

LEVEL 5 ASSESSMENT SPECIFICATION

Student name:	Tom Wolsey	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	21.7
Achieved 78% for Task 1. Achieved 80% for Task 2. Achieved 68% for Task 3. Achieved 70% for Task 4. Achieved 65% for Task 5. Excellent, detailed knowledge and showed an excellent understanding of the well-established concepts and principles of the covered by this assignment.		
3. Cognitive and Intellectual Skills	10	4.5
Largely descriptive work in the analysis of the test results.		
4. Practical Application Skills	50	38.5
Achieved 77% for Task 1. Achieved 80% for Task 2. Achieved 74% for Task 3. Achieved 74% for Task 4. Achieved 80% for Task 5. Excellent application of established methods in the use of Matlab, Multisim and Simulink. Required minimum support throughout the assessment.		
5. Transferable Skills for Life and Professional Practice	10	5.5
Mostly coherent, organised and accurate work, in a suitable structure and is for the most part clearly expressed.		

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%)

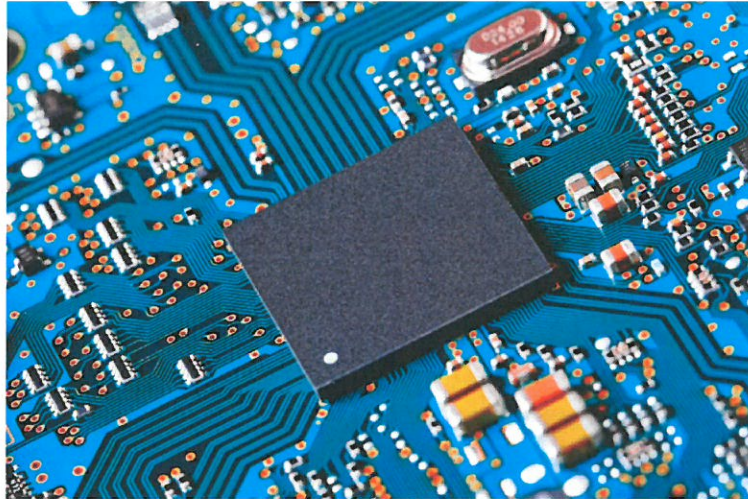
70%

Tom Wolsey 126039

PSDE 5009 – Electrical Principles, Methods and Simulation

Assignment 2 – Control Systems

Jason Pointer



Contents

Task 1 – RC Circuit First Order System	3
a).....	3
b)	4
c).....	7
d)	7
Task 2 - DC Motor Model	10
a).....	10
b)	11
c).....	12
d)	13
Task 3 - Position Control Servo.....	14
a).....	14
b)	14
c).....	15
d)	17
e).....	18
f)	22
Task 4.....	24
Task 5.....	28
a).....	28
b) /c.....	28
d)	31

Task 1 – RC Circuit First Order System

a)

Below the transfer function is determined

Should have stated the requirement of the task.

Kirchhoff law states

$$Ri(t) + \frac{1}{C} \int i(t) dt = V_{in}(t)$$

Taking the first derivative on both sides above gives

$$R \frac{di(t)}{dt} + \frac{1}{C} i(t) = \frac{dV_{in}(t)}{dt}$$

Using Laplace transform, the series electric RC circuit can be modelled by

$$sRI(s) + \frac{I(s)}{C} = sV_{in}(s)$$

$$I(s)[sR + 1/C] = sV_{in}(s)$$

$$I(s)[R + 1/Cs] = V_{in}(s)$$

$$V_{in}(s) = I(s)[R + 1/Cs]$$

eq (1)

Also

$$V_{out} = \frac{1}{C} \int i(t) dt$$

$$\frac{dV_{out}}{dt} = \frac{i(t)}{C}$$

Using Laplace transform

$$sV_{out}(s) = \frac{I(s)}{C}$$

$$I(s) = sV_{out}(s) \times C$$

Sub into eq (1)

$$V_{in}(s) = sV_{out}(s) \times C[R + 1/Cs]$$

$$\frac{V_{in}(s)}{V_{out}(s)} = sC[R + 1/Cs] = CRs + 1$$

So

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{CRs + 1}$$

Divide top and bottom by CR

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1/CR}{s + 1/CR}$$

To convert into a function of time take inverse Laplace transforms:

$$(v_{\text{out}}(t))/(v_{\text{in}}(t)) = 1/CR \times e^{(-t/CR)}$$

$$v_{\text{out}}(t) = v_{\text{in}}(t) \times 1/CR \times e^{(-t/CR)}$$

You might want to use equation editor next time.

b)

Below the time response plot can be seen along with the time constant, delay time and rise time

```

1. %File Name - Assignment_2_task_1_part_b.m
2. %Task 1 - B
3. %Author - Tom Wolsey
4. %Date 07/03/2019
5. %First order response of a RC circuit.
6.
7. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8. %Set value for Resistor, capacitor and voltage %
9. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10. R=input('Enter the value for R: '); %Program waits for user to input value
11. C=input('Enter the value for C: '); %Program waits for user to input value
12. V=input('Enter the value for V: '); %Program waits for user to input value
13.
14. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15. % Calculate time constant %
16. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17. Tc=R*C;
18.
19. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20. % Delay Time %
21. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22. dt=-Tc*log(0.5);
23.
24. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25. % Rise Time %
26. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27. rt=Tc*(log(0.9)-log(0.1));
28.
29. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30. % Set initial conditions %
31. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32. t=0;
33. Vout=0;
34. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35. % Determine the output voltage for value of time, t. %
36. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
37. for P=1:50
38.     t(P)=(P-1)*Tc/5;
39.     Vout(P)=V*(1-exp(-t(P)/Tc));
40. end;
41. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42. % Display the Vout against time %
43. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44. plot(t,Vout,'r-');
45. title('First Order Response - RC Circuit')
46. xlabel('Time (sec)')
47. ylabel('Voltage (V)')
48. grid
49.

```

```

50. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51. % Displays Time Constant %
52. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
53. disp(' Time Constant = ');
54. disp(Tc);
55.
56. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
57. % Displays Delay Time %
58. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59. disp(' Delay Time = ');
60. disp(dt);
61.
62. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
63. % Displays Rise Time %
64. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65. disp(' Rise Time = ');
66. disp(rt);

```

Very good program.

```

Enter the value for R: 1500
Enter the value for C: 0.0000015
Enter the value for V: 20
Time Constant =
    0.0023

```

```

Delay Time =
    0.0016

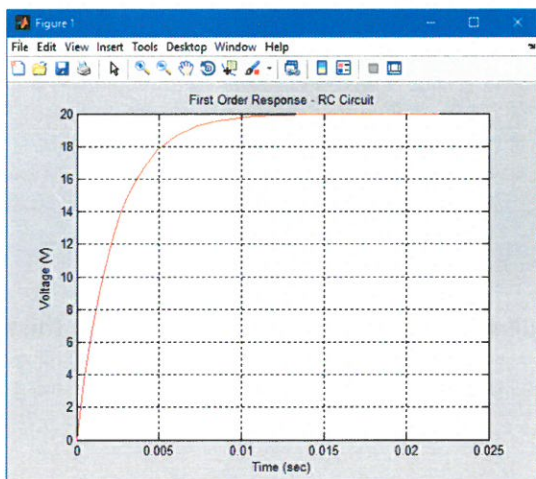
```

```

Rise Time =
    0.0049

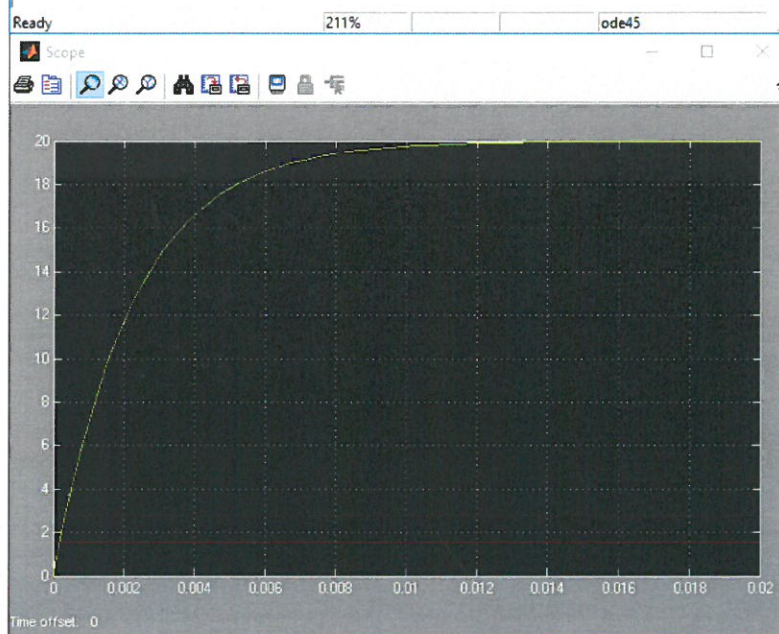
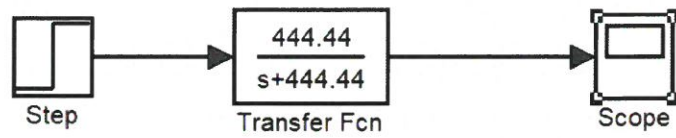
```

*Could have modify
your "disp" to include
units.*



The transfer function

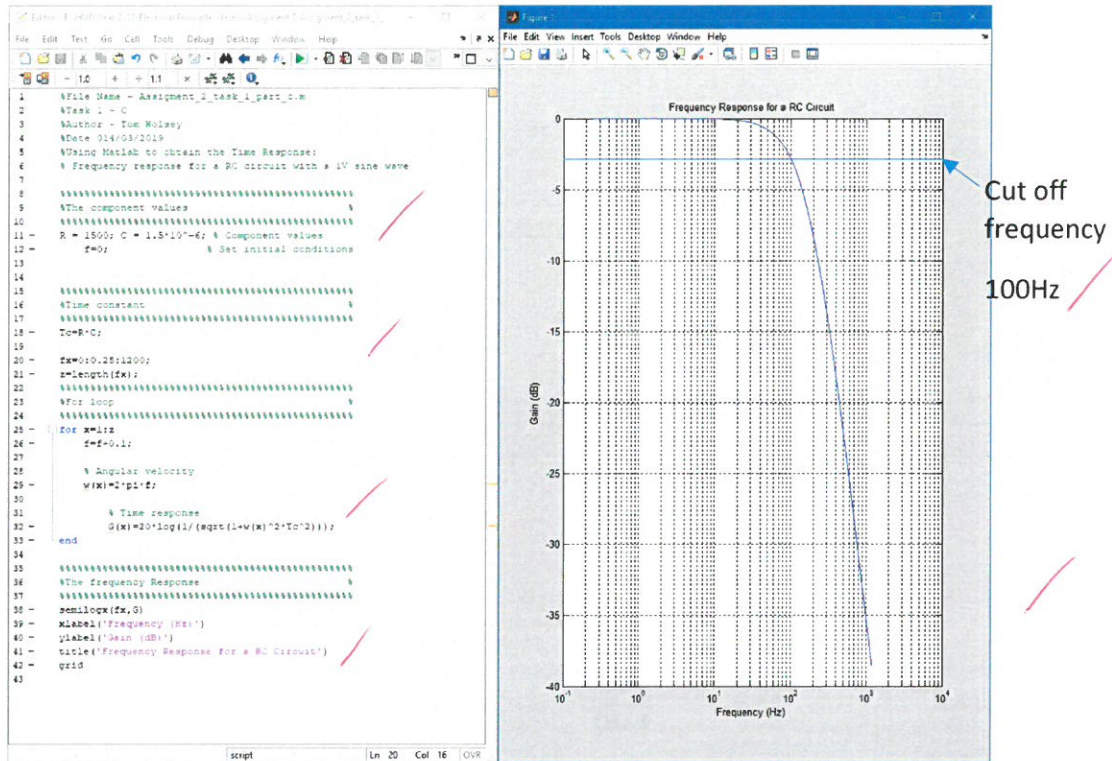
$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1/2.25 \times 10^{-3}}{s + 1/2.25 \times 10^{-3}} = \frac{444.44}{s + 444.44}$$



It can be seen that the MATLAB and Simulink results match up when comparing the two graphs.

c)

Below the cut off frequency has been determined.



A bit too small, difficult to read. Have seen the program run in the classroom.

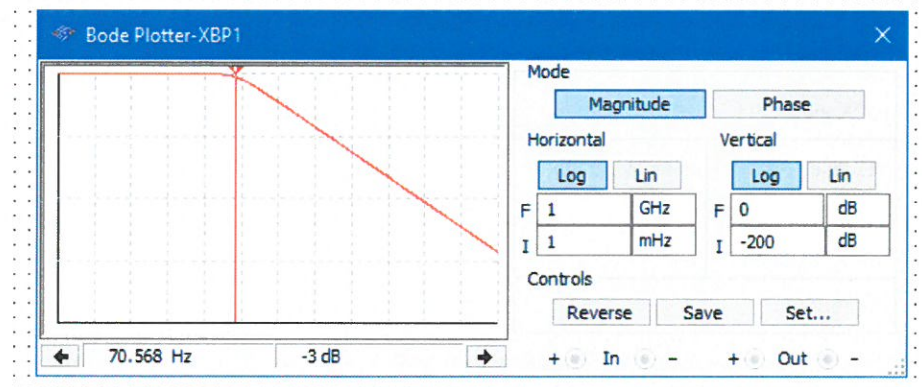
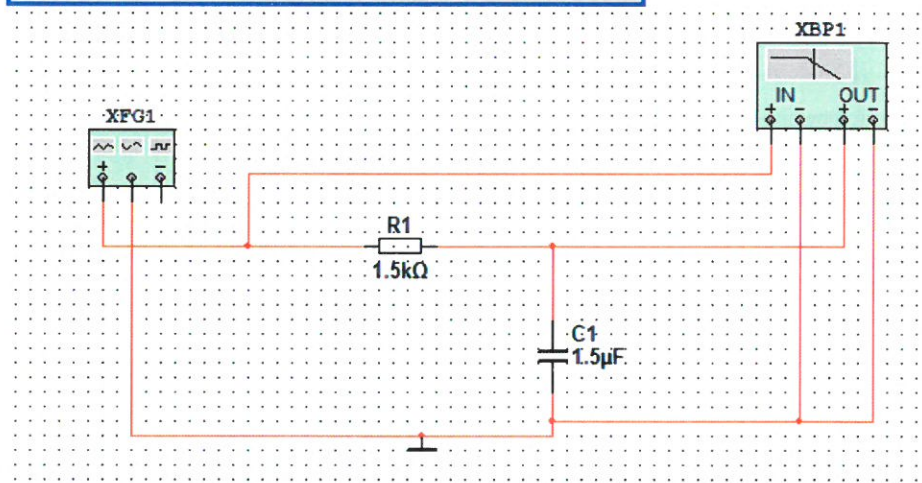
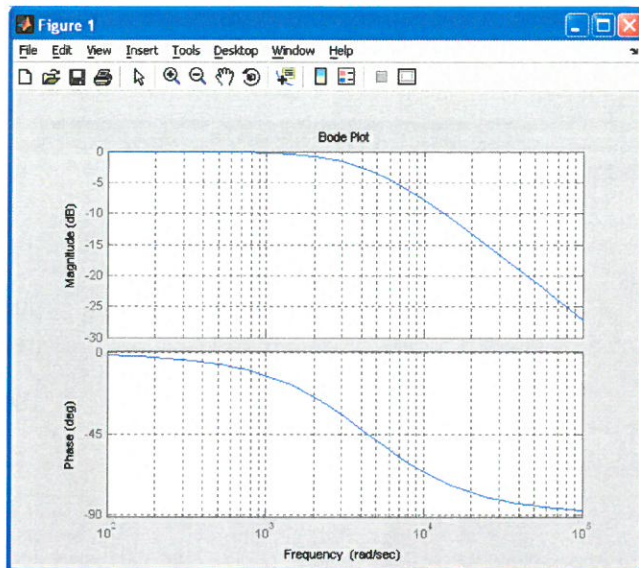
d)

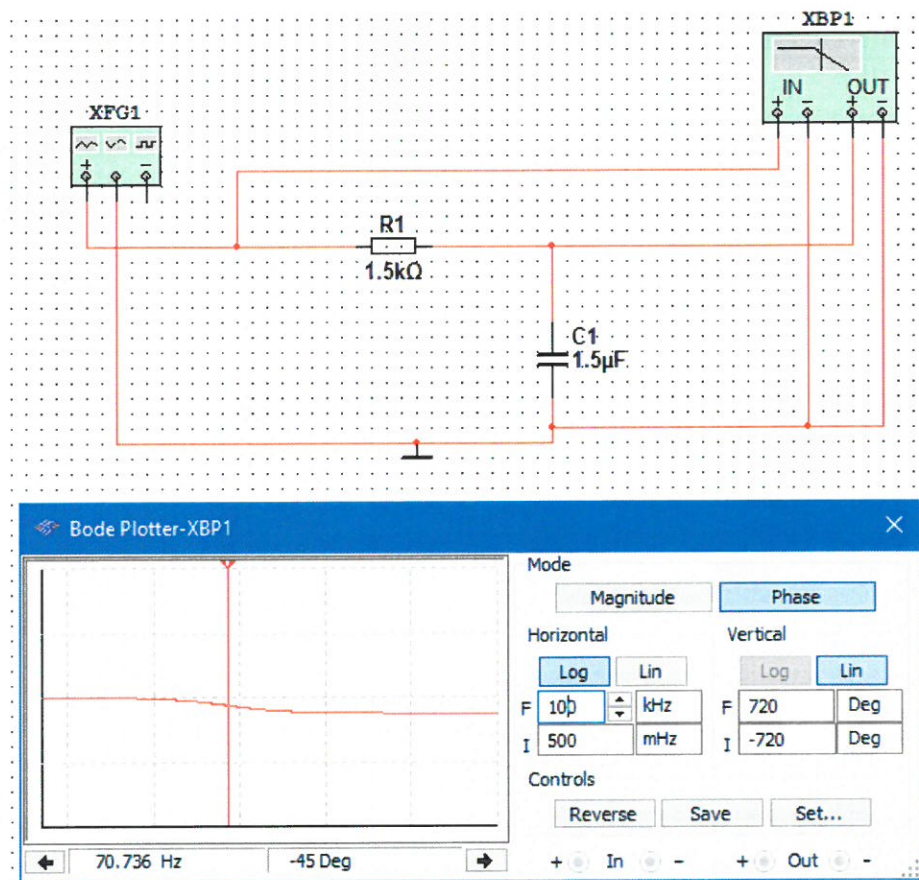
Below a bode plot and phase plots have been produced

```

1. %File Name - Assignment_2_task_1_part_d.m
2. %Task 1 - C
3. %Author - Tom Wolsey
4. %Date 014/03/2019
5. %Producing frequency response using Bode Plot
6.
7. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8. %Create Transfer Function %
9. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10. G=tf([1],[0.000225,1]);
11.
12. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13. %Bode Plot %
14. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15. bode(G)
16. grid
17. title('Bode Plot')

```





As can be seen when comparing the results from MATLAB and the multisim they show the same values

Confirm both results between matlab and multisim.

Could have compared the cut off frequency.

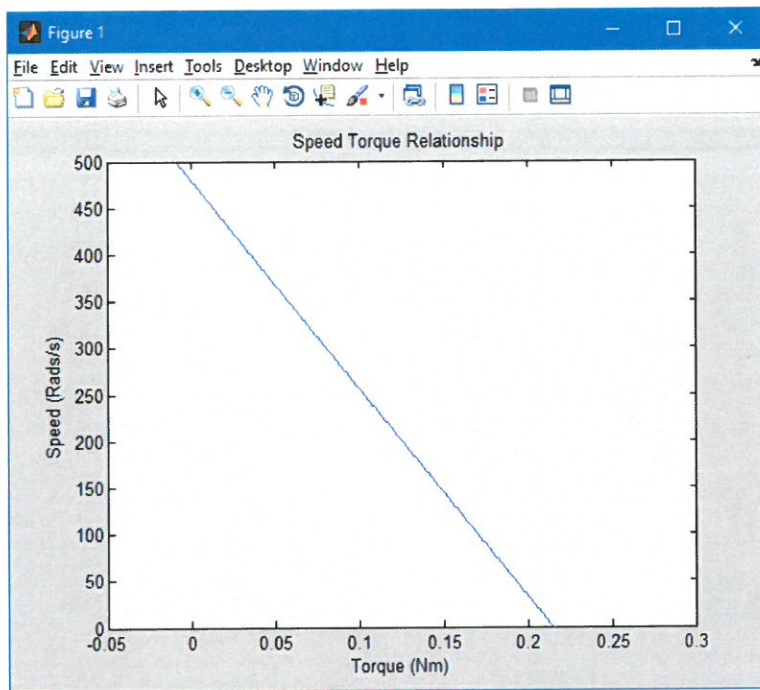
Task 2 - DC Motor Model

a)

Below the equivalent circuit for the armature voltage controlled DC motor

```
1. %File Name - Assignment_2_task_2.m
2. %Task 2
3. %Author - Tom Wolsey
4. %Date 14/03/2019
5. %DC Motor.
6.
7. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8. % Motor parameters %
9. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10. Ra=1.4; Va=12; Ts=0.21453;
11.
12. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13. % Calculation of Motor constant %
14. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15. k=(Ts*Ra)/Va;
16.
17. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18. % Max Speed %
19. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20. W0=Va/k;
21.
22. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23. % Speed Range %
24. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25. W=0:2:500;
26. T=k*(Va-k*W)/Ra;
27.
28. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29. % Display of results %
30. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31. disp('The Motor Constant, km')
32. disp(k)
33. disp('The Motor Max Speed, Rads/S')
34. disp(W0)
35. plot(T,W,'b-')
36. xlabel('Torque (Nm)')
37. ylabel('Speed (Rads/s)')
38. title('Speed Torque Relationship')
```

Good program.

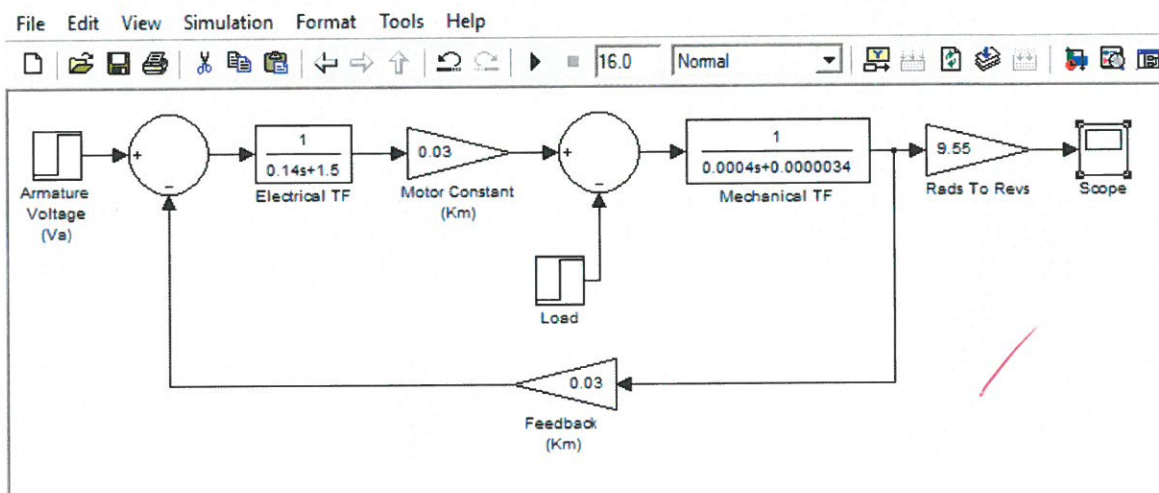


The Motor Constant, k_m
0.0250

The Motor Max Speed, Rads/S
479.4534

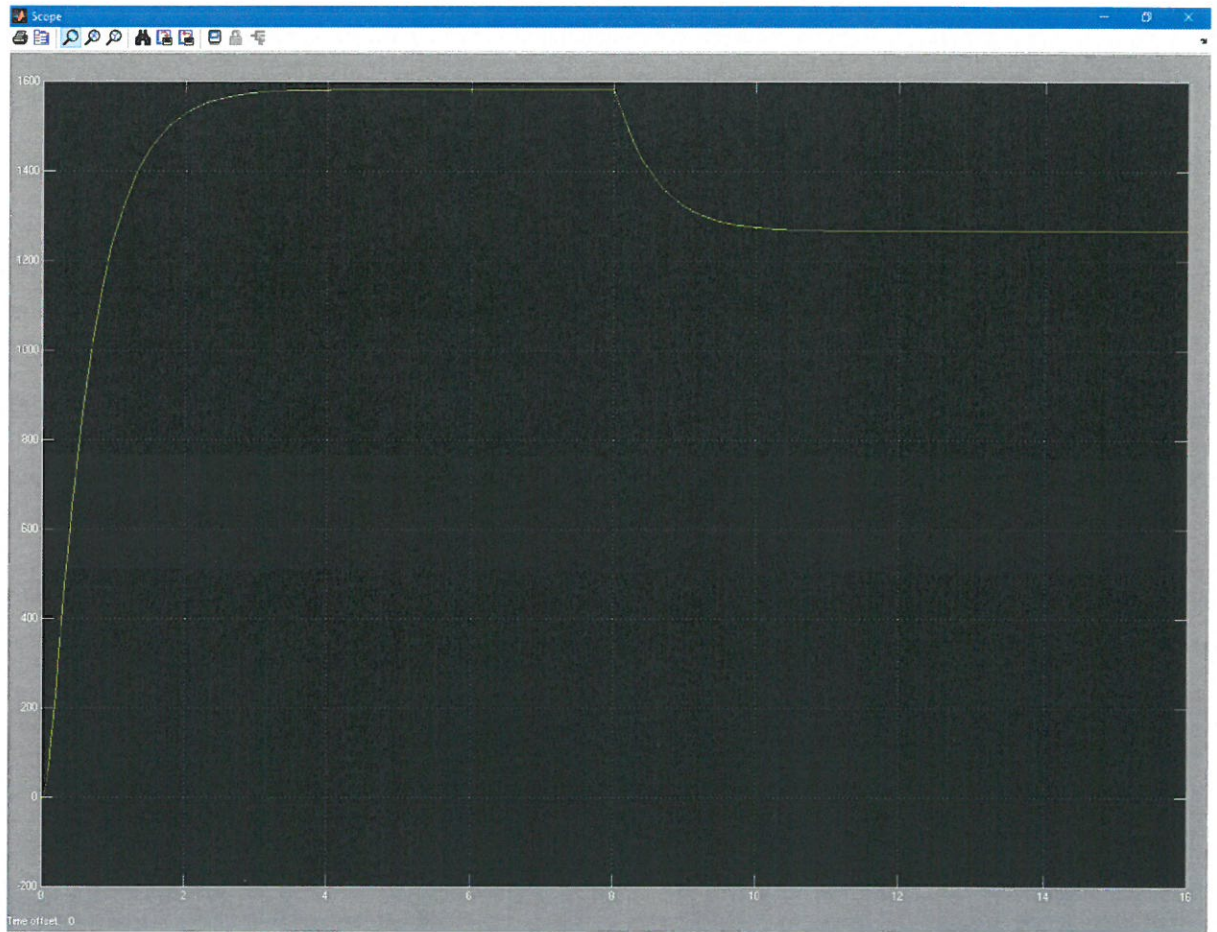
b)

A Simulink model for the armature-voltage controlled DC motor can be seen



c)

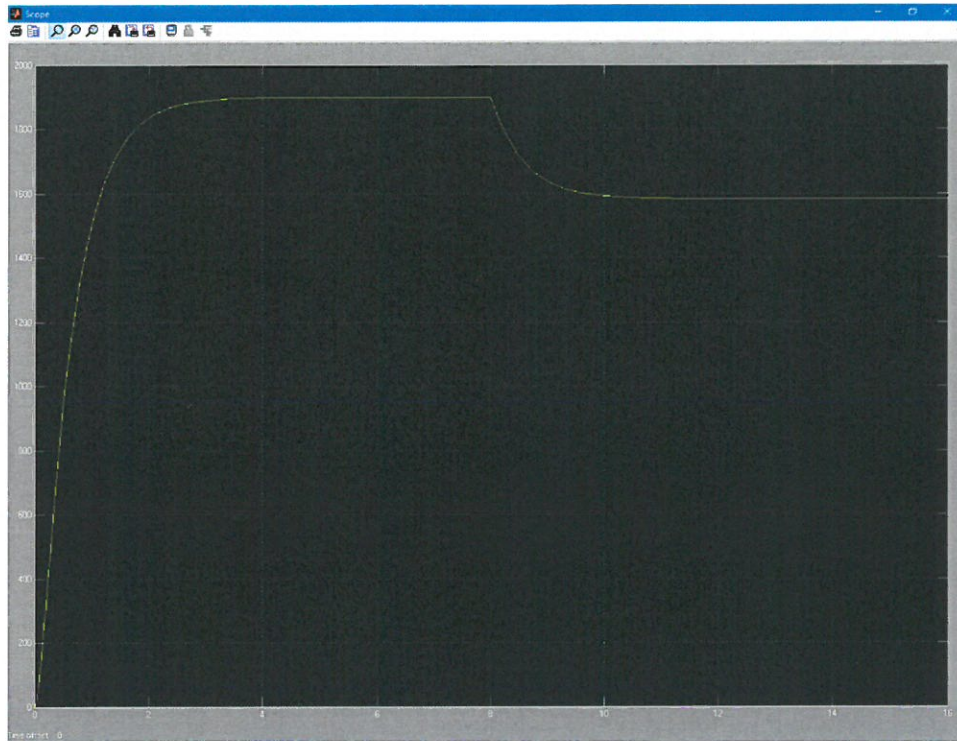
Below the speed response can be seen



d)

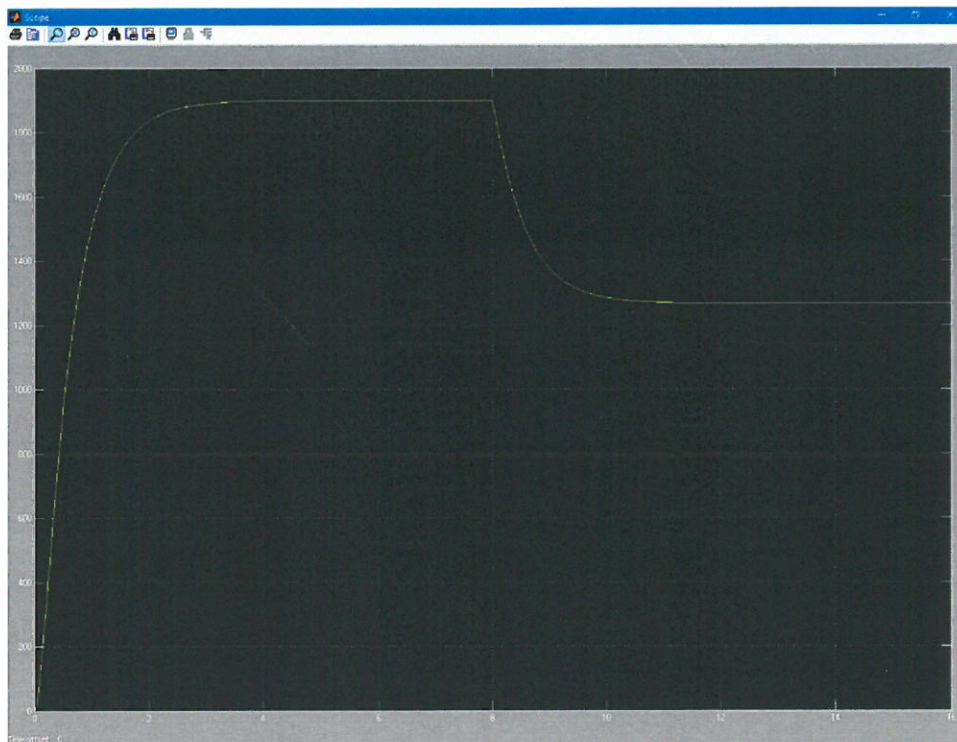
Below the steady state angular speeds have been determined.

$T = 0.02\text{Nm}$



Steady state RPM = 1580 *rpm* *Unit*

$T = 0.04\text{Nm}$

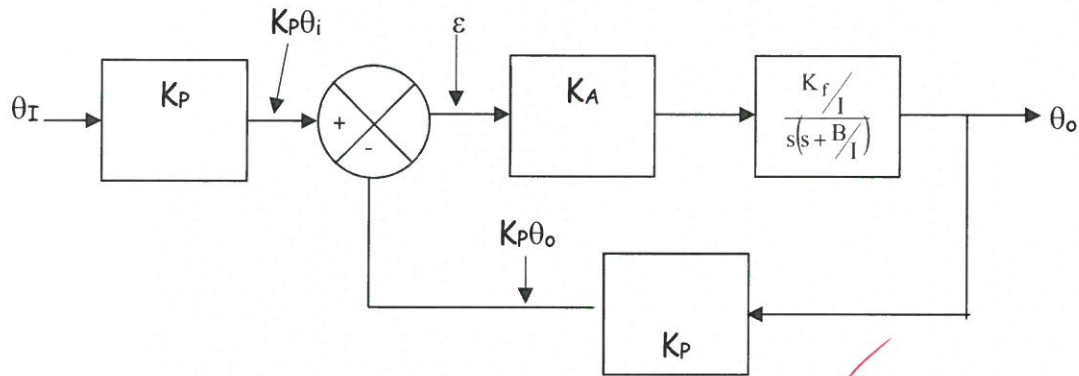


Steady state RPM = 1250

Task 3 - Position Control Servo

a)

Block diagram for the position controlled servo



b)

The transfer function for the system is as follows

$$\frac{\theta_o}{\theta_i} = \frac{G(s)}{1 + G(s)H(s)}$$

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

$$H(s) = K_P$$

Transfer function

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

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

The transfer function

↓ new page.

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

c)

Below can be seen the value for B being found, the value of B is then used to create a critically damped system

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

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

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

$$\omega_n^2 = 200$$

So Steady state value = $k_{ss} = 1$

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

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

for critically damping $\zeta = 1$

So $2 \times 1 \times 14.14 = 5000B$

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

To determine the poles for critically damped

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

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

MATLAB program for critically damped system

```

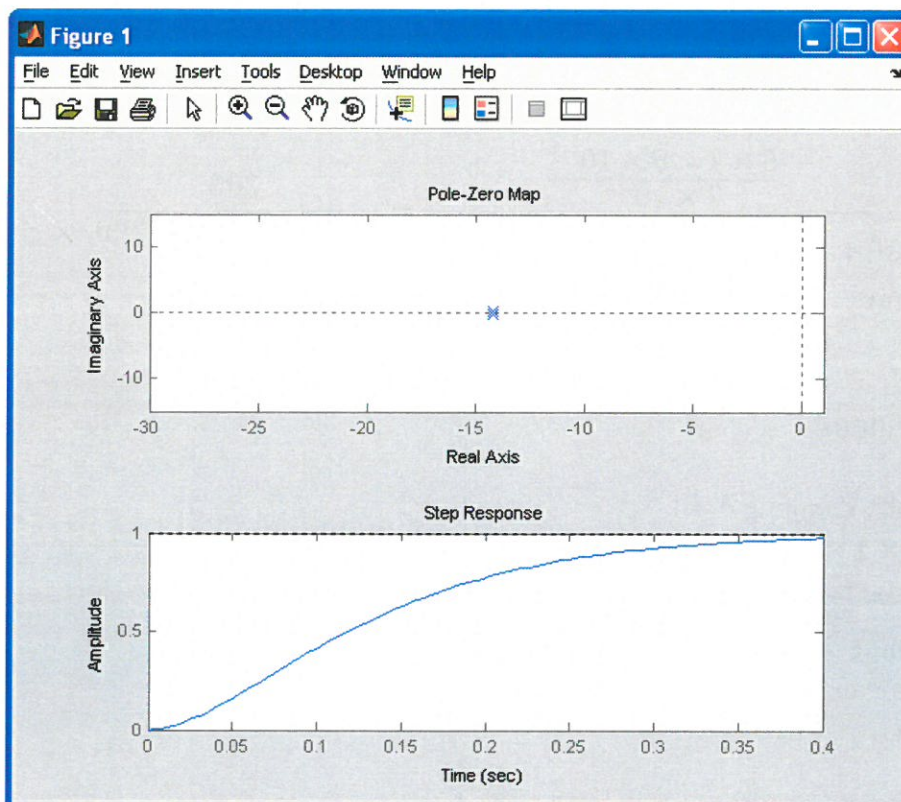
1. %File Name - Assignment_2_task_2_C.m
2. %Task 2 - C
3. %Author - Tom Wolsey
4. %Date 21/03/2019
5.
6. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7. %Create the numerator
8. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9. num=[200];
10.
11. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12. %Create the denominator
13. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14. den=[1 28.28 200];
15.
16. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17. %Transfer Function
18. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19. sys=tf(num,den)
20.
21. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

22. %Poles
23. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24. pole(sys)
25.
26.
27. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28. %Plotting the Poles
29. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30. subplot(211), pzmap(sys)
31. axis([-30 1 -15 15])
32.
33. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34. %Plotting the response for an unit step
35. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
36. subplot(212), step(sys)

```



could have extended the axis to show a more full response.

Transfer function:

200

$s^2 + 28.28s + 200$

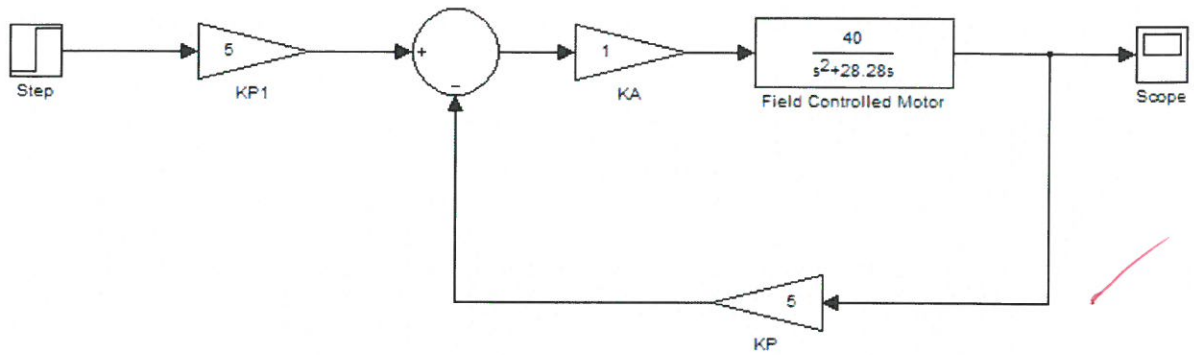
ans =

-14.1400 + 0.2458i

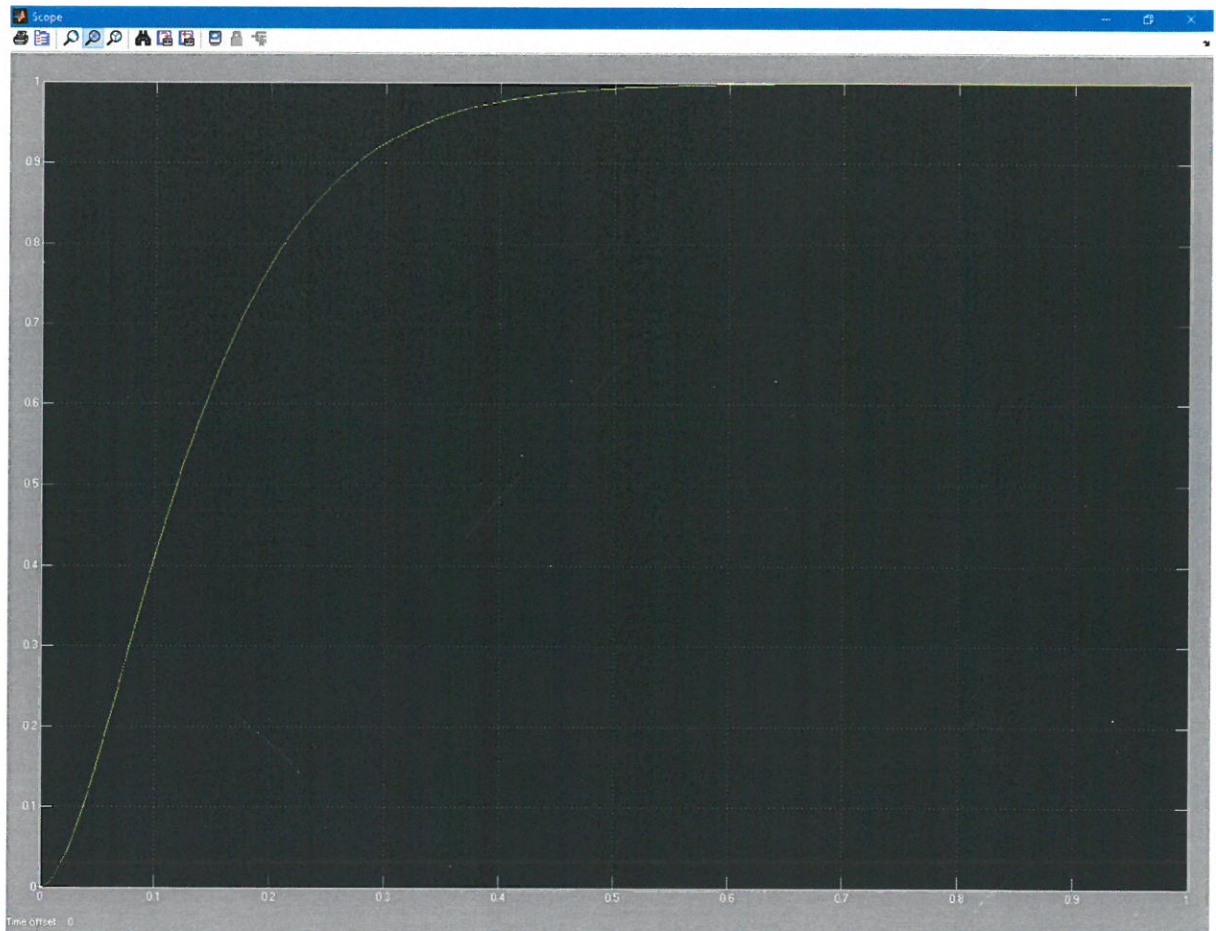
-14.1400 - 0.2458i

d)

The time response for the system has been obtained and can be seen below



Below can be seen the time response for the system, this was created using MATLAB, this has also been modelled in Simulink to compare the results.



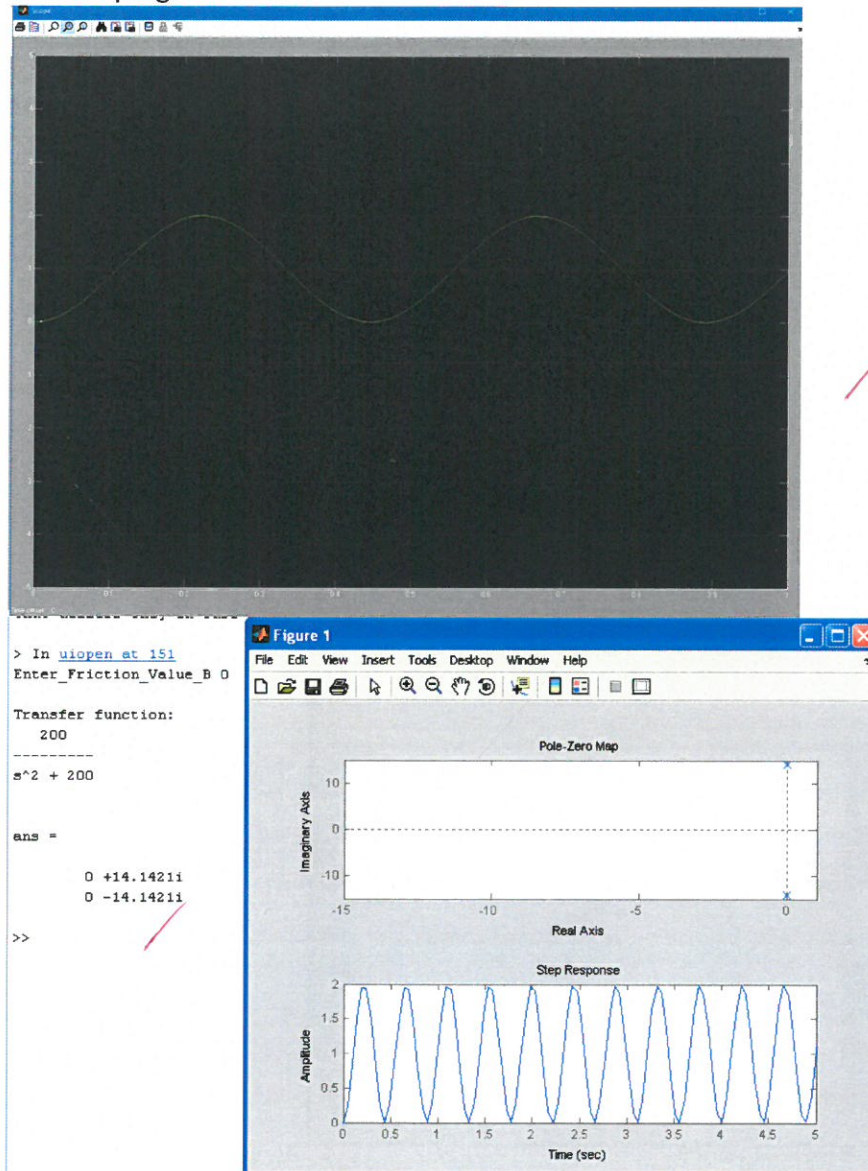
The MATLAB time response is show in Part C, this matches the one obtained from Simulink which can be seen above.

e)

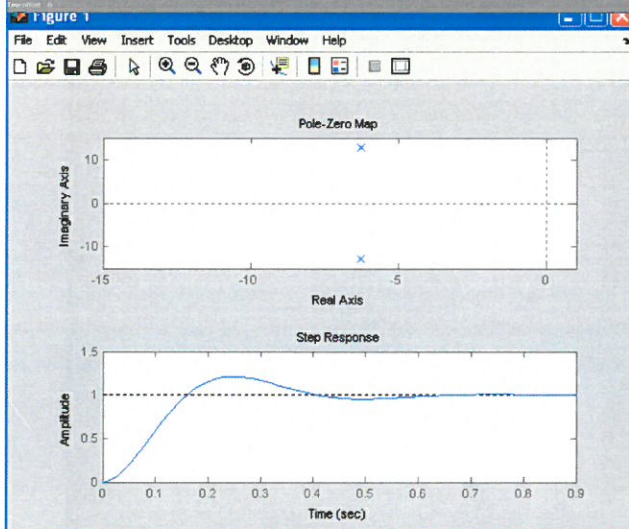
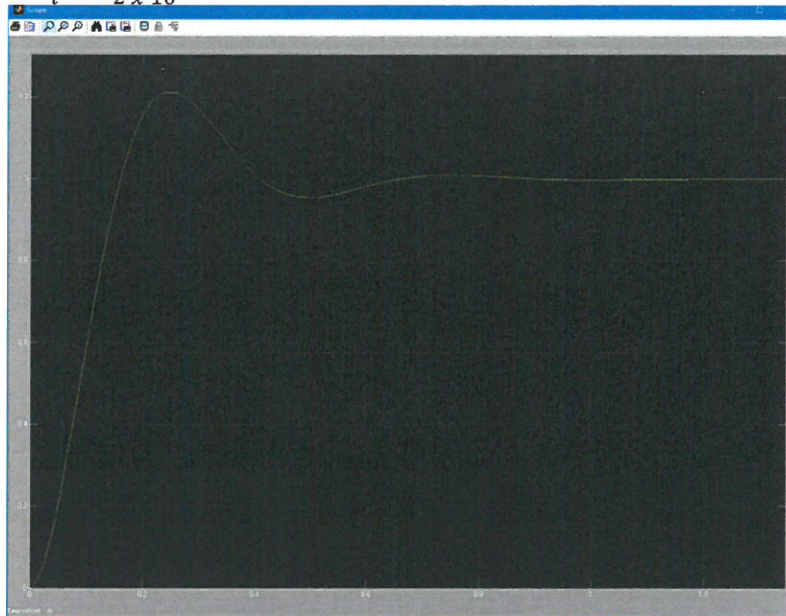
```
1. %File Name - Assignment_2_task_2_C.m
2. %Task 2 - C
3. %Author - Tom Wolsey
4. %Date 21/03/2019
5.
6. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7. %Create the numerator          %
8. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9.
10. B=input('Enter Friction Value B '); %Program waits for user to input value
11. BI=B/(2*10^-4); %Calculation B/I
12. num=[200];
13.
14. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15. %Create the denominator          %
16. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17. den=[1 BI 200];
18.
19. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20. %Transfer Function          %
21. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22. sys=tf(num,den)
23.
24. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25. %Poles          %
26. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27. pole(sys)
28.
29.
30. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31. %Plotting the Poles          %
32. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33. subplot(211), pzmap(sys)
34. axis([-15 1 -15 15])
35.
36. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
37. %Plotting the response for an unit step          %
38. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39. subplot(212), step(sys)
```

Critically damped = 5.657×10^{-3}

- No damping = 0



- Under damped Let $B = 2.5 \times 10^{-3}$
- $$So \frac{b}{i} = \frac{2.5 \times 10^{-3}}{2 \times 10^{-4}} = 12.5$$



Enter_Friction_Value_B 0.0025

Transfer function:
200

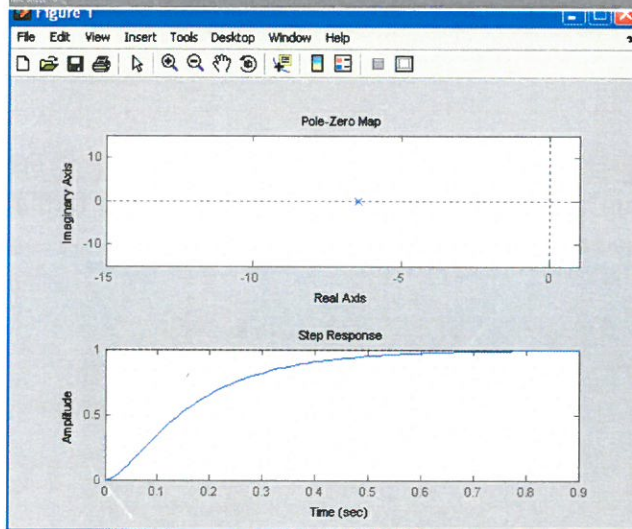
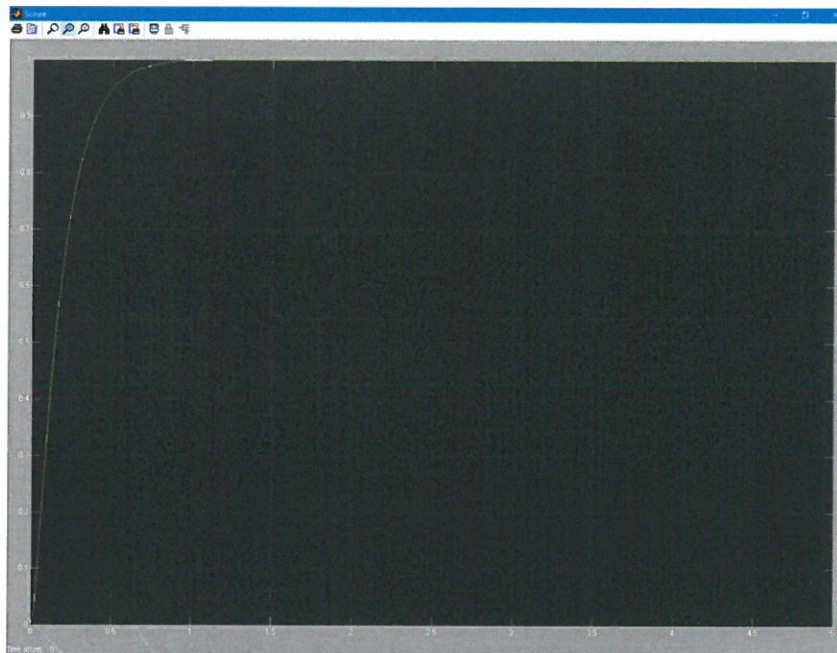
 $s^2 + 12.5 s + 200$

ans =

-6.2500 +12.6861i
-6.2500 -12.6861i

>>

- Over damped Let $B = 7.5 \times 10^{-3}$
- $So \frac{b}{i} = \frac{7.5 \times 10^{-3}}{2 \times 10^{-4}} = 37.5$



Enter_Friction_Value_B 0.0075

Transfer function:
200

 $s^2 + 37.5 s + 200$

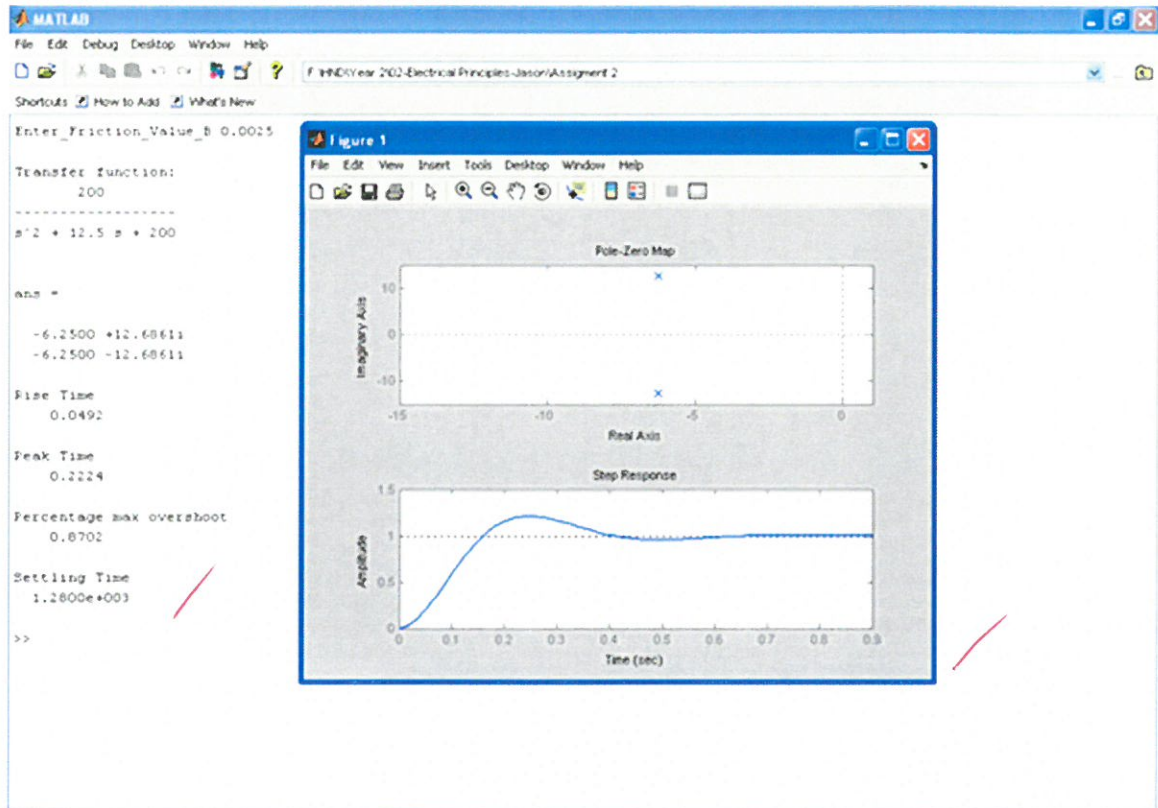
ans =

-31.0611
-6.4389

>>

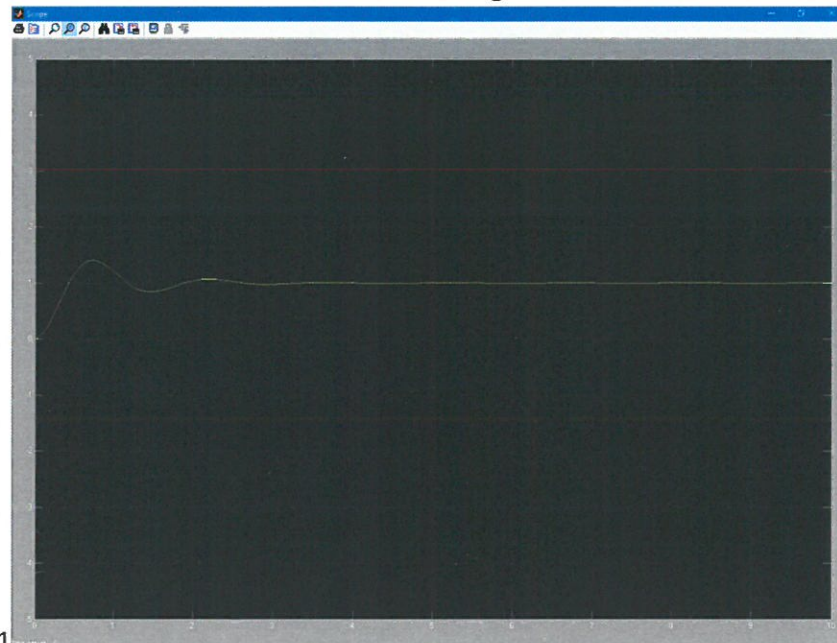
f)

Below can be seen the Rise time, Peak time, percentage maximum overshoot and settling time when the system is in underdamped.

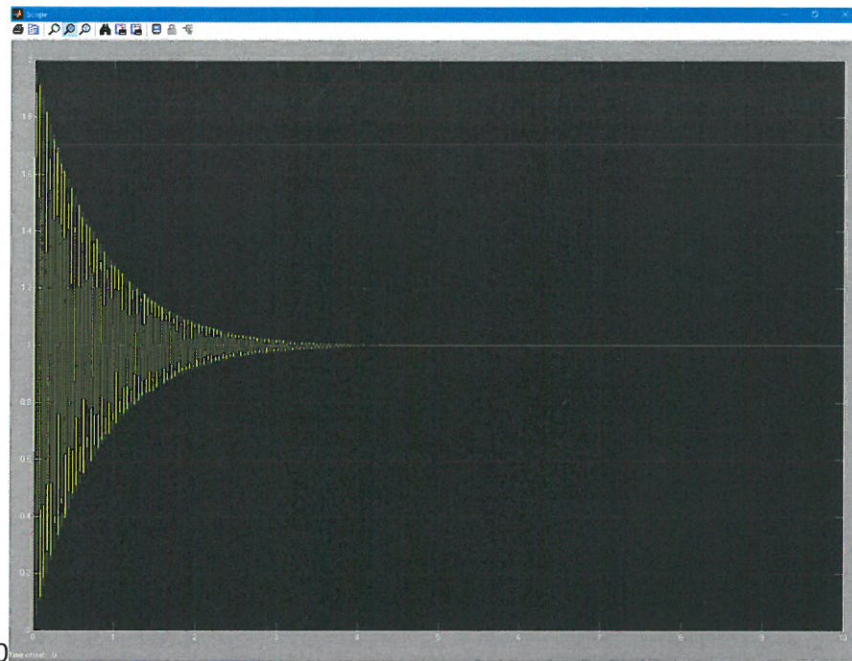


Should have included the modified program.

The MATLAB program was also run in two other conditions to see how the performance would be affected when the value of B was changed.



KA = 0.1



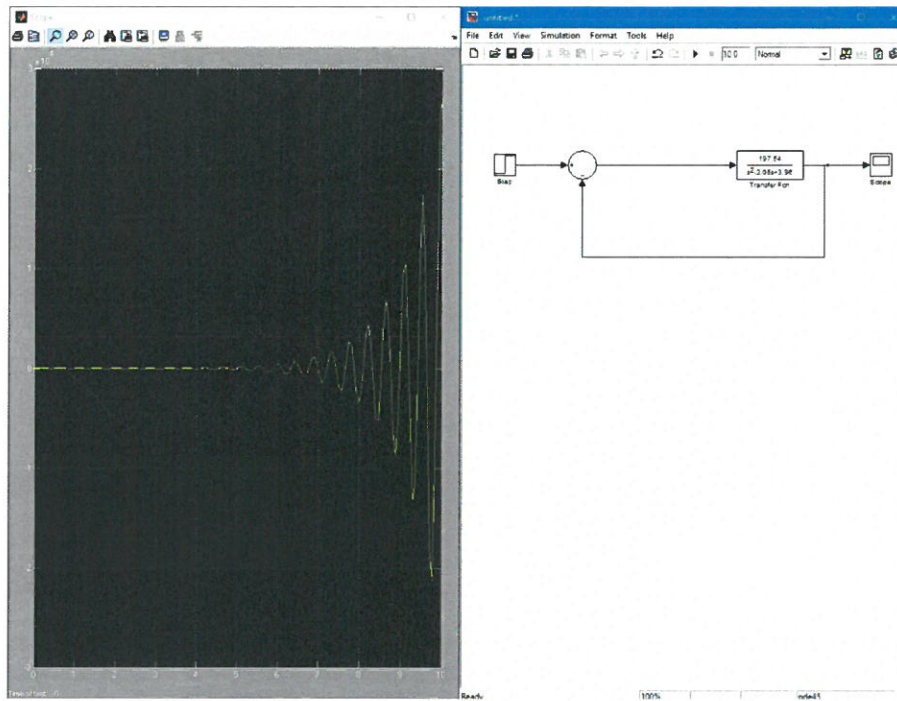
KA = 100



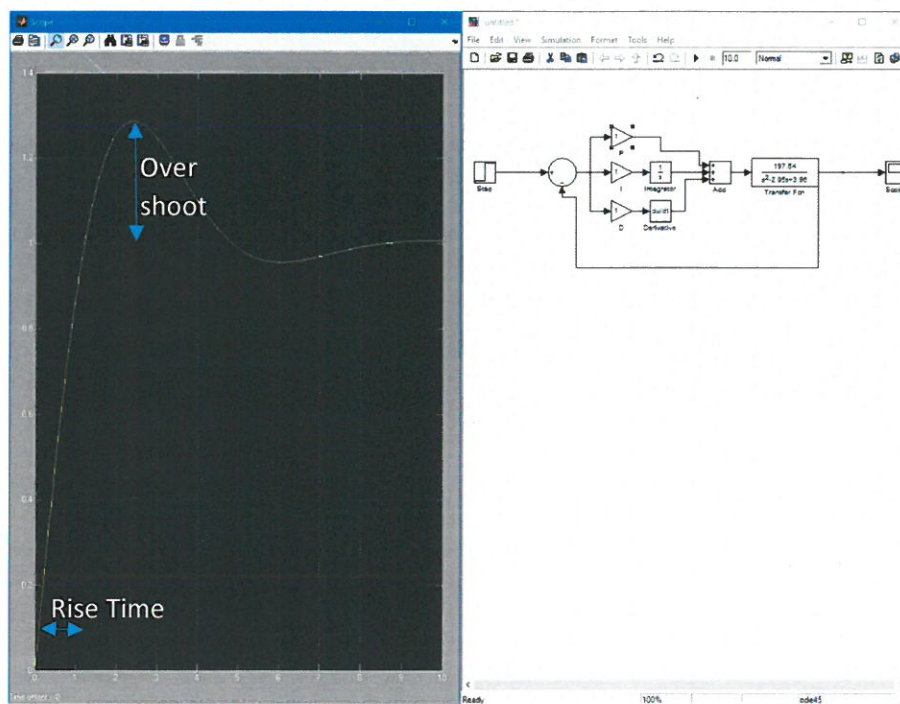
KA = 1

Increasing the gain of the system causes a faster response but there is also more oscillations.

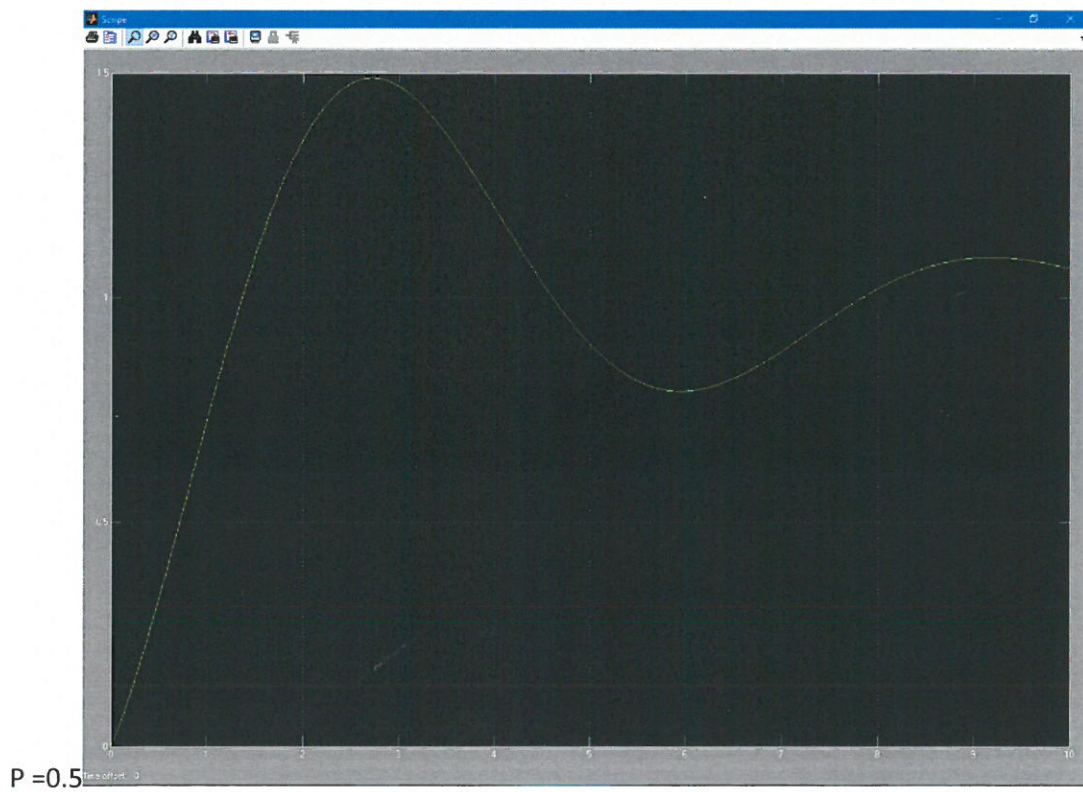
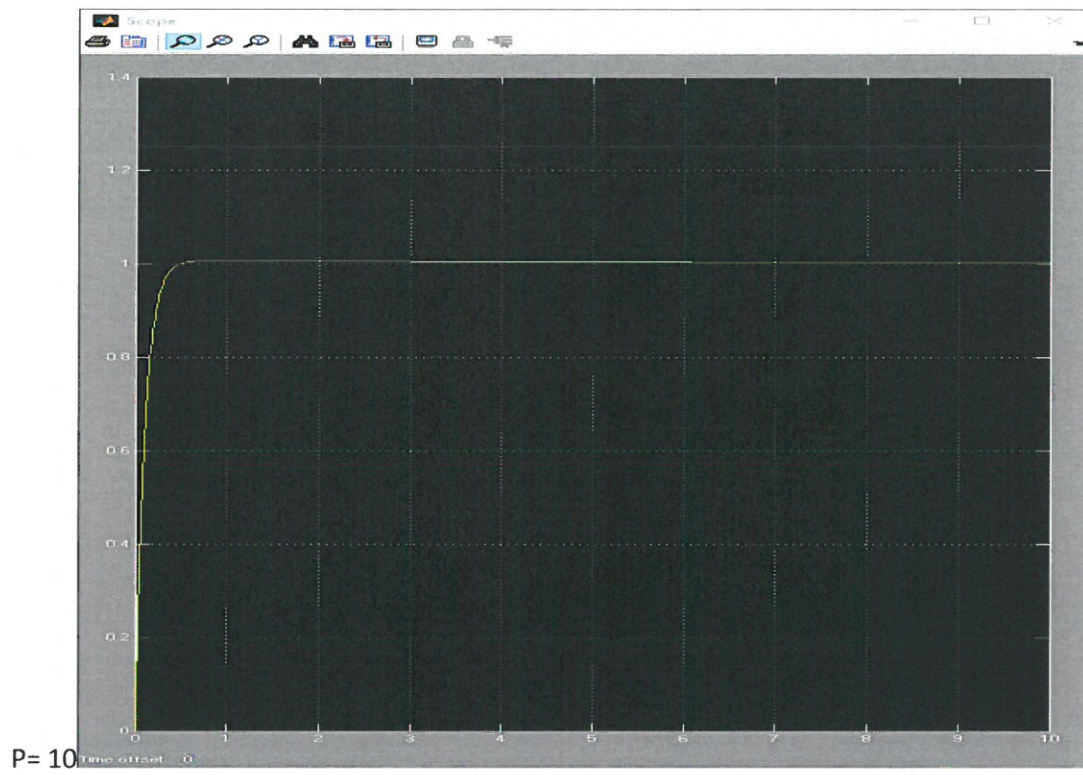
Task 4

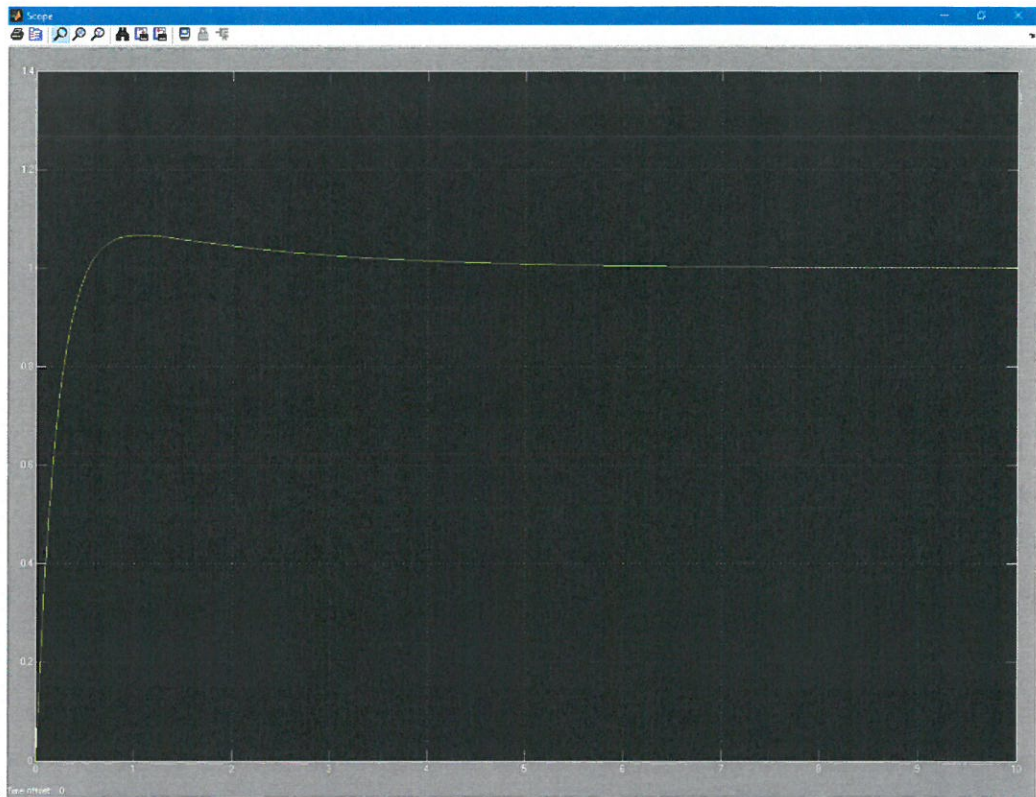


PID Intel state

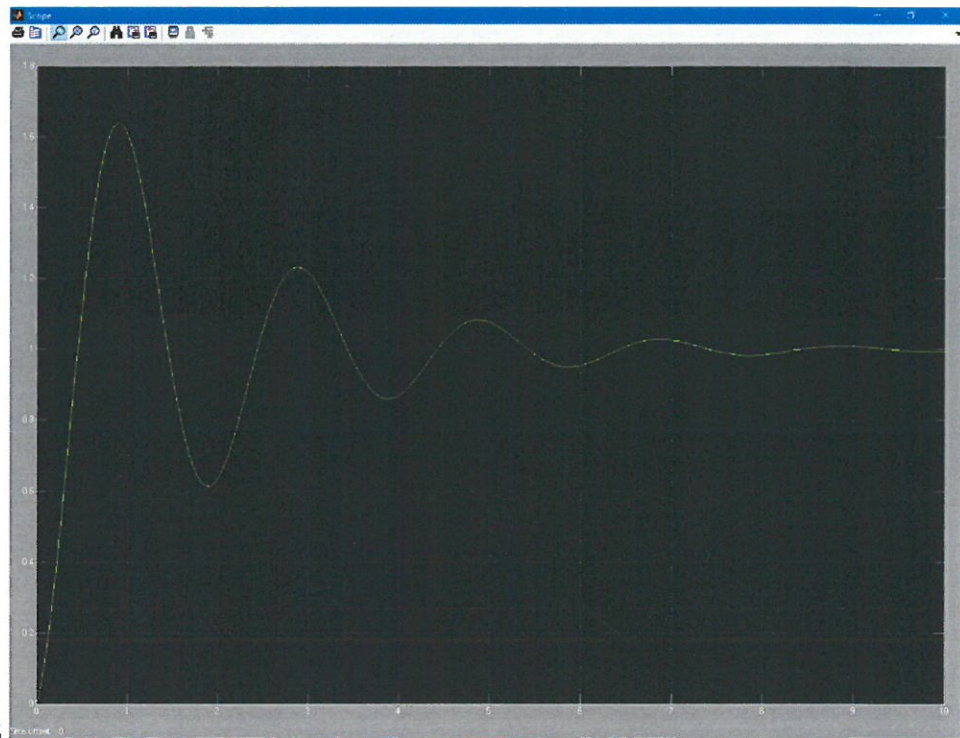


$Tr \Rightarrow 10 \text{ to } 90\%$



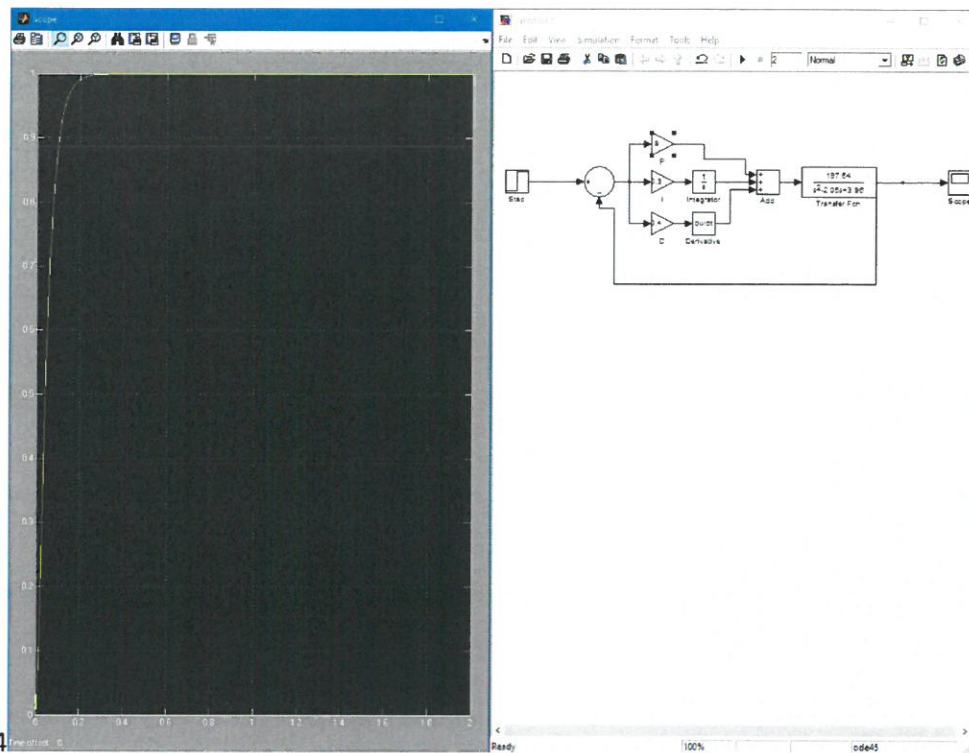


$P = 5, I = 2.5$



$P = 0.5, I = 5, D = 0.5$

$P = 8 \quad I = 0.3 \quad D = 0.4$



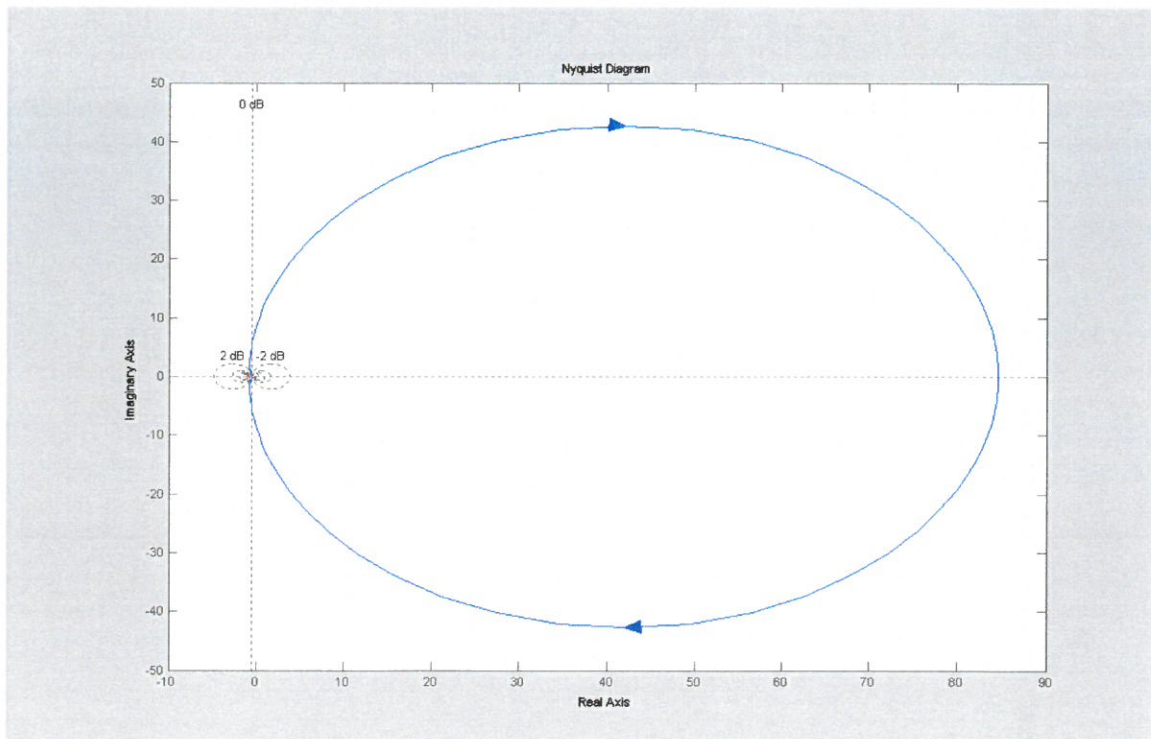
Task 5

a)

A program to plot a Nyquist diagram for an open loop transfer function of a servo control rig.

```

1. %File Name - Assignment_2_task_5_part_a.m
2. %Task 5 - a
3. %Author - Tom Wolsey
4. %Date 02/04/2019
5. %Using Matlab to obtain the Myquist Diagram
6.
7. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8. %The component values
9. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10.
11. num=[0.03868 -5.369 -97.34 13450];
12. den=[1 169.8 1730 159];
13.
14. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15. %Create the transfer function
16. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17.
18. Ny=tf([num],[den])
19.
20. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21. %Plot the Nyquist response
22. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23.
24. nyquist(Ny)
25. grid
  
```



b) /c

Close Loop

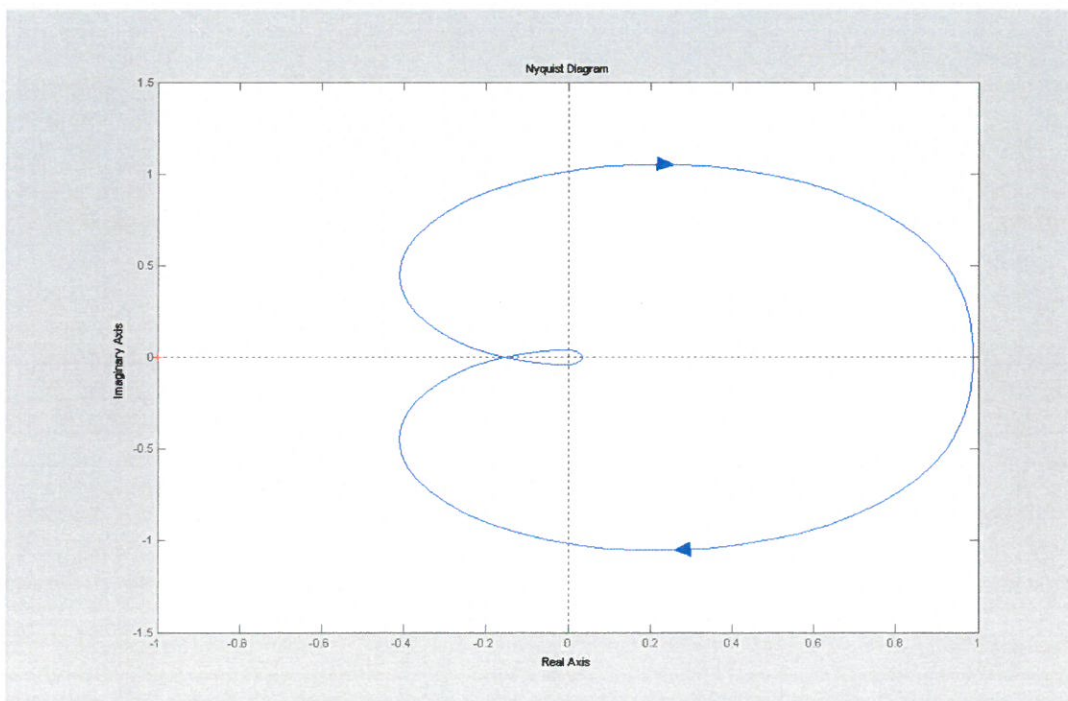
$$G_C(s) = \frac{0.03868s^3 - 5.369s^2 - 97.34s + 13450}{s^3 + 169.8s^2 + 1730s + 159 + 0.03868s^3 - 5.369s^2 - 97.34s + 13450}$$

$$G_C(s) = \frac{0.03868s^3 - 5.369s^2 - 97.34s + 13450}{1.03868s^3 + 164.431s^2 + 1632.66s + 13609}$$

```

1. %File Name - Assignment_2_task_5_part_b.m
2. %Task 5 - b
3. %Author - Tom Wolsey
4. %Date 02/04/2019
5. %Using Matlab to obtain the Myquist Diagram
6.
7. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8. %The component values
9. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10. num=[0.03868 -5.369 -97.34 13450];
11. den=[1.03868 164.431 1632.66 13609];
12.
13. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14. %Create the transfer function
15. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16.
17. Ny=tf([num],[den]) % Create the transfer function
18.
19. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20. %Plot the Nyquist response
21. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22.
23. nyquist(Ny) % Plot the Nyquist response
24. pause
25.
26. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27. %Plot the Bode plot
28. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29.
30. bode(Ny) % Plot the Bode plot
31. grid

```



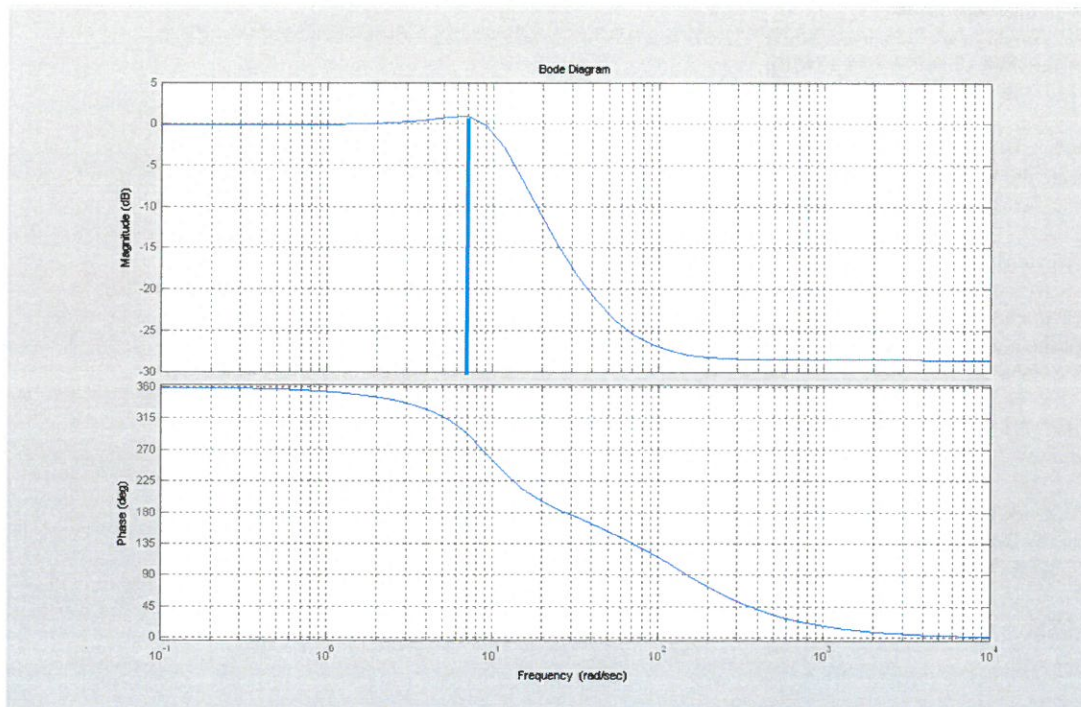
The system is stable the response is in the negative axis.

The phase margin is infinite because the gain is less than one.

Crosses the real axis at -0.17.

$$\text{Gain margin, } GM = \frac{1}{0.17} = 5.88$$

$$GM_{(dB)} = 20\log(5.88) = 15.39\text{dB}$$

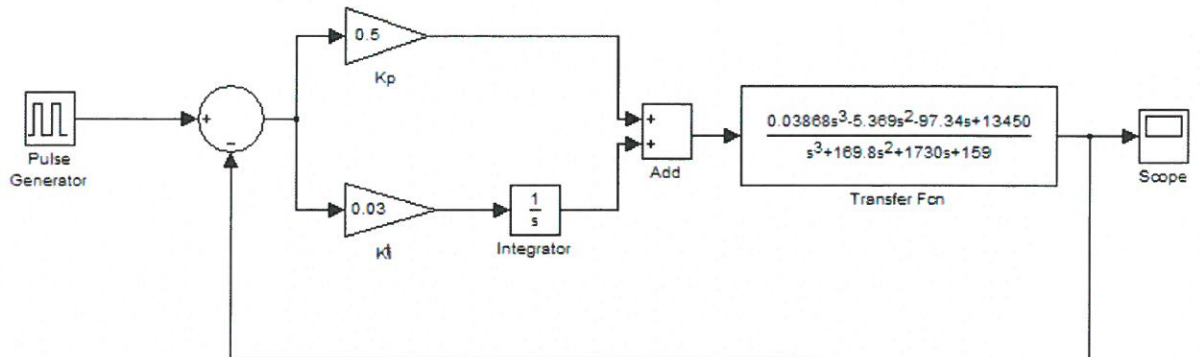


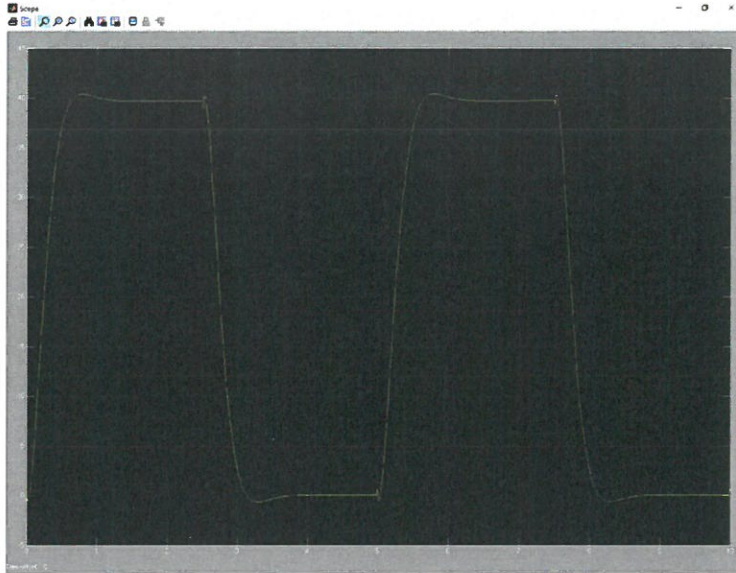
Resonance
 $M_p = 7\text{Hz}$

Bandwidth ω_b
 $= 30 - 9 = 21\text{Hz}$

d)

The final setup for the tuning of the system can be seen below in the screenshot of the PI system





What about changing K_p and also you have not explained the method of tuning.