

LEVEL 5 ASSESSMENT SPECIFICATION

Student name:	Joe Potter	Student ID number:	
Programme:	HND/HNC Electrical/Electronic Engineering, HNC Instrumentation Engineering & HNC Power System Engineering		
Module:	Electrical Principles, Methods and Simulation		
Module code:	PSDE5009	Contribution to Overall Module Assessment (%):	50%
Lecturer:	Jason Pointer	Internal Verifier:	Steve Roberts
Assignment Title:	Control Systems	Word count (or equivalent):	2000
Submission deadline:	16/05/19	Return date of provisional marks & written feedback:	
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 40%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

Module Learning Outcomes (from module syllabus)

1. Analyse and evaluate electrical circuits, electronic systems and **control systems**.
2. Simulate standard electrical circuits, electronic systems and **control systems**.

STUDENT FEEDBACK FORM

This section details the *extent* to which the assessment criteria are demonstrated by you, which in turn determines your mark. The marks available for each category of skill are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills	0	0
2. Knowledge and Understanding Skills	30	18.3
Achieved 40% for Task 1. Achieved 72% for Task 2. Achieved 58% for Task 3. Achieved 70% for Task 4. Achieved 65% for Task 5. Knowledge is reasonably extensive and accurate in most parts. Show a very good understanding of most concepts and principles covered by this assignment but not all of the elements were covered for some of the task.		
3. Cognitive and Intellectual Skills	10	3.5
Largely descriptive work in the analysis of the test results.		
4. Practical Application Skills	50	34.1
Achieved 68% for Task 1. Achieved 70% for Task 2. Achieved 60% for Task 3. Achieved 75% for Task 4. Achieved 68% for Task 5. A very good application of established methods in the use of Matlab, Multisim and Simulink but in places not all of the required work was produced.		
5. Transferable Skills for Life and Professional Practice	10	4.8
Mostly ordered presentation and structure but in places does lack some coherence. Must state the requirements of all tasks.		

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)

Late Submission Penalties (tick if appropriate)

Up to 1 week late (40% Max)

Over 1 week late (0%)

61%

Student Name: Joseph Potter

Student Number: 121306

Lecturer: Jason Pointer

Assignment: Control Systems

Module Code: PDSE5009

Assignment 2 of 2

Contents Page

Task 1 _____ Page 3 -7

Task 2 _____ Page 8 - 10

Task 3 _____ Page 11 - 17

Task 4 _____ Page 18 - 19

Task 5 _____ Page 20 - 24

Task 1

a)

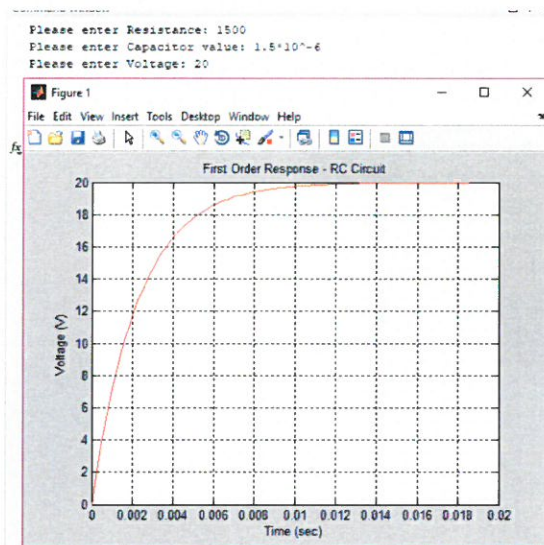
$$\frac{1}{s+a} e^{-at}$$

$$\frac{V_{out}(t)}{V_{in}(t)} = \frac{1}{CR} \times e^{-\frac{t}{CR}}$$

$$V_{out}(t) = V_{in}(t) \times \frac{1}{CR} e^{-\frac{t}{CR}}$$

You should state the requirement of the task.

You are missing the beginning for deriving the formula.



(fig.2)

The code written in matlab is:

```
% First Order Response of an RC circuit

% Values for the Resistor, capacitor and voltage
R = input('Please enter Resistance: ');
C = input('Please enter Capacitor value: ');
V = input('Please enter Voltage: ');

Tc=R*C;      % Calculate the Time Constant
t=0;         % Set the Initial Conditions
Vout=0;

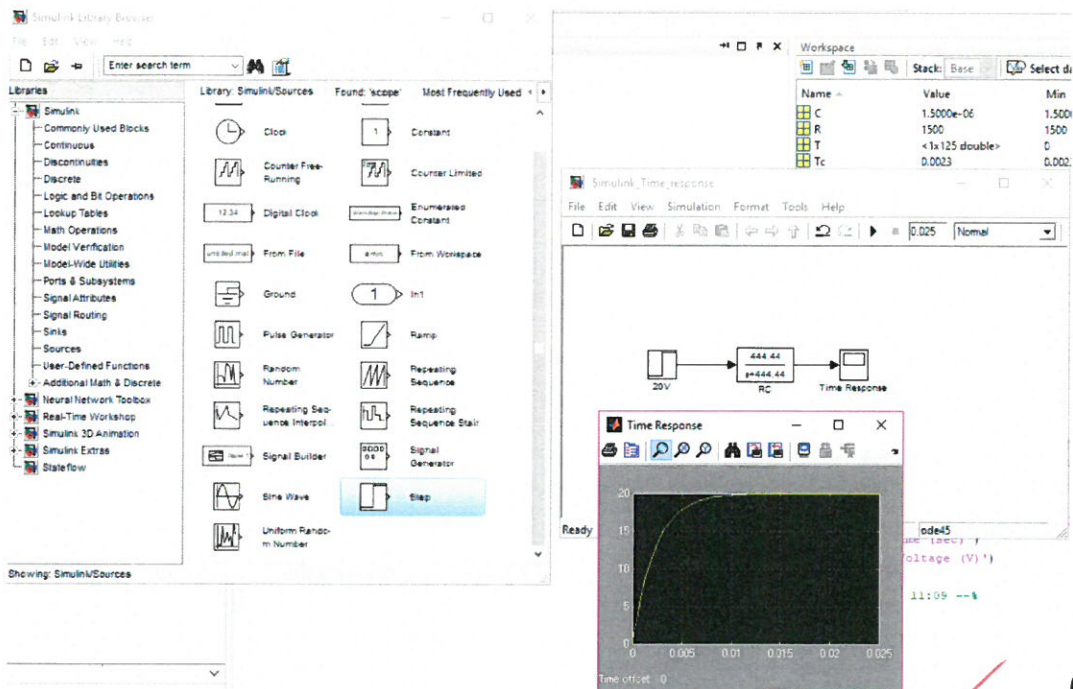
% Determine the output voltage for value of time, t
for k=1:125
    t(k)=(k-1)*Tc/15;
    Vout(k)=V*(1-exp(-t(k)/Tc));
end;
```

```
% Display the Vout against time
plot(t,Vout,'r-');
title('First Order Response - RC Circuit')
xlabel('Time (sec)')
ylabel('Voltage (V)')
grid
```

b)

Simulink simulation

Have not included delay and rise time.



(fig.3)

Simulink

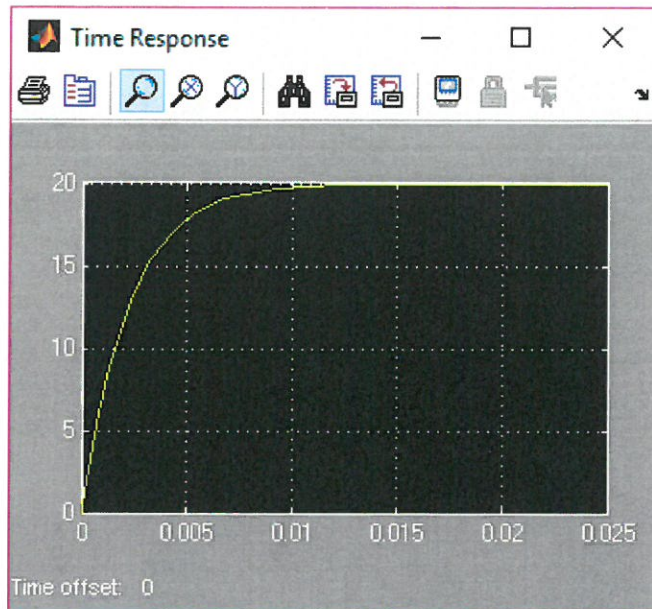
$$\frac{V_{out}(s)}{V_{in}(s)} =$$

$$\frac{1/1.5 \times 10^3 \times 1.5 \times 10^{-6}}{s + 1/1.5 \times 10^3 \times 1.5 \times 10^{-6}}$$

$$= \frac{1/2.25 \times 10^{-3}}{s + 1/2.25 \times 10^{-3}}$$

$$= \frac{444.44}{s + 444.44}$$

$$= \frac{444.44}{s + 444.44}$$



The plot above shows that that the MATLAB simulation was correct as they are both similar.

c)

```
% Joe Potter Task 1c
% Frequency response for a RC circuit with a 1V sine wave

% The component values
% Values for the Resistor, capacitor and frequency
R = input('Please enter Resistance: ');
C = input('Please enter Capacitor value: ');
f = 0;

% Time constant
Tc=R*C;

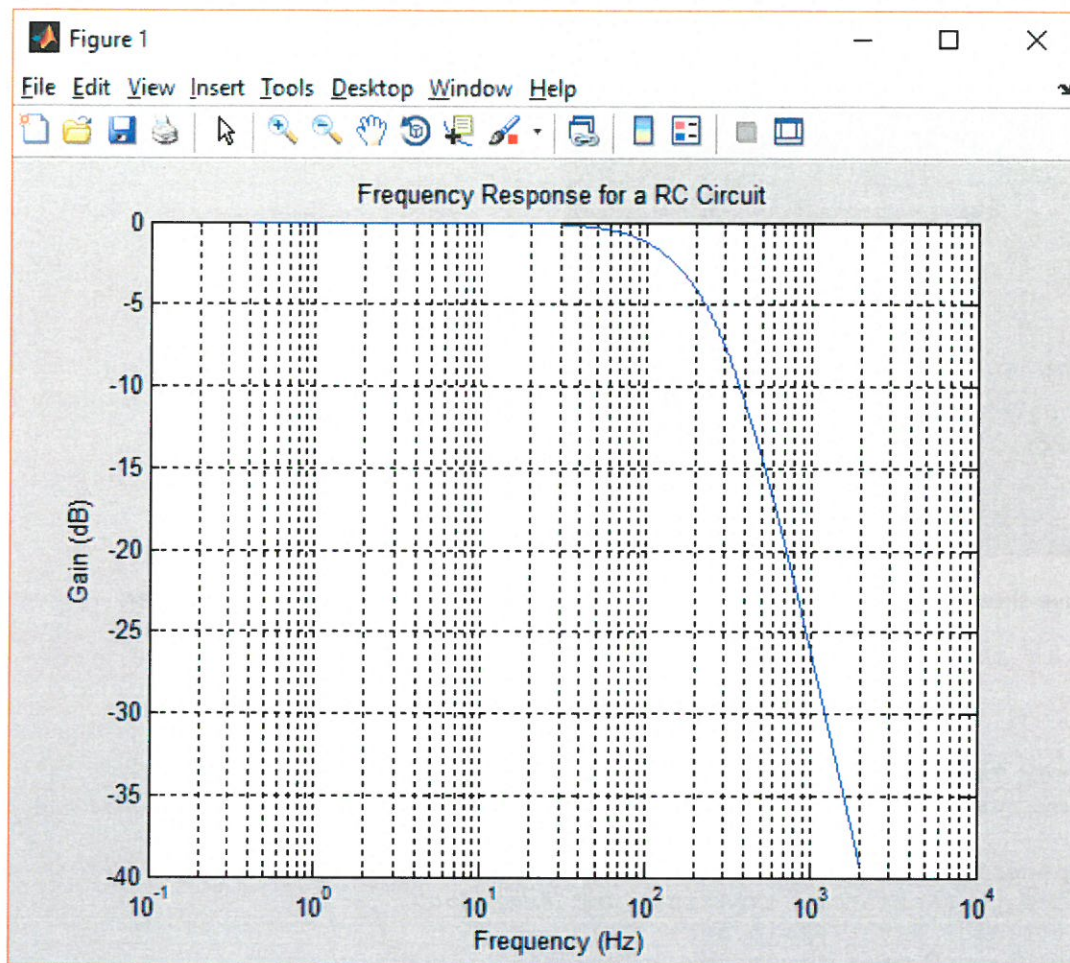
fx=0:0.4:2000;
final=length(fx);

% For loop
for x=1:final
    f=f+0.1;

    % Angular velocity
    w(x)=2*pi*f;

    % Time response
    G(x)=20*log(1/(sqrt(1+w(x)^2*Tc^2)));
end

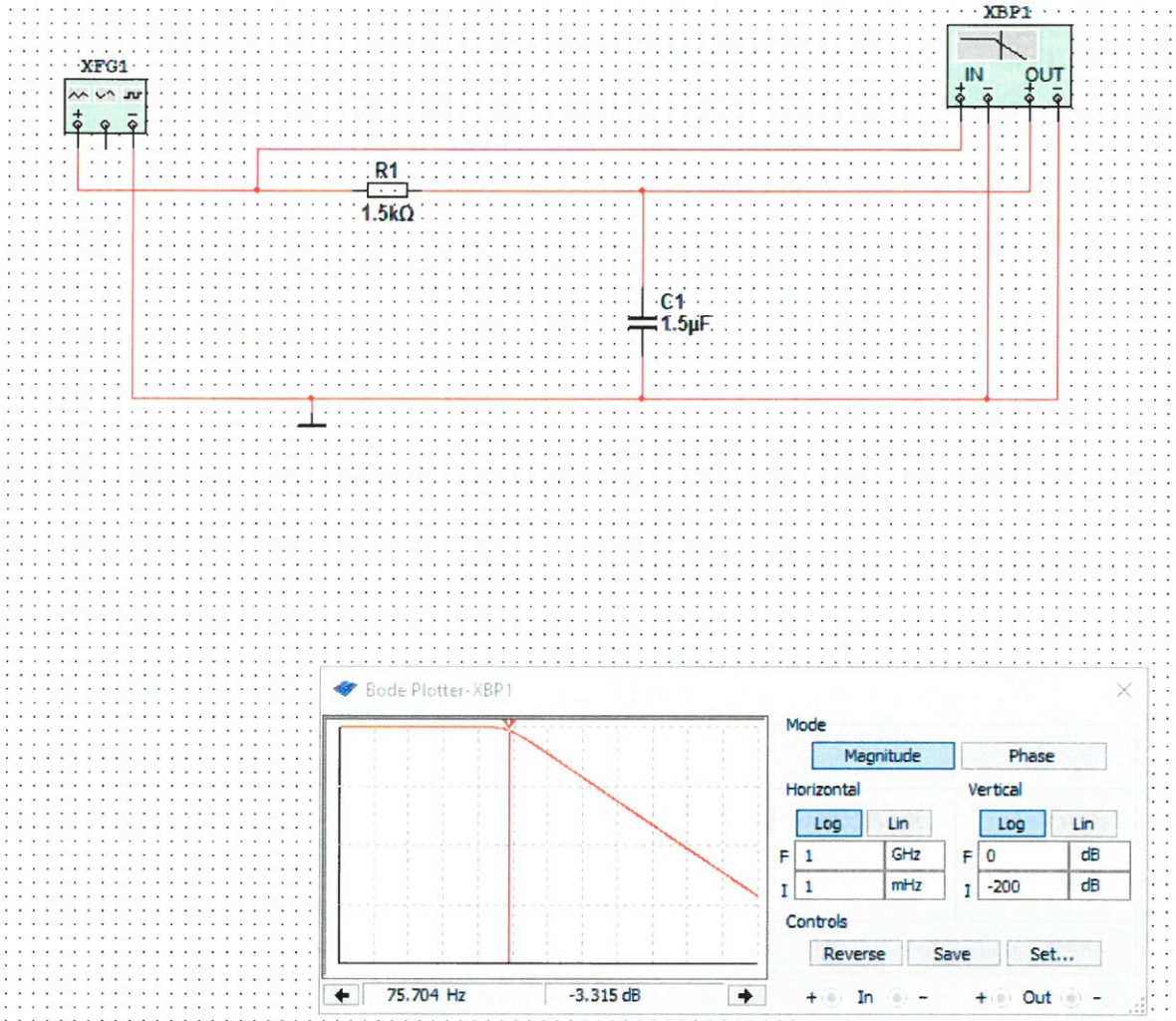
% The frequency Response for first order
semilogx(fx,G)
xlabel('Frequency (Hz)')
ylabel('Gain (dB)')
title('Frequency Response for a RC Circuit')
grid
```



Cut off frequency = 110Hz

d)

Multisim simulation



Cut off frequency = 75.704Hz.

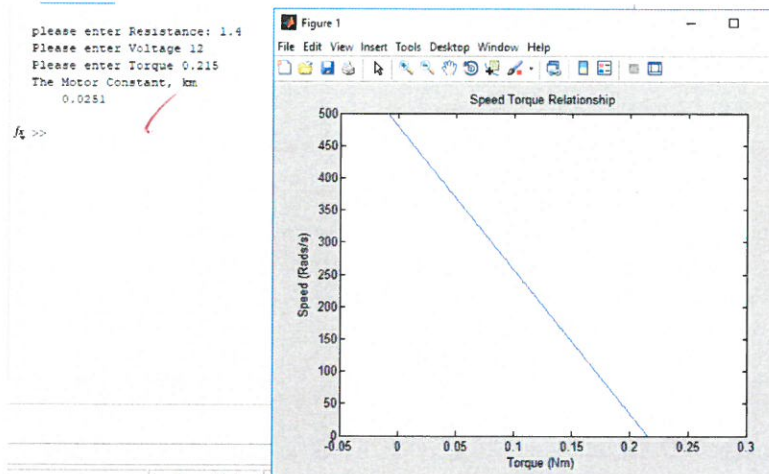
Between the two frequency responses, they both show different answers, this could be because of the different software being used.

Could have included the phase plot.

You are missing the Matlab program for the Bode plot.

Task 2

a)



% DC Motor

Ra = input('please enter Resistance: '); % Motor Resistance

Va = input('Please enter Voltage '); % Motor voltage

Ts = input('Please enter Torque '); % Motor torque

km=(Ts*Ra)/Va; % Calculation of Motor constant

% Max Speed

W0=Va/km;

% Speed

W=0:2:500;

T=km*(Va-km*W)/Ra;

% Results

disp('The Motor Constant, km')

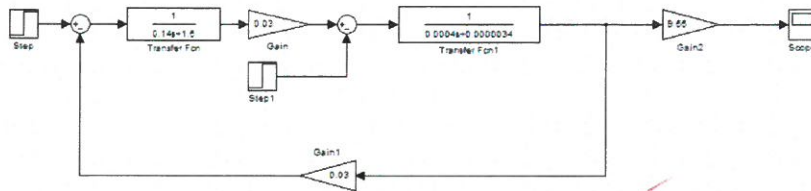
disp(km)

plot(T,W,'B-')

xlabel('Torque (Nm)')

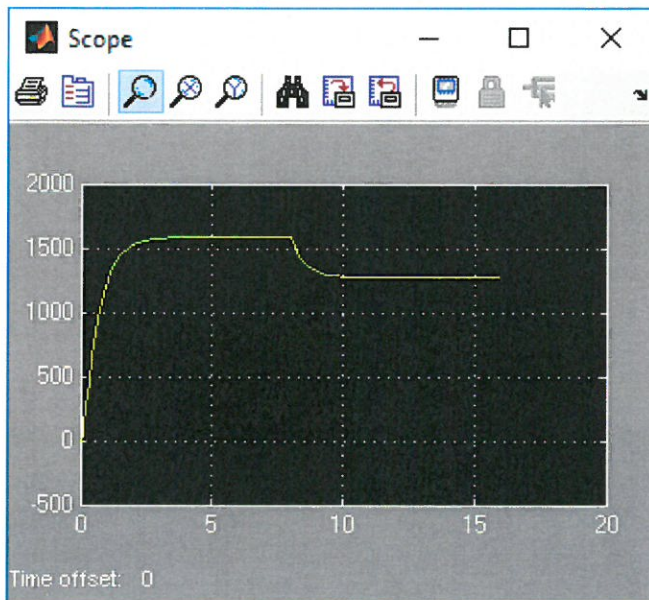
ylabel('Speed (Rads/s)')

title('Speed Torque Relationship')



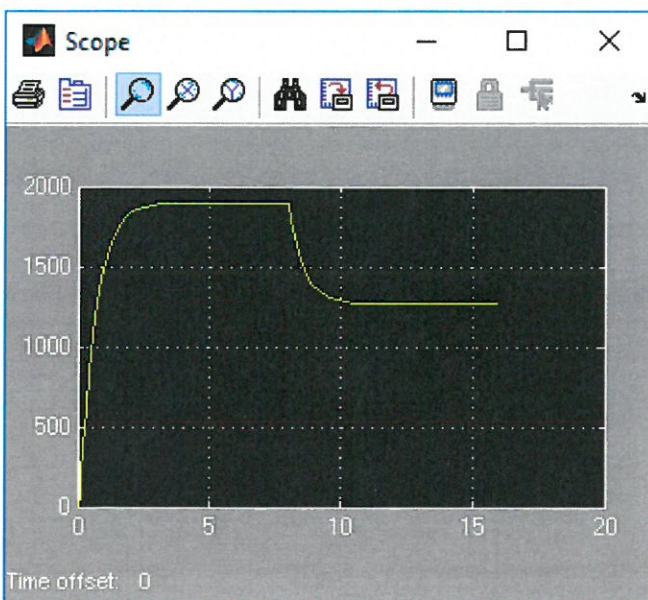
a bit small and should have labeled the blocks.

b and c



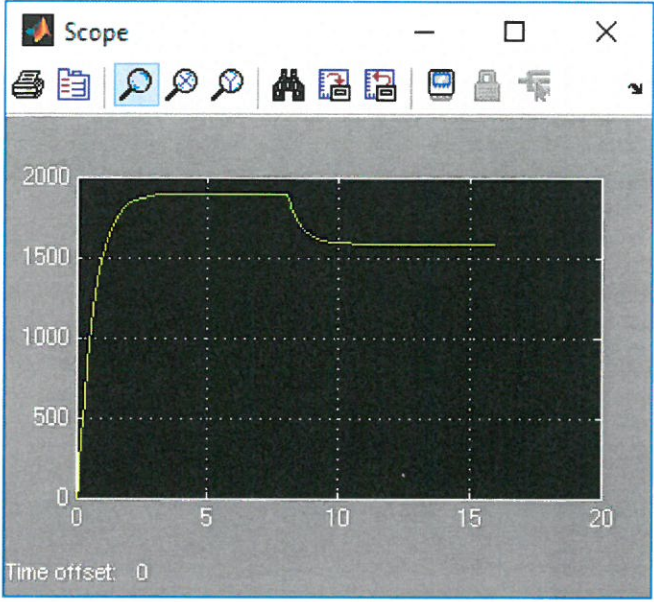
D

At 0.04



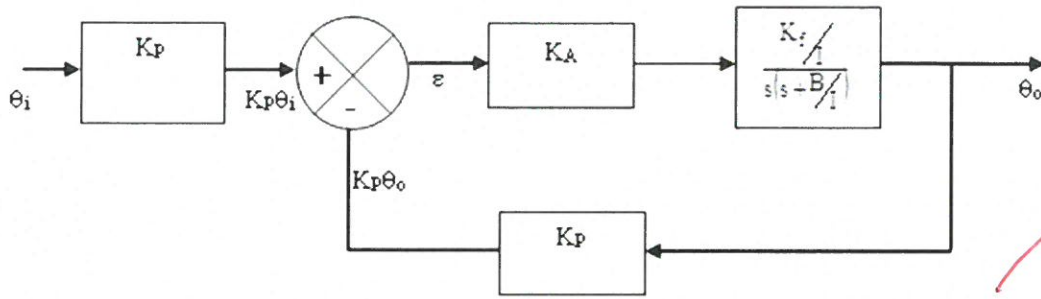
Steady state speed = 1250rpm

At 0.02



Steady state speed = 1550rpm

Task 3



a)

Standard equation for negative feedback system. $\frac{\theta_o}{\theta_i} = \frac{G(s)}{1 + G(s)H(s)}$

$$G(s) = K_A \times \frac{K_f/I}{s(s+B/I)}$$

$$H(s) = K_P$$

So

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_P \times \frac{K_A K_F/I}{s(s+B/I)}}{1 + K_P \times \frac{K_A K_F/I}{s(s+B/I)}}$$

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\frac{(s^2 + Bs/I) \times K_P K_A K_F/I}{s^2 + Bs/I}}{(s^2 + Bs/I) \times 1 + \frac{(s^2 + Bs/I) \times K_P K_A K_F/I}{s^2 + Bs/I}}$$

$$= \frac{\frac{K_P K_A K_F}{I}}{s^2 + \frac{Bs}{I} + \frac{K_P K_A K_F}{I}}$$

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\frac{K_P K_A K_F}{I}}{s^2 + \frac{Bs}{I} + \frac{K_P K_A K_F}{I}}$$

c)

$$\frac{\theta_o(s)}{\theta_i(s)} = k_{ss} \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\frac{5 \times 1 \times 8 \times 10^{-3}}{2 \times 10^{-4}}}{s^2 + \frac{Bs}{2 \times 10^{-4}} + \frac{5 \times 1 \times 8 \times 10^{-3}}{2 \times 10^{-4}}} = \frac{200}{s^2 + 5000Bs + 200}$$

$$\omega_n^2 = 200$$

$$\text{Natural frequency} = \omega_n = \sqrt{200} = 14.14 \text{ rad/sec}$$

$$2\zeta\omega_n = 5000B$$

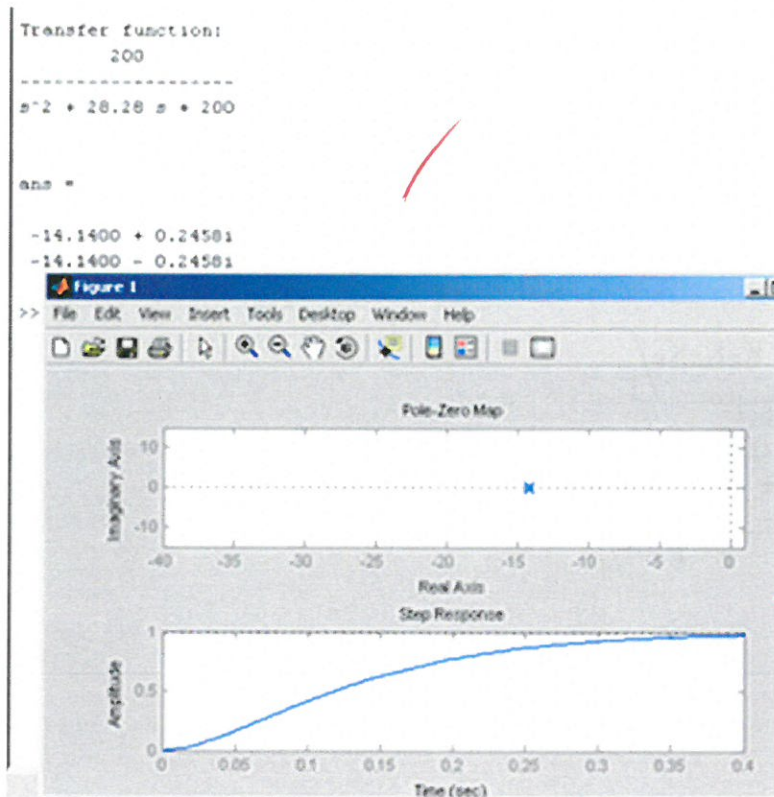
for critically damping $\zeta = 1$

$$B = \frac{28.28}{5000} = 5.657 \times 10^{-3} \text{ N/ms}^{-1}$$

To determine the poles for critically damped

$$(s^2 + 2\omega_n s + \omega_n^2) = 0$$

$$s = -\omega_n, -\omega_n = -14.14$$



```
%Joe Potter
%Numerator
num=[200];

% Denominator
den=[1 28.28 200];

% Transfer Function
```

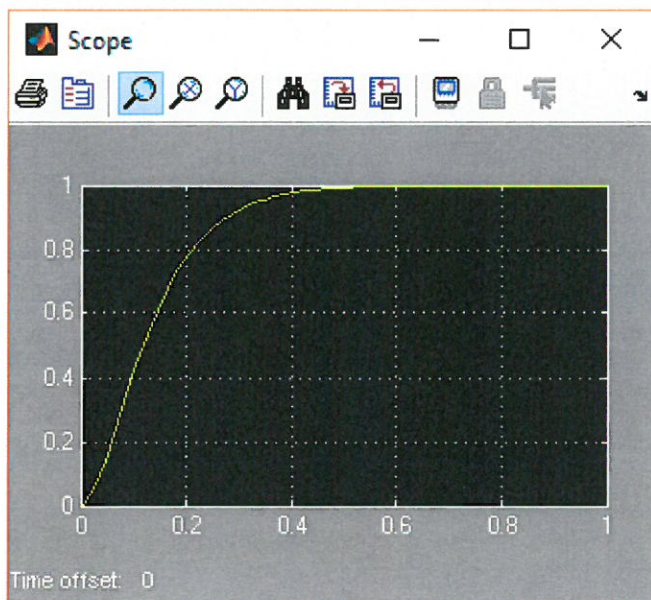
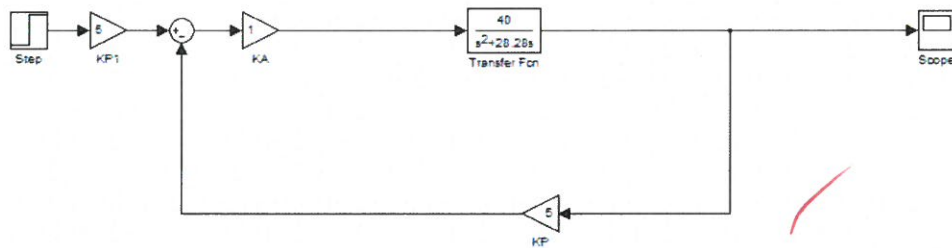
```
sys=tf(num,den)
```

```
% Poles
pole(sys)
```

```
% Plotting the Poles
subplot(211), pzmap(sys)
axis([-40 1 -15 15])
```

```
% Plotting the response for an unit step
subplot(212), step(sys)
```

d)



Looking at the graph above, it is the same as the MATLAB response.

e)

```
% Joe Potter
% Friction Value
B = input('Please ent value of friction, B:');
BI=B/(2*10^-4);
```

```
% Numerator
num= [200];
```

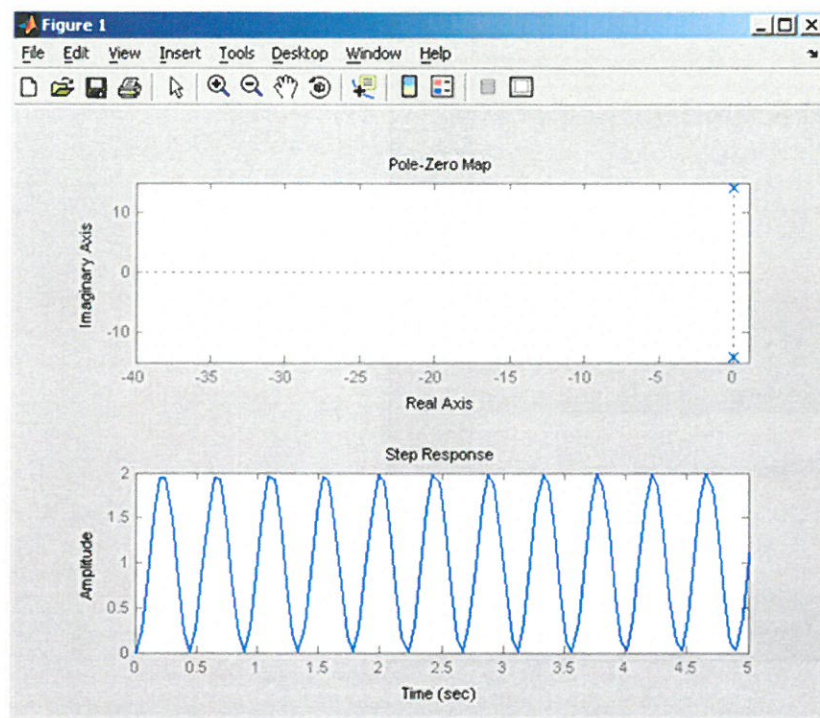
```
% Denominator
den=[1 BI 200];

% Transfer Function
sys=tf(num,den)

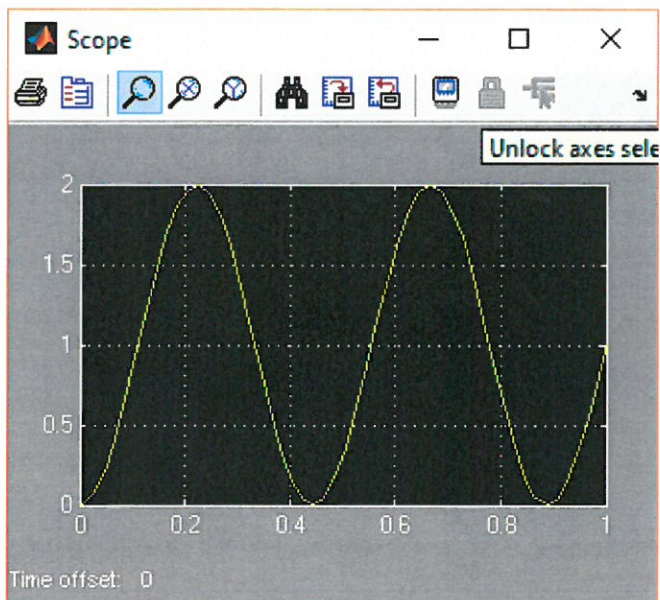
% Poles
pole(sys)

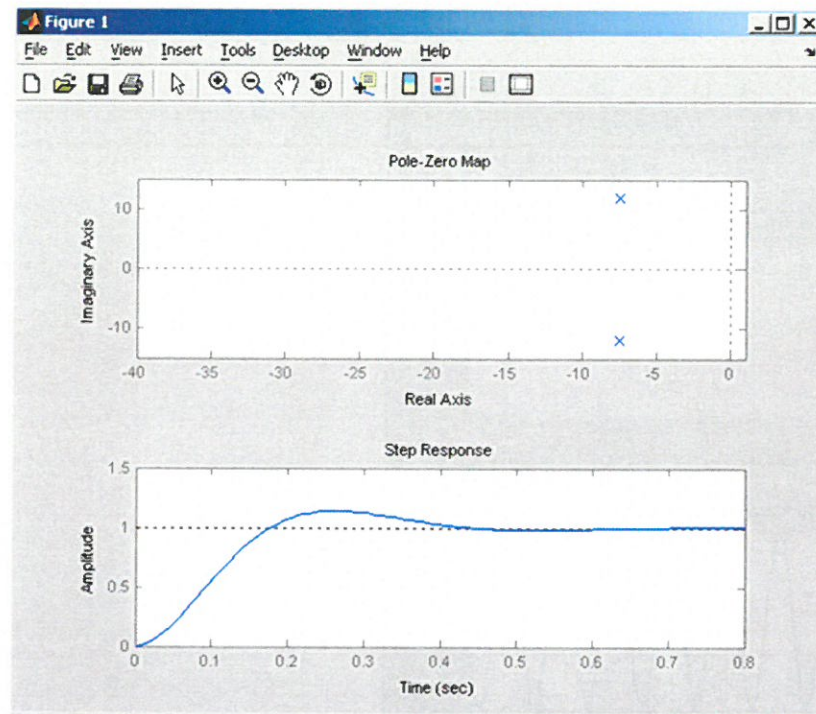
% Plotting the Poles
subplot(211), pzmap(sys)
    axis([-40 1 -15 15])

% Plotting the response for an unit step
subplot(212), step(sys)
```

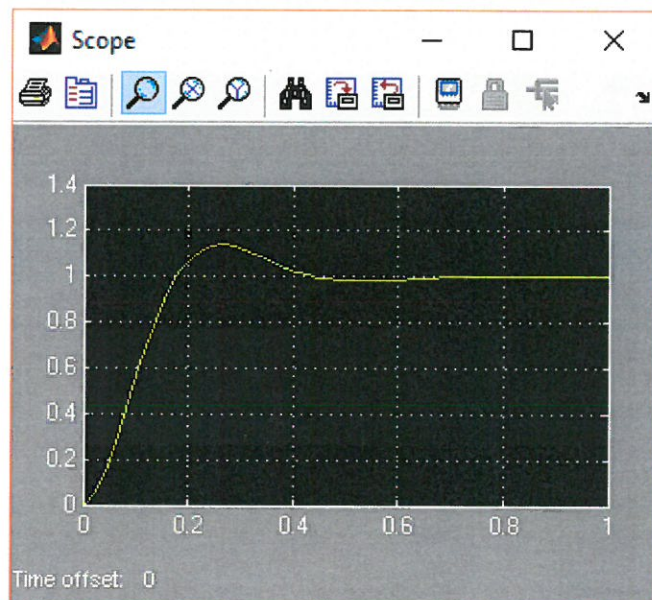


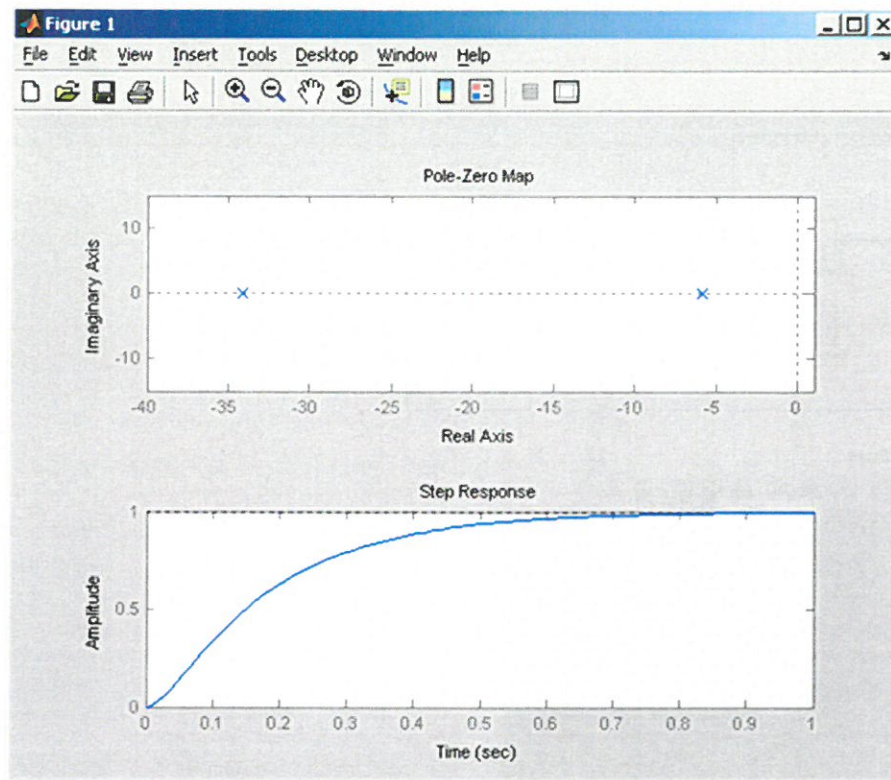
No damping $B = 0$



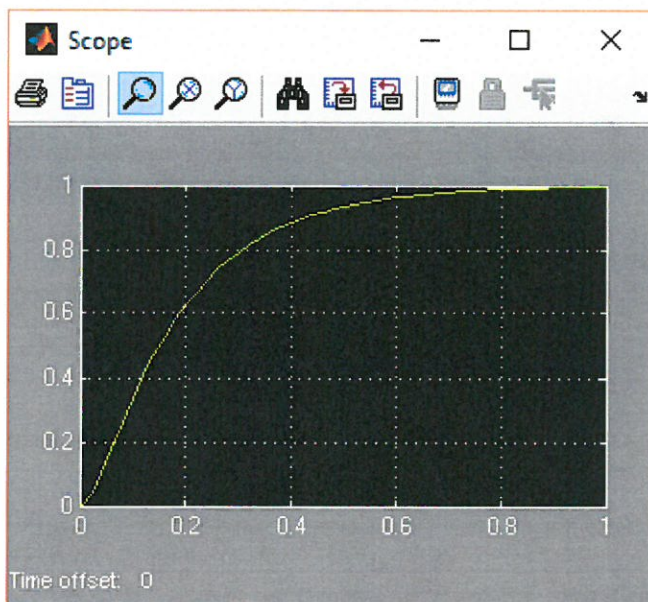


Under damped $B = \frac{3 \times 10^{-3}}{2 \times 10^{-4}} = 15$



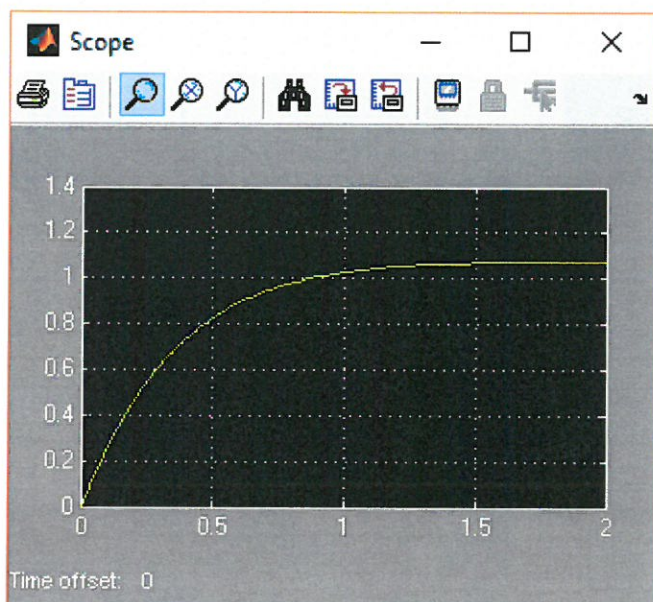
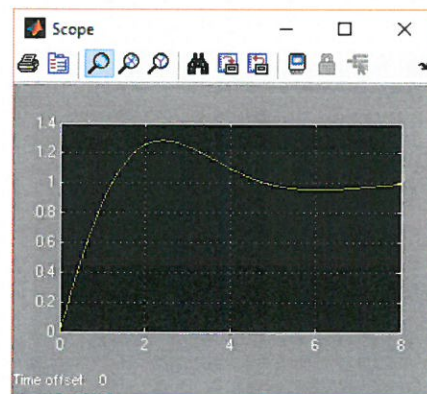
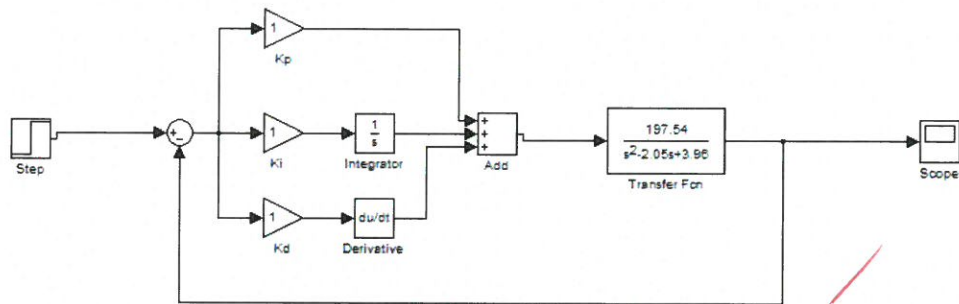


Over damped $B = \frac{8 \times 10^{-3}}{2 \times 10^{-4}} = 40$



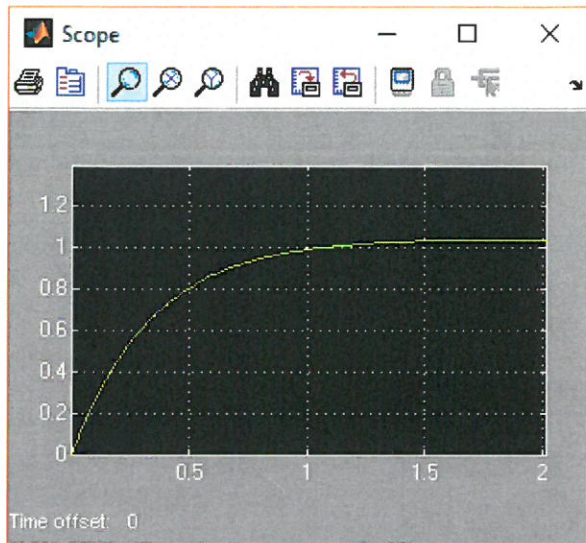
You are missing part f) for this task.

Task 4

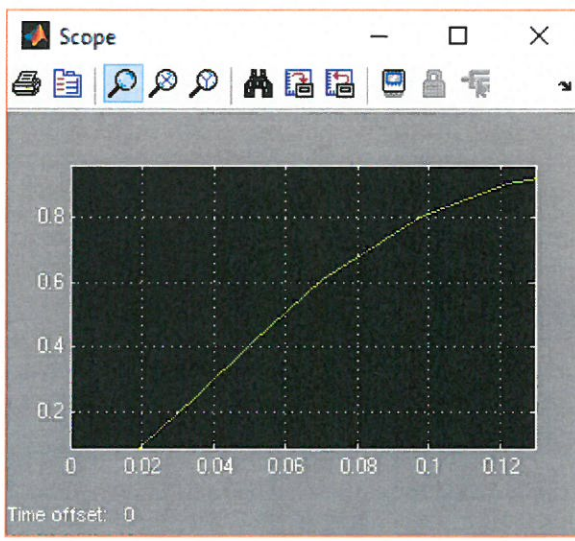


Kp = 3

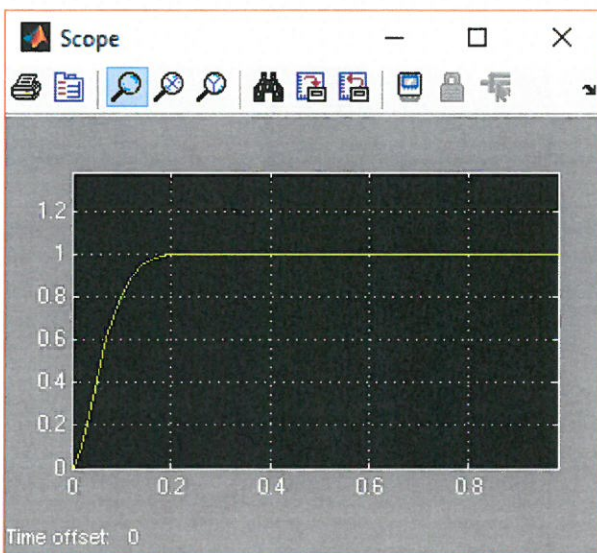
The circuit still has overshoot, however, the circuit does run faster.



Changing the integrator value, still shows overshoot but reduced.



Rise time is now shown less than 0.18seconds. As well as showing zero overshoot



Task 5

$$G_C(s) = \frac{0.03868s^3 - 5.369s^2 - 97.34s + 13450}{s^3 + 169.8s^2 + 1730s + 159 + 0.03868s^3 - 5.369s^2 - 97.34s + 13450}$$

$$= \frac{0.03868s^3 - 5.369s^2 - 97.34s + 13450}{1.03868s^3 + 162.431s^2 + 1632.66s + 13609}$$

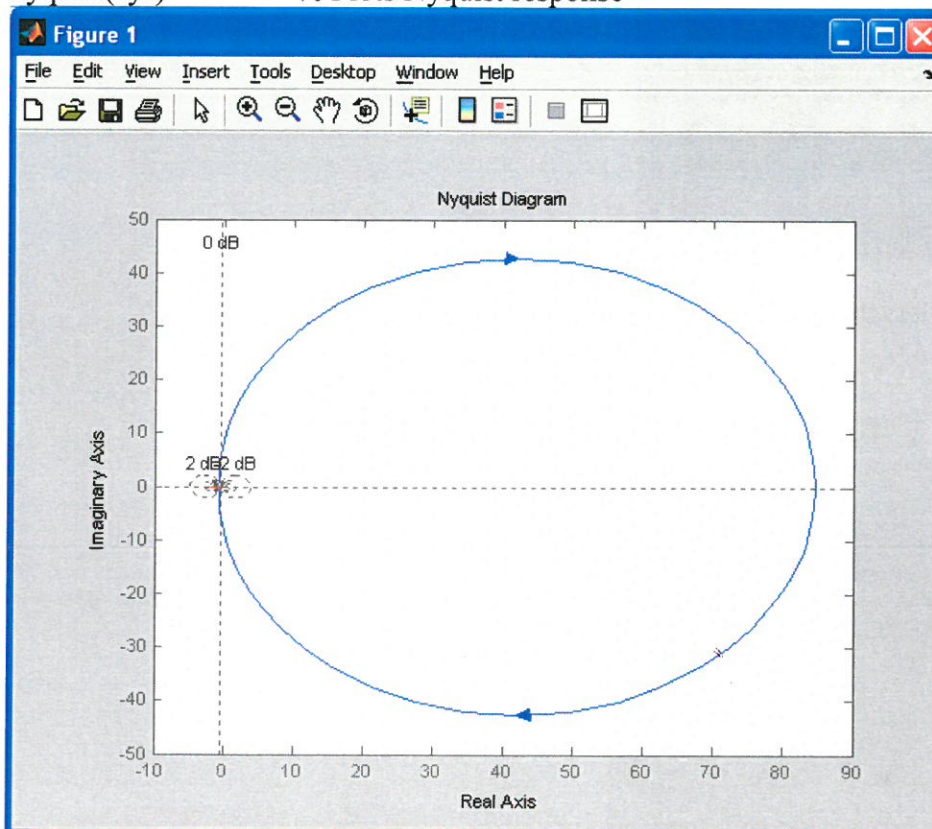
Task 5 (a)

```
num=[0.039 -5 -96.5 12850];
```

```
den=[1 168.5 1680 140];
```

```
sys=tf([num],[den]) % Transfer function
```

```
nyquist(sys) % Plots Nyquist response
```



Task 5 b and c

%b and c

```
num=[0.03868 -5.369 -97.34 13450];  
den=[1.03868 162.431 1632.66 13609];
```

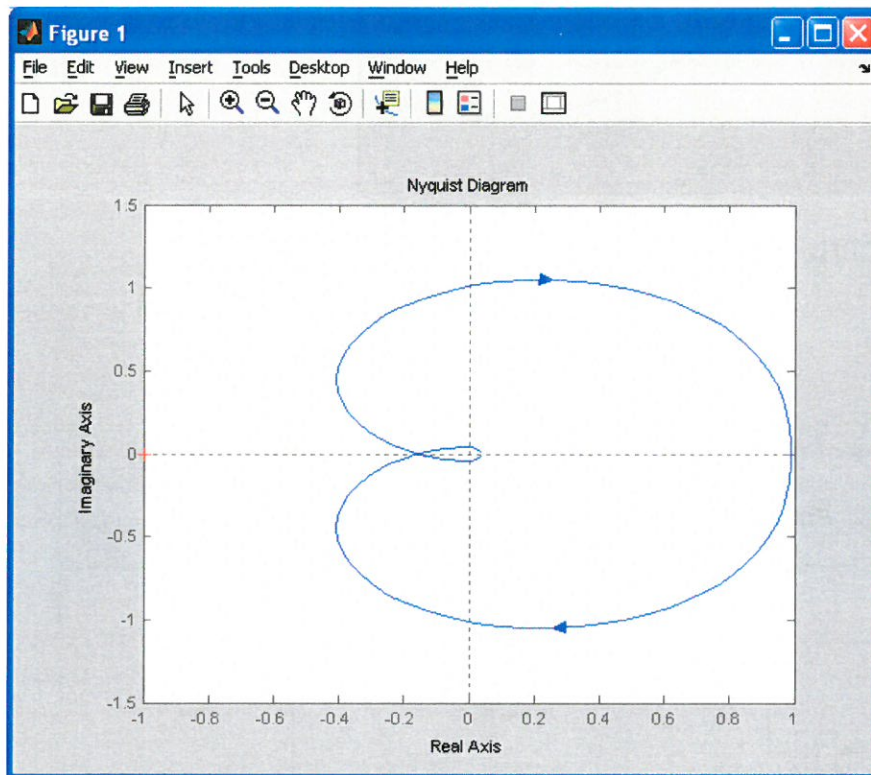
```
sys=tf([num],[den]) % Transfer function
```

```
nyquist(sys) % Plots Nyquist response
```

```
pause
```

```
bode(sys)
```

```
grid
```

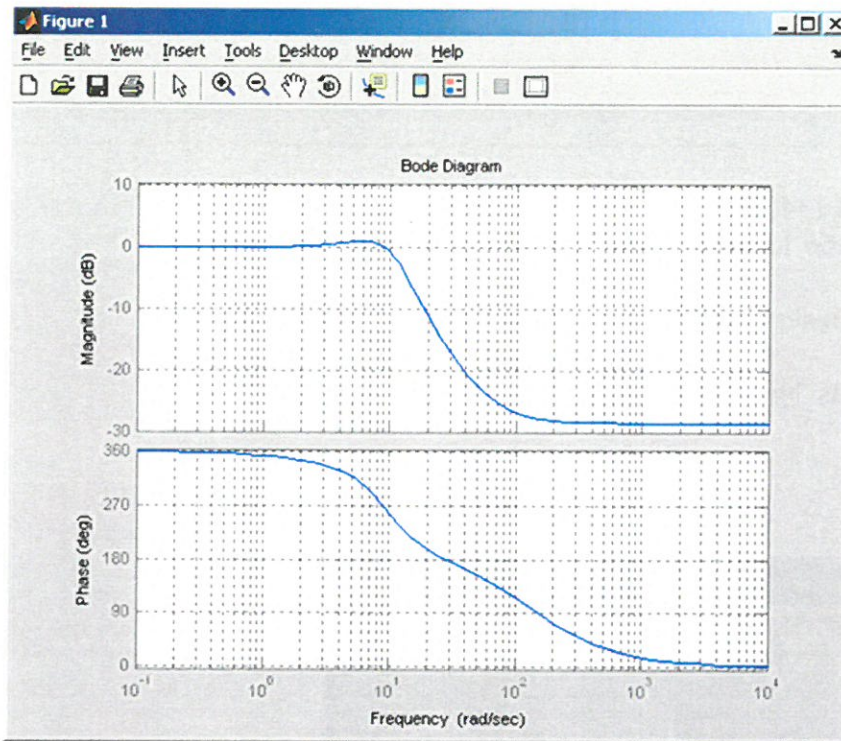


The system is stable, because the response is in the negative 'real axis'.

Phase margin is infinity.

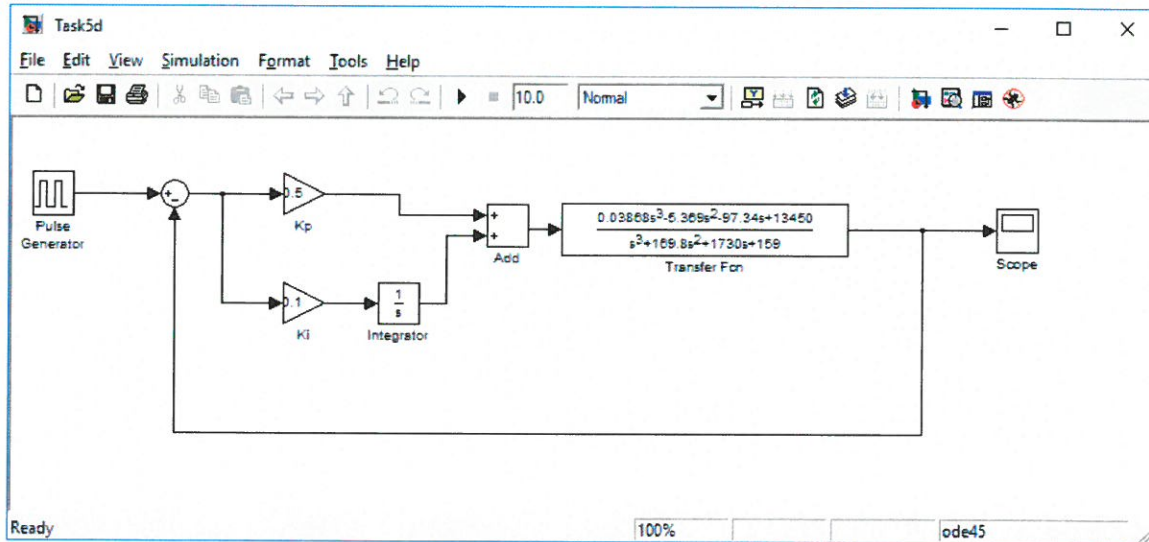
Gain margin is equal to $20\log(1/0.18) = 14.89\text{dB}$.

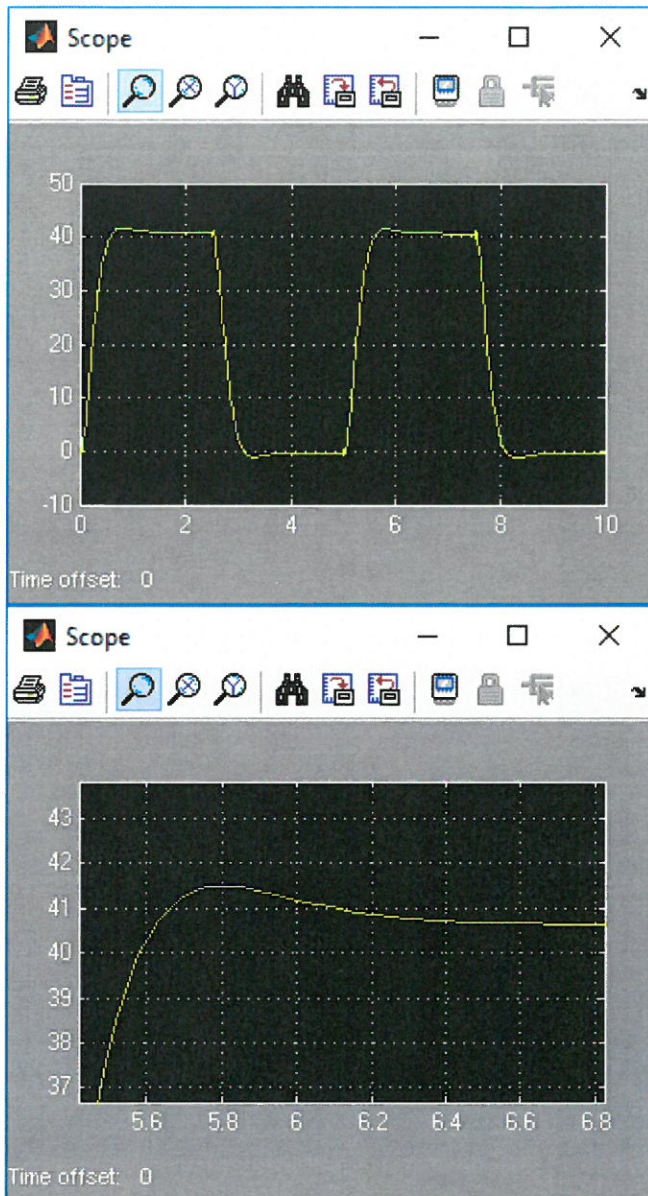
Good.



Peak resonance $M_p = 7\text{Hz}$
 Bandwidth $\omega_b = 30 - 10 = 20\text{Hz}$

5D





Change K_i to 0.03

You have not describe the tuning method.

