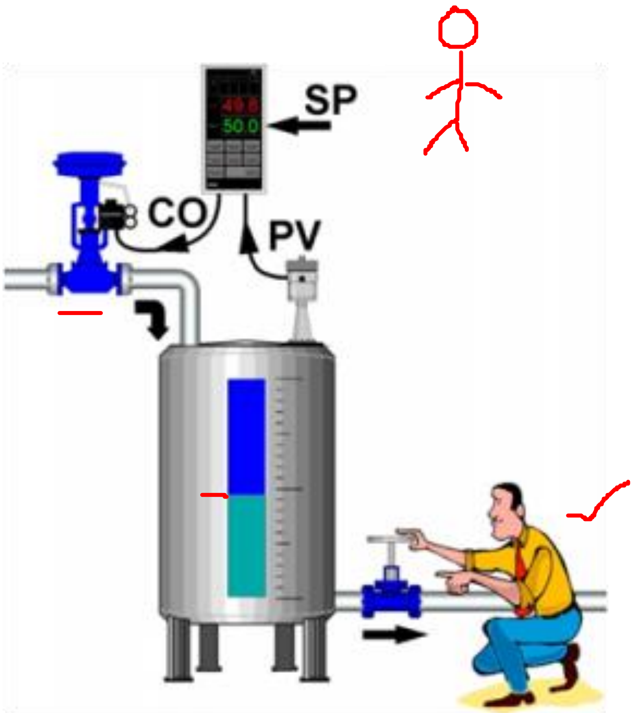
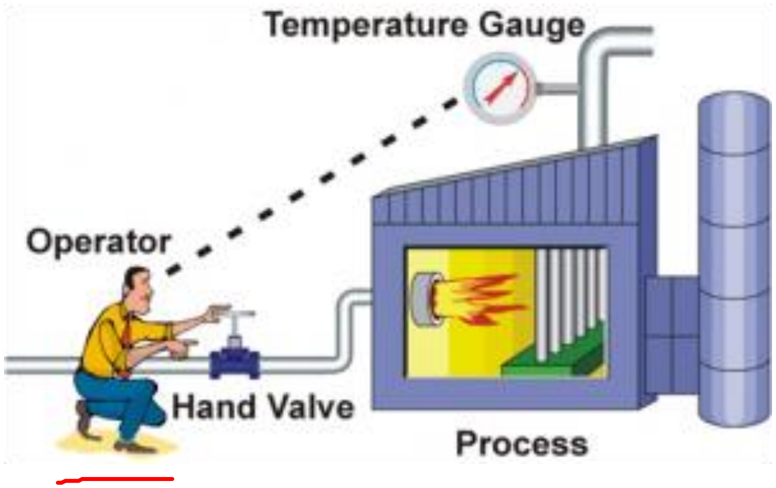
PID Controllers

$$CO = K_P e + K_I \int_0^t e(\tau)\,d\tau + K_D \frac{de}{dt}$$

$$= K_P \left(e + \frac{1}{T_i} \int + T_d \frac{de}{dt} \right)$$

Manual vs Automatic Control

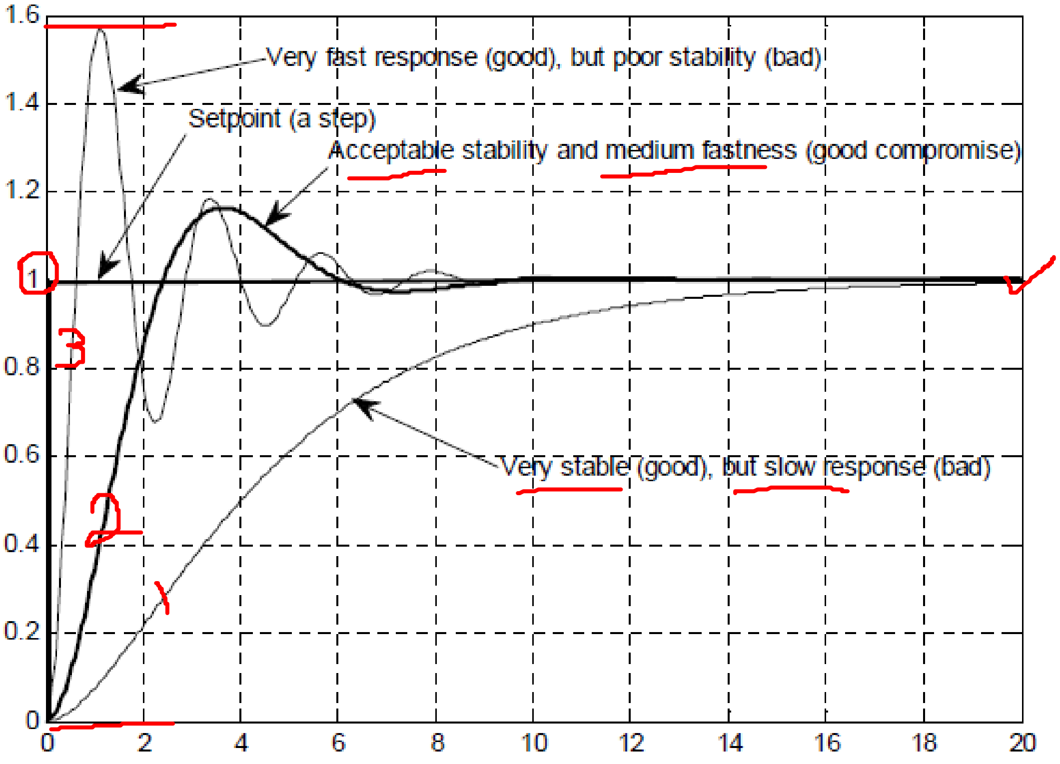




Acceptable stability, and medium fastness of response

What is the aim of the controller tuning? If it was possible to obtain, we would like to obtain both of the following for the control system:

- Fast responses, and
- Good stability
- The faster response, the worse stability, and
- The better stability, the slower response.



PID Controllers

Introduction

- More than half of the industrial controllers in use today utilize PID or modified PID control schemes.
- Many different types of tuning rules have been proposed in the literature.
  - ☞ Manual tuning on-site
  - ☞ On-line automatic tuning
  - ☞ Gain scheduling
- When the mathematical model of the plant is not known and therefore analytical design methods cannot be used, PID controls prove to be most useful.

PID Controller Design (Tuning)

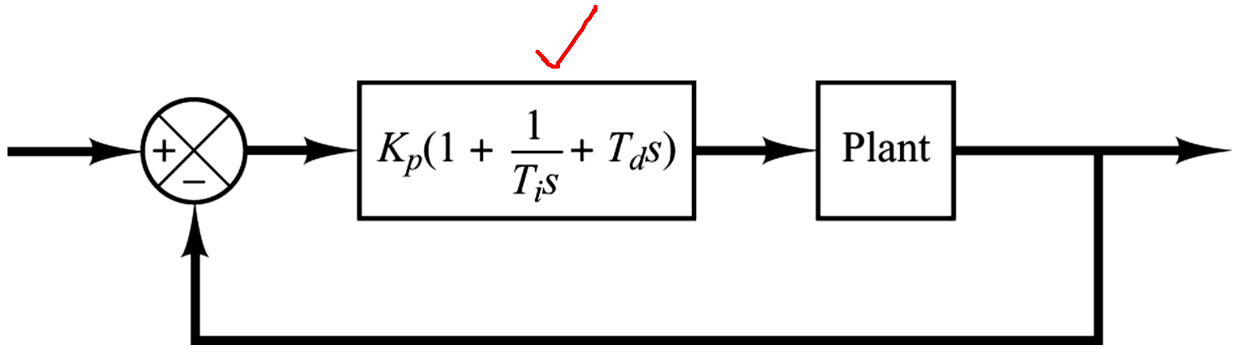


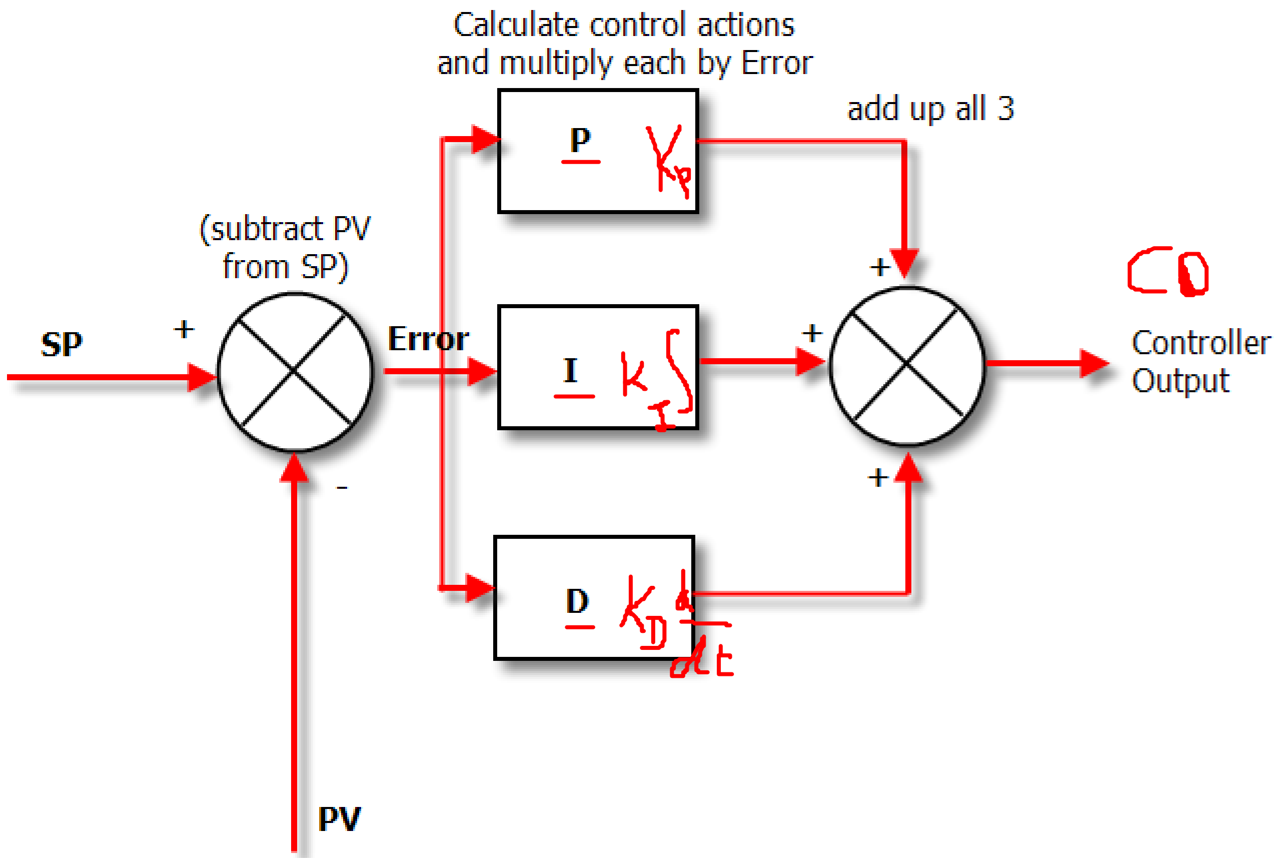
Figure 10-1 **PID control of a plant.**

<u>Design PID control</u>

- Know mathematical model ☞ various design techniques
- Plant is complicated, can't obtain mathematical model ☞ experimental approaches to the tuning of PID controllers

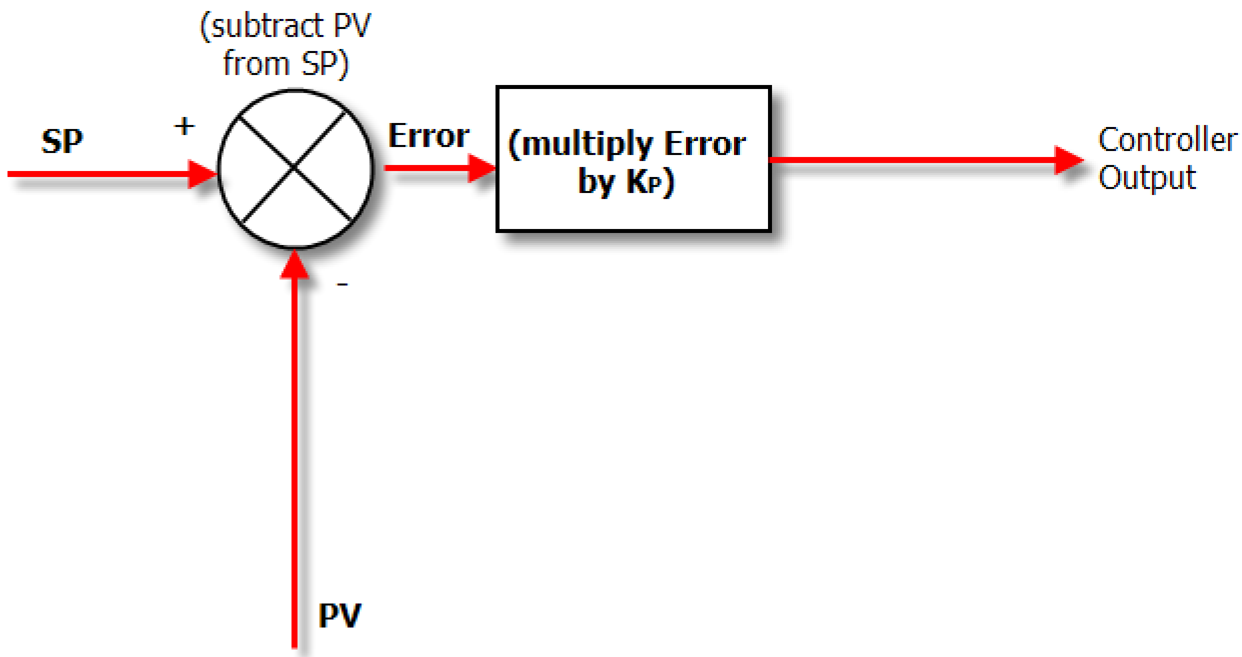
PID Controller Structure

$K_p \uparrow$ $t_r \downarrow$



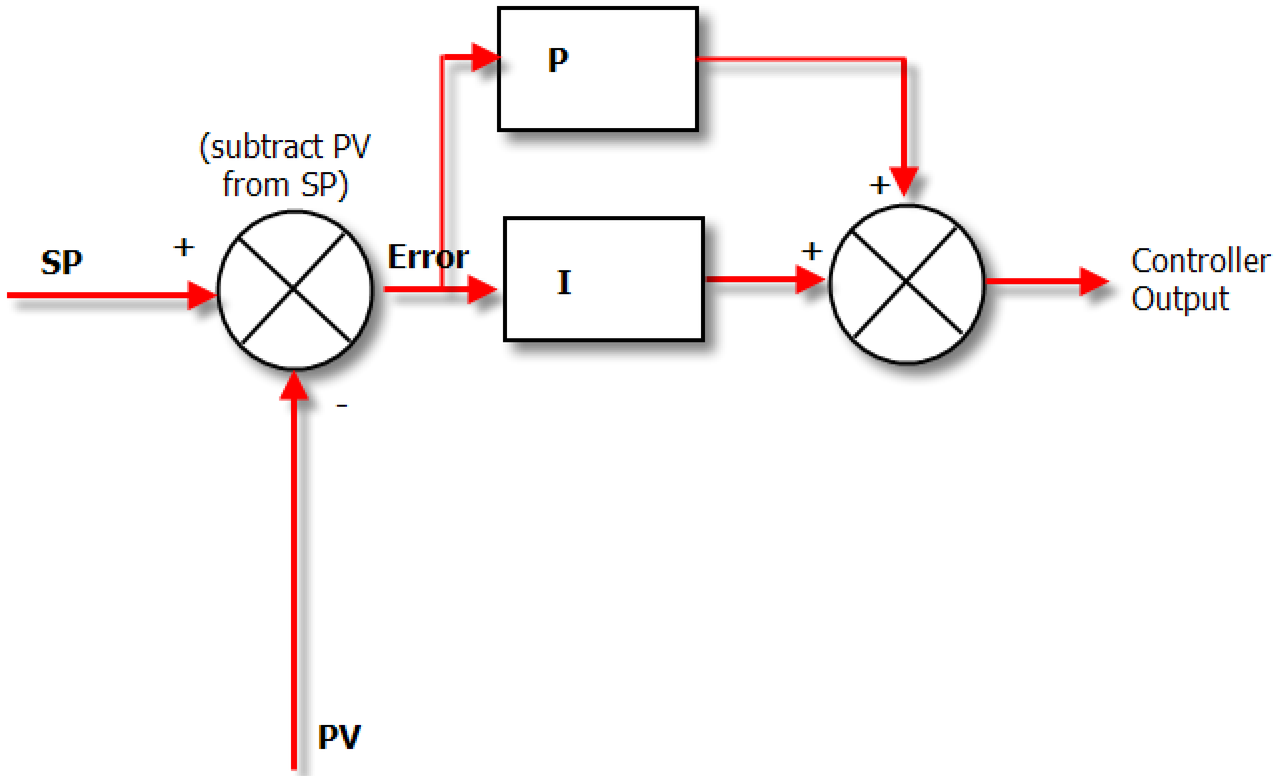
PID Controller Structure

Proportional Controller



PID Controller Structure

Proportional Controller



Ziegler-Nichols Rules for Tuning PID Controllers

-Ziegler and Nichols proposed rules for determining values of the proportional gain K_p, integral time T_i, and derivative time T_d based on the transient response characteristics of a given plant.

-Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on-site by experiments on the plant.

-Such rules suggest a set of values of K_p, T_i, and T_d that will give a stable operation of the system. However, the resulting system may exhibit a large maximum overshoot in the step response, which is unacceptable.

- We need series of fine tunings until an acceptable result is obtained.

Ziegler-Nichols 1st Method of Tuning Rule

- We obtain experimentally the response of the plant to a unit-step input, as shown in Figure 10-2.

-The plant involves neither integrator(s) nor dominant complex-conjugate poles.

-This method applies if the response to a step input exhibits an S-shaped curve.

- Such step-response curves may be generated experimentally or from a dynamic simulation of the plant.

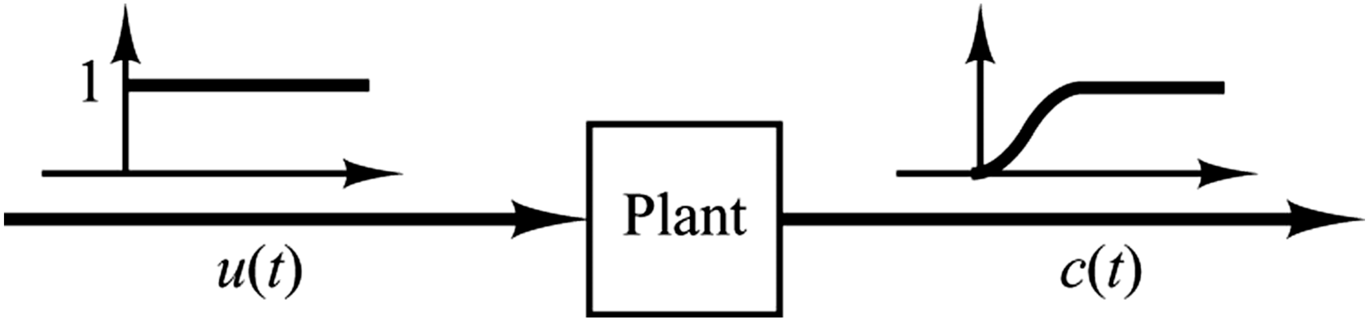
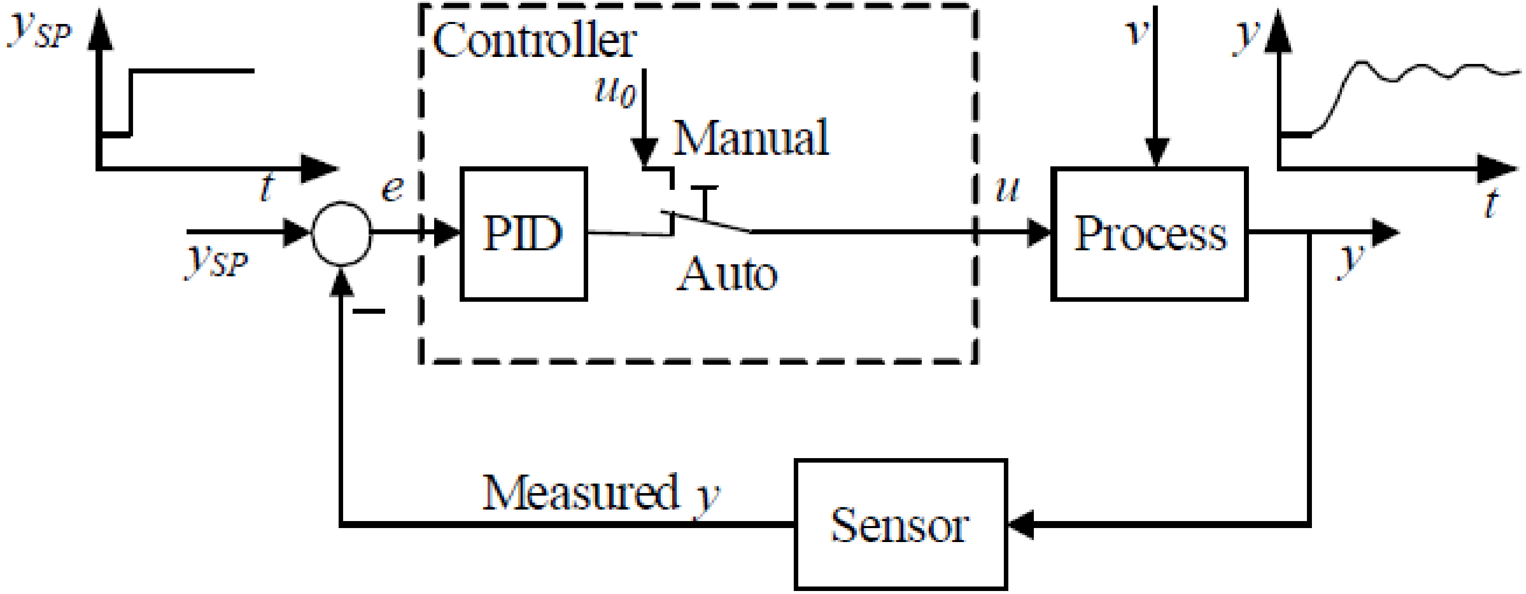


Figure 10-2 **Unit-step response of a plant.**

Ziegler-Nichols 1st Method of Tuning Rule

Needs to identify the system using a step response by putting the controller in manual mode



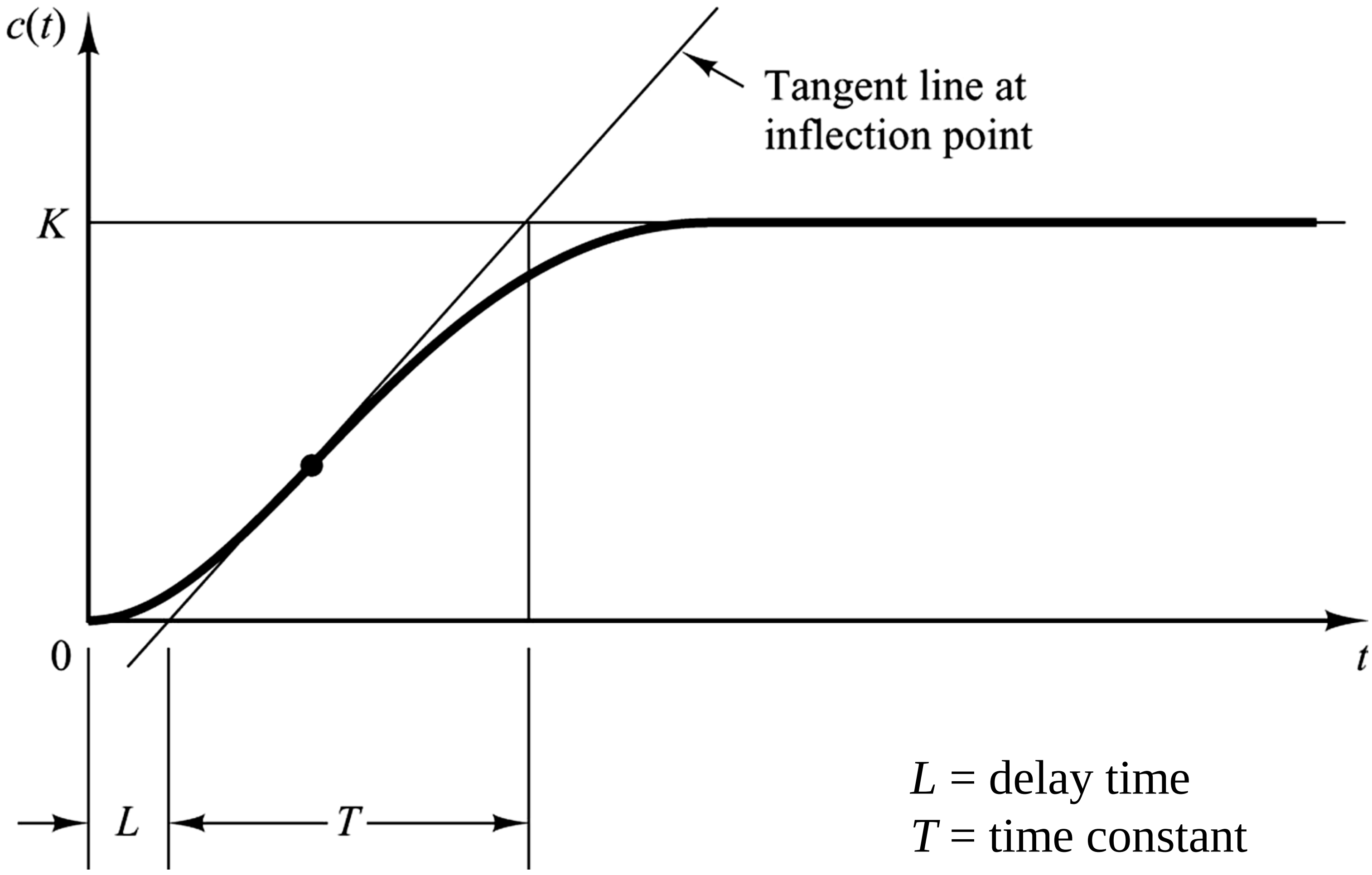


Figure 10-3 **S-shaped response curve.**

Transfer function: $$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1}$$

Ziegler and Nichols suggested to set the values of K_p, T_i, and T_d according to the formula shown in Table 10–1.

Table 8-1 **Ziegler–Nichols Tuning Rule Based on Step Response of Plant (First Method)**

Type of Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

Notice that the PID controller tuned by the first method of Ziegler–Nichols rules gives

$$
\begin{aligned}
G_c(s) &= K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \\
&= 1.2\,\frac{T}{L}\left(1 + \frac{1}{2Ls} + 0.5Ls\right) \\
&= 0.6T\,\frac{\left(s + \dfrac{1}{L}\right)^2}{s}
\end{aligned}
$$

Thus, the PID controller has a pole at the origin and double zeros at $s = -1/L$.

Ziegler-Nichols 2nd Method of Tuning Rule

1. We first set $T_i = \infty$ and $T_d = 0$. Using the proportional control action only (see Figure 10-4).

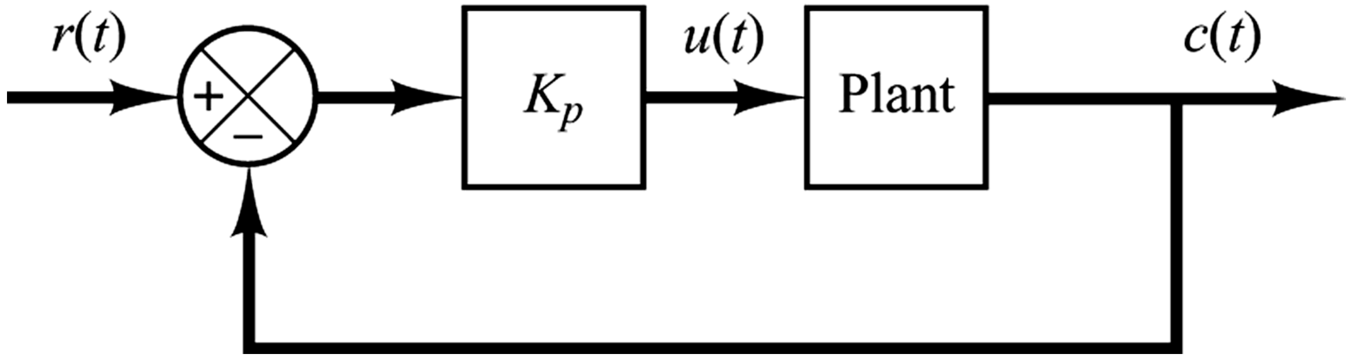


Figure 10-4 **Closed-loop system with a proportional controller.**

2. Increase K_p from 0 to a critical value K_{cr} at which the output first exhibits sustained oscillations.

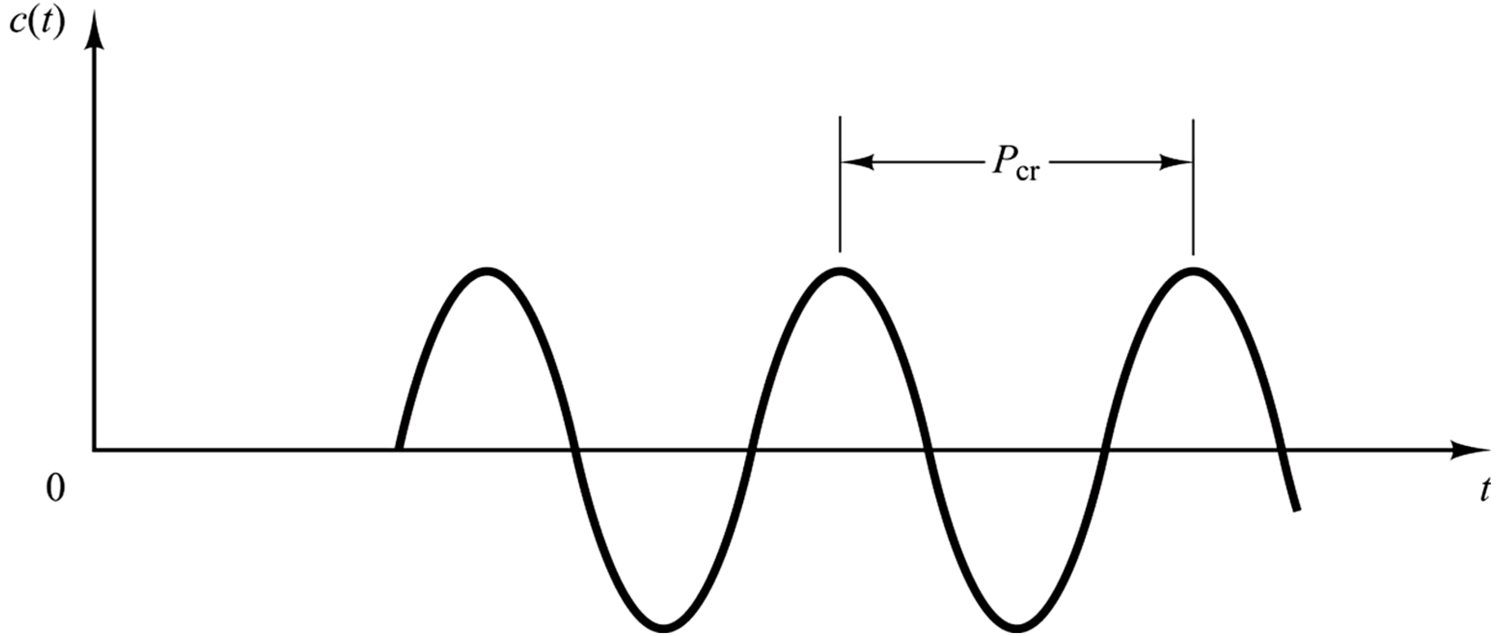


Figure 8-5 **Sustained oscillation with period P_{cr}. (P_{cr} is measured in sec.)**

☞ Ziegler and Nichols suggested that we set the values ofvthe parameters *K*,, *T*,, and *Td* according to the formula shown in Table 10-2.

Table 10-2 **Ziegler–Nichols Tuning Rule Based on Critical Gain K_{cr} and Critical Period P_{cr} (Second Method)**

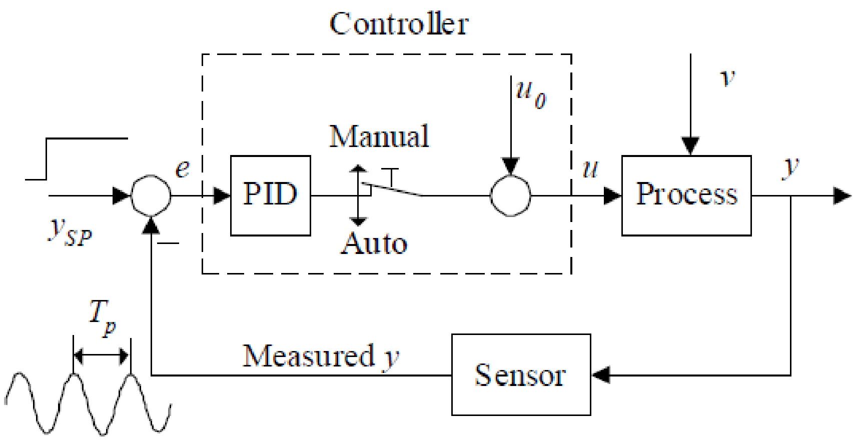
Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Ziegler-Nichols 2nd Method of Tuning Rule

1. Bring the process to (or as close to as possible) the specified *operating point* of the control system to ensure that the controller during the tuning is "feeling" representative process dynamic[6] and to minimize the chance that variables during the tuning reach limits. You can bring the process to the operating point by manually adjusting the control variable, with the controller in manual mode, until the process variable is approximately equal to the setpoint.

2. Turn the PID controller into a *P controller* by setting set $T_i = \infty$[7] and $T_d = 0$. Initially set gain $K_p = 0$. Close the control loop by setting the controller in automatic mode.

3. Increase K_p until there are *sustained oscillations* in the signals in the control system, e.g. in the process measurement, after an excitation of the system. (The sustained oscillations corresponds to the system



Notice that the PID controller tuned by the second method of Ziegler–Nichols rules gives

$$\begin{aligned} G_c(s) &= K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \\ &= 0.6K_{cr}\left(1 + \frac{1}{0.5P_{cr}s} + 0.125P_{cr}s\right) \\ &= 0.075K_{cr}P_{cr}\frac{\left(s + \frac{4}{P_{cr}}\right)^2}{s} \end{aligned}$$

Thus, the PID controller has a pole at the origin and double zeros at $s = -4/P_{cr}$.

Note that if the system has a known mathematical model (such as the transfer function), then we can use the root-locus method to find the critical gain K_{cr} and the frequency of the sustained oscillations ω_{cr}, where $2\pi/\omega_{cr} = P_{cr}$. These values can be found from the crossing points of the root-locus branches with the $j\omega$ axis. (Obviously, if the root-locus branches do not cross the $j\omega$ axis, this method does not apply.)

Comments. Ziegler–Nichols tuning rules (and other tuning rules presented in the literature) have been widely used to tune PID controllers in process control systems where the plant dynamics are not precisely known. Over many years, such tuning rules proved to be very useful. Ziegler–Nichols tuning rules can, of course, be applied to plants whose dynamics are known. (If the plant dynamics are known, many analytical and graphical approaches to the design of PID controllers are available, in addition to Ziegler–Nichols tuning rules.)

EXAMPLE 10–1

Consider the control system shown in Figure 10–6 in which a PID controller is used to control the system. The PID controller has the transfer function

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Although many analytical methods are available for the design of a PID controller for the present system, let us apply a Ziegler–Nichols tuning rule for the determination of the values of parameters K_p, T_i, and T_d. Then obtain a unit-step response curve and check to see if the designed system exhibits approximately 25% maximum overshoot. If the maximum overshoot is excessive (40% or more), make a fine tuning and reduce the amount of the maximum overshoot to approximately 25% or less.

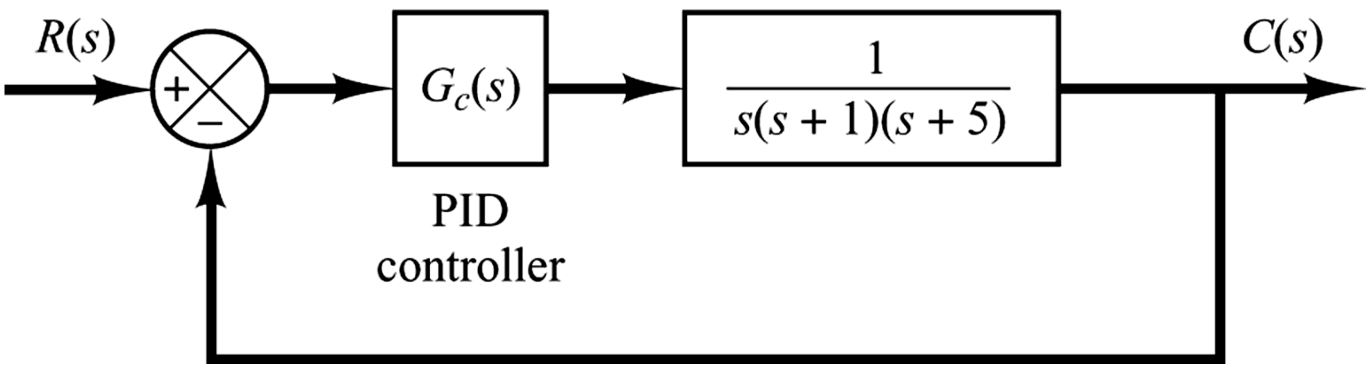


Figure 10-6 **PID-controlled system.**

Since the plant has an integrator, we use the second method of Ziegler–Nichols tuning rules. By setting $T_i = \infty$ and $T_d = 0$, we obtain the closed-loop transfer function as follows:

$$\frac{C(s)}{R(s)} = \frac{K_p}{s(s+1)(s+5) + K_p}$$

The value of K_p that makes the system marginally stable so that sustained oscillation occurs can be obtained by use of Routh's stability criterion. Since the characteristic equation for the closed-loop system is

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

the Routh array becomes as follows:

$$\begin{array}{lll} s^3 & 1 & 5 \\ s^2 & 6 & K_p \\ s^1 & \dfrac{30 - K_p}{6} & \\ s^0 & K_p & \end{array}$$

Examining the coefficients of the first column of the Routh table, we find that sustained oscillation will occur if $K_p = 30$. Thus, the critical gain K_{cr} is

$$K_{cr} = 30$$

With gain K_p set equal to K_{cr} $(= 30)$, the characteristic equation becomes

$$s^3 + 6s^2 + 5s + 30 = 0$$

To find the frequency of the sustained oscillation, we substitute $s = j\omega$ into this characteristic equation as follows:

$$(j\omega)^3 + 6(j\omega)^2 + 5(j\omega) + 30 = 0$$

or

$$6(5 - \omega^2) + j\omega(5 - \omega^2) = 0$$

from which we find the frequency of the sustained oscillation to be $\omega^2 = 5$ or $\omega = \sqrt{5}$. Hence, the period of sustained oscillation is

$$P_{cr} = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{5}} = 2.8099$$

Referring to Table 10–2, we determine K_p, T_i, and T_d as follows:

$$K_p = 0.6K_{cr} = 18$$

$$T_i = 0.5P_{cr} = 1.405$$

$$T_d = 0.125P_{cr} = 0.35124$$

The transfer function of the PID controller is thus

$$
\begin{aligned}
G_c(s) &= K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \\
&= 18\left(1 + \frac{1}{1.405s} + 0.35124s\right) \\
&= \frac{6.3223(s + 1.4235)^2}{s}
\end{aligned}
$$

The PID controller has a pole at the origin and double zero at $s = -1.4235$. A block diagram of the control system with the designed PID controller is shown in Figure 10–7.

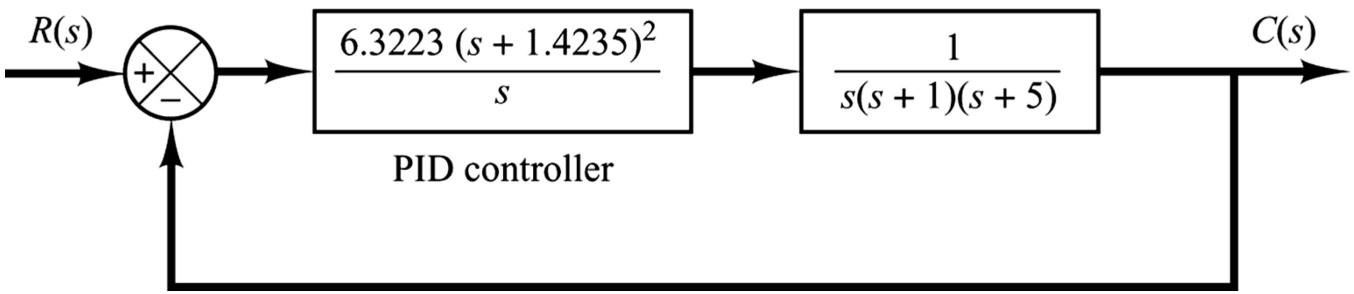


Figure 10-7 **Block diagram of the system with PID controller designed by use of the Ziegler–Nichols tuning rule (second method).**

Next, let us examine the unit-step response of the system. The closed-loop transfer function $C(s)/R(s)$ is given by

$$\frac{C(s)}{R(s)} = \frac{6.3223s^2 + 18s + 12.811}{s^4 + 6s^3 + 11.3223s^2 + 18s + 12.811}$$

MATLAB Program 10–1

```
% ---------- Unit-step response ----------
num = [0  0  6.3223  18  12.811];
den = [1  6  11.3223  18  12.811];
step(num,den)
grid
title('Unit-Step Response')
```

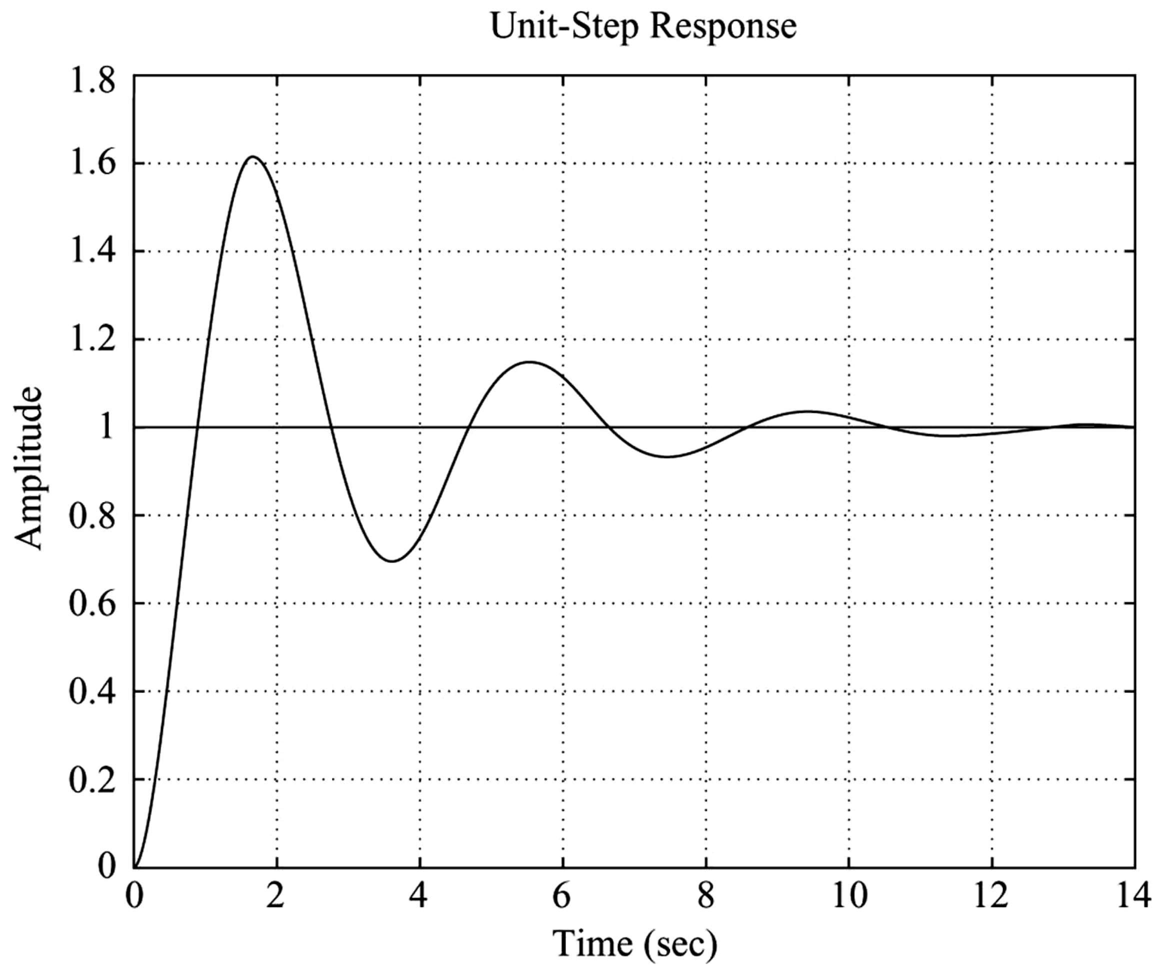


Figure 10-8 **Unit-step response curve of PID-controlled system designed by use of the Ziegler–Nichols tuning rule (second method).**

☞ The maximum overshoot in the unit-step response is approximately 62%.The amount of maximum overshoot is excessive. It can be reduced by fine tuning the controller parameters. Such fine tuning can be made on the computer. We find that by keeping $K_p = 18$ and by moving the double zero of the PID controller to $\boldsymbol{s} = -0.65$, that is, using the PID controller

$$G_c(s) = 18\left(1 + \frac{1}{3.077s} + 0.7692s\right) = 13.846\frac{(s + 0.65)^2}{s} \qquad (10\text{–}1)$$

the maximum overshoot in the unit-step response can be reduced to approximately 18%

☞ See Figure 10-9

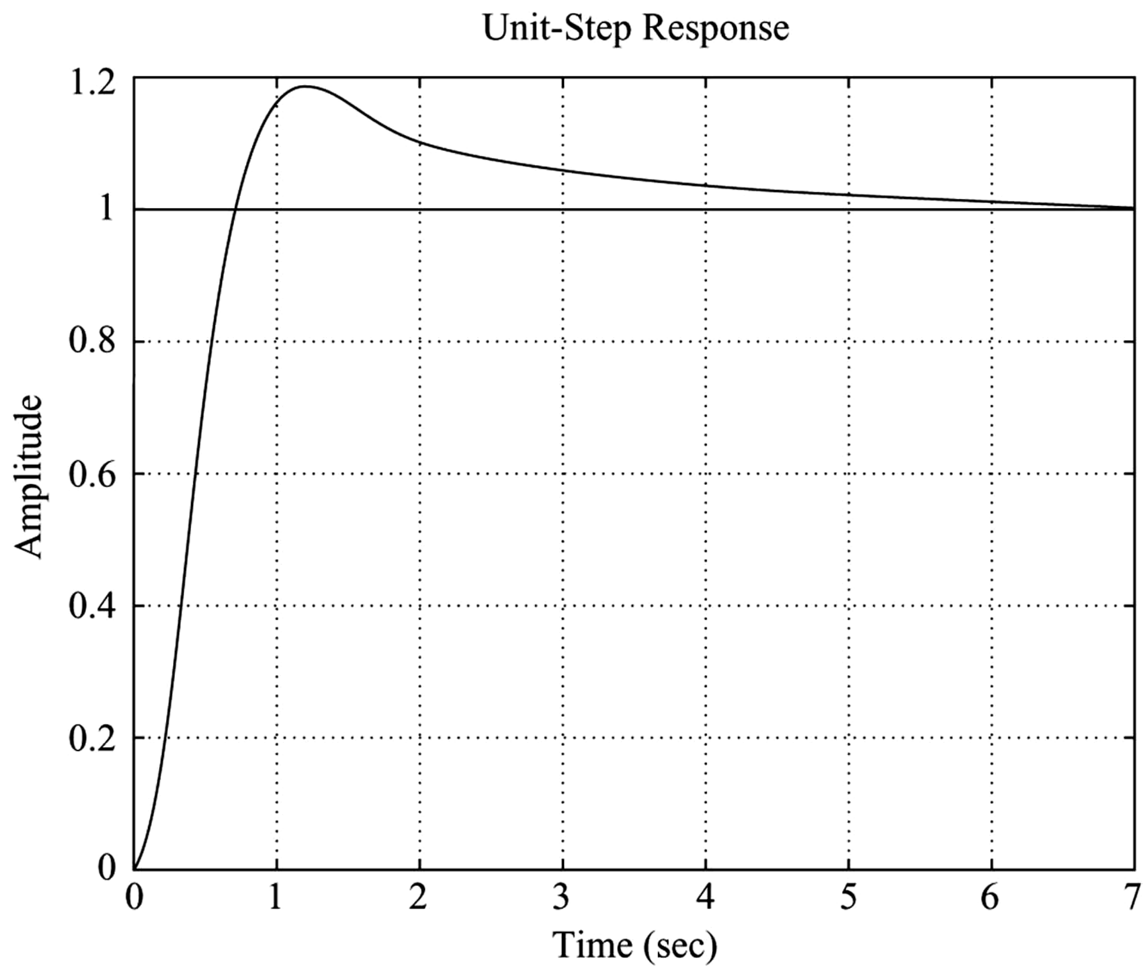


Figure 10-9 **Unit-step response of the system shown in Figure 8–6 with PID controller having parameters $K_p = 18$, $T_i = 3.077$, and $T_d = 0.7692$.**

If the proportional gain K_p is increased to 39.42, without changing the location of the double zero ($s = -0.65$), that is, using the PID controller

$$G_c(s) = 39.42\left(1 + \frac{1}{3.077s} + 0.7692s\right) = 30.322\frac{(s + 0.65)^2}{s} \tag{10–2}$$

then the speed of response is increased, but the maximum overshoot is also increased to approximately 28%, as shown in Figure 10–10.

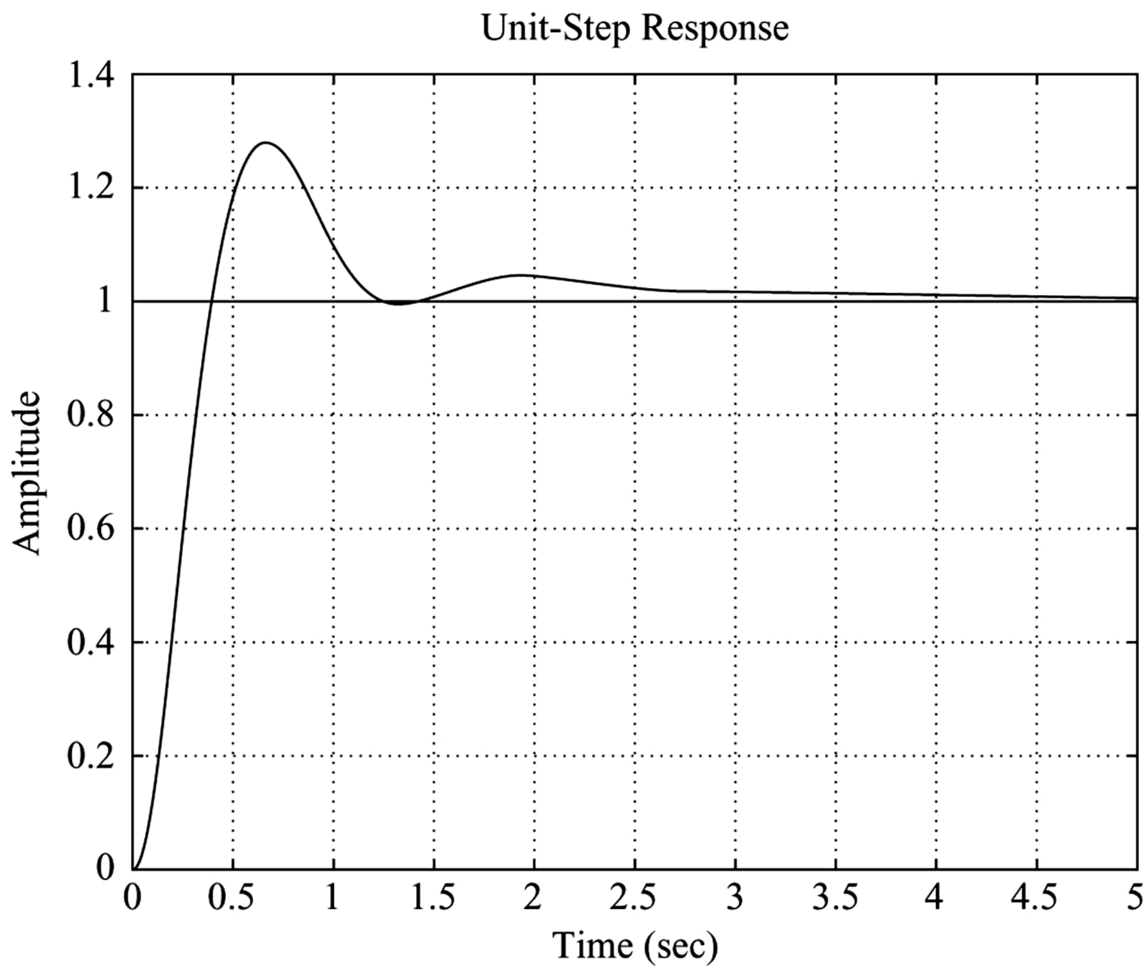
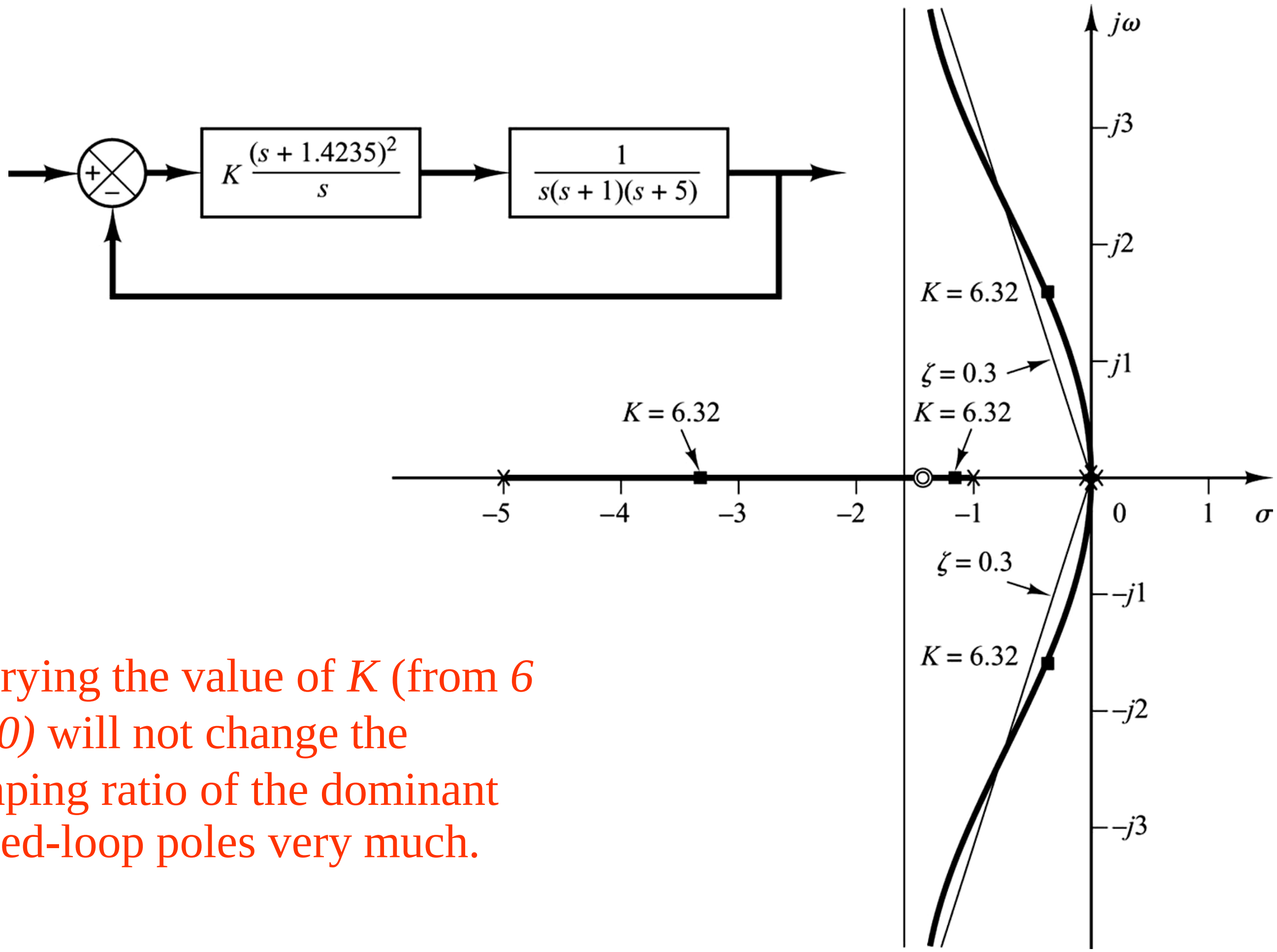


Figure 10–10
Unit-step response of the system shown in Figure 10–6 with PID controller having parameters $K_p = 39.42$, $T_i = 3.077$, and $T_d = 0.7692$.

Since the maximum overshoot in this case is fairly close to 25% and the response is faster than the system with $G_c(s)$ given by Equation (10–1), we may consider $G_c(s)$ as given by Equation (10–2) as acceptable. Then the tuned values of K_p, T_i, and T_d become

$$K_p = 39.42, \qquad T_i = 3.077, \qquad T_d = 0.7692$$

It is interesting to observe that these values respectively are approximately twice the values suggested by the second method of the Ziegler–Nichols tuning rule. The important thing to note here is that the Ziegler–Nichols tuning rule has provided a starting point for fine tuning.



- Varying the value of *K* (from *6* to *30)* will not change the damping ratio of the dominant closed-loop poles very much.

Figure 10-11 **Root-locus diagram of system when PID controller has double zero at *s* = –1.4235.**

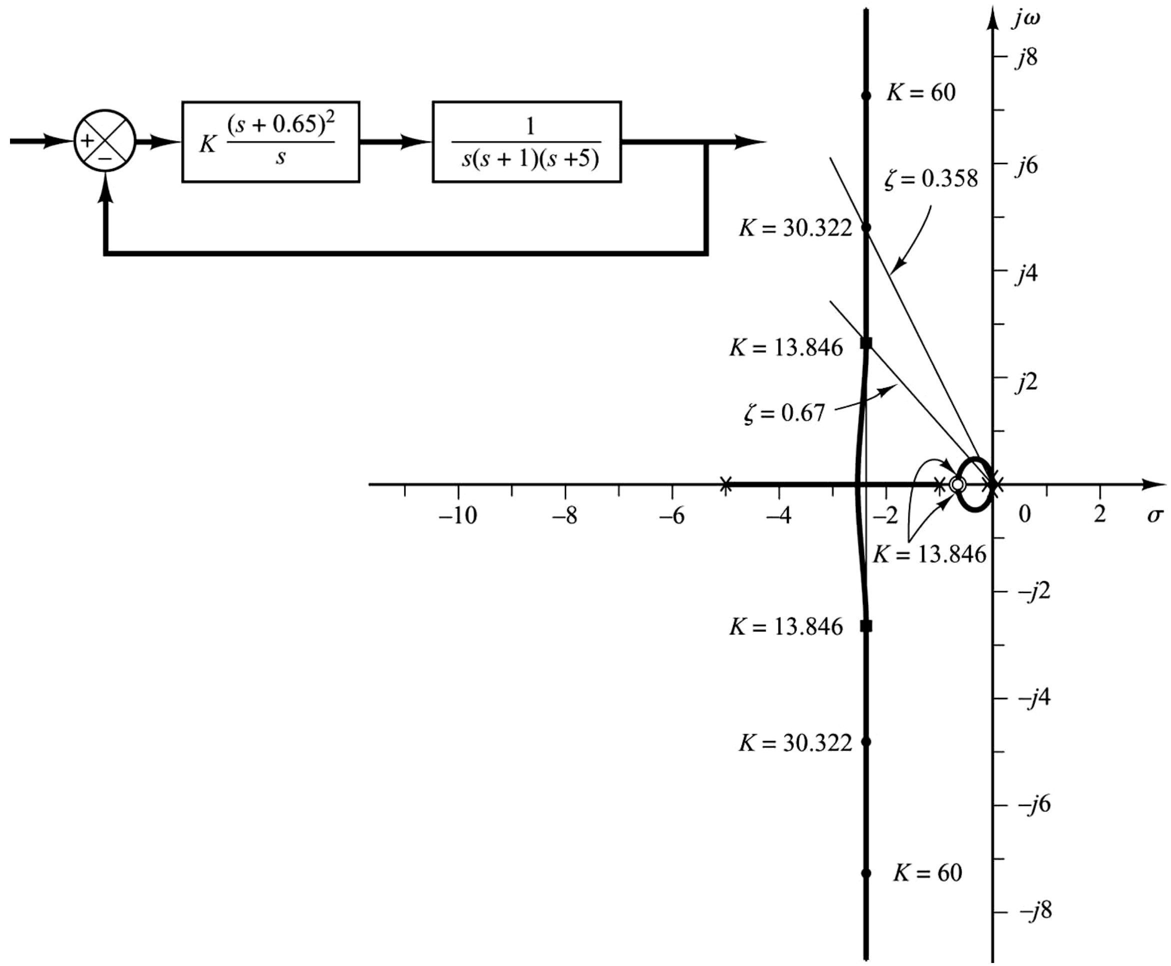


Figure 10-12 **Root-locus diagram of system when PID controller has double zero at $s = -0.65$. $K = 13.846$ corresponds to $G_c(s)$ given by Equation (10–1) and $K = 30.322$ corresponds to $G_c(s)$ given by Equation (10–2).**

10–3 COMPUTATIONAL APPROACH TO OBTAIN OPTIMAL SETS OF PARAMETER VALUES

EXAMPLE 10–2

Consider the PID-controlled system shown in Figure 10–13. The PID controller is given by

$$G_c(s) = K\frac{(s + a)^2}{s}$$

It is desired to find a combination of K and a such that the closed-loop system is underdamped and the maximum overshoot in the unit-step response is less than 10%, but more than 5%, to avoid an overdamped or a close-to-overdamped response. (Other conditions can be included, such as that the settling time be less than a specified value and the rise time be less than a certain specified value.)

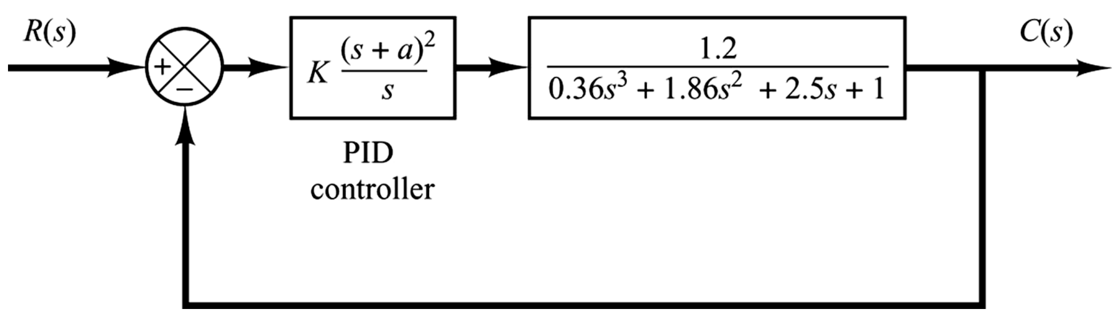


Figure 10-13 **PID-controlled system.**

$$2 \leq K \leq 5 \qquad \text{and} \qquad 0.5 \leq a \leq 1.5$$

To avoid an overly large amount of computation in this problem, let us choose the step size to be reasonable—say, 0.2 for both K and a.

MATLAB Program 10–2

```
t = 0:0.01:8;
for K = 5:-0.2:2; % Starts the outer loop to vary the K values
  for a = 1.5:-0.2:0.5; % Starts the inner loop to vary the a values
    num = [0  0  1.2*K  2.4*K*a  1.2*K*a^2];
    den = [0.36  1.86  2.5+1.2*K  1+2.4*K*a  1.2*K*a^2];
      y = step(num,den,t);
      m = max(y);
      if m < 1.1 & m > 1.05
      break; % Breaks the inner loop
    end
    end
  if m < 1.1 & m > 1.05
  break; % Breaks the outer loop
 end
 end
```

```
plot(t,y)
grid
title('Unit-Step Response')
xlabel('t Sec')
ylabel('Output')
KK = num2str(K); % String value of K to be printed on plot
aa = num2str(a); % String value of a to be printed on plot
text(4.25,0.54,'K = '), text(4.75,0.54,KK)
text(4.25,0.46,'a = '), text(4.75,0.46,aa)
sol = [K;a;m]

Sol =

4.2000
0.7000
1.0962
```

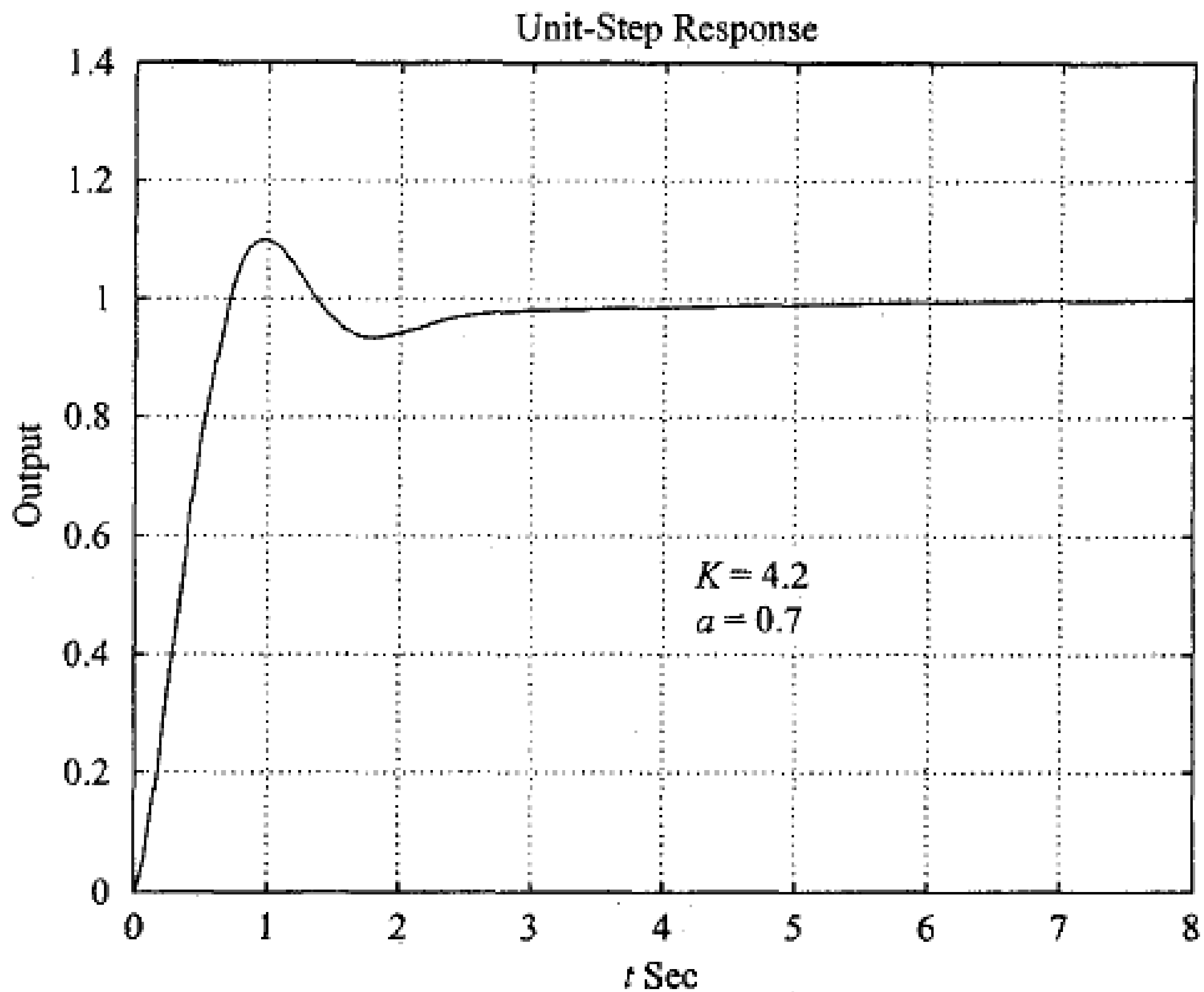


Figure 10–14
Unit-step response curve obtained by use of MATLAB Program 10–2.

MATLAB Program 10–3 is basically the same as MATLAB Program 10–2.

MATLAB Program 10–3

```
t = 0:0.01:8;
for K = 5:-0.2:2; % Starts the outer loop to vary the K values
  for a = 1.5:-0.2:0.5; % Starts the inner loop to vary the a values
        num1 = K*[1  2*a  a^2];
        den1 = [0  1  0];
        tf1 = tf(num1, den1);
        num2 = [0  0  0  1.2];
        den2 = [0.36  1.86  2.5  1];
        tf2 = tf(num2,den2);
        tf3 = tf1*tf2;
        sys = feedback(tf3,1);
        y = step(sys,t);
        m = max(y);
        if m < 1.1 & m > 1.05;
        plot(t,y);
        grid;
        title('Unit-Step Response')
        xlabel('t Sec')
        ylabel('Output')
        sol = [K;a;m]
        break; % Breaks the inner loop
     end
     end
  if m < 1.1 & m > 1.05;
  break; % Breaks the outer loop
end
end
```

```
sol =
      4.2000
      0.7000
      1.0962
text(6.2, 0.35,'K = '), text(6.65, 0.35,num2str(K))
text(6.2, 0.25,'a = '), text(6.65, 0.25,num2str(a))
```

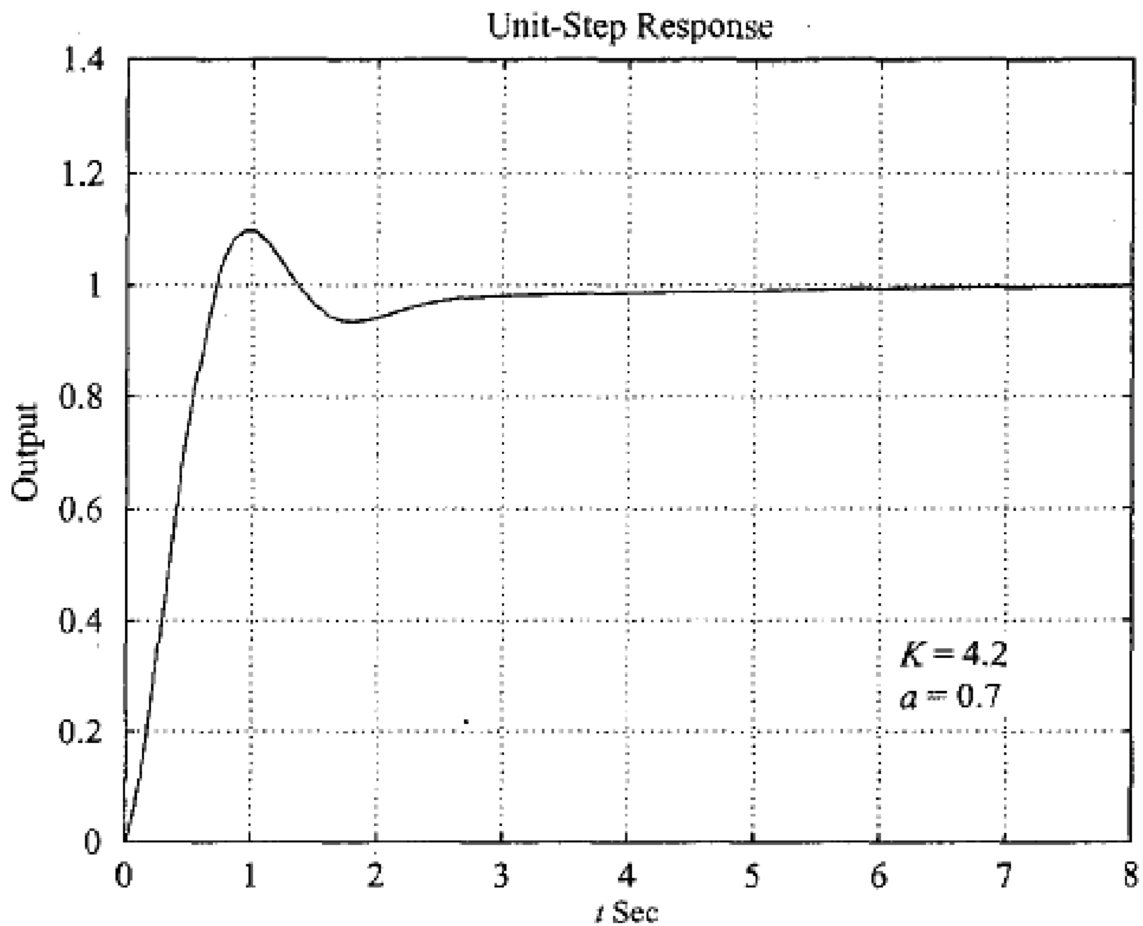


Figure 10–15 Unit-step response curve obtained by use of MATLAB Program 10–3.

EXAMPLE 10–3

In Example 10–2 we wrote MATLAB programs to find the first set of parameters to satisfy the given specifications. There may be more than one set of parameters that satisfy the specifications. In this example, we shall obtain all sets of parameters that satisfy the given specifications.

Consider the same system as in Example 10–2, except that the problem here is to find all sets of K and a that will satisfy the given specification that the maximum overshoot in the unit-step response be less than 10%. (This means that overdamped systems are included.) Assume the search region to be

$$2 \leq K \leq 3, \qquad 0.5 \leq a \leq 1.5$$

In the actual design process, the step size should be sufficiently small. In this example problem, however, we choose a fairly large step size to make the total number of search points reasonable. Thus, we choose the step size for both K and a to be 0.2.

To solve this problem it is possible to write many different MATLAB programs. We present here one such program, MATLAB Program 10–4.

MATLAB Program 10–4

```
%'K' and 'a' values to test

K = [2.0  2.2  2.4  2.6  2.8  3.0];
a = [0.5  0.7  0.9  1.1  1.3  1.5];

% Evaluate closed-loop unit-step response at each 'K' and 'a' combination
% that will yield the maximum overshoot less than 10%

t = 0:0.01:5;
g = tf([0  0  0  1.2],[0.36  1.86  2.5  1]);
k = 0;
for i = 1:6;
   for j= 1:6;
      gc = tf(K(i)*[1  2*a(j)  a(j)^2], [0  1  0]);  % controller
         G = gc*g/(1 + gc*g);  % closed-loop transfer function
         y = step(G,t);
```

```
         m = max(y);
         if m < 1.10
         k = k+1;
         solution(k,:) = [K(i)  a(j)  m];
         end
      end
   end
solution  % Print solution table
```

```
solution =

    2.0000  0.5000  0.9002
    2.0000  0.7000  0.9807
    2.0000  0.9000  1.0614
    2.2000  0.5000  0.9114
    2.2000  0.7000  0.9837
    2.2000  0.9000  1.0772
    2.4000  0.5000  0.9207
    2.4000  0.7000  0.9859
    2.4000  0.9000  1.0923
    2.6000  0.5000  0.9283
    2.6000  0.7000  0.9877
    2.8000  0.5000  0.9348
    2.8000  0.7000  1.0024
    3.0000  0.5000  0.9402
    3.0000  0.7000  1.0177
sortsolution = sortrows(solution,3)  % Print solution table sorted by
                                     % column 3
```

```
sortsolution =

    2.0000  0.5000  0.9002
    2.2000  0.5000  0.9114
    2.4000  0.5000  0.9207
    2.6000  0.5000  0.9283
    2.8000  0.5000  0.9348
    3.0000  0.5000  0.9402
    2.0000  0.7000  0.9807
    2.2000  0.7000  0.9837
    2.4000  0.7000  0.9859
    2.6000  0.7000  0.9877
    2.8000  0.7000  1.0024
    3.0000  0.7000  1.0177
    2.0000  0.9000  1.0614
    2.2000  0.9000  1.0772
    2.4000  0.9000  1.0923
```

```
% Plot the response with the largest overshoot that is less than 10%

K = sortsolution(k,1)

K =

   2.4000

a = sortsolution(k,2)

a =

   0.9000

gc = tf(K*[1  2*a  a^2], [0  1  0]);
G = gc*g/(1 + gc*g);
step(G,t)
grid  % See Figure 10-16
```

```
% If you wish to plot the response with the smallest overshoot that is
% greater than 0%, then enter the following values of 'K' and 'a'

K = sortsolution(11,1)

K =

   2.8000

a = sortsolution(11,2)

a =

   0.7000

gc = tf(K*[1  2*a  a^2], [0  1  0]);
G = gc*g/(1 + gc*g);
step(G,t)
grid  % See Figure 10-17
```

Note that for a specification that the maximum overshoot be between 10% and 5%, there would be three sets of solutions:

$$K = 2.0000, \quad a = 0.9000, \quad m = 1.0614$$

$$K = 2.2000, \quad a = 0.9000, \quad m = 1.0772$$

$$K = 2.4000, \quad a = 0.9000, \quad m = 1.0923$$

Unit-step response curves for these three cases are shown in Figure 10–18. Notice that the system with a larger gain K has a smaller rise time and larger maximum overshoot. Which one of these three systems is best depends on the system's objective.

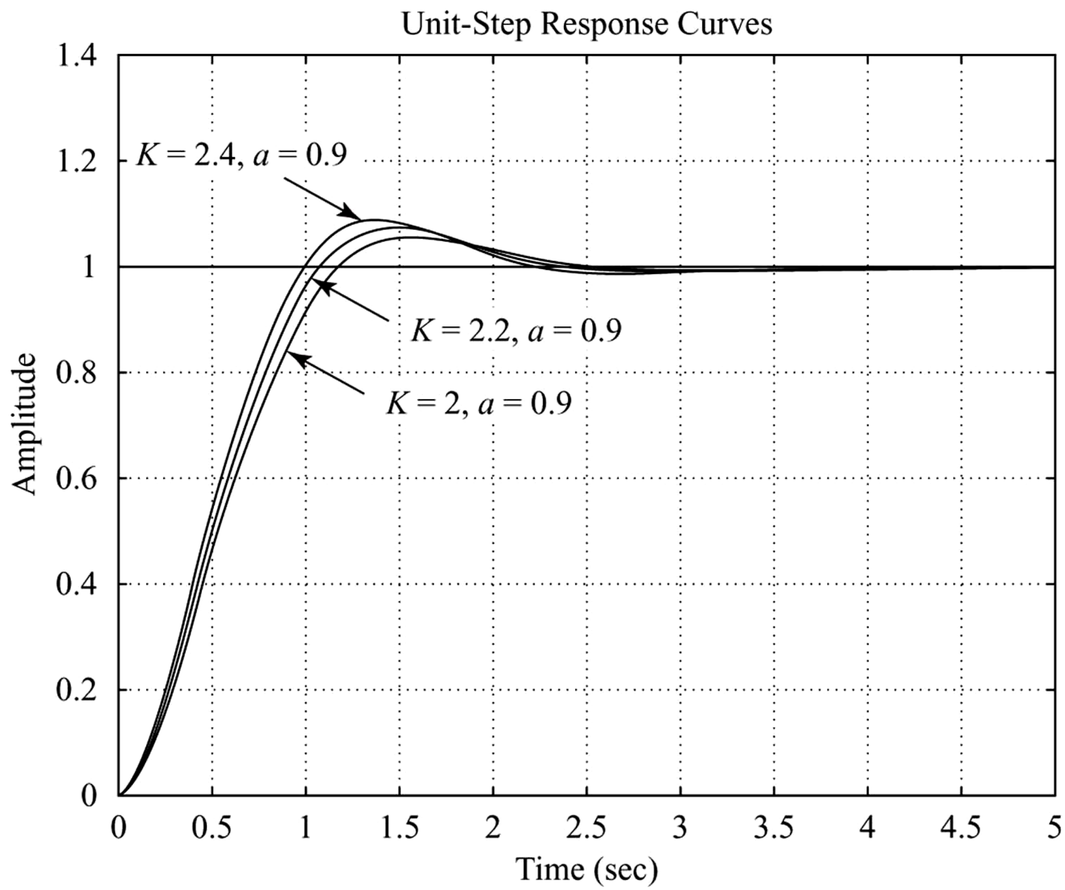
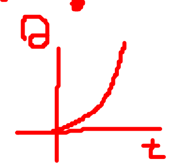


Figure 10–18 Unit-step response curves of system with $K = 2$, $a = 0.9$; $K = 2.2$, $a = 0.9$; and $K = 2.4$, $a = 0.9$.

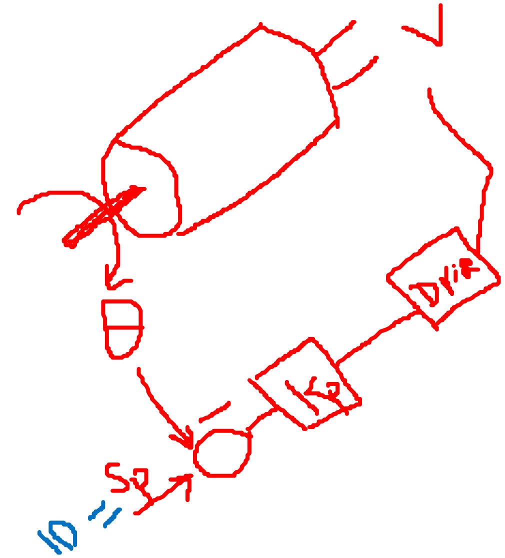
How does the PID works?

open loop → unstable

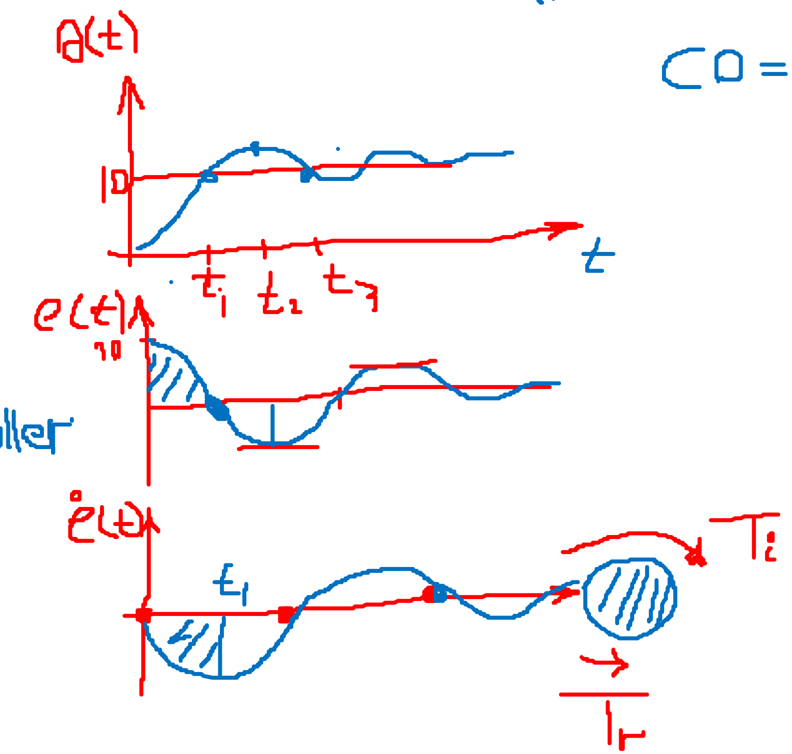


Closed loop → stable

Performance ←



Prob: Controller didn't stop the Motor at SP



$CO = K_p e$

Causes:

✓ 0 – t_1 initial e seen by Controller is Large

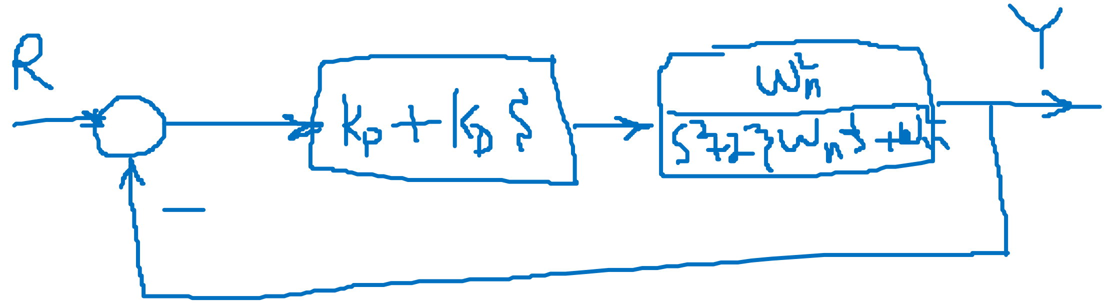
✓ t_1 – t_2 Small retarding Torque

PD controller

$CO = K_p e + K_D \dot{e}$

Mathematically

$$CO = (K_P + K_D s)\, E(s)$$



The CL

$$\frac{Y}{R} = \frac{\omega_n^2 [K_P + K_D s]}{s^2 + 2\zeta\omega_n s + \omega_n^2 + \omega_n^2 K_P + \omega_n^2 K_D s}$$

$$D(s) = s^2 + \underbrace{(2\zeta\omega_n + \omega_n^2 K_D)}\, s + (1 + K_P)\,\omega_n^2$$

$K_D \uparrow$ ⟶ More damping ⟶ less overshoot

PI controller

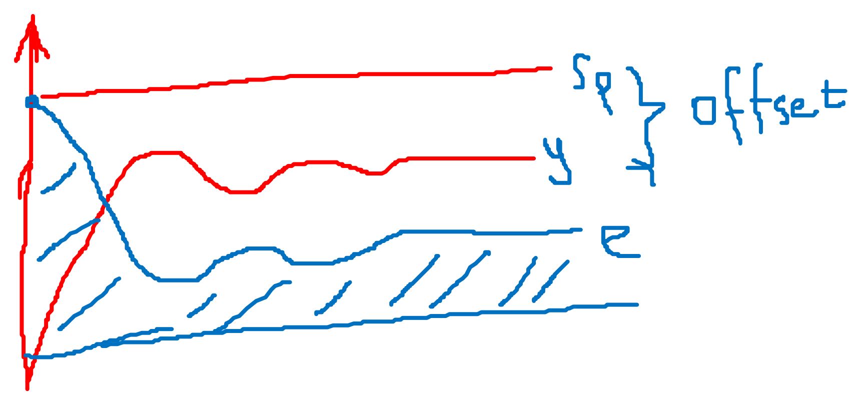
P – control

$$CO = K_p e + \text{bias}$$

or

$$CO = K_p e$$

$y_1 = y + \text{offset}$ Not practical



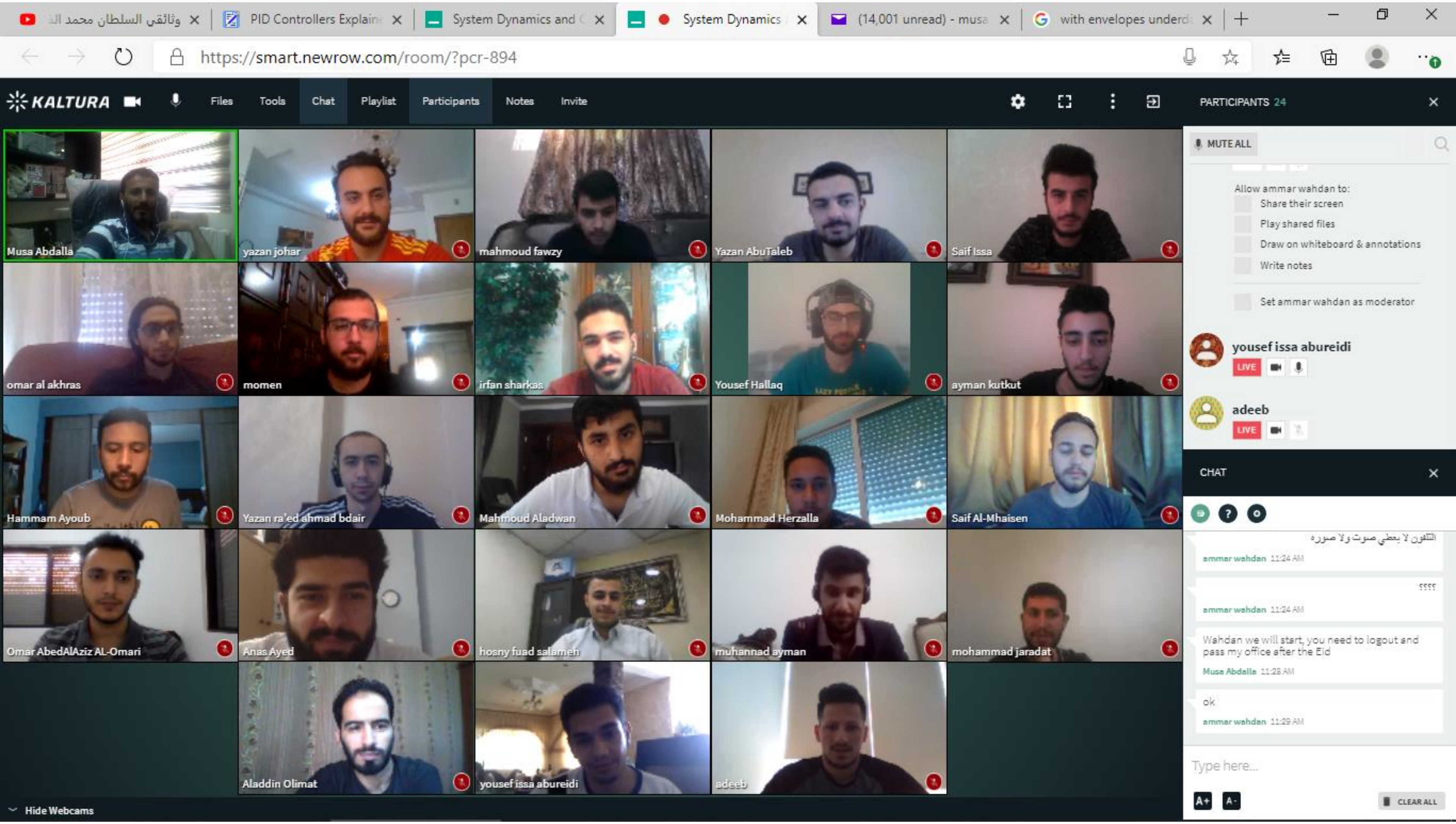
PI – Controller

$$CO = K_p e + K_I \int_0^t e(\tau)\, d\tau \quad \text{idea}$$

$$CO = \left[K_P + \frac{K_I}{s}\right] E(s) = \frac{K_I + K_P s}{s} E(s)$$

System Type ↑ ⟶ e_{ss} ↓

Mid Exam


https://smart.newrow.com/room/?pcr-894
KALTURA
Files
Tools
Chat
Playlist
Participants
Notes
Invite
PARTICIPANTS 24
MUTE ALL
Allow ammar wahdan to:
Share their screen
Play shared files
Draw on whiteboard & annotations
Write notes
Set ammar wahdan as moderator
yousef issa abureidi
adeeb
CHAT
Wahdan we will start, you need to logout and pass my office after the Eid
Musa Abdalla
ok
ammar wahdan
Musa Abdalla
yazan johar
mahmoud fawzy
Yazan AbuTaleb
Saif Issa
omar al akhras
momen
irfan sharkas
Yousef Hallaq
ayman kutkut
Hammam Ayoub
Yazan ra'ed ahmad bdair
Mahmoud Aladwan
Mohammad Herzalla
Saif Al-Mhaisen
Omar AbedAlAziz AL-Omari
Anas Ayed
hosny fuad salameh
muhannad ayman
mohammad jaradat
Aladdin Olimat
yousef issa abureidi
adeeb
Hide Webcams