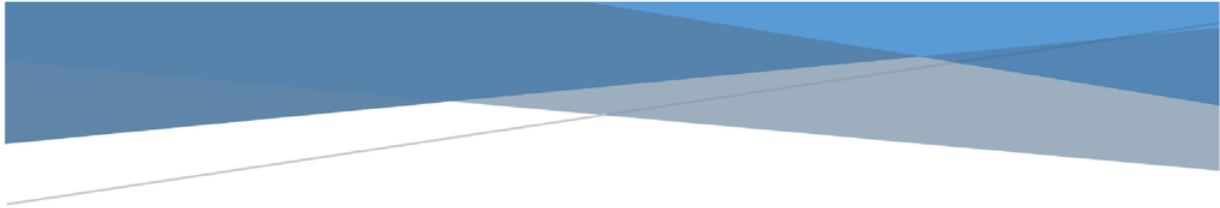


Assignment 2 PSDE5009

by Daniel Ince

FILE	14401_DANIEL_INCE_ASSIGNMENT_2_PSDE5009_158265_644629115.D OCX		
TIME SUBMITTED	13-MAY-2018 09:50PM (UTC+0100)	WORD COUNT	2845
SUBMISSION ID	89094786	CHARACTER COUNT	16160



ASSIGNMENT 2

PSDE5009

R. Davies

Electrical Principles, Methods and Simulation

13/05/2018

Daniel Ince

Contents

1. First Order System	2
2. DC Motor System Analysis	6
3. Types of Motor Speed Controller	12
Proportional speed control system.....	12
Proportional-integral speed control system.....	14
Proportional-integral-derivative speed control system.....	16
Controller comparisons.....	19
4. Conclusions and Recommendations.....	21

1. First Order System

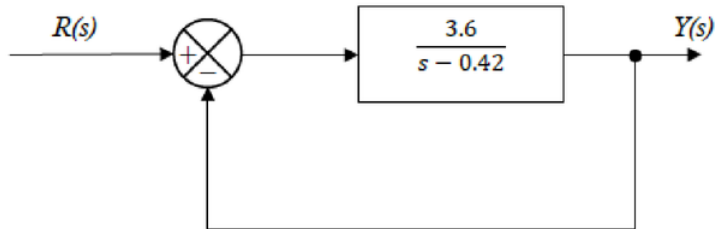


Figure 1: First order system

Figure 1 shows the first order system, currently without feedback, the transfer function puts the pole when $s - 0.42 = 0$ resulting as $s = +0.42$, this puts the zero in the right hand section of the plot. The feedback in this diagram takes the entire output back to the comparison, meaning 'K' is set to 1.

So, to work out the transfer function as a whole for this system, the value of K as 1 must be factored in.

$$\frac{G(s)}{1+G(s)H(s)} \text{ Where } G(s) = \frac{3.6}{s-0.42} \text{ and } H(s) = 1.$$

So $T(s) = \frac{\frac{3.6}{s-0.42}}{1 + \frac{3.6}{s-0.42} \times 1}$ Therefore, when we simplify by multiplying both top and bottom by $(s - 0.42)$ to bring the transfer function back into the first order format, we get:

$$T(s) = \frac{\frac{3.6}{(s-0.42)} \times (s-0.42)}{(s-0.42)(1 + \frac{3.6}{(s-0.42)})} \text{ To form: } T(s) = \frac{3.6}{s-0.42+3.6} \rightarrow \frac{3.6}{s+3.18}$$

This means that the pole (root of the denominator) becomes $s + 3.18 = 0$ so $s = -3.18$. This is in the left plane making the system stable, so it can be said that the feedback makes the pole shift from $+0.42$ to -3.18 , unstable to stable.

A root locus Matlab simulation can be run in order to determine if this is true. The transfer function can also be altered to work out the value for K when $s = 0$. For this to be correct, s needs to be 0 and the gain (K) must provide the minimum amount to make the system stable.

To do this, the transfer function $T(s) = \frac{\frac{3.6}{s-0.42}}{1 + \frac{3.6}{s-0.42} \times K}$ is used, and $H(s)$ is replaced with K. This makes:

$$T1(s) = \frac{\frac{3.6}{s-0.42}}{1 + \frac{3.6}{s-0.42} \times K} \text{ Simplifying this becomes: } T1(s) = \frac{3.6}{s-0.42+3.6K} \rightarrow \frac{3.6}{s+(3.6K-0.42)} \text{ To calculate K}$$

when $s = 0$, it must be set:

$$s + 3.6K - 0.42 = 0 \rightarrow 0 = 3.6K - 0.42 = 0$$

$3.6K = 0.42 \rightarrow K = 0.1167$ This value can also be plot in a root locus to set the pole to 0 and check the theory.

2 *Using a different standard format, you could have predicted the time-constant*

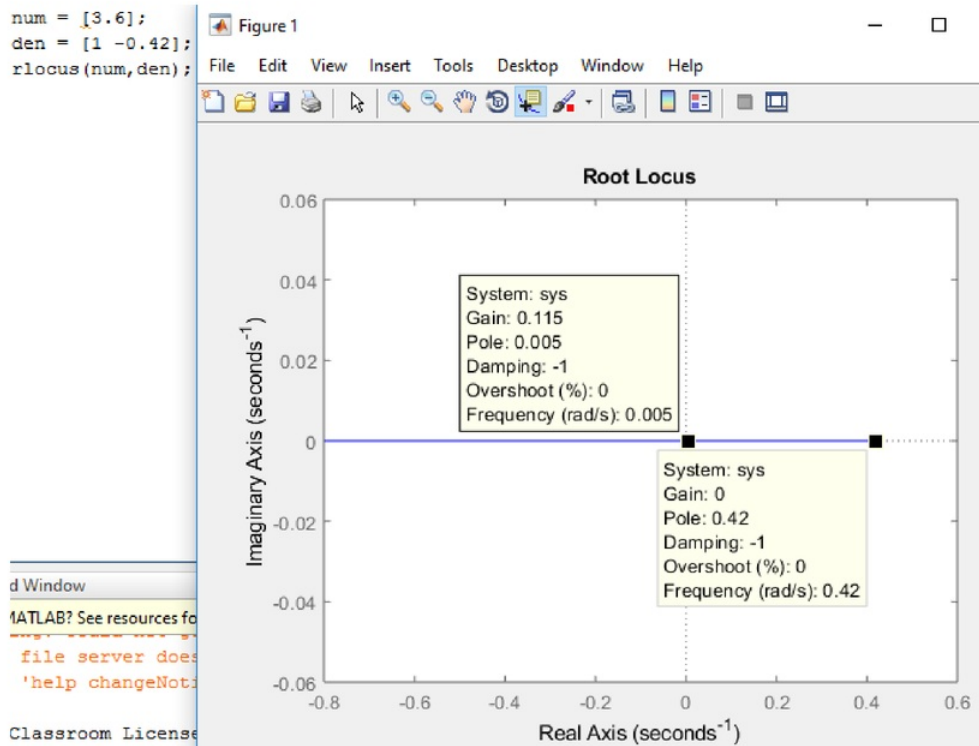


Figure 2: Open loop transfer function root locus

Figure 2 shows the root locus argand diagram for the transfer function: $\frac{3.6}{s-0.42}$. The pole can be seen at +0.42 suggesting instability whereas the zero tends towards minus infinity and cannot be seen in this scale. The gain when $s=0$ can be seen as approximately 0.115 when the pole is approximately 0, this suggests that the calculation is correct. A matlab simulation will now be run of the first order system with the closed loop feedback and the open loop feedback to prove the instability/stability.

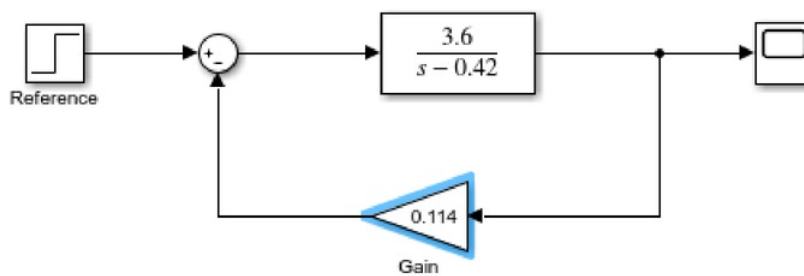


Figure 3: Open loop transfer function with gain of 0.114

Figure 3 shows the open loop transfer function with a gain value of K set to 0.114, this is intentionally below the minimum stability value of 0.1167 to test if the response is unstable.

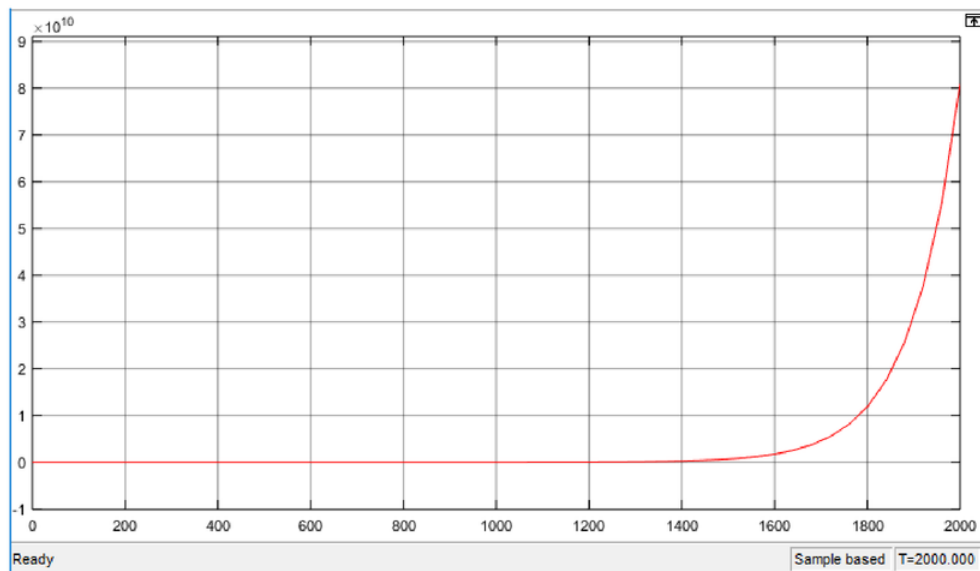


Figure 4: Unstable response

An unstable response can be seen in figure 4 as the graph tends towards infinity when the gain is not above the minimum required value. This means the system is underdamped and is not stable, when compared to figure 2, it proves this due to the pole being on the right side of the argand diagram.

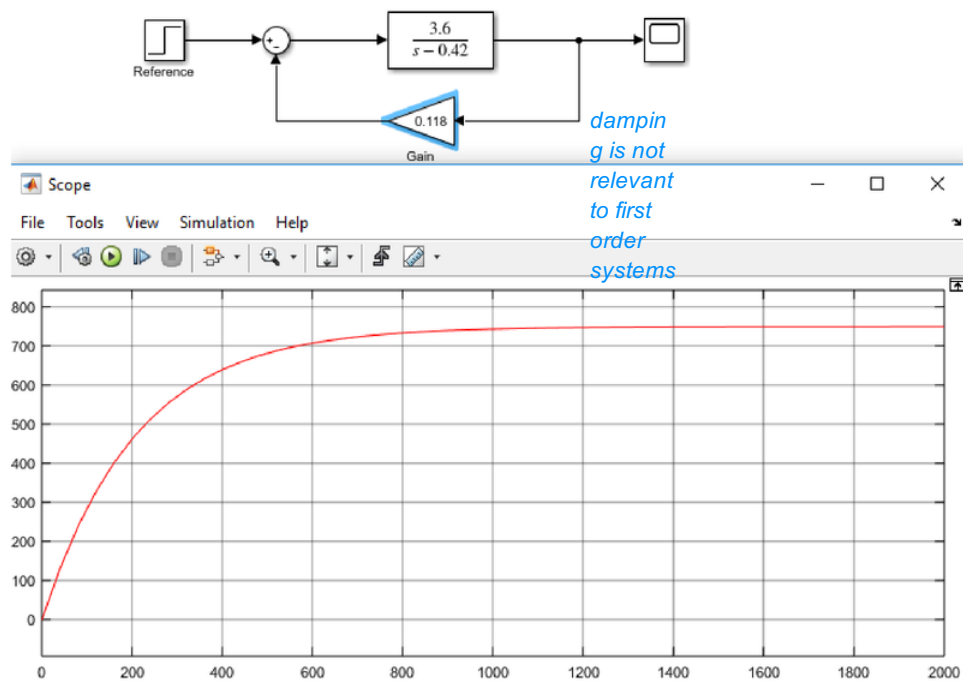


Figure 5: Stable response

Very good work. However, you could have found the location of the pole for $K = 0.118$ and $K = 1$ and the time constant and overall gains in each case since this is a first order system

When a gain of 0.118 is set, just above the stability threshold, the response is entirely different, it no longer tends towards infinity and displays a stable steady state period following the transient response. This system is pretty much critically damped, the original transfer function with a gain K value of 1 will be largely over damped but will still remain a stable system, shown below in figure 6.

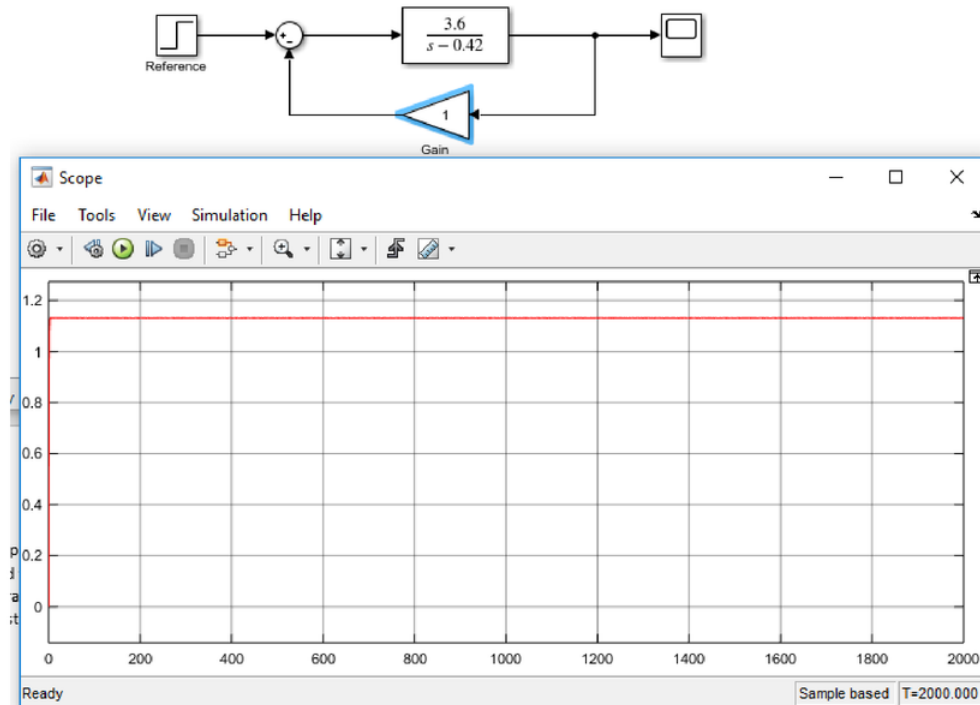


Figure 6: stable with a gain of 1

You could have measured the time constant and overall gain by zooming in on the transient part and comparison with theory

2. DC Motor System Analysis

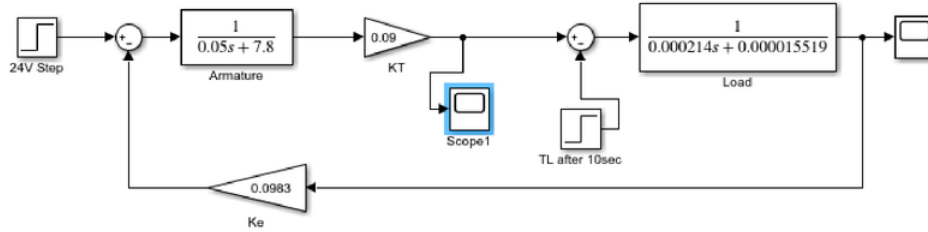


Figure 7: DC motor model

Above in figure 7 is the motor model developed through Simulink software. The details of this model are as follows:

Armature resistance[**R_a**] = 7.8 Ohms • Armature inductance[**L_a**] = 0.05H • Rotor inertia[**J**] = 0.000214 Kg m² • Friction coefficient[**b**] = 15.519 x 10⁻⁶ Nm/rad/s • Motor voltage constant[**K_e**] = 0.0983 V/rad/s • Motor torque constant[**K_t**] = 0.09 Nm/A • Maximum armature voltage = 40V

V_a = 24v

Below is the transfer function for the motor model.

$$G(s) = \frac{\frac{K_T}{L_a J}}{s^2 + s\left(\frac{b}{J} + \frac{R_a}{L_a}\right) + \frac{R_a b + K_T K_e}{L_a J}}$$

$$\frac{K_T}{L_a J} = \frac{0.09}{0.05 \times 0.000214} = 8411.215$$

$$\left(\frac{b}{J} + \frac{R_a}{L_a}\right) = \left(\frac{15.519 \times 10^{-6}}{0.000214}\right) + \left(\frac{7.8}{0.05}\right) = 0.0725 + 156 = 156.0725$$

$$\frac{R_a b + K_T K_e}{L_a J} = \frac{7.8 \times 15.519 \times 10^{-6} + 0.09 \times 0.0983}{0.05 \times 0.000214} = 838.135$$

Therefore:

$$G(s) = \frac{8411.215}{s^2 + 156.0725s + 838.135} \rightarrow \frac{K \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}$$

So, to work out the natural frequency, $\omega_n = \text{square root of } 838.135 = 28.95$.

The damping factor $\zeta = 2\zeta \omega_n = 156.07 \rightarrow \zeta = 2.7$ as this is greater than 1, it can be predicted that the response will be over damped.

The stall torque can be deduced from: $\frac{K_T V_a}{R_a} = \frac{0.09 \times 24}{7.8} = 0.277 \text{ Nm}$

The no load speed: $\frac{K_T V_a}{b R_a + K_t K_e} = \frac{0.09 \times 24}{(15.519 \times 10^{-6} \times 7.8) + (0.09 \times 0.0983)} = 240.86 \text{ rad/s}$

Loaded speed: $\frac{V_a - \frac{R_a I}{K_t}}{\frac{R_a b}{K_t} + K_e} \rightarrow \frac{24 - \frac{7.8 \times 0.025}{0.09}}{\frac{7.8 \times 15.519 \times 10^{-6}}{0.09} + 0.0983} = 219.11 \text{ rad/s}$

As gain is rad/s per volt, the gain can be worked out to be $240/24 = 10$ for an unloaded motor. This means that the speed of the unloaded motor should be approximately 240 rad/s when running. This will now be proved using simulink software.

Figure 8 shows the response of the motor speed when unloaded. The steady state speed value is 240.9 rad/s which ties into the 240.86 rad/s calculated value. The difference being down to the rounding of the software after two decimal places. The speed sharply rises upon starting the motor with no overshoot, but due to the transfer function zeta damping factor being well above 1, it is overdamped.

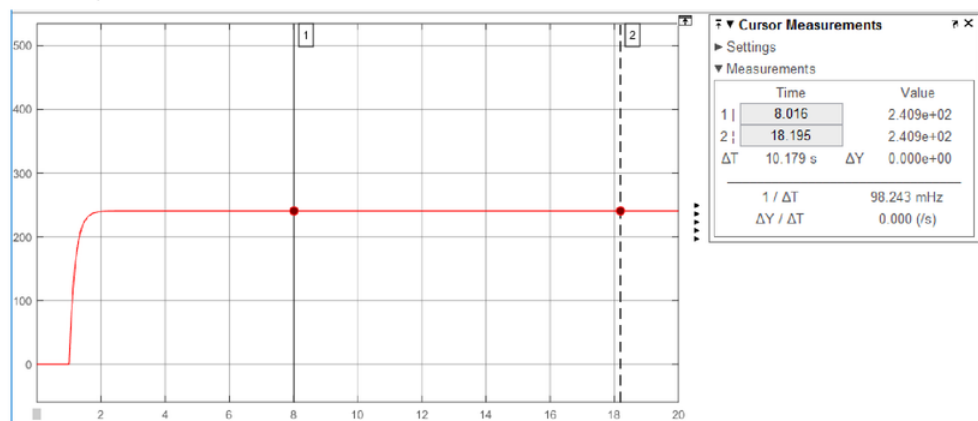


Figure 8: Over damped motor speed

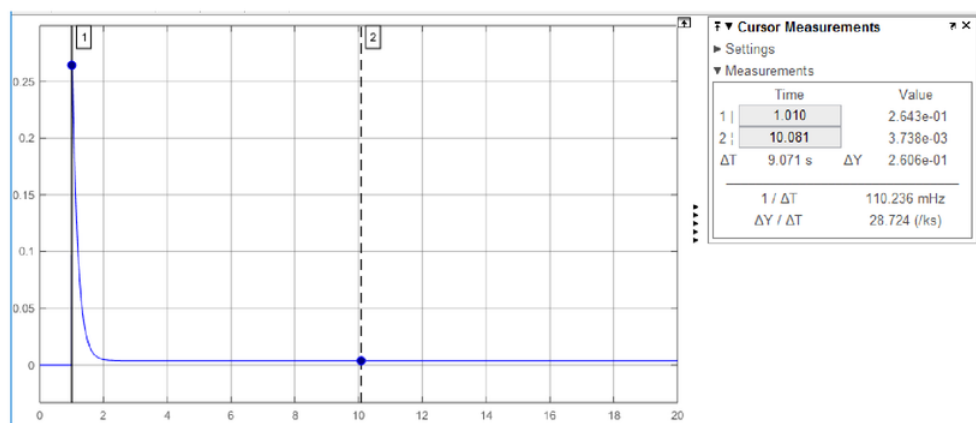


Figure 9: Overdamped motor torque

The large torque is in fact due to the lack of back EMF at start up, giving a very high current thus producing a large torque

Figure 9 displays the unloaded motor torque response. It rises abruptly on starting the motor due to the torque required to overcome the inertia and get the motor turning from standstill. The response begins rapidly dropping as the motor speed increases as less torque is needed due to the reduction of load. The settled value for the torque at maximum motor speed is 0.00373Nm, requiring almost no torque at all to keep the motor turning. The peak of maximum torque can be seen as 0.2643nm at 1 second in. This value is less than the calculated 0.277Nm stall torque value and evident from the figure, the motor does not stall when under this torque.

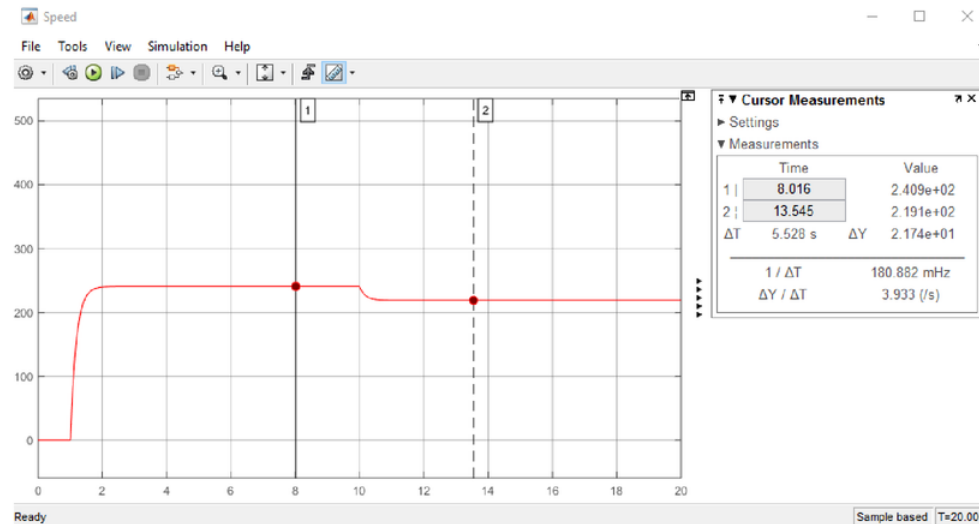


Figure 10: Motor load disturbance at 10 seconds speed

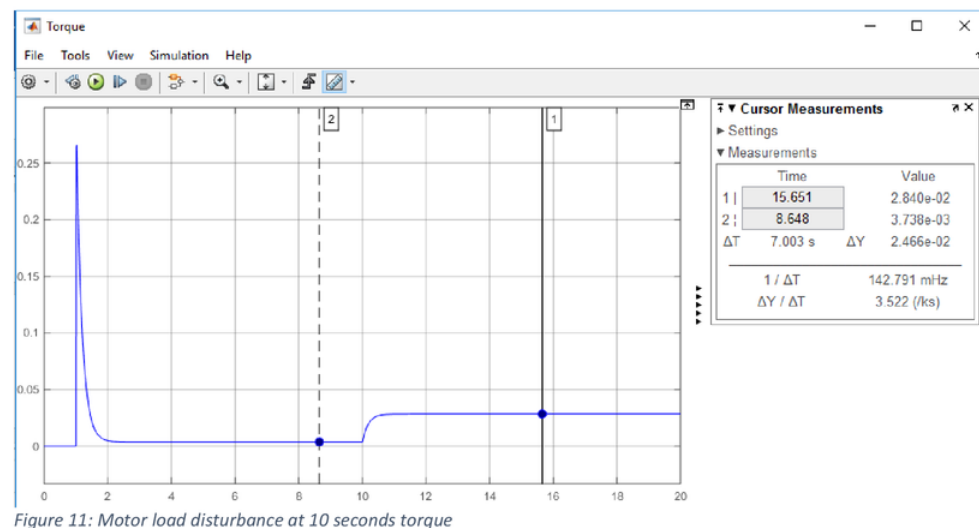


Figure 11: Motor load disturbance at 10 seconds torque

*should be at 11 seconds since it starts at 1
(10+1 = 11)*

When adding the torque load disturbance of 0.025Nm at 10 seconds the response changes to suit as seen in figures 10 and 11. Figure 10 shows the motor speed, up until 10 seconds in, the response is exactly the same as the unloaded motor. However at the point of disturbance, the speed drops off and then steadies out at a lower speed of 219.1rad/s. When compared to the calculated value these are identical, proving the theory. The torque shown in figure 11 can be seen to rise when the load changes, this is down to more torque being needed to drive the motor around and overcome the load. Overall the torque increases and the motor speed drops.

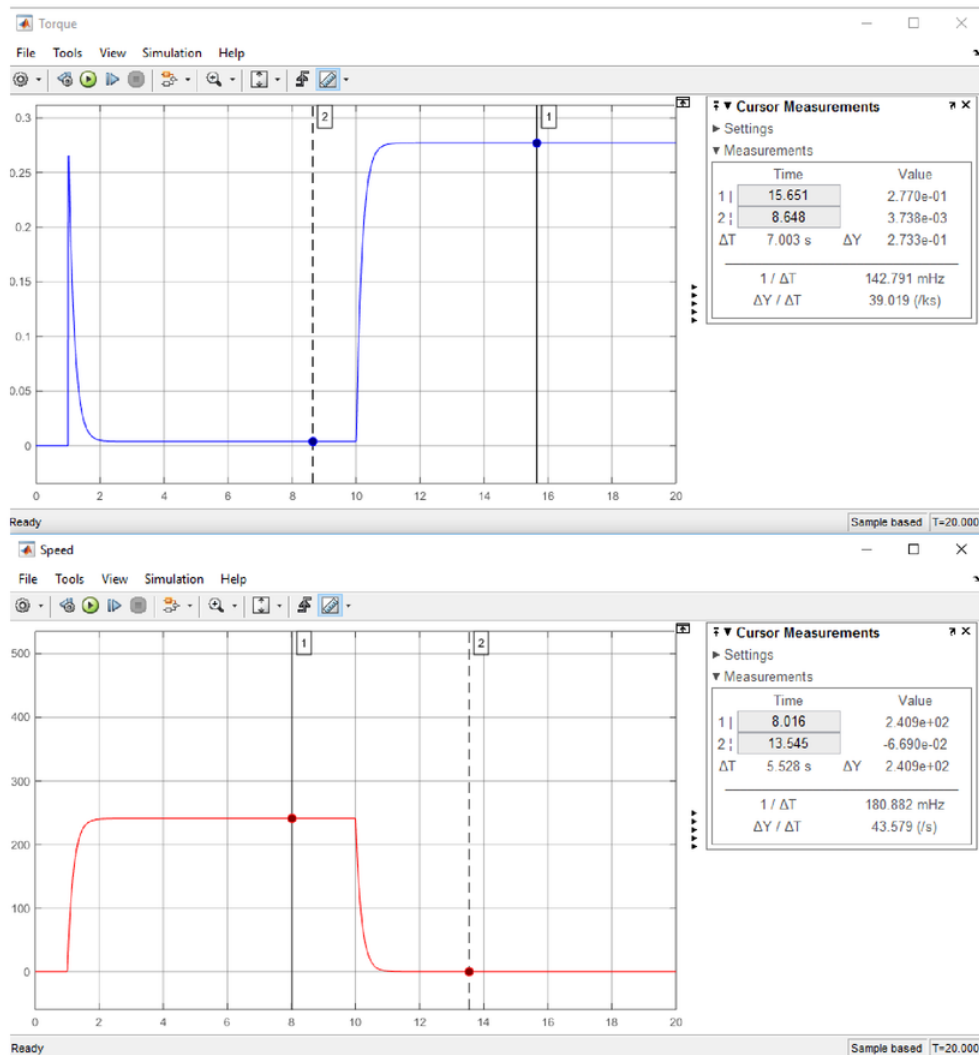


Figure 12: Stall torque proving

Another simulation was run with a disturbance, this time the load was set to the calculated stall torque value of 0.277Nm to determine if this works in practice. Figure 12 shows the disturbance added at ten seconds. It can be seen that the speed drops off almost instantly as the motor winds down to the point of stalling, and the torque increases, remaining there at a value of 0.277Nm due to the load remaining but motor still attempting to turn. This proves that the calculations are correct

*The motor has actually stopped since the
speed is virtually zero!*

for stall torque and that due to the equation, a higher voltage and lower armature resistance will provide a higher stall torque, meaning the motor can deal with more load before stalling out.

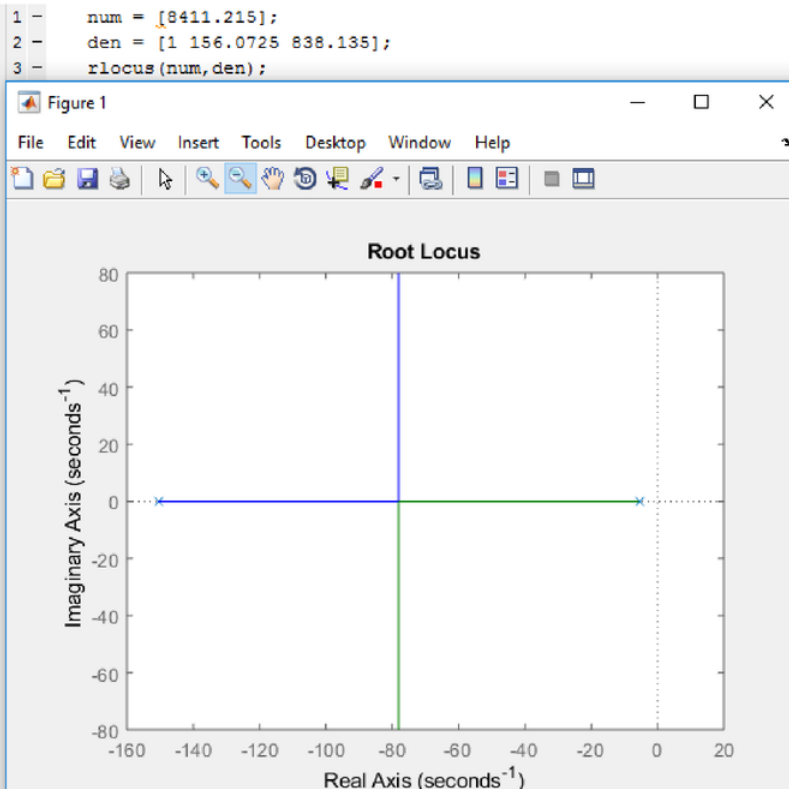


Figure 13: Argand diagram

To further analyse the system stability, the poles of the transfer function can be determined. Poles, being the root of the denominator can be found by putting the bottom line of the transfer function equal to zero.

$$G(s) = \frac{8411.215}{s^2 + 156.0725s + 838.135} \rightarrow s^2 + 156.0725s + 838.135 = 0$$

This leaves values which can be substituted into a quadratic equation to provide the two poles for this second order system.

$$\text{poles} = \frac{-156.0725 \pm \sqrt{156.0725^2 - 4 \times 1 \times 838.135}}{2 \times 1} = (-5.57[\text{pos}]) \text{ and } (-150.5[\text{neg}])$$

When inputting the transfer function into matlab and analysing the simulated argand diagram in figure 13, no zeroes are visible due to them tending towards infinity, this is because of the lack of the 's' value in the numerator. The poles are at -151 and -5.81 and can be seen in the figure, these are extremely close to the calculated values through using the quadratic. -5.57 is the dominant pole, it is the closest to 0 yet has the most effect due to it representing the mechanical time constant. The electrical time constant pole -151 is the recessive pole, having the least effect and is the furthest left on the diagram. As the poles are in the left half of the diagram, the system is said to be stable, this is proved by the fact that none of the response graphs are unstable. However if the dominant pole was to shift to the right past 0 it would swing the system into an unstable one.

Value	Calculated	Simulated
Stall torque	0.277Nm	0.277Nm
No load speed	240.86rad/s	240.9rad/s
Loaded speed	219.11rad/s	219.1rad/s

good summary

Above is a table containing all calculated and simulated values, overall all values tie in to one another with only slight differences due to the rounding of the graph on the simulation. This proves that this transfer function is correct and the theory used to produce these values is accurate.

3. Types of Motor Speed Controller

Proportional speed control system

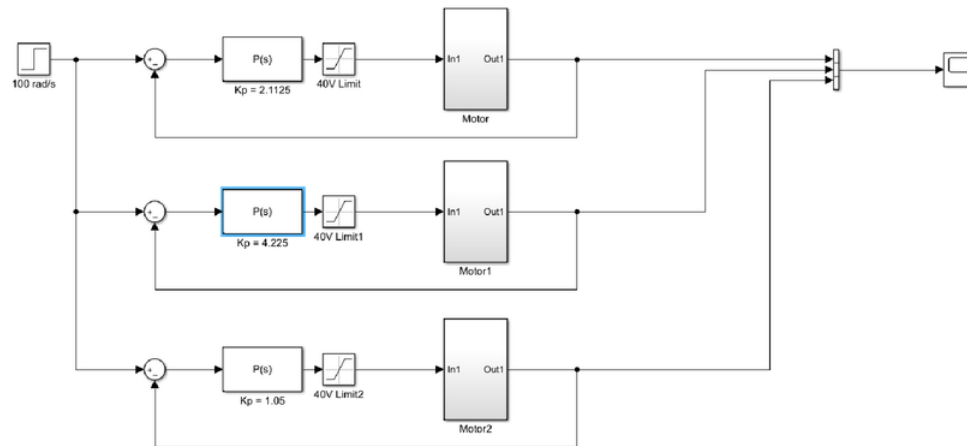


Figure 14: Proportional control simulated model

The motor model from task 2 was compressed into a subsystem and used as a block within this proportional controller. A step input of 100rad/s was used and the 0.025Nm disturbance at 10 seconds remained in. A PID block, set to only P was added along with a voltage limiter of 40V. Three of these systems were made and the results combined to be analysed in a three input scope with varying values of proportional gain. The resulting system model can be seen in figure 14. When autotuning the system, the optimum value for K_p was found to be 2.1125, this was used for the first of the three systems. Values above and below this were chosen for the other two systems, 4.225 and 1.05, roughly double and half the optimum value, the resulting responses were then analysed.

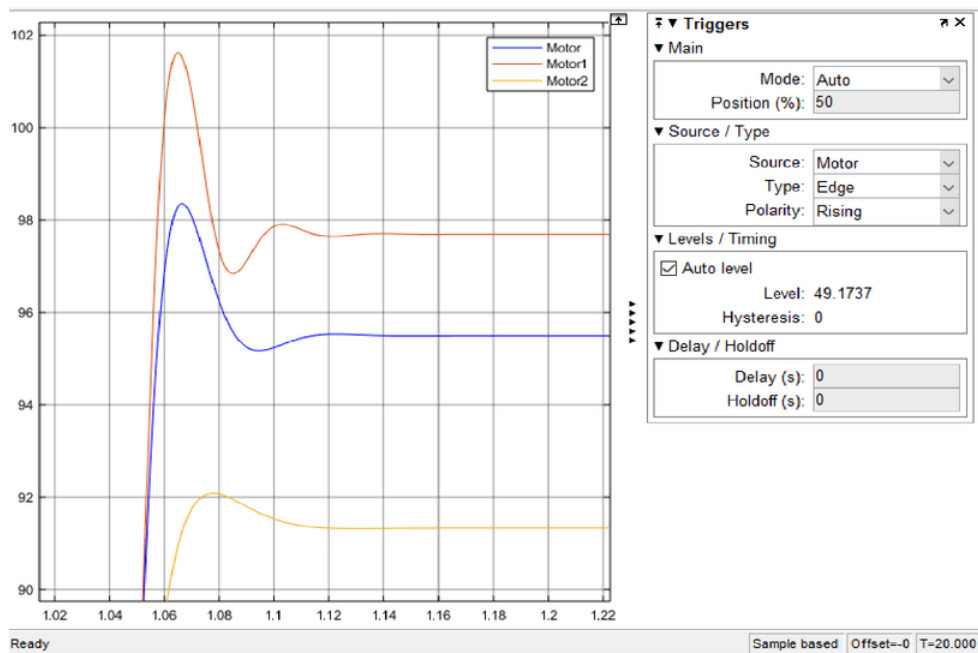


Figure 15: P response

Figure 15 shows the initial response by all three of the motors within the proportional control system.

a good summary

Kp	Rise time	Overshoot	Steady state error	Settling time
2.115	0.064	98.4-95.4=3	4.6	1.15
4.225	0.063	101.6-97.6=4	2.4	1.15
1.05	0.075	92.2-91.4=0.8	8.8	1.15

The rise time, overshoot, steady state error and settling time for all three motors was measured and the results compiled in the above table. From comparing the figure and table, it is apparent that the higher the value for Kp the larger the overshoot. While an increase of overshoot is apparent, a decrease in steady state error is also seen, the trend 'motor' is the auto tuned value and can be seen as a balance between the two. All three have the same settling time regardless of the proportional factor.

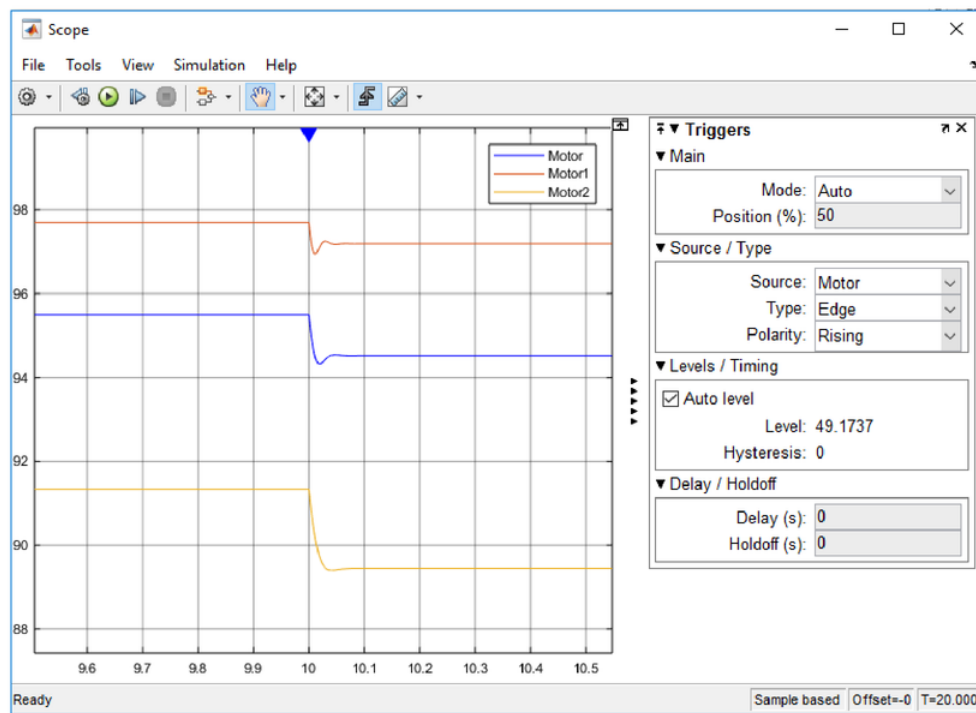


Figure 16: P control disturbance

Through also analysing the disturbance response, seen in figure 16, the value of Kp determines how the system responds to the load. The higher than optimum Kp value of 4.225 can be seen to react fast with more overshoot, but ultimately ends up with the least steady state error. The lower Kp value of 1.05 (motor 2 trend) responds slower and deviates even further from the set point of 100 when the disturbance is added than the other two do. Once again all three settle out at the same time.

Proportional-integral speed control system

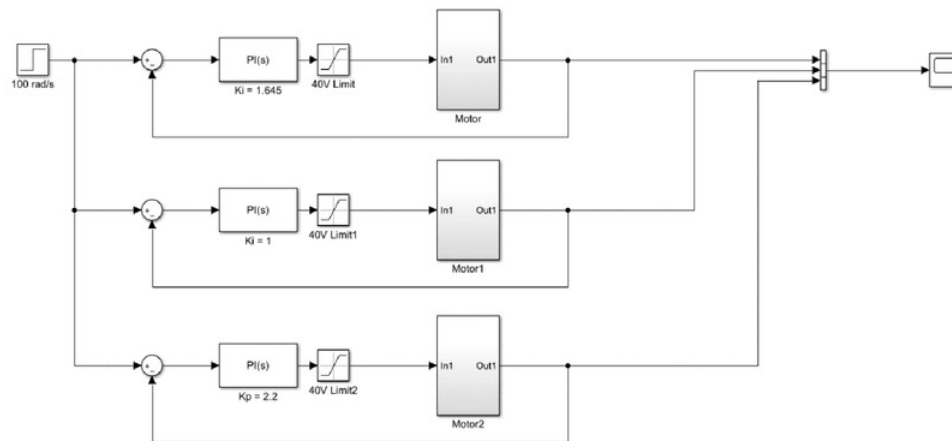


Figure 17: PI control system model

This type of control system has the proportional and integral factors on the PID block. The top motor is autotuned for PI, the following two have the exact same K_p value from the autotuned top motor but have different values for K_i . This is to isolate the integral factor as the changing variable and analyse only its effect on the response.

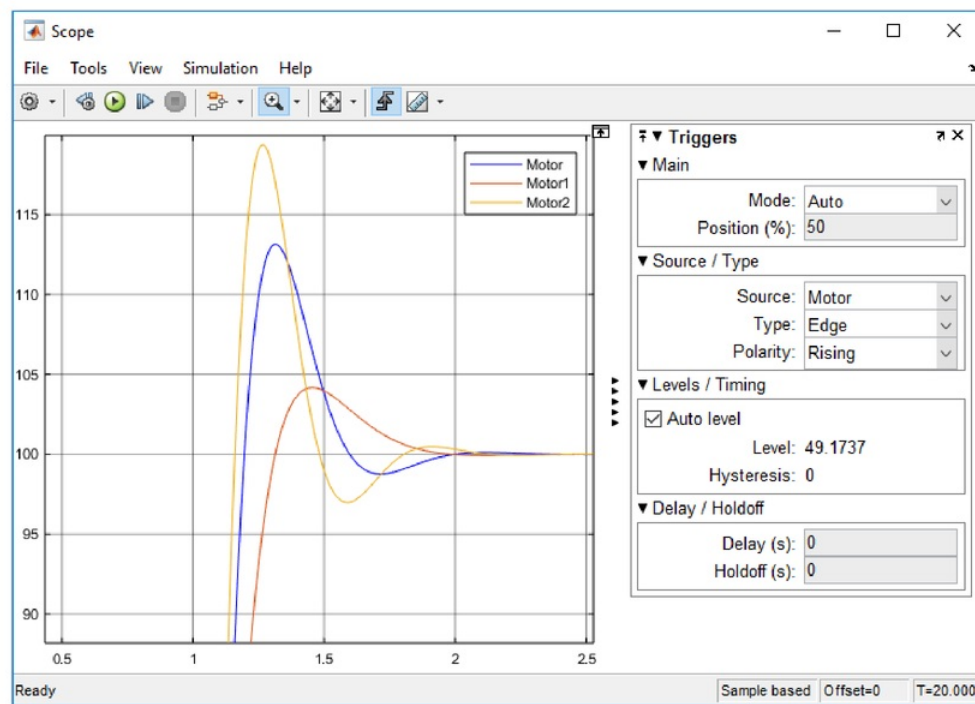


Figure 18: PI response

Ki	Rise time	Overshoot	Steady state error	Settling time
1.645	0.3	12.5	0	2.24
1	0.4	4	0	2.24
2.2	0.25	19.3	0	2.24

The table above shows the results from measuring the PI response graph shown in figure 18. The most notable change is that the steady state error for all three motors is 0 and they have achieved the desired set point of 100. The rise time is different for all three, with the largest value for Ki having the shortest rise time but also the most overshoot. The auto tuned response balances the overshoot and rise time to produce the optimum response for proportional-integral control.

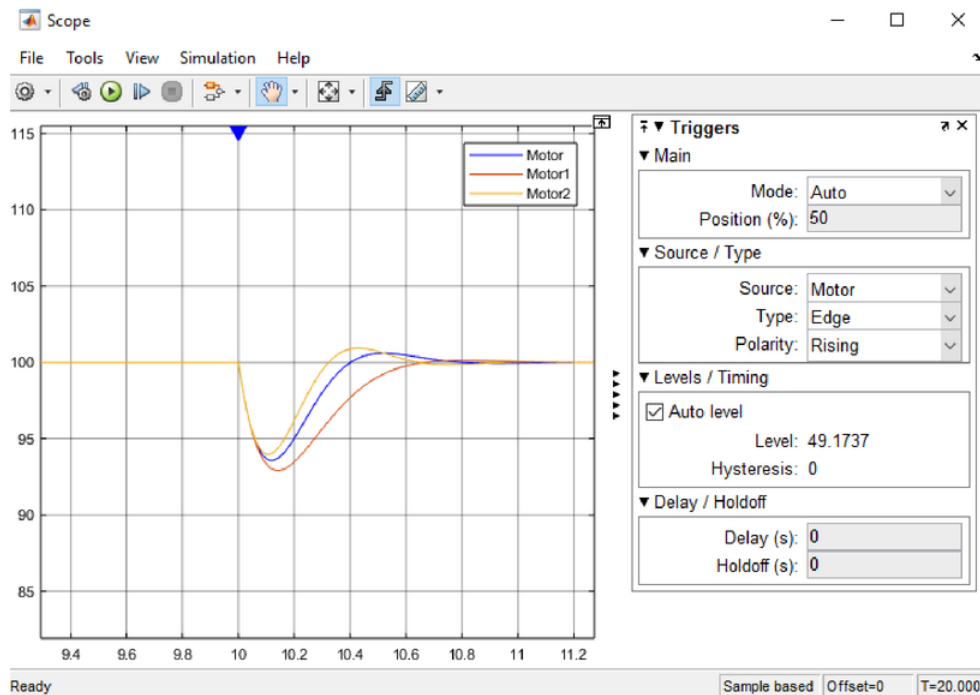


Figure 19: PI disturbance response

Figure 19 shows the 0.025Nm disturbance after 10 seconds, the integral factor can be seen to once again remove the steady state and following the disturbance, the set point of 100rad/s is met. The yellow 'motor 2' trend responds fastest but has the largest overshoot, this is explained by the highest value for Ki.

Proportional-integral-derivative speed control system

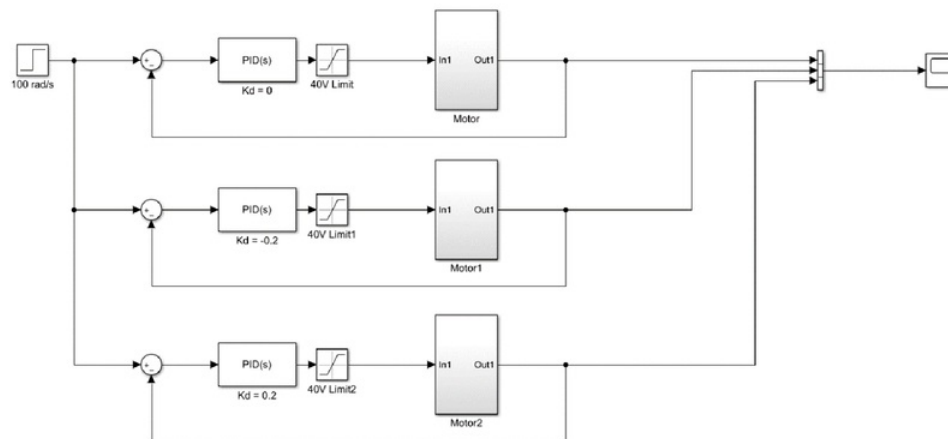


Figure 20: PID controller

The Simulink model for the PID controller is shown in figure 20, the block was adjusted to enable the proportional, integral and derivative factors and then auto tuned. The result of the auto tune is shown below in figure 21. The block was updated for the top system and the values for K_p and K_i were input as constants for the other two systems. The derivative K_d factor was altered either side of the optimum response. In this case the auto tuned value for K_d is 0, so the other K_d values were 0.2 and -0.2.

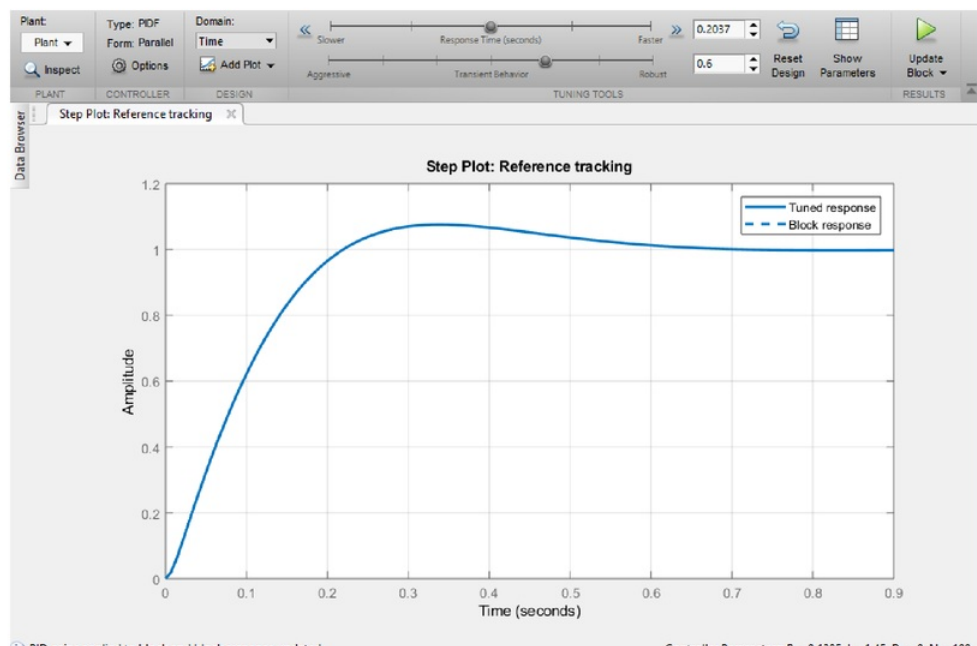


Figure 21: Autotuned response

Although Simulink suggests the optimum value of K_d is zero, it would have been better to have introduced a small value for comparison as put a negative value in for K_d produces positive feedback, no wonder it

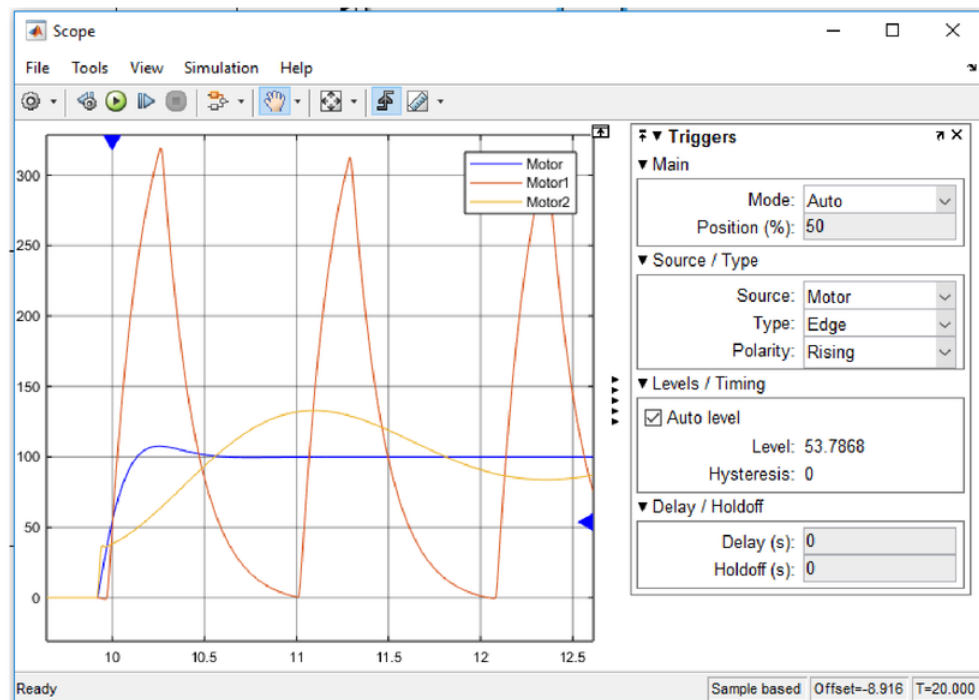


Figure 22: PID response

The autotuned PID response can be seen in the blue 'motor' trend, and is the optimum response for this system. The negative -0.2 Kd value can be seen in the orange 'motor1' trend, this trend does not settle and endlessly cycles being unstable. There is no steady state, and it has a massive overshoot. This is down to the derivative factor altering the settle time, the tuned response has the fastest settle time, rise time and has the least overshoot, combining all three of the factors.

Kd	Rise time	Overshoot	Steady state error	Settling time
0	0.25	8	0	0.5
-0.2	0.25	225	0	n/a
0.2	1	32	0	n/a

As the tuned response for this system has a value of 0 for the derivative, there is no action from the derivative factor and so it can be determined that it is not needed and that the proportional-integral action alone is suitable for this particular system. Adjusting the derivative factor even slightly proves huge changes to the system and provokes an unstable response.

Figure 23 shows how the system reacts to a disturbance, the 'motor1' trend is already very unstable and so there is almost no discernable difference in the trend. However, the tuned response returns to the set point much faster than the 'motor2' trend, showing that no derivative factor is better in this case and no reduction in overshoot is needed.

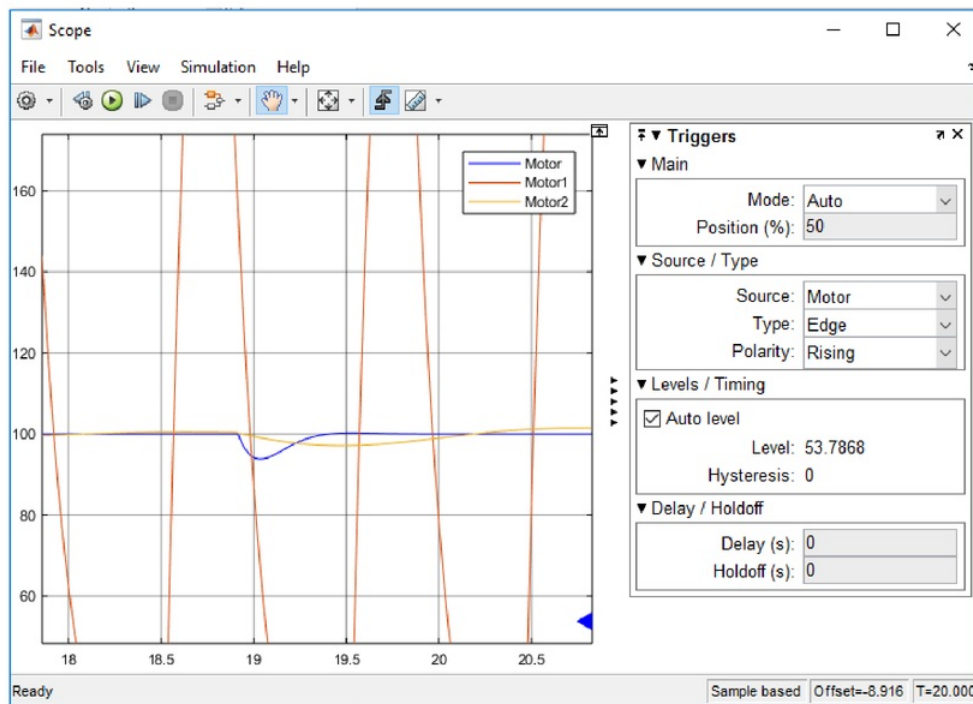


Figure 23: PID disturbance response

Controller comparisons

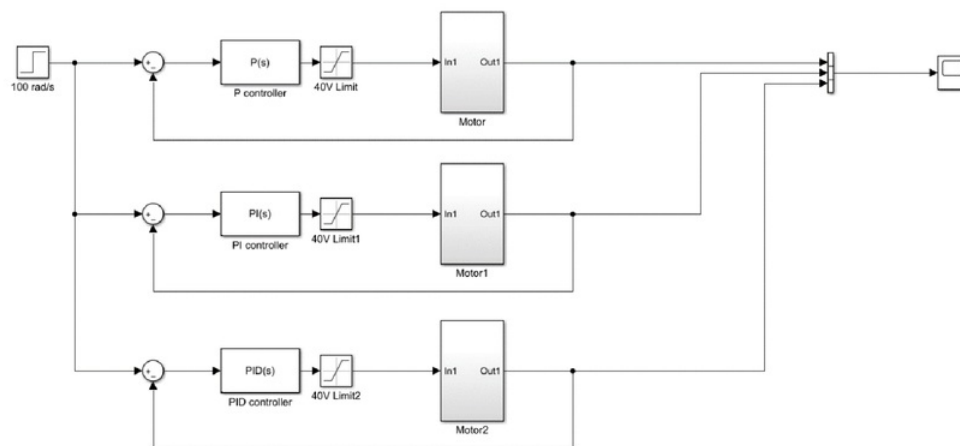


Figure 24: P, PI and PID comparison

This comparison involved putting together the autotuned control systems for P, PI and PID control types shown in figure 24 and comparing their responses on the scope.

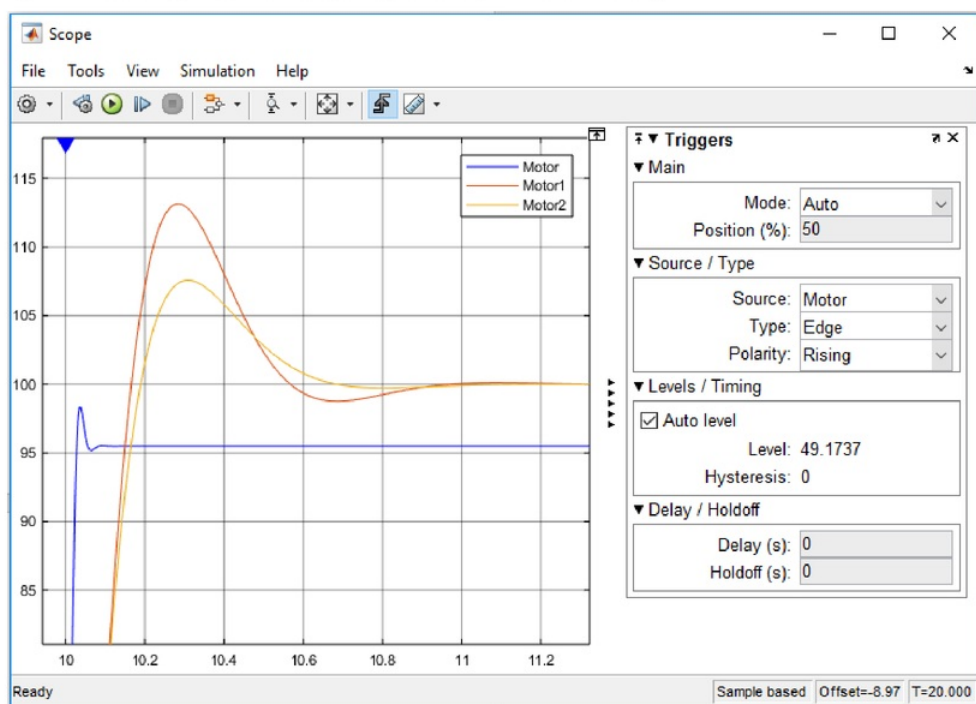


Figure 25: Comparison response

In figure 25, the effects of each controller can be seen. The first motor only has the proportional factor and so despite the fastest response, leaves a large amount of steady state error and along with some overshoot. The error is the main problem here as this would not be suitable for an industry process control system due to business and safety requirements.

The PI system shown as the orange 'motor1' trend removes this steady state error, proving that the integral factor removes this. However, there is still a large amount of overshoot and it also has the slowest time to settle.

The PID system shown as the yellow 'motor2' trend responds the best, although the tuned values for this produced a derivative value of 0. Meaning that although the PID block is being used, there is no derivative action required. This trend also has no steady state error and a notable amount of overshoot, however it is slightly less than that of the PI controller. Should the derivative action be increased, the overshoot would decrease but the settle time would increase and the system could become unstable if over done. Therefore the optimal solution in this case is to only have a PI action, this is different for all control systems and is not always the case.

A good analysis carried out with correct conclusions

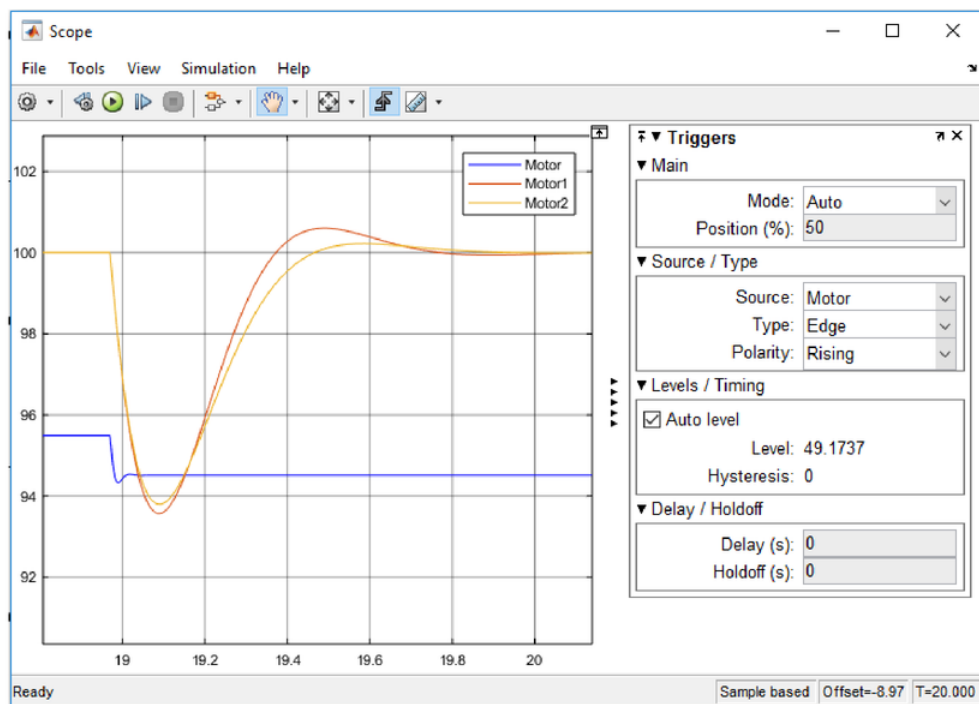
4. Conclusions and Recommendations

Throughout these tasks, an understanding of the differences between first and second order control systems has been developed. Along with how the transfer functions directly affect the responses of the systems. An unstable open loop system can be deduced to be unstable from the transfer function, however with the addition of some feedback, an overall transfer function can be produced and the poles move to the left, showing the system to have become stable. This system when simulated proves the theory and is useful for real-life process control situations.

Calculating predicted values and then simulating a model has provided an insight into further aspects of control such as turning the model into a subsystem and then applying PID control. Investigation into this control has helped understanding of the fine tuning of each individual part of the proportional, integral and derivative action within this system.

It was a surprise that some systems are better off left with only PI control as oppose to using all three control actions. Discovering this through simulation has helped with applying the knowledge gained into a practical aspect and having the freedom to change different control actions with the addition of a disturbance to the system and observe how it responds.

*Good conclusions but no
recommendations ?*



Assignment 2 PSDE5009

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

73/100

PAGE 1

PAGE 2

PAGE 3

Text Comment. Using a different standard format, you could have predicted the time-constant

PAGE 4

PAGE 5

Text Comment. damping is not relevant to first order systems

Text Comment. Very good work. However, you could have found the location of the pole for $K = 0.118$ and $K = 1$ and the time constant and overall gains in each case since this is a first order system

PAGE 6

Text Comment. You could have measured the time constant and overall gain by zooming in on the transient part and comparison with theory

PAGE 7

PAGE 8

PAGE 9

Text Comment. The large torque is in fact due to the lack of back EMF at start up, giving a very high current thus producing a large torque

PAGE 10

Text Comment. should be at 11 seconds since it starts at 1 ($10+1 = 11$)

Text Comment. The motor has actually stopped since the speed it virtually zero!

Text Comment. This is a good deduction and shows you have a good grasp of this subject

Text Comment. good summary

Text Comment. a good summary

Text Comment. Although Simulink suggests the optimum value of K_d is zero, it would have been better to have introduced a small value for comparison as put a negative value in for K_d produces positive feedback, no wonder it is oscillating!

Text Comment. A good analysis carried out with correct conclusions

Text Comment. What are the differences?

Text Comment. Good conclusions but no recommendations ?

RESEARCH-INFORMED LITERATURE

 n/a

KNOWLEDGE AND UNDERSTANDING

 Section 1: Explanation of first order system – Excellently explained (8/10) Section 2: Explanation of DC motor model – Good explanation given. However, you could have spent more time explaining each part of the model. (6/10) Section 3: Explanation of types of speed controllers – Excellent (8/10) **22**

ANALYSIS

 Section 1: Testing of first order system – Very good analysis. However, you could have considered the time constants and overall gain. Damping should not be part of this analysis as it does not apply to a first order system. (7/10) Section 2: Testing of DC motor model – Excellent analysis. (9/10) Section 3: Testing of speed controllers – Generally, very good analysis though setting K_d to a negative value was not a good idea. You could have showed the auto tune responses that would explain where the results came from (15/20) Section 4: Conclusions and recommendations – Brief conclusions given but no recommendations. (5/10) **36**

PRACTICAL APPLICATION AND DEP.



Sections 1, 2 and 3: Use of Simulink and presentation quality of results - Excellent (8/10)

8

SKILLS FOR PROFESSIONAL PRAC.



All sections – A well structured and presented report. Figures could have been listed at the front of the report. (7/10)

7