

EE3114 Systems and Control

Lab - 1

Semester A, 2020-21

MatLab

Practice + Simple Exercises (60 minutes)

Main Exercise (30 minutes)

Break (10 minutes)

Simulink

Practice (40 minutes)

Exercise (30 minutes)

Practice

Transfer function

$$G(s) = \frac{b_1 s^m + b_2 s^{m-1} + \dots + b_m s + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \dots + a_n s + a_{n+1}}$$

Matlab code for $G(s)$:

```
num = [b1, b2, ..., bm, bm+1];  
den = [a1, a2, ..., an, an+1];  
G = tf(num, den); % function tf() is used to create real-valued or  
                  complex-valued transfer function models
```

Example:

```
num=[2 3 4];  
den=[5 6 7];  
G=tf(num,den)
```

Result:

Transfer function:

$$\frac{2s^2 + 3s + 4}{5s^2 + 6s + 7}$$

Note 1: The transfer function is sometimes given by polynomial factor multiplications

Example: For simplicity, the transfer function

$$G(s) = \frac{2(s+1)(s^2+3s+2)}{(s+4)^3(s^2+10)}$$

can be described by the following Matlab code:

```
s=tf('s'); %Define s as Laplace variable  
G=2*(s+1)*(s^2+3*s+2)/((s+4)^3*(s^2+10))
```

Result:

Transfer function:

$$\frac{2 s^3 + 8 s^2 + 10 s + 4}{s^5 + 12 s^4 + 58 s^3 + 184 s^2 + 480 s + 640}$$

Note 2: Function `tfdata()` is used to find the numerator and denominator of the transfer function

Example:

```
s=tf('s');  
G=2*(s+1)*(s^2+3*s+2)/((s+4)^3*(s^2+10));  
[num den]=tfdata(G,'v')    % parameter 'v' means that the function  
                             returns the numerator and denominator  
                             as row vectors rather than cell arrays
```

Result:

```
num =                                %The polynomial coefficients of  
                                the terms s^5 and s^4 are 0  
    0    0    2    8   10    4  
  
den =  
    1   12   58  184  480  640
```

Note 3: Transfer function may contain time delay term, i.e., the delayed transfer function is represented by

$$G(s)e^{-as}$$

Example:

```
s=tf('s');  
G=2*(s+1)*(s^2+3*s+2)/((s+4)^3*(s^2+10));  
set(G,'ioDelay',4); %set time delay as 4  
G %display G
```

Result:

Transfer function:

$$\exp(-4s) * \frac{2s^3 + 8s^2 + 10s + 4}{s^5 + 12s^4 + 58s^3 + 184s^2 + 480s + 640}$$

Zeros and Poles of transfer function

$$G(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}$$

Matlab code for $G(s)$:

```
z = [z1; z2; ...; zm];  
p = [p1; p2; ...; pn];  
G = zpk(z, p, K);
```

Note: Elements in the sets z and p must be separated by semicolon rather than comma or space.

Example:

```
z=[-1;-2+2i;-2-2i];  
p=[-1;-2;-3;-4];  
G=zpk(z,p,3)
```

Result:

Zero/pole/gain:

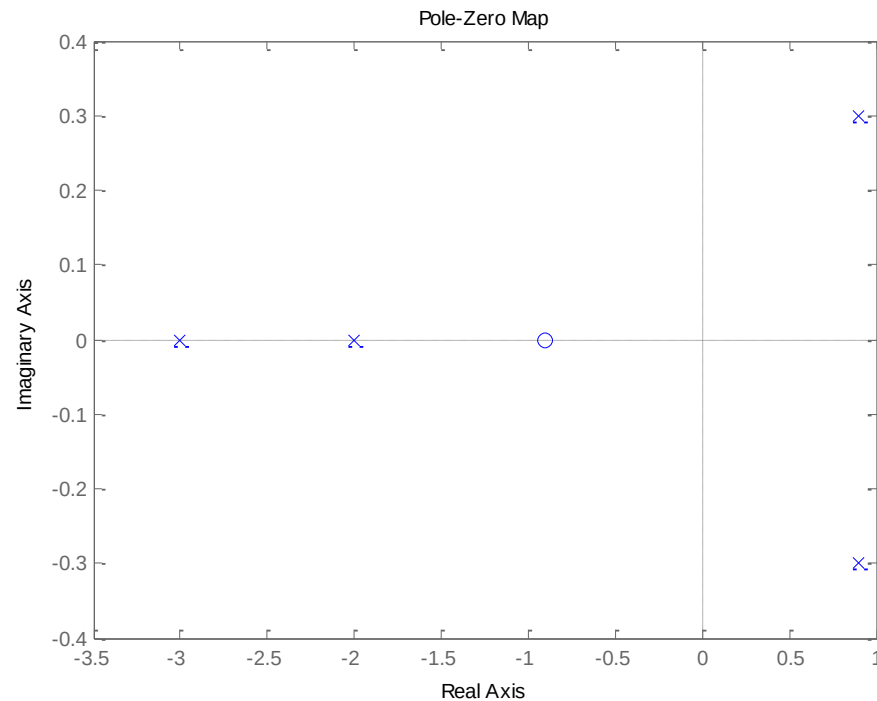
$$\frac{3 (s+1) (s^2 + 4s + 8)}{(s+1) (s+2) (s+3) (s+4)}$$

Note 1: Function pzmap() is used to plot poles and zeros of transfer function

Example:

```
s=tf('s');  
G=(0.05*s+0.045)/((s^2-1.8*s+0.9)*(s^2+5*s+6));  
pzmap(G);
```

Result:



Note 2: Function `zpkdata()` is used to find zeros, poles and the gain of the transfer function

Example: Determine the zeros, poles and the gain of transfer function:

$$G(s) = \frac{5s^3 + 50s^2 + 155s + 150}{s^4 + 11s^3 + 41s^2 + 61s + 30}$$

Matlab code:

```
num=[5 50 155 150];  
den=[1 11 41 61 30];  
G=tf(num,den);  
  
[z,p,k]=zpkdata(G,'v')
```

Result:

$z =$

-5.0000
-3.0000
-2.0000

$p =$

pole-zero cancellation

-5.0000
-3.0000
-2.0000
-1.0000

$k =$

5

Pole-zero cancelation and minimal realization

Minimal Realization of a Transfer Function is its simplest form after zero-pole cancellation

Example: What is the minimal realization of the following transfer function?

$$G(s) = \frac{5s^3 + 50s^2 + 155s + 150}{s^4 + 11s^3 + 41s^2 + 61s + 30}$$

Matlab code:

```
num=[5 50 155 150];  
den=[1 11 41 61 30];  
G=tf(num,den);  
zpk(G); %zero-pole-gain model  
Gm=minreal(G) %minimal realization
```

Result:

Transfer function:

$$\frac{5}{s + 1}$$

Exercise

Use MatLab

Find the minimal realization of the following transfer function:

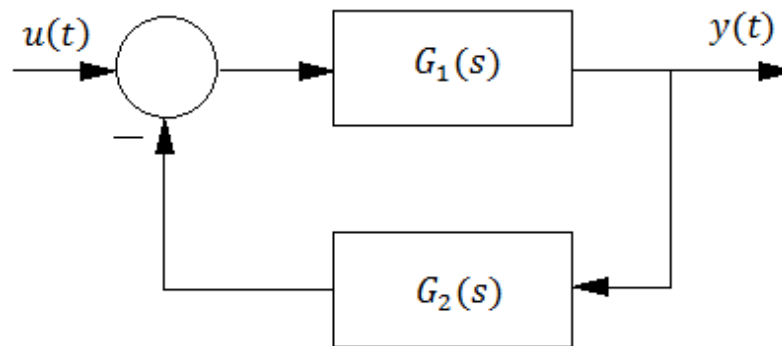
$$G(s) = \frac{s(s^2 - 1)}{s^4 + s^3 - s - 1}$$

Show your MatLab code

0.5 marks

Transfer function of feedback system

Negative Feedback system



Matlab code for the transfer function of the closed-loop system:

```
G=feedback(G1,G2);
```

Example:

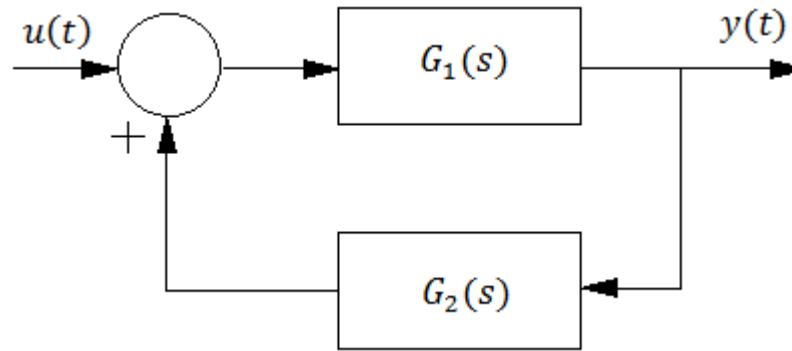
```
s=tf('s');  
G1=1/(s+1);  
G2=(s+2)/(s^2+7*s+10);  
feedback(G1,G2)
```

Result:

Transfer function:

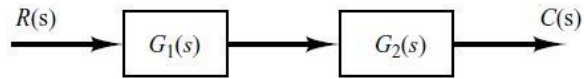
$$\frac{s^2 + 7s + 10}{s^3 + 8s^2 + 18s + 12}$$

Positive feedback system

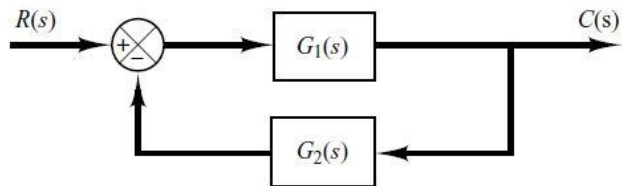
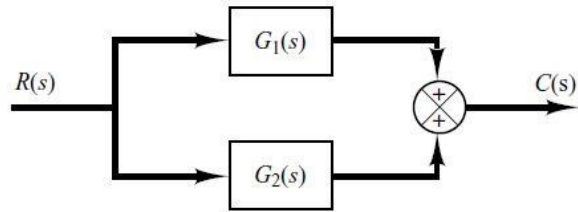


Matlab code for the transfer function of the closed-loop system:

```
G=feedback(G1,G2,1);
```



```
[num, den] = series(num1,den1,num2,den2)
[num, den] = parallel(num1,den1,num2,den2)
[num, den] = feedback(num1,den1,num2,den2)
```



Example:

$$G_1(s) = \frac{10}{s^2 + 2s + 10} = \frac{\text{num1}}{\text{den1}}$$

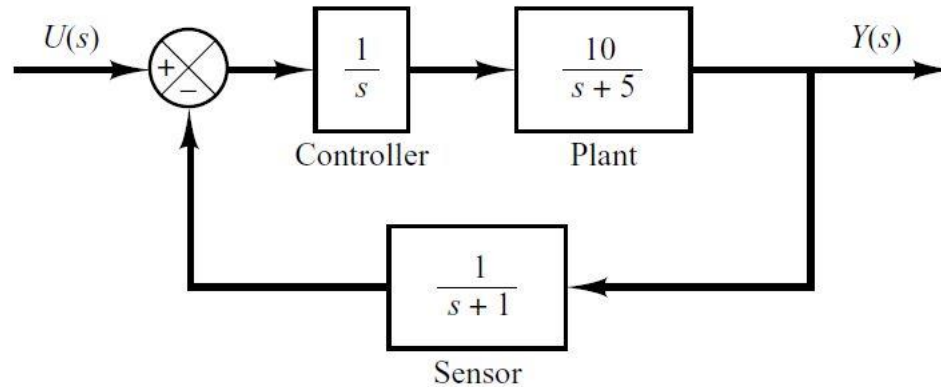
$$G_2(s) = \frac{5}{s + 5} = \frac{\text{num2}}{\text{den2}}$$

```
num1 = [10];
den1 = [1 2 10];
num2 = [5];
den2 = [1 5];
[num, den] =
series(num1,den1,num2,den2);
printsys(num,den)
```

num/den =

$$\frac{50}{s^3 + 7s^2 + 20s + 50}$$

Exercise



Find the Transfer Function of the feedback system:

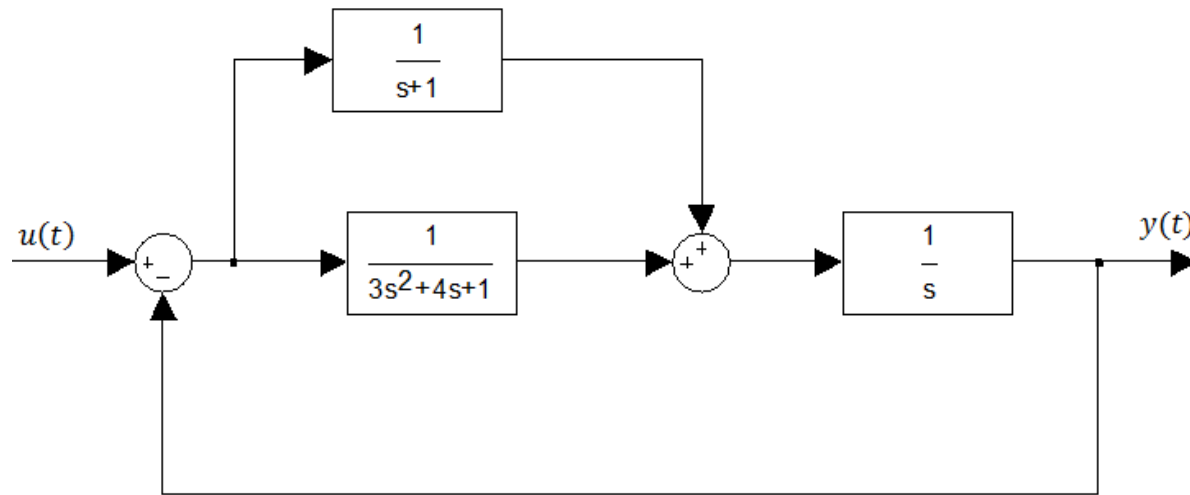
$$Y(s)/U(s) =$$

Show your MatLab code

1.0 marks

Reduction of complicated block diagrams

Example: What is the transfer function of the following closed-loop system?



Matlab code:

```
num1=1;  
den1=[1 1];  
G1=tf(num1,den1);
```

```
num2=1;  
den2=[3 4 1];  
G2=tf(num2,den2);
```

```
num3=1;  
den3=[1 0];  
G3=tf(num3,den3);
```

```
Gp=G1+G2;           %blocks in parallel
```

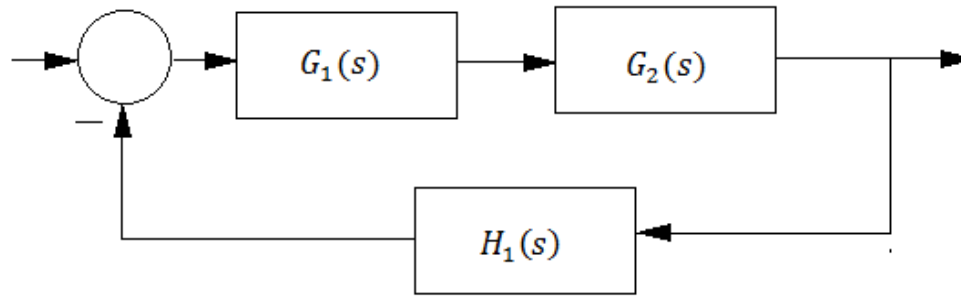
```
Gs=G3*Gp;           %blocks in cascade
```

```
Gc=feedback(Gs,1);  %unity feedback system
```

```
minreal(Gc)          %minimal realization
```

Impulse Response of a control system

Example: Find the open-loop and closed-loop impulse responses of the system:



where

$$G_1(s) = \frac{4}{s^3 + 2s^2 + 3s + 4}$$

$$G_2(s) = \frac{s - 3}{s + 3}$$

$$H_1(s) = \frac{1}{0.01s + 1}$$

Matlab code:

```
num=4;
den=[1 2 3 4];
G1=tf(num,den); %block G1(s)

num2=[1 -3];
den2=[1 3];
G2=tf(num2,den2); %block G2(s)

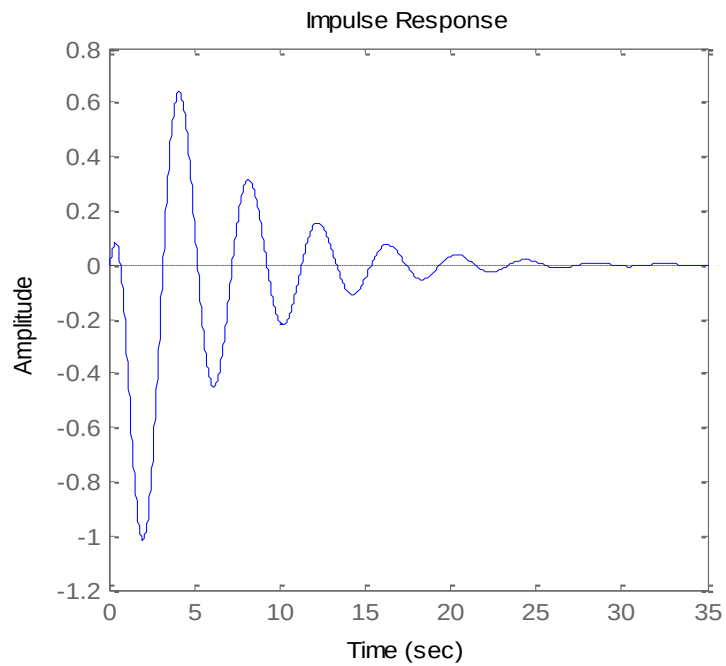
num3=1;
den3=[0.01 1];
H1=tf(num3,den3); %feedback block H1(s)

G=G1*G2;
G3=feedback(G,H1);

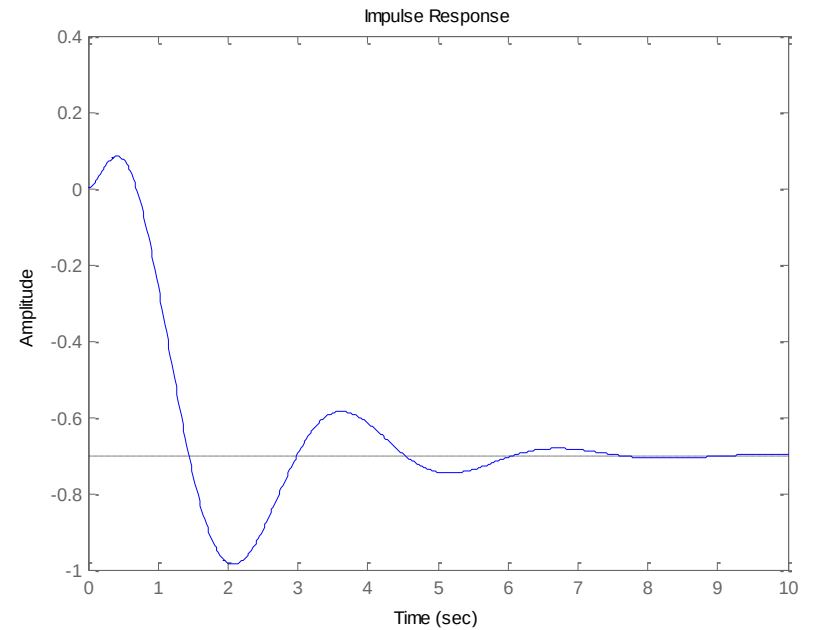
impulse(G); %response of the open-loop system

figure();
impulse(G3); % response of the closed-loop system
```

Results:



Response of open-loop system



Response of closed-loop system

Step Response of a control system

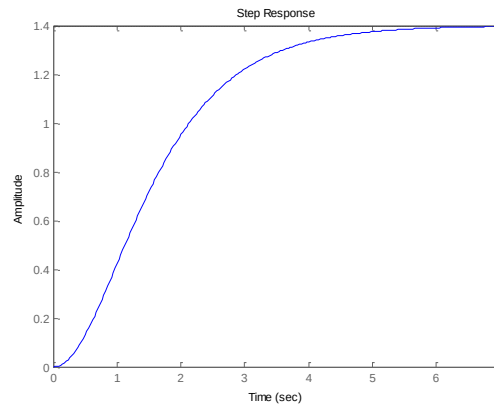
Example: Find the step response of the system:

$$G(s) = \frac{s^2 + 13s + 42}{s^4 + 11s^3 + 41s^2 + 61s + 30}$$

Matlab code:

```
s=tf('s');  
G=(s^2+13*s+42)/(s^4+11*s^3+41*s^2+61*s+30);  
step(G);
```

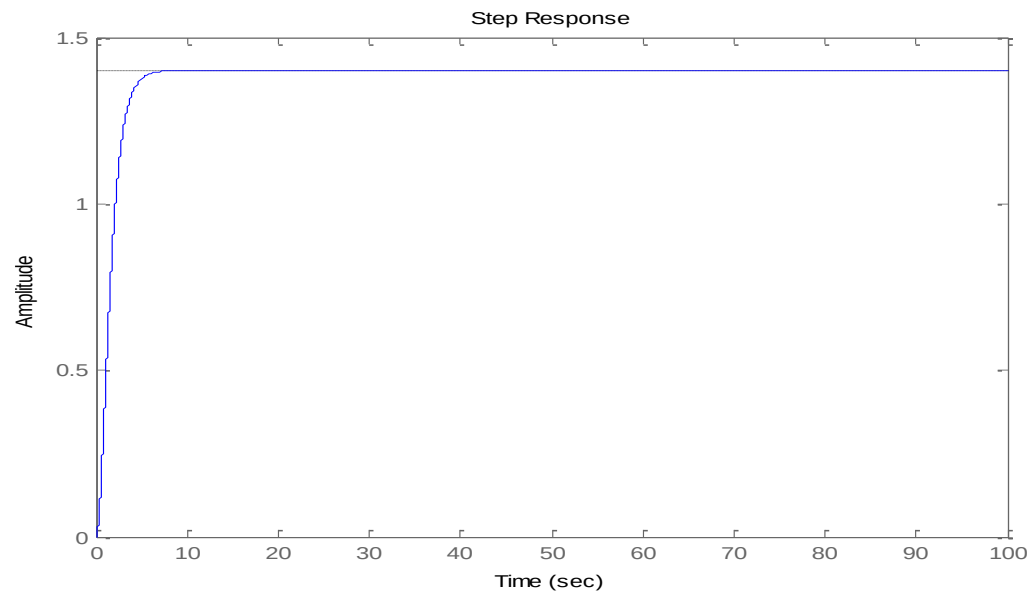
Result:



If specifying the simulation time t, Matlab code can be modified to:

```
s=tf('s');  
G=(s^2+13*s+42)/(s^4+11*s^3+41*s^2+61*s+30);  
t=0:0.01:100; %simulation time range  
step(G,t);
```

Result:



Ramp Response of a control system

Example: Find the ramp response of the system:

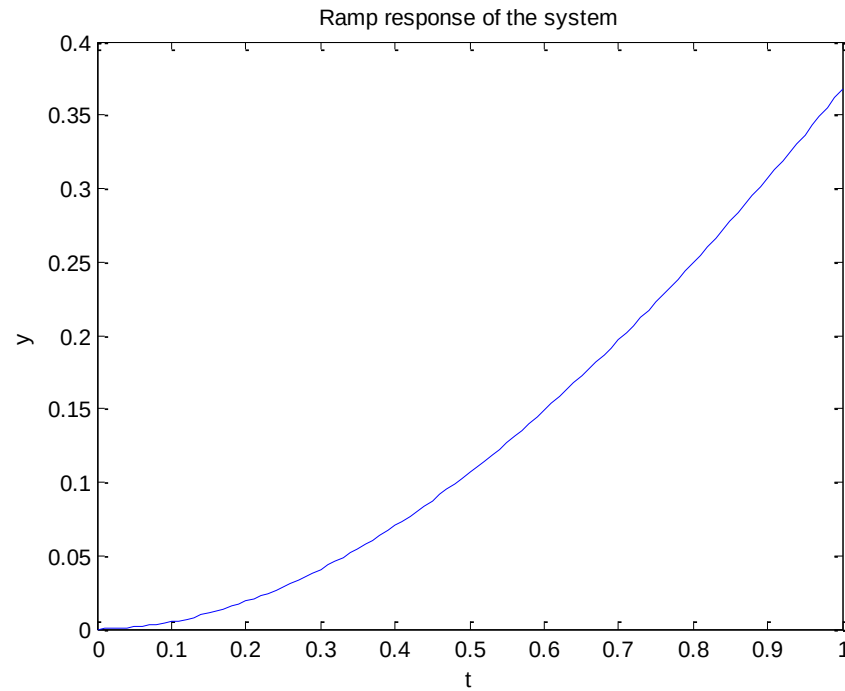
$$G(s) = \frac{K}{Ts + 1}.$$

where $K=1$, and $T=1$.

Matlab code:

```
K=1;
T=1;
s=tf('s');
G=K/(T*s+1);
t=0:0.01:1;
u=t;
[y,t]=lsim(G,u,t);
plot(t,y);
xlabel('t');
ylabel('y');
title('Ramp response of the system');
```

Result:



Note: Function `lsim()` can be used to simulate the response of a system to any input function u .

Step Response of a control system with delay

Example: Find the step response of the system:

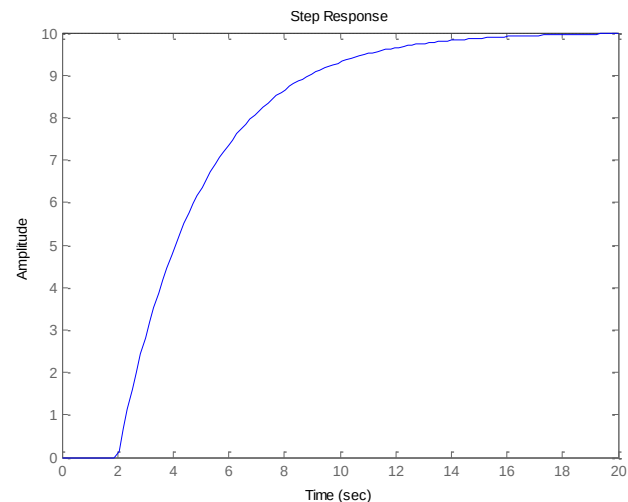
$$G(s) = \frac{K}{Ts + 1} e^{-as}$$

Where $K=10$, $T=3$, and $a=2$.

Matlab code:

```
K=10;  
T=3;  
s=tf('s');  
G=K/(T*s+1);  
set(G,'ioDelay',2);  
step(G);
```

Result:



Exercise

Find the step response of the following system:

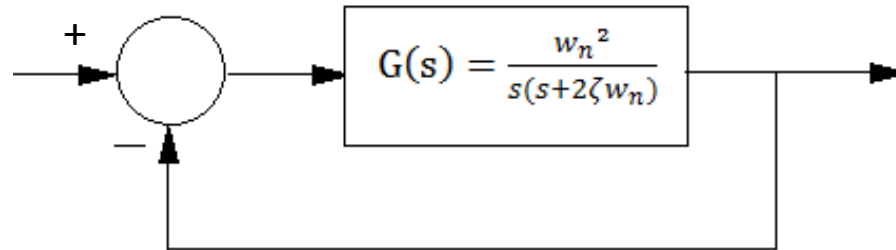
$$G(s) = \frac{3s^3 + 25s^2 + 72s + 80}{s^4 + 8s^3 + 40s^2 + 96s + 80}$$

Show your MatLab code

0.5 marks

Time-domain analysis of control systems

Example: In the following system, damping ratio is $\zeta = 0.6$ and natural frequency is $\omega_n = 5$ rad/sec. Obtain the rise time t_r , peak time t_p , and maximum overshoot M_p when the system is subject to a Unit Step input

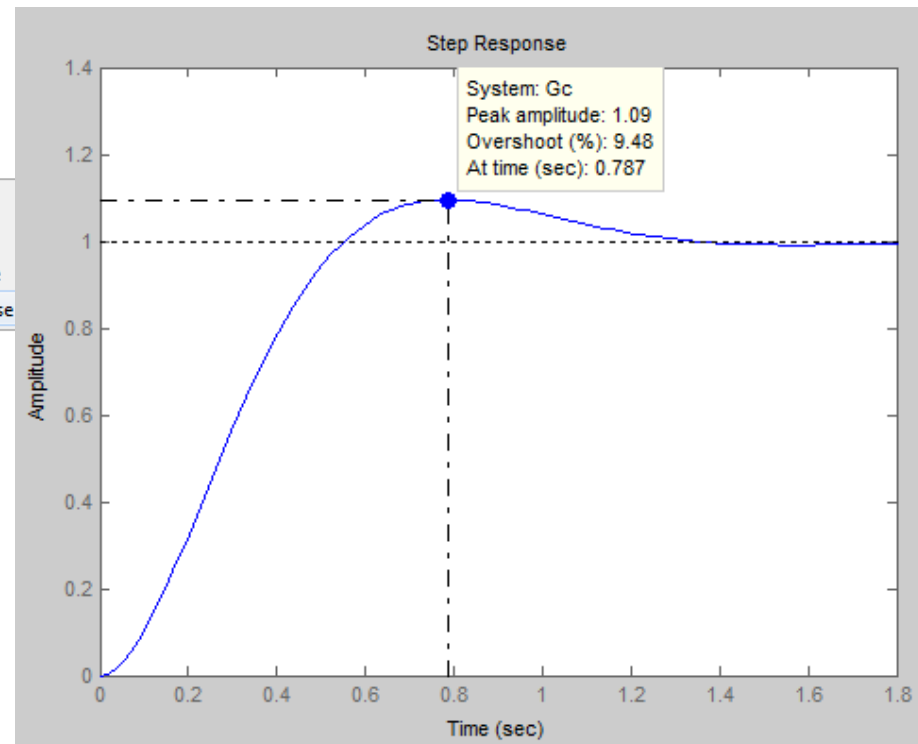
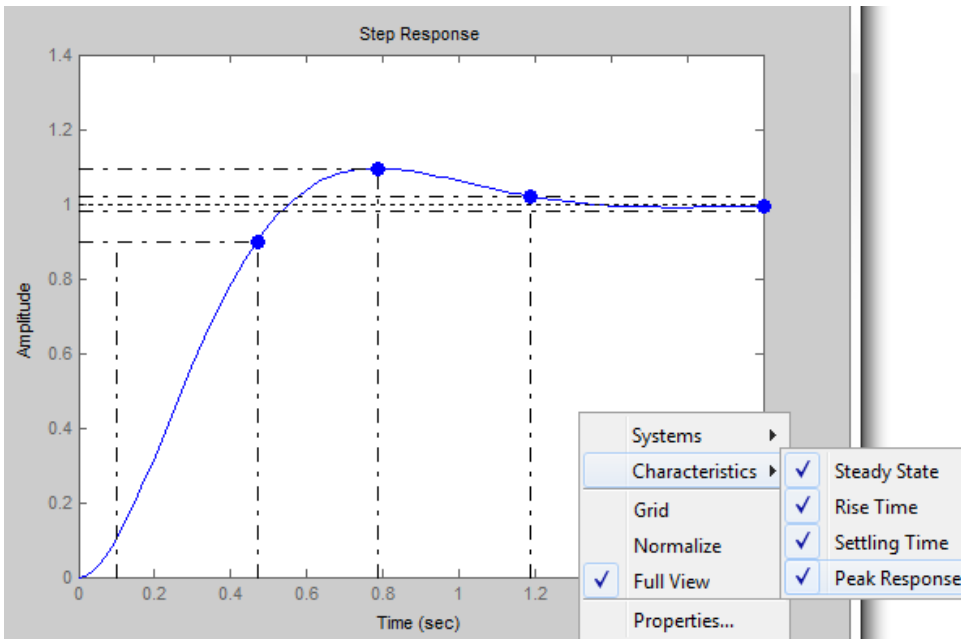


Matlab code:

```
zet=0.6;  
wn=5;  
s=tf('s');  
G=wn^2/(s*(s+2*zet*wn));  
Gc=feedback(G,1);  
step(Gc);
```

Step 1: Right click on the created figure, and choose “Characteristics”.

Step 2: Select “Peak Response” and put the mouse on the peak point of the figure to display the overshoot and the peak time.



Routh-Hurwitz stability criterion

Characteristic equation:

$$D(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0$$

Example: Determine the stability of the system with characteristic equation

$$D(s) = s^3 + s^2 + 2s + 24$$

Matlab code:

```
D=[1 1 2 24];      %Coefficients of the characteristic equation
l=length (D);
disp('-----')
disp('Roots of characteristic equation is:')
roots(D)
if mod(l,2)==0
    m=zeros(l,l/2);
    [cols,rows]=size(m);
    for i=1:rows
        m(1,i)=D(1,(2*i)-1);
        m(2,i)=D(1,(2*i));
    end
else
    m=zeros(l,(l+1)/2);
    [cols,rows]=size(m);
    for i=1:rows
        m(1,i)=D(1,(2*i)-1);
    end
    for i=1:((l-1)/2)
        m(2,i)=D(1,(2*i));
    end
end
end
```

Results:

Roots of characteristic equation is:

ans =

-3.0000

1.0000 + 2.6458i

1.0000 - 2.6458i

(Continuing on the next page)

Reference: <https://cn.mathworks.com/matlabcentral/fileexchange/25956-routh-hurwitz-stability-test?focused=5140598&tab=function>

```

for j=3:cols
    if m(j-1,1)==0
        m(j-1,1)=0.001;
    end
    for i=1:rows-1
        m(j,i)=(-1/m(j-1,1))*det([m(j-2,1) m(j-2,i+1);m(j-1,1) m(j-1,i+1)]);
    end
end

```

```

disp('-----The Routh-Hurwitz array is:-----')
m

```

```

% Checking for sign change
Temp=sign(m);a=0;
for j=1:cols
    a=a+Temp(j,1);
end
if a==cols
    disp('      ----> System is Stable <----')
else
    disp('      ----> System is Unstable <----')
end

```

Results:

-----The Routh-Hurwitz array is:-----

m =

1	2
1	24
-22	0
24	0

----> System is Unstable <----

Reference: <https://cn.mathworks.com/matlabcentral/fileexchange/25956-routh-hurwitz-stability-test?focused=5140598&tab=function>

Bode Plot

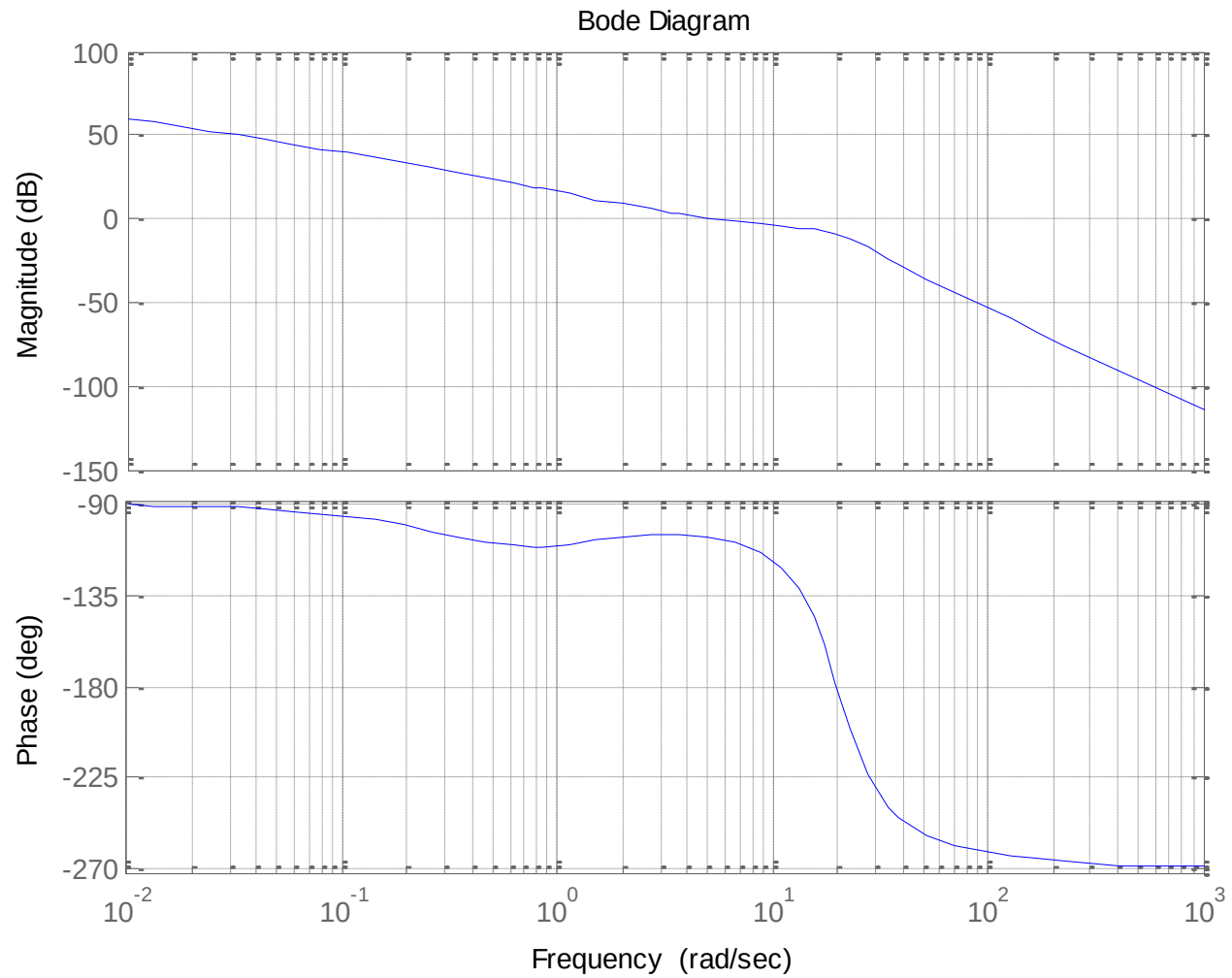
Example: For the system with the following open-loop transfer function, draw its Bode plot.

$$G(s) = \frac{2000(s + 1)}{s(s + 0.5)(s^2 + 14s + 400)}$$

Matlab code:

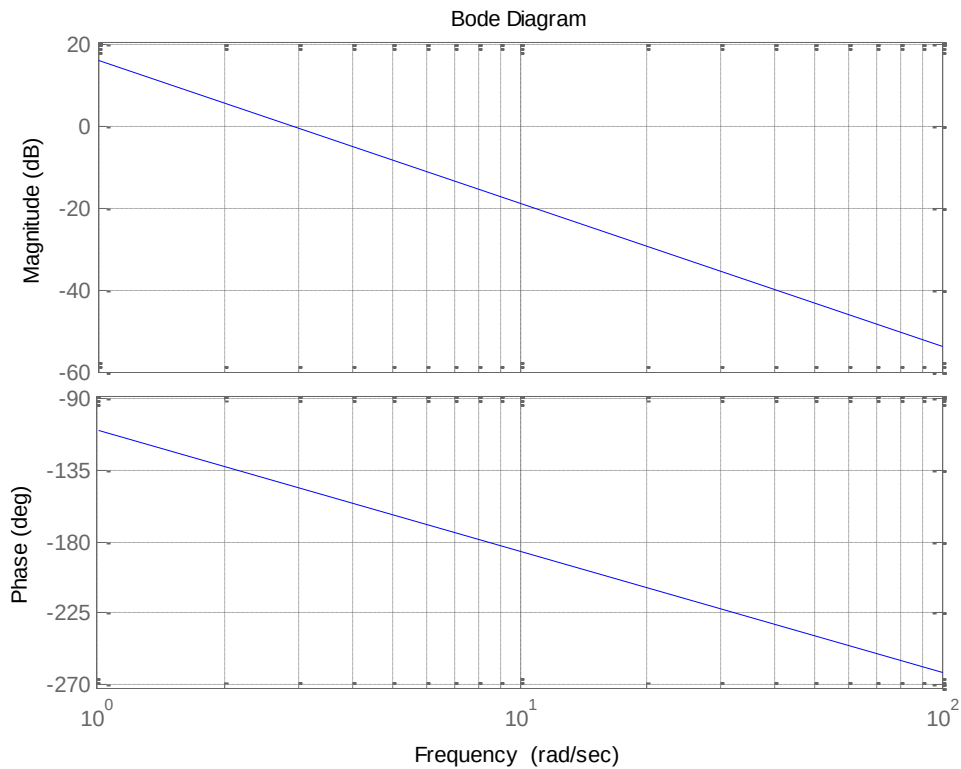
```
s=tf('s');  
G=2000*(s+1)/(s*(s+0.5)*(s^2+14*s+400));  
bode(G);  
grid;
```

Result:



If specifying the frequency range, the Matlab code can be modified to:

```
s=tf('s');  
G=2000*(s+1)/(s*(s+0.5)*(s^2+14*s+400));  
w=[1 100]; %frequency range from 1 to 100  
bode(G,w);  
grid;
```



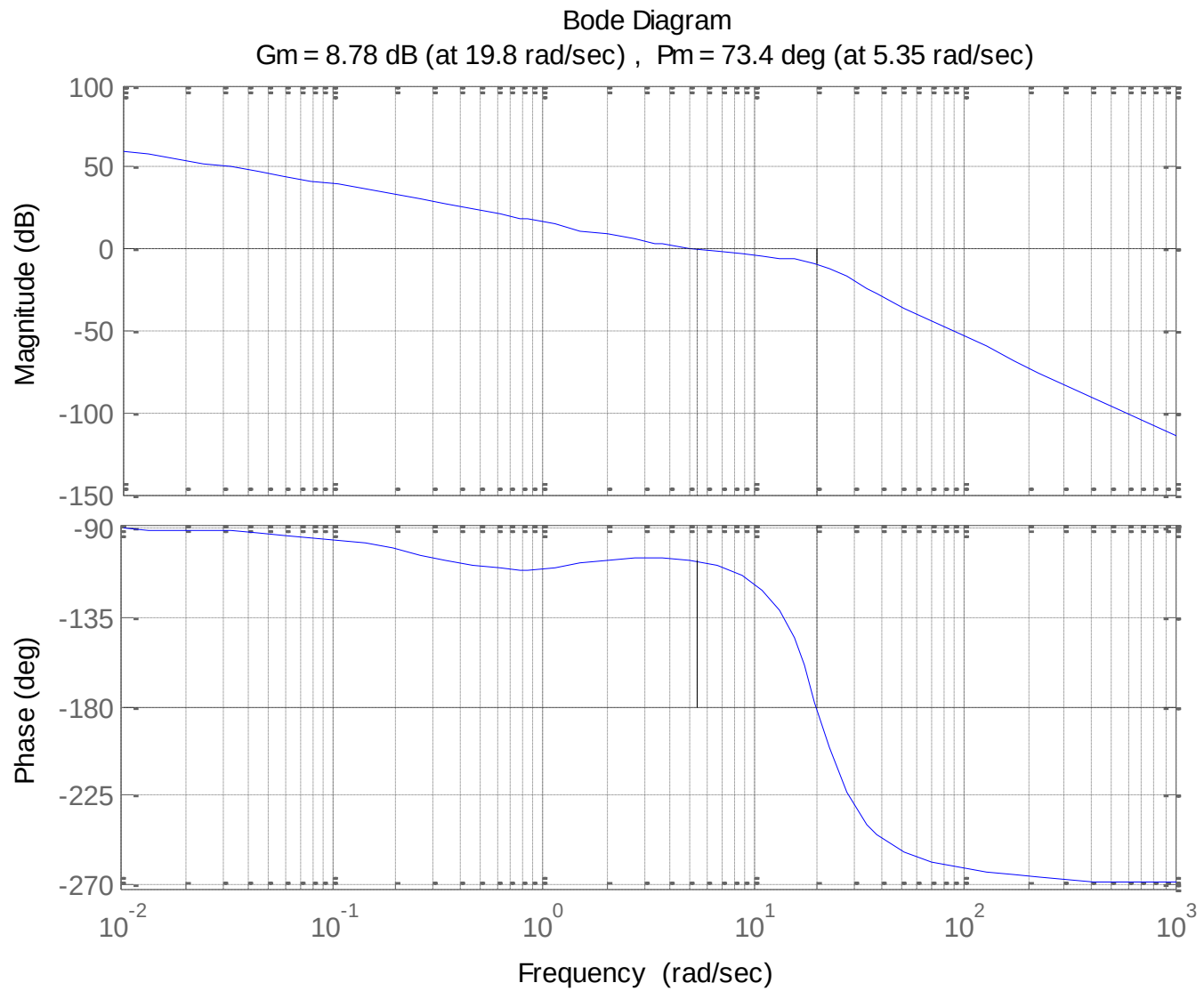
Example: For the system with the following open-loop transfer function, calculate its phase and gain margins, as well as phase and gain crossover frequencies.

$$G(s) = \frac{2000(s + 1)}{s(s + 0.5)(s^2 + 14s + 400)}$$

Matlab code:

```
s=tf('s');  
G=2000*(s+1)/(s*(s+0.5)*(s^2+14*s+400));  
margin(G);  
grid;
```

Result:



Nyquist Plot

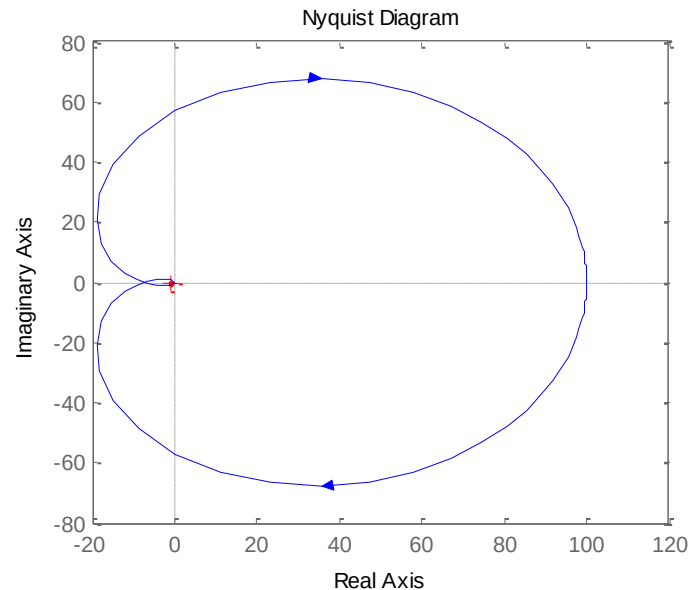
Example: For the system with the following open-loop transfer function, draw its Nyquist plot.

$$G(s) = \frac{1000}{(s + 5)(s^2 + 3s + 2)}$$

Matlab code:

```
s=tf('s');  
G=1000/((s^2+3*s+2)*(s+5));  
nyquist(G);
```

Result:



Root Locus

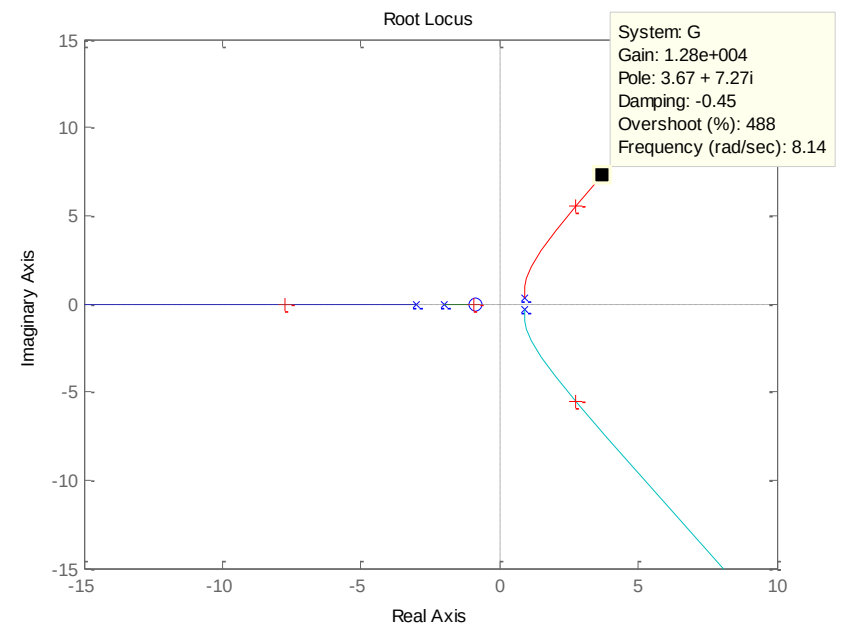
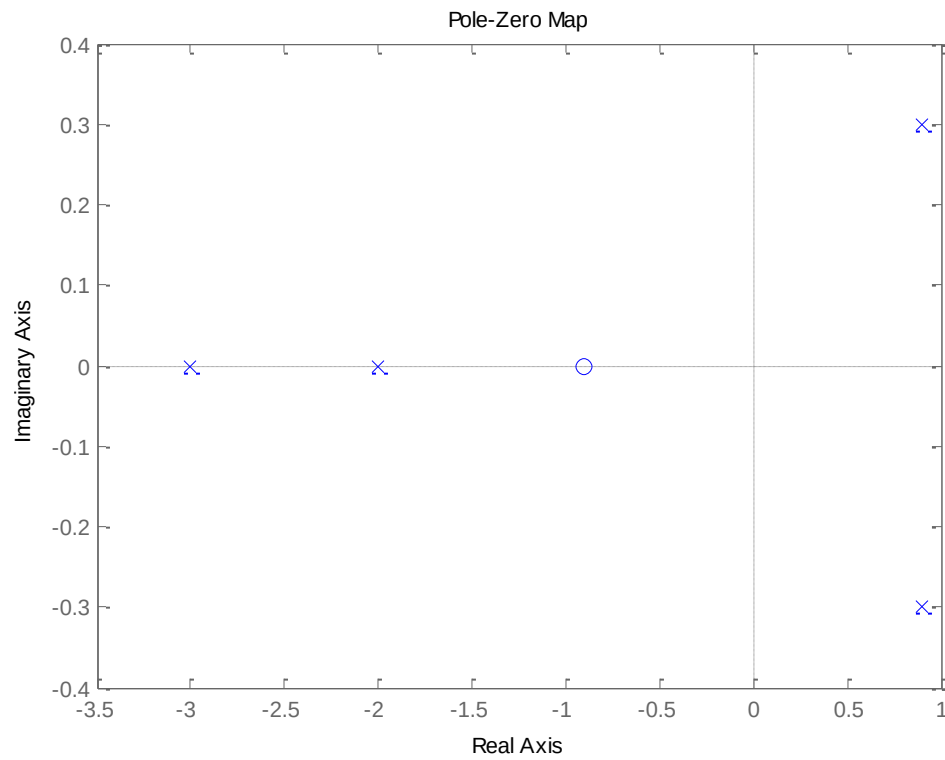
Example: Draw the pole-zero map and the root locus of the following open loop transfer function.

$$G(s) = \frac{0.05s + 0.045}{(s^2 - 1.8s + 0.9)(s^2 + 5s + 6)}$$

Matlab code:

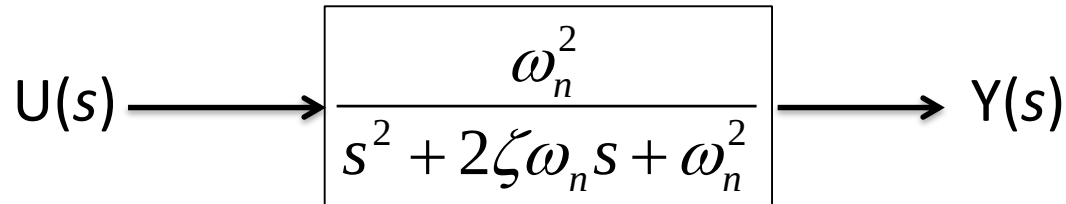
```
s=tf('s');  
G=(0.05*s+0.045)/((s^2-1.8*s+0.9)*(s^2+5*s+6));  
  
pzmap(G); %plot the pole-zero map  
  
figure();  
rlocus(G); %plot the root locus  
[K,Poles]=rlocfind(G); % Find root locus gain for a root interactively  
selecting from the graphics window.
```

Results:



Exercise (30 minutes)

In the standard second-order system:



with damping ratio: $\zeta = 0.5$

natural frequency: $\omega_n = 10$ rad/sec.

Suppose that the input is the Unit Step Function.

Use MatLab to find (Show your MatLab codes):

- rise time $t_r = ?$
- peak time $t_p = ?$
- maximum overshoot $M_p = ?$
- system stability

2.0 marks

BREAK

10 minutes

MatLab

Practice

Exercise

Simulink

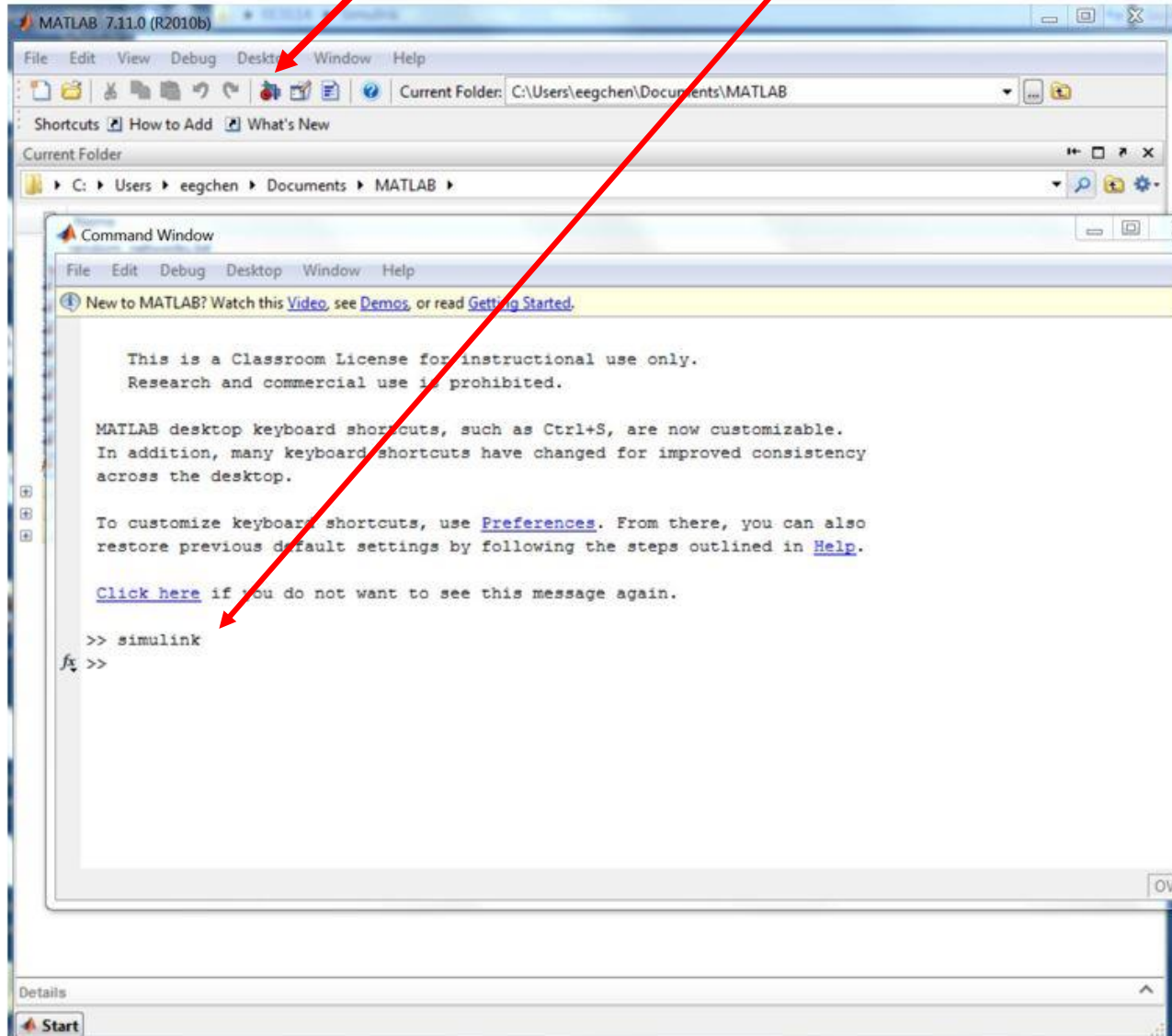
Practice (40 minutes)

Exercise (30 minutes)

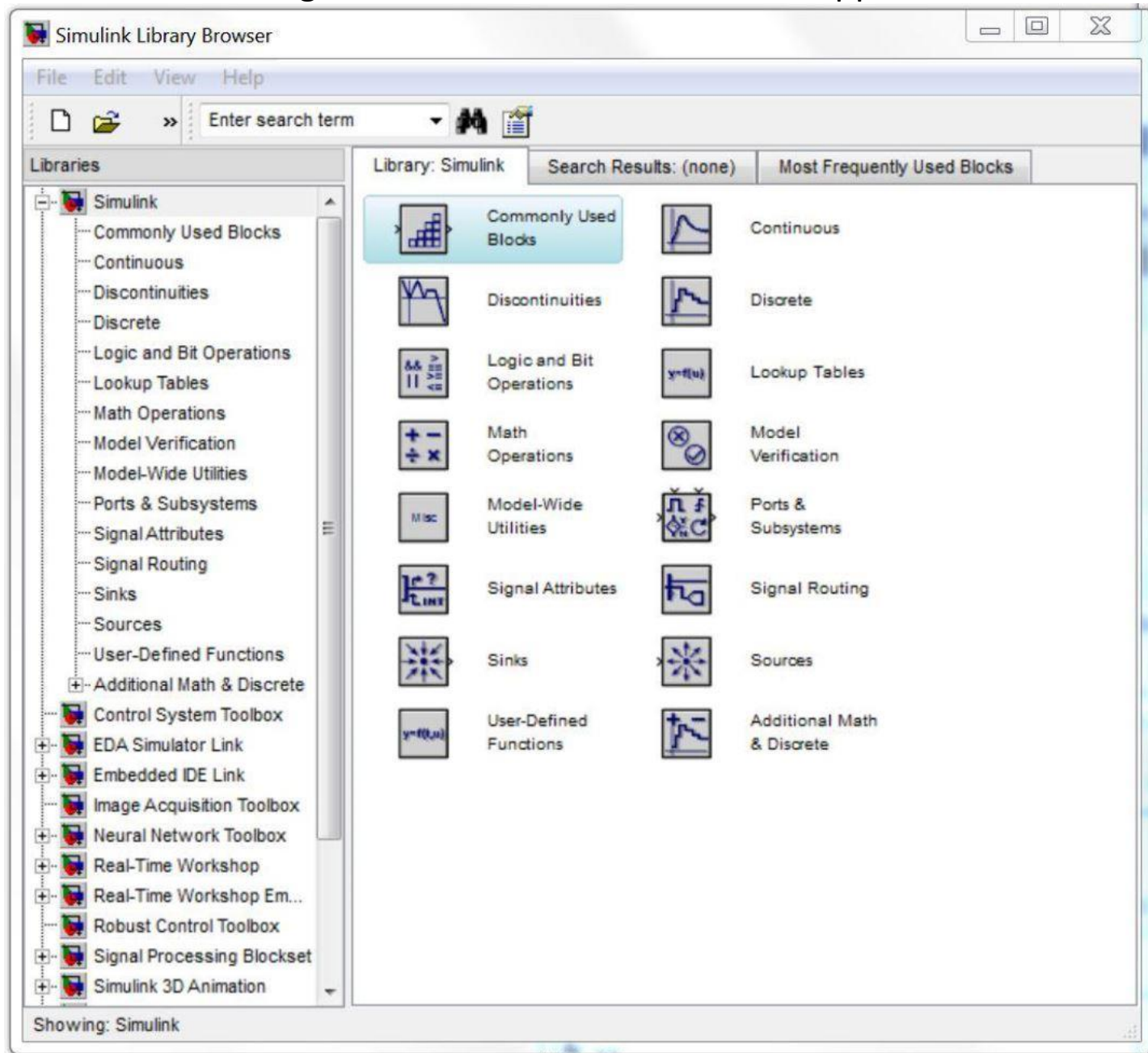
Practice

Start MatLab

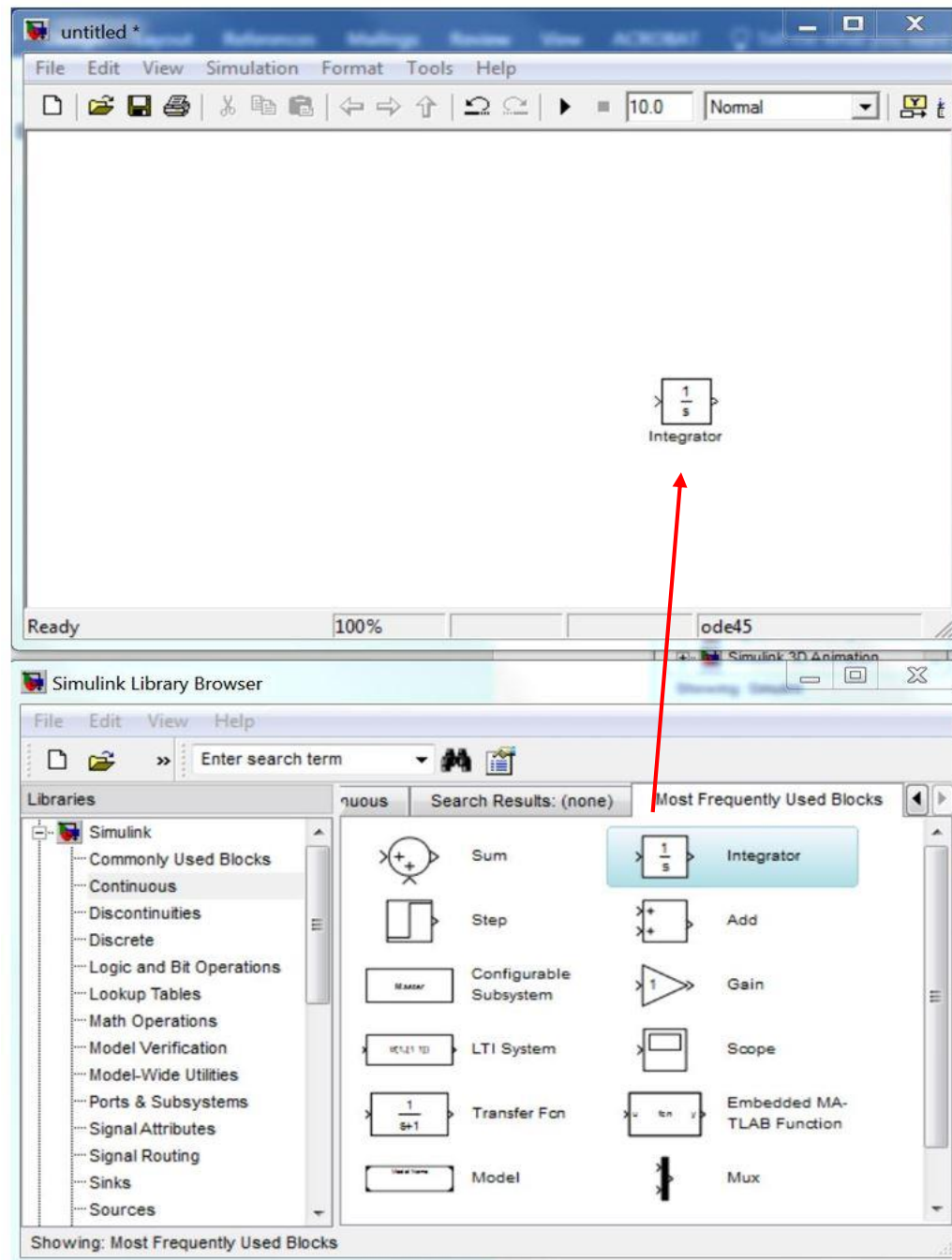
Click "Simulink Icon" or Input "Simulink" Command



The following Simulink interface window will appear:



- Start a new Simulink model page: File > New > Model



Example:

Second-order spring-damper system

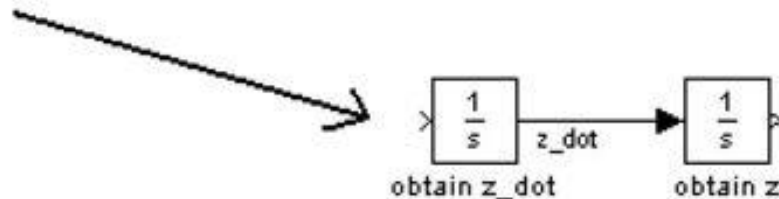
$$m\ddot{z}(t) + c\dot{z}(t) + kz(t) = u(t)$$



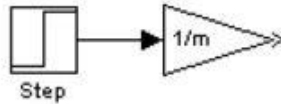
$$\ddot{z}(t) = \frac{1}{m}u(t) - \frac{k}{m}z(t) - \frac{c}{m}\dot{z}(t)$$

On the above Simulink model page:

construct the signal $\ddot{z}(t)$

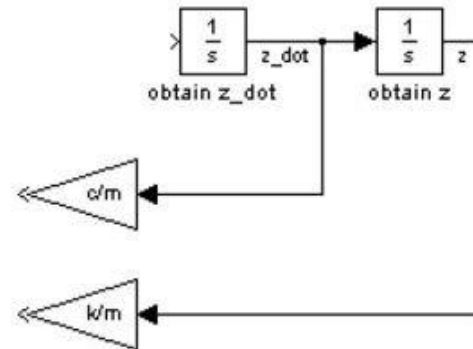


Add some basic blocks, as



“Unit step
input” block

“Gain” block



Next, suppose:

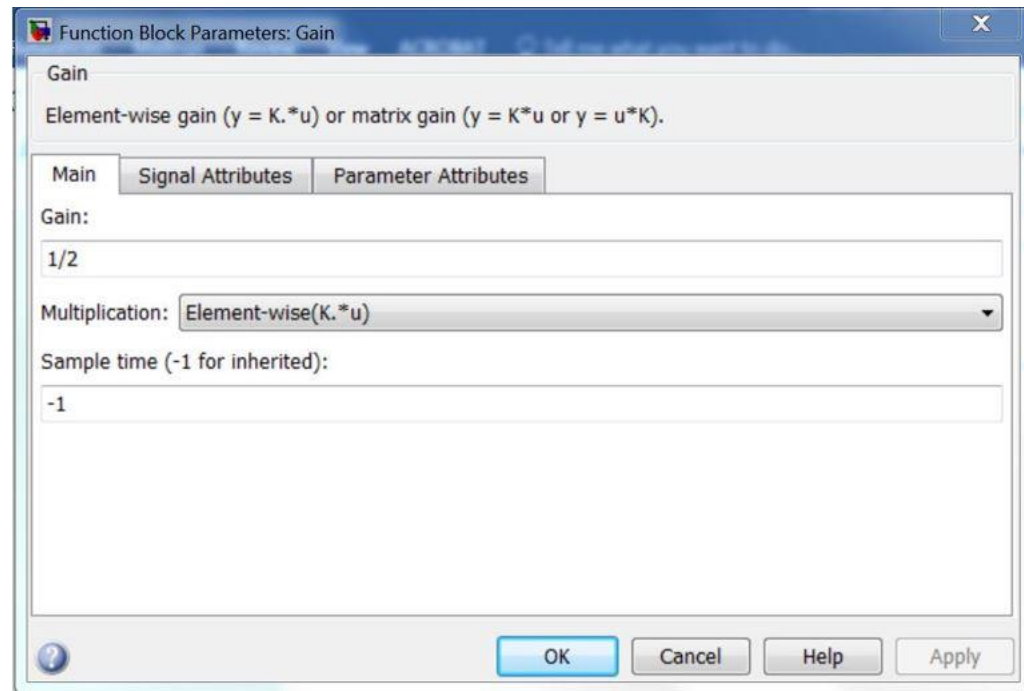
$$k = 2$$

$$c = 0.5$$

$$m = 2$$



Add “Gain” block
and change the
parameter 1/m
to be 1/2, as



Add "Sum" block and change parameters as

Function Block Parameters: Sum

Sum

Add or subtract inputs. Specify one of the following:

- a) string containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
- b) scalar, ≥ 1 , specifies the number of input ports to be summed.

When there is only one input port, add or subtract elements over all dimensions or one specified dimension

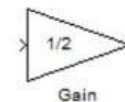
Main Signal Attributes

Icon shape: round

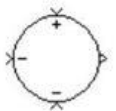
List of signs: +|-|-

Sample time (-1 for inherited): -1

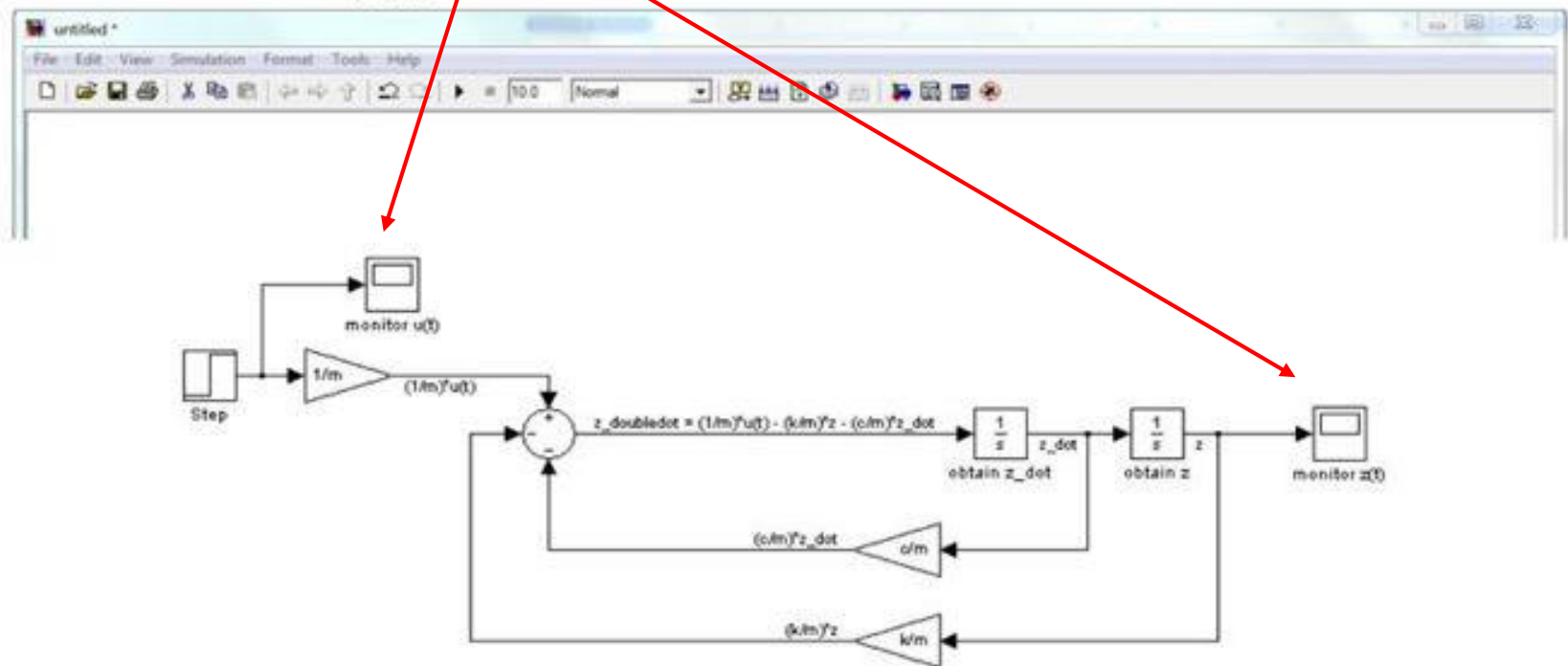
OK Cancel Help Apply



"Sum" block

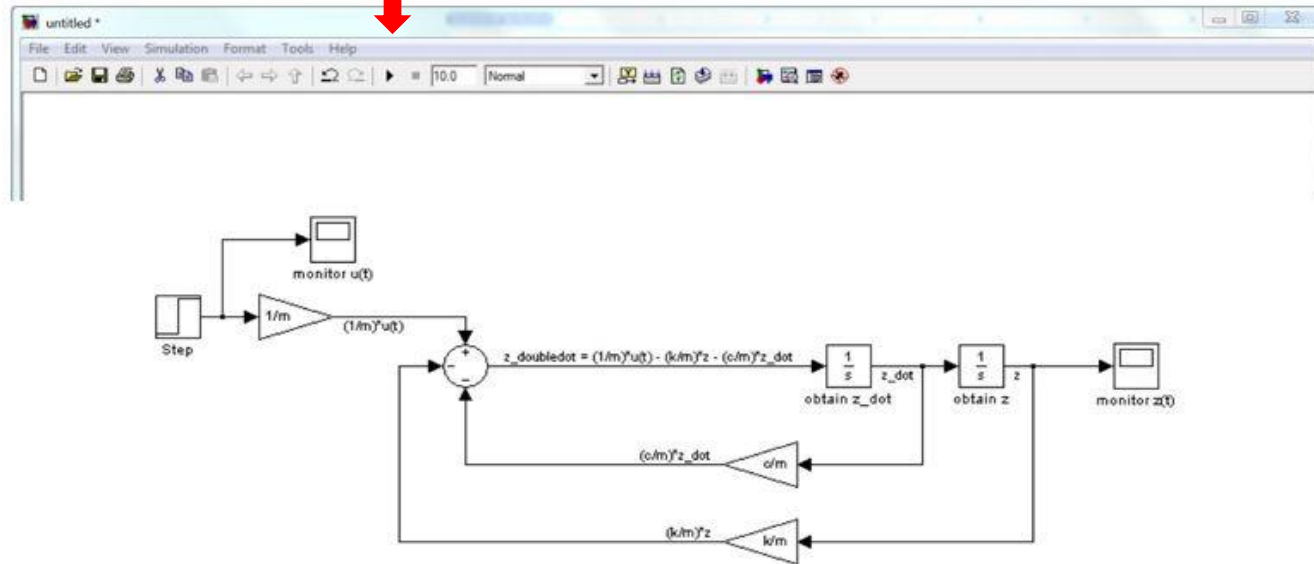


Continue ... add “Scope” block (in Sinks group) at the end of output

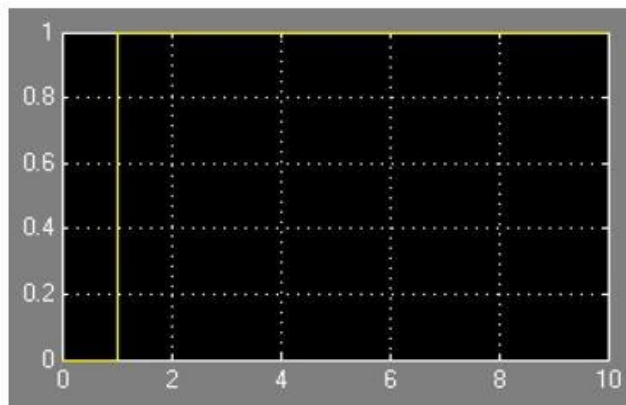


Continue ... Finally, obtain

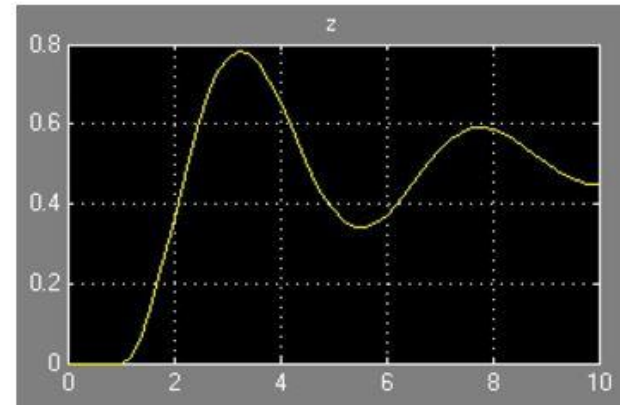
Run



Input

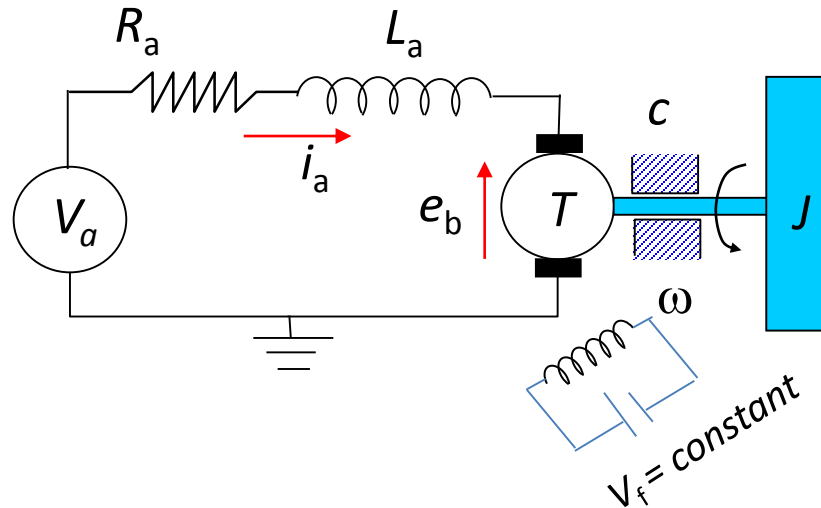


Output

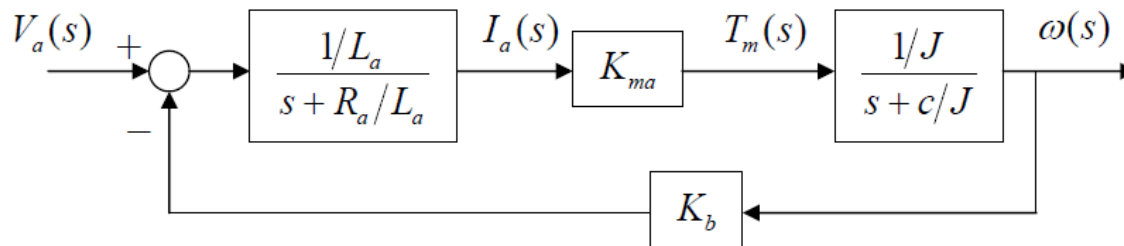


Exercise (30 minutes)

Consider the armature-controlled D.C. motor:



with block diagram:

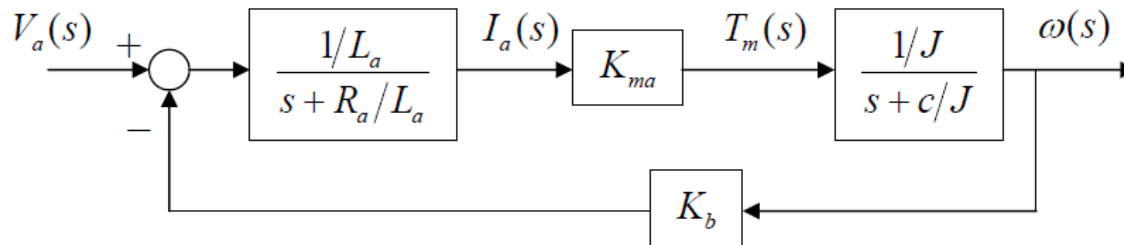


physical parameters:

$$\begin{aligned} J &= 8 \times 10^{-4} \text{ kg.m}^2 \\ c &= 3 \times 10^{-4} \text{ N.m/(rad/s)} \\ R_a &= 1.2 \Omega \\ L_a &= 0.020 \text{ H} \\ K_b &= 5 \times 10^{-2} \text{ V/(rad/s)} \\ K_{ma} &= 0.043 \text{ N.m/A} \end{aligned}$$

Block diagram:

Simplified parameters:



$$\begin{aligned} J &= 0.1 \\ c &= 0.5 \\ R_a &= 50.0 \\ L_a &= 2.0 \\ K_b &= 5.0 \\ K_{ma} &= 1.5 \end{aligned}$$

Use **Simulink** to

- draw this block diagram (with simplified parameters)
- simulate its unit step response

2.0 marks

END

For reference only

Nichol's Chart

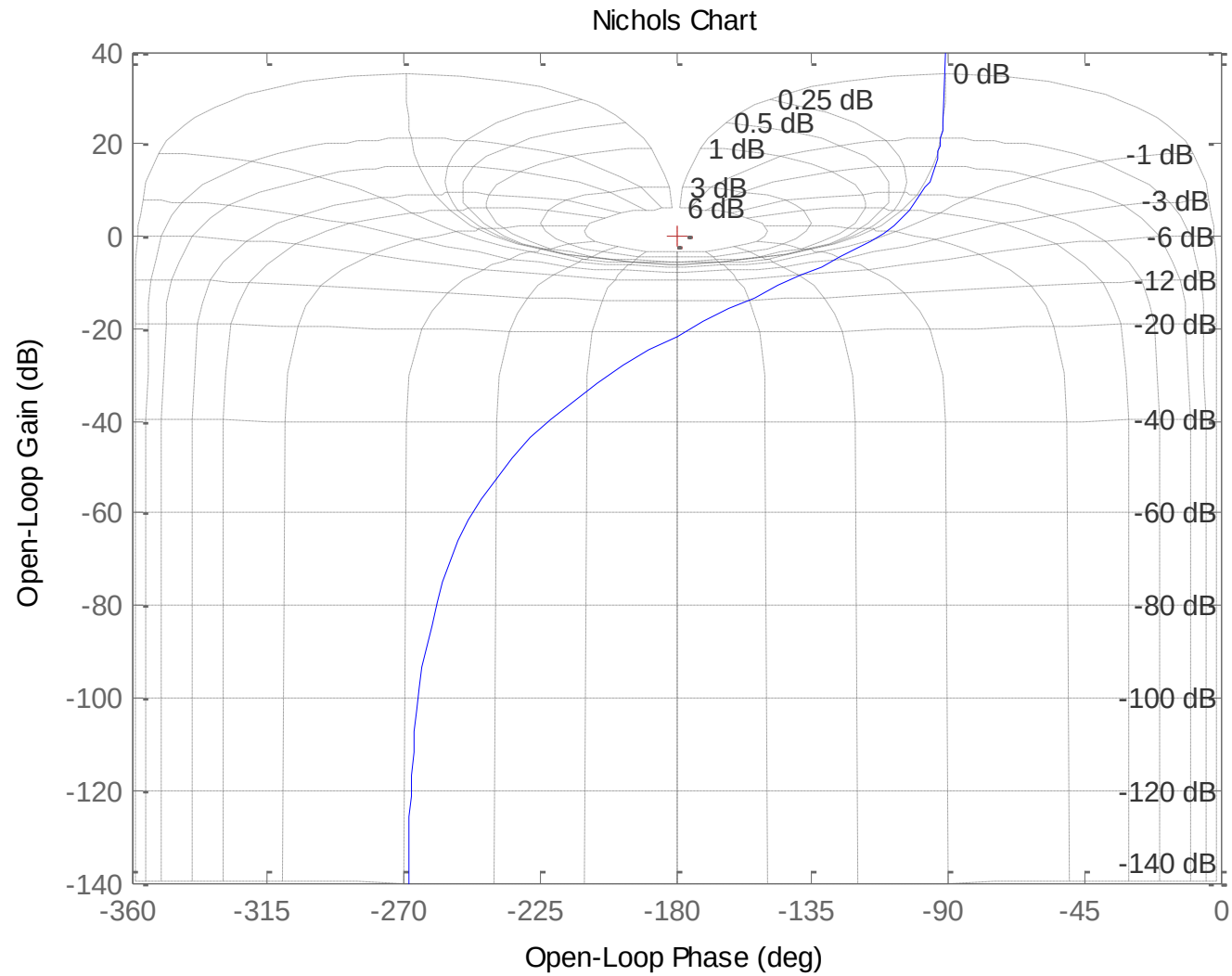
Example: For the system with the following open-loop transfer function, draw its Nichol's Chart and obtain the gain and phase margins.

$$G(s) = \frac{1}{s(s+1)(s+3)}$$

Matlab code:

```
s=tf('s');  
G=1/(s*(s+1)*(s+3));  
nichols(G);  
ngrid;
```

Result:



Step 1: Delete the last row of the matlab code, run the program again, right click on the created figure, and choose “Characteristics”.

Step 2: Select “All Stability Margins”, and put the mouse on the blue point of the figure to display the phase or gain margin.

