

Control System Design

Jacob Smith

Instrumentation & Control

ACDA5011

**Wales Institute for Science and Art
University of Wales Trinity Saint David**



Abstract

Modern control systems typically use PID controllers and mathematical process known as transfer functions. A PID controller uses a feedback loop to help control the system it is incorporated within. A Transfer function uses mathematical formulas to represent the relationship an input and output signal has on a system. This document will explore a basic PID controller to achieve a goal and produce a transfer function to represent this process whilst trying to achieve a stable system that can bring a system under control within 2 seconds and have an overshoot of less than 5%.

Introduction

For the purpose of this report the system that has been chosen to be simulated the closed loop angular control system of the Close In Weapon System (CIWS) used by the Royal Navy, which goes by the name of Phalanx. For the simulation that will be carried out in this report the controller used has been simplified into a basic transfer function block diagram, as shown below in figure 1, with table 1 explaining each component of the diagram.

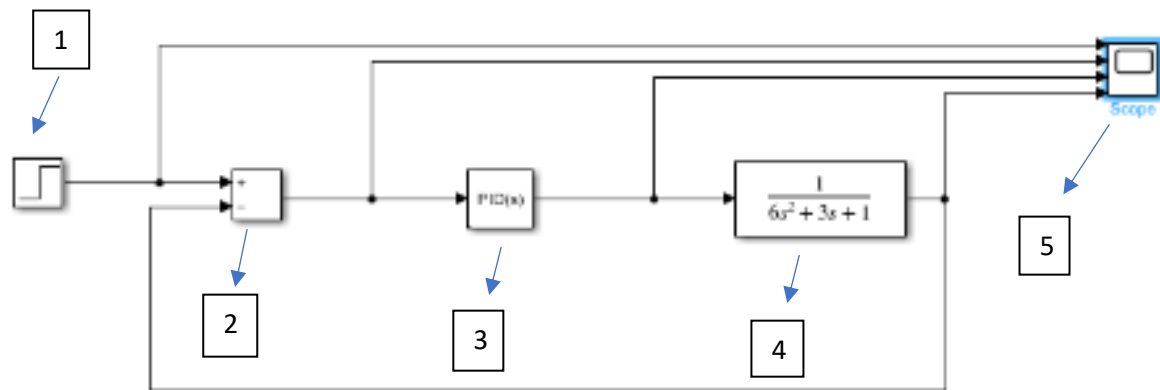


Figure 1:: Transfer function block diagram

Number	Description
1	Signal input
2	Initial signal and feedback signal amalgamation
3	PID controller
4	Transfer function second order equation also known as the Plant
5	Oscilloscope

Table 1: Figure 1 description

This is a basic PID control system where P I D stands for Proportional-Integral-Derivative, this type of controller is widely used due to its simplicity and effectiveness. It is a controller that uses a feedback loop to control the output signal. It constantly produces a steady state error value $e(t)$ against the initial set point $r(t)$ and the measured output $y(t)$ (as shown below in figure 2). It then provides correctional outputs using Proportional, Integral, and Derivative values, by doing this it allows the controller to apply greater control and accuracy over the system it is controlling [1].

The Proportional part of the controller applies proportionate output signal in accordance with the error value. If the error value is large the proportional output will be large in response, in an effort to get the error value to as small as a number as possible, which in turns means the output is as close to the set point as possible. But with a higher output there is a side effect of the possibility of inducing oscillation and instability into the system. As the error value approaches zero the applied correction also reaches zero meaning the final signal will never reach the desired output

The Integral part of the controller uses the error value after the initial proportional signal by adding a control effect in an attempt to reduce the error value to zero. This works by driving the signal to as close to zero as possible, but this has the adverse effect of inducing an overshoot, where the final output can exceed the desired position. This can induce an oscillating effect to a system.

The Derivative part of the controller uses the proportionate and integration output to predict the future signal outputs of the system. By doing this is can apply correction factors to the signal as it approaches the target value. This reduces overshoot and helps prevent oscillations. [1]

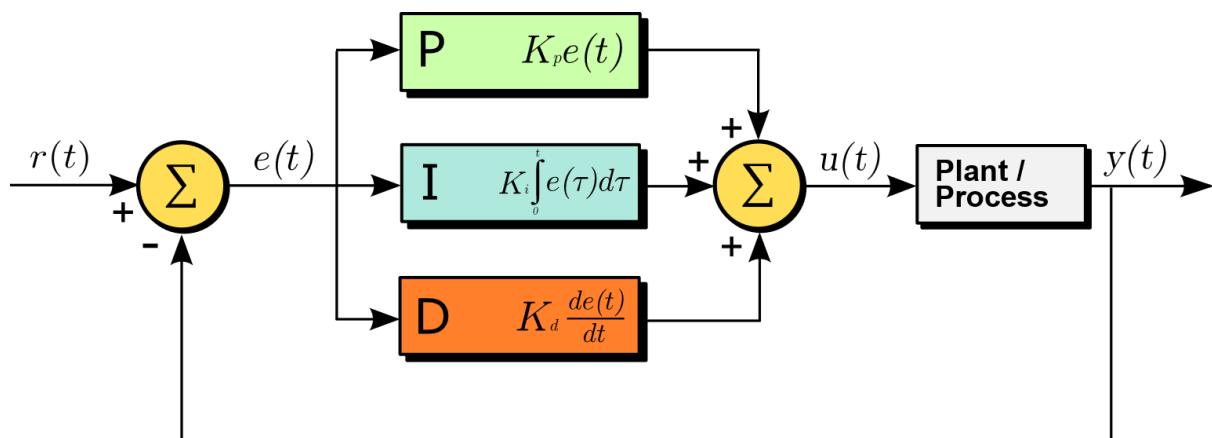


Figure 2: PID controller operation [2]

PID controllers are widely used in everyday systems we interact with daily, from a temperature controller for an oven to a vehicles cruise control. Any system that requires a desired variable output will utilise a PID controller as the simplest and most cost-effective way of controlling the system.

Simulation setups

The simulation which forms the basis of this report was carried out on a computer software known as MATLAB Simulink, which enable users to create virtual control systems to run simulations and graphically represent the results, with numerous analytical tools [3].

As described in the introduction a simple block diagram of the angular controller has been constructed in MATLAB Simulink shown below in figure 3.

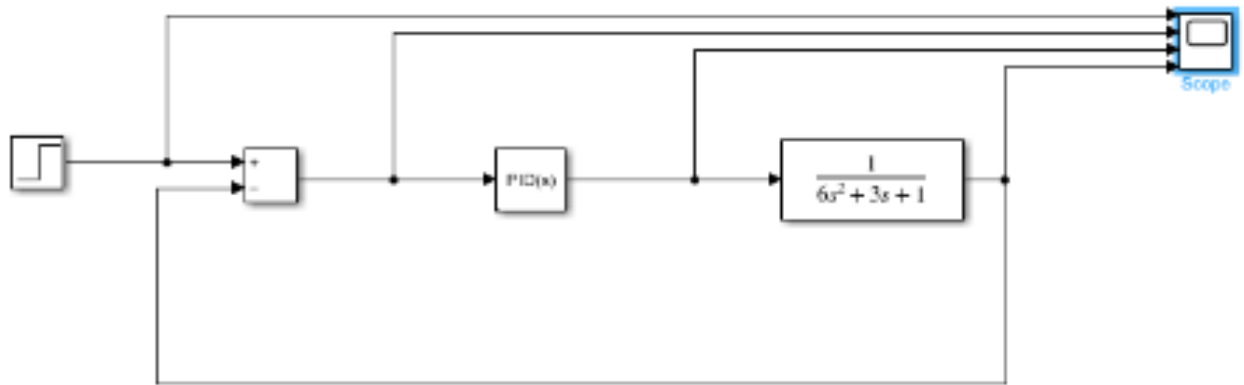


Figure 3: Simulation block diagram

All the control functions described in the introduction are the basic driving principles in a transfer function and in this case will be known as $G(s)$ this can be presented by the equation below, which is found in the above block diagram shown in figure 3. This equation also represents the Plant/Process shown in figure 2.

$$G(s) = \frac{1}{as^2 + bs + c}$$

For the purpose of the simulation and report, the constants a , b , and c are to be represented by the last 3 digits of the authors student number and in this case, they correspond to $a = 6$, $b=3$, and $c=1$ and the equation can be rewritten to what is shown below. This will be the main transfer function going forward for the simulation.

$$G(s) = \frac{1}{6s^2 + 3s + 1}$$

After the constant of the transfer function have been input into the system, the simulation was run where the software simulated an instruction for the CIWS to rotate to a desired position. Thanks to the oscilloscope attached into the system at various points the results can be shown in a graphic form showing the time between the initial set point, the error signal generation, the controller output and the actual output of the system. The results also show any overshoot and settling time associated with the system.

After the initial simulation the system then had to be tuned to meet to criteria as set out in the assignment brief. The system needed to be able to rotate the CIWS within 2 seconds of the initial input and with an overshoot of less than 5% of the input figure.

Further analysis was carried out by using a more complex transfer function equation to remove the PID controller from the system to prove mathematically the system was working as desired. These equations can be found in Appendix 1. The equational output from the system was then compared to the original system output.

Results

The results of the initial simulation can be seen below in figure 4 with the Y axis representing the response magnitude and the X axis representing time. For all further graphs like this the axis labels will remain the same. The 4 graphs represent the following:

1. Set point.
2. Error signal
3. Controller output
4. Actual system output

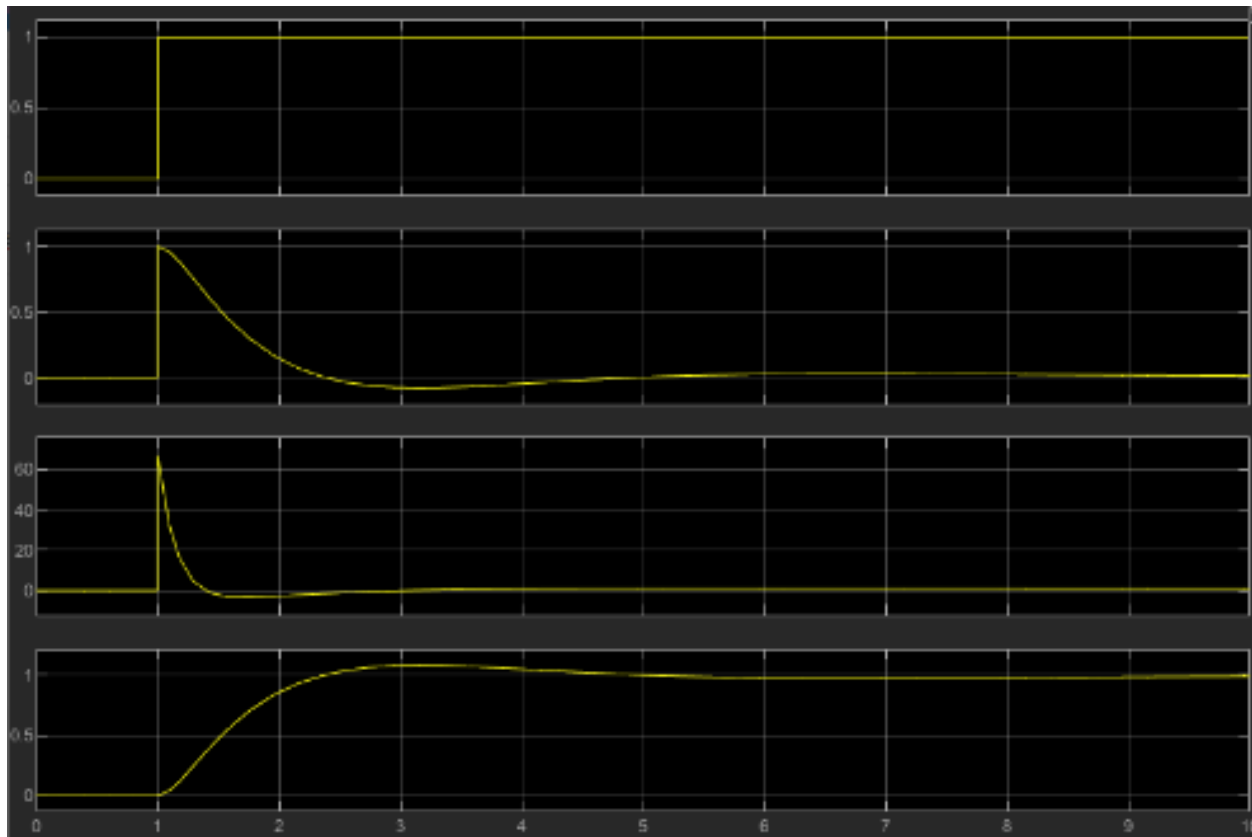


Figure 4: initial simulation results

As can be seen in graphs above the response time is slow and there is a period of several seconds where the system is oscillating before settling at the correct point.

In figures 5 and 6 below we can see the initial output of the automated tuning provided by the simulation system. The X axis is again represented by time and the Y axis is the amplitude of the system response to the new set point. The dotted line is the pre tuning results and the solid line represents the tuned output signal. These outputs are simulating the tuned output of graph 4 seen in figure 4 above.

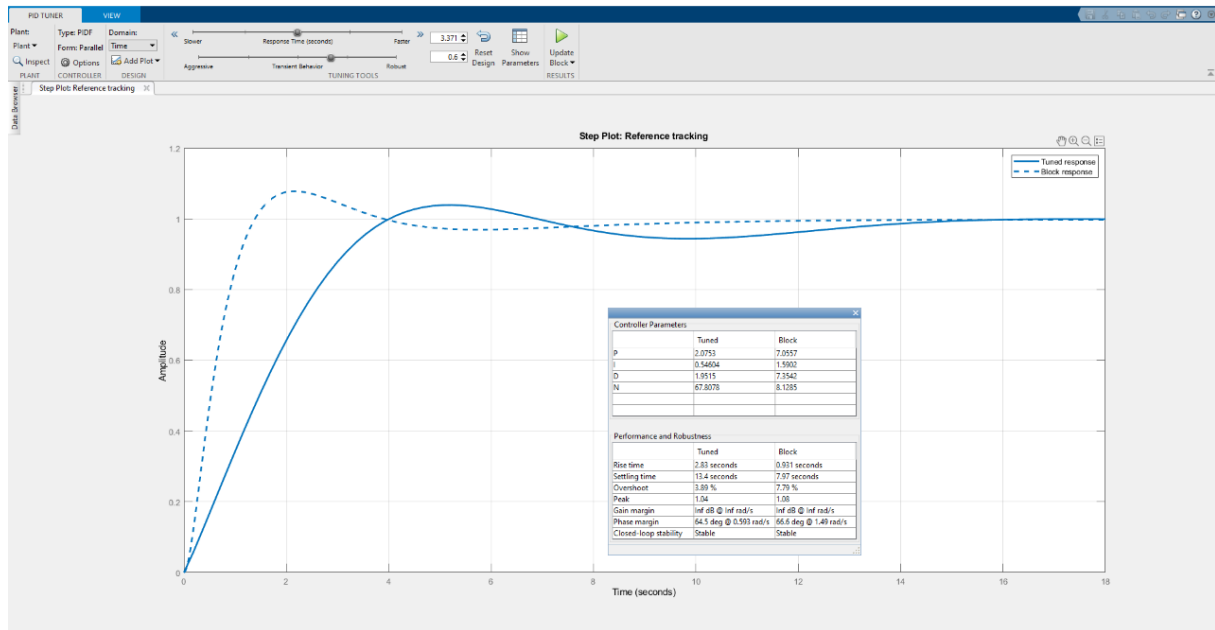


Figure 5: Automated tuning

Controller Parameters		
	Tuned	Block
P	2.0753	7.0557
I	0.54604	1.5902
D	1.9515	7.3542
N	67.8078	8.1285
Performance and Robustness		
	Tuned	Block
Rise time	2.83 seconds	0.931 seconds
Settling time	13.4 seconds	7.97 seconds
Overshoot	3.89 %	7.79 %
Peak	1.04	1.08
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	64.5 deg @ 0.593 rad/s	66.6 deg @ 1.49 rad/s
Closed-loop stability	Stable	Stable

Figure 6: Initial tuning figures

Due to the settling time being over the targeted 2 seconds further manual tuning was needed. By adjusting the response time and transient behaviour sliders shown in figure 7 the output could be tuned to achieve the desired outputs. Figure 9 and 10 below shows the final output achieved by manual tuning, bringing the settling time and overshoot percentage within the desired parameters.

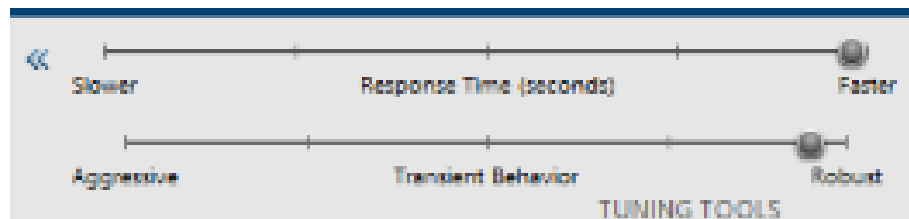


Figure 7: Manual tuning controls

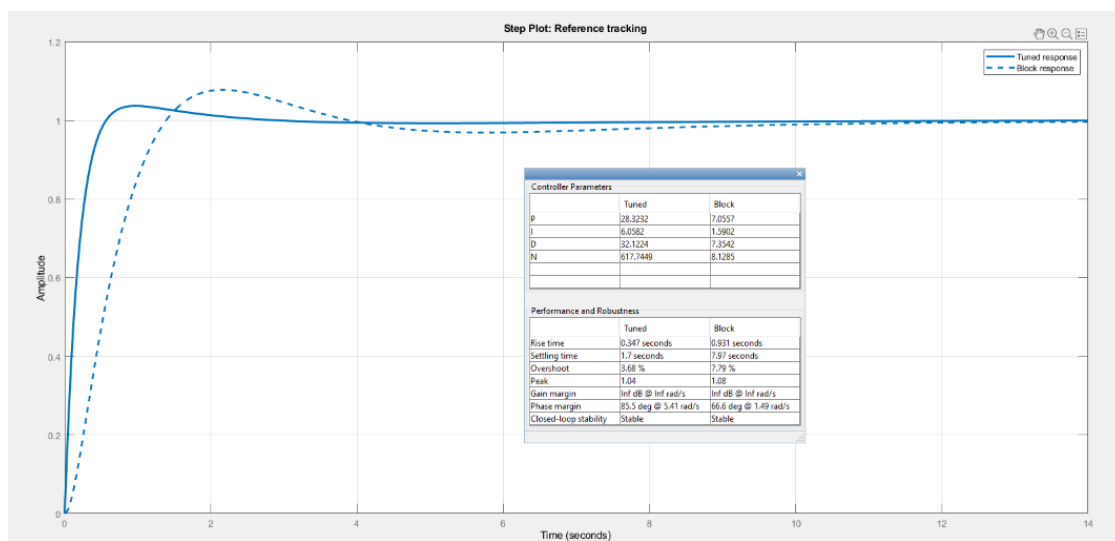


Figure 8: manual tuning results

Controller Parameters		
	Tuned	Block
P	28.3232	7.0557
I	6.0582	1.5902
D	32.1224	7.3542
N	617.7449	8.1285
Performance and Robustness		
	Tuned	Block
Rise time	0.347 seconds	0.931 seconds
Settling time	1.7 seconds	7.97 seconds
Overshoot	3.68 %	7.79 %
Peak	1.04	1.08
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	85.5 deg @ 5.41 rad/s	66.6 deg @ 1.49 rad/s
Closed-loop stability	Stable	Stable

Figure 9: Manual tuning figures

Figure 10 below shows the oscilloscope output results after the completion of tuning the system to achieve the desired output results.

1. Set point.
2. Error signal
3. Controller output
4. Actual system output

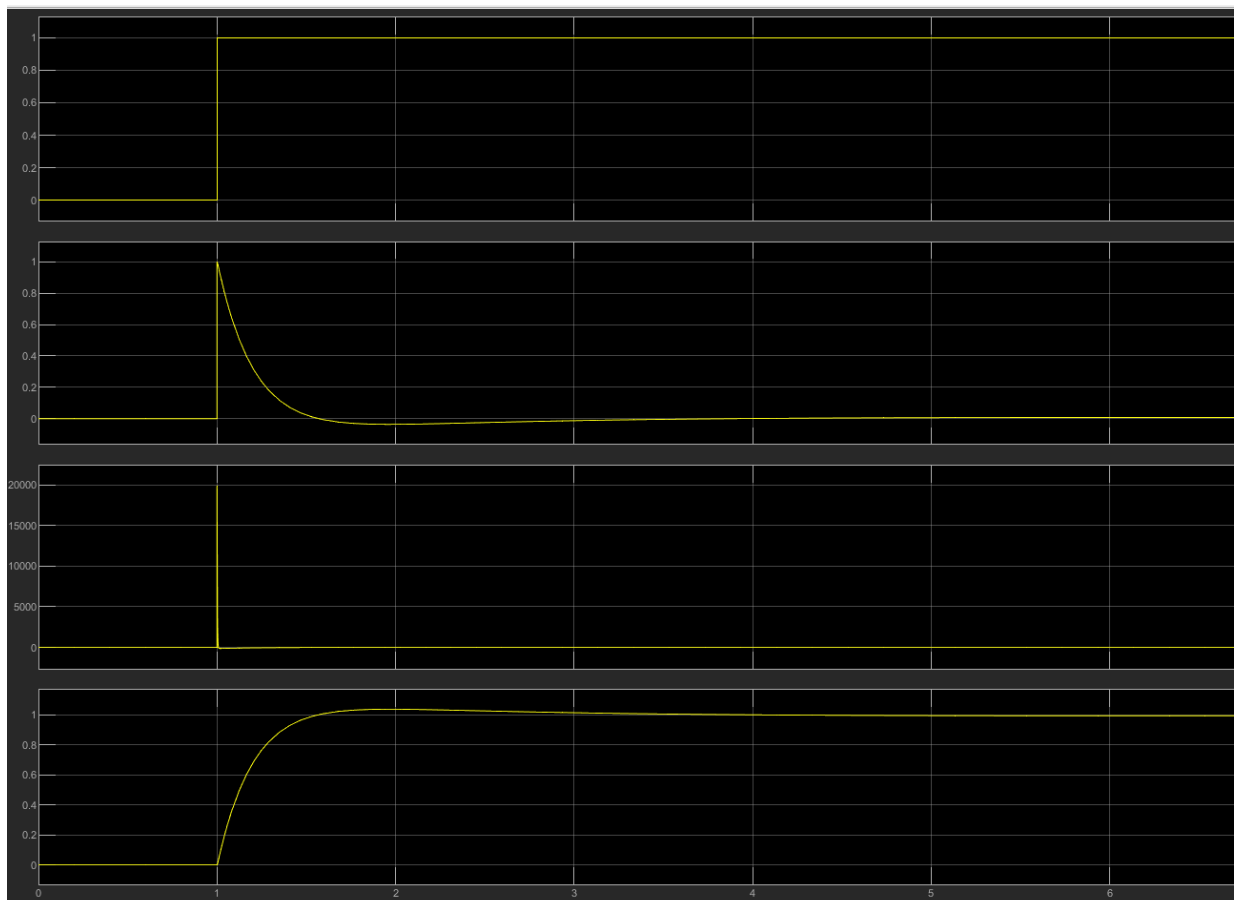


Figure 10:Tuned oscilloscope output

Figures 11 and 12 below show the results plotted in a Nyquist diagram. A Nyquist diagram is a type of polar plot format that shows frequency response, this allows for a simple visualization of stability in a feedback system [4]

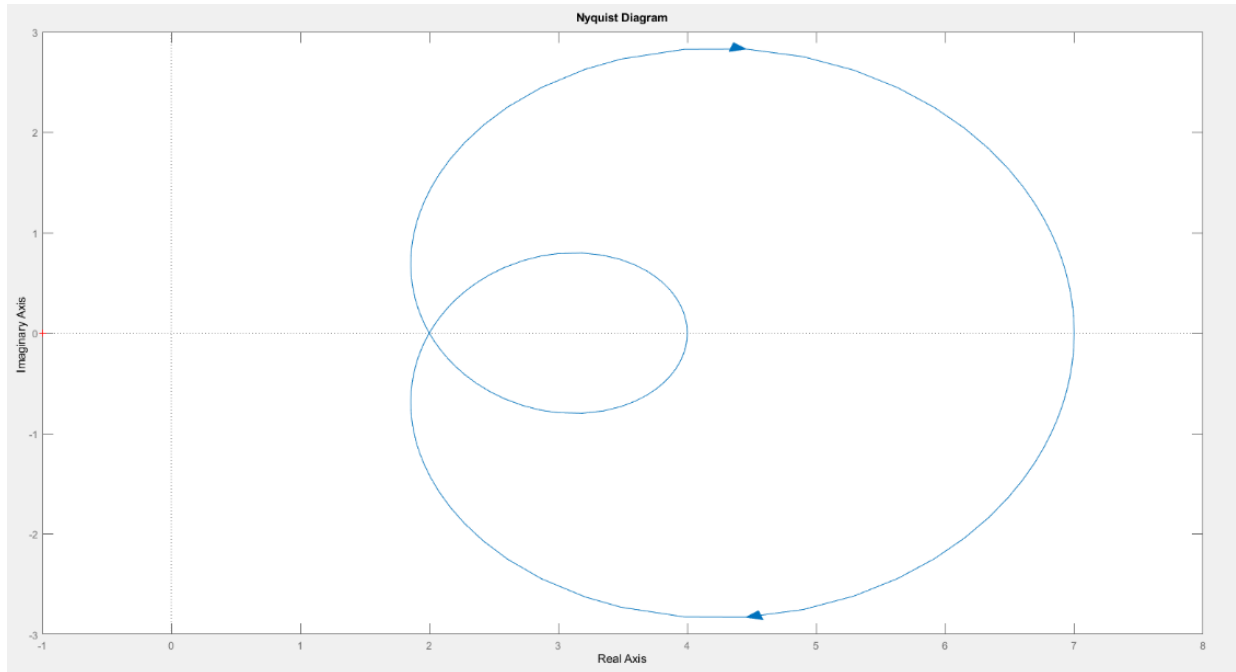


Figure 11: Nyquist diagram

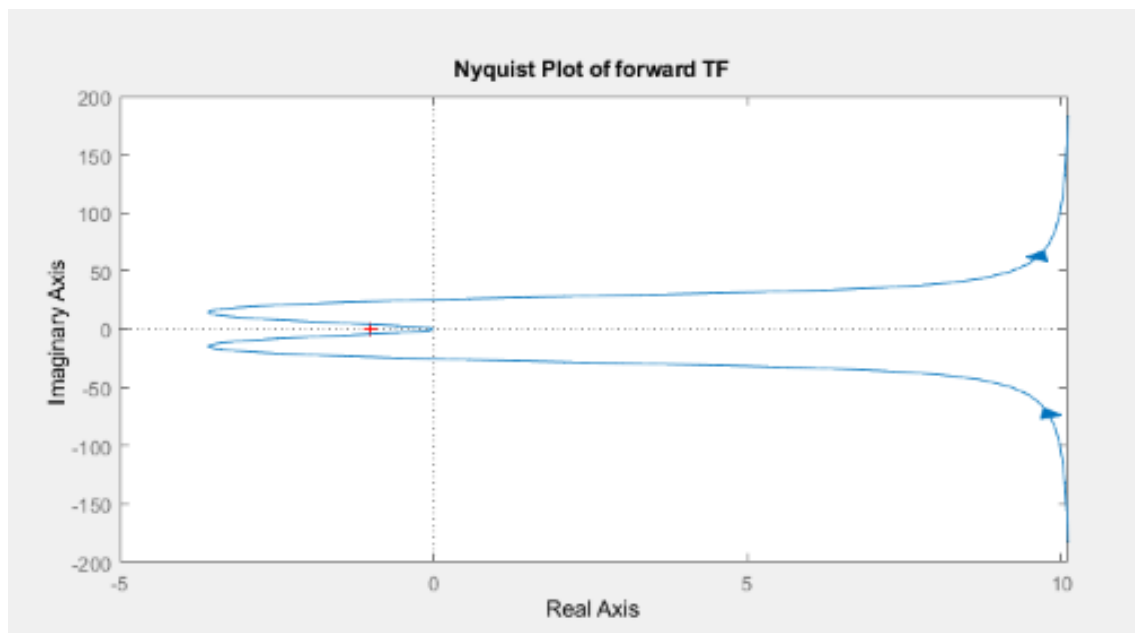


Figure 12: Nyquist plot of forward Transfer function

Figure 13 below shows a Bode diagram which shows frequency response within the system.

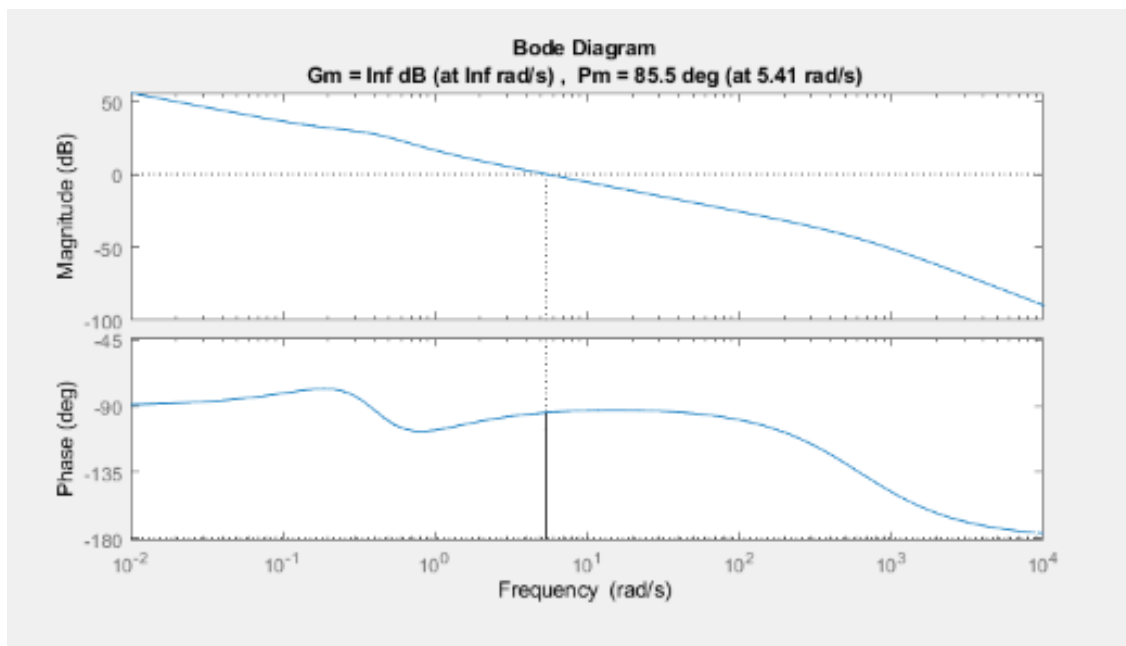


Figure 13: Bode Diagram

Figures 14 to 19 below shows the Pole-Zero plots against the output of the simulation. With these plots we can determine the stability of the system.

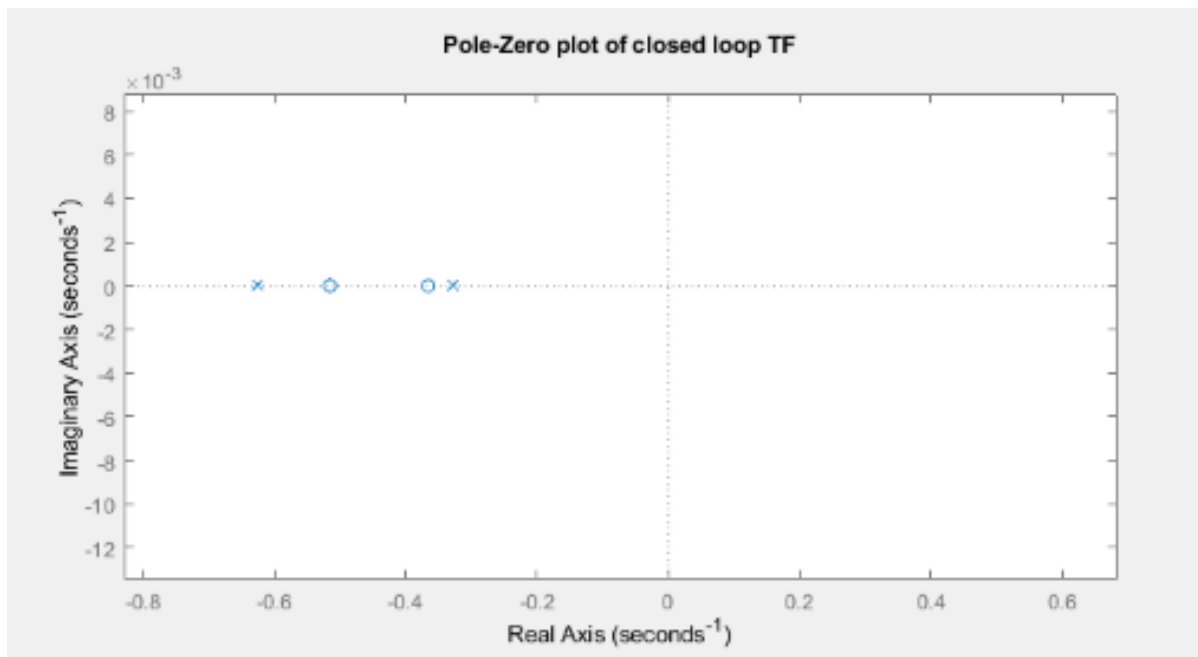


Figure 14: Pole-Zero plot closed loop transfer function

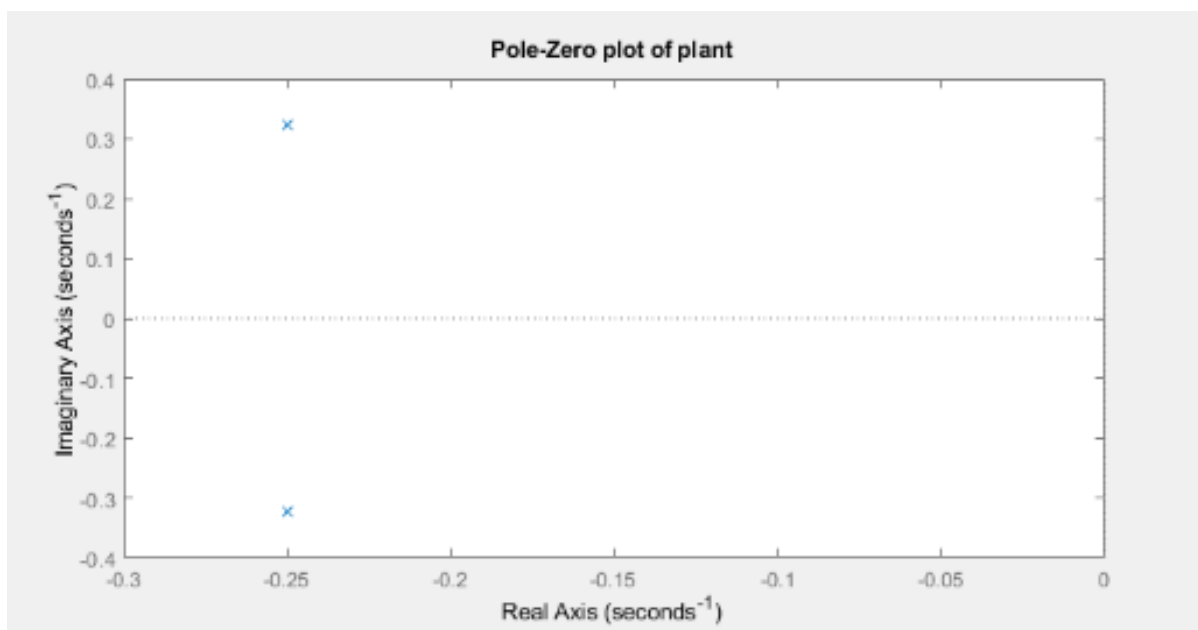


Figure 15: conjugate pair Pole-Zero plot pf plant

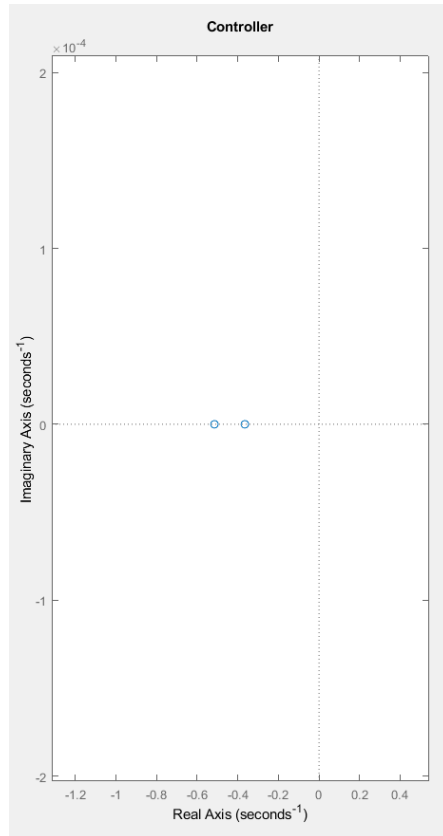


Figure 16: Pole Zero plot

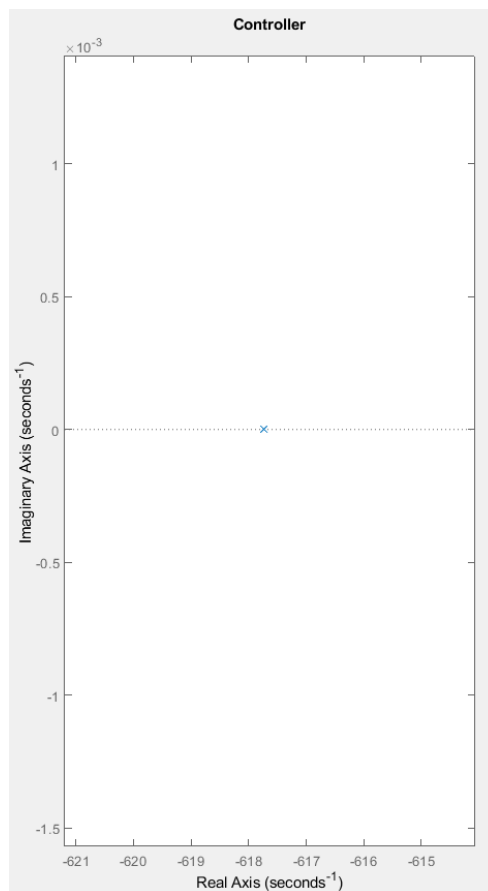


Figure 17: Pole Zero plot

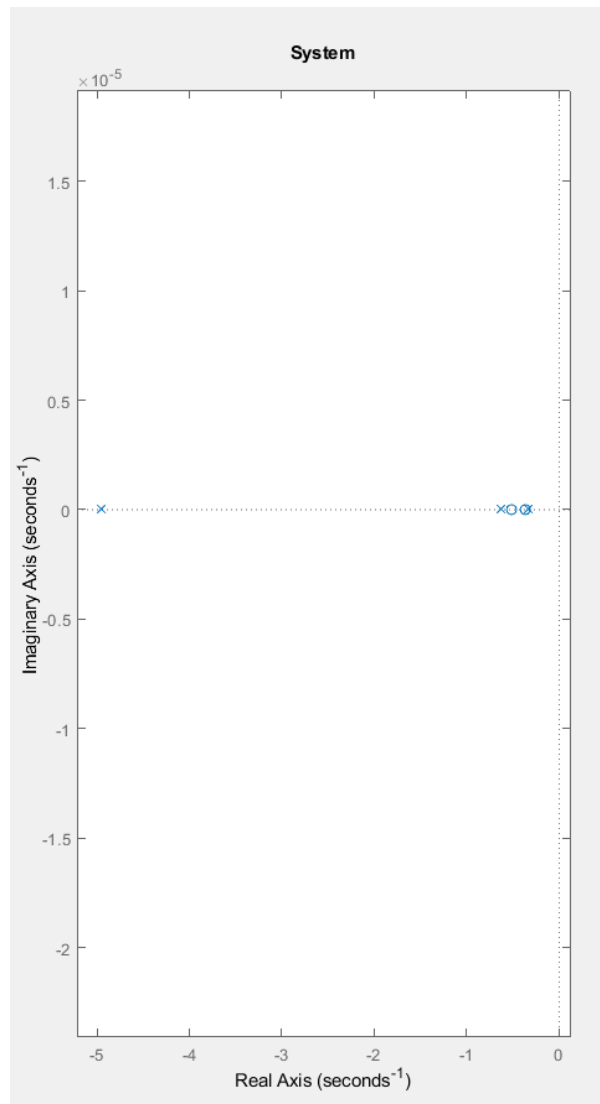


Figure 18:Pole-Zero plot

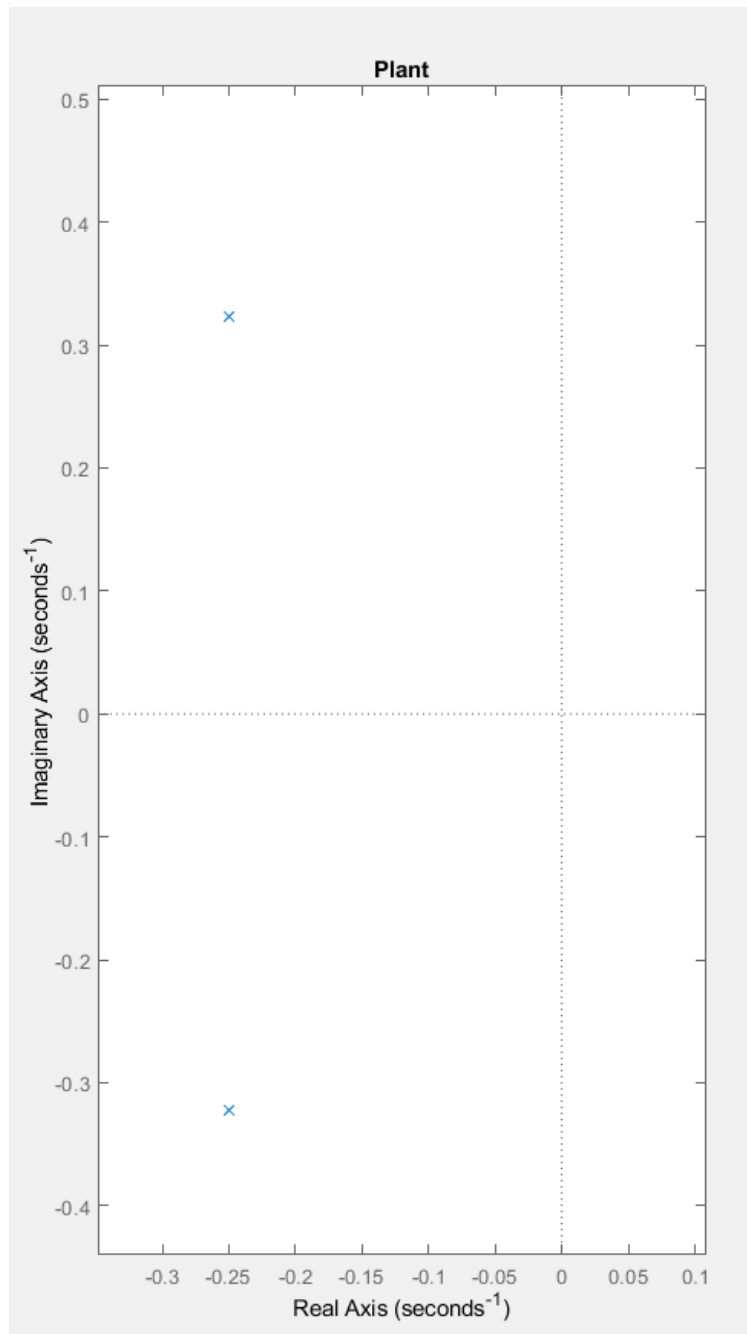


Figure 19: Conjugate pair of poles

Below in figure 20 we can see the final output data required by each of the PID sections and the ideal value needed to obtain the optimum output.

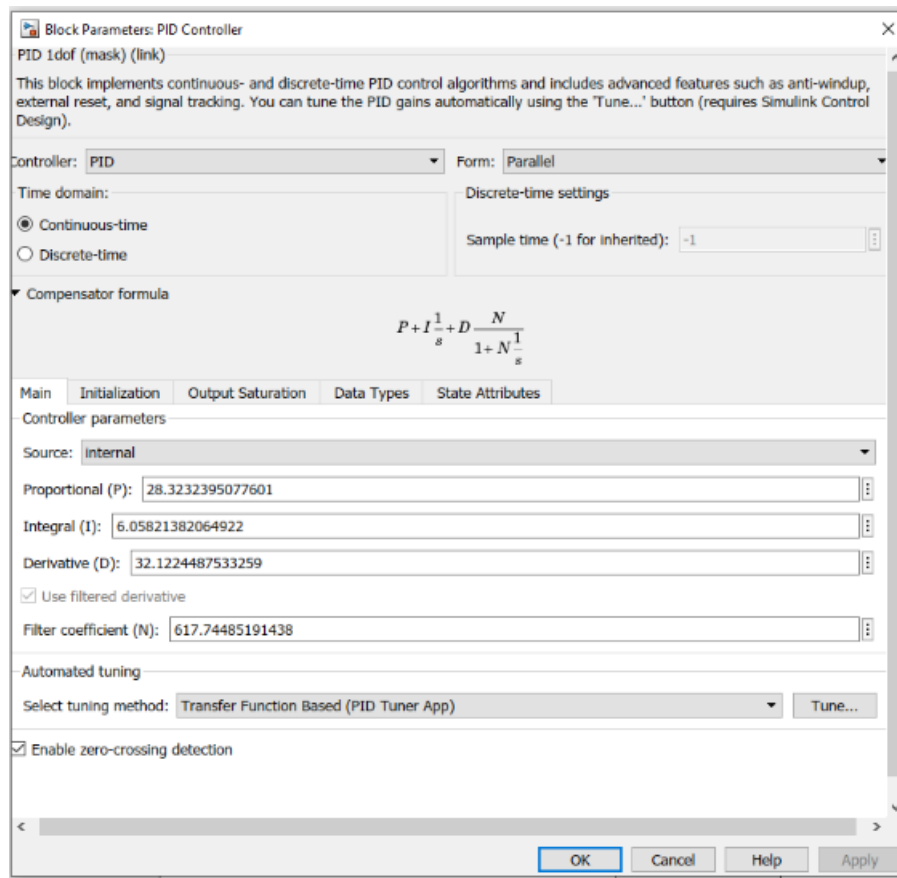


Figure 20: Final PID figures

Figure 21 below shows the next evolution of the circuit where the PID tuned figures have been used in a transfer function equation from Appendix 1.

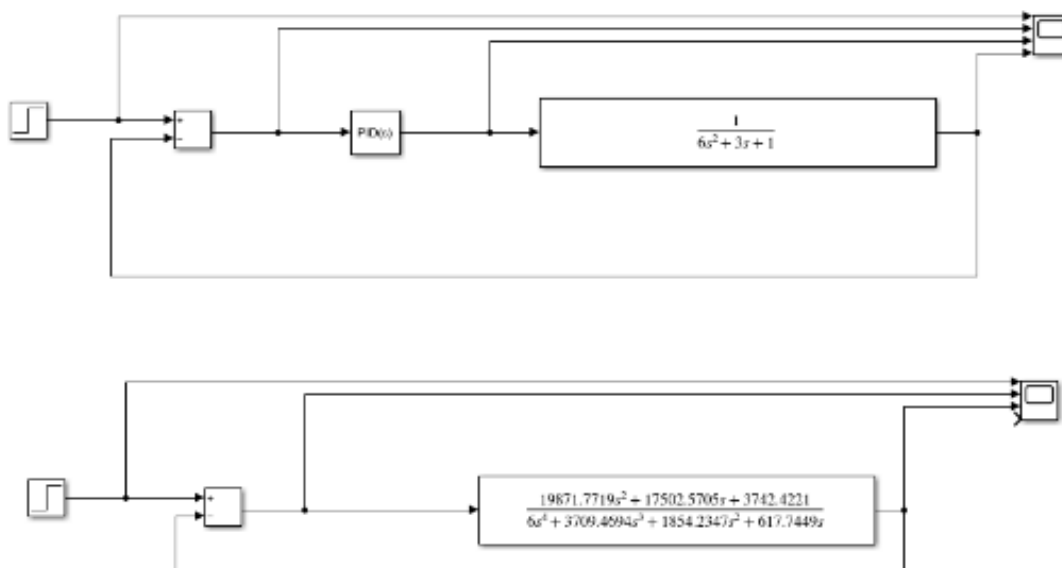


Figure 21: Circuit with Transfer function equation replacing PID controller

Figure 22 below shows the output signal from the new circuit layout. There are only 3 graphs due to the removal of the PID controller output signal. Each graph represents the following

1. Set point.
2. Error signal
3. System output

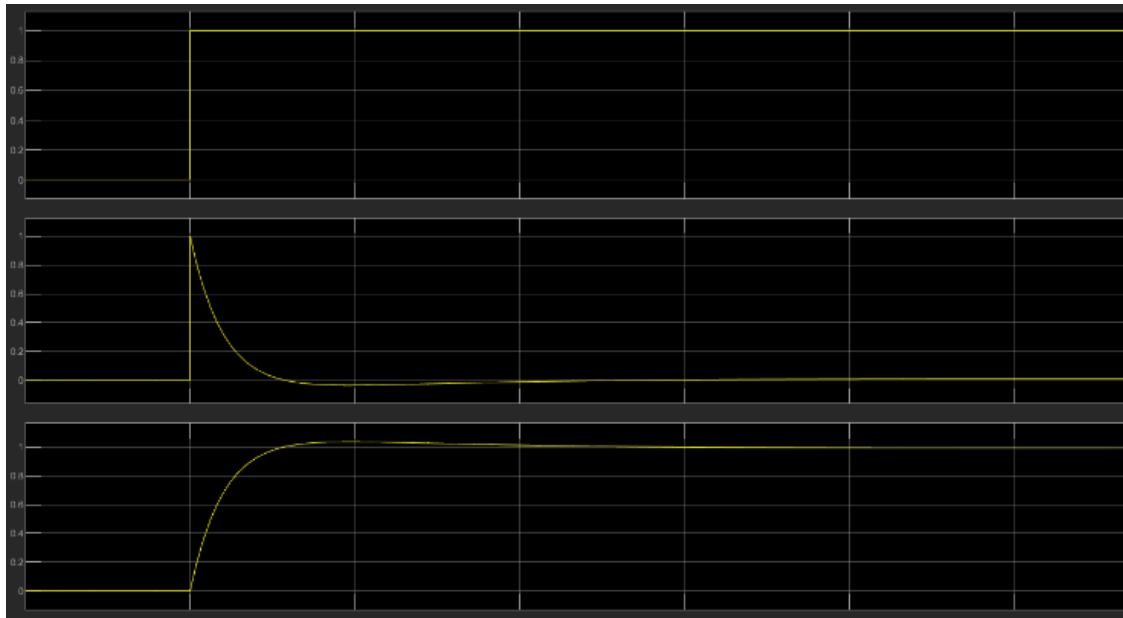


Figure 22: Output from figure 21 oscilloscope

Figure 23 below shows the final circuit layout after calculating out the feedback loop using the transfer function equation Appendix 1.

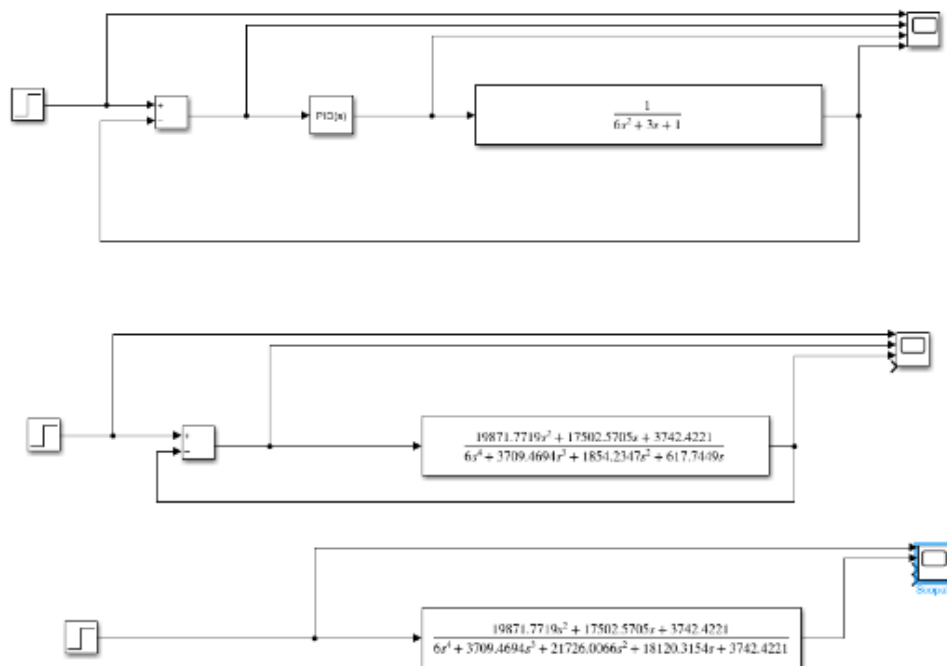


Figure 23: Circuit with Feedback correction formula applied

Figure 24 below shows the oscilloscope output signals. There are only 2 graphs due to the removal of the PID controller output signal and the feedback error signal. Each graph represents the following

1. Set point.
2. System output

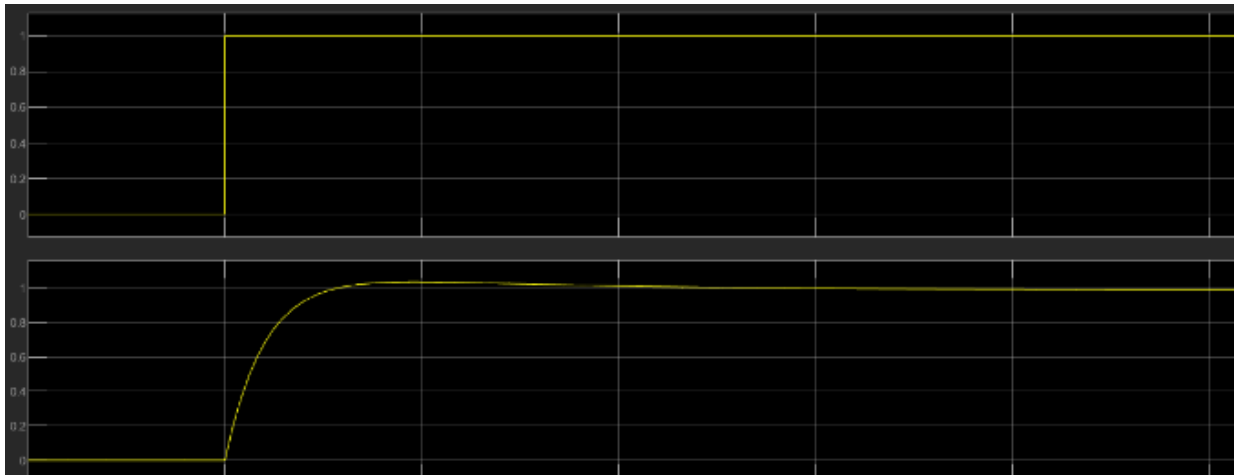


Figure 24:Output from figure 23 oscilloscope

Figure 25 below shows the comparison graph between the system output of all 3 circuits.

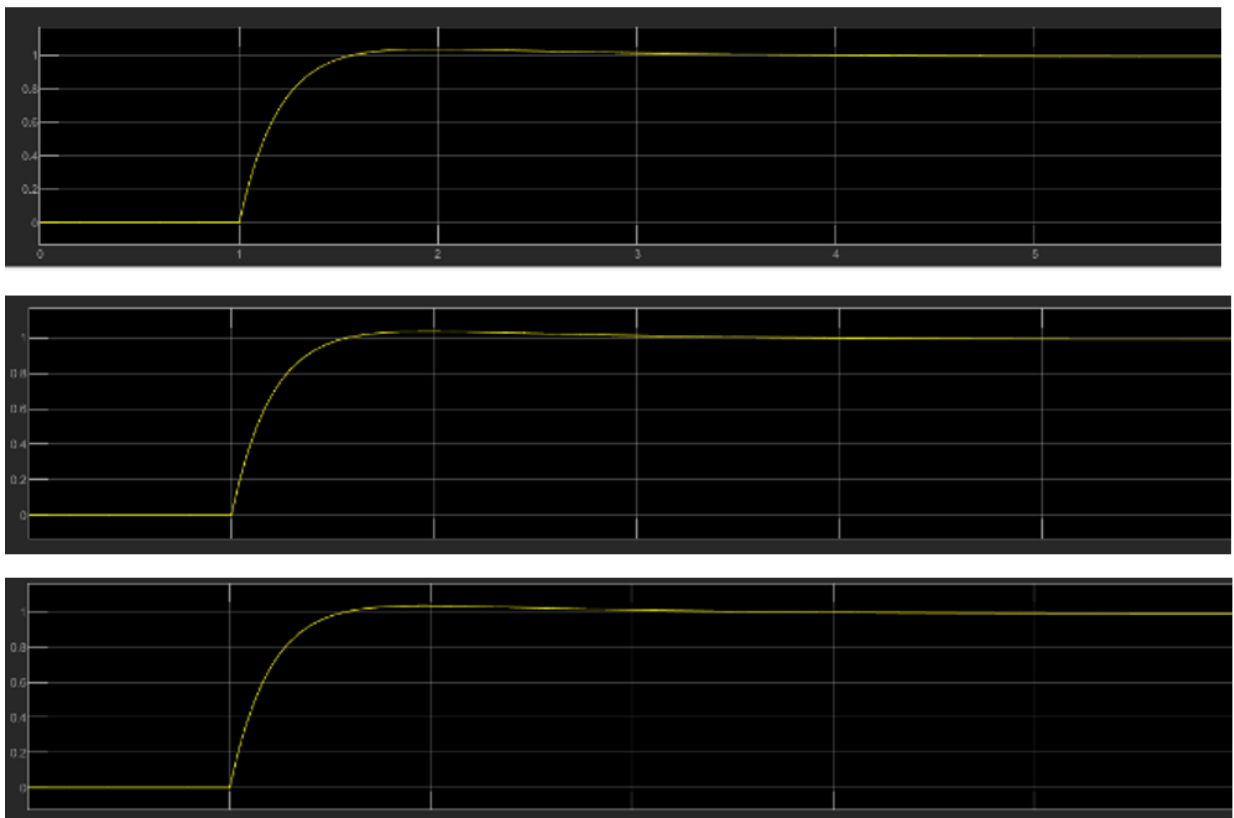


Figure 25:System output comparison between all 3 circuits

Figure 26 below shows the final circuit layout where all 3 output signals are combined into one oscilloscope for signal comparison.

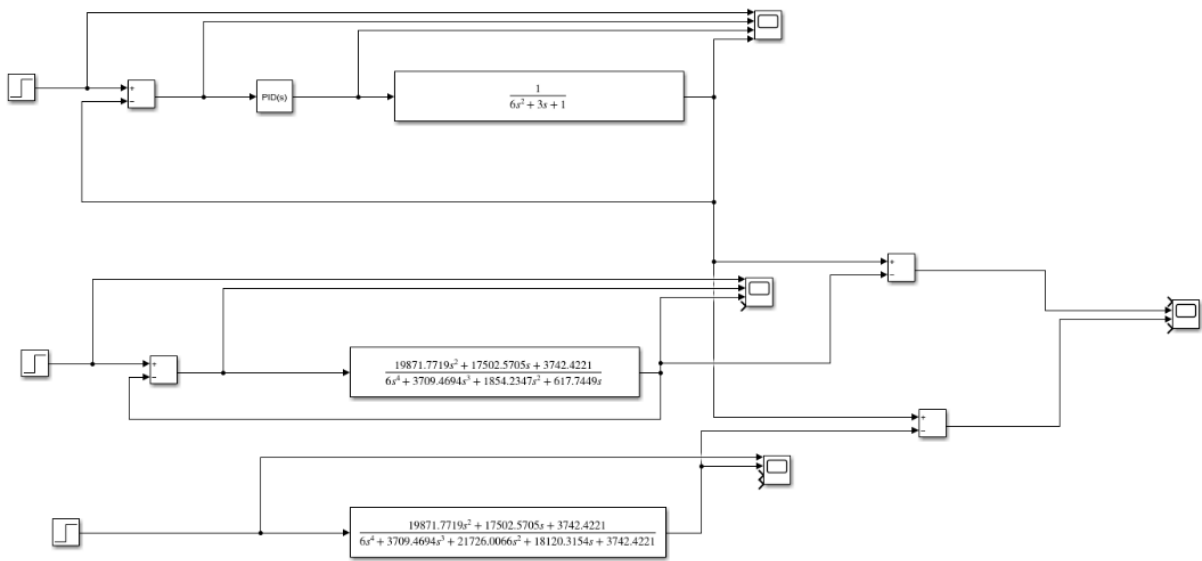


Figure 26: Final circuit configuration for combining output signals

Figure 27 below shows the combined output signal for all 3 system. With the top graph showing the combined signal of the first and second circuit, the second graph shows the combined output of the first and third circuit.

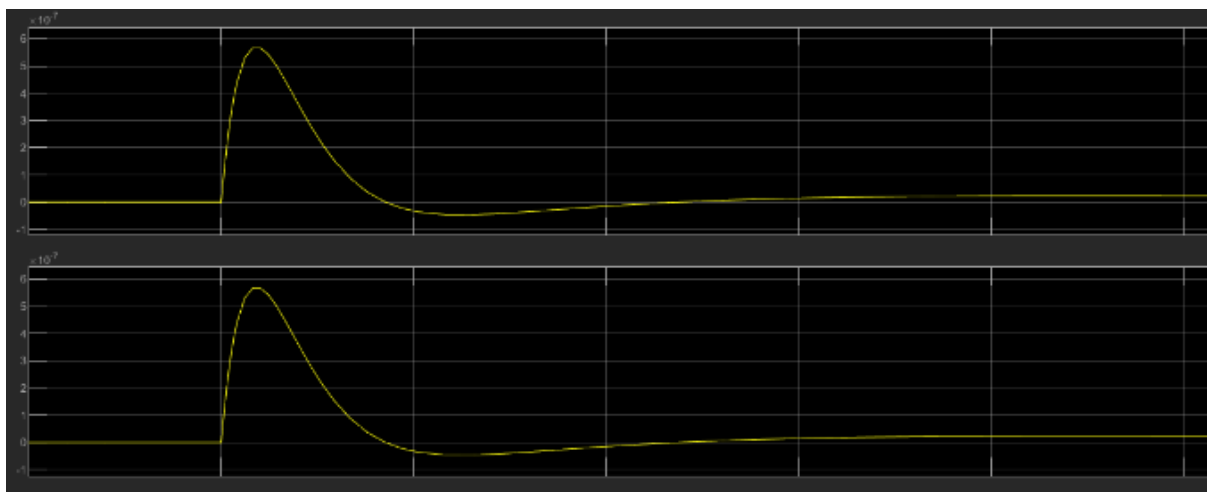


Figure 27: Combined output signal

Discussion

With the completion of the simulation, we can confirm through the mathematical process that the system is accurately simulating the process we are trying to replicate. In figure 3 above MATLAB carried out all calculations to produce an output signal it determines to be appropriate. Due to the requirements of the system to achieve a desired output of 2 second settling time and no more than 5% overshoot, the PID controller required tuning to achieve this goal.

The system is capable of automatically applying a tuning effect on the output as shown in figure 5, but as can be seen this required manual input to adjust the output figures to the desired parameters. By adjusting the response time and transient behaviour the output was tuned to achieve the requirements set out above. With figure 8 and 9 showing the final response curve and output figures.

By using the tuned figures found for the Proportional, Integral and Derivative from the system we can then complete the transfer function equation for the PID controller. By doing this it allows us to simplify the system by removing the PID controller and only having the plant equation control the process (Appendix 1). In figure 21 and 22 we can see the new circuit layout and the outputs on the oscilloscope. The circuit can then be further simplified by removing the feedback loop. This is done by completing the transfer function equation which incorporates the feedback into the whole system (Appendix 1). In figure 23 and 24 we can see the simplified version of this circuit and its output signals.

We can prove the mathematical equations were carried out correctly by comparing the output signal from all 3 circuits. In Figure 25 we can see all 3 circuits follow the exact same paths, with the rise time and settling time all being the same. This proves the theory, and the simulation are correct as they are producing the same output across the different circuits.

To further prove the accuracy of the equation in figure 26 the output signal from the 3 circuits were combined and a comparison is shown in figure 27. This comparison graph shows the error factor between the 3 circuits, with a lower number being the desired results. The graphs show an error factor of $\times 10^{-7}$ which is on the higher end of the comparison. This however can be explained by the rounding of the numbers used in the manual equation. MATLAB will use every figure after the decimal point to carry out its calculations. For time and simplicity, the manual equations in Appendix 1 were rounded to four decimal places.

Despite meeting the desired outputs, these outputs could be considered irrelevant if the system is not stable. An unstable system would result in uncontrollable oscillations moving the system further away from the desired output. To help determine if a system is in a stable state we use Pole zero plots, Bode diagrams and Nyquist plots, all of which can be seen in figures 11 to 19. Figure 11 shows a Nyquist plot and is a good visual representation of a stable system, due to the point at which the lines cross the real number and the distance between the 2 crossing points this equates to a stable system [5].

In figure 13 we can see the bode plot which represents frequency in response to magnitude and phase. Gain and phase margin represent the point at which the system could become unstable. The larger the margin the more stable the system is. With a gain margin of infinity this further proves the system in question is stable.

Using the Pole zero plots in figures 14 to 19 we can see a visual response of a system with a known transfer function (as calculated in Appendix 1) and they graphically show the stability of a system. All

the plots within figures 14 to 19 are shown in negative numbers. Due to the plots being on the negative side of the graph it represents a stable system due to an exponential decay to zero [6] with the conjugate pairs in figure 19 represents oscillation in the system. This is represented below in figure 28.

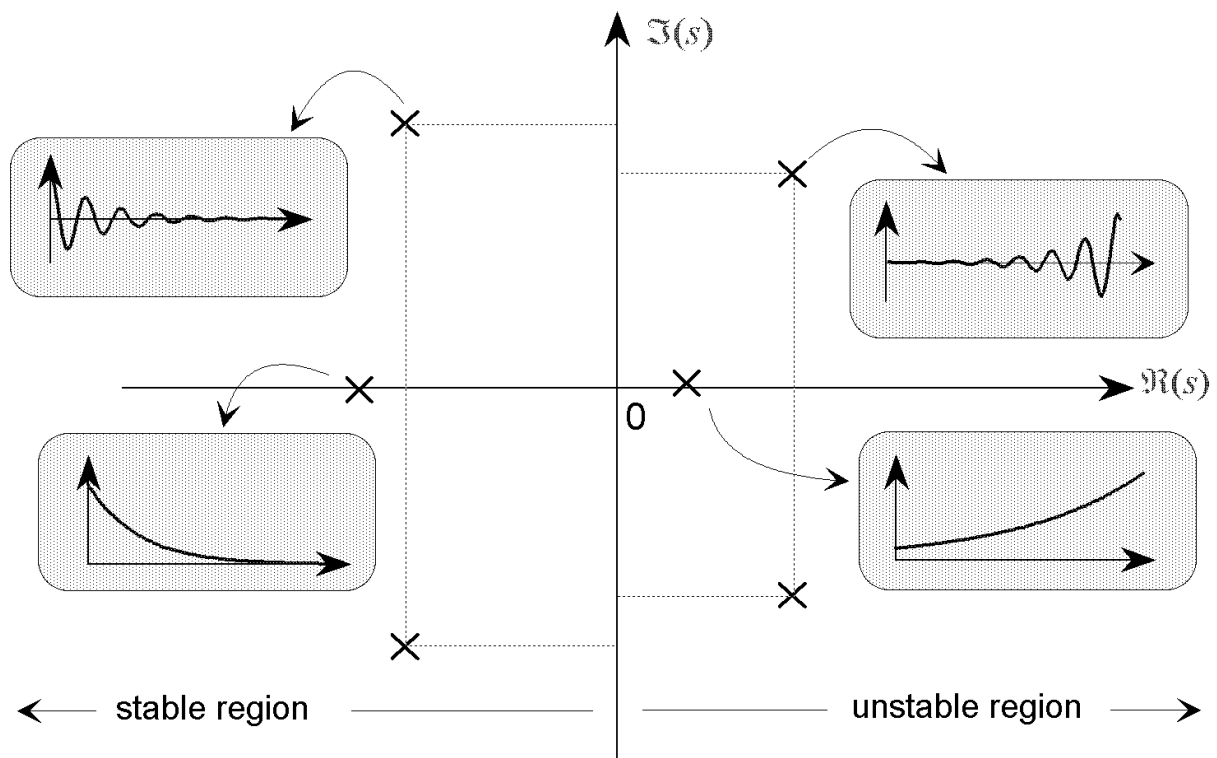


Figure 28: Pole Zero locations and their representative signals [6]

Conclusions

In the simulations described above a suitable control system has been designed with the use of a PID controller as well as sole use of a transfer function. The outputs have been compared together providing reassurance that both systems would react with the same output if they were presented with the same input.

An analysis of the response characteristics with the use of Nyquist plots, Bode Diagram and Pole-Zero diagrams determined that the system is stable. This further proves the system is suitable for the desired use of rotating a CIWS to the correct position within the desired time and overshoot parameters.

With the simulations, equations and results proving the proposed control system is adequate for the task in question, the next stage in the development would be to produce a physical representation of the control system. By doing this the theory that has been proven in these simulations can be tested in a real-world situation. Virtual simulations are great for planning and down selecting a bunch of ideas/designs into a final version to take forward. This helps to keep project and research & development costs to a minimum. The downside to this is the fact the simulations take place in a sanitary and controlled environment using virtual components that work exactly as they are supposed to at the levels they should. Whereas in the real work components have tolerances levels

they are designed to work within. So, a virtual simulation might say the control system is suitable to achieve the 2 second settling time and no more than 5% overshoot, but in a real-world example there could be vastly different results.

References

- [1] A. M., "PID Control," *CONTROL SYSTEMS, ROBOTICS, AND AUTOMATION*, vol. II.
- [2] "Tranics Zone," 22 03 2020. [Online]. Available: <https://www.tronicszone.com/blog/pid-controller-using-microcontrollers/>. [Accessed 18 04 2023].
- [3] "Math Works," [Online]. Available: <https://uk.mathworks.com/help/simulink/>. [Accessed 18 04 2023].
- [4] R. M. Tony Roskilly, "Frequency Domain anaklysis," in *Marine Systems Identification, Modeling and Control*, Butterworth-Heinemann, 2015, pp. 123-147.
- [5] R. Kein, "All about circuits," 05 07 2019. [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/how-to-use-nyquist-plots-for-stability-analysis/#:~:text=With%20a%20Nyquist%20plot%2C%20you,that%20is%20more%20reliably%20stable..> [Accessed 24 04 2023].
- [6] Massachusetts Institue of Technology, Department of Mechanical Engineering, "Analysis and Design of Feedback Control System".

Appendix 1

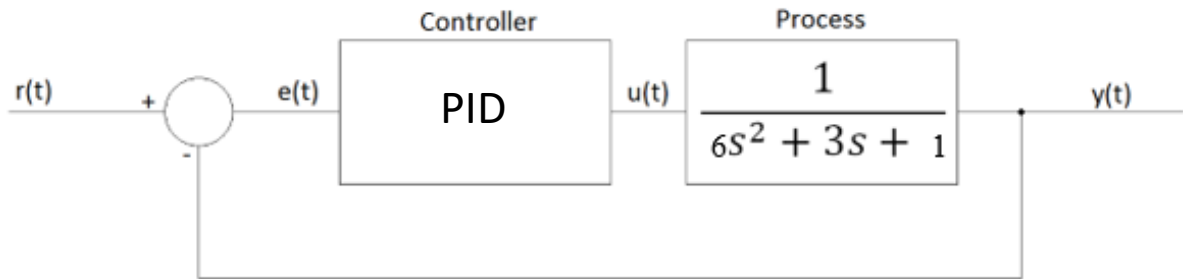
Transfer function calculations

$$G(s) = \frac{1}{as^2 + bs + c}$$

Taking a = 6, b = 3, c = 1 our transfer function now becomes:

$$G(s) = \frac{1}{6s^2 + 3s + 1}$$

When drawn as a block diagram, the system now looks like this:



Using the PID tuner in MATLAB, the following values were obtained:

K_p: 28.3232

K_i: 6.0582

K_d: 32.1224

N: 617.7449

So the transfer function equation for the PID controller will be:

$$K_p + K_i \frac{1}{s} + K_d \frac{s}{\frac{1}{N}s + 1}$$

$$28.3232 + \frac{6.0582}{s} + 32.1224 \frac{s}{\frac{1}{617.7449}s + 1}$$

To calculate the forward path transfer function of the system, the controller's transfer function must be multiplied by the transfer function of the process.

$$\left(28.3232 + \frac{6.0582}{s} + 32.1224 \frac{s}{\frac{1}{617.7449}s + 1} \right) \times \left(\frac{1}{6s^2 + 3s + 1} \right)$$

$$\left(28.3232 + \frac{6.0582}{s} + 32.1224 \frac{s}{\frac{s+617.7449}{617.7449}}\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

$$\left(28.3232 + \frac{6.0582}{s} + 32.1224 \left(s \div \frac{s+617.7449}{617.7449}\right)\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

$$\left(28.3232 + \frac{6.0582}{s} + 32.1224 \left(s \left(\frac{617.7449}{s+617.7449}\right)\right)\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

$$\left(28.3232 + \frac{6.0582}{s} + 32.1224 \left(\left(\frac{617.7449s}{s+617.7449}\right)\right)\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

$$\left(28.3232 + \frac{6.0582}{s} + \left(\frac{19843.44878s}{s+617.7449}\right)\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

$$\left(\frac{((s(s+617.7449)) \times 28.3232) + (s+617.7449)6.0582 + s(19843.4487s)}{s(s+617.7449)}\right) \times \left(\frac{1}{6+3s+1}\right)$$

$$\left(\frac{28.3232s^2 + 17496.51235s + 6.0582s + 3742.4221 + 19843.44878s^2}{s^2 + 617.7449s}\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

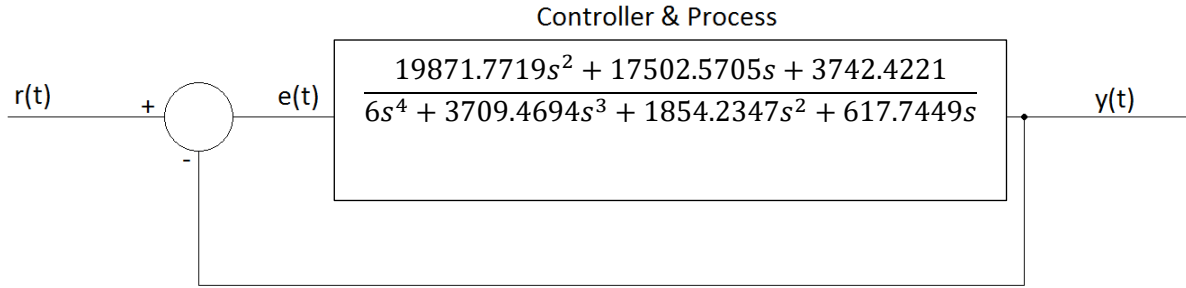
$$\left(\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{s^2 + 617.7449s}\right) \times \left(\frac{1}{6s^2+3s+1}\right)$$

$$\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{(s^2 + 617.7449s)(6s^2 + 3s + 1)}$$

$$\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3s^3 + 1s^2 + 3706.4694s^3 + 1853.2347s^2 + 617.7449s}$$

$$\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}$$

The block diagram can now be redrawn as:



Now we remove the feedback loop and calculate the transfer function of the system as a whole.

Using the following equation:

$$TF = \frac{G}{1 + GH}$$

Where TF is the transfer function of the whole system, G is the forward path transfer function, and H is the transfer function of the feedback loop, in this case $H = 1$

$$TF = \frac{\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}{1 + \left(\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s} \right) \times 1}$$

$$TF = \frac{\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}{\frac{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s} + \frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}$$

$$TF = \frac{\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}{\frac{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s + 19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}$$

$$TF = \frac{\frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}{\frac{6s^4 + 3709.4694s^3 + 21726.0066s^2 + 18120.3154s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}}$$

$$TF = \frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s} \times \frac{6s^4 + 3709.4694s^3 + 1854.2347s^2 + 617.7449s}{6s^4 + 3709.4694s^3 + 21726.0066s^2 + 18120.3154s + 3742.4221}$$

$$TF = \frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 21726.0066s^2 + 18120.3154s + 3742.4221}$$

$$TF = \frac{19871.7719s^2 + 17502.5705s + 3742.4221}{6s^4 + 3709.4694s^3 + 21726.0066s^2 + 18120.3154s + 3742.4221}$$

The block diagram can now be redrawn again as a single block.

