

For a unity feedback system, with  $G(s) = \frac{1}{s(s+1)(s+5)}$ , we want to design a PID controller of this form:

$$G_{PID}(s) = K \frac{(s+a)^2}{s},$$

where:

- $2 \leq K \leq 50$  and  $0.05 \leq a \leq 2$ ,
- the desired maximum overshoot is between 2 and 10 percent,
- the settling time is less than 3 seconds,
- Time simulation is for 5 seconds,
- You can iterate  $K$  and  $a$  with increments of 1 and 0.1 respectively.

In this problem, you will have to write a MATLAB program that finds the optimal value for  $K$  and  $a$  given the above ranges for  $K$  and  $a$  and the design specifications. First, you will have to obtain the CLTF in terms of  $K$  and  $a$  (in standard form). After that, you'll have to write MATLAB code with two for loops (one that iterates  $K$  and other that iterates  $a$ ) and if conditions that compute settling time and maximum overshoot. The output of your program should be the optimal  $K$  and  $a$ , as well as a plot that shows the step response.

Here's a pseudo-code that you can follow:

```
time = 0:0.01:5; % defining time
for K = 2:1:50 % for loop for K
for a = 0.05:0.1:2; $ for loop for a
num = [.....]; % Numerator for the CLTF in terms of K, a
den = [.....]; % Denominator for the CLTF in terms of K, a
y_step = step(num,den,time); % generating the step response for
% the specific K and a
.... % Your code here
.... % Your code here
.... % Your code here
.... % Here, in these few lines of code (above), you have to write
% statements which includes testing for the design specs that
% are required from you. You can use the max MATLAB function, if-else statements,
% breaks, and while to do that.
end % ending the first for loop
end % ending the second for loop
plot(time,y_step)
ylabel('Output')
solution = .... % Extract your solution from the loops
```

---

**Solutions (from Ogata):**

The closed-loop transfer function  $C(s)/R(s)$  is given by

$$\frac{C(s)}{R(s)} = \frac{Ks^2 + 2Kas + Ka^2}{s^4 + 6s^3 + (5 + K)s^2 + 2Kas + Ka^2}$$

A possible MATLAB program that will produce the first set of variables  $K$  and  $a$  that will satisfy the given specifications is given in MATLAB Program 8–15. In this program we use two ‘for’ loops. The specification for the settling time is interpreted by the following four lines:

```
s = 501; while y(s) > 0.98 and y(s) < 1.02;
s = s - 1; end;
ts = (s - 1) * 0.01
ts < 3.0
```

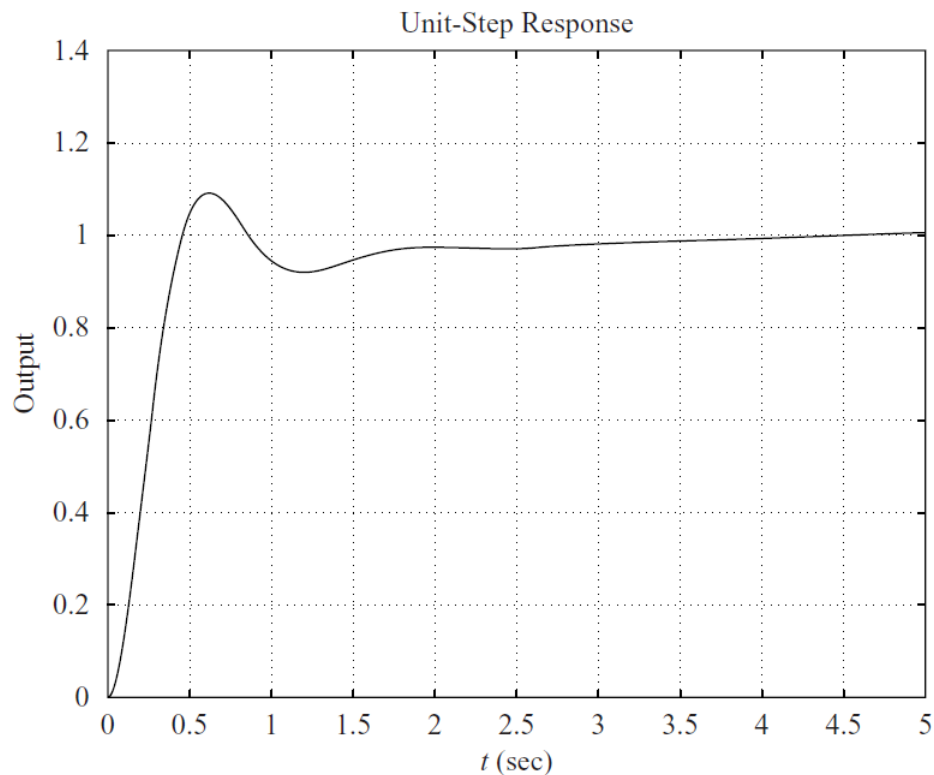
Note that for  $t = 0:0.01:5$ , we have 501 computing time points.  $s = 501$  corresponds to the last computing time point.

The solution obtained by this program is

$$K = 32, \quad a = 0.2$$

with the maximum overshoot equal to 9.69% and the settling time equal to 2.64 sec. The resulting unit-step response curve is shown in Figure 8–64.

MATLAB Program 8–15
<pre>t = 0:0.01:5; for K = 50:-1:2;   for a = 2:-0.05:0.05;     num = [K 2*K*a K*a^2];     den = [1 6 5+K 2*K*a K*a^2];     y = step(num,den,t);     m = max(y);     s = 501; while y(s) &gt; 0.98 &amp; y(s) &lt; 1.02;     s = s-1; end;     ts = (s-1)*0.01;     if m &lt; 1.10 &amp; m &gt; 1.02 &amp; ts &lt; 3.0       break;     end   end   if m &lt; 1.10 &amp; m &gt; 1.02 &amp; ts &lt; 3.0     break   end end plot(t,y) grid title('Unit-Step Response') xlabel('t sec') ylabel('Output') solution = [K;a;m;ts]  solution =  32.0000 0.2000 1.0969 2.6400</pre>



Next, we shall consider the case where we want to find all sets of variables that will satisfy the given specifications. A possible MATLAB program for this purpose is given in MATLAB Program 8-16. Note that in the table shown in the program, the last row of the table (k,:) or the first row of the sorttable should be ignored. (These are the last  $K$  and  $a$  values for searching purposes.)

```

MATLAB Program 8-16
t = 0:0.01:5;
k = 0;
for i = 1:49;
    K(i) = 51-i*1;
    for j = 1:40;
        a(j) = 2.05-j*0.05;
        num = [K(i) 2*K(i)*a(j) K(i)*a(j)*a(j)];
        den = [1 6 5+K(i) 2*K(i)*a(j) K(i)*a(j)*a(j)];
        y = step(num,den,t);
        m = max(y);
        s = 501; while y(s) > 0.98 & y(s) < 1.02;
            s = s-1; end;
        ts = (s-1)*0.01;
        if m < 1.10 & m > 1.02 & ts < 3.0
            k = k+1;
            table(k,:) = [K(i) a(j) m ts];
        end
    end
end
table(k,:) = [K(i) a(j) m ts]
table =

```

*(continues on next page)*

```

32.0000 0.2000 1.0969 2.6400
31.0000 0.2000 1.0890 2.6900
30.0000 0.2000 1.0809 2.7300
29.0000 0.2500 1.0952 1.7800
29.0000 0.2000 1.0726 2.7800
28.0000 0.2000 1.0639 2.8300
27.0000 0.2000 1.0550 2.8900
2.0000 0.0500 0.3781 5.0000

```

```
sorttable = sortrows(table,3)
```

```
sorttable =
```

```

2.0000 0.0500 0.3781 5.0000
27.0000 0.2000 1.0550 2.8900
28.0000 0.2000 1.0639 2.8300
29.0000 0.2000 1.0726 2.7800
30.0000 0.2000 1.0809 2.7300
31.0000 0.2000 1.0890 2.6900
29.0000 0.2500 1.0952 1.7800
32.0000 0.2000 1.0969 2.6400

```

```
K = sorttable(7,1)
```

```
K =
```

```
29
```

```
a = sorttable(7,2)
```

```
a=
```

```
0.2500
```

```
num = [K 2*K*a K*a^2];
```

```
den = [1 6 5+K 2*K*a K*a^2];
```

```
y = step(num,den,t);
```

```
plot(t,y)
```

```
grid
```

```
hold
```

```
Current plot held
```

```
K = sorttable(2,1)
```

```
K=
```

```
27
```

```
a = sorttable(2,2)
```

```
a=
```

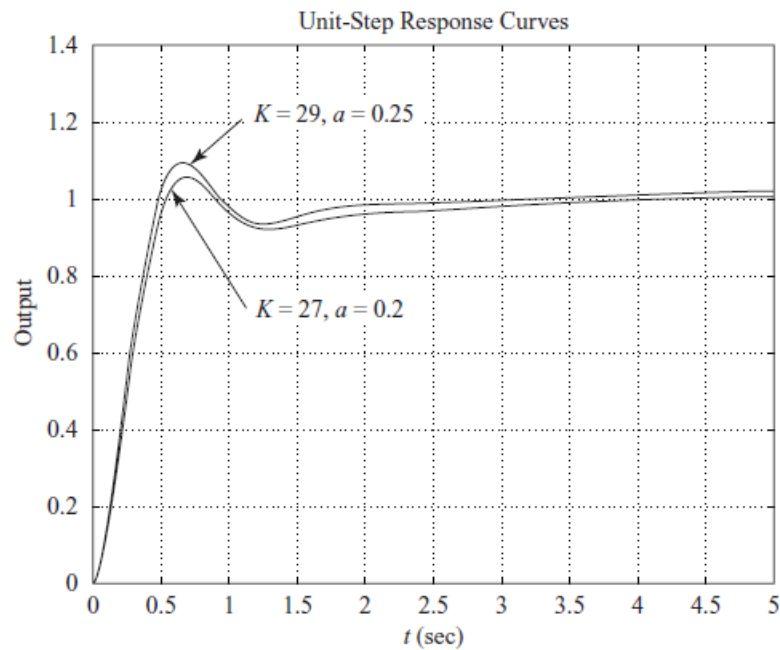
```
0.2000
```

*(continues on next page)*

```

num = [K 2*K*a K*a^2];
den = [1 6 5+K 2*K*a K*a^2];
y = step(num,den,t);
plot(t,y)
title('Unit-Step Response Curves')
xlabel('t (sec)')
ylabel('Output')
text(1.22,1.22,'K = 29, a = 0.25')
text(1.22,0.72,'K = 27, a = 0.2')

```



From the sortable, it seems that

$K = 29, a = 0.25$  (max overshoot = 9.52%, settling time = 1.78 sec)

and

$K = 27, a = 0.2$  (max overshoot = 5.5%, settling time = 2.89 sec)

are two of the best choices. The unit-step response curves for these two cases are shown in Figure 8–65. From these curves, we might conclude that the best choice depends on the system objective. If a small maximum overshoot is desired,  $K = 27, a = 0.2$  will be the best choice. If the shorter settling time is more important than a small maximum overshoot, then  $K = 29, a = 0.25$  will be the best choice.