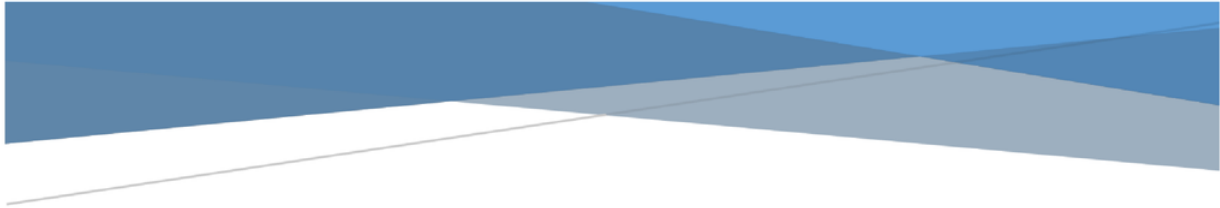


Assignment 2

by Declan Davies

FILE	11279_DECLAN_DAVIES_ASSIGNMENT_2_158265_1725029893.DOCX		
TIME SUBMITTED	11-MAY-2018 11:09PM (UTC+0100)	WORD COUNT	4425
SUBMISSION ID	89030834	CHARACTER COUNT	28748



ANALYSIS, SIMULATION AND EVALUATION OF CONTROL SYSTEMS

Assignment 2

Davies, Declan (YST)
[Email address]

Contents

1. First Order Control System	2
2. DC Motor Model Analysis	10
3. Types Of Motor Speed Controller	17
3.1 P Controller	17
3.2 PI Controller	23
3.3 PID Controller	28
3.4 Comparison Of P, PI and PID Controllers	32

1. First Order Control System

The first step in the process is to use the general solution that has been showed in the diagram below:-

This is not a general solution, $G(S)$ is the open loop transfer function

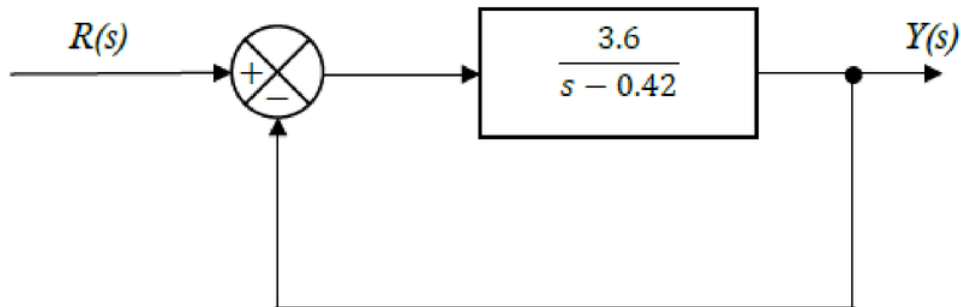


Figure 1 - First Order Control System

The values that need to be used for calculations throughout the process can be seen in the figures below:-

$$G(S) = \frac{3.6}{S - 0.42}$$

Figure 2 - General Solution

$$H(S) = 1$$

Figure 3 - $H(S)$ Value

The values in figure 2 and figure 3 will be used and without feedback the open loop for this circuit will look like the following:-

$$S - 0.42 = 0$$

Figure 4 - S value

$$S = 0.42$$

Figure 5 - S Value

In figure 4 it shows the bottom part of the general solution from figure 1 the next step is to get the value of S on its own which can be seen in figure 5 shows the value of s is 0.42.

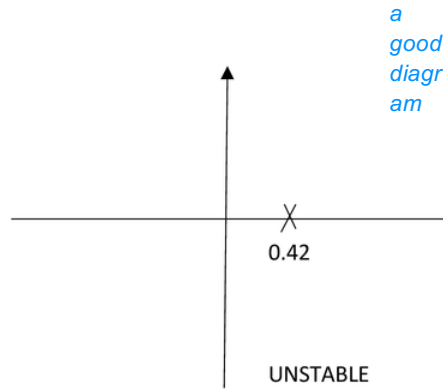


Figure 6 - Argand Diagram

In figure 6 it shows the argand diagram for the circuit with an open loop pole that it is unstable and the next step is to work out the value of S in an closed loop system.

$$\frac{G(S)}{1 + G(S)H(S)}$$

Figure 7 - First Step Work Out Feedback

In figure 7 it shows the formula that needs to be used in order to work out the value of S in a closed loop pole. In order to do this it means that the values from figure 2 and 3 need to be substituted into the formula above.

$$\frac{\frac{3.6}{(s - 0.42)}}{1 + \frac{3.6}{(s - 0.42)}} \times 1$$

Figure 8 - Formula With Values

After substituting the general solution and H values into the formula the next step in the process requires multiplying both the top and bottom by s - 0.42.

$$\frac{\frac{3.6}{(s - 0.42)} \times (s - 0.42)}{(s - 0.42) \left(1 + \frac{3.6}{(s - 0.42)} \right)}$$

Figure 9 - Multiply Top and Bottom

In figure 9 it shows that both the top and bottom will be multiplied by s-0.42 and on the top the general solution on the left also contains a s-0.42, which means they both cancel each other leaving 3.6 on the top line.

$$\frac{3.6}{s - 0.42 + 3.6}$$

Figure 10 - Take away - 0.42 from 3.6

The next step in the process will be to take away -0.42 from 3.6 which results in 3.18 and the result from this can be seen in figure 11.

$$\frac{3.6}{s + 3.18}$$

Figure 11 – Result

The next step in the process is to root the denominator of the circuit to calculate the value of s which will then allow me to draw the argand diagram to determine whether its stable or not.

$$s + 3.18 = 0$$

Figure 12 – Denominator

$$s = -3.18$$

Figure 13 - Calculated S Value

In figure 13 it shows the S value to be equal to -3.18 and now that I know this it can be plotted on an argand diagram to determine whether its stable.

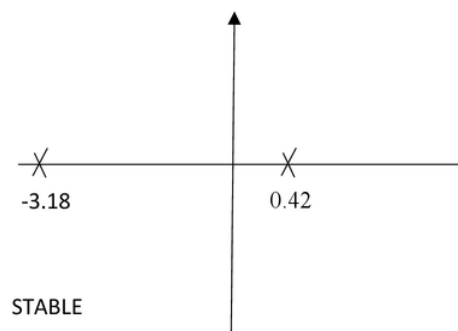


Figure 14 - Argand Diagram

In figure 14 it shows that the closed loop is stable meaning the next step in the process is to determine the minimum gain for the circuit.

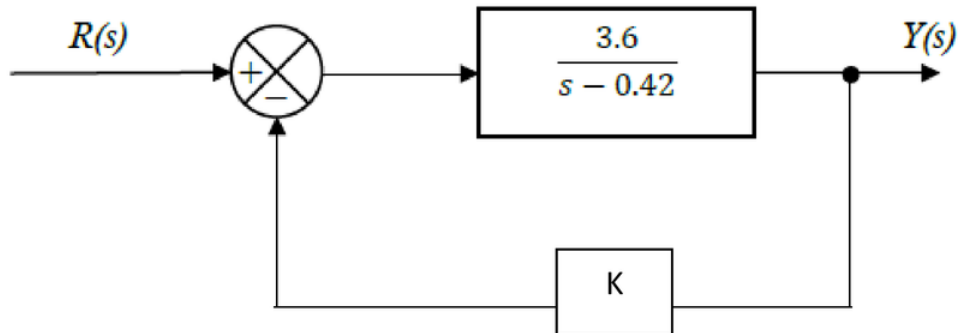


Figure 15 - Circuit With Feedback

The next step is to determine the value of k for the circuit in figure 15 that start by using the equation below as the value of K is equal to the value of H(S).

$$T(s) = \frac{\frac{3.6}{s - 0.42}}{1 + \frac{3.6k}{s - 0.42}}$$

Figure 16 – Formula

$$T(s) = \frac{3.6}{s - 0.42 + 3.6k}$$

Figure 17 - Formula 2

$$= \frac{3.6}{s + (3.6k - 0.42)}$$

Figure 18 - Formula 3

The next step is to find out the value of K when the value of S is equal to 0.

$$0 + 3.6k - 0.42 = 0$$

Figure 19 - After S=0

In figure 19 it shows the result when the value of s is equal to 0 that means the next step in the process is to get the value of K on its own that means -0.42 needs to be taken over the equal sign making it +0.42.

$$3.6k = 0.42$$

Figure 20 - Aim To Get K On It's Own

After that the value of 3.6 needs to be taken over making it -3.6 which will be used to divide the value of 0.42.

$$k = 0.116$$

Figure 21 - K Value

This shows that the value of K will be equal to 0.116 that means the minimum gain in the circuit for feedback will need to be 0.116. Following this the circuit needs to be simulated in matlab that can be seen below:-

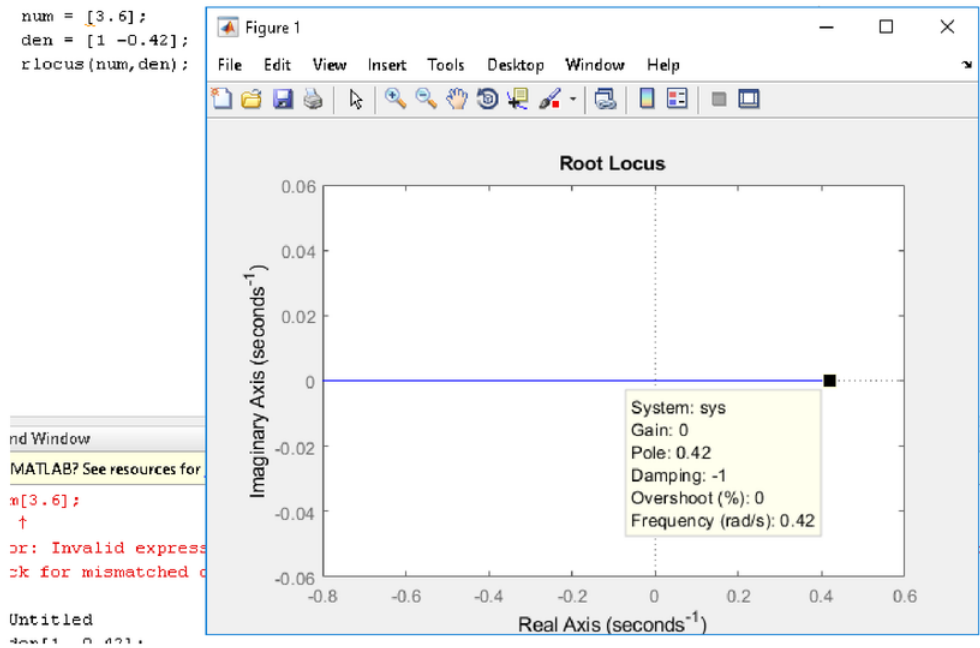
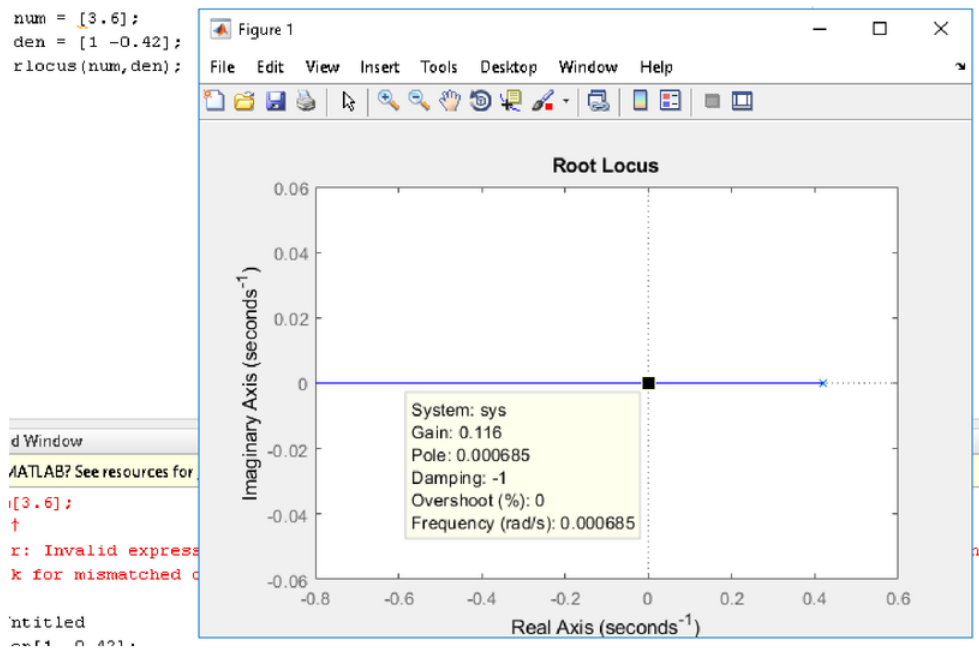
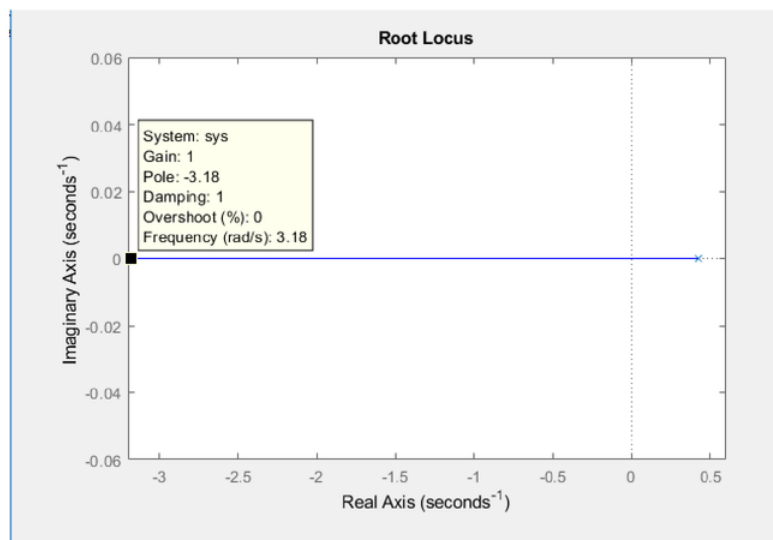


Figure 22 - Unstable Circuit

In figure 22 it shows the unstable side of the circuit matching the argand diagram from figure 6 that shows the theory has matched the simulated result. The next step in the process will be to find the stable argand diagram from the matlab circuit.



In figure 23 it shows the minimum gain for the circuit before it goes from being a stable circuit to an unstable circuit.



In figure 24 it shows the stable side of the circuit matching up with the argand diagram in figure 14 that shows the simulation is matching the calculations so far. The next step in the process is to build the circuit in Simulink and display the results of the graphs based upon the feedback entered.

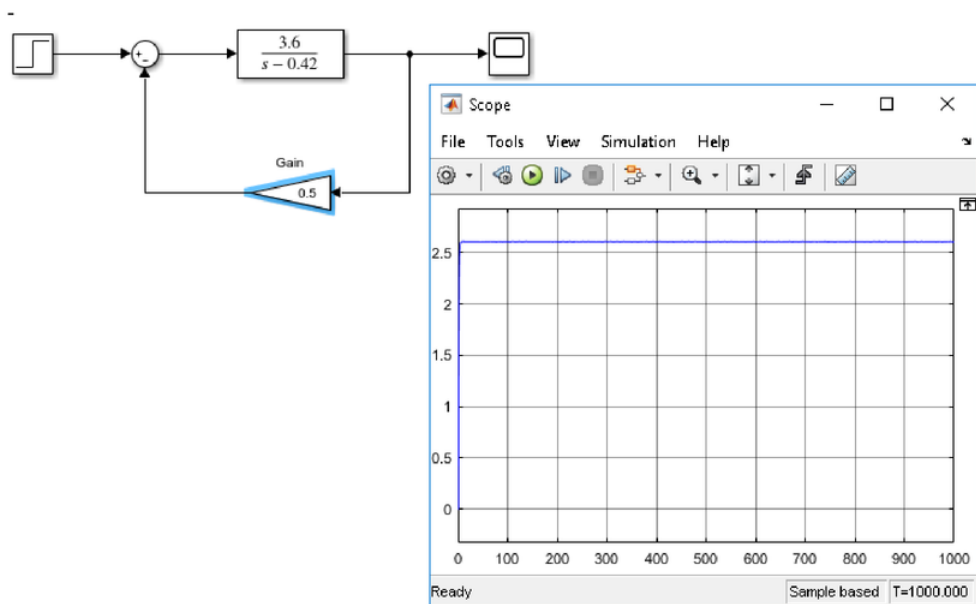


Figure 25 - Simulink Simulation Gain = 0.5

In figure it shows the circuit designed with a feedback loop where in this case the gain was equal to 0.5 which was a lot higher than the minimum required of 0.116 and as a result of this the graph showed a sharp quick response as a result. The next graph will show the circuit when the response is slower and the feedback is closer to the minimum required.

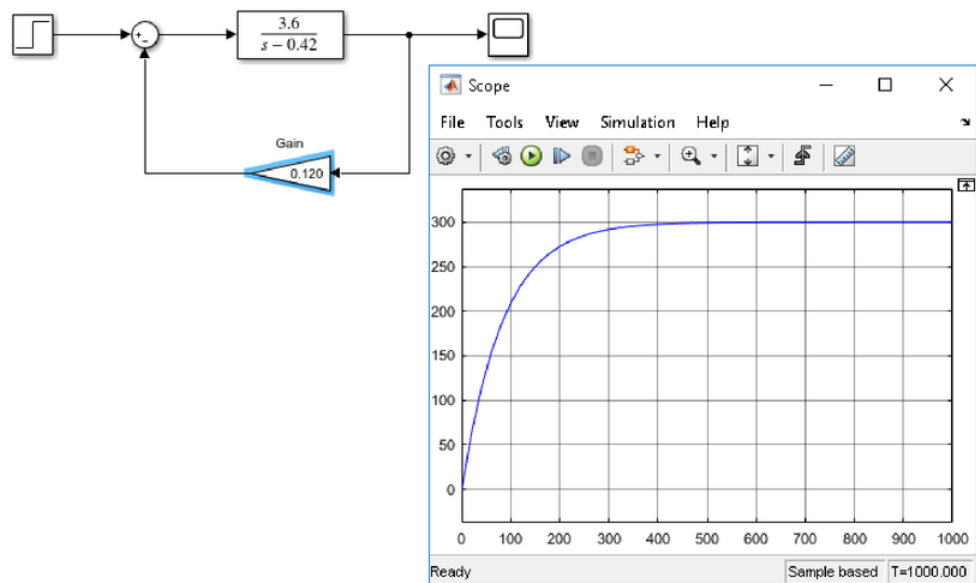


Figure 26 - Circuit With 0.120

In figure 26 the feedback in the circuit is much closer to the minimum required which results in the response being a lot slower than in figure 25 where the response was rapid and sharp on the graph. This means this circuit must be slightly overdamped and the next step will be to simulate the circuit with a gain that results in the circuit becoming slightly unstable.

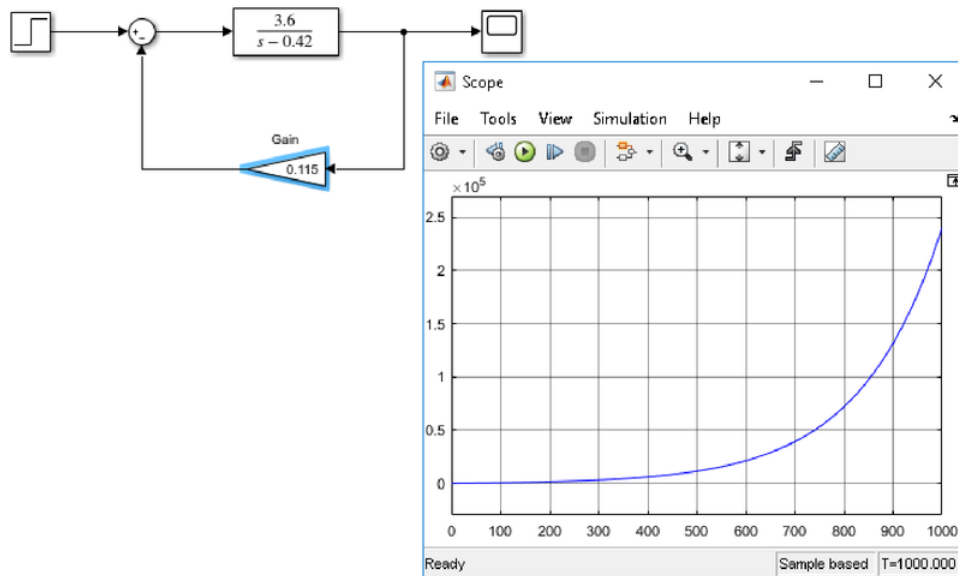


Figure 27 - Feedback 0.115

This table should have showed the results for the different values of K
In figure 27 the feedback selected was 0.115 which is just 0.001 below the minimum gain that results in the circuit becoming slightly unstable compared to the previous two circuits.

Values	Simulated Result	Calculated Result
0.42	Stable	Stable
3.18	Unstable	Unstable
Feedback	0.116	0.116

The table above shows that the open loop circuit resulted in the value of S equalling 0.42 which meant the circuit would be unstable and this matched the simulation as the pole ended up being on the right side of the graph. Whereas when calculating the value of s on a closed loop it resulted in the value of s equalling 3.18 that meant it was unstable that matched up with the simulation. The minimum gain required for the feedback in the calculations resulted in it being 0.116 so in order to find the minimum gain in the simulation it meant finding the pole at 0 and seeing the gain that matched the calculated value.

You have already shown that with unity feedback the pole lies at - 3.18 which you have already said is stable.

2. DC Motor Model Analysis

The first step in the process is to design a DC motor model that is based upon the following parameters:-

- Armature resistance = 7.8 Ohms
- Armature inductance = 0.05H
- Rotor inertia = 0.000214 Kg m²
- Friction coefficient = 15.519 x 10⁻⁶ Nm/rad/s
- Motor voltage constant = 0.0983 V/rad/s
- Motor torque constant = 0.09 Nm/A
- Maximum armature voltage = 40V

avoid use of first person

Based upon these parameters I will first need to develop a mathematical model of the motor and its load along with producing the transfer function with the disturbance initially set to zero. In order to do this I need to calculate the value of ω_n^2 . The formula for this can be seen below:-

$$\omega_n^2 = \frac{Ra + KTKE}{La * J}$$

Figure 28 - ω_n^2 Formula

In figure 28 using this formula the values from the DC motor parameters will need to be substituted into the formula in order to get the correct result.

$$\frac{7.8 \times 15.519 \times 10^{-6} + 0.09 \times 0.983}{0.05 \times 2.14 \times 10^{-6}} = 838.135$$

Figure 29 - ω_n^2 Values Substituted In

After calculating the value of the ω_n^2 that means the value of ω_n can now be calculated by finding the square root of the result in figure 29.

$$\omega_n = \sqrt{838.135} = 28.95$$

Figure 30 - ω_n Calculation

After calculating the values of ω_n this then allows the calculation of $2\zeta\omega_n$ that requires the use of the formula in figure 31.

$$2\zeta\omega_n = \frac{b}{J} + \frac{Ra}{La}$$

Figure 31 - $2\zeta\omega_n$ Formula

Using the formula found in figure 31 the parameter values need to be substituted into the formula in order to get the result.

$$= \frac{15.519 \times 10^{-6}}{2.14 \times 10^{-4}} + \frac{7.8}{0.05} = 156.07$$

Figure 32 - Result

The motor model should have been developed and explained first.

The next calculation that needs to be carried out can be seen in figure 33 below:-

$$\zeta = \frac{0.0725 + 156}{2 \times 28.95} = 2.696$$

Figure 33 - Result 2

After calculating that value the next step in the process is to calculate the value of KT that can be seen below and that will allow the value of K to then be calculated.

$$KT = \frac{0.09}{0.05(2.14 \times 10^{-4})} = 8411.21$$

Figure 34 - Calculated value of KT

Once the value of KT had been calculate this meant that the value of K could be calculated by dividing the result from figure 34 by the result from figure 29 that can be seen in figure 35.

$$K = \frac{8411.21}{838.135} = 10.04$$

Figure 35 - K Calculation

After calculating the value of K that meant that the next step was to carry out the calculation for the value of ΩNL by using the formula below:-

$$\Omega NL = \frac{KTVa}{bRa + KTK E}$$

Figure 36 - ΩNL Formula

By substituting in the correct values the result can be gained as can be seen in figure 37.

$$= \frac{(0.09)(24)}{15.519 \times 10^{-6}(7.8) + (0.09)(0.0983)} = 240.86$$

Figure 37 - OHM NL Result

Now that that result has been calculated this means that the calculation of OHM NL can now be carried out the formula and result can be seen below:-

$$\Omega L = \frac{Va - \frac{RaTL}{KT}}{\frac{Rab}{KT} + KE}$$

Figure 38 - OHM L

After all the values have been calculated this means the general solution can be put together as can be seen in figure 39.

$G(s)$ is a transfer function not a general solution

$$G(s) = \frac{8411.25}{s^2 + 156.0725s + 838.135}$$

Figure 39 - General Solution

Once the general solution had been found this meant that a quadratic equation could be used to this means for the quadratic equation the value of A would equal 1, the value of B would equal 156.0725 and the value of C 838.135 basically all the values from the denominator.

$$s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Figure 40 - Quadratic Equation

$$= \frac{-156.0725 \pm \sqrt{156.0725^2 - 4(1)(838.135)}}{2} = -150.5 \text{ or } -150.5$$

Figure 41 - Quadratic Equation Result

Now that the results from the quadratic equation have been gained this means that they can be plotted on an argan diagram to determine whether the poles are stable or not.

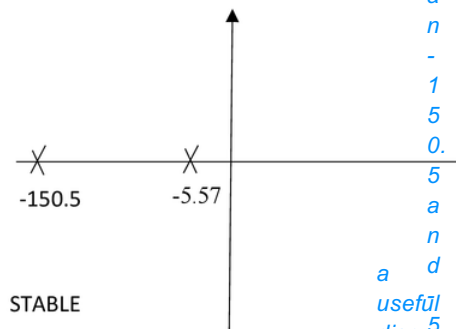


Figure 42 - Argan Diagram

The argand diagram displayed in figure 42 is the theoretical prediction, which in this case shows that the poles are both stable unlike in task 1 where one pole was in a stable condition and the other in an unstable condition. After this the next step is to design the model in matlab and the circuit with no load can be seen in figure 43.

The pole actually moved in task 1, it was a first order system and there were never 2 poles

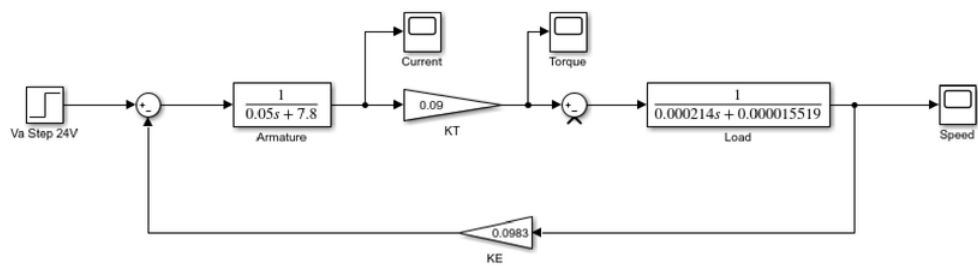


Figure 43 No Load Model

After designing the no load model the next step was to add on the load torque into the system as can be seen in figure 44.

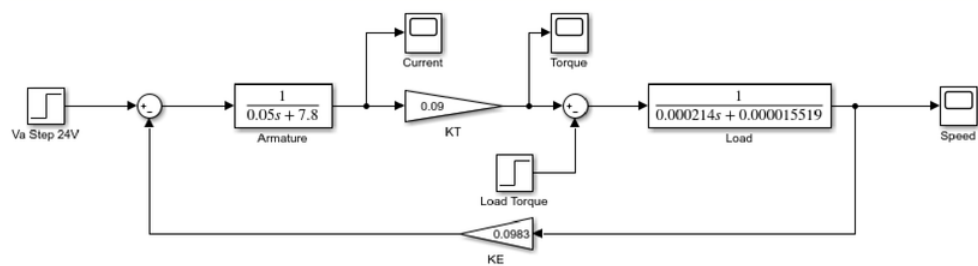


Figure 44 - Load Model *you could have explained each part of this motor model*

After adding the load torque for the circuit the next step was to find out the speed for the load modal after 10s. This meant when running the Simulink circuit the scope labelled speed would need to be checked in order to determine the results as can be seen in figure 45.

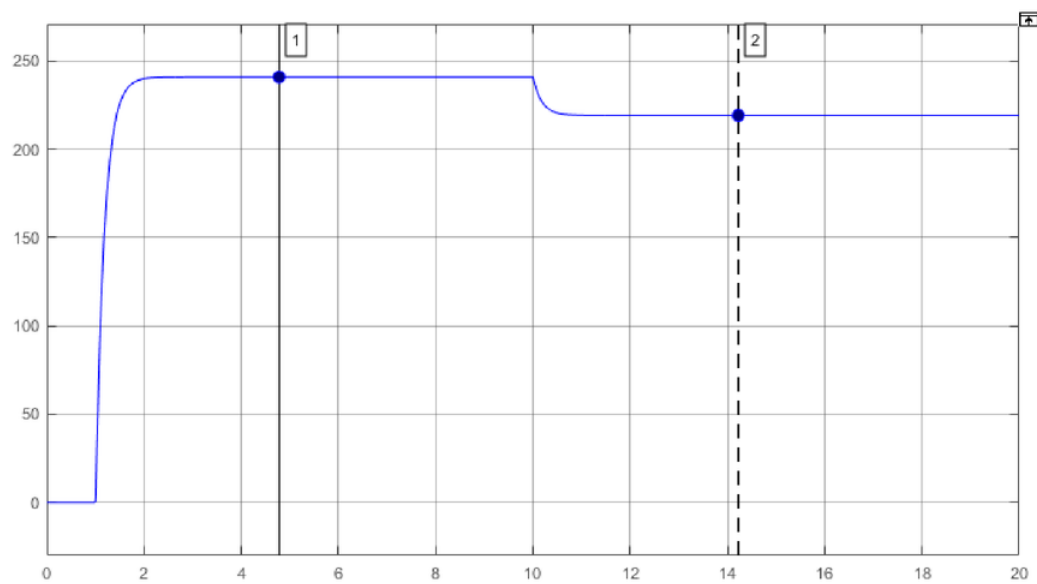


Figure 45 - After 10s Graph

In figure 45 it shows the graph associated with the speed scope that shows initially there is a rapid response before levelling out after about 2s. In order to do this I needed to put to cursors on the graph to determine the speed after 10s.

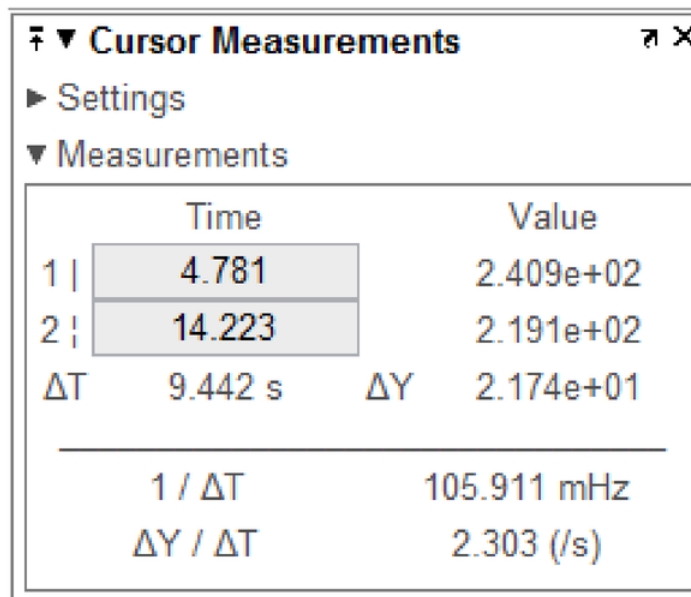


Figure 46 – Measurements

In figure 46 it shows after the speed had steadied at about 2s the speed was the until 10s that's why the measurements above shows a time of 4.7 as that value would be the same as the value at 10s. This shows that at 240.9 there was no load speed associated with the motor and that once the motor had been moved into under load the speed immediately reduces after 10s to 2.19. After this the current and torque graphs needed to be evaluated to see if they were performing as expected.

How do these results compare with the calculated ones?

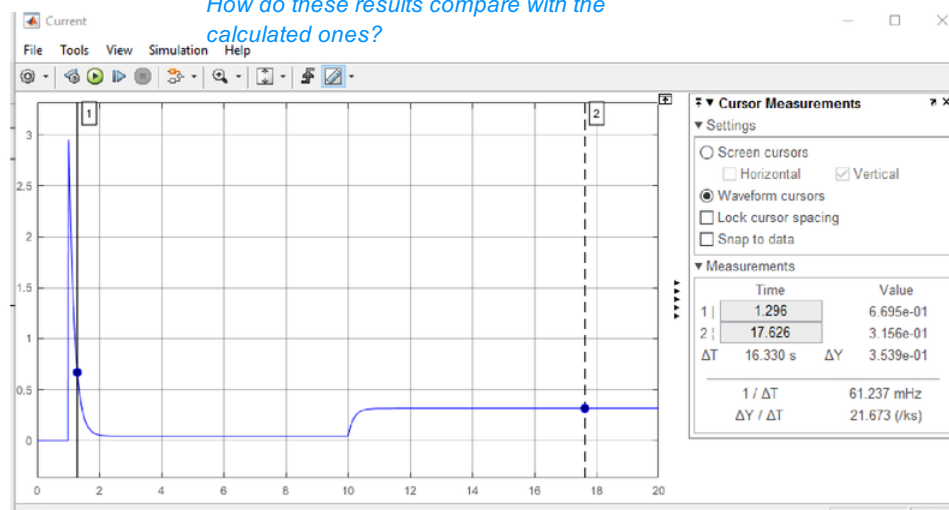


Figure 47 - Current Graph

The current graph shows after 1s it has a rapid raise up to a value of around 2.8 before rapidly declining before reading out until after 10s where it slightly increase in value.

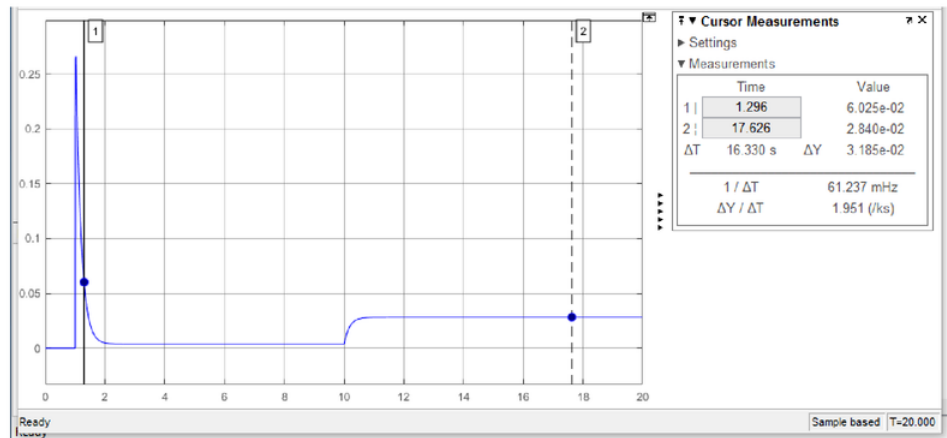


Figure 48 - Torque Graph

In figure 48 it display the torque graph which can easily be identified as being the exact same as that of the current graph as it shows they both show once load added they rapidly rise in value. The next step in the process will to calculate the stall value for the motor the formula and result can be seen in figure 49.

$$TM(Stall) = \frac{KT(Va)}{Ra} = \frac{0.09 \times 24}{7.8} = 0.277$$

Figure 49 - Stall Calculation

After calculating the stall load in the circuit this means that it can now be added into the circuit to see if the motor stalling. In order to see this the graph that will need to be looked at is the speed graph.

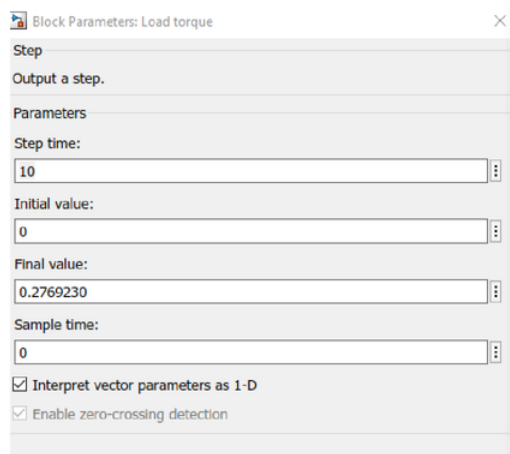


Figure 50 - Adding Load Stall

IN figure 50 it shows the load stall being added to the circuit with the step time being entered as 10 and the final value is where the load stall will be entered.

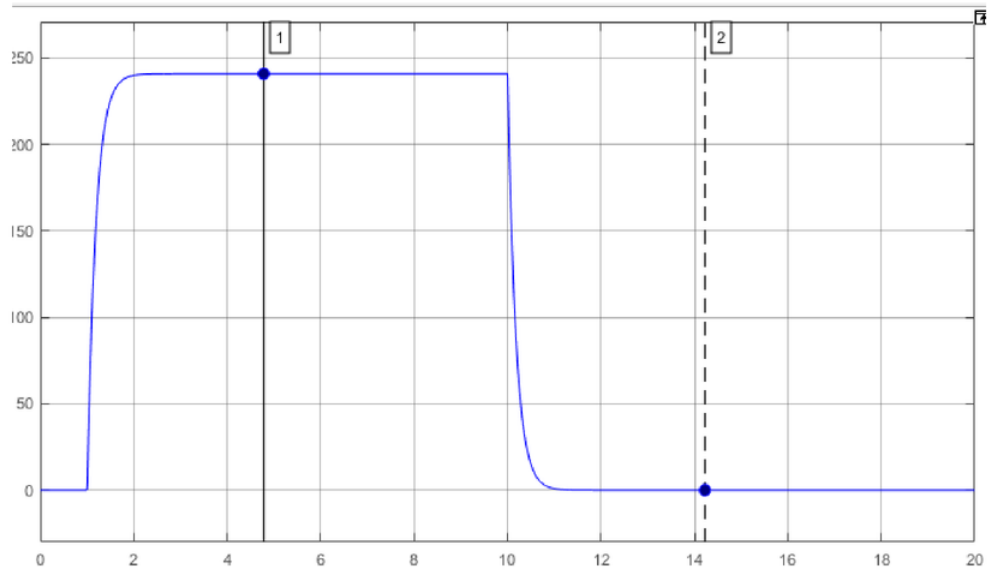


Figure 51 - Speed Graph With Load Stall

In figure 51 it shows that the load stall performed as expected matching the predications preciously made.

How do you know it has stalled?

3. Types Of Motor Speed Controller

In this section the three different types of motors that I will look at are a P controller, PI controller and a PID controller.

3.1 P Controller

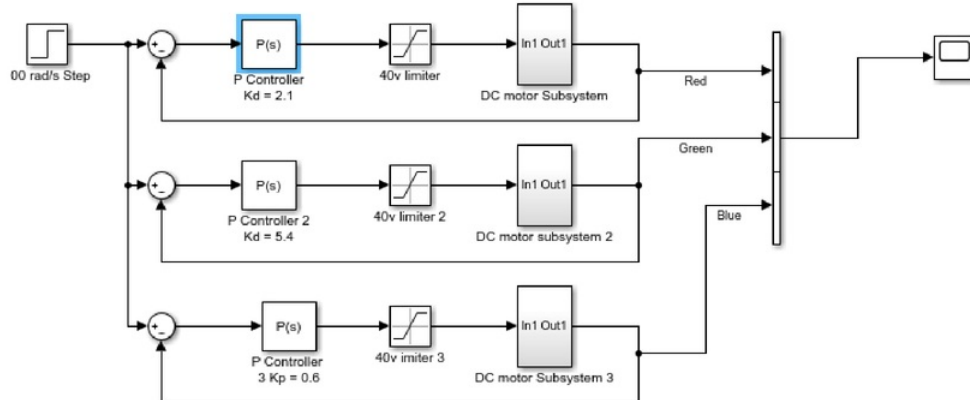


Figure 52 - P Controller

In figure 52 it shows the P controller for the circuit and each of the P controllers have different P values with the first being a tuned value of 2.1125, 5.5 and 0.6. The first thing that needs to be done is to simulate the tuned value that can be seen in figure 53.

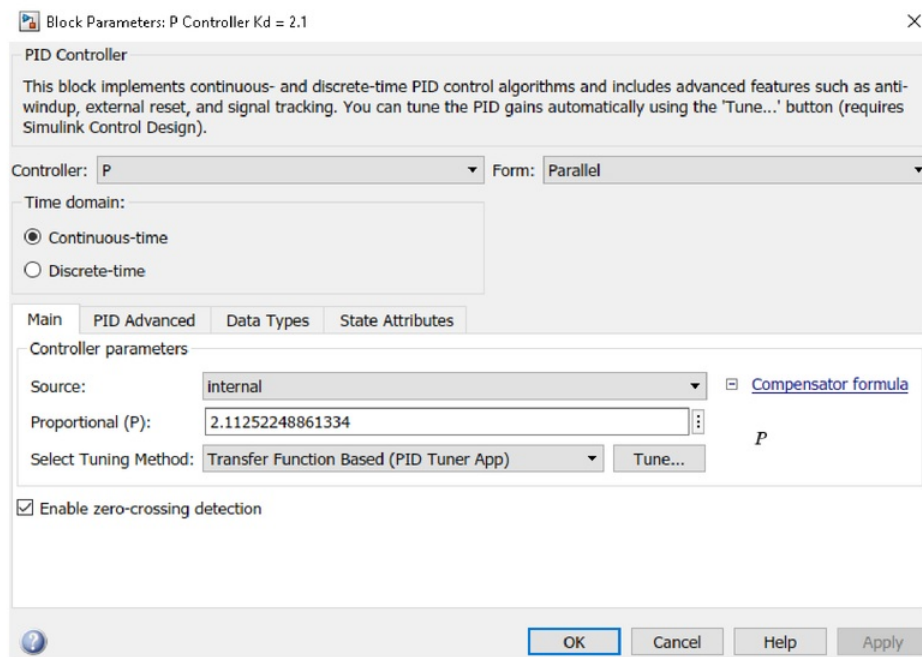


Figure 53 - P Controller Tuned Value

In figure 53 it shows the tuned value for the P Controller with the P value being 2.1 and the graph for this can be seen in figure 54.

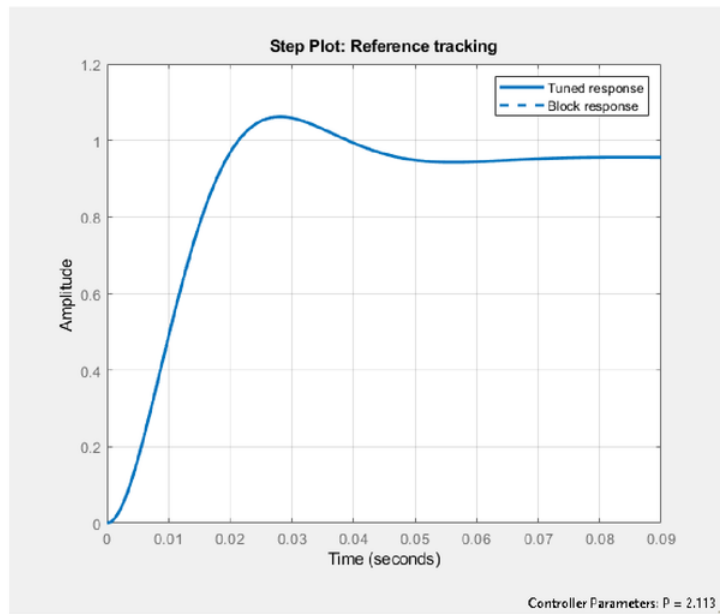


Figure 54 - P Controller Tuned Value Graph

Controller Parameters		
	Tuned	Block
P	2.1125	2.1125
I	n/a	n/a
D	n/a	n/a
N	n/a	n/a
Performance and Robustness		
	Tuned	Block
Rise time	0.0131 seconds	0.0131 seconds
Settling time	0.0431 seconds	0.0431 seconds
Overshoot	11.2 %	11.2 %
Peak	1.06	1.06
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	60 deg @ 98.6 rad/s	60 deg @ 98.6 rad/s
Closed-loop stability	Stable	Stable

Figure 55 - P Controller values

In figure 55 it shows the P vales that are associated with the tuned value of 2.1 it gives the overshoot percentage of 11.2%. The next step in the process is to simulate the P controller with a P value of 5.4 that can be seen in figure 56.

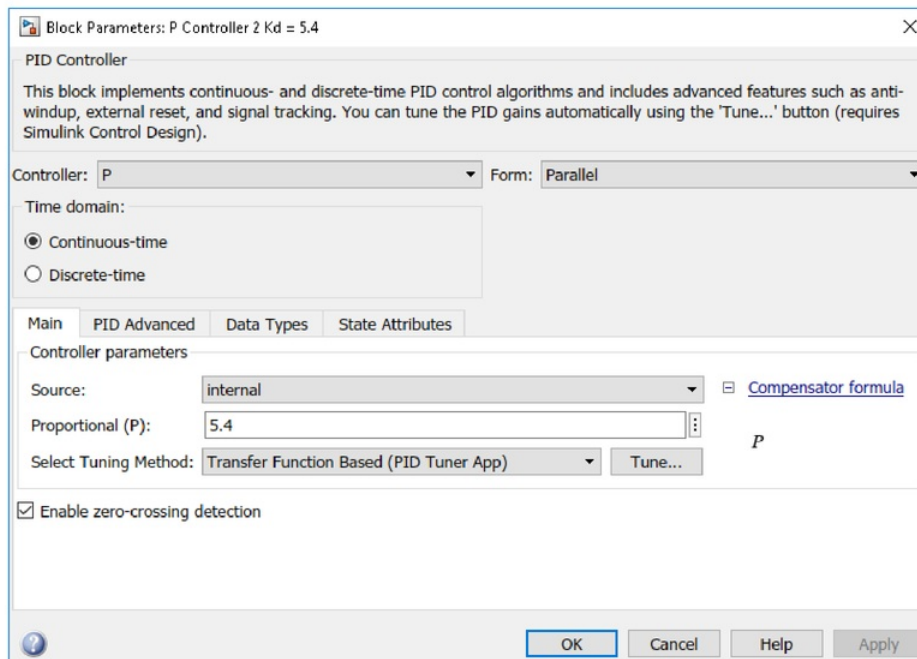


Figure 56 - P Controller 5.4

In figure 56 it shows the P controller has a P value of 5.4 which is almost double the tuned value and the graph associated with this can be seen in figure 57.

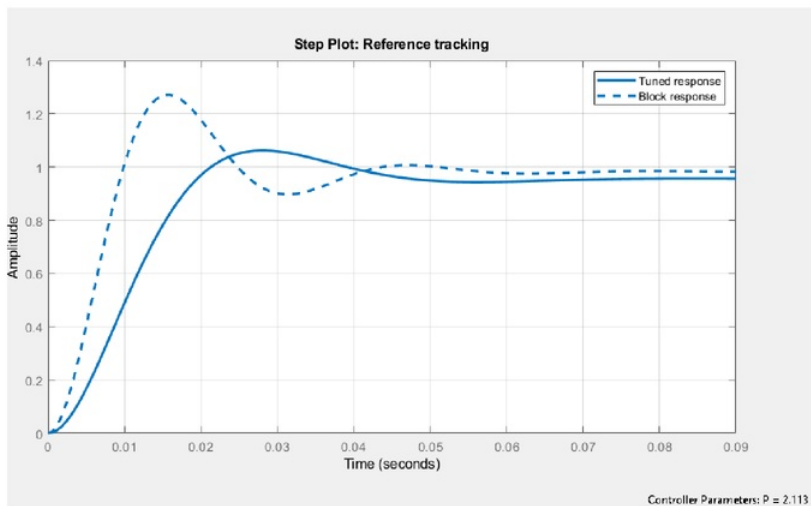


Figure 57 - P Controller 5.4 Graph

In figure 57 it shows the graph for the P controller with a P value of 5.4 and the graph associated with this can be seen in figure 58.

Controller Parameters		
	Tuned	Block
P	2.1125	5.4
I	n/a	n/a
D	n/a	n/a
N	n/a	n/a

Performance and Robustness		
	Tuned	Block
Rise time	0.0131 seconds	0.00656 seconds
Settling time	0.0431 seconds	0.0504 seconds
Overshoot	11.2 %	29.4 %
Peak	1.06	1.27
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	60 deg @ 98.6 rad/s	40.3 deg @ 188 rad/s
Closed-loop stability	Stable	Stable

Figure 58 - P Controller 5.4

In figure 58 it displays the values for P controllers with the P value of 5.4 and it shows that it has an overshoot of 29.4%.

Block Parameters: P Controller 3 Kp = 0.6

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: P Form: Parallel

Time domain:

☒ Continuous-time

☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Source: internal [Compensator formula](#)

Proportional (P): 0.6 P

Select Tuning Method: Transfer Function Based (PID Tuner App) [Tune...](#)

☒ Enable zero-crossing detection

OK Cancel Help Apply

Figure 59 - P Controller With P Value Of 0.7

In figure 59 it shows the P controller with a value of 0.6, which is almost a third of the tuned value and the graph for this can be seen in figure 60.

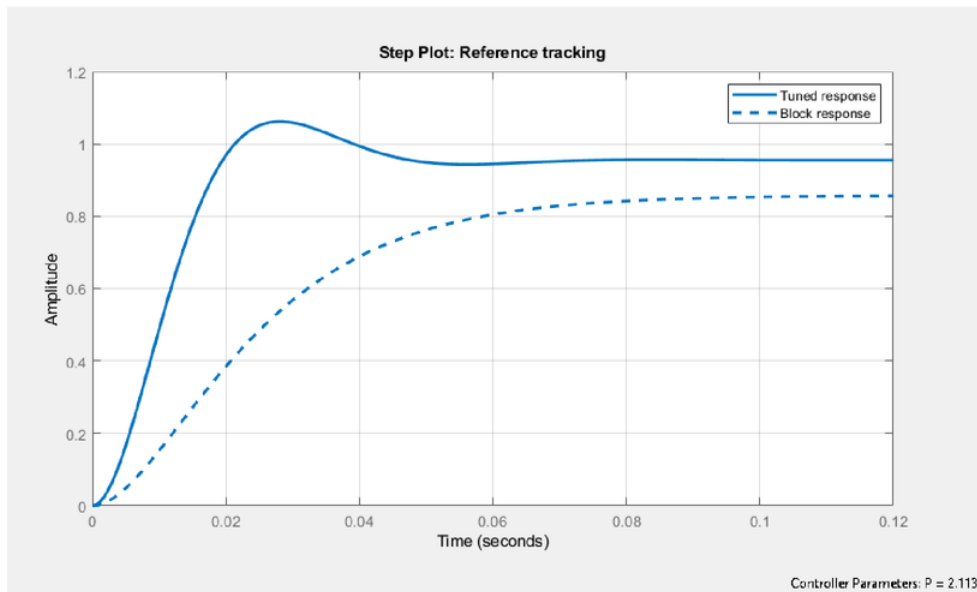


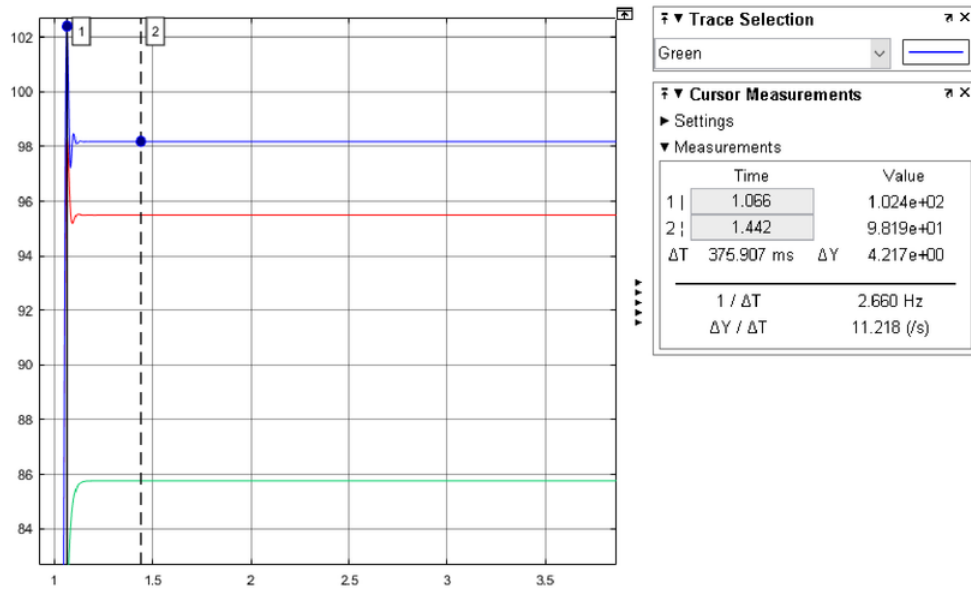
Figure 60 - P Controller P Value 0.6

In figure 60 it shows the P Controllers value with a P value of 0.6 and the values associated with this can be seen in figure 61.

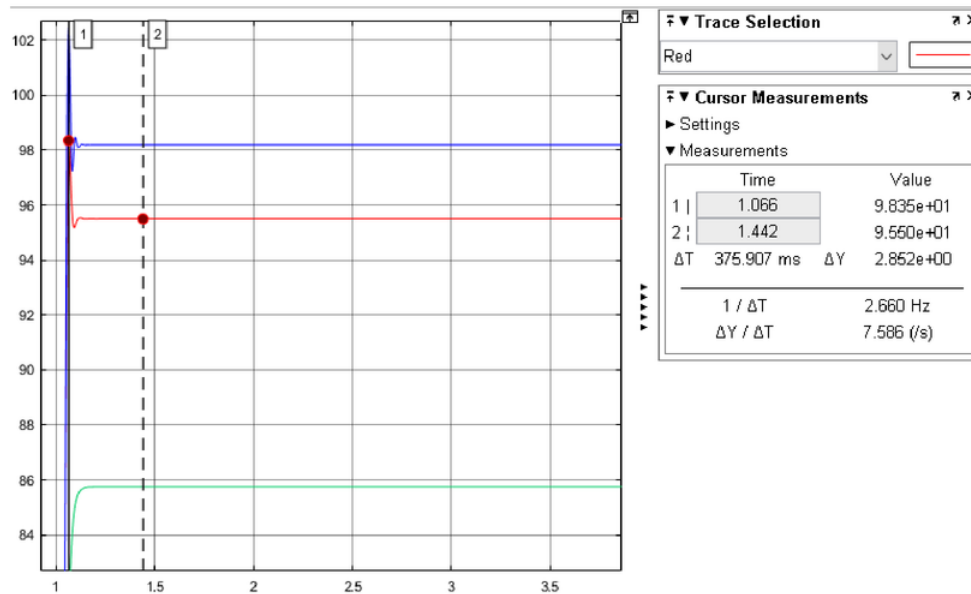
Controller Parameters		
	Tuned	Block
P	2.1125	0.6
I	n/a	n/a
D	n/a	n/a
N	n/a	n/a
Performance and Robustness		
	Tuned	Block
Rise time	0.0131 seconds	0.0449 seconds
Settling time	0.0431 seconds	0.0786 seconds
Overshoot	11.2 %	0 %
Peak	1.06	0.857
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	60 deg @ 98.6 rad/s	87.7 deg @ 32.3 rad/s
Closed-loop stability	Stable	Stable

Figure 61 - P Controller P Value 0.6

In figure 61 it shows that it has an overshoot value 0% which is significantly lower than the other two P controllers.



In figure 62 it display the P controller with a Kp value of 5.4 and it shows that it has the biggest overshoot of all the systems along with having the least steady state errors.



In figure 63 it shows the P controller with a P value that is tuned as 2.1 and it has a bit of an overshoot along with a steady state error.

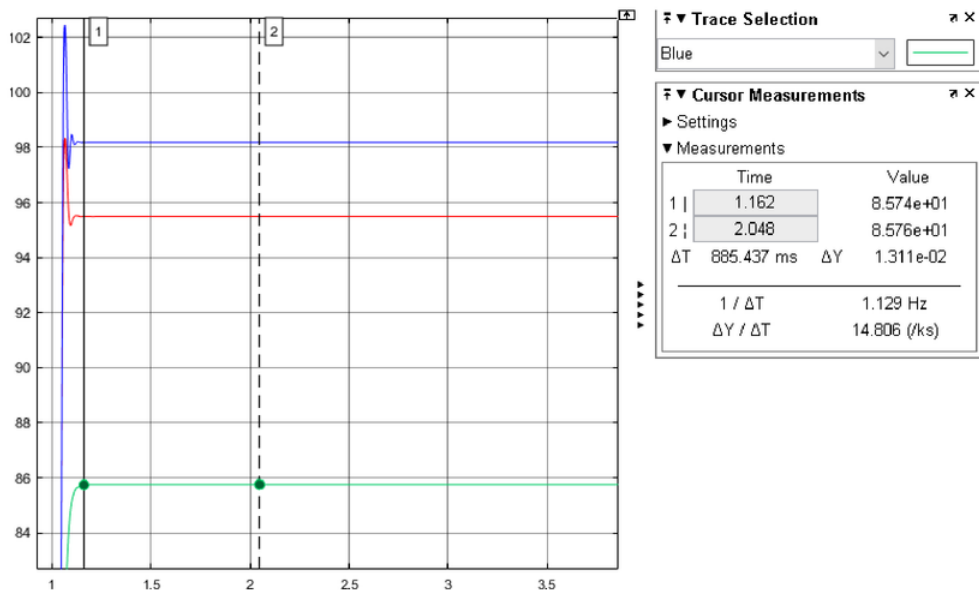


Figure 64 - P Controller 0.6

In figure 64 for the P Controller with a P value of 0.6 that has the lowest Kp value of all the systems we have designed and it also has the zero overshoot along with a large steady state error.

Kp value	Rise time (S)	Transient error	Overshoot (%)
2.1125	0.0131	Med	11.2
5.4	0.00656	Low	29.4
0.6	0.0449	High	0

This is a good table but what are you

3.2 PI Controller *conclusions*

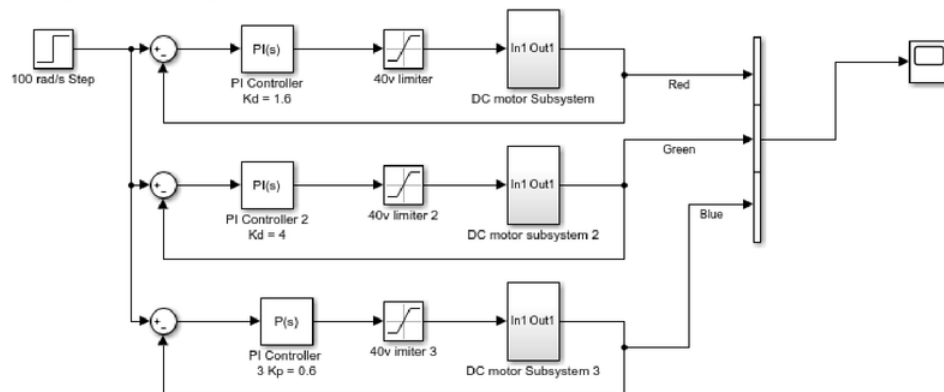


Figure 65 - PI Controller

IN figure 65 it shows the three PI Controllers designed to drive DC motors and the tuned PI controller has a value of 1.6, while the other two values that I have selected was 4 and 0.6.

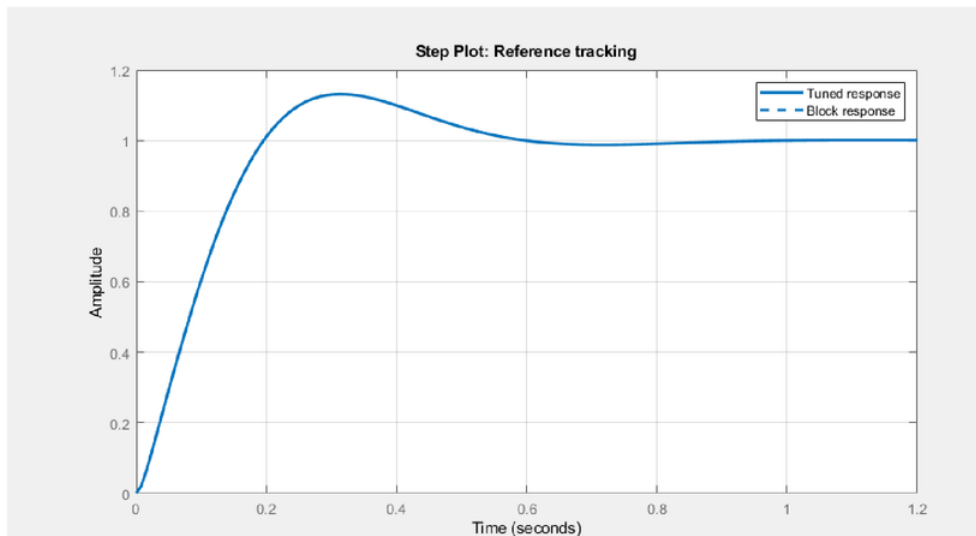


Figure 66 - PI Controller Tuned

In figure 66 it shows the tuned PI Controller and the values associated with this controller can be seen in figure 67.

Controller Parameters		
	Tuned	Block
P	0.11371	0.11371
I	1.6445	1.6445
D	n/a	n/a
N	n/a	n/a
Performance and Robustness		
	Tuned	Block
Rise time	0.142 seconds	0.142 seconds
Settling time	0.537 seconds	0.537 seconds
Overshoot	13.1 %	13.1 %
Peak	1.13	1.13
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	60 deg @ 9.82 rad/s	60 deg @ 9.82 rad/s
Closed-loop stability	Stable	Stable

Figure 67 - PI Controller Tuned 1.6

IN figure 67 it shows the tuned PI Controller has an overshoot of 13.1% along with having a rise time of 0.142 second.

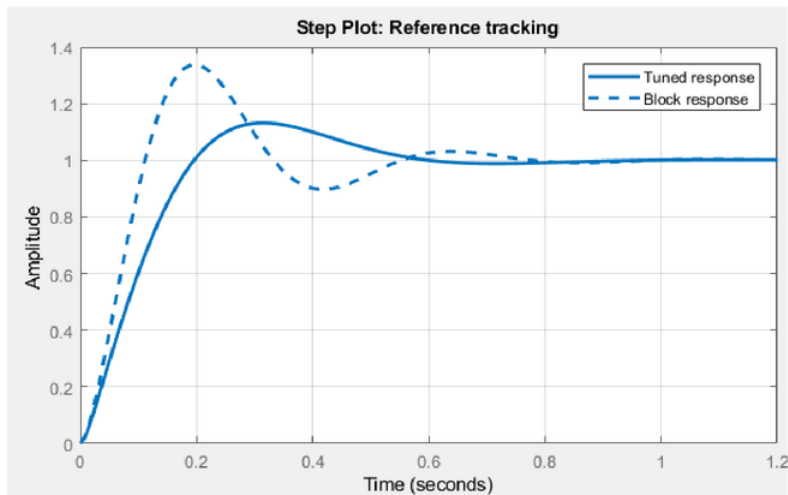


Figure 68 - PI Controller 4

In figure 68 it shows the PI controller with a value of 4 graph that displays the block response against the tuned graph and the values associated with it can be seen in figure 69.

Controller Parameters		
	Tuned	Block
P	0.11371	0.11371
I	1.6445	4
D	n/a	n/a
N	n/a	n/a
Performance and Robustness		
	Tuned	Block
Rise time	0.142 seconds	0.0806 seconds
Settling time	0.537 seconds	0.703 seconds
Overshoot	13.1 %	33.7 %
Peak	1.13	1.34
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	60 deg @ 9.82 rad/s	37.8 deg @ 15.1 rad/s
Closed-loop stability	Stable	Stable

Figure 69 - PI Controller Values

In figure 69 it shows the PI controller where it has a I value of 4, which has an overshoot of 33.7% which is much higher than the previous PI controller.

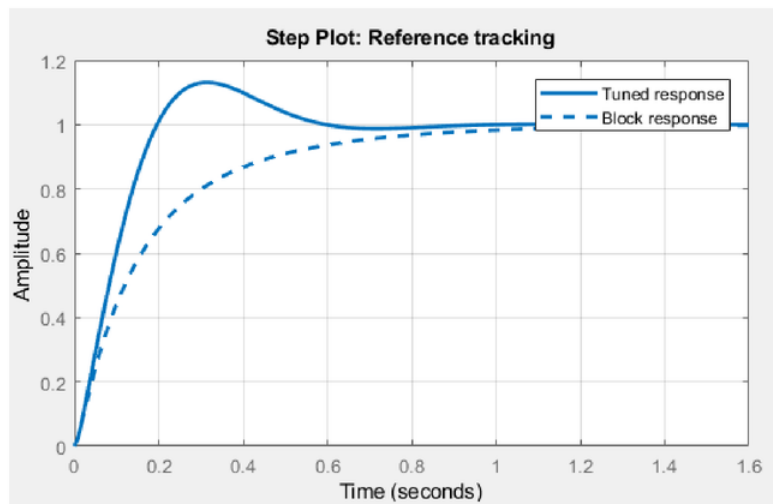


Figure 70 - PI Controller $I = 0.5$

In figure 70 it shows the graph for the PI controller with a value of 0.6 for I and the values associated with this in figure 71.

Controller Parameters		
	Tuned	Block
P	0.11371	0.11371
I	1.6445	0.6
D	n/a	n/a
N	n/a	n/a
Performance and Robustness		
	Tuned	Block
Rise time	0.142 seconds	0.353 seconds
Settling time	0.537 seconds	0.668 seconds
Overshoot	13.1 %	0 %
Peak	1.13	0.999
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	60 deg @ 9.82 rad/s	89.2 deg @ 6.2 rad/s
Closed-loop stability	Stable	Stable

Figure 71 - PI Controller $I = 0.6$

In figure 71 it shows the PI controller where the I value is 0.6 and it has an overshoot value of 0% which is significantly lower than the others.

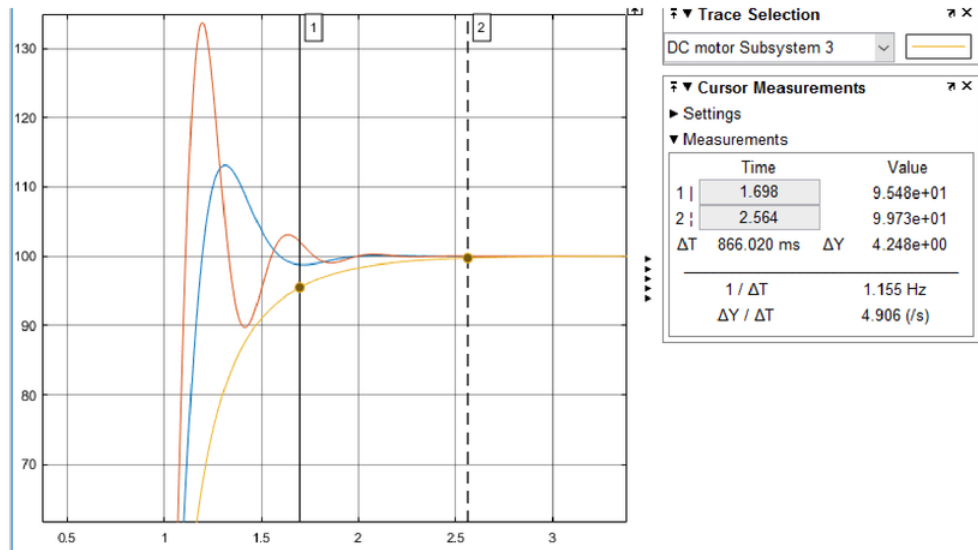


Figure 72 - PI Controller 0.6

In figure 72 it is the circuit that has the least K_i value and it can be seen there is no overshoot with a very slow rise time.

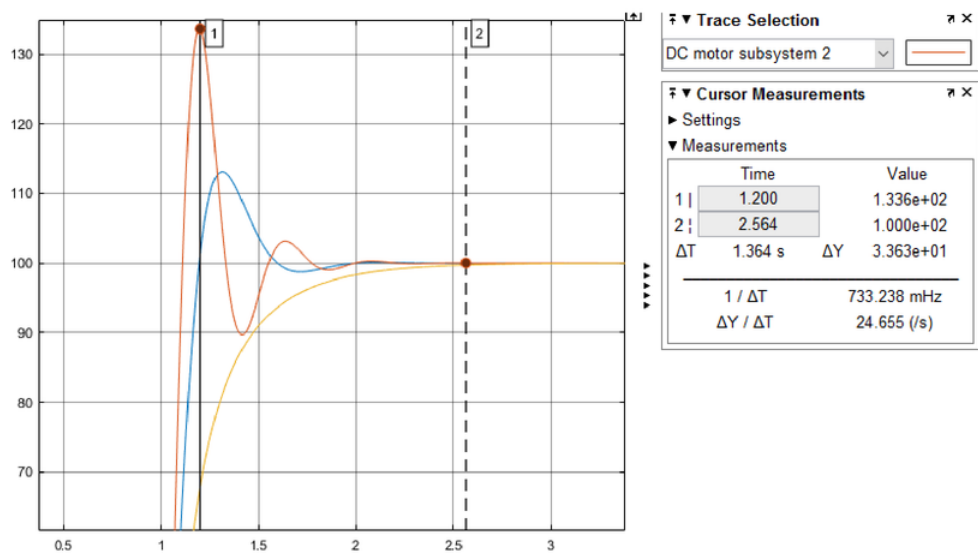


Figure 73 - PI Controller 4

In figure 73 this is the circuit with the highest K_i value that has a value of 4 and this has the largest overshoot with a rapid rise time.

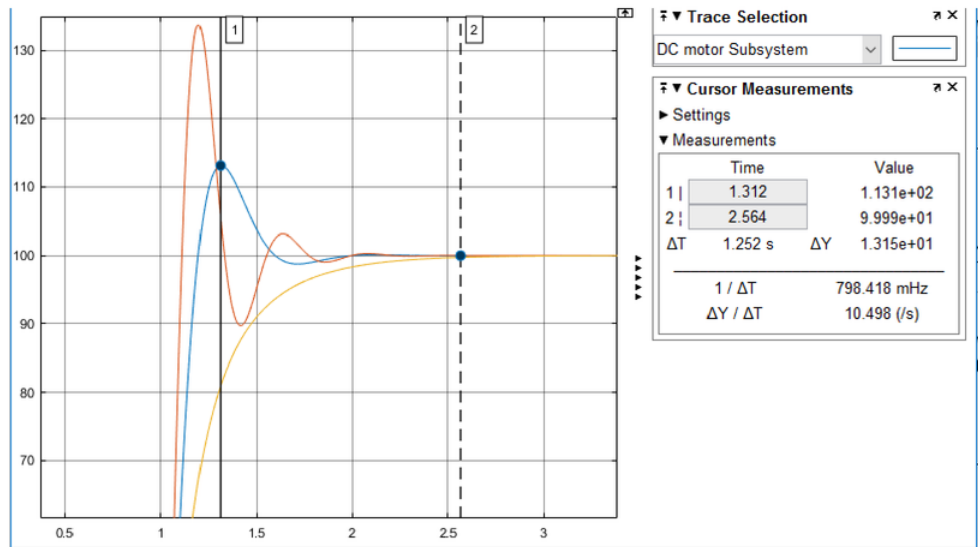


Figure 74 - Tuned PI Controller

In figure 74 it's the graph associated with the tuned value with a K_I value of 4 and it only has a slight overshoot and a very fast rise time.

K_I value	Rise time (S)	Transient error	Overshoot (%)
1.6445	0.142	--	13.1
4	0.0806	--	33.7
0.6	0.353	--	0

This is a good table but what are you conclusions

3.3 PID Controller

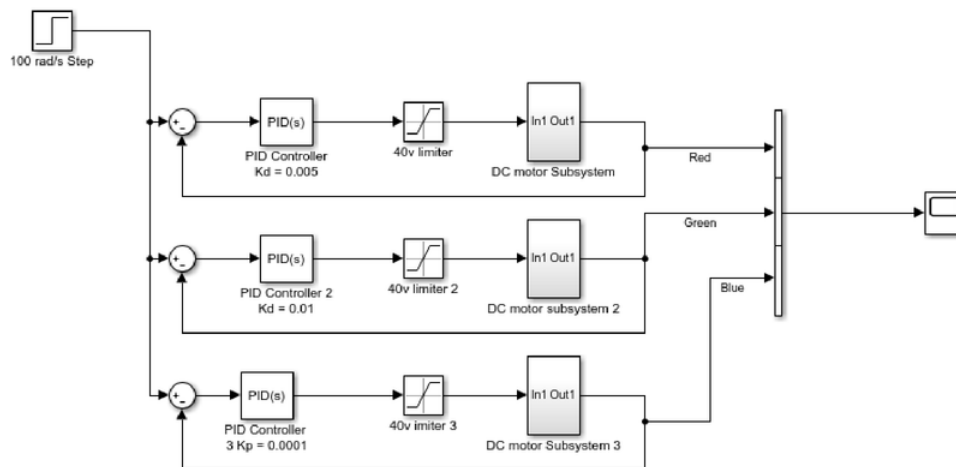


Figure 75 - PID Controller

In figure 75 it shows the PID controller for the circuit where the tuned value is 0.005, then the other is 0.01 and the last one is 0.0001.

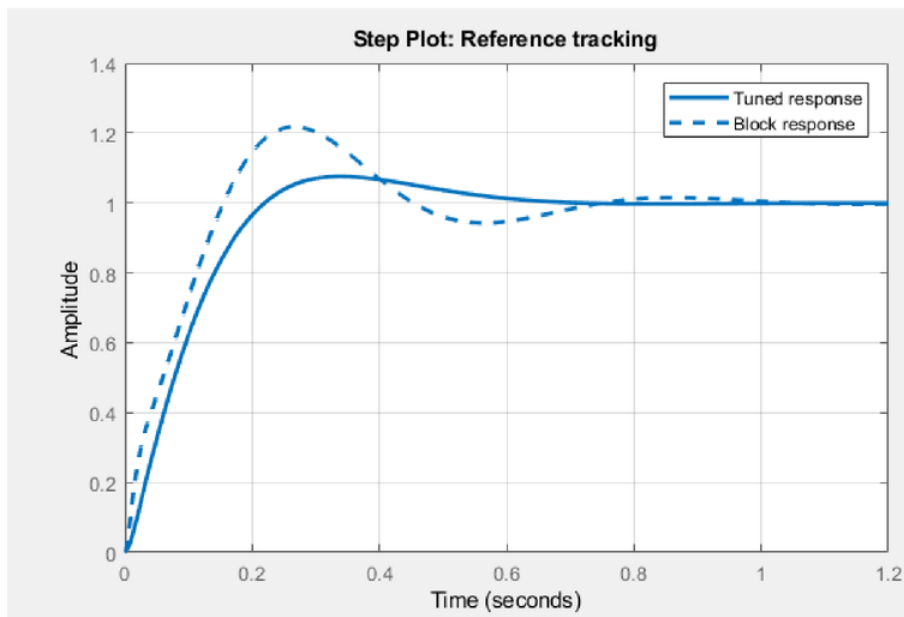


Figure 76 - PID Controller 0.005

In figure 76 it shows the controller that has been tuned and assigned a value of 0.005, which gives a very fast rise time with an overshoot of 21.7% as can be seen in figure 77.

Controller Parameters		
	Tuned	Block
P	0.13849	0.11371
I	1.4497	3
D	0	0.005
N	100	100
Performance and Robustness		
	Tuned	Block
Rise time	0.153 seconds	0.124 seconds
Settling time	0.563 seconds	0.693 seconds
Overshoot	7.57 %	21.7 %
Peak	1.08	1.22
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	69 deg @ 9.82 rad/s	52.2 deg @ 11.7 rad/s
Closed-loop stability	Stable	Stable

Figure 77 - PID Controller Tuned Values

The next PID Controller has a D value of 0.01 and the graph for this can be seen in figure 78.

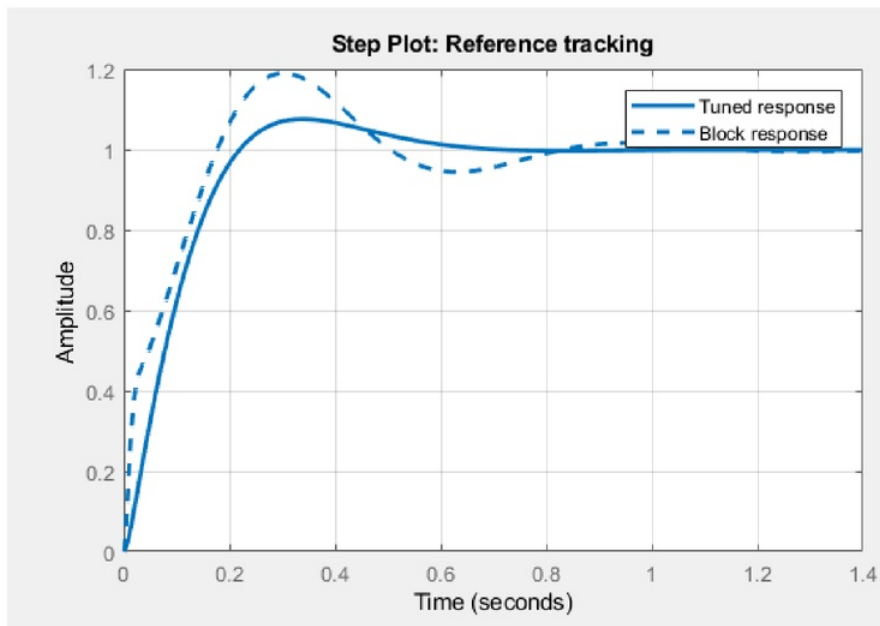


Figure 78 - PID Controller 0.01

In figure 78 it shows the graph associated with the PID controller with a D value of 0.01 and this has a bit of an overshoot and a rapid rise.

Controller Parameters		
	Tuned	Block
P	0.13849	0.11371
I	1.4497	3
D	0	0.01
N	100	100
Performance and Robustness		
	Tuned	Block
Rise time	0.153 seconds	0.142 seconds
Settling time	0.563 seconds	0.77 seconds
Overshoot	7.57 %	19 %
Peak	1.08	1.19
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	69 deg @ 9.82 rad/s	58.9 deg @ 10.7 rad/s
Closed-loop stability	Stable	Stable

Figure 79 - PID Controller 0.01 Values

In figure 79 it shows that the rise time for the PID Controller is 0.142 seconds with an overshoot of 19%.

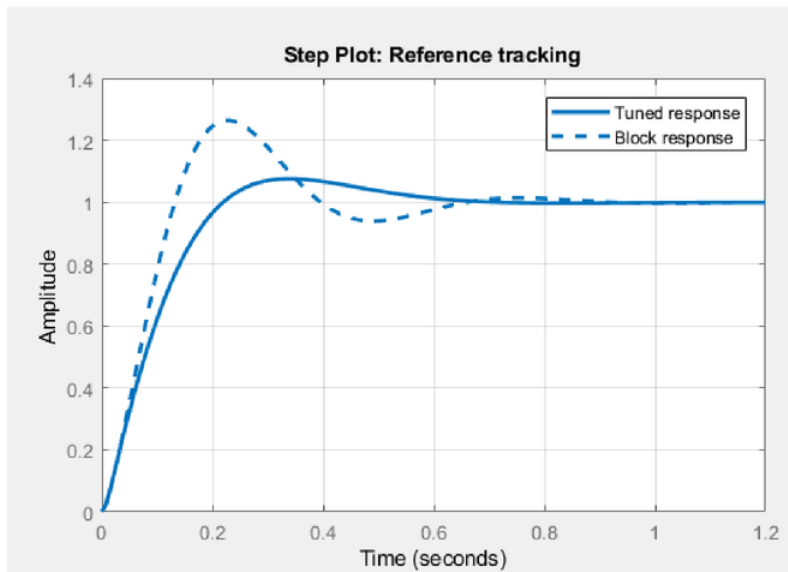


Figure 80 - PID Controller D value 0.0001

In figure 80 it shows the PID controller that has a value of 0.0001 associated with it and it has a quick rise time with a large overshoot.

Controller Parameters		
	Tuned	Block
P	0.13849	0.11371
I	1.4497	3
D	0	0.0001
N	100	100
Performance and Robustness		
	Tuned	Block
Rise time	0.153 seconds	0.0967 seconds
Settling time	0.563 seconds	0.608 seconds
Overshoot	7.57 %	26.4 %
Peak	1.08	1.26
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	69 deg @ 9.82 rad/s	44.6 deg @ 13.1 rad/s
Closed-loop stability	Stable	Stable

Figure 81 - PID Controller 0.0001

In figure 81 it shows the values associated to the PID controller that has the lowest Kd value and it has a rise time of 0.0967 seconds and an overshoot of 26.4%.

What are your overall conclusions from this table?

<u>Kd value</u>	Rise time (S)	Overshoot (%)
0.005	0.124	21.7
0.01	0.142	19
0.0001	0.0967	26.4

Figure 82 - Comparison PID Controller

3.4 Comparison Of P, PI and PID Controllers

In order to carry out a comparison between the three different types of controllers that means that a circuit needed to be created with one of each controller tuned. As can be seen in figure 83.

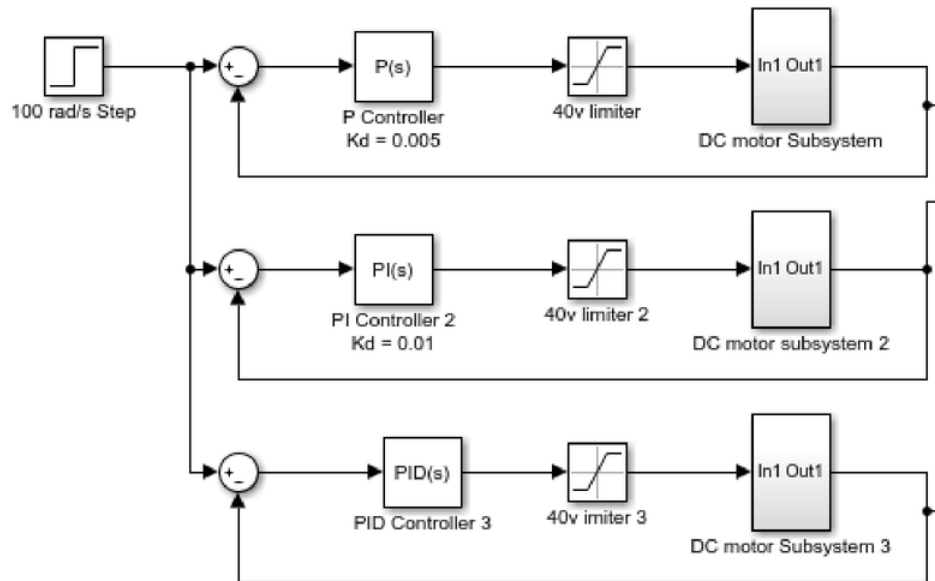


Figure 83 - Controller Comparison

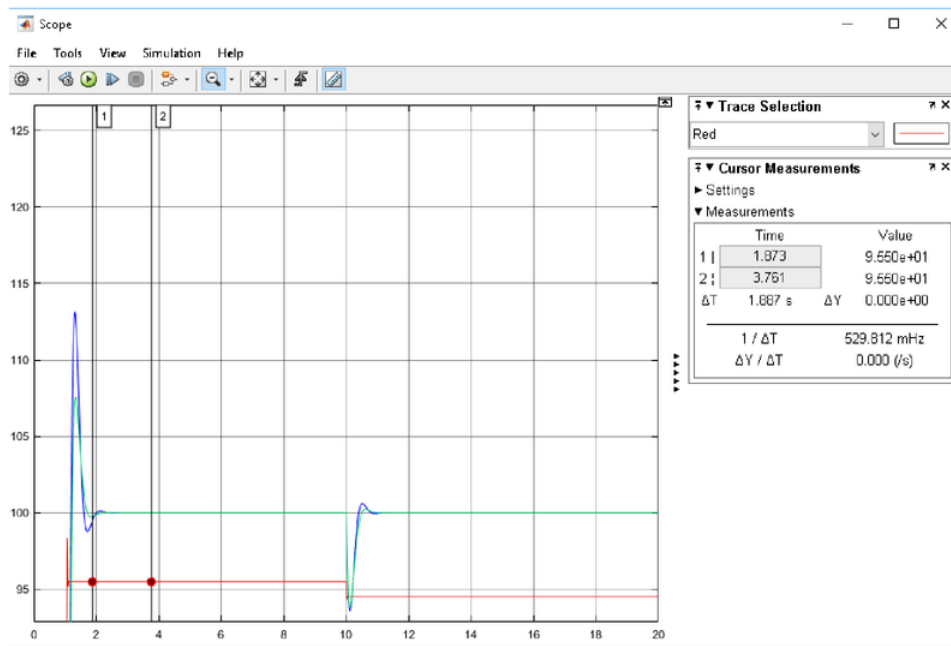


Figure 84 - Comparison Controllers

In figure 84 it shows that the circuit in figure 83 has been run in order to produce the graph above to display the overshoot of each along with their rise time. As can be seen the blue line has the largest overshoot of all the controllers along with a rapid rise time, whereas the red controller has quite a large steady state error, which for this application would make it unsuitable. On the other had the PID Controller which is represented by the green line shows a lower overshoot than the blue line with a quick rise time and no steady state error.

4. Conclusion

Don't write in the first person

In conclusion I learnt how to calculate theoretically calculate the values for first order systems along with how to plot them on argan diagram to work out whether the poles are stable or unstable depending upon their position. Then I learnt how to run this simulation on matlab to confirm the theoretical calculations carried out were correct using matlab software. In order to confirm the calculations against the simulation the minimum gain needed to be calculated and in order to confirm this in simulation on the rlocus on matlab go to the 0 to get the gain along with confirming pole values were correct, which in this case was true. While Also learning to model a DC motor modal by carrying out multiple calculations in order to use a quadratic equation to gain the pole values to see if they are stable or unstable which in this case both were stable and run the simulation on Simulink to confirm the calculations. Then following this using the motor modal to with different types of controllers by trying three different values one tunes one larger and once much smaller to compare the overshoot and rise time between the three, every time the one with the highest value of P, I or D would have the highest overshoot.

Figure 1 - First Order Control System	2
Figure 2 - General Solution	2
Figure 3 - H(S) Value.....	2
Figure 4 - S value	2
Figure 5 - S Value.....	2

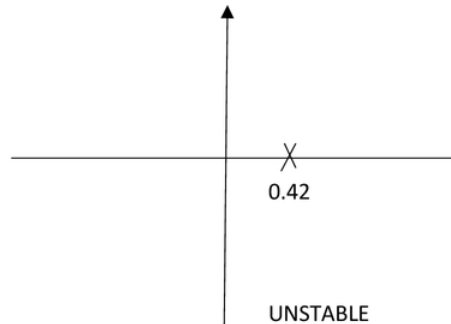


Figure 6 - Argand Diagram	3
Figure 7 - First Step Work Out Feedback	3
Figure 8 - Formula With Values.....	3
Figure 9 - Multiply Top and Bottom.....	3
Figure 10 - Take away - 0.42 from 3.6.....	4
Figure 11 – Result.....	4
Figure 12 – Denominator	4
Figure 13 - Calculated S Value.....	4
Figure 14 - Argand Diagram	4
Figure 15 - Circuit With Feedback.....	5
Figure 16 – Formula	5
Figure 17 - Formula 2	5
Figure 18 - Formula 3	5
Figure 19 - After S=0	5
Figure 20 - Aim To Get K On It's Own.....	5
Figure 21 - K Value	6
Figure 22 - Unstable Circuit.....	6
Figure 23 - Shows the Mimium Gain.....	7
Figure 24 - SHowS The Stable Side Of The Circuit	7
Figure 25 - Simulink Simulation Gain = 0.5	8
Figure 26 - Circuit With 0.120	8
Figure 27 - Feedback 0.115	9
Figure 28 - WN2 Formula	10
Figure 29 - WN2 Values Substituted In	10
Figure 30 - WN Calculation	10
Figure 31 - $2\zeta W_n$ Formula	10
Figure 32 – Result.....	10
Figure 33 - Result 2	11
Figure 34 - Calculated value of KT.....	11
Figure 35 - K Calculation	11
Figure 36 - QNL Formula	11

Figure 37 - OHM NL Result	11
Figure 38 - OHM L	11
Figure 39 - General Solution	12
Figure 40 - Quadratic Equation	12
Figure 41 - Quadratic Equation Result	12
Figure 42 - Argan Diagram	12
Figure 43 No Load Model	13
Figure 44 - Load Modal	13
Figure 45 - After 10s Graph	13
Figure 46 – Measurements	14
Figure 47 - Current Graph	14
Figure 48 - Torque Graph	15
Figure 49 - Stall Calculation	15
Figure 50 - Adding Load Stall	15
Figure 51 - Speed Graph Wlth Load Stall	16
Figure 52 - P Controller	17
Figure 53 - P Controller Tuned Value	17
Figure 54 - P Controller Tuned Value Graph	18
Figure 55 - P Controller values	18
Figure 56 - P Controller 5.4	19
Figure 57 - P Controller 5.4 Graph	19
Figure 58 - P Controller 5.4	20
Figure 59 - P Controller With P Value Of 0.7	20
Figure 60 - P Controller P Value 0.6	21
Figure 61 - P Controller P Value 0.6	21
Figure 62 - P Controller with P Value 5.4	22
Figure 63 P Controller Tuned P Value	22
Figure 64 - P Controller 0.6	23
Figure 65 - PI Controller	23
Figure 66 - PI Controller Tuned	24
Figure 67 - PI Controller Tuned 1.6	24
Figure 68 - PI Controller 4	25
Figure 69 - PI Controller Values	25
Figure 70 - PI Controller I = 0.5	26
Figure 71 - PI Controller = 0.6	26
Figure 72 - PI Controller 0.6	27
Figure 73 - PI Controller 4	27
Figure 74 - Tuned PI Controller	28
Figure 75 - PID Controller	28
Figure 76 - PID Controller 0.005	29
Figure 77 - PID Controller Tuned Values	29
Figure 78 - PID Controller 0.01	30
Figure 79 - PID Controller 0.01 Values	30
Figure 80 - PID Controller D value 0.0001	31
Figure 81 - PID Controller 0.0001	31
Figure 82 - Comparison PID Controller	32
Figure 83 - Controller Comparison	32
Figure 84 - Comparison Controllers	33

Assignment 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

59/100

PAGE 1

PAGE 2

PAGE 3

Text Comment. This is not a general solution, $G(S)$ is the open loop transfer function

PAGE 4

Text Comment. a good diagram

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

Text Comment. This table should have showed the results for the 3 different values of K

Text Comment. You have already shown that with unity feedback the pole lies at -3.18 which you have already said is stable.

PAGE 11

Text Comment. avoid use of first person

Text Comment. where did this equation come from?

Text Comment. this needs to be explained

Text Comment. where does this equation come from?

PAGE 12

Text Comment. The motor model should have been developed and explained first.

PAGE 13

Text Comment. $G(S)$ is a transfer function not a general solution

Text Comment. you mean -150.5 and -5.56

Text Comment. a useful diagram

Text Comment. The pole actually moved in task 1, it was a first order system and there were never 2 poles

PAGE 14

QM

Spelling

Spelling error

Text Comment. you could have explained each part of this motor model

QM

Spelling

Spelling error

PAGE 15

Text Comment. How do these results compare with the calculated ones?

PAGE 16

QM

Spelling

Spelling error

QM

Word/s missing

PAGE 17

Text Comment. How do you know it has stalled?

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

Text Comment. This is a good table but what are you conclusions

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

Text Comment. This is a good table but what are you conclusions

PAGE 30

PAGE 31

PAGE 32

PAGE 33

Text Comment. What are your overall conclusions from this table?

PAGE 34

PAGE 35

Text Comment. Don't write in the first person

PAGE 36

PAGE 37

PAGE 38

GRADING FORM: CSG-RUBIC

RESEARCH-INFORMED LITERATURE

 n/a

KNOWLEDGE AND UNDERSTANDING



Section 1: Explanation of first order system – Very good (7/10) Section 2: Explanation of DC motor model – Generally good, but there is much information that should have been explained about the model before introducing the equations. (6/10) Section 3: Explanation of types of speed controllers – Very good explanations given. (7/10)

20

ANALYSIS



Section 1: Testing of first order system – Very good testing but your analysis for the feedback contradicts itself and your table should have compared results for different values of K. (6/10) Section 2: Testing of DC motor model – Good tests carried out but a more detailed comparison with theory needed. Results should be summarized in a table for comparison. (6/10) Section 3: Testing of speed controllers – Good evidence of testing with results summarized in a table for comparison for each type of controller. However, no conclusions have been drawn at the end of each section except for the last test of the three controllers. (10/20)

25

Section 4: Conclusions and recommendations –
Little conclusions given and no recommendations.
(3/10)

PRACTICAL APPLICATION AND DEP.



Sections 1, 2 and 3: Use of Simulink and
presentation quality of results - Very good use of
the software and results are well presented. (8/10)

8

SKILLS FOR PROFESSIONAL PRAC.



All sections – Well structured and presented.
however, there are spelling errors and you should
not write in the first person unless asked to. (6/10)

6