Version:

TEX'd:
Feb. 17, 2014

1 Control Applications for Robotics

PREVIEW

We show that the transfer function and conditions of stability for linear systems can be studied using Laplace transforms.

Table 1.1 summarizes example applications of Laplace transform techniques. This chapter emphasizes control applications but briefly describes the other application areas.

TABLE 1.1 *Applications of Laplace transforms*

Area	*Application*
Stability	Stability of a linear system can be determined by analyzing the transfer function given by the Laplace transform.
Control	Control systems can be analyzed and designed using Laplace transforms.

THE TRANSFER FUNCTION

The *transfer function* is defined for linear systems with zero initial conditions called *relaxed* systems. Let $h(t)$ be the impulse response and $H(s)$ its Laplace transform. For an input $f(t)$ with transform $F(s)$ and a response $y(t)$ with transform $Y(s)$, the transfer function can be defined in two equivalent ways:

1. in the Laplace domain as the ratio of the output $Y(s)$ to the input $F(s)$;
2. as the transform of the impulse response $h(t)$.

Since the convolution $y(t) = f(t) * h(t)$ describes the response of the relaxed system, the transformed equivalent is

$$Y(s) = F(s)H(s) \quad \text{or} \quad H(s) = Y(s)/F(s).$$

The transfer function $H(s)$ can be analyzed in many ways. Analyzing the pole-zero model is an approach often taken in stability and control applications.

THE POLE-ZERO MODEL

Consider a transfer function that can be written in the form

$$\begin{aligned} H(s) = \frac{Y(s)}{X(s)} &= \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0} \\ &= K\frac{(s-z_1)\cdots(s-z_m)}{(s-p_1)\cdots(s-p_n)}. \end{aligned} \tag{1.1}$$

Notice that if this transfer function describes a linear time-invariant system, the poles define the impulse response of the system since $Y(s) = 1$ and all the $b_i = 0$ except for $b_0 = 1$. Thus, the inverse transform yields the *impulse response* for the system in the form

$$h(t) = \sum_{i=1}^{n} A_i \exp(p_i t) \quad t \geq 0$$

where the coefficients A_i can be found from the partial fraction expansion of $H(s)$.

□ EXAMPLE 1.1 ***Pole-Zero Analysis***

Consider the transform

$$Y(s) = \frac{2s(s+1)}{(s+3)(s^2+4s+5)}$$

with zeros at $s = 0$ and $s = -1$. The poles are at $s = -3$ and $s = -2 \pm i$. These poles and zeros are sometimes called the finite poles and finite zeros respectively. Also, there is a zero at ∞ since $Y(s) = 0$ as $s \to \infty$ because the degree of the denominator is one greater than that of the numerator. In most applications, we consider only the finite poles and zeros.

■ **THEOREM 1.1** ***Stability from Laplace transform***

For the function

$$y(t) = \mathcal{L}^{-1}\left\{\frac{P(s)}{Q(s)}\right\}$$

to be stable, it is necessary and sufficient that the equation $Q(s) = 0$ have no roots to the right of the imaginary axis in the complex s plane and that any roots on the imaginary axis be unrepeated.

Considering the values of the poles of $Y(s)$ and Theorem 1.1, we conclude that the function $y(t) = \mathcal{L}^{-1}[Y(s)]$ is stable since the poles have negative real parts.

□

1.1 MATLAB AND LAPLACE APPLICATIONS

MATLAB has commands that can be used to apply Laplace transforms to analyze linear systems for control systems problems. The *Control System Toolbox* has additional commands for this purpose. Table 1.2 lists a few such commands that are useful for applications of Laplace transforms to continuous linear systems. To see the complete list of commands available use the command **help control**.

TABLE 1.2 *MATLAB commands for control applications*

Command	*Result*
Control System Toolbox:	
bode	Bode plot (Frequency response)
impulse	Impulse response
lsim	Response to arbitrary inputs
pole	Poles
pzmap	Pole zero plot
rlocus	Root locus plot
step	Step response
tf	Transfer function

□ EXAMPLE 1.2 *MATLAB System Analysis*

This example shows how to use MATLAB commands to aid in determining the characteristics including the step response of the system

$$Y(s) = \frac{F(s)}{(s+1)(s+2)}.$$

It is easily shown that the impulse response for the system is

$$y_{\text{impulse}}(t) = e^{-t} - e^{-2t} \quad t \geq 0$$

and the step response is the integral of the impulse response,

$$y_{\text{step}}(t) = \frac{1}{2} - e^{-t} + \frac{1}{2}e^{-2t} \quad t \geq 0.$$

You can check these results using the initial and final value theorems.

The M-file for this example first defines the system characteristics from the coefficients of the numerator and denominator polynomial using the command transfer function **tf**. This command creates the transfer function for the system for other commands. The command **pole** computes the poles of the transfer function ($s = -1, -2$).

Then, the transform of the step response is defined. The call to **pzmap** plots the poles of the function corresponding to the poles of the step response including the pole at $s = 0$ as shown in Figure 1.1. In this case, there are no zeros in the function. Finally, the command **step** plots the step response of the system as Figure 1.2.

The first figure shows the pole-zero plot of the step response with poles at $s = 0, -1, -2$. The second figure is the step response. The differential equation for the system step response is given by

$$(s^2 + 3s + 2)Y(s) = 1/s$$

or $\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 1$. Note that as $t \to \infty$, $y(t) = 1/2$ in steady state. By the Final Value Theorem $y(t) = \lim_{s \to 0} sY(s) = 1/2$.

MATLAB Script

```
Example 1.2
% PZEXAMP.M Example of pole-zero plot and step response of the
%  system with response H(s)F(s)=F(s)/(s^2+3s+2).
%  CALLS: pole, pzmap, step, tf in the Controls System Toolbox
%
clear,clf
num=[1];         % Numerator coefficients F(s)=1
den=[1 3 2];     % Denominator coefficients for system
sys=tf(num,den); % Define system for pole, step.
pole(sys)        % Compute poles and display results
% Poles and zeros of step response F(s)=1/s
figure(1)
denstep=[1 3 2 0];       % Denominator for step input
sysstep=tf(num,denstep)  % Compute and display transform
pzmap(sysstep)           % Plot the poles and zeros
title(’Pole-zero plot for step response’)
```

```
% Step response
figure(2)
step(sys)        % Compute and plot step response
```

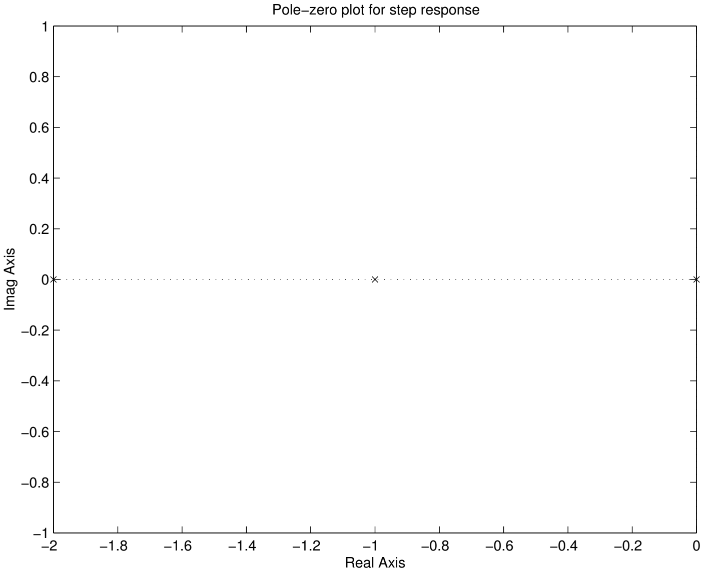

FIGURE 1.1 *Pole-zero plot for step response*

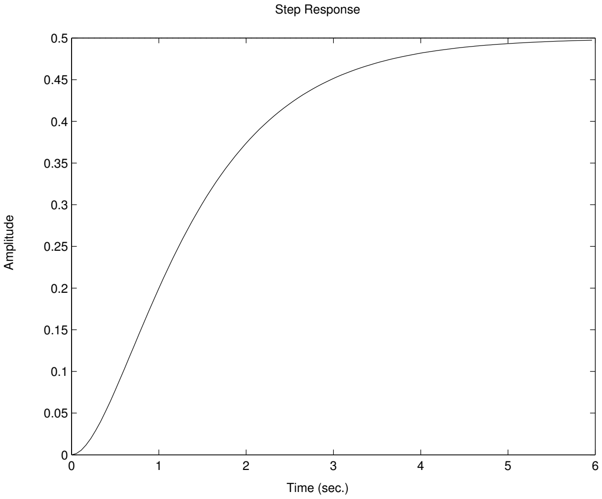

FIGURE 1.2 *Step response*

□

1.2 PHYSICAL SYSTEMS AND THEIR RESPONSE

In this section, we consider the response of linear time-invariant systems using the Laplace transform approach. In particular, first- and second-order systems are considered in some detail. Such systems are important as models of physical systems and also the response of higher-order systems often can be analyzed as the sum of the responses of first- and second-order systems.

FIRST ORDER SYSTEMS

Consider the system described by the first-order differential equation

$$\frac{dy(t)}{dt} + \frac{1}{\tau}y(t) = \frac{K}{\tau}f(t) \tag{1.2}$$

with τ and K constants with the Laplace transform

$$sY(s) - y(0) + \frac{1}{\tau}Y(s) = \frac{K}{\tau}F(s).$$

For the relaxed system, $y(0) = 0$ so that

$$Y(s) = \frac{(K/\tau)F(s)}{s + (1/\tau)} \tag{1.3}$$

with the step response ($F(s) = 1/s$),

$$y(t) = K(1 - e^{-t/\tau}) \quad t \geq 0. \tag{1.4}$$

For control applications, the step response and the frequency response of the system are of fundamental importance. First, we present these characteristics analytically and then use MATLAB to analyze first-order systems.

Step Response of First-Order Systems The parameter τ in Equation 1.4 is called the *time constant* of the system and has units of seconds if t is in seconds. Calculation of the magnitude of the step response verses multiples of the time constant shows that the response rises from zero to about 63% of the final value in time $t = \tau$. After $t = 4\tau$, the response is within 98.12% of the value K.

□ **EXAMPLE 1.3** ***Time Constants for First-Order Systems***

Given the transform of the step response of the system

$$Y(s) = \frac{(K/\tau)}{s[s + (1/\tau)]}, \tag{1.5}$$

the pole at $s = -1/\tau$ corresponds to a time constant of τ. The accompanying MATLAB script creates the system parameters for three values of τ and plots the step responses in Figure 1.3 for a system with $K = 5$. The poles at $s = -4, -2, -0.2$ correspond to the values $\tau = 0.25, 0.5, 5.0$, respectively.

MATLAB Script

```
Example 1.3
% TSTSTEP.M Plot of step response for several systems
%   Test case Y(s)= (5/tau)/[s(s+(1/tau))]  for tau = .25,.5, 5
%
clear, clf
num=[5/.25 5/.5 5/5]  % K/tau
den1=[1 4]      % tau =.25
den2=[1 2]      % tau = .5
den3=[1 .2]     % tau = 5
%
sys1=tf(num(1),den1);sys2=tf(num(2),den2);sys3=tf(num(3),den3)
step(sys1,sys2,sys3,0:.01:10)
grid
gtext('\tau = 5.0');gtext('\tau = 0.5');gtext('\tau = 0.25')
```

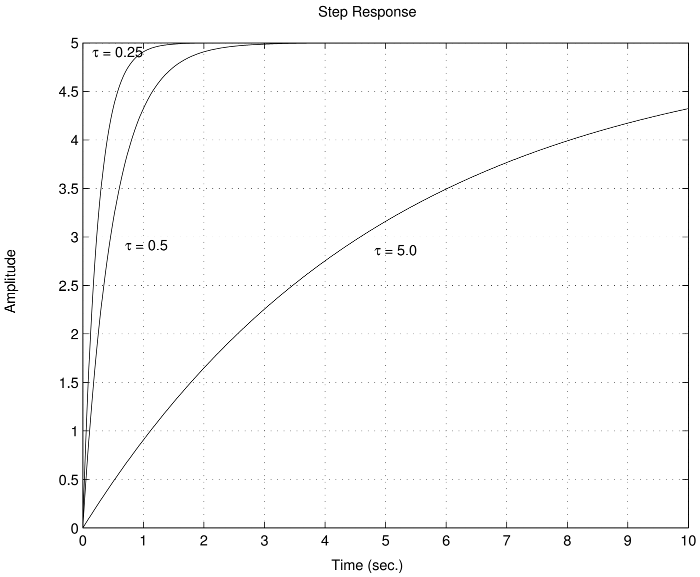

FIGURE 1.3 *Step responses*

□

The conclusion from the figure is that as the time constant of the system increases, the time to steady state increases proportionally. Steady state is sometimes defined as the time at which the response is 95% of the final value. This is about 3 time constants when $1 - e^{-t/\tau} = .05$.

Frequency Response of First-Order Systems The *frequency response* of a system is determined by using a sinusoid as the input function and computing (or measuring) the output for a range frequencies of interest. In terms of the transfer function for a first-order system

$$H(s) = \frac{Y(s)}{F(s)} = \frac{(K/\tau)}{s + (1/\tau)}, \tag{1.6}$$

the frequency response is given by $G(s)$, evaluated at $s = i\omega$. Thus, the frequency response of a system with transfer function $H(s)$ is defined as

$$H(i\omega) \quad 0 \le \omega < \infty. \tag{1.7}$$

From the first-order transfer function of Equation 1.6, we find that

$$H(i\omega) = \frac{K}{\tau\,(i\omega) + 1} = |H(i\omega)|\, e^{i\phi(\omega)}$$

so that the magnitude and phase of the frequency response can be written

$$|H(i\omega)| = \frac{K}{[1 + \tau^2\,\omega^2]^{1/2}} \qquad \phi(\omega) = -\tan^{-1} \tau\,\omega. \tag{1.8}$$

□ EXAMPLE 1.4 ***Bandwidth of First-Order Systems***

Let ω_{B} be the frequency at which the magnitude of the first-order transfer function is $1/\sqrt{2} = 0.707$ times the amplitude at very low frequencies.[1] Equating the first-order transfer function of Equation 1.8 at $\omega = \omega_{\mathrm{B}}$ to $K/\sqrt{2}$ yields

$$|H(i\omega_{\mathrm{B}})| = \frac{K}{[1 + \tau^2\omega_{\mathrm{B}}^2]^{1/2}} = \frac{K}{\sqrt{2}}$$

so that

$$\omega_{\mathrm{B}} = \frac{1}{\tau}$$

is the bandwidth in radians per second of a first-order system with time constant τ seconds.

To visualize the frequency response, the magnitude and phase of $H(i\omega)$ can be plotted versus frequency on linear scales. Another representation is the *Bode* plot or diagram, named for H. W. Bode, who plotted the magnitude on a logarithmic scale with the logarithm of frequency as the ordinate. On a Bode diagram, the phase of the transfer function is also plotted versus the logarithm of frequency.

The plot of the magnitude is in units of decibels for which

$$dB = 20 \log A$$

where A is the value to be converted to dB. To examine $H(i\omega)$ in dB, use the property of logarithms

$$\log\left(\frac{ab}{cd}\right) = \log(ab) - \log(cd) = \log a + \log b - \log c - \log d.$$

[1] This is often called the *half-power point* since if $|H|$ is measured in volts, the associated power is proportional to $|H|^2$. This is also called the -3dB point for $|H|$.

The magnitude of the first-order transfer function defined by Equation 1.8 in dB is

$$20 \log |H(i\omega)| = 20 \log K - 20 \log \left[1 + \left(\frac{\omega}{\omega_{\mathrm{B}}} \right)^2 \right]^{1/2} \tag{1.9}$$

if τ is replaced with $1/\omega_{\mathrm{B}}$.

For our numerical example, let $K = 1$ in Equation 1.9 so that at $\omega = 0$,

$$|H(0)| = 0 \text{ dB}.$$

Then, at $\omega = \omega_{\mathrm{B}}$,

$$|H(i\omega_{\mathrm{B}})| = -3.01 \text{ dB}$$

and at $\omega = 10\omega_{\mathrm{B}}$,

$$|H(i10\ \omega_{\mathrm{B}})| = -20.04 \text{ dB}.$$

The magnitude of the transfer function is reduced by approximately 3 dB at the frequency defining the bandwidth and by about 20 dB a decade higher in frequency.

The MATLAB script and the plots of magnitude and phase show that the bandwidth (-3 dB point) occur where $\omega = 1/\tau$. The phase is then -45 degrees.

MATLAB Script

```
Example 1.4
% BODETST.M Bode plots for first-order systems
%   Test case Y(s)= (1/tau)/[s(s+(1/tau))]  for tau = .25,.5, 5
%
clear, clf
num=[1/.25 1/.5 1/5]  % K/tau
den1=[1 4]      % tau =.25
den2=[1 2]      % tau = .5
den3=[1 .2]     % tau = 5
%
sys1=tf(num(1),den1);sys2=tf(num(2),den2);sys3=tf(num(3),den3)
bode(sys1,sys2,sys3)
grid
gtext('\tau = 5.0');gtext('\tau = 0.5');gtext('\tau = 0.25')
```

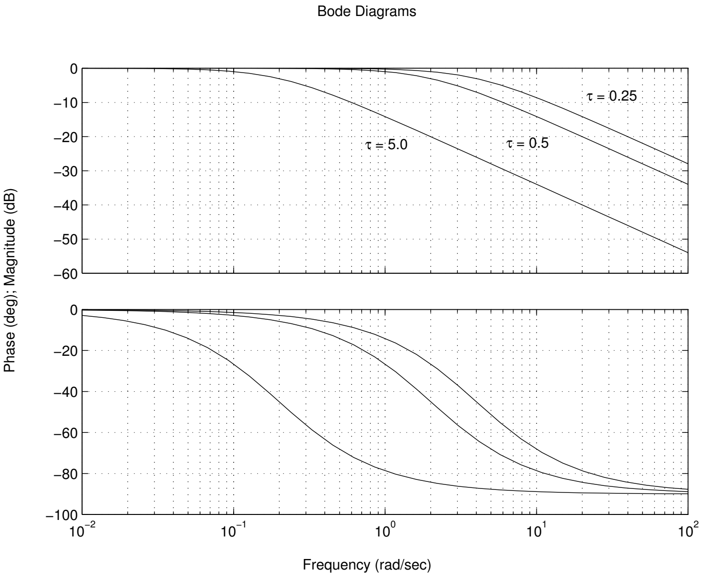

FIGURE 1.4 *Bode plots for first-order system*

□

Notice that for the equation

$$\dot{y}(t) + ay(t) = f(t), \qquad y(0) = 0 \tag{1.10}$$

where a is a constant, the solution can be written

$$y(t) = \int_0^t f(\lambda)\, e^{-a(t-\lambda)}\, d\lambda. \tag{1.11}$$

If $f(t) = \delta(t)$, using the sifting property of $\delta(t)$ we find

$$y(t) = \int_0^t \delta(\lambda)\, e^{-a(t-\lambda)}\, d\lambda = e^{-at} \quad t \geq 0$$

as the impulse response of the system. Note that a is the inverse of the time constant of the system in these equations. Using this result shows that Equation 1.11 is the convolution with $f(t)$ and the impulse response of the system. Chapter 5 in Harman presents another approach to finding the solution of the differential equation.

SECOND ORDER SYSTEMS

Consider the system described by the second-order differential equation

$$\frac{d^2y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + {\omega_n}^2 y(t) = {\omega_n}^2 f(t) \tag{1.12}$$

where ζ is the *damping ratio* and ${\omega_n}^2$ is the *natural* or *undamped* frequency. Both of these parameters are assumed positive constants in our discussion. The transfer function for this second-order system is

$$H(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} \tag{1.13}$$

which when written in this form has the *dc gain* $H(0) = 1$. The poles of the transfer function are

$$s_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2}.$$

If $\zeta = 0$, there is no damping and the inverse transform yields a sinusoidal impulse response of sustained amplitude. For $0 < \zeta < 1$, the impulse response is a damped sinusoid. For $\zeta \geq 1$, there is no oscillation in the response. The step response is explored in Example 1.5.

□ EXAMPLE 1.5 *Second-Order Step Response*

The accompanying script computes and plots the step response for a second-order system of Equation 1.12 using the value $\omega_n = 1$ with the different damping ratios

$$\zeta = 0.25, 0.5, 1.0, 5.0.$$

MATLAB Script

```
Example 1.5
% TSTSTEP2.M Plot of step response for several systems
%   Test case Y(s)= (1)/[s(s+2*zeta*+1)]  for zeta=0.25,.5,1,5
%
clear, clf
zeta=[0.25 0.5 1 5];
for k=1:4
   num=[1];                 %
   den=[1 2*zeta(k) 1];     %
   step(num,den)            % Plot the result
   hold on
end
%
grid
gtext(’\zeta = .25’);gtext(’\zeta = 0.5’);
gtext(’\zeta = 1.0’);gtext(’\zeta = 5.0’);
```

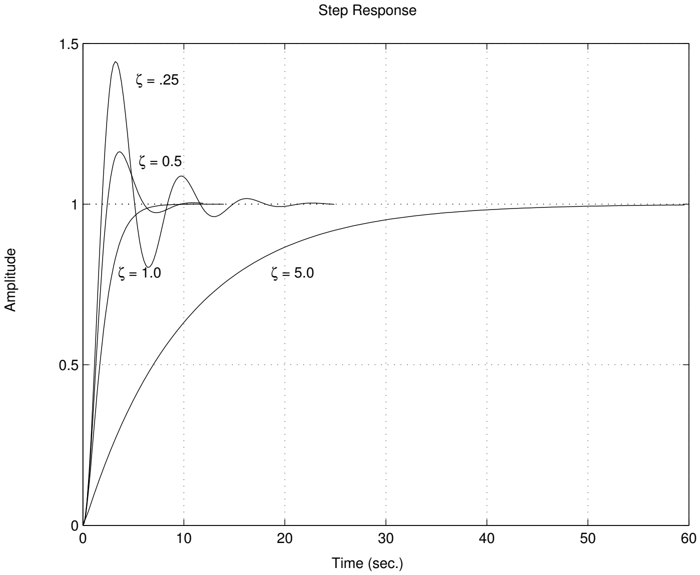

FIGURE 1.5 *Step response for second-order system*

□

Frequency Response of Second-Order Systems The frequency response function for the second-order system of Equation 1.13 is

$$H(i\omega) = \frac{{\omega_n}^2}{(i\omega)^2 + 2\zeta\omega_n(i\omega) + {\omega_n}^2}. \qquad \textbf{(1.14)}$$

The magnitude of $H(i\omega)$ becomes

$$\begin{aligned} |H(i\omega)| &= \left| \frac{1}{\left(i\frac{\omega}{\omega_n}\right)^2 + 2\zeta\left(i\frac{\omega}{\omega_n}\right) + 1} \right| \\ &= \frac{1}{\sqrt{\left(1 - \frac{\omega^2}{\omega_n^2}\right)^2 + \left(2\zeta\frac{\omega}{\omega_n}\right)^2}} \end{aligned} \qquad \textbf{(1.15)}$$

From the result, we find that if $\omega << \omega_n$, the low-frequency asymptote is

$$-20\log 1 = 0\text{dB}$$

and the high-frequency response is a line having a slope of -40dB/decade since for $\omega >> \omega_n$,

$$-20\log\frac{\omega^2}{\omega_n^2} = -40\log\frac{\omega}{\omega_n}.$$

□ EXAMPLE 1.6 *Second-Order Frequency Response*

The accompanying MATLAB script calculates and plots the magnitude of the frequency response of the second-order system of Equation 1.15 for various values of damping.

MATLAB Script

```
Example 1.6
% FREQ2ORD.M Frequency response of second order system for values
%  of damping  zeta=[0.25 0.5 0.707 1]
%
clear;clf
w=0:.05:3;                 % Frequency points (rad/sec)
zeta=[0.25 0.5 0.707 1]    % Values of damping
for k=1:4
   Hnum=[0 0 1];
   Hden=[1 2*zeta(k) 1];
   Hiomega=freqs(Hnum,Hden,w);
   Hmag=abs(Hiomega);
   plot(w,Hmag)
     title('Frequency response of second-order system')
     xlabel('\omega/\omega_n')
     ylabel('|H(i \omega)|')
      hold on
end
grid
gtext('\zeta = .25');gtext('\zeta = 0.5');
gtext('\zeta = .707');gtext('\zeta = 1.0');
```

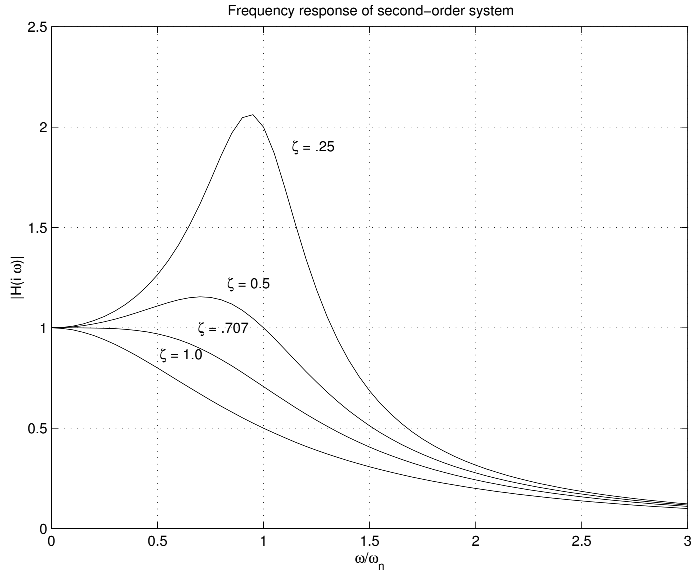

FIGURE 1.6 *Frequency response for second-order system*

□

1.3 CONTROL APPLICATIONS OF LAPLACE TRANSFORMS

This section introduces feedback control systems and their analysis by application of Laplace transform techniques. For our purposes, a *feedback control system* is a system that maintains a prescribed relationship between the output and some reference input. When the control action is based on the difference between the desired output and the actual output, the system is often called a *closed-loop control system*. For comparison, a control system without feedback is called an *open-loop control system*. Table 1.3 compares these and presents examples of each.

TABLE 1.3 *Comparison of control systems*

Open-loop system	*Closed-loop system*
Output is not measured or compared to input.	Controller brings output to desired value.
Stability is inherent.	Must be designed to be stable.
Calibration necessary.	Relatively insensitive to internal changes or external disturbances.
Examples	
Timed traffic lights	Thermostatic control
Washing machine	Automobile cruise control

BLOCK DIAGRAM MODEL

The block diagram of Figure 1.7 shows a closed-loop control system. The designation within the blocks indicate the transfer functions for various components of the system. For example, $G(s)$ can represent the transfer function of the portion of the system to be controlled, sometimes called the *plant*, and the controller or compensator that varies the input to the plant to obtain the desired output. For our present purposes, it is not necessary to further define $G(s)$ in terms of its components. The function $H_f(s)$ is the transfer function of the feedback element, usually a sensor that measures the output $Y(s)$. In this simple diagram, $E(s)$ is called the error signal and $R(s)$ is the reference input signal.

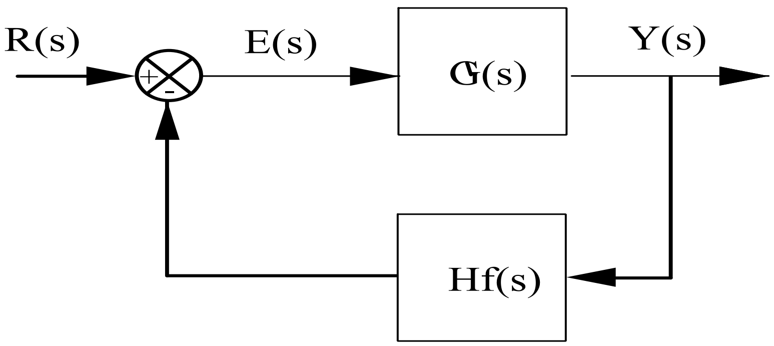

FIGURE 1.7 *Feedback control model*

If the input $R(s)$ is constant, we say the control system is a *regulator* since the object is to maintain the output at some constant value in the presence of disturbances. Temperature control or speed-control closed-loop systems are of this type. Other control systems are designed to allow the output to follow some time function as input. Such systems that control mechanical position or motion are called *servomechanisms*.

The transfer function $Y(s)/R(s)$ for the system in Figure 1.7 is derived by observing that

$$\begin{aligned} Y(s) &= G(s)E(s) \\ E(s) &= R(s) - H_f(s)Y(s) \end{aligned}$$

and eliminating $E(s)$ from these equations to yield

$$Y(s) = G(s)[R(s) - H_f(s)Y(s)]$$

so that

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H_f(s)}. \tag{1.16}$$

This is the *closed-loop transfer function*.

□ **EXAMPLE 1.7** ***First-Order Feedback Response***

Consider the transfer function of a first-order system in the form

$$G(s) = \frac{Ka}{s+a}$$

where a is the reciprocal of the time constant for the system and K is a positive constant. According to Example 1.4, this parameter a is the bandwidth of the system. The dc gain for this system is K as seen by letting $s = 0$ in the expression for $G(s)$. One characteristic of the system that is of interest in some problems is the *gain-bandwidth product* which is

$$\text{GainBW} = Ka.$$

For purposes of comparison, this will be called the open-loop system.

In Figure 1.7, let $G(s)$ be the first-order transfer function and let the system have unity feedback $H_f(s) = 1$ so that the closed-loop transfer function is

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{Ka}{s+a(1+K)}.$$

according to Equation 1.16. In this case, the gain-bandwidth product is

$$\text{GainBW} = \frac{K}{1+K}a(1+K) = Ka,$$

the same as for the open-loop system. The conclusion is that the closed-loop gain is decreased by $1+K$ and the bandwidth is increased by the same factor as compared to the system without feedback. However, for first-order systems, the gain-bandwidth product is constant.

□

CONTROL METHODS

Consider the system shown in Figure 1.8. This represents a control loop with a *compensator* or *controller* that serves to produce a control signal that reduces the error to zero or a small value.

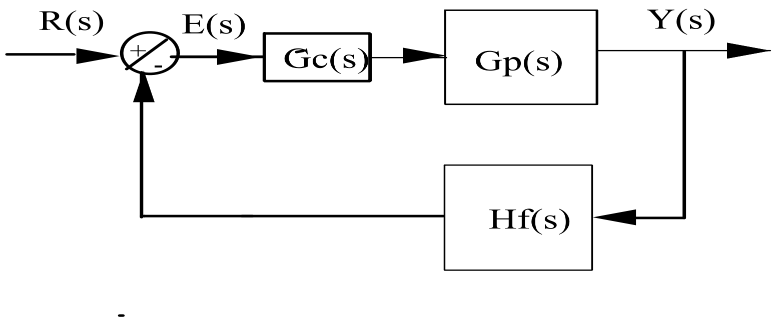

FIGURE 1.8 *Feedback control model with compensator*

In the figure, $G_c(s)$ is the transfer function of the *compensator* and $G_p(s)$ is the transfer function of the plant or process to be controlled. Using Equation 1.16, the closed-loop transfer function for the system is found to be

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)H_f(s)}. \tag{1.17}$$

Many control systems can be modeled this way and the design problem is to select $G_c(s)$ to get the desired system response. Our approach in this section is to analyze systems with different forms of $G_c(s)$.

PID Controllers Perhaps the most common form of compensator used in feedback control systems is the proportional-plus-integral-plus-derivative (PID) compensator. In Figure 1.8, consider $e(t)$ the input signal to the compensator and $m(t)$ the output in the time domain. The PID compensator or controller is defined by the equation

$$m(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de(t)}{dt}. \qquad \textbf{(1.18)}$$

The Laplace transform of $m(t)$ yields the transfer function is

$$G_c(s) = \frac{M(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s. \qquad \textbf{(1.19)}$$

In practice, all three terms may not be implemented in a control systems. For example, a PI compensator is commonly used with the characteristics described by Equation 1.18 after setting $K_D = 0$. We will study several types of compensators in this section.

Proportional Control If $G_c(s) = K_P$, a non-zero positive constant, in Figure 1.8, the control method is called *gain compensation* or more commonly *proportional control*. To study the effect of proportional control, we analyze the response of a first-order system controlled by a compensator with constant gain. The transfer function for the first-order system

$$G_p = \frac{K/\tau}{s + 1/\tau}$$

with unity feedback is

$$\begin{aligned} T(s) = \frac{Y(s)}{R(s)} &= \frac{K_P G_p(s)}{1 + K_P G_p(s)} \\ &= \frac{K_P K/\tau}{s + (1 + K_P K)/\tau} \end{aligned}$$

The dc gain is

$$T(0) = \frac{K_P K}{1 + K_P K} = \frac{K}{K + 1/K_P}$$

so that the dc gain is reduced from the open-loop gain of G_p since $K_P \geq 0$.

The accompanying MATLAB script computes the step function response for the values of the proportional constant $K_P = 0.1, 0.3, 0.5$.

MATLAB Script

```
% PROCNTL1.M Proportional control example for the plant
%  Gp(s)=K/[s(s+1)(s+2)] for three values of Kp.
%
clear, clf
t=0:.01:20;       % Time steps
Kp=[0.1 0.3 0.5];
for I=1:3
   Gpnum=[0 0 0 4*Kp(I)]; Gpden=[1 3 2 0];
   Tnum=Gpnum; Tden=Gpnum+Gpden;step(Tnum,Tden,t)
   hold on
end; hold off
gtext(’K_{p}= 0.1’);gtext(’K_{p} = 0.3’);gtext(’K_{p}= 0.5’)
```

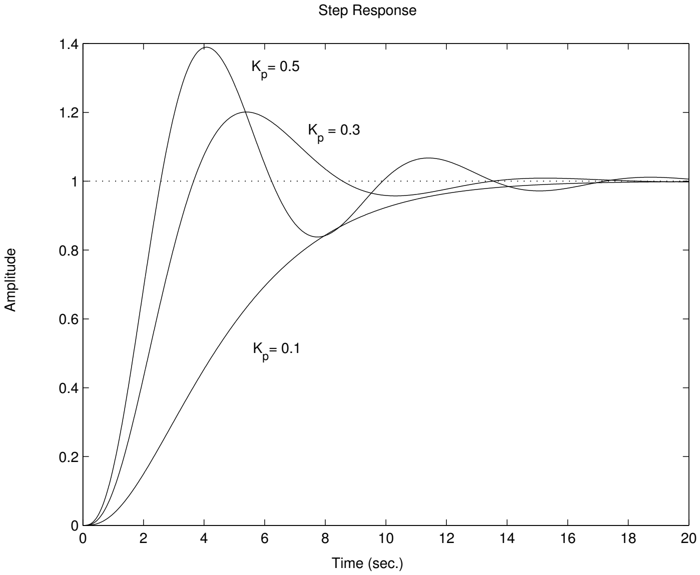

FIGURE 1.9 *Responses for proportional control*

In Figure 1.9 notice that the overall system response becomes more oscillatory as the proportional gain is increased. This implies that even an overdamped plant or process may become underdamped (oscillatory) with proportional control and an appropriate value of K_P.

To analyze the system in more detail we could consider the system characteristic equation given by Equation 1.17 as

$$1 + K_P G(s) H_f(s) = 0.$$

The roots of this equation could be plotted for various values of K_P to form a *root locus*. The MATLAB command **rlocus** computes the root locus and plots the result.

Comparison of Control Methods Each term in the controller equation

$$G_c(s) = \frac{M(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s.$$

has a different effect on the response of a system to an input signal.

□ **EXAMPLE 1.8** ***Control of First-Order System***

Using the first-order system of Example 1.3, the accompanying MATLAB script computes and plots the step response for the plant

$$G_p(s) = \frac{K/\tau}{s + 1/\tau} = \frac{2}{s+2}$$

for which the plant gain is 1 and the time constant is $\tau = 0.5$. The step response is computed for the open-loop plant, then in a closed-loop with proportional compensation, and then with integral compensation.

MATLAB Script

```
Example 1.8
% PID1.M Plot of step response for several control methods
%   Test case Open-loop Y(s)= (K/tau)/[s+(1/tau)]
%
clear, clf
tau=0.5;
K=1;                % Plant Gain
KP=1.5;             % Proportional Gain
KI=2.0;
numopen=[0 K/tau];   % K/tau
denopen=[1 1/tau];   % tau = 5
%
sysopen=tf(numopen,denopen);
% Proportional Control
numpro=KP*numopen; denpro=[1 (1+KP*K)/tau];
syspro=tf(numpro,denpro)
% Integral Control
numint=KI*numopen;
denint=[1 1/tau KI*K/tau];
sysint=tf(numint,denint)
%
step(sysopen,’-’,syspro,’--’,sysint,’:’,0:.005:6)
grid
legend(’Open-loop’)
gtext(’\tau =0.5’);gtext(’KP=2.0’);gtext(’KI = 2.0’)
%
% Note: Many of the expressions can be simplified
```

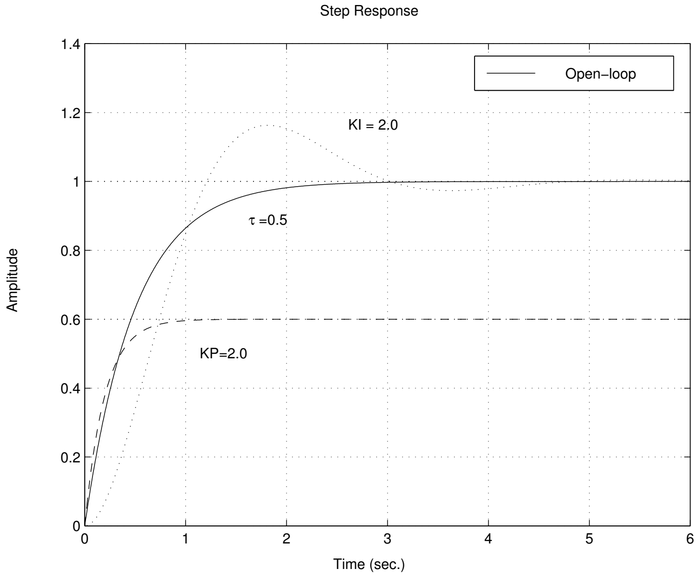

FIGURE 1.10 *First-order step responses for various control methods*

□

□ **EXAMPLE 1.9**

Control of First-Order System

A simpler way to find the transfer function of a closed loop system is to use the commands **feedback** after the transfer function for G_cG_p is computed by the command **series**.

MATLAB Script

```
Example 1.9
% PIDCNTL1.M Plot of step response for several control methods
%   Test case Open-loop Y(s)= (K/tau)/[s+(1/tau)]
% Use Control System Toolbox - Call feedback and series
%
clear, clf
tau=0.5
K=1                 % Plant Gain
KI=2.0              % Integral Gain
numopen=[0 K/tau]   % K/tau
denopen=[1 1/tau]   % tau = 0.5
sysopen=tf(numopen,denopen);
% Integral Control  KI/s
numintc=[KI]
denintc=[1 0]
sysintc=tf(numintc,denintc)
% Form feedback loop
syscp=series(sysintc,sysopen)
clsys=feedback(syscp,1,-1)
```

```
step(sysopen,’-’,clsys,’--’,0:.005:6)
grid
legend(’Open-loop’,’Integral’)
gtext(’\tau =0.5’);gtext(’KI = 2.0’)
```

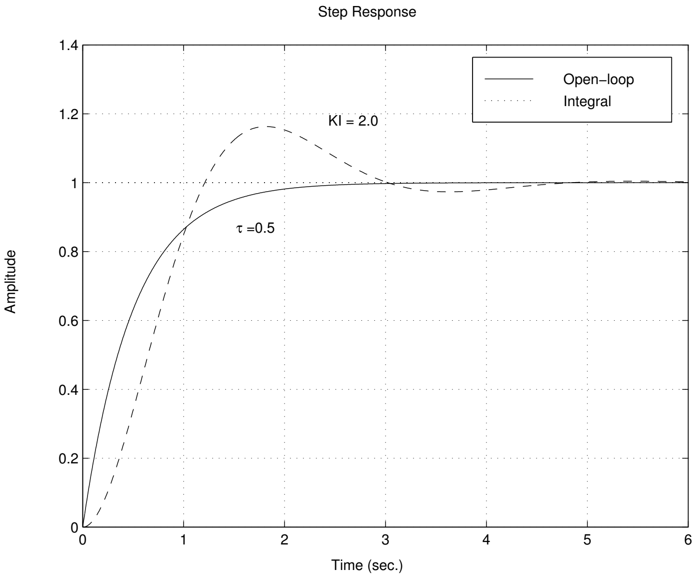

FIGURE 1.11 *First-order step response for PI control*

□