SCHOOL OF ENGINEERING
AND BUILT ENVIRONMENT

Control Systems

MMH624556

Course Work 1

Author:	Luke Byrne
Matriculation Number:	S1823325
Program Code:	EBE_CC_BMEEE-20-AB
Program Title:	Meng Electronic and Electrical Engineering
Module Code:	MMH624556-20-A
Module Title:	Control Systems
Date of Submission:	20/01/2021

ABSTRACT

This report provides an analysis of tuning techniques for a PI (Proportional-Integral) controller, when given an open loop plant step response. The tuning relations investigated here are: Haglund Astrom, Cohen & Coon, and Ziegler Nichols. Each controlled system was tested for unit step and unit disturbance. From this analysis it is concluded that the Haglund Astrom tuning method provides the best step response parameters, however the Ziegler Nichols method provides the best disturbance rejection.

The impact of including a saturation block within the system is also investigated. It is concluded that a saturation block could substantially change step and disturbance responses for all controlled systems. An outline of anti-windup techniques to compensate for a saturation block is presented. However, no specific parameters for the saturation block were given, and therefore no specific tuning relations for the anti-windup techniques may be recommended.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

1 INTRODUCTION

PI (Proportional-Integral) controllers are used to improve output responses of industrial systems [1]. This report provides an analysis of tuning techniques for a PI controller used within a chemical processing plant. The tuning relations investigated here are: Haglund Astrom, Cohen & Coon, and Ziegler Nichols, however many other tuning relations are available [1]. Each controlled system was tested for unit step and unit disturbance.

The impact of including a saturation block within the system is also provided. It is concluded that a saturation block could substantially change step and disturbance responses for all controlled systems. An outline of anti-windup techniques to compensate for a saturation block is presented. However, no specific parameters for the saturation block were given, and therefore no specific tuning relations for the anti-windup techniques may be recommended.

1.1 AIMS AND OBJECTIVES

- Use given plant step response data to derive a plant transfer function
- Design three controllers, using methods given in question
- Investigate step response, and disturbance response parameters of each controlled system
- Evaluate the pros and cons of each controller, ultimately making a recommendation
- Investigate the significance of saturation “Block X” in the alternative system diagram
- Outline changes required to compensate for Block X, demonstrating anti-windup implementation

2 METHODOLOGY

MATLAB [2] was used for all calculation and simulation in this analysis, except when investigating the significance of Block X in the alternative diagram. Simulink [3] was used for the Block X investigation, as the diagram shows Block X as a saturation block [4]. Saturation is not a linear time invariant function, and therefore cannot be computed using ordinary MATLAB. Simulink was also used for validation of the MATLAB results.

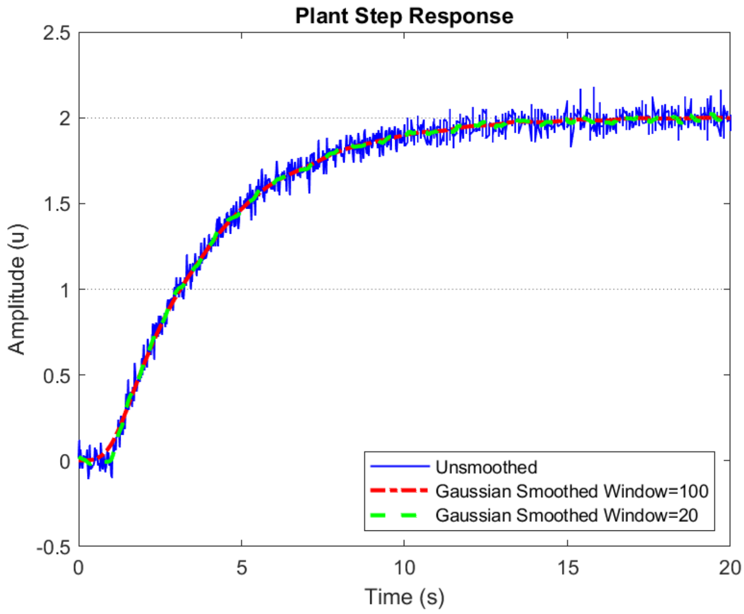
2.1 DERIVING OPEN LOOP PLANT TRANSFER FUNCTION

Smoothing data

The given plant data contained significant amounts of noise. In order to make this data useable it was necessary to smooth out this noise. To this end, two Gaussian smoothing functions were applied [5]. One function used a wide window of 100, for use in deriving the magnitude K and time constant τ. However, a wide window was inappropriate for determining θ, the time delay. For this a short window of 20 was used.

The output of this process may be seen in Figure 1:

FIGURE 1: OPEN LOOP PLANT STEP RESPONSE



Extracting θ, τ, K

Once smoothed, it was possible to extract θ, τ, and K for use in deriving the open loop plant transfer function. K is the final resting value of the open loop step response. Therefore, K was set as the final value for the heavily smoothed data. θ is the time delay between step input and plant response. At time θ the gradient of the response transitions between 0 and the maximum observable gradient. As such, θ was set to the time when the second derivative of the lightly smoothed step response data was highest. τ, the time constant is defined as the time when the plant step response is equal to 63% of K, minus the time delay θ [6]. Please see appendix A for the complete MATLAB script and explanation.

This gave: K = 1.9976 θ = 1.0200 τ = 3.0200

Calculating open loop plant transfer function

Once the values for θ, τ, K were determined it was possible to derive the open loop transfer function for the plant, using the generic formula [6]:

$$G(s) = \frac{K\, e^{-\theta s}}{\tau s + 1}$$

EQUATION 1: GENERIC PLANT TRANSFER FUNCTION

Giving the open loop plant transfer functions as:

$$G(s) = \frac{1.998e^{-1.0200s}}{3.02s + 1}$$

EQUATION 2: OPEN LOOP PLANT TRANSFER FUNCTION

Plotting the derived transfer function over the given plant response data verifies that it is approximately correct, as seen in Figure 2:

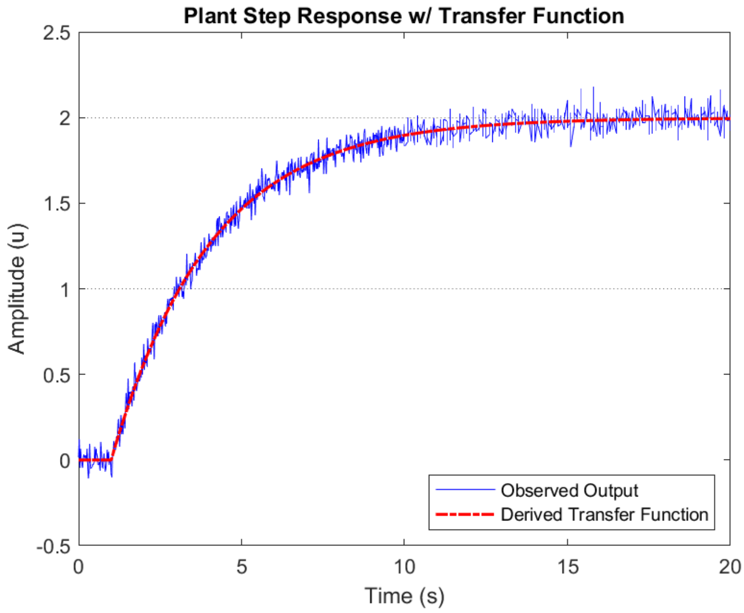


FIGURE 2: PLANT STEP RESPONSE WITH DERIVED TRANSFER FUNCTION

2.2 CALCULATING HAGLUND ASTROM CONTROLLER TRANSFER FUNCTION

The following table was used to determine the values for K_c, and τ_I:

G(s)	K_c	τ_I
$\frac{Ke^{-\theta s}}{s}$	$\frac{0.35}{K\theta}$	7θ
$\frac{Ke^{-\theta s}}{\tau s+1}$	$\frac{0.14}{K}+\frac{0.28\tau}{\theta K}$	$0.33\theta+\frac{6.8\theta\tau}{10\theta+\tau}$

TABLE 1: HAGLUND ASTROM CONTROLLER SETTINGS

As the plant transfer function G(s), shown in Equation 1, matches the form given in row three of Table 1, the following equations were used:

$$K_c=\frac{0.14}{K}+\frac{0.28\tau}{\theta K}$$

$$\tau_I=0.33\theta+\frac{6.8\theta\tau}{10\theta+\tau}$$

EQUATION 3: HAGLUND ASTROM PI TUNING RELATIONS

Giving $K_c = 0.4851$ and $\tau_I = 1.9211$

Using the standard PI Controller equation [6]:

$$G(s) = K_c\left[1 + \frac{1}{\tau_I}\right]$$

EQUATION 4: STANDARD PI CONTROLLER EQUATION

The Haglund Astrom controller transfer function was determined to be:

$$G(s) = \frac{0.9319s + 0.4851}{1.921s}$$

EQUATION 5: HAGLUND ASTROM CONTROLLER TRANSFER FUNCTION

2.3 CALCULATING COHEN AND COON CONTROLLER TRANSFER FUNCTION

From the question, the following equations were used to calculate K_c, and τ_I:

$$K_c = \frac{1}{K}\frac{\tau}{\theta}\left[0.9 + \frac{\theta}{12\tau}\right]$$

$$\tau_I = \frac{\theta\left[30 + 3\left(\frac{\theta}{\tau}\right)\right]}{9 + 20\left(\frac{\theta}{\tau}\right)}$$

EQUATION 6: COHEN & COON PI TUNING RELATIONS

Giving $K_c = 1.3757$ and $\tau_I = 2.0078$

Using the standard PI Controller equation given in Equation 2 the Cohen and Coon controller transfer function was determined to be:

$$G(s) = \frac{2.762s + 1.376}{2.008s}$$

EQUATION 7: COHEN & COON CONTROLLER TRANSFER FUNCTION

2.4 ZEIGLER NICHOLS TUNING

For Ziegler-Nichols tuning R_N is the ratio between the maximum gradient of the open loop plant step response and the magnitude of the input signal, and L is the time delay previously given as θ in this report [7]. In this case, the magnitude of the input signal is 1, and therefore R_N is equal to the maximum gradient of the step response. This was extracted directly using the MATLAB gradient function [8]. Please see Appendix A for details.

Giving $R_N = 0.6400$

The line of maximum gradient may be seen in Figure 3:

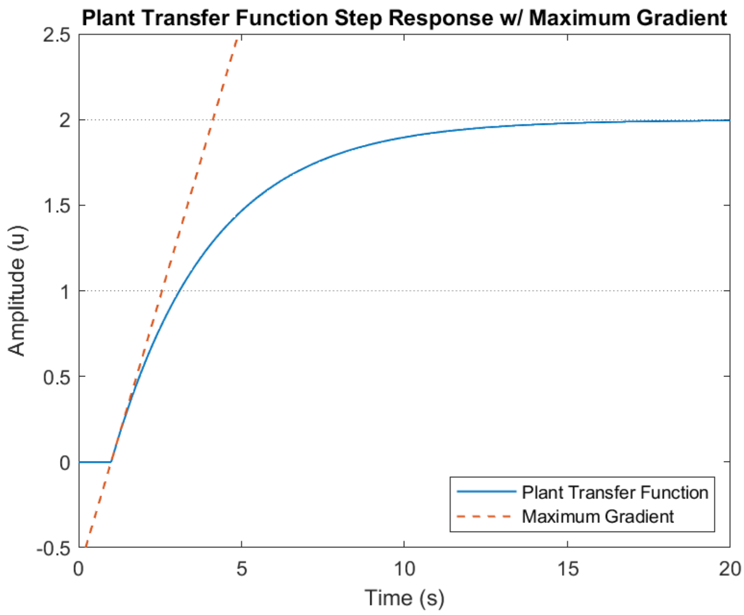


FIGURE 3: OPEN LOOP PLANT TRANSFER FUNCTION STEP RESPONSE WITH LINE OF MAXIMUM GRADENT

From the question, the following table was used to calculate K_c, and τ_I:

Controller Structure		Proportional Gain K_P	Integral Time Constant τ_I	Derivative Time Constant τ_D
Case (i)	P	$\frac{1}{R_N L}$		
Case (ii)	PI	$\frac{0.9}{R_N L}$	$3L$	
Case (iii)	PID	$\frac{1.2}{R_N L}$	$2L$	$0.5L$

TABLE 2: ZIEGLER NICHOLS CONTROLLER SETTINGS

As the controller structure being investigated is a PI controller, the following equations were used:

$$K_c = K_P = \frac{0.9}{R_N L}$$

$$\tau_I = 3L$$

EQUATION 8: ZIEGLER NICHOLS PI TUNING RELATIONS

Giving $K_c = 1.3786$ and $\tau_I = 3.0600$

Using the standard PI Controller equation given in Equation 4, the Zeigler-Nichols controller transfer function was determined to be:

$$G(s) = \frac{4.219s + 1.379}{3.06s}$$

EQUATION 9: ZIEGLER NICHOLS CONTROLLER TRANSFER FUNCTION

2.5 INVESTIGATING RESPONSES TO UNIT STEP DISTURBANCE

The controlled systems were further tested for their response to a unit step disturbance at 40 seconds. As the MATLAB function stepplot [9] cannot be nested inside subsequent stepplot functions, it was necessary to manually add the disturbance transfer function before passing to stepplot. However, this method produces repeated disturbances, with an interval of 40s. This anomaly is due to how MATLAB calculates transfer functions and is noted in the documentation for the feedback function [10]. However, limiting the systems' output plots to 80 seconds in length avoids inclusion of the repetitions. This repetition is not present when simulating the systems in Simulink.

2.6 INVESTIGATING SIGNIFICANCE OF BLOCK X

The diagrammatic representation of Block X was compared to the Simulink library. By doing so, Block X was assumed to be a saturation block. The Simulink documentation for saturation was then checked to ensure this was a reasonable choice [4]. Saturation blocks limit outputs within a pre-determined upper and lower limit. Saturation in often introduced into a system by physical limits [11]. For example, voltage limitations of the controller output signal, or maximum and minimum positions of an inlet valve for the plant [12].

As saturation is not a linear time invariant function, regular MATLAB is incapable of modelling this system and Simulink was used instead [4]. The saturation limits of Block X are unspecified, and therefore upper and lower limits of +-0.75 were implemented for demonstrative purposes.

The following Simulink model was used for all saturation simulations:

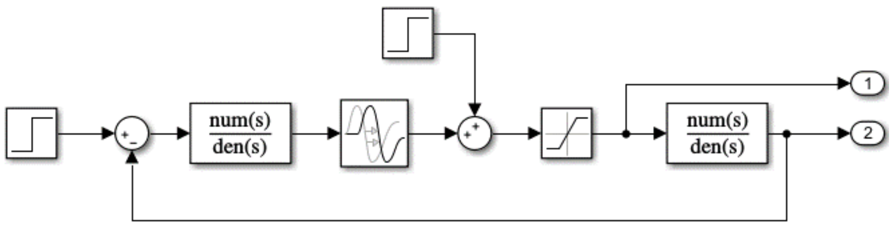


FIGURE 4: SIMULINK MODEL WITH SATURATION

2.7 IMPLEMENTING ANTI-WINDUP

Anti-windup is required to limit the impact of saturation blocks within control systems [13]. Windup appears when there is an inconsistency between the controller output and plant input. When this happens the feedback loop is effectively broken, and the system output is poor. In the case of PI and PID controllers, windup causes the integral I term to become inflated, and can lead the control signal to be outside of the saturated operating region for a long period of time. This in turn can lead to significant overshoots. Anti-windup, therefore, is the process of selectively lowering the I term to keep the controller output within the saturation limits. The basic block diagram for a controlled system with anti-windup may been seen in Figure 5:

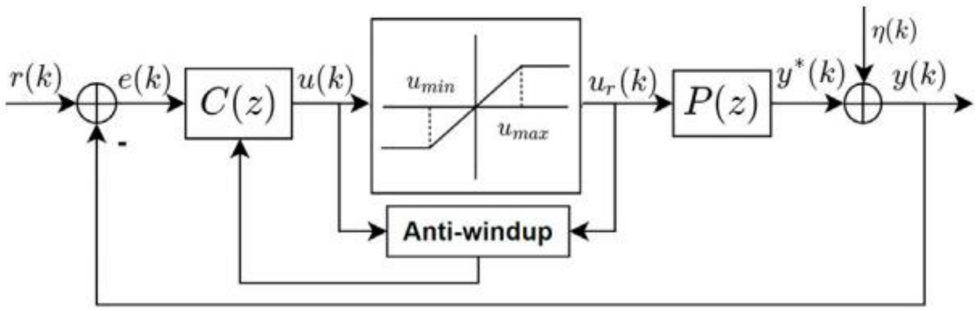


FIGURE 5: GENERAL SYSTEM WITH ANTI-WINDUP [13]

Two of the most common methods are back-calculation, and integrator clamping [13]. These methods are were implemented for the purposes of demonstration. However, it is possible to implement more sophisticated anti-windup mechanisms by using a state space approach [14].

With back calculation, the difference between the controller output, and the plant input is measured and subtracted from Ki (the integrator gain of the controller) [11]. This dynamically limits the output of the integrator, such that changes to the I term are proportional to the error of the plant input. The gain of the back-calculation feedback must be tuned to compensate for the specific limits of the saturation block. This gain is generally equal to $\frac{1}{T_t}$ where T_t is a parameter called the tracking time constant [13]. As no limits were specified for Block X, no tuning of the back-calculation gain was possible, therefore the gain was fixed at 1. The Simulink model used for back calculation may be seen in Figure 6. With reference to Figure 5, when using back calculation the new controller integral term $u_i(k)$ is given as:

$$u_i(k) = u_i(k-1) + \left[K_i e(k) + \frac{1}{T_t} e_p(k)\right] T_s$$

EQUATION 10: BACK CALCULATION EQUATION

Integrator clamping imposes limits on the output signal of the integrator directly [15]. This method does not alter the integrator gain of the controller, instead it simply limits the integrator output to stop the I term becoming too large. The integrator limits may either be fixed or dynamically adjusted. Dynamically adjusted limits tend to produce better system responses. Again, the limits for integrator clamping must be tuned to the specific system. This was not possible within this report, therefore fixed limits of +- 0.5 were selected for this investigation. The Simulink model used for integrator clamping may be seen in Figure 7.

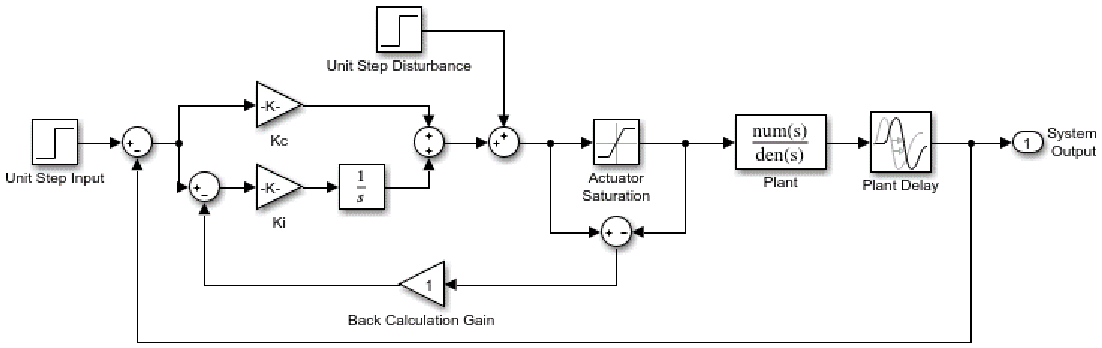


FIGURE 6: SIMULINK MODEL WITH BACK CALCULATION

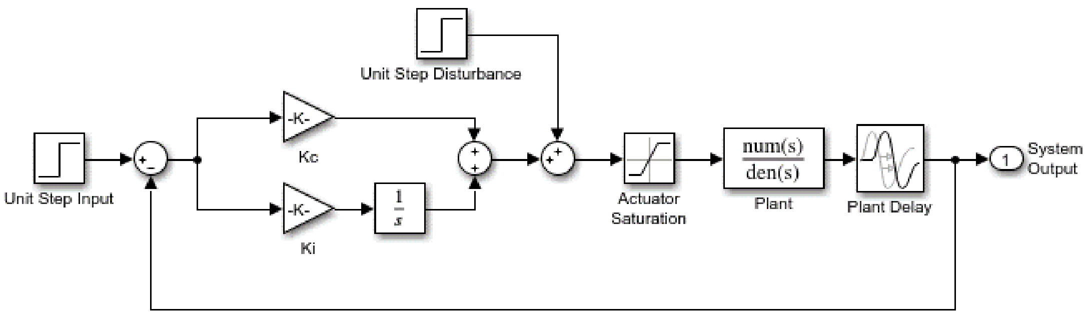


FIGURE 7: SIMULINK MODEL WITH INTEGRATOR CLAMPING

3 RESULTS AND DISCUSSION

3.1 CLOSED LOOP RESPONSE

The closed loop step responses of the three controlled systems is shown below in Figure 8:

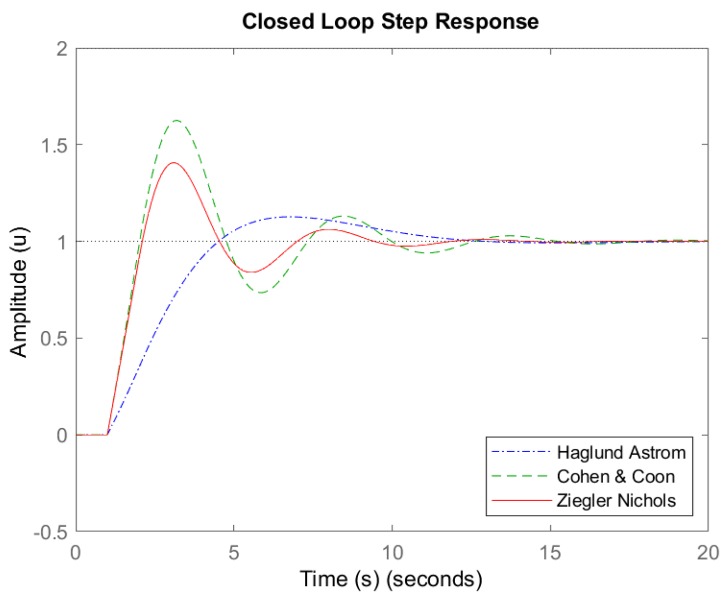


FIGURE 8: CLOSED LOOP STEP RESPONSE OF CONTROLLED SYSTEMS

Table 3 shows the step response parameters of each system:

Controller Type	Rise Time (s)	Settling Time (s)	Overshoot (%)	Peak Magnitude	Peak Time (s)
Haglund Astrom	2.5998	11.3742	12.6587	1.1266	6.7844
Cohen & Coon	0.8156	14.4034	62.4695	1.6247	3.1455
Ziegler-Nichols	0.8793	10.9390	40.6674	1.4067	3.0811

TABLE 3: STEP RESPONSE PARAMETERS OF CONTROLLED SYSTEMS

From this it can be seen that the Cohen & Coon, and Ziegler Nichols controllers had very similar rise times and peak times. However, the rise time of the Haglund Astrom controller was over three times longer than the other two. The Haglund Astrom controller also had a peak time over two times longer than the others. However, the Haglund Astrom controller also had an overshoot percentage approximately 3 times smaller than the Ziegler Nichols controller, and 5 times smaller than the Cohen & Coon controller.

In all applications a balance must be struck between rise time and overshoot percentage. However, typically overshoot percentages of 40%+ would be considered unreasonable. As such, the Haglund Astrom controlled system may be said to have the best step response characteristics.

Controller Type	Gain Margin (dB)	Phase Margin (Deg)	Gain Margin Frequency (Radians)	Phase Margin Frequency (Radians)
Haglund Astrom	12.8	53.9709	1.4507	0.4046
Cohen & Coon	3.88	26.1840	1.4654	0.9682
Ziegler-Nichols	4.74	38.0727	1.5734	0.9105

TABLE 4: GAIN AND PHASE MARGIN PARAMETERS OF CONTROLLED SYSTEMS

Table 4 shows the gain and phase margin characteristics for each controlled system. The gain and phase margins of all systems are positive, and within reasonable ranges, demonstrating that all systems are stable. However, the margins for the Haglund Astrom controller are significantly larger than the Cohen & Coon and Ziegler Nichols. This, coupled with the substantially lower overshoot percentage, means it is reasonable to conclude that the Haglund Astrom controlled system is the most conservative.

3.2 DISTURBANCE RESPONSE

All three controllers were tested for their disturbance rejection capacities using a unit step disturbance at 40 seconds. Figure 9 shows the step and disturbance responses of all systems:

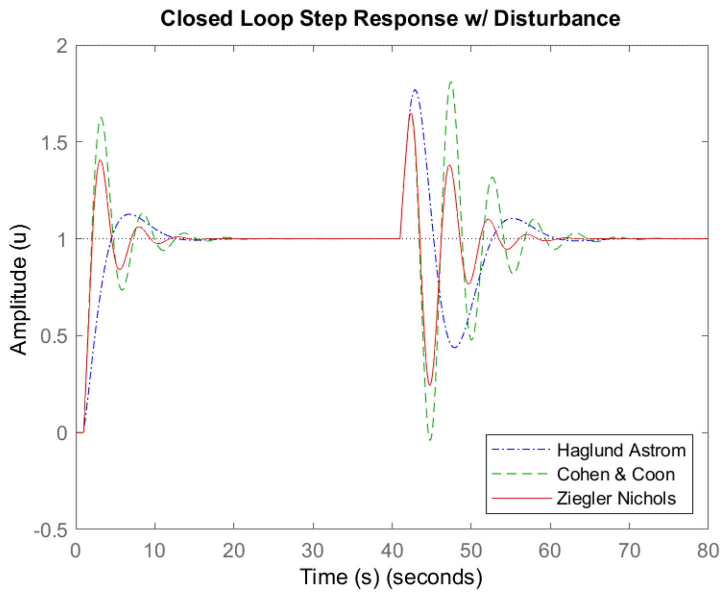


FIGURE 9: STEP AND DISTRUBANCE RESPONSE OF CONTROLLED SYSTEMS

Controller Type	Overshoot (%)	Settling Time (s)	Peak Magnitude	Peak Time (s)
Haglund Astrom	76.8347	19.7777	1.7683	2.8908
Cohen & Coon	81.0295	23.8242	1.8103	7.4216
Ziegler-Nichols	64.9262	17.6672	1.6493	2.3705

TABLE 5: DISTURBANCE RESPONSE PARAMETERS OF CONTROLLLED SYSTEMS

From the disturbance response parameters shown in Table 5, it can be seen that the Ziegler Nichols controller had the smallest overshoot percentage and settling time. The Cohen & Coon controller had the largest overshoot and longest settling time. The peak time for the Cohen & Coon controller was over twice the length of the others, owed to the fact that the peak of the Cohen & Coon system's response was seen in the first oscillation, rather than in the initial rise. For these reasons it may be said neither the Haglund Astrom or Cohen & Coon controllers performed better than the Ziegler Nichols baseline. This is in contrast to the step response characteristics, where the Haglund Astrom controller fared better. As such, consideration must be given as to whether step response, or disturbance response characteristics are more important for the given system.

3.3 IMPACT OF BLOCK X

Figure 10 shows the impact that including the saturation block has had on the signal entering the plant. As can be seen, the signals from the controllers have been limited to a peak magnitude of 0.75 . This has a significant impact on the step and disturbance responses of all controllers, as shown in Figure 11.

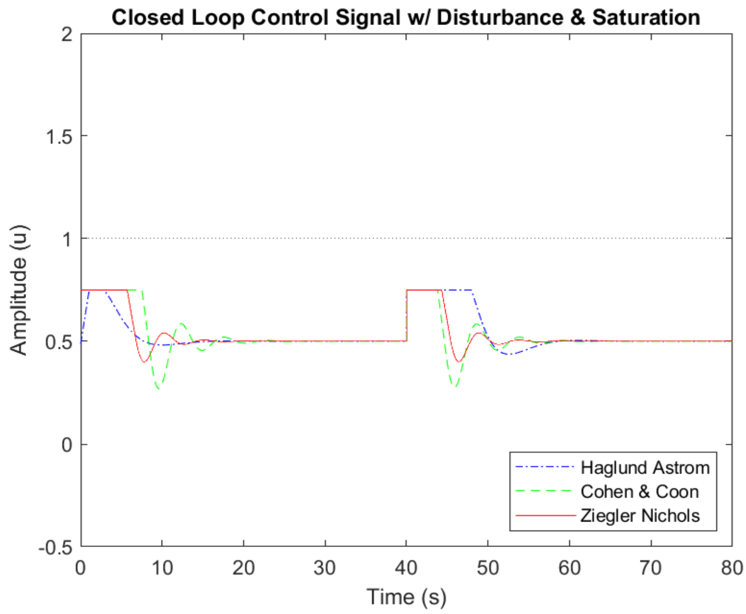


FIGURE 10: SIGNAL INTO PLANT OF CONTROLLED SYSTEMS WITH SATURATION

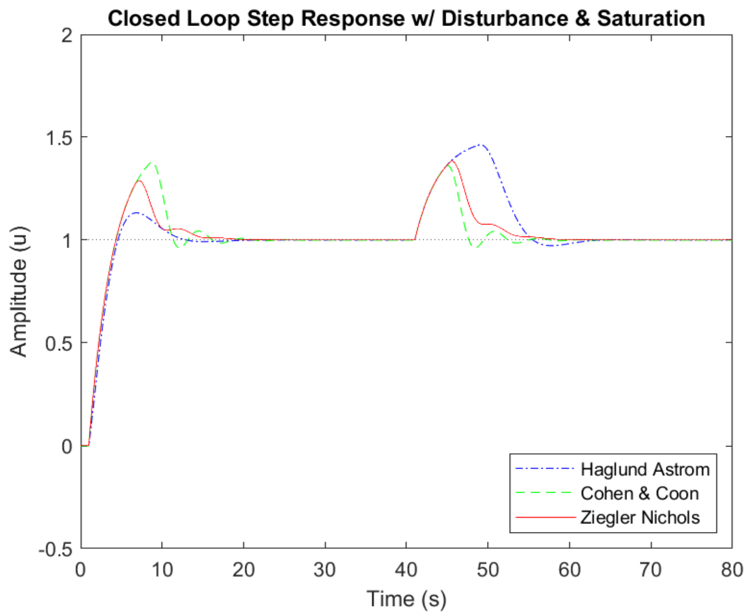


FIGURE 11: STEP AND DISTURBANCE RESPONSE OF CONTROLLED SYSTEMS WITH SATURATION

Tables 6 and 7 show the step response, and disturbance response parameters of the controlled systems with saturation:

Controller Type	Rise Time (s)	Settling Time (s)	Overshoot (%)	Peak Magnitude	Peak Time (s)
Haglund Astrom	2.6726	11.4904	13.2359	1.1324	6.8430
Cohen & Coon	2.5632	15.5734	37.8740	1.3787	8.7290
Ziegler-Nichols	2.5632	14.2393	28.9246	1.2892	7.1740

TABLE 6: STEP RESPONSE PARAMETERS OF CONTROLLED SYSTEMS WITH SATURATION

Controller Type	Overshoot (%)	Settling Time (s)	Peak Magnitude	Peak Time (s)
Haglund Astrom	46.3516	21.3232	1.4635	9.0950
Cohen & Coon	36.2616	16.4528	1.3626	5.0380
Ziegler-Nichols	38.3995	17.2944	1.3840	5.5680

TABLE 7: DISTURBANCE RESPONSE PARAMETERS OF CONTROLLED SYSTEMS WITH SATURATION

With saturation at +- 0.75, step response parameters for the Haglund Astrom system have not changed beyond a very slight increase in overshoot percentage and peak time. This is owed to the little clipping of the control signal as seen in Figure 10. However, the step response parameters of the other systems changed drastically. Overshoot percentages for the Cohen & Coon and Ziegler Nichols controlled systems have decreased substantially. The rise times of these systems have also increased as have settling times and peak times.

The disturbance response characteristics of all systems have also changed substantially. Overshoot percentages for all systems have nearly halved. However, peak times for the Haglund Astrom and Ziegler Nichols controllers have increased dramatically. Settling times for the Haglund Astrom and Ziegler Nichols controllers have not been effected to a great extent, however the Cohen & Coon system saw a reduction in settling time of approximately 7.4 seconds.

However, it is important to note that these results can only be used for demonstrative purposes. As no limits have been specified for the real saturation block (Block X).

3.4 ANTI-WINDUP

Figure 12 shows the changes to step and disturbance response when anti-windup techniques were implemented on the Haglund Astrom controlled system. The specific response parameters are given in Tables 8 and 9. It can be seen that integrator clamping at +-0.5 has resulted in far better step response characteristics, but has not substantially affected disturbance response. Whereas back calculation with a gain of 1 has not significantly effected step response, but has improved disturbance rejection.

However, it is important to note that the values for back calculation gain, and integrator clamping limits included in this report are for demonstrative purposes only. In practice, these values must be tuned to the specific parameters of the saturation block in question.

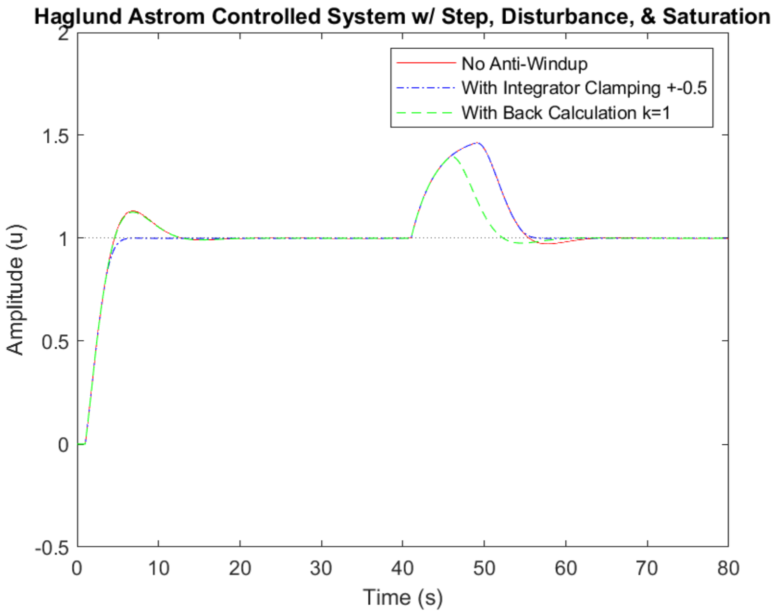


FIGURE 12: EFFECTS OF ANTI-WINDUP ON THE HAGLUND ASTROM CONTROLLED SYSTEM

Controller Type	Rise Time (s)	Settling Time (s)	Overshoot (%)	Peak Magnitude	Peak Time (s)
No Anti-windup	2.6726	11.4907	13.2312	1.1324	6.8430
Integrator Clamped	2.7917	5.1472	0.0689	1.0007	7.1360
Back Calculation	2.6770	11.4614	12.6566	1.1266	6.8620

TABLE 8: STEP RESPONSE PARAMETERS OF HAGLUND ASTROM CONTROLLED SYSTEM WITH AND WITHOUT ANTI-WINDUP

Controller Type	Overshoot (%)	Settling Time (s)	Peak Magnitude	Peak Time (s)
No Anti-windup	46.3521	21.3242	1.4635	9.0950
Integrator Clamped	46.3474	15.5670	1.4635	9.0950
Back Calculation	39.6327	18.1555	1.3963	6.0100

TABLE 9: DISTURBANCE RESPONSE PARAMETERS OF HAGLUND ASTROM CONTROLLED SYSTEM WITH AND WITHOUT ANTI-WINDU

4 CONCLUSIONS

This report compared Haglund Astrom, Cohen & Coon, and Ziegler Nichols tuning methods for a PI controller within a chemical plant. These systems were compared in terms of gain and phase margin, step response, and disturbance response parameters. From this, it is clear that the Haglund Astrom system has the best step response, and the Ziegler Nichols system has the best disturbance rejection. The Haglund Astrom controlled system also had the largest gain and phase margins. Consideration must be given to the overall likelihood and magnitude of disturbance within the system. Unless significant and frequent disturbances are likely then the Haglund Astrom controller is the better choice for the given plant.

The impact of a saturation block (Block X) was also investigated, as were common anti-windup methods which compensate for saturation. It is concluded that a saturation block could substantially change step and disturbance responses for all controlled systems. However, appropriately tuned back calculation or integrator clamping can reduce this, and improve overall system performance. It was observed that back calculation and integrator clamping have different effects on step and disturbance response. As such, further investigation would be necessary to determine which controller + anti-windup technique pairing would be most appropriate for the system. As no saturation limits were given for Block X, anti-windup tuning and controller + anti-windup technique pairing were considered outside the scope of this report.

5 REFERENCES

[1] A. O'Dwyer, “A SUMMARY OF PI AND PID CONTROLLER TUNING RULES FOR PROCESS WITH TIME DELAY. PART 1: PI CONTROLLER TUNING RULES,” Terrassa, Spain, 2000.

[2] MathWorks, “MATLAB: About,” 2021. [Online]. Available: https://uk.mathworks.com/products/matlab.html.

[3] MathWorks, “Simulink: About,” 2020. [Online]. Available: https://uk.mathworks.com/products/simulink.html.

[4] MathWorks, “Saturation,” 2020. [Online]. Available: https://uk.mathworks.com/products/simulink.html.

[5] MathWorks, “Smoothdata,” 2020. [Online]. Available: https://uk.mathworks.com/help/matlab/ref/smoothdata.html.

[6] M. Idi, “Proportional Integral (PI) Controller for a Process Plant System,” vol. 4, no. 1, 2015.

[7] j. M. R. P. Vladimir Boba, “TUNING OF DIGITAL PID CONTROLLERS BASED ON ZIEGLER - NICHOLS METHOD,” 1997.

[8] MathWorks, “Gradient,” [Online]. Available: https://uk.mathworks.com/help/matlab/ref/gradient.html.

[9] MathWorks, “stepplot,” 2020. [Online]. Available: https://uk.mathworks.com/help/control/ref/stepplot.html.

[10] MathWorks, “feedback,” [Online]. Available: https://uk.mathworks.com/help/control/ref/feedback.html.

[11] Sang-HoonKim, “Chapter 2 - Control of direct current motors,” in *Electric Motor Control: DC, AC, and BLDC Motors*, 2017.

[12] T. Hu and Z. Lin, Control systems with actuator saturation: analysis and design, Birkhauser, 2001.

[13] L. d. Silva, R. C. Flesch and J. E. Normey-Rico, “Analysis of Anti-windup Techniques in PID Control of Processes with Measurement Noise,” vol. 51, no. 4, 2018.

[14] L. Wang, “Control design methods with antiwindup mechanism for disturbance rejection and reference following,” *Transactions of the Institute ofMeasurement and Control,* vol. 38, no. 6, pp. 625-638, 2016.

[15] X.-l. Li, J.-G. Park and H.-B. Shin, “Comparison and Evaluation of Anti-Windup PI Controllers,” vol. 11, no. 1, 2011.

APENDIX

A. MATLAB SCRIPT

```
% This matlab script takes open-loop plant step response data,
smooths
% it, derives a transfer function, designs 3 different PI
controllers
% for the system, and tests the closed-looop step response of the
new
% controlled systems.
%
% The controller tuning methods investigated are the Haglund Astrom,
% Cohen & Coon, and Ziegler Nichols.
%
% The systems are tested using a unit step at time 0s,
% and a unit step with a following unit disturbance at time 40s.
%
% Written by Luke Byrne at Glasgow Caledonian University 23/12/2020


% Wipe workspace
clear

% Import the given open loop plant step response data,
% and acompanying timestamps.
% Data files must be in the current working directory.
StepResponse = importdata('response.dat');
RealTime = importdata('time.dat');

% Smooth data, to make it easier to derive the transfer function.
% Gaussian Smoothing with two windows were used. A wide window to
find
% the maximum amplitude K, and a short window to find the break
point Tao.
% Changing these window sizes will change the values for K, Tao and
Theta
% used throughout the rest of the analysis
WideWindowSize=100;
ShortWindowsSize=20;
StepResponseHeavilySmoothed = smoothdata(StepResponse,'gaussian',
WideWindowSize);
StepResponseLightlySmoothed = smoothdata(StepResponse,'gaussian',
ShortWindowsSize);

% Imported timestamps have irregular intervals.
% Create new array with same length, but in regular intervals.
IdealTime=[0:0.1:20];

% Plot graph of plant step response data.
```

```
p=plot(RealTime, StepResponse, 'b', RealTime,
StepResponseHeavilySmoothed, 'r-.', RealTime,
StepResponseLightlySmoothed, 'g--');
 yline(2, ':')
 yline(1, ':')
 p(1).LineWidth = 1;
 p(2).LineWidth = 2;
 p(3).LineWidth = 2;
 legend('Unsmoothed', strcat('Gaussian Smoothed Window=',
num2str(WideWindowSize)), strcat('Gaussian Smoothed Window=',
num2str(ShortWindowsSize)), 'Location', 'southeast')
 title('Plant Step Response');
 xlabel('Time (s)');
 ylabel('Amplitude (u)');


% Find K, Theta, and Tao from smoothed data.

% K = The final value of the heavily smoothed data.
K= StepResponseHeavilySmoothed(size(StepResponseHeavilySmoothed, 1))

% Theta is the point in time when the second derivative of the plant
step
% response is largest
% For this the lightly smoothed data is used
StepResponseSecondDerivative=gradient(gradient(StepResponseLightlySm
oothed));
[StepResponseSecondDerivativeMax,
StepResponseSecondDerivativeIndexAtMax] =
max(double(StepResponseSecondDerivative));
Theta=RealTime(StepResponseSecondDerivativeIndexAtMax)

% Tao equals the point in time at whch the plant step response
% is equal to 0.63 times K, minus Theta
[val,idx]=min(abs(StepResponseHeavilySmoothed-K*0.63));
Tao=RealTime(idx)-Theta


% Dreive open loop plant transfer function, and step response.
PlantTF=tf(K, [Tao 1],'InputDelay', 1)
PlantTFStepResponse=step(PlantTF, IdealTime);

% Plot plant transfer function on top of noisy data.
plot(RealTime, StepResponse,'b')
 hold on
  plot(IdealTime, PlantTFStepResponse,'r-.', 'LineWidth', 1.5)
  yline(2, ':')
  yline(1, ':')
  legend('Plant Step Response','Derived Transfer Function',
'Location', 'southeast')
  title('Plant Step Response w/ Transfer Function');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
```

```
 hold off

% Find values of K_c, and Tao_i,
% and calculate Haglund Astrom controller transfer function.
K_c = (0.14/K)+(0.28*Tao)/(Theta*K)
Tao_i= (0.33*Theta)+(6.8*Theta*Tao)/(10*Theta+Tao)
K_i = K_c/Tao_i
HaglundAstromControllerTF=K_c+(tf(K_c, [Tao_i 0]))

% Find values of K_c, Tao_i,
% and calculate Cohen & Coon controller transfer function
K_c=(1/K)*(Tao/Theta)*(0.9+Theta/(12*Tao))
Tao_i=Theta*(30+3*(Theta/Tao))/(9+20*(Theta/Tao))
K_i = K_c/Tao_i
CohenAndCoonControllerTF=K_c+(tf(K_c, [Tao_i 0]))


% Find the maximum gradient of the open loop plant transfer function
step response
PlantTFGradient=gradient(PlantTFStepResponse)*10;
[PlantTFMaxGradient, PlantTFIndexAtMax] =
max(double(PlantTFGradient));
PlantTFMaxGradient
PlantTFMaxGradientTime=IdealTime(PlantTFIndexAtMax)
PlantTFMaxGradientAmplitude=PlantTFStepResponse(PlantTFIndexAtMax)

% Plot open loop plant transfer function with line of maximum
gradient.
plot(IdealTime,PlantTFStepResponse, 'LineWidth', 1)
 hold on
  plot (IdealTime,
PlantTFMaxGradientAmplitude+PlantTFMaxGradient*(IdealTime-
PlantTFMaxGradientTime),'--',  'LineWidth', 1)
  yline(2, ':')
  yline(1, ':')
  ylim([-0.5 2.5])
  legend('Plant Transfer Function', 'Maximum Gradient', 'Location',
'southeast')
  title('Plant Transfer Function Step Response w/ Maximum
Gradient');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
 hold off

% Using maxium gradient, find L, R_n, K_c, and Tao_i
% and calculate Ziegler Nichols Controller transer function.
L=Theta
R_n=PlantTFMaxGradient
K_c=0.9/(R_n*L)
Tao_i=3*L
K_i = K_c/Tao_i
ZieglerNicholsControllerTF=K_c+(tf(K_c, [Tao_i 0]))
```

```
% Calculate closed loop system transfer functions for all 3
controllers.
ClosedLoopHaglundAstrom =
feedback(HaglundAstromControllerTF*PlantTF,1);
ClosedLoopCohenAndCoon =
feedback(CohenAndCoonControllerTF*PlantTF,1);
ClosedLoopZieglerNichols =
feedback(ZieglerNicholsControllerTF*PlantTF,1);

% Calculate gain and phase margins for each controlled system
[HAGm,HAPm,HAWcg,HAWcp] = margin(HaglundAstromControllerTF*PlantTF)
margin(HaglundAstromControllerTF*PlantTF)
[CCGm,CCPm,CCWcg,CCWcp] = margin(CohenAndCoonControllerTF*PlantTF)
margin(CohenAndCoonControllerTF*PlantTF)
[ZNGm,ZNPm,ZNWcg,ZNWcp] = margin(ZieglerNicholsControllerTF*PlantTF)
margin(ZieglerNicholsControllerTF*PlantTF)

% Calculate step info for all controllers
ClosedLoopHaglundAstromInfo=stepinfo(ClosedLoopHaglundAstrom)
ClosedLoopCohenAndCoonInfo=stepinfo(ClosedLoopCohenAndCoon)
ClosedLoopZieglerNicholsInfo=stepinfo(ClosedLoopZieglerNichols)

% Plot 3 controllers closed loop unit step response.
stepplot(IdealTime, ClosedLoopHaglundAstrom, 'b-.')
 hold on
  stepplot(IdealTime, ClosedLoopCohenAndCoon, 'g--')
  stepplot(IdealTime, ClosedLoopZieglerNichols, 'r')
  yline(2, ':')
  yline(1, ':')
  ylim([-0.5 2])
  legend('Haglund Astrom', 'Cohen & Coon', 'Ziegler Nichols',
'Location', 'southeast')
  title('Closed Loop Step Response');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
 hold off
```

```
% Create a unit step disturbance of magnatude M at time t.
% Change value of M to see system response to different
% amounts of disturbance
M=1;
t=40;
DisturbanceTF=tf(M,'InputDelay', t);

% Calculate closed loop system transfer fucntions for all 3
controllers,
% with a unit step disturbance introduced
ClosedLoopHaglundAstromDisturbance =
feedback((HaglundAstromControllerTF+DisturbanceTF)*PlantTF,1);
ClosedLoopCohenAndCoonDisturbance =
feedback((CohenAndCoonControllerTF+DisturbanceTF)*PlantTF,1);
```

```
ClosedLoopZieglerNicholsDisturbance =
feedback((ZieglerNicholsControllerTF+DisturbanceTF)*PlantTF,1);

% Plot closed loop system responses, with disturbance
stepplot([0:0.1:80], ClosedLoopHaglundAstromDisturbance, 'b-.')
 hold on
  stepplot([0:0.1:80], ClosedLoopCohenAndCoonDisturbance, 'g--')
  stepplot([0:0.1:80], ClosedLoopZieglerNicholsDisturbance, 'r')
  yline(2, ':')
  yline(1, ':')
  ylim([-0.5 2])
  legend('Haglund Astrom', 'Cohen & Coon', 'Ziegler Nichols',
'Location', 'southeast')
  title('Closed Loop Step Response w/ Disturbance');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
 hold off

% Cut off the first 40s of the closed loop responses with
disturbances,
% and pass the result to a stepinfo() function to get the rise time
etc
% for the disturbances only.
% A resolution of 0.0001s in needed to prevent rounding errors
a=step(ClosedLoopHaglundAstromDisturbance, [0:0.0001:80]);
b=a([t*10000:800000]);
HaglundAstromDisturbanceInfo=stepinfo(b, [0:0.0001:40], 1)

a=step(ClosedLoopCohenAndCoonDisturbance, [0:0.0001:80]);
b=a([t*10000:800000]);
CohenAndCoonDisturbanceInfo=stepinfo(b, [0:0.0001:40], 1)

a=step(ClosedLoopZieglerNicholsDisturbance, [0:0.0001:80]);
b=a([t*10000:800000]);
ZieglerNicholsDisturbanceInfo=stepinfo(b, [0:0.0001:40], 1)
```

```
% Open and Run Simulink systems
% System files must be inside the current working directory
directory
Saturation=sim("Saturation.slx");
AntiWindup=sim("AntiWindup.slx");
```

```
% Plot Simulink saturation block outputs
plot(Saturation.HaglundAstrom, 'b-.')
 hold on
  plot(Saturation.CohenAndCoon, 'g--')
  plot(Saturation.ZieglerNichols, 'r')
  yline(2, ':')
  yline(1, ':')
  ylim([-0.5 2])
  legend('Haglund Astrom', 'Cohen & Coon', 'Ziegler Nichols',
'Location', 'southeast')
```

```
  title('Closed Loop Control Signal w/ Disturbance & Saturation');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
 hold off

% Plot SIMULINK Saturated system outputs
plot(Saturation.SystemHaglundAstrom, 'b-.')
 hold on
  plot(Saturation.SystemCohenAndCoon, 'g--')
  plot(Saturation.SystemZieglerNichols, 'r')
  yline(2, ':')
  yline(1, ':')
  ylim([-0.5 2])
  legend('Haglund Astrom', 'Cohen & Coon', 'Ziegler Nichols',
'Location', 'southeast')
  title('Closed Loop Step Response w/ Disturbance & Saturation');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
 hold off


% Extract data from simulink saturation system outputs,
% for use when extracting step info
SaturationHaglundAstrom=Saturation.HaglundAstrom.Data;
SaturationCohenAndCoon=Saturation.CohenAndCoon.Data;
SaturationZieglerNichols=Saturation.ZieglerNichols.Data;

SystemHaglundAstrom=Saturation.SystemHaglundAstrom.Data;
SystemCohenAndCoon=Saturation.SystemCohenAndCoon.Data;
SystemZieglerNichols=Saturation.SystemZieglerNichols.Data;

% Split Simulink saturation output data into only the step,
% and only the disturbance for use in stepinfo() function
HaglundAstromWithSaturationStep=SystemHaglundAstrom([1:40001]);
HaglundAstromWithSaturationDisturbance=SystemHaglundAstrom([40001:80
001]);

CohenAndCoonWithSaturationStep=SystemCohenAndCoon([1:40001]);
CohenAndCoonWithSaturationDisturbance=SystemCohenAndCoon([40001:8000
1]);

ZieglerNicholsWithSaturationStep=SystemZieglerNichols([1:40001]);
ZieglerNicholsWithSaturationDisturbance=SystemZieglerNichols([40001:
80001]);


% Get step info for saturation outputs
HaglundAstromWithSaturationStepInfo=stepinfo(HaglundAstromWithSatura
tionStep, [0:0.001:40], 1)
CohenWithSaturationStepInfo=stepinfo(CohenAndCoonWithSaturationStep,
[0:0.001:40], 1)
ZieglerNicholsWithSaturationStepInfo=stepinfo(ZieglerNicholsWithSatu
rationStep, [0:0.001:40], 1)
```

```
HaglundAstromWithSaturationDisturbanceInfo=stepinfo(HaglundAstromWit
hSaturationDisturbance, [0:0.001:40], 1)
CohenAndCoonWithSaturationDisturbanceInfo=stepinfo(CohenAndCoonWithS
aturationDisturbance, [0:0.001:40], 1)
ZieglerNicholsWithSaturationDisturbanceInfo=stepinfo(ZieglerNicholsW
ithSaturationDisturbance, [0:0.001:40], 1)
```

```
% Plot SIMULINK anti-Windup results
plot(AntiWindup.NoAntiWindup, 'r')
 hold on
  plot(AntiWindup.IntegratorClamped, 'b-.')
  plot(AntiWindup.BackCalculation, 'g--')
  yline(2, ':')
  yline(1, ':')
  ylim([-0.5 2])
  legend('No Anti-Windup', 'With Integrator Clamping +-0.5', 'With
Back Calculation k=1')
  title('Haglund Astrom Controlled System w/ Step, Disturbance, &
Saturation');
  xlabel('Time (s)');
  ylabel('Amplitude (u)');
 hold off

% Extract data from simulink anti-windup system outputs,
% for use when extracting step info
NoAntiWindup=AntiWindup.NoAntiWindup.Data;
IntegratorClamped=AntiWindup.IntegratorClamped.Data;
BackCalculation=AntiWindup.BackCalculation.Data;

SaturationHaglundAstrom=Saturation.HaglundAstrom.Data;
SaturationCohenAndCoon=Saturation.CohenAndCoon.Data;
SaturationZieglerNichols=Saturation.ZieglerNichols.Data;

SystemHaglundAstrom=Saturation.SystemHaglundAstrom.Data;
SystemCohenAndCoon=Saturation.SystemCohenAndCoon.Data;
SystemZieglerNichols=Saturation.SystemZieglerNichols.Data;


% Split Simulink saturation output data into only the step,
% and only the disturbance for use in stepinfo() function
NoAntiWindupStep=AntiWindup.NoAntiWindup.Data([1:40001]);
NoAntiWindupDisturbance=AntiWindup.NoAntiWindup.Data([40001:80001]);

IntegratorClampedStep=AntiWindup.IntegratorClamped.Data([1:40001]);
IntegratorClampedDisturbance=AntiWindup.IntegratorClamped.Data([4000
1:80001]);

BackCalculationStep=AntiWindup.BackCalculation.Data([1:40001]);
BackCalculationDisturbance=AntiWindup.BackCalculation.Data([40001:80
001]);
```

```
% Get step info for anti-windup outputs
NoAntiWindupStepInfo=stepinfo(NoAntiWindupStep, [0:0.001:40], 1)
IntegratorClampedStepInfo=stepinfo(IntegratorClampedStep,
[0:0.001:40], 1)
BackCalculationStepInfo=stepinfo(BackCalculationStep, [0:0.001:40],
1)

NoAntiWindupDisturbanceInfo=stepinfo(NoAntiWindupDisturbance,
[0:0.001:40], 1)
IntegratorClampedDisturbanceInfo=stepinfo(IntegratorClampedDisturban
ce, [0:0.001:40], 1)
BackCalculationDisturbanceInfo=stepinfo(BackCalculationDisturbance,
[0:0.001:40], 1)

B.
```