# MATLAB MODULE 4

### Feedback System Simulation

## System Response

The transfer function manipulations give us a transfer function model $M(s)$ between command input $R(s)$ and output $Y(s)$; model $M_w(s)$ between disturbance input$W(s)$ and output $Y(s)$; a model between command input $R(s)$ and control $U(s)$, *etc*. It is now easy to use some of the control analysis commands available from the Control System Toolbox. **impulse(M)** and **step(M)** commands represent common control analysis operations that we meet in this book. Also frequently used in the book are frequency-response plots.

**Example M4.1**

Consider the block diagram in Fig. M4.1.

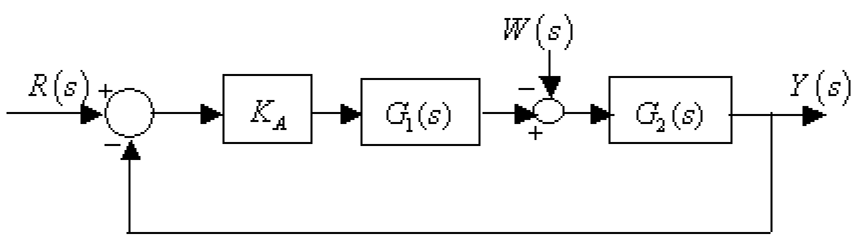


**Fig. M4.1**

For value of gain $K_A$ = 80, the following two MATLAB sessions evaluate the step responses with respect to reference input $R(s)$ and disturbance signal $W(s)$ for

$$G_1(s) = \frac{5000}{(s+1000)}; G_2(s) = \frac{1}{s(s+20)}$$

```
>> %Step response with respect to R(s)
>> s = tf('s');
>> KA = 80;
>> G1 = 5000/(s+1000);
>> G2 = 1/(s*(s+20));
>> M = feedback(series(KA*G1,G2),1)
>> step(M)
```

The MATLAB responds with

```
Transfer function:

              400000
--------------------------------------------------
s^3 + 1020 s^2 + 20000 s + 400000
```

and step response plot shown in Fig. M4.2. The grid has been introduced in the plot by right clicking on the plot and selecting **Grid** option.

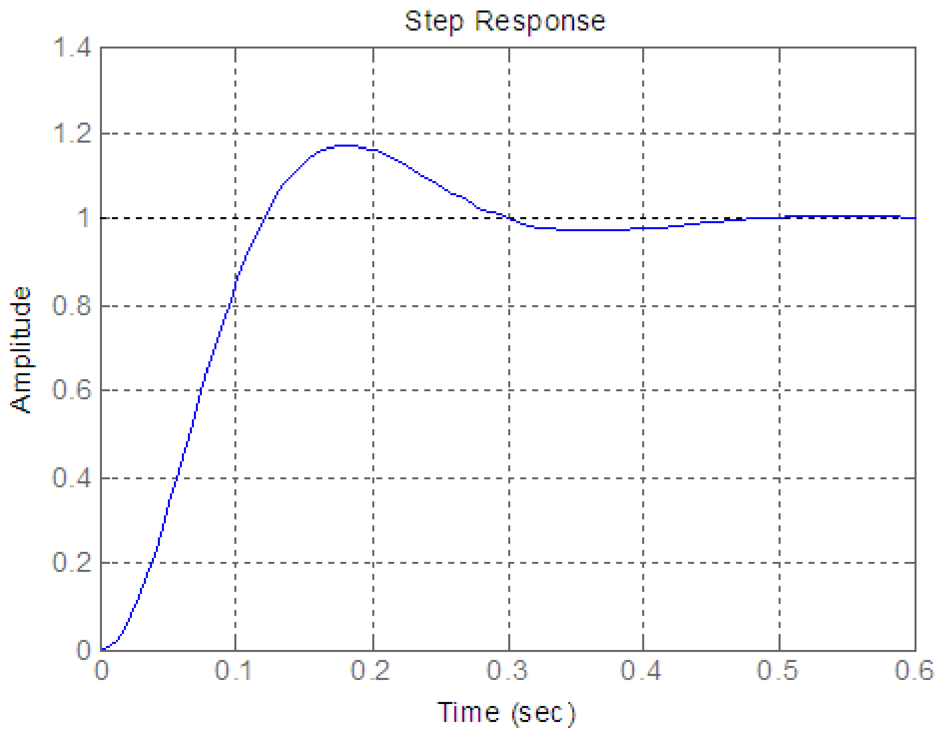


**Fig. M4.2**

```
>> %Step response with respect to W(s)

>> s = tf('s');

>> KA = 80;

>> G1 = 5000/(s+1000);

>> G2 = 1/(s*(s+20));

>> Mw = (-1) * feedback(G2, KA*G1)

>> step(Mw)
```

MATLAB responds with

```
Transfer function:

                -s - 1000
--------------------------------------------------
s^3 + 1020 s^2 + 20000 s + 400000
```

and step response plot shown in Fig. M4.3.

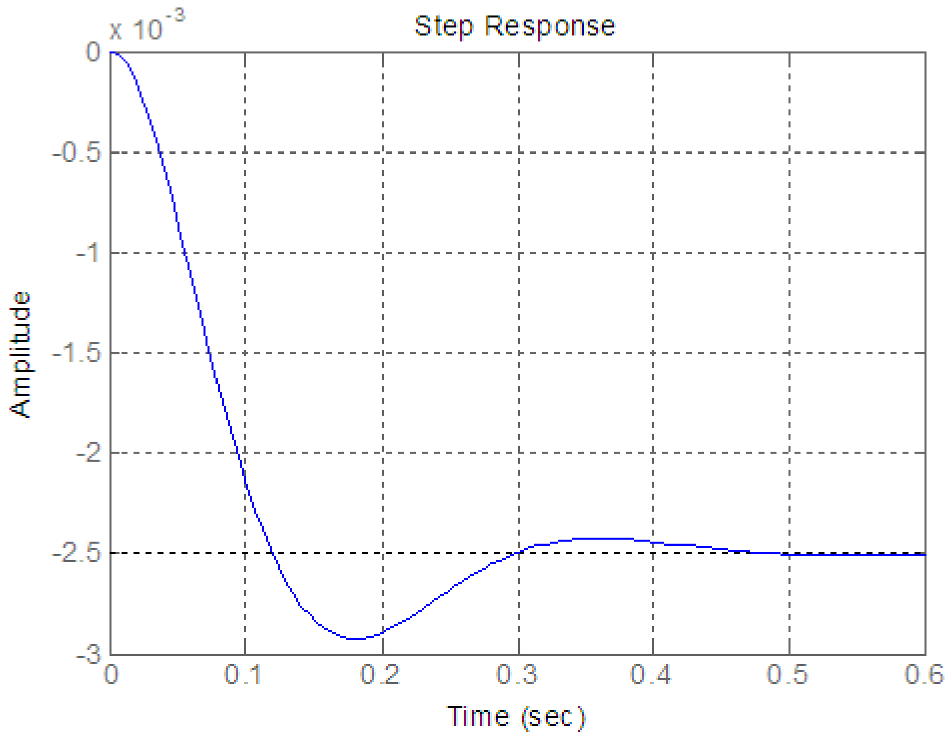


**Fig. M4.3**

**Example M4.2**

Let us examine the sensitivity of the feedback system represented by the transfer function

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

The system sensitivity to parameter $K$ is

$$S_K^M = \frac{\Delta M / M}{\Delta K / K} = \frac{\partial M}{\partial K}\left(\frac{K}{M}\right) = \frac{s(s+1)}{s^2 + s + K}$$

Figure M4.4 shows the magnitudes of $\left|S_K^M(j\omega)\right|$ and $\left|M(j\omega)\right|$ versus frequency $\omega$ for $K$ = 0.25; generated using the following MATLAB script. Text arrows have been introduced in the plot by following **Insert** from the main menu and selecting the option **Text Arrow**.

Note that the sensitivity is small for lower frequencies, while the transfer function primarily passes low frequencies.

```
w = 0.1:0.1:10;

M = abs(0.25./((j*w).^2+j*w+0.25));

SMK = abs((j*w .* (j*w + 1))./((j*w).^2 + j*w +0.25));

plot(w,M,'r',w,SMK,'b');

xlabel('Frequency (rad/sec)');

ylabel('Magnitude');
```

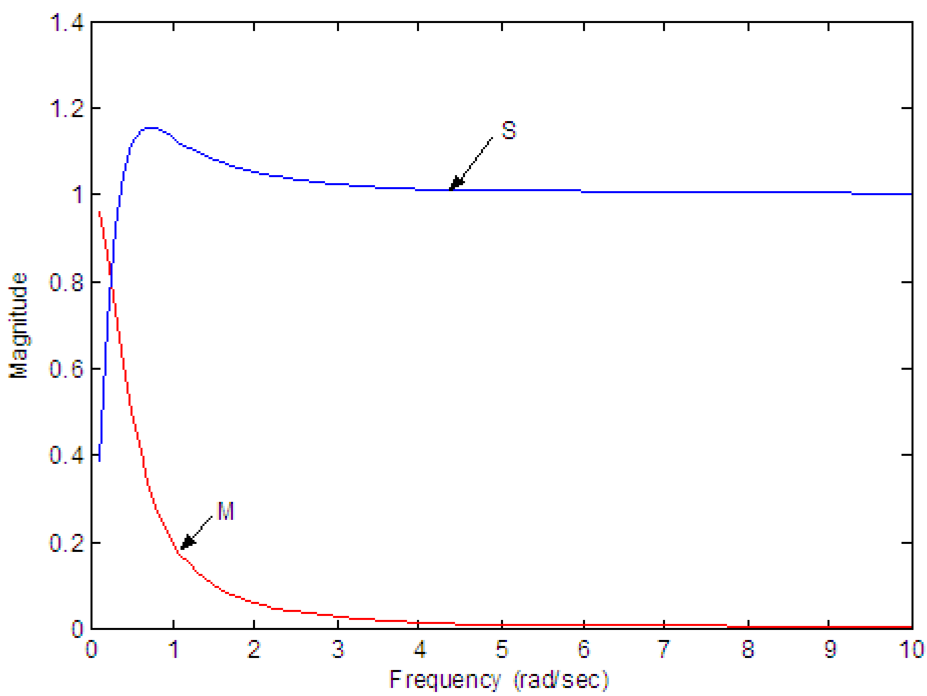


**Fig. M4.4**

Of course, the sensitivity $S$ only represents robustness for small changes in gain $K$. If $K$ changes from 1/4 within the range $K$ = 1/16 to $K$ = 1, the resulting range of step responses, generated by the following MATLAB script, is shown in Fig. M4.5. This system, with an expected wide range of $K$, may not be considered adequately robust. A robust system would be expected to yield essentially the same (within an agreed-upon variation) response to selected inputs.

```
s = tf('s');

S = (s*(s+1))/(s^2+s+0.25);

M1 = 0.0625/(s^2+s+0.0625);

M2 = 0.25/(s^2+s+0.25);

M3 = 1/(s^2+s+1);

step(M1);

hold on;

step(M2);

step(M3);
```

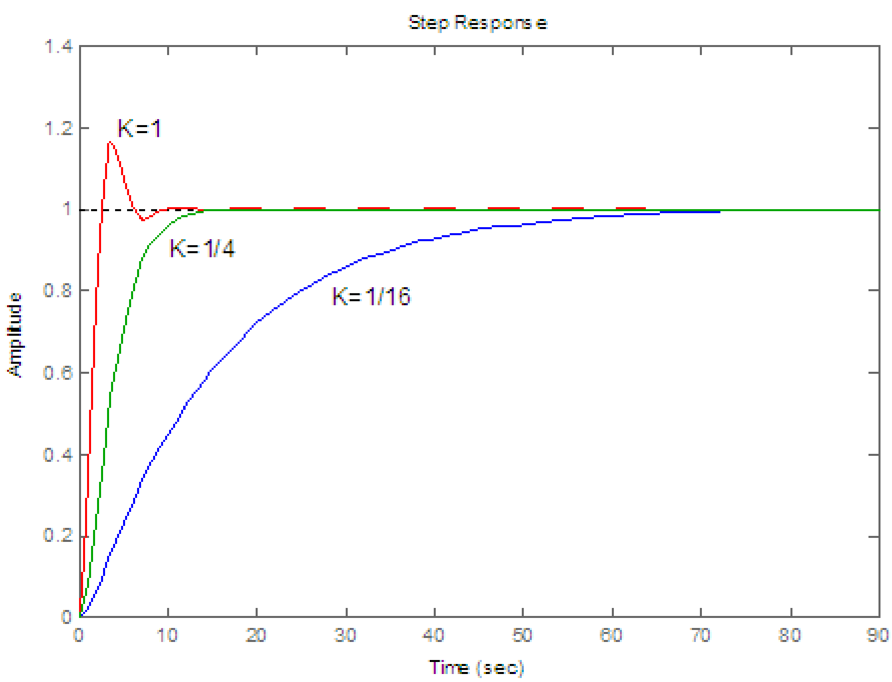


**Fig. M4.5**

# MATLAB MODULE 4

### Feedback System Simulation

## Simulink Simulation

Simulink simulation is an alternative to block diagram manipulation followed by time-response analysis. From the Simulink model of a control system, output *y* in response to command *r*, output *y* in response to disturbance *w*, control *u* in response to command *r*, and all other desired internal variables can be directly obtained.

**Example M4.3**

Control system design methods discussed in this course are based on the assumption of availability of linear time invariant (LTI) models for all the devices in the control loop.

Consider a speed control system. The actuator for the motor is a power amplifier. An amplifier gives a saturating behaviour if the error signal input to the amplifier exceeds *linear range* value.

MATLAB simulink is a powerful tool to simulate the effects of nonlinearities in a feedback loop. After carrying out a design using LTI models, we must test the design using simulation of the *actual* control system, which includes the nonlinearities of the devices in the feedback loop.

Figure M4.6 is the simulation diagram of a feedback control system: the amplifier gain is 100 and the transfer function of the motor is 0.2083/(*s* +1.71). We assume the amplifier of gain 100 saturates at +5 or -5volts. The result of the simulation is shown in Fig. M4.7.

The readers are encouraged to construct the simulink model using the procedure described in Module 3. All the parameter settings can be set/seen by double clicking on related blocks.

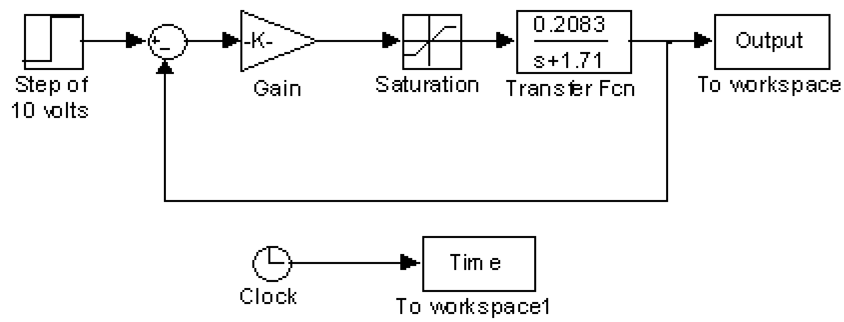


**Fig. M4.6** ( download )

Time and Output response data have been transferred to workspace using **To Workspace** block from **Sinks** main block menu. **Clock** block is available in**Sources** main menu. These variables are stored in the structure **Output** and **Time** in the workspace, along with the information regarding simulink model name. For example,

```
>> Output

Output =

time: [ ]

signals: [1x1 struct]

blockName: 'M4_3/To workspace Output'

>> Time

Time =

time: [ ]

signals: [1x1 struct]

blockName: 'M4_3/To workspace Time'
```

To access output and time values, one needs to access **Output.signals.values** and **Time.signals.values** . The step response plot has been generated by the following MATLAB script.

```
>> plot(Time.signals.values,Output.signals.values)
>> xlabel('Time (sec)');
>> ylabel('Output');
>> title('Step Response');
```

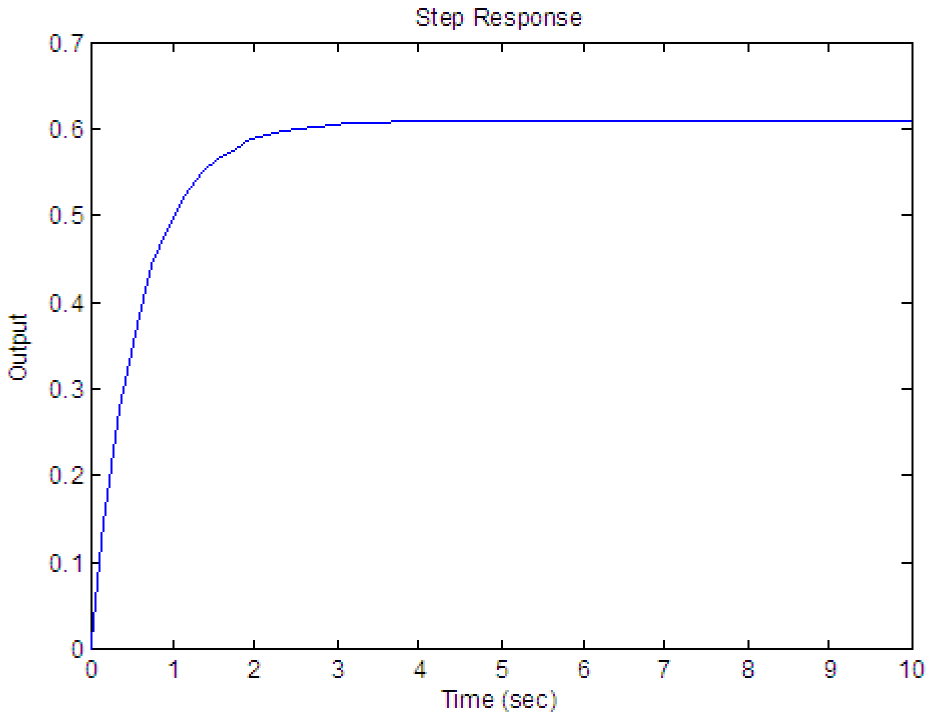


**Fig. M4.7**

## Example M4.4

In this example, we simulate a temperature control system with measurement noise added to the feedback signal. The process transfer function is

$$G(s) = \frac{3e^{-s\tau_D}}{3.1s+1}$$

The deadtime $\tau_D = 0.15$ minutes. The measurement noise parameters we have used are: mean of 0, variance of 0.5, initial seed of 61233, and sample time of 0. The simulink inputs a step of 30 to the system (Fig. M4.8). Deadtime block in this figure is **Transport Delay** block from **Continuous** library, and **Random Number** block is from **Sources** library.

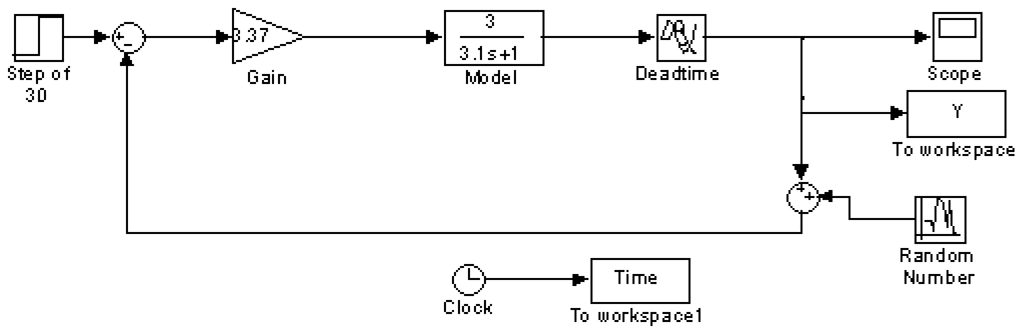


**Fig. M4.8** ( download )

The data has been transferred to the workspace using **To Workspace** block. The step response, generated using the following MATLAB script is shown in Fig. M4.9.

```
>> plot(Time.signals.values,Y.signals.values);

>> ylabel('Output (Y)');

>> xlabel('Time(min)');

>> title('Step Response');
```

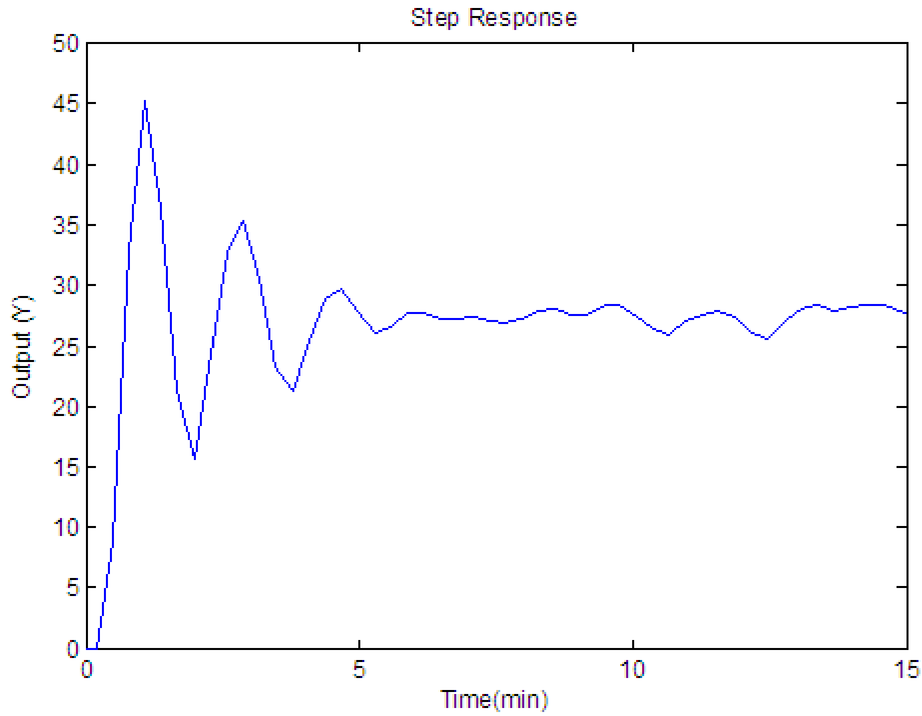


**Fig. M4.9**

The performance of the system with measurement noise removed, is shown in Fig. M4.10. To remove the effect of noise, simply disconnect the **Random number** block from the **Sum** block in the feedback path.

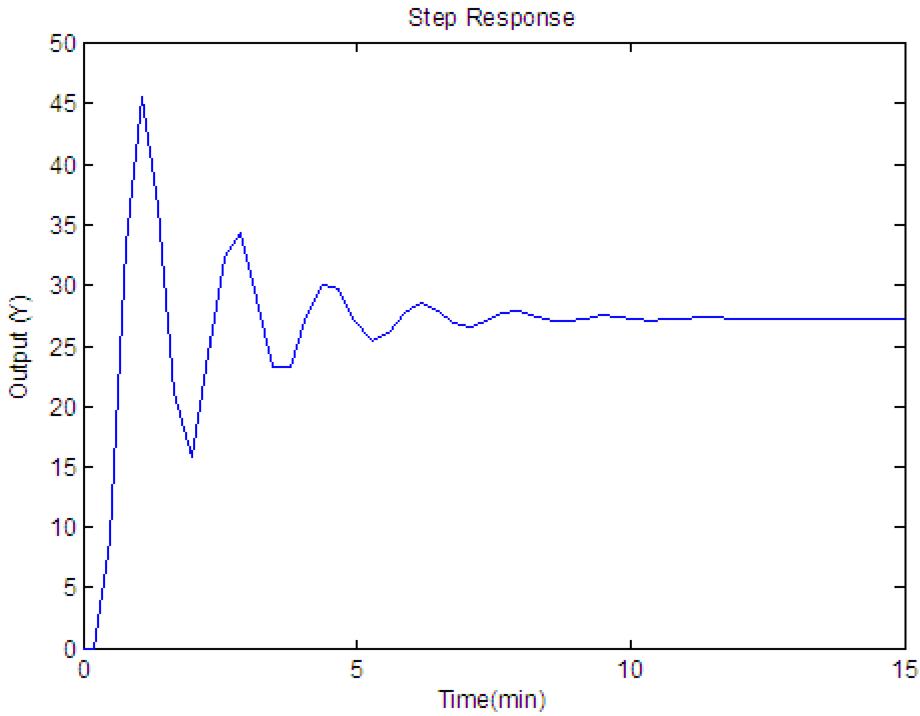


**Fig. M4.10**

5. Find the transfer function, $T(s) = C(s)/R(s)$, for the system shown in Figure P5.5. Use the following methods:

[redacted]

b. MATLAB. Use the following transfer functions: MATLAB ML

$G_1(s) = 1/(s+7)$, $G_2(s) = 1/(s^2+2s+3)$,
$G_3(s) = 1/(s+4)$, $G_4(s) = 1/s$,
$G_5(s) = 5/(s+7)$, $G_6(s) = 1/(s^2+5s+10)$,
$G_7(s) = 3/(s+2)$, $G_8(s) = 1/(s+6)$.

Hint: Use the append and connect commands in MATLAB's Control System Toolbox.

Ex M4.1

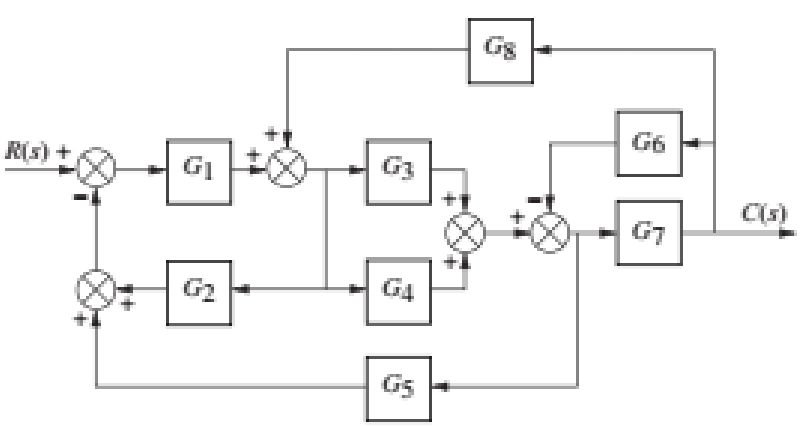


**FIGURE P5.5**