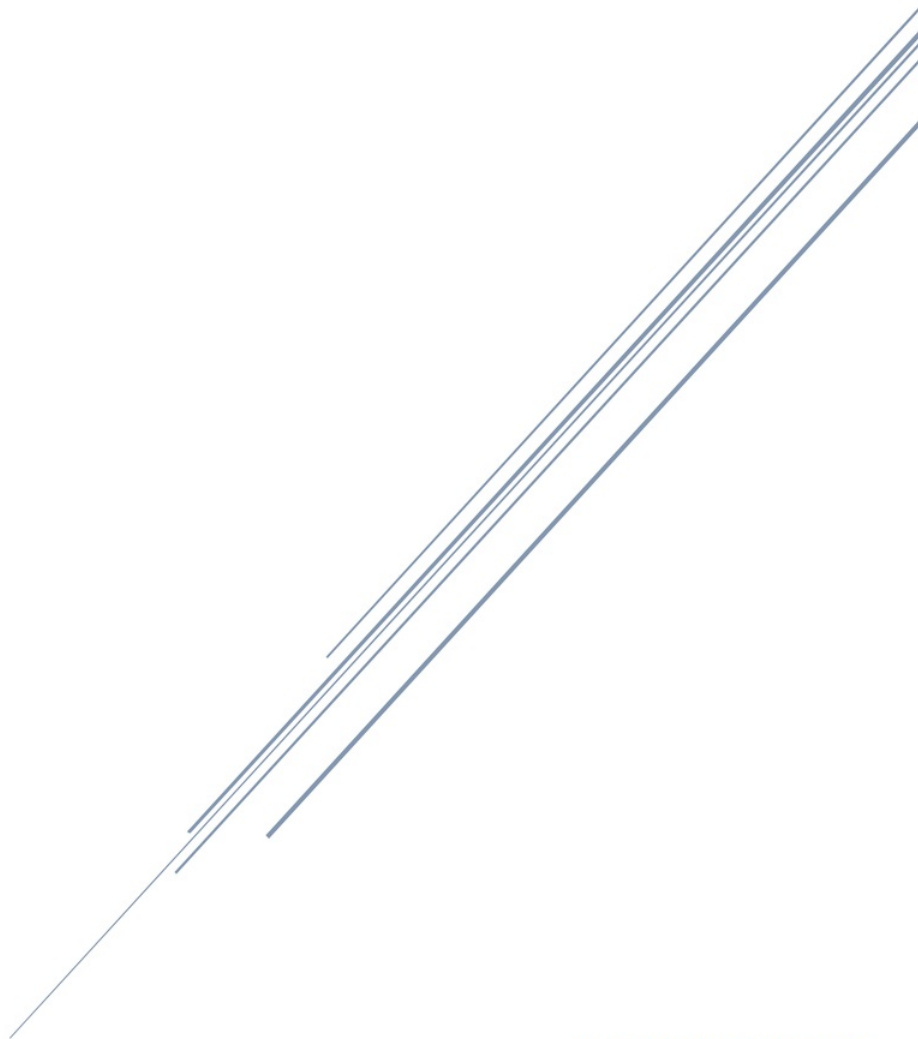


assignment 2

by Trystan Daniels

FILE	11396_TRYSTAN_DANIELS_ASSIGNMENT_2_158265_2068390392.DOCX		
TIME SUBMITTED	11-MAY-2018 07:01PM (UTC+0100)	WORD COUNT	1807
SUBMISSION ID	89019701	CHARACTER COUNT	9534

HNC ELECTRICAL/ELECTRONIC ENGINEERING
ELECTRICAL PRINCIPLES,
METHODS AND SIMULATION
PSDE5009 – ASSIGNMENT 2



TRYSTAN DANIELS

Table of Contents

INTRODUCTION.....	2
1. First order control system (task1).....	3
2. Dc motor model analysis (task2).....	8
3. Types of motor speed controller (task3, 4, 5 and 6).....	14
P controller.....	14
PI controller.....	19
PID controller	22
Comparison of P, PI and PID controllers	26
Conclusion.....	28

INTRODUCTION

Throughout this assignment theory will be predicted on first and second order systems. A model of DC motor will be developed and a P, PI and PID controller will be analysed and tested. The theory will be proven after using the aid of matlab to support the predicted results and outcomes.

The first task will be a first order system, producing a overall transfer function, drawing argand diagrams to show whether it's a stable system. The minimum gain will be calculated for the system to be stable.

The second task will be to analyse a DC motor model and simulate, developing a mathematical model of the motor and load. Looking into damping factor, stall torque, load speed and no load speed. This will be analysed and simulated on matlab.

The next tasks will be on a P, PI and PID controller and the effects of increasing and reducing the value of P, I and D. This will be simulated and analysed on multisim driving the Dc motor model used in the task above. The rise time, overshoot and steady state error will be analysed and compared. Finally all three controllers will be compared to see the results.

1. First order control system (task1)

This section should begin with an explanation

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

$$H(s) = 1$$

Without feedback

Open loop pole is:

$$s - 0.42 = 0$$

$$s = 0.42$$

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

you could have drawn the argand diagram here and shown the open loop system is unstable

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

Multiply top and bottom by (s - 0.42)

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

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

$$T(s) = \frac{3.6}{s + 3.18}$$

Root of the denominator gives the closed loop pole

$$s + 3.18 = 0$$

$$s = -3.18$$

Pole is in the left plane therefore stable.

$$H(s) = K$$

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

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

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

good development but a more detailed explanation is needed.

Pole is found when:

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

For pole at $s=0$

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

$$3.6k = 0.42$$

$$k = 0.116$$

This is k_{min} to stabilise the system.

Simulating on matlab

Num = [3.6];

Den = [1 -0.42];

Rlocus(num,den);

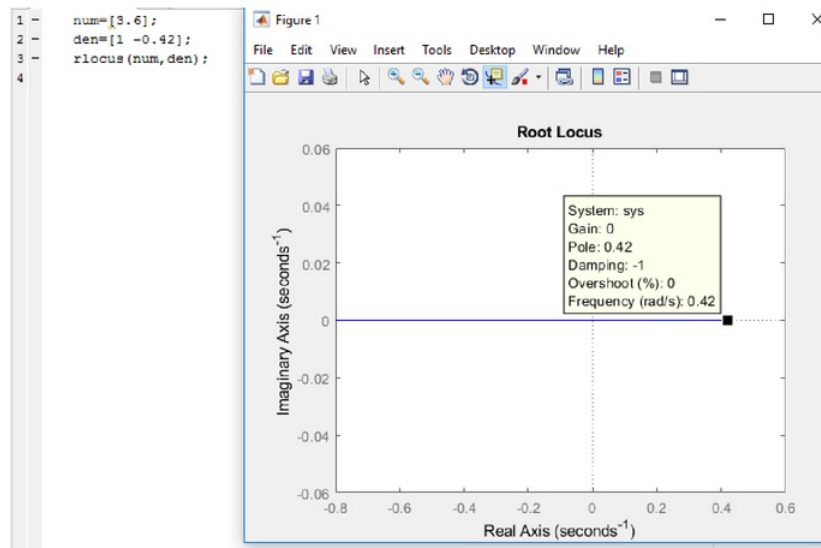


Figure 1

Figure 1 shows the pole on the argand diagram in the right plane (positive), therefore this means its an unstable system. This is with no gain at all. This backs the theory above.

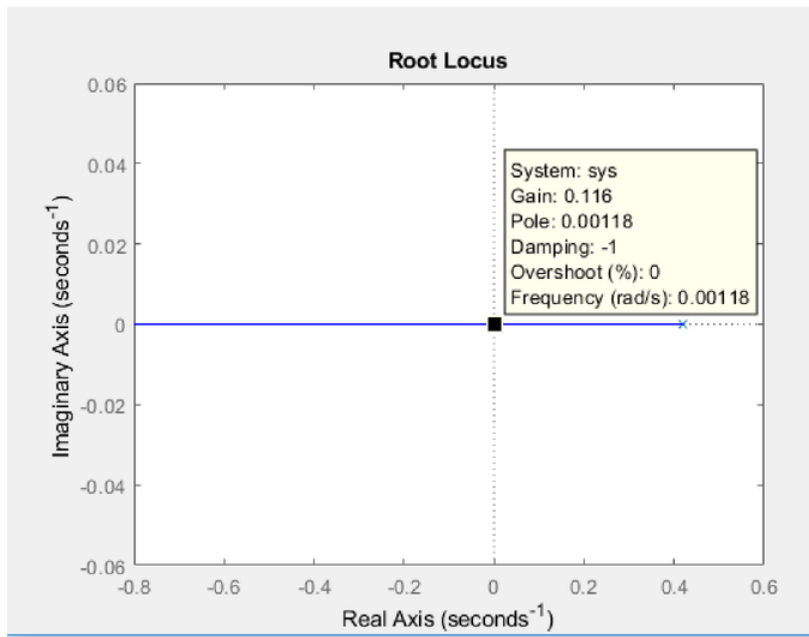


Figure 2

Figure 2 shows the system with a feedback gain of 0.116. With this feedback the pole has reached 0. This is the Kmin value as stated through theory.

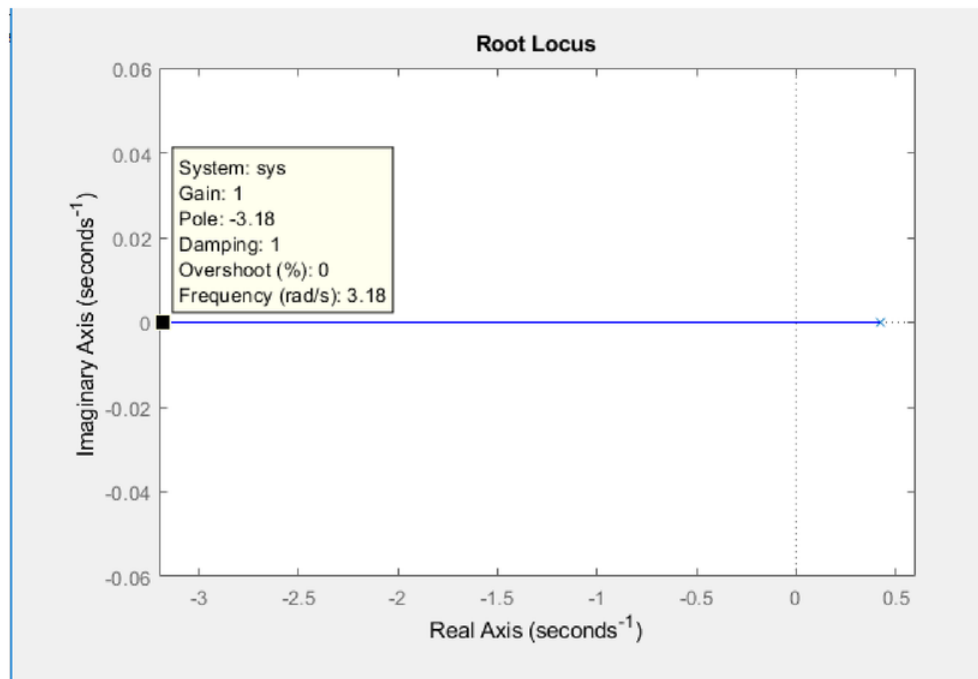


Figure 3

The figure above shows the system with a gain of 1. The argand diagram shows the system would be stable with a feedback gain of 1 as the pole is in the left plane.

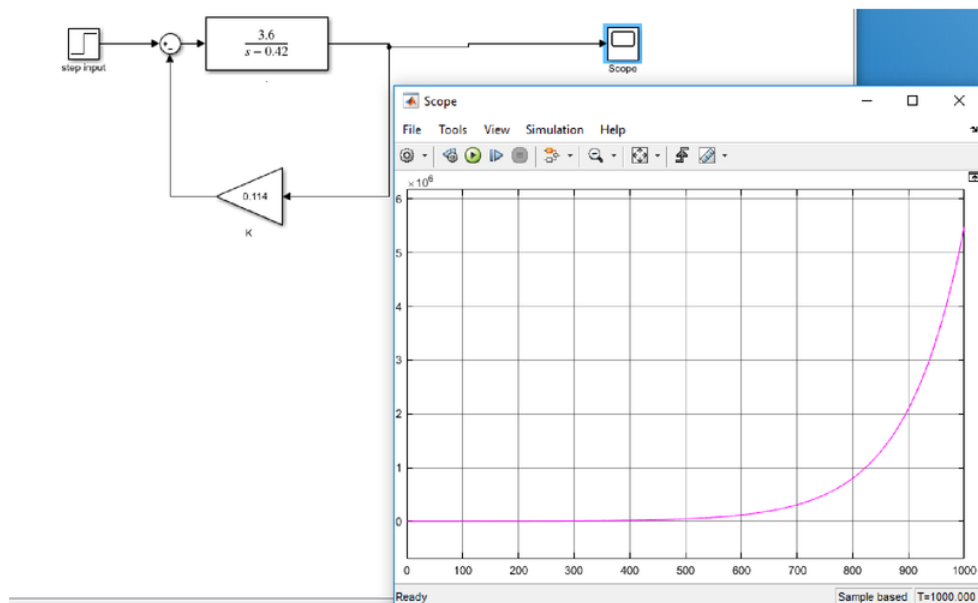


Figure 4

Figure 4 shows the step response with a 0.114 feedback gain. The response graph show the system slightly unstable.

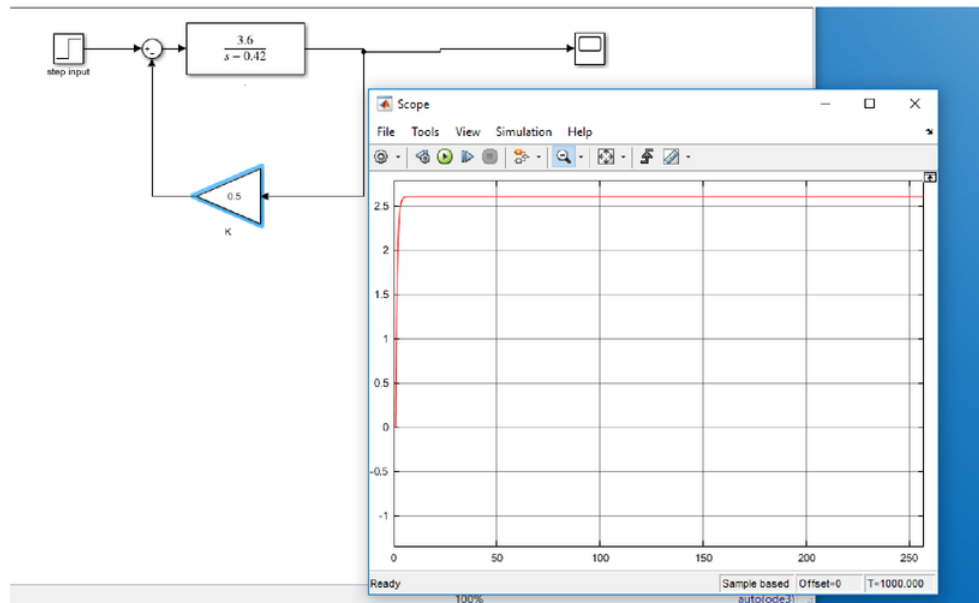


Figure 5

Figure 5 above shows the step response with a 0.5 gain. The graph shows a sharp response.

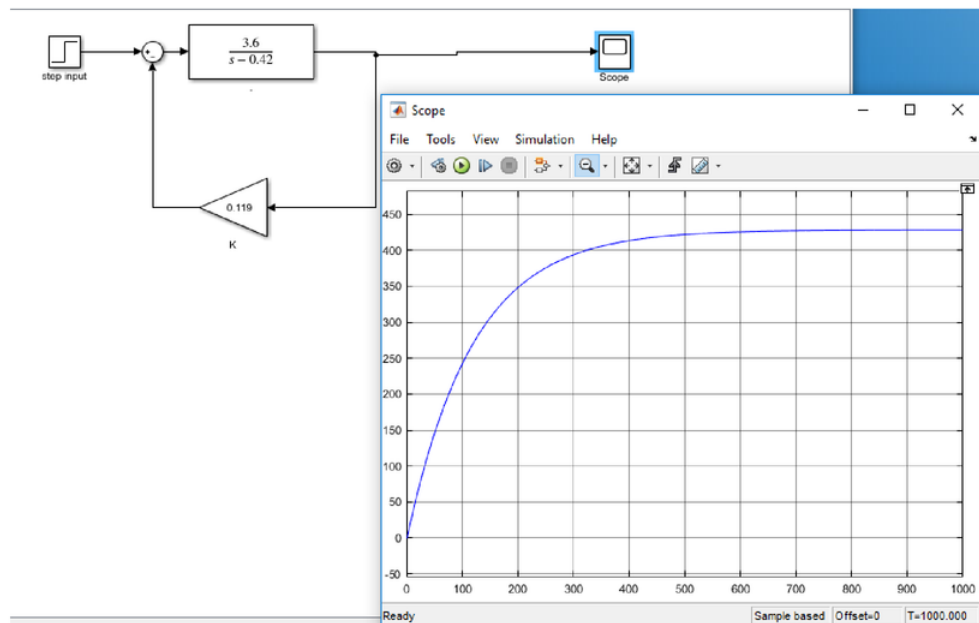


Figure 6

Figure 6 shows the step response with a 0.119 gain, the response is slow and overdamped.

You could have measured the overall gain and the time constant for comparison

2. Dc motor model analysis (task2)

Motor analysis

$R_a = 7.8$

$L_a = 0.05$

Explanation needed together with block diagram

$b = 15.519 \times 10^{-6} \text{ NM/rads}$

$J = 2.14 \times 10^{-4} \text{ kgm}^2$

$K_e = 0.983 \text{ v/rads}$

$K_t = 0.09 \text{ Nm/A}$

Max armature voltage = 40v

$$\begin{aligned} \omega_n^2 &= \frac{R_a b + K_t K_e}{L_a J} \\ &= \frac{7.8 \times 15.519 \times 10^{-6} + 0.09 \times 0.983}{0.05 \times 2.14 \times 10^{-6}} \end{aligned}$$

$$= 838.135$$

$$\begin{aligned} \omega_n &= \sqrt{838.135} \\ &= 28.95 \end{aligned}$$

$$\begin{aligned} 2\zeta\omega_n &= \frac{b}{J} + \frac{R_a}{L_a} \\ &= \frac{15.519 \times 10^{-6}}{2.14 \times 10^{-4}} + \frac{7.8}{0.05} \\ &= 156.07 \end{aligned}$$

$$\begin{aligned} \zeta &= \frac{0.0725 + 156}{2 \times 28.95} \\ &= 2.696 \end{aligned}$$

what does this mean?

$$\begin{aligned} K &= \frac{8411.21}{838.135} \\ &= 10.04 \end{aligned}$$

Explanations needed

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

$$= 8411.21$$

$$\begin{aligned}\Omega NL &= \frac{KTVa}{bRa + KTKE} \\ &= \frac{(0.09)(24)}{15.519 \times 10^{-6}(7.8) + (0.09)(0.0983)} \\ &= 240.86\end{aligned}$$

*unit
s?*

$$\begin{aligned}\Omega L &= \frac{Va - \frac{RaTL}{KT}}{\frac{Rab}{KT} + KE} \\ \Omega L &= \frac{24 - 2.167}{1.34 \times 10^{-3} + 0.0983} \\ &= 219.12\end{aligned}$$

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

$$A = 1$$

$$B = 156.0725$$

$$C = 838.135$$

$$\begin{aligned}s &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ &= \frac{-156.0725 \pm \sqrt{156.0725^2 - 4(1)(838.135)}}{2} \\ &= -5.57 \text{ or } -150.5\end{aligned}$$

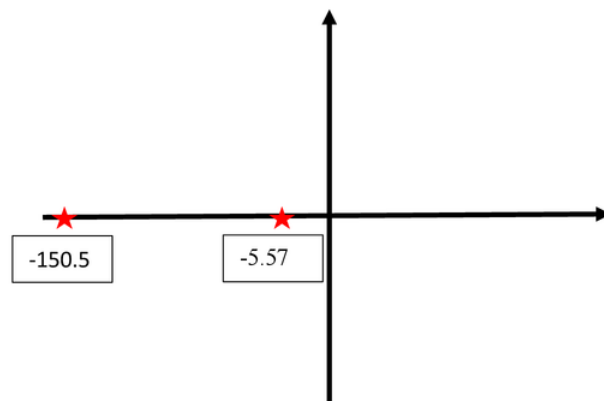


Figure 7

Figure 7 shows the argand diagram from theory prediction. This diagram shows that the system is stable as the poles are in the left plane. Also the system is overdamped with one dominant pole being at -5.57.

Below in figure 8 and 9 are the matlab model of the motor system. Figure 8 show the no load model and figure 9 shows the load model.

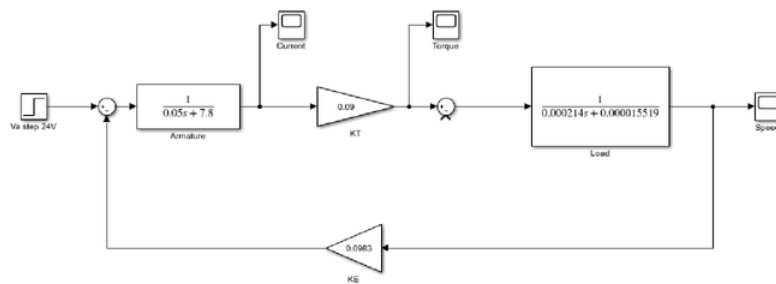


Figure 8

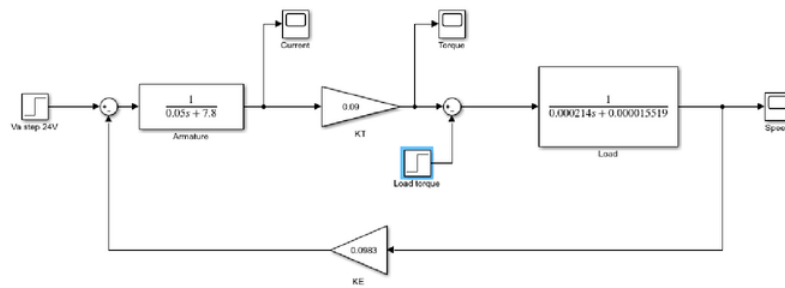


Figure 9

Below in figure 10 shows the speed graph of the loaded model of 0.025nm after 10s. The first 10s is the no load speed for the motor being 240.9. This matches the calculated theory above. After 10s when the motor is put underload the speed reduces to 219.1, this again supports the predicted theory above.

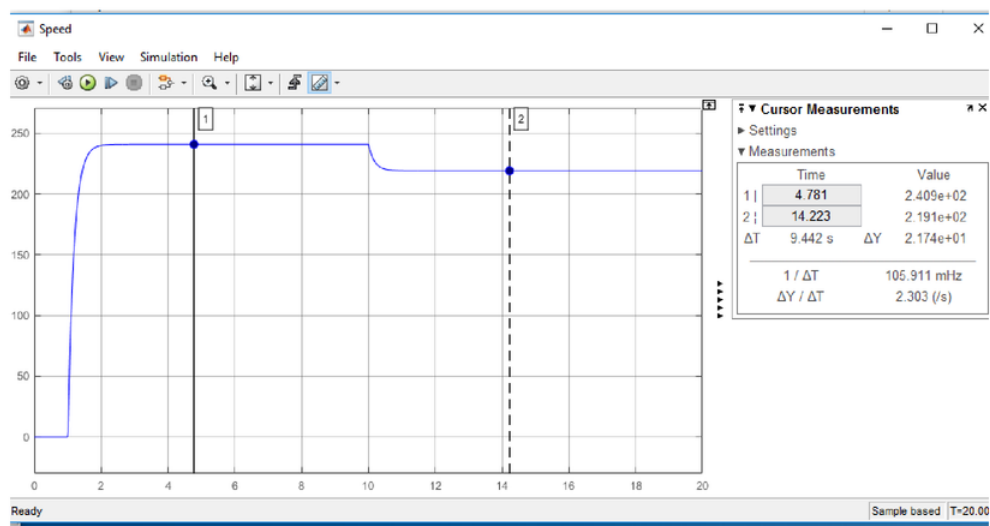


Figure 10

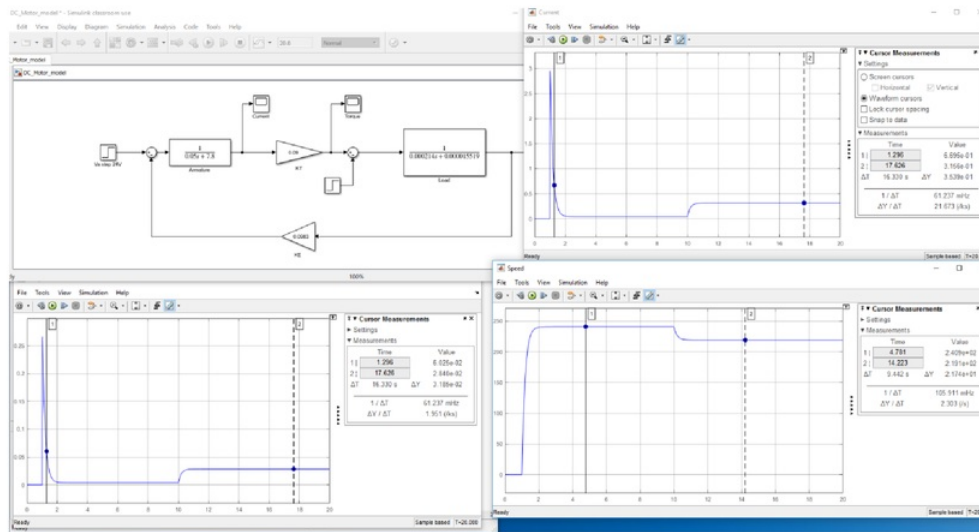


Figure 11

Figure above shows the current and torque graphs, after the load is added both graphs rise in value and both graphs are identical.

$$\begin{aligned}
 TM(stall) &= \frac{KT(Va)}{Ra} \\
 &= \frac{0.09 \times 24}{7.8} \\
 &= 0.277
 \end{aligned}$$

Below in figure 12 shows the calculated stall load inserted into the model and the speed graph. The graph shows the motor stalling once the load is added as predicted.

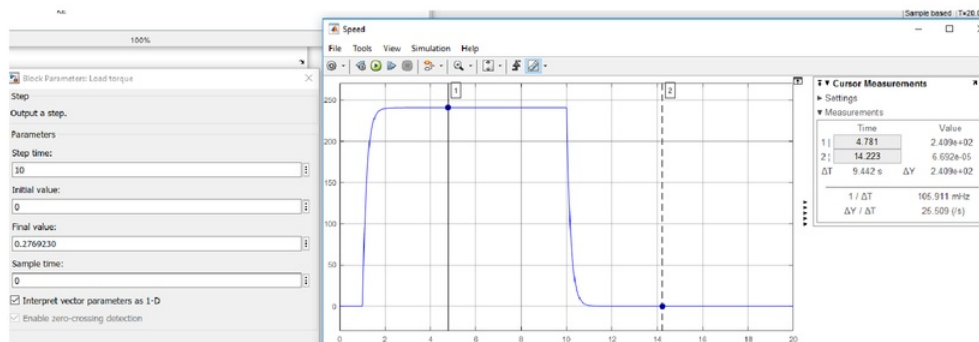


Figure 12

how do you know it is stalled?

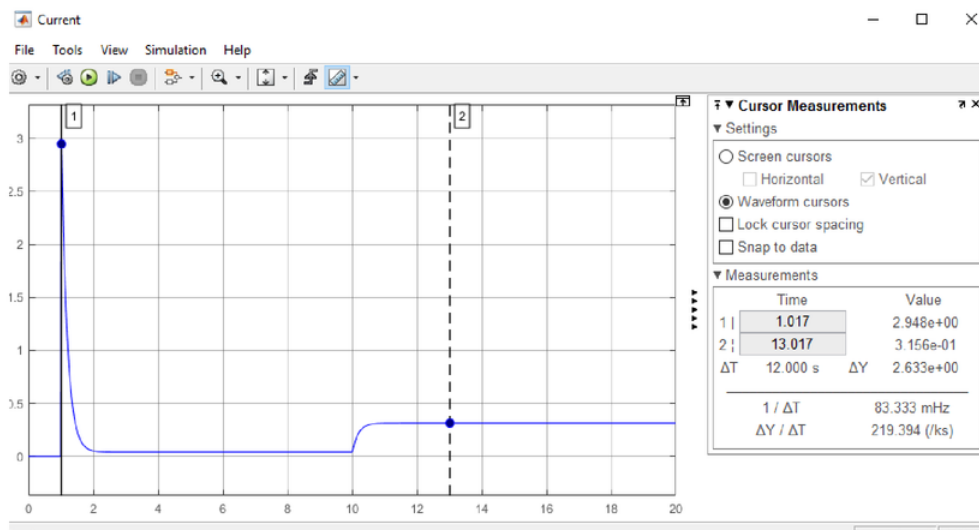


Figure 13

Above in figure 13 is the current graph for the model. The current spikes at 1s then returns to a low value until the load is added at 10s therefore creating more current to be drawn.

3. Types of motor speed controller (task3, 4, 5 and 6)

P controller

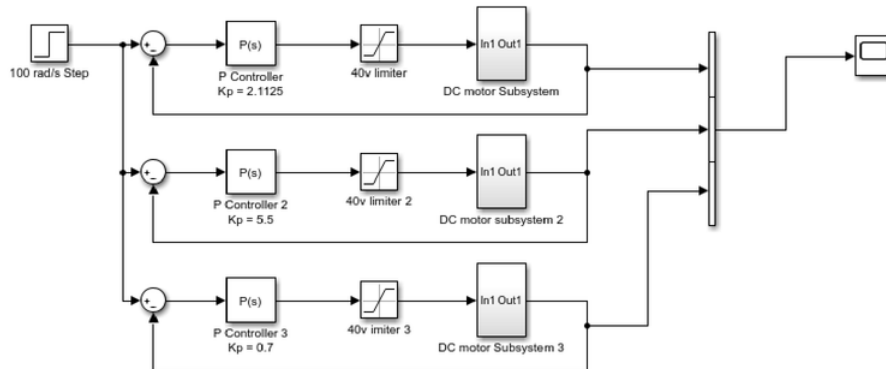


Figure 14

Figure 14 shows the three P controllers, with different P values of 2.1125 as a tuned value, 5.5 and 0.7 to simulate the change in P.

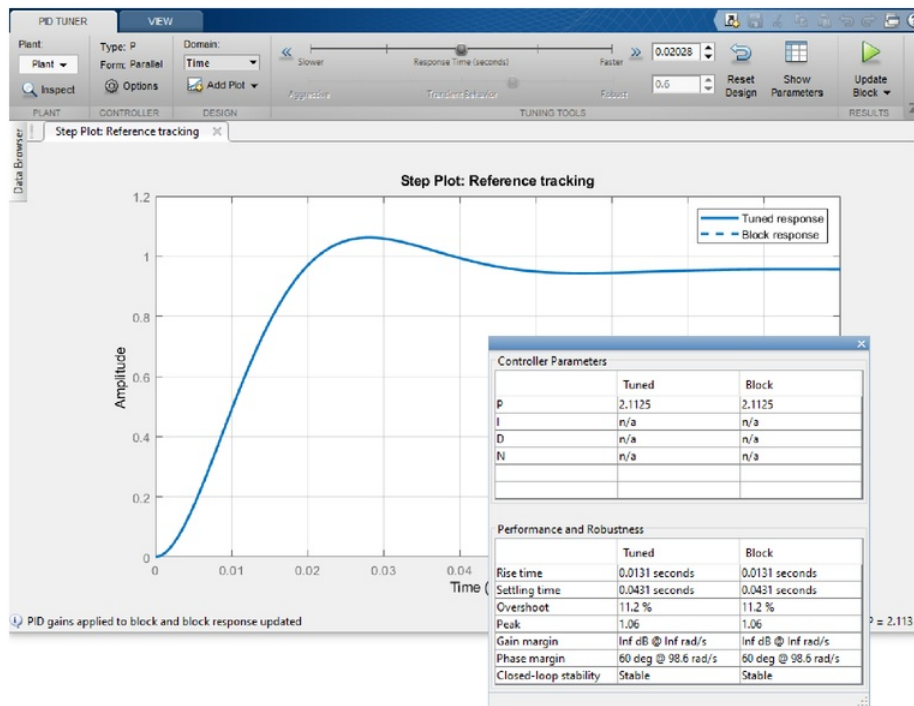


Figure 15

Figure 15 shows the P values being tuned to 2.1125. With this K_p value, the overshoot percentage of 11.2%.

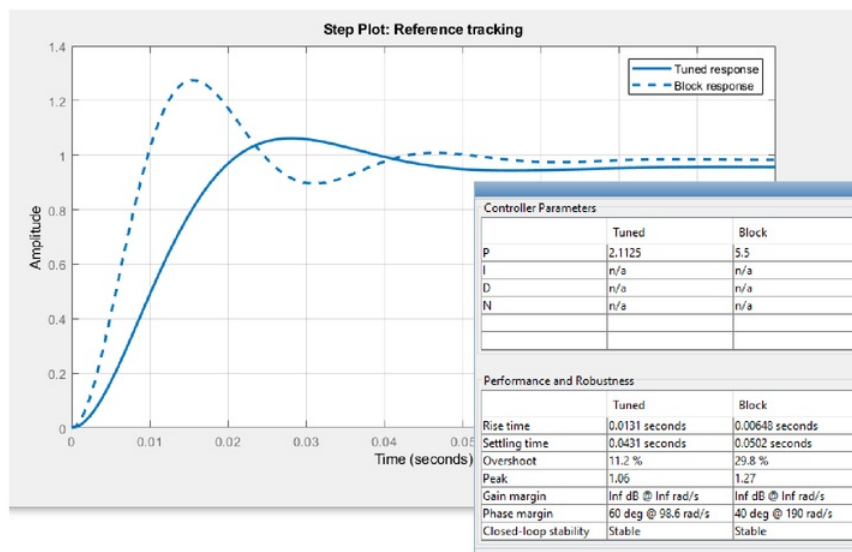


Figure 16

Figure 16 shows the second value chosen at k_p of 5. This gives an overshoot of 29.8%, however this gives less of a steady state error and gives a faster rise time.

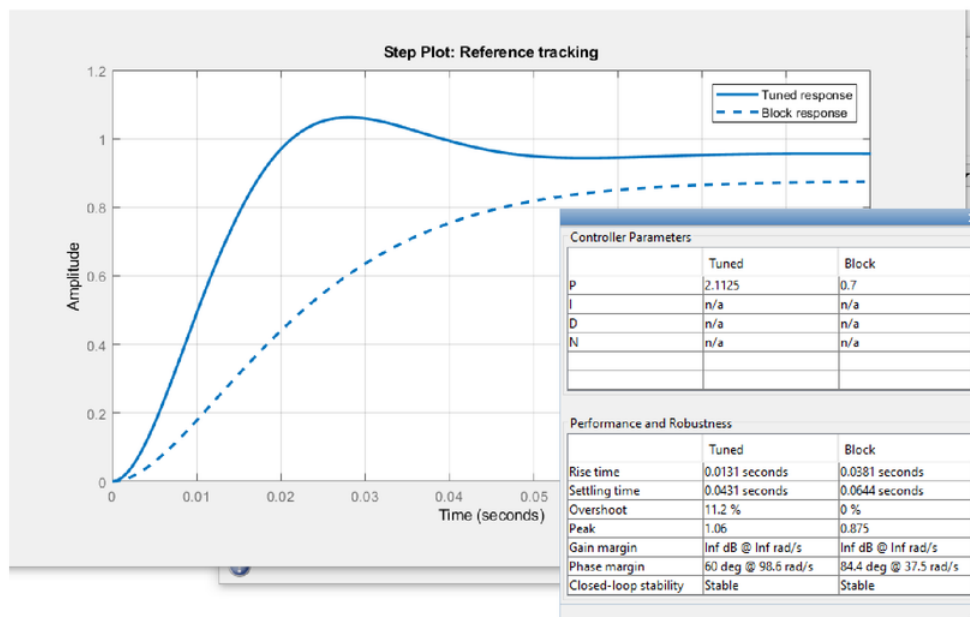


Figure 17

Above in figure 17 is the 3rd k_p value of 0.7. This gives a slower rise time, no overshoot and more of a steady state error.

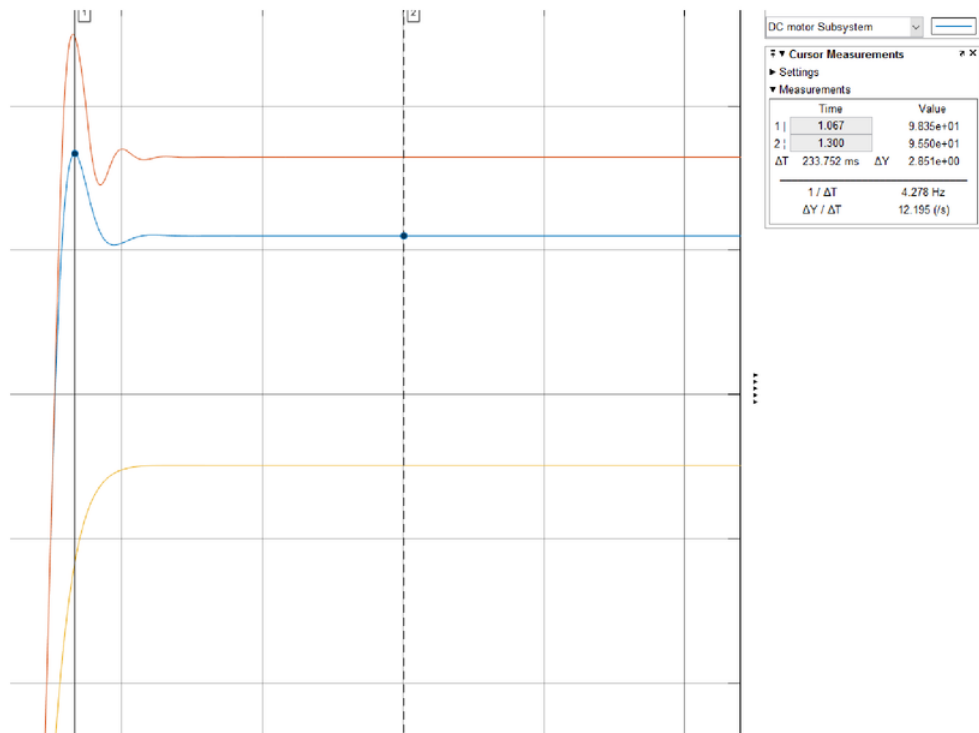


Figure 18

Figure 18 shows the step response for the tuned Kp value (Blue). Here you can see the steady state error is worse than the subsystem 2 (Kp=5) and a slight overshoot.

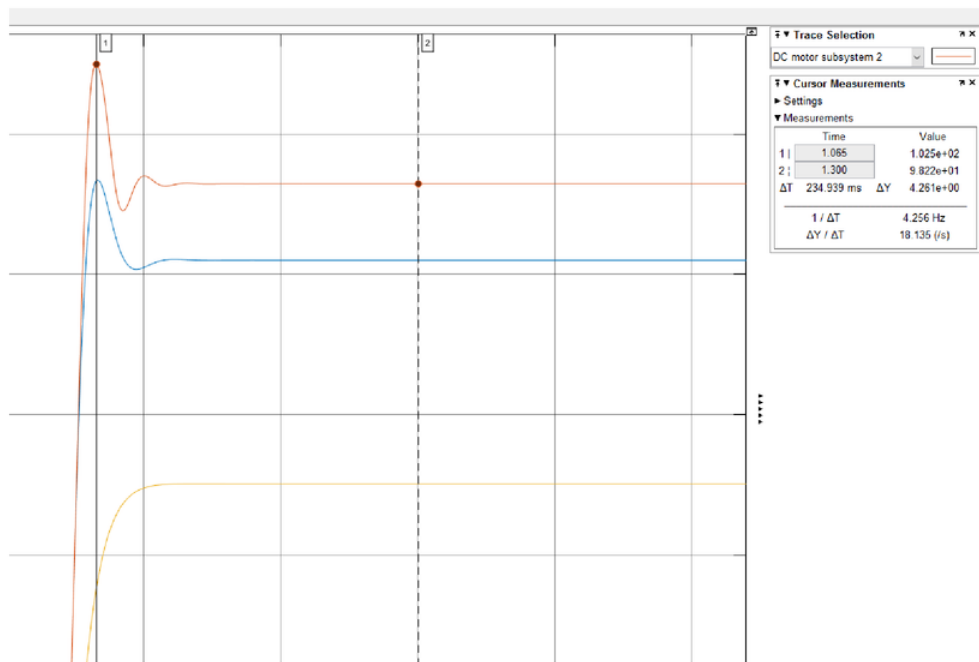


Figure 19

Figure 19 shows subsystem 2 with a Kp value of 5. This has a large overshoot with less steady state error than the other 2 subsystems. The rise time is also a little quicker than the tuned subsystem.

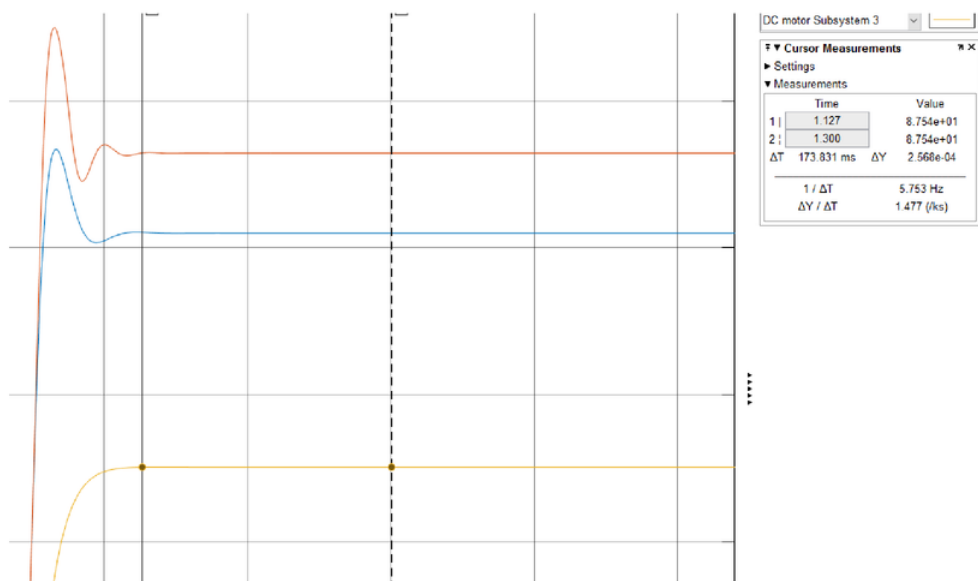


Figure 20

Figure 20 shows the worst of the three systems being the lowest Kp value. Having no overshoot however a slow rise time and a large steady state error.

Kp value	Rise time (S)	Transient error	Overshoot (%)
2.1125	0.0131	Med	11.2
5	0.00648	Low	28.9
0.7	0.0381	High	0

From the table above it can be said that as you increase the value of P the rise time decreases and the steady state error however the overshoot increases.

PI controller

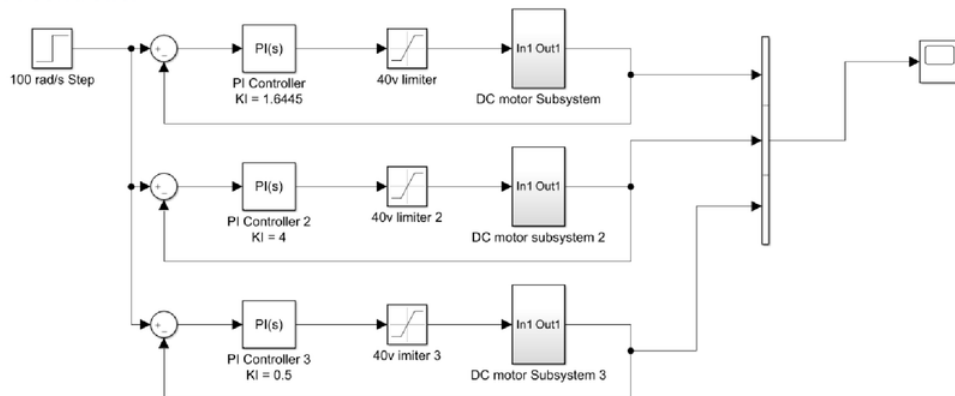


Figure 21

Figure 21 shows the 3 PI controllers designed driving DC motors. The tuned value of PI is 1.6445, therefore I choose the two other values of PI of 4 and 0.5 to simulate the difference in adjusting I. the P value is keep the same in the 3 systems.

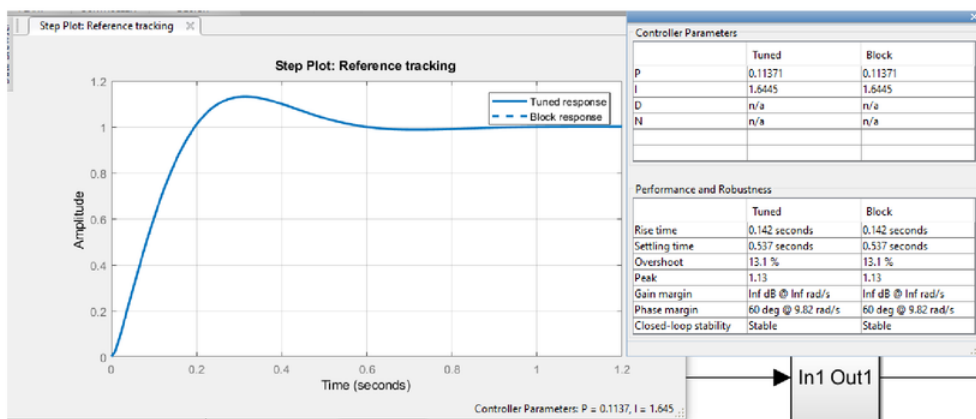


Figure 22

Figure 22 shows the tuned value of KI of 1.6445, with a rise time of 0.142 and overshoot of 13.1%. There is no overshoot.

good conclusion

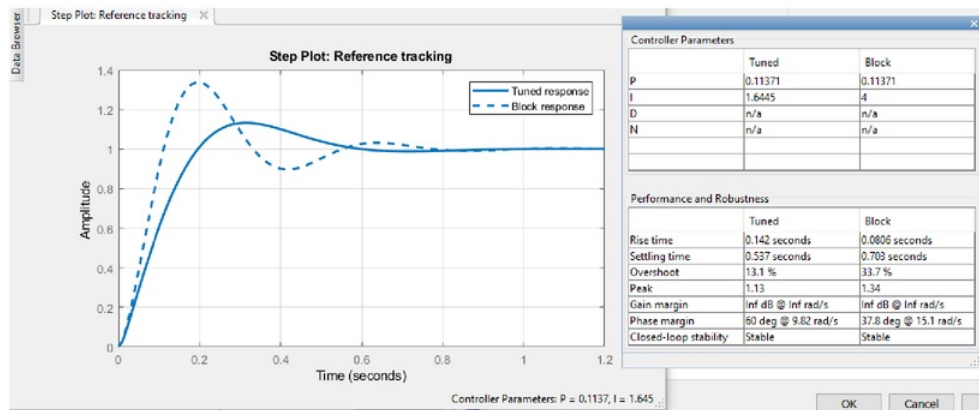


Figure 23

Figure 23 shows the second system with a KI of 4. There is a bigger overshoot of 33.7% with no transient error.

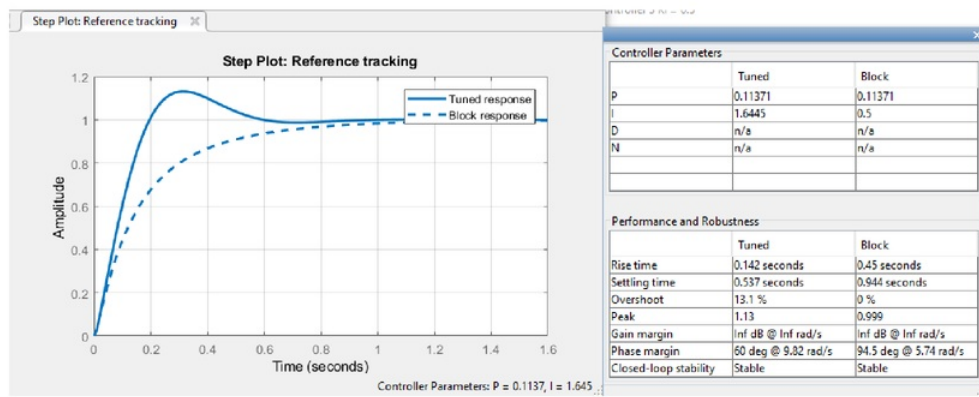


Figure 24

The figure above shows the third system with a KI of 0.5. The rise time is slow compared to the other circuits being 0.45. There is no overshoot for this Ki value however.

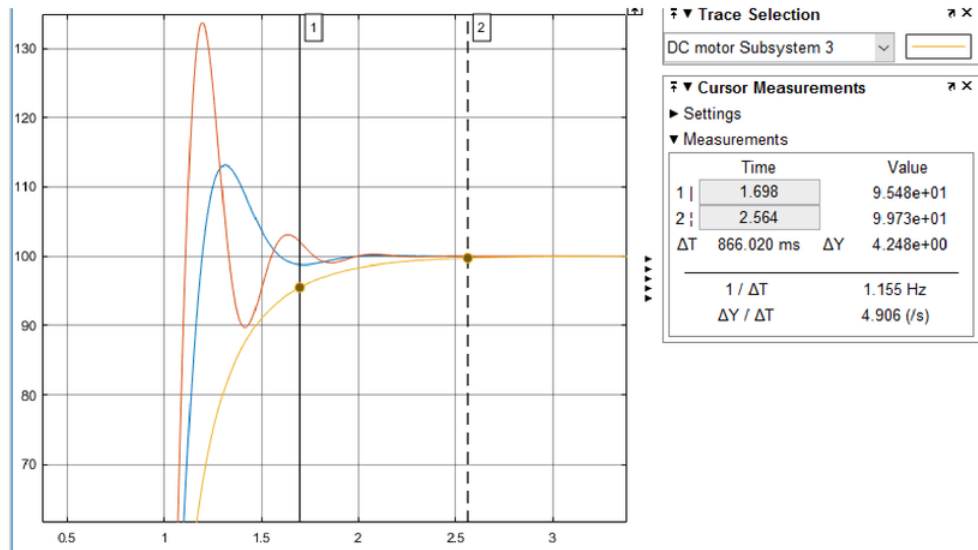


Figure 25

Figure 25 shows circuit three the smallest value of K_i , here it can be seen there is no overshoot and a slow rise time.

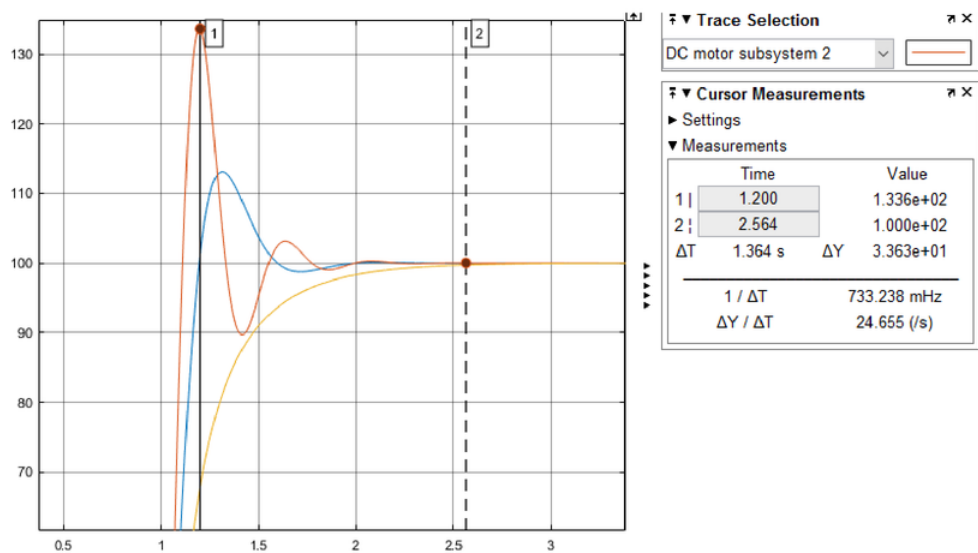


Figure 26

Figure 26 shows the second circuit with a K_i of 4. This has a large overshoot, but a very quick rise time.

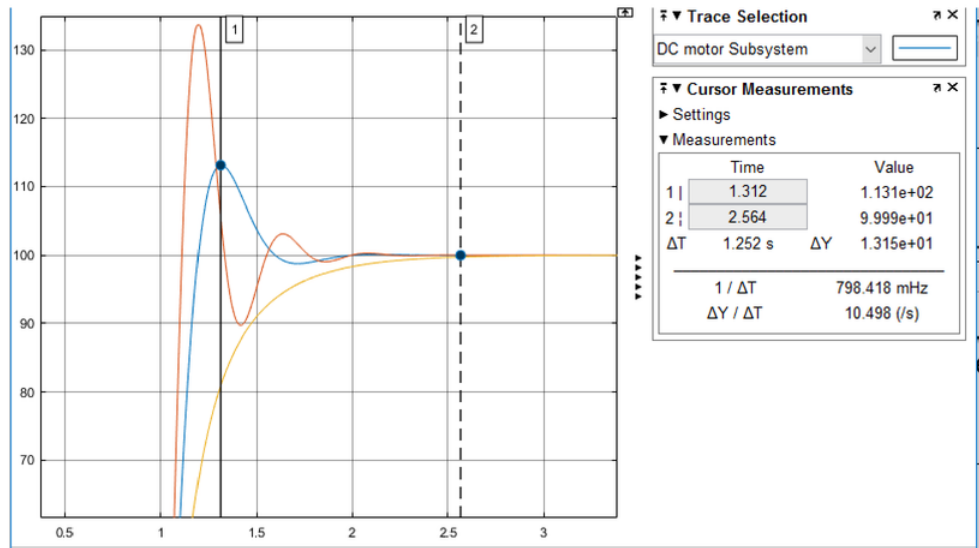


Figure 27

Figure 27 shows the tuned value of KI of 1.6445. it has a slight overshoot with a fast rise time.

Ki value	Rise time (S)	Transient error	Overshoot (%)
1.6445	0.142	--	13.1
4	0.0806	--	33.7
0.5	0.45	--	0

In conclusion I part will get rid of the transient error. As you increase the value of I the rise time reduces however the overshoot increases, this is proportional.

PID controller

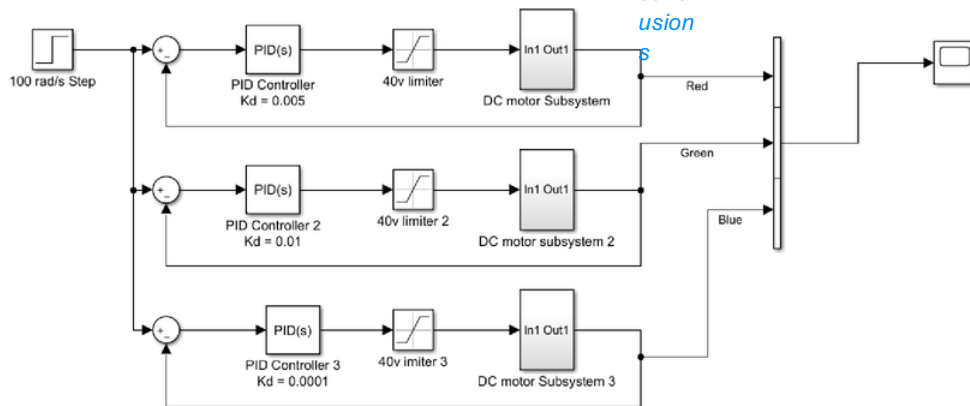


Figure 28

Figure 28 shows the three PID controllers. For this DC motor drive system the best tuned system has no Kd which means a PI controller would be the best option for this system. However some Kd were added to show the effect of D.

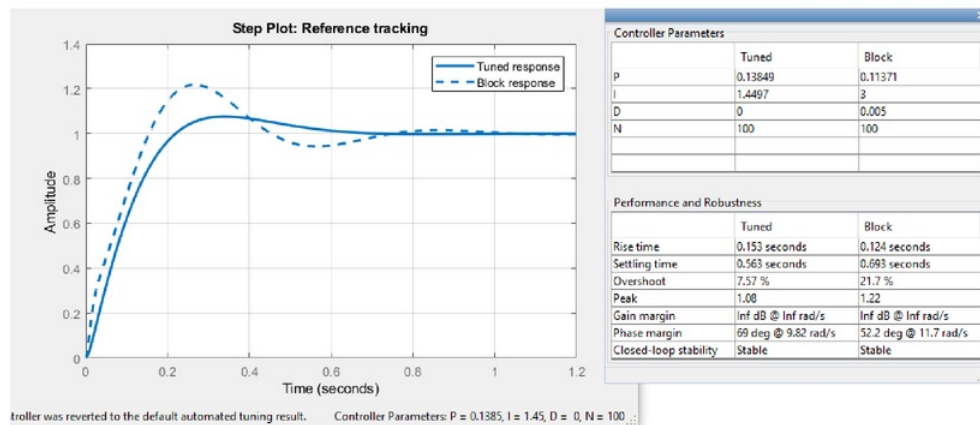


Figure 29

Figure 29 shows the Kd of 0.005, this gives a faster rise time however gives overshoot.

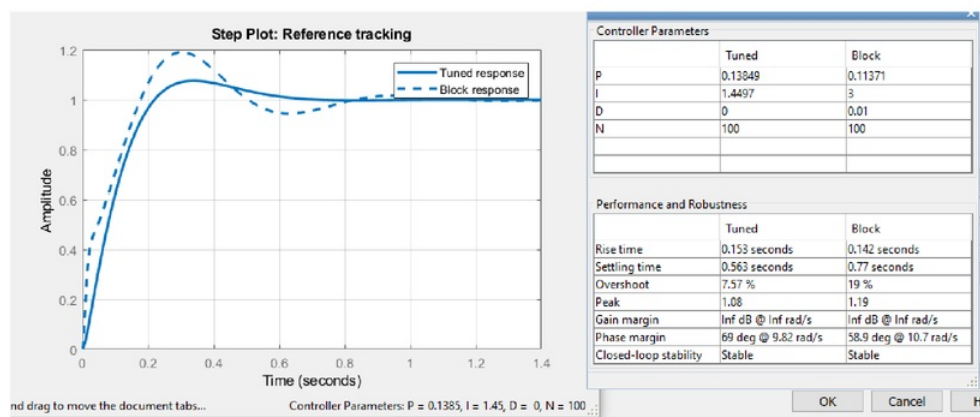


Figure 30

Figure 30 shows another Kd value of 0.01, this gives a slower rise time however a little less overshoot.

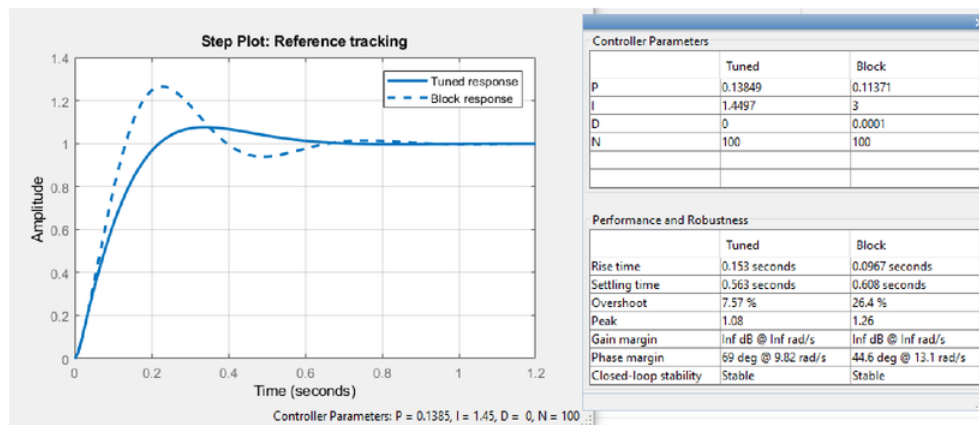


Figure 31

Figure 31 shows the smallest value of K_d . This gives a faster rise time. However, it gives more overshoot as the K_d is very small.

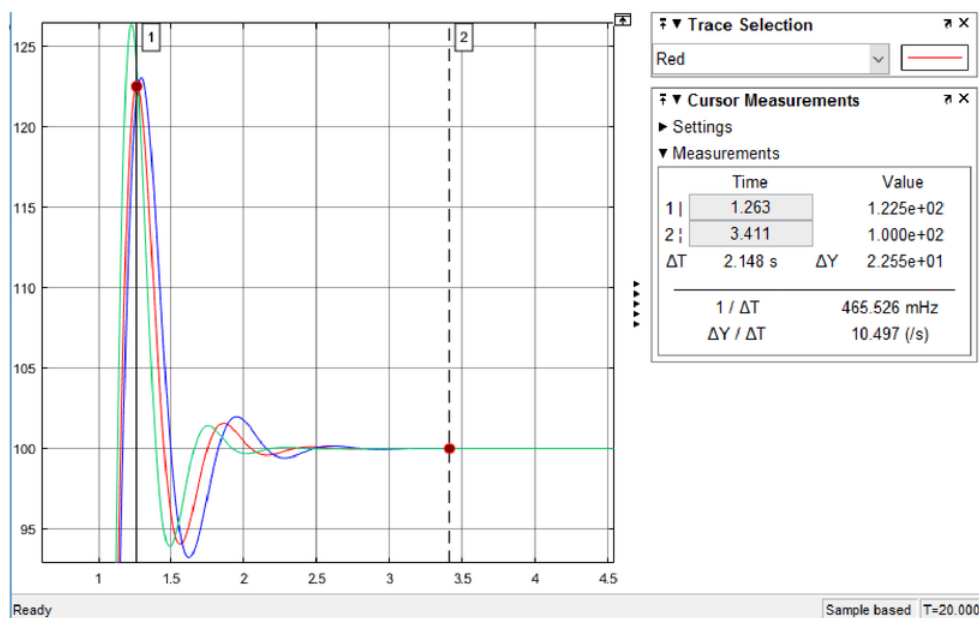


Figure 32

Figure 32 above shows the graph for all three different values of K_d . The highlighted line of 0.01 of K_d . This gives the least overshoot and reasonable rise time.

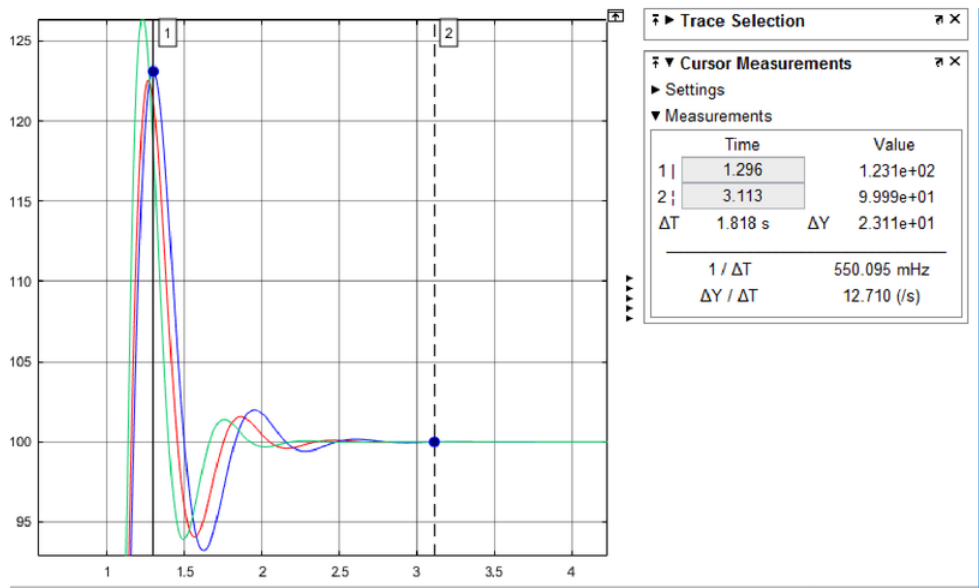


Figure 33

Figure 34 shows the blue line of K_d 0.005, this has a bigger overshoot due to $1e K_d$ in the controller.

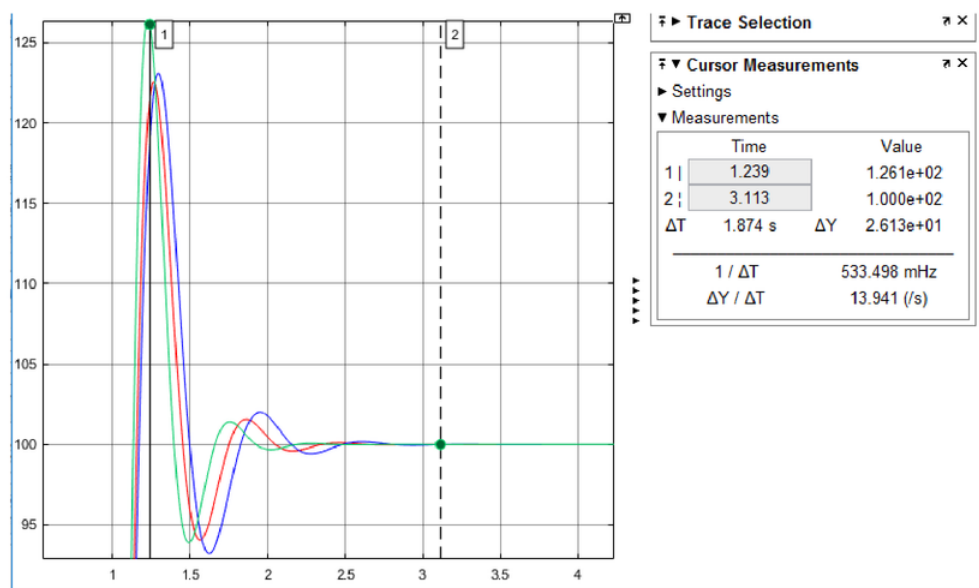


Figure 34

Figure 35 shows the lowest value of K_d , hence such high overshoot.

Kd value	Rise time (S)	Transient error	Overshoot (%)
0.005	0.124	--	21.7
0.01	0.142	--	19
0.0001	0.0967	--	26.4

The above table is the comparison of the change in Kd in a PID controller. As the value of Kd increases the rise time slows, however the amount of overshoot decreases.

Comparison of P, PI and PID controllers

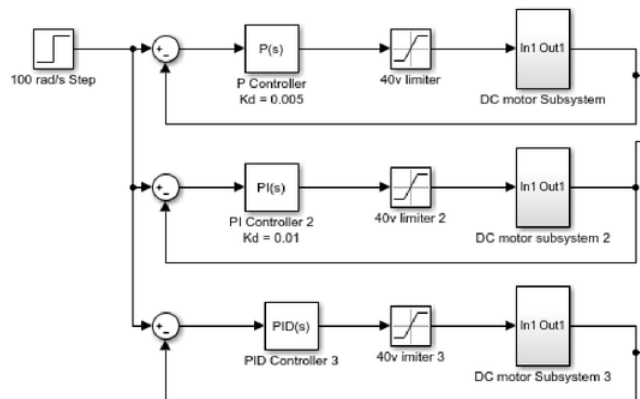


Figure 35

Figure 35 shows the three separate systems. One P controller, One PI controller and one PID controller tuned to their best for a comparison between the different controllers.

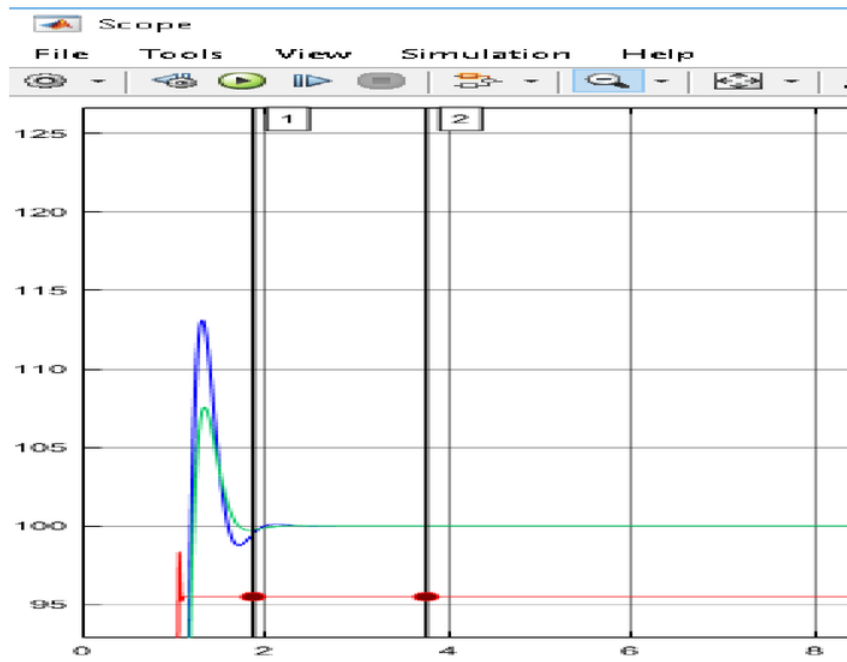


Figure 36

Figure above shows the three different controllers, the red line shows the P controller which has a large steady state error, for this system the P controller doesn't suit. The blue line is the PI controller which has a fast rise time, a high overshoot and not steady state error. The green line, the PID controller is the best as for is lower overshoot, good rise time and no steady state error.

Controller	Rise time (S)	Transient error	Overshoot (%)
P	slow	high	0
PI	fast	--	High
PID	fast	--	Low

Conclusion

Through theory prediction and simulation, systems can be simulated actually using software such as matlab, this reduces a lot of mathematical calculations. Knowledge of system stability were gained throughout the assignment and how to work out the minimum gain to ensure the system is stable.

With PID controllers knowledge was gained in the difference and increasing P, I and D values were gained where PID not necessarily was the best controller in all situations, with this assignment the best suited response would be a PI controller.

*there is much more you could have
said*

assignment 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

62/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

Text Comment. This section should begin with an explanation

Text Comment. you could have drawn the argand diagram here and shown the open loop system is unstable

PAGE 5

Text Comment. good development but a more detailed explanation is needed.

PAGE 6

PAGE 7

PAGE 8

Text Comment. You could have measured the overall gain and the time constant for comparison

PAGE 9

Text Comment. Explanation needed together with block diagram

Text Comment. what does this mean?

Text Comment. Explanations needed

PAGE 10

Text Comment. units?

Text Comment. graphs are too small

QM

Spelling

Spelling error

Text Comment. how do you know it is stalled?

Text Comment. good conclusion

Text Comment. good table

Text Comment. good conclusions

Text Comment. there is much more you could have said

GRADING FORM: CSG-RUBIC

TRYSTAN DANIELS

62

RESEARCH-INFORMED LITERATURE

 n/a

KNOWLEDGE AND UNDERSTANDING

 Section 1: Explanation of first order system – Very little explanation given except by equations (which are well presented). This section needs more words! (5/10) Section 2: Explanation of DC motor model – Good equations but again, explanations are needed. (5/10) Section 3: Explanation of types of speed controllers – Good. (7/10)

17

ANALYSIS

 Section 1: Testing of first order system – Good evidence of testing using the Matlab rlocus function and Simulink features. You could have measured the time-constant and overall gain values. (7/10) Section 2: Testing of DC motor model – . Good results with some analysis. (7/10) Section 3: Testing of speed controllers – Excellent with some very good analysis. (16/20) Section 4: Conclusions and recommendations – Briefly stated but no

32

recommendations. (2/10)

PRACTICAL APPLICATION AND DEP.



Sections 1, 2 and 3: Use of Simulink and presentation quality of results - Excellent use of the software except that some of the graphs are very small. (8/10)

8

SKILLS FOR PROFESSIONAL PRAC.



All sections – Well presented and structured. Figures should be listed. (5/10)

5