VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA
NEW ZEALAND

Demo 2: Feedback Control Systems

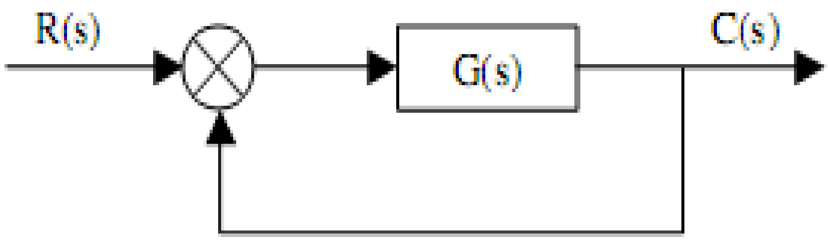
XMUT315 Control Systems Engineering

Topics

1. Feedback systems.
2. Second order system approximation.
3. Reduction of block diagrams.
4. Responses of the system (e.g. unit step response for given damping ratios).

Exercise 1 (Feedback System)

For the unity feedback system shown in the following diagram, $G(s)$ is given as:



$$G(s) = \frac{30(s^2 - 5s + 3)}{(s+1)(s+2)(s+4)(s+5)}$$

Determine the closed-loop step response using MATLAB. If this system is to be approximated as a second order system, analyse if this is correct?

Answer

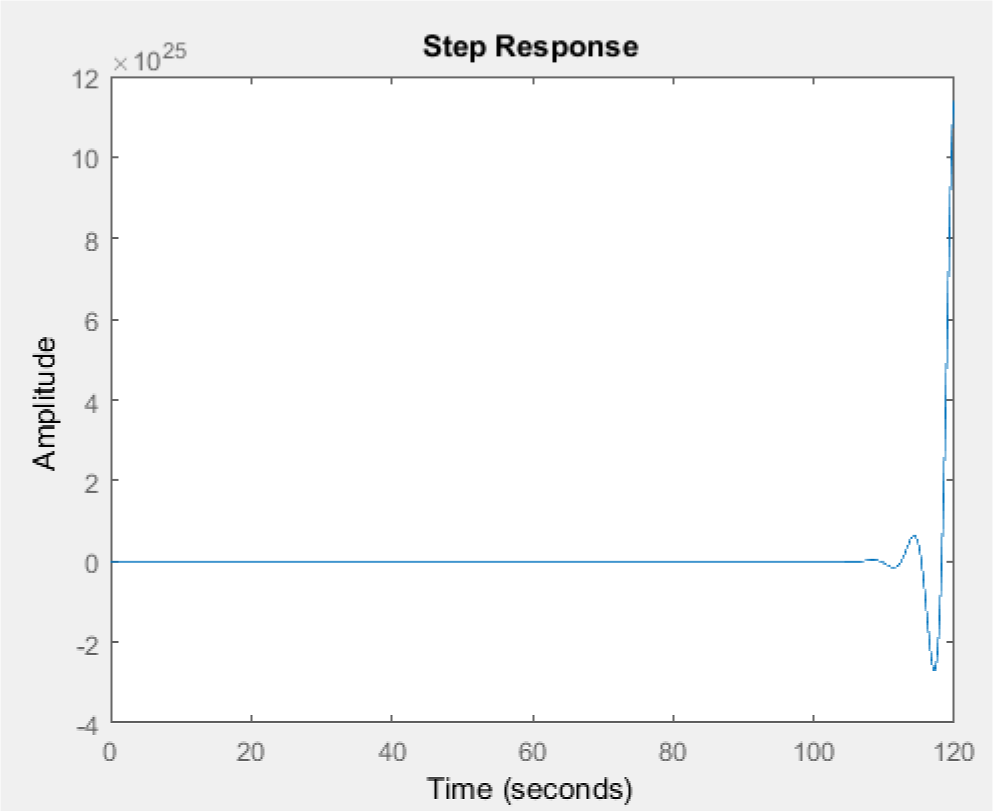
To model step response of the feedback system in MATLAB, you are required to do these things:

- Model of the transfer functions of the system:
    - Define and assign the numerator of the transfer function (e.g. `numg` vector)
    - Define and assign the denumerator of the transfer function (e.g. `deng` vector, treat the vector as a polynomial equation e.g. use `poly()` function).
- Assign the numerator and denumerator vectors into the transfer function (e.g. use `tf()` function).
- Assign the feedback system as (e.g. `feedback()` function).
- Assign the unit step as (e.g. use `step()` function).

MATLAB code:

```
% demo21.m

numg=30*[1 -5 3];
deng=poly([-1 -2 -4 -5]);
G=tf(numg,deng);
T=feedback(G,1);
step(T);
```

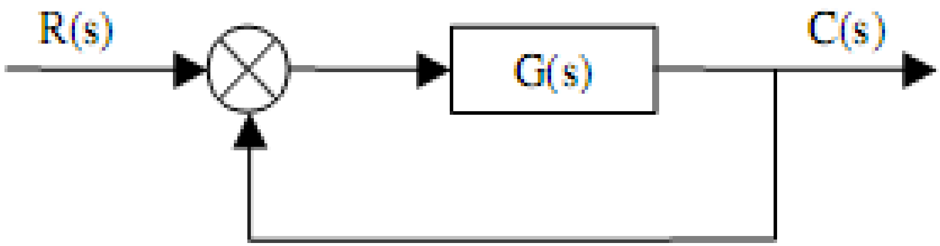
This will give the following simulation output in the MATLAB.



The simulation shows a big overshoot response and non-minimum phase behaviour. Hence the second-order approximation is not valid.

Exercise 2 (Second Order System Approximation)

Determine the accuracy of the second-order approximation using MATLAB to simulate the unity feedback system shown in the figure below where:



$$G(s) = \frac{15(s^2 + 3s + 7)}{(s^2 + 3s + 7)(s + 1)(s + 3)}$$

Answer

To model the step response of the feedback system in MATLAB, you are required to do these things:

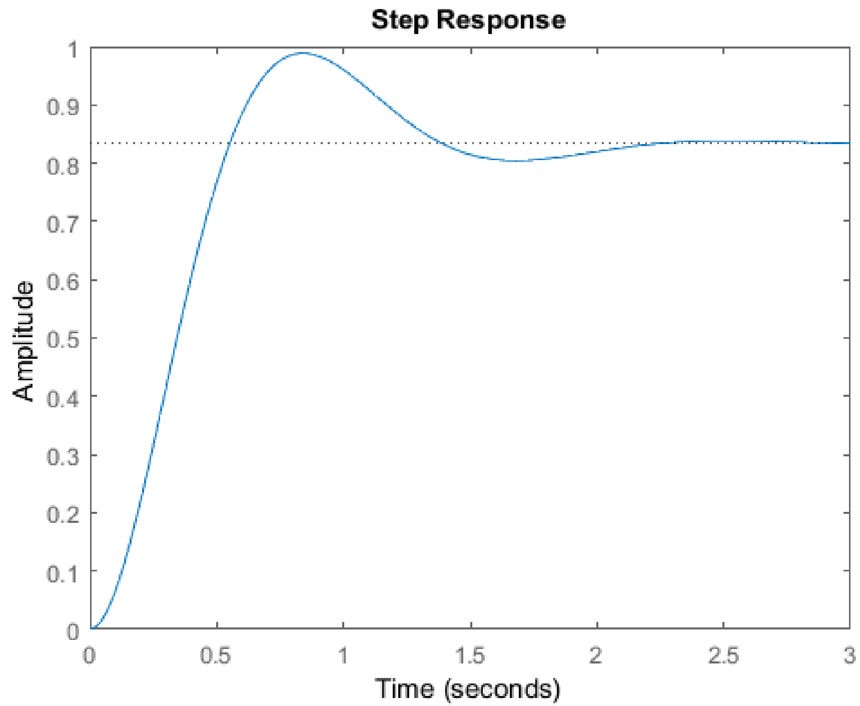
- Model of the transfer functions of the system:
    - Define and assign the numerator of the transfer function (e.g. `numg` vector)
    - Define and assign the denumerator of the transfer function (e.g. `deng` vector, treat the vector as a polynomial equation e.g. use `poly()` function).
- This time use convolution function in MATLAB to combine the parameters in the denumerator (e.g. use `conv()` function).

- Assign the numerator and denumerator vectors into the transfer function (e.g. use `tf()` function).
- Assign the feedback system as (e.g. `feedback()` function).
- Assign the unit step as (e.g. use `step()` function).

MATLAB code:

```
% demo22.m

numg=15*[1 3 7];
deng=conv([1 3 7], poly([-1 -3]));
G=tf(numg,deng);
T=feedback(G,1);
step(T);
```

The response of the simulation is as given below.



It looks like the second order approximation of the given system resembles to the actual response of the given system.

Exercise 3 (Reduction of Block Diagrams)

Reduce the system shown below to a single transfer function, $T(s) = C(s)/R(s)$ using MATLAB.

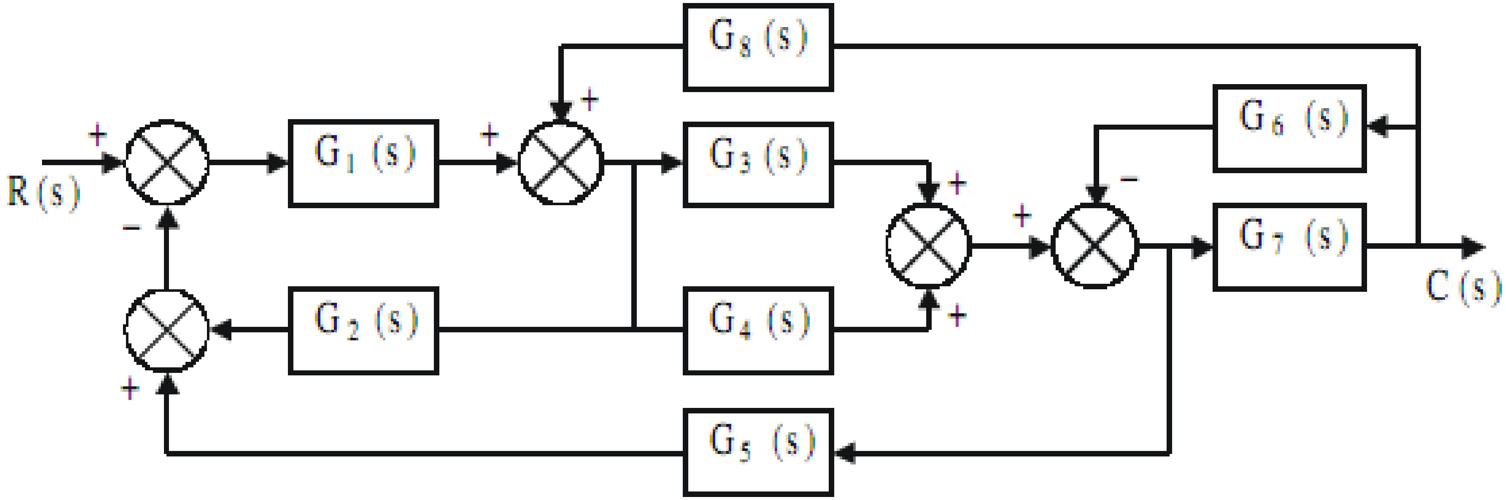
The transfer functions of the blocks in the system are given as:

$$G_1(s) = \frac{1}{s+7} \qquad G_2 = \frac{1}{s^2 + 6s + 5} \qquad G_3(s) = \frac{1}{s+8}$$

$$G_4(s) = \frac{1}{s} \qquad G_5(s) = \frac{7}{s+3} \qquad G_6(s) = \frac{1}{s^2 + 7s + 5}$$

$$G_7(s) = \frac{5}{s+5} \qquad G_8(s) = \frac{1}{s+9}$$

The overall system is represented as a block diagram shown below:

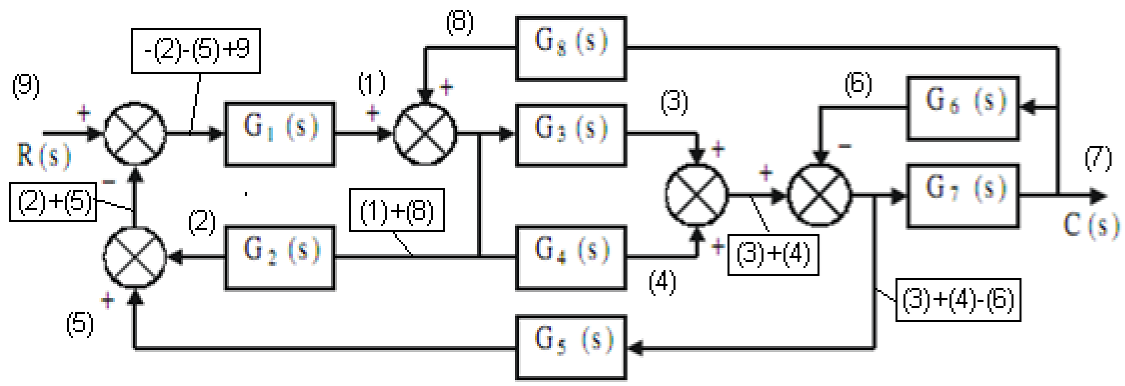


Answer

To model the system as block diagram model in MATLAB, you are required to do these things:

- Model of the blocks in the system as the transfer functions of the subsystems in the system (e.g. use `tf()` function)
- Combine these blocks into the overall system (use `append()` function)
- Links determination and connecting these block following arrangement in the given control systems (e.g. use `connect()` function)

Links determination:



Subsystems	Output	Input 1	Input 2	Input 3
G1	1	-2	-5	-9
G2	2	1	8	0
G3	3	1	8	0
G4	4	1	8	0
G5	5	3	4	-6
G6	6	7	0	0
G7	7	3	4	-6
G8	8	7	0	0

MATLAB code:

```
% demo23.m

G1=tf([0 0 1],[0 1 7]);
G2=tf([0 0 1],[1 6 5]);
G3=tf([0 0 1],[0 1 8]);
G4=tf([0 0 1],[0 1 0]);
G5=tf([0 0 7],[0 1 3]);
G6=tf([0 0 1],[1 7 5]);
G7=tf([0 0 5],[0 1 5]);
G8=tf([0 0 1],[0 1 9]);
G9=tf([0 0 1],[0 0 1]);
```

```
T1=append(G1,G2,G3,G4,G5,G6,G7,G8,G9);
Links=[1 -2 -5 9;2 1 8 0;3 1 8 0;4 1 8 0;5 3 4 -6;6
7 0 0;7 3 4 -6;8 7 0 0];
Inputs=9;
Outputs=7;
Ts=connect(T1,Links,Inputs,Outputs);
T=tf(Ts);
```

If you type T in the command window, this results in MATLAB as follows:

```
T =

                  10 s^7 + 290 s^6 + 3350 s^5 + 1.98e04 s^4 + 6.369e04 s^3 + 1.089e05 s^2 + 8.895e04 s + 2.7e04
  ---------------------------------------------------------------------------------------------------------------
  s^10 + 45 s^9 + 866 s^8 + 9305 s^7 + 6.116e04 s^6 + 2.533e05 s^5 + 6.57e05 s^4 + 1.027e06 s^3 + 8.909e05 s^2 + 3.626e05 s + 42000

Continuous-time transfer function.
```

Exercise 4 (Unit Step Response& Damping Ratios)

For the closed-loop system defined by:

$$\frac{C(s)}{R(s)} = \frac{1}{s^2 + 2\zeta s + 1}$$

a. Plot the unit-step response curves $c(t)$ for damping ratio, ζ= 0, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8, and 1.0. Natural frequency (ω_n) is normalized to 1.

b. Plot a three dimensional plot of (a).

To plot the two-and three-dimensional responses of the system:

- Initialise and assign time and damping ratio vectors e.g. `t` and `zeta`.
- Apply the contents of the vectors above inside the `For.. Loop` to the numerator and denumerator vectors before being assigned to the step function (e.g. use `step()` function).
- Plot the two-dimensional graph of the response of the system (e.g. use `plot()` function).
- Assign appropriate labelling of the axes (e.g. use `xlabel()` and `ylabel()` functions) and parameters (use `text()` function) and name of the graph title (e.g. use `title()` function)
- Plot the three-dimensional graph of the response of the system (e.g. use `mesh()` function).
- Assign appropriate labelling of the axes (e.g. use `xlabel()`, `ylabel()` and `zlabel()` functions) and name of the graph title (e.g. use `title()` function).

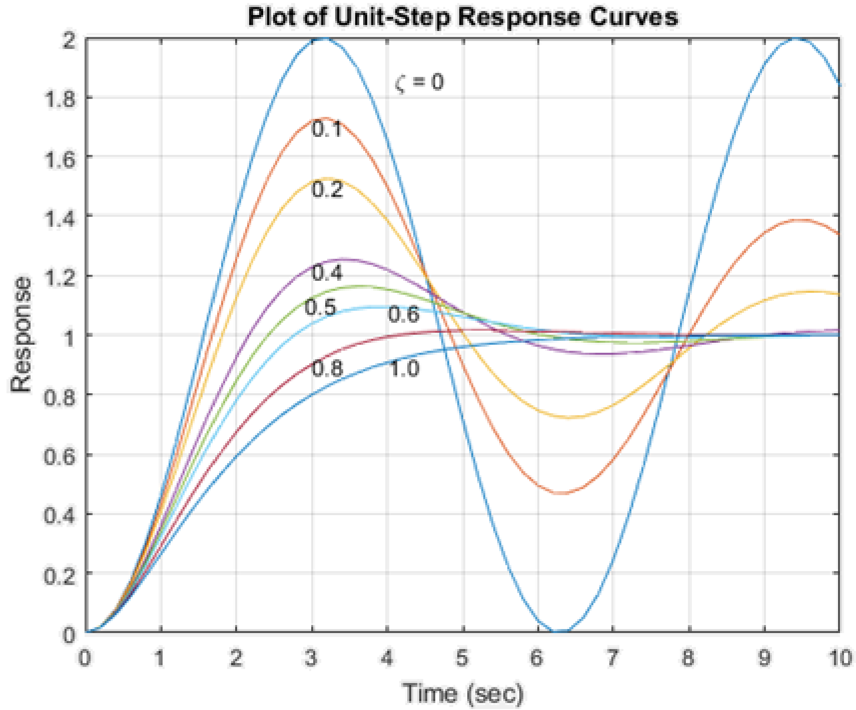
MATLAB code:

```
% demo24.m

% Two-dimensional plot and three-dimensional plot of unit-
step
% response curves for the standard second-order system with
wn = 1
% and zeta = 0, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8, and 1.0

t=0:0.2:10;
zeta=[0 0.1 0.2 0.4 0.5 0.6 0.8 1.0];

for n=1:8
       num=[0 0 1];
       den=[1 2*zeta(n) 1];
       [y(1:51,n),x,t]=step(num,den,t);
end
```

```
% Two-dimensional diagram with the command plot (t, y)

plot(t,y)

grid

title('Plot of Unit-step Response Curves')

xlabel('Time (sec)')
ylabel('Response')

text(4.1,1.86,'\zeta = 0')
text(3.0,1.7,'0.1')
text(3.0,1.5,'0.2')
text(3.0,1.22,'0.4')
text(2.9,1.1,'0.5')
text(4.0,1.08,'0.6')
text(3.0,0.9,'0.8')
text(4.0,0.9,'1.0')
```

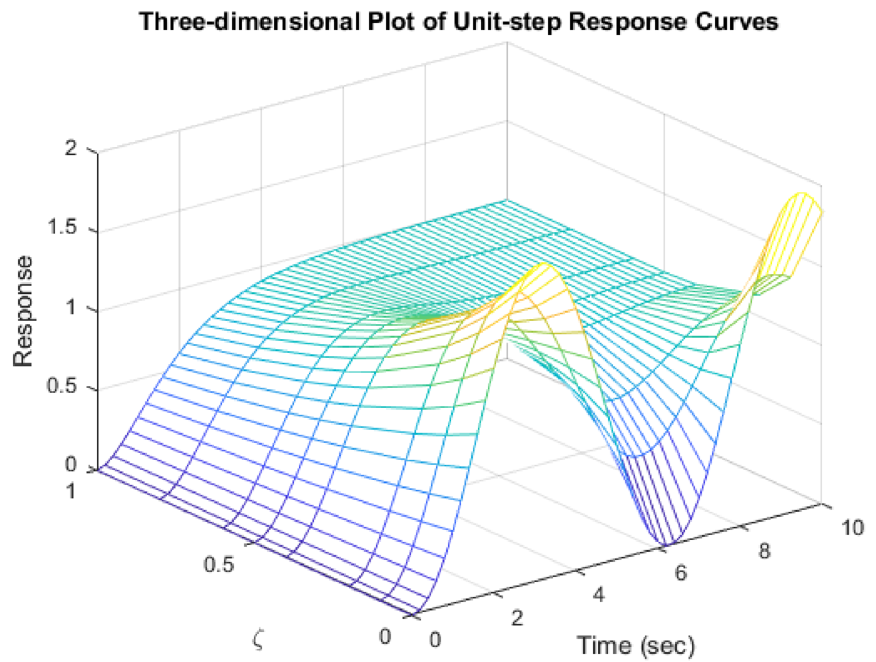
```
% For three dimensional plot, we use the command mesh (t, zeta, y')
figure
mesh(t,zeta,y')
title('Three-dimensional Plot of Unit-step Response Curves')
xlabel('Time (sec)')
ylabel('\zeta')
zlabel('Response')
```

The following diagram shows a plot of unit-step response curves.



- The graph shows various system responses according to various types of damping constants (ζ) i.e. 0, 0.1, 0.2 … 1.
- In ζ = 0, the response of the system does not experience any damping e.g. oscillation, whereas 0.1 < ζ < 0.6 the responses of the system are underdamped, and for ζ = 0.8 and 1 the responses of the system are overdamped.

The following diagram outlines the three-dimensional plot of unit-step response curves.



- The graph shows the three-dimensional contour of the plot of response, damping ratio (ζ) and time.
- It shows the progression of response of the system from oscillation response, underdamped response and overdamped response as the damping ratio is increased.