:

$$LC\, \ddot{v}_R(t) + RC\, \dot{v}_R(t) + v_R(t) = RC\, \dot{v}_I(t) \tag{2.5}$$

This model is obtained by differentiating both sides of Equation 2.4 and then substituting $\dot{v}_C = \frac{1}{RC} v_R$. The coefficients of this linear ODE are $a_2 = LC$, $a_1 = RC$, $a_0 = 1$, $b_1 = RC$, and $b_0 = 0$. Note that the derivative of the input signal appears in this simple model. △

Example 2.3. Figure 2.2 shows a BMW 750iL (left) and a free-body diagram of the forces acting on the car (right). Let v denote the longitudinal velocity of the car in $\frac{m}{sec}$. The key forces along the direction of vehicle motion consist of:

1. Gravitational force, F_{grav}: If the car is moving up a slope of angle $\theta(t)$ (in $rads$) then this force is $mg\sin(\theta(t))$. Here $g = 9.81\ \frac{m}{sec^2}$ is the gravitational constant and $m = 2,085kg$ is the vehicle mass (without passengers).

2. Rolling resistance, F_{roll}: This force is due to friction at the interface of the tire and road.

3. Aerodynamic drag, F_{aero}: This force is modeled as $c_D v^2$ where c_D is the drag coefficient.

4. Engine/brake force, F_{net}: The forces generated to accelerate or decelerate the car are complicated to model. They include engine combustion, drivetrain, brake hydraulics, and tire/road slip dynamics. For simplicity, we'll neglect these complications and simply treat the net engine/brake force F_{net} as a control input that can be generated through proper control of the throttle and brakes.

By Newton's second law, the longitudinal motion of the car is modeled by the following first-order, nonlinear ODE:

$$\dot{v}(t) = \frac{1}{m}\left(F_{net}(t) - c_D v^2(t) - F_{roll} - mg\sin(\theta(t))\right) \tag{2.6}$$

Note that this model is only valid for $v > 0$, e.g. rolling resistance and drag would have the opposite sign for a car moving in reverse. Additional details on vehicle modeling can be found in [23]. The parameters $c_D = 0.4\frac{N\cdot sec^2}{m^2}$ and $F_{roll} = 228N$ were obtained from coast-down experiments [9]. Equation 2.6 is a grey-box model since the form of the model is from first-principles (Newton's laws) but experiments were used to obtain some parameters. The velocity is the output ($y := v$) and net engine/brake force is the controllable input ($u := F_{net}$). The road slope $\theta(t)$ changes with time and is a disturbance input acting on the vehicle. Hence this model has two inputs and one output, i.e. it is a multiple-input, single-output model. The car velocity can be solved for a given initial velocity, net engine/brake force, and slope. $\triangle$

To summarize, the dynamics of a system can often be modeled by nonlinear or linear ODEs. It is important to note that such models are only an approximation of the real system behavior. The model always involves some inaccuracies. For control system design we need the simplest model that captures the essential dynamics. In addition, our design and analysis must account for the impact of model errors.

2.1.2 Principle of Superposition

An important fact is that linear ODEs satisfy the principle of superposition. In particular, assume that $y_1(t)$ is the solution of the linear ODE (Equation 2.1) with input $u_1(t)$ and zero IC. In addition, assume $y_2(t)$ is the solution with input $u_2(t)$ and zero IC. Then the linear ODE satisfies the following two properties:

- **Scaling:** For any constant $c \in \mathbb{R}$, the solution of the linear ODE with input $u_S(t) = c\,u_1(t)$ and zero IC is given by $y_S(t) = c\,y_1(t)$.
- **Additivity:** The solution of the linear ODE with input $u_A(t) = u_1(t) + u_2(t)$ and zero IC is given by $y_A(t) = y_1(t) + y_2(t)$.

These superposition properties are key to the easy analysis of linear ODE models. Nonlinear ODEs do not satisfy these properties and, as a result, they are more challenging to analyze.

Example 2.4. As a simple example, consider the linear ODE $\dot{y}(t) + 2y(t) = 4u(t)$. The left subplot of Figure 2.3 shows the response of this system $y_1(t)$ with zero IC and input $u_1(t) = 1$ (blue dashed). It also shows the response $y_S(t)$ with zero IC and input $u_S(t) = 2u_1(t)$ (red solid). By the principle of superposition scaling property, the results are related by $y_S(t) = 2y_1(t)$. The right subplot of Figure 2.3 shows the responses $y_1(t)$ and $y_2(t)$ to zero IC and inputs $u_1(t) = 1$ and $u_2(t) = \sin(3t)$ (blue dashed). It also shows the response $y_A(t)$ with zero IC and input $u_A(t) = u_1(t) + u_2(t)$ (red solid). By the principle of superposition additive property, the results are related by $y_A(t) = y_1(t) + y_2(t)$.

$\triangle$

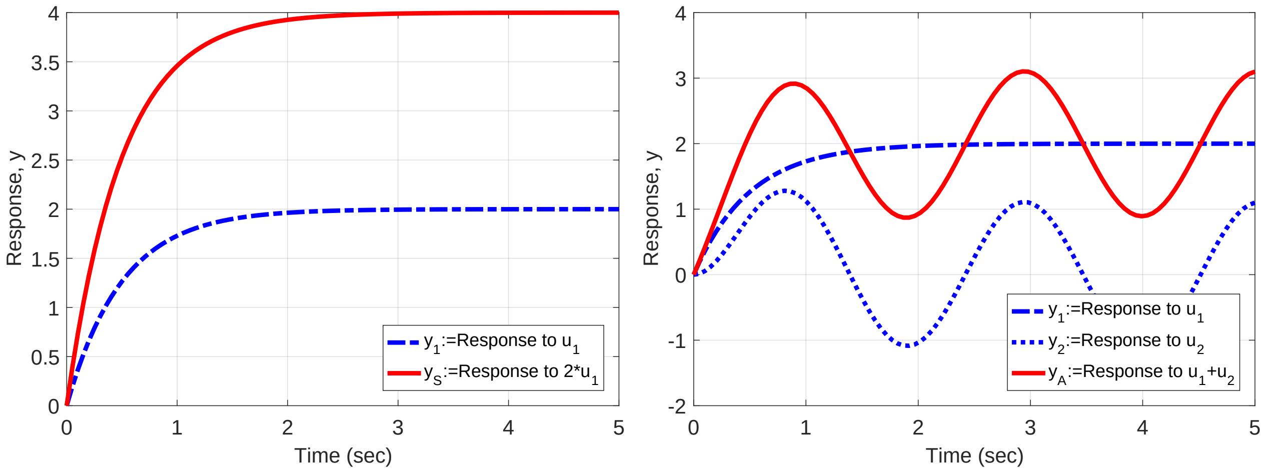


Figure 2.3:
(Left) Responses $y_1(t)$ and $y_S(t)$ to inputs $u_1(t) = 1$ and $u_S(t) = 2u_1(t)$ with zero IC.
(Right) Responses $y_1(t)$, $y_2(t)$, and $y_A(t)$ to inputs $u_1(t) = 1$, $u_2(t) = \sin(3t)$, and $u_A(t) = u_1(t) + u_2(t)$ with zero IC.

2.2 Equilibrium Points and Linearization

Summary: An equilibrium point is essentially a constant solution to a nonlinear ODE. A nonlinear ODE can be approximated as a linear ODE near an equilibrium point by Jacobian linearization. A nonlinear system can have many equilibrium points and each equilibrium point can have a different linear ODE approximation.

Consider an n^{th} nonlinear ODE of the following form:

$$y^{[n]}(t) = f(y(t), \dot{y}(t), \ldots, y^{[n-1]}(t), u(t), \dot{u}(t), \ldots, u^{[m]}(t)) \tag{2.7}$$

An *equilibrium point* consists of (constant) values $\bar{y} \in \mathbb{R}$ and $\bar{u} \in R$ such that

$$0 = f(\bar{y}, 0, \ldots, 0, \bar{u}, 0, \ldots, 0) \tag{2.8}$$

This is called an equilibrium point because if the input is held constant at $u(t) = \bar{u}$ for all $t \geq 0$ and the initial conditions are specified as $y(0) = \bar{y}$ and $\dot{y}(0) = \cdots = y^{[n-1]}(0) = 0$ then the solution of Equation 2.7 is $y(t) = \bar{y}$ for all $t \geq 0$. Finding an equilibrium point $(\bar{y}, \bar{u})$ is often called "trimming" the system. In this case Equation 2.8 must be solved for two unknowns. Hence the equilibrium point is typically not unique since there are fewer equations than unknowns.

Example 2.5. Example 2.3 introduced a model for the BMW 750iL. If the road is level $(\theta(t) = 0 rads)$ then this model simplifies to the following nonlinear ODE:

$$\begin{aligned} \dot{v}(t) &= \frac{1}{m}\left(F_{net}(t) - c_D v^2(t) - F_{roll}\right) := f(v(t), F_{net}(t)) \qquad (2.9)\\ \text{IC: } & v(0) = v_0 \end{aligned}$$

The function $f : \mathbb{R}^2 \to \mathbb{R}$ describes the dynamics for the system with input F_{net} and output v. By definition, an equilibrium point $(\bar{v}, \bar{F}_{net})$ for the car satisfies $f(\bar{v}, \bar{F}_{net}) = 0$. For example, if $\bar{F}_{net} = 400N$ is given then $f(\bar{v}, \bar{F}_{net}) = 0$ can be solved for the equilibrium velocity. This yields $\bar{v} = 20.7\frac{m}{sec}$ $(\approx 46\frac{miles}{hr})$. Physically this means that if the initial velocity is $v(0) = \bar{v}$ and the input force is maintained at $F_{net}(t) = \bar{F}_{net}$ for all $t \geq 0$ then the car will remain at the velocity $v(t) = \bar{v}$ for all $t \geq 0$. Alternatively, if $\bar{v} = 29\frac{m}{sec}$ $(\approx 65\frac{miles}{hr})$ is given then we can solve for $\bar{F}_{net} = 564N$. Note that the equilibrium point is not unique. The equilibrium velocity can be solved for a given input force or the required input force can be solved for a given velocity.

△

If the solution of a nonlinear ODE remains "near" an equilibrium point then the dynamics can be approximated by a linear ODE. The precise steps of this approximation are known as *Jacobian linearization*. This is extremely useful because many tools for control design are applicable for linear ODE models. Moreover, many controllers are designed to keep a system near a particular equilibrium point. Hence, the controller, if properly designed, will ensure the linear approximation is valid. Jacobian linearization uses the *Taylor series* expansion to approximate the dynamics of a nonlinear system near an equilibrium point. Taylor series is typically introduced in Calculus and is briefly reviewed in Appendix 2.4.1. For simplicity, the Jacobian linearization process is initially described for a first-order, nonlinear ODE of the following form:

$$\dot{y}(t) = f(y(t), u(t)) \tag{2.10}$$

where $f : \mathbb{R}^2 \to \mathbb{R}$ is a given function. The general case for n^{th} order nonlinear ODEs is given at the end of the section. Assume $(\bar{y}, \bar{u})$ is an equilibrium point, i.e. $f(\bar{y}, \bar{u}) = 0$. We know that if the system is initialized at $y(0) = \bar{y}$ and the input is held constant at $u(t) = \bar{u}$ for all $t \geq 0$ then the solution of Equation 2.10 stays at $y(t) = \bar{y}$ for all $t \geq 0$. Jacobian linearization is used to approximate the solution $y(t)$ to the nonlinear ODE when $y(0)$ is slightly different from $\bar{y}$ and/or the input $u(t)$ is slightly different from $\bar{u}$. The first step is to define variables that measure the deviation of the nonlinear solution (y, u) from the equilibrium point $(\bar{y}, \bar{u})$:

$$\delta_y(t) := y(t) - \bar{y} \tag{2.11}$$

$$\delta_u(t) := u(t) - \bar{u} \tag{2.12}$$

Note that $\dot{\delta}_y(t) = \dot{y}(t)$ because $\bar{y}$ is a constant. Hence the nonlinear ODE in Equation 2.10 can be rewritten in terms of these new deviation variables as:

$$\dot{\delta}_y(t) = f(\bar{y} + \delta_y(t), \bar{u} + \delta_u(t)) \tag{2.13}$$

Next perform a multi-variable Taylor series expansion around $(\bar{y}, \bar{u})$ to obtain a linear approximation for the nonlinear function f:

$$f(\bar{y} + \delta_y(t), \bar{u} + \delta_u(t)) \approx f(\bar{y}, \bar{u}) + \frac{\partial f}{\partial y}(\bar{y}, \bar{u}) \cdot \delta_y + \frac{\partial f}{\partial u}(\bar{y}, \bar{u}) \cdot \delta_u \tag{2.14}$$

We have dropped the higher order terms (quadratics, etc) in the expansion. The function f has two arguments and hence the Taylor series requires the partial derivatives of f with respect to both y and u. Moreover, note that $f(\bar{y}, \bar{u}) = 0$ because $(\bar{y}, \bar{u})$ is assumed to be an equilibrium point. Thus substituting Equation 2.14 into Equation 2.13 yields a first-order *linear ODE*:

$$\dot{\delta}_y(t) + a_0 \delta_y(t) = b_0 \delta_u(t) \tag{2.15}$$

where $b_0 := \frac{\partial f}{\partial u}(\bar{y}, \bar{u})$ and $a_0 := -\frac{\partial f}{\partial y}(\bar{y}, \bar{u})$. Note the sign convention in the definition of a_0 is chosen so that the linear ODE approximation is in the standard form introduced in Section 2.1. The solution of this linear ODE will approximate the solution of the nonlinear ODE (Equation 2.10) as long as $y(t)$ remains "near" $\bar{y}$ and $u(t)$ remains "near" $\bar{u}$, i.e. as long as both $\delta_y(t)$ and $\delta_u(t)$ have "small" magnitudes.

Example 2.6. The BMW 750iL on a level road is modeled with the following nonlinear ODE:

$$\dot{v}(t) = \frac{1}{m}\left(F_{net}(t) - c_D v^2(t) - F_{roll}\right) := f(v(t), F_{net}(t)) \tag{2.16}$$

$$\text{IC: } v(0) = v_0$$

As shown in Example 2.5, this model has an equilibrium point at $(\bar{v}, \bar{F}_{net}) = (20.74\frac{m}{sec}, 400N)$. The left subplot of Figure 2.4 shows vehicle velocity (solid blue line) computed from the nonlinear ODE with initial condition $v(0) = 19.74\frac{m}{sec}$ and input $F_{net}(t) = 400 - 140\sin(2t)$. A Jacobian linearization for this model requires two partial derivatives:

$$\frac{\partial f}{\partial v}(\bar{v}, \bar{F}_{net}) = \left.\frac{-2c_D\bar{v}}{m}\right|_{(\bar{v},\bar{F}_{net})} = -0.008\frac{1}{sec}$$

$$\frac{\partial f}{\partial F_{net}}(\bar{v}, \bar{F}_{net}) = \left.\frac{-1}{m}\right|_{(\bar{v},\bar{F}_{net})} = 4.8 \times 10^{-4}\frac{1}{kg}$$

where $m = 2,085kg$ and $c_D = 0.4\frac{N\cdot sec^2}{m^2}$ as introduced in Example 2.3. This yields the Jacobian linearization for the BMW:

$$\dot{\delta}_v(t) + 0.008\delta_v(t) = (4.8 \times 10^{-4})\,\delta_F(t) \tag{2.17}$$

The solution of this linear ODE can be used to approximate the solution of the nonlinear ODE. For example, consider the initial condition and input force specified above for the nonlinear ODE. This is equivalent to an initial condition $\delta_v(0) = -1\frac{m}{sec}$ and input $\delta_F(t) = -140\sin(2t)$ for the linear ODE. The linear ODE can be solved (analytically in this case) to obtain $\delta_v(t)$. This yields the approximate velocity $\delta_v(t) + \bar{v}$ which is shown as the red dashed curve in Figure 2.4. The solution to the linear ODE is almost indistinguishable from the solution of the nonlinear ODE. The approximation is accurate because the velocity remains near the equilibrium value $\bar{v}$. However, the approximation is not quite as good for velocities further from the equilibrium value. For example, the right subplot of Figure 2.4 shows the solution of both the nonlinear ODE and the linear ODE for the same input force but with initial condition $v(0) = -10.74\frac{m}{sec}$. There is a noticeable difference in the responses because the velocity is far from the equilibrium velocity used to construct the linearization. The linear ODE is constructed around a specific equilibrium point. A Jacobian linearization constructed near the velocity $-10.74\frac{m}{sec}$ would provide a better approximation for the nonlinear response shown in the right subplot of Figure 2.4.

△

Next consider the more general case of an n^{th} order nonlinear ODE:

$$y^{[n]}(t) = f(y(t), \dot{y}(t), \ldots, y^{[n-1]}(t), u(t), \dot{u}(t), \ldots, u^{[m]}(t)) \tag{2.18}$$

The Jacobian linearization process is similar in this case but with additional notation. Specifically, let $(\bar{y}, \bar{u})$ be an equilibrium point. Then Jacobian linearization yields a linear ODE approximation of the form:

$$\delta_y^{[n]}(t) + a_{n-1}\delta_y^{[n-1]}(t) + \cdots + a_1\dot{\delta}_y(t) + a_0\delta_y(t) = b_m\delta_u^{[m]}(t) + \cdots + b_1\dot{\delta}_u(t) + b_0\delta_u(t) \tag{2.19}$$

where $\delta_y(t) := y(t) - \bar{y}$ and $\delta_u(t) := u(t) - \bar{u}$ are the deviations of the output and input from the equilibrium point. Moreover, the coefficients of the linear ODE are given by the following

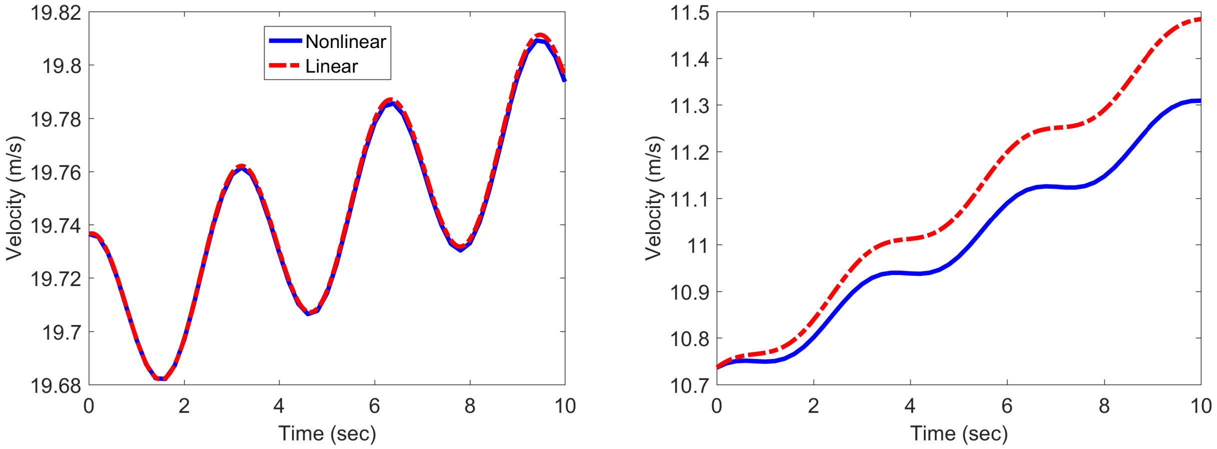


Figure 2.4: BMW output velocity for $F_{net}(t) = 400 - 140\sin(2t)N$ with initial conditions $v(0) = 19.7\frac{m}{sec}$ (left subplot) and $v(0) = 10.7\frac{m}{sec}$ (right subplot).

partial derivatives:

$$\begin{aligned}
a_0 &= -\frac{\partial f}{\partial y}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) & b_0 &= \frac{\partial f}{\partial u}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \\
a_1 &= -\frac{\partial f}{\partial y^{[1]}}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) & b_1 &= \frac{\partial f}{\partial u^{[1]}}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \\
&\vdots & &\vdots \\
a_{n-1} &= -\frac{\partial f}{\partial y^{[n-1]}}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) & b_m &= \frac{\partial f}{\partial u^{[m]}}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0)
\end{aligned}$$

Again, the coefficients $(a_0, a_1, \dots, a_{n-1})$ are defined as the negative of the corresponding partial. This sign convention yields a linear ODE in our standard form (Equation 2.19) with all terms involving y and its derivatives on the left side. It is important to emphasize that nonlinear systems can, in general, have many equilibrium points. The Jacobian linearization is performed at a particular equilibrium point and different linear ODE approximations are obtained at each equilibrium point.

To summarize, Jacobian linearization is used to construct a linear ODE at an equilibrium point $(\bar{y}, \bar{u})$. Let $y(t)$ denote the solution of the nonlinear ODE to an input $u(t)$ with initial condition $y(0)$. The linear ODE has an an equivalent input $\delta_u(t) = u(t) - \bar{u}$ and initial condition $\delta_y(0) = y(0) - \bar{y}$. The solution of the linear ODE is $\delta_y(t)$ and this yields an approximation for the nonlinear solution (assuming small deviations from equilibrium) as $y(t) \approx \delta_y(t) + \bar{y}$.

2.3 Alternative Model Representations

Summary: This section reviews three alternative model representations. First, the transfer function is introduced as an alternative notation for an n^{th} order linear ODE. Second, linear state-space models are a set of n coupled, first-order ODEs and can be used to represent an n^{th} order linear ODE. Third, nonlinear state-space models are briefly introduced.

2.3.1 Transfer Functions

An n^{th}-order linear ODE is given by:

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^{[m]}(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t) \tag{2.20}$$

The <u>*transfer function*</u> for this ODE is defined as:

$$G(s) := \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{2.21}$$

At this point Equation 2.21 is merely a new notation for the ODE, i.e. the transfer function is simply a different way of representing the ODE. We will see later that the transfer function has several uses beyond simply being another notation for the ODE. The function `tf` can be used to create a transfer function in Matlab. The syntax is `G=tf(num,den)` where `num` is $1 \times m$ row vector of numerator coefficients and `den` is a $1 \times n$ row vector of denominator coefficients. Additional help and documentation can be found by typing "`help tf`" and "`doc tf`" at the command line, respectively. The function `tfdata` can be used to extract the numerator and denominator coefficients from a transfer function. A simple example is given below.

Example 2.7. Consider the second-order ODE:

$$6\ddot{y}(t) + 9\dot{y} + 2y = 4\dot{u} + 8u \tag{2.22}$$

The transfer function for this ODE is given by $G(s) = \frac{4s+8}{6s^2+9s+2}$. This transfer function can be created in Matlab with the `tf` function:

```
>> G=tf([4 8],[6 9 2])
G =
       4 s + 8
  ---------------
  6 s^2 + 9 s + 2
Continuous-time transfer function.
>> [num,den]=tfdata(G);
>> num{1}
ans =
     0     4     8
>> den{1}
ans =
     6     9     2
```

Note that `tfdata` returns the coefficients `num` and `den` as cell arrays. The syntax `num{1}` and `den{1}` extracts the actual row vectors of coefficients from the cell array. △

The Laplace Transform can be used to make a formal connection between the ODE and its transfer function. This connection is briefly discussed in Appendix 2.4.2. However, the Laplace Transform is not required in the remainder of these notes and hence Appendix 2.4.2 can be skipped with no loss of continuity.

2.3.2 Linear State-Space

The control design tools developed in these notes primarily use ODE and transfer function models. This is commonly referred to as "classical control" design. Alternatively, the system dynamics can be described by a state-space model as defined below. The use of state-space models leads to an alternative set of control design tools commonly referred to as "modern" or "state-space" design. State-space models are only briefly introduced here and details on state-space design can be found in [4,8,10].

An n^{th} order *linear state-space model* with input u and output y takes the form:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\\ \text{IC: } &x(0) = x_0\end{aligned} \tag{2.23}$$

Here $x \in \mathbb{R}^n$ is an n-dimensional vector known as the *state*. The model dynamics are defined by the matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, $C \in \mathbb{R}^{1\times n}$, and $D \in \mathbb{R}$. There is a non-uniqueness in the state-space representation. In other words, there are different choices for (A, B, C, D) that represent the same dynamics from input u to output y [†]. State-space models can handle multiple-input, multiple-output systems with only minor notational changes. Thus, u and y can, in general, be vectors although this discussion will focus on the case where they are scalars. The model is completed with a single, vector-valued initial condition $x(0) = x_0 \in \mathbb{R}^n$. Equation 2.23 expresses the dynamics as a first-order, vector differential equation, i.e. it is a set of n first-order ODEs that are coupled together by (A, B, C, D).

The n^{th} order linear ODE in Equation 2.20 can always be written in state-space form. This is easiest to see when the derivatives of the input u do not appear in the ODE. Specifically, consider the following form for the ODE:

$$\begin{aligned}&y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0 y(t) = b_0 u(t)\\ &\text{IC: } y(0) = y_0;\ \dot{y}(0) = \dot{y}_0;\ \ldots;\ y^{[n-1]}(0) = y_0^{[n-1]}\end{aligned} \tag{2.25}$$

[†] Define a new set of state variables $z := Tx$ where T is a nonsingular $n \times n$ matrix. Then dynamics from input u to output y can be equivalently represented with the state z:

$$\begin{aligned}\dot{z}(t) &= \left(TAT^{-1}\right) z(t) + (TB)u(t)\\ y(t) &= \left(CT^{-1}\right) z(t) + Du(t)\end{aligned} \tag{2.24}$$

The coefficient of $y^{[n]}$ has been normalized ($a_n = 1$) to simplify the notation. This normalization can be done by simply dividing both sides of the ODE by a_n. Define the state variables $x_1 := y$, $x_2 := \dot{y}$, …, $x_n := y^{[n-1]}$. The first $n-1$ state variables satisfy the simple relations $\dot{x}_1(t) = x_2(t)$, $\dot{x}_2(t) = x_3(t)$, etc. Moreover, the linear ODE can be used to express $\dot{x}_n(t) = y^{[n]}(t)$ in terms of the state variables and input. As a result, the linear ODE in Equation 2.25 can be expressed in state-space form with the following state-matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_0 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad D = 0$$

The initial condition for the state-space model is $x(0) = \left[y_0,\, \dot{y}_0, \ldots, y_0^{[n-1]} \right]^T$. An n^{th}-order ODE (Equation 2.20) that contains derivatives of the input u can also be written in state-space form. The state matrices are more complicated in this case and details can be found in [6, 16, 18]. The reverse direction also holds: a state-space model can be converted to an equivalent n^{th}-order linear ODE. This conversion is discussed in Appendix 2.4.3.

The function `ss` can be used to create a transfer function in Matlab. The syntax is `G=ss(A,B,C,D)` where `A`, `B`, `C`, and `D` have appropriate dimensions. Additional help and documentation can be found by typing "`help ss`" and "`doc ss`" at the command line. The function `ssdata` can be used to extract the state-space matrices. The functions `ss` and `tf` can also be used to convert between state-space and transfer function representations. A simple example is given below.

Example 2.8. Consider the third-order ODE:

$$\begin{aligned} & y^{[3]}(t) + 8\ddot{y}(t) + 9\dot{y}(t) + 2y(t) = -3u(t) \\ & \text{IC: } y(0) = y_0;\ \dot{y}(0) = \dot{y}_0;\ \ddot{y}(0) = \ddot{y}_0 \end{aligned} \tag{2.26}$$

Define the variables $x_1 := y$, $x_2 := \dot{y}$, and $x_3 := \ddot{y}$. The single third-order ODE can be re-written as three coupled first-order ODEs: $\dot{x}_1(t) = x_2(t)$, $\dot{x}_2(t) = x_3(t)$, and

$$\dot{x}_3(t) = -2x_1(t) - 9x_2(t) - 8x_3(t) - 3u(t). \tag{2.27}$$

These three first-order ODEs can be compactly expressed as a vector, first-order ODE:

$$\begin{aligned} & \dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -9 & -8 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ -3 \end{bmatrix} u(t) \\ & y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) \\ & \text{IC: } x(0) = \begin{bmatrix} y_0 & \dot{y}_0 & \ddot{y}_0 \end{bmatrix}^T \end{aligned} \tag{2.28}$$

This state-space model is constructed in the Matlab code below.

```
>> A=[0 1 0; 0 0 1; -2 -9 -8];
>> B=[0;0;-3]; C=[1 0 0]; D=0;
>> G=ss(A,B,C,D);

% Comment: tf() converts G from SS to TF form. Note that we
% recover the TF for the original 3rd-order ODE.
>> tf(G)
ans =
          -3
  ---------------------
  s^3 + 8 s^2 + 9 s + 2

% We can also construct the original TF and convert from TF to SS.
>> G2 = tf(-3,[1 8 9 2]);    % Construct original TF
>> G3=ss(G2);                % ss() converts G2 from TF to SS form

% Note that A3 is not the same as A given above. This is due to
% the non-uniqueness of state-space models, i.e. both G and G3
% represent the same dynamics but with different state matrices.
>> [A3,B3,C3,D3]=ssdata(G3);
>> A3
A3 =
   -8.0000   -2.2500   -0.5000
    4.0000         0         0
         0    1.0000         0
```

△

2.3.3 Nonlinear State-Space

The linear state-space models introduced in the previous section can be generalized to nonlinear dynamics. Specifically, an n^{th} order *nonlinear state-space model* with input u and output y takes the form:

$$\begin{aligned} &\dot{x}(t) = f(x(t), u(t)) \\ &y(t) = h(x(t), u(t)) \\ &\text{IC: } x(0) = x_0 \end{aligned} \tag{2.29}$$

Here $x \in \mathbb{R}^n$ is an n-dimensional vector known as the *state*. The model dynamics are defined by the functions $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ and $h : \mathbb{R}^{n+1} \to \mathbb{R}$. Again, u and y can, in general, be vectors although the notation here is for the case where they are scalars. The model is completed with a single, vector-valued initial condition $x(0) = x_0 \in \mathbb{R}^n$. The Jacobian linearization presented in Section 2.2 can be extended to these nonlinear state-space models. This generalization mainly involves additional notation and is not needed in the remainder of these notes. Details on this extension of Jacobian linearization are provided in Appendix 2.4.4.

2.4 Appendix: Background and Additional Results

Summary: This appendix first provides a review of Taylor series expansion. Next the Laplace transform is defined and used to formally connect the ODE and its associated transfer function. This is followed by a discussion of the steps required to convert from a state-space model to a transfer function / ODE model. Finally, Jacobian linearization for nonlinear state-space models is briefly described.

2.4.1 Taylor Series

To start, first consider a function of one variable: $f : \mathbb{R} \to \mathbb{R}$. The Taylor series of f at a point $\bar{x} \in \mathbb{R}$ is given by:

$$f(x) = f(\bar{x}) + \frac{df}{dx}(\bar{x}) \cdot (x - \bar{x}) + \text{Higher Order Terms (Quadratic, etc)} \tag{2.30}$$

If x is "near" $\bar{x}$ then the higher order terms can be neglected. This yields a linear function that approximates f near $\bar{x}$:

$$f(x) \approx f(\bar{x}) + \frac{df}{dx}(\bar{x}) \cdot (x - \bar{x}) \tag{2.31}$$

The error in making this linear approximation is on the order of $(x - \bar{x})^2$. Here the linear approximation passes through the equilibrium point $(\bar{x}, f(\bar{x})$ with slope $\frac{df}{dx}(\bar{x})$.

Example 2.9. Consider the quadratic drag term $f(v) = c_D v^2$ that appears in the vehicle model with $c_D = 0.4 \frac{N \cdot sec^2}{m^2}$. This function is shown Figure 2.5 below as the blue solid line. The linear Taylor series approximation near $\bar{v} = 29 \frac{m}{sec}$ is:

$$\begin{aligned} c_D v^2 &\approx c_D \bar{v}^2 + (2 c_D \bar{v}) \cdot (v - \bar{v}) \\ &= 336.4N + \left(23.2 \frac{N \cdot sec}{m}\right) \cdot \left(v - 29 \frac{m}{sec}\right) \end{aligned} \tag{2.32}$$

The linear approximation is also shown in the figure (red dashed). The linear Taylor series approximates the nonlinear drag for velocities near $\bar{v} = 29 \frac{m}{sec}$. For example, if $v = 30 \frac{m}{sec}$ then the actual drag is $c_D v^2 = 360N$. The linear Taylor series gives the approximate drag of $359.6N$ $(= 336.4 + 23.2 \times 1)$. The approximation is accurate since v is near $\bar{v}$. If we instead select $v = 10 \frac{m}{sec}$ then the actual drag is $c_D v^2 = 40N$. The linear Taylor series gives the approximate drag of $-104.4N$ $(= 336.4 + 23.2 \times -19)$. The approximation is quite poor in this case since v is far from $\bar{v}$. In fact, the linear Taylor series yields a negative (non-physical) value for drag. △

Additional details on Taylor Series can be found in most Calculus textbooks.

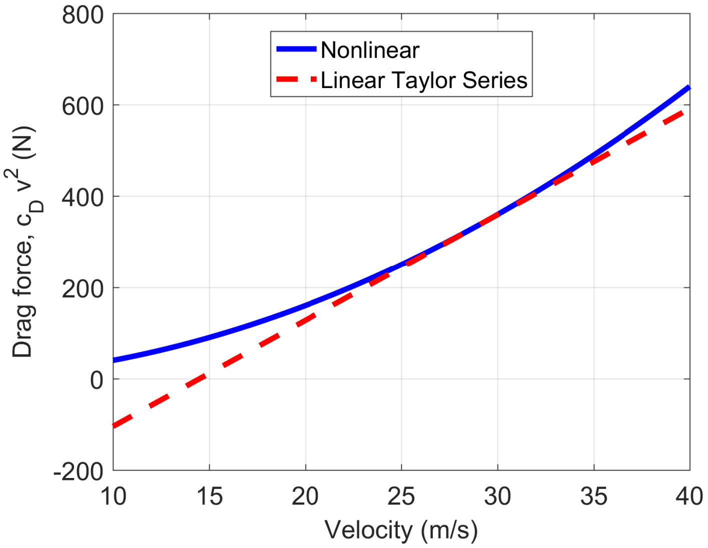


Figure 2.5: Quadratic drag and linear Taylor series approximation

2.4.2 Laplace Transform

Consider a signal $y(t)$ defined on $t \geq 0$. The *one-sided Laplace transform* of $y(t)$ is defined as:

$$Y(s) := \int_0^\infty y(t)e^{-st}dt \tag{2.33}$$

where $s \in \mathbb{C}$. The complex function $Y : \mathbb{C} \to \mathbb{C}$ is defined on a region of convergence. Specifically, the Laplace transform of y is defined at values of $s \in \mathbb{C}$ for which the integral converges. The transform of $y(t)$ will be denoted by either $Y(s)$ or $\mathcal{L}\{y(t)\}$ The Laplace transform has a number of useful properties that make it suitable for analysis of signals and systems. In this context, $y(t)$ is typically considered a function of time t and $Y(s)$ is a function of a (complex) frequency s. Hence the Laplace transform takes a signal in the time domain and maps it to a signal in the frequency domain. The following relation is used to connect an ODE and to its corresponding transfer function:

$$\mathcal{L}\{\dot{y}(t)\} = sY(s) - y(0) \tag{2.34}$$

This relation can be shown from the definition of the Laplace transform and integration by parts. Thus if the signal has zero initial value $y(0) = 0$ then differentiation in the time domain is equivalent to multiplication by "s" in the frequency domain. Similarly $\mathcal{L}\{y^{[k]}(t)\} = s^k Y(s)$ assuming that the y has zero ICs: $y(0) = \dot{y}(0) = \cdots = y^{[k-1]}(0) = 0$. Finally, consider an n^{th} order ODE with input u and output y:

$$a_n y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0 y(t) = b_m u^{[m]}(t) + \cdots + b_1\dot{u}(t) + b_0 u(t) \tag{2.35}$$

Applying the Laplace transform to this ODE with zero ICs yields:

$$\left(a_n s^n + a_{n-1}s^{n-1} + \cdots a_1 s + a_0\right) Y(s) = \left(b_m s^m + \cdots b_1 s + b_0\right) U(s) \tag{2.36}$$

Thus the input and output are related in the frequency domain by $Y(s) = G(s)U(s)$ where $G(s)$ is the transfer function:

$$G(s) := \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{2.37}$$

This brief discussion demonstrates that, in the frequency domain, the output of an ODE is given by the product of the transfer function and the input. Thus the transfer function is not simply another notation for the ODE. Additional details on the Laplace transform can be found in [6,16,18].

2.4.3 State-Space to Transfer Function

This section describes the steps to convert a state-space model to an ODE/transfer function representation. Recall that the state-space model has the following form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.38}$$

The Laplace transform, introduced in Appendix 2.4.2, can be used to show that the corresponding transfer function is $G(s) = C(sI - A)^{-1}B + D$. The numerator and denominator of $G(s)$ are polynomials in s that represent an equivalent linear ODE / transfer function.

This section will briefly describe an alternative derivation to convert from a state-space to a transfer function model. This requires one fact from linear algebra. The function $p(s) = \det(sI - A)$ is a polynomial in s and hence there are coefficients $\{a_0, \ldots, a_n\}$ such that:

$$p(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 \tag{2.39}$$

Recall that λ is an eigenvalue of A if and only if λ satisfies the characteristic equation $p(\lambda) = 0$. The Cayley-Hamilton theorem [8] states that A also satisfies this characteristic equation:

$$A^n + a_{n-1}A^{n-1} + \cdots + a_1 A + a_0 I = 0 \tag{2.40}$$

To derive an n^{th} order linear ODE, first differentiate the output of the state-space model (Equation 2.38) and substitute for $\dot{x}$:

$$\dot{y} = C\dot{x} + D\dot{u} = CAx + CBu + D\dot{u} \tag{2.41}$$

Continue differentiating the output n times and use the state-space model to substitute for $\dot{x}$ after each differentiation. Stacking these output derivatives yields:

$$\begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^{[n]} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{[n]} \end{bmatrix} x + \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ CA^{n-1}B & CA^{n-2}B & \cdots & D \end{bmatrix} \begin{bmatrix} u \\ \dot{u} \\ \vdots \\ u^{[n]} \end{bmatrix} \tag{2.42}$$

Multiply this equation on the left by the row vector $\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} & 1 \end{bmatrix}$. By the Cayley-Hamilton theorem the term involving x drops out. This yields a linear ODE of the form:

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0 y(t) = b_n u^{[n]}(t) + \cdots + b_1\dot{u}(t) + b_0 u(t) \tag{2.43}$$

where the coefficients $\{b_0, \ldots, b_n\}$ are defined as:

$$\begin{bmatrix} b_0 & b_1 & \cdots & b_{n-1} & b_n \end{bmatrix} := \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} & 1 \end{bmatrix} \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ CA^{n-1}B & CA^{n-2}B & \cdots & D \end{bmatrix} \tag{2.44}$$

The corresponding transfer function is $G(s) = \frac{b_n s^n + \cdots + b_1 s + b_0}{s^n + \cdots + a_1 s + a_0}$. The denominator of the transfer function is equivalent to the characteristic equation $p(s)$ for the matrix A. Thus the matrix A and the transfer function / linear ODE have the same characteristic equation. Moreover, the eigenvalues of A are the same as the roots of this characteristic equation. It can be shown with some additional algebra and another application of the Cayley Hamilton theorem that the numerator of this transfer function is equivalent to: $C\left(p(s)(sI - A)^{-1}\right)B + p(s)D$. The denominator of the transfer function is the characteristic polynomial $p(s)$. Hence the transfer function can be written as $G(s) = C(sI - A)^{-1}B + D$ as obtained via the Laplace transform.

2.4.4 Jacobian Linearization

Consider the following n^{th} order *nonlinear state-space model* with input u and output y:

$$\begin{aligned} &\dot{x}(t) = f(x(t), u(t)) \\ &y(t) = h(x(t), u(t)) \\ &\text{IC: } x(0) = x_0 \end{aligned} \tag{2.45}$$

An *equilibrium point* consists of (constant) values $\bar{x} \in \mathbb{R}^n$, $\bar{y} \in \mathbb{R}$ and $\bar{u} \in R$ such that

$$0 = f(\bar{x}, \bar{u}) \tag{2.46}$$

$$\bar{y} = h(\bar{x}, \bar{u}) \tag{2.47}$$

Define deviation variables $\delta_x(t) := x(t) - \bar{x}$, $\delta_y(t) := y(t) - \bar{y}$, and $\delta_u(t) := u(t) - \bar{u}$. Applying the multivariable Taylor series approximation around the equilibrium point $(\bar{x}, \bar{y}, \bar{u})$ yields the following linear state-space model:

$$\begin{aligned} &\dot{\delta}_x(t) = A\delta_x + B\delta_u \\ &\delta_y(t) = C\delta_x + D\delta_u \\ &\text{IC: } \delta_x(0) = x(0) - \bar{x} \end{aligned} \tag{2.48}$$

The entries of the state matrices are defined by the appropriate partials:

$$A_{i,j} := \frac{\partial f_i}{\partial x_j}(\bar{x}, \bar{u}), \quad B_{i,j} := \frac{\partial f_i}{\partial u_j}(\bar{x}, \bar{u})$$

$$C_{i,j} := \frac{\partial h_i}{\partial x_j}(\bar{x}, \bar{u}), \quad D_{i,j} := \frac{\partial h_i}{\partial u_j}(\bar{x}, \bar{u})$$

Chapter 3

System Response

This chapter covers the response of systems focusing primarily on simple first and second order systems. Section 3.1 reviews the use of numerical integration to compute the response of a system. The remainder of the chapter describes analytical methods for understanding the system response. Sections 3.2 and 3.3 review the procedure to solve for the free (initial condition) and forced response of an input. The stability characteristics of the system is related to the roots of a related characteristic equation. Next, Sections 3.4-3.6 derive the response of first and second order systems due to a step input. The key features include the final (steady-state) value, settling time, peak overshoot, rise time, and undershoot. Finally, Section 3.7 briefly summarizes key response features for general n^{th} order ODEs.

3.1 Numerical Simulation

Summary: This section describes the basic approach to numerical integration. Then several simulation tools within Matlab are reviewed. This includes command line functions for simulating linear and nonlinear systems as well as the graphical tool Simulink.

3.1.1 Numerical Integration

This section provides a basic outline of numerical integration methods. Consider a simple scalar, nonlinear ODE:

$$\dot{x}(t) = f(x(t), u(t)) \tag{3.1}$$

A scalar ODE is considered here only to simplify the discussion and numerical integration algorithms can typically handle higher order ODEs. Assume the initial condition $x(0) = x_0 \in \mathbb{R}$ is given and the input $u(t)$ is specified for $t \geq 0$. The next few sections provide an exact analytical solution when the ODE is linear. However, in most cases it is not possible to compute such analytical solutions. The objective of numerical integration is to compute an approximate solution using evaluations of the function f. A simplistic approach to numerical integration is based on the approximation of the time derivative for small Δt as:

$$\dot{x}(t) \approx \frac{x(t+\Delta t) - x(t)}{\Delta t} \tag{3.2}$$

Substitute this approximation into Equation 3.1 and solve for $x(t+\Delta t)$ to obtain:

$$x(t+\Delta t) \approx x(t) + f(x(t), u(t)) \cdot \Delta t \tag{3.3}$$

Thus the given $x(0) = x_0$ and input $u(0)$ can be used to approximately compute $x(\Delta t)$:

$$x(\Delta t) \approx x_0 + f(x_0, u(0)) \cdot \Delta t \tag{3.4}$$

Next, this approximation for $x(\Delta t)$ along with the given input $u(\Delta t)$ can be used to approximately compute $x(2\Delta t)$:

$$x(2\Delta t) \approx x(\Delta t) + f(x(\Delta t), u(\Delta t)) \cdot \Delta t \tag{3.5}$$

We can continue stepping ahead in time using this approximation. This yields the following iteration to compute an approximate solution to the nonlinear ODE for $k = 0, 1, \ldots$:

$$x((k+1)\Delta t) \approx x(k\Delta t) + f(x(k\Delta t), u(k\Delta t)) \cdot \Delta t \tag{3.6}$$

This algorithm is known as Euler integration. It assumes the step size Δt is fixed and it only requires a single evaluation of the function f at each step of the iteration. There are more sophisticated and computationally efficient (fixed-step) solvers that evaluate f multiple times at each step, e.g. the Runge-Kutta method. There are also numerical integration techniques that vary the step size Δt adaptively as the iteration progresses. These variable step solvers can be even faster and more accurate. A key point of this discussion is that numerical integration only approximately solves the ODE. For many problems the approximate solution will be of sufficient accuracy. However, if the solution is not sufficiently accurate (e.g. the solution looks "jagged") then there are typically settings that can be modified to improve the accuracy.

3.1.2 Command Line Functions

Matlab contains several command line functions to simulate linear systems including:

- **initial**: The syntax `[y,t]=initial(G,x0,TFINAL)` computes the free (initial condition) response for a system `G` with IC $x(0)=$`x0` from $t=0$ to $t=$`TFINAL`. This function requires `G` to be given as a linear state-space system.
- **lsim**: The syntax `[y,t]=lsim(G,u,t,x0)` computes the forced response for a system `G` with IC $x(0)$=`x0` and input specified by (`u,t`). The system `G` can be given either as a state-space or transfer function model. However the initial condition is used only if `G` is a linear state-space system and is ignored if `G` is a transfer function.
- **step**: The syntax `[y,t]=step(G,TFINAL)` computes the forced unit step response for a system `G` from $t=0$ to $t=$`TFINAL`. This is the response with zero IC and input $u(t)=1$ for $t\geq 0$. Again, `G` can be either a state-space or transfer function model. The response due a step of any (non-unit) magnitude can be obtained by simply rescaling the output of `step`. For example, the response y with zero IC and input $u(t)=2$ for $t\geq 0$ is obtained by `[yunit,t]=step(G,TFINAL)` and `y=2*yunit`. This follows from the principle of superposition as discussed in Section 2.1.2: if $(u(t),y(t))$ satisfy the linear ODE with zero IC then $(c\,u(t),c\,y(t))$ also satisfy the linear ODE for any constant $c\in\mathbb{R}$.

The help and documentation for these functions provides details including additional syntax options and examples. The integration is performed using specialized code that exploits the properties of linear systems.

Matlab also contains a variety of command line numerical integration solvers for nonlinear state-space systems including `ode45`, `ode23`, and `ode23s`. For example, the syntax `[t,x] = ode45(ODEFUN,[T0 TFINAL],x0)` integrates the system $\dot{x}(t)=f(t,x(t))$ with IC $x($`T0`$)=$`x0` from $t=$ `T0` to $t=$`TFINAL`. The input argument `ODEFUN` is itself a function that specifies the system dynamics f by `XDOT=ODEFUN(T,X)`. Note that f is allowed to depend explicitly on time and this can be used to model the effects of an input. The `ode45` routine uses the Runge-Kutta (4,5) formula to perform the numerical integration. A variety of numerical integration options can be specified using the `odeset`. See the documentation and help for additional details. These notes will not make use of the command line numerical integration routines for nonlinear systems. This brief summary is only intended to make you aware of these functions. We will instead use Simulink which is introduced in the next section.

Example 3.1. The functions `lsim` and `step` are demonstrated with the following linear ODE:

$$\begin{aligned}&2\ddot{y}(t)+0.5\dot{y}(t)+y(t)=0.3\dot{u}(t)+7u(t)\\&\text{IC: } y(0)=0;\ \dot{y}(0)=0\end{aligned}$$

The left subplot of Figure 3.1 shows two step responses with $u(t)=1$ and $u(t)=0.5$ for $t\geq 0$. The Matlab code to generate this figure is given below (omitting code to add labels, etc). The response with $u(t)=0.5$ is obtained by simply scaling the response with $u(t)=1$.

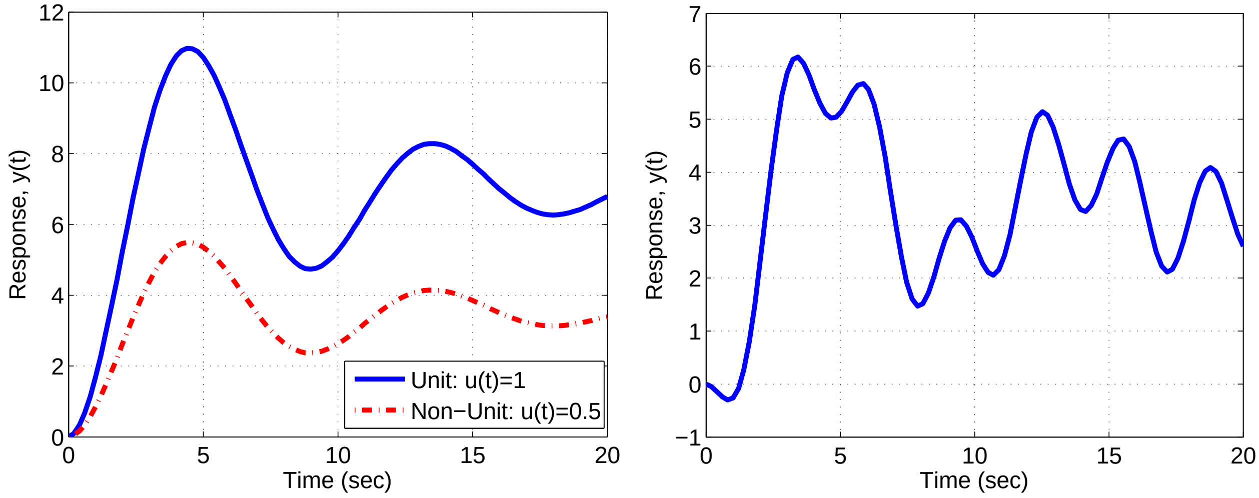


Figure 3.1: Forced response of $G(s) = \frac{0.3s+7}{2s^2+0.5s+1}$ with step input (left) and $u(t) = 0.5 - \cos(2t)$ (right). Both responses are with zero IC.

```
>> G = tf([0.3 7],[2 0.5 1]);
>> Tf = 20;
>> [yunit,tunit]=step(G,Tf);
>> yscaled=0.5*yunit;
>> plot(tunit,yunit,'b',tunit,yscaled,'r-.');
```

The right subplot of Figure 3.1 shows the forced response with $u(t) = 0.5 - \cos(2t)$ for $t \geq 0$. The Matlab code to generate this figure is given below (again omitting some additional code):

```
>> G = tf([0.3 7],[2 0.5 1]);
>> Tf = 20;
>> t = linspace(0,Tf,100);
>> u = 0.5-cos(2*t);
>> [y,t]=lsim(G,u,t);
>> plot(t,y);
```

△

3.1.3 Simulink

Matlab and Simulink are computational tools used to design, analyze and simulate control systems. Simulink is a graphical simulation tool that is based on the block diagram concept. It can be used to model interconnections of linear and/or nonlinear systems. One advantage of Simulink (as compared to a command-line solver like `ode45`) is that the block diagram framework easily allows components of a complex system to be independently modeled and interconnected. A drawback of this graphical, block-diagram framework is that certain standard programming concepts, e.g. "for-loops" and logical "if/else" statements, are cumbersome to implement in Simulink.

There are a variety of freely available tutorials on Matlab and Simulink. For example, Matlab offers several introductory tutorials in both video and written format:

- *Interactive Simulink Tutorial:* The link below contains a variety of video tutorials combined into three groups: (i) “Simulink On-Ramp”, (ii) “Using Simulink to Model Continuous Dynamical Systems”, and (iii) “Using Simulink to Model Discrete Dynamical Systems”. Each of these groups has a variety sub-topics. At this point it should be sufficient to watch “Constructing and Running a Simple Model” in Group (i) ($\approx 14min$), “Modeling Transfer Functions” in Group (ii) ($\approx 13min$) and “Modeling a System of Differential Equations also in Group (ii) ($\approx 7min$). The remaining videos in Groups (i) and (ii) are useful but not required. Discrete dynamical systems will not be a focus of this course and hence the videos in Group (iii) can be skipped.

  `http://www.mathworks.com/academia/student_center/tutorials/sltutorial_launchpad.html?s_cid=0409_webg_sltutorial_294710`

- *Learn Simulink Basics:* The link below contains more detailed written documentation on basic Simulink. The videos specified above should be sufficient for this course. The written document can serve as an additional resource.

  `http://www.mathworks.com/support/learn-with-matlab-tutorials.html`

It is worth noting that Simulink can be used within a *model-based design* process. Specifically, the “old” process is to use Matlab / Simulink (or another, similar tool) for the control design, analysis, and simulation. The final control design is then handed off from the control engineer to a software engineer. The software engineer translates the designed controller into code in a true programming language, e.g. C++. This can then be compiled and implemented on a production embedded processor. Bugs can be introduced in this translation step and the process must be repeated if the controller is updated. As a result, this translation step is costly both in terms of time and money. It is becoming standard practice in industry to instead use a model-based design process. One aspect of model-based design is to automatically generate production code. For example, Simulink has tools (Real-time Workshop) to directly generate code from the Simulink model. This allows the control design, analysis, and simulation to be done entirely within the Matlab/Simulink environment. The translation from Simulink to code is done automatically and the hand-off from the control to software engineer is avoided.

3.2 Free (Initial Condition) Response

Summary: A procedure to solve for the free response of a system is briefly reviewed. Next, two important aspects of the free response are discussed. First, the system is defined to be stable if the free response decays to zero for all initial conditions. The system is stable if and only if all roots of the characteristic equation have negative real part. Second, the speed of response is characterized by a time constant.

3.2.1 Free Response Solution

The previous section reviewed numerical integration. Numerical integration is used in most cases to solve for the free or forced response of a system. However, our control design tools require a better understanding of linear ODEs. Hence the next few sections will review the explicit (analytical) solution for the free and forced response of a linear n^{th} order ODE.

The *free response* of a linear system with input u and output y is obtained by setting the input to zero: $u(t) = 0$ for all $t \geq 0$. In this case the system response is modeled as:

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = 0 \tag{3.7}$$

$$\text{IC: } y(0) = y_0; \; \ldots \; y^{[n-1]}(0) = y_0^{[n-1]} \tag{3.8}$$

The free response of the system is equivalently called the *initial condition response*.

Solving for the free response requires a characterization of the homogeneous solutions of the ODE. Specifically, any solution to the ODE in Equation 3.7 (but not necessarily satisfying the IC in Equation 3.8) is called a *homogeneous solution*. Assume that $y(t) = e^{st}$ is a homogeneous solution for some number $s \in \mathbb{C}$. Note that the first two derivatives are $\dot{y}(t) = se^{st}$ and $\ddot{y}(t) = s^2 e^{st}$. In general, the k^{th} derivative is $y^{[k]}(t) = s^k e^{st}$. Substitute the assumed homogeneous solution y and its derivatives into the ODE to obtain:

$$\left(a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0 \right) e^{st} = 0 \tag{3.9}$$

The exponential e^{st} is nonzero for any values of s and t. Hence, $y(t) = e^{st}$ is a homogeneous solution if and only if s satisfies:

$$a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0 = 0 \tag{3.10}$$

Equation 3.10 is known as the *characteristic equation* and any solution s is called a *root*. The characteristic equation is an n^{th} order polynomial and hence it has n roots $\{s_1, \ldots, s_n\} \subset \mathbb{C}$ by the fundamental theorem of algebra. For simplicity assume the roots are distinct so that there are no repeated roots. It can be shown [2,5] that y is a homogeneous solution if and only if there exists coefficients $\{c_1, \ldots, c_n\} \subset \mathbb{C}$ such that

$$y(t) = \sum_{i=1}^{n} c_i e^{s_i t} \tag{3.11}$$

Based on this fact, the free response (assuming distinct roots) can be solved as follows:

1. Solve for the n roots $\{s_1, \dots, s_n\} \subset \mathbb{C}$ of the characteristic equation.
2. Form the general solution as $y(t) = \sum_{i=1}^n c_i e^{s_i t}$.
3. Use the n initial conditions to solve for the n unknown coefficients $\{c_1, \dots, c_n\} \subset \mathbb{C}$.

Similar steps hold even if the system has repeated roots. The only distinction is that the terms in the general homogeneous solution (Equation 3.11) must be altered if there are repeated roots. Additional details can be found in textbooks on ODEs [2,5]. It should also be noted that if time t has units of *sec* then the roots s_i have units of $\frac{rad}{sec}$. This implies that the product $s_i t$ has units of *rad*. The units $\frac{rad}{sec}$ should be assumed for the roots s_i whenever they are not explicitly specified. The three-step solution process is demonstrated next with a simple example.

Example 3.2. Consider a system modeled by a second-order linear ODE:

$$\begin{aligned} &2\ddot{y}(t) - 2\dot{y}(t) - 12y(t) = 4u(t) \\ &\text{IC: } y(0) = 11;\ \dot{y}(0) = -2 \end{aligned}$$

To find the free response ($u(t) = 0$), first compute the roots of the characteristic equation $2s^2 - 2s - 12 = 0$. The two roots of this polynomial are $s_1 = -2\frac{rad}{sec}$ and $s_2 = 3\frac{rad}{sec}$. The second step is to note that all homogeneous solutions have the form $y(t) = c_1 e^{-2t} + c_2 e^{3t}$. The third and final step is to use the initial conditions to solve for the coefficients c_1 and c_2. Substituting the general solution y into the initial conditions yields two equations with two unknowns:

$$\begin{aligned} 11 &= y(0) = c_1 + c_2 \\ -2 &= \dot{y}(0) = -2c_1 + 3c_2 \end{aligned}$$

The solution is $c_1 = 7$ and $c_2 = 4$. Hence the free response is given by $y(t) = 7e^{-2t} + 4e^{3t}$. $\triangle$

Both roots in the example above are real numbers. However, the roots of the characteristic equation can, in general, be complex numbers.[*] If s_i is complex then the complex exponential $e^{s_i t}$ appears in the homogeneous solution. Some additional clarification is needed to demonstrate that a real-valued solution is obtained in this case. First note that if the ODE coefficients $\{a_0, a_1, \dots, a_n\}$ are real then any complex roots of the characteristic equation come in complex conjugate pairs. Let $s_1 = \alpha + j\beta$ and $s_2 = \alpha - j\beta$ be a complex conjugate pair of roots where $j := \sqrt{-1}$. These roots lead to complex exponential terms in the solution of the form $c_1 e^{s_1 t} + c_2 e^{s_2 t}$. It follows from Euler's formula that these terms can be re-written as:

$$c_1 e^{s_1 t} + c_2 e^{s_2 t} = e^{\alpha t}(c_1 + c_2)\cos(\beta t) + j e^{\alpha t}(c_1 - c_2)\sin(\beta t) \tag{3.12}$$

Define $\tilde{c}_1 := c_1 + c_2$ and $\hat{c}_1 := j(c_1 - c_2)$. If the ICs are real numbers then it can be shown that both $\tilde{c}_1$ and $\hat{c}_1$ will be real numbers. Thus the complex exponential terms that appear in the solution will, in fact, yield real-valued terms $\tilde{c}_1 e^{\alpha t}\cos(\beta t) + \hat{c}_1 e^{\alpha t}\sin(\beta t)$. The solution procedure can be modified to explicitly express the homogeneous solution as real-valued signals:

[*] See Appendix 3.8.1 for a brief review of complex numbers.

1. Solve for the n roots of the characteristic equation. Group the real roots $\{s_1, \ldots, s_k\} \subset \mathbb{R}$ and complex conjugate pairs $\{\alpha_1 \pm j\beta_1, \ldots, \alpha_l \pm j\beta_l\} \subset \mathbb{C}$ of the characteristic equation.
2. Form the general solution as $y(t) = \sum_{i=1}^{k} c_i e^{s_i t} + \sum_{i=1}^{l} \tilde{c}_i e^{\alpha_i t} \cos(\beta_i t) + \hat{c}_i e^{\alpha_i t} \sin(\beta_i t)$.
3. Use the n initial conditions to solve for the n unknown coefficients $\{c_1, \ldots, c_k\} \subset \mathbb{R}$ and $\{\tilde{c}_1, \hat{c}_1, \ldots, \tilde{c}_l, \hat{c}_l\} \subset \mathbb{R}$.

Example 3.3. Consider a system modeled by a second-order linear ODE:

$$\begin{aligned} &\ddot{y}(t) + 2\dot{y}(t) + 5y(t) = u(t) \\ &\text{IC: } y(0) = 6;\ \dot{y}(0) = -14 \end{aligned}$$

To compute the free response ($u(t) = 0$), first solve for the roots of the characteristic equation $s^2 + 2s + 5 = 0$. The roots are $s_1 = -1 + 2j\ \frac{rad}{sec}$ and $s_2 = -1 - 2j\ \frac{rad}{sec}$. The second step is to form the general solution. This can be done with complex exponentials yielding $y(t) = c_1 e^{(-1+2j)t} + c_2 e^{(-1-2j)t}$. The third and final step is to use the IC to solve for the coefficients c_1 and c_2. Substituting y into the IC yields two equations with two unknowns:

$$\begin{aligned} 6 &= y(0) = c_1 + c_2 \\ -14 &= \dot{y}(0) = (-1 + 2j)c_1 + (-1 - 2j)c_2 \end{aligned}$$

The solution is $c_1 = 3 + 2j$ and $c_2 = 3 - 2j$. These coefficients are complex conjugates as expected from the discussion above. By Euler's formula, the free response $y(t) = (3 + 2j)e^{(-1+2j)t} + (3 - 2j)e^{(-1-2j)t}$ can be rewritten as $y(t) = e^{-t}(6\cos(2t) - 4\sin(2t))$.

We can obtain the same result using the general solution expressed with real sinusoids. Specifically, all homogeneous solutions have the form $y(t) = \tilde{c}_1 e^{-t}\cos(2t) + \hat{c}_1 e^{-t}\sin(2t)$. The initial conditions can be used to solve the unknown coefficients yielding $\tilde{c}_1 = 6$ and $\hat{c}_1 = -4$. △

3.2.2 Stability

Stability is a property of the free response as $t \to \infty$ as formalized in the next definition.

Definition 3.1. A linear system is *stable* if the free response returns to zero ($y(t) \to 0$ as $t \to \infty$) for any initial condition. The system is called *unstable* if it is not stable.

The free response is a linear combination of exponentials $e^{s_i t}$ where s_i is a characteristic equation root. If s_i is real then the exponential can decay to zero (if $s_i < 0$), grow unbounded (if $s_i > 0$) or remain constant (if $s_i = 0$). If s_i is part of a complex pair $\alpha_i \pm j\beta_i$ then it contributes terms of the form $e^{\alpha_i t}\cos(\beta_i t)$ and $e^{\alpha_i t}\sin(\beta_i t)$. These terms can decay to zero (if $\alpha_i < 0$), grow unbounded (if $\alpha_i > 0$) or oscillate (if $\alpha_i = 0$). This leads to the following fact:

Fact 3.1. A linear system is stable if and only if all roots of the characteristic equation have strictly negative real part, i.e. $Re\{s_i\} < 0$ for all i where Re denotes the real part of the root.

The system in Example 3.3 is stable ($Re\{s_i\} = -1 < 0$ for $i = 1, 2$) while the system in Example 3.2 is unstable ($s_2 = 3 > 0$). In general if a system has at least one root with $Re\{s_i\} > 0$ then the free response grows unbounded. If the system has no roots with $Re\{s_i\} > 0$ but has (distinct) roots with $Re\{s_i\} = 0$, then the solution will neither decay to zero nor grow unbounded. Instead, the solution will either oscillate or remain constant. A system with these properties is sometimes called marginally stable but we will still consider it as unstable.

3.2.3 Time Constant

Next, the "speed" of the free response is discussed. To clarify the notion of "speed", consider a first order system $\dot{x}(t) + a_0 x(t) = 0$ with IC $x(0) = x_0$. The solution is $x(t) = x_0 e^{-a_0 t}$. For the concrete value $a_0 = 1$, the solution satisfies $x(3) = e^{-3} x_0 \approx 0.05 x_0$. In other words, the response is stable ($s = -1 \frac{rad}{sec}$) and decays to $\approx 5\%$ of its original value in $3 sec$. Alternatively if $a_0 = 4$ then $x(0.25) = e^{-3} x_0 \approx 0.05 x_0$. Again, the response is stable ($s = -4 \frac{rad}{sec}$) and decays to $\approx 5\%$ of its original value in $0.25 sec$. Note that the speed of convergence for a stable, first-order system is inversely related to the root. This concept is formalized in the next definition.

Definition 3.2. The time constant associated with a root $s \in \mathbb{C}$ is defined to be $\tau = \frac{1}{|Re\{s\}|}\, sec$.

As discussed above, the time constant associated with a real root s is directly related to the speed of convergence of e^{st}. Similarly, complex conjugate roots $s = \alpha \pm \beta$ give rise to terms of the form $e^{\alpha t} \cos(\beta t)$ and $e^{\alpha t} \sin(\beta t)$. If $\alpha < 0$ (stable response) and t is three time constants (i.e. $t = \frac{3}{|\alpha|}$) then $|e^{\alpha t} \cos(\beta t)| \leq e^{-3} \approx 0.05$ and $|e^{\alpha t} \sin(\beta t)| \leq 0.05$. **Thus all terms in the free response individually decay to $\approx$5% of their original value after three time constants.** The free response $y(t)$ of an n^{th} order system is a sum of n such terms. Thus it is not easy to precisely characterize the time for a stable response $y(t)$ to decay to 5% of $y(0)$. Roughly the slowest term (longest time constant) will dominate the speed of response. This dominant root approximation will be discussed further in Section 3.7.

Figure 3.2 shows the free response terms for various values of the root. Specifically, the top subplot shows the term e^{st} for real roots $s = -2, -1, 0,$ and $0.5 \frac{rad}{sec}$. The response decays to zero for $s = -2$ and -1, grows unbounded for $s = 0.5$, and remains constant for $s = 0$. The inset graph shows the location of the roots in the complex plane. It is common to state Fact 3.1 as: "a system is stable if and only if all roots are strictly in the left half of the complex plane (LHP)". The roots $s = -2$ and -1 have time constants of $0.5 sec$ and $1 sec$. Thus the responses for $s = -2$ and -1 converge to 0.05 in approximately three time constants ($1.5 sec$ and $3 sec$, respectively). Note that a slower response corresponds to a larger time constant. Thus, (stable) roots closer to the imaginary axis in the complex plane correspond to a slower response, e.g. $s = -1$ is slower than $s = -2$. Similarly, unstable roots closer to the imaginary axis diverge more slowly than those farther to the right, e.g. $s = +2$ (not shown) diverges more quickly than $s = +0.5$.

The bottom subplot shows the free response term $e^{\alpha t} \cos(\beta t)$ for complex roots $s = \alpha \pm j\beta$ with $\alpha = -2, -1, 0, 0.5$ and $\beta = 2$. The related term $e^{\alpha t} \sin(\beta t)$ has similar behavior and is not shown. The response decays to zero for $\alpha = -2$ and -1, grows unbounded for $\alpha = 0.5$, and oscillates for $\alpha = 0$. The inset graph shows the location of the root $\alpha + j\beta$ in the complex plane. The other conjugate root $\alpha - j\beta$ is not shown. The roots with $\alpha = -2$ and -1 have time constants of $0.5 sec$ and $1 sec$. The responses for $\alpha = -2$ and -1 converge to 0.05 in approximately three time constants ($1.5 sec$ and $3 sec$, respectively). Again, a slower response corresponds to a larger time constant and roots closer to the imaginary axis correspond to slower response. The oscillations in the response are related to the imaginary part β. For example, $\cos(\beta t) = 0$ for $t = \frac{\pi}{2\beta}, \frac{3\pi}{2\beta}, \ldots$. Thus all responses in the bottom subplot of Figure 3.2 are zero at $t \approx 0.79$ and $2.36 sec$. These oscillations are discussed further in Section 3.3.

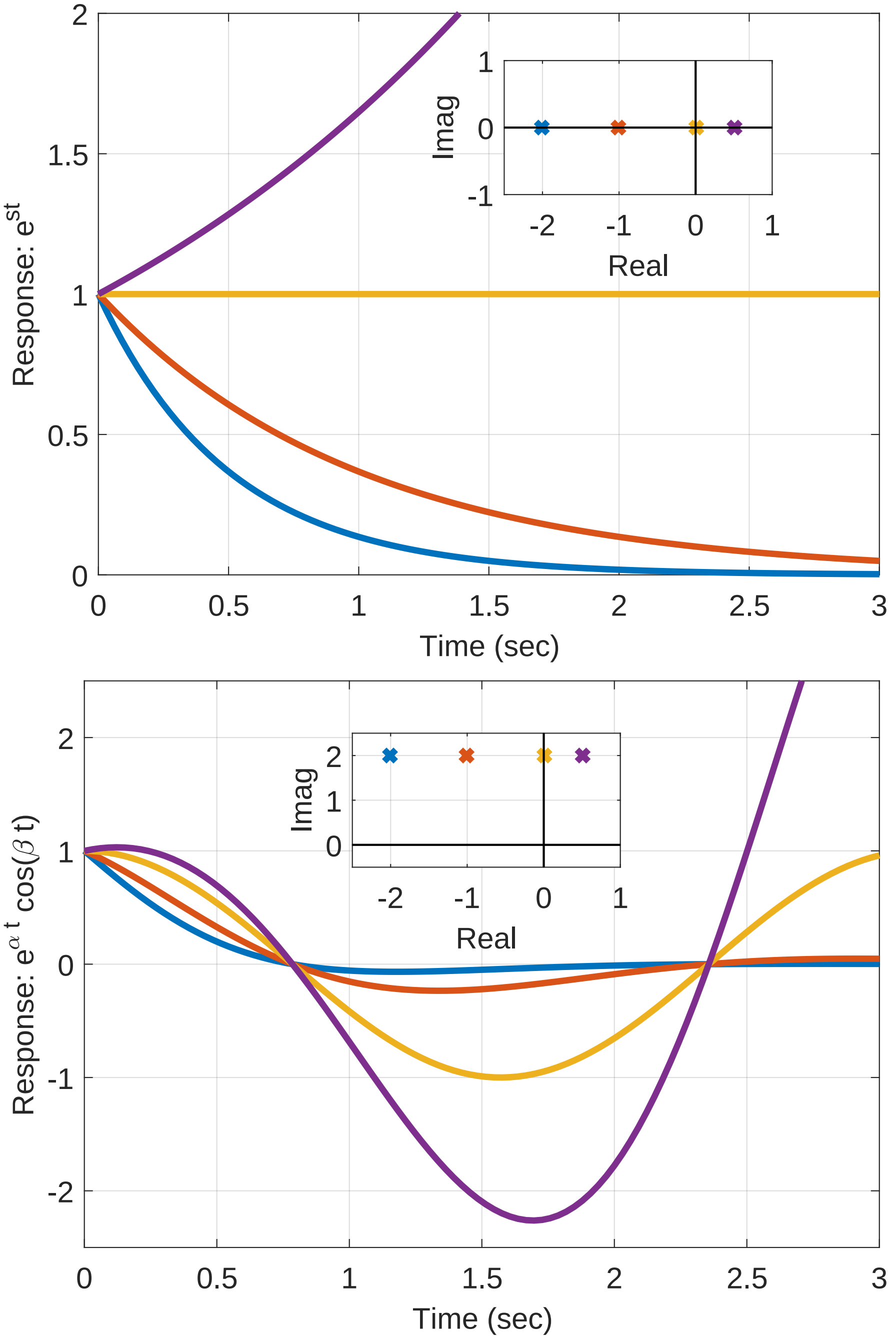


Figure 3.2: Free response terms due to real roots (top) and complex roots (bottom).

3.3 Forced Response

Summary: A procedure to solve for the forced response of a system is briefly reviewed. Next, a technical point is discussed regarding minimal and non-minimal systems. Finally, the system is defined to be bounded-input, bounded-output (BIBO) stable if the output remains bounded for any bounded input. A minimal system is BIBO stable if and only if all roots of the characteristic equation have negative real part. Thus the conditions for BIBO stability are identical to those given for stability of the free response.

3.3.1 Forced Response Solution

The *forced response* of a linear system with input u and output y is obtained with nonzero inputs. In this case the system response is modeled as:

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^{[m]}(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t) \qquad (3.13)$$

$$\text{IC: } y(0) = y_0; \; \ldots \; y^{[n-1]}(0) = y_0^{[n-1]} \qquad (3.14)$$

Any solution to the forced ODE in Equation 3.13 (but not necessarily satisfying the IC in Equation 3.14) is called a *particular solution*. The forced response can be solved by building on the procedure described for the free response. Again assume for simplicity that the characteristic equation has no repeated roots. Then the forced response can be solved as follows:

1. Solve for the n roots $\{s_1, \ldots, s_n\} \subset \mathbb{C}$ of the characteristic equation.
2. Find any particular solution $y_P(t)$.
3. Form the general solution as $y(t) = y_P(t) + \sum_{i=1}^{n} c_i e^{s_i t}$.
4. Use the n initial conditions to solve for the n unknown coefficients $\{c_1, \ldots, c_n\} \subset \mathbb{C}$.

The solution procedure is similar if the system has repeated roots but with some modification to the terms in the general homogeneous solution. One example of the required modifications is given in Section 3.6.3. Moreover, the complex exponential terms arising from complex roots can be re-written as real terms as discussed in Section 3.2. There are a variety of methods to solve for a particular solution in Step 3 including the method of undetermined coefficients. These methods won't be described in detail because we'll mainly consider simple input functions, e.g. steps and sinusoids. Additional details on the forced response solution can be found in [2,5].

Example 3.4. Example 2.6 derived a model for a BMW 750iL linearized around the equilibrium point $(\bar{v}, \bar{F}_{net}) = (20.74 \frac{m}{sec}, 400N)$. The linearized dynamics are given by:

$$\dot{\delta}_v(t) + 0.008 \delta_v(t) = (4.8 \times 10^{-4})\, \delta_F(t) \qquad (3.15)$$

The exact forced response with initial condition $\delta_v(0) = 0$ and step input $\delta_F(t) = 50N$ for $t \geq 0$ will be derived using the solution procedure described above. First, the characteristic equation has only a single (stable) root $s = 0.008 \frac{rad}{sec}$. Second, the input δ_F is a constant and hence there is a constant particular solution $\delta_v(t) = \bar{\delta}_v$. This particular solution must satisfy

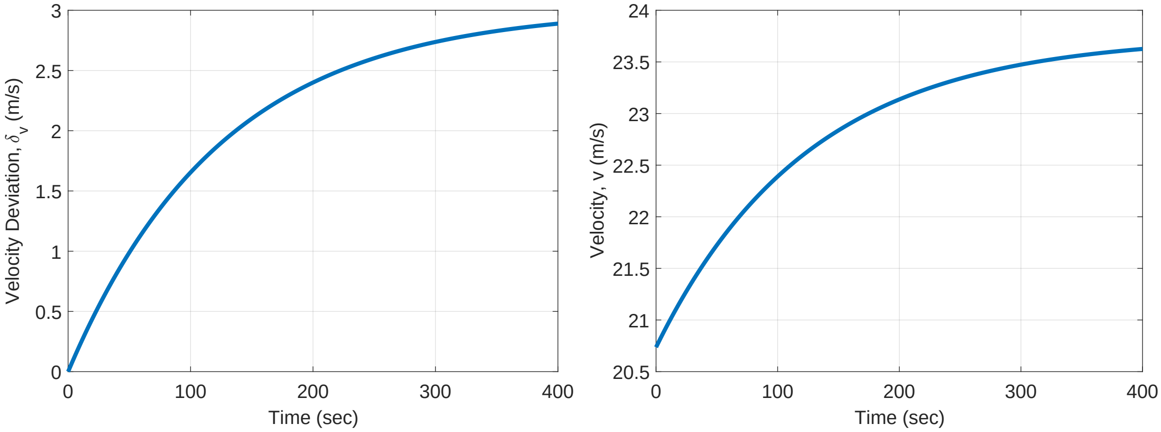


Figure 3.3: Step response for linearized BMW dynamics with step force $\delta_F = 50N$. Velocity deviation from trim (left) and actual velocity (right) are shown.

the ODE which, for a constant solution, simplifies to $0.008\bar{\delta}_v = (4.8 \times 10^{-4}) \times (50)$. This yields the particular solution $\delta_v(t) = 3\frac{m}{sec}$. Third, the general solution is $\delta_v(t) = 3 + c_1 e^{-0.008t}$. The final step is to use the initial condition $\delta_v(0) = 0$ to solve for $c_1 = -3$. Hence the forced step response for the BMW is given by $\delta_v(t) = 3 - 3e^{-0.008t}\frac{m}{sec}$. This solution is shown in the left subplot of Figure 3.3. The final value in this case is simply the particular solution, $\delta_v(t) \to 3\frac{m}{sec}$. The time constant associated with the root is $\tau = \frac{1}{0.008} = 125sec$. It takes approximately three time constants ($\approx 375sec$) for the step response to converge to within 5% of its final value. Note that δ_v is the deviation of the car from its trim velocity. The actual vehicle velocity is obtained by simply adding back the trim velocity, i.e. $v(t) = \delta_v(t) + \bar{v}$. This is shown in the right subplot of Figure 3.3. The forced step response can also be approximately computed using the numerical methods described in Section 3.1. For example, the code below uses the `step` command to compute the (approximate) solution:

```
>> G = tf(4.8e-4,[1 0.008]); % Linearized dynamics
>> Tf = 400;                 % Final simulation time, sec
>> dF = 50;                  % Step input force, N
>> [dv_unit,t]= step(G,Tf);  % Unit step response (dF=1);
>> dv = dv_unit*dF;          % Step response with specified dF
>> plot(t,dv)
```

△

3.3.2 Minimal Realizations

This section will clarify a technical point regarding “non-minimal” ODEs. To highlight the issue, consider the ODE $\dot{y} + y = 2\dot{u} + 2u$. For any (differentiable) input u this ODE has $y = 2u$ as a particular solution. Hence the first-order model $\dot{y} + y = 2\dot{u} + 2u$ is not minimal in the sense that the same input-output dynamics is obtained by the (zero-order) model $y = 2u$.

To generalize this discussion, consider the transfer function for an n^{th} order linear ODE:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{3.16}$$

Up to this point the transfer function has simply been used as another notation for the ODE. Additional insight can be gained by viewing G as a complex function. See Appendix 3.8.1 for a brief review of complex functions. This viewpoint is used in the following definitions.

Definition 3.3. A number $s \in \mathbb{C}$ is a *pole* of the system if $a_n s^n + \cdots + a_1 s + a_0 = 0$. Hence the terms "poles" and "roots" have the same meaning and will be used interchangeably.

Definition 3.4. A number $z \in \mathbb{C}$ is a *zero* of the system if $b_m z^m + \cdots + b_1 z + b_0 = 0$. The zero z is called a *right-half plane* (RHP) zero or *non-minimum phase* zero if $Re\{z\} \geq 0$.[†]

A system that has no common poles and zeros is called *minimal*. For minimal systems, z is a zero if and only if $G(z) = 0$. Similarly s is a pole if and only if $|G(r)| = \infty$. Also note that if no input derivatives appear in the ODE then the transfer function numerator is simply $b_0 \neq 0$ and the system has no zeros. Thus the zeros are associated with input derivative terms.

A system that has a pole and zero at the same location is called *non-minimal*. In this case the transfer function is not well-defined at the location of the common pole/zero. For example, the system $\dot{y} + y = 2\dot{u} + 2u$ discussed above has the transfer function $G(s) = \frac{2(s+1)}{s+1}$. This system has a common pole and zero at $s = -1$. Thus $G(-1) = \frac{0}{0}$ is not well-defined. The common pole/zero can be canceled. This yields the transfer function $G(s) = 2$ corresponding to the zero-order model $y = 2u$. In general, common poles/zeros in non-minimal systems can be canceled to obtain a minimal representation with equivalent input-output dynamics. This cancellation "hides" some unobserved dynamics. In most cases, we'll consider systems with minimal representations. However, this technical point regarding non-minimal realizations will appear when considering feedback systems later in the course. Minimal and non-minimal realizations are discussed further in textbooks on state-space modeling and control [4, 8, 10].

The Matlab commands `pole` and `zero` can be used to compute the poles (roots) and zeros of a general n^{th} order system. The command `minreal` is used to compute a minimal realization for a system. The help and documentation for these functions provides additional details.

Example 3.5. The Matlab functions `pole`, `zero`, and `minreal` are demonstrated below. The step responses for the non-minimal and minimal realizations (Figure 3.4) are identical.

```
>> G = tf([3042 10140 3042],[1 13 199 507]);
>> zero(G) % Zeros are roots of 3024s^2+10140s+3042 = 0
ans =
   -3.0000
   -0.3333
>> pole(G) % Poles are roots of s^3+13s^2+199s+507 = 0
ans =
  -5.0000 +12.0000i
  -5.0000 -12.0000i
  -3.0000 + 0.0000i

>> Gmin = minreal(G)    % Cancels common pole/zero at s=-3
```

[†] The term "non-minimum phase" arises from the forced response with a sinusoidal input as discussed later.

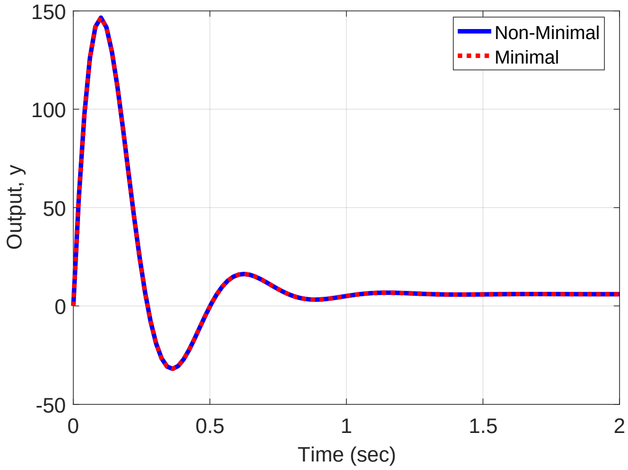


Figure 3.4: Step response for non-minimal realization $G(s)$ and minimal realization $G_{min}(s)$.

```
Gmin =
   3042 s + 1014
  ----------------
  s^2 + 10 s + 169
>> step(G,'b',Gmin,'r')  % Step responses are identical                △
```

3.3.3 Stability

Definition 3.1 in Section 3.2 defines stability in terms of the free (initial condition) response generated with zero input ($u(t) = 0$). There is another notion of stability related to the forced response with non-zero inputs.

Definition 3.5. A linear system is *bounded-input, bounded-output (BIBO) stable* if the output remains bounded for every bounded input. The system is called *(BIBO) unstable* if it is not (BIBO) stable.

More precisely, an input u is said to be bounded if there is some number $N_u < \infty$ such that $|u(t)| < N_u$ for all $t \geq 0$. If a system is BIBO stable then a bounded input u will lead to a bounded output y. Thus there will exist some other number $N_y < \infty$ such that $|y(t)| < N_y$ for all $t \geq 0$. There is a simple condition for BIBO stability in terms of the system poles.

Fact 3.2. A minimal, linear system is BIBO stable if and only if all roots of the characteristic equation have strictly negative real part, i.e. $Re\{s_i\} < 0$ for all i.

The proof of Fact 3.2 requires additional concepts and will be given in Section XXX. This fact is stated for minimal systems to avoid the technical issues associated with canceled dynamics (as discussed in the previous subsection). Note that the condition for stability of the free response (Fact 3.1) is identical to this condition for BIBO stability. Hence the two stability definitions (free response and BIBO) are equivalent for minimal, linear systems. Thus we won't distinguish between them when using the term "stable". The two notions of stability are not equivalent, in general, for nonlinear systems.

3.4 Step Response

Summary: This section briefly discusses the key qualitative features of the step response. The main focus is on step responses for stable systems. The key features include the final (steady-state) value, settling time, peak overshoot, rise time, and undershoot.

Time domain performance specifications can be used to design simple controllers. These specifications are usually given in terms of the forced response of the system with zero initial conditions and a step input: $u(t) = \bar{u}$ for $t \geq 0$. The step response can be computed with the solution procedure in Section 3.3.1. This yields a step response of the form $y(t) = y_P(t) + \sum_{i=1}^{n} c_i e^{s_i t}$ where $\{s_1, \ldots, s_n\} \subset \mathbb{C}$ are the roots, y_P is a particular solution and the coefficients $\{c_1, \ldots, c_n\} \subset \mathbb{C}$ are determined from the zero initial conditions. The step response characteristics depend on the stability of the system:

- **Stability:** Figure 3.5 shows a collection of unit step responses for stable (left) and unstable (right) systems. The locations of the system poles in the complex plane are also shown inset. The step function is a bounded input. Hence, if the system is stable then the step response will remain bounded. As noted in the previous section, the system is (free response and BIBO) stable if and only if all poles have negative real part. If the system is unstable then, in most cases, the step response will grow unbounded. However, it is possible for the step response of an unstable system to remain bounded. For example $\ddot{y}(t) + y(t) = u(t)$ has poles $s = \pm j$. This system is unstable and yet the unit step response is bounded/oscillatory as shown in the right subplot of Figure 3.5.[‡]

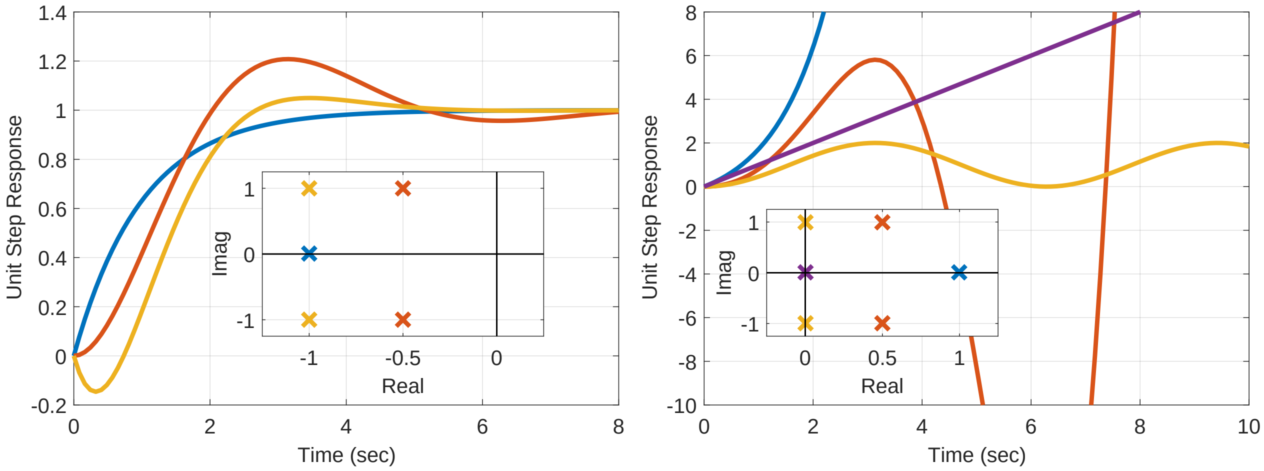


Figure 3.5: Unit step responses for stable (left) and unstable (right) systems. The stable systems are $\frac{1}{s+1}$, $\frac{1.25}{s^2+s+1.25}$, and $\frac{-s+2}{s^2+2s+2}$. The unstable systems are $\frac{1}{s-1}$, $\frac{1.25}{s^2-s+1.25}$, $\frac{1}{s^2+1}$, and $\frac{1}{s}$.

Typically a control system is designed to ensure, at a minimum, that the response is stable.

[‡] BIBO stability means the output remains bounded *for all* bounded inputs. A system is BIBO unstable if the output grows unbounded *for at least one* bounded input. The system $\ddot{y}(t) + y(t) = u(t)$ has a bounded output for a unit step but it is unstable because the output grows unbounded for the bounded input $u(t) = \sin(t)$.

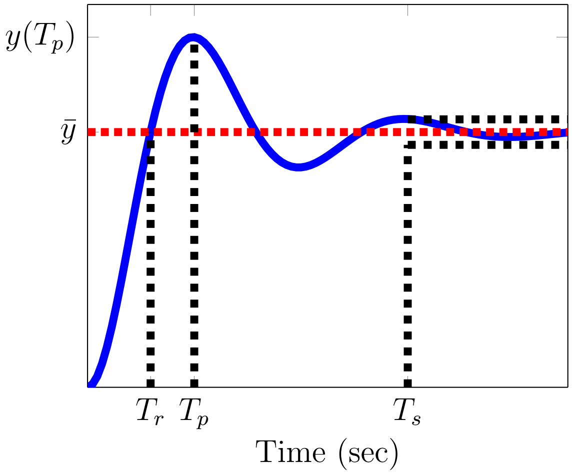


Figure 3.6: Key features of a stable step response.

Hence we will focus mainly on this case. Figure 3.6 labels several key features for a typical stable step response. These features are discussed below.

- **Final Value:** As discussed above, if the system is stable then the step response remains bounded. In addition, the response converges as $t \to \infty$ to a *final or steady-state value*. The steady-state value $\bar{y}$ for a stable step response is obtained by setting all derivatives of y and u to zero in the ODE. This yields the algebraic relation $a_0\bar{y} = b_0\bar{u}$ which can be solved for the steady-state output $\bar{y} := \frac{b_0}{a_0}\bar{u}$.[§] Note that $y_P(t) = \bar{y}$ is a particular solution for a step input $u(t) = \bar{u}$ and thus the step response solution is $y(t) = \bar{y} + \sum_{i=1}^{n} c_i e^{s_i t}$. If the system is stable then all exponential terms decay to zero and $y(t) \to \bar{y}$ as $t \to \infty$.

- **Settling Time:** The *settling time* T_s is the time for the output to converge within ±5% of the steady-state value. This corresponds to the time it takes to get and stay within the interval $[0.95\bar{y},\, 1.05\bar{y}]$. Slightly different definitions are occasionally used, e.g. 1% or 2% settling times. The settling time is one measure for the system speed of response.

- **Peak Overshoot:** The step response for $\frac{1}{s+1}$ shown in Figure 3.5 smoothly rises to the final value. However, the other two responses overshoot (exceed) the final value and oscillate before converging. Roughly, the step response for $\frac{1}{s+1}$ is *overdamped* while the other two responses are *underdamped*. These terms will be defined more precisely in Section 3.6. The peak (maximum) value of the output is $y(T_p)$ where T_p is the peak time. The *peak overshoot* is defined as:

$$M_p = \frac{y(T_p) - \bar{y}}{\bar{y}} \tag{3.17}$$

  The numerator $y(T_p) - \bar{y}$ is the amount of overshoot beyond the final value. This is normalized in the denominator by the final value $\bar{y}$ so that M_p is unitless. As a result

[§] The definition of $\bar{y}$ requires $a_0 \neq 0$ in order to be well-defined. If the coefficient of y in the ODE is zero, i.e. $a_0 = 0$, then characteristic equation has the form $a_n s^n + \cdots + a_1 s = 0$. Hence $a_0 = 0$ implies $s = 0$ is a root of the characteristic equation and the system is unstable. Conversely if the system is stable then it must be true that $a_0 \neq 0$. Hence $\bar{y} := \frac{b_0}{a_0}\bar{u}$ is well-defined for a stable system.

of this normalization, M_P does not depend on the step magnitude $\bar{u}$. Occasionally the peak overshoot is expressed as a percent, e.g. $100 \times M_p\%$. Overdamped responses do not overshoot the final value and hence M_p is not defined in this case.

- **Rise Time:** The *rise time* T_r is the time required for the response to first reach the steady-state value: $y(T_r) = \bar{y}$. This is a measure of the initial speed of response. This is in contrast to the settling time which requires any oscillations to decay to a small value. The rise time, as defined here, is not well-defined for overdamped responses. Other definitions for the rise time are occasionally used, e.g. the time for the output to go from 10% to 90% of the steady state value.

- **Undershoot:** The step response for $\frac{-s+2}{s+2s+2}$ shown in Figure 3.5 initially moves negative before reversing direction toward the final value. The step response shown in Figure 3.6 does not exhibit this undershoot behavior. This tendency to initially move in the wrong direction is called *undershoot*. This behavior is associated with derivatives of the input appearing in a certain form in the ODE. For example, the input of the system $\frac{-s+2}{s+2s+2}$ enters as $-\dot{u} + 2u$. The negative coefficient on $\dot{u}$ causes this undershoot. This will be discussed further in Section 3.7. Undershoot is not common but does occur in some systems. Such behavior introduces challenges in the control design.

A step input is a very simple forcing input. However, the key features defined above are sufficient to understand the overall performance of simple systems. The following sections provide more quantitative descriptions for first, second, and higher order systems. In particular, the characteristics of the system poles (time constant, real/imaginary parts) will be useful to infer characteristics of the system response (settling time, rise time, overshoot).

3.5 First Order Step Response

Summary: The exact step response solution for a first-order system is derived. Then two key features (settling time, final value) for a stable response are discussed. In particular, the system has only a single real pole and its associated time constant. The settling time of the system is approximately three time constants. The final value is computed from a simple algebraic equation obtained by setting all derivative terms in the ODE to zero.

Consider the following first-order system:

$$\dot{y}(t) + a_0 y(t) = b_0 u(t) \tag{3.18}$$

$$\text{IC: } y(0) = 0 \tag{3.19}$$

The coefficient of $\dot{y}$ is normalized to simplify the derivation. If $a_1 \neq 1$ then the ODE can be normalized by dividing out a_1. This ODE also does not have the term $b_1\dot{u}(t)$ and the impact of this term is discussed in Section 3.7.

Section 3.3.1 provides a procedure to solve for the unit step response ($u(t) = 1$) of the first-order ODE given above. First, the characteristic equation $s + a_0 = 0$ has a single root $s = -a_0$. The remainder of the solution depends on the value of this root:

- $s = -a_0 < 0$: In this case $a_0 \neq 0$ and hence the constant $\bar{y} := \frac{b_0}{a_0}$ is well defined. The function $y_P(t) = \bar{y}$ is a particular solution for a unit step input $u(t) = 1$. In addition, the initial condition can be used to solve for the unknown coefficient in the resulting general solution. This yields the following forced response solution:

$$y(t) = \bar{y}\left(1 - e^{st}\right) \text{ where } s = -a_0 \text{ and } \bar{y} = \frac{b_0}{a_0}. \tag{3.20}$$

  The solution is stable as the only root satisfies $s < 0$. Two key features of this stable response are the final (steady-state) value and the settling time. In particular, the solution $y(t)$ begins at $y(0) = y_0$ and converges exponentially to $\bar{y}$ as $t \to \infty$. Hence $\bar{y}$ is the final value. The 5% settling time is approximately three time constants where the time constant is $\tau = \frac{1}{|s|} = \left|\frac{1}{a_0}\right|$. The solution is overdamped and hence neither overshoot nor rise time are defined.

- $s = -a_0 > 0$: In this case, the solution is again given by Equation 3.20. The main distinction is that the response is unstable as the only root satisfies $s > 0$. Hence the exponential e^{st} grows unbounded with time.

- $s = -a_0 = 0$: This implies $a_0 = 0$ which means the ODE is $\dot{y}(t) = b_0 u(t)$. A particular solution for a unit step input is $y_P(t) = b_0 t$ and the general solution is $y(t) = y_P(t) + c_1$. Using the initial condition to solve for c_1 yields the forced response solution for $s = 0$:

$$y(t) = b_0 t \tag{3.21}$$

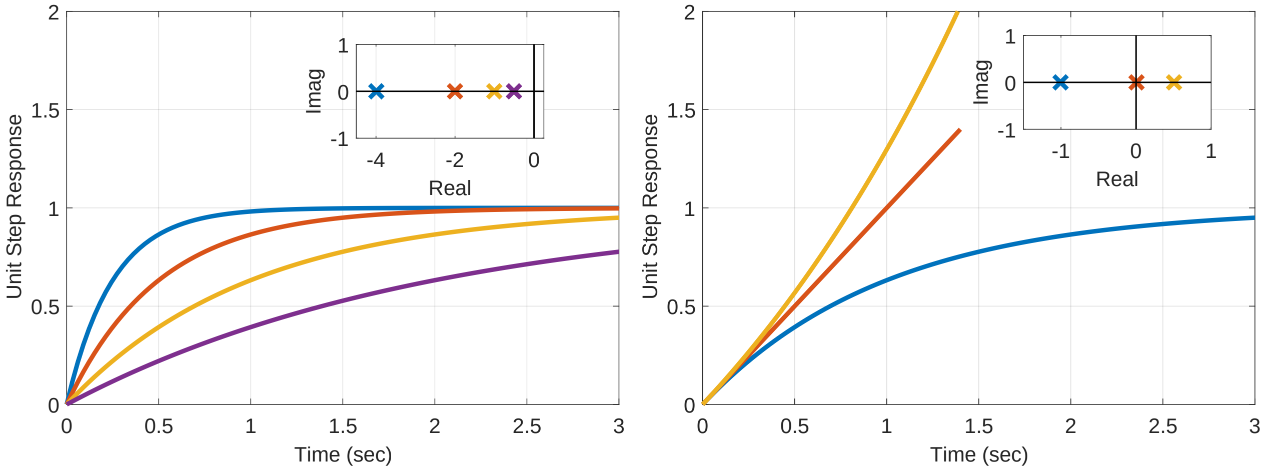


Figure 3.7: First-order step responses with stable roots (left) and stable/unstable roots (right)

Figure 3.7 shows a collection of unit step responses with $y(0) = 0$. The left plot shows stable step responses with $a_1 = 1$ and $(a_0, b_0) = \{(4,4),\ (2,2),\ (1,1),\ (0.5, 0.5)\}$. Each of these responses corresponds to a steady state value of $\bar{y} = 1$. The four responses have stable roots $s = \{-4,\ -2,\ -1,\ -0.5\}\frac{rad}{sec}$. These correspond to time constants $\tau = \{0.25,\ 0.5,\ 1,\ 2\}sec$ and settling times $3\tau = \{0.75,\ 1.5,\ 3,\ 6\}sec$. The solutions converge more slowly for roots that are closer to the imaginary axis. The right plot compares three responses with $a_1 = 1$, $b_0 = 1$, and $a_0 = \{-1,\ 0,\ +1\}$. These three responses have roots $s = \{+1,\ 0,\ -1\}$. The response with $s = -1$ is stable while the responses with $s = 0$ and $s = +1$ are unstable. Note that the unstable responses grow unbounded either linearly ($s = 0$) or exponentially ($s = +1$) with time. These responses demonstrate that the systems with $s = 0$ and $s = +1$ are BIBO unstable, i.e. the responses grow unbounded for the bounded step input.

The derivation given above can be modified to handle $a_1 \neq 0$, non-zero initial conditions $y(0) = y_0$, and non-unit step inputs $u(t) = \bar{u}$ for $t \geq 0$. For example, if $s \neq 0$ then the more general solution to handle these cases is:

$$y(t) = \bar{y}\left(1 - e^{st}\right) + y_0 e^{st} \text{ where } s = -\frac{a_0}{a_1} \text{ and } \bar{y} = \frac{b_0}{a_0}\bar{u}. \tag{3.22}$$

This solution starts at $y(0) = y_0$. If the response is stable ($s < 0$) then it converges to the steady-state value $\bar{y}$. It takes three time constants to converge 95% of the way from y_0 to $\bar{y}$. Note that the steady-state value for a stable response is obtained by setting $\dot{y}(t) = 0$ and solving the resulting algebraic equation $a_0\bar{y} = b_0\bar{u}$.

3.6 Second Order Step Response

Summary: The step response solution for a second-order system is discussed. The solution can be overdamped (distinct real roots), critically damped (repeated real roots), or underdamped (complex roots). For stable systems, the solution is typically expressed in terms of the natural frequency ω_n, damping ratio ζ, and damped natural frequency ω_d. The overdamped ($\zeta > 1$) and critically damped ($\zeta = 1$) solutions are similar to a first-order response. The underdamped response ($\zeta < 1$) has overshoot and oscillations. Simple formulas are given for the settling time, peak overshoot, and rise time of the underdamped response in terms of ω_n, ζ, and ω_d.

3.6.1 Overview of Solution

Consider the following second-order system:

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0 u(t) \tag{3.23}$$

$$\text{IC: } y(0) = 0,\ \dot{y}(0) = 0 \tag{3.24}$$

The coefficient of $\ddot{y}$ is normalized to simplify the derivation. If $a_2 \neq 1$ then the ODE can be normalized by dividing out a_2. This ODE also does not have the terms $b_2\ddot{u}(t)$ and $b_1\dot{u}(t)$. The impact of these terms is discussed in Section 3.7.

Section 3.3.1 provides a procedure to solve for the unit step response ($u(t) = 1$). First, the characteristic equation $s^2+a_1 s+a_0 = 0$ has two roots $s_{1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_0}}{2}$. The solution depends on the sign of the discriminant $a_1^2 - 4a_0$. The two roots can be distinct real ($a_1^2 - 4a_0 > 0$), repeated real ($a_1^2 - 4a_0 = 0$), or complex ($a_1^2 - 4a_0 < 0$). In all three cases, the roots have negative real part and the system is stable if and only if $a_1 > 0$ and $a_0 > 0$.

Unstable step responses can grow unbounded or remain bounded but oscillate, e.g. as shown in Figure 3.5 (right). As noted previously, control systems are typically designed to ensure stability. Hence the remainder of the section focuses on the stable case. For stable systems, the ODE coefficients are typically redefined as $a_1 := 2\zeta\omega_n$ and $a_2 := \omega_n^2$:

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = b_0 u(t) \tag{3.25}$$

Here ω_n is the <u>*natural frequency*</u> ($\frac{rad}{sec}$) and ζ is the <u>*damping ratio*</u> (unitless). These new coefficients are related to the original coefficients by $\omega_n = \sqrt{a_0}$ and $\zeta = \frac{a_1}{2\sqrt{a_0}}$. The stability conditions $a_1 > 0$ and $a_2 > 0$ imply that $\omega_n > 0$ and $\zeta > 0$. The two roots can be written as:

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \tag{3.26}$$

There are again three cases depending on the sign of the discriminant $\zeta^2 - 1$:

- **Overdamped,** $\zeta > 1$: There are two distinct real roots given by Equation 3.26.
- **Critically damped,** $\zeta = 1$: There are two repeated real roots at $s_{1,2} = -\omega_n$.
- **Underdamped,** $0 < \zeta < 1$: There are two complex roots $s_{1,2} = -\zeta\omega_n \pm j\,\omega_n\sqrt{1-\zeta^2}$ where $j := \sqrt{-1}$. The real part is $-\zeta\omega_n$. The imaginary part $\omega_d := \omega_n\sqrt{1-\zeta^2}$ is called

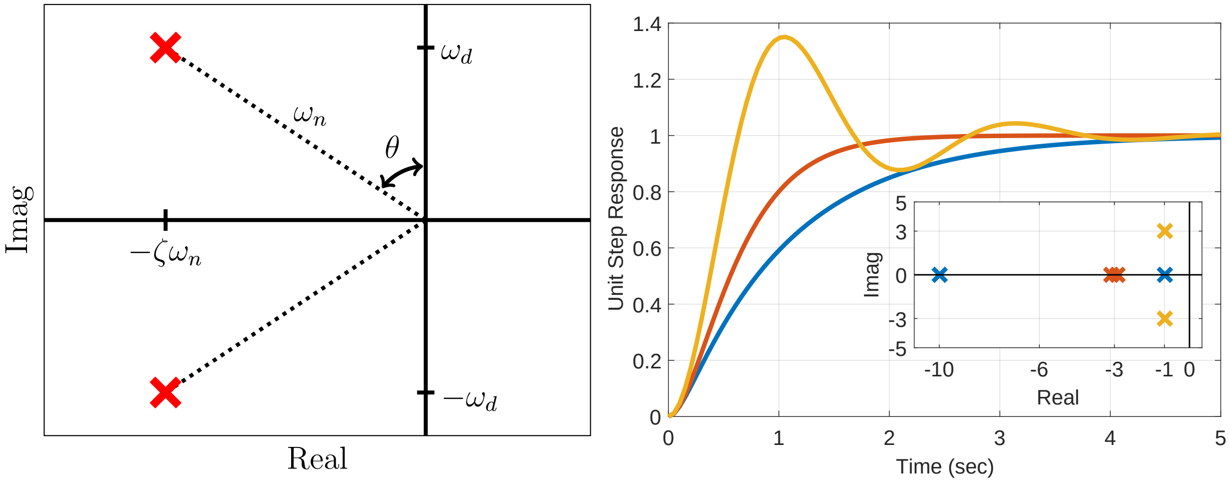


Figure 3.8: Notation for complex roots (left) and stable second-order unit step responses (right). The stable step responses are shown for $\frac{10}{s^2+11s+10}$, $\frac{9}{s^2+6s+9}$, and $\frac{10}{s^2+2s+10}$.

the <u>*damped natural frequency*</u> $(\frac{rad}{sec})$. The magnitude of each root is given by $|s_{1,2}| = \omega_n$. In addition, the damping ratio satisfies $\zeta = \sin\theta$. The left subplot of Figure 3.8 shows these various quantities in the complex plane for a complex pair of roots.

Figure 3.8 (right) shows one unit step response corresponding to each of the three cases above. The critically damped and overdamped responses both rise uniformly to the final value similar to a first-order response. The underdamped response has overshoot and oscillations before settling down. The unit step response solutions are discussed further below. By the principle of superposition, the response for non-unit step inputs $u(t) = \bar{u}$ can be obtained by scaling: if $y(t)$ is the unit step response then $\bar{u} \cdot y(t)$ is the response to a non-unit step $u(t) = \bar{u}$.

3.6.2 Overdamped

The system has two distinct real roots $s_1 < 0$ and $s_2 < 0$. The unit step response is:

$$y(t) = \bar{y}\left(1 - \frac{s_2}{s_2 - s_1}e^{s_1 t} + \frac{s_1}{s_2 - s_1}e^{s_2 t}\right) \text{ where } \bar{y} := \frac{b_0}{\omega_n^2} \tag{3.27}$$

Appendix 3.8.2 gives a brief derivation of this solution. Both exponentials in the response decay uniformly to zero and $y(t)$ converges to the final value $\bar{y}$. The response neither oscillates nor overshoots so peak overshoot and rise time are not defined. These features of an overdamped second-order step response are similar to those of a first-order response. The settling time is dominated by the "slower" root. Specifically, s_1 and s_2 have time constants $\tau_1 = \frac{1}{|s_1|}$ and $\tau_2 = \frac{1}{|s_2|}$. The "slower" root is closer to the imaginary axis and has a larger time constant. Assume s_1 is significantly slower than s_2, e.g. $\tau_1 > 10\tau_2$. This inequality implies $e^{s_2 t}$ decays much more quickly than $e^{s_1 t}$. It also implies $\frac{s_2}{s_2 - s_1} \approx 1$ and $\frac{s_1}{s_1 - s_2} \approx 0$. Thus the unit step response is approximately $y(t) \approx \bar{y}(1 - e^{-s_1 t})$. To summarize, $\tau_1 > 10\tau_2$ implies the response is approximately first-order with settling time $3\tau_1$. If both roots have similar time constants then the actual settling time for the overdamped response will be slightly larger than $3\tau_1$.

Example 3.6. Consider the second-order ODE $\ddot{y}(t) + 11\dot{y}(t) + 10y(t) = 10u(t)$ with corresponding transfer function $G(s) = \frac{10}{s^2+11s+10}$. The ODE coefficients are related to the damping ratio and natural frequency by $\omega_n^2 = 10$ and $2\zeta\omega_n = 11$. Hence $\omega_n = \sqrt{10} \approx 3.16\frac{rad}{sec}$ and $\zeta = \frac{11}{2\sqrt{10}} = 1.74$. This system is overdamped because $\zeta > 1$. The roots of the characteristic equation are $s_1 = -1$ and $s_2 = -10$ and the steady-state value for a unit step is $\bar{y} = 1$. The unit step response solution is $y(t) = 1 - \frac{10}{9}e^{-t} + \frac{1}{9}e^{-10t}$. This step response is shown in Figure 3.8 (right). The roots have time constants $\tau_1 = 1sec$ and $\tau_2 = \frac{1}{10}sec$. The root $s_1 = -1$ has a much slower response than the root $s_2 = -10$. Thus the response appears similar to the first order response $1 - e^{-t}$ and has settling time of $\approx 3sec$.

As another example, the system $G(s) = \frac{1.1}{s^2+2.1s+1.1}$ is overdamped and has roots at $s_1 = -1$ and $s_2 = -1.1$. The rough settling time is again $3\tau_1 = 3sec$ based on the slower root s_1. However, these two roots have similar time constants. As a result the actual response is closer to $4.5sec$ which is slower than given by our rough estimate from the slow root. The actual settling time was computed via numerical simulation with the `step` function. △

3.6.3 Critically Damped

The system has repeated real roots at $s_{1,2} = -\omega_n < 0$ and the unit step response is:

$$y(t) = \bar{y}\left(1 - (1 + \omega_n t)e^{-\omega_n t}\right) \text{ where } \bar{y} := \frac{b_0}{\omega_n^2} \tag{3.28}$$

Appendix 3.8.2 briefly derives this solution. The critically damped response converges uniformly to the final value $\bar{y}$ and neither oscillates nor overshoots. Thus peak overshoot and rise time are not defined. The time constant of the repeated roots is $\tau_{1,2} = \frac{1}{\omega_n}$ and the settling time is roughly $3\tau_{1,2}$. As noted in the previous subsection, the actual settling time is slightly slower than this rough estimate when the roots are similar. Note that $(1 + \omega_n t)e^{-\omega_n t} \approx 0.05$ when $t = 4.75\tau_{1,2}$. Hence the actual 5% settling time is $\approx 4.75\tau_{1,2}$ for repeated roots. However, we will still use the rough estimate $3\tau_{1,2}$ for convenience.

Example 3.7. Consider the second-order ODE $\ddot{y}(t)+6\dot{y}(t)+9y(t) = 9u(t)$ with corresponding transfer function $G(s) = \frac{9}{s^2+6s+9}$. The system is critically damped with $\omega_n = \sqrt{9} = 3\frac{rad}{sec}$ and $\zeta = \frac{6}{2\sqrt{9}} = 1.0$. The roots are repeated at $s_{1,2} = -3$ and the steady-state value for a unit step is $\bar{y} = 1$. The unit step response, as given above, is $y(t) = 1 - (1 + 3t)e^{-3t}$. This step response is shown in Figure 3.8 (right). The time constant of the repeated roots is $\tau_{1,2} = \frac{1}{3}sec$ and hence the rough settling time is $3\tau_{1,2} = 1sec$. However, the response in Figure 3.8 (right) takes $1.6sec$ ($\approx 4.75\tau_{1,2}$) to converge within 5%. This demonstrates that repeated roots converge more slowly than expected by the approximation $3\tau_{1,2}$. △

3.6.4 Underdamped

The system has complex roots at $s_{1,2} = -\zeta\omega_n \pm j\omega_d$ where $\omega_d := \omega_n\sqrt{1-\zeta^2}$. The unit step response, as derived in Appendix 3.8.2, is given by:

$$y(t) = \bar{y}\left(1 - e^{-\zeta\omega_n t}\cos(\omega_d t) - e^{-\zeta\omega_n t}\frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t)\right) \text{ where } \bar{y} := \frac{b_0}{\omega_n^2} \tag{3.29}$$

This response overshoots and oscillates before converging to the final value $\bar{y}$. The features of the underdamped response depend on the root locations in the complex plane. Figure 3.9 shows several responses with varying real part $-\zeta\omega_n$ (left) and varying imaginary part ω_d (right). The complex roots come in a conjugate pair and only the root with positive imaginary part is shown. The settling time is strongly affected by the real part while the rise time is strongly affected by the imaginary part. Figure 3.10 shows several responses with varying damping ratio ζ (left) and varying natural frequency ω_n (right). As noted previously, the complex roots have magnitude ω_n and $\zeta = \sin\theta$ where θ is the angle of the root from the imaginary axis (See Figure 3.8). Thus increasing ζ with fixed ω_n (left plot) causes the roots to move along a circle in the complex plane. At $\zeta = 0$, the roots are $s_{1,2} = \pm j\omega_n$. This is unstable and the response oscillates. The overshoot decreases significantly as ζ is increased. The right plot shows increasing ω_n with fixed ζ. This causes the roots to move on a line with fixed angle relative to the imaginary axis. This has a strong affect on settling and rise time. There are simple formulas that explain the key features of the underdamped responses shown in Figures 3.9 and 3.10. The following formulas are derived in Appendix 3.8.3:

- **Settling Time:** The time constant associated with the complex pair is $\tau_{1,2} = \frac{1}{|Re\{s_{1,2}\}|} = \frac{1}{\zeta\omega_n}$. The settling time is approximately $3\tau_{1,2}$. This behavior is observed in Figure 3.9. Specifically, the settling time decreases significantly as the real part $-\zeta\omega_n$ becomes more negative (left plot) but is not affected by changing the imaginary part ω_d (right plot).

- **Peak Overshoot:** The peak time and value are $T_p = \frac{\pi}{\omega_d}$ and $y(T_p) = \bar{y}(1 + e^{-\zeta\omega_n T_p})$. The peak overshoot is given by $M_p = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi}$. This behavior is observed in Figure 3.10. Specifically, the overshoot decreases significantly as ζ increases (left plot) but is not affected by changing the natural frequency ω_n (right plot).

- **Rise Time:** The rise time is $T_r = \frac{\frac{\pi}{2}+\sin^{-1}(\zeta)}{\omega_d}$. The rise time decreases significantly with increasing imaginary part ω_d as shown in the right plot of Figure 3.9. The dependence on ζ is more clear by expressing rise time as $T_r = \frac{\frac{\pi}{2}+\sin^{-1}(\zeta)}{\omega_n\sqrt{1-\zeta^2}}$. This form shows that increasing ω_n decreases rise time. This behaviors is seen in the right plot of Figure 3.10. The damping ratio ζ affects both the numerator and denominator. The numerator ranges from $\frac{\pi}{2}$ to π as ζ increases from 0 to 1. The dependence in the denominator is the stronger effect because $\frac{1}{\sqrt{1-\zeta^2}} \to \infty$ as $\zeta \to 1$. In summary, the rise time increases significantly as ζ increases. This behavior can be seen in the left plot of Figure 3.10.

Example 3.8. Consider the second-order ODE $\ddot{y}(t) + 2\dot{y}(t) + 10y(t) = 10u(t)$ with transfer function $G(s) = \frac{10}{s^2+2s+10}$. The system is underdamped with $\omega_n = \sqrt{10} \approx 3.2\frac{rad}{sec}$ and $\zeta = \frac{2}{2\sqrt{10}} = 0.32$. The roots of the characteristic equation are $s_{1,2} = -1 \pm 3j$ and the steady-state value for a unit step is $\bar{y} = 1$. The unit step response solution is $y(t) = 1 - e^{-t}\cos(3t) - \frac{1}{3}e^{-t}\sin(3t)$. This step response is shown in Figure 3.8 (right). The roots have time constant $\tau_{1,2} = 1sec$ and the settling time is $\approx 3sec$. The peak time and value are $T_p = \frac{\pi}{3}$ and $y(T_p) = 1 + e^{-\frac{\pi}{3}} \approx 1.35$. Hence the peak overshoot is $M_p = \frac{y(T_p)-\bar{y}}{\bar{y}} = 0.35$. The rise time is $T_r = \frac{\frac{\pi}{2}+\sin^{-1}(0.32)}{3} \approx 0.63sec$. △

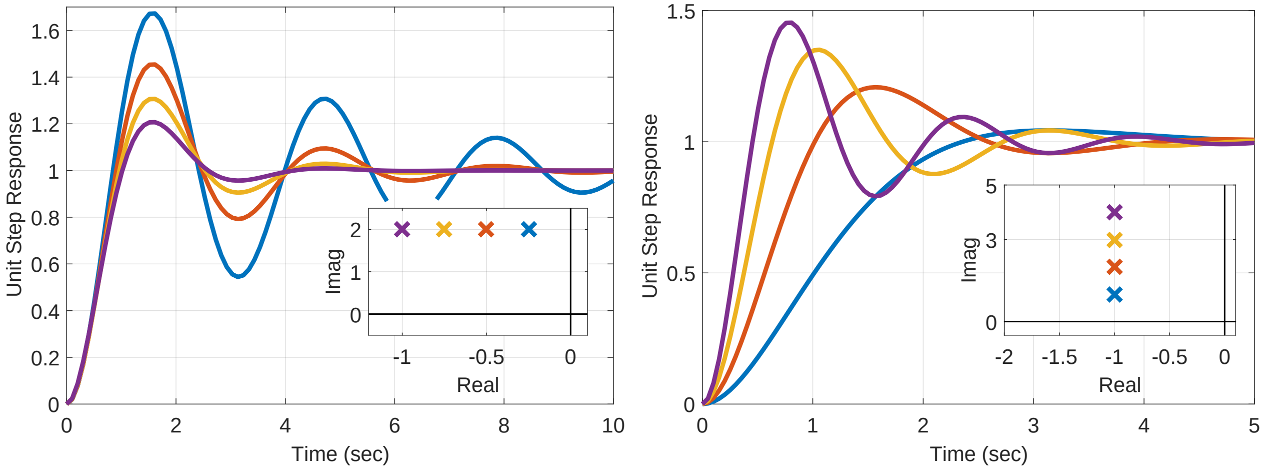


Figure 3.9: Underdamped step responses: Left plot has varying real part $-\zeta\omega_n = \{-1, -0.75, -0.5, -0.25\}$ and fixed imaginary part $\omega_d = 2$. Right plot has fixed real part $-\zeta\omega_n = -1$ and varying imaginary part $\omega_d = \{1, 2, 3, 4\}$.

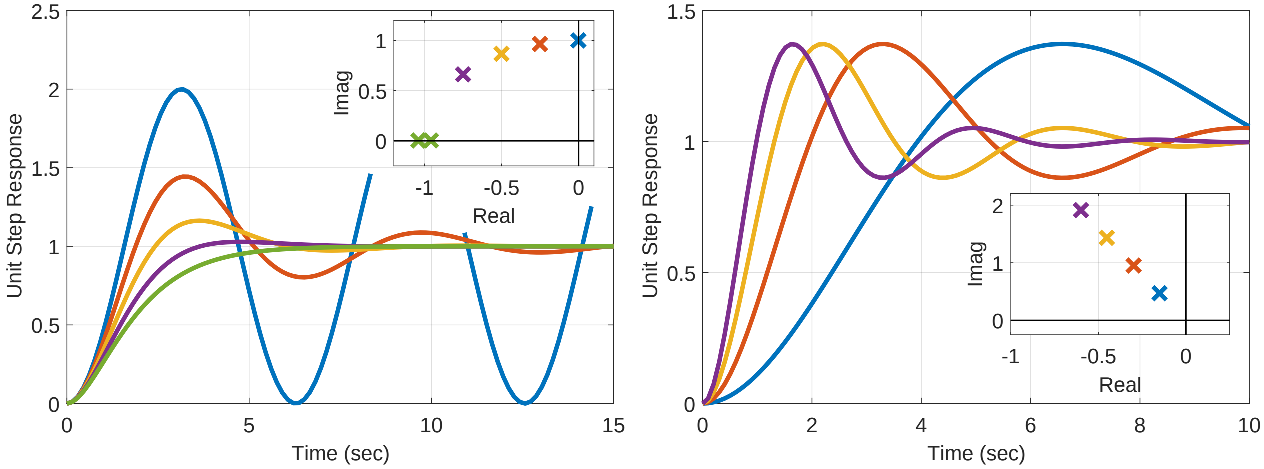


Figure 3.10: Underdamped step responses: Left plot has varying $\zeta = \{0, 0.25, 0.5, 0.75, 1\}$ and fixed $\omega_n = 1$. Right plot has fixed $\zeta = 0.3$ and varying $\omega_n = \{0.5, 1, 1.5, 2\}$.

3.7 Higher Order Systems

Summary: This section considers the dynamics of an n^{th} order ODE. First, we define the steady-state (DC) gain of a transfer function. Next the effect of an input derivative term $b_1 \dot{u}(t)$, i.e. a zero, is studied. Zeros in the LHP increase the step response overshoot and reduce the rise time. Zeros in the RHP cause undershoot. Finally, a method is given to construct a first or second-order ODE to approximate a general n^{th} order ODE.

3.7.1 Steady-state gain

The past few sections have studied the transient response characteristics for first and second order systems. This section provides some initial results for general n^{th} order systems including ODEs that contain derivatives of the input. A more complete understanding of general n^{th} order systems will be given using additional (frequency domain) tools in Chapter 5.

The transfer function corresponding to a general n^{th} order linear ODE is:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{3.30}$$

As noted previously, we can gain additional insight into the system dynamics by viewing G as a complex function. In particular, the value of the transfer function evaluated at $s = 0$, i.e. $G(0)$, has importance.

Definition 3.6. The *DC gain* or *steady-state gain* of the system is $G(0)$.

The DC gain is simply $G(0) = \frac{b_0}{a_0}$. As discussed in Section 3.4 the step response of a stable n^{th} order system converges to a final value $\bar{y} = \frac{b_0}{a_0}\bar{u}$. This final value is simply the DC gain multiplied by the step input size, i.e. $\bar{y} = G(0)\bar{u}$.

The Matlab command `dcgain` computes the DC gain of the system. The Matlab command `damp` is also useful for higher order systems. It displays a table with the damping ratio, natural frequency, and time constant associated with each pole. These commands are demonstrated in the example below. The help and documentation for these functions provides additional details.

Example 3.9. Various Matlab functions are demonstrated on the third-order system below.

```
>> G = tf([3042 1014],[1 13 199 507]);

>> zero(G) % Zeros are roots of 3024s+1014 = 0
ans =
   -0.3333

>> pole(G) % Poles are roots of s^3+13s^2+199s+507 = 0
ans =
  -5.0000 +12.0000i
  -5.0000 -12.0000i
  -3.0000 + 0.0000i

>> damp(G)  % Displays additional info for poles

         Pole              Damping       Frequency      Time Constant
                                       (rad/seconds)      (seconds)


 -3.00e+00                 1.00e+00       3.00e+00          3.33e-01
 -5.00e+00 + 1.20e+01i     3.85e-01       1.30e+01          2.00e-01
 -5.00e+00 - 1.20e+01i     3.85e-01       1.30e+01          2.00e-01

>> dcgain(G) % DC gain is G(0)= 1014/507 =2
ans =
    2
```

△

3.7.2 Zeros

Consider the following system given as both an ODE and transfer function:

$$G(s) = \frac{b_1 s + 4}{s^2 + 2s + 4}, \qquad \ddot{y}(t) + 2\dot{y}(t) + 4y(t) = b_1\dot{u}(t) + 4u(t) \tag{3.31}$$

This is a second order, underdamped system with $\omega_n = 2\frac{rad}{sec}$, $\zeta = 0.5$, and DC gain equal to 1. If $b_1 = 0$ then the system has no zeros but if $b_1 \neq 0$ then the system has a single zero at $z = -\frac{4}{b_1}$. The poles (roots of the characteristic equation) are $s = -1 \pm 1.73$. The results in Section 3.6.4 can be used to predict the features of the underdamped unit step response:

- **Settling Time:** $T_s = \frac{1}{\zeta\omega_n} \approx 3sec$
- **Peak Overshoot:** $M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \approx 0.16$
- **Rise Time:** $T_r = \frac{\frac{\pi}{2}+\sin^{-1}(\zeta)}{\omega_d} \approx 1.21sec$
- **Final Value**: $\bar{y} = G(0) = 1$

The effects of the zero can be determined from the solutions of the ODEs with and without the zero. Specifically, if y_0 is the response with $b_1 = 0$ then $y_1(t) = y_0(t) - \frac{1}{z}\dot{y}_0(t)$ is the response with $b_1 \neq 0$. Roughly, this can be shown by noting that if y_0 is the response due only to $b_0 u(t)$ then $\frac{b_1}{b_0}\dot{y}_0(t)$ is the response due only $b_1\dot{u}(t)$. Details are given in Appendix 3.8.4. The left plot of Figure 3.11 shows the response with various values of $b_1 \geq 0$. The inset diagram shows both the poles ('x') and zeros ('o'). All systems have have the same poles and only the pole with positive imaginary part is shown. The response (blue line) with $b_1 = 0$ (no zero) matches the second-order predictions given above. The responses with $b_1 = 0.8, 2, 4, 8$ correspond to zeros at $z = -5, -2, -1, -0.5$. **The effect of the left-half plane (LHP) zero is to increase overshoot and decrease rise time.** The LHP zero has negligible effect on the settling time

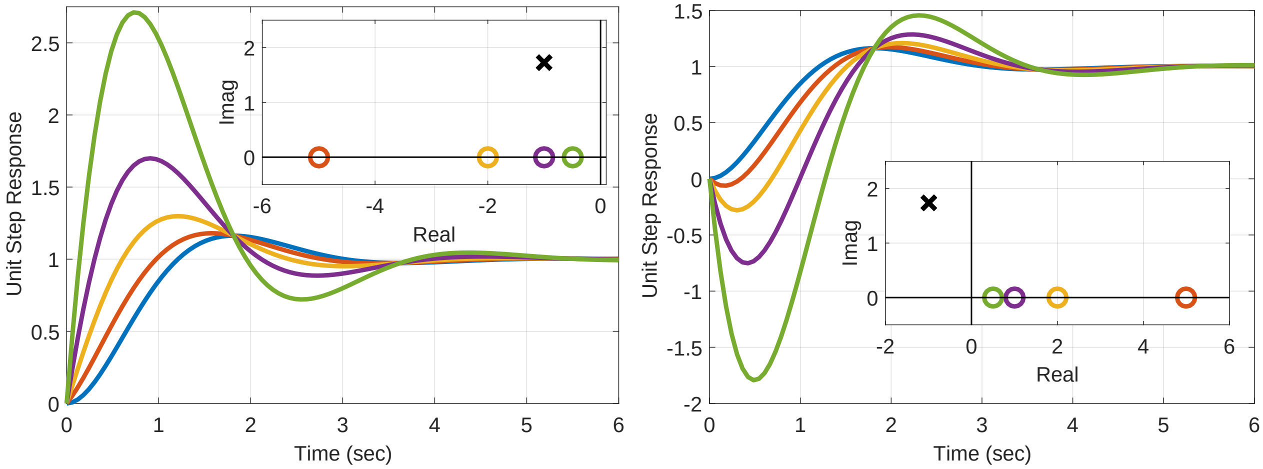


Figure 3.11: Unit step responses of G (Equation 3.31). Left plot has $b_1 = 0, 0.8, 2, 4, 8$ and $z =$ none, $-5, -2, -1, -0.5$. Right plot has $b_1 = 0, -0.8, -2, -4, -8$ and $z =$ none, $5, 2, 1, 0.5$.

and final value. This behavior follows from the relation $y_1(t) = y_0(t) - \frac{1}{z}\dot{y}_0(t)$. Specifically, $y_0(t)$ initially rises quickly and hence $\dot{y}_0(t)$ has a large positive value during this transient. If $z < 0$ then $y_0(t)$ and $-\frac{1}{z}\dot{y}_0(t)$ sum together to give a larger overshoot and shorter rise time in $y_1(t)$. However the zero has negligible effect on the settling time and final value because $\dot{y}_0(t) \to 0$ as $t \to \infty$. The effect on rise time and overshoot is inversely related to z. This effect is small if the zero is far in the LHP ($|z|$ is large) but increases as the zero moves toward the imaginary axis. The effect is significant when the zero is within a factor of ≈ 5 of the poles.

The right plot of Figure 3.11 shows the response with various values of $b_1 \leq 0$. The response (blue line) with $b_1 = 0$ (no zero) is again shown. The responses with $b_1 = -0.8, -2, -4, -8$ correspond to zeros at $z = 5, 2, 1, 0.5$. **The effect of the right-half plane (RHP) zero is to cause undershoot, i.e. the response initially goes in the wrong direction.** This behavior can again be explained from the relation $y_1(t) = y_0(t) - \frac{1}{z}\dot{y}_0(t)$. If $z > 0$ then $-\frac{1}{z}\dot{y}_0(t)$ causes a large negative initial transient leading to the undershoot. The RHP zero again has negligible effect on the settling time and final value because $\dot{y}_0(t) \to 0$ as $t \to \infty$. The undershoot effect is small if the zero is far in the RHP ($|z|$ is large) but increases as the zero moves right to left toward the imaginary axis. The effect is significant when the zero is within a factor of ≈ 5 of the poles.

3.7.3 Low Order Approximations

Many systems can be modeled by first or second order ODEs. This is a key reason for the detailed analysis of these cases in the previous sections. An understanding of first and second order systems is useful even when the dynamics are described by a general, n^{th}-order ODE. In many cases, the n^{th}-order ODE can be accurately approximated by a lower order ODE. As one case, the overdamped second order system in Example 3.6 has roots at $s_1 = -1$ and $s_2 = -10$. This second-order system is well-approximated by a first order system with root $s_1 = -1$. The "fast" root at $s_2 = -10$ tends to slow down the response. However, the root s_2 has a small effect when it is much faster than the slow root ($\tau_2 \leq 0.1\tau_1$).

The following procedure can be used to obtain a low order approximation for a stable system by removing "fast" dynamics. Consider the n^{th} order system:

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_0 u(t) \tag{3.32}$$

Let $\{s_1, \ldots, s_n\}$ denote the poles (roots) of the system ordered from slowest to fastest ($\tau_1 \geq \cdots \geq \tau_n$). The "slowest" poles are the dominant dynamics. There are two cases:

1. **Slowest root s_1 is real:** The unit step response solution is $y(t) = \bar{y} + c_1 e^{s_1 t} + c_2 e^{s_2 t} + \cdots c_n e^{s_n t}$ where $\bar{y} = \frac{b_0}{a_0}$ is the DC gain of the system. Assume s_1 is significantly slower than the remaining poles, e.g. $\tau_1 \geq 10\tau_2$. Then the exponential terms associated $\{s_2, \ldots, s_n\}$ decay faster than $e^{s_1 t}$. In addition, it can be shown that $c_1 \approx -\bar{y}$.[¶] Hence, the response is approximately first order $y(t) \approx \bar{y}(1 - e^{s_1 t})$. The low-order model is thus given by:

$$\dot{y}(t) - s_1 y(t) = -s_1 \bar{y}\, u(t) \tag{3.33}$$

   This first-order approximation retains the slow pole at s_1 and also has DC gain $\bar{y} = \frac{b_0}{a_0}$.

2. **Slowest roots s_1 and s_2 are complex:** Let ζ and ω_n be the damping ratio and natural frequency of the complex pair $s_{1,2}$. Assume $s_{1,2}$ are significantly slower than the remaining poles, e.g. $\tau_{1,2} \geq 10\tau_3$. Following similar steps as above, a low-order approximation is:

$$\ddot{y}(t) + 2\zeta\omega_n \dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 \bar{y}\, u(t) \tag{3.34}$$

   This second-order approximation retains the slow poles at $s_{1,2}$ and also has DC gain $\frac{b_0}{a_0}$.

Some care must be taken in applying this procedure. In particular, the neglected "fast" dynamics can contain significant transients before decaying to zero. Thus it is important to compare the step responses of the low order and n^{th} order systems to ensure the accuracy of the approximation. This procedure can be generalized to construct lower order models with zeros and/or more than two poles. However, this requires additional mathematical tools.

Example 3.10. The system $G_1(s) = \frac{9\times 10^4}{3s^4+291s^3+6270s^2+101400s+1.8\times 10^5}$ is fourth-order with DC gain of 0.5 and poles at $s = -2, -10\pm 17.3j, -75$. The slow pole at $s_1 = -2$ yields the first order approximation $G_{low,1}(s) = \frac{1}{s+2}$. Next, the system $G_2(s) = \frac{2.7\times 10^5}{s^5+98s^4+2194s^3+36555s^2+107100s+2.7\times 10^5}$ is fifth order with DC gain of 1.0 and poles at $s = -1.5 \pm 2.6j, -10 \pm 17.3j, -75$. The slow complex poles at $s_{1,2} = -1.5 \pm 2.6j$ have $\omega_n = 3\frac{rad}{sec}$ and $\zeta = 0.5$. This yields the second-order approximation $G_{low,2}(s) = \frac{9}{s^2+3s+9}$. Figure 3.12 shows that the low-order systems accurately approximate the step response of their corresponding higher order system. △

[¶] After some algebra, the zero ICs imply $c_1 = -\bar{y} \cdot \Pi_{k=2}^{n} \frac{s_k}{s_k - s_1}$. If s_1 is "slow" then $\frac{s_k}{s_k - s_1} \approx 1$ and $c_1 \approx -\bar{y}$.

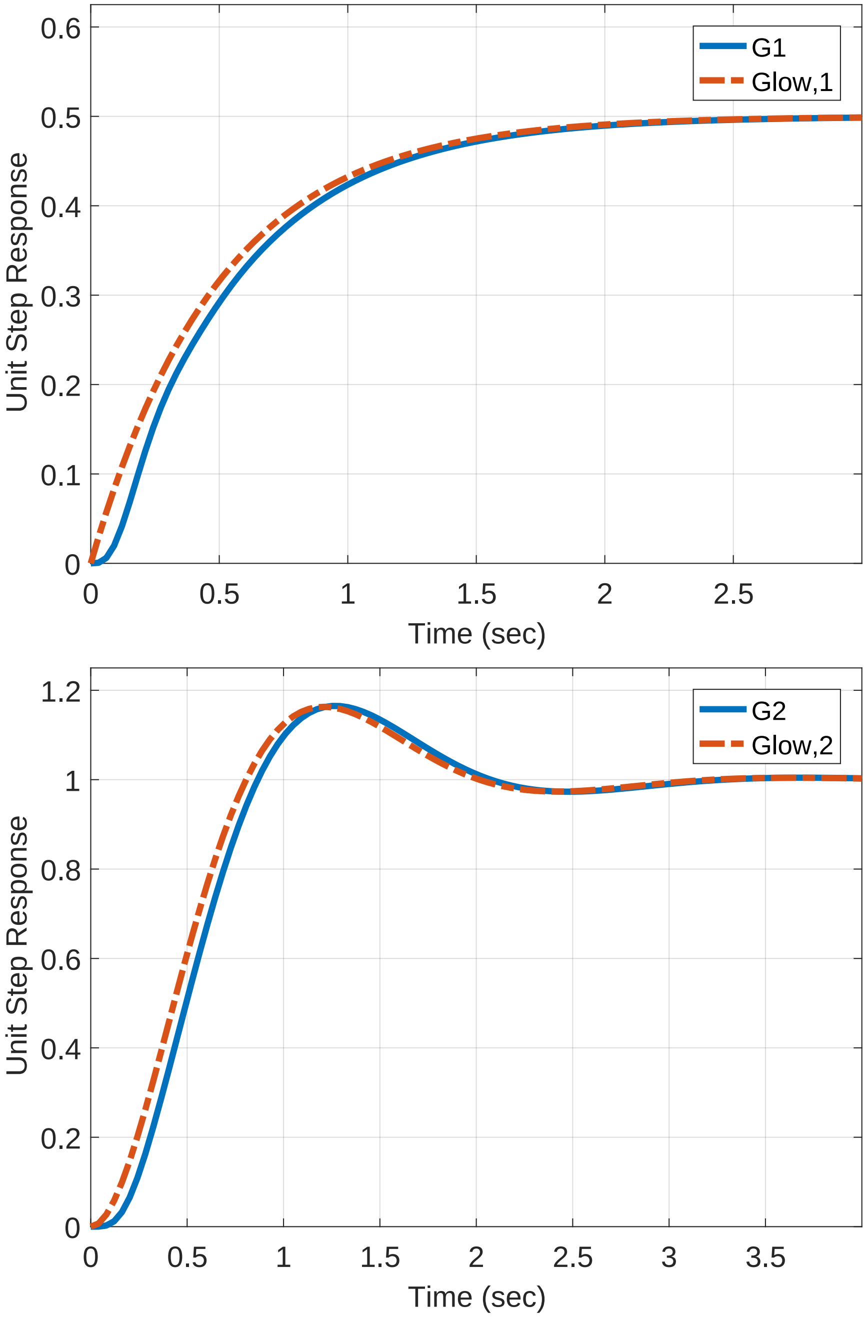


Figure 3.12: Unit step responses for $G_1(s)/G_{low,1}(s)$ (Left) and $G_2(s)/G_{low,2}(s)$ (Right).

3.8 Appendix: Background and Additional Results

Summary: This appendix provides a brief review of complex numbers and complex functions. Next, the appendix derives the unit step response solution for a stable second order system. Separate derivations are given for the overdamped, critically damped, and underdamped cases. The appendix also derives formulas to estimate the settling time, peak overshoot, and rise time for an underdamped unit step response. Finally, the appendix briefly describes the forced response solution for an ODE with input derivative term $b_1\dot{u}(t)$, i.e. with a single zero.

3.8.1 Complex Numbers and Functions

Let $j := \sqrt{-1}$ so that $j^2 = -1$. This unit imaginary number is often denoted as i but we'll use the notation j. A complex number $s \in \mathbb{C}$ has the form $s = a+jb$ where a and b are real numbers. As shown in Figure 3.13, s can be plotted in the complex plane with real part $Re\{s\} = a$ on the horizontal axis and imaginary part $Im\{s\} = b$ on the vertical axis. The magnitude and phase (angle) of this vector are given by $|s| = \sqrt{a^2+b^2}$ and $\phi = \tan^{-1}\left(\frac{b}{a}\right)$. The phase is expressed in units of $rads$ and is also denoted by $\angle r$. Conversely the real and imaginary parts can be expressed as $a = |s|\cos\phi$ and $b = |s|\sin\phi$. Thus the complex number can be rewritten as $s = |s|\cdot(\cos\phi + j\sin\phi)$. Euler's formula for any real number ϕ is $e^{j\phi} = \cos(\phi) + j\sin(\phi)$. This gives rise to the "polar" form for the complex number $s = |s|e^{j\phi}$. The notation $\angle s$ will often be used to represent the angle ϕ of a complex number. Finally, the complex conjugate of s is given by $\bar{s} = a - jb$. Note that $s\bar{s} = a^2 + b^2 = |s|^2$. This relation can be used to express fractions of complex numbers in real/imaginary form. For example, $\frac{1}{s}$ can be equivalently written as $\frac{\bar{s}}{s\bar{s}} = \frac{a-jb}{a^2+b^2}$. Hence $Re\{\frac{1}{s}\} = \frac{a}{a^2+b^2}$ and $Im\{\frac{1}{s}\} = \frac{-b}{a^2+b^2}$.

Example 3.11. The complex number $s = 3+4j$ has magnitude $|s| = \sqrt{3^2+4^2} = 5$ and phase $\angle s = \tan^{-1}(\frac{4}{3}) = 0.93rads$. The polar form is $s = 5e^{j0.93}$. The complex conjugate is $\bar{s} = 3-4j$. Finally, $\frac{1}{s} = \frac{\bar{s}}{|s|^2} = \frac{3-4j}{25}$. Hence $Re\{\frac{1}{s}\} = 0.12$ and $Im\{\frac{1}{s}\} = -0.16$. △

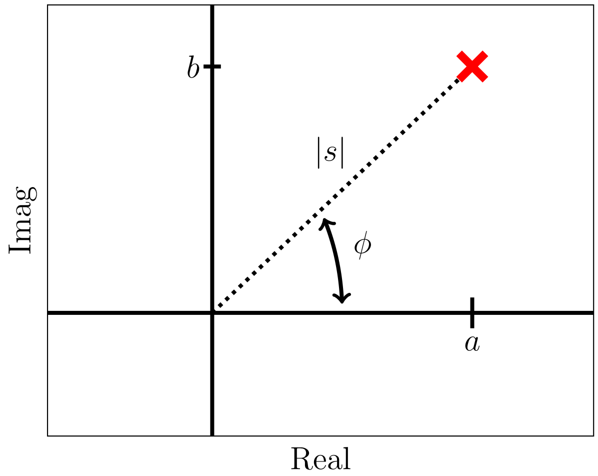


Figure 3.13: Complex number $s = a + jb = |s|e^{j\phi}$.

The polar form can also be used to obtain simple expressions for products of two complex numbers. Let s_1 and s_2 be two complex numbers with polar form $|s_1|e^{j\angle s_1}$ and $|s_2|e^{j\angle s_2}$. Their product is given by $s = s_1 s_2 = |s_1||s_2|\, e^{j(\angle s_1 + \angle s_2)}$. Thus for a product of complex numbers the phases add, i.e. $\angle s = \angle s_1 + \angle s_2$, and the magnitudes multiply, i.e. $|s| = |s_1||s_2|$.

Example 3.12. Consider two complex numbers $s_1 = 3 + 4j$ and $s_2 = 12 + 5j$. Their product $s = s_1 s_2$ can be computed by multiplying the real and imaginary parts:

$$s = (3 + 4j) \cdot (12 + 5j) = 16 + 63j \tag{3.35}$$

This product has magnitude $|s| = 65$ and phase $\angle s = \tan^{-1}(\frac{63}{16}) = 1.32 rad$. This product can also be computed using the polar form. Specifically, s_1 has magnitude $|s_1| = 5$ and phase $\angle s_1 = 0.93 rads$ as shown in the previous example. Similarly, s_2 has magnitude $|s_2| = \sqrt{12^2 + 5^2} = 13$ and phase $\angle s_2 = \tan^{-1}(\frac{5}{12}) = 0.39 rads$. Thus the product $s = s_1 s_2$ has magnitude $|s| = 5 \cdot 13 = 65$ and phase $\angle s = 0.93 + 0.39 = 1.32 rad$. The polar form for computing products will be more convenient later in the text. △

Insight into system dynamics can be gained by viewing the transfer function as a complex function. Specifically a function $G : \mathbb{C} \to \mathbb{C}$ is a complex function if it takes a complex number $s \in \mathbb{C}$ as input and returns a complex number $G(s) \in \mathbb{C}$. Thus the input s has both real/imaginary part (or equivalently magnitude/phase) and the returned value $G(s)$ also has both real/imaginary part (or magnitude/phase). A simple example is provided below.

Example 3.13. Let $G(s) = \frac{1}{s-2}$ be a complex function. The function can be evaluated at any complex number s. At the particular value $s_0 = 1 + 5j$ the function is $G(s_0) = \frac{1}{-1+5j}$. The real and imaginary parts of $G(s_0)$ can be computed by multiplying the numerator and denominator by the conjugate $-1 - 5j$. This yields $G(s_0) = \frac{-1-5j}{26}$. The magnitude and phase are $|G(s_0)| = \frac{1}{26}$ and $\angle G(s_0) = -1.77 rads$. Since s and $G(s)$ are both complex numbers it is difficult to visualize this function. The right plot of Figure 3.14 shows only the magnitude $|G(s)|$ versus the complex variable s. The vertical axis is $|G(s)|$ while the horizontal axes are the real and imaginary parts of s. Notice that $|G(s)|$ shoots to infinity when $s = 2$, i.e $|G(s)| = \infty$ when $Re\{s\} = 2$ and $Im\{s\} = 0$. This plot looks like a tent with a pole placed at $s = 2$ holding up the tent. Thus we say that $G(s)$ has a ”pole” at $s = 2$ as defined more precisely in Section 3.3.2. △

3.8.2 Stable Second-Order Step Response Solution

The ODE for stable second-order system can be written as:

$$\ddot{y}(t) + 2\zeta\omega_n \dot{y}(t) + \omega_n^2 y(t) = b_0 u(t) \tag{3.36}$$

Overdamped

The overdamped case corresponds to $\zeta > 1$ with two distinct roots $s_1 < 0$ and $s_2 < 0$. A particular solution for a unit step input is $\bar{y} = \frac{b_0}{\omega_n^2}$. Hence the general solution is:

$$y(t) = \bar{y} + c_1 e^{s_1 t} + c_2 e^{s_2 t} \tag{3.37}$$

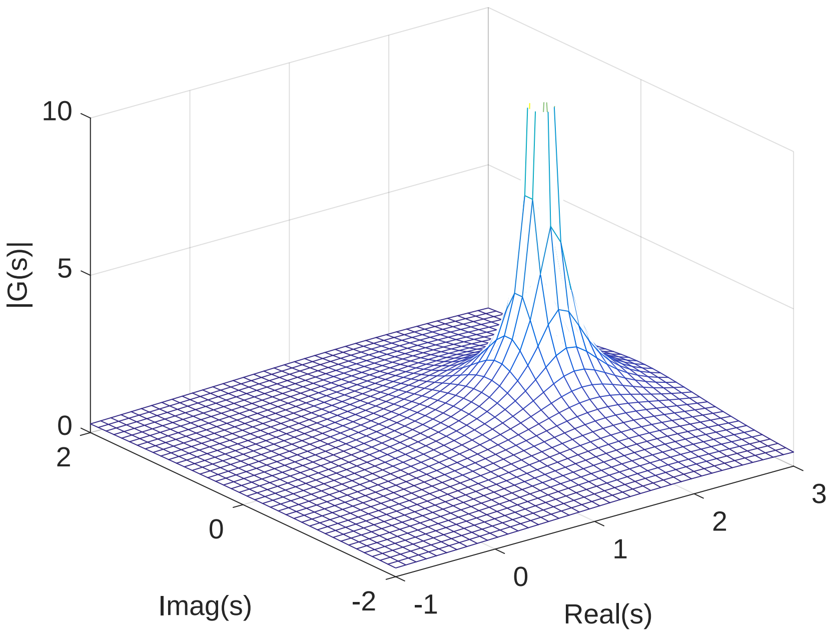


Figure 3.14: Magnitude of $G(s) = \frac{1}{s-2}$.

The ICs can be used to solve for c_1 and c_2:

$$0 = y(0) = \bar{y} + c_1 + c_2 \tag{3.38}$$

$$0 = \dot{y}(0) = s_1 c_1 + s_2 c_2 \tag{3.39}$$

These equations are solved by $c_1 = -\bar{y}\frac{s_2}{s_2-s_1}$ and $c_2 = \bar{y}\frac{s_1}{s_2-s_1}$. Hence the unit step response is:

$$y(t) = \bar{y}\left(1 - \frac{s_2}{s_2 - s_1}e^{s_1 t} + \frac{s_1}{s_2 - s_1}e^{s_2 t}\right) \text{ where } \bar{y} := \frac{b_0}{\omega_n^2} \tag{3.40}$$

Critically Damped

The critically damped case corresponds to $\zeta = 1$ with two repeated real roots at $s_{1,2} = -\omega_n$. Sections 3.2 and 3.3 give a general solution procedure for free and forced responses but only for the case of distinct roots. This procedure must be slightly modified if the system has repeated roots. In particular, if there are repeated roots at $s_{1,2} = -\omega_n$ then the general homogeneous solution takes the form $c_1 e^{-\omega_n t} + c_2 t e^{-\omega_n t}$. A particular solution for a unit step input is $\bar{y} = \frac{b_0}{\omega_n^2}$. Hence the general solution is:

$$y(t) = \bar{y} + c_1 e^{-\omega_n t} + c_2 t e^{-\omega_n t} \tag{3.41}$$

The ICs can be used to solve for c_1 and c_2:

$$0 = y(0) = \bar{y} + c_1 \tag{3.42}$$

$$0 = \dot{y}(0) = -\omega_n c_1 + c_2 \tag{3.43}$$

These equations are solved by $c_1 = -\bar{y}$ and $c_2 = -\omega_n \bar{y}$. Hence the unit step response is:

$$y(t) = \bar{y}\left(1 - (1+\omega_n t)e^{-\omega_n t}\right) \text{ where } \bar{y} := \frac{b_0}{\omega_n^2} \tag{3.44}$$

Underdamped

The underdamped case corresponds to $0 < \zeta < 1$ with complex roots at $s_{1,2} = -\zeta\omega_n \pm j\omega_d$ where $\omega_d := \omega_n\sqrt{1-\zeta^2}$. A particular solution for a unit step input is $\bar{y} = \frac{b_0}{\omega_n^2}$. Hence the general solution is:

$$y(t) = \bar{y} + c_1 e^{s_1 t} + c_2 e^{s_2 t} \tag{3.45}$$

The complex exponential terms can be re-written as real terms as discussed in Section 3.2:

$$y(t) = \bar{y} + \tilde{c}_1 e^{-\zeta\omega_n t}\cos(\omega_d t) + \tilde{c}_2 e^{-\zeta\omega_n t}\sin(\omega_d t) \tag{3.46}$$

The ICs can be used to solve for $\tilde{c}_1$ and $\tilde{c}_2$:

$$0 = y(0) = \bar{y} + \tilde{c}_1 \tag{3.47}$$

$$0 = \dot{y}(0) = -\zeta\omega_n\tilde{c}_1 + \omega_d\tilde{c}_2 \tag{3.48}$$

These equations are solved by $\tilde{c}_1 = -\bar{y}$ and $\tilde{c}_2 = -\bar{y}\frac{\zeta}{\sqrt{1-\zeta^2}}$. Hence the unit step response is:

$$y(t) = \bar{y}\left(1 - e^{-\zeta\omega_n t}\cos(\omega_d t) - e^{-\zeta\omega_n t}\frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t)\right) \text{ where } \bar{y} := \frac{b_0}{\omega_n^2} \tag{3.49}$$

3.8.3 Underdamped Step Response Features

This section derives formulas to estimate the settling time, peak overshoot, and rise time for an underdamped unit step response.

Settling Time

The following trigonometric identity holds for any real numbers c_1, c_2, and α:

$$c_1\cos(\alpha) + c_2\sin(\alpha) = \sqrt{c_1^2 + c_2^2}\cos\left(\alpha - \tan^{-1}\left(\frac{c_2}{c_1}\right)\right) \tag{3.50}$$

This identity can be used to rewrite the underdamped unit step response (Equation 3.49) in the following alternative form:

$$y(t) = \bar{y}\left(1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos\left(\omega_d t - \theta\right)\right) \text{ where } \theta := \tan^{-1}\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right) \tag{3.51}$$

The quantity θ is the same as the angle θ shown in the left subplot of Figure 3.8. The cosine term oscillates between $[-1, +1]$. Hence, the underdamped response can never be larger than

$\bar{y}\left(1+\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\right)$ and never be smaller than $\bar{y}\left(1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\right)$. These two curves form an envelope that bounds the step response. The upper and lower envelopes converge within ±5% of the final value $\bar{y}$ when the following inequality holds:

$$e^{-\zeta\omega_n t} \leq 0.05\sqrt{1-\zeta^2} \tag{3.52}$$

For small damping ratios, e.g $\zeta < 0.5$, this inequality holds approximately for $t \geq \frac{3}{\zeta\omega_n}$. Recall that the underdamped roots $s_{1,2} = -\zeta\omega_n \pm j\omega_d$ have a time constant $\tau_{1,2} = \frac{1}{\zeta\omega_n}$. Thus the 5% settling time is approximately $3\tau_{1,2}$ when $\zeta < 0.5$. The speed of response slows slightly and the settling time increases for larger values of $\zeta \in [0.5, 1]$. In fact, the settling time for a critically damped system ($\zeta = 1$) is $4.75\tau_{1,2}$ as discussed in Section 3.6.3. Thus the settling time for an underdamped system increases to $4.75\tau_{1,2}$ as $\zeta \to 1$. However, we will still use the rough $3\tau_{1,2}$ approximation for the 5% settling time of an underdamped system for all values of ζ.

Peak Overshoot

The underdamped response rises to the peak value and then oscillates as it decays to the steady-state value. The local maxima and minima of the response occur at times where $\dot{y}(t) = 0$. Differentiating the underdamped unit step response in Equation 3.49 and simplifying yields:

$$\dot{y}(t) = \bar{y}e^{-\zeta\omega_n t}\frac{\omega_n}{\sqrt{1-\zeta^2}}\sin(\omega_d t) \tag{3.53}$$

Hence $\dot{y}(t) = 0$ if and only if $\sin(\omega_d t) = 0$. Thus the local maxima and minima occur when $\omega_d t = \pi, 2\pi, 3\pi, \ldots$. The largest (global peak) of the step response occurs at the first local maxima: $T_p = \frac{\pi}{\omega_d}$. The corresponding peak value is:

$$y(T_p) = \bar{y}\left(1 + e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}\right) \tag{3.54}$$

The peak overshoot is $M_p = \frac{y(T_p)-\bar{y}}{\bar{y}}$. Substituting the peak value into this formula yields:

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{3.55}$$

Rise Time

Substitute $y(t) = \bar{y}$ into the alternative form for the underdamped solution (Equation 3.51):

$$\bar{y} = \bar{y}\left(1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos\left(\omega_d t - \theta\right)\right) \text{ where } \theta := \tan^{-1}\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right) \tag{3.56}$$

After some simplification, the condition for $y(t) = \bar{y}$ is:

$$0 = \cos\left(\omega_d t - \theta\right) \tag{3.57}$$

Thus $y(t) = \bar{y}$ for times such that $\omega_d t - \theta = \frac{\pi}{2}, \frac{3\pi}{2}, \ldots$. The rise time T_r corresponds to the first time that $y(T_r) = \bar{y}$, i.e.:

$$T_r = \frac{\theta + \frac{\pi}{2}}{\omega_d} \tag{3.58}$$

3.8.4 Forced Response With a Zero

Consider the general n^{th} order system with zero ICs:

$$a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_1 \dot{u}(t) + b_0 u(t) \tag{3.59}$$

$$\text{IC: } y(0) = 0; \; \ldots \; y^{[n-1]}(0) = 0 \tag{3.60}$$

In addition, define two transfer functions:

$$G_0(s) = \frac{b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{3.61}$$

$$G_1(s) = \frac{b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{3.62}$$

G_1 is the transfer function associated with the ODE in Equation 3.59. It has a single zero at $z = -\frac{b_0}{b_1}$. G_0 is the transfer function for another system with the same poles but without the zero. The goal of this section is to related the forced responses of these two systems.

Let u be a differentiable input (not a step) with $u(0) = 0$. In addition, let y_0 and y_1 denote the forced responses of G_0 and G_1 with this input. Hence the pair (u, y_1) satisfy the ODE in Equation 3.59 while the pair (u, y_0) satisfy

$$a_n y_0^{[n]}(t) + a_{n-1} y_0^{[n-1]}(t) + \cdots + a_1 \dot{y}_0(t) + a_0 y_0(t) = b_0 u(t) \tag{3.63}$$

Differentiate both sides of Equation 3.63 and multiply by $\frac{b_1}{b_0}$ to obtain:

$$a_n \frac{d^n}{dt^n}\left(\frac{b_1}{b_0}\dot{y}_0\right) + \cdots a_1 \left(\frac{b_1}{b_0}\dot{y}_0\right) + a_0 \left(\frac{b_1}{b_0}\dot{y}_0\right) = b_1 \dot{u}(t) \tag{3.64}$$

Sum Equations 3.63 and 3.64 to show that $y_0(t) + \frac{b_1}{b_0}\dot{y}_0 = y_0(t) - \frac{1}{z}\dot{y}_0(t)$ satisfies the ODE in Equation 3.59 associated with G_1. Moreover, $y_0(t) - \frac{1}{z}\dot{y}_0(t)$ satisfies the zero ICs. This follows because y_0 has zero ICs ($y_0(0) = \cdots = y_0^{[n-1]}(0) = 0$) and it can also be shown that $u(0) = 0$ implies $y^{[n]}(0) = 0$. Finally, the forced response solution to Equation 3.59 with zero ICs is unique. Hence $y_1(t) = y_0(t) - \frac{1}{z}\dot{y}_0(t)$.

One remaining technical point to address are the assumptions that $u(t)$ is differentiable and $u(0) = 0$. Step inputs jump discontinuously at $t = 0$ and hence these inputs are not differentiable. A step input of magnitude $\bar{u}$ is handled via the following approximation:

$$u_\epsilon(t) = \begin{cases} \frac{\bar{u}}{\epsilon} t & 0 \le t < \epsilon \\ \bar{u} & t \ge \epsilon \end{cases} \tag{3.65}$$

For all $\epsilon > 0$ this function $u_\epsilon(t)$ is differentiable and satisfies $u_\epsilon(0) = 0$. Hence the discussion above is valid. As $\epsilon \to 0$, the input $u_\epsilon(t)$ converges to a step input and the outputs converge to the corresponding step responses. Thus the relation $y_1(t) = y_0(t) - \frac{1}{z}\dot{y}_0(t)$ holds even for step inputs. Precise details require the use of generalized functions called “distributions”.

Chapter 4

PID Control

This chapter covers the basics of proportional-integral-derivative (PID) control. Section 4.1 summarizes the key design issues associated with modeling and control of systems. Section 4.2 describes an open-loop control strategy. This open-loop strategy requires no additional sensors but typically has poor performance. This motivates the use of feedback. Specifically, Section 4.3 describes a simple feedback strategy known as proportional control. This strategy computes an error by comparing the desired reference with a measurement of the plant output. The control command is set proportional to this error. Next, Sections 4.4 and 4.5 describe two additional feedback strategies known as proportional-integral (PI) and proportional-derivative (PD) control. Two examples of PID design are given in Section 4.6 Finally, modifications to basic PID and implementation details are given in Sections 4.7 and 4.8.

4.1 Summary of Control Design Issues

Summary: This section summarizes the key design issues associated with modeling and control of systems. The discussion focuses on a DC motor for concreteness. There are two important points. First, the models used for control design are often simplified and contain a variety of inaccuracies including uncertain parameters, unmodeled dynamics, nonlinear effects, and implementation effects. It is common to design the controller with the simplified model and then check performance on a more accurate model. Second, the control design involves many, often conflicting, objectives and hence performance trade-offs are required.

4.1.1 DC Motor Model

This section summarizes the design issues associated with modeling and control. Sections 4.2-4.4 then study various control strategies for first-order systems. The control design will be based on the dynamic characteristics of low-order systems as studied in the previous chapter. As a concrete example, consider a small DC motor driving an inertia as shown in Figure 4.1 (left). The input is the voltage u (V) applied to the motor and the output is the angular velocity y $(\frac{rad}{sec})$ of the motor shaft. The goal is to develop strategies to control the motor speed using the input voltage. This specific design problem is applicable to DC motors found in a variety of applications including multicopters or small fixed-wing aircraft. However, the discussion covers issues that are broadly applicable to most control systems.

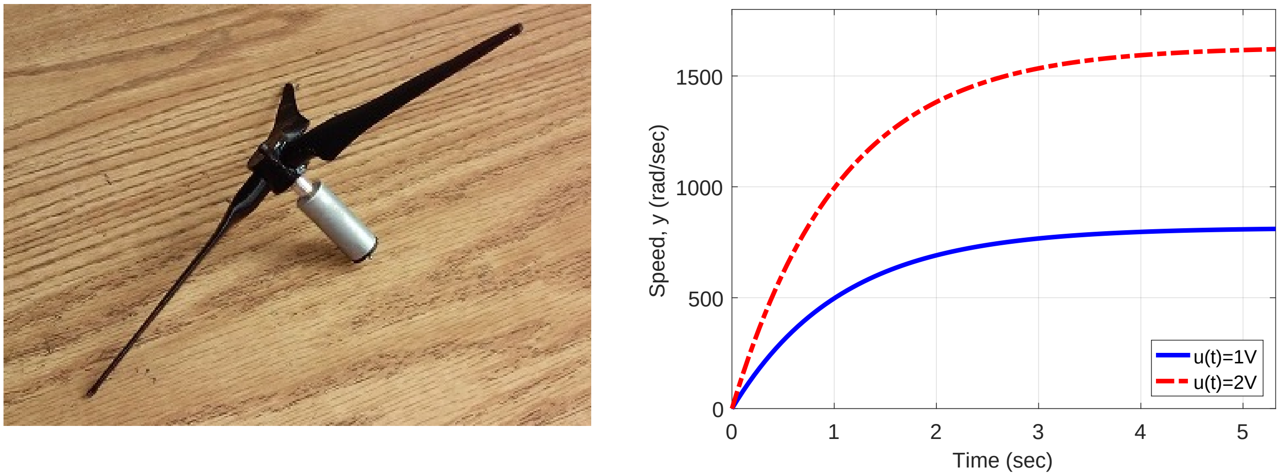


Figure 4.1: Left: Precision Microdrives DC Motor 112-002 with small inertia. Right: Step responses for "nominal" DC motor model.

The control design will use a model of the DC motor. Modeling the system is an important step in any control design. As mentioned previously, modeling is a domain specific task, i.e. it typically requires knowledge specific to an individual field of engineering. Thus we will not emphasize the motor modeling here but details can be found in Appendix 4.9.1. These modeling details are not required to understand the control design and can be skipped.

Briefly, the motor involves coupled electrical and mechanical (rotational inertia) dynamics. We'll assume the electrical dynamics are much faster than the mechanical dynamics. This leads to a first-order ODE model for the motor:

$$\begin{aligned} &\dot{y}(t) + a_0\, y(t) = b_0\, u(t) + b_0\, d(t) \\ &\text{where: } a_0 = 0.94 \frac{1}{sec} \text{ and } b_0 = 766.8 \frac{rad}{sec^2\, V} \end{aligned} \tag{4.1}$$

The model coefficients a_0 and b_0 were determined from the specification sheet for a Precision Microdrives DC Motor 112-002 [14]. In addition, the ODE includes another input $d(t)$ (V) to model the effects of environmental disturbances. For example, DC motors are used on multicopters to rotate propellers and generate thrust. This creates a reaction load torque that opposes the motor shaft rotation. This reaction torque depends on the multicopter velocity and attitude (angular orientation). Such load torques are difficult to model precisely and instead the simple disturbance input $d(t)$ (V) is used to capture these effects.

Figure 4.1 (right) shows step responses for the motor model with $u(t) = 1V$ and $u(t) = 2V$. The transfer function for the first-order model is $G(s) = \frac{766.8}{s+0.94}$. This has a pole at $s_1 = -0.94$, time constant $\tau_1 = 1.06\, sec$, and steady state gain $G(0) = 815.7 \frac{rad}{sec\, V}$. The step response characteristics in Figure 4.1 agree with these values. Specifically, both responses have a settling time of $3\tau_1 = 3.18\, sec$. Moreover, the final values are $G(0) \cdot 1V = 815.7 \frac{rad}{sec}$ and $G(0) \cdot 2V = 1631.4 \frac{rad}{sec}$ for the responses with $u(t) = 1V$ and $u(t) = 2V$. These responses are labeled as "nominal" because the first order model was developed under certain approximations. As a result, the step responses of an actual Precision Microdrives DC Motor 112-002 will likely differ from these nominal model responses. This is discussed further in the next subsection.

4.1.2 Model Simplifications and Uncertainties

It is important to recognize that the models used for control design are often simplified and contain a variety of uncertainties/inaccuracies:

- **Uncertain Parameters:** The ODE coefficients a_0 and b_0 depend on various parameters including motor resistance, inductance, and inertia. The values of a_0 and b_0 given in Equation 4.1 were computed using typical motor parameter values obtained from the specification sheet. These parameters can (and do) vary from motor to motor due to manufacturing tolerances. As a consequence, the ODE coefficients a_0 and b_0 will vary with some uncertainty. For example, the left plot of Figure 4.2 shows the unit step response with the nominal values $a_0 = 0.94$ and $b_0 = 766.8$ (solid blue). It also shows four additional responses (dashed red) with ±10% variation in these coefficients: $(a_0, b_0) = \{(0.85, 690.1), (0.85, 843.5), (1.03, 690.1), (1.03, 843.5)\}$. Note the small change in settling time and the significant effect on the steady state gain.

- **Unmodeled Dynamics:** The model used for control design typically neglects some dynamics. In many cases, these unmodeled dynamics are neglected on purpose to simplify the control design. For example, the first order, linear ODE (Equation 4.1) captures the dominant rotational inertia dynamics. It neglects the "fast" electrical dynamics.

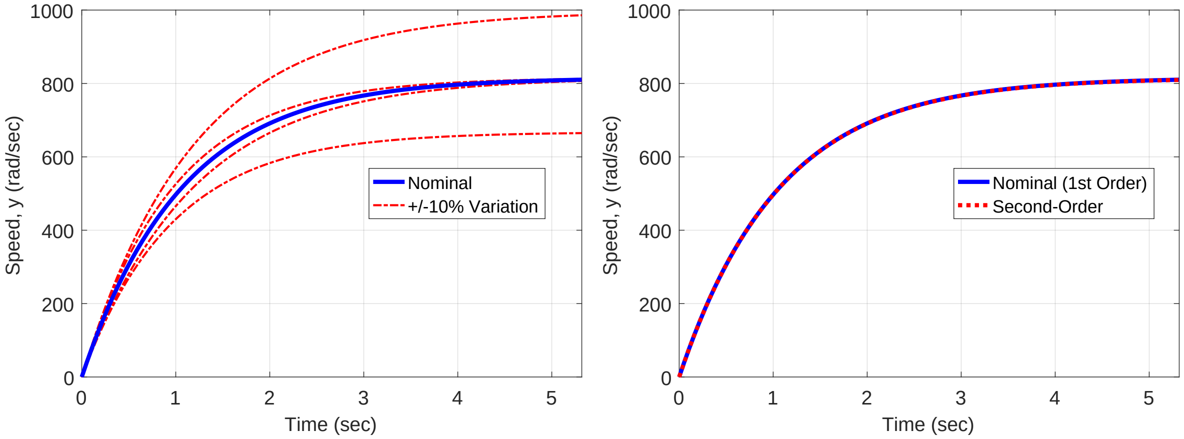


Figure 4.2: Left: Unit step responses with nominal ODE coefficients and ±10% variation. Right: Unit step response of nominal first-order response and second-order response.

Appendix 4.9.1 derives a second-order model that captures these unmodeled electrical dynamics. The transfer function of the second order model is:

$$G_2(s) = \frac{1.23 \times 10^{-3}}{3 \times 10^{-10}\, s^2 + 1.6 \times 10^{-6}\, s + 1.505 \times 10^{-6}} \tag{4.2}$$

The right subplot of Figure 4.2 shows the unit step response with the nominal first-order model (solid blue) and the second order model (dashed red). The responses are almost identical and cannot be distinguished on the plot. The fast electrical dynamics appear to be negligible. However, these unmodeled dynamics may impact the system performance depending the design of the controller.

- **Nonlinear Effects:** Our linear ODE models also neglect nonlinear effects. For example, the linear ODE may be an approximation of a nonlinear ODE obtained via linearization (as discussed in Section 2.2). As another example, the DC motor has a maximum operating voltage of $u_{max} = 3V$. The input must saturate (i.e. not exceed) u_{max} otherwise the motor could be damaged. This saturation is a nonlinear effect that is not included in the linear ODE model. It is difficult to analyze the effect of such nonlinearities. In many cases simulations are used to determine their impact on performance. This is discussed further in Section XXX.

- **Implementation Effects:** The models used for control design in these notes are in the form of linear ODEs. The control algorithms designed in the following sections will also be given in the form of linear ODEs. In most cases the control algorithm must be implemented on a microprocessor using measurements obtained at specific sampling intervals. This raises a number of implementation effects that are not captured by our linear ODE models. These implementation effects can typically be neglected during the design of the controller. This is discussed further in Section 4.8.

Each of these issues must be considered in the control design. Certain effects may be negligible for a particular problem while others may be significant. A typical approach is to design the controller using a simplified linear ODE model of the system. The controller is then checked on a more accurate, higher fidelity model that includes some/all of these additional effects. For example, the controller performance can be evaluated by simulation using a model that includes unmodeled dynamics, nonlinear terms, and/or implementation details. In addition, simulations can be performed with different parameter values, e.g. ±10% variation in ODE coefficients. We will explore some of these effects further in the DC motor control design.

4.1.3 Control Design Objectives

There are numerous, often conflicting, objectives in the design of most control systems. This typically necessitates performance trade-offs. This section discusses common objectives in the context of the DC motor control design problem. For this problem, the goal is to have the motor speed $y(t)$ follow a desired reference speed $r(t)$. The tracking error is defined to be $e(t) := r(t) - y(t)$. The steady state (or final value) for the error is denoted e_{ss}. Specific design requirements often include the following:

- **Stability:** At a minimum, the combination of the controller and the plant should be (free response and BIBO) stable as defined in Chapter 3.

- **Reference Tracking:** The controller should be designed so that the system output tracks the desired reference command. In particular, the system output should follow reference commands with small overshoot and steady state error. In addition, the system should respond quickly, i.e. the rise and settling times should be small.

- **Disturbance Rejection:** The controller should be designed so that disturbances have small effect on tracking, i.e. result in small errors. For example, the steady state error due to a constant disturbance should remain within some specified tolerance.

- **Actuator Effort:** The input should remain within allowable levels. For the DC motor, the input voltage should satisfy $u \in [0, u_{max}] = [0V,\, 3V]$ to avoid damaging the motor.

- **Noise Rejection:** In most cases the controller relies on a measurement. It is typically required that any measurement inaccuracies, e.g. noise, have small effect on tracking.

- **Robustness to Model Uncertainty:** As noted in the previous section, the model used for control design is typically simplified. The controller must be robust, i.e. insensitive, to model errors introduced by this simplified model.

The DC motor control design will explore some of these performance trade-offs. In general the precise design requirements must be carefully considered. The dominant issues would depend on the particular problem. For example, model uncertainty may be the dominant issue in one problem but be less significant in another problem. A specific design example with concrete performance requirements is given in Section 4.6.

4.2 Open-Loop Control

Summary: This section describes an open-loop strategy to control a DC motor. The goal is to select the input voltage so that the motor maintains a desired speed. The open-loop controller selects the motor voltage using only the desired motor speed and a model of the motor. This open-loop strategy requires no additional sensors but has poor disturbance rejection and is sensitive to model variations.

4.2.1 Open-Loop Design

We'll consider a simple *open-loop control* strategy: i) the user specifies the desired motor speed $r(t)$, and ii) the controller sets the input voltage proportional to the desired speed, $u(t) = K_{ol}\, r(t)$ where K_{ol} is a gain to be selected. A key point is that the open-loop controller is based only on the desired speed and the gain is selected using a model of the motor. A block diagram for this open-loop strategy is shown in Figure 4.3. The motor itself is modeled by a first-order system (Equation 4.1) with transfer function $G(s) = \frac{b_0}{s+a_0}$. To model the combined controller/motor system, substitute $u(t) = K_{ol}\, r(t)$ into the motor model:

$$\dot{y}(t) + a_0\, y(t) = (b_0 K_{ol})\, r(t) + b_0\, d(t) \tag{4.3}$$

This ODE describes the dynamics of the combined open-loop controller/motor from inputs r and d to output y. The simple open-loop controller does not change the speed of response. In particular, for any K_{ol} there is a single pole $s_1 = -a_0$ with associated time constant $\tau_1 = \frac{1}{|a_0|}$. This is the same as for the motor itself (with no control). Moreover, the open-loop controller does nothing to reject disturbances, i.e. the steady state gain from d to y is unchanged by K_{ol}. The beneficial effect of K_{ol} is that, by proper selection, we can ensure $y(t)$ eventually follows $r(t)$. In particular, the choice $K_{ol} = \frac{a_0}{b_0}$ in Equation 4.3 yields a steady state gain equal to one from r to y. Hence $K_{ol} = \frac{a_0}{b_0}$ ensures that $y(t) \to \bar{r}$ for step reference commands $r(t) = \bar{r}$. This open-loop gain simply inverts the motor DC gain: $K_{ol} = \frac{a_0}{b_0} = \frac{1}{G(0)}$. Thus perfect knowledge of the motor DC gain is required to ensure $y(t) \to \bar{r}$, i.e. the value of $G(0)$ used in the open-loop controller must exactly match the DC gain of the real motor.

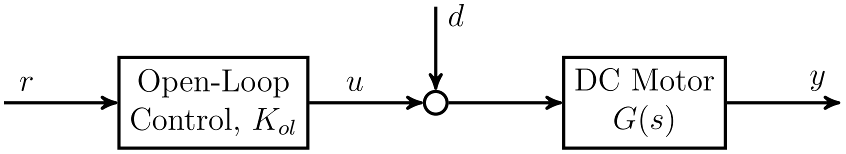


Figure 4.3: Block diagram for open-loop control of DC motor.

These results can be confirmed via simulation. The specific values $a_0 = 0.94$ and $b_0 = 766.8$ for the motor model yield a gain $K_{ol} = \frac{a_0}{b_0} = 0.0012 \frac{V\,sec}{rad}$. The system pole is $s_1 = -0.94$ with associated settling time $3\tau_1 = 3.18 sec$. Figure 4.4 shows the response (open-loop controller and motor) with IC $y(0) = 0$ and for two different step reference commands: $r(t) = \bar{r}$ with $\bar{r} = 1000$ and $2000 \frac{rad}{sec}$. A step disturbance is also applied: $d(t) = 0V$ for $t \in [0, 5]sec$ and $d(t) = -0.2V$ for $t \geq 5sec$. The left plot shows the input voltage. The right plot shows

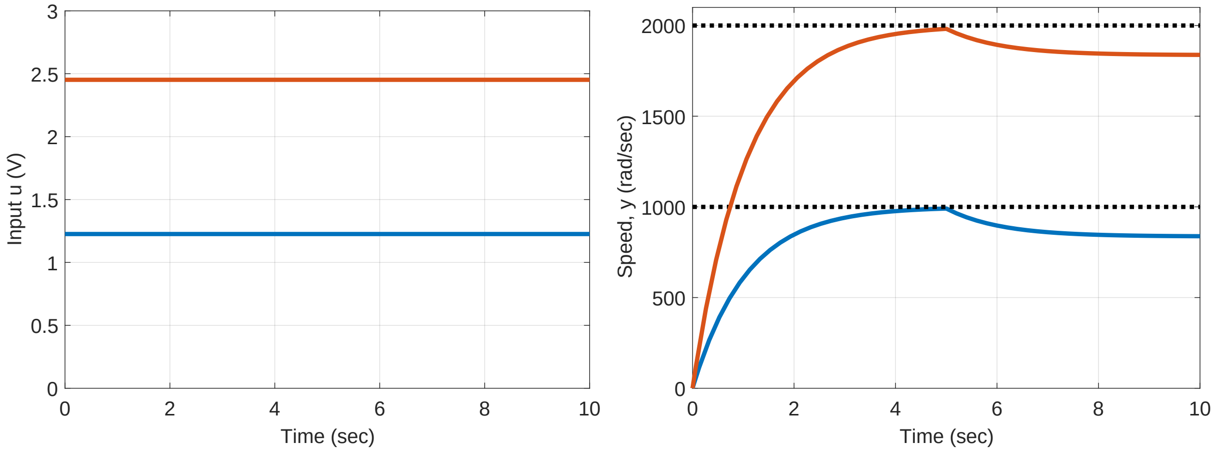


Figure 4.4: Step responses for combined open-loop control / motor system with $\bar{r} = 1000$ and $2000\,\frac{rad}{sec}$ and $d(t) = -0.2V$ for $t \geq 5$. Left: Input voltage. Right: Motor speed (solid) and desired speed (dashed).

the resulting motor speed (solid line) and desired speed (dashed line) for each case. In both cases, $y(t) \rightarrow \bar{r}$ for $t \leq 5$. The settling time is $3.18sec$ which is the same as for the motor itself. The input voltage remains unchanged when the step disturbance occurs at $t = 5sec$ because the open-loop controller was designed with no knowledge of the disturbance. As a result, the disturbance causes $y(t)$ to diverge from $r(t)$. The steady state gain from d to y is $\frac{b_0}{a_0} = 815.7\frac{rad}{secV}$. Hence the $-0.2V$ disturbance causes the motor speed to deviate by $-163\frac{rad}{sec}$.

These simulation results also demonstrate the linearity of the model. Both output responses are simply the the response due only to the command $r(t)$ (with $d(t) = 0$) plus the response due only to the disturbance $d(t)$ (with $r(t) = 0$). Moreover, the output response for $t \leq 5$ with $\bar{r} = 2000\,\frac{rad}{sec}$ is simply twice the response with $\bar{r} = 1000\,\frac{rad}{sec}$. Thus multiple simulations at different values of the reference command and/or disturbance are not required to understand the controller performance. In most cases, simulations will only be performed at a single reference command and/or disturbance.

4.2.2 Impact of Model Uncertainty

Section 4.1.2 emphasized that the models used for control design are simplified and contain inaccuracies. The effects of unmodeled dynamics and uncertain model parameters are briefly discussed for the DC motor open-loop controller. The DC motor model neglects fast electrical dynamics. These unmodeled dynamics have negligible impact on the performance of the open-loop controller and hence no results are shown.

The model parameter variations pose a more significant issue. The open-loop gain $K_{ol} = 0.0012$ was computed using the nominal ODE coefficients $a_0 = 0.94$ and $b_0 = 766.8$. As noted in Section 4.1.2 these nominal values were based on the motor specification sheet. The actual ODE coefficients vary from motor to motor. Figure 4.5 shows the impact of the parameter variations on the performance of the open-loop control. It shows a step response with $r(t) = 1000\frac{rad}{sec}$ and $K_{ol} = 0.0012$. The right plot shows the response with nominal DC

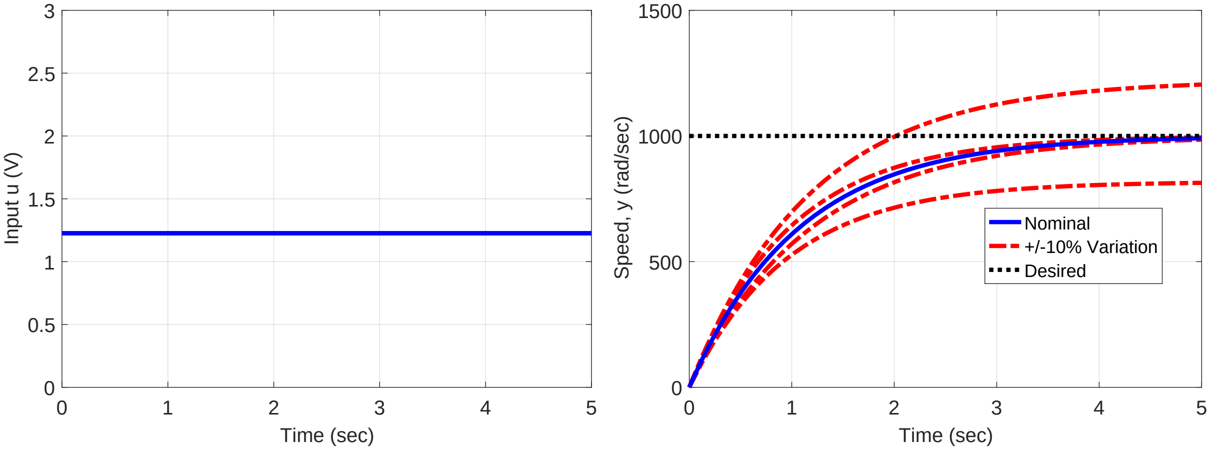


Figure 4.5: Step responses of open-loop controller/motor for nominal ODE coefficients and ±10% variation. Left: Input voltage. Right: Motor speed (solid) and desired speed (dashed).

motor coefficients (solid blue) and with ±10% variation in the ODE coefficients (dashed red): $(a_0, b_0) = \{(0.85, 690.1), (0.85, 843.5), (1.03, 690.1), (1.03, 843.5)\}$. Note that $K_{ol} = 0.0012$ is the same in all responses and only the ODE coefficients in the DC motor model are changed. The left plot shows the motor input voltage. This is the same for all responses since the open-loop controller only depends on r and the (fixed) value of K_{ol}. It is important to emphasize that the open-loop controller does not change the input voltage based on modeling errors. As a consequence, the responses with ±10% variation fail to converge to the desired value (some with larger errors than others).

One solution for parameter variations is to calibrate the open-loop controller for each specific motor. This would require each individual motor to be experimentally tested to determine the appropriate values of (a_0, b_0). The open-loop gain K_{ol} would then be computed for each specific motor rather than using nominal (typical) values of (a_0, b_0). Such calibration is used in some control problems. However, this is generally a more expensive and time-consuming process than the alternative (feedback designs) to be presented in the next few sections. This is especially true if the control system needs to work for thousands or millions of motors. In many cases precise calibration is not possible. In these cases alternative designs based on feedback measurements (discussed next) are required.

4.2.3 Summary of Open-Loop Performance

In this section we briefly summarize the performance of the open-loop controller with respect to the design requirements given in Section 4.1.3:

- **Stability:** The controller/motor combination will be stable for any open-loop gain K_{ol}. More generally, an open-loop controller cannot change the stability properties of the plant. Thus an open-loop controller cannot be used to stabilize an unstable plant.

- **Reference Tracking:** The open-loop strategy $u(t) = K_{ol} r(t)$ ensures tracking of the desired motor speed in steady state. This requires an accurate estimate for the steady state gain of the motor. The open-loop controller does not change the dynamic characteristics of the motor. It is possible to design more complicated (dynamic) open-loop strategies that modify the dynamic characteristics. However, this requires even more accurate models for the dynamics of the system.

- **Disturbance Rejection:** The open-loop controller has no affect on disturbances. It is typically true that open-loop controllers have poor disturbance rejection characteristics.
- **Actuator Effort:** The open-loop controller remains within the allowable actuator limits. This can generally be achieved by placing simple limits on the reference command.
- **Noise Rejection:** The open-loop controller does not require any measurements. Hence sensor noise is not an issue. Moreover, it means the system can be designed more cheaply as the cost of the sensor is not required.
- **Robustness to Model Uncertainty:** The open-loop controller requires an accurate estimate of the model. Hence it is sensitive to variations in the model as discussed in Section 4.2.2.

In summary, open-loop control can be effective if the plant is stable, the disturbances are small, and the model is accurate. If any of these conditions fails, then open-loop control will either fail to achieve stability (if the plant is unstable) or will not provide accurate tracking. Feedback control strategies are described next that can address unstable plants as well as problems with large disturbances and/or significant model uncertainty.

4.3 Proportional Control

Summary: This section describes a proportional controller for a DC motor. The controller computes the error between the desired motor speed and the measured motor speed. The motor voltage is then set proportional to this error. A larger proportional gain will reduce steady state errors and give a faster response. However, large gains also increase the motor inputs (possibly beyond the allowable voltage limits). Moreover, unmodeled dynamics will cause oscillations and overshoot if the gain is selected too large.

4.3.1 Proportional Control Design

We'll continue with the DC motor control problem introduced in the previous section. This section will consider a simple *closed-loop control* strategy known as *proportional (P) control*: i) the user specifies the desired motor speed $r(t)$, ii) the tracking error between the desired and actual speeds $e(t) = r(t) - y(t)$ is computed, and iii) the controller sets the input voltage proportional to the tracking error, $u(t) = K_p\,(r(t) - y(t))$ where K_p is a gain to be selected. **A key point is that the proportional controller requires a sensor to measure the motor speed.** A block diagram for this strategy, shown in Figure 4.6, involves a loop of cause and effect. In particular, the input voltage u affects the motor speed y and then y is used by the controller to compute the input voltage u. This is known as closed-loop control because feeding the measurement back to the controller creates a loop around the plant. The combined proportional controller and DC motor is referred to as the *closed-loop system*.

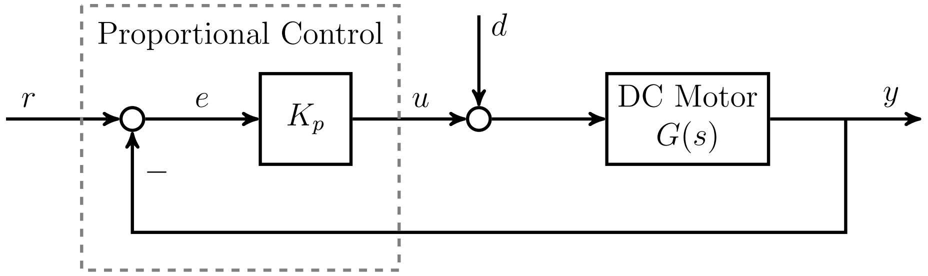


Figure 4.6: Block diagram for closed-loop proportional control of DC motor.

To model the closed-loop system, substitute $u(t) = K_p\,(r(t) - y(t))$ into the motor model (Equation 4.1) to obtain the following ODE:

$$\dot{y}(t) + a_0\,y(t) = b_0 K_p\,(r(t) - y(t)) + b_0\,d(t) \tag{4.4}$$

Bring all terms involving $y(t)$ to the left side. This yields a model for the closed-loop system:

$$\dot{y}(t) + (a_0 + b_0 K_p)\,y(t) = (b_0 K_p)\,r(t) + b_0\,d(t) \tag{4.5}$$

The closed-loop is modeled by a first-order ODE from inputs r and d to output y. The pole of the closed-loop system is $s_1 = -(a_0 + b_0 K_p)$. Thus the closed-loop is stable if the proportional gain satisfies $a_0 + b_0 K_p > 0$. For the remainder of the discussion it will be assumed that K_p satisfies this condition so that the closed-loop is stable. The closed-loop time constant is $\tau_1 = \frac{1}{|a_0+b_0K_p|}$. Thus the time constant decreases (faster response) as we increase the gain K_p. The gain also impacts the tracking and disturbance rejection. Consider the case of a step reference command $r(t) = \bar{r}$ and a step disturbance $d(t) = \bar{d}$. Equation 4.5 is a linear ODE and hence the response due to both $r(t)$ and $d(t)$ is simply the sum of the responses due to each individual input. Thus the motor speed converges in steady state to the following:

$$y_{ss} = \frac{b_0K_p}{a_0+b_0K_p}\bar{r} + \frac{b_0}{a_0+b_0K_p}\bar{d} \tag{4.6}$$

This yields the following formula for the steady state error:

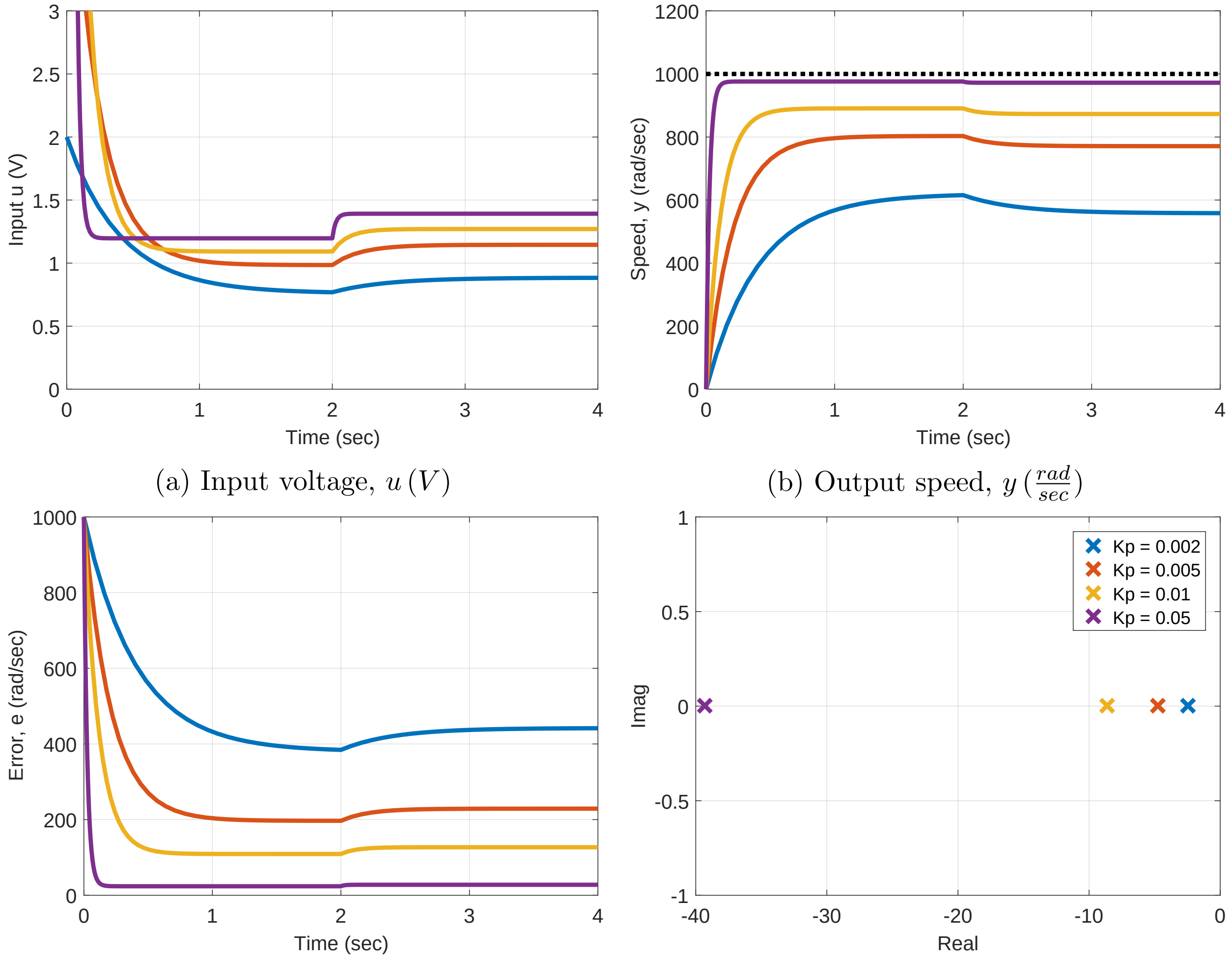
$$e_{ss} = \bar{r} - y_{ss} = \frac{a_0}{a_0+b_0K_p}\bar{r} - \frac{b_0}{a_0+b_0K_p}\bar{d} \tag{4.7}$$

In general there will be a non-zero steady state error. In other words, the proportional controller does not exactly track the reference command nor does it reject disturbance. However, the steady state error can be made arbitrarily small by choosing K_p sufficiently large. **To summarize, increasing the gain K_p will decrease the time constant and reduce the steady state error due to $\bar{r}$ and/or $\bar{d}$.** This seems to imply that the gain K_p should be chosen to be as large as possible. However, there are additional factors that limit the magnitude of K_p as discussed below.[*]

These results can be confirmed via simulation. For the DC motor, closed-loop stability requires $K_p > -\frac{a_0}{b_0} = -0.0012$. This is not a significant constraint as $K_p > 0$ in any practical design. Figure 4.7 shows the closed-loop response with IC $y(0) = 0$, step reference command $r(t) = 1000\,\frac{rad}{sec}$, and a step disturbance: $d(t) = 0V$ for $t \in [0, 2]sec$ and $d(t) = -0.2V$ for $t \geq 2sec$. These simulations are performed with four different proportional gains. The figure shows: (a) input voltage u, (b) output speed y, (c) tracking error e, and (d) location of closed-loop poles. Figures 4.7b and 4.7c confirm that larger proportional gains lead to a faster response and reduce the errors due to both the reference commands and disturbances. Figures 4.7d shows that the closed-loop pole moves to the left (faster response) as the gain is increased. This type of plot (location of poles with a varying gain) is called a *root locus*.

The plot of input voltage (a) shows one drawback of increasing gain. **Specifically, increasing the gain also increases the required voltage input especially during the initial transient response.** In fact, the choice $K_p = 0.002$ is the only response that remains within the maximum motor voltage limit of $u_{max} = 3V$. All other responses exceed this limit. The input voltage at $t = 0$ can be easily computed. The initial error is $e(0) = \bar{r} - y(0) = \bar{r}$ and hence the initial voltage is $u(0) = K_pe(0) = K_p\bar{r}$. For example, if $K_p = 0.002$ and $\bar{r} = 1000\frac{rad}{sec}$ then $u(0) = 2V$. The relation $u(0) = K_p\bar{r}$ shows that larger values of K_p increase the initial voltage input. This will typically limit the magnitude of K_p.

[*] This discussion is in the context of the DC motor but, for the most part, holds for any first order system. Some slight changes to the discussion are required if $a_0 = 0$ or $b_0 < 0$. For example, the DC motor has $b_0 = 766.8 > 0$ and K_p would also be positive in any real design. The time constant decreases as we "increase" K_p by making it more positive. For some systems the coefficient b_0 is negative. If $b_0 < 0$ then the gain K_p must also be negative so that the product b_0K_p remains positive. In this case, the time constant decreases as we "increase" the magnitude of K_p thus making it more negative. Also note that if $a_0 = 0$ then the system will have zero steady state error whenever $\bar{d} = 0$.



(a) Input voltage, $u\,(V)$

(b) Output speed, $y\,(\frac{rad}{sec})$

(c) Tracking error, $e = r - y\,(\frac{rad}{sec})$

(d) Locus of closed-loop poles as gain K_P varies.

Figure 4.7: Step responses for closed-loop with $r(t) = 1000\,\frac{rad}{sec}$ and $d(t) = -0.2V$ for $t \geq 2$.

4.3.2 Impact of Model Uncertainty

The effects of unmodeled dynamics and uncertain model parameters, introduced in Section 4.1.2, are briefly discussed for the DC motor proportional controller. First, consider model parameter variations. These variations have a much smaller effect on the performance of the closed-loop controller as compared to the open-loop controller. Figure 4.8 shows the response with proportional gain $K_p = 0.01$ and step command $r(t) = 1000\frac{rad}{sec}$. The right plot shows the response with nominal DC motor coefficients $(a_0, b_0) = (0.94, 766.8)$ (solid blue) and with ±10% variation in the ODE coefficients (dashed red). The left plot shows the motor input voltage. The effect of parameter variations shows up in the measurement of motor speed. As a result, the proportional controller is able to respond and counteract these parameter variations. All responses are quite similar and the performance is not as sensitive to the parameter variations. Larger proportional gains result in less sensitivity to the variations in the ODE coefficients. However, larger gains also cause large input voltages that exceed the allowable value $u_{max} = 3V$

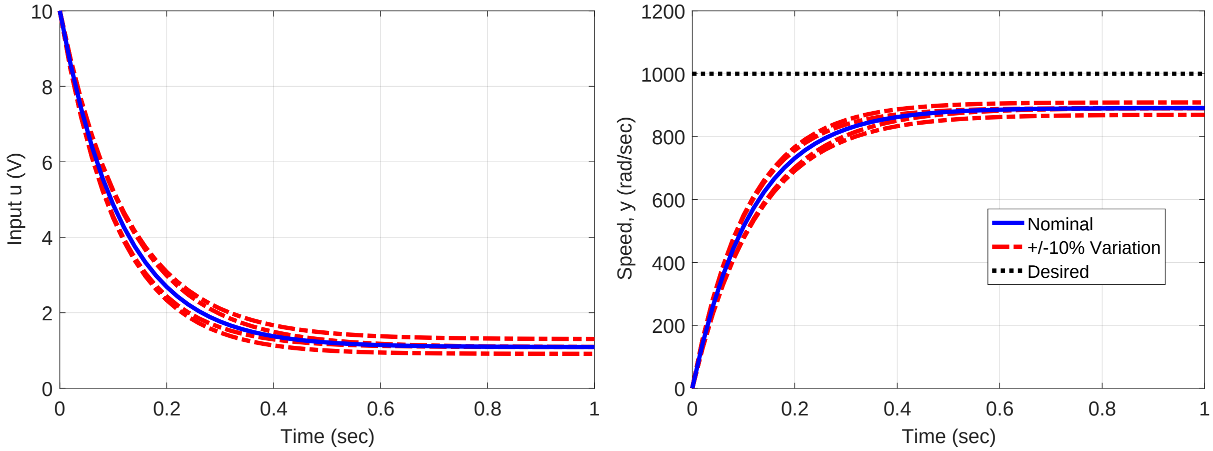


Figure 4.8: Step responses of closed-loop (proportional controller/motor) for nominal ODE coefficients and ±10% variation. Proportional gain is $K_p = 0.01$. Left: Input voltage. Right: Motor speed (solid) and desired speed (dashed).

as discussed above.

Next, unmodeled dynamics can have a significant impact on the closed-loop performance for large values of K_p. In particular, the first-order DC motor model neglects fast electrical dynamics. Equation 4.2 gives a second-order motor model that includes these electrical dynamics. Figure 4.9 shows the motor speed output with a proportional controller and the second-order motor model in Equation 4.2. The response is simulated with $r(t) = 1000\frac{rad}{sec}$ and several proportional gains. The response has the expected first-order characteristic for $K_p = 1$, 1.5, and 2. However, increasing the gain further ($K_p = 8$ and 16) leads to overshoot and oscillations but no reduction in the settling time. To understand this behavior, express the second-order motor model (Equation 4.2) as follows (without the controller and the disturbance):

$$a_2\ddot{y}(t) + a_1\dot{y}(t) + a_0\, y(t) = b_0 u(t) \tag{4.8}$$

where $b_0 = 0.001227$, $a_2 = 3 \times 10^{-10}$, $a_1 = 1.6 \times 10^{-6}$, and $a_0 = 1.505 \times 10^{-6}$. Substitute $u(t) = K_p\,(r(t) - y(t))$ and group all terms involving $y(t)$ to the left side. This yields a second-order model for the closed-loop that includes the fast electrical dynamics:

$$a_2\ddot{y}(t) + a_1\dot{y}(t) + (a_0 + b_0K_p)\, y(t) = (b_0K_p)\, r(t) \tag{4.9}$$

Divide both sides of this ODE by a_2 to put this model in standard second-order form with damping ratio ζ and natural frequency ω_n:

$$\ddot{y}(t) + \underbrace{\frac{a_1}{a_2}}_{:=2\zeta\omega_n}\dot{y}(t) + \underbrace{\frac{(a_0 + b_0K_p)}{a_2}}_{:=\omega_n^2} y(t) = \frac{(b_0K_p)}{a_2}\, r(t) \tag{4.10}$$

Table 4.1 shows the closed-loop poles, natural frequency, and damping ratio for several values of K_p. The row for $K_p = 0$ corresponds to no control and this row gives the poles of the second-order motor model itself. The remaining rows show that increasing K_p will both increase the

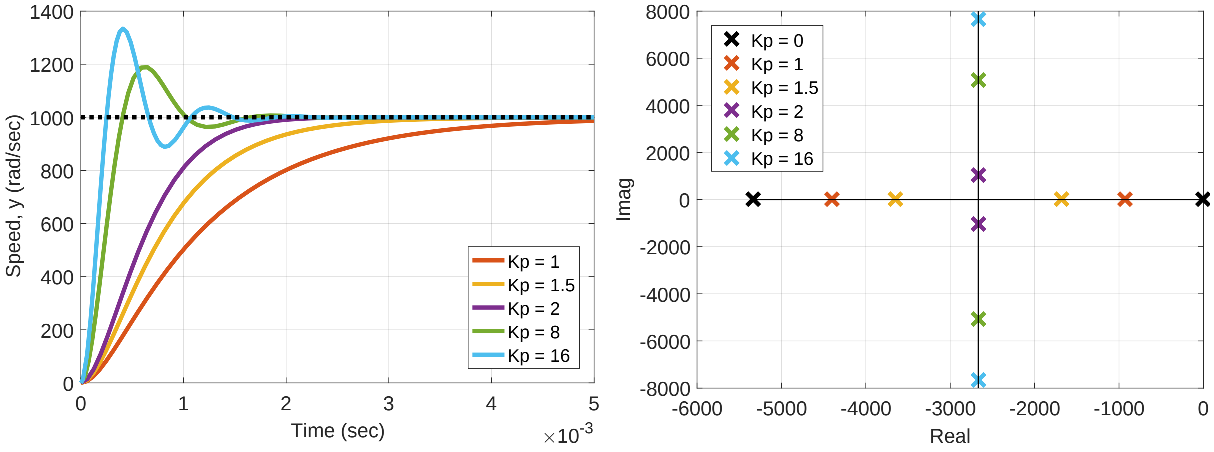


Figure 4.9: Step responses of closed-loop (proportional controller/motor) with second order model. Left: Motor speed (solid) and desired speed (dashed). Right: Root Locus.

K_p	$s_{1,2}$	ζ	ω_n
0	-0.94, -5332	37.6	71
1	-930, -4404	1.32	2023
1.5	-1681, -3653	1.08	2478
2	-2667 ± 1036 j	0.93	2861
8	-2667 ± 5061 j	0.47	5720
16	-2667 ± 7637 j	0.33	8089

Table 4.1: Closed-loop poles for several gains K_p using second order motor model.

natural frequency ω_n and decrease the damping ratio ζ. The right plot of Figure 4.9 provides a graphical version of this data in the form of a root locus plot of the closed-loop poles as the gain K_p is changed. The poles start at $(-0.94, -5332)$ for $K_p = 0$. Increasing K_p will initially cause the slow pole to move leftward while the fast pole moves rightward in the complex plane. This corresponds to speeding up the system response because the slow pole dominates the response. Eventually the poles join and at -2667 for $K_p = 1.7$. Further increasing the gain will cause the two poles to split and follow a vertical line with real part -2667. As a result, the response becomes oscillatory with more overshoot and no improvement in the settling time if the gain is increased above $K_p = 1.7$. To summarize, fast dynamics can typically be neglected in the design for reasonable values of the gain. **However, large values of the proportional gain will eventually degrade the performance (oscillations and overshoot) with no improvements in settling time.**

4.3.3 Summary of Proportional Control Performance

The proportional controller has the following effects with respect to the design requirements given in Section 4.1.3:

- **Stability:** For the DC motor closed-loop stability requires $K_p > -\frac{a_0}{b_0} = -0.0012$. Note that the motor itself (with no control) is stable but that the closed-loop system is unstable if $K_p < -0.0012$. Thus closed-loop control can, if designed poorly, destabilize a stable plant. Conversely, if the plant (without control) is unstable then a properly designed feedback controller can be used to obtain a stable closed-loop system. Thus it is important to distinguish between the stability of the plant (without control) and the stability of the closed-loop (with feedback control).

- **Reference Tracking:** The proportional controller does not track the desired motor speed even if there is no disturbance. However, increasing the gain K_p will decrease the steady state error due to reference commands. It also decreases the closed-loop time constant (faster response).

- **Disturbance Rejection:** Increasing the gain K_p will also decrease the steady state error due to disturbances.

- **Actuator Effort:** Large gains K_p will cause the input voltage to exceed the allowable bounds. The largest inputs tend to occur during the initial transients and this may limit the choice of the gain K_p.

- **Noise Rejection:** The proportional controller requires a measurement of motor speed and sensor noise may be an issue. Generally larger gains tend to amplify noise leading to degraded performance. This is explored further in Section XXX.

- **Robustness to Model Uncertainty:** The proportional controller makes the closed-loop response less sensitive to model parameter variations. However, very large gains will eventually cause a degradation in performance (overshoot and oscillations) due to any unmodeled dynamics. For more general systems (higher than first-order), increasing gains can even cause the closed-loop to become unstable.

In summary, closed-loop proportional control requires trade-offs between reference tracking, disturbance rejection, speed of response, and sensitivity to parameter variations (all of which tend improve with increasing gains) versus actuator effort, robustness to unmodeled dynamics, and sensor noise (all of which tend to degrade with increasing gains. The proportional controller only has a single gain K_p that can be selected and this limits the ability to make these various trade-offs. More advanced feedback control strategies are described next that include additional dynamics and gains to improve this trade-off.

4.4 Proportional-Integral Control

Summary: This section describes a proportional-integral controller for a DC motor. The controller computes the motor input voltage using a term proportional to the error and another term proportional to the integral of the error. The main property of integral control is that it achieves zero error in steady state. In addition, the initial transient is dominated by the proportional term while the steady state is dominated by the integral term. Thus the two terms can be used to provide better trade-offs in the control design.

4.4.1 Proportional-Integral Control Design

The previous section discussed proportional control for the DC motor. Increasing the gain K_p has several effects including reduced steady state error, faster settling time, and larger control inputs. However, the proportional control strategy is limiting because a single gain affects both the transient behavior (e.g. control effort for small t) as well as the steady state behavior (e.g. error as $t \to \infty$). Conceptually it would be useful to have an alternative strategy with gains that can be tuned to independently modify the transient and steady state behaviors. This section introduces such a strategy known as *proportional-integral (PI) control*: i) the user specifies the desired motor speed $r(t)$, ii) the tracking error between the desired and actual speeds $e(t) = r(t) - y(t)$ is computed, and iii) the controller sets the input voltage as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau \tag{4.11}$$

where K_p and K_i are proportional and integral gains to be selected. A block diagram for the closed-loop is shown in Figure 4.10.

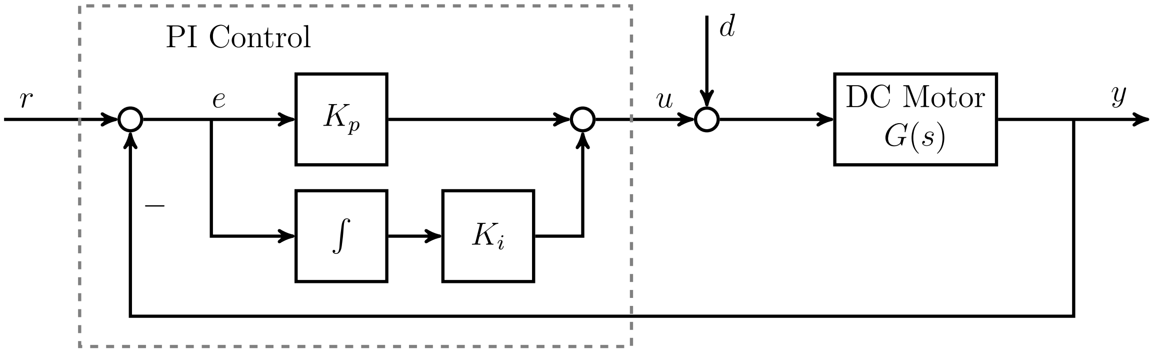
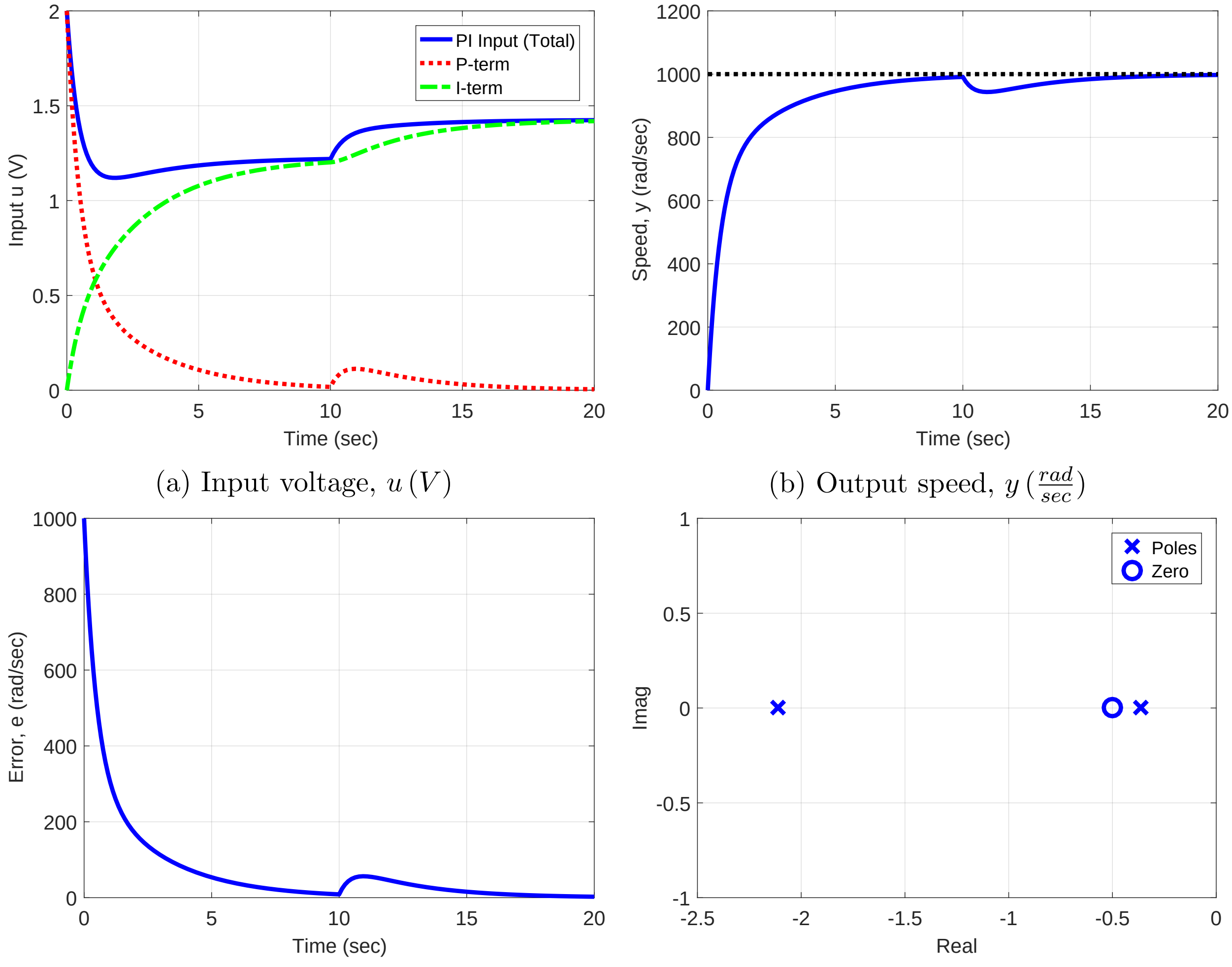


Figure 4.10: Block diagram for proportional-integral control of DC motor.

The first (proportional) term in Equation 4.11 reacts to the “present” as measured by the current error. This term dominates during the initial transient. The second (integral) term reacts to the “past” as measured by the accumulated integral of the error. This term dominates in the steady state as $t \to \infty$. The closed-loop simulation shown in Figure 4.11 displays this behavior. The simulation is performed with the first-order DC motor model (Equation 4.1) and a PI controller with gains $K_p = 0.002$ and $K_i = 0.001$. The simulation uses the IC $y(0) = 0$,



(a) Input voltage, $u\,(V)$

(b) Output speed, $y\,(\frac{rad}{sec})$

(c) Tracking error, $e = r - y\,(\frac{rad}{sec})$

(d) Closed-loop poles and zero from r to y.

Figure 4.11: Step responses for closed-loop with $r(t) = 1000\,\frac{rad}{sec}$ and $d(t) = -0.2V$ for $t \geq 10$. PI gains are $K_p = 0.002$ and $K_i = 0.001$.

step reference command $r(t) = 1000\,\frac{rad}{sec}$, and a step disturbance: $d(t) = 0V$ for $t \in [0, 10]sec$ and $d(t) = -0.2V$ for $t \geq 10sec$. Figure 4.7a shows the control input u (solid blue) along with the contributions of the proportional (red dotted) and integral (green dash-dotted) terms. At $t = 0$, the proportional term is $K_p e(0) = K_p \bar{r} = 2V$ and the integral term is zero, i.e. the initial control input is entirely due to the proportional term. As time goes on, the area under the curve of $e(t)$ vs. t (Figure 4.7c) increases and hence the integral term increases. Eventually $e(t) \to 0$ so the control input in steady state is due entirely to the integral term.

To further understand the simulations results shown in Figure 4.11, first note that the PI controller itself is a dynamical system with inputs (r, y) and output u. It can be expressed as an ODE by differentiating both sides of Equation 4.11 with respect to time:

$$\dot{u}(t) = K_p \dot{e}(t) + K_i e(t) \tag{4.12}$$

To model the closed-loop, differentiate the the motor model ODE (Equation 4.1) and substitute for $\dot{u}$ using Equation 4.12:

$$\ddot{y}(t) + a_0\,\dot{y}(t) = b_0 K_p \dot{e}(t) + b_0 K_i e(t) + b_0 \dot{d}(t) \tag{4.13}$$

Finally, bring all terms involving $y(t)$ to the left side of the ODE:

$$\ddot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=2\zeta\omega_n}\,\dot{y}(t) + \underbrace{(b_0 K_i)}_{:=\omega_n^2}\,y(t) = (b_0 K_p)\,\dot{r}(t) + (b_0 K_i)\,r(t) + b_0 \dot{d}(t) \tag{4.14}$$

This is a second-order model for the closed-loop from inputs r and d to output y. Both poles are in the left half plane if and only if $a_0 + b_0 K_p > 0$ and $b_0 K_i > 0$. For the remainder of the discussion it is assumed that these conditions are satisfied so that the closed-loop is stable.

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

$$T_{r\to y} = \frac{(b_0 K_p)s + (b_0 K_i)}{s^2 + (a_0 + b_0 K_p)s + (b_0 K_i)} \tag{4.15}$$

The closed-loop poles have natural frequency $\omega_n = \sqrt{b_0 K_i}$ and damping ratio $\zeta = \frac{a_0 + b_0 K_p}{2\sqrt{b_0 K_i}}$. The second-order response characteristics (rise time, overshoot, settling time) depend on (ω_n, ζ) as discussed in Section 3.6. Increasing K_i will increase ω_n, decrease ζ and leave $\zeta\omega_n$ unchanged. Thus increasing K_i should reduce damping (causing overshoot / oscillations) but have no effect on settling time. Moreover, increasing K_p leaves ω_n unchanged but increases ζ and $\zeta\omega_n$. Thus increasing K_p should increase damping and reduce settling time. The actual conclusions are more complicated because the transfer function from r to y has a zero at $z = -\frac{K_i}{K_p}$. As discussed in Section 3.7.2, this zero impacts the initial transient; it tends to cause overshoot and decrease the rise time. It is difficult to precisely state the effect of the gains K_p and K_i due to this zero. Hence some trial and error is typically required in selecting these gains. Figure 4.11d shows the closed-loop pole and zero for the DC motor and PI controller.[†]

The main benefit of PI control is the impact on tracking and disturbance rejection. Consider the case of a step reference command $r(t) = \bar{r}$ and a step disturbance $d(t) = \bar{d}$. Based on Equation 4.14, the motor speed converges to $y_{ss} = \bar{r}$, i.e. there is no steady state error for any values of $\bar{r}$ and $\bar{d}$. This behavior is observed in Figures 4.11b and 4.11c. **This is a key property of integral control: If the system converges to a steady state then there is no error ($e_{ss} = 0$).** This property can be shown with the following argument. Assume the system has reached a steady state at time t_0 such that $u(t) \approx u_{ss}$ and $e(t) \approx e_{ss}$ for $t \geq t_0$. Based on this approximation the PI control input (Equation 4.11) at time t_0 is:

$$u(t_0) \approx K_p e_{ss} + K_i \int_0^{t_0} e(\tau)\, d\tau \tag{4.16}$$

Similarly, split the integral term to approximate the PI control input for any $t \geq t_0$ as follows:

$$\begin{aligned} u(t) &= K_p e(t) + K_i \int_0^{t_0} e(\tau)\, d\tau + K_i \int_{t_0}^{t} e(\tau)\, d\tau \\ &\approx \left(K_p e_{ss} + K_i \int_0^{t_0} e(\tau)\, d\tau \right) + K_i e_{ss} \cdot (t - t_0) \end{aligned} \tag{4.17}$$

[†] This discussion, for the most part, holds for PI control on any first order system. As in the previous section, some slight changes to the discussion are required for systems with $b_0 < 0$.

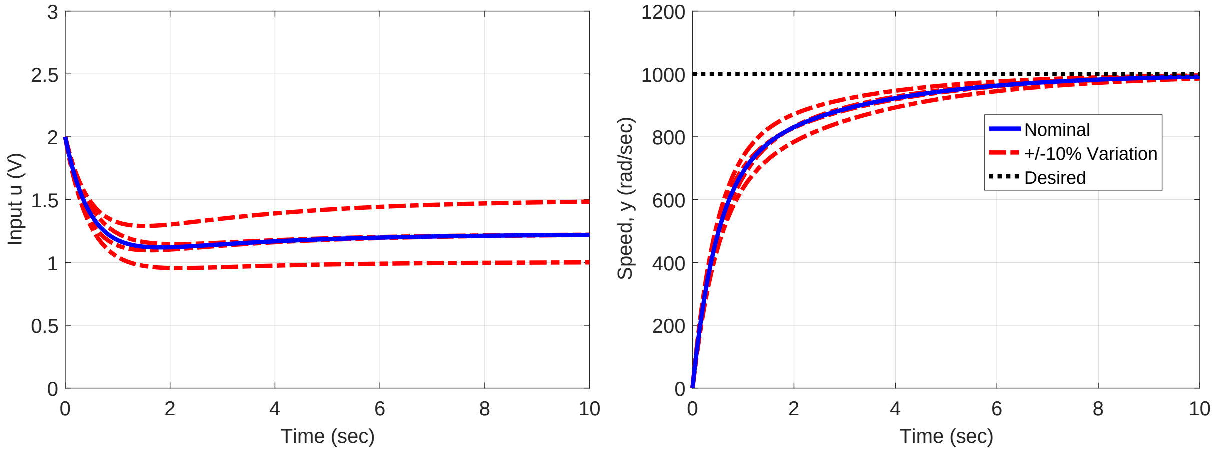


Figure 4.12: Step responses of closed-loop (PI controller/motor) for nominal ODE coefficients and ±10% variation. PI gains are $K_p = 0.002$ and $K_i = 0.001$. Left: Input voltage. Right: Motor speed (solid) and desired speed (dashed).

The term in parentheses is approximately $u(t_0)$ by Equation 4.16. Thus Equation 4.17 can be rewritten as: $K_i e_{ss} \cdot (t - t_0) \approx u(t) - u(t_0)$. Finally this implies $K_i e_{ss} \cdot (t - t_0) \approx 0$ because $u(t) \approx u(t_0)$. Hence there is no steady state error, $e_{ss} = 0$. This argument is only based on the existence of a steady state and the use of an integral term in the controller. Thus, the remarkable property of zero steady state error is general, e.g. it holds for PI control on higher-order plants and in the presence of constant disturbances.

4.4.2 Impact of Model Uncertainty

The effects of unmodeled dynamics and uncertain model parameters, introduced in Section 4.1.2, are briefly discussed for the DC motor PI controller. First, consider model parameter variations. Figure 4.12 shows the response with gains $(K_p, K_i) = (0.002, 0.001)$ and step command $r(t) = 1000\frac{rad}{sec}$. The right plot shows the response with nominal DC motor coefficients $(a_0, b_0) = (0.94, 766.8)$ (solid blue) and with ±10% variation in the ODE coefficients (dashed red). The left plot shows the motor input voltage. There is a small spread in the motor speed responses. This demonstrates that the controller reduces sensitivity to model parameter variations. There are two significant differences between these results and those obtained with the proportional control (Figure 4.8). First, all motor speed responses in Figure 4.12 converge in steady-state to the reference command. This is due to the important zero steady state error property of integral control. In other words, perfect tracking is achieved in steady state as long as the parameter variations don't cause instability. Second, the input voltage remains within the allowable limits $u_{max} = 3V$. This is achievable because the PI controller has two terms that can be used to independently tune the transient and steady-state response. In particular, the PI controller has a proportional gain $K_p = 0.002$ chosen to ensure that the input voltage remains within the allowable limits during the initial transient. The integral term causes the input voltage to converge to the "correct" value required for zero steady-state error.

Next, unmodeled dynamics can have a significant impact on the closed-loop performance for large gains (K_p, K_i). In particular, the first-order DC motor model neglects fast electrical dynamics. Equation 4.2 gives a second-order motor model that includes these electrical dynamics. These fast dynamics can typically be neglected in the design for reasonable values of the gains. However, large gains will eventually degrade the performance leading to oscillations, overshoot and even instability. This behavior is similar to that obtained for the proportional controller in Section 4.3.2 and hence no results are shown.

4.4.3 Summary of Proportional-Integral Control Performance

The PI controller has the following effects on the design requirements given in Section 4.1.3:

- **Stability:** Closed-loop stability requires $a_0 + b_0 K_p > 0$ and $b_0 K_i > 0$. Again, the motor itself (with no control) is stable but that the closed-loop system is unstable if the gains fail to satisfy these conditions. Thus closed-loop control can, if designed poorly, destabilize a stable plant. Conversely, if the plant (without control) is unstable then a properly designed feedback controller can be used to obtain a stable closed-loop system.

- **Reference Tracking:** If the closed-loop is stable then the PI controller perfectly tracks the desired motor speed in steady state. The closed-loop has two poles that can be placed arbitrarily in the left half by proper selection of K_p and K_i. Thus the transient response characteristics (overshoot, rise time, settling time) can be tuned by placing the poles using the relations derived for second order systems (Section 3.6). The main complication is that the transfer function from r to y has a zero at $-\frac{K_i}{K_p}$. This zero modifies the transient response characteristics as discussed in Section 3.7.2.

- **Disturbance Rejection:** If the closed-loop is stable then the PI controller perfectly rejects any constant disturbance, i.e. constant disturbances cause no steady-state error.

- **Actuator Effort:** Large gains (K_p, K_i) will cause the input voltage to exceed the allowable bounds. The control effort in the initial transient is due entirely to the proportional term while the control effort in steady-state is entirely due to the integral term. The two gains allow the controller to be more easily tuned to remain within actuator effort limits.

- **Noise Rejection:** The PI controller requires a measurement of motor speed and sensor noise may be an issue. Generally larger gains tend to amplify noise leading to degraded performance. This is explored further in Section XXX.

- **Robustness to Model Uncertainty:** The proportional controller makes the closed-loop response less sensitive to model parameter variations. Moreover, there will be zero steady-state error even if the parameters vary from their nominal values as long as the closed-loop is stable. Unmodeled (fast) dynamics will cause performance degradation (overshoot, oscillations, and even instability) if the gains are selected too large.

The main property of PI control is that it achieves zero error in steady state. The proportional and integral gains can be tuned to provide better trade-offs than can be achieved by proportional control.

4.5 Proportional-Derivative Control

Summary: This section introduces a second-order model for the directional heading of a rocket. The dynamics are unstable and cannot be stabilized by either P or PI control. This motivates the introduction of proportional derivative (PD) control. This controller computes the input using a term proportional to the error and another term proportional to the derivative of the error. This strategy is able to stabilize the rocket attitude dynamics. However, a basic implementation of PD control causes large control inputs when there are step changes in the reference command. In addition, PD control tends to amplify the effects of sensor noise. It is typical to implement a "smoothed" derivative term to alleviate these undesired effects.

4.5.1 Rocket Heading Dynamics

Proportional or Proportional-Integral controllers are generally sufficient for the control of first-order plants. However, additional issues arise for control second (and higher order) plants. We'll explore some of these issues using a second order model for the heading, i.e. attitude, dynamics of a rocket. Rockets require precise control of their heading direction to ensure that they reach their desired final destination. Figure 4.13 (left) shows a simple diagram showing the key variables involved in the attitude dynamics. The output is the rocket heading angle y (rad). Modern rockets control their heading by thrust vectoring (also known as gimballed thrust) [15]. Specifically, the rocket nozzle can be rotated to change the direction of the thrust T. The input u (rad) is the angle between the rocket thrust and the centerline of the rocket. The goal is to control the rocket heading angle y using the input thrust angle u.

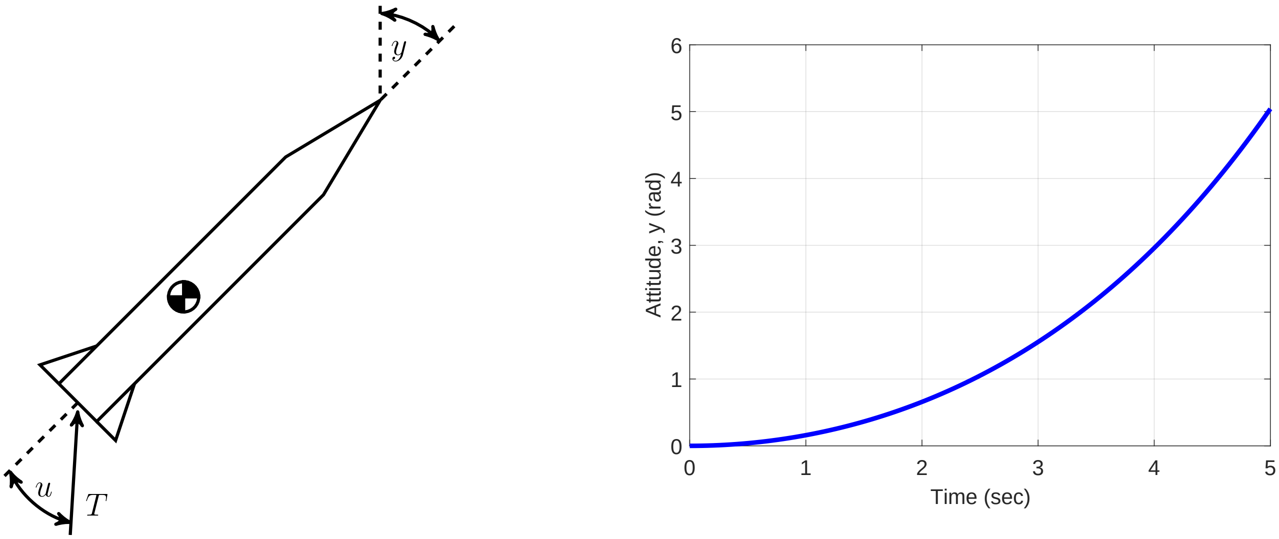


Figure 4.13: Left: Diagram for rocket attitude. Right: Step response for "nominal" rocket attitude model with $u(t) = 0.05rad$.

A simple nonlinear model for the rocket attitude dynamics is developed in Appendix 4.9.2 based on the work in [11]. Linearizing around an equilibrium condition with $(\bar{y}, \dot{\bar{y}}, \bar{u}) = (0, 0, 0)$

yields the following second-order linear ODE model:

$$\begin{aligned}&\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0\, u(t) + b_0\, d(t)\\ &\text{where: } a_1 = 0\frac{1}{sec},\ a_0 = -0.12\frac{1}{sec^2},\ \text{and } b_0 = 6.32\frac{1}{sec^2}\end{aligned} \tag{4.18}$$

The ODE includes another input $d(t)$ (rad) to model various disturbances acting on the rocket. This "nominal" model fails to capture uncertainties in the ODE coefficients and also neglects certain dynamics (e.g. the dynamics of the sloshing in the fuel tank). It is also important to note that the rocket nozzle can only physically rotate by small angles. It is assumed that the nozzle angle input saturates at $|u| \leq u_{max} = 0.2rad$ $(\approx 11.5^o)$.

Figure 4.13 (right) shows a step response for the rocket attitude model with $u(t) = 0.05rad$ and ICs $y(0) = \dot{y}(0) = 0$. The transfer function for the second-order model is $G(s) = \frac{6.32}{s^2-0.12}$. This has poles at $s_1 = -0.346$ and $s_2 = 0.346$. Hence, the plant dynamics are unstable due to the positive root s_2. The step response in Figure 4.13 is dominated by a growing exponential with time constant $\tau_2 = \frac{1}{|s_2|} = 2.89sec$ corresponding to the positive root.

Consider the use of proportional control for the rocket heading system. Substitute $u(t) = K_p(r(t) - y(t))$ into Equation 4.18 and collect all terms involving $y(t)$ on the left side:

$$\ddot{y}(t) + \underbrace{a_1}_{2\zeta\omega_n}\dot{y}(t) + \underbrace{(a_0 + b_0K_p)}_{\omega_n^2}\, y(t) = (b_0K_p)\, r(t) + b_0\, d(t) \tag{4.19}$$

The proportional gain affects the natural frequency and does not affect the damping term a_1. For the rocket attitude system $a_1 = 0$ and hence the characteristic equation for the closed-loop is $s^2 + (a_0 + b_0K_P) = 0$. If $a_0 + b_0K_p < 0$ then the closed-loop is unstable with one pole in the left half of the complex plane (LHP) and one pole in the RHP. If $a_0 + b_0K_P \geq 0$ then the closed-loop is unstable with two poles on the imaginary axis. **Thus it is not possible to stabilize the rocket attitude dynamics using proportional control. It can also be shown that the rocket attitude dynamics cannot be stabilized by a PI control law.**[‡] In general, P and PI control cannot stabilize any second-order plant with $a_1 \leq 0$. Moreover, P and PI will be ineffective if the plant is stable but with low damping, e.g. a plant with $(a_1, a_0) = (2, 100)$ corresponding to $(\zeta, \omega_n) = (0.1, 10)$. For such systems, P and PI are not able to increase the damping and hence the closed-loop response will be oscillatory with large overshoot. This motivates the need for the additional control strategy introduced next.

4.5.2 Proportional-Derivative Control Design

This section introduces a strategy known as *proportional-derivative (PD) control*: i) the user specifies the desired motor speed $r(t)$, ii) the tracking error between the desired and actual speeds $e(t) = r(t) - y(t)$ is computed, and iii) the controller sets the input voltage as follows:

$$u(t) = K_pe(t) + K_d\dot{e}(t) \tag{4.20}$$

[‡] To model the closed-loop with PI control, first differentiate Equation 4.18 and then substitute for $\dot{u}$ using the ODE form of a PI controller in Equation 4.12. The closed-loop characteristic equation is $s^3 + (a_0 + b_0K_p)s + (b_0K_i) = 0$. The characteristic equation can be factored as $(s - p_1)(s - p_2)(s - p_3) = 0$ where $\{p_1, p_2, p_3\}$ are the poles. Comparing the s^2 terms in both forms of the characteristic equation yields $-(p_1 + p_2 + p_3) = 0$. This implies that $Re\{p_i\} \geq 0$ for at least one pole, i.e. the closed-loop is unstable for any choice of (K_i, K_p).

where K_p and K_d are proportional and derivative gains. A block diagram for the closed-loop is shown in Figure 4.14. The first (proportional) term in Equation 4.20 reacts to the "present" as measured by the current error. The second (derivative) term reacts to the "future" as measured by the rate of change of the error. For example, suppose $e(t) = 0$ and $\dot{e}(t) > 0$ at some time t. In this case the proportional term $K_p e(t)$ is zero. However, $\dot{e}(t) > 0$ implies the error is growing and will be positive shortly after t. The derivative term $K_d \dot{e}(t)$ anticipates this change and increases $u(t)$ accordingly.[§] Conversely, if $e(t) = 0$ and $\dot{e}(t) < 0$ then the error is decreasing and will be negative shortly after t. In this case the derivative term anticipates the change due to $\dot{e}(t) < 0$ and decreases $u(t)$.

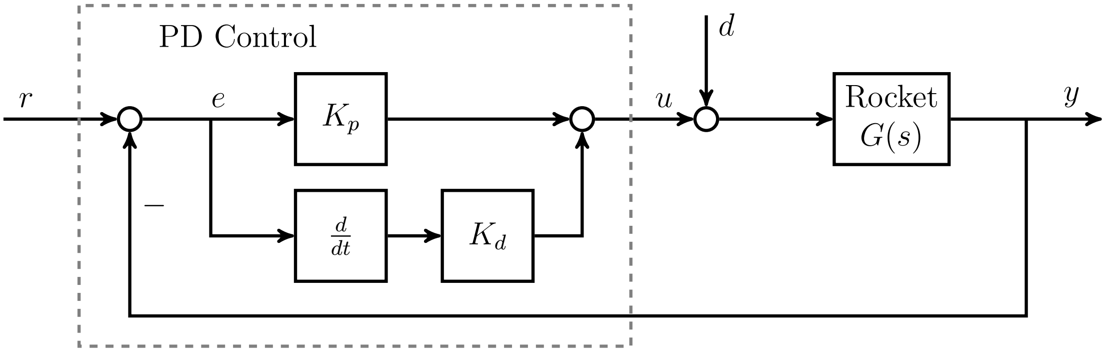


Figure 4.14: Block diagram for proportional-derivative control of rocket attitude.

A model of the closed-loop system can be used to build on this intuition. Substitute the PD controller (Equation 4.20) into the model for the rocket heading (Equation 4.18) and bring all terms involving y to the left side:

$$\ddot{y}(t) + \underbrace{(a_1 + b_0 K_d)}_{:=2\zeta\omega_n} \dot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=\omega_n^2} y(t) = (b_0 K_d)\, \dot{r}(t) + (b_0 K_p)\, r(t) + b_0 d(t) \tag{4.21}$$

This is a second-order model for the closed-loop from inputs r and d to output y. The gain K_p affects the natural frequency while K_d affects the damping term. Both poles are in the left half plane if and only if $a_1 + b_0 K_d > 0$ and $a_0 + b_0 K_p > 0$. For the remainder of the discussion it is assumed that these conditions are satisfied so that the closed-loop is stable.

The closed-loop has two poles that can be placed arbitrarily through proper selection of (K_p, K_d). In addition, the transfer function from r to y includes a zero at $-\frac{K_p}{K_d}$ which influences the response. For example, suppose the PD gains are selected as $(K_p, K_d) = (0.75, 0.47)$. As a result, the closed-loop poles have the following natural frequency and damping ratio:

$$\omega_n = \sqrt{a_0 + b_0 K_p} = 2.14sec \;\text{ and }\; \zeta = \frac{a_1 + b_0 K_d}{2\sqrt{a_0 + b_0 K_p}} = 0.7 \tag{4.22}$$

This corresponds to complex poles at $s_{1,2} = -1.5 \pm 1.53j$. Section 3.6 relates the second order step response characteristics to (ω_n, ζ). The predicted settling time, rise time, and peak

[§] This statement assumes $K_d > 0$ which is the typical case when $b_0 > 0$. If $b_0 < 0$ then typically $K_d < 0$. In this case if $\dot{e}(t) > 0$ then the derivative term decreases $u(t)$.

overshoot for the closed-loop response are:

$$t_s \approx \frac{3}{\zeta\omega_n} = 2sec, \;\; t_r = \frac{\frac{\pi}{2} + \sin^{-1}(\zeta)}{\omega_n\sqrt{1-\zeta^2}} = 1.53sec, \;\text{ and }\; M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}} = 0.046 \qquad (4.23)$$

For the selected gains, the closed-loop zero is located at $-\frac{K_p}{K_d} = -1.57$. This zero is almost the same as the real part of the poles $s_{1,2}$. Hence the zero is expected to reduce the rise time and increase the overshoot compared to the predictions in Equation 4.23.

Figure 4.15 shows results with this PD controller and the rocket attitude model (Equation 4.18). The simulation uses the ICs $\dot{y}(0) = y(0) = 0$, step reference command $r(t) = 0.1\,rad$, and a step disturbance: $d(t) = 0rad$ for $t \in [0, 5]sec$ and $d(t) = -0.01rad$ for $t \geq 5sec$. Figure 4.15b shows the heading angle response. The peak heading angle is $y_p = 0.123rad$ and the heading angle just before the disturbance is $y(5) = 0.103rad$. Hence the observed peak overshoot is $M_p = \frac{0.123-0.103}{0.103} = 0.19$. This is significantly larger than the predicted value $M_p = 0.046$. Moreover, the observed rise time is approximately $t_r = 0.55sec$. This is significantly shorter than the predicted value $t_r = 1.53sec$. These deviations from the predictions are due to the zero at $-\frac{K_p}{K_d} = -1.57$ as mentioned above. The observed settling time (before the disturbance occurs) is close to the predicted value of $t_s = 2sec$. Also note the steady state error both before and after the disturbance (Figure 4.15c) due to the lack of integral control.

Figure 4.15a shows the control input u (solid blue) along with the proportional (red dotted) and derivative (green dash-dotted) terms. In steady state $\dot{e} = 0$ and the derivative term has no impact. However, the derivative term has a significant impact during transients. The inset diagram shows the signals zoomed for $t \leq 0.1sec$. At $t = 0$, the proportional term is $K_p e(0) = K_p\bar{r} = 0.0746rad$. The derivative term is extremely large and far exceeds the allowable thrust angle $u_{max} = 0.2rad$. In theory, the derivative term at $t = 0$ is $K_d\dot{e}(0) = K_d\dot{r}(0) = \infty$ because the desired value steps discontinuously from $r(0^-) = 0rad$ to $r(0^+) = 0.1rad$. The simulation uses a more practical implementation to avoid this infinite transient. **Specifically, it is common to implement a "smoothed" derivative term as $K_d\dot{v}(t)$ where**

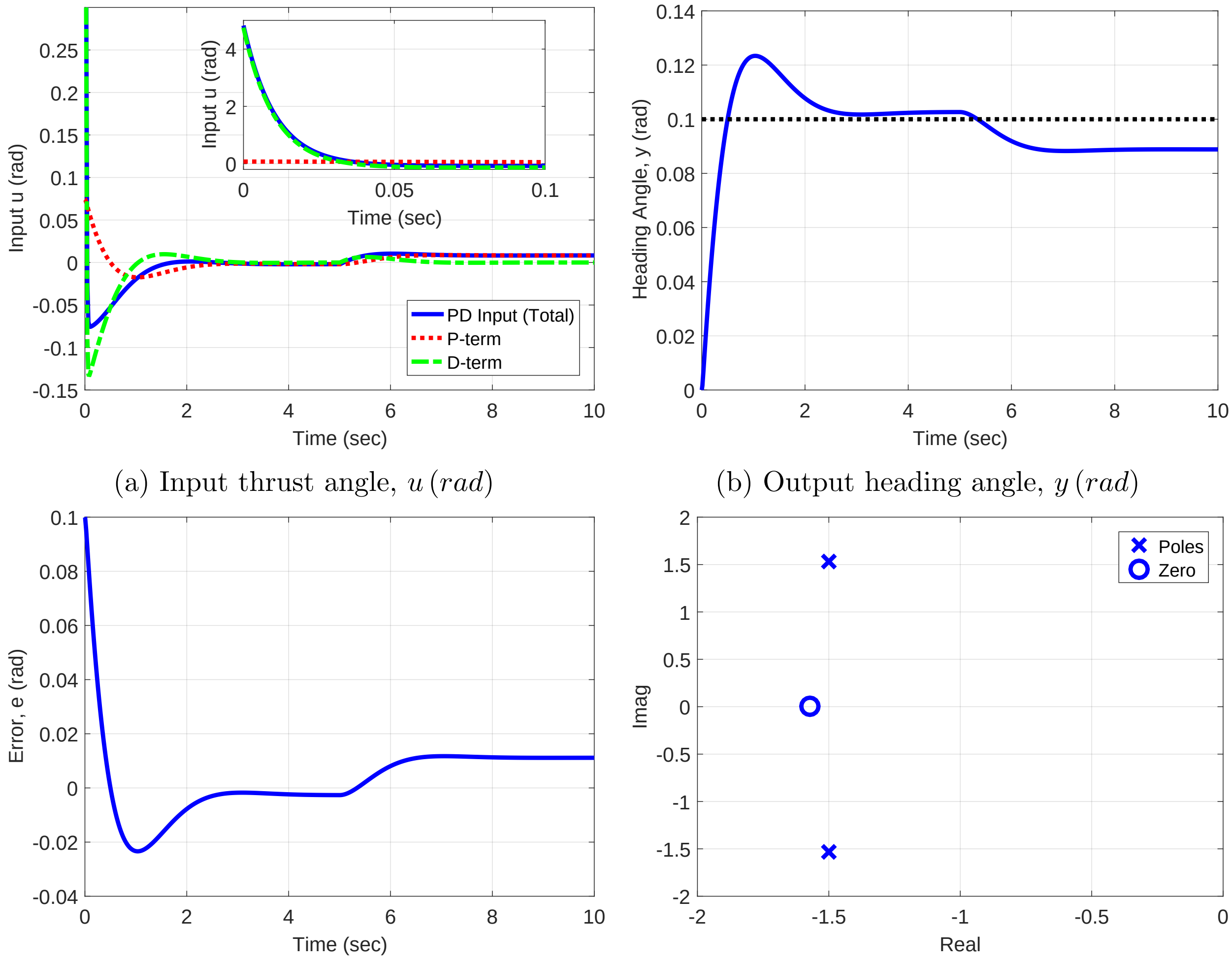
$$\alpha_1\dot{v}(t) + v(t) = e(t) \quad \text{IC: } v(0) = 0 \qquad (4.24)$$

If $\alpha_1 = 0$ then Equation 4.24 simply yields $v(t) = e(t)$ and the derivative term is $K_d\dot{e}(t)$ as before. If α_1 is "small" then this ODE first smooths the error before differentiating. Specifically, when $e(t)$ jumps discontinuously at $t = 0$ then $v(t)$ will respond smoothly with a characteristic first-order response. Hence $v(t)$ will be differentiable and the derivative term will not be infinite. The value $\alpha_1 = 0.01$ is used in the simulation. This yields $\dot{v}(0) = \frac{e(0)}{\alpha_1} = \frac{\bar{r}}{\alpha_1} = 10$. The resulting derivative term is then $K_d\dot{v}(0) = 4.7rad$ as observed in Figure 4.15a.

4.5.3 Summary of Proportional-Derivative Control Performance

The impact of model uncertainty (parameter variations and unmodeled dynamics) is similar to the discussions for P/PI control. Hence no additional results are shown for the PD controller. The PD controller has the following effects on the design requirements in Section 4.1.3:

- **Stability:** P and PI controllers cannot stabilize a second-order system if $a_1 \leq 0$. However, PD control can be used to stabilize such systems. In particular, closed-loop stability of a second-order system with PD control requires $a_1 + b_0K_d > 0$ and $a_0 + b_0K_p > 0$.



(a) Input thrust angle, $u\,(rad)$

(b) Output heading angle, $y\,(rad)$

(c) Tracking error, $e = r - y\,(rad)$

(d) Closed-loop poles and zero from r to y.

Figure 4.15: Step responses for closed-loop with $r(t) = 0.1\,rad$ and $d(t) = -0.01rad$ for $t \geq 5$. PD gains are $K_p = 0.74$ and $K_d = 0.47$.

- **Reference Tracking:** A PD controller does not perfectly track reference commands due to the lack of integral control. Increasing K_p will reduce the steady state error but K_d has no affect on the steady-state. The closed-loop has two poles that can be placed arbitrarily in the left half by proper selection of K_p and K_d. Thus the transient response characteristics (overshoot, rise time, settling time) can be tuned by placing the poles using the relations derived for second order systems (Section 3.6). The main complication is that the transfer function from r to y has a zero at $-\frac{K_p}{K_d}$. This zero modifies the transient response characteristics as discussed in Section 3.7.2.

- **Disturbance Rejection:** Again, increasing K_p will decrease the steady-state error due to constant disturbances but K_d will have no affect on the steady-state.

- **Actuator Effort:** The derivative term can lead to extremely large control inputs especially when the error term changes rapidly. It is typical to implement the derivative term using the smoothed derivative in Equation 4.24.

- **Noise Rejection: A significant impact of the derivative term is that it tends to amplify the effect of sensor noise thus degrading performance.** This is discussed further in Section XXX.

- **Robustness to Model Uncertainty:** The PD controller has similar features with respect to model uncertainty as a proportional control. The proportional term reduces sensitivity to parameter variations. However large gains can lead to performance degradations due to fast unmodeled dynamics.

The derivative term of a PD controller directly impacts the damping. However, the derivative term can yield large control inputs during transients and can also amplify sensor noise.

4.6 PID Tuning

Summary: This section focuses on tuning of PID controllers for first and second order systems. The design requirements are specified in terms of the step response characteristics of the closed-loop system. The controller gains are tuned by *pole placement* using two steps. First, the closed loop poles are selected to satisfy the design requirements using the step response characteristics. Second, the controller gains are chosen to place the poles at the desired locations.

4.6.1 Approach

Each feedback system has unique design issues. As a result, it is difficult to provide a single procedure for tuning of PID controllers that will be successful in all cases. The textbook [1] is a good reference covering many details on PID tuning. This section will focus on tuning for the special case with plants modeled by first and second order systems. The main design requirements are specified in terms of the step response characteristics of the closed-loop system. **The basic design approach described here is called *pole placement*. This means the controller gains can be tuned to place the closed-loop poles anywhere in the left half plane. The closed loop poles are placed to satisfy the design requirements using the step response characteristics described in Sections 3.5 and 3.6.** The remainder of the section provides examples of this design procedure for first and second order systems. Procedures to tune PID controllers for higher order systems are given in [1].

4.6.2 PI for First-Order Systems

Consider a plant modeled by the following first-order ODE:

$$\begin{aligned} &\dot{y}(t) + a_0\, y(t) = b_0\, u(t) + b_0\, d(t) \\ &\text{where: } a_0 = 2 \text{ and } b_0 = 3 \end{aligned} \tag{4.25}$$

The plant has a single pole at $s = -2\frac{rad}{sec}$. Thus the plant is stable with time constant $\tau = 0.5sec$ and settling time $3\tau = 1.5sec$. The design requirements for the controller are:

- **Stability:** The closed-loop system should be stable.
- **Reference Tracking:** The error $e(t) := r(t) - y(t)$ should satisfy the steady state bound $|e_{ss}| \leq 0.01$ for a unit reference $r(t) = 1\,\frac{rad}{sec}$ with no disturbance $d(t) = 0$. The step response should also have peak overshoot $M_p \leq 0.05$ and settling time $\leq 0.75sec$.
- **Disturbance Rejection:** The steady state error due to a constant unit disturbance $d(t) = 1$ with no reference $r(t) = 0$ should satisfy $|e_{ss}| \leq 0.02$.[¶]
- **Actuator Effort:** The input should remain within $|u(t)| \leq 10$.

[¶] Note that the requirements for reference tracking and disturbance rejection are written independently with unit norm inputs. The principle of superposition can be used to bound the error when there are both reference and disturbance inputs. In particular, if $r(t) = \bar{r}$ and $d(t) = \bar{d}$ then the requirements imply $|e_{ss}| \leq 0.01\bar{r} + 0.02\bar{d}$.

- **Noise Rejection:** No requirement involving sensor noise will be considered in this example. However, noise rejection typically restricts the speed of response. In other words, noise is typically amplified if the closed-loop settling time is too small. This will be discussed in more detail later in the course.
- **Robustness to Model Uncertainty:** As noted previously, the model used for control design is typically simplified. In this case, assume the first order model neglects “fast” dynamics. If the closed-loop is too fast then these unmodeled dynamics will degrade performance. Thus the closed-loop settling time is required to be $\geq 0.1 sec$.

Proportional control is simple and should be considered as an initial option if the plant is first-order. However, this approach only has one (proportional) gain that affects both the settling time and steady state error. The reference tracking requirement will generally require a large proportional gain. As discussed in Section 4.3.2, large proportional gains may actually degrade performance due to unmodeled dynamics. In such cases, PI control is a good option. The proportional and integral terms can be used to independently modify the transient response (settling time) and steady state tracking.

Based on these considerations a PI controller will be designed for this example system:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau \tag{4.26}$$

The closed-loop model is derived following similar steps to those in Section 4.4.1: i) Substituting the control (Equation 4.26) into the plant dynamics (Equation 4.25), ii) Differentiate both sides of the resulting ODE, and iii) collect all terms involving $y(t)$ on the left side. This yields the following closed-loop model:

$$\ddot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=2\zeta\omega_n} \dot{y}(t) + \underbrace{b_0 K_i}_{:=\omega_n^2} y(t) = b_0 K_p \, \dot{r}(t) + b_0 K_i \, r(t) + b_0 \dot{d}(t) \tag{4.27}$$

Here ζ and ω_n are the closed-loop damping ratio and natural frequency. The pole placement procedure involves selecting the location of the closed-loop poles in order to satisfy the design requirements. Closed-loop stability is achieved by placing the poles anywhere in the LHP. Moreover, the use of integral control (and closed-loop stability) implies that $e(t) \to 0$ for any constant reference and/or disturbance. Hence the steady-state reference tracking and disturbance rejection requirements are automatically satisfied. Thus the main requirements involve the peak overshoot $M_p \leq 0.05$ and settling time $\leq 0.75 sec$ due to a unit reference.

Section 3.6 discussed the step response features for a second order system (with no zero). Recall that if the closed-loop has damping ratio $\zeta \leq 1$ and natural frequency ω_n then the closed-loop has roots at $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$. The settling time is approximately $\frac{3}{\zeta\omega_n}$ and the peak overshoot is $M_p = \exp(-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi)$. **The pole placement method is to select the closed-loop (ω_n, ζ) based on the settling time and peak overshoot requirements. The controller gains can then be computed from Equation 4.27: $K_i = \frac{\omega_n^2}{b_0}$ and $K_p = \frac{2\zeta\omega_n - a_0}{b_0}$.** This discussion focused on the case $\zeta \leq 1$. Similar results can be obtained if it is desired to have an overdamped ($\zeta > 1$) closed-loop with two real poles.

Results for two different PI control designs are shown in Table 4.2. Both controllers are designed to just meet the settling time requirement $\tau_{settle} = 0.75 sec$. Reducing settling time further might cause issues with control effort, noise, and/or robustness. The first design $K_1(s)$ is critically damped and hence the estimated overshoot (without the zero) is zero. The second design $K_2(s)$ is underdamped and the estimated overshoot (without the zero) is $M_p = 0.046$. Overshoot is generally undesirable but a small amount of overshoot has the benefit of significantly reducing the rise time, e.g. refer back to the right plot of Figure 3.9. Note that the closed-loop (Equation 4.27) depends on $\dot{r}(t)$ and, as a result, has a zero at $s = -\frac{K_i}{K_p}$. This zero will reduce the rise time but also cause increase overshoot in both designs. This zero will likely cause the second design to exceed the overshoot requirement.

Design	ζ	ω_n, $\frac{rad}{sec}$	Poles, $s_{1,2}$	M_p	τ_{settle}, sec	K_p	K_i
$K_1(s)$	1	4	-4, -4	0	0.75	2	5.33
$K_2(s)$	0.7	5.71	$-4 \pm 4.08j$	0.046	0.75	2	10.88

Table 4.2: Data for two PI controllers. M_p and τ_{settle} are estimated neglecting the zero at $-\frac{K_i}{K_p}$.

Figure 4.16 shows the closed-loop responses for the two designs with zero ICs, step reference command $r(t) = 4$, and a step disturbance: $d(t) = 0$ for $t \in [0, 2] sec$ and $d(t) = -2$ for $t \geq 2 sec$. The right plot shows the output. Note that the use of integral control allows both designs to converge back to the reference command even after the disturbance occurs. The peak value for the second design is $y_p \approx 4.5$. This corresponds to a peak overshoot $M_p = 0.125$ ($= \frac{4.5-4}{4}$). As expected, this exceeds the overshoot requirement due to the zero at $s = -\frac{K_i}{K_p}$. The left plot of Figure 4.16 shows the control signal. Both designs remain with the allowable bounds. If the control exceeded the allowable bounds then the speed of response would likely need to be reduced. This might need an increase in the required settling time. **In general, a reasonable initial PI design for a first-order system can be obtained by placing critically damped ($\zeta = 1$) poles. This will result in repeated poles $s_{1,2} = -\omega_n, -\omega_n$ and the natural frequency can be tuned to set the desired settling time. Some iteration will be required due to the effect of the closed-loop zero at $s = -\frac{K_i}{K_p}$.**

4.6.3 PID for Second-Order Systems

Consider a plant modeled by the following second-order ODE:

$$\begin{aligned} &\ddot{y}(t) + a_1 \dot{y}(t) + a_0\, y(t) = b_0\, u(t) + b_0\, d(t) \\ &\text{where: } a_1 = -2, \;\; a_0 = 17, \;\; \text{and } b_0 = 17 \end{aligned} \tag{4.28}$$

The plant has a poles in the RHP at $s = 1 \pm 4j \frac{rad}{sec}$ and is thus unstable. The poles are lightly damped with natural frequency of $4.1 \frac{rad}{sec}$. The design requirements for the controller are:

- **Stability:** The closed-loop system should be stable.

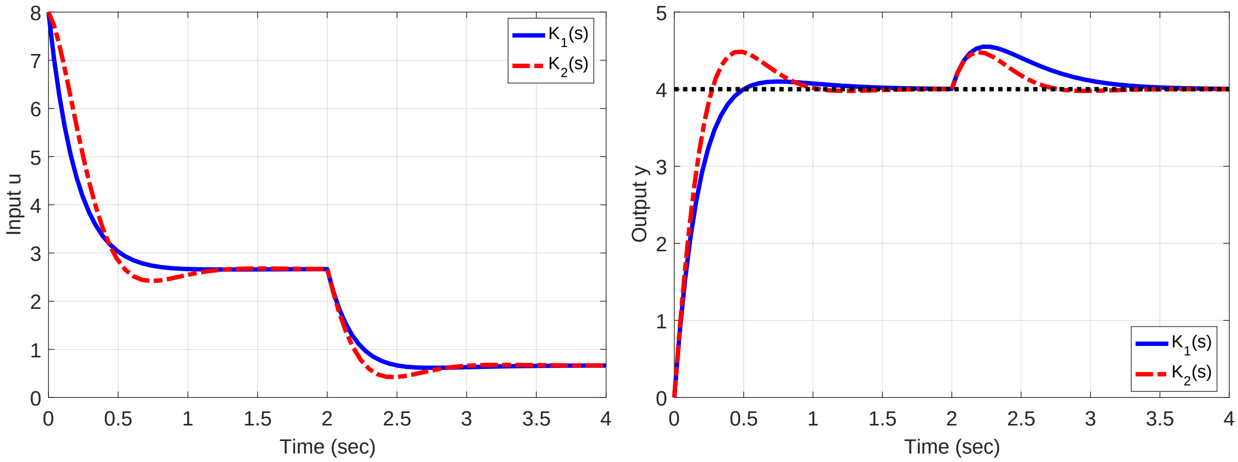


Figure 4.16: Step responses for closed-loop with $r(t) = 4$ and $d(t) = 2$ for $t \geq 2$. Results shown for two PI controllers

- **Reference Tracking:** The error $e(t) := r(t) - y(t)$ should satisfy the steady state bound $|e_{ss}| \leq 0.01$ for a unit reference $r(t) = 1\,\frac{rad}{sec}$ with no disturbance $d(t) = 0$. The step response should also have peak overshoot $M_p \leq 0.1$ and settling time $\leq 0.6sec$.

- **Disturbance Rejection:** The steady state error due to a constant unit disturbance $d(t) = 1$ with no reference $r(t) = 0$ should satisfy $|e_{ss}| \leq 0.02$.

- **Actuator Effort:** The input should remain within $|u(t)| \leq 20$.

- **Noise Rejection:** No requirement involving sensor noise will be considered in this example. However, noise rejection typically restricts the speed of response.

- **Robustness to Model Uncertainty:** Assume the second order model neglects “fast” dynamics. If the closed-loop is too fast then these unmodeled dynamics will degrade performance. Thus the closed-loop settling time is required to be $\geq 0.1sec$.

Control is required to stabilize the system, i.e. move the poles from the RHP to the LHP, and to add damping to the poles. A derivative term is required to meet these objectives for the example system. Hence a PID controller of the following form will be designed:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau - K_d \dot{y}(t) \tag{4.29}$$

Note that the derivative term is written as $-K_d\dot{y}(t)$. This avoids differentiating the reference signal which, for step reference commands, can cause a large spike in the control $u(t)$. If the plant dynamics are overdamped (or heavily damped) then the derivative term may not be needed and should be removed to simplify the control design.

The closed-loop model is derived by: i) Substituting the control (Equation 4.29) into the plant dynamics (Equation 4.28), ii) Differentiating both sides of the resulting ODE, and iii)

Collecting all terms involving $y(t)$ on the left side. This yields the following closed-loop model:

$$y^{[3]}(t) + (a_1 + b_0 K_d)\ddot{y}(t) + (a_0 + b_0 K_p)\dot{y}(t) + (b_0 K_i)y(t) = b_0 K_p\, \dot{r}(t) + b_0 K_i\, r(t) + b_0 \dot{d}(t) \tag{4.30}$$

Pole placement involves selecting the location of the closed-loop poles in order to satisfy the design requirements. Closed-loop stability is achieved by placing the poles anywhere in the LHP. Moreover, the use of integral control (and closed-loop stability) implies that $e(t) \to 0$ for any constant reference and/or disturbance. Hence the steady-state reference tracking and disturbance rejection requirements are automatically satisfied. Thus the main requirements involve the peak overshoot $M_p \leq 0.1$ and settling time $\leq 0.6sec$ due to a unit reference.

Assume the closed loop poles are placed at $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$ and $s_3 = -p < 0$ for some (ω_n, ζ, p). Then the closed-loop characteristic equation has the form:

$$0 = (s^2 + 2\zeta\omega_n s + \omega_n^2)(s + p) = s^3 + (p + 2\zeta\omega_n)s^2 + (2\zeta\omega_n p + \omega_n^2)s + \omega_n^2 p \tag{4.31}$$

Match the coefficients of this desired characteristic equation to the corresponding one for the closed-loop model in Equation 4.30. This yields the following relations between the PID gains and the desired closed-loop pole locations:

$$\begin{aligned} a_1 + b_0 K_d &= p + 2\zeta\omega_n \\ a_0 + b_0 K_p &= 2\zeta\omega_n p + \omega_n^2 \\ b_0 K_i &= \omega_n^2 p \end{aligned} \tag{4.32}$$

The closed-loop is third-order but we can still use the results on second order step responses in Section 3.6 to select initial pole locations. Specifically, select the closed-loop (ω_n, ζ) based on the settling time and peak overshoot requirements. The third pole placed at $s = -p$ will affect the transient characteristics. As a starting point, it can be selected as $p \geq \omega_n$. The controller gains can then be computed from the chosen (ω_n, ζ, p) using Equation 4.32.

Results for two different PID control designs are shown in Table 4.3. Both controllers are designed to just meet the settling time requirement $\tau_{settle} = 0.6sec$. Reducing settling time further might cause issues with control effort, noise, and/or robustness. The first design $K_1(s)$ is critically damped and hence the estimated overshoot (neglecting the zero and extra pole) is zero. The second design $K_2(s)$ is underdamped and the estimated overshoot is $M_p = 0.046$. Note that the closed-loop (Equation 4.30) has a zero at $s = -\frac{K_i}{K_p}$. This zero will reduce the rise time but also cause increase overshoot in both designs. However, both designs also have an extra pole at $s = -p$. This will tend to slow down the response and partially offset the effect of the zero. Some iteration may be required in the design due to these additional effects.

Figure 4.17 shows the closed-loop responses for the two designs with zero ICs, step reference command $r(t) = 4$, and a step disturbance: $d(t) = 0$ for $t \in [0, 2]sec$ and $d(t) = -2$ for $t \geq 2sec$. The right plot shows the output. Note that the use of integral control allows both designs to converge back to the reference command even after the disturbance occurs. The peak value for the second design is $y_p \approx 4.73$. This corresponds to a peak overshoot $M_p = 0.185$ $(= \frac{4.73-4}{4})$. This exceeds the overshoot requirement due to the zero at $s = -\frac{K_i}{K_p}$. The left plot of Figure 4.17 shows the control signal. Both designs remain with the allowable bounds. The critically damped design has less overshoot but slower rise time (after the reference command) and slower response to the disturbance. The final gains can be tuned with further iteration.

Design	ζ	ω_n, $\frac{rad}{sec}$	p	Poles, $s_{1,2}$ and s_3	M_p	τ_{settle}, sec	K_p	K_i	K_d
$K_1(s)$	1	5	5	-5, -5, -5	0	0.6	3.41	7.35	1.0
$K_2(s)$	0.7	7.14	5	$-5 \pm 5.10j$, -5	0.046	0.6	4.94	15.0	1.0

Table 4.3: Data for two PID controllers. M_p and τ_{settle} are estimated neglecting the zero at $-\frac{K_i}{K_p}$ and extra pole at $s = -p$.

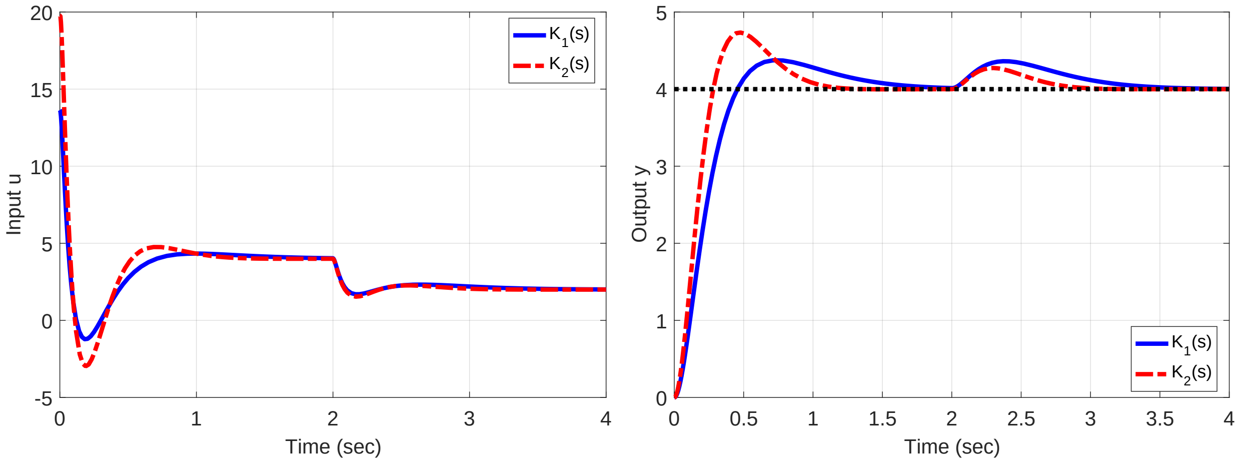


Figure 4.17: Step responses for closed-loop with $r(t) = 4$ and $d(t) = 2$ for $t \geq 2$. Results shown for two PID controllers

4.7 Modifications to Basic PID

Summary: This section describes a few variations to the basic PID algorithm that are useful in practice. First, actuator saturation can degrade systems that use integral control through an effect known as windup. Integrator anti-windup strategies are useful to avoid such performance degradation. Second, a PID controller (or its variations) can result in zeros in the complementary sensitivity (reference to output) transfer function. These zeros can be modified or completely removed by weighting the reference terms in a PID controller. A particular instance of this reference weighting is the use of rate feedback for the derivative term.

See notes.

4.8 Control Law Implementation

Summary: The controllers designed in this book can be represented by ODEs. It is common to implement these controllers on a microprocessor. In this case, the control u is recomputed at specific (discrete) time intervals. The (discrete-time) update equation is chosen to approximate the properties of the designed (ODE) controller. The approximate update equation can be implemented on a microprocessor with a few lines of code.

4.8.1 Implementation Architecture

Several variations of PID control have been introduced thus far including P, PI, and PD. These controllers can be represented by an ODE that relates tracking error $e(t)$ to control $u(t)$. For example, a PI control is represented by the following ODE:

$$\dot{u}(t) = K_p\dot{e}(t) + K_i e(t) \tag{4.33}$$

The corresponding transfer function is $K(s) = \frac{K_p s + K_i}{s}$. The next few chapters will develop additional control design methods beyond the PID architecture. These new design methods also yield controllers that can be represented by an ODE. It is possible to physically implement such controllers using standard circuit components (resistors, capacitors, inductors, op-amps). However, it is common to instead implement the control law on a microprocessor.

A simple model for the controller implementation on a microprocessor is shown in Figure 4.18. This figure also shows several signals involved in the control loop. The control u is re-computed at specific (discrete) time intervals, e.g. every $\Delta t = 0.1sec$. This consists of three key steps: 1) Sampling, 2) Difference Equation Update, and 3) Zero-Order Hold (ZOH). Each of these steps is described in more detail below. The overall approach taken in this book is referred to as "continuous-time design": a controller $K(s)$ is designed assuming continuous time and then an approximation is obtained for implementation. An alternative approach is to directly design a controller in discrete-time [17]. This "discrete-time design" requires the plant dynamics $G(s)$ to be approximated by a discrete-time system.

1. Sampling: The output of the plant $y(t)$ is a *continuous-time* signal, i.e. it is a function of time t where t is a continuous variable. The microprocessor samples $y(t)$ once every Δt seconds to obtain a sequence of measurements: $y_1 := y(\Delta t)$, $y_2 := y(2\Delta t)$, $y_3 := y(3\Delta t)$, etc. The sampled measurement y_k is a *discrete-time* signal, i.e. it is a function of a time index k where $k = 1, 2, 3, \ldots$ is a positive integer. The value $y_k := y(k\Delta t)$ is the measurement at the k^{th} time that the microprocessor runs its code. The microprocessor also requires the reference as a discrete-time signal r_k. In many systems the reference is also obtained by sampling, e.g. the pilot inputs on an airplane are sampled to generate the reference. However, the reference could simply be stored in memory on the microprocessor. The sampling time Δt depends on the particular application but it is assumed to be "fast". Roughly this means the sampling rate should be 10× faster than the relevant dynamics. For example, if $G(s)$ has important dynamics with time constant near $0.5sec$ then the sampling time should satisfy $\Delta t \leq \frac{0.5}{10}sec = 50msec$. If the sampling is "fast" then the implemented controller will closely match the behavior of the continuous-time design $K(s)$. Figure 4.18 shows the output in continuous and discrete time.

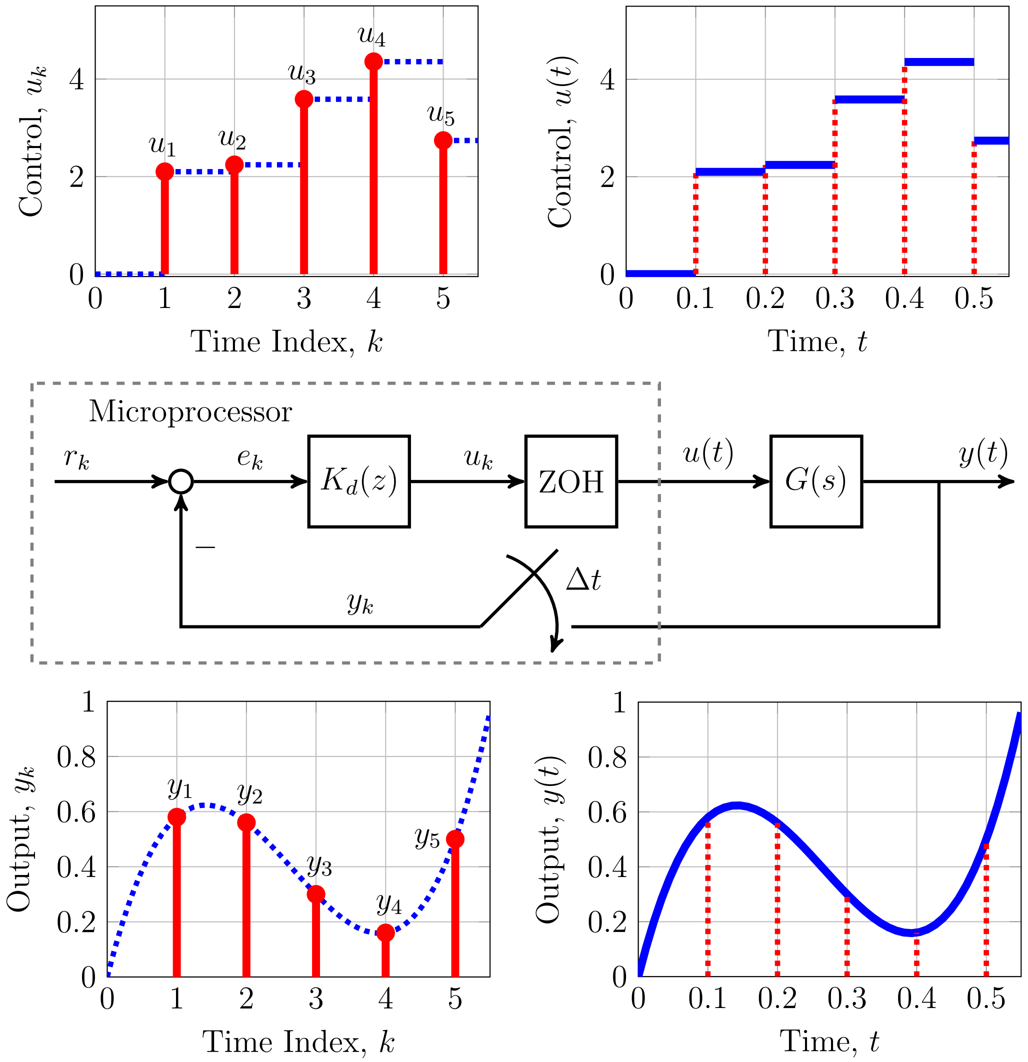


Figure 4.18: Discrete-time controller implementation. The signals $y(t)$ and $u(t)$ are in continuous-time. The signals y_k and u_k are in discrete-time with sampling time $\Delta t = 0.1sec$.

2. Control Update: The microprocessor uses the sampled measurement y_k and reference command r_k to compute the error as $e_k = r_k - y_k$. The discrete-time control u_k is then updated based on the error e_k. The discrete-time update relation, denoted by $K_d(z)$ in Figure 4.18, is chosen to approximate the behavior of the continuous-time controller $K(s)$. Details on this discrete-time update process are covered in Section 4.8.2. For now, it is sufficient to provide a particular example for such an update relation:

$$u_k = u_{k-1} + 5e_k - 4.9e_{k-1} \tag{4.34}$$

A relation of this form is called a *difference equation*. If the previous control u_{k-1} and error e_{k-1} are stored in memory then the current control u_k can be computed from e_k using Equation 4.34. One possible implementation is demonstrated in the pseudocode below.

```
Initialize: uprev=0, eprev=0
while(1)
   Obtain new samples: (r,y)
   Compute error: e = r - y
   Update control: u = uprev + 5*e - 4.9*eprev
   Update previous values:  uprev = u, eprev = e
end
```

3. Zero-order Hold: The microprocessor only updates the control u_k when it receives a new sample (r_k, y_k). A rule is required to convert the discrete-time signal u_k back into a continuous-time signal $u(t)$. One simple rule is a *zero-order hold (ZOH)*. This rule holds $u(t)$ constant in between the sample updates:

$$u(t) = u_k \text{ for } t \in [k\Delta t,\, (k+1)\Delta t) \tag{4.35}$$

For example, $u(t) = u_1$ for $t \in [\Delta t,\, 2\Delta t)$ and $u(t) = u_2$ for $t \in [2\Delta t,\, 3\Delta t)$. Figure 4.18 shows the control signals in both discrete and continuous time. The continuous-time control signal $u(t)$ has a stair-stepped behavior due to the zero-order hold.

Figure 4.18, as drawn, assumes the microprocessor can instantaneously perform its update calculations. For example, a measurement y_3 sampled at time $t = 3\Delta t$ immediately yields an updated control signal $u(t)$ at $t = 3\Delta t$. In reality, there will be some amount of computation delay τ_{delay}. Hence the measurement y_3 sampled at $t = 3\Delta t$ will cause the control signal to be updated at $t = 3\Delta t + \tau_{delay}$. Such time delays can degrade control performance and even cause instability as discussed further in Section 7.5.3.

4.8.2 Control Update

This section focuses on the update equation for the discrete-time control signal u_k. In particular, assume a controller $K(s)$ has been designed in continuous-time. The transfer function $K(s)$ represents an ODE that relates the continuous-time error $e(t)$ and control $u(t)$. The main objective is to approximate this continuous-time controller with an update equation that relates the discrete-time error e_k and control u_k.

To demonstrate the basic approach, consider a PI controller designed in continuous-time:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau \tag{4.36}$$

The discrete-time implementation updates the control every Δt seconds. Thus the objective is to compute the control $u_k := u(k\Delta t)$ at sample time $t = k\Delta t$ based on the current error $e_k := e(k\Delta t)$. In addition, the update equation can also depend on previous samples of the error and control, e.g. e_{k-1} and u_{k-1}. To obtain a discrete-time approximation, first note that the control signals produced by $K(s)$ at consecutive sample times are related by:

$$u(k\Delta t) - u\left((k-1)\Delta t\right) = K_p e(k\Delta t) - K_p e\left((k-1)\Delta t\right) + K_i \int_{(k-1)\Delta t}^{k\Delta t} e(\tau)\, d\tau \tag{4.37}$$

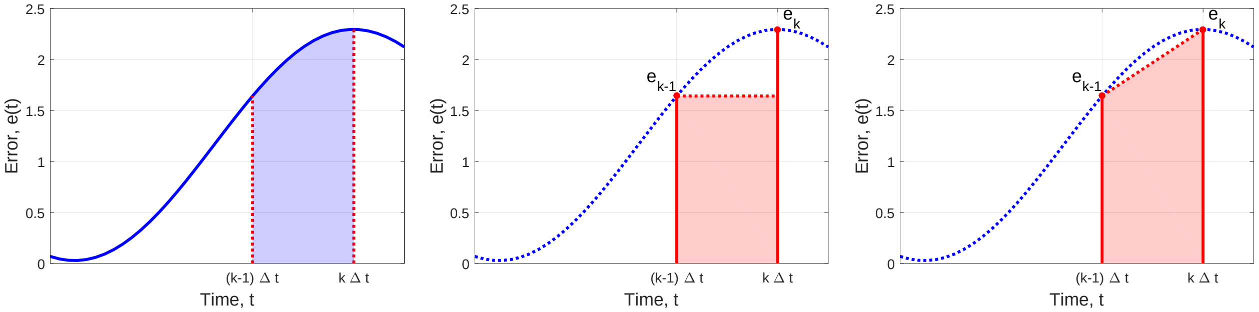


Figure 4.19: Integrals: Continuous-time (left), ZOH approximation (middle), and FOH approximation (right).

The continuous-time signals u and e evaluated at sample times $t = (k-1)\Delta t$ and $t = k\Delta t$ can be replaced by their discrete-time equivalents, e.g. replace $u(k\Delta t)$ by u_k. The main issue is that the integral term in Equation 4.37 involves the error signal $e(t)$ for the entire interval $t \in [(k-1)\Delta t,\, k\Delta t)$. An example of the exact (continuous-time) integral is represented by the shaded blue area in the left plot of Figure 4.19. The discrete-time update requires an approximation of this integral using e_k and/or e_{k-1}. There are many possible approximations. Two simple approximations are the *zero-order hold (ZOH)* and *first-order (FOH)*. A ZOH treats $e(t)$ as a constant on the interval, e.g. $e(t) = e_{k-1}$.[‖] The ZOH assumption approximates the integral by the shaded rectangular area as shown in the middle plot of Figure 4.19. A FOH instead treats $e(t)$ as varying linearly between the samples. This yields a better approximation for the integral given by the shaded trapezoidal area shown in the right plot of Figure 4.19. The terminology "zero-order" and "first-order" refer to the fact that a constant and linear relation are 0^{th} and 1^{st} order polynomials. An update equation for u_k is obtained by replacing the integral Equation 4.37 by one of these approximations. For example, the FOH approximation $\int_{(k-1)\Delta t}^{k\Delta t} e(\tau)\, d\tau \approx 0.5 \cdot (e_k + e_{k-1})\, \Delta t$ yields the following relation:

$$u_k = u_{k-1} + \left(K_P + K_i \frac{\Delta t}{2}\right) e_k - \left(K_p - K_i \frac{\Delta t}{2}\right)\, e_{k-1} \tag{4.38}$$

This is a difference equation that approximates the continuous-time controller. The FOH should be used since it provides a better approximation than a ZOH. There are many other continuous-to-discrete approximation methods but all provide similar difference equations if the sampling time Δt is sufficiently fast.

The Matlab function `c2d` converts a continuous-time system to a discrete-time approximation. The syntax is `Kd=c2d(K,DeltaT)` where `K` is a continuous-time controller, `DeltaT` is the sample time, and `Kd` is the discrete-time approximation. This default syntax uses a ZOH approximation. A FOH approximation can be obtained with the syntax `Kd=c2d(K,DeltaT,'foh')`. Additional approximation methods are available and details can be found in the help and documentation. Two simple examples are given at the end of this section.

[‖] The term ZOH is used in two different ways. It was first introduced for the simple rule to convert the control from discrete-time u_k to continuous-time $u(t)$. Here ZOH refers to an assumption on the continuous-time signal $e(t)$ that is used to approximate the controller in terms of the discrete-time signal e_k.

In general, discretizing an n^{th} order ODE yields an n^{th} order difference equation. For example, discretizing a second-order ODE yields a second-order difference equation:

$$a_2 u_{k+2} + a_1 u_{k+1} + a_0 u_k = b_2 e_{k+2} + b_1 e_{k+1} + b_0 e_k \tag{4.39}$$

Difference equations can also be compactly denoted using a transfer function notation. The transfer function associated with Equation 4.39 is:

$$K_d(z) = \frac{b_2 z^2 + b_1 z + b_0}{a_2 z^2 + a_1 z + a_0} \tag{4.40}$$

The transfer function is used here simply as a different representation of the difference equation. The variable "z" corresponds to shifting the discrete-time index by one unit. The transfer function $K_d(s)$ uses the subscript d to emphasize that it is a discrete-time representation. It is also important to note that Equation 4.39 can be equivalently written as:

$$a_2 u_k = -a_1 u_{k-1} - a_0 u_{k-2} + b_2 e_k + b_1 e_{k-1} + b_0 e_{k-2} \tag{4.41}$$

This simply re-defines the time index and re-arranges some terms. The form in Equation 4.39 is more standard. However, the form in Equation 4.41 can be directly interpreted as an update for u_k based on the current error e_k as well as the past two values of the control $\{u_{k-2}, u_{k-1}\}$ and error $\{e_{k-2}, e_{k-1}\}$.

Example 4.1. Consider a continuous-time PI controller with $K_p = 5$ and $K_i = 1$. The code below obtains ZOH and FOH approximations with the sample time $\Delta t = 0.1sec$.

```
% Construct continuous-time PI controller
>> Kp = 5;
>> Ki = 1;
>> K = tf([Kp Ki],[1 0]);

% ZOH Approximation
>> DeltaT = 0.1;
>> Kd = c2d(K,DeltaT)
Kd =
  5 z - 4.9
  ---------
    z - 1
Sample time: 0.1 seconds
Discrete-time transfer function.

% FOH Approximation
>> Kd = c2d(K,DeltaT,'foh')
Kd =
  5.05 z - 4.95
  -------------
      z - 1
Sample time: 0.1 seconds
Discrete-time transfer function.
```

The FOH yields a transfer function $K_d(z)$ that represents the following difference equation:

$$u_k = u_{k-1} + 5.05e_k - 4.95e_{k-1} \tag{4.42}$$

This agrees with the FOH approximation given in Equation 4.38. △

Example 4.2. This example computes a discrete-time approximation for the following second-order controller:

$$K(s) = \frac{s^2 + 2s + 3}{4s^2 + 5s + 6} \tag{4.43}$$

The discretization is performed with a sample time $\Delta t = 0.01sec$.

```
>> K = tf([1 2 3],[4 5 6]);
>> DeltaT = 0.01;
>> Kd = c2d(K,DeltaT,'foh')
Kd =
  0.2509 z^2 - 0.4968 z + 0.246
  -----------------------------
     z^2 - 1.987 z + 0.9876
Sample time: 0.01 seconds
Discrete-time transfer function.
```

The function yields discrete-time transfer function $K_d(z)$ that represents the following difference equation:

$$u_k = 1.987u_{k-1} - 0.9876u_{k-2} + 0.2509e_k - 0.4968e_{k-1} + 0.246e_{k-2} \tag{4.44}$$

△

4.9 Appendix: Modeling Details

Summary: This appendix provides details on the models used in this chapter for the DC motor and rocket attitude dynamics.

4.9.1 DC Motor Model Details

Figure 4.20 shows a simple diagram for a DC motor that includes both the electrical and mechanical components. The input voltage u (V) causes a current i (Amp) in the electrical circuit. The current through the motor windings generates a torque T_M (Nm) which rotates the motor shaft and inertia. The output y $(\frac{rad}{sec})$ is the rotational speed of the shaft. The rotation of the motor creates a back EMF, i.e. voltage, e (V) on the circuit. In addition, there may be a load torque T_L (Nm) which tends to decelerate the inertia. For example, if the motor is used to propel a rotor on multicopter then the aerodynamic forces will generate an opposing load torque on the motor shaft.

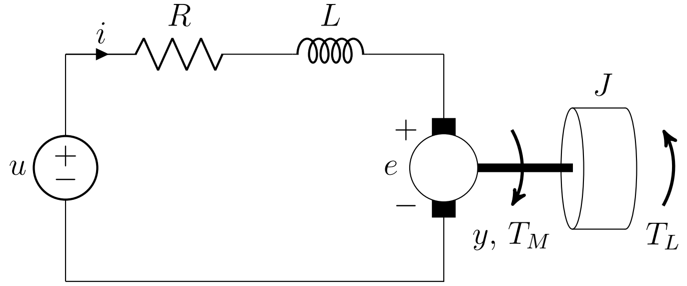


Figure 4.20: Diagram for a DC motor driving a rotational inertia.

The dynamic equations for the motor are given by two coupled first-order ODEs:

$$J\dot{y}(t) = T_M - T_L - b_c\,\mathrm{sign}(y(t)) \tag{4.45}$$

$$u(t) - L\frac{di}{dt}(t) - Ri(t) - e(t) = 0 \tag{4.46}$$

The first equation is Newton's second law for rotational systems: rotational inertia J times angular acceleration $\dot{y}(t)$ is equal to the sum of the torques on the motor shaft. The term $b_c\mathrm{sign}(y(t))$ represents a Coulomb friction that opposes rotation of the motor shaft. This friction does not change with the magnitude of the motor speed. A viscous friction term $b_v y(t)$ grows proportional to the motor speed. Viscous friction is neglected here but could be included in a more detailed model. The second equation is Kirchoff's voltage law: the sum of voltages around the circuit is equal to zero. This equation also assumes ideal relationships for the resistor and capacitor.

For an ideal motor, the back EMF is proportional to motor speed: $e(t) = K_e y(t)$ where K_e $(\frac{V\,sec}{rad})$ the back EMF constant. In addition, the motor torque is proportional to the circuit current: $T_M(t) = K_t i(t)$ where K_t $(\frac{Nm}{Amp})$ is the torque constant. Conservation of power can be used to show that $K_t = K_e$. In particular, the power balance with $y(t) > 0$ is:

$$u(t)i(t) = i(t)^2 R + L\frac{di}{dt}(t)\, i(t) + b_c y(t) + (T_M(t) - b_c)\, y(t) \tag{4.47}$$

The left side is the input electrical power. The first two terms on the right side are the losses in the resistor and the instantaneous power to the inductor. The remaining two terms are the mechanical energy losses due to the Coulomb friction and mechanical power output due to the net torque on the motor shaft. Re-arrange using Kirchoff's voltage law (Equation 4.46):

$$e(t)i(t) = T_M(t)y(t) \tag{4.48}$$

Finally, substitute $e(t) = K_e y(t)$ and $T_M(t) = K_t i(t)$ to show that $K_t = K_e$.

The motor model has the five parameters to be identified: J, L, R, b_c, and $K_t = K_e$. In addition, the input voltage should remain below a maximum operation voltage u_{max} to avoid damage to the motor. A specification sheet for a motor can be used to estimate typical values for these parameters. For example, Table 4.4 provides parameter values for a Precision Microdrives DC Motor 112-002 [14] with a small inertia attached to the motor shaft. The values for L, R, and u_{max} are directly from the specification sheet. The value for J is estimated based on the three blade inertia shown in Figure 4.1 (left). The remaining parameters can be estimated from data in the specification sheet. In particular, the specification sheet provides data under no load conditions ($T_L = 0\,Nm$) with the rated voltage $u_{rated} = 2.4\,V$. For these conditions the motor converges to a speed $y_{NL} = 17000\,rpm = 1780\,\frac{rad}{sec}$ and current $i_{NL} = 0.27\,Amp$. Kirchoff's voltage law (Equation 4.46) in steady state is:

$$u_{rated} - Ri_{NL} - K_e\, y_{NL} = 0 \tag{4.49}$$

Use the given no-load data to solve for $K_e = 0.0012\,\frac{V\,sec}{rad}$. As derived above, $K_t = K_e = 0.0012\,\frac{V\,sec}{rad}$. Newton's law (Equation 4.45) in steady-state with no load (assuming $y(t) > 0$) is:

$$0 = K_t i_{NL} - b_c \tag{4.50}$$

Thus the Coulomb friction is $b_c = 3.3 \times 10^{-4}\,Nm$. This friction will be neglected to simplify the remainder of the discussion but it should be included in a more detailed model.

At this point the model is complete but it takes the form of the two coupled first-order ODEs in Equations 4.45 and 4.46. It will be useful to combine these into a single second-order equation. A short-cut will be given later (Section XXX) to simplify some of the required algebra. For now, the algebra will be cumbersome but straightforward. Specifically, form $L\frac{d}{dt}$(Eq 4.45) $+ R \cdot$ (Eq 4.45) to obtain (neglecting Coulomb friction):

$$J(L\ddot{y}(t) + R\dot{y}(t)) = K_t\left(L\frac{di}{dt}(t) + Ri(t)\right) - L\dot{T}_L(t) - RT_L(t) \tag{4.51}$$

Use Equation 4.46 to substitute on the right side to obtain:

$$J(L\ddot{y}(t) + R\dot{y}(t)) = K_t(u - K_e y(t)) - L\dot{T}_L(t) - RT_L(t) \tag{4.52}$$

Parameter	Definition	Nominal Value	Units
J	Inertia	2×10^{-6}	$kg\,m^2$
L	Inductance	150×10^{-6}	H
R	Resistance	0.8	y
b_c	Coulomb friction	3.3×10^{-4}	Nm
K_t	Torque constant	1.23×10^{-3}	$\frac{Nm}{Amp}$
K_e	Back EMF constant	1.23×10^{-3}	$\frac{V\,sec}{rad}$
u_{max}	Maximum Voltage	3.0	V

Table 4.4: Parameters for Precision Microdrives DC Motor 112-002 [14] with small inertia.

Collect all terms involving the output $y(t)$ on the left side:

$$(JL)\,\ddot{y}(t) + (JR)\,\dot{y}(t) + (K_tK_e)y(t) = K_tu(t) - L\dot{T}_L(t) - RT_L(t) \tag{4.53}$$

This second-order ODE has two inputs: i) a controllable input u and, ii) a load torque input T_L. For simplicity, gather the terms involving the load torque into a single disturbance $d(t) := -\frac{1}{K_t}(L\dot{T}_L(t) + RT_L(t))$. With this definition, the ODE model is:

$$(JL)\,\ddot{y}(t) + (JR)\,\dot{y}(t) + (K_tK_e)y(t) = K_tu(t) + K_td(t) \tag{4.54}$$

The benefit of this definition is that the ODE from d to y is the same as the ODE from u to y. Using the nominal parameter values specified in Table 4.4 yields the following motor model:

$$(3 \times 10^{-10})\,\ddot{y}(t) + (1.6 \times 10^{-6})\,\dot{y}(t) + (1.51 \times 10^{-6})\,y(t) = (1.23 \times 10^{-3})\,u(t) \tag{4.55}$$

The corresponding transfer function for this second order model is:

$$G_2(s) = \frac{1.23 \times 10^{-3}}{3 \times 10^{-10}\,s^2 + 1.6 \times 10^{-6}\,s + 1.505 \times 10^{-6}} \tag{4.56}$$

This second-order system has a DC gain of $G_2(0) = 813\frac{rad}{sec V}$. In addition, the two poles are at $s_1 = -0.94$ and $s_2 = -5.33 \times 10^{-3}$ with corresponding time constants $\tau_1 = -1.06\,sec$ and $\tau_2 = -1.9 \times 10^{-4}\,sec$. The slow pole s_1 is much slower than the fast pole s_2. Hence, the motor dynamics are approximately first order with settling time 3.2 sec ($= 3\tau_1$). An approximate, first-order model can be constructed using the procedure described in Section 3.7.3. This procedure constructs the first-order model with the slow pole at s_1 and with DC gain equal $G_2(0)$. A similar first-order approximation can be obtained by setting $L = 0$ in Equation 4.46. This neglects the "fast" electrical dynamics and retains the "slower" rotational inertia dynamics. Setting $L = 0$ in Equation 4.54 yields the following first-order approximation:

$$\dot{y}(t) + 0.94\,y(t) = 768.8\,u(t) + 768.8\,d(t) \tag{4.57}$$

4.9.2 Rocket Attitude Model

The free-body diagram in Figure 4.21 below shows the key forces involved in the attitude control problem. T (N) is the thrust force produced by the engine rocket and F_a (N) is the aerodynamic force. The rocket attitude, i.e. the heading angle, is denoted by y (rad). The direction of the thrust can be changed by rotating the rocket nozzle. The input u (rad) is the angle between the rocket thrust and the centerline of the rocket. The objective is to use the control input u to ensure that the rocket attitude y tracks some desired reference heading r.

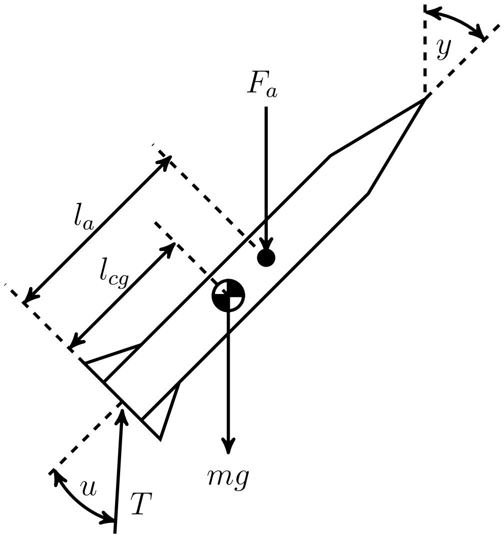


Figure 4.21: Rocket Free-body Diagram

Table 4.5 shows the key model parameter for the rocket attitude dynamics. Several parameters for the model are taken from [11]. The model makes several simplifying assumptions, e.g. it ignores the moments due to the fuel sloshing in the fuel tank. In addition, it assumes that F_a acts vertically downward as drawn in the diagram. However, the simplified model will be sufficient to highlight the key issues in attitude control. The rocket attitude dynamics are given by Newton's second law for rotational systems. Summing the moments about the center of gravity gives:

$$J\ddot{y}(t) = T\ l_{cg} \sin u(t) + F_a\,(l_a - l_{cg}) \sin y(t) \tag{4.58}$$

This is a nonlinear ODE due to the $\sin u$ and $\sin y$ terms. The system has an equilibrium point at $(\dot{\bar{y}}, \bar{y}, \bar{u}) = (0, 0, 0)$. Linearizing around this trim condition gives the following linear ODE:

$$J\ddot{y}(t) = T\ l_{cg} u(t) + F_a\,(l_a - l_{cg})\, y(t) \tag{4.59}$$

After plugging in the parameter values from Table 4.5 and re-arranging terms we obtain:

$$\ddot{y}(t) - 0.1225\, y(t) = 6.3163\, u(t) \tag{4.60}$$

Parameter	Definition	Nominal Value	Units
J	Moment of inertia about CG	2.49×10^8	$kg\, m^2$
l_{cg}	Length from nozzle to CG	30.48	m
l_a	Length from nozzle to aero. center	33.53	m
T	Engine thrust	5.16×10^7	N
m	Rocket mass	1.456×10^6	kg
F_a	Aerodynamic force	1×10^7	N

Table 4.5: Parameters for Rocket Attitude Dynamics. Several parameters are from [11].

Chapter 5

Frequency Response

The previous chapter discussed various versions of PID for the control of first and second order systems. The designs involved tuning a small number of gains based on relations between the following properties of the closed-loop system:

- Coefficients of the ODE, e.g. (ζ, ω_n)
- Poles/roots of the characteristic equation and their locations in the complex plane
- Transient response characteristics with a specific focus on the step response (rise time, settling time, and overshoot)

This approach is based in the *time-domain*, i.e. it focuses on the transient response of the system. This approach becomes more difficult to apply for higher order systems and/or more complicated controllers. More advanced design methods are based in the *frequency-domain*. In particular, these methods rely on an understanding of the system response to sinusoidal inputs. This chapter presents the basic tools for understanding the frequency response characteristics. Subsequent chapters use these tools to develop new control design and analysis methods.

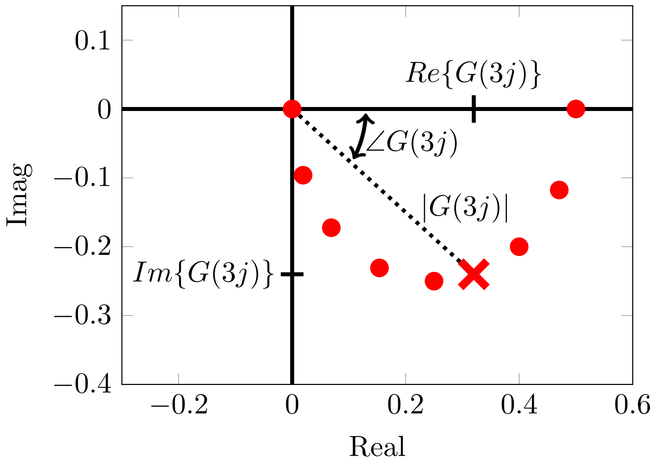
5.1 Steady-State Sinusoidal Response

Summary: The transfer function $G(s)$ is used to express the solution of a stable linear system forced by a sinusoidal input. If the input is $u(t) = \sin(\omega t)$ then the response satisfies $y(t) \to |G(j\omega)| \sin(\omega t + \angle G(j\omega))$ as $t \to \infty$. The output converges to a sinusoid at the same frequency as the input but with amplitude scaled by $|G(j\omega)|$ and phase shifted by $\angle G(j\omega)$.

5.1.1 Revisiting the Transfer Function

The transfer function $G(s)$ has mainly been used as notation for an ODE. However, it will now play a key role in describing the solution of the ODE forced by a sinusoidal input. In particular, the sinusoidal solution depends on the transfer function evaluated at a purely imaginary number $s = j\omega$ where $\omega \in \mathbb{R}$ is the frequency in $\frac{rad}{sec}$. The result $G(j\omega)$ is a complex number that can be expressed either in Cartesian form by its real/imaginary parts or in polar form by its magnitude $|G(j\omega)|$ and phase $\angle G(j\omega)$. A simple example demonstrating this notation is given below. Refer back to Appendix 3.8.1 for a brief review of complex functions.

Example 5.1. $G(s) = \frac{2}{s+4}$ is the transfer function for a stable, first order system. The value at $\omega = 0$ is $G(0) = 0.5$ which is the DC gain of the system. At the frequency $\omega = 3\frac{rad}{sec}$ the function is $G(3j) = \frac{2}{3j+4}$. Multiply the numerator and denominator by the conjugate $-3j + 4$. This yields the Cartesian form: $G(3j) = \frac{8-6j}{25} = 0.32 - 0.24j$. The magnitude and phase are $|G(3j)| = \sqrt{0.32^2 + 0.24^2} = 0.4$ and $\angle G(3j) = \tan^{-1}\left(-\frac{0.24}{0.32}\right) = -0.64 rads$. The complex number $G(3j)$ is shown graphically in the left plot of Figure 5.1 (red x). The plot also shows $G(j\omega)$ at several other frequencies (red dots). The corresponding complex numbers are given in both Cartesian and polar forms (Figure 5.1 on right). The phase is between $[-\frac{\pi}{2}, 0]$ for all frequencies and this corresponds to the points remaining in the fourth quadrant. Note that $G(j\omega) \to \frac{2}{j\omega} = -\frac{2j}{\omega}$ as $\omega \to \infty$. In other words, $G(j\omega)$ approaches the origin along the negative imaginary axis as $\omega \to \infty$. This corresponds to a phase $\approx -\frac{\pi}{2}$ as $\omega \to \infty$.



ω	$G(j\omega)$, Cartesian	$G(j\omega)$, Polar
0	0.500	$0.500e^{0j}$
1	$0.471 - 0.118j$	$0.485e^{-0.245j}$
2	$0.400 - 0.200j$	$0.447e^{-0.463j}$
3	$0.320 - 0.240j$	$0.400e^{-0.644j}$
4	$0.250 - 0.250j$	$0.354e^{-0.785j}$
6	$0.154 - 0.231j$	$0.277e^{-0.983j}$
10	$0.069 - 0.172j$	$0.186e^{-1.190j}$
20	$0.019 - 0.096j$	$0.098e^{-1.373j}$
∞	0	$0e^{-\frac{\pi}{2}j}$

Figure 5.1: Transfer function $G(s) = \frac{2}{s+4}$ evaluated at $s = j\omega$ for several frequencies. Results are shown in the complex plane (left) and as tabulated data (right).

△

5.1.2 Sinusoidal Response for First Order Systems

The term *sinusoidal response* refers to the forced response of a system with a sinusoidal input, e.g. $u(t) = \sin(\omega t)$ or $u(t) = \cos(\omega t)$ where ω is a given frequency in $\frac{rad}{sec}$. The form of the forced response solution as $t \to \infty$ will be of particular interest. This is typically called the *steady-state sinusoidal response.*[*] This section derives the steady-state sinusoidal response solution for a first order system of the following form:

$$\begin{aligned} &\dot{y}(t) + a_0 y(t) = b_0 u(t) \\ &\text{IC: } y(0) = y_0 \end{aligned} \tag{5.1}$$

The corresponding transfer function is $G(s) = \frac{b_0}{s+a_0}$. It is assumed that $a_0 > 0$. This ensures that the system is (BIBO) stable and hence the response $y(t)$ due to the sinusoidal input remains bounded. The solution can be derived for $u(t) = \sin(\omega t)$ or $u(t) = \cos(\omega t)$. However, it will simplify the algebra to first derive the forced response solution for the complex sinusoid $u(t) = e^{j\omega t}$. This will yield solutions for both (real) sine and cosine inputs.

The procedure in Section 3.3.1 is applied to solve for the forced response of Equation 5.1 with input $u(t) = e^{j\omega t}$. First, the characteristic equation $s + a_0 = 0$ has only a single (real) root $s = -a_0$. Next, a particular solution $y_P(t)$ is found for the forced ODE yielding a general solution of the form $y(t) = y_P(t) + c_1 e^{-a_0 t}$. Finally, the initial condition $y(0) = y_0$ is used to solve for the coefficient c_1. The system is assumed to be stable ($a_0 > 0$). As a result, the term $c_1 e^{-a_0 t}$ decays to zero after the settling time $3\tau_1 = \frac{3}{a_0} sec$. This implies $y(t) \to y_P(t)$ as $t \to \infty$ and the steady-state sinusoidal response is given by the particular solution. An expression for $y_P(t)$ can be obtained from the method of undetermined coefficients [2, 5]. This method requires a "guess" at the solution $y_P(t)$ of the forced ODE:

$$\dot{y}_P(t) + a_0 y_P(t) = b_0 e^{j\omega t} \tag{5.2}$$

Assume y_P has the same form as the input: $y_P(t) = c_P e^{j\omega t}$ where $c_P \in \mathbb{C}$ is a complex coefficient to be determined. Substitute this assumed particular solution into Equation 5.2:

$$(j\omega + a_0) c_P e^{j\omega t} = b_0 e^{j\omega t} \tag{5.3}$$

The complex exponential $e^{j\omega t}$ is nonzero for any value of ω and t. Hence $y_P(t) = c_P e^{j\omega t}$ is a particular solution if and only if $c_P = \frac{b_0}{j\omega + a_0}$. This yields the important connection back to the transfer function: $c_P = G(j\omega)$ or, expressed in polar form, $c_P = |G(j\omega)| e^{j\angle G(j\omega)}$. Thus the steady state sinusoidal response due to the input $u(t) = e^{j\omega t}$ is given by the particular solution:

$$y_P(t) = |G(j\omega)| e^{j(\omega t + \angle G(j\omega))} \tag{5.4}$$

The complex functions $(u(t), y_P(t))$ satisfy the forced ODE in Equation 5.2. To obtain real-valued solutions, take the imaginary part of the forced ODE:

$$\frac{d}{dt} Im\{y_P(t)\} + a_0\, Im\{y_P(t)\} = b_0 Im\{e^{j\omega t}\} \tag{5.5}$$

[*] Note that the response is oscillating as $t \to \infty$ so it is not "steady" in the sense of being a constant.

Thus the real-valued functions $(Im\{u(t)\}, Im\{y_P(t)\})$ also satisfy the ODE. Next recall Euler's formula (Appendix 3.8.1): $e^{j\phi} = \cos(\phi) + j\sin(\phi)$ for any real number ϕ. By Euler's formula, the imaginary part of the input is $Im\{u(t)\} = Im\{e^{j\omega t}\} = \sin(\omega t)$. Similarly the imaginary part of the particular solution is given by $Im\{y_P(t)\} = |G(j\omega)| \sin(\omega t + \angle G(j\omega))$. Putting together these facts leads to the following conclusion:

Conclusion: Assume the first order system in Equation 5.1 is stable. If the input is $u(t) = \sin(\omega t)$ then the response satisfies $y(t) \to |G(j\omega)| \sin(\omega t + \angle G(j\omega))$ as $t \to \infty$. In other words, the output is a sinusoid at the same frequency as the input but the amplitude is scaled by $|G(j\omega)|$ and the phase is shifted by $\angle G(j\omega)$.

Similarly, $y(t) \to |G(j\omega)| \cos(\omega t + \angle G(j\omega))$ if the input is $u(t) = \cos(\omega t)$. This can be shown by taking the real parts of the forced ODE in Equation 5.2. It is a key property of linear systems that the steady-state sinusoidal solution can be expressed in terms of the transfer function. The next section demonstrates that this holds even for higher order, stable linear systems. The control analysis and design methods developed in the remainder of the class are based on this fact.

Example 5.2. Consider the following stable, first order system:

$$\begin{aligned} &\dot{y}(t) + 4y(t) = 2u(t) \\ &\text{IC: } y(0) = 3 \end{aligned} \tag{5.6}$$

The transfer function for this system is $G(s) = \frac{2}{s+4}$. This is the same transfer function considered in Example 5.1. Evaluating the transfer function at $s = j\omega$ yields:

$$G(j\omega) = \frac{2}{j\omega + 4} = \frac{-2j\omega + 8}{\omega^2 + 16}, \quad |G(j\omega)| = \frac{\sqrt{4\omega^2 + 64}}{\omega^2 + 16}, \quad \text{and } \angle G(j\omega) = \tan^{-1}\left(-\frac{\omega}{4}\right)$$

Thus if $u(t) = \sin(\omega t)$ then the output converges to the following steady-state response:

$$y_{ss}(t) = \frac{\sqrt{4\omega^2 + 64}}{\omega^2 + 16} \sin\left(\omega t + \tan^{-1}\left(-\frac{\omega}{4}\right)\right) \tag{5.7}$$

As a specific example, if $\omega = 2\frac{rad}{sec}$ then $|G(2j)| = 0.447$ and $\angle G(2j) = -0.464 rads$. Thus if $u(t) = \sin(2t)$ then the output converges to $y_{ss}(t) = 0.447 \sin(2t - 0.464)$. This input/output pair is shown in the left plot of Figure 5.2. This graph confirms that the output converges to a sinusoid with amplitude 0.447 after an initial transient. The steady-state output can be re-written as $y_{ss}(t) = 0.447 \sin(2(t - 0.232))$. This implies that the output is shifted to the right by $0.232 sec$ relative to the input. For example, the input $u(t)$ crosses zero at $t = 2\pi sec$ and the output crosses zero at $t = 2\pi + 0.232 \approx 6.515 sec$.

As another example, if $\omega = 4\frac{rad}{sec}$ then $|G(4j)| = 0.354$ and $\angle G(4j) = -0.785 rads$. Thus if $u(t) = \sin(4t)$ then the output converges to $y_{ss}(t) = 0.354 \sin(4t - 0.785)$. This input/output pair is shown in the right plot of Figure 5.2. This graph confirms that the output converges to a sinusoid with amplitude 0.354 after an initial transient. The steady-state output can be re-written as $y_{ss}(t) = 0.354 \sin(4(t - 0.196))$. This implies that the output is shifted to the right by $0.196 sec$ relative to the input. For example, the input $u(t)$ crosses zero at $t = 2\pi sec$ and the output crosses zero at $t = 2\pi + 0.196 \approx 6.480 sec$.

△

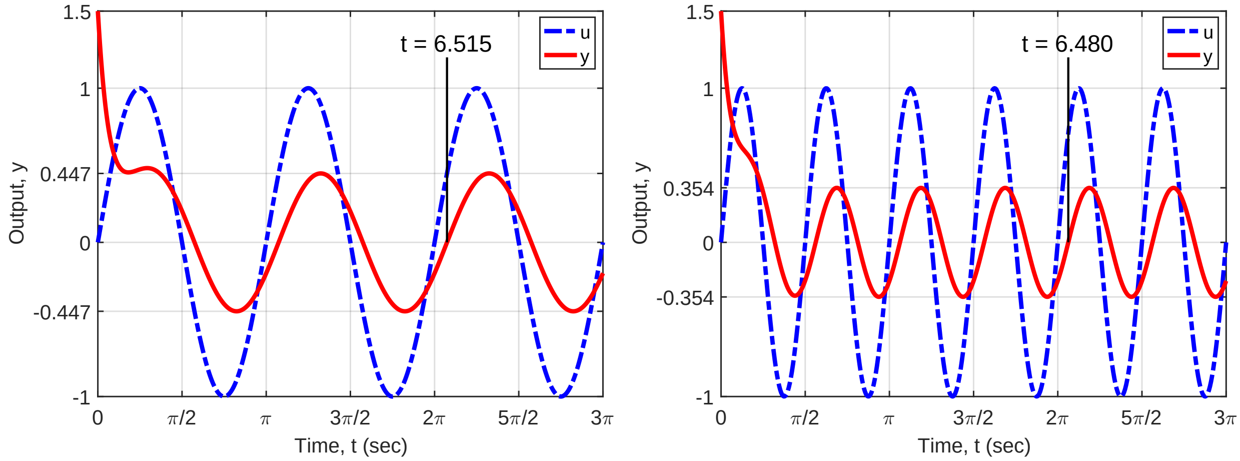


Figure 5.2: Response for $G(s) = \frac{2}{s+4}$ with input $u(t) = \sin(\omega t)$. Input frequencies are $\omega = 2\frac{rad}{sec}$ (left) and $\omega = 4\frac{rad}{sec}$ (right).

In general the steady-state frequency response can be re-written as:

$$y_{ss}(t) = |G(j\omega)| \sin\left(\omega(t - t_{shift})\right) \quad \text{where} \quad t_{shift} := -\frac{\angle G(j\omega)}{\omega} \tag{5.8}$$

If $\angle G(j\omega) < 0$ then $t_{shift} > 0$ represents the amount that the steady-state output is shifted to the right relative to the input. Thus $\angle G(j\omega) < 0$ is called *phase lag* because the output “lags” behind the input. Conversely, if $\angle G(j\omega) > 0$ then the steady-state output is shifted to the left relative to the input. Thus $\angle G(j\omega) > 0$ is called *phase lead* because the output “leads” ahead of the input.

5.1.3 Sinusoidal Response for Higher Order Systems

This section derives the sinusoidal steady-state response for higher order linear systems. The basic steps are only sketched as they are similar to those used to obtain the solution for a first order system. Consider the following n^{th} order system:

$$\begin{aligned}&a_n y^{[n]}(t) + a_{n-1} y^{[n-1]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^{[m]}(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t)\\ &\text{IC: } y(0) = y_0;\ \dot{y}(0) = \dot{y}_0;\ \ldots;\ y^{[n-1]}(0) = y_0^{[n-1]}\end{aligned} \tag{5.9}$$

The corresponding transfer function for this system is:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{5.10}$$

It is assumed that the system is stable, i.e. all poles are in the LHP. This ensures that the system is (BIBO) stable and hence the response $y(t)$ due to the sinusoidal input remains bounded. As before, the solution is first derived for $u(t) = e^{j\omega t}$. This will yield solutions for both (real) sine and cosine inputs.

The procedure in Section 3.3.1 is again applied to solve for the forced response of Equation 5.9 with input $u(t) = e^{j\omega t}$. First, the characteristic equation $a_n s^n + \cdots + a_1 s + a_0 = 0$ has n roots $\{s_1, \ldots, s_n\}$. Next, a particular solution $y_P(t)$ is found for the forced ODE yielding a general solution of the form $y(t) = y_P(t) + \sum_{i=1}^{n} c_i e^{s_i t}$. Finally, the initial conditions are used to solve for the coefficients $\{c_1, \ldots, c_n\}$. The system is assumed to be stable ($Re\{s_i\} < 0$ for each i) and hence each term $c_i e^{s_i t}$ decays to zero. The settling time for this decay can be determined based on the real part of the poles. This implies $y(t) \to y_P(t)$ as $t \to \infty$ and the steady-state sinusoidal response is given by the particular solution.

An expression for $y_P(t)$ can again be obtained from the method of undetermined coefficients [2, 5]. This method requires a "guess" at the solution $y_P(t)$ of the forced ODE. Assume $y_P(t) = c_P e^{j\omega t}$ where $c_P \in \mathbb{C}$ is a complex coefficient to be determined. Substitute both this assumed particular solution as well as $u(t) = e^{j\omega t}$ into Equation 5.9:

$$\left(a_n \left(j\omega\right)^n + \cdots + a_1 \left(j\omega\right) + a_0\right) c_P e^{j\omega t} = \left(b_m \left(j\omega\right)^m + \cdots + b_1 \left(j\omega\right) + b_0\right) e^{j\omega t} \tag{5.11}$$

The complex exponential $e^{j\omega t}$ is nonzero for any value of ω and t. Hence $y_P(t) = c_P e^{j\omega t}$ is a particular solution if and only if $c_P = G(j\omega)$. Thus the steady state sinusoidal response due to the input $u(t) = e^{j\omega t}$ is given by the particular solution:

$$y_P(t) = |G(j\omega)| e^{j(\omega t + \angle G(j\omega))} \tag{5.12}$$

The complex functions $(u(t), y_P(t))$ satisfy the forced ODE (Equation 5.9). The remaining steps to convert this into real-valued solutions are identical to those given in the previous section. This leads to the similar important conclusion regarding the steady-state sinusoidal response for higher order systems:

Conclusion: Assume the n^{th} order system in Equation 5.9 is stable. If the input is $u(t) = \sin(\omega t)$ then the response satisfies $y(t) \to |G(j\omega)| \sin\left(\omega t + \angle G(j\omega)\right)$ as $t \to \infty$. In other words, the output is a sinusoid at the same frequency as the input but the amplitude is scaled by $|G(j\omega)|$ and the phase is shifted by $\angle G(j\omega)$.

Similarly, $y(t) \to |G(j\omega)| \cos\left(\omega t + \angle G(j\omega)\right)$ if the input is $u(t) = \cos(\omega t)$. More generally if $u(t) = A \sin(\omega t + \theta)$ for some numbers (A, θ) then the steady-state response is

$$y_{ss}(t) = A |G(j\omega)| \sin\left(\omega t + \theta + \angle G(j\omega)\right) \tag{5.13}$$

In other words, the output amplitude is simply the input amplitude multiplied by $|G(j\omega)|$. Moreover, the output phase is simply the input phase plus the additional phase $\angle G(j\omega)$.

5.2 Bode Plots

Summary: A Bode plot consists of two subplots: gain vs. frequency and phase vs. frequency. Such plots are useful to understand the steady-state response to sinusoids of different frequencies. Several simple examples, including a differentiator and integrator, are given to provide insight for the data contained in a Bode plot.

5.2.1 An Overview of Bode Plots

A *Bode Plot* is a common tool used to understand the frequency response of a linear system. As shown in Section 5.1, the steady-state sinusoidal response of a linear system is described by the magnitude $|G(j\omega)|$ and phase $\angle G(j\omega)$ of the transfer function. It is useful to have graphical displays of these quantities to understand how the system responds to sinusoids of different frequencies. A Bode plot, named after Hendrik Bode, consists of two subplots:

- **Bode Magnitude (Gain) Plot:** This is a plot of gain vs. frequency. The horizontal axis is ω on a log (base 10) scale in units of $\frac{rad}{sec}$. The vertical axis is the gain expressed by the quantity $|G(j\omega)|_{dB} := 20\log_{10}|G(j\omega)|$. The subscript dB denotes that this quantity is the magnitude in units of decibels. The unit decibel, named after Alexander Graham Bell, was originally used for power measurements in telecommunications systems. Values in dB can be converted back to actual units by the formula $|G(j\omega)| = 10^{(|G(j\omega)|_{dB}/20)}$. The table below shows some conversions between actual units and dB. **Increasing or decreasing by a multiplicative factor of 10 in actual units corresponds to increasing or decreasing additively by $+20$ or -20 in units of dB.**

$\lvert G(j\omega)\rvert$	0.001	0.01	0.1	0.25	0.5	$\frac{1}{\sqrt{2}}$	1	$\sqrt{2}$	2	4	10	100	1000
$\lvert G(j\omega)\rvert_{dB}$	-60	-40	-20	-12	-6	-3	0	3	6	12	20	40	60

- **Bode Phase Plot:** This is a plot of phase vs. frequency. The horizontal axis is ω on a log (base 10) scale in units of $\frac{rad}{sec}$. The vertical axis is the phase $\angle G(j\omega)$ in degrees.

There are a few important comments regarding these plots. First, the two subplots can sometimes appear in slightly different forms. For example, the horizontal axis can appear in units of Hz ($\frac{cycles}{sec}$) where $1\,Hz = 2\pi\frac{rad}{sec}$. Thus Bode plots should be carefully examined to determine the precise data and axes being used. Second, the steady-state frequency response solution was derived in Section 5.1 for stable systems. However, Bode plots can still be drawn for unstable systems and these will be used later for control design and analysis. Finally, it takes some effort to become accustomed to the use of log scales and dB units. However, there are several benefits of generating the gain and phase plots in this way. For example, this format provides a better display of data across wide ranges of frequency and gain. In addition, the properties of logarithms will allow these plots to be easily used for control design.

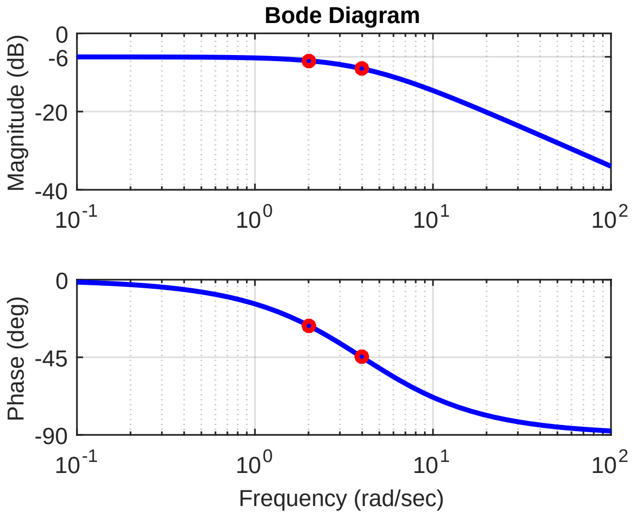
Bode plots can be generated with the Matlab function `bode`. The syntax is `bode(G)` where `G` is a system. This default syntax generates the gain and phase plots in the form described above. The alternative syntax `[mag,phase,w]=bode(G)` returns the magnitude (in actual units), phase (in *degs*), and frequency (in $\frac{rad}{sec}$). The magnitude in units of dB can then be calculated as `magdb = 20*log10(mag)`. Additional options are available and details can be found in the help and documentation. A simple example is given below.

Example 5.3. $G(s) = \frac{2}{s+4}$ is the transfer function for a stable, first order system. This transfer function was previously studied in Examples 5.1 and 5.2. The Bode plot for $G(s)$ is shown on the left of Figure 5.3. A similar plot can be created with the following Matlab commands:

```
>> G = tf(2,[1 4]);
>> bode(G);
```

Figure 5.3 also shows the tabulated magnitude (dB) and phase (deg) at several frequencies. The red circles on the Bode plot correspond to the frequencies $\omega = 2$ and $4 \frac{rad}{sec}$. These are the input frequencies for the sinusoidal responses shown in Figure 5.2. Both the magnitude (in dB) and phase are more negative at $\omega = 4\frac{rad}{sec}$. This corresponds to a smaller amplitude and increased phase shift for the response to $u(t) = \sin(4t)$ (right plot in Figure 5.2).

The Bode plot provides additional information about the response. For example, at low frequencies the magnitude and phase satisfy $|G(j\omega)|_{dB} \approx -6dB$ and $\angle G(j\omega) \approx 0degs$. Converting the magnitude to actual units yields $|G(j\omega)| \approx 10^{-6/20} \approx 0.5$. Hence for low frequencies the input $u(t) = \sin(\omega t)$ yields a steady-state output $y_{ss}(t) \approx 0.5\sin(\omega t)$. As another example, $|G(j\omega)|_{dB} \to -\infty$ as $\omega \to \infty$. In actual units, this corresponds to $|G(j\omega)| \to 0$ as $\omega \to \infty$. Thus for high frequencies the input $u(t) = \sin(\omega t)$ yields an output sinusoid of very small amplitude. This roughly means that the system does not respond to high frequency sinusoids.



ω $(\frac{rad}{sec})$	$G(j\omega)$ Polar	$\lvert G(j\omega)\rvert_{dB}$ (dB)	$\angle G(j\omega)$ (deg)
0	$0.500e^{0j}$	-6.0	0
1	$0.485e^{-0.245j}$	-6.3	-14.0
2	$0.447e^{-0.464j}$	-7.0	-26.6
3	$0.400e^{-0.644j}$	-8.0	-36.9
4	$0.354e^{-0.785j}$	-9.0	-45.0
6	$0.277e^{-0.983j}$	-11.1	-56.3
10	$0.186e^{-1.190j}$	-14.6	-68.2
20	$0.098e^{-1.373j}$	-20.2	-78.7
∞	$0e^{-\frac{\pi}{2}j}$	$-\infty$	-90

Figure 5.3: Bode plot (left) for $G(s) = \frac{2}{s+4}$ and data evaluated at several $s = j\omega$ (right). The red circles on the Bode plot highlight the data at $\omega = 2$ and $4\frac{rad}{sec}$.

△

5.2.2 Bode Plot: Differentiator

This section focuses on a simple system known as a differentiator. The analysis of this system provides additional insight regarding the graphical data in a Bode plot. It will also be useful to understand the properties of derivative control. Consider the following first order system:

$$y(t) = \dot{u}(t) \tag{5.14}$$

The corresponding transfer function is $G(s) = \frac{s}{1} = s$.[†] This system is called a *differentiator* because the output is simply the derivative of the input. If $u(t) = \sin(\omega t)$ then differentiation yields $y(t) = \omega\cos(\omega t)$. The output can be equivalently written in the form $y(t) = \omega\sin(\omega t + \frac{\pi}{2})$. Thus the output amplitude is scaled by ω and the phase is shifted by $\frac{\pi}{2} rad = +90 deg$. The right side of Figure 5.4 shows the response for $\omega = 0.5$ and $2\,\frac{rad}{sec}$. In both cases the output achieves its peak one quarter cycle before ($= +90deg$) the input, i.e the output "leads" the input. Also note that the output amplitude grows proportional to frequency. **Thus the differentiator "amplifies" sinusoids at higher frequencies.**

A Bode plot provides a graphical summary of these results. The frequency response for the differentiator is $G(j\omega) = j\omega$. This is a positive imaginary number for all values of $\omega > 0$. Thus the phase is $\angle G(j\omega) = \frac{\pi}{2} rad = 90deg$ for all $\omega > 0$. This agrees with the results above. Moreover, the magnitude in actual units is $|G(j\omega)| = \omega$, i.e. magnitude is proportional to frequency. At frequencies $\{0.1, 0.5, 1, 2, 10\}\ \frac{rad}{sec}$, the magnitude is $\{0.1, 0.5, 1, 2, 10\}$ in actual units and $\{-20, -6, 0, +6, +20\}$ in dB. The Bode plot for the differentiator shown in Figure 5.4 displays these results. **It is common to state that the magnitude plot has a slope of $+20dB$ per decade.** This means increasing by a (multiplicative) factor of 10 on the frequency axis corresponds to an (additive) increase of $20dB$ on the magnitude axis. In actual units, multiplying frequency by a factor of 10 corresponds to multiplying the magnitude by 10.

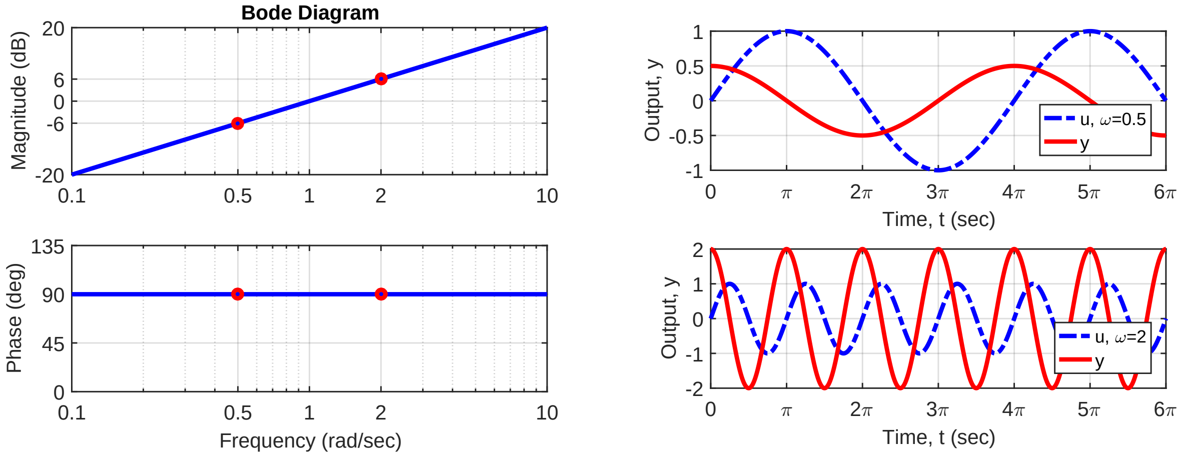


Figure 5.4: Bode plot for a differentiator $G(s) = s$ (left) and frequency responses for $\omega = 0.5$ and $2\,\frac{rad}{sec}$ (right).

5.2.3 Bode Plot: Integrator

This section focuses on a simple system known as an integrator. Again, the analysis of this system is intended to provide insight regarding Bode plots. It will also be useful to understand

[†] The differentiator is non-proper because the ODE contains the first derivative of u but no derivatives of y. This is the reason the transfer function numerator is a polynomial of higher degree than the denominator.

the properties of integral control. Consider the following first order system:

$$\dot{y}(t) = u(t) \tag{5.15}$$

The transfer function is $G(s) = \frac{1}{s}$. This system is an *integrator* because the output is the integral of the input: $y(t) = y(0) + \int_0^t u(\tau)d\tau$. If $u(t) = \sin(\omega t)$ then integration yields:

$$y(t) = \left(y(0) + \frac{1}{\omega}\right) - \frac{1}{\omega}\cos(\omega t) \tag{5.16}$$

The integrator has a pole at $s = 0$ and hence an integrator is an unstable system. Thus the effect of the initial condition $y(0)$ does not decay as $t \to \infty$. However, if $y(0) = -\frac{1}{\omega}$ then $y(t) = -\frac{1}{\omega}\cos(\omega t)$. This is a particular solution for Equation 5.15 forced by $u(t) = \sin(\omega t)$. This solution can be re-written as $y(t) = \frac{1}{\omega}\sin(\omega t - \frac{\pi}{2})$. The amplitude is scaled by $\frac{1}{\omega}$ and the phase is shifted by $\frac{\pi}{2}rad = -90deg$. The right side of Figure 5.5 shows the response for $\omega = 0.5$ and $2\,\frac{rad}{sec}$. In both cases the output achieves its peak one quarter cycle after ($= -90deg$) the input, i.e the output "lags" the input. Also note that the output amplitude is inversely related to frequency. **Thus the integrator "amplifies" sinusoids at lower frequencies.**

A Bode plot again provides a graphical summary of these results. The frequency response for an integrator is $G(j\omega) = \frac{1}{j\omega} = \frac{-j}{\omega}$. This is a negative imaginary number for all values of $\omega > 0$. Thus the phase is $\angle G(j\omega) = -\frac{\pi}{2}rad = -90deg$ for all $\omega > 0$. The magnitude in actual units is $|G(j\omega)| = \frac{1}{\omega}$, i.e. magnitude is inversely related to frequency. At frequencies $\{0.1, 0.5, 1, 2, 10\}$ $\frac{rad}{sec}$, the magnitude is $\{10, 2, 1, 0.5, 0.1\}$ in actual units and $\{+20, +6, 0, -6, -20\}$ in dB. The Bode plot for the integrator shown in Figure 5.5 confirms these results. **It is common to state that this magnitude plot has a slope of** $-20dB$ **per decade.** This means that increasing by a (multiplicative) factor of 10 on the frequency axis corresponds to an (additive) decrease of $-20dB$ on the magnitude axis. In actual units, multiplying frequency by a factor of 10 corresponds to dividing the magnitude by 10.

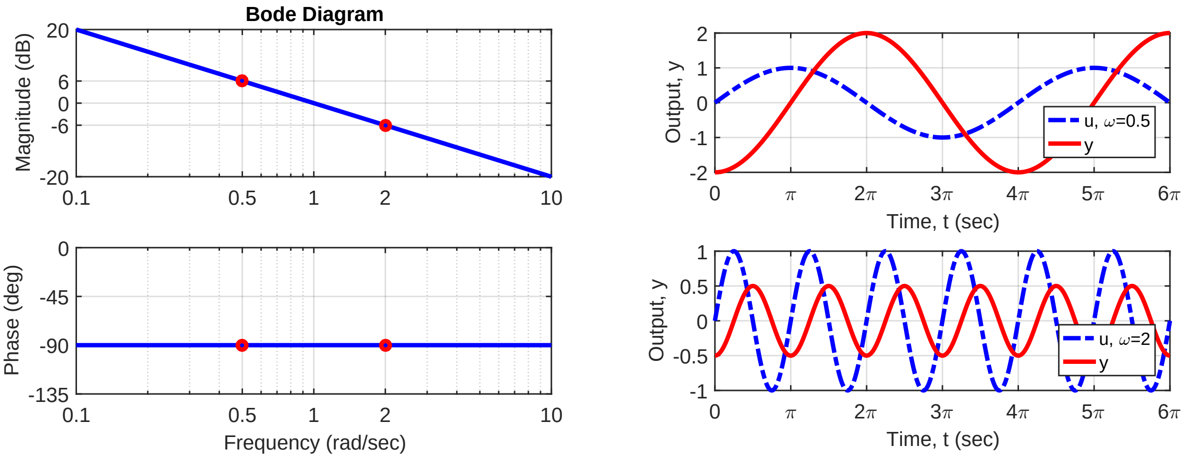


Figure 5.5: Bode plot for an integrator $G(s) = \frac{1}{s}$ (left) and frequency responses for $\omega = 0.5$ and $2\,\frac{rad}{sec}$ (right).

5.3 Bode Plots: First Order Systems

Summary: This section focuses on Bode plots for first order systems. A pole defines a corner frequency for the system. The magnitude plot is flat at low frequencies and rolls off at $-20dB$ per decade at high frequencies. The phase plot transitions by $\pm 90^o$ near the corner frequency but the precise details depend on the signs of the ODE coefficients. A first order zero has similar features but its magnitude plot rolls up at $+20dB$ per decade at high frequencies.

5.3.1 First Order System

The next few sections will briefly describe how to sketch approximate Bode plots by hand. The actual Bode plot can always be created in Matlab using the command `bode`. However, the ability to sketch approximate Bode plots by hand will serve a number of purposes. It will help connect the Bode plot to properties of the system transfer function including its transient response characteristics. In addition, the ability to approximately sketch Bode plots will be used to design more advanced controllers.

To start, consider the following first order system:

$$\dot{y}(t) + a_0 y(t) = b_0 u(t) \tag{5.17}$$

The transfer function for this system is $G(s) = \frac{b_0}{s+a_0}$. Assume the system is stable ($a_0 > 0$) and has positive input gain ($b_0 > 0$). Systems where these sign conditions do not hold are discussed later in the section. The left side of Figure 5.6 shows the Bode plot for the specific system $G(s) = \frac{2}{s+4}$. The magnitude is flat at $|G(0)|_{dB} = -6dB$ up to about $\omega = 4\frac{rad}{sec}$ and then "rolls off" at -20 dB per decade at higher frequencies. The phase plot transitions from 0^o at low frequencies to -90^o at high frequencies. It passes through -45^o of phase at $\omega = 4\frac{rad}{sec}$.

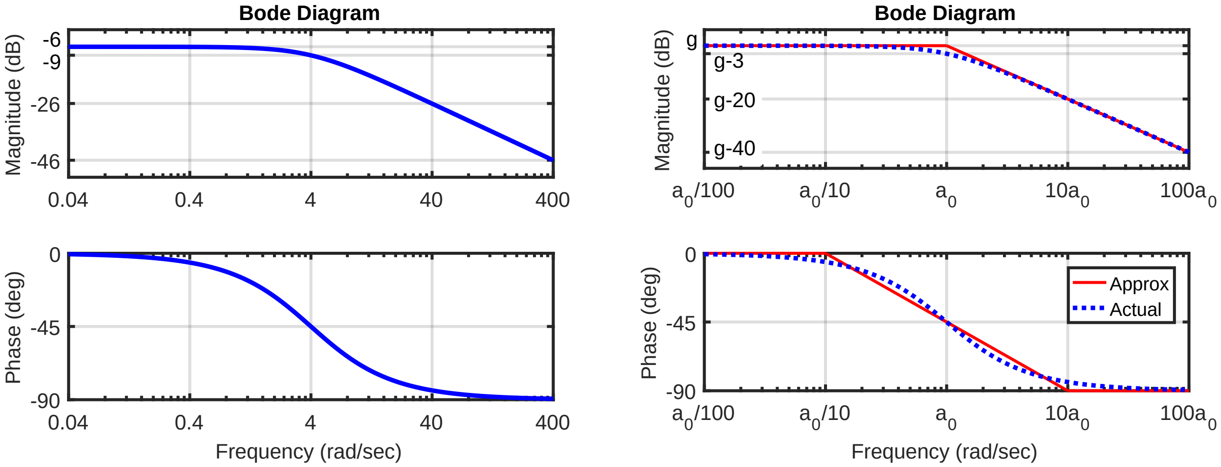


Figure 5.6: Left: Bode plot for $G(s) = \frac{2}{s+4}$. Right: Bode plot for generic first order system $G(s) = \frac{b_0}{s+a_0}$ where $a_0 > 0$ and $b_0 > 0$. The notation $g := |G(0)|_{dB}$ is used on the magnitude plot. The Bode plot also includes straight-line asymptotic approximations.

In fact every first order system $G(s) = \frac{b_0}{s+a_0}$ has similar characteristics when both $a_0 > 0$ and $b_0 > 0$. The right side of Figure 5.6 shows the Bode plot for this general case (dotted blue). It also shows *straight-line Bode approximations* (solid red). The Bode plot for a general first order system has the following features:

- **Corner Frequency:** The magnitude plot roughly begins its high frequency roll-off at the frequency $\omega = a_0$. This is known as the *corner frequency.* The transfer function at the corner frequency is $G(ja_0) = \frac{b_0}{ja_0+a_0}$. This can be rewritten in terms of the DC gain as $G(ja_0) = \frac{G(0)}{j+1}$. Thus the magnitude (in actual units) and phase at the corner frequency are $|G(ja_0)| = \frac{G(0)}{\sqrt{2}}$ and $\angle G(ja_0) = -\frac{\pi}{4} rad = -45^o$.[‡] To convert the magnitude to dB, recall that $\log_{10}(\frac{c_1}{c_2}) = \log_{10}(c_1) - \log_{10}(c_2)$ for any real numbers c_1 and c_2. Thus the magnitude in dB at the corner frequency is:

$$\begin{aligned} |G(ja_0)|_{dB} &= 20\log_{10}\left(\frac{G(0)}{\sqrt{2}}\right) = 20\log_{10} G(0) - 20\log_{10}\sqrt{2} \\ &= |G(0)|_{dB} - 3 \end{aligned} \tag{5.18}$$

  The equation above uses the fact $20\log_{10}\sqrt{2} \approx 3$. To summarize, the exact phase is -45^o at the corner frequency $\omega = a_0$ and the magnitude $-3dB$ below the DC gain. These features are displayed in both plots in Figure 5.6.

- **Low Frequency Approximation:** If $\omega \leq \frac{a_0}{10}$ then $G(j\omega) = \frac{b_0}{j\omega+a_0} \approx \frac{b_0}{a_0}$. Hence the magnitude and phase satisfy $|G(j\omega)|_{dB} \approx |G(0)|_{dB}$ and $\angle G(j\omega) \approx 0^o$ at low frequencies.

- **High Frequency Approximation:** If $\omega \geq 10a_0$ then $G(j\omega) = \frac{b_0}{j\omega+a_0} \approx \frac{b_0}{j\omega} = -\frac{b_0 j}{\omega}$. Hence the magnitude and phase satisfy $|G(j\omega)|_{dB} \approx \left|\frac{b_0}{\omega}\right|_{dB}$ and $\angle G(j\omega) \approx -90^o$ at high frequencies. The magnitude is inversely proportional to frequency ω and hence it rolls off at $-20dB$ per decade similar to an integrator.

- **Middle Frequency Approximation:** The middle frequency "straight-line" approximation connects the low and high frequency approximations. In the magnitude plot, this corresponds to extending the low and high frequency approximations so they intersect at the corner frequency. This straight line extension yields the approximate gain of $|G(0)|_{dB}$ at the corner frequency $\omega = a_0$. This approximation is slightly higher than the actual gain $|G(ja_0)|_{dB} = |G(0)|_{dB} - 3dB$. In the phase plot, the middle frequency approximation connects 0^o of phase at $\omega = \frac{a_0}{10}$ to -90^o of phase at $\omega = 10a_0$. This approximation passes through the exact phase of $\angle G(ja_0) = -45^o$ at the corner frequency.[§]

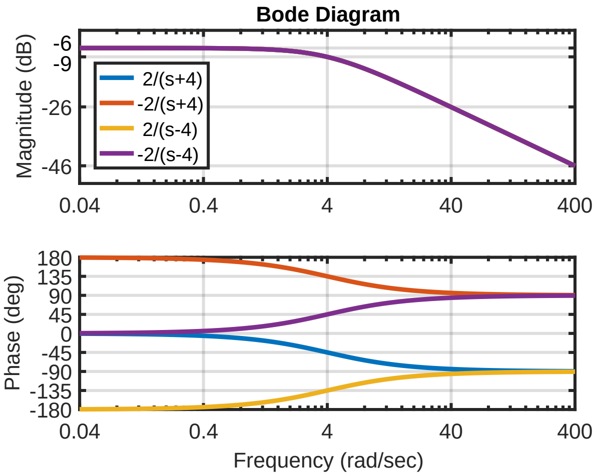
It is useful to connect these results back to the first order step response response characteristics covered in Section 3.5. Recall that the time constant is $\tau_1 = \frac{1}{a_0} sec$ and the settling time is $3\tau_1 = \frac{3}{a_0} sec$. Thus increasing a_0 corresponds to decreasing settling time, i.e. faster response. On a Bode plot, increasing a_0 corresponds to increasing the corner frequency. Thus

[‡] At the corner frequency, $G(ja_0) = \frac{G(0)}{j+1} \cdot \frac{-j+1}{-j+1} = 0.5G(0) - 0.5G(0)j$. Thus the magnitude is $|G(ja_0)| = G(0)\sqrt{0.5^2 + 0.5^2} = \frac{G(0)}{\sqrt{2}}$. The phase is $\angle G(ja_0) = \tan^{-1}\left(\frac{-0.5G(0)}{0.5G(0)}\right) = \tan^{-1}(-1) = -\frac{\pi}{4} rad$.

[§] The middle frequency phase approximation is a straight line on the log scale. The precise formula is not particularly important but for completeness it is given by: $\angle G(j\omega) \approx -90^o \cdot \log_{10}(\frac{\omega}{0.1a_0}) / \log_{10}(\frac{10a_0}{0.1a_0})$.

larger corner frequencies are associated with faster response. Also recall that if $u(t) = \bar{u}$ then the output converges to the steady-state value $\bar{y} = \frac{b_0}{a_0}\bar{u} = G(0)\bar{u}$. On a Bode plot, $G(0)$ is the low frequency approximate gain. Thus the gain for low frequency sinusoids is approximately the gain observed for step inputs.

Next consider the case where either $a_0 < 0$ and/or $b_0 < 0$. The left side of Figure 5.7 shows Bode plots corresponding to all four combinations for the signs of a_0 and b_0. Note that the magnitude plot is identical for all four cases. This is essentially due to the fact that $|j\omega + 4| = |(j\omega - 4)|$. However, the phase is different for each case. The right side shows a table of the transfer function and phase evaluated at DC, the corner frequency, and as $\omega \rightarrow \infty$ for each of the four transfer functions. This data can be used to sketch the phase plot. In particular, the straight line approximation transitions from the low frequency phase to the high frequency phase as described above. It passes through the exact phase at the corner frequency.



	$\omega\left(\frac{rad}{sec}\right)$	$\frac{2}{j\omega+4}$	$\frac{-2}{j\omega+4}$	$\frac{2}{j\omega-4}$	$\frac{-2}{j\omega-4}$
$G(j\omega)$	0	0.5	-0.5	-0.5	0.5
	4	$\frac{-j+1}{4}$	$\frac{j-1}{4}$	$\frac{-j-1}{4}$	$\frac{j+1}{4}$
	$\rightarrow \infty$	$\frac{-2j}{\omega}$	$\frac{2j}{\omega}$	$\frac{-2j}{\omega}$	$\frac{2j}{\omega}$
$\angle G(j\omega)$ (deg)	0	0	180	-180	0
	4	-45	135	-135	45
	$\rightarrow \infty$	-90	90	-90	90

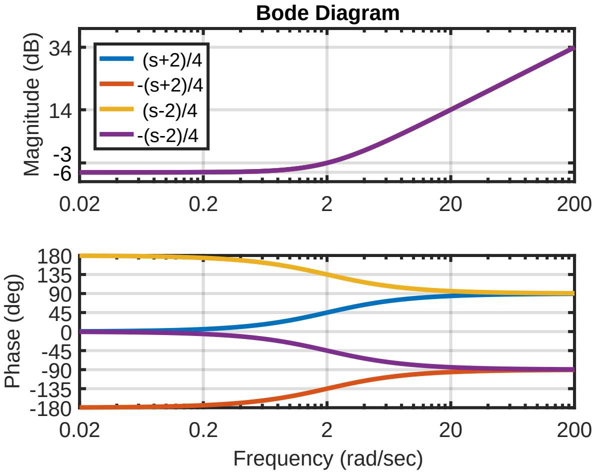
Figure 5.7: Left: Bode plots for $G(s) = \frac{\pm 2}{s \pm 4}$. The magnitude plots are identical for all four cases but the phase plots differ. Right: Transfer function and phase evaluated at several $s = j\omega$.

5.3.2 First Order Zero

Next consider the following first order system:

$$a_0 y(t) = \dot{u}(t) + b_0 u(t) \tag{5.19}$$

The transfer function is $G(s) = \frac{s+b_0}{a_0}$. This has a zero at $s = -b_0$ and this defines the *corner frequency* $\omega = |b_0|\frac{rad}{sec}$ for the system. Analysis of this system is motivated by PD control which has a similar form. Figure 5.8 shows results for the specific values $a_0 = \pm 4$ and $b_0 = \pm 2$. The magnitude plot is identical for all four cases. In particular, the magnitude is flat at $|G(0)|_{dB} = -6dB$ up to about $\omega = 2\frac{rad}{sec}$ and then "rolls up" at $+20dB$ per decade at higher frequencies. The phase plot depends on the signs of a_0 and b_0 but in all cases it transitions by $\pm 90^o$ around the frequency $\omega = 2\frac{rad}{sec}$. The right side of Figure 5.8 shows a table of the transfer function and phase evaluated at DC, the corner frequency, and as $\omega \rightarrow \infty$ for each of the four systems. This data can be used to sketch the phase plot.



	$\omega\left(\frac{rad}{sec}\right)$	$\frac{j\omega+2}{4}$	$\frac{j\omega+2}{-4}$	$\frac{j\omega-2}{4}$	$\frac{j\omega-2}{-4}$
$G(j\omega)$	0	0.5	-0.5	-0.5	0.5
	2	$\frac{2j+1}{2}$	$\frac{-2j-1}{2}$	$\frac{2j-1}{2}$	$\frac{-2j+1}{2}$
	$\to\infty$	$\frac{j\omega}{4}$	$\frac{-j\omega}{4}$	$\frac{j\omega}{4}$	$\frac{-j\omega}{4}$
$\angle G(j\omega)$	0	0	-180	180	0
	2	45	-135	135	-45
(deg)	$\to\infty$	90	-90	90	-90

Figure 5.8: Left: Bode plots for $G(s) = \frac{s\pm 2}{\pm 4}$. The magnitude plots are identical for all four cases but the phase plots differ. Right: Transfer function and phase evaluated at several $s = j\omega$.

In fact every first order zero $G(s) = \frac{s+b_0}{a_0}$ has similar characteristics. Straight-line Bode approximations can be drawn based on the following features:

- **Corner Frequency:** The magnitude plot roughly begins its high frequency roll-up at the corner frequency $\omega = |b_0|$. The absolute value is used here to account for both cases $b_0 > 0$ and $b_0 < 0$. The transfer function at the corner frequency is $G(j|b_0|) = \frac{j|b_0|+b_0}{a_0}$. This can be rewritten in terms of the DC gain as $G(j|b_0|) = G(0)\,(\pm j + 1)$. Thus the magnitude (in actual units) at the corner frequency is $|G(jb_0)| = |G(0)|\sqrt{2}$. To convert the magnitude to dB, recall that $\log_{10}(c_1 c_2) = \log_{10}(c_1) + \log_{10}(c_2)$ for any real numbers c_1 and c_2. Thus the magnitude in dB at the corner frequency is:

$$\begin{aligned} |G(j|b_0|)|_{dB} &= 20\log_{10}\left(|G(0)|\sqrt{2}\right) = 20\log_{10}|G(0)| + 20\log_{10}\sqrt{2} \\ &= |G(0)|_{dB} + 3 \end{aligned} \tag{5.20}$$

  The equation above uses the fact $20\log_{10}\sqrt{2} \approx 3$. Thus the exact magnitude at the corner frequency is $+3dB$ above the DC gain. The phase depends on the signs of a_0 and b_0 as demonstrated by the table in Figure 5.6.

- **Low Frequency Approximation:** If $\omega \leq \frac{|b_0|}{10}$ then $G(j\omega) = \frac{j\omega+b_0}{a_0} \approx \frac{b_0}{a_0}$. Hence the magnitude satisfies $|G(j\omega)|_{dB} \approx |G(0)|_{dB}$ at low frequencies. The phase $\angle G(j\omega) \approx \angle G(0)$ which can be 0^o or $\pm 180^o$ depending on the signs of a_0 and b_0.

- **High Frequency Approximation:** If $\omega \geq 10|b_0|$ then $G(j\omega) = \frac{j\omega+b_0}{a_0} \approx \frac{j\omega}{a_0}$. Hence the magnitude satisfies $|G(j\omega)|_{dB} \approx \left|\frac{b_0}{\omega}\right|_{dB}$. The magnitude is proportional to frequency ω and hence it rolls up at $+20dB$ per decade similar to an differentiator. The phase $\angle G(j\omega) \approx \angle\frac{j\omega}{a_0}$ which can be $\pm 90^o$ depending on the sign of a_0.

- **Middle Frequency Approximation:** The middle frequency "straight-line" approximation connects the low and high frequency approximations. In the magnitude plot, this corresponds to extending the low and high frequency approximations so they intersect at the corner frequency. This straight line extension yields the approximate gain of $|G(0)|_{dB}$ at the corner frequency $\omega = |b_0|$. This approximation is slightly lower than the actual gain $|G(j|b_0|)|_{dB} = |G(0)|_{dB} + 3dB$. In the phase plot, the middle frequency approximation connects the phases from the low and high frequency approximations. For example, if both $a_0 > 0$ and $b_0 > 0$ then the the system has 0^o of phase at low frequencies and $+90^o$ at high frequencies. The middle frequency approximation connects 0^o of phase at $\omega = \frac{|b_0|}{10}$ to $+90^o$ of phase at $\omega = 10|b_0|$. This approximation passes through the exact phase of $\angle G(j|b_0|) = +45^o$ at the corner frequency.[¶]

[¶] This middle frequency phase approximation is a straight line on the log scale. The precise formula is not particularly important but for completeness it is given by: $\angle G(j\omega) \approx +90^o \cdot \log_{10}(\frac{\omega}{0.1|b_0|})/\log_{10}(\frac{10|b_0|}{0.1|b_0|})$.

5.4 Bode Plots: Second Order Systems

Summary: This section focuses on Bode plots for second order systems. A second order differentiator has $+180deg$ of phase and magnitude slope of $+40NdB$ per decade. A double integrator has $-180deg$ of phase and magnitude slope of $-40NdB$ per decade. Next, an underdamped second order system is characterized by its damping ratio ζ and natural frequency ω_n. Such a system has a corner frequency at ω_n. The magnitude plot is flat at low frequencies and rolls off at $-40dB$ per decade at high frequencies. For stable systems with positive input gain, the phase plot transitions from 0^o to -180^o. If the system has low damping ($\zeta \ll 1$) then it will have a resonant (peak) gain of $|G(j\omega_n)| \approx \frac{1}{2\zeta}$ near the natural frequency.

5.4.1 Second Order Differentiator and Integrator

Sections 5.2.2 and 5.2.3 discuss first order differentiators and integrators. This section reviews similar results for the second order case. First, consider the following second order system:

$$y(t) = \ddot{u}(t) \tag{5.21}$$

The corresponding transfer function is $G(s) = \frac{s^2}{1} = s^2$. This is a second order differentiator because the output is simply the second-derivative of the input. If $u(t) = \sin(\omega t)$ then differentiating twice yields $y(t) = -\omega^2 \sin(\omega t) = \omega^2 \sin(\omega t + \pi)$. Thus the output amplitude is scaled by ω^2 and the phase is shifted by $\pi rad = +180deg$. The Bode plot in Figure 5.9 (left) displays these results. Specifically, the frequency response is $G(j\omega) = (j\omega)^2 = -\omega^2$. This is a negative real number and hence the phase is $\angle G(j\omega) = +180deg$. Moreover, the magnitude in actual units is $|G(j\omega)| = \omega^2$. At frequencies $\{0.1, 0.5, 1, 2, 10\}$ $\frac{rad}{sec}$, the magnitude is $\{0.01, 0.25, 1, 4, 100\}$ in actual units and $\{-40, -12, 0, +12, +40\}$ in dB. It is common to state that the magnitude plot has a slope of $+40dB$ per decade. This means increasing by a (multiplicative) factor of 10 on the frequency axis corresponds to an (additive) increase of $40dB$ on the magnitude axis. In actual units, multiplying frequency by a factor of 10 corresponds to multiplying the magnitude by 100. Thus the second order differentiator has a large amplification of sinusoids at high frequencies. **Generalizing these results, a N^{th} order derivative $G(s) = s^N$ has phase $+90Ndeg$ and magnitude slope of $+20NdB$ per decade.**

Similar results can be obtained for a second order integrator:

$$\ddot{y}(t) = u(t) \tag{5.22}$$

The transfer function is $G(s) = \frac{1}{s^2}$. This system is a second order integrator (also called a "double integrator") because the output $y(t)$ is obtained by integrating the input $u(t)$ twice. Thus if $u(t) = \sin(\omega t)$ then integrating twice and accounting for ICs $y(0)$ and $\dot{y}(0)$ yields:

$$y(t) = y(0) + \left(\dot{y}(0) + \frac{1}{\omega}\right) t - \frac{1}{\omega^2}\sin(\omega t) \tag{5.23}$$

The system has two poles at $s = 0$ and hence is unstable. Thus the terms involving ICs $y(0)$ and $\dot{y}(0)$ do not decay as $t \to \infty$. In fact, one term grows unbounded with time. However, if $y(0) = 0$ and $\dot{y}(0) = -\frac{1}{\omega}$ then $y(t) = -\frac{1}{\omega^2}\sin(\omega t)$. This is a particular solution for Equation 5.22

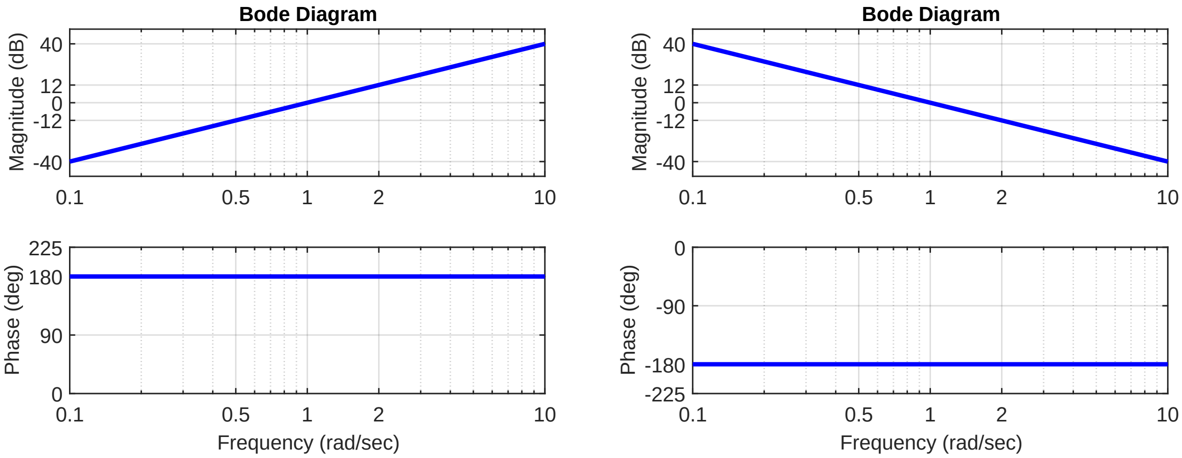


Figure 5.9: Bode plots for $G(s) = s^2$ (left) and $G(s) = \frac{1}{s^2}$ (right).

forced by $u(t) = \sin(\omega t)$. This solution can be re-written as $y(t) = \frac{1}{\omega^2}\sin(\omega t - \pi)$. The amplitude is scaled by $\frac{1}{\omega^2}$ and the phase is shifted by $-\pi rad = -180 deg$. The Bode plot for the integrator shown in Figure 5.9 (right) confirms these results. The frequency response is $G(j\omega) = \frac{1}{(j\omega)^2} = \frac{-1}{\omega^2}$. This is a negative real number and the phase is $\angle G(j\omega) = -180deg$. The magnitude in actual units is $|G(j\omega)| = \frac{1}{\omega^2}$, i.e. magnitude is inversely related to frequency squared. At frequencies $\{0.1, 0.5, 1, 2, 10\}\ \frac{rad}{sec}$, the magnitude is $\{100, 4, 1, 0.25, 0.01\}$ in actual units and $\{+40, +12, 0, -12, -40\}$ in dB. It is common to state that this magnitude plot has a slope of $-40dB$ per decade. This means that increasing by a (multiplicative) factor of 10 on the frequency axis corresponds to an (additive) decrease of $-40dB$ on the magnitude axis. In actual units, multiplying frequency by a factor of 10 corresponds to dividing the magnitude by 100. **Generalizing these results, a N^{th} order integrator $G(s) = \frac{1}{s^N}$ has phase $-90Ndeg$ and magnitude slope of $-20NdB$ per decade.**

5.4.2 Underdamped Second Order System

This section will discuss the Bode plot for a stable, underdamped second order system. Recall from Section 3.6 that the ODE for such a system can be expressed in terms of the damping ratio ζ and natural frequency ω_n:

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = b_0 u(t) \qquad (5.24)$$

The transfer function for this system is $G(s) = \frac{b_0}{s^2+2\zeta\omega_n s+\omega_n^2}$. In addition, assume the system has positive input gain ($b_0 > 0$). The left side of Figure 5.10 shows the Bode plot for two specific cases: $(b_0, \zeta, \omega_n) = (16, 0.7, 4)$ and $(16, 0.05, 4)$ corresponding to $G(s) = \frac{16}{s^2+5.6s+16}$ and $G(s) = \frac{16}{s^2+0.4s+16}$. In both cases the magnitude is flat at $|G(0)|_{dB} = 0dB$ for low frequencies ($\omega \le 0.4\frac{rad}{sec}$) and "rolls off" at -40 dB per decade at high frequencies ($\omega \ge 40\frac{rad}{sec}$). The main distinction is that the lightly damped system ($\zeta = 0.05$) has a large $+20dB$ peak near the natural frequency. Both phase plots transition from 0^o at low frequencies to -180^o at high frequencies. Moreover, both pass through -90^o of phase at $\omega = 4\frac{rad}{sec}$. The main distinction is that the lightly damped system ($\zeta = 0.05$) has a very sharp phase transition from 0^o to -180^o. The system with higher damping ($\zeta = 0.7$) has a much smoother transition.

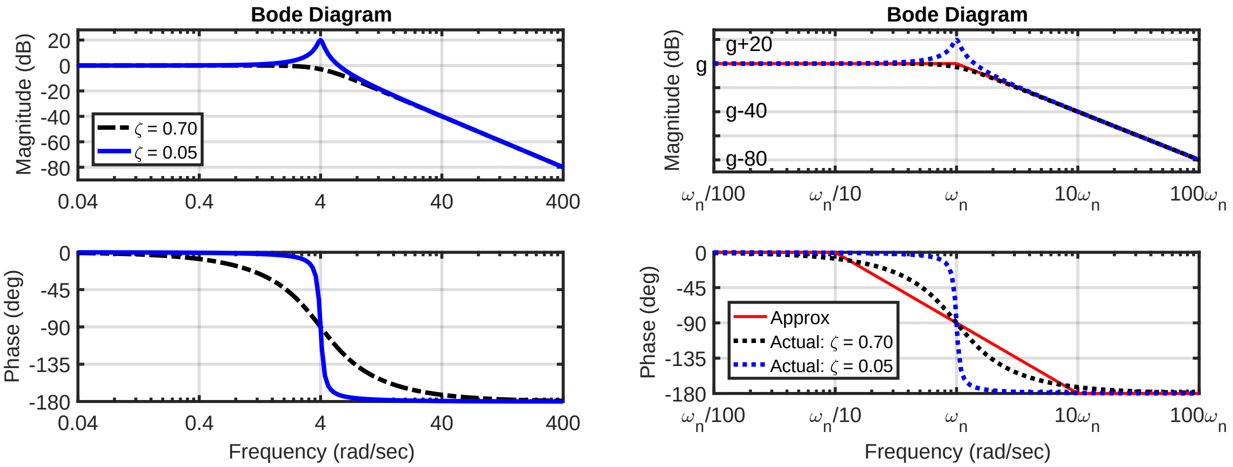


Figure 5.10: Left: Bode plots for $G(s) = \frac{16}{s^2+8\zeta s+16}$ with $\zeta = 0.7,\, 0.05$. Right: Bode plot for generic first order system $G(s) = \frac{b_0}{s^2+2\zeta\omega_n s+\omega_n^2}$ with $b_0 > 0$ and $\zeta = 0.7,\, 0.05$. The notation $g := |G(0)|_{dB}$ is used on the magnitude plot. The Bode plot includes straight-line approximations.

In fact every stable, underdamped second order system $G(s) = \frac{b_0}{s^2+2\zeta\omega_n s+\omega_n^2}$ has similar characteristics when $b_0 > 0$. The right side of Figure 5.10 shows the Bode plot for this general case for two different damping ratios (dotted lines). It also shows *straight-line Bode approximations* (solid red). The Bode plot for the general case has the following features:

- **Corner Frequency:** The magnitude plot roughly begins its high frequency roll-off at $\omega = \omega_n$. Thus the corner frequency for the underdamped system is the natural frequency. The transfer function at the corner frequency is $G(j\omega_n) = \frac{b_0}{2\zeta\omega_n^2 j}$. This can be rewritten in terms of the DC gain as $G(j\omega_n) = -\frac{G(0)j}{2\zeta}$. Thus the magnitude (in actual units) and phase at the corner frequency are $|G(j\omega_n)| = \frac{G(0)}{2\zeta}$ and $\angle G(j\omega_n) = -\frac{\pi}{2} rad = -90^o$. The magnitude in dB at the corner frequency is:

$$|G(j\omega_n)|_{dB} = 20\log_{10}\left(\frac{G(0)}{2\zeta}\right) = |G(0)|_{dB} - 20\log_{10}(2\zeta) \tag{5.25}$$

  If $\zeta = 0.7$ then the $|G(j\omega_n)|_{dB} \approx |G(0)|_{dB} - 3dB$, i.e. the gain at the corner frequency is $\approx -3dB$ below the DC gain. This is similar to the characteristics of a first-order pole. However, if $\zeta = 0.05$ then $|G(j\omega_n)|_{dB} \approx |G(0)|_{dB} + 20dB$. Thus lightly damped systems ($\zeta \ll 1$) have a large *resonant peak*. The precise resonance (peak) frequency is derived via calculus in Appendix 5.8. Specifically, if $\zeta < \frac{1}{\sqrt{2}}$ then the peak frequency is $\omega_p = \omega_n\sqrt{1-2\zeta^2}$ and the peak gain is $|G(j\omega_p)| = \frac{|G(0)|}{2\zeta\sqrt{1-\zeta^2}}$. **More importantly, if $\zeta \ll 1$ then this resonance (peak) frequency is near the natural frequency $\omega_p \approx \omega_n$ and the peak gain is $|G(j\omega_p)| \approx \frac{G(0)}{2\zeta}$.** These features are displayed in both plots in Figure 5.10.

- **Low Frequency Approximation:** If $\omega \leq \frac{\omega_n}{10}$ then $G(j\omega) \approx \frac{b_0}{\omega_n^2}$. Hence the magnitude and phase satisfy $|G(j\omega)|_{dB} \approx |G(0)|_{dB}$ and $\angle G(j\omega) \approx 0^o$ at low frequencies.

- **High Frequency Approximation:** If $\omega \geq 10\omega_n$ then $G(j\omega) \approx \frac{b_0}{(j\omega)^2} = -\frac{b_0 j}{\omega^2}$. Hence the magnitude and phase satisfy $|G(j\omega)|_{dB} \approx \left|\frac{b_0}{\omega^2}\right|_{dB}$ and $\angle G(j\omega) \approx -180^o$ at high frequencies. The magnitude is inversely proportional to ω^2 and hence it rolls off at $-40dB$ per decade similar to a double integrator.

- **Middle Frequency Approximation:** The middle frequency "straight-line" approximation connects the low and high frequency approximations. In the magnitude plot, this corresponds to extending the low and high frequency approximations so they intersect at the corner frequency. This straight line extension yields the approximate gain of $|G(0)|_{dB}$ at the corner frequency $\omega = \omega_n$. The actual gain at the corner frequency can be significantly higher than $|G(0)|_{dB}$ if the system is lightly damped. In the phase plot, the middle frequency approximation connects 0^o of phase at $\omega = \frac{\omega_n}{10}$ to -180^o of phase at $\omega = 10\omega_n$.[‖] This phase approximation is more accurate for highly damped systems. For lightly damped systems, the phase transitions very sharply from 0^o to -180^o.

It is useful to connect these results back to the second order step response response characteristics covered in Section 3.6. Recall that the settling time and rise time are given by $T_s \approx \frac{3}{\zeta\omega_n}$ and $T_r = \frac{\frac{\pi}{2}+\sin^{-1}(\zeta)}{\omega_n\sqrt{1-\zeta^2}}$, Thus increasing ω_n corresponds to decreasing both settling and rise times, i.e. faster response. On a Bode plot, increasing ω_n corresponds to increasing the corner frequency. Thus larger corner frequencies are associated with faster response. Also recall that the peak overshoot is given by $M_p = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}\pi}}$. Decreasing the damping ratio ζ will increase peak overshoot. It will also increase settling time (slower convergence) and reduce rise time (faster initial response). The effect of damping ratio on a Bode plot is mainly observed near the natural frequency. Specifically, small damping ratios yield a large resonant peak on the magnitude plot and a sharp transition on the phase plot.

Figure 5.11 shows the effect of the resonance peak on the sinusoidal response. The left plot shows the Bode plot for $G(s) = \frac{16}{s^2+0.4s+16}$ corresponding to $(b_0, \zeta, \omega_n)(16, 0.05, 4)$. This is the same lightly damped system considered above. It has a peak of $+20dB$ near the natural frequency and the phase transitions sharply from 0^o to -180^o. The right plot shows the response to $u(t) = \sin(\omega t)$ for $\omega = 0.4$ and $8\frac{rad}{sec}$. These frequencies are also marked on the Bode plot. The frequency $\omega = 8\frac{rad}{sec}$ is beyond the corner frequency yielding $|G(j8)| \approx 0.33(= -9.5dB)$ and $\angle G(j8) \approx -180^o$. Thus the input $u(t) = \sin(8t)$ yields an output with amplitude attenuated (scaled down) by 0.33 as shown in the top right subplot. On the other hand, the frequency $\omega 4\frac{rad}{sec}$ is at the corner frequency. This is approximately the resonance frequency yielding a large gain $|G(j4)| \approx 10.0(= +20dB)$ and phase $\angle G(j4) \approx -90^o$. Thus the input $u(t) = \sin(4t)$ yields an output with amplitude amplified (scaled up) by 10.0 as shown in the bottom right subplot. Notice that the two time domain plots have different y-axis limits. The output amplitude for $\omega = 4\frac{rad}{sec}$ is significantly larger than for $\omega = 8\frac{rad}{sec}$.

[‖] The middle frequency phase approximation is a straight line on the log scale. The precise formula is not particularly important but for completeness it is given by: $\angle G(j\omega) \approx -180^o \cdot \log_{10}(\frac{\omega}{0.1\omega_n}) / \log_{10}(\frac{10\omega_n}{0.1\omega_n})$.

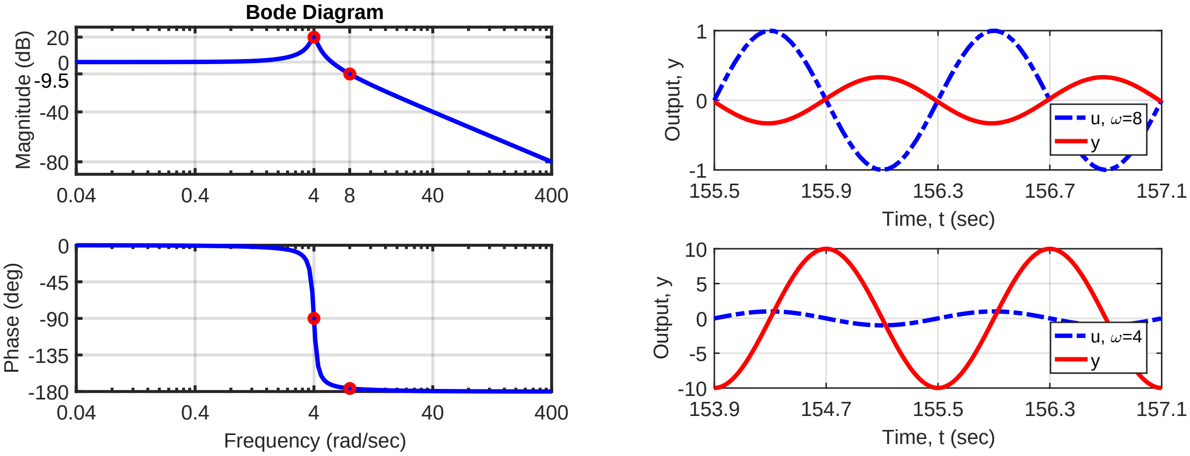


Figure 5.11: Bode plot for $G(s) = \frac{16}{s^2+0.4s+16}$ (left) and frequency responses for $\omega = 0.4$ and $8\,\frac{rad}{sec}$ (right).

There are several additional cases regarding second order systems. These cases are only briefly reviewed here. Second order systems with $b_0 < 0$ and/or complex poles in the right half plane (unstable) have identical magnitude plots to those shown in Figure 5.10. However, the phase plot will be different if $b_0 < 0$ and/or the system has unstable complex poles. The correct phase can be found by studying the transfer function at the corner frequency, low frequency limit, and high frequency limit. Systems can also have second order complex zeros, e.g. $G(s) = \frac{s^2+2\zeta\omega_n s+\omega_n^2}{a_0}$. Such zeros have similar characteristics to the first-order case covered in Section 5.3.2. In the second-order case, the magnitude is roughly flat up to the corner frequency and then rises at $+40dB$ per decade. If the damping is light $\zeta \ll 1$ then the magnitude will have a sharp “notch” (drop in gain) near the corner frequency. Finally, second order overdamped systems with two real poles are discussed in Section 5.5.3.

5.5 Bode Plots: Higher Order Systems

Summary: Consider a system whose transfer function $G(s)$ is expressed as a product of two terms: $G(s) = G_1(s)G_2(s)$. The Bode phase plot of $G(s)$ is simply the sum of the phase plots of $G_1(s)$ and $G_2(s)$. Similarly, the Bode magnitude plot of $G(s)$ (in dB) is simply the sum of the magnitude plots of $G_1(s)$ and $G_2(s)$. This section provides several examples demonstrating the use of this fact to draw Bode plots for higher order systems.

5.5.1 Products of Transfer Functions

Consider a system whose transfer function $G(s)$ is expressed as a product of two terms: $G(s) = G_1(s)G_2(s)$. The response $G(j\omega)$ at a frequency ω is a complex number given by the product of the two complex numbers $G_1(j\omega)$ and $G_2(j\omega)$. Recall from Appendix 3.8.1 (and Example 3.12) that such products can be easily written in polar form. In particular, the product of two complex numbers is obtained by adding their phases and multiplying their magnitudes. Based on this fact, the magnitude and phase of $G(j\omega)$ are given by:

$$|G(j\omega)| = |G_1(j\omega)| \cdot |G_2(j\omega)| \tag{5.26}$$

$$\angle G(j\omega) = \angle G_1(j\omega) + \angle G_2(j\omega) \tag{5.27}$$

Next, recall that $\log_{10}(c_1c_2) = \log_{10}(c_1) + \log_{10}(c_2)$ for any real numbers c_1 and c_2. Thus the magnitude in actual units (Equation 5.26) can be rewritten in dB as follows:

$$|G(j\omega)|_{dB} = |G_1(j\omega)|_{dB} + |G_2(j\omega)|_{dB} \tag{5.28}$$

To summarize, the magnitude (in dB) and phase of $G(s)$ are simply the sum of the magnitudes (in dB) and phases of $G_1(s)$ and $G_2(s)$. **Therefore the Bode phase plot of $G(s)$ is simply the sum of the phase plots of $G_1(s)$ and $G_2(s)$. Similarly, the Bode magnitude plot of $G(s)$ (in dB) is simply the sum of the magnitude plots of $G_1(s)$ and $G_2(s)$.** This is a key property that will be used to design more advanced controllers in stages.

Example 5.4. The transfer function $G(s) = \frac{10}{s}$ can be expressed as the product of an integrator $G_1(s) = \frac{1}{s}$ and a (constant) gain $G_2(s) = 10$. The left side of Figure 5.12 shows the Bode plots for G, G_1, and G_2. As discussed in Section 5.2.3, the integrator has a magnitude slope of $-20dB$ per decade and a phase of -90^o. Moreover, the integrator has a gain of 1 at $\omega = 1\frac{rad}{sec}$, i.e. $|G_1(j1)|_{dB} = 0dB$. The gain $G_2(s) = 10$ has magnitude of $20dB$ and phase of 0^o at all frequencies. The Bode plot for the product $G(s) = G_1(s)G_2(s)$ can be obtained by simply adding the individual plots for $G_1(s)$ and $G_2(s)$. As a result, the magnitude of $G(s)$ is shifted upward by $+20dB$ at all frequencies relative to the integrator. The phase of $G(s)$ is the same as the phase of the integrator because $\angle G_2(j\omega) = 0^o$.

The right side of Figure 5.12 shows a similar example $G(s) = \frac{-0.1}{s}$. $G(s)$ is a product of an integrator $G_1(s) = \frac{1}{s}$ with gain $G_2(s) = -0.1$. In this case, the gain has magnitude of $20\log_{10}|-0.1| = -20dB$ and phase of $+180^o$ at all frequencies. Hence the magnitude of $G(s)$ is shifted downward by $-20dB$ at all frequencies relative to the integrator. Moreover, the phase of $G(s)$ is the sum of the -90^o of phase from the integrator and the $+180^o$ of phase from the gain. Hence $\angle G(j\omega) = +90^o$ at all frequencies.

△

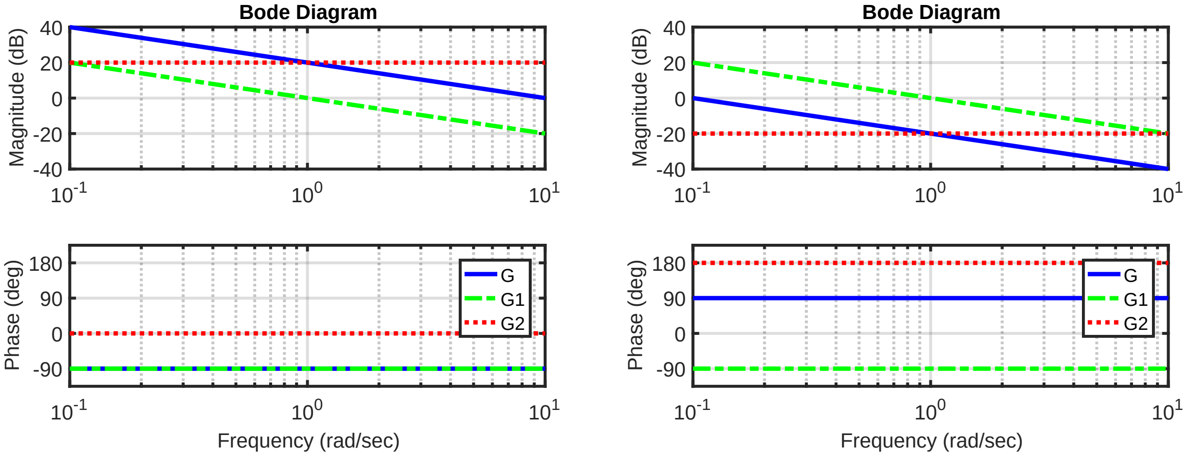


Figure 5.12: Bode plots for $G_1(s)$, $G_2(s)$, and $G(s) = G_1(s)G_2(s)$.
Left: $G_1(s) = \frac{1}{s}$ and $G_2(s) = 10$. Right: $G_1(s) = \frac{1}{s}$ and $G_2(s) = -0.1$.

5.5.2 Lead Controller

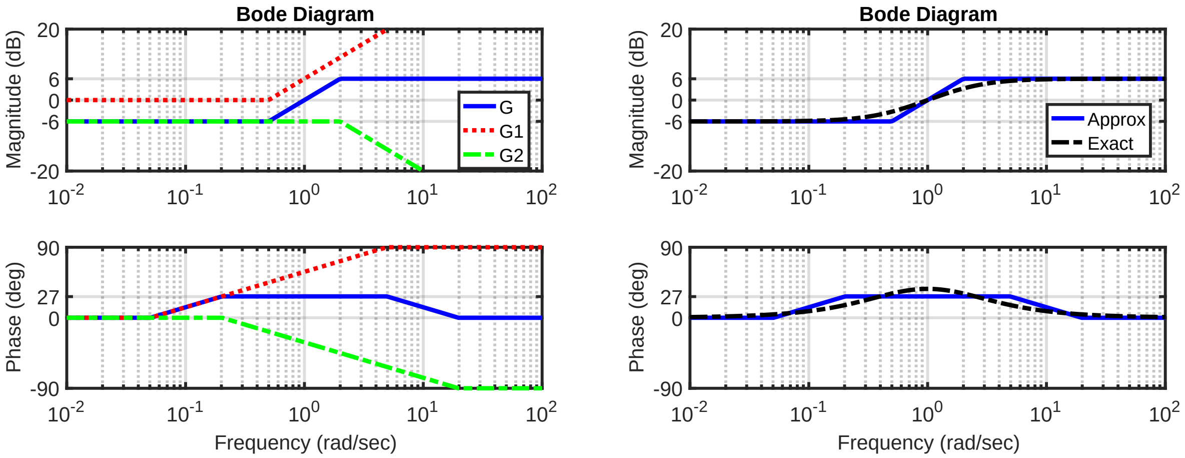
Consider the following first-order system with input $e(t)$ and output $u(t)$:

$$\dot{u}(t) + 2u(t) = 2\dot{e}(t) + e(t) \tag{5.29}$$

The corresponding transfer function is $G(s) = \frac{2s+1}{s+2}$. This is a first-order system with both a pole and a zero. This system is an instance of a lead controller as will be discussed in Section XXX. Note that the transfer function can be expressed as the product $G(s) = G_1(s)G_2(s)$ where $G_1(s) = 2s + 1$ and $G_2(s) = \frac{1}{s+2}$.

The left side of Figure 5.13 shows the straight-line Bode approximations for $G_1(s)$, $G_2(s)$, and $G(s)$. The system $G_1(s)$ has a zero at $s = -0.5$ and a low frequency gain of 1 $(= 0dB)$. Hence the magnitude plot for $G_1(s)$ is $0dB$ at low frequencies and rolls up at $+20dB$ per decade above the corner frequency of $\omega_1 := 0.5\frac{rad}{sec}$. Moreover, the phase plot is 0^o at low frequencies and $+90^o$ at high frequencies. It transitions from 0^o at $0.05\frac{rad}{sec}$ $(= \omega_1/10)$ to 90^o at $5\frac{rad}{sec}$ $(= 10\omega_1)$. Similarly, $G_2(s)$ has a pole at $s = -2$ and a low frequency gain of 0.5 $(= -6dB)$. Hence the magnitude plot for $G_2(s)$ is $-6dB$ at low frequencies and rolls off at $-20dB$ per decade above the corner frequency of $\omega_2 = 2\frac{rad}{sec}$. The phase plot is 0^o at low frequencies and -90^o at high frequencies. It transitions from 0^o at $0.2\frac{rad}{sec}$ $(= \omega_2/10)$ to 90^o at $20\frac{rad}{sec}$ $(= 10\omega_2)$.

The Bode plot for $G(s)$ is obtained by adding the Bode plots for $G_1(s)$ and $G_2(s)$. As a result the magnitude of $G(s)$ is $-6dB$ at low frequencies. For $\omega \in [\omega_1, \omega_2]$, $G(s)$ rises at $+20dB$ per decade due to the zero in $G_1(s)$. However, for $\omega \geq \omega_2$ the slopes of $G_1(s)$ and $G_2(s)$ cancel so that the magnitude of $G(s)$ remains flat at high frequencies. Similar arguments can be used to generate the phase plot for $G(s)$. The right side of Figure 5.13 shows the straight-line approximation and exact Bode plot for $G(s)$. The straight-line approximation generated by adding the Bode plots of $G_1(s)$ and $G_2(s)$ is reasonably accurate. Note that the phase of $G(s)$ is positive and the exact phase has a peak $\geq 30^o$ at a frequency of $1\frac{rad}{sec}$. If this system (Equation 5.29) is excited by a sinusoidal input $e(t) = \sin \omega t$ then the output sinusoid will "lead" the input sinusoid due to the positive phase. This is similar to the lead behavior observed in Figure 5.4 for a differentiator.



Bode Diagram

Bode Diagram

Figure 5.13: Left: Straight-line Bode approximations for $G_1(s) = 2s + 1$, $G_2(s) = \frac{1}{s+2}$ and $G(s) = G_1(s)G_2(s)$. Right: Straight-line approximation and exact Bode for G(s).

5.5.3 Overdamped Second Order System

This section briefly discusses the construction of Bode plots for overdamped second order systems. Such systems have two real poles. For example, consider the following system with input $u(t)$ and output $y(t)$:

$$\ddot{y}(t) + 1.2\dot{y}(t) + 0.2y(t) = 0.5u(t) \tag{5.30}$$

The corresponding transfer function is $G(s) = \frac{0.5}{s^2+1.2s+0.2}$. This system has natural frequency $\omega_n = \sqrt{0.2} \approx 0.45$ and damping $\zeta = \frac{1.2}{2\omega_n} \approx 1.34$. The system is overdamped with poles at $s = -1$ and $s = -0.2$. The transfer function can be expressed as the product $G(s) = G_1(s)G_2(s)$ where $G_1(s) = \frac{1}{s+0.2}$ and $G_2(s) = \frac{0.5}{s+1}$.

The left side of Figure 5.14 shows the straight-line Bode approximations for $G_1(s)$, $G_2(s)$, and $G(s)$. The system $G_1(s)$ has a pole at $s = -0.2$ and a low frequency gain of 5 ($= 14dB$). Hence the magnitude plot for $G_1(s)$ is $14dB$ at low frequencies and rolls off at $-20dB$ per decade above the corner frequency of $\omega_1 := 0.2\frac{rad}{sec}$. Moreover, the phase plot transitions from 0^o at low frequencies to -90^o at high frequencies passing through -45^o at the corner frequency ω_1. Similarly, $G_2(s)$ has a pole at $s = -1$ and a low frequency gain of 0.5 ($= -6dB$). Hence the magnitude plot for $G_2(s)$ is $-6dB$ at low frequencies and rolls off at $-20dB$ per decade above the corner frequency of $\omega_2 = 1\frac{rad}{sec}$. The phase plot transitions from 0^o at low frequencies to -90^o at high frequencies passing through -45^o at the corner frequency ω_2.

The Bode plot for $G(s)$ is obtained by adding the Bode plots for $G_1(s)$ and $G_2(s)$. As a result the magnitude of $G(s)$ is $8dB$ ($= 14dB - 6dB$) at low frequencies. For $\omega \in [\omega_1, \omega_2]$, $G(s)$ falls at $-20dB$ per decade due to the pole in $G_1(s)$. However, for $\omega \geq \omega_2$ the slopes of $G_1(s)$ and $G_2(s)$ add so that the magnitude of $G(s)$ falls at $-40dB$ per decade at high frequencies. Similar arguments can be used to generate the phase plot for $G(s)$. The right side of Figure 5.13 shows the straight-line approximation and exact Bode plot for $G(s)$. The straight-line approximation generated by adding the Bode plots of $G_1(s)$ and $G_2(s)$ is reasonably accurate.

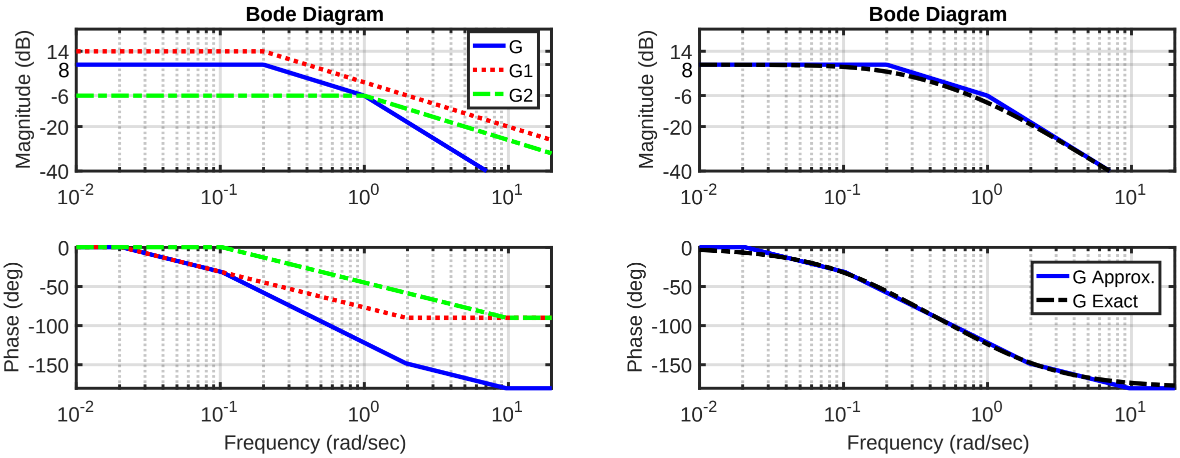


Figure 5.14: Left: Straight-line Bode approximations for $G_1(s) = \frac{1}{s+0.2}$, $G_2(s) = \frac{0.5}{s+1}$ and $G(s) = G_1(s)G_2(s) = \frac{0.5}{s^2+1.2s+0.2}$. Right: Straight-line approximation and exact Bode for $G(s)$.

5.6 Frequency Content of Signals

Summary: The steady-state sinusoidal response for a stable system can be computed using the transfer function. More generally, if the input is a sum of sinusoids then the steady state response is given by summing the responses due to each input sinusoid. This result is powerful because general signals can be expressed as a sum of sinusoids using the Fourier Series. Signals can then be roughly classified as "low frequency" and "high frequency" based on the Fourier Series coefficients. The Bode plot can then be used to gain intuition for how the system will respond to more general low and high frequency signals.

5.6.1 Low and High Frequency Signals

The information in a Bode magnitude and phase plot can be used to compute the steady state response due to a sinusoidal input. For example, consider the linear system with transfer function $G(s) = \frac{10}{s+10}$. This first order system has a pole at $s_1 = -10$ and time constant $\tau_1 = 0.1sec$. The Bode plot for this system is shown on the left of Figure 5.15. The Bode magnitude plot for this system has the typical first order characteristics with DC gain $G(0) = 1$, corner frequency of $10\frac{rad}{sec}$, and high frequency roll off of -20 dB per decade. Thus if $u(t) = \sin(\omega t)$ and $\omega \ll 10\frac{rad}{sec}$ then the steady state response will be $y_{ss}(t) \approx u(t)$. Similarly, if $u(t) = \sin(\omega t)$ and $\omega \gg 10\frac{rad}{sec}$ then the steady state response will be $y_{ss}(t) \approx 0$.

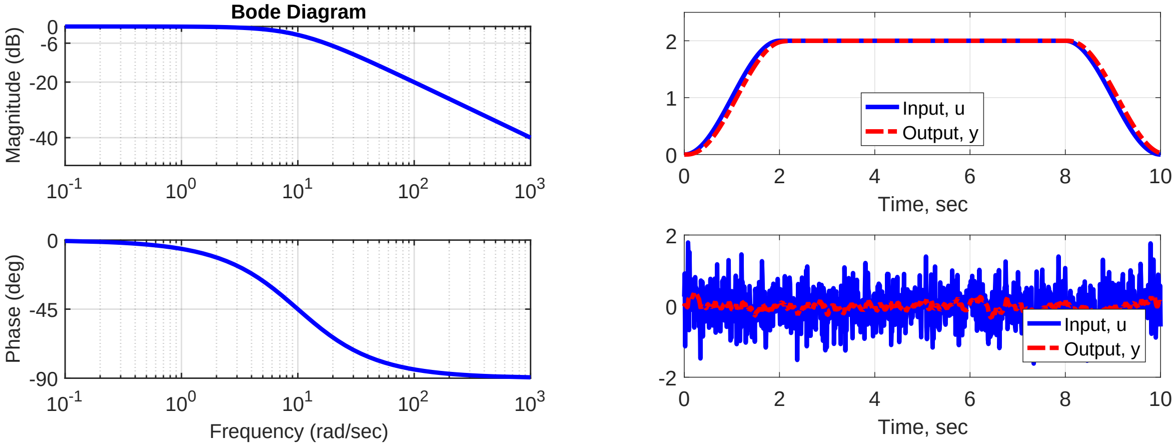


Figure 5.15: Left: Bode plot for $G(s) = \frac{10}{s+10}$. Right: Response due to a "low frequency" signal (top) and "high frequency" input (bottom).

The Bode plot can also be used to gain intuition for how the system will respond to more general signals. The right side of Figure 5.15 shows the response of this system due to two different input signals. In particular, the top subplot shows the response for the input:

$$u(t) = \begin{cases} 1 - \cos(0.5\pi t) & t < 2sec \\ 2 & t \in [2,8)sec \\ 1 + \cos\left(0.5\pi(t-8)\right) & t \in [8,10) \\ 0 & t \geq 10 \end{cases} \tag{5.31}$$

This input signal is slow relative to the system time constant / corner frequency. Thus the steady-state output is mainly affected by the low frequency characteristics of the system: $|G(j\omega)| \approx 1$ and phase $\angle G(j\omega) \approx 0$ from the Bode plot. As a result, the output closely follows the input signal with only a small deviation. In summary, the system $G(s)$ has negligible effect on the magnitude and phase of "low frequency" signals.

The bottom subplot on the right of Figure 5.15 shows the response for an input generated by random (unit variance, Gaussian) noise. In this case the input signal changes rapidly. Thus the steady-state output is mainly affected by the high frequency characteristics of the system: $|G(j\omega)| \approx 0$ for $\omega \gg 10\frac{rad}{sec}$. As a result, the output of $G(s)$ is not able to follow this rapidly changing input. Hence the output signal is attenuated, i.e. its magnitude is significantly reduced, relative to the input. In summary, the system $G(s)$ significantly attenuates such "high frequency" signals.

This rough terminology of "low frequency" vs. "high frequency" will often be used in the next few chapters of this text. This section provides a more rigorous foundation for this terminology.

5.6.2 Superposition for Sinusoidal Inputs

Section 5.1 discussed the steady-state sinusoidal response for a stable system described by a transfer function $G(s)$. A general version of this result is as follows: If the input is $u(t) = A\cos(\omega t + \theta)$ for some numbers (A, θ) then the output response satisfies

$$y(t) \to A|G(j\omega)| \cos\left(\omega t + \theta + \angle G(j\omega)\right) \text{ as } t \to \infty \tag{5.32}$$

This result is more powerful when extended to inputs that are sums of sinusoids. Specifically, consider an input given by the sum of N_c cosine terms:

$$u(t) = \sum_{k=1}^{N_c} A_k \cos(\omega_k t + \theta_k) \tag{5.33}$$

where $\{\omega_k\}_{k=1}^{N_c}$, $\{A_k\}_{k=1}^{N_c}$, and $\{\theta_k\}_{k=1}^{N_c}$ describe the frequency, amplitude, and phase of each cosine term. The general solution of the system $G(s)$ forced by this input is of the form $y(t) = y_P(t) + \sum_{i=1}^{n} c_i e^{s_i t}$ where $\{s_1, \ldots, s_n\}$ are the roots of the characteristic equation. The system is assumed to be stable and hence $\sum_{i=1}^{n} c_i e^{s_i t} \to 0$ as $t \to \infty$ with convergence dominated by the slowest root. Thus stability implies $y(t) \to y_P(t)$ where a particular solution, extending the procedure in Section 5.1, is given by:

$$y_P(t) = \sum_{k=1}^{N_c} A_k |G(j\omega_k)| \cos\left(\omega_k t + \theta_k + \angle G(j\omega_k)\right) \tag{5.34}$$

In other words, if the input is a sum of sinusoids (Equation 5.33) then the steady state response (Equation 5.34) is given by summing the responses due to each input sinusoid. The next example demonstrates this fact.

Example 5.5. Consider the system $G(s) = \frac{10}{s+10}$ forced by the following input with zero IC:

$$u(t) = 1.2 + 0.9\cos\left(10t + \frac{\pi}{2}\right) + 0.9\cos\left(100t + \frac{\pi}{6}\right) \tag{5.35}$$

This input signal is shown on the left of Figure 5.16. The first (constant) term can be interpreted as a cosine at zero (DC) frequency, i.e. $(\omega_1, A_1, \theta_1) = (0, 1.2, 0)$. The remaining terms can be interpreted as low frequency ($\omega_2 = 10$) and high frequency ($\omega_3 = 100$) components.

The Bode magnitude plot for the system $G(s)$ has the typical first order characteristics (see right plot in Figure 5.6) with DC gain $G(0) = 1$, corner frequency of $10\frac{rad}{sec}$, and high frequency roll off of -20 dB per decade. The high frequency component (third term) in the input is well above the corner frequency. Thus it is expected that this term should be attenuated (magnitude reduced) in the steady-state output of $G(s)$. To verify this expectation, first evaluate the transfer function at the various frequencies $\{\omega_k\}_{k=1}^{3}$:

$$G(0) = 1 \qquad G(10) = 0.5 - 0.5j = 0.707e^{-j\frac{\pi}{4}} \qquad G(100) = 0.01 - 0.1j = 0.1e^{-j1.47} \tag{5.36}$$

Thus the steady-state solution (Equation 5.34) as $t \to \infty$ is:

$$y(t) \to 1.2 + 0.64\cos\left(10t + \frac{\pi}{4}\right) + 0.09\cos\left(100t - 0.95\right) \tag{5.37}$$

The right side of Figure 5.16 shows this steady state response (blue dashed) along with the actual system output (red solid). The system $G(s)$ has a single pole at $s_1 = -10$ with corresponding time constant of $\tau_1 = 0.1sec$. Hence the actual response converges to the steady state response after the settling time $3\tau_1 = 0.3sec$. As expected the high frequency component ($\omega_3 = 100$) is attenuated in the output, i.e. the amplitude of this term is significantly reduced. As a consequence, the output is dominated by the constant and low frequency terms.

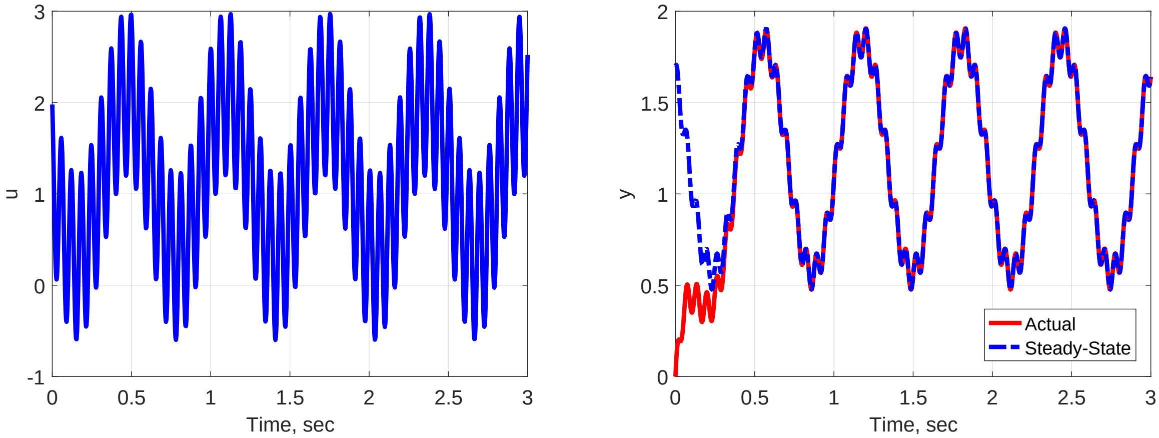


Figure 5.16: Left: Signal $u(t)$ consisting of three cosine terms (Equation 5.35). Right: Output $y(t)$ of linear system $G(s) = \frac{10}{s+10}$ due to input $u(t)$ (Equation 5.35) and zero IC. Both actual response and steady-state sinusoidal response are shown.

△

5.6.3 Fourier Series

The previous section demonstrated that if the input is a sum of sinusoids then the steady state output is simply the sum of the response due to each input term. The result is powerful because general signals can be decomposed into a sum of sinusoidal components. In particular, a (real) input signal $u(t)$ defined on a finite time interval $[0, T]$ can be expressed as a Fourier Series, i.e. as a sum of complex exponentials:

$$\begin{aligned} u(t) &= \sum_{k=-\infty}^{\infty} U_k e^{j\omega_k t} \\ \text{where } U_k &:= \frac{1}{T}\int_0^T u(t) e^{-j\omega_k t} \text{ and } \omega_k := \frac{2\pi k}{T} \end{aligned} \tag{5.38}$$

U_k is the k^{th} (complex) Fourier coefficient and ω_k is the k^{th} harmonic frequency. This complex expansion can be equivalently rewritten with (real) cosine terms:

$$\begin{aligned} u(t) &= \sum_{k=0}^{\infty} A_k \cos(\omega_k t + \theta_k) \\ \text{where } A_k &:= |U_k| \text{ and } \theta_k := \angle U_k \end{aligned} \tag{5.39}$$

There are many technical details omitted in this brief summary of the Fourier Series, e.g. convergence of the infinite sum of sinusoids. Such details on the Fourier Series (and related transforms) are beyond the scope of this course but can be found in textbooks on signal processing, e.g. [19].

The Fourier Series and sinusoidal superposition can be combined to compute the steady state response for any input signal $u(t)$ defined on $[0, T]$:[**]

1. Express $u(t)$ as an infinite sum of cosine terms using the Fourier Series (Equation 5.39).
2. Compute the steady-state output due to the k^{th} input term:

$$A_k |G(j\omega_k)| \cos\left(\omega_k t + \theta_k + \angle G(j\omega_k)\right) \tag{5.40}$$

3. By superposition, the total steady-state response as $t \to \infty$ is given by the sum of the individual responses:

$$y(t) \to \sum_{k=1}^{\infty} A_k |G(j\omega_k)| \cos\left(\omega_k t + \theta_k + \angle G(j\omega_k)\right) \tag{5.41}$$

This procedure will give the correct output response in steady-state after the effect of the initial condition has decayed to zero. We won't use this formal procedure in the remainder of the

[**] These steps assume T is "large" so that the output has time to converge to the steady-state response. Here "large" simply means that T is long relative to the time constants of the system poles so that the homogeneous term has time to decay, $\sum_{i=1}^{n} c_i e^{s_i t} \to 0$

text. However, it does motivate the use of informal terms such as "low frequency" and "high frequency" signals. It also provides additional intuition for how linear systems will respond to general low and high frequency signals.

As an example, recall the responses for the system $G(s) = \frac{10}{s+10}$ shown in Figure 5.15. This figure includes the response due to the "low frequency" input in Equation 5.31. The "low frequency" nature of this signal can now be made precise. Specifically, the top left plot of Figure 5.17 shows the Fourier Series amplitudes for this signal. The amplitudes are essentially zero above $5\frac{rad}{sec}$. As a consequence, the linear system $G(s) = \frac{10}{s+10}$, which satisfies $G(j\omega) \approx 1$ below its corner frequency at $10\frac{rad}{sec}$, has minimal effect on this input signal. The bottom left plot of Figure 5.17 show the Fourier Series amplitudes for the output response due to this input. As expected, the output amplitudes are essentially unchanged relative to the input. As a result, the output signal $y(t)$ closely matches in the input signal $u(t)$ as shown in Figure 5.15.

As another example, Figure 5.15 shows the response of $G(s) = \frac{10}{s+10}$ due to a "high frequency" random input. The top right plot of Figure 5.17 shows the Fourier Series amplitudes for this random signal. Note that the amplitudes are relatively flat all the way beyond $300\frac{rad}{sec}$. Thus this random signal has significant high frequency content. The amplitudes at high frequencies are attenuated by $G(s) = \frac{10}{s+10}$ because its magnitude rolls off by $-20dB$ per decade above its corner frequency at $10\frac{rad}{sec}$. The bottom right plot of Figure 5.17 show the Fourier Series amplitudes for the output response due to this input. As expected, the output amplitudes are significantly reduced at high frequencies. As a result, the output signal $y(t)$ is significantly attenuated relative to the input signal $u(t)$ as shown in Figure 5.15.

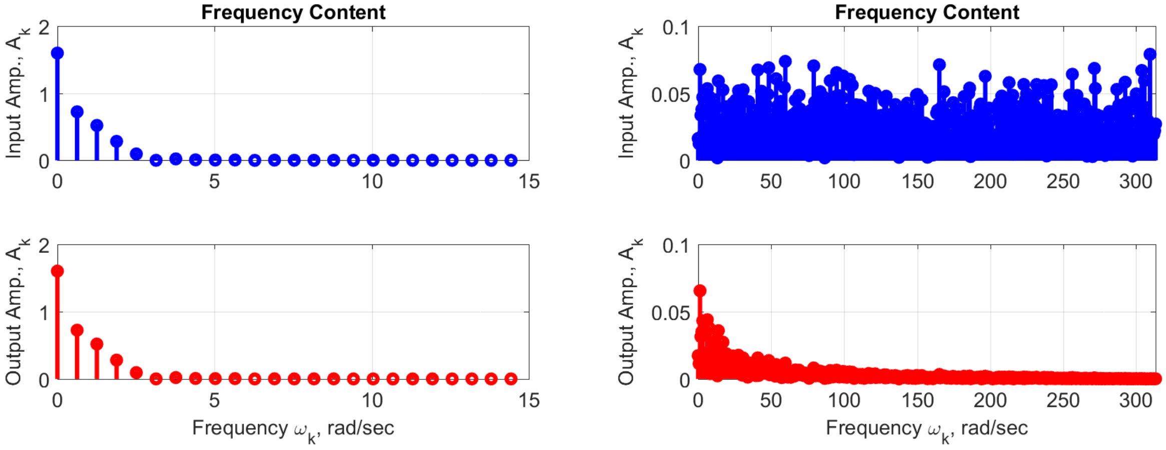


Figure 5.17: Fourier Series amplitudes for input and output with low frequency input (left) and high frequency input (right).

5.7 System Identification

Summary: A model for a system can be constructed using experimental input/output data. This section describes a frequency domain (black-box) approach to system identification. The steps of this approach are to: i) Force the system with a sinusoid at frequency ω_k, ii) Measure the amplitude and phase of the system output in steady-state, iii) Use this experimental data to estimate the system frequency response $G(j\omega_k)$ at ω_k, iv) Repeat steps i)-iii) at a variety of frequencies and use the frequency response points to estimate an ODE/transfer function model. This basic approach can be time consuming as it requires running many experiments, i.e. Steps i)-iii). Less time-consuming methods for system identification will be briefly discussed.

See notes.

5.8 Appendix: Resonance

Summary: This appendix derives the peak gain and resonance frequency for an underdamped second order system.

The transfer function for a stable, underdamped second-order system is $G(s) = \frac{b_0}{s^2+2\zeta\omega_n s+\omega_n^2}$. The precise frequency where this system achieves its peak gain can be determined via calculus. Briefly, the magnitude (in actual units) at an arbitrary frequency is given by:

$$|G(j\omega)| = \frac{|b_0|}{\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2}} \tag{5.42}$$

The magnitude achieves its peak value at a frequency where $\frac{d}{d\omega}|G(j\omega)| = 0$. Differentiating Equation 5.42 with respect to frequency yields:

$$\frac{d}{d\omega}|G(j\omega)| = \frac{-0.5|b_0|\ (-4\omega(\omega_n^2 - \omega^2) + 8\zeta^2\omega_n^2\omega)}{\left((\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2\right)^{1.5}} \tag{5.43}$$

This derivative is equal to zero if and only if:

$$0 = \omega^3 + (2\zeta^2 - 1)\omega_n^2\,\omega \tag{5.44}$$

There are two cases depending on the value of ζ:

- $2\zeta^2 - 1 \geq 0$: In this case $\zeta \geq \frac{1}{\sqrt{2}}$. The peak frequency is $\omega_p = 0$ and the peak gain is $|G(0)|$. This corresponds to a heavily damped system for which no resonance occurs.
- $2\zeta^2 - 1 < 0$: In this case $\zeta < \frac{1}{\sqrt{2}}$. The peak frequency is $\omega_p = \omega_n\sqrt{1-2\zeta^2}$ and the peak gain is $|G(j\omega_p)| = \frac{|G(0)|}{2\zeta\sqrt{1-\zeta^2}}$. This corresponds to a lightly damped system with resonance at frequency ω_p.

Chapter 6

Frequency Domain Control Design

This chapter introduces loopshaping for control design. This will allow us to design more advanced controllers that achieve better performance tradeoffs than possible with PID. The chapter first uses the transfer function notation to derive models for different system interconnections including serial, parallel and feedback connections. This enables the ODEs for closed-loop systems to be derived using easy algebraic manipulations. The stability of the feedback system is then discussed and conditions for stability are given. Next, the performance conditions for the feedback system are specified in the frequency domain. Most control design requirements can be specified on either the sensitivity function $S(s) = \frac{1}{1+G(s)K(s)}$ or complementary sensitivity function $T(s) = \frac{G(s)K(s)}{1+G(s)K(s)}$. Loopshaping is a design procedure that focuses on the loop transfer function $L(s) = G(s)K(s)$. First, the requirements on $S(s)$ and $T(s)$ are translated into related requirements on the loop $L(s)$. Next the controller is designed in steps to meet the requirements on the loop $L(s)$. Each step uses a specific controller component to improve the loop characteristics in the low, middle, or high frequencies. The remainder of the chapter discusses the effects of various controller components and provides an example of loopshaping design.

6.1 Interconnections of Systems

Summary: This section uses the transfer function notation to derive models for different system interconnections. The parallel interconnection of $G_1(s)$ and $G_2(s)$ is given by $H(s) = G_1(s) + G_2(s)$. The serial interconnection of $G_1(s)$ followed by $G_2(s)$ is given by $H(s) = G_2(s) \cdot G_1(s)$. Finally, the negative feedback interconnection of $G_1(s)$ and $G_2(s)$ is given by $H(s) = \frac{G_1(s)}{1+G_1(s)G_2(s)}$. Some care is required with this final expression because it results in a non-minimal representation. Instead the Matlab command `feedback` should be used to correctly compute the (minimal) transfer function for feedback interconnections.

6.1.1 Revisiting the Transfer Function Notation

A model for a feedback system can be obtained by combining models for the plant and controller. This is relatively easy for first/second order plants and simple PID controllers. However, the manipulations become cumbersome for higher-order plants and more complex controllers. Transfer functions can be used to simplify the algebraic manipulations for such derivations. To demonstrate these manipulations, first consider a general n^{th} order linear ODE:

$$a_n y^{[n]}(t) + \cdots + a_1 \dot{y}(t) + a_0 y(t) = b_m u^{[m]}(t) + \cdots + b_1 \dot{u}(t) + b_0 u(t) \tag{6.1}$$

The corresponding transfer function is:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{6.2}$$

The transfer function was first introduced as simply another notation for the ODE. In particular, the numerator and denominator polynomials provide the coefficients for the right (input) and left (output) sides of the ODE. The k^{th} power of s corresponds, notationally, to the k^{th} derivative with respect to time. It will be useful to further extend this notation. Let $N(s)$ and $D(s)$ denote the polynomials in the numerator and denominator of the transfer function: $G(s) = \frac{N(s)}{D(s)}$. The linear ODE can then be denoted as:

$$\underbrace{(a_n s^n + \cdots + a_1 s + a_0)}_{:=D(s)} Y(s) = \underbrace{(b_m s^m + \cdots + b_1 s + b_0)}_{:=N(s)} U(s) \tag{6.3}$$

The equation $D(s)Y(s) = N(s)U(s)$ is simply a compact notation for the linear ODE (Equation 6.1) that relates input u to output y.* It represents the derivatives in the ODE by polynomials in s. A useful feature of this notation is that algebraic manipulations involving differential equations can be equivalently performed using polynomials. Two simple examples are provided below to demonstrate these manipulations. Further details (including proofs using linear differential operators) can be found in [7].

*Again, the Laplace Transform (Appendix 2.4.2) can be used to derive a more formal connection. For example, if $Y(s)$ is the Laplace transform of $y(t)$ then the Laplace transform of $y^{[k]}(t)$ is $s^k Y(s)$ (assuming zero ICs). This fact can be used to derive Equation 6.3 by taking the Laplace Transform of the ODE in Equation 6.1. However, the formal connection provided by the Laplace Transform is not needed here.

Example 6.1. Consider the following two relations from $u(t)$ to $y_1(t)$ and $y_2(t)$:

$$y_1(t) = \dot{u}(t) + 4u(t) \quad \text{and} \quad y_2(t) = 2\ddot{u}(t) - 5.7\dot{u}(t) - u(t) \tag{6.4}$$

The sum $y(t) = y_1(t) + y_2(t)$ is thus given by $y(t) = 2\ddot{u}(t) - 4.7\dot{u}(t) + 3u(t)$. This can also be derived using the polynomial representation for differential equations. In particular, the relations from input $u(t)$ to outputs $y_1(t)$ and $y_2(t)$ are denoted by:

$$Y_1(s) = (s+4)\,U(s) \quad \text{and} \quad Y_2(s) = (2s^2 - 5.7s - 1)\,U(s) \tag{6.5}$$

Summing the polynomials yields $Y(s) = Y_1(s) + Y_2(s) = (2s^2 - 4.7s - 1)\,U(s)$. This denotes $y(t) = 2\ddot{u}(t) - 4.7\dot{u}(t) + 3u(t)$ as derived above. △

Example 6.2. Consider the following two relations from $u(t)$ to $w(t)$ and from $w(t)$ to $y(t)$:

$$w(t) = 3\dot{u}(t) - 7u(t) \quad \text{and} \quad y(t) = -5\dot{w}(t) + w(t) \tag{6.6}$$

A model from $u(t)$ to $y(t)$ can be obtained by substituting $w(t)$ from the first relation into the second relation. This yields $y(t) = -15\ddot{u}(t) + 38\dot{u}(t) - 7u(t)$. This can also be derived using the polynomial representation for the differential equations. In particular, the relations from $u(t)$ to $w(t)$ and from $w(t)$ to $y(t)$ are denoted by:

$$W(s) = (3s-7)\,U(s) \quad \text{and} \quad Y(s) = (-5s+1)\,W(s) \tag{6.7}$$

Thus $Y(s) = (-5s+1)(3s-7)\,U(s) = (-15s^2 + 38s - 7)\,U(s)$. This denotes $y(t) = -15\ddot{u}(t) + 38\dot{u}(t) - 7u(t)$ as derived above. △

In the first example a sum of two ODEs corresponds to a sum of their polynomial representations. In the second example a concatentation of one ODE followed by another ODE corresponds to a multiplication of their polynomial representations. These examples are sufficiently simple that they don't necessarily demonstrate the utility of the polynomial representation. For example, only non-proper systems appear in both examples above and no derivatives of the output $y(t)$ appear in the final result. The remainder of the section derives more general results involving interconnections of systems described by transfer functions.

6.1.2 Parallel Interconnection

Consider the parallel interconnection of two subsystems $G_1(s)$ and $G_2(s)$ as shown in Figure 6.1. The outputs of the subsystems are summed to obtain $y(t) = y_1(t) + y_2(t)$. **The parallel interconnection is modeled by the transfer function $H(s) = G_1(s)+G_2(s)$.** An informal sketch of this result is:

$$Y(s) = Y_1(s) + Y_2(s) = (G_1(s) + G_2(s))\; U(s) \tag{6.8}$$

This result is formally shown in Appendix 6.7.1 using the polynomial representation for ODEs. Parallel interconnections can be computed in Matlab using + as shown in the example below.

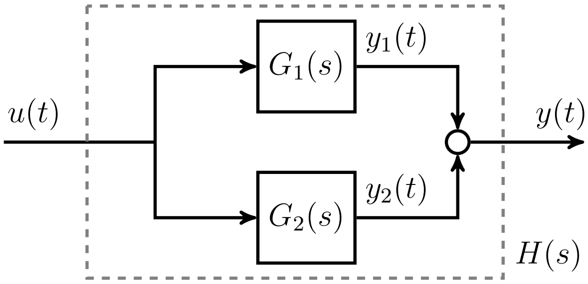


Figure 6.1: Parallel Interconnection of G_1 and G_2.

Example 6.3. Consider the systems $G_1(s) = \frac{3}{s+4}$ and $G_2(s) = \frac{5}{9s-6}$. The parallel interconnection $H(s) = G_1(s) + G_2(s)$ is given by:

$$H(s) = \frac{3}{s+4} + \frac{5}{9s-6} = \frac{3(9s-6)+5(s+4)}{(s+4)(9s-6)} = \frac{32s+2}{9s^2+30s-24}$$

This transfer function denotes the following ODE from input $u(t)$ to output $y(t)$:

$$9\ddot{y}(t) + 30\dot{y}(t) - 24y(t) = 32\dot{u}(t) + 2u(t) \tag{6.9}$$

The same result can be obtained in Matlab:

```
>> G1 = tf(3,[1 4]);
>> G2 = tf(5,[9 -6]);
>> H = G1+G2
   H =
        32 s + 2
    -----------------
    9 s^2 + 30 s - 24
```

Deriving the ODE for this parallel interconnection would have been more complicated without the polynomial representation. △

6.1.3 Serial (Cascade) Interconnection

Consider the serial (cascade) interconnection of two subsystems $G_1(s)$ and $G_2(s)$ as shown in Figure 6.2. The output of $G_1(s)$ is connected as the input to $G_2(s)$. **The serial interconnection is modeled by the transfer function $H(s) = G_2(s) \cdot G_1(s)$.** An informal sketch of this result is:

$$Y(s) = G_2(s)\,W(s) = G_2(s)G_1(s)\,U(s) \tag{6.10}$$

This result is formally shown in Appendix 6.7.2 using the polynomial representation for ODEs. Serial interconnections can be computed in Matlab using `*` as shown in the example below.

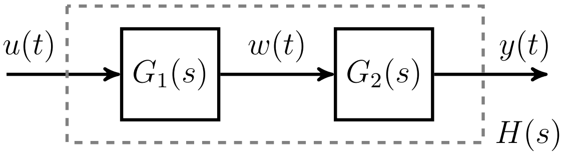


Figure 6.2: Cascade (Serial) Interconnection of G_1 and G_2.

Example 6.4. Consider the systems $G_1(s) = \frac{3}{s+4}$ and $G_2(s) = \frac{5}{9s-6}$. The serial interconnection $H(s) = G_2(s) \cdot G_1(s)$ is given by:

$$H(s) = \frac{5}{9s-6} \cdot \frac{3}{s+4} = \frac{15}{9s^2+30s-24}$$

This transfer function denotes the following ODE from input $u(t)$ to output $y(t)$:

$$9\ddot{y}(t) + 30\dot{y}(t) - 24y(t) = 15u(t) \tag{6.11}$$

The same result can be obtained in Matlab:

```
>> H = G2*G1
   H =
           15
    ----------------
    9 s^2 + 30 s - 24
```

△

6.1.4 Feedback Interconnection

Consider the negative feedback interconnection of two subsystems $G_1(s)$ and $G_2(s)$ as shown in Figure 6.3. $G_1(s)$ is said to be in the *forward path* and $G_2(s)$ is in the *feedback path.* This is called *negative feedback* because the output of $G_2(s)$ enters the summing junction with a negative sign so that $e(t) = r(t) - w(t)$. *Positive feedback* refers to the case where $e(t) = r(t) + w(t)$.

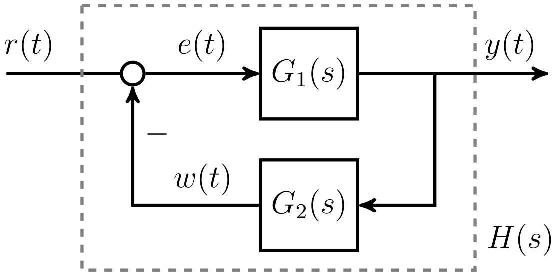


Figure 6.3: Negative Feedback Interconnection of G_1 and G_2.

The transfer function relations in Figure 6.3 are

$$Y(s) = G_1(s)\,(R(s) - W(s)) \tag{6.12}$$

$$W(s) = G_2(s)Y(s) \tag{6.13}$$

Eliminating $W(s)$ and solving for $Y(s)$ yields:

$$Y(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)} R(s) \tag{6.14}$$

Based on this informal sketch, the negative feedback interconnection is modeled by the transfer function $H(s) = \frac{G_1(s)}{1+G_1(s)G_2(s)}$. A similar derivation for positive feedback interconnections yields $H(s) = \frac{G_1(s)}{1-G_1(s)G_2(s)}$. Some care is required with these expressions because they are non-minimal as defined in Section 3.3.2, i.e. they contain common poles and zeros at the same location. In particular, a more formal derivation for the negative feedback case is given in Appendix 6.7.3 using the polynomial representation for ODEs. The result (Equation 6.48) involves a more complicated expression for $H(s)$ in terms of the numerator and denominator polynomials of $G_1(s)$ and $G_2(s)$. In fact, the transfer function obtained from Equation 6.14 has common poles and zeros at the roots of $D_1(s) = 0$ where $D_1(s)$ is the denominator polynomial of $G_1(s)$. These common poles and zeros must be cancelled to obtain the correct (minimal) expression for $H(s)$. **As a result, negative feedback interconnections should not be computed in Matlab using the expression `G1/(1+G1*G2)`. Instead, they should be computed using the syntax `H=feedback(G1,G2)`.** The simpler expression $H(s) = \frac{G_1(s)}{1+G_1(s)G_2(s)}$ will often be used with the understanding that this represents the minimal representation after common poles/zeros have been removed. The example below demonstrates these issues.

Example 6.5. Consider the systems $G_1(s) = \frac{3}{s+4}$ and $G_2(s) = \frac{5}{9s-6}$. The negative feedback interconnection can be obtained in Matlab:

```
>> H = feedback(G1,G2)
   H =
       27 s - 18
    ----------------
    9 s^2 + 30 s - 9
```

This transfer function denotes the following ODE from input $u(t)$ to output $y(t)$:

$$9\ddot{y}(t) + 30\dot{y}(t) - 9y(t) = 27\dot{u}(t) - 18u(t) \tag{6.15}$$

The formal expression (Equation 6.48) in the appendix is $H(s) = \frac{D_2(s)N_1(s)}{D_2(s)D_1(s)+N_2(s)N_1(s)}$. This yields the same transfer function:

$$H(s) = \frac{(9s-6)\cdot 3}{(s+4)(9s-6) + 5\cdot 3} = \frac{27s - 18}{9s^2 + 30s - 9}$$

Finally, $H(s) = \frac{G_1(s)}{1+G_1(s)G_2(s)}$ yields the same result but with additional poles and zeros at the roots of $D_1(s) = 0$ where $D_1(s) = (s+4)$ is the denominator of $G_1(s)$. Specifically, this form for $H(s)$ introduces an additional pole and zero at $s = -4$. The Matlab commands below demonstrate this behavior:

```
>> H2 = G1/(1+G1*G2)
   H2 =
        27 s^2 + 90 s - 72
    ---------------------------
    9 s^3 + 66 s^2 + 111 s - 36
>> pole(H2).'
   ans =
     -4.0000   -3.6103    0.2770
>> zero(H2).'
 ans =
     -4.0000    0.6667
```

Again, this form is non-minimal due to the common pole and zero at $s = -4$. The `minreal` function can be used cancel the common pole and zero to obtain a minimal representation:

```
>> minreal(H2)
   ans =

        3 s - 2
   -----------------
   s^2 + 3.333 s - 1
```

This transfer function represents the same system as derived above. Multiply the numerator and denominator by 9 to show this equivalence. **This example emphasizes that feedback interconnections should be computed in Matlab using the `feedback` function.** △

6.2 Stability of Feedback Systems

Summary: This section considers a generic feedback system with many inputs and outputs. Two special transfer functions associated with feedback system are the sensitivity $S(s) = \frac{1}{1+G(s)K(s)}$ and complementary sensitivity $T(s) = \frac{G(s)K(s)}{1+G(s)K(s)}$. The transfer functions $S(s)$ and $T(s)$ are "complementary" in the sense that $S(s) + T(s) = 1$ for all $s \in \mathbb{C}$. Next, the feedback system is defined to be stable if every individual input/output transfer function is stable. It is shown that if the controller $K(s)$ is designed to cancel a RHP pole or zero in the plant $G(s)$ then the feedback system will be unstable. Hence the controller should never be designed to have such cancellations. Assuming such a cancellation is avoided, then the feedback system will be stable if and only if $1 + G(s)K(s)$ has no zeros in the RHP.

6.2.1 General Feedback System

Chapter 4 presented techniques to design PID controllers for first and second order systems. The controller is designed to ensure stability, track reference commands, reject disturbances and be insensitive to sensor noise. Figure 6.4 shows a general feedback system that includes the signals required to model these issues. The systems include the plant $G(s)$ and controller $K(s)$. The signals include the reference command r, tracking error e, control command u, input disturbance d, plant input v, plant output y, sensor noise n, and measured plant output m. Most applications will have competing design requirements related to the various signals shown in this generic feedback diagram. The frequency domain procedure, introduced later in this chapter, will enable better trade-offs in the control design.

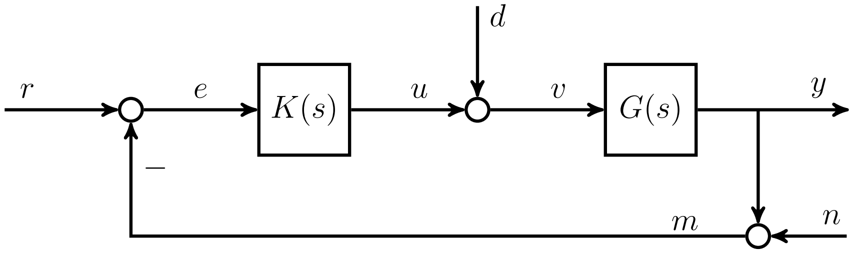


Figure 6.4: General feedback system

The first step is to notice that there are many different inputs and outputs in Figure 6.4. Each input / output pair has an associated transfer function that models the dynamics from the chosen input to output. For example, the transfer function from input r to output y, denoted by $T_{r\to y}(s)$, can be computed by setting $d = 0$ and $n = 0$ in Figure 6.4. In this case $Y(s) = G(s)K(s)\,(R(s) - Y(s))$ and solving for $Y(s)$ yields $T_{r\to y}(s) = \frac{G(s)K(s)}{1+G(s)K(s)}$. This result is a special case of the feedback interconnection discussed in Section 6.1.4 with $G(s)K(s)$ in the forward path and unity feedback path. As discussed in Section 6.1.4, this should be interpreted as the minimal transfer function obtained after canceling any common poles and zeros.

The feedback diagram contains three input signals (r, d, n) and five "internal" signals that can be selected as outputs (e, u, v, y, m). Thus there are a total of 15 $(= 3 \cdot 5)$ possible input / output transfer functions. Each possible transfer function can be computed using similar algebraic manipulations as used to compute $T_{r\to y}(s)$. As one additional example, consider the transfer function from input r to output e, denoted $T_{r\to e}(s)$. Again set $d=0$ and $n=0$ in Figure 6.4 so that $E(s)=R(s)-Y(s)$ and $Y(s)=G(s)K(s)E(s)$. Eliminating $Y(s)$ and solving for $E(s)$ yields $T_{r\to e}(s)=\frac{1}{1+G(s)K(s)}$. This result is a special case of the feedback interconnection discussed in Section 6.1.4 with unity forward path and $G(s)K(s)$ in the feedback path.

The two transfer functions computed above have special significance. **In particular, $S(s)=\frac{1}{1+G(s)K(s)}$ is called the sensitivity transfer function and $T(s)=\frac{G(s)K(s)}{1+G(s)K(s)}$ is the complementary sensitivity function.**[†] The name for $S(s)$ arises because this transfer function is a measure of the sensitivity of the closed-loop to small changes in the plant model $G(s)$.[‡] The two transfer functions $S(s)$ and $T(s)$ will be explored in detail later. **For now it is sufficient to notice that $S(s)+T(s)=1$ for all values of $s\in\mathbb{C}$. Thus $S(s)$ and $T(s)$ are "complementary" in the sense that they sum to 1.** All 15 possible input / output transfer functions, computed using similar manipulations as above, can now be expressed in terms of S and T as shown in the following matrix array:

$$\begin{bmatrix} Y(s) \\ U(s) \\ E(s) \\ M(s) \\ V(s) \end{bmatrix} = \begin{bmatrix} T(s) & -T(s) & G(s)S(s) \\ K(s)S(s) & -K(s)S(s) & -T(s) \\ S(s) & -S(s) & -G(s)S(s) \\ T(s) & S(s) & G(s)S(s) \\ K(s)S(s) & -K(s)S(s) & S(s) \end{bmatrix} \begin{bmatrix} R(s) \\ N(s) \\ D(s) \end{bmatrix} \tag{6.17}$$

This array should be interpreted using standard matrix multiplication. For example, the second output $U(s)$ is given by the second row of the matrix multiplied by the input vector:

$$U(s)=K(s)S(s)R(s)-K(s)S(s)N(s)-T(s)D(s) \tag{6.18}$$

This formula is a consequence of the principle of superposition for linear systems (Section 2.1.2). The individual effects of the inputs (r,n,d) are summed together to determine their combined effect on the output.

6.2.2 Stability Definition

The primary performance requirement for the controller is that it must ensure that the feedback system is stable. There are 15 input/output pairs in Equation 6.17. It is possible that some

[†] The notation $T(s)$ with no subscript will always denote the complementary sensitivity. If a subscript appears, e.g. $T_{r\to y}(s)$, then it denotes the input and output pair.

[‡] A brief sketch is as follows. $T(s)$ is the transfer function from reference $r(t)$ to output $y(t)$. A differential change dG in the plant model causes a differential change dT in T. A measure of the system sensitivity is $\frac{dT/T}{dG/G}$. This is the relative change in the closed loop divided by a relative change in the plant dynamics. This can be rewritten as $\frac{dT}{dG}\cdot\frac{G}{T}$. The derivative is given by:

$$\frac{dT}{dG}=\frac{d}{dG}\left[\frac{GK}{1+GK}\right]=\frac{K}{(1+GK)^2} \tag{6.16}$$

Subsituting this derivative yields $\frac{dT}{dG}\cdot\frac{G}{T}=S$. In other words, $S(s)$ measures the sensitivity of the closed-loop (from r to y) to small (differential) changes in the plant model.

of these transfer functions are stable and some are unstable. As a specific example, consider a feedback system with $K(s) = \frac{1}{s-1}$ and $G(s) = \frac{s-1}{s+3}$. Figure 6.5 shows this system along with the output $y(t)$ and control $u(t)$ response due to a unit step reference command. These responses can be understood by examining the appropriate transfer functions. First, a minimal realization for the product $G(s)K(s)$ is $\frac{1}{s+3}$. Note that $K(s)$ has a RHP pole at $s = +1$ that is cancelled by a zero at the same location in $G(s)$. The complementary senstivity function (from r to y) for this example is $T(s) = \frac{G(s)K(s)}{1+G(s)K(s)} = \frac{1}{s+4}$. This is a stable first-order system (one pole at $s = -4$) with time constant of $\tau_1 = 0.25sec$ and DC gain of $T(0) = 0.25$. Thus a unit step reference $r(t) = 1$ yields an output response $y(t)$ that converges in steady state to 0.25 with a settling time of $3\tau_1 = 0.75sec$. On the other hand, the transfer function from r to u is given by $T_{r\to u}(s) = \frac{K(s)}{1+G(s)K(s)} = \frac{s+3}{(s-1)(s+4)}$. This is an unstable system due to the pole at $s = +1$. As a consequence, a unit step reference $r(t) = 1$ causes the control command $u(t)$ to grow unbounded. All these features are shown in Figure 6.5.

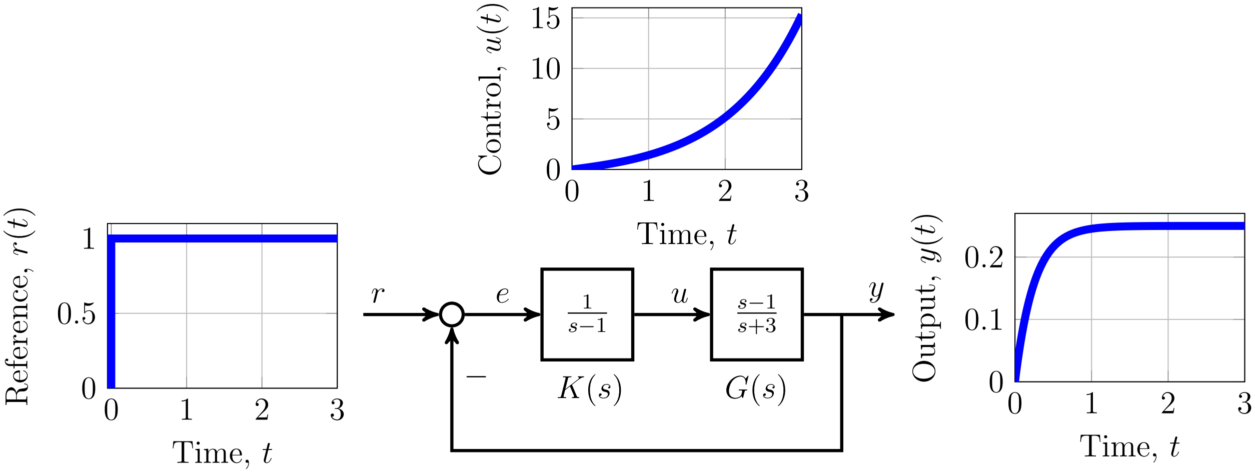


Figure 6.5: Feedback system with RHP pole/zero cancellation between $G(s)$ and $K(s)$

The behavior shown in Figure 6.5 would be unacceptable in practice even though $y(t)$ has a stable, first-order response. Specifically, all real systems have saturation limits on the magnitude of the input. At some point the unbounded control signal $u(t)$ in this example will exceed such limits. The system will either be damaged or the output $y(t)$ will also grow unbounded due to the (nonlinear) saturation of the input. Either outcome would be undesirable. In general, there will be undesirable effects if any of the 15 input / output transfer functions is unstable. This motivates the following definition for stability of the feedback system.

Definition 6.1. The feedback system is defined to be stable if all possible transfer functions in the system (r to e, d to u, n to y, etc) are stable.

It is important to note that the stability of the feedback system can be different from the stability of the individual components G and K. For example, the feedback system can be unstable even if both G and K are stable. Conversely, the feedback system can be stable even if G and/or K is unstable.

6.2.3 Stability Condition for Feedback Systems

This section derives a simple condition to check for the stability of the feedback system. Based on Definition 6.1, this requires checking the stability of the 15 transfer functions in Equation 6.17. This can be done using the equivalent conditions for free (IC) response (Fact 3.1) and BIBO (Fact 3.2) stability. In other words, each of the 15 transfer functions must be checked to ensure that all poles lie in the LHP. Upon closer inspection, the array of input/output transfer functions in Equation 6.17 has only four unique entries (neglecting sign): $S(s)$, $T(s)$, $G(s)S(s)$, and $K(s)S(s)$. Note that the sign does not affect the stability properties, e.g. $S(s)$ and $-S(s)$ have the same poles. Hence stability of these four transfer functions is necessary and sufficient for stability of the feedback system.

The simple stability condition is derived by considering the numerator and denominator polynomials for the plant and controller:

$$G(s) = \frac{N_G(s)}{D_G(s)} \quad \text{and} \quad K(s) = \frac{N_K(s)}{D_K(s)} \tag{6.19}$$

Moreover, assume that these are minimal representations for $G(s)$ and $K(s)$. Thus $N_G(s)$ and $D_G(s)$ are polynomials with no common roots that can be canceled. Similarly, $N_K(s)$ and $D_K(s)$ have no common roots. Minimal realizations for the four main input/output transfer functions listed above can be expressed in terms of these polynomials:

$$\begin{aligned}
S(s) &= \frac{1}{1+\frac{N_G(s)N_K(s)}{D_G(s)D_K(s)}} = \frac{D_G(s)D_K(s)}{D_G(s)D_K(s)+N_G(s)N_K(s)} \\
T(s) &= \frac{\frac{N_G(s)N_K(s)}{D_G(s)D_K(s)}}{1+\frac{N_G(s)N_K(s)}{D_G(s)D_K(s)}} = \frac{N_G(s)N_K(s)}{D_G(s)D_K(s)+N_G(s)N_K(s)} \\
G(s)S(s) &= \frac{\frac{N_G(s)}{D_G(s)}}{1+\frac{N_G(s)N_K(s)}{D_G(s)D_K(s)}} = \frac{N_G(s)D_K(s)}{D_G(s)D_K(s)+N_G(s)N_K(s)} \\
K(s)S(s) &= \frac{\frac{N_K(s)}{D_K(s)}}{1+\frac{N_G(s)N_K(s)}{D_G(s)D_K(s)}} = \frac{N_K(s)D_G(s)}{D_G(s)D_K(s)+N_G(s)N_K(s)}
\end{aligned} \tag{6.20}$$

Note that $D_G(s)D_K(s)+N_G(s)N_K(s)$ appears in the denominator of all four transfer functions. The following *characteristic equation* is key for the stability of the feedback system:

$$D_G(s)D_K(s) + N_G(s)N_K(s) = 0 \tag{6.21}$$

An important observation is that the feedback system will be unstable if there is a RHP pole/zero cancellation between $G(s)$ and $K(s)$. One example of this behavior was given in the previous section. To make this rigorous, consider the case where $K(s)$ has a pole at $s = p_0$ in the RHP and $G(s)$ has a zero at the same location. Thus both $D_K(p_0) = 0$ and $N_G(p_0) = 0$ so that p_0 is a root of the characteristic equation in Equation 6.21. Moreover, both $N_K(p_0) \neq 0$ and $D_G(p_0) \neq 0$ because the representations in Equation 6.19 are assumed to be minimal. Thus $K(s)S(s)$ has a pole at $s = p_0$ because the denominator is zero and the numerator is non-zero. Hence the feedback system is unstable in this case.[§] A similar argument can be used to show that the feedback system is unstable when $G(s)$ has a pole at $s = p_0$ in the RHP and $K(s)$ has a zero at the same location. **Thus a controller should never be designed to cancel a RHP pole or zero in the plant as this will result in an unstable feedback system.** The simple stability condition for the feedback system is now stated in terms of the transfer functions $G(s)$ and $K(s)$.

Fact 6.1. Consider a feedback system where $G(s)$ and $K(s)$ have the minimal realizations given in Equation 6.19. Moreover, assume there are no RHP pole/zero cancellations in the product $G(s)K(s)$. Then the feedback system is stable if and only if $1+G(s)K(s)$ has no RHP zeros. Equivalently, the feedback system is stable if and only if the sensitivity $S(s)$ is stable.

This fact will be proved by first showing that neither $S(s)$ nor $1 + G(s)K(s)$ has a RHP pole/zero cancellation. In particular the sensitivity $S(s)$ is given by (Equation 6.20):

$$S(s) = \frac{D_G(s)D_K(s)}{D_G(s)D_K(s) + N_G(s)N_K(s)} \tag{6.22}$$

Suppose the numerator of $S(s)$ has a zero at some $s = p_0$ in the RHP. Then either $D_G(p_0) = 0$ or $D_K(p_0) = 0$. If $D_G(p_0) = 0$ then $N_G(p_0) \neq 0$ (because the realization of $G(s)$ is assumed to be minimal) and $N_K(p_0) \neq 0$ (because $G(s)$ and $K(s)$ are assumed to have no RHP pole/zero cancellations). Hence if $D_G(p_0) = 0$ then the denominator must be non-zero at $s = p_0$. Similarly, if $D_K(p_0) = 0$ then the denominator must be non-zero at $s = p_0$. Thus $S(s)$ has no RHP pole/zero cancellations. In fact, the RHP poles of $S(s) = \frac{1}{1+G(s)K(s)}$ are precisely the RHP zeros of $1 + G(s)K(s)$.

The proof is completed by showing the following statement: The feedback system is unstable if and only if $S(s)$ is unstable. Two steps are required to show this equivalence:

1. *Show that if $S(s)$ is unstable then the feedback system is unstable:* The feedback system is unstable, by definition, if any of the possible input-output transfer functions is unstable. Hence if $S(s)$, the transfer function from reference to error, is unstable then the feedback system is unstable.

2. *Show that if the feedback system is unstable then $S(s)$ is unstable:* If the feedback system is unstable then at least one of the four transfer functions in Equation 6.20 is unstable. All four transfer functions have the same denominator and hence this implies that the characteristic equation (Equation 6.21) has a root at some $s = p_0$ in the RHP. The denominator of the sensitivity is the characteristic polynomial. As shown above, $S(s)$ has no RHP pole/zero cancellations and hence $|S(p_0)| = +\infty$. Thus if the feedback system is unstable then $S(s)$ has a pole at $s = p_0$.

[§] The other three transfer functions in Equation 6.20 are not necessarily unstable because their numerator may have a zero that cancels the root of the characteristic equation. For example, it was shown in the previous section that $K(s) = \frac{1}{s-1}$ and $G(s) = \frac{s-1}{s+3}$ causes $K(s)S(s)$ to be unstable while $T(s)$ is stable. However, the feedback system is still, by definition, unstable if at least one input/output transfer function is unstable.

6.3 Frequency Domain Performance Specifications

Summary: Most control design requirements can be specified in the frequency domain as bounds on the sensitivity $S(s)$ and complementary sensitivity $T(s)$. In particular, good reference tracking and disturbance rejection require $|S(j\omega)| \ll 1$ at low frequencies. Good rejection noise rejection requires $|T(j\omega)| \ll 1$ at high frequencies. A rrequirement on the control $u(t)$ is given by a specific bound on $T(s)$ across all frequencies. Finally, a rule of thumb for good robustness is to have $|S(j\omega)| \leq 2.5$ for all frequencies.

6.3.1 Revisiting Control Design Objectives

This section revisits the basic control design objectives, originally introduced in Section 4.1.3, using the generic feedback interconnection in Figure 6.6. In particular, each performance objective can be stated as a frequency domain requirement on a particular input/output transfer function. The controller $K(s)$ should be designed to satisfy the following objectives:

- **Stability:** The previous section focused on stability of the feedback system. At a minimum, the feedback system should be stable as defined in Definition 6.1. Fact 6.1 can be used to check for stability of the feedback system. Specifically, $G(s)$ and $K(s)$ should have no RHP pole/zero cancellations and $S(s) = \frac{1}{1+G(s)K(s)}$ should be stable.

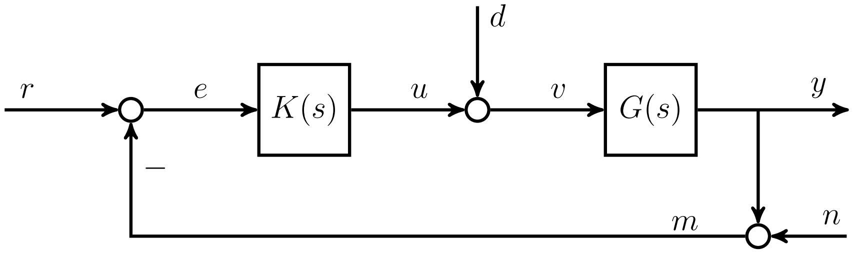


Figure 6.6: General feedback system

- **Reference Tracking:** The controller should be designed so that the system output $y(t)$ tracks, i.e. closely follows, the reference command $r(t)$. It will be useful to interpret tracking in terms of the reference to error transfer function. In the absence of noise the tracking error is given by $e(t) = r(t) - y(t)$. Thus $E(s) = S(s)R(s)$ (Equation 6.17) where $S(s) = \frac{1}{1+G(s)K(s)}$ is the sensitivity. If the reference command is $r(t) = A\cos(\omega t)$ then the steady-state tracking error satisfies:

$$e(t) \to A|S(j\omega)|\cos(\omega t + \angle S(j\omega)) \tag{6.23}$$

  For example, if $|S(j\omega)| = 0.05$ then the error amplitude is only 5% of the reference amplitude at the frequency ω. Thus good tracking at frequency ω requires $|S(j\omega)| \ll 1$.

- **Disturbance Rejection:** The controller should be designed so that a disturbance $d(t)$ has small effect on the output $y(t)$. These signals are related by $Y(s) = G(s)S(s)\,D(s)$ (Equation 6.17). Thus rejection of a sinusoidal disturbance $d(t) = A\cos(\omega t)$ requires

$|G(j\omega)S(j\omega)|$ to be "small". One notion of "small" is that the controller should reduce the effect of disturbance as compared to the performance with no control. In particular, if there is no control, i.e. $K(s) = 0$, then the disturbance and output are related by $Y(s) = G(s)D(s)$. Thus the controller reduces the effect of a disturbance if $|G(j\omega)S(j\omega)| \ll |G(j\omega)|$. In other words, $|S(j\omega)| \ll 1$ implies good disturbance rejection, i.e. that the effect of the disturbance with control is significantly less than the effect with no control.

- **Actuator Effort:** The control $u(t)$ should remain within allowable levels. Consider the relation from reference command to control: $U(s) = K(s)S(s)\,R(s)$ (Equation 6.17). This implies that the control effort required to track a reference signal at frequency ω is determined by $|K(j\omega)S(j\omega)|$. Note that the requirement on actuator effort is not simply $|K(j\omega)S(j\omega)| \ll 1$ because some control is required to perform the tracking. To clarify this statement consider the input required to track a constant reference $\bar{r}$ in steady-state. Recall that the DC (steady-state) gain of the plant is $G(0)$ (Section 3.7.1). Thus if $y(t) \to \bar{r}$ and $u(t) \to \bar{u}$ then $\bar{r} = G(0)\bar{u}$. In other words, tracking $\bar{r}$ in steady-state (assuming $d(t) = 0$) requires the control $\bar{u} = \frac{1}{G(0)}\bar{r}$. As a consquence, it does not make sense to require $|K(0)S(0)| < \frac{1}{G(0)}$ as this would prevent steady state tracking. More generally, tracking of dynamic signals requires additional control beyond the steady-state value. As a rule of thumb, a sensible requirement on reference $r(t)$ to control $u(t)$ is:

$$|K(j\omega)S(j\omega)| \leq \frac{2.5}{|G(0)|} \tag{6.24}$$

  The factor of 2.5 allows for control beyond that required for perfect steady state tracking. A more convenient form for the actuator requirement is obtained by multiplying Equation 6.24 by $G(s)$. Note that $G(s)K(s)S(s) = \frac{G(s)K(s)}{1+G(s)K(s)}$ which is the complementary sensitivity $T(s)$. Hence the actuator requirement can be rewritten as follows:

$$|T(j\omega)| \leq 2.5 \left| \frac{G(j\omega)}{G(0)} \right| \tag{6.25}$$

- **Noise Rejection:** It is typically required that any measurement inaccuracies, e.g. noise $n(t)$, have small effect on the output $y(t)$. This input/output pair is related by $Y(s) = -T(s)N(s)$ (Equation 6.17). If the noise is $n(t) = -A\cos(\omega t)$ then the steady-state output (due only to the noise) satisfies $y(t) \to A|T(j\omega)|\cos(\omega t + \angle T(j\omega))$. Thus good noise rejection at frequency ω requires $|T(j\omega)| \ll 1$. It is also typically required tha the noise does not cause large control effort. The relation from noise to control is $U(s) = -K(s)S(s)\,N(s)$ (Equation 6.17). This is the same as from $r(t)$ to $u(t)$ (except for the negative sign). Hence Equation 6.25 also ensures reasonable gain from noise to control.

- **Robustness to Model Uncertainty:** As noted in the previous section, the model used for control design is typically simplified. The controller must be robust, i.e. insensitive, to model errors introduced by this simplified model. Frequency domain methods for robustness analysis will be considered further in Chapter 7. **It will be shown that a rule of thumb for good robustness is to have $|S(j\omega)| \leq 2.5$ for all ω.**

Good reference tracking and disturbance rejection at a frequency ω both require $|S(j\omega)| \ll 1$. Good noise rejection at a frequency ω requires $|T(j\omega)| \ll 1$. These requirements are in direct conflict. Specifically, recall that $S(j\omega) + T(j\omega) = 1$ at all frequencies. Hence it is not possible to make both $|S(j\omega)|$ and and $|T(j\omega)|$ small at a specific frequency. This conflict is resolved by splitting the design objectives based on frequency. Typically, reference commands and disturbances dominate at low frequencies while noise dominates at high frequencies.

As a concrete example, recall the "low frequency" (Equation 5.31) and "high frequency" (random) signals discussed in Section 5.6. These signals and their Fourier Series amplitudes are shown in Figure 6.7. Reference signals tend to have the characteristics of the "low frequency" signal $r(t)$ shown on the left. In the time domain, such signals tend to be slowly varying, e.g. smooth transitions between constant values.[¶] In the frequency domain, such signals have significant energy (Fourier Series amplitudes) at low frequencies. The features of a disturbance signal depend on the specific problem but these signals also tend to be "low frequency". On the other hand, noise tends to have the characteristics of the "high frequency" random signal $n(t)$ shown on the right of Figure 6.7. In the time domain, such signals change rapidly. In the frequency domain, such signals tend to have roughly constant Fourier Series amplitudes across all frequencies. Note that the Fourier Series amplitudes for $n(t)$ are large relative to the amplitudes of $n(t)$ $\omega \geq 5\frac{rad}{sec}$. As a result, the noise dominates at high frequencies.

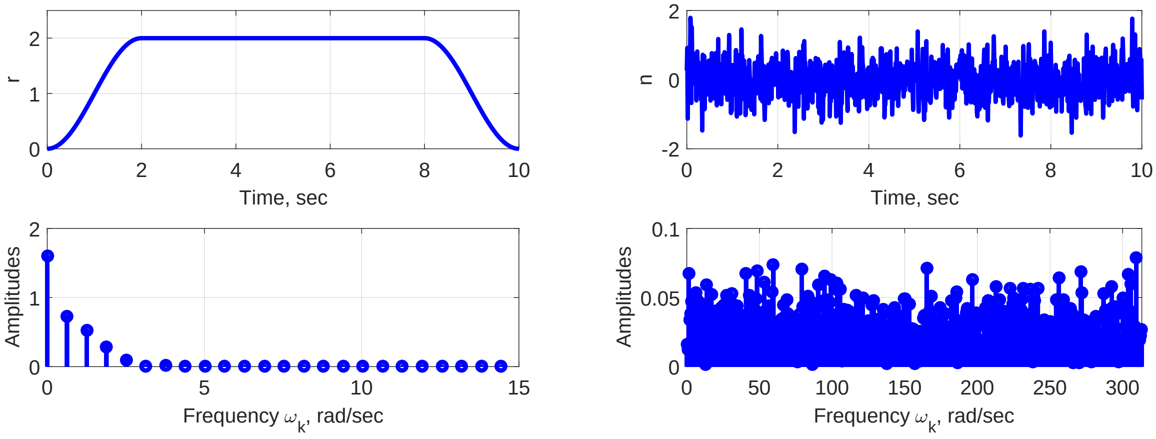


Figure 6.7: Left: Low Frequency signal (Equation 5.31) and its Fourier Series amplitudes. Right: High frequency (random) signal and its Fourier Series amplitudes.

To summarize, good reference tracking and disturbance rejection (with dominant energy below a frequency ω_1) require $|S(j\omega)| \ll 1$ for low frequencies $\omega \leq \omega_1$. Good rejection of "high frequency" noise (with dominant energy above a frequency ω_2) requires $|T(j\omega)| \ll 1$ for high frequencies $\omega \geq \omega_2$. Finally, a reasonable requirement on the control $u(t)$ is given by the bound on $T(s)$ in Equation 6.25.

[¶] The PID design in Chapter 4 considered step reference commands that jumped from one constant value to another. Step commands are useful for analysis as the system response can be analytically computed using the techniques in Chapter 3. However, it is typical for the reference command to smoothly transition from one constant value to another constant value. If the transition time is short then this smooth transition can be approximated by a step change for the purposes of analysis.

6.3.2 Example: Basic Frequency Domain Tradeoffs

This section illustrates the basic $S(s)$ and $T(s)$ tradeoffs using the following simple example:

$$\text{Plant: } \dot{y}(t) = u(t) \qquad\qquad \text{Control: } u(t) = K_p e(t) \tag{6.26}$$

The plant is a single integrator with transfer function $G(s) = \frac{1}{s}$. The controller is a proportional gain, $K(s) = K_p$. The sensitivity and complementary sensitivity for this example are:

$$S(s) = \frac{1}{1+G(s)K(s)} = \frac{s}{s+K_p} \quad \text{and} \quad T(s) = \frac{G(s)K(s)}{1+G(s)K(s)} = \frac{K_p}{s+K_p} \tag{6.27}$$

The left side of Figure 6.8 shows the Bode magnitude plot for $S(s)$ and $T(s)$ with $K_p = 1$. The plot also shows the *loop transfer function* $L(s) = G(s)K(s)$. There are three frequency regions of importance:

- **Low Frequencies:** In this region, the loop gain is large $|L(j\omega)| \gg 1$ and, as a result, $|S(j\omega)| \ll 1$. The small sensitivity corresponds to good reference tracking and disturbance rejection at low frequencies. Another consequence of the large loop gain is that $|T(j\omega)| \approx 1$. This implies that noise appears in the output with little attenuation, i.e. there is poor noise rejection at low frequencies. The poor noise rejection is a consequence of the good reference tracking and disturbance rejection obtained by relying on the feedback measurement at low frequencies.

- **High Frequencies:** In this region, the loop gain is small $|L(j\omega)| \ll 1$ and, as a result $|T(j\omega)| \ll 1$. The small complementary sensitivity corresponds to good noise rejection at high frequencies. Another consequence of the small loop gain is that $|S(j\omega)| \approx 1$. This implies that both the reference tracking and disturbance rejection are poor in this frequency range. The poor tracking and disturbance rejection are a consequence of the good noise rejection obtained by having small feedback at high frequencies.

- **Middle Frequencies:** The loop transfer function for this simple example has unity gain, i.e. $|L(j\omega_c)| = 1 = 0dB$, at a single frequency $\omega_c = K_p = 1\frac{rad}{sec}$. This frequency defines the *(loop) bandwidth* which is a measure of the system speed of response. Note that the closed-loop (see Equation 6.27) has a single pole $s_1 = -1\frac{rad}{sec}$ at this same frequency. This corresponds to a time constant of $\tau_1 = 1sec$. More precise definitions of bandwidth will be given in the next section. It will be shown later that the characteristics of the middle frequency region impact the stability and robustness of the closed loop.

The right plot of Figure 6.8 shows the output response $y(t)$ (red dashed) due to both a reference $r(t)$ (blue solid) and noise $n(t)$ input. The reference $r(t)$ and noise $n(t)$ inputs are given by the "low" and "high" frequency signals shown in Figure 6.7. For $K_p = 1$ the system bandwidth, as defined above, is $\omega_c = 1\frac{rad}{sec}$. This is sufficiently low that most of the noise is rejected and the output $y(t)$ is fairly smooth. However, the output has a noticeable lag relative to the reference signal. This is because the reference signal has frequency content in the $1 - 3\frac{rad}{sec}$ range (see Figure 6.7) which is above the system bandwdith.

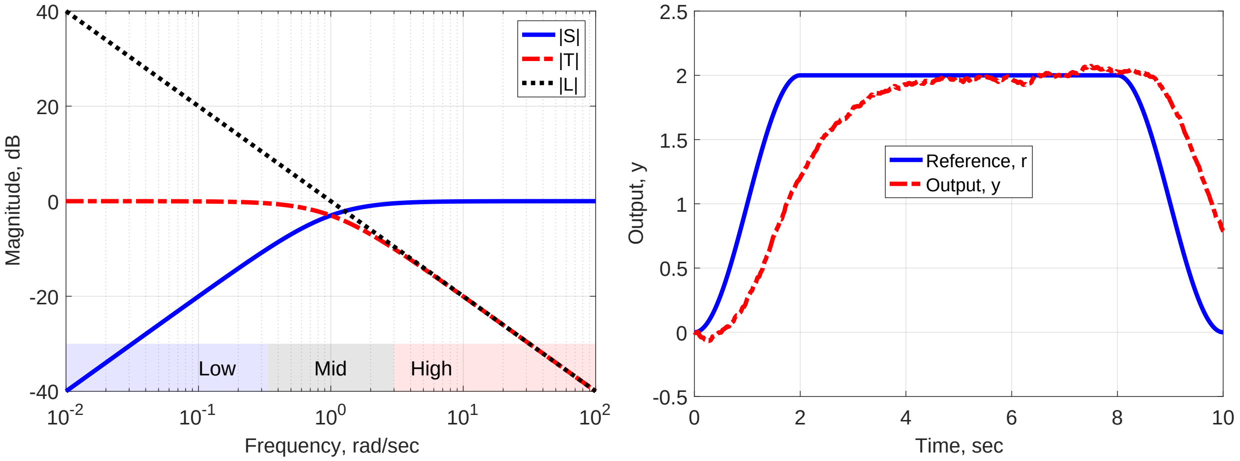


Figure 6.8: Left: Bode magnitude plot of $S(s)$, $T(s)$, and $L(s) = G(s)K(s)$ for $G(s) = \frac{1}{s}$ and $K(s) = 1$. Right: Time response due to reference $r(t)$ and noise $n(t)$ inputs.

Increasing the proportional gain to $K_P = 10$ yields the results shown in Figure 6.9. The left plot again shows the Bode magnitude plots for $S(s)$, $T(s)$, and $L(s)$. The larger gain pushes the system bandwidth to $10\frac{rad}{sec}$. For this gain the closed-loop (see Equation 6.27) pole is at $s_1 = -K_p = -10\frac{rad}{sec}$ with corresponding time constant $\tau_1 = 0.1sec$. Thus larger bandwidths correspond to a faster speed of response. The good tracking and disturbance rejection extends over a wider "low" frequency region. The disadvantage is that the complementary sensitivity function has larger gain at high frequencies. This corresponds to degraded noise rejection. The right plot of Figure 6.9 shows the time domain response due to reference and noise inputs. The output responds faster to the reference command but is also degraded due to the noise. Both of these features support the conclusions from the frequency domain plots (better reference tracking but degraded noise rejection).

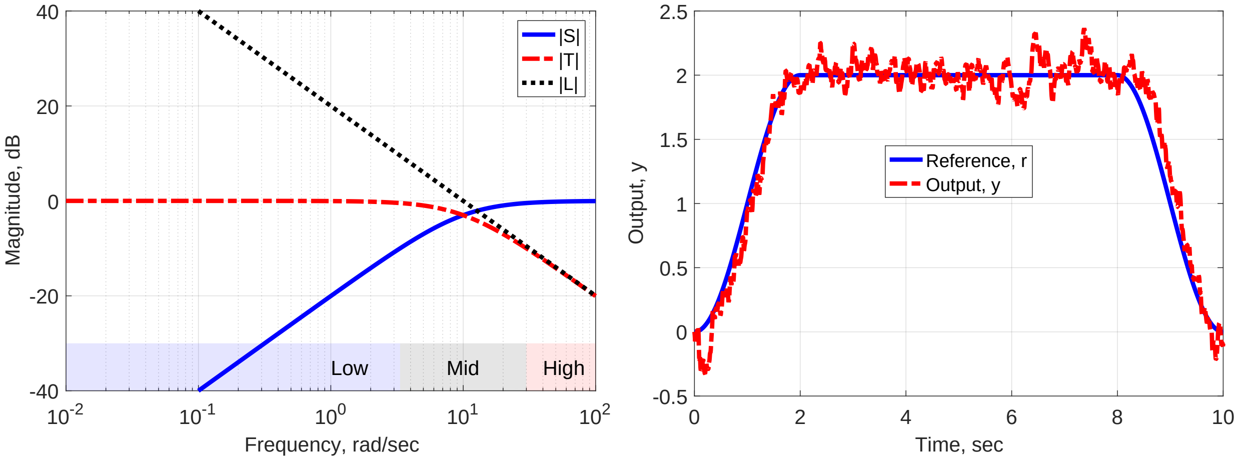


Figure 6.9: Left: Bode magnitude plot of $S(s)$, $T(s)$, and $L(s) = G(s)K(s)$ for $G(s) = \frac{1}{s}$ and $K(s) = 10$. Right: Time response due to reference $r(t)$ and noise $n(t)$ inputs.

6.4 Introduction to Loopshaping

Summary: Loopshaping is a design procedure that focuses on the loop transfer function $L(s) = G(s)K(s)$. Requirements on $S(s)$ and $T(s)$ are translated into related requirements on the loop $L(s)$. In particular, good reference tracking and disturbance rejection require large loop gain at low frequencies. Good noise rejection requires small loop gain at high frequencies. Stability and robustness are associated with a shallow slope in the middle frequencies. The speed of response is characterized by the system bandwidth as defined by the loop crossover frequency. In loopshaping, the controller is designed in steps to meet the requirements on the loop $L(s)$. Each step uses one of the following components: (a) Proportional Gain, (b) Low Frequency and Integral Boosts, (c) High Frequency Rolloff, and (d) Lead.

6.4.1 Approach

The goal is to design $K(s)$ such that the feedback system is stable. The system should also satisfy various performance and robustness requirements. Note that each requirement in the previous section can be specified as a bound on either $|S(j\omega)|$ or $|T(j\omega)|$. The controller $K(s)$ appears in the denominator of both $S(s)$ and $T(s)$. As a result, it is difficult to design a controller to directly satisfy the requirements on $|S(j\omega)|$ and $|T(j\omega)|$.

Loopshaping **is a design procedure that instead focuses on the loop transfer function** $L(s) = G(s)K(s)$**. First, the requirements on** $S(s)$ **and** $T(s)$ **are translated into related requirements on the loop** $L(s)$**.** This is discussed further in Section 6.4.2. **Next the controller is designed in steps to meet the requirements on the loop** $L(s)$**. Each step uses a specific controller component to improve the loop characteristics in the low, middle, or high frequencies.** Important components for the control design are introduced in Section 6.4.3. Note that the loop transfer function $L(s) = G(s)K(s)$ has a simple (linear) dependence on the controller $K(s)$. This property is useful in the design procedure as shown in Section 6.6. The overall process described in this chapter closely follows the notes in [20]. Loopshaping can more easily handle higher order plants than our previous PID design approach. In addition, it extends naturally to systems with multiple inputs and multiple outputs. This extension to multiple inputs/outputs is not covered in this text but details can be found in more advanced textbooks [21, 24].

6.4.2 Requirements on Loop $L(s) = G(s)K(s)$

Section 6.3.1 introduced several performance requirements on $|S(j\omega)|$ and $|T(j\omega)|$. The first step in loopshaping is to translate these into requirements on the loop transfer function. Figure 6.10 shows sample performance bounds on $|S(j\omega)|$, $|T(j\omega)|$, and $|L(j\omega)|$. The black dashed curves in each figure correspond to an example, sensitivity, complementary sensitivity, and loop transfer functions for the plant $G(s) = \frac{1}{s^2+0.7s+0.01}$ and controller $K(s) = 1$. Again, there are three frequency regions of importance:

- **Low Frequencies:** The upper left plot of Figure 6.10 shows a sample upper bound on $|S(j\omega)|$. In particular, $|S(j\omega)| \leq \epsilon_1 \ll 1$ for $\omega \leq \omega_1$ yields good reference tracking

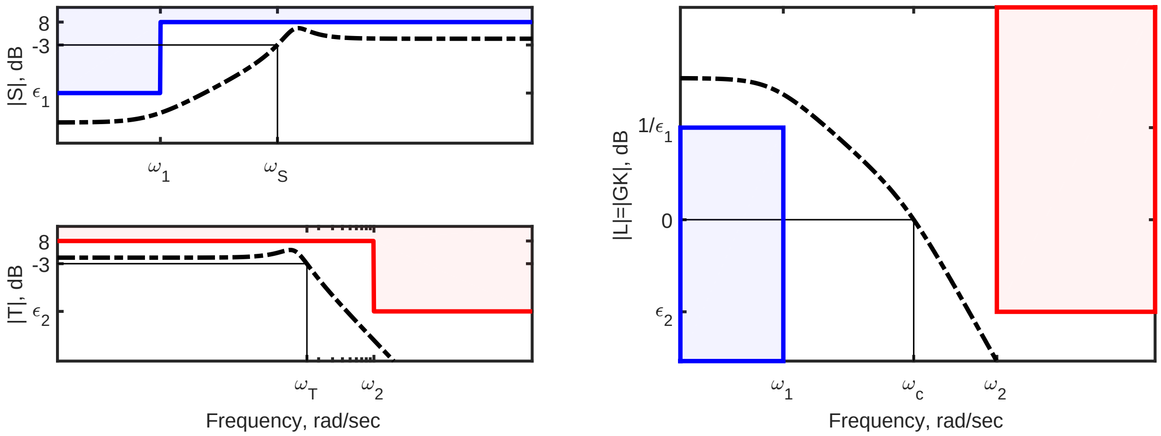


Figure 6.10: Requirement bounds on $|S(j\omega)|$ (upper left), $|T(j\omega)|$ (lower left) and $|L(j\omega)|$ (right). All plots show sample result (black dashed) for $G(s) = \frac{1}{s^2+0.7s+0.01}$ and $K(s) = 1$.

and disturbance rejection. The sensitivity can be expressed in terms of the loop transfer function: $S(s) = \frac{1}{1+L(s)}$. Therefore, if $|S(j\omega)|$ is small then the loop gain $|L(j\omega)|$ must be large. Roughly, $|S(j\omega)| \leq \epsilon_1 \ll 1$ is approximately equivalent to $|L(j\omega)| \geq \frac{1}{\epsilon_1} \gg 1$. This lower bound on $|L(j\omega)|$ for $\omega \leq \omega_1$ is shown in the right plot of Figure 6.10. The plots are drawn for the specific values $\epsilon_1 = 0.1 = -20dB$ an $\frac{1}{\epsilon_1} = 10 = +20dB$. Typical requirements would likely be more stringent (smaller value of ϵ_1).

Recall that $S(j\omega) + T(j\omega) = 1$ for all ω. As a result, $|S(j\omega)| \ll 1$ at low frequencies implies $|T(j\omega)| \approx 1$. Thus bounds on $|T(j\omega)|$ at low frequencies, e.g. actuator requirements, are typically less important than the bounds on $|S(j\omega)|$.

- **High Frequencies:** The lower left plot of Figure 6.10 shows a sample upper bound on $|T(j\omega)|$. The bound $|T(j\omega)| \leq \epsilon_2 \ll 1$ for $\omega \geq \omega_2$ yields good noise rejection. The complementary sensitivity can also be expressed in terms of the loop transfer function: $T(s) = \frac{L(s)}{1+L(s)}$. Therefore, if $|T(j\omega)|$ is small then the loop gain $|L(j\omega)|$ must also be small. Roughly, $|T(j\omega)| \leq \epsilon_2 \ll 1$ is approximately equivalent to $|L(j\omega)| \leq \epsilon_2 \ll 1$. This upper bound on $|L(j\omega)|$ for $\omega \geq \omega_2$ is shown in the right plot of Figure 6.10. The plots are drawn for the specific value $\epsilon_2 = 0.1 = -20dB$. Again, typical requirements would likely be more stringent (smaller value of ϵ_2).

  Note that $|T(j\omega)| \ll 1$ at high frequencies implies $|S(j\omega)| \approx 1$. This again follows from $S(j\omega) + T(j\omega) = 1$ for all ω. Thus bounds on $|S(j\omega)|$ at high frequencies are typically less important than the bounds on $|T(j\omega)|$.

- **Middle Frequencies:** A rule of thumb for good robustness is $|S(j\omega)| \leq 2.5 \approx 8dB$ as discussed further in Chapter 7. For now it is sufficient to state that this robustness requirement leads to a corresponding rule of thumb requirement on the loop gain $|L(j\omega)|$. Specifically, the slope of the loop gain should be $\geq -30dB$ per decade in the middle frequency region. If the slope is shallow, e.g. $-20dB$ per decade, then the feedback system will typically be stable with good robustness. If the slope is too steep, e.g.

$-40dB$ per decade, then there will be stability or robustness issues. As a result, the loop gain should not transition too steeply through the middle frequency region. This will thus require a separation between the low and high frequency regions, i.e. ω_1 and ω_2 cannot be too close together. These statements will be made more precise in Chapter 7.

One final aspect for discussion is the speed of response. The step response settling time was previously used to measure the speed of a simple system. An alternative, frequency domain measure for the speed of response is the "bandwidth". There are three distinct methods to measure the bandwidth. First, a *loop crossover frequency* ω_c is defined by $|L(j\omega_c)| = 1 = 0dB$. Typically there is only a single loop crossover frequency but in some cases there can be more than one. The *loop bandwidth* is given by the lowest loop crossover frequency.[‖] A higher bandwidth corresponds to a faster speed of response. A second measure of speed is the *sensitivity bandwidth*, denoted ω_S. This is defined as the highest frequency such that $|S(j\omega)| \leq \frac{1}{\sqrt{2}} = -3dB$ for all $\omega \leq \omega_S$. This is roughly a measure of the frequency range for tracking and disturbance rejection. The third and final measure of speed is the *complementary sensitivity bandwidth*, denoted ω_T. This is defined as the lowest frequency such that $|T(j\omega)| \leq \frac{1}{\sqrt{2}} = -3dB$ for all $\omega \geq \omega_T$. This is roughly a measure of the frequency range for noise rejection. These three different bandwidths are typically ordered by $\omega_S \leq \omega_c \leq \omega_T$. They are all labelled in Figure 6.10 for the example system. The loop bandwidth ω_c is the most common of these three bandwidth definitions and will be used most frequently.

To summarize, good reference tracking and disturbance rejection require large loop gain at low frequencies. Good noise rejection requires small loop gain at high frequencies. Stability and robustness are associated with a shallow slope in the middle frequencies. The speed of response is characterized by the system bandwidth as commonly given by the loop crossover frequency.

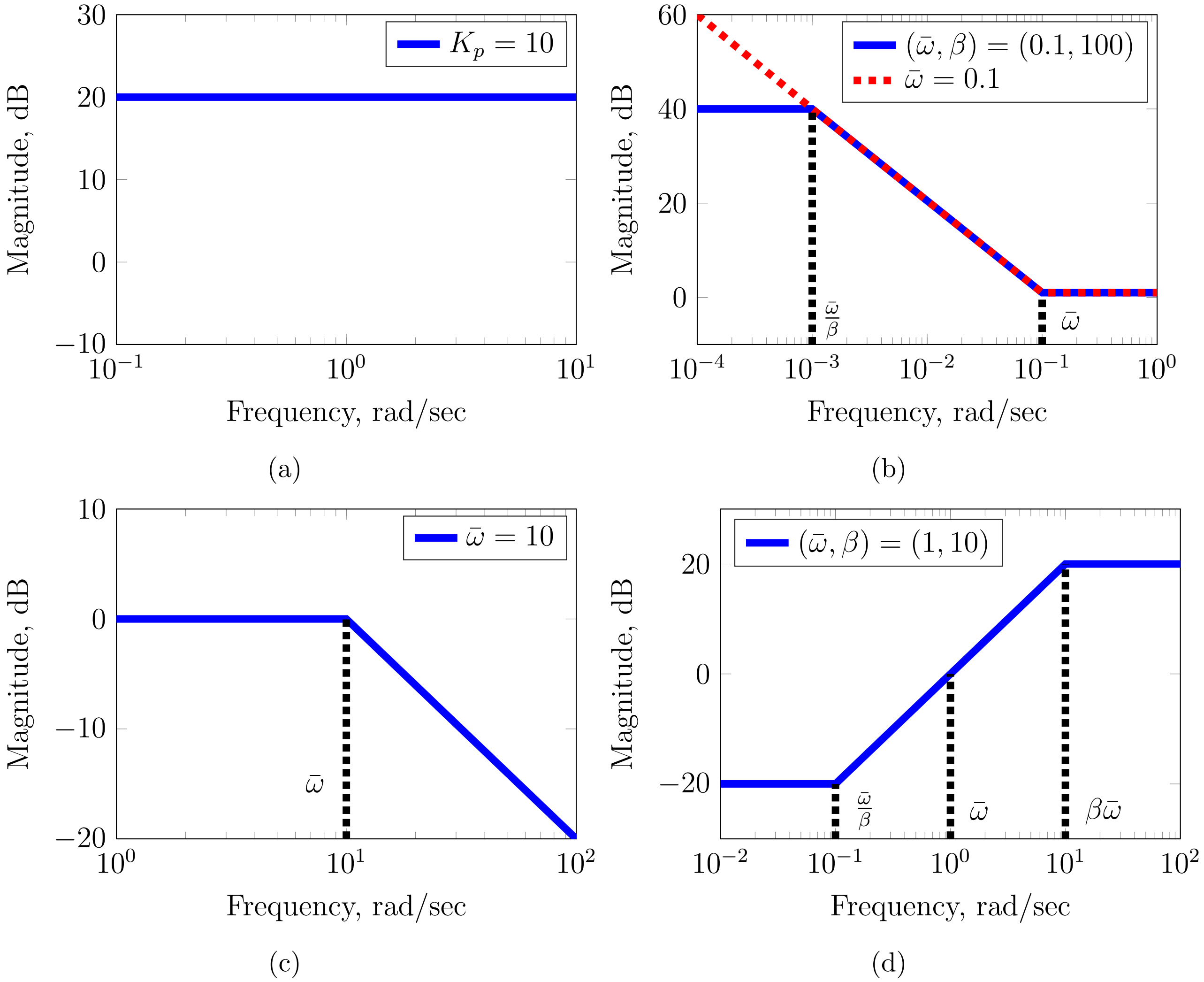
6.4.3 Controller Components for Loopshaping

The main components used in the loopshaping design process are: (a) Proportional Gain, (b) Low Frequency and Integral Boosts, (c) High Frequency Rolloff, and (d) Lead. Figure 6.11 shows an example Bode magnitude plot (straight-line approximation) for each of these components. This terminology and the precise form of these components is taken from [20]. Additional components can be defined but these four components will be sufficient for most designs. The purpose of each key component is briefly summarized as follows:

- **Proportional Gain:** The proportional gain K_p has constant magnitude across frequency. The effect of this component is to shift the entire loop shape magnitude up or down. This will primarily be used to set the crossover frequency of the loop shape.

- **Integral Boost:** The integral boost is $K(s) = \frac{s+\bar{\omega}}{s}$ where $\bar{\omega}$ is a tunable parameter. This is used to increase the low frequency gain of the loop shape. The integral boost has a high frequency gain of $|K(\infty)| = 1$, a pole at $s = 0$, and a zero at $s = -\bar{\omega}$. The slope

[‖] The *loop bandwidth* and *loop crossover frequency* are the same for systems with only one loop crossover. This is the typical case.



(a)

(b)

(c)

(d)

Figure 6.11: Straight-line Bode magnitude approximations for: (a) Proportional Gain, (b) Low Frequency (solid blue) and Integral (red dashed) Boosts, (c) Rolloff, and (d) Lead.

of the Integral Boost is $-20dB$ per decade for $\omega \leq \bar{\omega}$, i.e the gain increases by a factor of 10 for every decrease in frequency by a factor of 10. The gain continues to increase at low frequencies and hence the integral boost has infinite DC gain: $|K(0)| = \infty$. These features can be observed in the straight line approximation (dashed red) shown in Figure 6.11(b). Note that an integral boost corresponds to a PI controller. Specifically, if $U(s) = K(s)E(s)$ then the input $e(t)$ and output $u(t)$ are related by:

$$\dot{u}(t) = \dot{e}(t) + \bar{\omega}e(t) \tag{6.28}$$

Integrating both sides of this ODE yields:

$$u(t) = e(t) + \bar{\omega} \int_0^t e(\tau)\, d\tau \tag{6.29}$$

This has the form of a PI controller with proportional gain of 1 and integral gain of $\bar{\omega}$.

- **Low Frequency Boost:** The low frequency boost is $K(s) = \frac{s+\bar{\omega}}{s+\bar{\omega}/\beta}$ where $\bar{\omega}$ and β are tunable parameters. This is used to increase the low frequency gain of the loop shape. This component has a low frequency gain of $|K(0)| = \beta$ and a high frequency gain of $|K(\infty)| = 1$. It has a pole at $s = -\frac{\bar{\omega}}{\beta}$ and a zero at $s = -\bar{\omega}$. These features can be observed in the straight line approximation (solid blue) shown in Figure 6.11(b). This component is similar to an integral boost except that the gain "levels off" at $|K(0)| = \beta$ for low frequencies $\omega \leq \frac{\bar{\omega}}{\beta}$.

- **High Frequency Rolloff:** The rolloff is $K(s) = \frac{\bar{\omega}}{s+\bar{\omega}}$ where $\bar{\omega}$ is a tunable parameter. This is used to reduce the high frequency gain of the loop. This component has a low frequency gain of $|K(0)| = 1$ and a pole at $s = -\bar{\omega}$. The slope of the rolloff is $-20dB$ per decade for $\omega \geq \bar{\omega}$, i.e the gain decreases by a factor of 10 for increase in frequency by a factor of 10. The gain continues to decrease at high frequencies. Hence the rolloff gain goes to zero at high frequencies: $|K(j\omega)| \approx \frac{\bar{\omega}}{\omega}$ as $\omega \to \infty$. These features can be observed in the straight line approximation shown in Figure 6.11(c).

- **Lead:** The lead is $K(s) = \frac{\beta s+\bar{\omega}}{s+\beta\bar{\omega}}$ where $\bar{\omega}$ and β are tunable parameters. This is used to increase the slope of the loop near $\omega = \bar{\omega}$. This improves the stability and robustness properties of the feedback system as discussed in Chapter 7. This component has a low frequency gain of $|K(0)| = \beta^{-1}$ and a high frequency gain of $|K(\infty)| = \beta$. It has a pole at $s = -\beta\bar{\omega}$ and a zero at $s = -\frac{\bar{\omega}}{\beta}$. These features can be observed in the straight line approximation shown in Figure 6.11(d).

6.5 Effects of Components

Summary: This section demonstrates the effects of the proportional gain, integral boost, and rolloff components for loopshaping. A proportional gain (greater than 1) increases the loop magnitude at all frequencies. This will increase the system bandwidth, reduce steady state error, and increase sensitivity to noise. An integral boost increases the low frequency gain but leaves the high frequencies unchanged. The use of integral control results in zero steady state error due to constant reference commands. However, the integral boost has negligible effect on the bandwidth and sensitivity to noise. Finally, the rolloff decreases the high frequency gain but leaves the low frequencies unchanged. The rolloff decreases sensitivity to noise but has negligible effect on bandwidth and steady-state error.

6.5.1 Example System

This section demonstrates the effects of the proportional gain, integral boost, and rolloff components for loopshaping. The low frequency boost component has similar effects as the integral boost and hence will not be considered in detail. The lead component will be discussed in Section XXX. Consider the general feedback diagram shown in Figure 6.12.[**] For concreteness, the example will focus on a plant with the following first-order dynamics:

$$\dot{y}(t) + 2y(t) = 5u(t) \tag{6.30}$$

The corresponding transfer function for this system is $G(s) = \frac{5}{s+2}$. The top left subplot of Figure 6.13 shows the Bode magnitude plot for this system. The system has DC gain $G(0) = 2.5 = 8dB$ and corner frequency of $2\frac{rad}{sec}$.

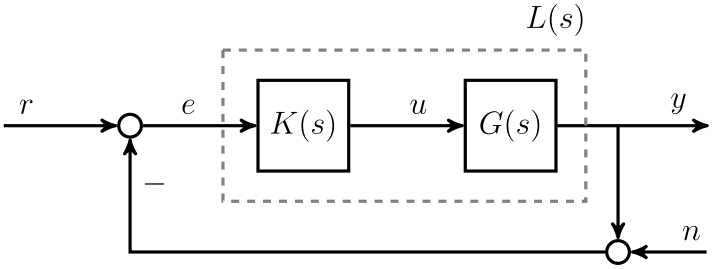


Figure 6.12: General feedback system

Next, consider the performance of the (closed-loop) feedback system with this plant $G(s)$ and a simple proportional gain $K(s) = 1$. In this case the loop is simply the plant, i.e. $L(s) = G(s)$. The closed-loop sensitivity and complementary sensitivity are:

$$S(s) = \frac{1}{1+G(s)} = \frac{s+2}{s+7} \quad \text{and} \quad T(s) = \frac{G(s)}{1+G(s)} = \frac{5}{s+7} \tag{6.31}$$

[**] The example will focus on the reference $r(t)$ and noise $n(t)$ inputs. The disturbance $d(t)$ is not included in the feedback diagram. However, recall that both the reference tracking and disturbance rejection performance are related to the sensitivity function $S(s)$. Thus the impact of a particular loopshaping component on the reference tracking will have a similar effect on disturbance rejection.

The Bode magnitude plots for $S(s)$ and $T(s)$ are shown in the lower left subplot of Figure 6.13. The right plot of Figure 6.13 shows the (closed-loop) output response $y(t)$ (black dotted) due to both a reference $r(t)$ (blue solid) and noise $n(t)$ input. Recall that that the bandwidth of the feedback system is defined by the loop crossover frequency ω_c such that $|L(j\omega_c)| = 1 = 0dB$. For $K(s) = 1$ this is the same frequency ω_c such that $|G(j\omega_c)| = 1 = 0dB$. For the example system, this crossover occurs at $\omega_c = 4.58\frac{rad}{sec}$. The output response shown in Figure 6.13 (right) roughly follows the reference command. However, there is a noticeable lag in the output relative to reference in the initial response ($0 \le t \le 2$). Moreover, the output shows a large steady state error and significant effect from the noise.

Note that the steady-state error due to the reference command can be computed from the DC gain of the sensitivity $S(s)$. In particular, $S(s) = \frac{s+2}{s+7}$ is the transfer function from reference to error. This corresponds to the following closed-loop ODE:

$$\dot{e}(t) + 7e(t) = \dot{r}(t) + 2r(t) \tag{6.32}$$

The closed-loop is stable and hence if the reference converges to a constant, $r(t) \to \bar{r}$, then the error also converges to a constant $e(t) \to \bar{e}$. Setting derivatives to zero in Equation 6.32 yields the following steady-state relation: $\bar{e} = \frac{2}{7}\bar{r} = S(0)\bar{r}$. For the example response shown in Figure 6.13, the reference converges to $\bar{r} = 2$. As a result, the error converges to $\bar{e} = \frac{4}{7} \approx 0.57$. This corresponds to a steady state output $\bar{y} = \bar{r} - \bar{e} = 1.43$. This is the response due only to the reference and the noise causes some variation around this value. **For any stable closed-loop system, the error due to a constant reference input $\bar{r}$ converges to $\bar{e} = S(0)\bar{r}$.**

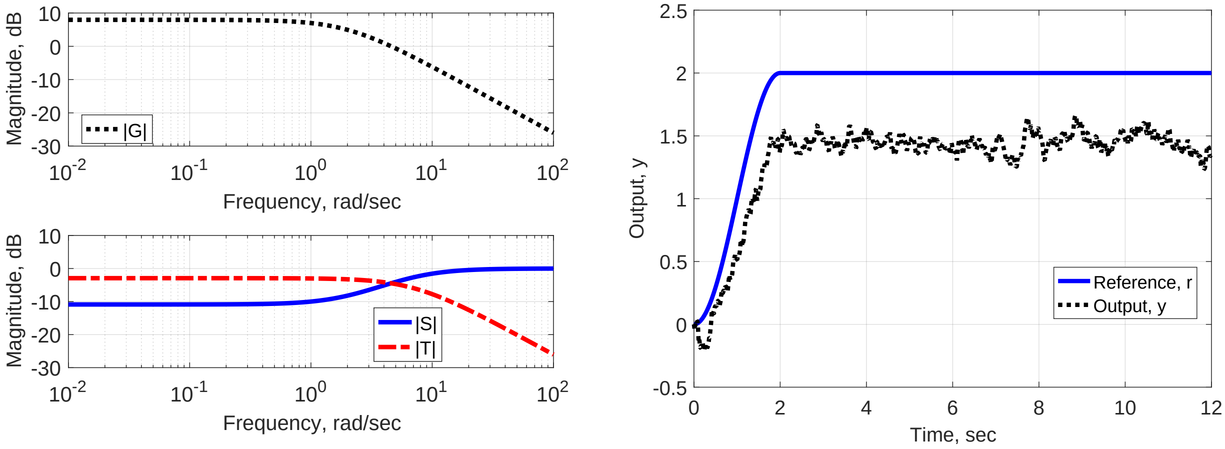


Figure 6.13: Left: Bode magnitude plots for $G(s) = \frac{5}{s+2}$ (top), $S(s) = \frac{1}{1+G(s)}$ and $T(s) = \frac{G(s)}{1+G(s)}$ (bottom). Right: Time response of closed-loop with reference $r(t)$ and noise $n(t)$ inputs.

6.5.2 Proportional Gain

This subsection considers the effect of the proportional gain. As a specific example, consider the gain $K(s) = 3 = 9.5dB$. Figure 6.14 (left) shows the Bode magnitude plot for $G(s)$, $K(s)$, and $L(s) = G(s)K(s)$. Recall that the following fact regarding magnitudes (in dB):

$$20\log_{10}|G(j\omega)K(j\omega)| = 20\log_{10}|G(j\omega)| + 20\log_{10}|K(j\omega)| \tag{6.33}$$

In other words, the Bode magnitude plot of $L(s)$ is simply the sum of the Bode magnitude plots of $G(s)$ and $K(s)$. The proportional gain has constant magnitude across all frequencies. Hence the effect of this loopshaping component is to shift the loop magnitude by the same amount at all frequencies. For this example, note that $|L(j\omega)|$ in Figure 6.14 is simply $|G(j\omega)|$ shifted upward by a gain of $3 = 9.5dB$.

There are several consequences of this upward shift in the loop gain. First, the DC gain of the loop increases from $G(0) = 2.5$ to $L(0) = 7.5$. As a result the DC gain of the sensitivity is $S(0) = \frac{1}{1+L(0)} = 0.12$. Thus the steady state error for a reference $\bar{r} = 2$ is reduced to $\bar{e} = 0.24$. Second, the crossover frequency increases from $4.58\frac{rad}{sec}$ for $G(s)$ to $14.87\frac{rad}{sec}$ for $L(s)$. Third, the high frequency gain is shifted upward by $9.5dB$ from $|G(j\omega)|$ to $|L(j\omega)|$. These effects appear in the time responses shown in the right plot of Figure 6.14. This plot shows the time response of the closed loop with $K(s) = 3$ (red dashed). For comparison it also shows the time response of the closed loop with $K(s) = 1$ (black dotted). The proportional gain $K(s) = 3$ yields faster response due to the increased bandwidth, lower steady state error due to the larger loop gain at DC, and larger sensitivity to noise due to larger loop gain at high frequencies.

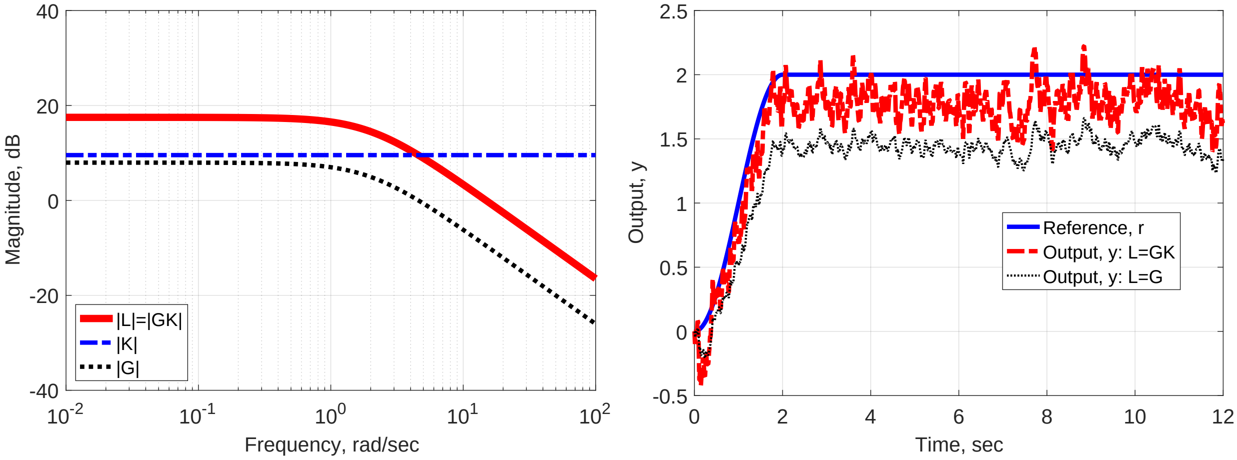


Figure 6.14: Left: Bode magnitude plots for $G(s) = \frac{5}{s+2}$, $K(s) = 3$, and $L(s) = G(s)K(s)$. Right: Time responses of closed-loop with reference $r(t)$ and noise $n(t)$ inputs.

6.5.3 Integral Boost

This subsection considers the effect of the integral boost. As a specific example, consider the boost with $\bar{\omega} = 0.5\frac{rad}{sec}$, i.e. $K(s) = \frac{s+0.5}{s}$. The left plot of Figure 6.15 shows the Bode magnitude plot for $G(s)$, $K(s)$, and $L(s) = G(s)K(s)$. Again, the Bode magnitude plot of $L(s)$ is simply the sum of the Bode magnitude plots of $G(s)$ and $K(s)$. The integral boost has a gain of approximatley $1 = 0dB$ well above above the corner frequency of $\bar{\omega} = 0.5\frac{rad}{sec}$. As a result, the integral boost has negligible effect above this corner frequency: $|L(j\omega)| \approx |G(j\omega)|$

for $\omega \geq 4\bar{\omega} = 2\frac{rad}{sec}$. Below the corner frequency the boost has a slope of $-20dB$ per decade. Thus the boost raises the loop gain by 10 for every decrease in frequency by a factor of 10. The gain continues to rise at low frequencies.

There are several consequences of the integral boost. First, the DC gain of the loop increases from $G(0) = 2.5$ to $L(0) = \infty$. As a result the DC gain of the sensitivity is $S(0) = \frac{1}{1+L(0)} = 0$ and thus the steady state error for a reference $\bar{r} = 2$ is $\bar{e} = 0$. **The use of integral control results in zero steady state error due to constant reference commands.** Second, the crossover frequency and the high frequency gain of the loop $L(s) = G(s)K(s)$ are basically the same as for $G(s)$. This is because the integral boost has gain approximately equal to $1 = 0dB$ at high frequencies. These effects appear in the time responses shown in the right plot of Figure 6.15. This plot shows the time response of the closed loop with $K(s) = \frac{s+0.5}{s}$ (red dashed). For comparison the time response of the closed loop with $K(s) = 1$ (black dotted) is also shown. The integral boost yields zero steady state error on average (neglecting the effect of noise). However, the initial transient and the sensitivity to noise are unchanged by the integral boost. This is because the boost has negligible effect on the bandwidth and high frequency gain.

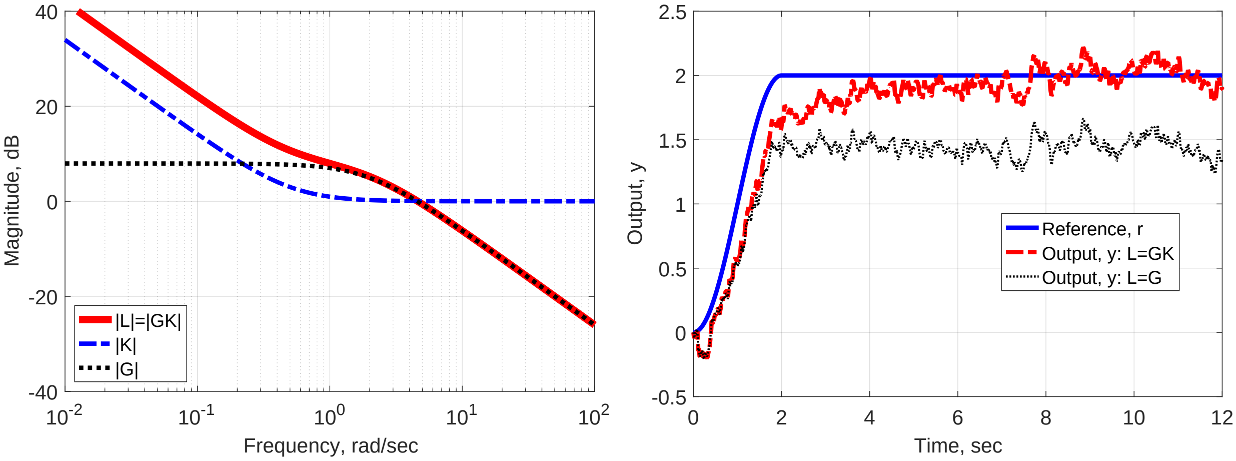


Figure 6.15: Left: Bode magnitude plots for $G(s) = \frac{5}{s+2}$, $K(s) = \frac{s+0.5}{s}$, and $L(s) = G(s)K(s)$. Right: Time responses of closed-loop with reference $r(t)$ and noise $n(t)$ inputs.

6.5.4 Rolloff

This subsection considers the effect of the high frequency rolloff. As a specific example, consider the rolloff with $\bar{\omega} = 10\frac{rad}{sec}$, i.e. $K(s) = \frac{10}{s+10}$. The left plot of Figure 6.16 shows the Bode magnitude plot for $G(s)$, $K(s)$, and $L(s) = G(s)K(s)$. Again, the Bode magnitude plot of $L(s)$ is simply the sum of the Bode magnitude plots of $G(s)$ and $K(s)$. The rolloff has a gain approximately equal to $1 = 0dB$ well below the corner frequency of $\bar{\omega} = 10\frac{rad}{sec}$. As a result, the integral boost has negligible effect below this corner frequency: $|L(j\omega)| \approx |G(j\omega)|$ for $\omega \leq 0.25\bar{\omega} = 2.5\frac{rad}{sec}$. Above the corner frequency the rolloff has a slope of $-20dB$ per decade. Thus the rolloff reduces the loop gain by 10 for every increase in frequency by a factor of 10.

There are several consequences of the rolloff. First, the DC gain of the loop is unchanged: $G(0) = L(0) = 2.5$ As a result the DC gain of the sensitivity and the steady state error due to reference commands are unchanged. Second, the crossover frequency is also basically unchanged because the rolloff has gain approximately equal to $1 = 0dB$ at low frequencies. Third, the high frequency gain of the loop $L(s)$ is reduced by the rolloff as compared to $G(s)$. These effects appear in the time responses shown in the right plot of Figure 6.16. This plot shows the time response of the closed loop with $K(s) = \frac{10}{s+10}$ (red dashed). For comparison the time response of the closed loop with $K(s) = 1$ (black dotted) is also shown. The rolloff has negligible effect on the steady state error and the initial transient response. However, the effect of the noise is reduced by the rolloff and the output is smoother. The improved noise rejection can be more easily observed in the inset plot which enlarges the steady-state response.

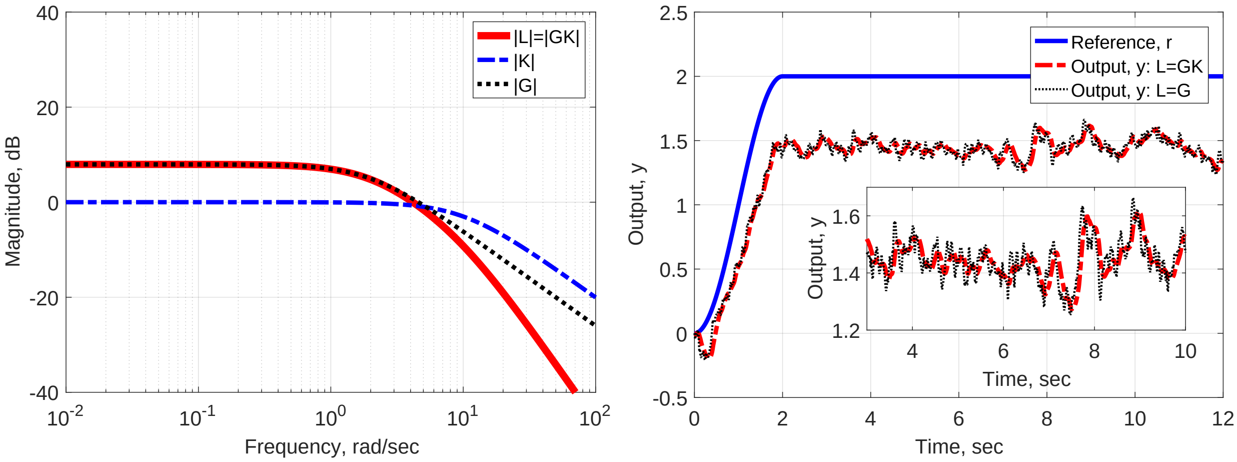


Figure 6.16: Left: Bode magnitude plots for $G(s) = \frac{5}{s+2}$, $K(s) = \frac{10}{s+10}$, and $L(s) = G(s)K(s)$. Right: Time responses of closed-loop with reference $r(t)$ and noise $n(t)$ inputs.

6.6 Loopshaping Design

Summary: Performance specifications include closed loop stability, desired loop crossover frequency, bounds on $|S(j\omega)|$ for tracking performance, and bounds on $|T(j\omega)|$ for noise rejection. The loopshaping design first converts the bounds on $|S(j\omega)|$ and $|T(j\omega)|$ into roughly equivalent bounds on $|L(j\omega)|$. Next, the various controller components are designed in a series of steps to satisfy these conditions. A reasonable process is to: i) Design a proportional controller to select the desired loop crossover frequency, ii) Use a low frequency or integral boost to achieve the desired low frequency gain for $|L(j\omega)|$, and iii) Use a high frequency roll-off to achieve the high frequency roll-off for $|L(j\omega)|$. A fourth step, discussed in the next chapter, may be required. This involves designing a lead controller to make the closed-loop more robust to model uncertainties.

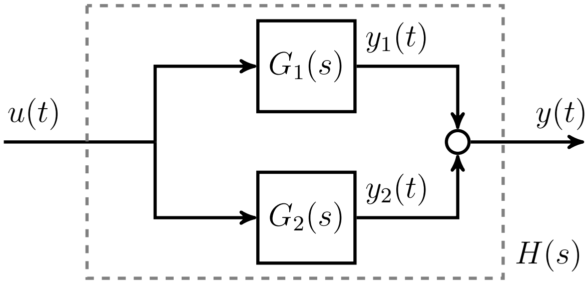
See posted Matlab example.

6.7 Appendix: Additional Results

Summary: This appendix provides additional details on the derivations of transfer functions for various system interconnections.

6.7.1 Derivation for Parallel Interconnections

Consider the parallel interconnection of two subsystems $G_1(s) = \frac{N_1(s)}{D_1(s)}$ and $G_2(s) = \frac{N_2(s)}{D_2(s)}$ as shown in Figure 6.1 (repeated below for convenience). The outputs of the subsystems are summed to obtain $y(t) = y_1(t) + y_2(t)$.



Parallel Interconnection of G_1 and G_2.

The goal is to obtain a model $H(s)$ from input $u(t)$ to output $y(t)$ in terms of the models for $G_1(s)$ and $G_2(s)$. The ODEs for the two subsystems have the following polynomial representations:

$$D_1(s)Y_1(s) = N_1(s)U(s) \tag{6.34}$$

$$D_2(s)Y_2(s) = N_2(s)U(s) \tag{6.35}$$

To derive a model for the parallel interconnection, first multiply Equation 6.34 by $D_2(s)$ and Equation 6.35 by $D_1(s)$:

$$D_2(s)D_1(s)Y_1(s) = D_2(s)N_1(s)U(s) \tag{6.36}$$

$$D_1(s)D_2(s)Y_2(s) = D_1(s)N_2(s)U(s) \tag{6.37}$$

Summing these equations yields:

$$(D_1(s)D_2(s))\, Y(s) = (D_2(s)N_1(s) + D_1(s)N_2(s))\, U(s) \tag{6.38}$$

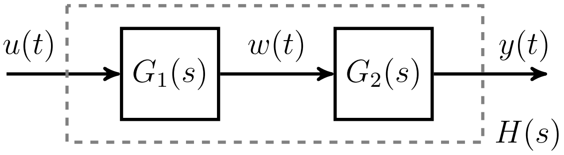
This result uses the fact that the order of multiplication can be freely interchanged, e.g. $D_2(s)D_1(s) = D_1(s)D_2(s)$. Thus the transfer function for the parallel interconnection is

$$H(s) = \frac{D_2(s)N_1(s) + D_1(s)N_2(s)}{D_1(s)D_2(s)} = \frac{N_1(s)}{D_1(s)} + \frac{N_2(s)}{D_2(s)} \tag{6.39}$$

The parallel interconnection of $G_1(s)$ and $G_2(s)$ is modeled by the sum $H(s) = G_1(s) + G_2(s)$.

6.7.2 Derivation for Serial Interconnections

Consider the cascade (serial) interconnection of two subsystems $G_1(s) = \frac{N_1(s)}{D_1(s)}$ and $G_2(s) = \frac{N_2(s)}{D_2(s)}$ as shown in Figure 6.2 (repeated below for convenience). The output of $G_1(s)$ is connected as the input to $G_2(s)$.



Cascade (Serial) Interconnection of G_1 and G_2.

The goal is to obtain a model $H(s)$ from input $u(t)$ to output $y(t)$ in terms of the models for $G_1(s)$ and $G_2(s)$. The ODEs for the two subsystems have the following polynomial representations:

$$D_1(s)W(s) = N_1(s)U(s) \tag{6.40}$$

$$D_2(s)Y(s) = N_2(s)W(s) \tag{6.41}$$

To derive a model for the serial interconnection, first multiply Equation 6.41 by $D_1(s)$ and substitute for $D_1(s)W(s)$ using Equation 6.40:

$$D_1(s)D_2(s)Y(s) = N_2(s)D_1(s)W(s) = N_2(s)N_1(s)U(s) \tag{6.42}$$

This again uses the fact that the order of multiplication can be interechanged, e.g. $N_2(s)D_1(s) = D_1(s)N_2(s)$. Thus the transfer function for the serial interconnection is

$$H(s) = \frac{N_2(s)N_1(s)}{D_2(s)D_1(s)} \tag{6.43}$$

The serial interconnection of $G_1(s)$ and $G_2(s)$ is modeled by the product $H(s) = G_2(s) \cdot G_1(s)$.

6.7.3 Derivation for Negative Feedback Interconnections

Consider the negative feedback interconnection of two subsystems $G_1(s) = \frac{N_1(s)}{D_1(s)}$ and $G_2(s) = \frac{N_2(s)}{D_2(s)}$ as shown in Figure 6.3 (repeated below for convenience).

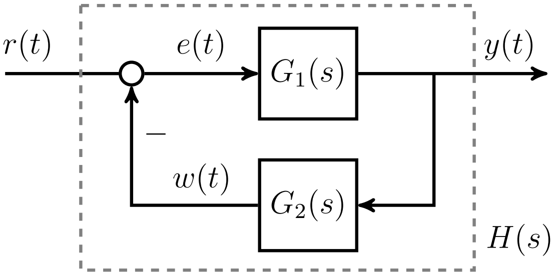
A polynomial representation for this interconnection is:

$$D_1(s)Y(s) = N_1(s)\left(R(s) - W(s)\right) \tag{6.44}$$

$$D_2(s)W(s) = N_2(s)Y(s) \tag{6.45}$$

To derive a model for the interconnection, first multiply Equation 6.44 by $D_2(s)$:

$$D_2(s)D_1(s)Y(s) = D_2(s)N_1(s)\left(R(s) - W(s)\right) \tag{6.46}$$



Negative Feedback Interconnection of G_1 and G_2.

Next, use Equation 6.45 to substitute for $D_2(s)W(s)$ and collect all terms involving $Y(s)$:

$$(D_2(s)D_1(s) + N_2(s)N_1(s))\,Y(s) = D_2(s)N_1(s)R(s) \tag{6.47}$$

Thus the transfer function for the feedback interconnection is

$$H(s) = \frac{D_2(s)N_1(s)}{D_2(s)D_1(s) + N_2(s)N_1(s)} \tag{6.48}$$

The informal sketch in Section 6.1.4 yielded the following form for the transfer function:

$$H(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)} \tag{6.49}$$

This expression in Equation 6.49 can be equivalently rewritten as follows:

$$\begin{aligned} H(s) &= \frac{N_1(s)}{D_1(s)} \frac{1}{1 + \frac{N_1(s)N_2(s)}{D_1(s)D_2(s)}} \\ &= \frac{N_1(s)}{D_1(s)} \frac{D_1(s)D_2(s)}{D_1(s)D_2(s) + N_1(s)N_2(s)} \\ &= \frac{D_1(s)}{D_1(s)} \frac{N_1(s)D_2(s)}{D_1(s)D_2(s) + N_1(s)N_2(s)} \end{aligned}$$

This is the same as the result from the formal derivation (Equation 6.48) except for the additional factor of $\frac{D_1(s)}{D_1(s)}$. Thus $\frac{G_1(s)}{1+G_1(s)G_2(s)}$ represents the negative feedback interconnection but with additional poles and zeros at the roots of $D_1(s) = 0$. Canceling these common poles and zeros yields the correct (minimal) representation.

Chapter 7

Stability Margins and Robustness

This chapter introduces techniques to assess the stability and robustness of a feedback system based on the properties of the loop transfer function $L(s)$. First, Nyquist plots are introduced. A Nyquist plot is a single plot of the frequency response $L(j\omega)$. It consists of the imaginary part $Im\{L(j\omega)\}$ on the vertical axis versus the real part $Re\{L(j\omega)\}$ on the horizontal axis. The Nyquist stability theorem relates the stability of the closed-loop to properties of the Nyquist plot for $L(s)$. Next, various safety factors are introduced to account for model uncertainty including gain, phase, time delay, and disk margins. Specifically, model uncertanity can cause the Nyquist curve to change from having the "correct" number of encirclements of the -1 point (stable closed-loop) to having an "incorrect" number of encirclements (unstable closed-loop). The robustness margins measure how close the Nyquist plot of the loop $L(s)$ approaches the critical -1. Finally, a basic result is presented to demonstrate that the loopshaping design procedure will yield a stable closed-loop with good performance and robustness. The chapter concludes with additional examples of loopshaping design.

7.1 Nyquist Plots

Summary: A Nyquist plot is a single plot of the frequency response $G(j\omega)$. It consists of the imaginary part $Im\{G(j\omega)\}$ on the vertical axis versus the real part $Re\{G(j\omega)\}$ on the horizontal axis. The convention is to draw this plot for both $\omega \geq 0$ and $\omega < 0$. Nyquist plots are used to understand the stability and robustness of a feedback system. Nyquist plots for first order systems (with or without a zero) are simply circles in the complex plane.

7.1.1 An Overview of Nyquist Plots

A *Nyquist Plot* is a common tool used to understand the stability and robustness of a feedback system. As shown in Section 5.1, the steady-state sinusoidal response of a linear system can be expressed in terms of the magnitude $|G(j\omega)|$ and phase $\angle G(j\omega)$ of the transfer function. Recall that a Bode plot displays both the magnitude and phase versus frequency on two separate subplots. A Nyquist plot, named after Harry Nyquist, displays the frequency response information in a different format. It consists of a single plot of the imaginary part $Im\{G(j\omega)\}$ on the vertical axis versus the real part $Re\{G(j\omega)\}$ on the horizontal axis. The convention is to draw the Nyquist plot for both $\omega \geq 0$ and $\omega < 0$. Nyquist plots can be generated with the Matlab function `nyquist`. In particular, the syntax `nyquist(G)` generates the Nyquist plot of a system `G`. The alternative syntax `[Gre,Gim,w]=nyquist(G)` returns the real part, imaginary part, and frequency (in $\frac{rad}{sec}$). Additional options are available and details can be found in the help and documentation. A simple example is given below.

Example 7.1. $G(s) = \frac{2}{s+4}$ is the transfer function for a stable, first order system. This transfer function was studied in Example 5.3 and the corresponding Bode plot was shown in Figure 5.3. The Nyquist plot for $G(s)$ is shown on the left of Figure 7.1. A similar Nyquist plot can be created with the following Matlab commands:

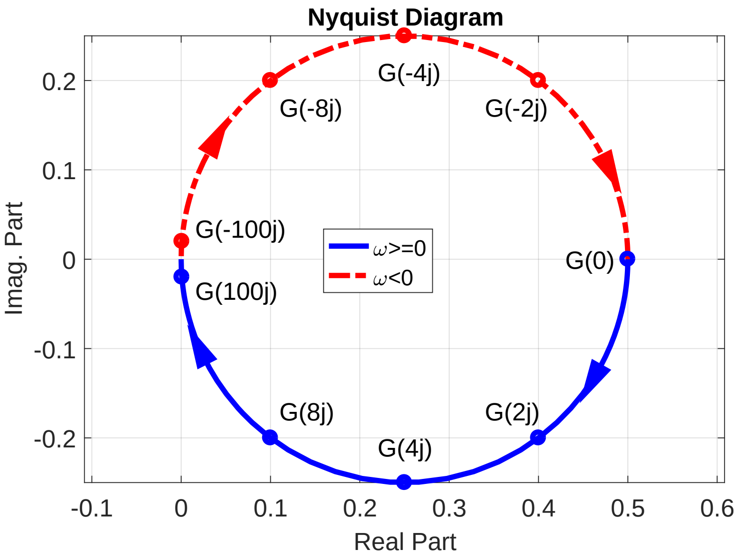
```
>> G = tf(2,[1 4]);
>> nyquist(G);
```

Figure 7.1 also shows the tabulated frequency response data (polar form, real part, and imaginary part) at several frequencies. The solid circles on the Nyquist plot highlight the data at these frequencies. The Nyquist plot for this system consists of a circular loop in the clockwise direction with the top half (dashed red) corresponding to $\omega < 0$ and the bottom half (solid blue) corresponding to $\omega \geq 0$. The curve (for $\omega < 0$) starts at $G(-\infty) = 0$ and moves upward in the complex plane toward $G(-4j) = 0.25 + 0.25j$. The curve bends back downward to $G(0) = 0.5$. The curve (for $\omega > 0$) continues downward in the complex plane toward $G(4j) = 0.25 - 0.25j$. The curve finally ends by moving back upward to $G(+\infty) = 0$ as $\omega \to +\infty$.

△

Note that the curve for $\omega < 0$ is simply the complex conjugate of the curve for $\omega > 0$. Specifically, if $G(j\omega) = c_R + jc_I$ (where c_R and c_I are the real and imaginary parts) then $G(-j\omega) = c_R - jc_I$. This property holds for the Nyquist plot of any transfer function.* Thus the Nyquist plot for $\omega < 0$ is simply the mirror image about the real axis of the plot for $\omega > 0$.

*Consider a generic transfer function $G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0}$. Let $\overline{G(j\omega)}$ denote the complex conjugate of



ω $(\frac{rad}{sec})$	$G(j\omega)$ Polar	Real Part	Imag. Part
$-\infty$	$0e^{\frac{\pi}{2}j}$	0	0
-100	$0.02e^{1.531j}$	0.0008	0.02
-8	$0.224e^{1.107j}$	0.10	0.20
-4	$0.354e^{0.785j}$	0.25	0.25
-2	$0.447e^{0.464j}$	0.40	0.20
0	$0.500e^{0j}$	0.500	0.000
2	$0.447e^{-0.464j}$	0.40	-0.20
4	$0.354e^{-0.785j}$	0.25	-0.25
8	$0.224e^{-1.107j}$	0.10	-0.20
100	$0.02e^{-1.531j}$	0.0008	-0.02
$+\infty$	$0e^{-\frac{\pi}{2}j}$	0	0

Figure 7.1: Nyquist plot (left) for $G(s) = \frac{2}{s+4}$ and data evaluated at several $s = j\omega$ (right). The solid circles on the Nyquist plot highlight the data at several frequencies.

7.1.2 Nyquist Plots: First-Order Systems

This section will briefly describe how to sketch a Nyquist plot for a first order system. Consider the following system:

$$\dot{y}(t) + a_0 y(t) = b_0 u(t) \tag{7.2}$$

Assume only that $a_0 \neq 0$. The transfer function for this system is $G(s) = \frac{b_0}{s+a_0}$. The real and imaginary parts of the frequency response are:

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \cdot \frac{-j\omega + a_0}{-j\omega + a_0} = \underbrace{\frac{b_0 a_0}{a_0^2 + \omega^2}}_{Re\{G(j\omega)\}} + j \underbrace{\frac{-b_0 \omega}{a_0^2 + \omega^2}}_{Im\{G(j\omega)\}} \tag{7.3}$$

It can be shown, after some algebraic manipulation, that the real and imaginary parts satisfy the following equation:

$$\left(Re\{G(j\omega)\} - \frac{b_0}{2a_0} \right)^2 + Im\{G(j\omega)\}^2 = \left(\frac{b_0}{2a_0} \right)^2 \tag{7.4}$$

This is the equation for a circle in the complex plane with radius $\frac{b_0}{2a_0}$ and center on the real axis at $\frac{b_0}{2a_0}$. The Nyquist plot circle passes through the following points for $\omega \geq 0$:

$G(j\omega)$. Use the fact that the ODE coefficients $(a_0, \ldots, a_n, b_0, \ldots, b_m)$ are real numbers to show:

$$\overline{G(j\omega)} = \frac{b_m \overline{(j\omega)^m} + \cdots + b_1 \overline{(j\omega)} + b_0}{a_n \overline{(j\omega)}^n + \cdots + a_1 \overline{(j\omega)} + a_0} = G(-j\omega) \tag{7.1}$$

- **Low (DC) Frequency:** If $\omega = 0$ then $G(j0) = \frac{b_0}{a_0}$. This is a purely real number and hence $G(0)$ lies on the real axis of the Nyquist plot.

- **High Frequency:** As $\omega \to +\infty$, $G(j\omega) \approx j\frac{-b_0}{\omega}$. This is purely imaginary. Thus as $\omega \to +\infty$ the Nyquist plot $G(j\omega)$ tends to zero along the negative imaginary axis.

- **Corner Frequency:** Recall that the corner frequency for this first order system is $\omega = |a_0| > 0$. The Nyquist plot at this frequency is $G(j|a_0|) = \frac{b_0}{2a_0} - j\frac{b_0}{2|a_0|}$.

As noted above, the Nyquist plot for $G(s) = \frac{b_0}{s+a_0}$ is a circle. For positive frequencies the circle passes through the points $G(0)$, $G(\infty)$, and $G(|a_0|j)$. For negative frequencies, the circle can be determined by conjugation, i.e. $G(-j\omega) = \overline{G(j\omega)}$. The direction of the curve will be important when we use the Nyquist plot for stability and robustness analysis. In particular, the Nyquist curve is oriented moving from $\omega = -\infty$ toward $\omega = +\infty$. The left plot in Figure 7.2 shows an example Nyquist curve for $G(s) = \frac{2}{s-4}$. The curve for $\omega \geq 0$ passes through $G(0) = -0.5$, $G(j\infty) = 0$, and $G(4j) = -0.25 - 0.25j$. The curve for $\omega < 0$ is a mirror image about the real axis. This curve is a circle with a counter-clockwise orientation.

A Nyquist plot is also a circle for a first order system with a zero:

$$\dot{y}(t) + a_0 y(t) = b_1 \dot{u}(t) + b_0 u(t) \tag{7.5}$$

Again, assume only that $a_0 \neq 0$. The transfer function for this system is $G(s) = \frac{b_1 s + b_0}{s+a_0}$. The frequency response can be written as follows:

$$G(j\omega) = \frac{b_1(j\omega + a_0) + (b_0 - b_1 a_0)}{j\omega + a_0} = b_1 + \frac{(b_0 - b_1 a_0)}{j\omega + a_0} \tag{7.6}$$

The second term has the form of a first-order frequency response with no zero. This term corresponds to a circle in a Nyquist plot. The first term is a constant that simply shifts the circle along the real axis. **Thus, the Nyquist curve for any first-order system $G(s) = \frac{b_1 s + b_0}{s+a_0}$ (with $a_0 \neq 0$) is simply a circle passing through $G(0)$, $G(\infty)$ and $G(j|a_0|)$.** The right plot in Figure 7.2 shows another example Nyquist curve for $G(s) = \frac{3s+5}{s-2}$. The curve for $\omega \geq 0$ passes through $G(0) = -2.5$, $G(j\infty) = 3$, and $G(2j) = 0.25 - 2.75j$. The curve for $\omega < 0$ is a mirror image about the real axis. This curve is a circle with a counter-clockwise orientation.

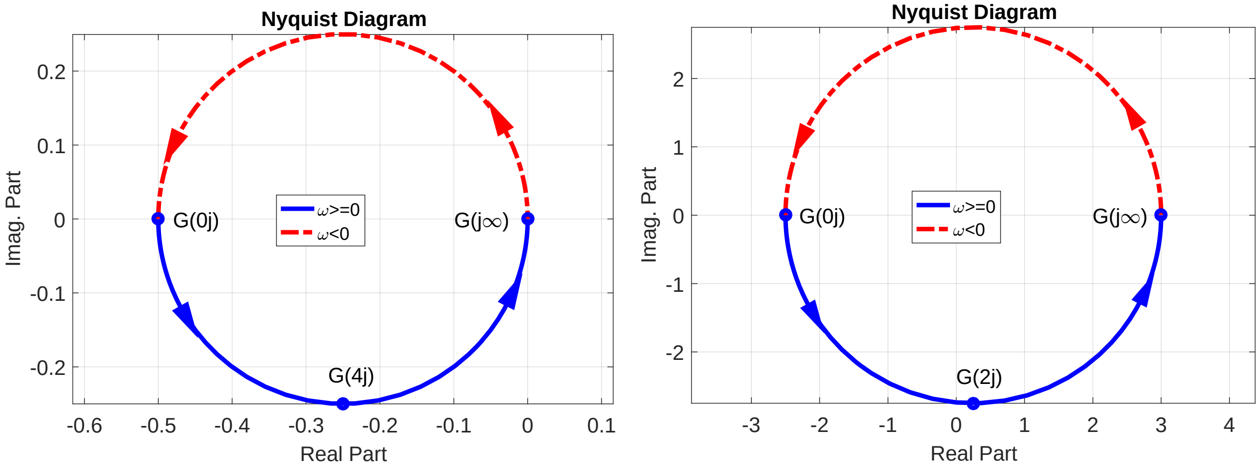


Figure 7.2: Nyquist plot for $G(s) = \frac{2}{s-4}$ (left) and $G(s) = \frac{3s+5}{s-2}$ (right).

7.2 Cauchy's argument principle

Summary: Nyquist plots will be used to state a theorem concerning the stability of a feedback system. A proper understanding of the Nyquist stability theorem requires one result from complex analysis: Cauchy's Argument Principle. This principle states that a transfer function $G(s)$ evaluated on a simple, closed curve will encircle the origin $N_z - N_p$. Here N_p and N_z are the number of poles and zeros of $G(s)$ that lie inside the closed curve.

Nyquist plots will be used to state a theorem concerning the stability of a feedback system. A proper understanding of the Nyquist stability theorem requires one result from complex analysis: Cauchy's Argument Principle. The basic idea will be sketched in this section but additional details can be found in textbooks on complex analysis [3]. To begin, let $G(s)$ be a transfer function for a system. In addition, let Γ be a simple, closed curve in the complex plane. A "simple" curve is one that does not intersect itself. The notation $G(\Gamma)$ is the curve obtained by mapping each complex number $s_0 \in \Gamma$ to another complex number $G(s_0)$. In general, $G(\Gamma)$ will be closed but need not be simple.

As an example, consider the specific curve Γ_R defined by three pieces as shown on the left in Figure 7.3. The first piece (red dotted) starts at $s_1 = -jR$ and rises along the imaginary axis to $s_2 = 0$. The second piece (blue solid) continues along the imaginary axis from $s_2 = 0$ to $s_3 = +jR$. The third piece (green dashed) follows a semicircular arc of radius R from $s_3 = +jR$ to $s_4 = +R$ and back to $s_1 = -jR$. This third piece corresponds to $s = Re^{j\phi}$ as ϕ goes from $\frac{\pi}{2}$ to $-\frac{\pi}{2}$. The curve Γ_R is shown for the specific value of $R = 50$. It is a simple, closed curve in the clockwise direction. The right side of Figure 7.3 shows the mapping $G(\Gamma_R)$ defined with the transfer function $G(s) = \frac{2}{s+4}$. $G(\Gamma_R)$ is also a closed curve and, for this transfer function, it is also simple. Note that $G(\Gamma_R)$ is similar to the Nyquist plot of $G(s) = \frac{2}{s+4}$ shown in Figure 7.1. The main difference is that $G(\Gamma_R)$ includes a small indentation ($G(s_3)$ to $G(s_4)$ to $G(s_1)$) due to the (green dashed) semicircular piece of Γ_R. The radius $R = 50$ is large and, as a result, $|G(s)| \approx 0$ along this piece. $G(\Gamma_R)$ **converges to the Nyquist plot of** $G(s)$ **as** $R \to \infty$**, .**

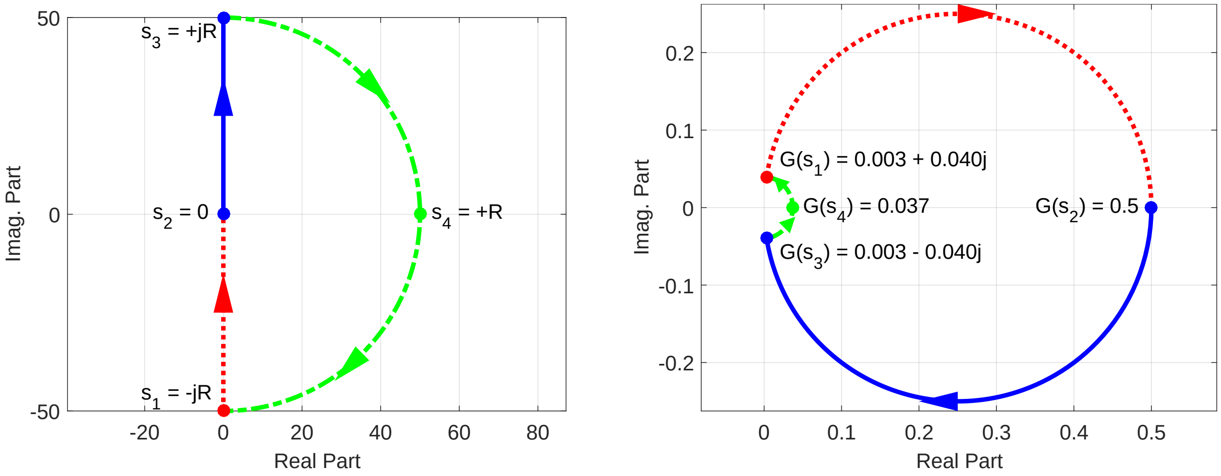


Figure 7.3: A simple, closed curve Γ_R defined with $R = 50$ (left) and its mapping $G(\Gamma_R)$ defined with $G(s) = \frac{2}{s+4}$ (right).

Let N_p denote the number of poles of $G(s)$ inside the curve Γ.[†] In addition, let N_z denote the number of zeros of $G(s)$ inside of Γ. Cauchy's argument principle can now be stated:

Fact 7.1 (Cauchy's Argument Principle)**.** Assume Γ does not pass through any poles or zeros of $G(s)$. Then the closed curve $G(\Gamma)$ encircles the origin $N_z - N_p$ times. If $N_z - N_p > 0$ then $G(\Gamma)$ encircles the origin clockwise. If $N_z - N_p < 0$ then $G(\Gamma)$ encircles the origin counter-clockwise.

Note that if Γ passes through a zero of $G(s)$ then $G(\Gamma)$ intersects the origin. If Γ passes through a pole of $G(s)$ then $G(\Gamma)$ diverges to ∞. Cauchy's argument principle does not apply in such cases where Γ passes through a zero or pole of $G(s)$. The proof of Cauchy's argument principle can be found in textbooks on complex analysis [3]. A few simple examples will be given here to demonstrate this principle.

Example 7.2. Figure 7.4 (left) shows the previously defined curve Γ_R with $R = 2$. The mapping $G(\Gamma_R)$ with $G(s) = s - 1$ is shown on the right of Figure 7.4. $G(\Gamma_R)$ has the same shape as Γ_R but it is shifted to the left by one unit. For example, $s_2 = 0$ gets mapped to $G(s_2) = -1$. $G(s)$ has a single zero $(s = +1)$ inside Γ_R but no poles inside Γ_R. The zero is labeled by the circle on the left plot of Figure 7.4. Hence $(N_z, N_p) = (+1, 0)$ and Cauchy's argument principle states that $G(\Gamma_R)$ will encircle the origin once in the clockwise direction $(N_z - N_p = 1 > 0)$.

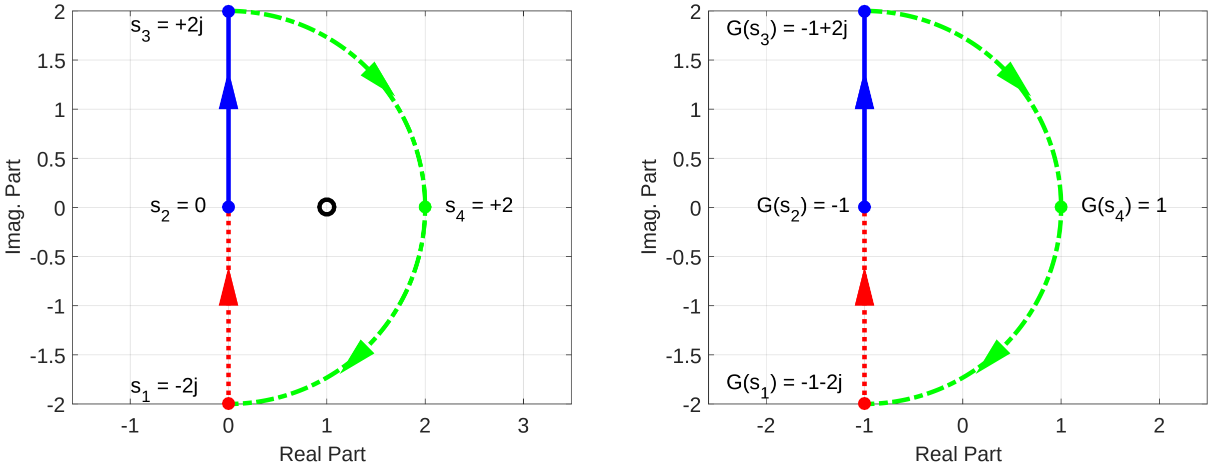


Figure 7.4: A simple, closed curve Γ_R defined with $R = 2$ (left) and its mapping $G(\Gamma_R)$ defined with $G(s) = s - 1$ (right).

△

Example 7.3. Figure 7.5 (left) shows the previously defined curve Γ_R with $R = 2$. The mapping $G(\Gamma_R)$ with $G(s) = s + 1$ is shown on the right of Figure 7.5. $G(\Gamma_R)$ has the same shape as Γ_R but it is shifted to the right by one unit. For example, $s_2 = 0$ gets mapped to $G(s_2) = +1$. $G(s)$ has a single zero $(s = -1)$ but it is outside Γ_R. The zero is labeled by the circle on the left plot of Figure 7.5. Thus $G(s)$ has no zeros or poles inside Γ_R, i.e. $(N_z, N_p) = (0, 0)$. By Cauchy's argument principle, $G(\Gamma_R)$ will not encircle the origin $(N_z - N_p = 0)$.

[†] Here Γ denotes any simple, closed curve and not necessarily the specific curve Γ_R shown in Figure 7.3.

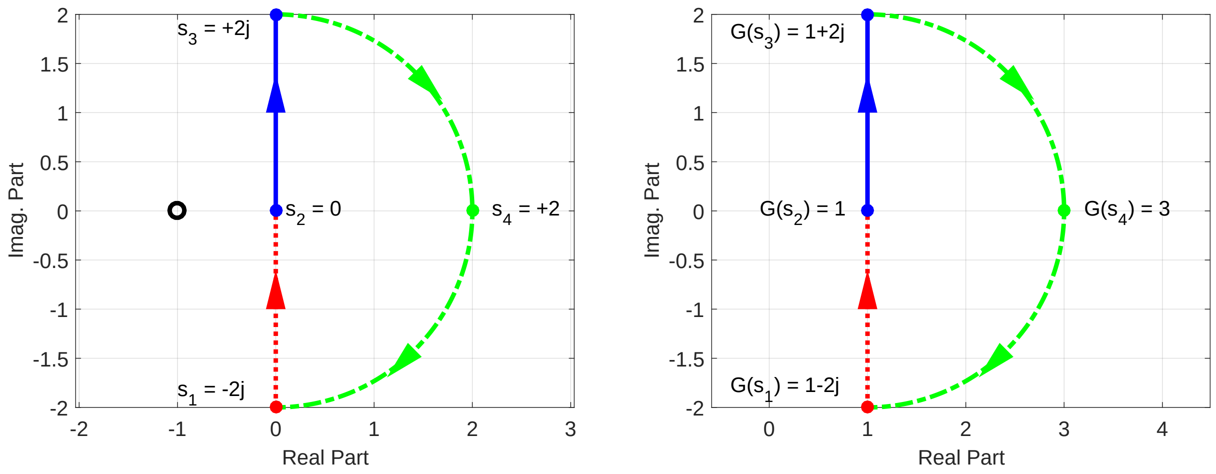


Figure 7.5: A simple, closed curve Γ_R defined with $R = 2$ (left) and its mapping $G(\Gamma_R)$ defined with $G(s) = s + 1$ (right).

△

Example 7.4. Figure 7.6 (left) shows the previously defined curve Γ_R with $R = 3$. The mapping $G(\Gamma_R)$ with $G(s) = s^2 - 3s + 2$ is shown on the right of Figure 7.6. $G(\Gamma_R)$ maps into a curve that intersects with itself, i.e. $G(\Gamma_R)$ is closed but not simple. The transfer function can be factored as $G(s) = (s - 1)(s - 2)$ demonstrating that $G(s)$ has zeros at $s = +1$ and $s = +2$. Both zeros are labeled by circles on the left plot of Figure 7.6. Thus $G(s)$ has two zeros and no poles inside Γ_R, i.e. $(N_z, N_p) = (2, 0)$. By Cauchy's argument principle, $G(\Gamma_R)$ encircles the origin twice in the clockwise direction $(N_z - N_p = 2 > 0)$.

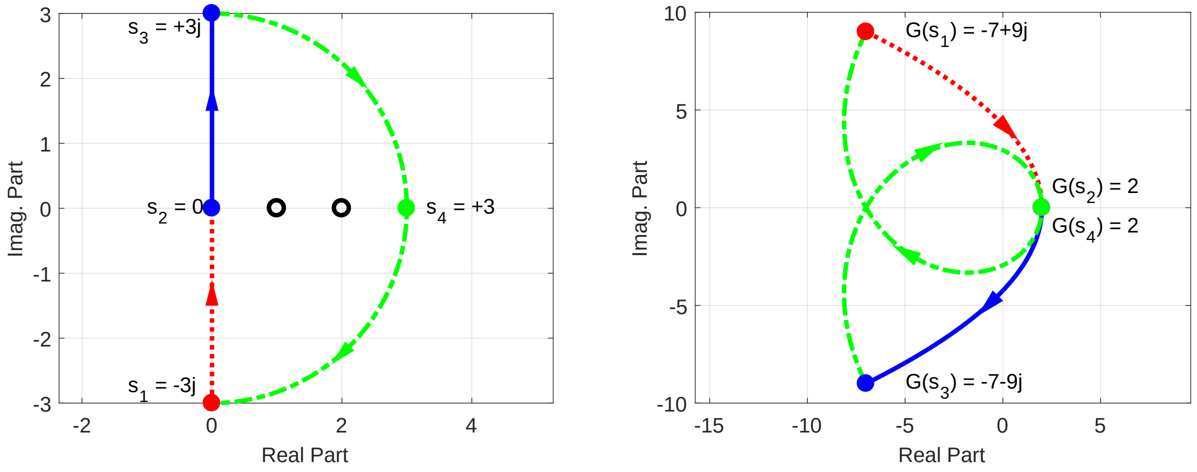


Figure 7.6: A simple, closed curve Γ_R defined with $R = 3$ (left) and its mapping $G(\Gamma_R)$ defined with $G(s) = s^2 - 3s + 2$ (right).

△

Example 7.5. Figure 7.7 (left) shows the previously defined curve Γ_R with $R = 3$. The mapping $G(\Gamma_R)$ with $G(s) = \frac{2s+4}{s-1}$ is shown on the right of Figure 7.7. $G(s)$ has a zero at $s = -2$ outside of Γ_R. It also has a pole at $s = +1$ inside of Γ_R. The zero and pole are labeled by a circle and x, respectively, on the left plot of Figure 7.7. Thus $(N_z, N_p) = (0, 1)$ and by Cauchy's argument principle, $G(\Gamma_R)$ encircles the origin once in the counter-clockwise direction ($N_z - N_p = -1 < 0$).

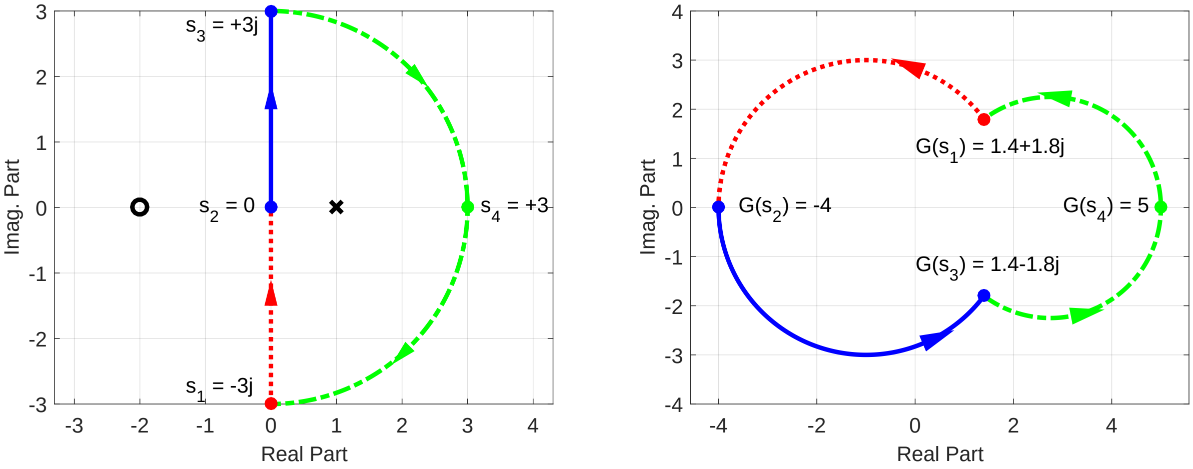


Figure 7.7: A simple, closed curve Γ_R defined with $R = 3$ (left) and its mapping $G(\Gamma_R)$ defined with $G(s) = \frac{2s+4}{s-1}$ (right).

△

7.3 Nyquist Stability Condition

Summary: The value $s = -1$ in the complex plane is a critical point on a Nyquist plot. The closed-loop is unstable if the Nyquist curve of the loop $L(s)$ passes through this critical point. Moreover, the Nyquist theorem states that the closed-loop is stable if and only if the Nyquist curve of $L(s)$ encircles the $s = -1$ point the "correct" number of times. The "correct" number of times is equal to the number of RHP poles of the loop $L(s)$.

7.3.1 The Critical -1 Point

The value $s = -1$ in the complex plane is a critical point on a Nyquist plot. This subsection briefly describes a relation between this critical -1 point and closed-loop stability. Consider the feedback system shown in Figure 7.8. Recall that the zeros of $1 + L(s)$ are poles of the closed-loop system (assuming the realization for $L(s)$ is minimal). For example, the sensitivity function is $S(s) = \frac{1}{1+L(s)}$. If $1 + L(s_0) = 0$ for some complex s_0 then $|S(s_0)| = \infty$, i.e. s_0 is a pole of $S(s)$.

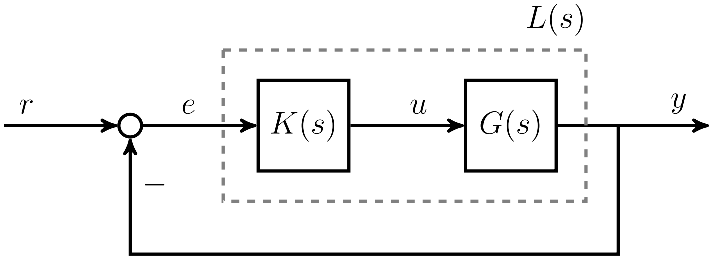


Figure 7.8: General Feedback System

Suppose that the Nyquist curve of the loop $L(s)$ passes through $s = -1$, i.e. there is some frequency ω_0 such that $L(j\omega_0) = -1$. This can be rewritten as $1 + L(j\omega_0) = 0$. In other words, $1 + L(s)$ has a zero at $s = j\omega_0$ and, by the discussion above, the closed-loop has pole at $s = j\omega_0$. **Hence if the Nyquist curve of $L(s)$ passes through the critical point $s = -1$ then the closed-loop has a pole on the imaginary axis and therefore the closed-loop is unstable.** This result can also be interpreted using sinusoidal frequency response. In particular, $L(j\omega_0) = -1$ implies that $r(t) = 0$ and $e(t) = \cos(\omega_0 t)$ are a (steady-state) solution for the feedback system. This follows because $L(j\omega_0) = -1$ means $|L(j\omega_0)| = 1$ and $\angle L(j\omega_0) = \pi\, rad$. Hence $e(t) = \cos(\omega_0 t)$ causes the steady state output $y(t) = \cos(\omega_0 t + \pi) = -\cos(\omega_0 t)$. This output, through negative feedback, satisfies $e(t) = -y(t)$, i.e. $e(t)$ and $y(t)$ can be in a sustained oscillation even if $r(t) = 0$. As noted above $|L(j\omega_0)| = -1$ also means that $|S(j\omega_0)| = \infty$. In other words, the transfer function from reference to error $S(s)$ has infinite gain at frequency ω_0. This infinite gain can be related to sinusoidal response solution with $r(0) = 0$ and $e(t) = \cos(\omega_0 t)$. Specifically, the reference can be interpreted as a sinusoid with amplitude zero, i.e. $r(t) = 0\ \cos(\omega_0 t)$, while the error is a sinusoid with amplitude one, i.e. $e(t) = 1\ \cos(\omega_0 t))$. Hence the gain from reference to error at frequency ω_0 is infinite $(= \frac{1}{0})$ as predicted by $|S(j\omega_0)| = \infty$.

7.3.2 Nyquist Theorem

The value $s = -1$ also plays a critical role in the Nyquist stability condition. Consider the feedback system shown in Figure 7.8. Recall that the feedback system is defined to be stable if all possible transfer functions in the system (r to e, etc) are stable (Definition 6.1). Moreover, recall that the feedback system is unstable if there is a RHP pole/zero cancellation between $G(s)$ and $K(s)$ (Section 6.2.3). Thus for the remainder of the section it is assumed that there are no such RHP pole/zero cancellations in the loop transfer function $L(s) = G(s)K(s)$. With this assumption, the feedback system is stable if and only if $1 + G(s)K(s)$ has no zeros in the RHP (Fact 6.1). The Nyquist theorem, given below, provides a condition for stability of the closed-loop system in terms of the Nyquist plot of the (open) loop $L(s)$. The following notation is used to state the Nyquist theorem:

- P_{CL}: This denotes the number of poles of the closed-loop system in the RHP (including the imaginary axis).
- P_{OL} : This denotes the number of poles of the open-loop transfer function $L(s)$ in the RHP (including the imaginary axis).
- N_{CCW}: This denotes the number of times the Nyquist curve of $L(s)$ encircles the critical -1 point in the counter-clockwise direction.

The Nyquist theorem is first stated for the simpler case where $L(s)$ has no poles on the imaginary axis.

Fact 7.2 (Nyquist Theorem)**.** Assume $L(s)$ has no poles on the imaginary axis. Then $P_{CL} = P_{OL} - N_{CCW}$. Thus the closed-loop is stable ($P_{CL} = 0$) if and only if $N_{CCW} = P_{OL}$.

The benefit of the Nyquist condition is that closed-loop stability can be determined from a Nyquist plot of the open loop transfer function $L(s)$. This will be useful for control design via the loopshaping process.

The proof of the Nyquist theorem follows from an application of Cauchy's argument principle. Define the transfer function $H(s) = 1 + L(s)$ and consider the simple, closed curve Γ_R shown on the left in Figure 7.3. Let N_p and N_z denote the number of poles and zeros of $H(s)$ inside the curve Γ_R. Cauchy's argument principle states that $H(\Gamma_R)$ has $N_z - N_p$ (clockwise) encirclements of the origin. The sign convention can be flipped to equivalently state this principle as: $H(\Gamma_R)$ has $N_p - N_z$ (counter-clockwise) encirclements of the origin. Three relations are needed to complete the proof:

1. First, note that $L(s) = H(s) - 1$ so that $L(\Gamma_R)$ is the curve $H(\Gamma_R)$ shifted to the left by one unit. Thus $H(\Gamma_R)$ encircles the origin counter-clockwise if and only if $L(\Gamma_R)$ encircles -1 counter-clockwise. As $R \to \infty$, $L(\Gamma_R)$ converges to the Nyquist curve of $L(s)$. Hence Cauchy's argument principle implies $N_{CCW} = N_p - N_z$.
2. Next, note that Γ_R contains the entire RHP as $R \to \infty$. Hence if R is sufficiently large then the RHP zeros of $H(s) = 1 + L(s)$ are precisely the closed-loop RHP poles, i.e. $N_z = P_{CL}$.

3. Finally, the RHP poles of $H(s)$ are precisely the RHP poles of the open-loop $L(s)$, i.e. $N_p = P_{OL}$. This follows because $L(s)$ has no RHP pole/zero cancellations and hence $|H(s_0)| = \infty$ at some s_o in the RHP if and only if $|L(s_0)| = \infty$.

Combining these three relations yields: $N_{CCW} = P_{OL} - P_{CL}$. This can be re-arranged to conclude $P_{CL} = P_{OL} - N_{CCW}$.

Several examples are given next to demonstrate the Nyquist stability theorem.

Example 7.6. The left plot of Figure 7.9 shows the Nyquist plots for $L_1(s) = \frac{2}{s+4}$ and $L_2(s) = \frac{-2s+2}{s+4}$. The loop transfer function $L_1(s)$ is stable so $P_{OL} = 0$. The Nyquist curve for $L_1(s)$ does not encircle the critical -1 point. Hence $N_{CCW} = 0$ and by the Nyquist theorem $P_{CL} = P_{OL} - N_{CCW} = 0$. Thus the feedback system with $L_1(s)$ is stable. This result can be verified by directly computing a closed-loop transfer function. For example, $S_1(s) = \frac{1}{1+L_1(s)} = \frac{s+4}{s+6}$ is stable.

The loop transfer function $L_2(s)$ is also stable so $P_{OL} = 0$. In addition, the Nyquist curve for $L_2(s)$ encircles the critical -1 point once in the clockwise direction. Hence $N_{CCW} = -1$ and by the Nyquist theorem $P_{CL} = P_{OL} - N_{CCW} = +1$. Thus the feedback system with loop $L_2(s)$ has one pole in the RHP and is unstable. This result can be verified by directly computing a closed-loop transfer function. For example, $S_2(s) = \frac{1}{1+L_2(s)} = \frac{-s-4}{s-6}$ has one pole in the RHP as predicted.

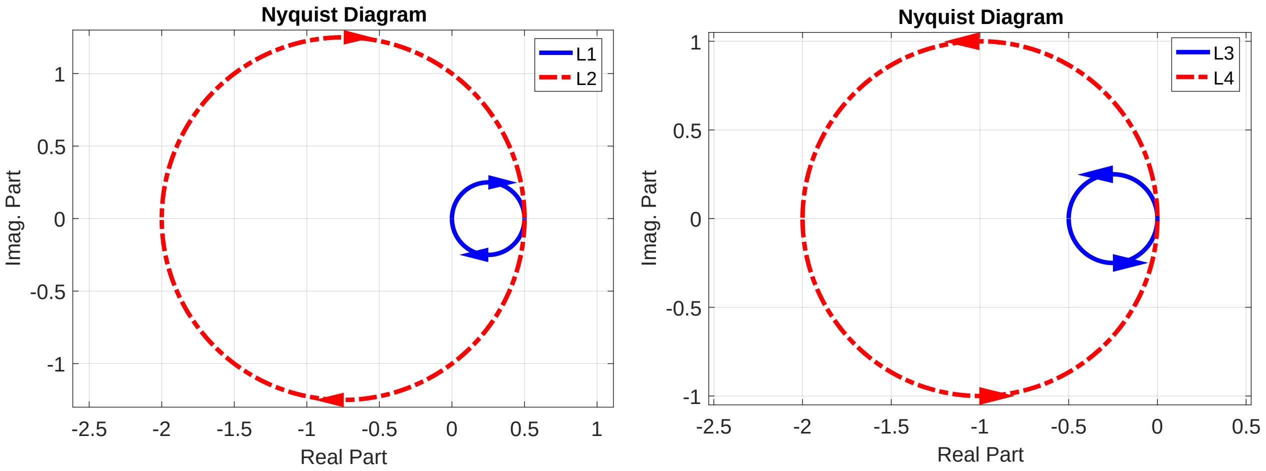


Figure 7.9: Left: Nyquist plots for $L_1(s) = \frac{2}{s+4}$ and $L_2(s) = \frac{-2s+2}{s+4}$. Right: Nyquist plots for $L_3(s) = \frac{2}{s-4}$ and $L_4(s) = \frac{8}{s-4}$.

$\triangle$

Example 7.7. The right plot of Figure 7.9 shows the Nyquist plots for $L_3(s) = \frac{2}{s-4}$ and $L_4(s) = \frac{8}{s-4}$. The loop transfer function $L_3(s)$ is unstable with a single pole in the RHP ($P_{OL} = 1$). The Nyquist curve for $L_3(s)$ does not encircle the critical -1 point. Hence $N_{CCW} = 0$ and by the Nyquist theorem $P_{CL} = P_{OL} - N_{CCW} = 1$. Thus the feedback system with $L_3(s)$ is unstable. This result can be verified by directly computing a closed-loop transfer function. For example, $S_3(s) = \frac{1}{1+L_3(s)} = \frac{s-4}{s-2}$ has one pole in the RHP as predicted.

The loop transfer function $L_4(s)$ is also unstable with a single pole in the RHP ($P_{OL} = 1$). In addition, the Nyquist curve for $L_4(s)$ encircles the critical -1 point once in the counter clockwise direction. Hence $N_{CCW} = 1$ and by the Nyquist theorem $P_{CL} = P_{OL} - N_{CCW} = 0$. Thus the feedback system with $L_4(s)$ has no poles in the RHP and is stable. This result can be verified by directly computation. For example, $S_4(s) = \frac{1}{1+L_4(s)} = \frac{s-4}{s+4}$ is stable. △

Example 7.8. The left plot of Figure 7.10 shows the Nyquist plot for the higher order system $L_5(s) = \frac{2}{s-1} \cdot \frac{100}{s^2+5s+100}$. This loop transfer function is unstable with a single pole in the RHP ($P_{OL} = 1$). The Nyquist curve for $L_5(s)$ encircles the critical -1 point once in the counter clockwise direction. Hence $N_{CCW} = 1$ and by the Nyquist theorem $P_{CL} = P_{OL} - N_{CCW} = 0$. Thus the feedback system with $L_5(s)$ is stable.

The right plot of Figure 7.10 shows another Nyquist plot for the higher order system $L_6(s) = \frac{1}{(s+1)^5}$. This loop transfer function is stable ($P_{OL} = 0$). The Nyquist curve for $L_6(s)$ does not encircle the critical -1 point. Hence $N_{CCW} = 0$ and by the Nyquist theorem $P_{CL} = P_{OL} - N_{CCW} = 0$. Thus the feedback system with $L_6(s)$ is stable.

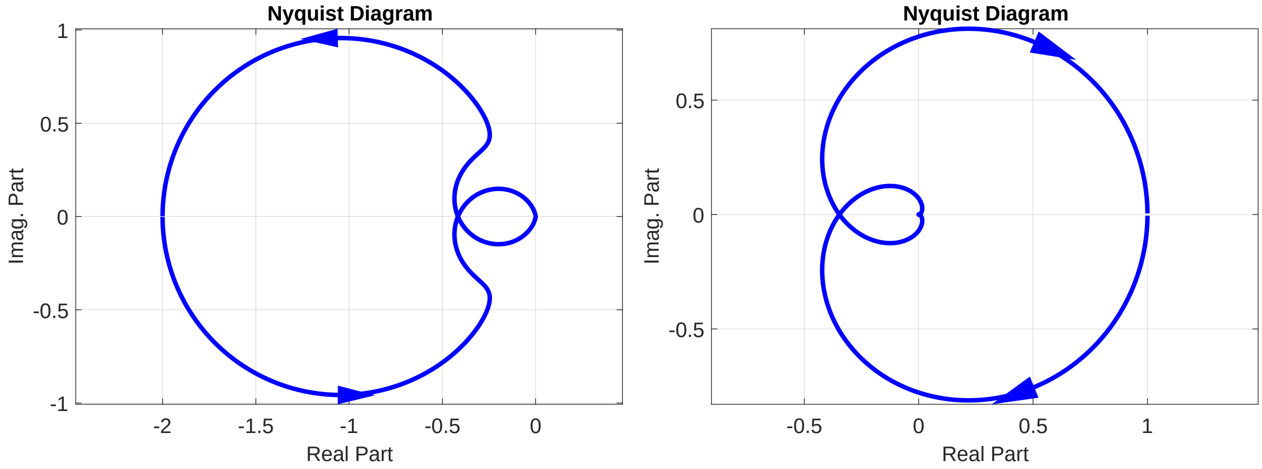


Figure 7.10: Nyquist plots for $L_5(s) = \frac{2}{s-1} \cdot \frac{100}{s^2+5s+100}$ (left) and $L_6(s) = \frac{1}{(s+1)^5}$ (right).

△

7.3.3 Extended Nyquist Theorem

The Nyquist theorem (Fact 7.2) was stated above under the assumption that $L(s)$ has no poles on the imaginary axis. This subsection will extend the result for the case where $L(s)$ has imaginary axis poles. This is an important extension. For example, the use of integral control leads to a loop $L(s)$ with a pole at $s = 0$. The main technical issue is that the Nyquist plot diverges to infinity if $L(s)$ has a pole on the imaginary axis. As a consequence, Cauchy's Argument Principle (Fact 7.1) cannot be applied. Specifically, the proof of the Nyquist theorem given above (assuming $L(s)$ has no poles on the imaginary axis) uses the simple, closed curve Γ_R shown on the left of Figure 7.3. If $L(s)$ has imaginary axis poles then the simple, closed curve used for analysis must be slightly perturbed (modified) to avoid these imaginary axis poles. The `nyquist` command in Matlab will not perform the required perturbations. In other words, if $L(s)$ has a pole on the imaginary axis then `nyquist` will show a Nyquist curve diverging to infinity. It is up to the user to understand how this curve should be modified using the perturbation described below (or any other similar perturbation).

There are many ways to perform such perturbations. The analysis here will be performed using a perturbed curve $\Gamma_{R,\epsilon}$ defined by three pieces as shown on the left in Figure 7.11. The perturbed curve $\Gamma_{R,\epsilon}$ is simply Γ_R shifted by ϵ along the real axis. The curve $\Gamma_{R,\epsilon}$ is shown for the specific values of $R = 1.5$ and $\epsilon = -0.1$. In this case the perturbed curve is shifted left by 0.1 since $\epsilon < 0$. The right side of Figure 7.11 shows the curve $L(\Gamma_{R,\epsilon})$ defined with the transfer function $L(s) = \frac{1}{s}$. The curve loops counter clockwise touching the negative real axis at $L(s_2) = \frac{1}{\epsilon} = -10$. As $\epsilon \to 0$, the curve $\Gamma_{R,\epsilon}$ converges to Γ_R. Hence $L(\Gamma_{R,\epsilon})$ grows unbounded as $\epsilon \to 0$ due to the pole at $s = 0$.

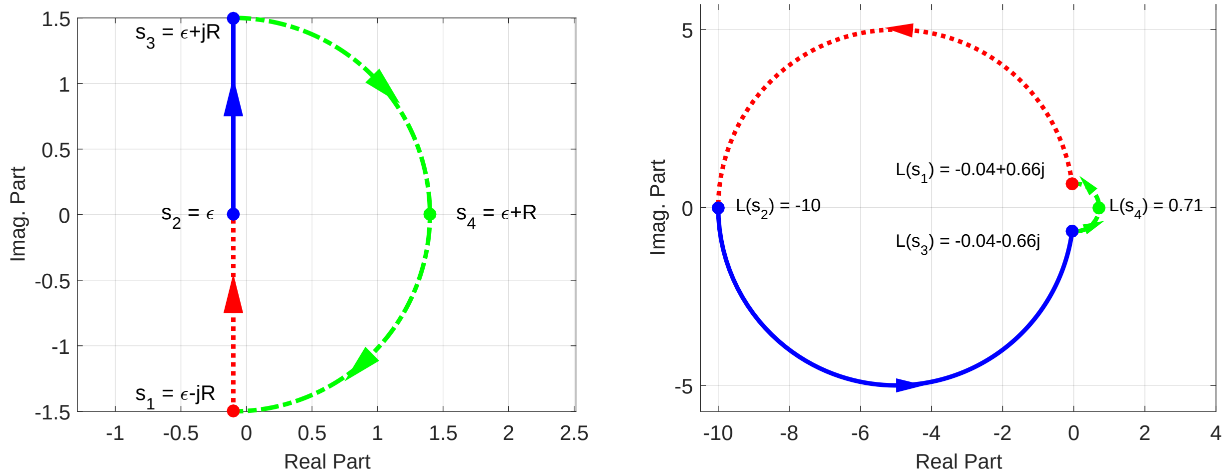


Figure 7.11: A simple, closed curve $\Gamma_{R,\epsilon}$ defined with $(R, \epsilon) = (1.5, -0.1)$ (left) and its mapping $L(\Gamma_{R,\epsilon})$ defined with $L(s) = \frac{1}{s}$ (right).

The following notation is used to state the extended Nyquist theorem:

- $P_{CL}(\epsilon)$: This denotes the number of poles of the closed-loop system with real part $\geq \epsilon$.
- $P_{OL}(\epsilon)$: This denotes the number of poles of the open-loop transfer function $L(s)$ with real part $\geq \epsilon$.
- $N_{CCW}(\epsilon)$: This denotes the number of times the Nyquist curve $L(\epsilon + j\omega)$ encircles the critical -1 point in the counter-clockwise direction. Note that the Nyquist curve $L(\epsilon + j\omega)$ corresponds to $L(\Gamma_{R,\epsilon})$ as $R \to \infty$.

The Nyquist theorem is stated next in a form that can be applied for loops $L(s)$ with poles on the imaginary axis.

Fact 7.3 (Extended Nyquist Theorem)**.** Assume $L(s)$ has no poles with real part equal to ϵ. Then $P_{CL}(\epsilon) = P_{OL}(\epsilon) - N_{CCW}(\epsilon)$. Thus $P_{CL}(\epsilon) = 0$ if and only if $N_{CCW}(\epsilon) = P_{OL}(\epsilon)$.

The initial Nyquist theorem (Fact 7.2) is a special case of this extended Nyquist theorem when $\epsilon = 0$. The proof of this extended Nyquist theorem is similar to the one given above for the basic Nyquist theorem. It follows from Cauchy's Argument Principle using the the transfer function $H(s) = 1 + L(s)$ and the simple, closed curve $\Gamma_{R,\epsilon}$. If $L(s)$ has poles on the imaginary axis then the extended Nyquist theorem should be applied for some "small" $\epsilon < 0$. Note that $\Gamma_{R,\epsilon}$ converges to $Re\{s\} \geq \epsilon$ as $R \to \infty$ and this contains the RHP if $\epsilon < 0$. Thus if $N_{CCW}(\epsilon) = P_{OL}(\epsilon)$ and $\epsilon < 0$ then $P_{CL}(\epsilon) = 0$ by the extended Nyquist theorem. In this case the closed-loop contains no poles in RHP and hence the closed-loop is stable. The next example demonstrates this result.

Example 7.9. Consider the loop transfer function $L(s) = \frac{1}{s}$. Figure 7.12 shows the curve $L(\epsilon + j\omega)$ for $\epsilon = -0.1$. This curve corresponds to $L(\Gamma_{R,\epsilon})$ as $R \to \infty$. The loop $L(s)$ has a single pole at $s = 0$ which lies within $\Gamma_{R,\epsilon}$, i.e. $P_{OL}(\epsilon) = 1$. In addition, the Nyquist curve $L(\epsilon + j\omega)$ encircles the critical -1 point once in the counter clockwise direction, i.e. $N_{CCW}(\epsilon) = 1$. By the extended Nyquist theorem $P_{CL}(\epsilon) = P_{OL}(\epsilon) - N_{CCW}(\epsilon) = 0$. Thus the feedback system with $L(s) = \frac{1}{s}$ has no poles with real part $\geq \epsilon = -0.1$. This implies the closed-loop has no poles in the RHP (including the imaginary axis) and hence it is stable. This result can be verified by directly computing a closed-loop transfer function. For example, $S(s) = \frac{1}{1+L(s)} = \frac{s}{s+1}$ has one pole at $s = -1$ and is stable.

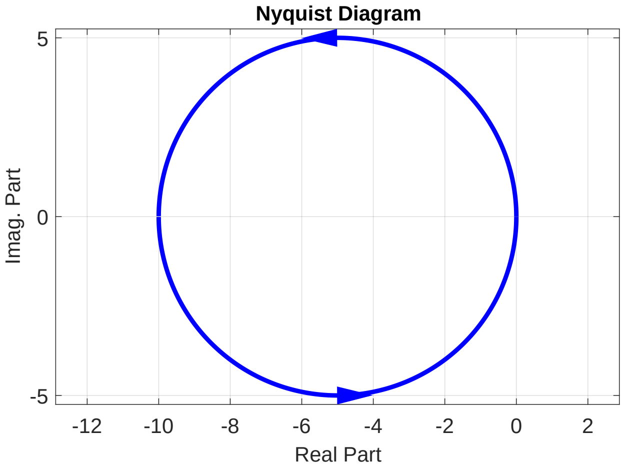


Figure 7.12: Nyquist plot for $L(s) = \frac{1}{s}$ along $s = \epsilon + j\omega$ with $\epsilon = -0.1$.

△

7.4 Gain Margins

Summary: This section discusses one a safety factor called the *gain margin* to account for model uncertainty. The gain margin is the amount of allowable variation in the gain of the plant before the closed-loop becomes unstable. As a rule of thumb, the closed-loop should remain stable for gain variations in the range $[0.5, 2]$ $(= \pm 6dB)$. It is shown that a (positive) gain variation g_0 causes a closed-loop pole at $s = \pm j\omega_0$ if and only if $L(j\omega_0) = -\frac{1}{g_0}$. Based on this fact, Bode plots can be used to quickly determine the gain margin of a system.

7.4.1 Gain Margin Definition

As discussed previously, the model used for control design is only an approximation for the "real" dynamics of the plant. Control engineers have developed different types of "safety factors" to account for the mismatch between the design model and the dynamics of the real system. This section will discuss one type of safety factor called the *gain margin.* The gain margin is the amount of allowable variation in the gain of the plant before the closed-loop becomes unstable. To make this concrete, consider the feedback diagram shown in Figure 7.13. $G(s)$ is the "nominal" model used to design the controller $K(s)$. The constant g is introduced to represent variations in the gain of the plant dynamics. In other words, the model used for design is $G(s)$ but the real dynamics might have a different gain as represented by $gG(s)$. Typically we assume that the gain of the design model at least has the correct sign and hence only positive gain variations $g > 0$ are of interest. The loop transfer function, including this gain variation, is given by $L_g(s) = gG(s)K(s)$. Note that the nominal loop transfer function corresponds to $g = 1$ and this is denoted as $L(s) = G(s)K(s)$.

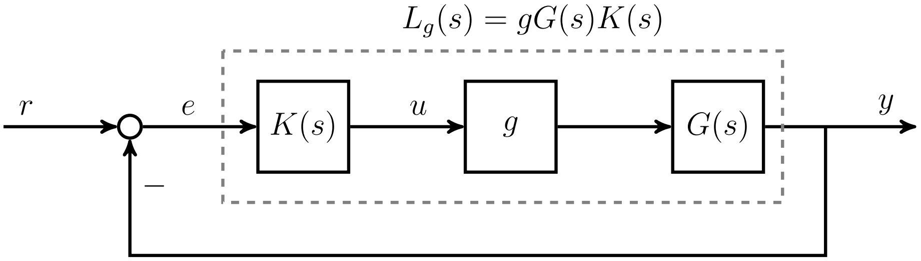


Figure 7.13: Feedback system including gain variation g in the plant.

Assume that the controller has been designed to stabilize the nominal design model $G(s)$, i.e. the closed-loop system is stable for $g = 1$. The closed-loop system may become unstable as the gain g is varied (either increased or decreased from $g = 1$). The gain margin specifies the minimum and maximum variation for which the closed-loop remains stable as defined below.

Definition 7.1 (Gain Margin)**.** The gain margin consists of an upper limit $\bar{g} \geq 1$ and a lower limit $\underline{g} \leq 1$ such that:

1. the closed-loop is stable for all positive gain variations g in the range $\underline{g} < g < \bar{g}$, and
2. the closed-loop is unstable for gain variations $g = \bar{g}$ (if $\bar{g} < \infty$) and also unstable for gain variations $g = \underline{g}$ (if $\underline{g} > 0$).

Thus the gain margins $\bar{g}$ and $\underline{g}$ correspond to the variations that lie at the boundary between stable and unstable closed-loops. In some cases the closed-loop remains stable for arbitrarily large increases and/or decreases in the gain g. The upper limit is $\bar{g} = +\infty$ if the closed-loop remains stable for any gain $g > 1$. The lower limit is $\underline{g} = 0$ if the closed-loop remains stable for any positive gain $g < 1$. **As a rule of thumb, the gain margin limits should satisfy $\bar{g} > 2$ and $\underline{g} < 2$ for good robustness to model uncertainty. This is frequently stated as $\pm 6dB$ of required gain margin for good robustness.**[‡] This rule of thumb is a good starting point but requirements may differ slightly depending on the specific problem.

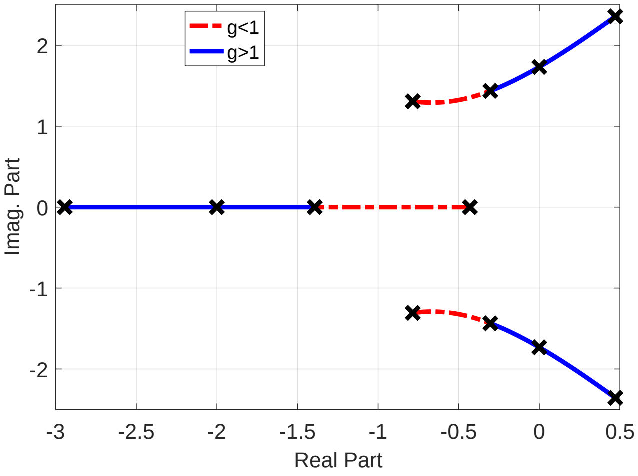
Example 7.10. Consider a feedback system with nominal plant model $G(s) = \frac{1}{s^3+2s^2+3s+1}$ and controller $K(s) = 2$. The nominal loop transfer function is $L(s) = G(s)K(s) = \frac{2}{s^3+2s^2+3s+1}$. The nominal sensitivity function is thus $S(s) = \frac{1}{1+L(s)} = \frac{s^3+2s^2+3s+1}{s^3+2s^2+3s+3}$. The poles of $S(s)$ are $s_{1,2} = -0.30 \pm 1.44j$ and $s_3 = -1.39$. All poles of $S(s)$ are in the left half plane and hence the nominal closed-loop is stable (Fact 6.1).

The left plot in Figure 7.14 shows the locus of closed-loop poles, i.e. poles of $S_g(s) = \frac{1}{1+gL(s)}$, as the gain g varies continuously in the range $0 \le g \le 8$. The closed-loop poles are marked for the specific values of $g = \{0,\ 1,\ 2.5, 8\}$ and are also shown in the table in Figure 7.14. There are several items to note. First, the poles of the "nominal" sensitivity function $S(s)$ (given above) correspond to the case $g = 1$. Second, the poles remain in the LHP for any non-negative gain satisfying $g < 1$. In fact, as $g \to 0$ the closed-loop poles converge to those of $L(s)$. Third, two closed-loop poles cross into the RHP as the gain g is increased above 1. The locus would continue for $g \to \infty$ but is only drawn for variations up to $g = 8$. If $g = 2.5$ then the closed-loop is unstable with poles on the imaginary axis at $s_{1,2} = \pm 1.73j$. The closed-loop is unstable if $g \ge 2.5$ and stable for non-negative gains $g < 2.5$. Based on this discussion, the upper and lower gain margins for this system are $\bar{g} = 2.5 \approx +8dB$ and $\underline{g} = 0$. $\triangle$

Example 7.11. Next consider a feedback system with nominal plant model $G(s) = \frac{1}{s^3+2s^2+3s-1}$ and controller $K(s) = 2$. The nominal loop transfer function is $L(s) = G(s)K(s) = \frac{2}{s^3+2s^2+3s-1}$. Note that $L(s)$ is unstable in this case because it has one pole at $s = 0.28$ in the RHP. The nominal sensitivity function in this case is $S(s) = \frac{1}{1+L(s)} = \frac{s^3+2s^2+3s-1}{s^3+2s^2+3s+3}$. The poles of $S(s)$ are $s_{1,2} = -0.78 \pm 1.31j$ and $s_3 = -0.43$. All poles of $S(s)$ are in the left half plane and hence the nominal closed-loop is stable (Fact 6.1).

The left plot in Figure 7.15 shows the locus of closed-loop poles, i.e. poles of $S_g(s) = \frac{1}{1+gL(s)}$, as the gain g varies continuously in the range $0 \le g \le 8$. The closed-loop poles are marked for the specific values of $g = \{0,\ 0.5,\ 1,\ 3.5, 8\}$ and are also shown in the table in Figure 7.15. There are several items to note. First, the poles of the "nominal" sensitivity function (given above) correspond to the case $g = 1$. Second, one of the closed-loop poles crosses into the RHP as the gain g is decreased below 1. If $g = 0.5$ then the closed-loop is unstable with a pole on the imaginary axis at $s_3 = 0$. Thus the closed-loop is unstable for positive gains $g \le 0.5$ and stable if $0.5 < g \le 1$. Third, two closed-loop poles cross into the RHP as the gain g is increased above 1. The locus would continue for $g \to \infty$ but is only drawn for variations up to $g = 8$. If $g = 3.5$ then the closed-loop is unstable with poles on the imaginary axis at $s_{1,2} = \pm 1.73j$.

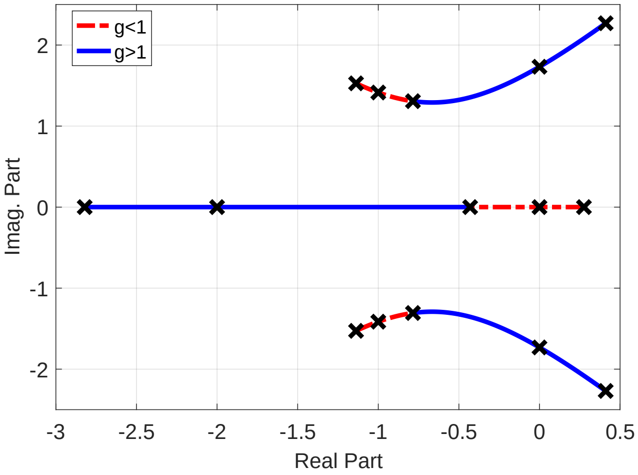
[‡] Recall that $\bar{g} = +6dB$ corresponds to an actual upper gain margin of $\bar{g} = +2$. Similarly, $\underline{g} = -6dB$ corresponds to an actual lower gain margin of $\underline{g} = 0.5$.



g	Pole 1	Pole 2	Pole 3
0.0	-0.78 + 1.31j	-0.78 - 1.31j	-0.43
1.0	-0.30 + 1.44j	-0.30 - 1.44j	-1.39
2.5	+1.73j	-1.73j	-2.00
8.0	0.47 + 2.36j	0.47 - 2.36j	-2.94

Figure 7.14: Left: Locus of closed-loop poles, i.e. poles of $S_g(s) = \frac{1}{1+gL(s)}$, as constant gain g varies from $g = 0$ to $g = 8$. The locus is drawn using the nominal loop $L(s) = \frac{2}{s^3+2s^2+3s+1}$. Right: Table of closed-loop pole locations for specific values of g.

Thus closed-loop is unstable if $g \geq 3.5$ and stable if $1 \leq g < 3.5$. Based on this discussion, the upper and lower gain margins for this system are $\bar{g} = 3.5 \approx +10.9dB$ and $\underline{g} = 0.5 \approx -6dB$.



g	Pole 1	Pole 2	Pole 3
0.0	-1.14 + 1.53j	-1.14 - 1.53j	0.28
0.5	-1.0 + 1.41j	-1.0 - 1.41j	0
1.0	-0.78 + 1.31j	-0.78 - 1.31j	-0.43
3.5	+1.73j	- 1.73j	-2
8.0	0.41 + 2.27j	0.41 - 2.27j	-2.82

Figure 7.15: Left: Locus of closed-loop poles, i.e. poles of $S_g(s) = \frac{1}{1+gL(s)}$, as constant gain g varies from $g = 0$ to $g = 8$. The locus is drawn using the nominal loop $L(s) = \frac{2}{s^3+2s^2+3s-1}$. Right: Table of closed-loop pole locations for specific values of g.

$\triangle$

7.4.2 Connection to Bode Plots

This subsection discusses the use of Bode plots to quickly determine the gain margin of a system. Assume the nominal closed-loop is stable, i.e. all poles are in the LHP. The gain variations g cause the closed-loop poles to move continuously in the complex plane as observed in Examples 7.10 and 7.11. It is possible for the closed-loop poles to move into the RHP (unstable closed loop) if the gain g is increased or decreased by a sufficiently large amount. The critical gains are values of g for which the closed-loop poles are on the imaginary axis. These gains mark the transition as poles move from the LHP (stable) into the RHP (unstable). To be concrete, a critical gain g_0 causes the closed-loop to have a pole on the imaginary axis $s = \pm j\omega_0$ for some frequency $\omega_0 \geq 0$. Thus the sensitivity $S_{g_0}(s) = \frac{1}{1+g_0 L(s)}$ has poles at $s = \pm j\omega_0$ which is equivalent to $1 + g_0 L(j\omega_0) = 0$ (Fact 6.1). **To summarize, a (positive) gain variation g_0 causes a closed-loop pole at $s = \pm j\omega_0$ if and only if $L(j\omega_0) = -\frac{1}{g_0}$.**

The condition for critical gains can be used to calculate the gain margin directly from the Bode plot of $L(s)$. Specifically, the condition $L(j\omega_0) = -\frac{1}{g_0}$ for some positive gain g_0 implies that $\angle L(j\omega_0) = \pm 180 deg$. The gain margins are computed from a Bode plot as follows:

1. Identify the frequencies $\{\omega_1, \ldots, \omega_N\}$ where $\angle L(j\omega_i) = \pm 180 deg$. These are called the phase crossing frequencies or gain margin frequencies. There may be zero, one, or many gain margin frequencies.

2. The associated critical gain for each frequency ω_i is $g_i = \frac{1}{|L(j\omega_i)|}$ $(i = 1, \ldots, N)$. This critical gain g_i results in a closed-loop pole at $s = \pm j\omega_i$.

3. The upper gain margin $\bar{g}$ is the smallest of the critical gains above 1. The lower gain margin $\underline{g}$ is the largest of the critical gains below 1. For example if the nominal closed-loop is stable and the critical gains are $\{0.4, 0.8, 1.5, 2.0\}$ then the closed-loop is stable for all gains in the range $\underline{g} = 0.8 < g < 1.5 = \bar{g}$.

Example 7.12. The left side of Figure 7.16 shows the Bode plot for $L(s) = \frac{2}{s^3+2s^2+3s+1}$. This is the same loop transfer function studied in Example 7.10. The Bode plot has a single gain margin frequency at $\omega_1 = 1.73 rad/sec$. This frequency is marked in both the phase and magnitude plot. The magnitude at this frequency is $|L(j\omega_1)| = -8dB = 0.4$. The corresponding critical gain is $g_1 = \frac{1}{|L(j\omega_1)|} = 2.5$. Thus the gain $g_1 = 2.5$ causes the closed-loop to have poles at $s = \pm 1.73j$. The gain margins are $\bar{g} = 2.5$ and $\underline{g} = 0$. These results agree with those given in Example 7.10.

The right side of Figure 7.16 shows the Bode plot for $L(s) = \frac{2}{s^3+2s^2+3s-1}$. This is the same loop transfer function studied in Example 7.11. The Bode plot has two gain margin frequencies at $\omega_1 = 1.73 rad/sec$ and $\omega_2 = 0 rad/sec$. The first frequency is marked in both the phase and magnitude plot. The magnitude at the first gain margin frequency is $|L(j\omega_1)| = -10.9dB \approx 0.29$. The corresponding critical gain is $g_1 = \frac{1}{|L(j\omega_1)|} = 3.5$. Thus the first critical gain $g_1 = 3.5$ causes the closed-loop to have poles at $s = \pm 1.73j$. The magnitude at the second gain margin frequency is $|L(j\omega_2)| = +6dB \approx 2$. The corresponding critical gain is $g_2 = \frac{1}{|L(j\omega_2)|} = 0.5$. Thus the second critical gain $g_2 = 0.5$ causes the closed-loop to have a pole at $s = 0$. Based on these results, the gain margins are $\bar{g} = 3.5$ and $\underline{g} = 0.5$. These results agree with those given in Example 7.11.

△

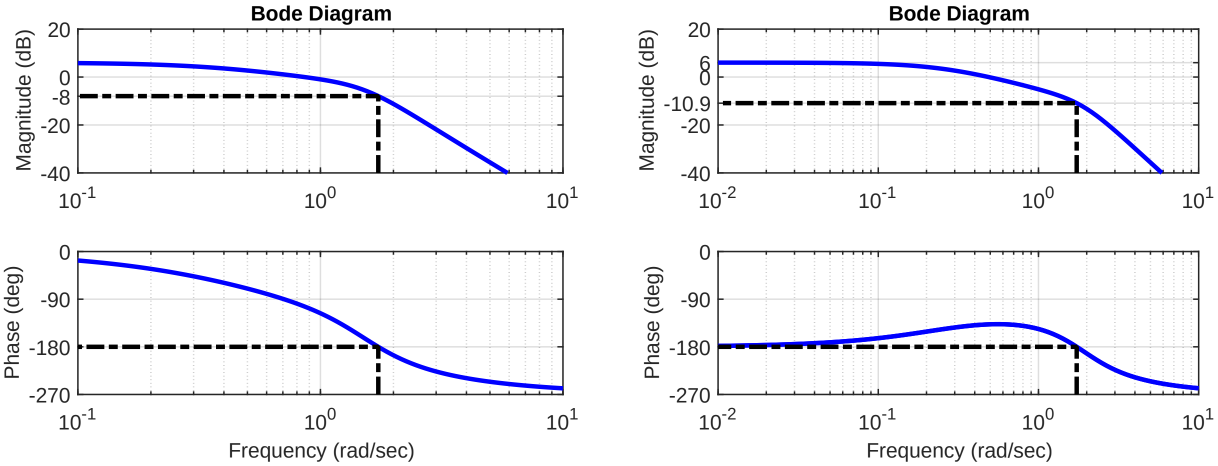


Figure 7.16: Bode plot for $L(s) = \frac{2}{s^3+2s^2+3s+1}$ (left) and $L(s) = \frac{2}{s^3+2s^2+3s-1}$ (right). Gain margin frequencies are marked in both cases.

7.5 Phase and Time Delay Margins

Summary: Control systems must be designed to tolerate a small amount of time delays. This motivates a safety factor called the *phase margin*. The phase margin is the amount of allowable variation in the phase of the plant before the closed-loop becomes unstable. As a rule of thumb, the closed-loop should remain stable for phase variations in the range $[-45, +45]deg$. It is shown that a phase variation θ_0 causes a closed-loop pole at $s = j\omega_0$ if and only if $e^{-j\theta_0} L(j\omega_0) = -1$. Based on this fact, Bode plots can be used to quickly determine the phase margin of a system. Time delay margins can also be computed.

7.5.1 Time Delays

Control laws are typically implemented on an embedded processor as discussed in Section 4.8. The computations take a certain amount of time and this introduces a delay in the feedback loop. A simple model for this effect is a pure time delay τ_d as shown on the left of Figure 7.17. The right plot in Figure 7.17 shows an example pair of input/output signals for the delay. The input signal is $u(t) = \sin(2t)$ and the time delay is $\tau_d = 0.7sec$. The output signal is simply shifted to the right (delayed): $v(t) = u(t - \tau_d) = \sin(2(t - 0.7))$ (for $t \geq \tau_d$). Note that $v(t)$ has the same amplitude as $u(t)$ but with an additional phase shift. In general, the response of a delay τ_d to an input $u(t) = \sin(\omega t)$ is given by the output $v(t) = \sin(\omega t - \omega \tau_d)$. Thus the frequency response of a delay τ_d has magnitude equal to one at all frequencies because the output amplitude is the same as the input amplitude. Moreover, the frequency response of the delay has phase equal to $-\omega\tau_d$. This magnitude and phase correspond to the complex number $e^{-j\omega\tau_d}$, i.e. the frequency response of a delay τ_d is given by $e^{-j\omega\tau_d}$. As notation, the delay is represented by the corresponding complex function $e^{-s\tau_d}$. The control law must be designed to ensure robustness to such delays.

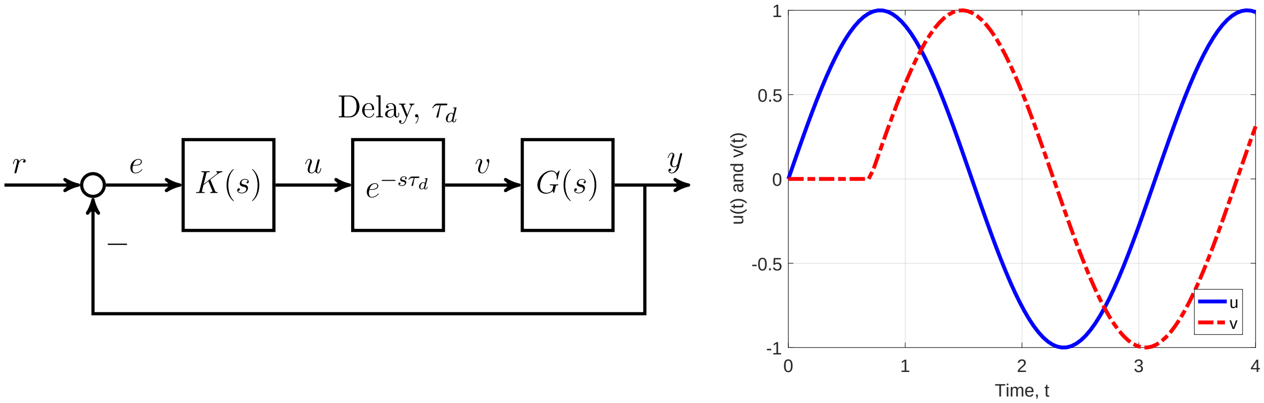


Figure 7.17: Feedback system with time delay τ_d (left) and an example input/output pair for time delay (right).

7.5.2 Phase Margins

This section will discuss another type of safety factor called the *phase margin*. The phase margin is the amount of allowable variation in the phase of the plant before the closed-loop becomes unstable. To make this concrete, consider the feedback diagram shown in Figure 7.18. Again, $G(s)$ is the "nominal" model used to design the controller $K(s)$. The complex number $e^{-j\theta}$ is introduced to represent phase variations in the plant dynamics. The loop transfer function, including this phase variation, is given by $L_\theta(s) = e^{-j\theta}G(s)K(s)$. Note that the nominal loop transfer function corresponds to $\theta = 0$ and this is denoted as $L(s) = G(s)K(s)$. The term phase variation arises because $\angle L_\theta(j\omega) = \angle L(j\omega) - \theta$, i.e. θ modifies the angle (phase) of the dynamics. As discussed in the previous subsection, phase variations can occur due to time delays in the feedback loop. Sufficient phase margin is required to ensure that such delays do not destabilize the system.

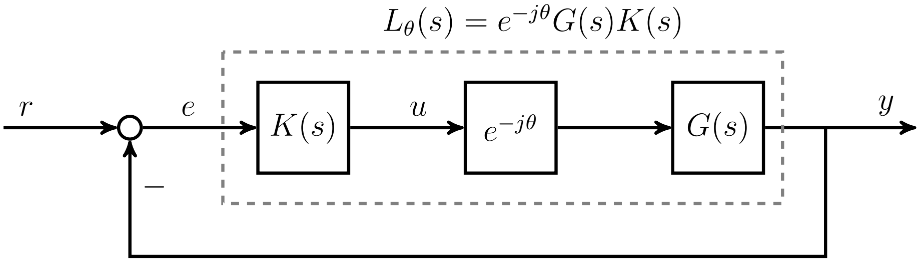


Figure 7.18: Feedback system including phase variation $e^{-j\theta}$ in the plant.

Assume that the controller has been designed to stabilize the nominal design model $G(s)$, i.e. the closed-loop system is stable for $\theta = 0$. The phase variations will cause the closed-loop poles to move continuously in the complex plane. It is possible for the closed-loop poles to move into the RHP (unstable closed loop) if the phase θ is increased or decreased by a sufficiently large amount. The critical phases are values for which the closed-loop poles are on the imaginary axis. These phases mark the transition as poles move from the LHP (stable) into the RHP (unstable). In particular, a critical phase θ_0 causes the closed-loop to have a pole on the imaginary axis $s = j\omega_0$ for some frequency $\omega_0 \geq 0$. Thus the sensitivity $S_{\theta_0}(s) = \frac{1}{1+e^{-j\theta_0}L(s)}$ has a pole $s = j\omega_0$ which is equivalent to $1 + e^{-j\theta_0}L(j\omega_0) = 0$ (Fact 6.1). **To summarize, a phase variation θ_0 causes a closed-loop pole at $s = j\omega_0$ if and only if $e^{-j\theta_0}L(j\omega_0) = -1$.**

Two additional facts are required before stating the formal phase margin definition. First, complex numbers repeat with every $360deg$ $(= 2\pi)$ change in phase, i.e. $e^{j\theta} = e^{j\theta+2\pi}$. Thus only phases in the range $-180deg \leq \theta \leq 180deg$ need to be considered. Second, positive and negative phase variations have a similar effect. Specifically, the complex conjugate of a transfer function $L(j\omega)$ is $L(-j\omega)$ as discussed in Section 7.1. Therefore the complex conjugate of $e^{-j\theta_0}L(j\omega_0) = -1$ is $e^{j\theta_0}L(-j\omega_0) = -1$. Thus θ_0 is a critical phase causing a closed loop pole at $s = j\omega_0$ if and only if $-\theta_0$ is a critical phase causing a closed-loop pole at $s = -j\omega_0$. Either sign of critical phase will cause instability. Based on these facts, the phase margin specifies the maximum variation for which the closed-loop remains stable as defined below.

Definition 7.2 (Phase Margin)**.** The phase margin consists of an upper limit $\bar{\theta} \geq 0$ such that:

1. the closed-loop is stable for all phase variations θ in the range $-\bar{\theta} < \theta < \bar{\theta}$, and
2. the closed-loop is unstable for $\theta = \bar{\theta}$ (if $\bar{\theta} < \infty$).

The phase margin $\bar{\theta}$ corresponds to the variation that lies at the boundary between stable and unstable closed-loops. In some cases the closed-loop remains stable for all phases in the range $0 \leq \theta \leq 180deg$. In this case, the upper limit is defined to be $\bar{\theta} = +\infty$. **As a rule of thumb, the phase margin limit should satisfy $\bar{\theta} > 45deg$ for good robustness to model uncertainty. This is frequently stated as $\pm 45deg$ of required phase margin for good robustness.** This rule of thumb is a good starting point but requirements may differ slightly depending on the specific problem.

The condition for critical phases can be used to calculate the phase margin directly from the Bode plot of $L(s)$. Specifically, the condition $e^{-j\theta_0} L(j\omega_0) = -1$ for some phase θ_0 implies that $|L(j\omega_0)| = 1 = 0dB$. The phase margins are computed from a Bode plot as follows:

1. Identify the frequencies $\{\omega_1, \ldots, \omega_N\}$ where $|L(j\omega_0)| = 1 = 0dB$. These are called the gain crossing frequencies or phase margin frequencies. There may be zero, one, or many phase margin frequencies.
2. The critical phase for each frequency ω_i satisfies $-\theta_i + \angle L(j\omega_i) = -180deg$. Thus the critical phases are $\theta_i = \angle L(j\omega_i) + 180deg$ $(i = 1, \ldots, N)$. To perform this calculation, first express $\angle L(j\omega_i)$ in the interval $[0, -360]deg$ by adding or subtracting factors of $360deg$. This yields a critical phase θ_i in the range $[-180, +180]deg$ and results in a closed-loop pole at $s = j\omega_i$. As an example, if $\angle L(j\omega_i) = -135deg$ then the critical phase is $\theta_i = -135 + 180 = 45deg$.
3. The phase margin $\bar{\theta}$ is the smallest (in magnitude) critical phase, i.e. it is the smallest value of $|\theta_i|$ for $i = 1, \ldots, N$.

Example 7.13. The left side of Figure 7.19 shows the Bode plot for $L(s) = \frac{2}{s^3+2s^2+3s+1}$. This is the same loop transfer function studied in Examples 7.10 and 7.12. The Bode plot has a single phase margin frequency at $\omega_1 = 0.86rad/sec$. This frequency is marked in both the phase and magnitude plot. The phase at this frequency is $\angle L(j\omega_1) = -103.7deg$. The corresponding critical phase is $\theta_1 = \angle L(j\omega_1) + 180deg = 76.3deg$. Thus the phase $\theta_1 = 76.3deg$ causes the closed-loop to have a pole at $s = 0.86j$. The phase margin is $\bar{\theta} = 76.3deg$.

The right side of Figure 7.19 shows the Bode plot for $L(s) = \frac{2}{s^3+2s^2+3s-1}$. This is the same loop transfer function studied in Examples 7.11 and 7.12. The Bode plot has a single phase margin frequency at $\omega_1 = 0.49rad/sec$. This frequency is marked in both the phase and magnitude plot. The phase at this frequency is $\angle L(j\omega_1) = -137.6deg$. The corresponding critical phase is $\theta_1 = \angle L(j\omega_1) + 180deg = 42.4deg$. Thus the phase $\theta_1 = 42.4deg$ causes the closed-loop to have a pole at $s = 0.49j$. The phase margin is $\bar{\theta} = 42.4deg$. △

7.5.3 Time Delay Margin

As discussed in Section 7.5.1, a pure time delay τ_d has the frequency response $e^{-j\omega\tau_d}$. For a fixed frequency this introduces a phase variation $e^{-j\theta}$ with $\theta = \omega\tau_d$. Thus the time delay margin can be computed from the phase margin as follows:

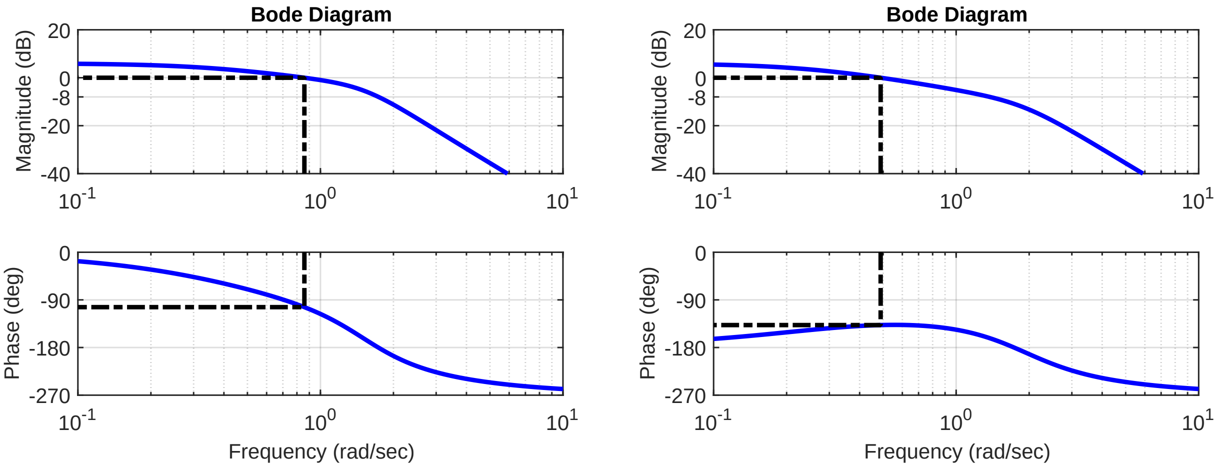


Figure 7.19: Bode plot for $L(s) = \frac{2}{s^3+2s^2+3s+1}$ (left) and $L(s) = \frac{2}{s^3+2s^2+3s-1}$ (right). Phase margin frequencies are marked in both cases.

1. Identify the phase margin frequencies $\{\omega_1, \ldots, \omega_N\}$ and corresponding critical phases $\{\theta_1, \ldots, \theta_N\}$ from a Bode plot as described in Section 7.5.2.
2. The critical delays are given by $\tau_{d,i} = \left|\frac{\theta_i}{\omega_i}\right| > 0$ for $i = 1, \ldots, N$.
3. The time delay margin $\bar{\tau}_d$ is the smallest critical delay.

Time delays can be modeled and simulated in two ways in Matlab. In Simulink there is a *transport delay* block. At the command line, it is possible to specify that a transfer function has either an input or output delay. For example, the code `G.InputDelay = 0.7` specifies that the transfer function `G` has a $0.7sec$ delay at the input. The next example provides simulation results for systems with a delay using the command line approach.

Example 7.14. Example 7.13 computed the phase margin with the loop $L(s) = \frac{2}{s^3+2s^2+3s+1}$ as $76.3deg = 1.33rad$. The corresponding phase margin frequency was $0.86rad/sec$. Thus the delay margin is $1.55sec$ ($= 1.33/0.86$) with the same frequency. The left plot of Figure 7.20 shows the step response for the closed-loop (reference to output) with the delay equal to the margin $1.55sec$. The step responses oscillates with a frequency of $0.86rad/sec$ as expected.

Example 7.13 also computed the phase margin with the loop $L(s) = \frac{2}{s^3+2s^2+3s-1}$ as $42.4deg = 0.74rad$. The corresponding phase margin frequency was $0.49rad/sec$. Thus the delay margin is $1.51sec$ ($= 0.74/0.49$) with the same frequency. The right plot of Figure 7.20 shows the step response for the closed-loop (reference to output) with the delay equal to the margin $1.51sec$. The step responses oscillates with a frequency of $0.49rad/sec$ as expected.

△

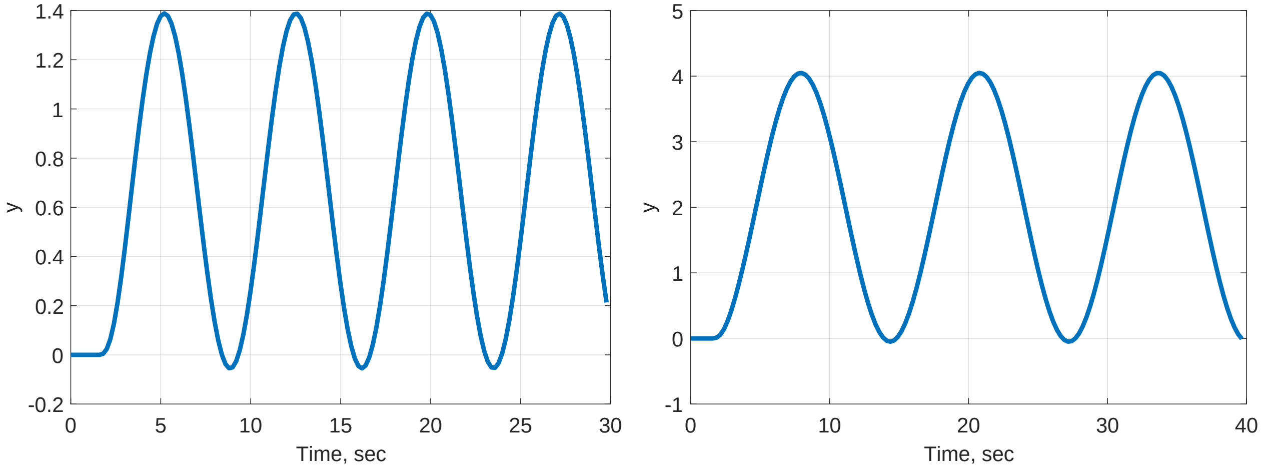


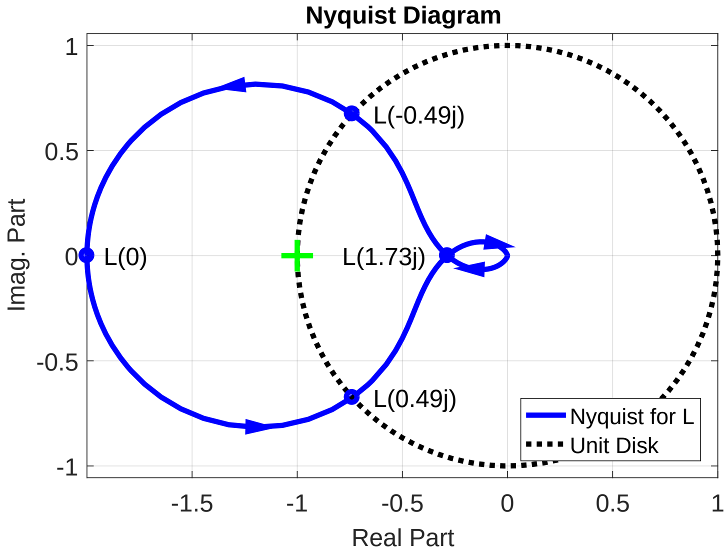
Figure 7.20: Step responses for closed-loop (reference to output) for $L(s) = \frac{2}{s^3+2s^2+3s+1}$ with delay of $1.55sec$ (left) and $L(s) = \frac{2}{s^3+2s^2+3s-1}$ with delay of $1.52sec$ (right).

7.6 Disk Margins

Summary: Nyquist plots provide another interpretation for gain and phase margins. Specifically, gain and phase variations can cause the Nyquist curve to change from having the "correct" number of encirclements of the -1 point (stable closed-loop) to having an "incorrect" number of encirclements (unstable closed-loop). These two margins measure how close the Nyquist plot of the loop $L(s)$ approaches the critical -1 point along two particular directions. Another robustness margin is obtained by directly considering the minimum distance d_{min} between $L(j\omega)$ and the critical -1 point. As a rule of thumb, $d_{min} \geq 0.4$ will provide good robustness margins. This corresponds to $|S(j\omega)| \leq 2.5$ for all ω. This rule of thumb implies that the Nyquist plot of $L(j\omega)$ does not enter a disk of radius 0. centered at -1. The requirement $d_{min} \geq 0.4$ is called a disk margin based on this graphical interpretation.

7.6.1 Gain/Phase Margins: Connection to Nyquist Plots

Nyquist plots provide another interpretation for gain and phase margins. Consider the (nominal) loop $L(s) = \frac{2}{s^3+2s^2+3s-1}$ which was studied in previous subsections. The poles of $L(s)$ are at $s_{1,2} = -1.14 \pm 1.53j$ and $s_3 = 0.28$. Thus $L(s)$ is unstable with a single pole in the RHP ($P_{OL} = 1$). The Nyquist curve of $L(s)$, shown on the left of Figure 7.21, has a single counterclockwise encirclement of the critical -1 point ($N_{CCW} = 1$). By the Nyquist Theorem (Fact 7.2), the (nominal) closed-loop has $P_{CL} = P_{OL} - N_{CCW} = 0$ poles in the RHP and hence the closed-loop is stable. This can be verified by directly computing the (nominal) sensitivity function: $S(s) = \frac{1}{1+L(s)} = \frac{s^3+2s^2+3s-1}{s^3+2s^2+3s+3}$. The poles of $S(s)$ are $s_{1,2} = -0.78 \pm 1.31j$ and $s_3 = -0.43$. All poles of $S(s)$ are in the left half plane and the nominal closed-loop is stable as predicted by the Nyquist theorem.



ω $(\frac{rad}{sec})$	$G(j\omega)$ Polar	Real Part	Imag. Part
-0.49	$1.0e^{2.41j}$	-0.74	0.67
0	$2.0e^{-\pi j}$	-2.0	0.0
0.49	$1.0e^{-2.41j}$	-0.74	-0.67
1.73	$0.29e^{-\pi j}$	-0.29	0.0

Figure 7.21: Nyquist plot (left) for $L(s) = \frac{2}{s^3+2s^2+3s-1}$ and data evaluated at several $s = j\omega$ (right). The solid circles on the Nyquist plot highlight the data at several frequencies.

Recall that a (positive) gain variation g_0 causes a closed-loop pole at $s = \pm j\omega_0$ if and only if $L(j\omega_0) = -\frac{1}{g_0}$. This condition implies that $\angle L(j\omega_0) = \pm 180deg$, i.e. $L(j\omega_0)$ is a real, negative number. The Nyquist curve shown in Figure 7.21 has two frequencies where $L(j\omega)$ is a real negative number: $\omega_1 = 1.73rad/sec$ and $\omega_2 = 0rad/sec$. Both frequencies are marked on the Nyquist plot and the corresponding frequency response is given in the table on the right of Figure 7.21. The magnitude at the first gain margin frequency is $|L(j\omega_1)| \approx 0.29$ and the corresponding critical gain is $g_1 = \frac{1}{|L(j\omega_1)|} = 3.5$. The magnitude at the second gain margin frequency is $|L(j\omega_2)| = 2.0$ and the corresponding critical gain is $g_2 = \frac{1}{|L(j\omega_2)|} = 0.5$. Based on these results, the gain margins are $\bar{g} = 3.5$ and $\underline{g} = 0.5$. These results agree with those given in Examples 7.11 and 7.12.

Next, recall that a phase variation θ_0 causes a closed-loop pole at $s = j\omega_0$ if and only if $e^{-j\theta_0}L(j\omega_0) = -1$. This condition implies that $|L(j\omega_0)| = 1$, i.e. $L(j\omega_0)$ intersects the disk of radius one centered at the origin (unit disk). The Nyquist curve shown in Figure 7.21 has one positive frequency where $L(j\omega)$ has magnitude equal to one: $\omega_1 = 0.49rad/sec$. This frequency is marked on the Nyquist plot and the corresponding frequency response is given in the table on the right of Figure 7.21. The phase at the phase margin frequency is $\angle L(j\omega_1) = -2.41rad \approx -137.6deg$. The corresponding critical phase is $\theta_1 = \angle L(j\omega_1) + 180deg = 42.4deg$. Thus the phase margin is $\bar{\theta} = 42.4deg$. Recall that there is a symmetry in phase margins, i.e $\theta_1 > 0$ causes an instability if and only if $-\theta_1$ causes instability. This symmetry appears on the Nyquist plot at the negative frequency $-0.49rad/sec$. Specifically $|L(-0.49j)|$ also has magnitude equal to one and the phase at this frequency is $\angle L(-0.49j) = 2.41rad \approx 137.6deg$. This yields a critical phase of $-42.4deg$. Thus the phase margin is $\bar{\theta} = 42.4deg$ and the closed-loop is stable for all phase variations in the range $-\bar{\theta} < \theta < \bar{\theta}$. These results agree with those given in Example 7.13.

The gain and phase margins mark the boundary between stable and unstable closed-loops. For example, a critical gain g_i causes the closed-loop to have a pole on the imaginary axis at $s = \pm j\omega_i$. This is equivalent to $g_i L(j\omega_0) = -1$, i.e. the Nyquist curve of $g_i L(j\omega_0)$ passes through the critical -1 point. The left plot of Figure 7.22 shows the Nyquist curve for the nominal loop $L(s) = \frac{2}{s^3+2s^2+3s-1}$ (blue) and the loop with the lower gain margin $\underline{g}L(s) = 0.5L(s)$ (red dashed). The nominal loop has a single counter-clockwise encirclement of -1 as discussed above ($N_{CCW} = +1$). This is the "correct" number of encirclements for closed-loop stability. The Nyquist curve shrinks as the gain is reduced until the curve $\underline{g}L(s)$ just passes through the -1 point. This indicates that the closed-loop has a pole at $s = j\omega_2 = 0rad/sec$. If the gain is reduced further ($g < \underline{g}$) then the Nyquist curve of $gL(s)$ will no longer encircle the -1 point ($N_{CCW} = 0$) indicating an unstable closed-loop ($P_{CL} = P_{OL} - N_{CCW} = 1$). The right plot of Figure 7.22 shows similar results using the Nyquist curve for the nominal loop $L(s)$ (blue) and the loop with the upper gain margin $\bar{g}L(s) = 3.5L(s)$ (red dashed). The Nyquist curve grows as the gain is increased until the curve $\bar{g}L(s)$ just passes through the -1 point. This indicates that the closed-loop has a pole at $s = j\omega_1 = 1.73rad/sec$. If the gain is increased further ($g > \bar{g}$) then the Nyquist curve of $gL(s)$ will encircle the -1 point once in the clockwise direction ($N_{CCW} = -1$) indicating an unstable closed-loop ($P_{CL} = P_{OL} - N_{CCW} = +2$). Similar results can be obtained for phase margins, i.e. the phase margin $\bar{\theta}$ causes the Nyquist curve $e^{\pm j\bar{\theta}}L(s)$ to pass through the critical -1 point indicating the transition from stable to unstable. **To**

summarize, gain and phase variations can cause the Nyquist curve to change from having the "correct" number of encirclements of the -1 point (stable closed-loop) to having an "incorrect" number of encirclements (unstable closed-loop).

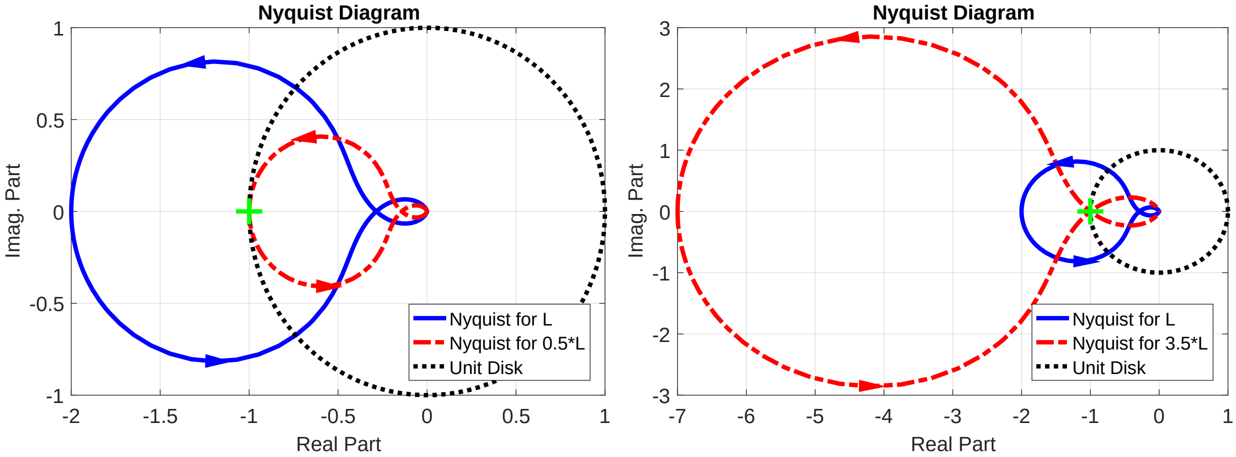


Figure 7.22: Left: Nyquist plot for $L(s) = \frac{2}{s^3+2s^2+3s-1}$ and $0.5L(s)$.
Right: Nyquist plot for $L(s)$ and $3.5L(s)$.

7.6.2 Disk Margins

Robustness margins measure how close the Nyquist plot of the (nominal) loop $L(s) = G(s)K(s)$ approaches the critical -1 point. It is assumed that the controller $K(s)$ has been designed so that the Nyquist curve of $L(s)$ encircles -1 the "correct" number of times to ensure closed-loop stability. However, if the Nyquist plot of $L(s)$ comes "near" to the -1 point then small variations in the model $G(s)$ can cause $L(j\omega)$ to pass through -1 or to have the wrong number of number of encirclements of -1. Both situations would imply that the closed-loop is unstable. The classical gain and phase margins measure how close $L(j\omega)$ comes to the critical -1 point along two specific directions. Specifically, gain margins measure the closeness between $L(j\omega)$ and -1 along the negative real axis. Phase margins measure the closeness between $L(j\omega)$ and -1 along a circle of radius 1. These two directions are highlighted by red arrows on the left plot of Figure 7.23.

Another robustness margin is obtained by directly considering the distance between $L(j\omega)$ and the critical -1 point. Specifically, the right plot of Figure 7.23 shows the vectors -1 and $L(j\omega)$ (black dotted). The vector between -1 and $L(j\omega)$ is given by $1 + L(j\omega)$ (red solid). The minimum distance between the Nyquist plot $L(j\omega)$ to -1 is a good robustness measure. This distance is given by:

$$d_{min} := \min_{\omega} |1 + L(j\omega)| \qquad (7.7)$$

If this minimum distance d_{min} is small then small variations (gain and/or phase) in the plant model can cause $L(j\omega)$ to pass through -1 or cause $L(j\omega)$ to have the wrong number of encirclements. Again, both cases would imply that the closed-loop is unstable. We can connect this metric to the sensitivity function $S(s) = \frac{1}{1+L(s)}$ as follows:

$$\frac{1}{d_{min}} = \max_{\omega} |S(j\omega)| \tag{7.8}$$

Thus a large peak value of $|S(j\omega)|$ corresponds to poor robustness, i.e. $L(j\omega)$ comes near to -1. Conversely, good robustness is ensured by small peak values of $|S(j\omega)|$, i.e. $L(j\omega)$ does not come near to -1.

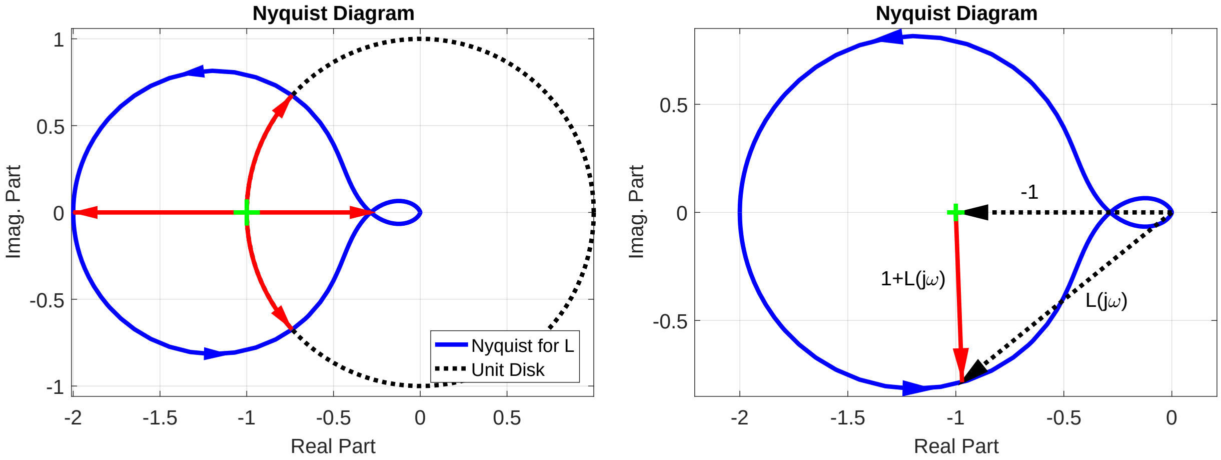


Figure 7.23: Left: Gain and Phase Margins measure the distance between $L(j\omega)$ and -1 along the red directions. Right: Actual distance between $L(j\omega)$ and -1 is given by $|1 + L(j\omega)|$.

As a rule of thumb, $d_{min} \geq 0.4$ will provide good robustness margins. This corresponds to $|S(j\omega)| \leq 2.5$ for all ω. This rule of thumb implies that the distance from the Nyquist curve $L(j\omega)$ to -1 is at least 0.4. In other words, the Nyquist plot of $L(j\omega)$ does not enter a disk of radius $0.$ centered at -1. This disk is shown (shaded green) in Figure 7.24. The requirement $d_{min} \geq 0.4$ is called a disk margin based on this graphical interpretation. The disk margins can be related to the classical gain/phase margins. Specifically, if $L(s)$ achieves a disk margin $d_{min} \geq 0.4$ then the Nyquist curve does not intersect the interval $[-1.4, -0.6]$. This implies that the system achieves gain margins of *at least* $\underline{g} \leq \frac{1}{1.4} \approx 0.71$ and $\bar{g} \geq \frac{1}{0.6} \approx 1.67$. Moreover, if $L(s)$ achieves a disk margin $d_{min} \geq 0.4$ then the Nyquist curve does not intersect the unit disk along the arc from -1 to $-0.92 - 0.39j$. This implies that the system achieves a phase margin of *at least* $\bar{\theta} \geq \tan^{-1}(\frac{0.39}{0.92} \approx 0.4rad \approx 23deg)$. These classical margins are slightly less than our previous rules of thumb of $\pm 6dB$ ($\underline{g} \leq 0.5$ and $\bar{g} \geq 2$) of gain margin and $\pm 45deg$ of phase margins. However, the values $\underline{g} \leq 0.71$, $\bar{g} \geq 1.67$, and $\bar{\theta} \geq 23deg$ are minimum bounds ensured by $d_{min} \geq 0.4$ and the actual margins are typically better than these minimums.

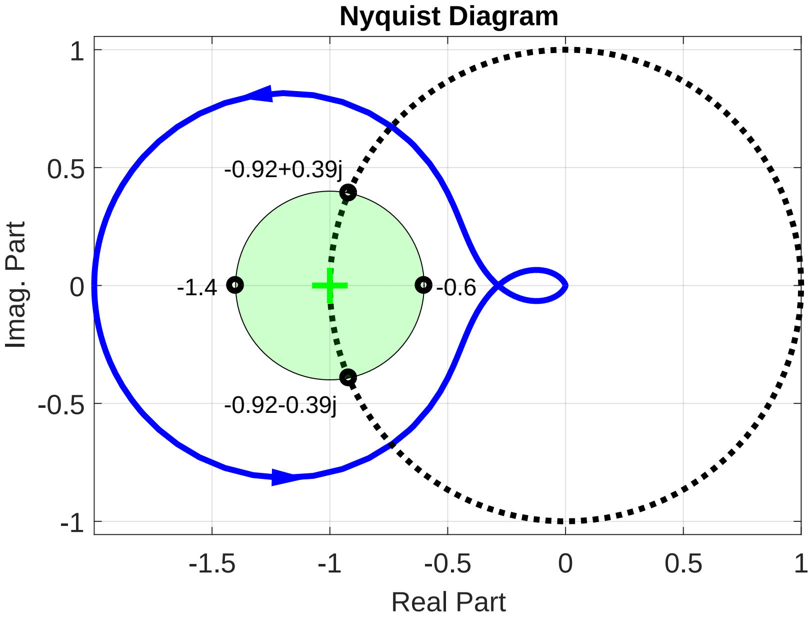


Figure 7.24: Disk Margin with $d_{min} = 0.4$.

7.7 Basic Loopshaping Theorem

Summary: This section presents a basic result to demonstrate that the loopshaping design procedure will yield a stable closed-loop with good performance and robustness. The result is based on the Nyquist theorem and the Bode Gain/Phase formula. The Bode Gain/Phase formula, presented in this section, connects the magnitude and phase plots for a transfer function with all poles and zeros in the LHP. This formula provides insight into the relation between the slope of $|L(j\omega)|$ near crossover and the stability/robustness of the closed-loop system.

See posted notes.

7.8 Lead Control

Summary: The feedback system will have poor robustness margins if the slope of $|L(j\omega)|$ is too steep ($< -30 dB/dec$) near the loop cross-over frequency. The Bode gain/phase formula can be used to make this statement more concrete. Specifically, if the slope of $|L(j\omega)$ is $> -30 dB/dec$ then, under some additional technical assumptions on L, the phase margin will be at least $45 deg$. A lead controller can be used to increase the slope (make it more shallow) near crossover and hence increase the phase margin. An example loopshaping design using lead control is provided in this section.

See posted notes.

7.9 Loopshaping Design

Summary: Performance specifications include closed loop stability, desired loop crossover frequency, bounds on $|S(j\omega)|$ for tracking performance, and bounds on $|T(j\omega)|$ for noise rejection. The loopshaping design first converts the bounds on $|S(j\omega)|$ and $|T(j\omega)|$ into roughly equivalent bounds on $|L(j\omega)|$. Next, the various controller components are designed in a series of steps to satisfy these conditions. A reasonable process is to: i) Design a proportional controller to select the desired loop crossover frequency, ii) Use a low frequency or integral boost to achieve the desired low frequency gain for $|L(j\omega)|$, and iii) Use a high frequency roll-off to achieve the high frequency roll-off for $|L(j\omega)|$. A fourth step, discussed in this chapter, may be required. This involves designing a lead controller to make the closed-loop more robust to model uncertainties.

See posted Matlab example.

Bibliography

[1] K. Åström and T. Hägglund. *PID Controllers: Theory, Design, and Tuning.* The Instrumentation, Systems, and Automation Society, 2nd edition, 1995.

[2] W. E. Boyce and R.C. DiPrima. *Elementary Differential Equations and Boundary Value Problems.* Wiley, 2008.

[3] J.W. Brown and R.V. Churchill. *Complex Variables and Applications.* McGraw-Hill Education, ninth edition, 2013.

[4] C.-T. Chen. *Linear System Theory and Design.* Oxford University Press, 1998.

[5] C.H. Edwards and D.E. Penney. *Differential Equations and Linear Algebra.* Pearson, 2008.

[6] G. Franklin, J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems.* Pearson, 2014.

[7] K. Hedrick, A. Packard, R. Horowitz, K. Poolla, and F. Borrelli. Me 132: Dynamic systems and feedback (class notes). University of California, Berkeley, Fall 2010.

[8] J.P. Hespanha. *Linear SYstems Theory.* Princeton University Press, 2009.

[9] J.K.Hedrick, D. Godbole, R. Rajamani, and P. Seiler. Stop and go cruise control, 1999.

[10] T. Kailath. *Linear Systems.* Prentice-Hall, 1980.

[11] K. Krishnaswamy and D. Bugajski. Inversion based multibody control: Launch vehicle with fuel slosh. In *AIAA Guidance, Navigation and Control Conference*, pages AIAA 2005–6149, 2005.

[12] L. Ljung. *System Identification: Theory for the User.* Prentice Hall, 1999.

[13] By Matthias93 (Own work) [Public domain], via Wikimedia Commons, 2010.

[14] Precision Microdrives. DC motor model 112-002 product data sheet. wwww.precisionmicrodrives.com, 2016.

[15] NASA. Gimbaled thrust. http://microgravity.grc.nasa.gov/education/rocket/gimbaled.html, 2016.

[16] N. Nise. *Control Systems Engineering.* Wiley, 2015.

[17] K. Ogata. *Discrete-Time Control Systems.* Pearson, 2nd edition, 1995.

[18] K. Ogata. *Modern Control Engineering.* Pearson, 2009.

[19] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, 3rd edition, 2009.

[20] A. Packard. Notes on loopshaping. University of California, Berkeley, 2011.

[21] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design.* John Wiley, 2nd edition, 2005.

[22] M.E. Taylor. *Introduction to Differential Equations.* American Mathematical Society, 2011.

[23] J.Y. Wong. *Theory of Ground Vehicles.* Wiley, 2008.

[24] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, 1st edition, 1995.