

LEVEL 5 ASSESSMENT SPECIFICATION

Student name:	Sean Beynon	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	21.1
Achieved 74% for Task 1. Achieved 75% for Task 2. Achieved 65% for Task 3. Achieved 75% for Task 4. Achieved 62% 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
Some evaluation and analysis of concepts and principles, though descriptive in parts.		
4. Practical Application Skills	50	36.3
Achieved 75% for Task 1. Achieved 75% for Task 2. Achieved 60% for Task 3. Achieved 78% for Task 4. Achieved 75% 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	6.5
Work is accurate, coherent, fluent, well-structured and organised.		

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%)

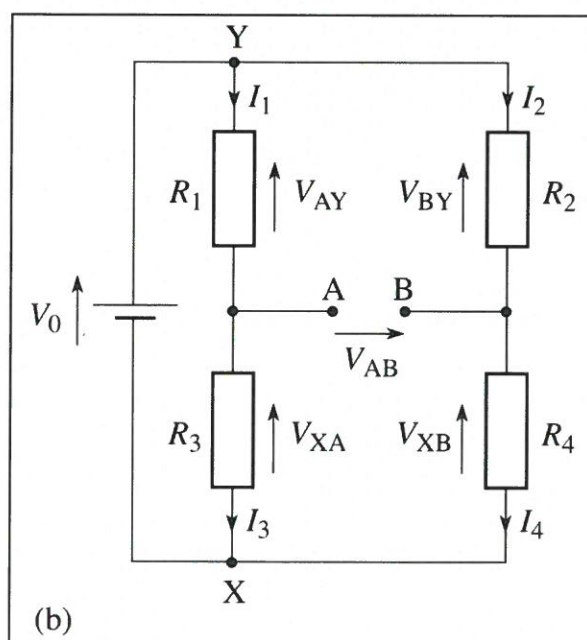
68%

HNC POWER SYSTEM ENGINEERING

Module: Electrical Principles, Methods and Simulations

Assignment: Control Systems

Sean Beynon
National Grid
2018



Summary

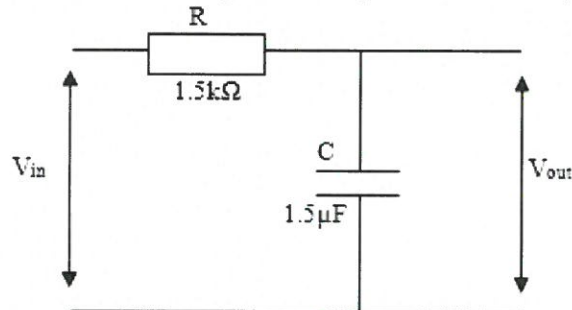
This Assignment provides a variety of techniques to scrutinize the performance of electrical and electronic circuits. The assignment offered the ability to cross reference values, this demonstrating the ability to perform tasks using mathematical simulation and computerised modelling techniques. These techniques together with the analysis of the circuits increases the understanding of circuit theory and principles.

Table of contents

Summary.....	2
Table of contents.....	2
Task 1- RC Circuit First Order System.....	3
Task 2- DC Motor Model	9
Task 3- Position Control Servo	12
Task 4	22
Task 5	26

Task 1- RC Circuit First Order System

Below is the RC circuit that is to be analysed using time and frequency response.



a)

Kirchhoff law states that:

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

Obtaining the first derivative on both sides provides:

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

Laplace transform can be used to model the series electric RC circuit:

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

$$I(s) \left[sR + \frac{1}{C} \right] = sV_{in}(s)$$

$$I(s) \left[R + \frac{1}{Cs} \right] = V_{in}(s)$$

$$V_{in}(s) = I(s) \left[R + \frac{1}{Cs} \right] \quad \text{[Equation (1)]}$$

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 Equation (1):

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

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

Therefore:

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

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:

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

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

b)

Matlab was used to determine the time constant, delay time and rise time of the circuit.

```

1  % First order response of a RC circuit by Sean Beynon 14/02/19.
2
3  % below is the values for the Resistor, capacitor and voltage
4 - R = 1.5*10^3; C = 1.5*10^-6; V = 20;
5
6 - Tc=R*C;          % Calculate time constant
7 -   t=0;           % Set the initial conditions
8 -   Vout=0;
9
10  %Rise Time calculation
11 -   TR= -Tc*(log(0.1)-log(0.9));
12
13  %Delay Time Calculation
14
15 -   TD= -Tc * (log(0.5));
16
17  % create the output voltage for value of time, t.
18 - for k=1:70
19 -     t(k)=(k-1)*Tc/6;
20 -     Vout(k)=V*(1-exp(-t(k)/Tc));
21 - end;
22
23  % Display the Vout against time
24 - plot(t,Vout,'r-');
25 -     title('First Order Response - RC Circuit')
26 -     xlabel('Time (sec)')
27 -     ylabel ('Voltage (V)')
28 -     grid
29 -     disp('time constant =')
30 -     disp(Tc)
31 -     disp('Rise Time =')
32 -     disp (TR)
33 -     disp('Delay Time =')
34 -     disp (TD)

```

The calculations produced the following results.

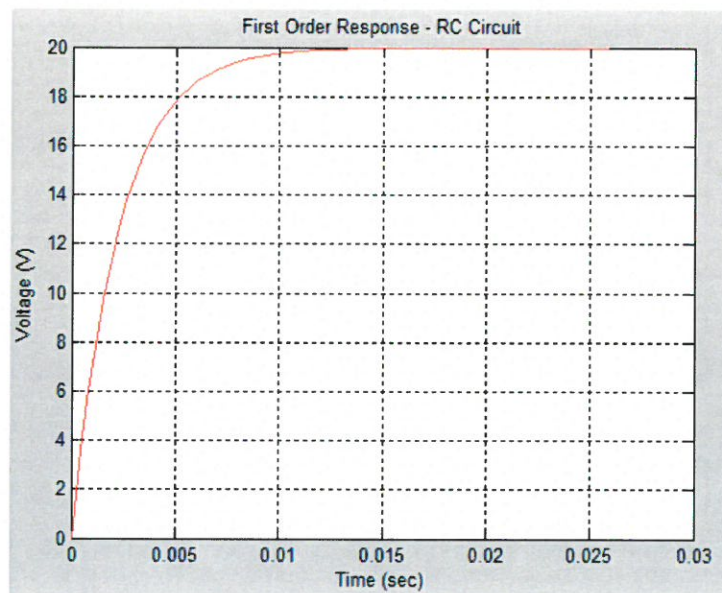
Time constant = 0.0023

Rise Time = 0.0049

Delay Time = 0.0016

You need to get your program to include units.

Matlab was also used to produce a time response plot of the circuit when $V_{in} = 20V$



The time response plot was confirmed using Simulink.

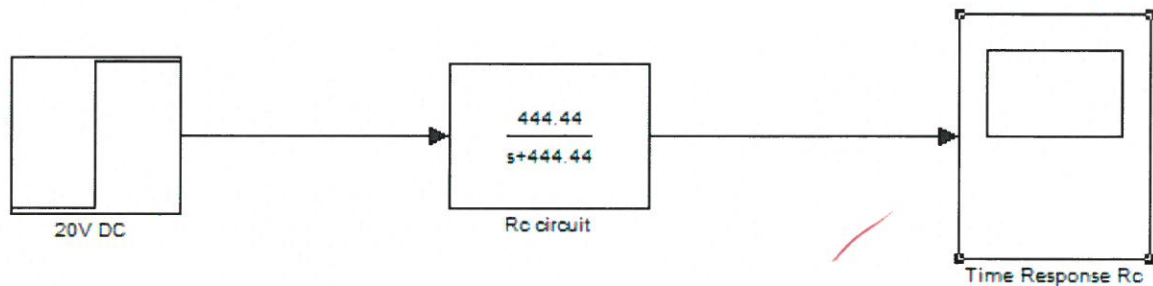
$$R = 1.5 \text{ k}\Omega \quad C = 1.5 \text{ }\mu\text{F}$$

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

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1/1.5 \times 10^{-3} * 1.5 \times 10^{-6}}{s + 1/1.5 \times 10^3 * 1.5 \times 10^{-6}}$$

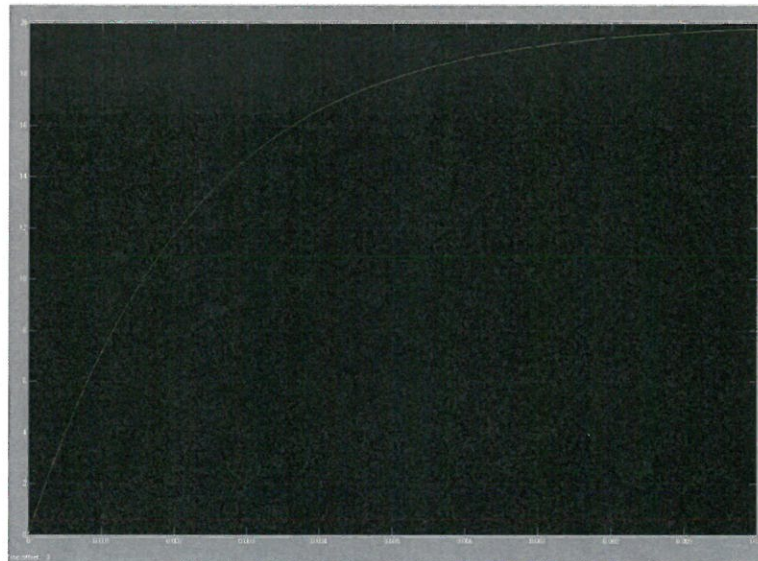
$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1/2.25 \times 10^{-3}}{s + 1/2.25 \times 10^{-3}} = \frac{444.44}{s + 444.44}$$

Diagram below shows the Simulink circuit created.



The Simulink time response plot confirms that the program created in Matlab functions correctly.

*Not very clear
but OK on the
electronic copy.*



c)

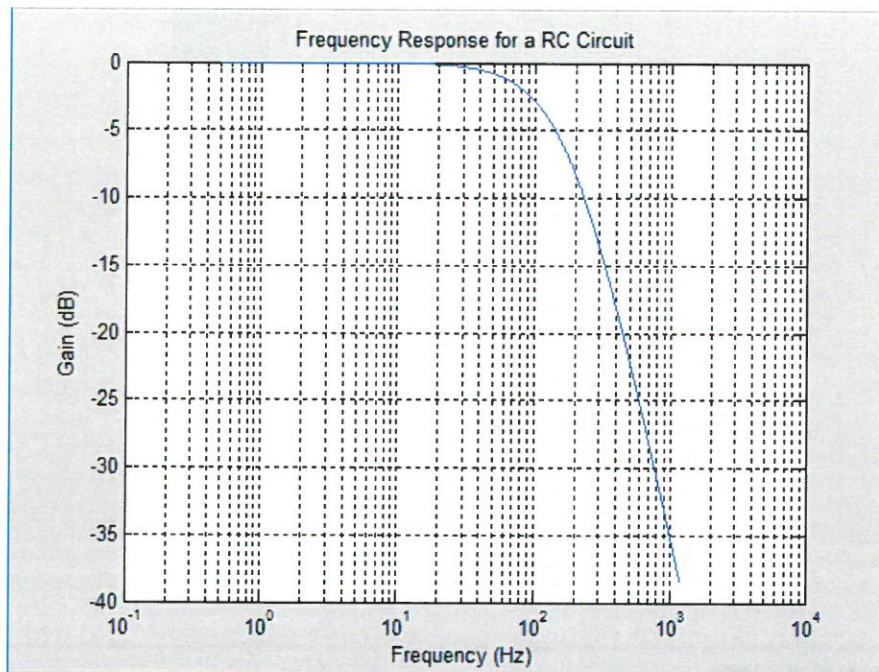
Shown below is the program created to determine the time response when a sine wave is applied to the input frequency.

```

1  % Time Response MATLAB Program by Sean Beynon
2  % Show frequency response with a 1V sine wave
3
4  R = 1500; C = 1.5*10^-6;          % Component values
5  f=0;                               % Set initial conditions
6
7  % Time constant
8  Tc=R*C;
9
10  fx=0:0.5:1200;
11  z=length(fx);
12  % For loop
13  for x=1:z
14      f=f+0.2;
15
16      % calculating angle
17      w(x)=2*pi*f;
18
19      % Time response
20      G(x)=20*log(1/(sqrt(1+w(x)^2*Tc^2)));
21  end
22
23  % The frequency Response
24  semilogx(fx,G)
25  xlabel('Frequency (Hz)')
26  ylabel('Gain (dB)')
27  title('Frequency Response for a RC Circuit')
28  grid
29

```

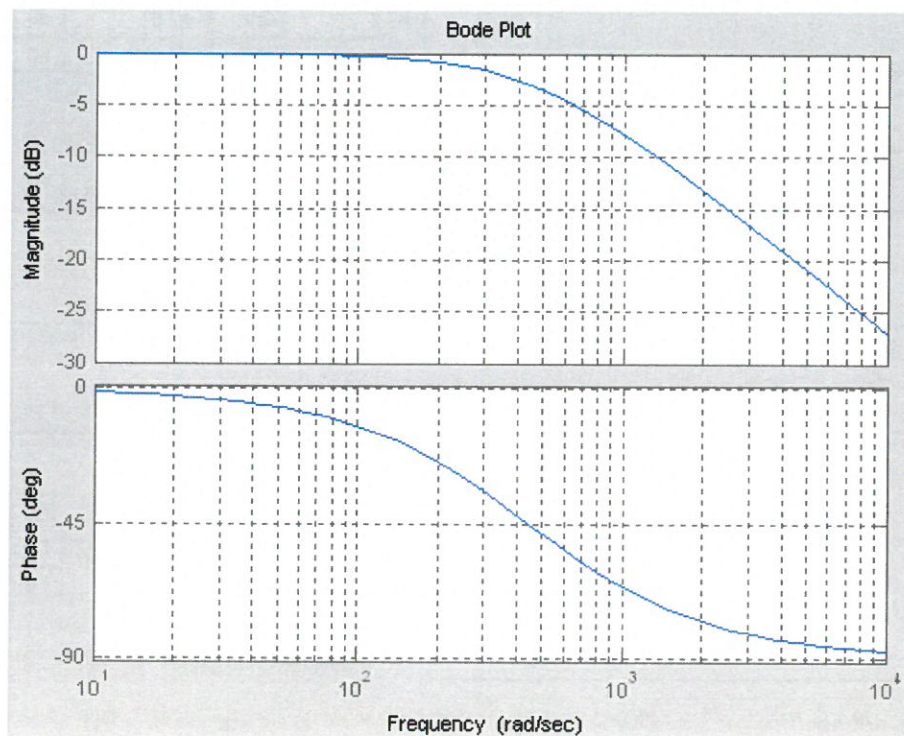
Frequency response plot when a sine wave is applied to the frequency.



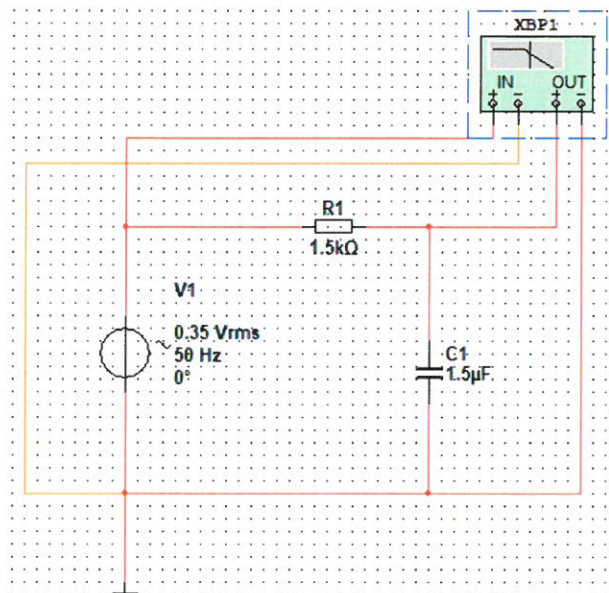
Have not
found the
cut off frequency

d)

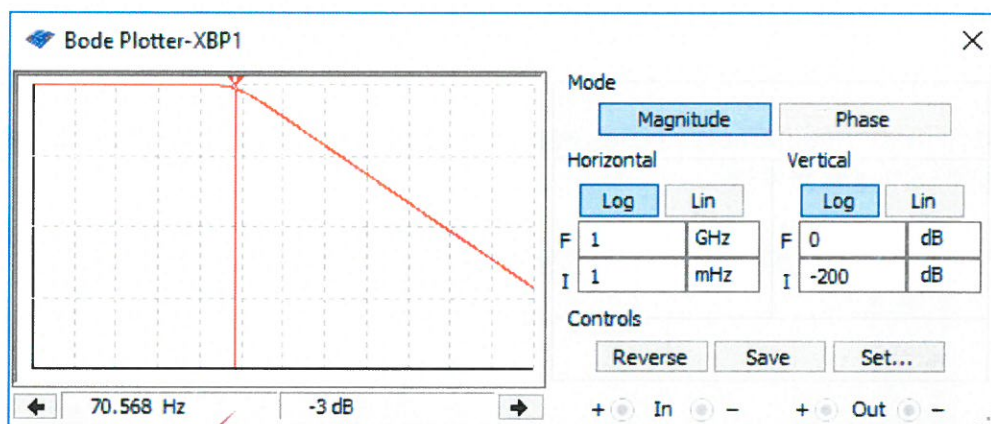
The frequency response can also be represented as a Bode plot. This includes a plot against magnitude and phase.



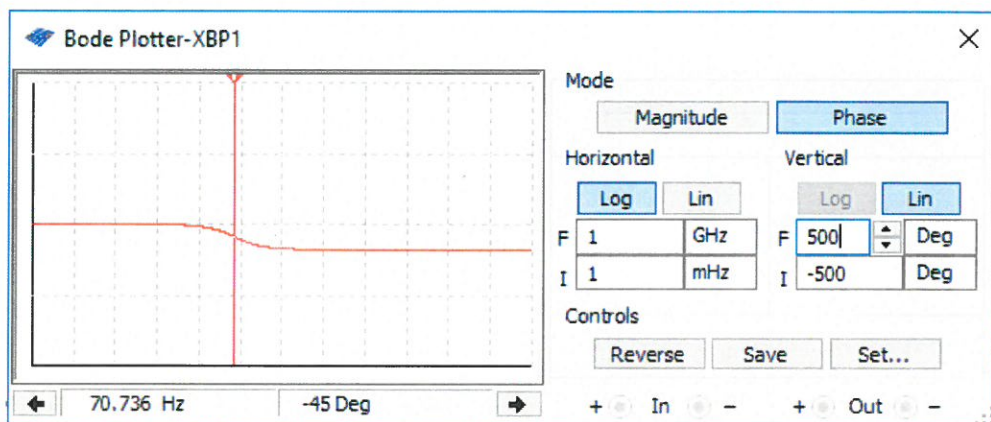
The circuit was constructed in Multisim to confirm the results.



Magnitude plot



Phase plot



Good.

The Multisim plots confirm that the results are correct from the Matlab program that was created.

Task 2- DC Motor Model

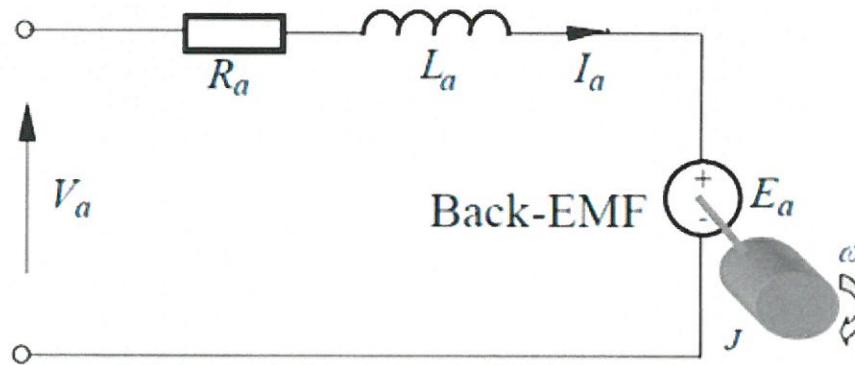
The following figure shows the equivalent circuit for an armature voltage controlled DC motor.

Motor Parameters

$$R_a = 1.4\Omega$$

$$V_a = 12V$$

$$T_s = 0.21453Nm$$



a)

The motor constant (K_M) was determined by creating a Matlab program.

```
% DC Motor program, Assignment 2 Task 2      by Sean Beynon
```

```
% Motor specification values
```

```
Ra=1.4; Va=12; Ts=0.214;
```

```
% Motor constant Calculation
```

```
km=(Ts*Ra)/Va;
```

```
% Speed Range
```

```
W=0:1.5:650;
```

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

```
% Display of results
```

```
disp('The Motor Constant, km')
```

```
disp(km)
```

```
plot(T,W,'B-')
```

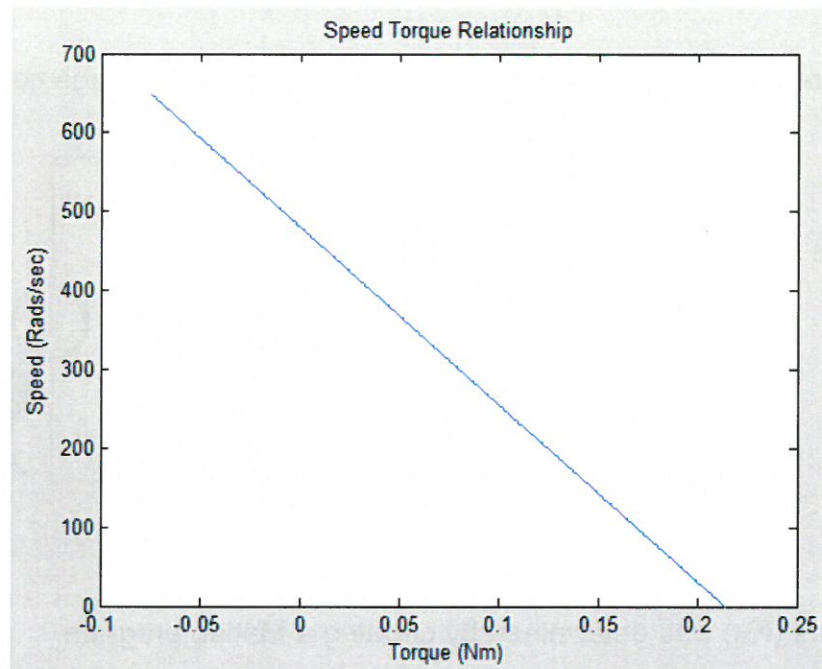
```
xlabel('Torque (Nm)')
```

```
ylabel('Speed (Rads/sec)')
```

```
title('Speed Torque Relationship')
```

Motor Constant = 0.0250

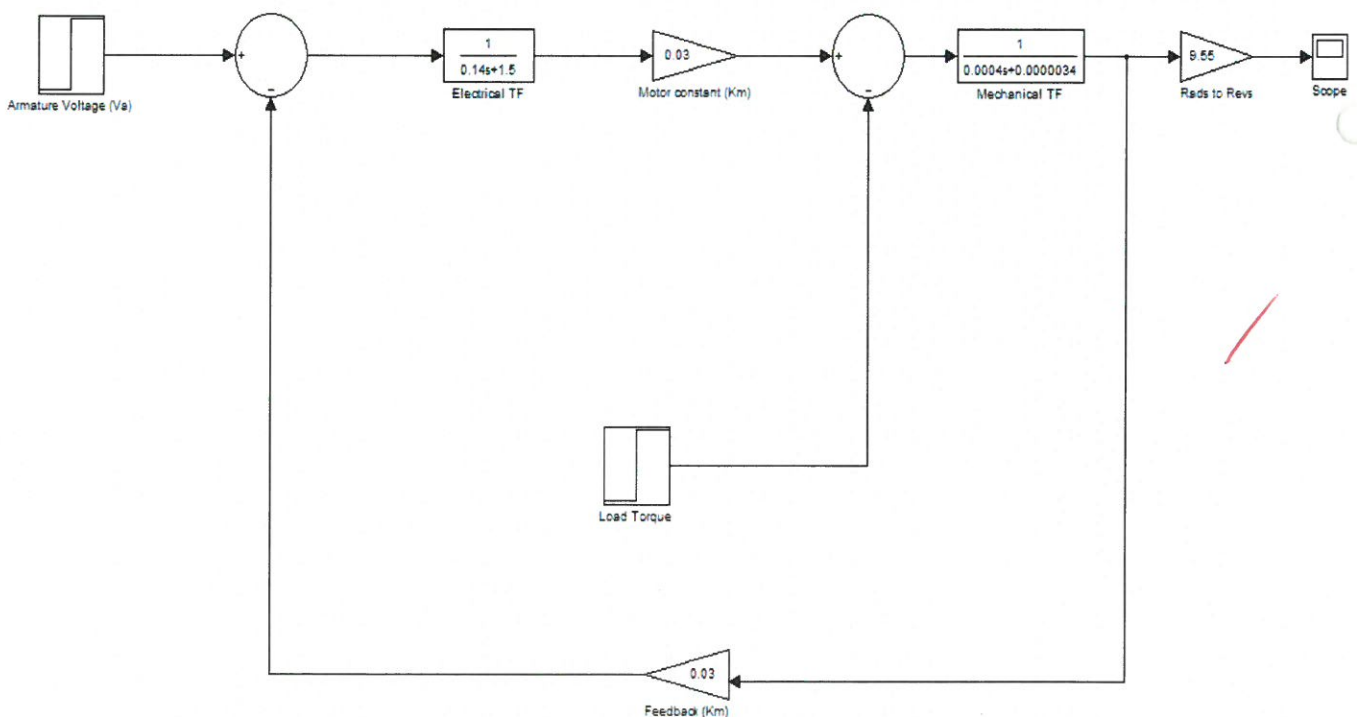
The program was also created to determine the relationship between speed and torque.



b)

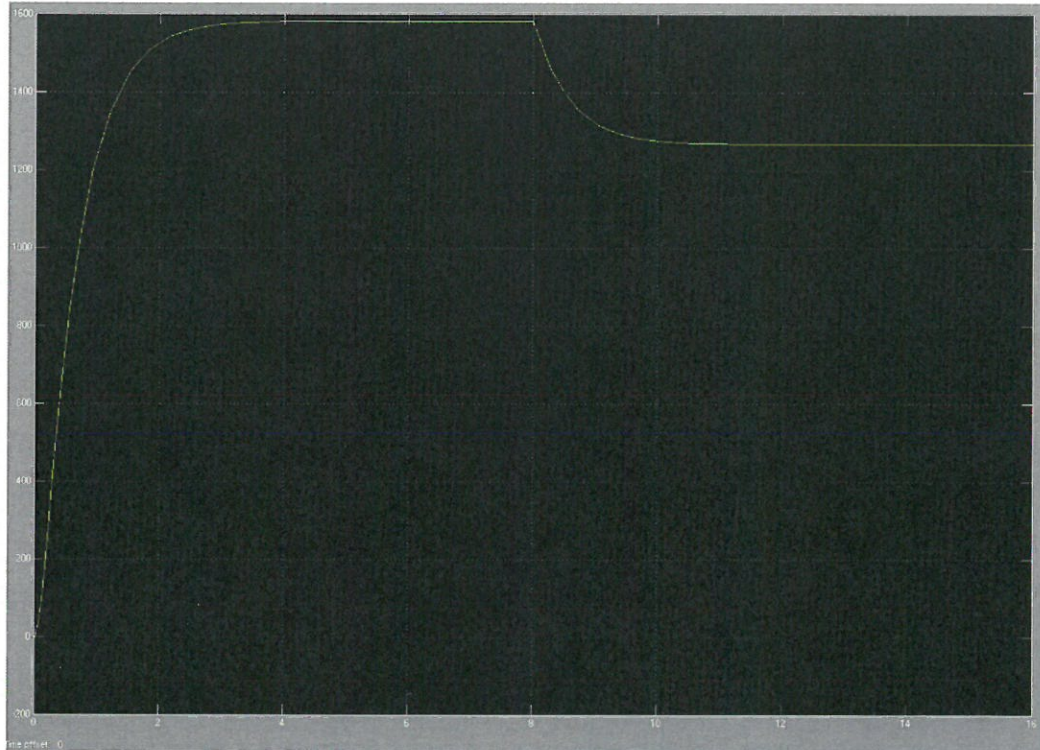
The Simulink model below shows an armature voltage controlled DC motor.

Before completing this a conversion from rad/sec to rev/minis required = $\omega \frac{60}{2\pi} = 9.55\omega$



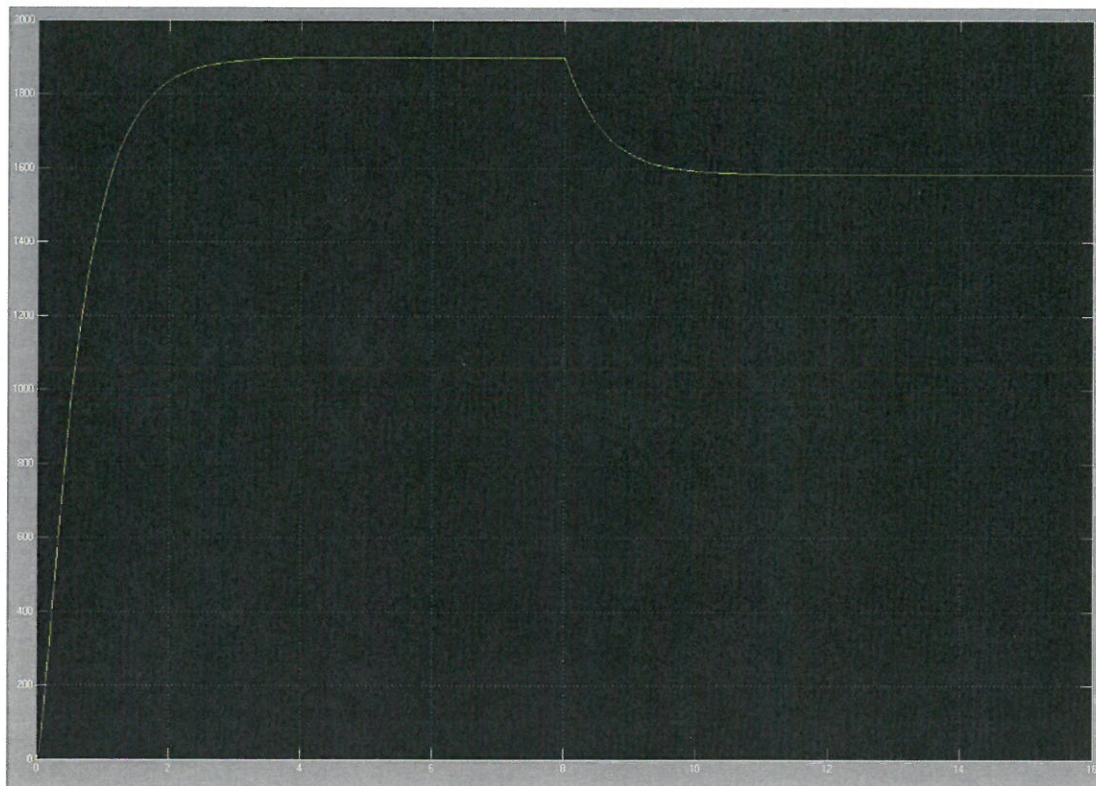
c)

Speed response plot when the load changes between 0.02Nm and 0.04Nm.

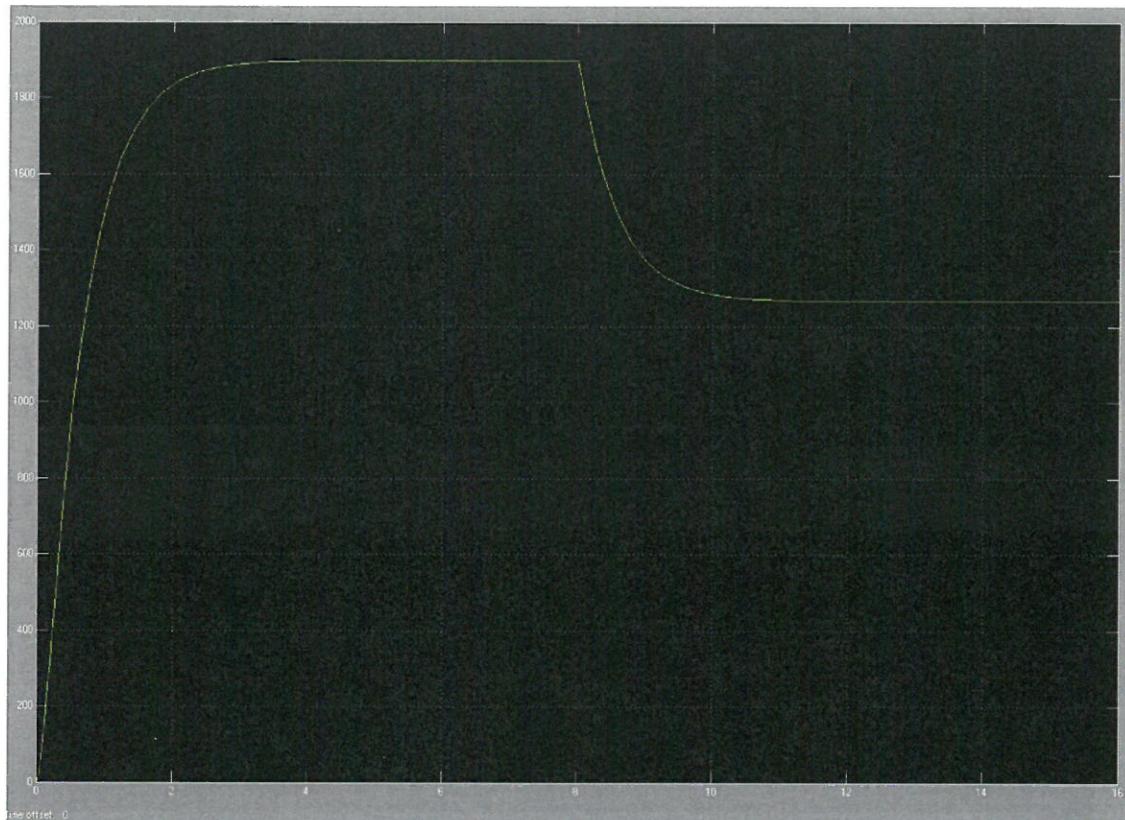


d)

Steady state plot at a load of 0.02Nm

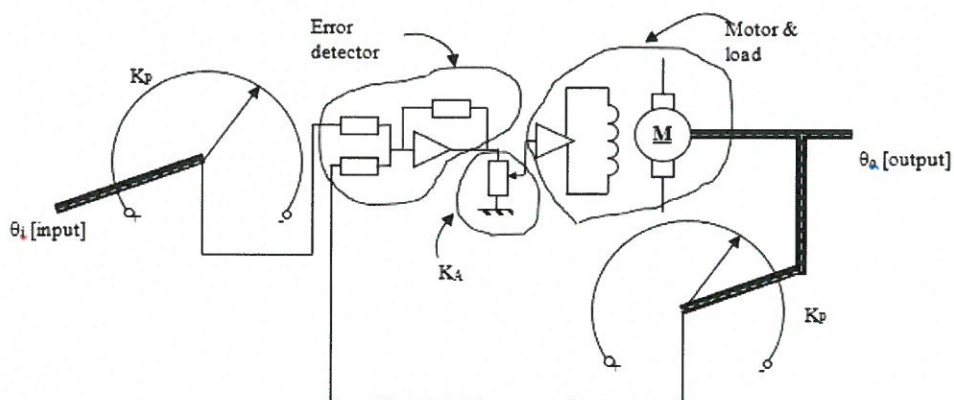


Steady state plot at a load of 0.04Nm

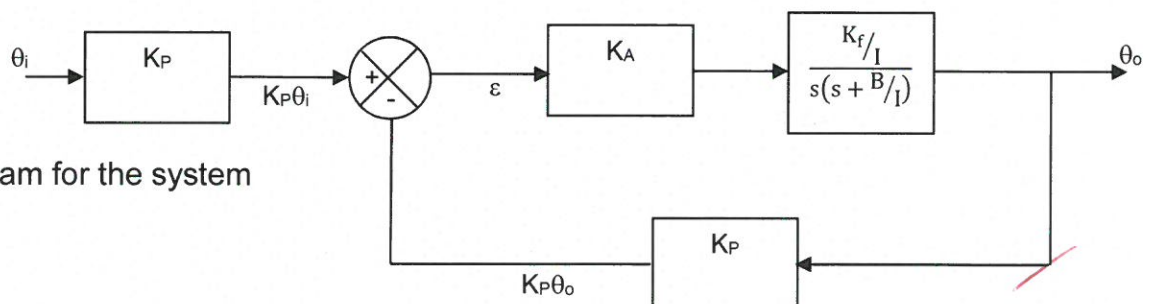


Need to state the steady state value.

Task 3- Position Control Servo



a)



Block Diagram for the system

b)

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

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

$$H(s) = K_P$$

The overall Transfer function

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

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

Multiply the top and bottom by $(s^2 + \frac{Bs}{I})$

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

The transfer function

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

c)

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

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

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

$$\omega_n^2 = 200$$

$$\text{Steady state value} = k_{ss} = 1$$

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

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

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

$$B = \frac{28.28}{5000} = 5.656 \times 10^{-3} \text{ N/ms}^{-1}$$

The following calculation is to determine the poles for critically damped.

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

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

The Matlab program is created to demonstrate that the system is critically damped.

This provides access to pole plot, highlighting the system stability.

```
% define the numerator
```

```
num=[200];
```

```
% define the denominator
```

```
den=[1 28.28 200];
```

```
% Transfer Function
```

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

```
% Poles
```

```
pole(pc)
```

```
% Plotting the Poles
```

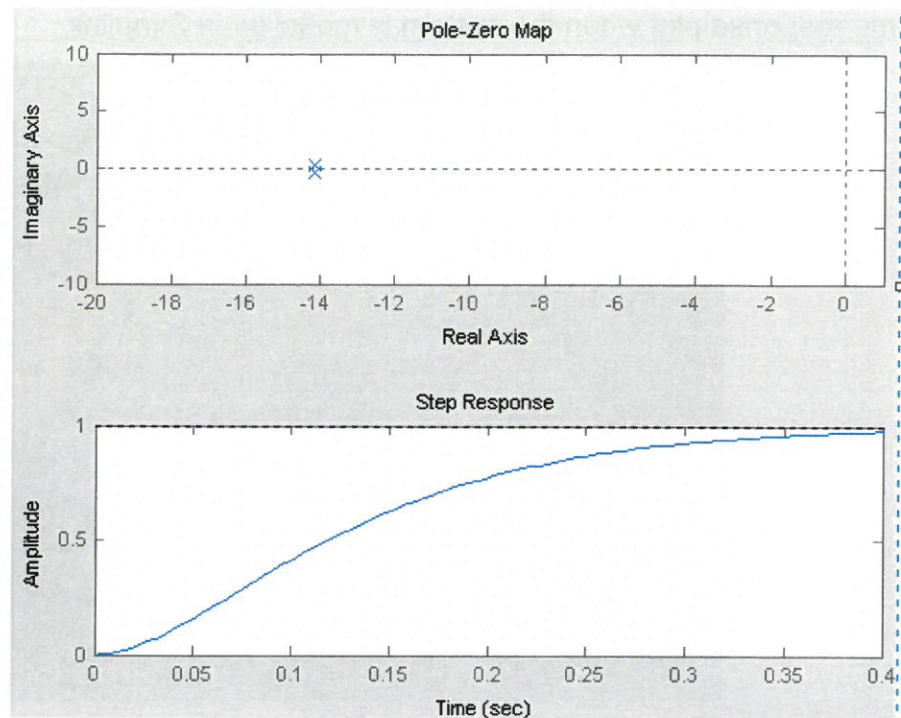
```
subplot(211), pzmap(pc)
```

```
axis([-20 1 -16 16])
```

```
% Plotting the response for an unit step
```

```
subplot(212), step(pc)
```

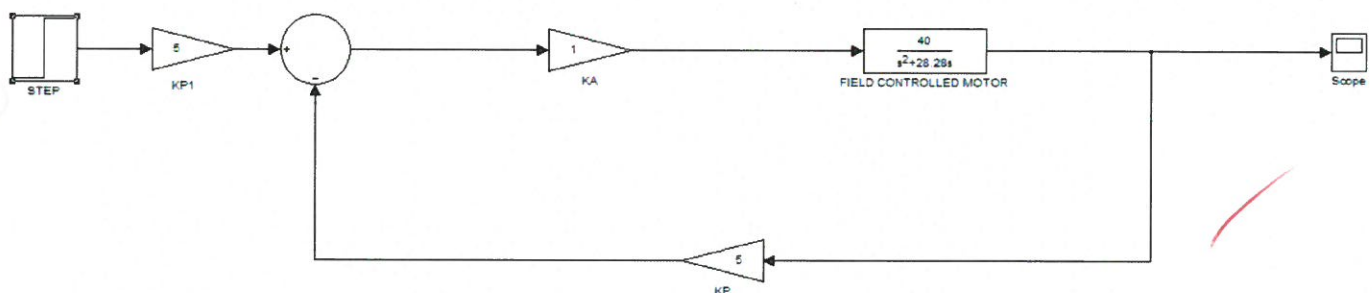
Pole plot showing the system is critically damped



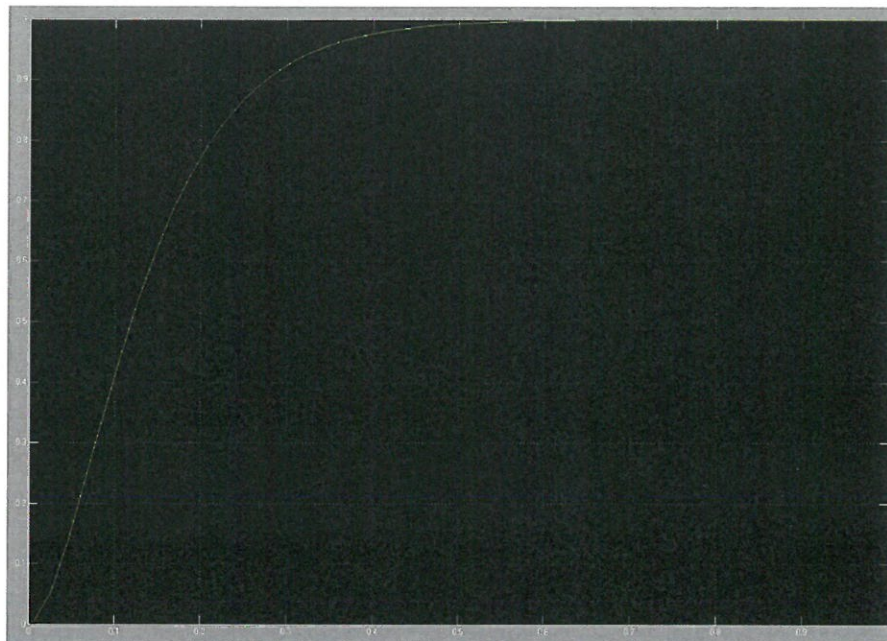
Could have shown more of the time response.

d)

The Matlab program was modelled in Simulink to compare its performance when determining the time response of the system.



Time response plot when the system is modelled in Simulink.



e)

The Matlab program was modelled to allow different values of B to be entered.

```
%Determine the poles and time response by Sean Beynon

% Variable B input
B=input('Enter friction value, B');

% Calculate B/I
y=B/2*10^-4;

% define the numerator
num=[200];

% denominator
den=[1 y 200];

% Transfer Function
pc=tf(num,den);

% Poles
pole(pole(pc))

% Plotting the Poles
subplot(211), pzmap(pc)
axis([-20 1 -16 16])

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

The Values for B that were entered are as followed:

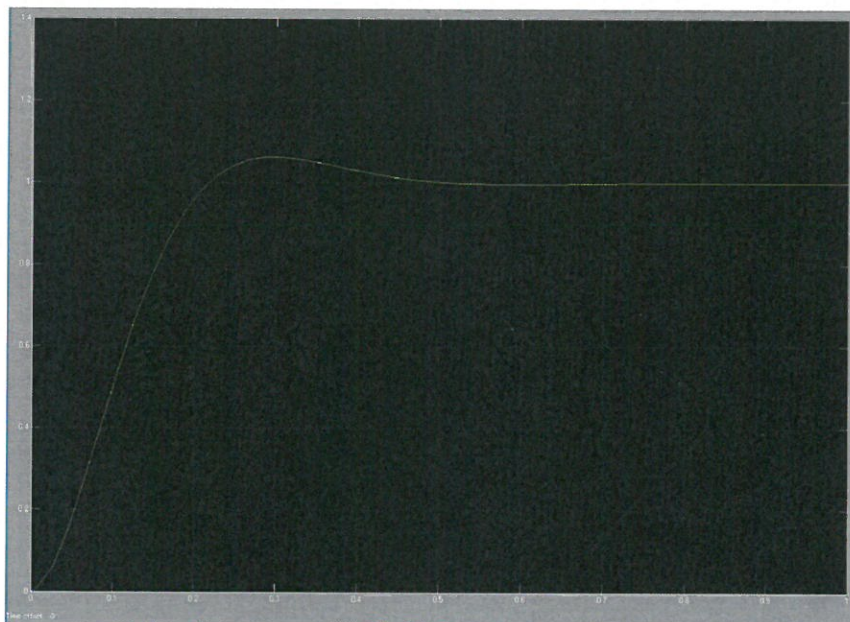
Critically damp - $B = 5.656 \times 10^{-3}$

No damping - $B = 0$

Underdamped - $B = 4.75 \times 10^{-3}$ SO: $B/I = 3.75 \times 10^{-3} / 2 \times 10^{-4} = 18.75$

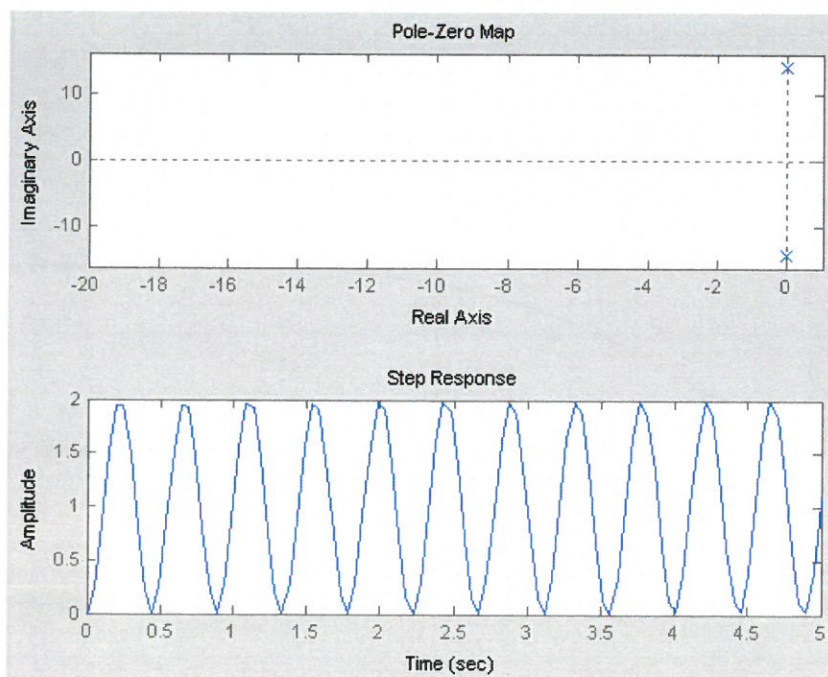
Overdamped - $B = 8.4 \times 10^{-3}$ So: $B/I = 8.4 \times 10^{-3} / 2 \times 10^{-4} = 42$

The following plot is for a 'No Damping' system.

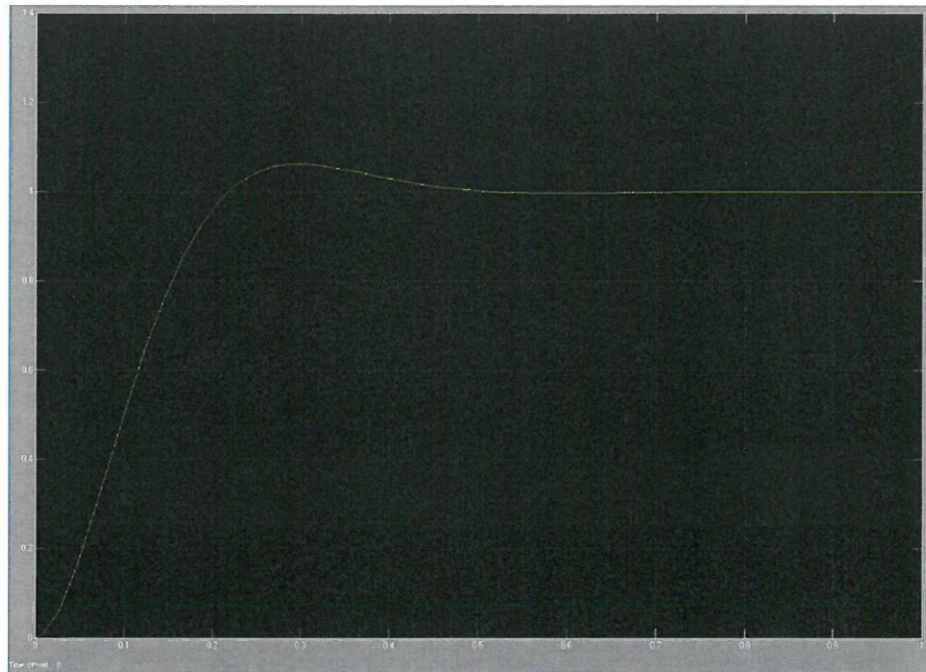


Wrong response, this is under damped.

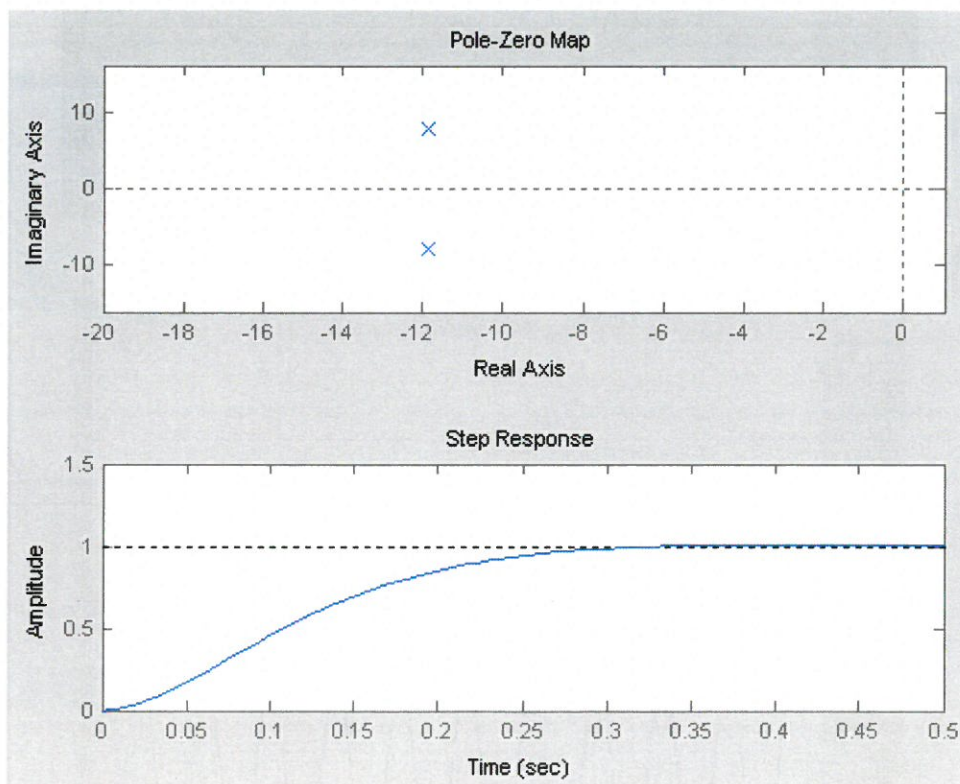
Time response was checked using Simulink.



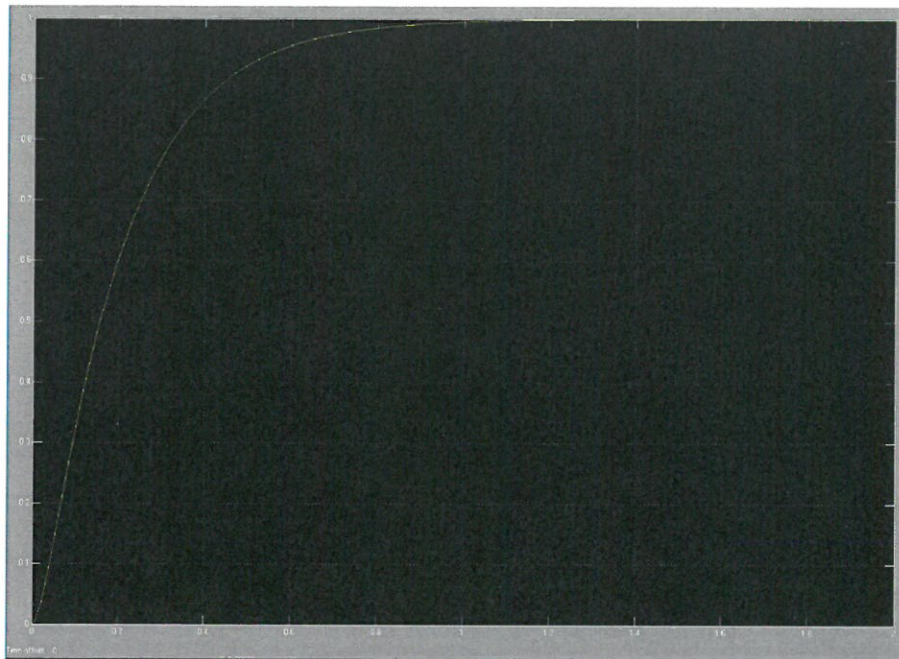
The following plot is for an 'Underdamped' system.



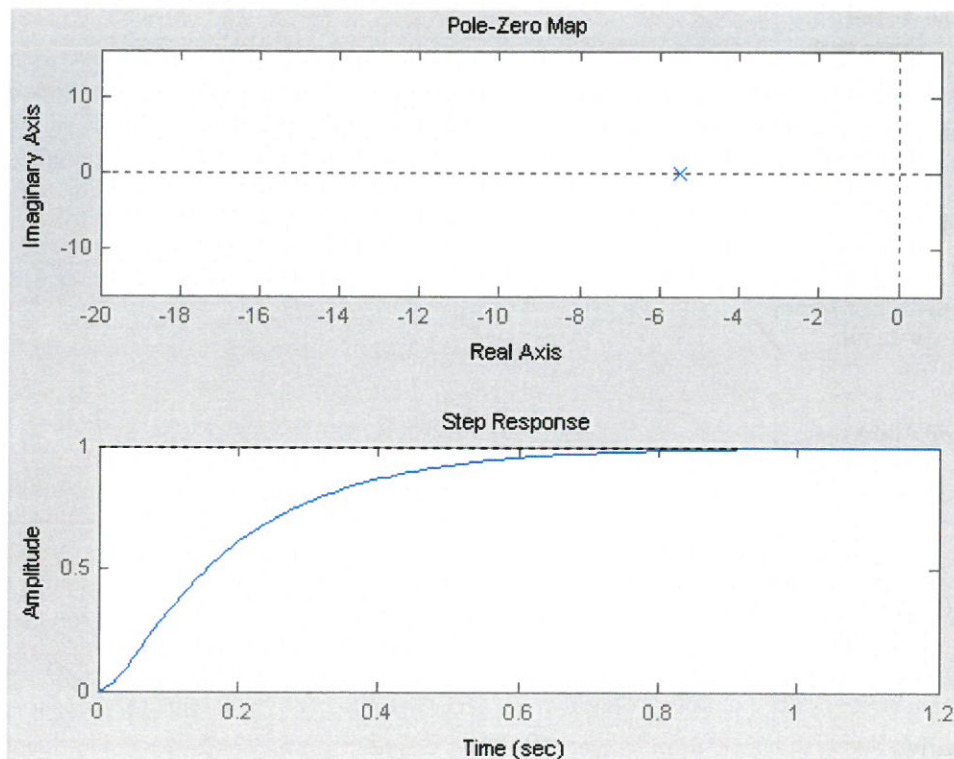
Time response was checked using Simulink.



The following plot is for an 'overdamped' system.



Time response was checked using Simulink.



If the real axis was expanded the other pole would be seen.

f)

The Matlab program was adapted for the under damped condition to obtain the rise time, peak time, percentage maximum overshoot and settling time. Below is the program developed:

```
%Determine the poles and time response by Sean Beynon

% Variable B input
B=input('Enter friction value, B')

% Calculate B/I
y=B/(2*10^-4);

% define the numerator
num=[200];
wn=sqrt(200);

% denominator
den=[1 y 200];

% Transfer Function
pc=tf(num,den);

% Poles
pole(pc)

% Plotting the Poles
subplot(211), pzmap(pc)
axis([-20 1 -16 16])

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

% Determining zeta
z=y/(2*wn);

% Damped natural frequency
wd=wn*sqrt(1-(z^2));

% Rise Time
tr=(2.16*z+0.6)/(wn);

% Peak Time
tp=pi/wd;

% Percentage of max overshoot
mp=100*exp(-(pi*z)/sqrt(1-z^2));

% Settling Time
ts=4/(z*wn);

% Display Results
disp('Rise Time')
disp(tr)
disp('Peak Time')
disp(tp)
disp('Percentage max overshoot')
disp(mp)
disp('Settling Time')
disp(ts)
```

Good.

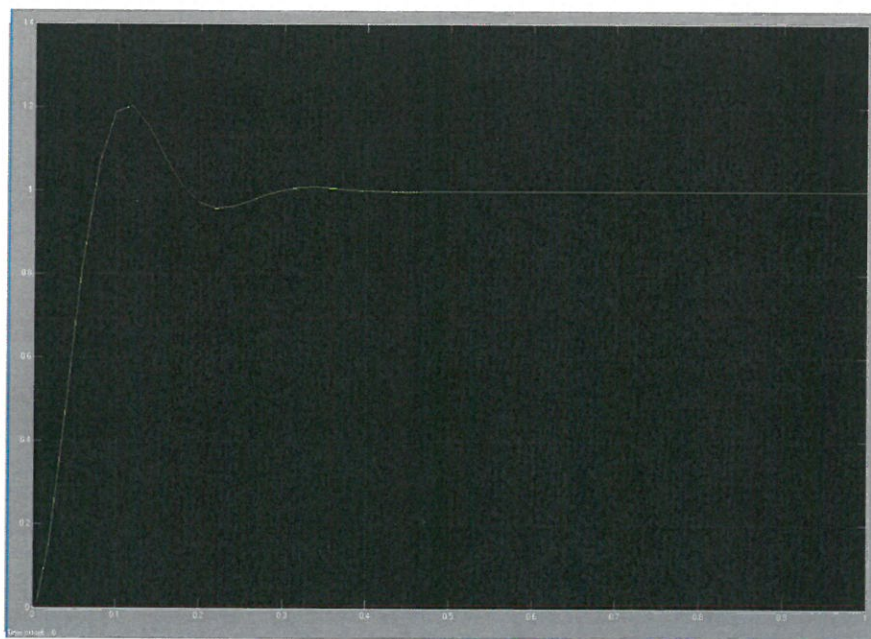
Rise Time = 170.7 ms

Peak Time = 409.1 ms

Percentage max overshoot = 77.7%

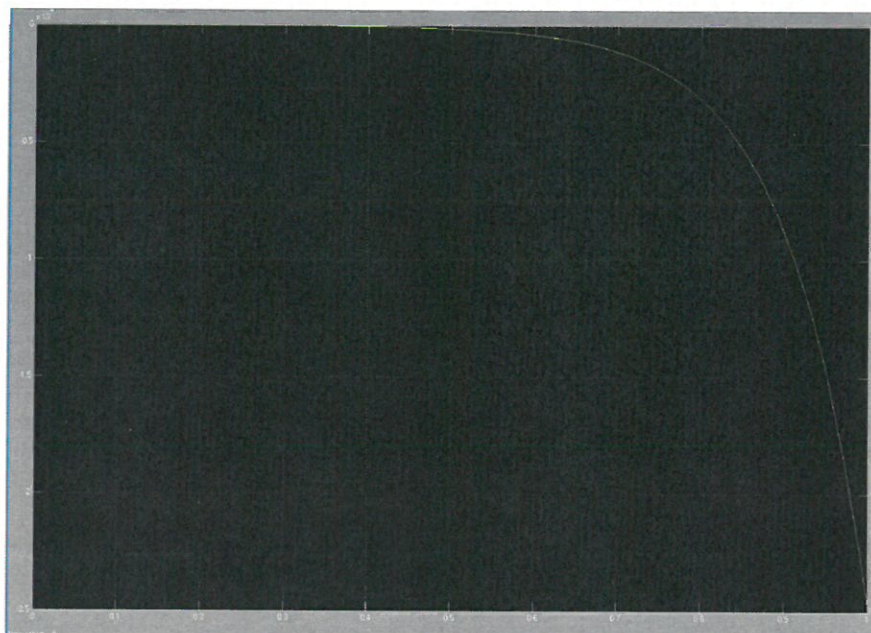
Settling Time = 336.8 ms

The proportional controller was increased to 5, to understand what effect it has to the performance of the system.



The proportional controller was also decreased to -2, to understand what effect it has to the performance of the system.

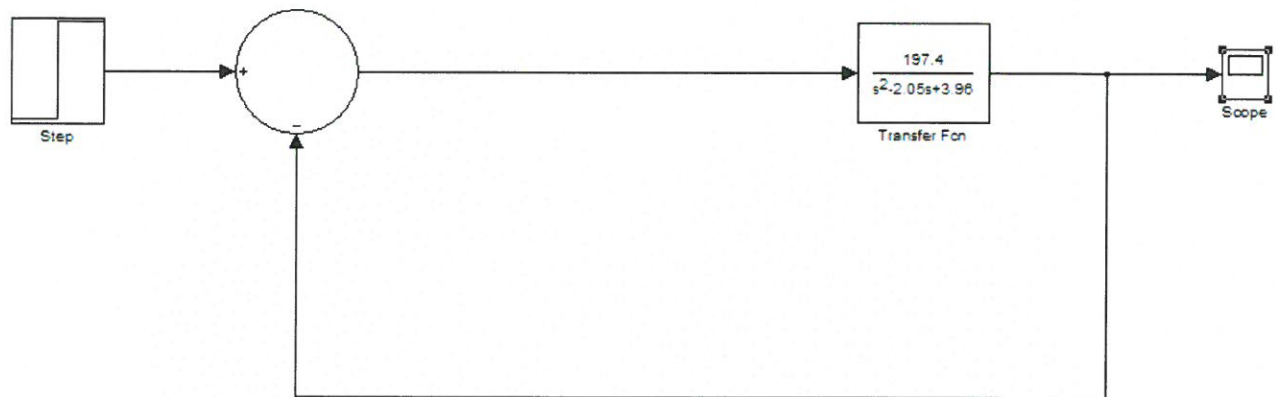
↖ No.



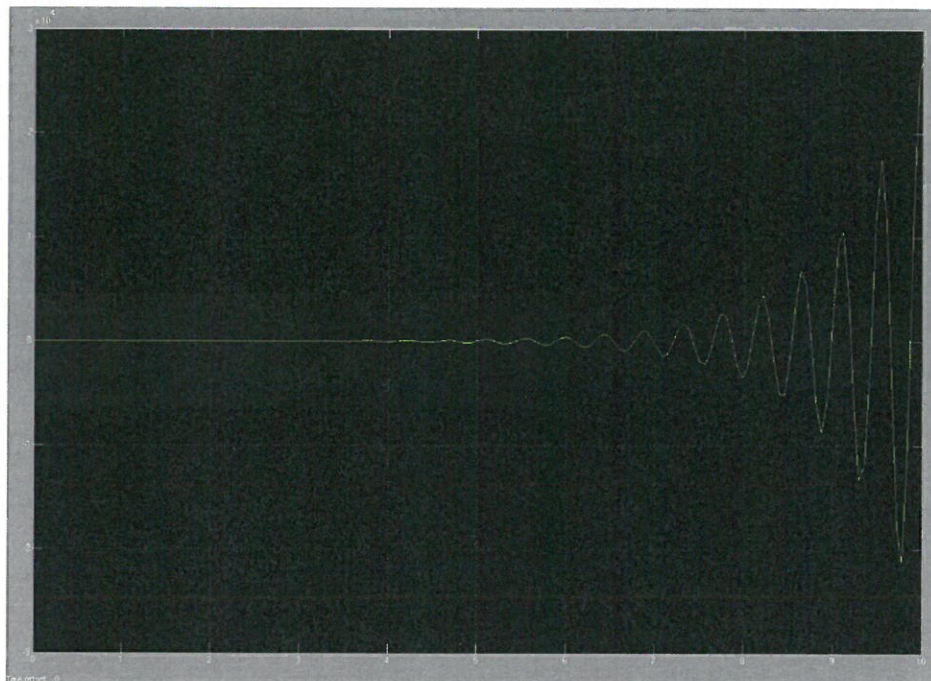
Task 4

An electrical system was modelled in Simulink by

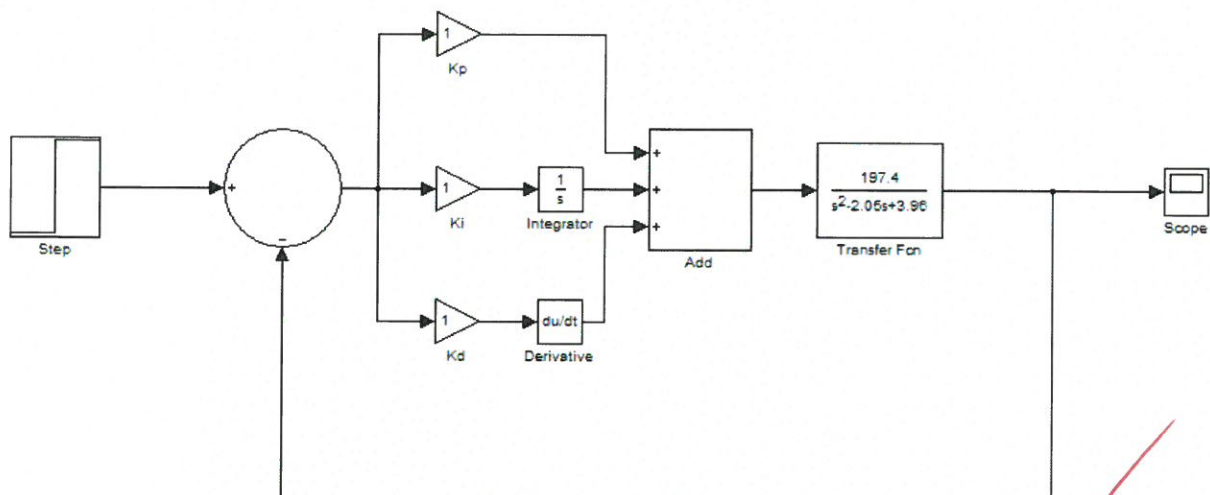
$$G(s) = \frac{197.54}{s^2 - 2.05s + 3.96}$$



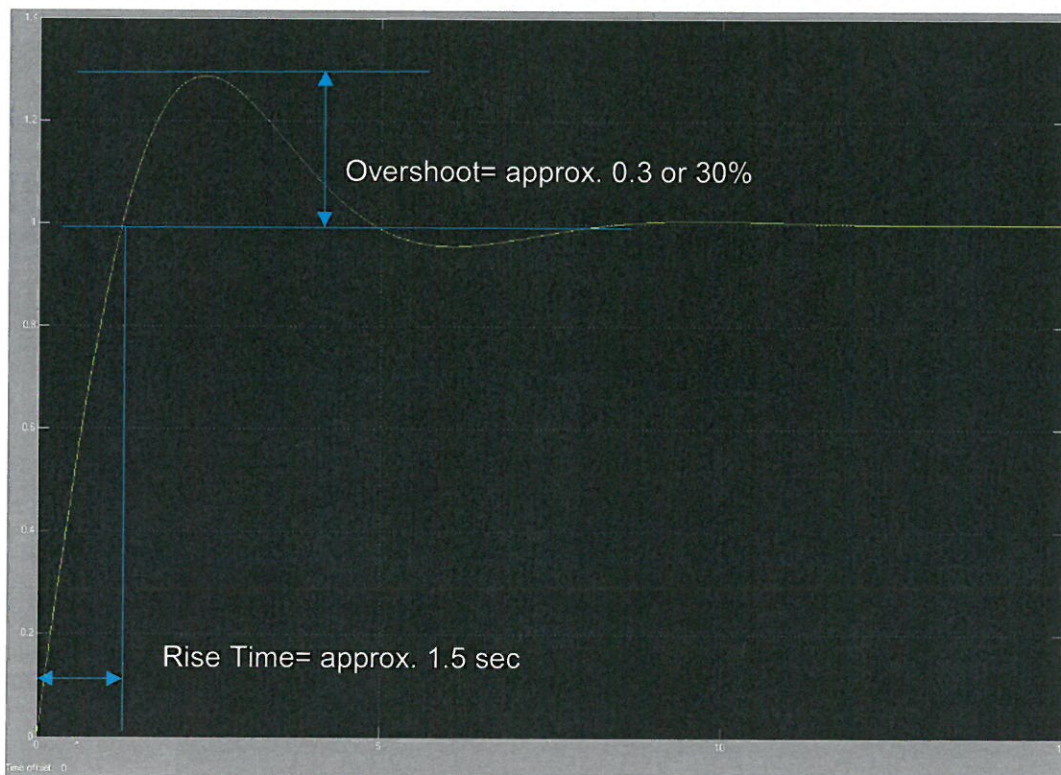
This model provides a totally unstable system.



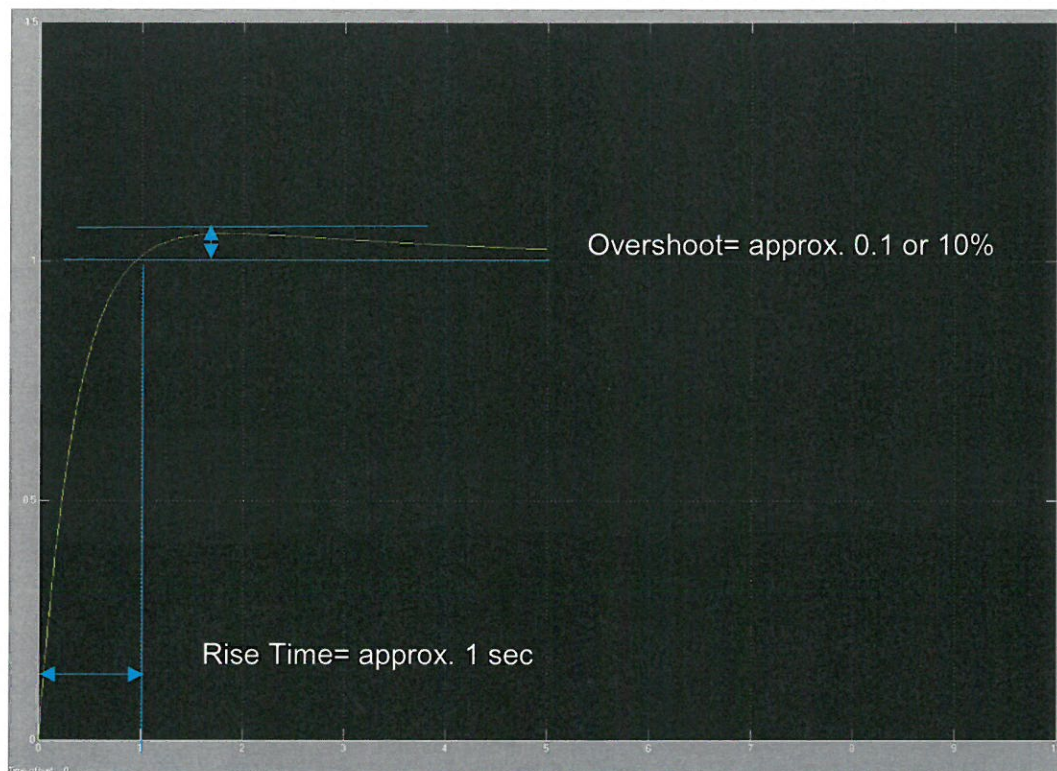
The model below shows the PID controller in Simulink.



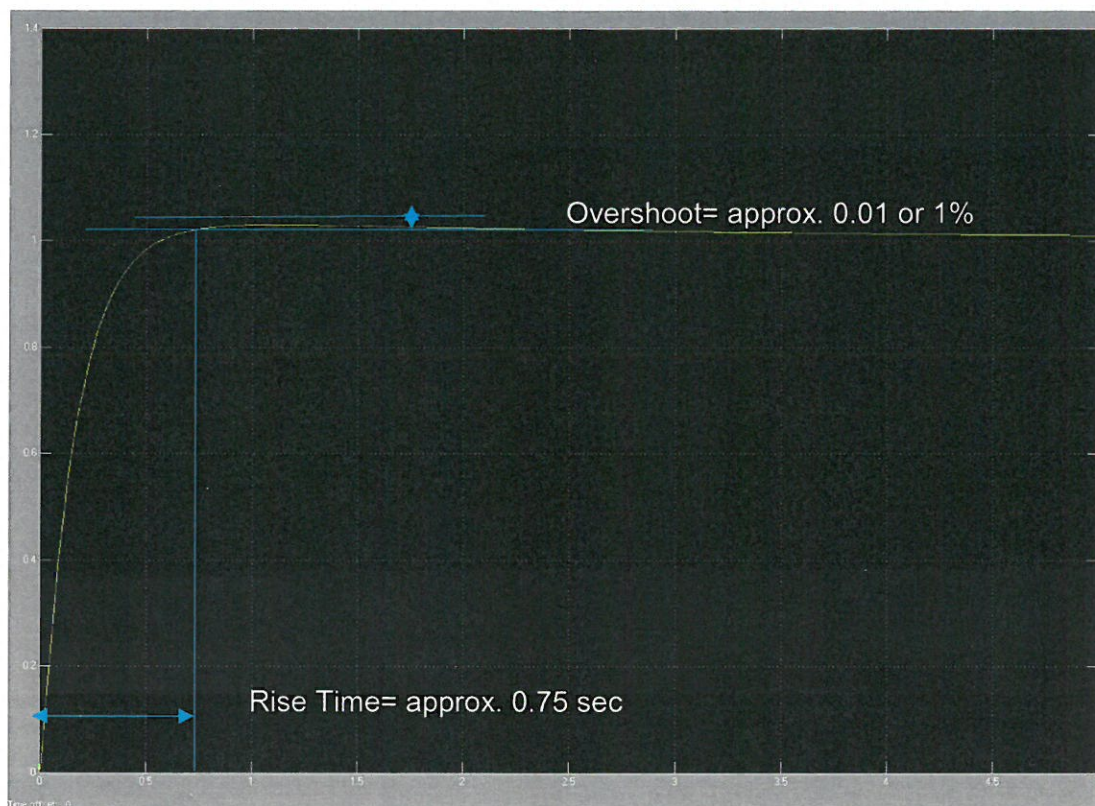
The parameters were set as follows: $K_p = 3$, $K_i = 0.8$, $K_d = 1$



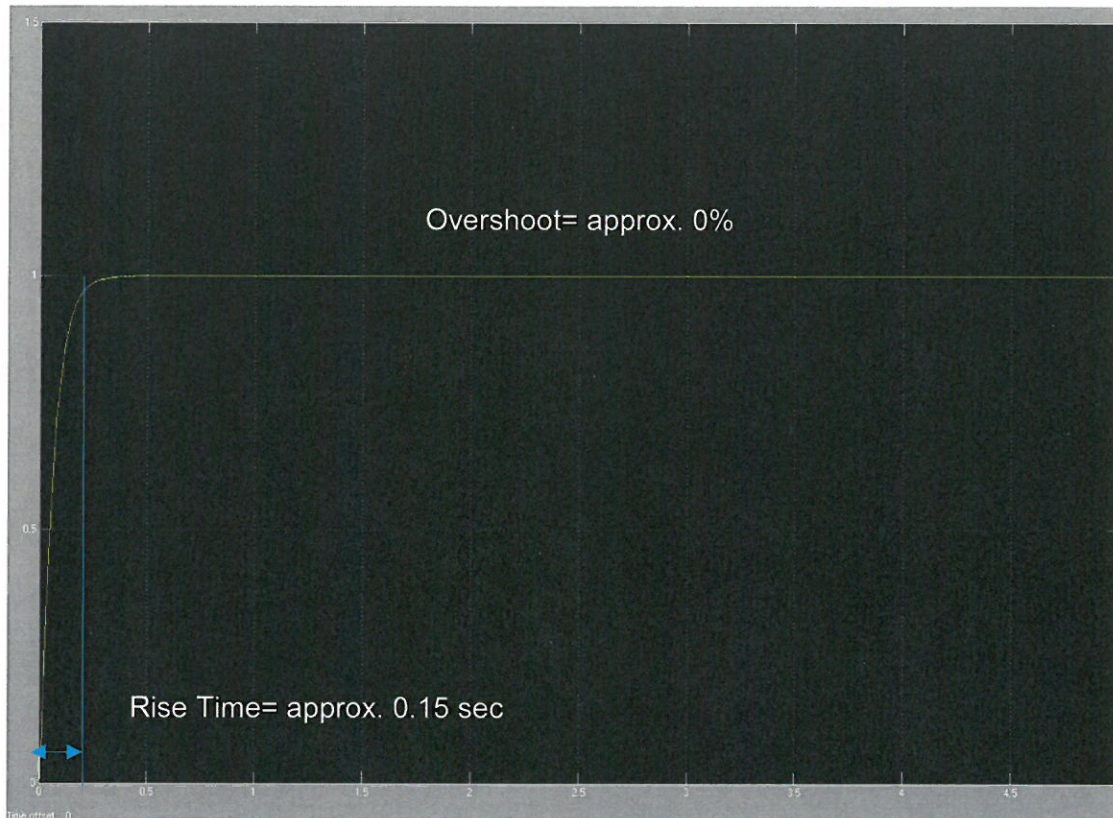
The parameters were set as follows: $K_p = 3$, $K_i = 0.8$, $K_d = 0.8$



The parameters were set as follows: $K_p = 3$, $K_i = 0.8$, $K_d = 0.5$



The parameters were set as follows: $K_p = 8$, $K_i = 0.3$, $K_d = 0.5$ ✓



Good work, clearly shows the tuning process.

Task 5

a)

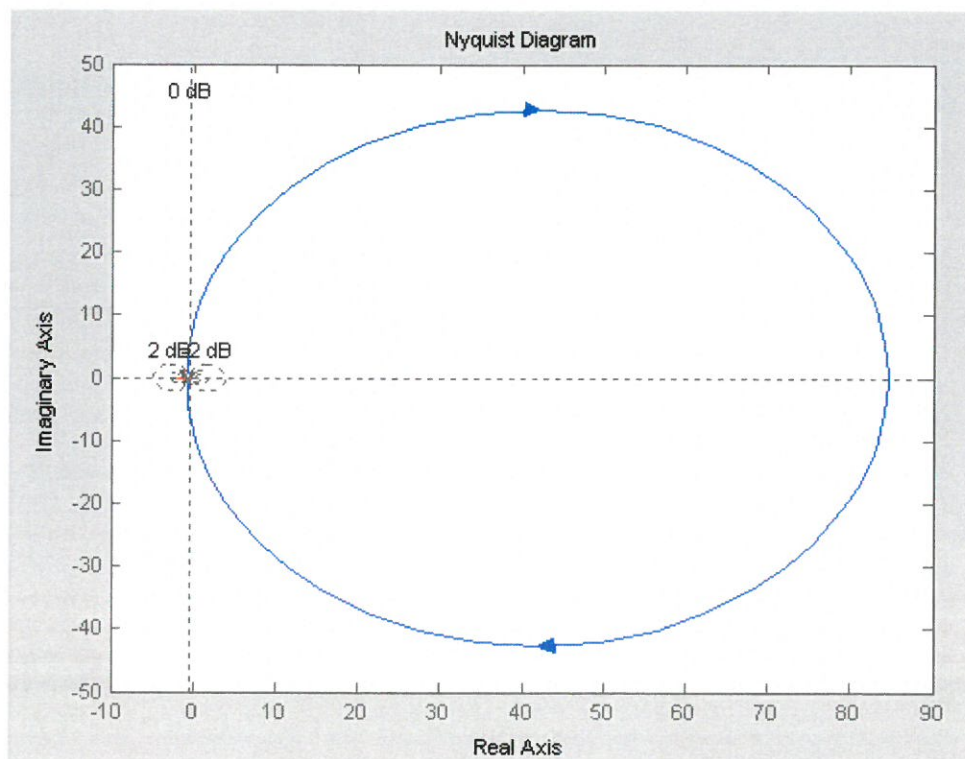
A Matlab program was developed, using the open loop transfer function below, to plot a Nyquist diagram of a servo control rig.

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

The simple program below enables a Nyquist diagram to be developed.

```
% Nyquist by Sean Beynon 2/05/19.  
  
x=[0.03868 -5.369 -97.34 13450];  
y=[1 169.8 1730 159];  
  
sys=tf([x],[y])    % transfer function  
  
nyquist(sys)       % Nyquist response  
  
grid
```

The following Nyquist diagram is created using the open loop transfer function provided in the question.

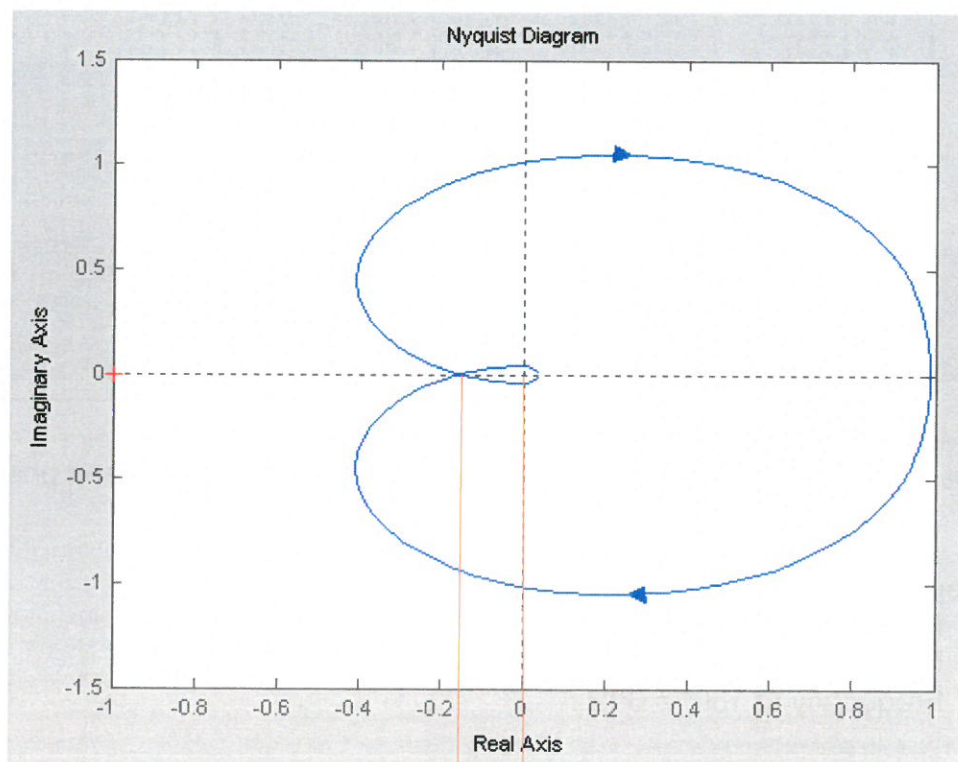


b & c)

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

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

```
1 % Nyquist part b) by Sean Beynon 2/05/19.
2
3 - x=[0.03868 -5.369 -97.34 13450];
4 - y=[1.03868 164.431 1632.66 13609];
5
6 - sys=tf([x],[y]) % transfer function
7
8 - nyquist(sys) % Nyquist response
9
10 - pause
11
12 - bode(sys) % Bode plot
13 - grid
14
15 |
```



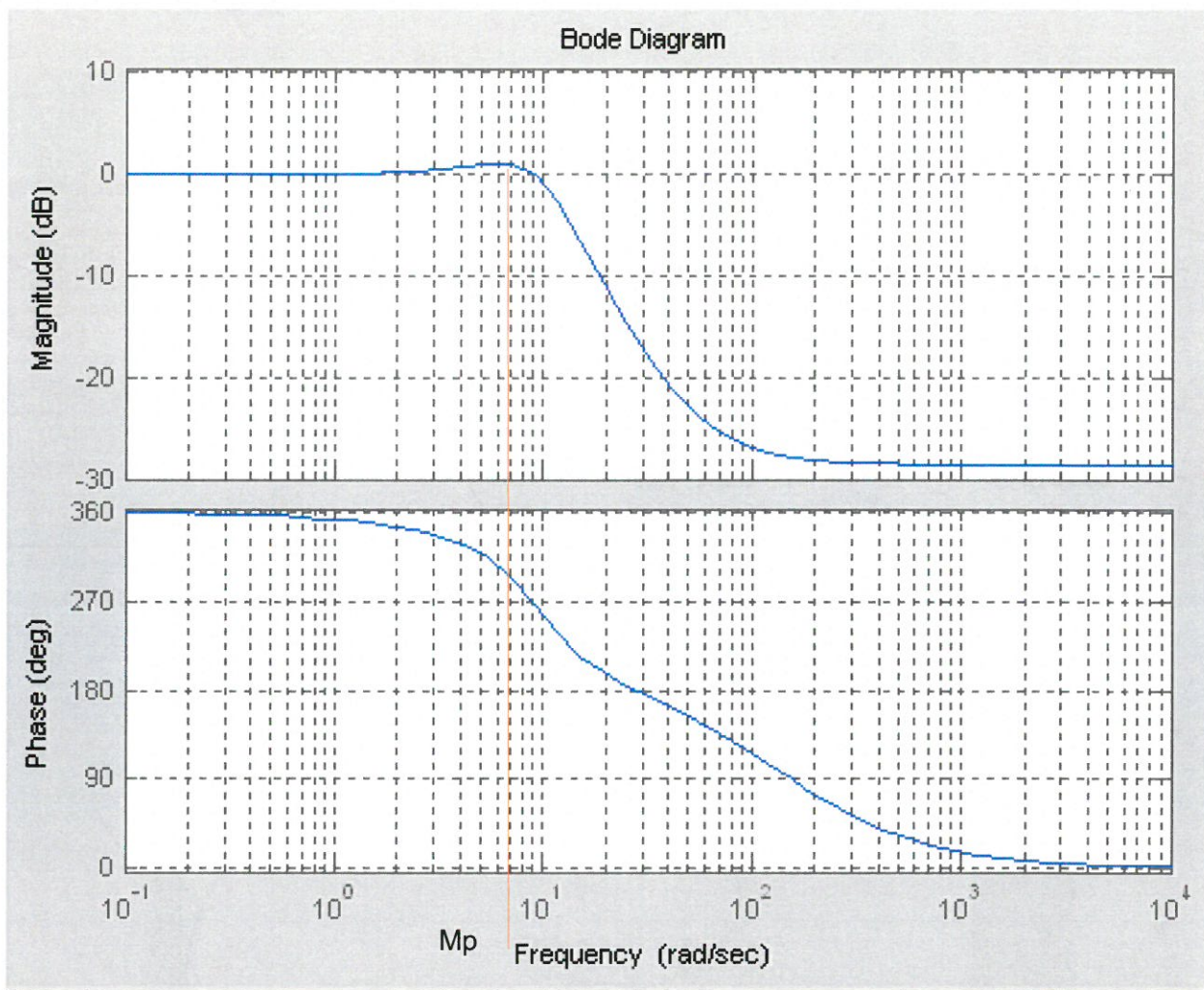
Crosses the real axis= -0.17

Yes the system is stable as it's in the negative real value, although the phase margin is infinite because the gain is always 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 occurs at the point of maximum magnitude.

Bandwidth ω_b is the frequency between when the magnitude cross 0dB and phase cross 180° .

For this system:

$$\text{Resonance } M_p = 7\text{Hz}$$

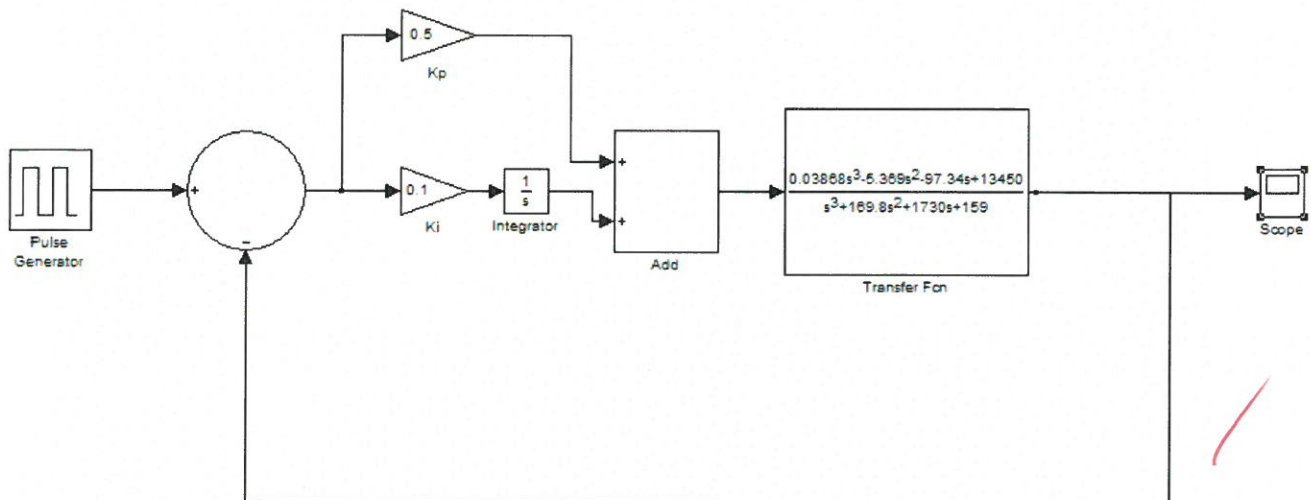
$$\text{Frequency at } 0\text{dB} = 9\text{Hz}$$

$$\text{Frequency at } 180^\circ = 30\text{Hz}$$

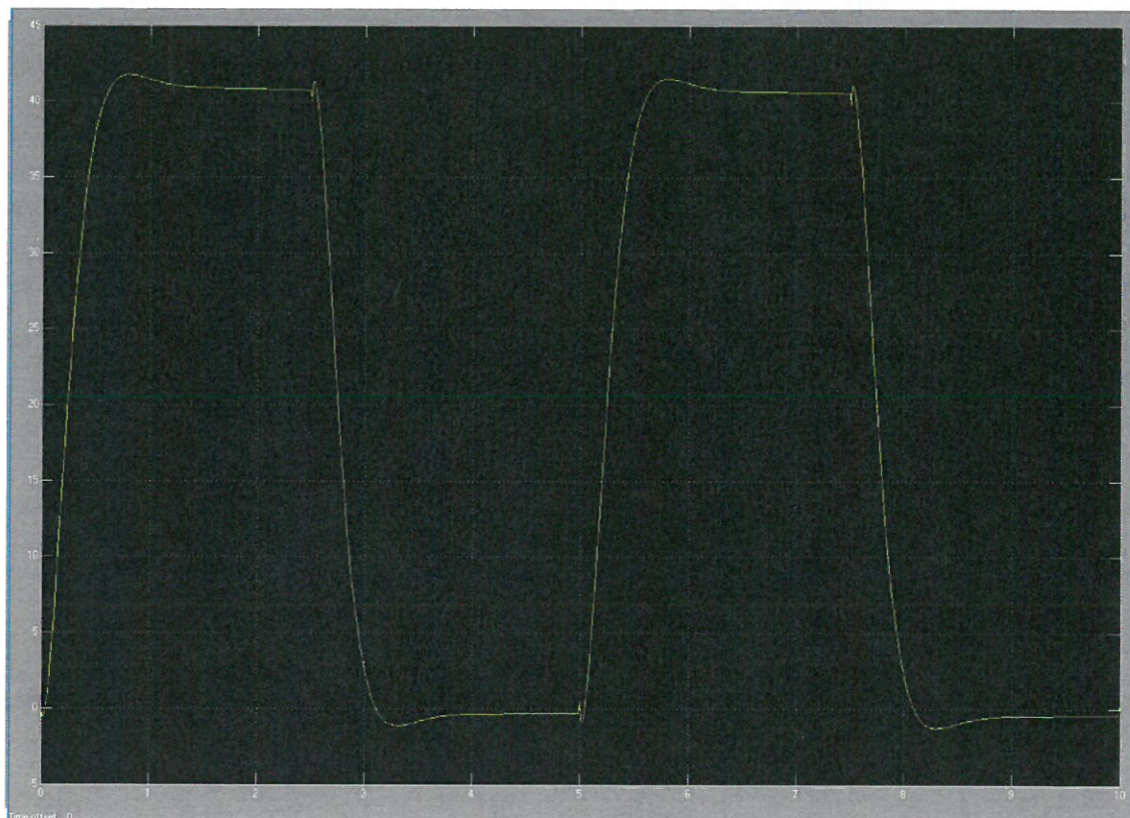
$$\omega_b = 30 - 9 = 21\text{Hz}$$

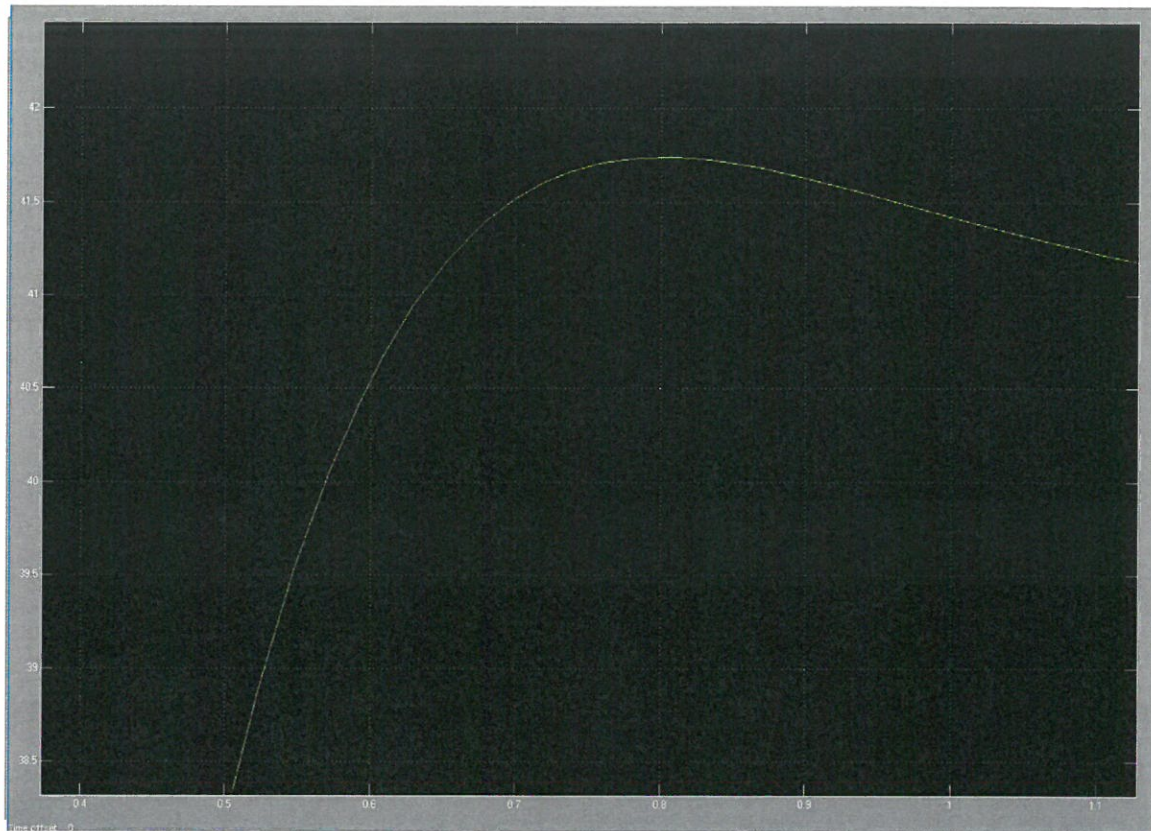
d)

Below is the system model with a PI controller gains and initial gains, which was created in Simulink under task specifications.



The response curve was plotted for the first 10 seconds and provides a square wave form.



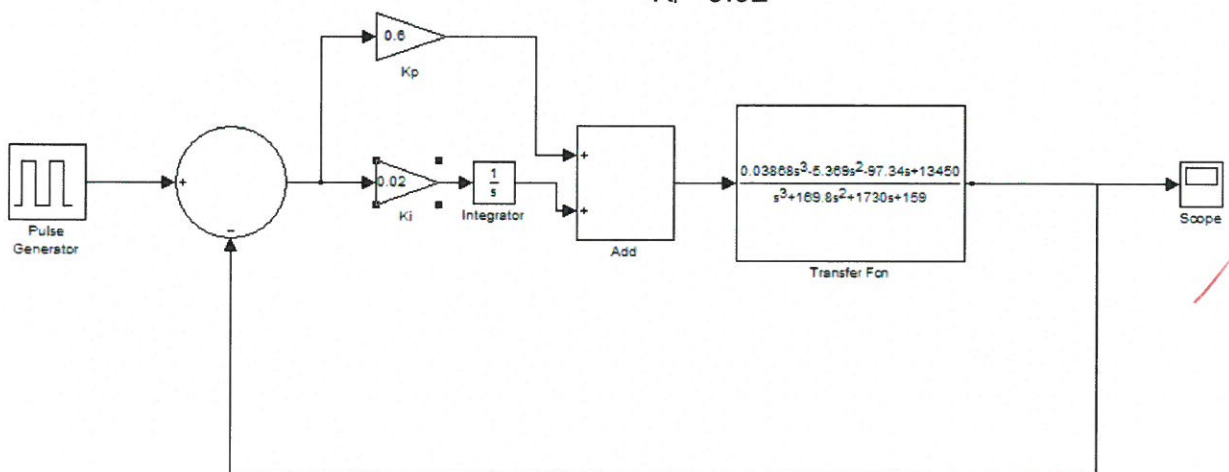


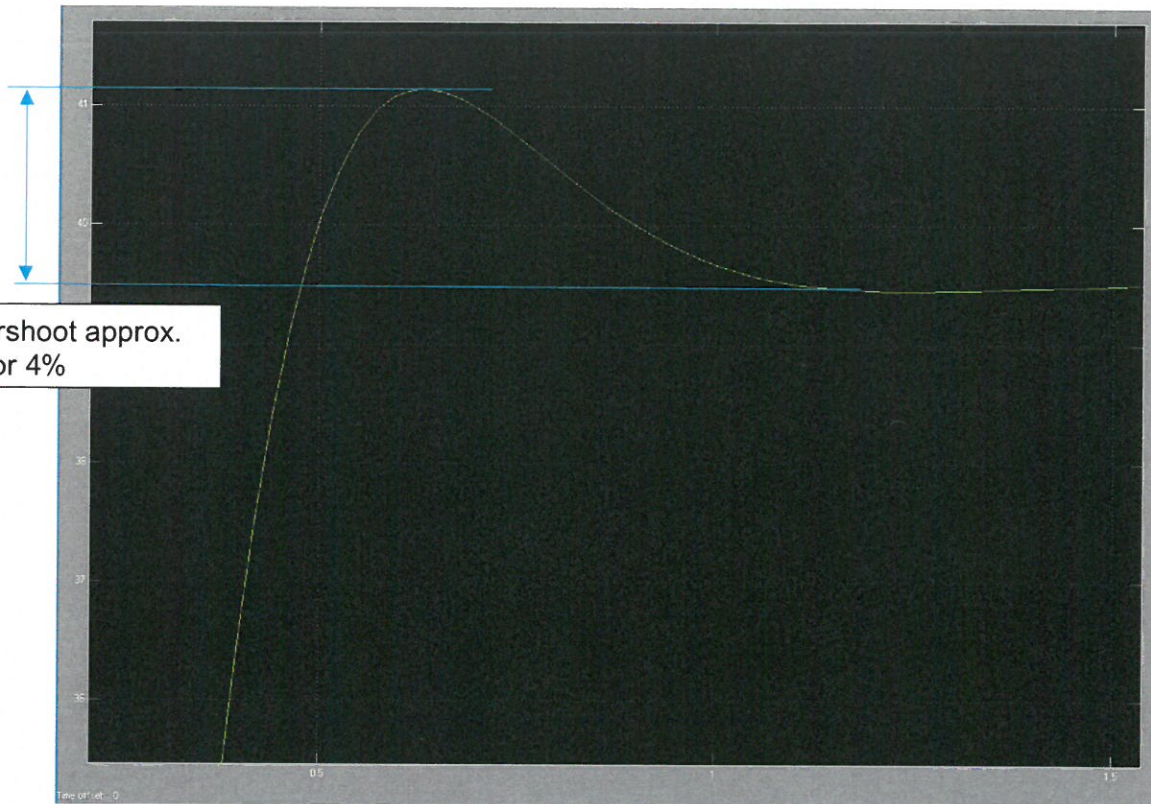
Manual changes were made to the PI controller's gains in the servo control rig to ensure the closed loop system can achieve overshoot $\sigma < 5\%$ and the settling time $t_s < 1.0 \text{ sec}$.

The following changes were made to the system:

$$K_p - 0.6$$

$$K_i - 0.02$$





What about the settling time?

