UNIVERSITY OF WATERLOO

CHE 524 Process Control Laboratory

Final Report for Continuous Stair Tank Heater

Author: Yung Chi Daniel Cho

Group 1

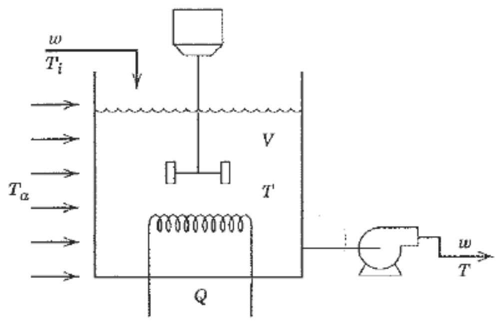
This report was written by the author under the guidance of TA Nathan Grishkewich and Course Instructor Luis A. Ricardez-Sandova. No other assistance were received.

CONTENTS

List of Figure

PERTINENT THEORY

In this experiment controller is designed for a continuous stair tank heater system. Water enters the stair tank at temperature T_i and Q amount of heat is supplied to the stair tank heater. The tank is well mixed such that water exits the tank at temperature T, which is the same as the temperature inside the tank.



A first order plus time delay model is used to model the changes of temperature T.

$$T'(t) = MK_p \times \left(1 - e^{-\frac{t-\theta}{\tau_p}}\right) \quad \textbf{(1)}$$

$$M - Amplitude\ of\ step\ change K_P - Process\ Gaint - time(s)\theta - time\ delay$$
$$\tau_p - Process\ time\ constant$$

$$T'(t) = T(t) - T(0) - Temperature\ in\ deviation\ variable$$

Experimental procedures

9 experiments were conducted in increasing and decreasing direction and at a several of temperature range as listed in the Lab Manual. The model is fitted to the experimental data to obtain K_P τ_p and θ using Sum of Square Error Minimization using Solver in Excel.

$$SSE = \sum_{t=0}^{N} (T'_{fitted}(t) - T'(t))^2$$

System Identification And Modelling (1st Session)

Result For First Session

The raw data of the experiment and their fitted curve is included in Appendix 1. Process parameters from each experiment are summarized in Table 1.

TABLE 1: PROCESS PARAMETERS FOR OPEN LOOP RESPONSE

Change	Mid Temp	Direction	M	KP	taud	taup	SSE	Comment
20 to 40	30	Increasing	20	0.42	2.23	327	22.9	
40 to 60	50	Increasing	20	0.63	3.12	288	30.9	
60 to 80	70	Increasing	20	0.36	0.00	225	21.1	Only consider the first 400 data points
80 to 60	70	Decreasing	-20	0.19	0.97	271	15.4	Only consider the first 400 data points
60 to 40	50	Decreasing	-20	0.50	9.97	289	26.3	
40 to 20	30	Decreasing	-20	0.35	6.02	303	14.9	
20 to 40	30	Increasing	20	0.35	0.00	300	18.9	
40 to 20	30	Decreasing	-20	0.35	1.00	316	15.6	
20 to 50	35	Increasing	30	0.45	4.01	294	23.2	

Linearity

Figure 1 shows all the fitted curves for open loop response for the system. We see that for step change in inlet temperature of 20, the steady state gains are not consistent. Comparing steady state gain when step change is positive, step change from 20 to 40,60 to 80 and 20 to 40 (repeated) have steady state gain of 6, while step change from 40 to 60 have steady state gain of 11. Comparing steady state gain when step change is negative, no common response is observed even though the magnitude of the step change is the same.

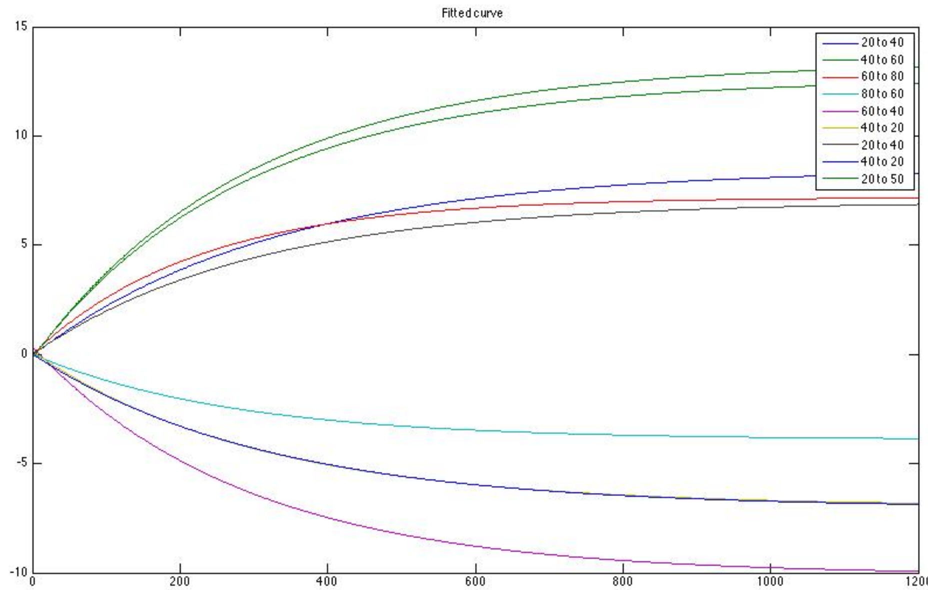
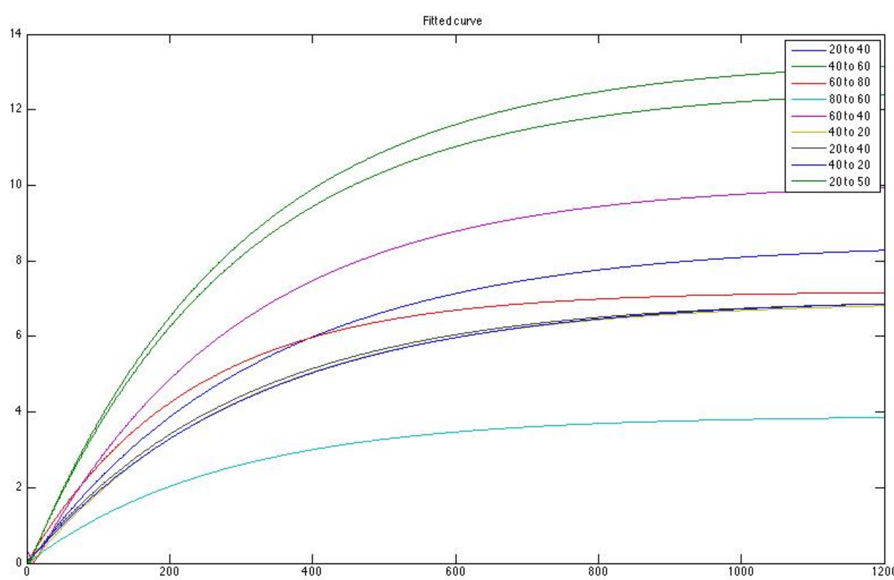


FIGURE 1: FITTED CURVE FOR OPEN LOOP RESPONSE

Figure 2 shows the absolute value of open loop response. This plot allows us to compare experiments with the same magnitude but opposite sign. From Figure 2, the magnitude of the steady state gain for experiment from 80 to 60 is less than 4, while for 60 to 80 it is about 6.5. Similarity, the magnitude of the steady state gain for experiment from 40 to 60 is 12 while for 60 to 40 is 10. The magnitude of the steady state gain for the same temperature range is 3 degree higher when the step is positive.



However the step change from 20 to 40 and 40 to 20 is linear. The magnitude of the steady state gain is about 6.5 for both directions. However, any step change other than 20 to 40 is not linear because different direction gives different steady state.

The nonlinearity of the system will make creating a controller significantly harder. We will have to average the result to get generalized system parameters. However, we will not be able to know how well the system will perform using the averaged result because of nonlinearity. This means that the generalized parameter only gives us an initial guess when developing a control system. Onsite tuning is required to optimize the performance of the controller.

We calculate the averaged process parameter and show it in Table 2. The average results will be used as the basis to design initial PI and PID controllers.

TABLE 2 : SUMMARY OF AVERAGED PROCESS PARAMETERS

	KP	taud	taup
Mean	0.40	3.04	290.38
STDEV.P	0.12	3.27	29.25
Confidence CI	0.11	3.00	26.82
Max (pred)	0.51	6.03	317.20
Min (pred)	0.29	0.04	263.55

Confidence interval is made with two sided t-distribution with alpha =0.025.

The result shows that the controller parameters (KP, taud, taup) for CSTH will lies between (0.29,0.51), (0,6), (264,317), respectively.

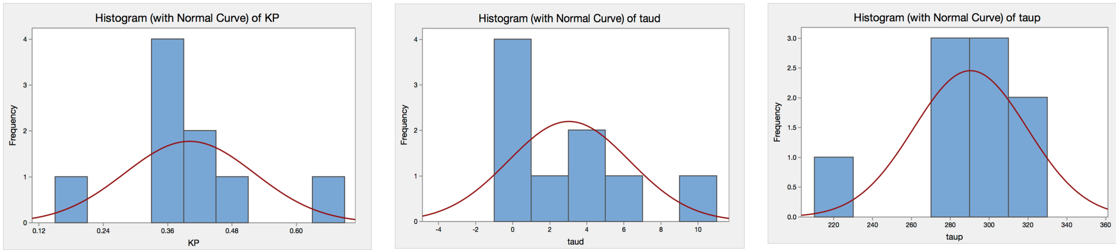
System Identification Analysis

Because the system is nonlinear, to design a good controller we will have to consider the temperature range and other operation factors.

We are interested in seeing how controller parameters distributed over the range, and how it varies with the operating temperature and direction. This allows us to predict how process will behave when the operating temperature is different from what we tested in System Identification.

Using a statistic tool Minitab Express, we can see the distribution of the controller parameters over its full range.

FIGURE 2 : HISTOGRAM OF KP, TAUP, AND THETA



The histogram shows that the data is hardly normal. To obtain better statistical soundness, it is recommended to increase the sample size and perform more experiments. However, in real world practices, such statistical soundness need to be balanced with time and cost consideration because engineering projects are time and cost sensitive.

Effect of Direction

To investigate the effect of direction of step change on controller parameters, we compare the data grouped with direction.

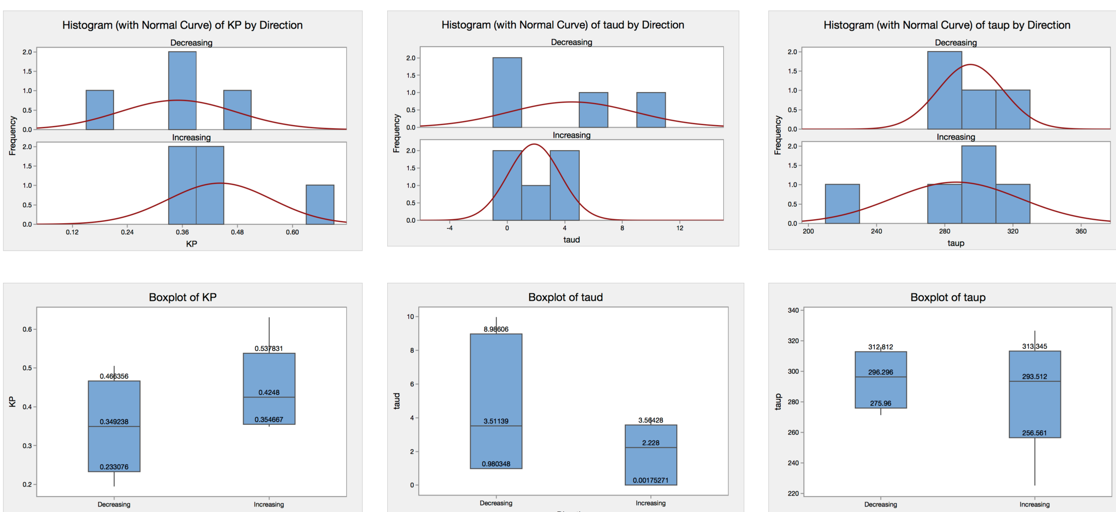


FIGURE 3 BOXPLOT OF KP THETA TAUP GROUPED BY DIRECTION OF THE STEP CHANGE

Figure 3 shows the box plot of Kp theta taup grouped by direction of the step change. Kp tends to be higher when system in increasing in temperature, which means the system have higher initial change when temperature is increasing than decreasing. Taud are similar on average regardless of direction. However, taud is more volatile and less predictable when decreasing temperature (higher variation). Taup are similar on average regardless of direction. But taup is less predictable when increasing temperature.

Effect of Temperature

To investigate how controller parameter varies with temperature, we regress each parameter with the midpoint of the step change.

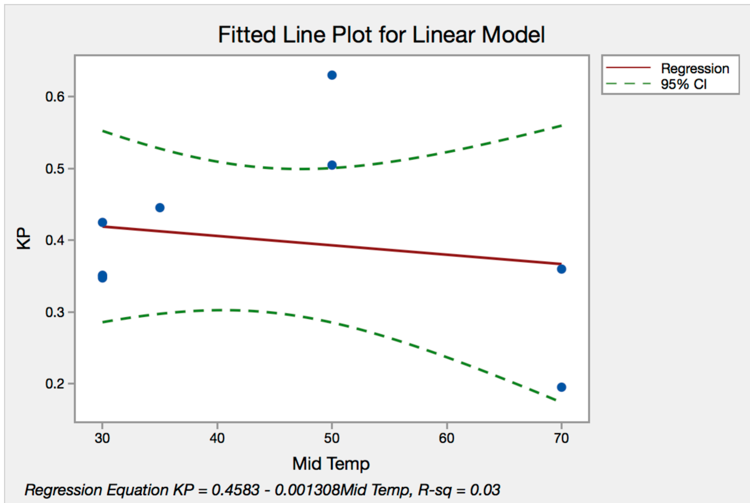


FIGURE 4: KP REGRESSION WITH TEMPERATURE

Kp fit poorly with operating temperature because the exceptionally high Kp at MidTemp =50. However, note that only the first 400 data points are used to solve for Kp at MidTemp =70, while MidTemp uses the full range of data. We might see a different correlation with temperature if we modify the data range for MidTemp = 50 as well.

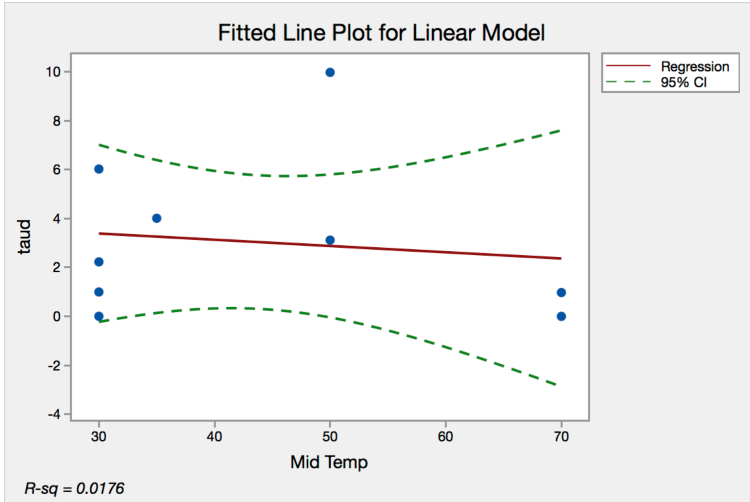


FIGURE 5: THETA REGRESSION WITH TEMPERATURE

Taud fit poorly with operating temperature as well. Data appear erratic and we concluded that Taud is not correlated with operating temperature.

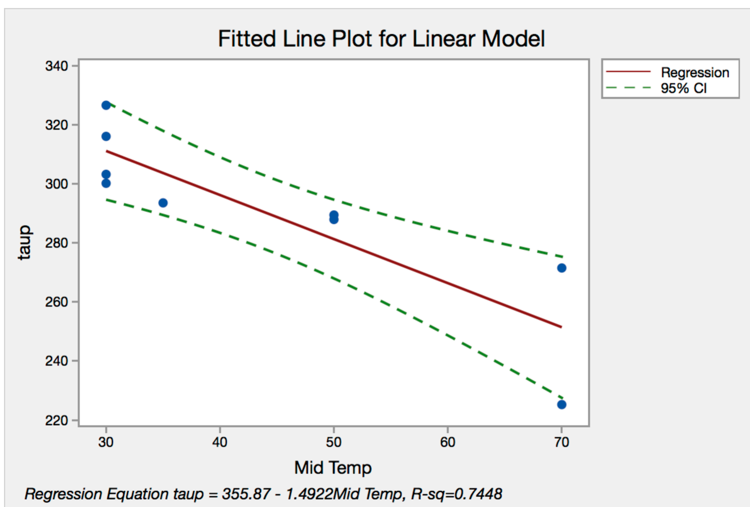


FIGURE 6 : TAUP REGRESSION WITH TEMPERATURE

Taup fits much better with operating temperature, showing a negative relationship as temperature goes up. This means that the system reach steady state faster when temperature is higher. When tuning controller we have to keep this in mind and select an appropriate taup for the operating temperature.

System Identification Conclusion

The proposed controller parameter set is the mean of experiment, where Kp, taud, taup equals, 0.4, 3, 290 respectively. We concluded that the direction of step change does not affect the parameters significantly in long term because parameter averages grouped by direction are close to each other. It is expected that the parameters Kp, taud, taup if adjusted will lie between (0.29,0.51), (0,6), (264,317) respectively. If the system operates at a low end of the temperature, taup should increase, and higher temperature, lower taup.

Designing PI And PID Controller Using Internal Model Control

Advantage and limitation

By assuming a process model is known, IMC can be used to obtain analytical expression of PI and PID tuning relations. The tuning method is well researched and the resulted tuning parameters are analytically stable and robust. However, in practice we often don't develop the process model from first principles. Instead, a first order model is assumed to approximate higher order system behaviours. This assumption simplifies system identification for complex systems, but lose the analytical certainty given by IMC.

Designing control parameter using IMC

Choosing τ_c

Using guideline published by Riera et al, 1986

$$\tau_c > 0.8\theta \;\; \tau_c > 0.1\tau_p$$

Substitute values of θ and $\tau_p \tau_c > 2.4 \;\; \tau_c > 29$

We will use $\tau_c = 29$

From table 12.1 in Seborg Case G, the PI and PID controller setting is as follow

PI Controller

$$K_c \mathrm{K_p} = \frac{\tau}{\tau_c + \theta} = \frac{290}{29 + 3} = 9.06$$

$$K_c = \frac{9.06}{K_p} = 22.65$$

$$\tau_I = \tau = 290$$

Sensitivity Analysis

Knowing that nonlinearity gives us a range of process parameters, we want to see how the variation in the process parameters affects the controller parameters using IMC. In sensitivity analysis, controller parameters are re-calculated with one process parameters in its extreme value while the rest are in their average value. The process is repeated for all extreme values and the effect of each extreme value can be compare in a Tornado Diagram.

Sensitivity Analysis for PI controller

We know that the range of theta and taup is (0,6), (264,317). We are interested to see how the variation of system parameters affects the tuning parameters. We performed 4 calculations each contains one extreme value while keeping the other parameters at average.

Kc and tauI, we calculate the following cases

Case	Theta	taup
1	6	290
2	0	290
3	3	264
4	3	317

The result is summarized in the chart below. We see that the variation of θ have a greater impact to K_c value than τ_p.

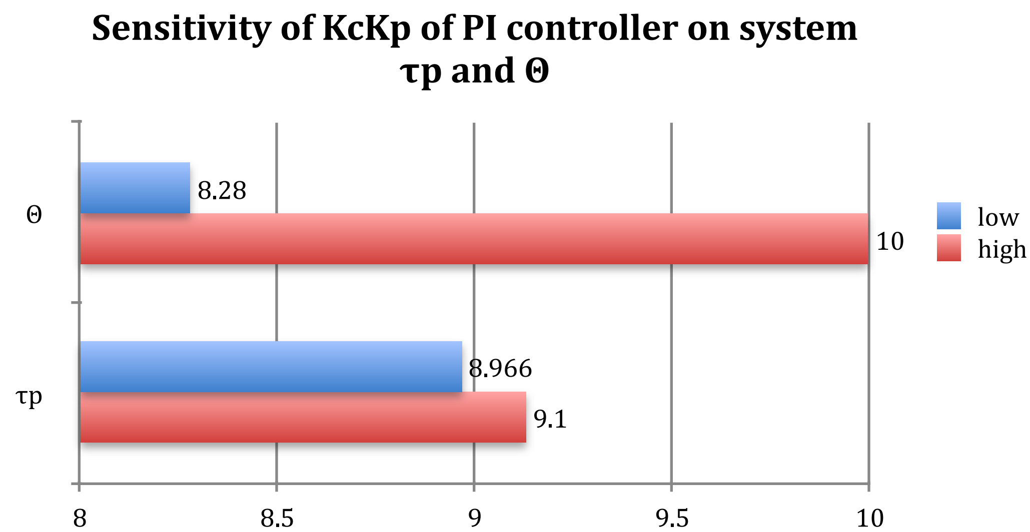


FIGURE 7: TORNADO DIAGRAM OF CONTROL PARAMETER IN PI CONTROLLER

PID CONTROLLER

From table 12.1 case H

$$K_c K_p = \frac{290 + \frac{3}{2}}{29 + \frac{3}{2}} = 9.552$$

$$K_c = \frac{9.552}{k_p} = \frac{9.552}{0.4} = 23.82$$

$$\tau_I = \tau + \frac{\theta}{2} = 290 + \frac{3}{2} = 291.89$$

$$\tau_D = \frac{\tau\theta}{2\tau + \theta} = \frac{291 \times 3}{2 \times 291 + 3} = 1.51$$

SENSITIVITY ANALYSIS FOR PID CONTROLLER

The same sensitivity analysis scheme is performed on PID controller. Again we see that delay have higher impact on gain than τ_p

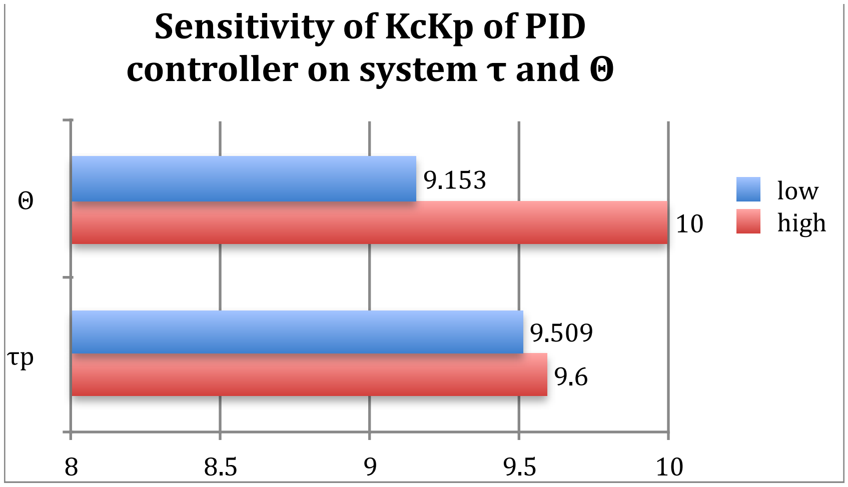


FIGURE 8: TORNADO DIAGRAM FOR KC IN PID CONTROLLER

τ_I *is highly influenced by* τ_p

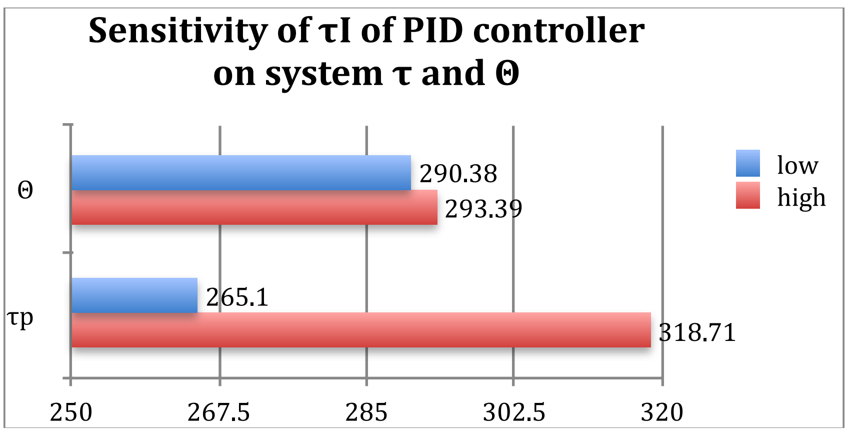


FIGURE 9: TORNADO DIAGRAM FOR TAUI IN PID CONTROLLER

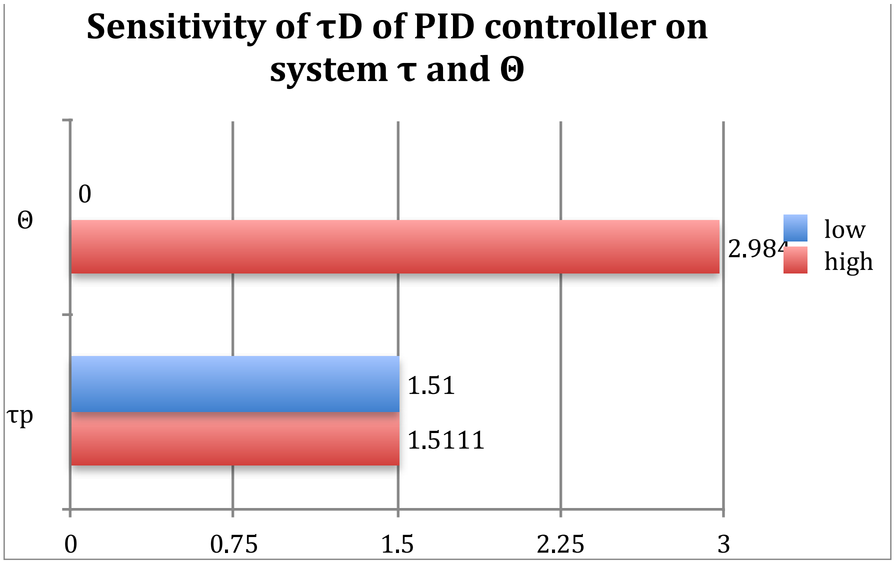


FIGURE 10 : TORNADO DIAGRAM IN TAUD IN PID

For our range, the value of τ_D only depends on delay.

MATLAB SIMULINK SIMULATION

A simulation is produced in Matlab Simulink.

$$G_p = \frac{0.4}{290s + 1}$$

Time delay is set to 3.

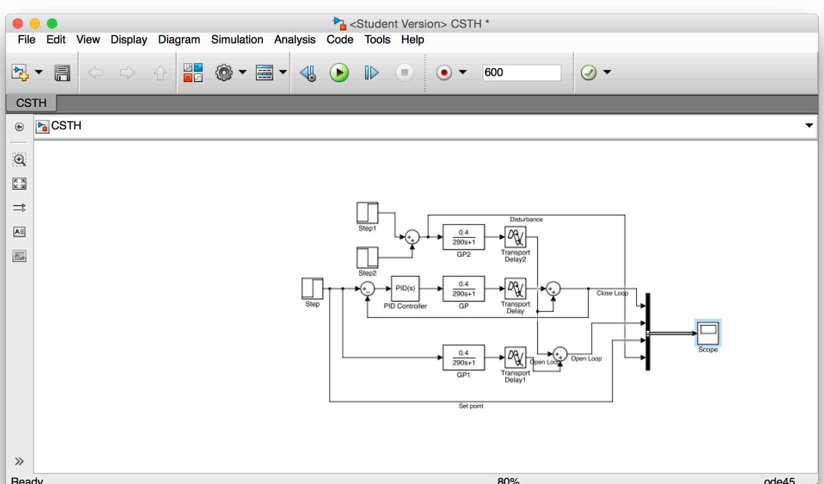


FIGURE 11 : SCREEN SHOOT OF SIMULINK SIMULATION OF PIPTD MODEL

The PID controller transfer function from the text book is

$$G_c(s) = K_c(1 + \frac{1}{\tau_I s} + \tau_D s)$$

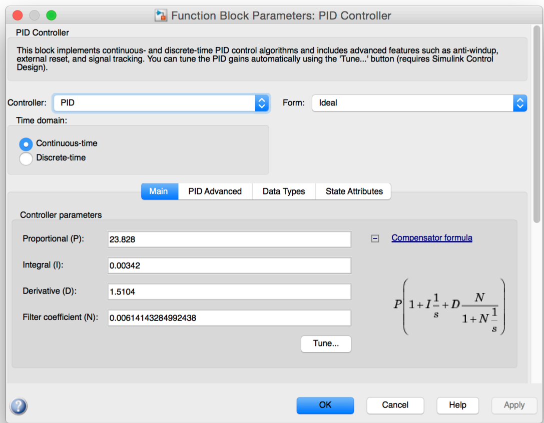


FIGURE 12 : SCREENSHOT OF PID CONTROLLER IN SIMULINK

The form Matlab expect for Ideal PID controller is different from the textbook.

Comparing the Matlab form to the textbook form we obtained

$$P = K_c$$

$$I = \frac{1}{\tau_I}$$

$$D = \tau_D$$

When N approach 0 the term

$$\frac{N}{1 + \frac{N}{S}} \rightarrow s$$

So we set N to be a very small number.

Simulation Set point change

Setting step change to 20 at t=10

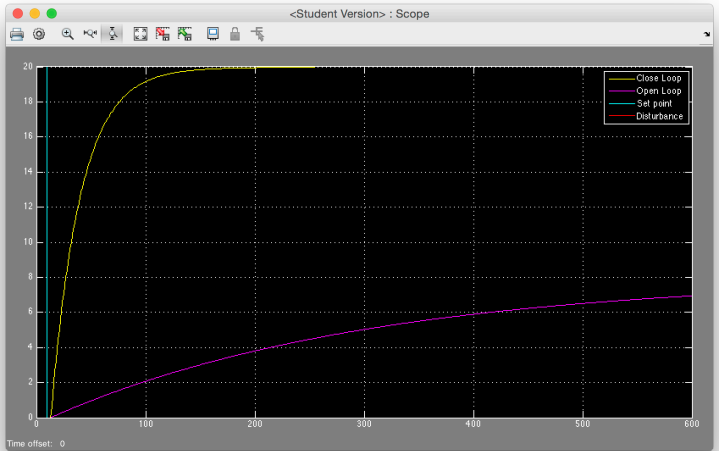


FIGURE 13 : SIMULINK OUTPUT FOR OPEN AND CLOSE LOOP RESPONSE

We see that system reach steady state before t=200 in close loop, which is significantly faster than open loop response.

SIMULATION DISTURBANCE

We set disturbance to 100 from t= 10 till t=20 and observed the following response.

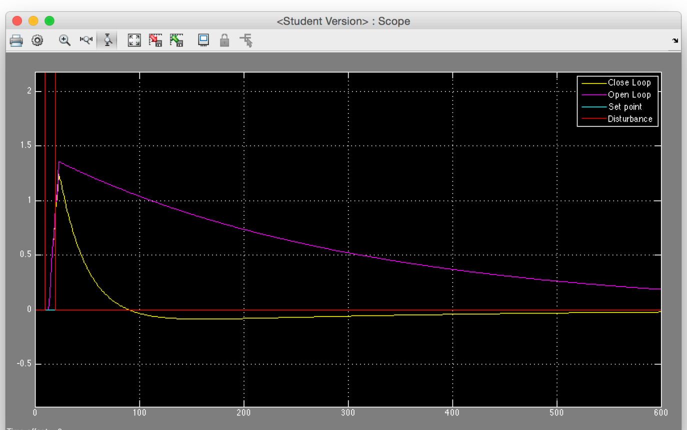


FIGURE 14 : SIMULATION OF OPEN AND CLOSE LOOP DISTURBANCE REJECTION

We see the close loop system rejects the disturbance a lot faster than open loop system. Both systems deviate from set point by 1.4 after the disturbance but close loop system recovers after 300s.

NUMERICAL STABILITY ANALYSIS OF RECOMMENDED CONTROL PARAMETERS

A controller must deliver stability for process safety concern. A Matlab code is developed to evaluate the stability for a typical first order process with $Kp = 0.4,\ \tau_p = 290.$ 300 random combinations of P and τ_I value are chosen and the root of numerator of the characteristic equation is evaluated. The PI setting is considered stable if and only if all roots of the characteristic equation are less than zero. See Appendix C for the Matlab code.

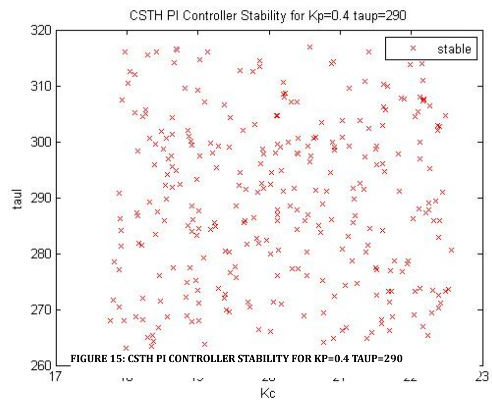


FIGURE 15: CSTH PI CONTROLLER STABILITY FOR KP=0.4 TAUP=290

Similar Matlab code is developed to evaluate the stability of PID controller in the recommended range. 300 combinations of Kp, τ_I and τ_p are randomly chosen and the root of the numerator of the characteristic equation is evaluated. The combination is considered stable if all roots are less than zero. See appendix C for the Matlab code.

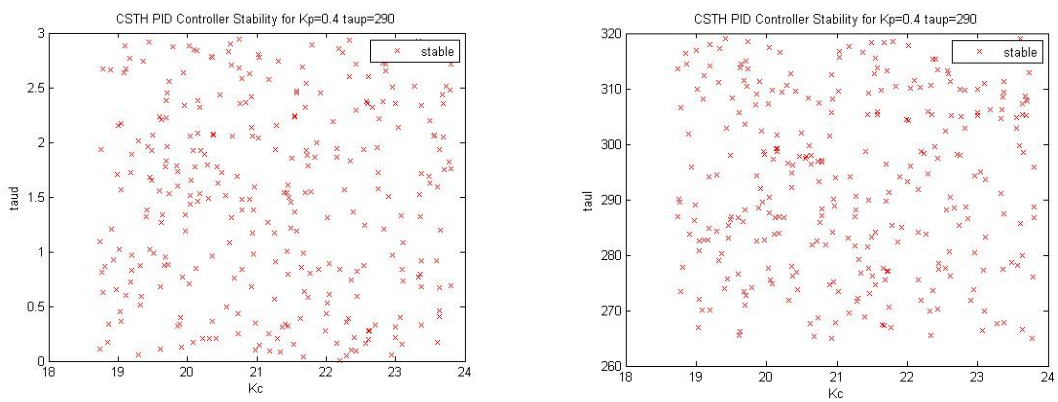


FIGURE 16: CSTH PID CONTROLLER STABILITY ANALYSIS

SUMMARY OF RECOMMENDED CONTROLLER PARAMETERS

The PI and PID controller parameters are as follow:

$$G_c = P\left(1 + I\frac{1}{s} + Ds\right)$$

TABLE 3: SUMMARY OF RECOMMANDED CONTROL PARAMETER

	P	I	D
PI (min)	17.75	0.00315	-
PI (recommend)	22.582	0.00344	-
PI (Max)	31.21	0.00379	-
PID (min)	18.73	0.00313	0
PID(Recommend)	23.828	0.00342	1.5104
PID(Max)	32.94	0.00377	2.96

We expect mediocre performance because the controller is developed using average system parameters. The nonlinearity by operating temperature and direction will increase the difficulty of designing a controller.

CONTROL TESTING (2ND SESSION)

EXPERIMENT PROCEDURE

Using the recommended control parameter as initial starting points, 3 experiments were conducted for PI and 3 for PID controller. The additional experiments tune the controller in the direction that improves performance.

PERTINENT THEORY

In close loop, a PID controller is added to change the manipulated variable in a way that will lead the controlled variable to the desired set point.

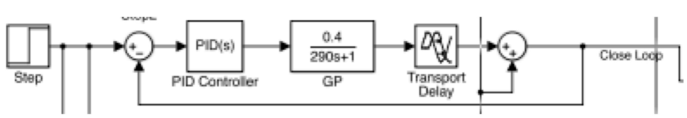


FIGURE 17 : BLOCK DIAGRAM OF CLOSE LOOP SYSTEM

CURVE FITTING

Experimental data are fitted using PI or PID and First Order Plus Time Delay model. Curve fitting is performed using Matlab lsqcurvefit and a custom PID and first order plus time delay model. See Appendix D for Matlab code. The model use Matlab function 'tf' to create the process transfer function. Time delay is specified by using 'InputDelay' option. Matlab function 'pid' is used to create the PI and PID transfer function. D is zero when the controller is PI. The PID and Process transfer function is multiplied and the loop is closed using Matlab function 'feedback'. The response in time domain is obtained with Matlab function 'step'. Amplitude of the step change is specified using 'stepDataOption' in 'StepAmplitude'.

Matlab function 'Lsqcurvefit' is used to search for process parameter Kp, taup, and theta. The purpose of curve fitting the data is to i) allow response comparison without the noise, especially when the response difference are minimum and the noise is getting in the way; ii) allow response comparison when the amplitude of the step change are not the same. The step response can be simulated using fitted data to get the most out of experimental data to draw conclusion.

Quantifying Error And Noise

We calculate the integral square error and sum of square residual for both PI and PID controller using experimental data:

$$ISE = \sum_{t=1}^{n} \left(M - T'(t)\right)^2$$

ISE is a good measurement of controller performance. It is the squared area between the curve y=M and the response curve. The smaller ISE value the better the performance of the controller because the performance is closer to the set point. Using square error instead of the area between curves allow us to capture the deviation from set point even when the response overshoot. Squaring the error make sure positive error and negative error don't cancel each other out.

$$Residual = \sum_{t=1}^{n} (T'_{fitted}(t) - T'(t))^2$$

Residual is a good measure of noise when the curve fitting is done reasonably well. As the fitted curve ideally pass through the center of the response data, the vertical distance between each data point and the fitted curve is noise. The sum of all the vertical distance squared is the residual, and also the sum of square of noise. Similar to ISE, the squaring ensure that positive noise don't cancel out negative noise.

Result

See Appendix B for the experimental data and curve fitting result.

TABLE 4 : SUMMARY OF CLOSE LOOP PERFORMANCE FOR SET POINT CHANGE FROM 55 TO 60

	Kc	tauI	taud	Kp	taup	theta	resid
PI baseline	22.6	290	0	0.095	50.477	6.833	30.522
PI with lower integrative term	22.4	260	0	0.054	43.038	5.69	20.753
PI with higher proportional term	27.4	290	0	0.033	43.433	4.544	30.269
PID baseline	23.8	291.9	1.5	0.046	44.524	6.902	24.38
PID with lower integrative term	23.8	261.9	1.5	0.043	43.102	7.397	16.846
PID with higher proportional term	27.8	291.9	1.5	0.036	48.426	2	26.106

Comparing Open Loop And Close Loop Performance

Comparing open loop and close loop performance is a challenge because the smallest step change performed in open loop has a magnitude of 20, while the largest step change performed in close loop has a magnitude of 10.

To take advantage of the data, the PI close loop data is curve fitted with a PI plus first order model with the known PI setting (P=23.82, I=291.89) to estimate process parameters. It was determined that Kp=0.047 and taup =234.211.

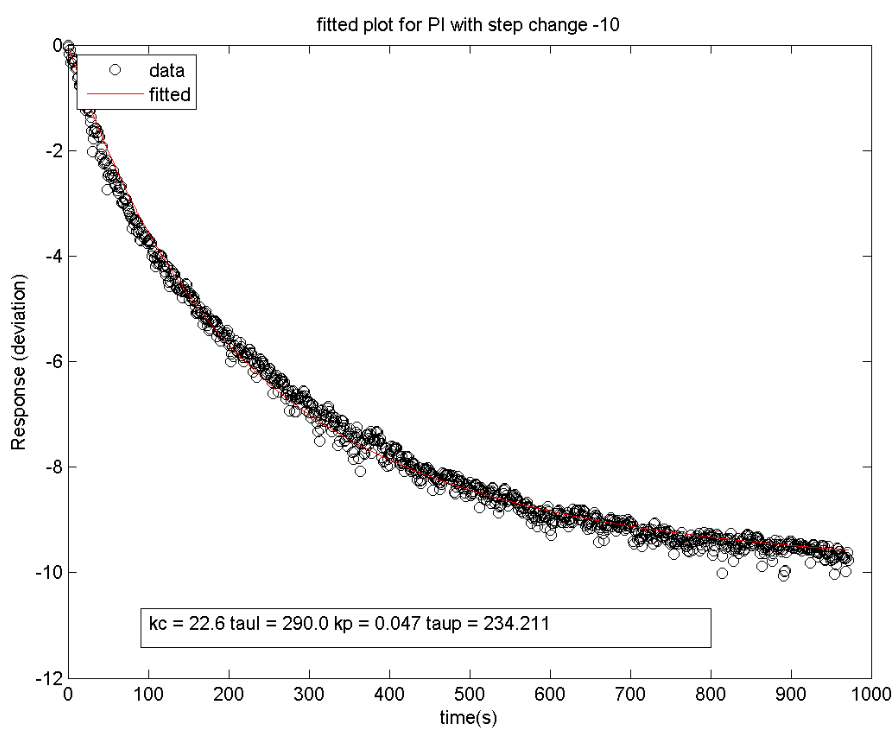


FIGURE 18: FITTED PLOT FOR PI WITH STEP CHANGE OF -10

Using this information, close loop response with step change of 20 is simulated using Simulink and compared with the 4 negative step change experiments in open loop.

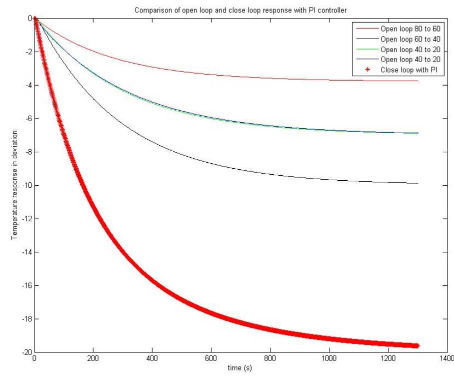


FIGURE 19 : COMPARISON OF OPEN LOOP AND CLOSE LOOP RESPONSE WITH PI CONTROLLER

The advantage of the PI controller in close loop is significant. Temperature response in open loop does not match the step change in the input in steady state. However, a step change in set point in close loop results in a change in the controlled variable in 22 minutes with very little offset. The response is stable and does not overshoot. PI controller in close loop improved the controllability and stability of the process.

CLOSE LOOP THEORETICAL EXPECTATION VS EXPERIMENTAL RESULT

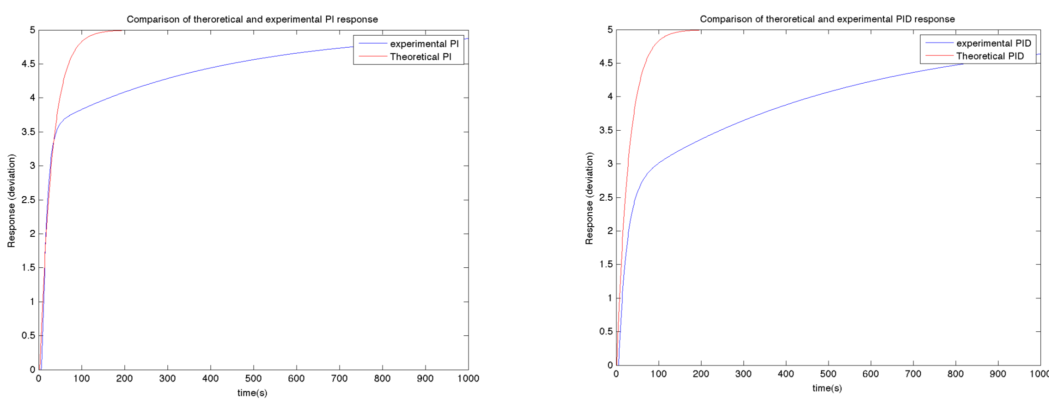


FIGURE 20 : COMPARISON OF CLOSE LOOP THEORETICAL EXPECTATION AND EXPERIMENTAL RESULT

Using the average process parameter (Kp=0.4 taup=290 theta =3) and the recommended control parameters (PI: P=22.6 I=290 PID: P=23.8 I=291.9 D=1.5) we simulate the theoretical expectation of set point change response of magnitude 5. The experimental curve is obtained using fitted experimental close loop response in PI and PID.

Figure 20 shows the difference between theoretical expectation and experimental result. Assuming process parameter will hold the same in close loop, we expected the process will reach steady state at around 200s. But in reality, for both PI and PID the experimental response is much slower than expected. The system response very fast initially for the first half of the change and took 20 times as long to complete the other half of the change.

This allows us to see the nonlinearity of the system. The process parameter Kp taup and theta will changes when we close the loop. Kp drop by one tenth and taup drop by one sixth when we close the loop. This kind of change is unpredictable with our knowledge.

In practice, this is going to be a common scenario. This means onsite tuning is often required to obtain optimal performance. The corresponding optimal controller setting cannot be determined with open loop system identification alone.

Fine-Tuning PI Controller

PI controller is tested in the recommended setting. During the lab, we also tested parameters that are likely to improve system response.

Figure 21 shows the plot of CSTH system response in recommended PI setting and in alternative setting. Note that all tests were performed with a set point change of 55 degree to 60 degree. Unfortunately, PI baseline response curve have lower gain because the test started at 55.7 degree instead of 55 degree. This error makes it hard to compare the alternative setting with the recommended setting.

TABLE 5 : REFINING PI CONTROLLER

Runs	P	I	D	M
PI baseline	22.65	290	0	4.3
PI with lower integrative term	22.4	260	0	5
PI with higher proportional term	27.4	290	0	5

To take advantage of the existing data, the PI baseline run is curve fitted with a PI and first order model to solve for Kp and taup for this particular run. The Kp and taup, along with PI setting is used to provide PI Baseline Simulated curve with step change of 5.

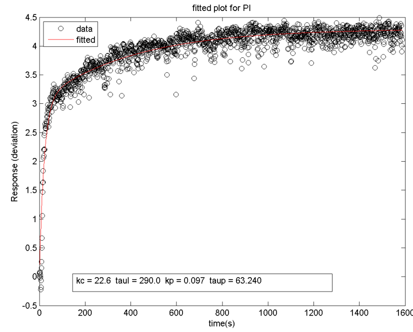


FIGURE 21: PI RESPONSE WITH STEP CHANGE OF 4.3

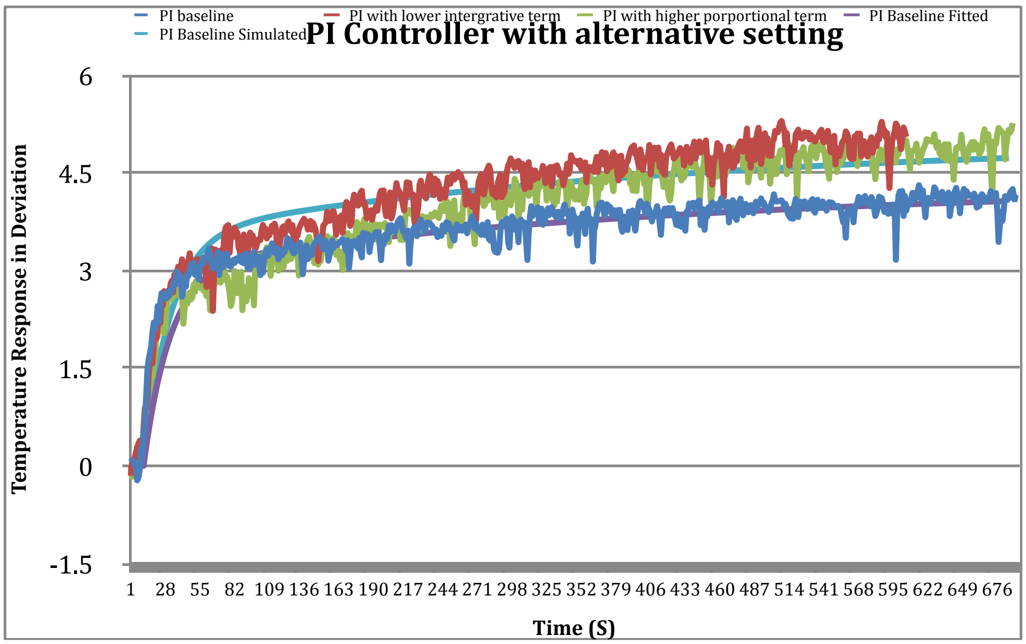


FIGURE 22 : COMPARISON OF PI INITIAL SETTING TO FINE-TUNING

Comparing PI baseline simulated curve with PI with alternative setting, it is visible that both alternative setting are faster than the baseline setting in reaching steady state. It is promising that system will reach steady state even faster if P is higher and I is lowered at the same time.

We calculate the integral square error between experimental curve and Y=5 curve. The results are listed in Table 6.

TABLE 6: INTEGRAL SQUARE ERROR OF PI CONTROLLER EXPERIMENT

Runs	ISE
PI baseline	2.0001e+03
PI with lower integrative term	940.6463
PI with higher proportional term	1.3547e+03

The lower the ISE the better the performance. The numerical analysis agrees with graphical analysis that PI with lower integrative term is the best controller out of all three. Both retuned setting are better than the initial setting.

STABILITY ANALYSIS OF PI CONTROLLER

Graphically, the temperature responded smoothly without overshooting. The PI controller reaches steady state within about 12 minutes. The controller delivered stability as predicted in numerical stability analysis.

FINE-TUNING PID CONTROLLER

PID controllers are tested in the recommended setting and the control parameters are fine-tuned. The magnitude of the tuning is similar to those performed in PI fine-tuning. Table 7 summarized the fine-tuned setting attempted for PID controller.

TABLE 7: RETURNING PID CONTROLLER

Runs	P	I	D
PID baseline	23.82	291.89	1.51
PID with lower integrative term	23.82	261.89	1.51
PID with higher proportional term	27.82	291.89	1.51

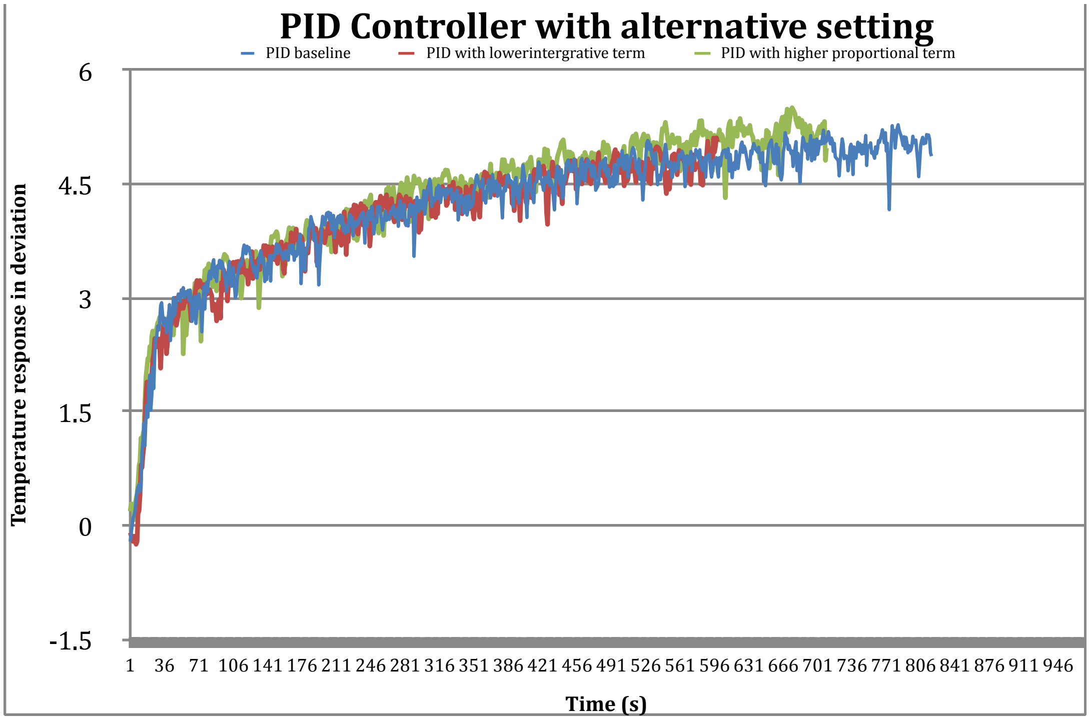


FIGURE 23 : COMPARING FINE-TUNED PID CONTROLLER PARAMETERS

Figure 23 compares the fine-tuned PID controller parameters. With the same magnitude of adjustment, the benefit of fine-tuned setting is not significant as in PI. Graphically, the response curve overlaps each other. One can argue that there is practically no difference between the settings. To improve the performance of PID controller, greater magnitude of change is need.

Stability Analysis Of Pid Controller

Graphically, the temperature responded smoothly without overshooting. The PID controller reaches steady state within about 12 minutes. The controller delivered stability as predicted in numerical stability analysis.

Comparison Of Pi And Pid Performance And Robustness

ISE is a good measurement of controller performance. The lower the ISE the smaller the squared area between set point and experimental response over time. From the ISE in Table 8 we can see that PI with lower integrative term performs the best out of all controllers. Changing P and changing I both significantly improve performances. PID controllers on average perform better than PI controllers except when integrative term is low.

TABLE 8: SUMMARY OF ISE, RESIDUAL OF CONTROLLED AN MANIPULATED VARIABLES

	ISE	Residual(T)	Residual(Manipulated)
PI baseline	2.0001e+03	30.52	26260.32
PI with lower integrative term	940.6463	20.75	10312.87
PI with higher proportional term	1.3547e+03	30.26	11684
PID baseline	1.1361e+03	24.38	39102.85
PID with lower integrative term	1.1774e+03	16.84	39395.99
PID with higher proportional term	1.0292e+03	26.10	52493.64

Residual is a good measure of noise when the curve fitting is done reasonably well. However the goodness of fit is not guaranteed and it varies from one curve to another. Residual is also proportional to the length of the data. After considering all factors influencing residual, we conclude that the amount of noise in PI and PID response is about the same.

A noisy controlled variable will have an impact on the controller especially for PID. PID controller anticipates future process change by based on derivative of the past and attempted to control them. A noisy data will lead PID controller to make control action based on noise, thus wearing off the valves.

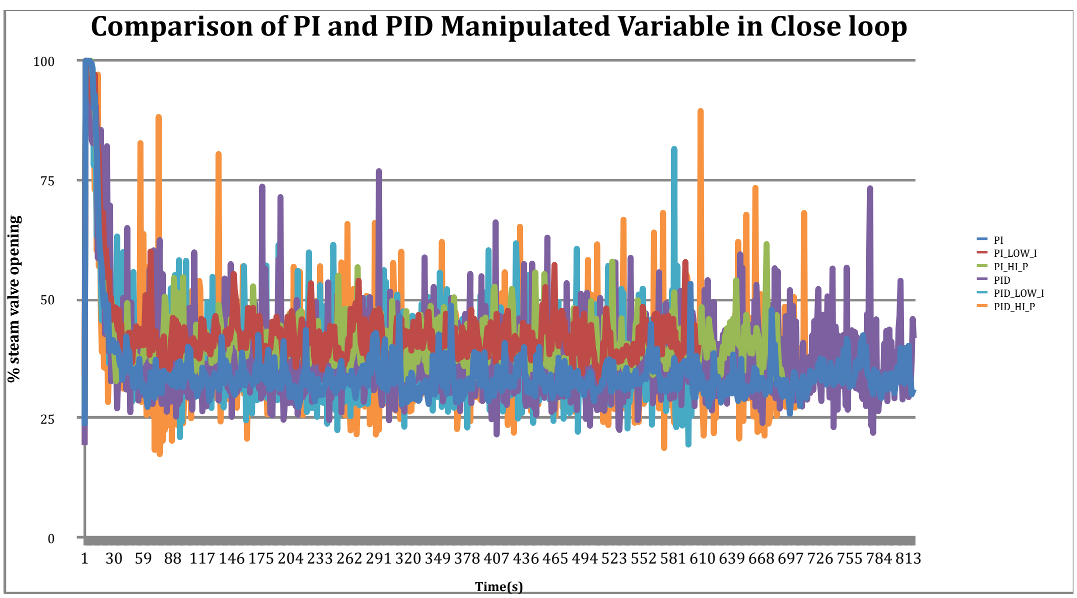


FIGURE 24: COMPARISON OF PI AND PID MANIPULATED VARIABLE IN CLOSE LOOP

Figure 24 shows the comparison of PI and PID in manipulated variable. A controller must make control action to bring the system to set point. However, if the magnitude of the action and the frequency of the action are too high, it will wear the valves very quickly, shorten the life span of the valve and increase operating cost in replacing valves. High magnitude of the control action might also subject the process to instability. From figure 24, it is clear that all the peaks that stick out are from PID controller. The magnitude of controlled actions is visibly higher in PID than in PI.

Assuming the valve opening stabilized at 39.5, we calculate the residual from t=30 to t=600 for all runs. Result is shown in Table 8. The numerical analysis agrees with graphical observation. PID has higher noise in its manipulated variable. Lowering I in PID controller does not affect the amount of control action. However, increasing gain in PID controller increase the magnitude of control action significantly. This means that on noisy data, P should be kept low for PID controller, if PID controller has to be used at all.

For PI controller, improving performance also results in lowering the amount of control action. This means PI is a good for noisy data.

Disturbance Rejection

Disturbance rejection is tested for both open and close loop by adding the same amount of ice to the system. Figure 25 shows how the system responses to the disturbance. Open loop system takes 12 minutes to reject the disturbance, while close loop systems takes about 6 minutes to reject the disturbance. This demonstrated the controller added stability and performance to the system.

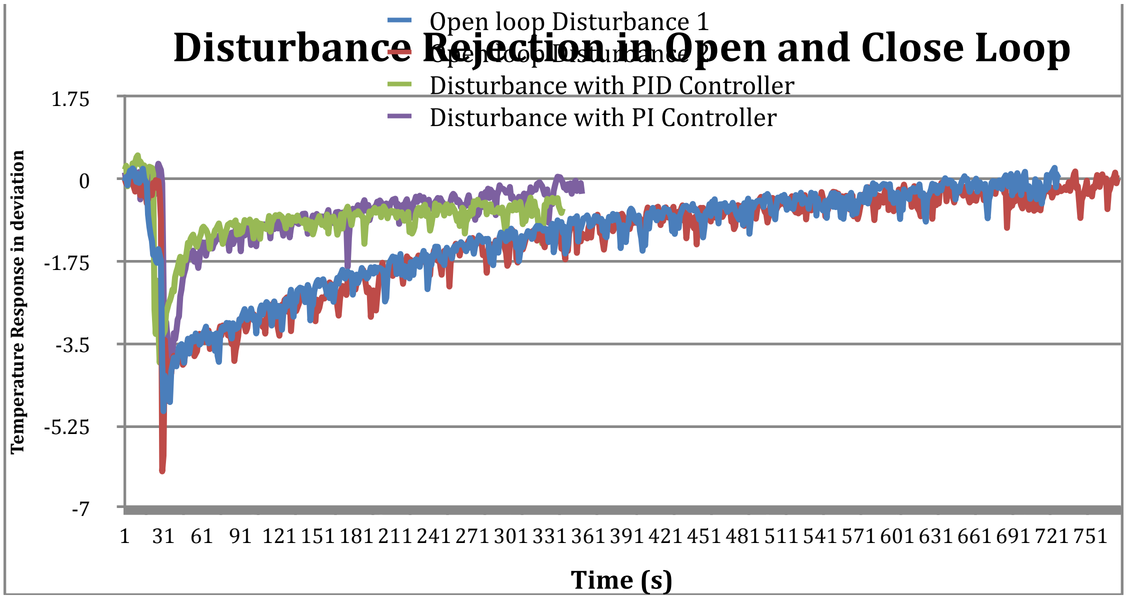


FIGURE 25: DISTURBANCE REJECTION IN OPEN AND CLOSE LOOPS

Table 9 shows the integral square error for disturbance rejection. Disturbance rejection is much better in close loop than in open loop.

TABLE 9: INTEGRAL SQUARE ERROR FROM THE LOWEST DEVIATION

	ISE
Disturbance with PID Controller	389
Disturbance with PI Controller	386
Disturbance in open loop 1	1790
Disturbance in open loop 2	2101

Disturbance Rejection With PI And PID

A closer look at PI and PID controller in figure 26 shows that PI controller gives less offset than the PID controller. However the difference is less than half of a degree. ISE calculation shows that PI controller is better than PID by a small margin.

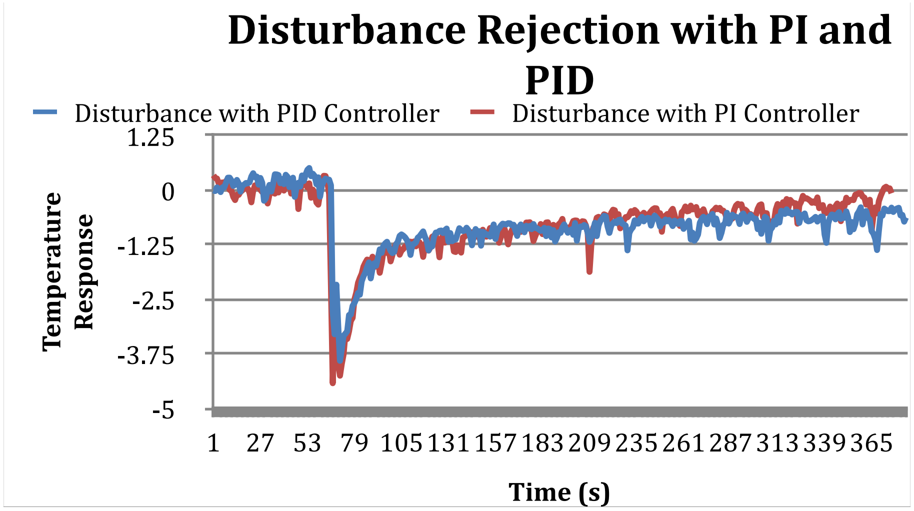


FIGURE 26 : DISTURBANCE REJECTION WITH PI AND PID

Conclusion

Close loop performance is far greater than open loop performance. In close loop, a step change in set point results in predictable steady state gain. This is true for all close loop experiment. In open loop, the steady state gain is a result of operating temperature and step change and is difficult to predict.

System is nonlinear with respect to operating temperature as well as loop arrangement. Closing the loop significantly changes process parameters.

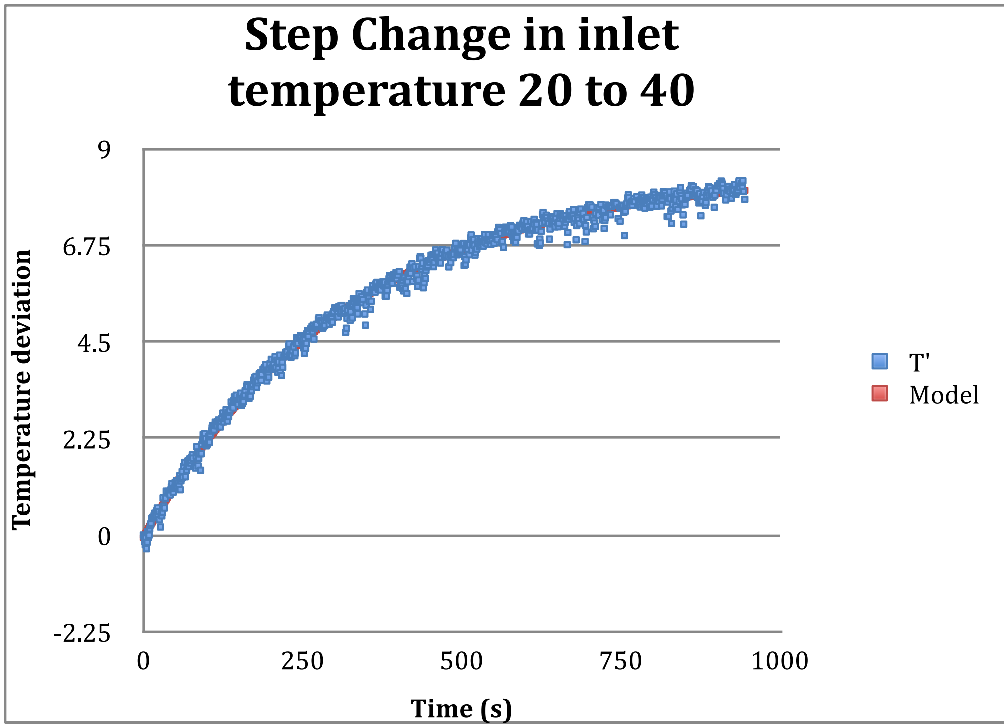
PI and PID controller delivered stability for the duration of the experiment in close loop as predicted in numerical stability analysis.

For the same magnitude of changes PI controllers are more sensitive to tuning than PID.

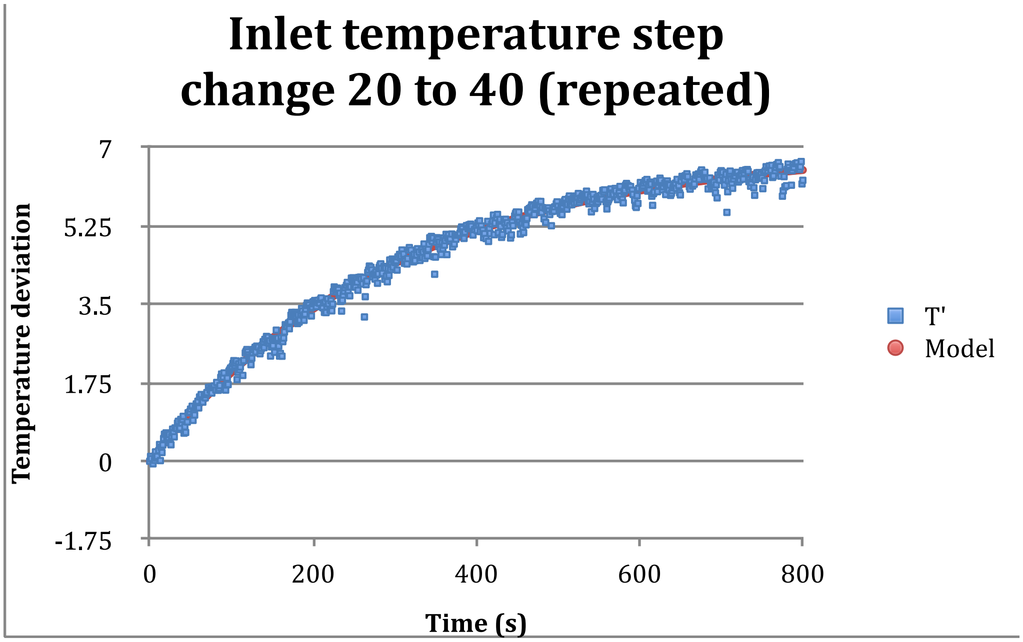
PID controller provides stronger control action than PI controller. This is due to noise in the controlled variable. Using PID controller for noisy system will wear off the valve.

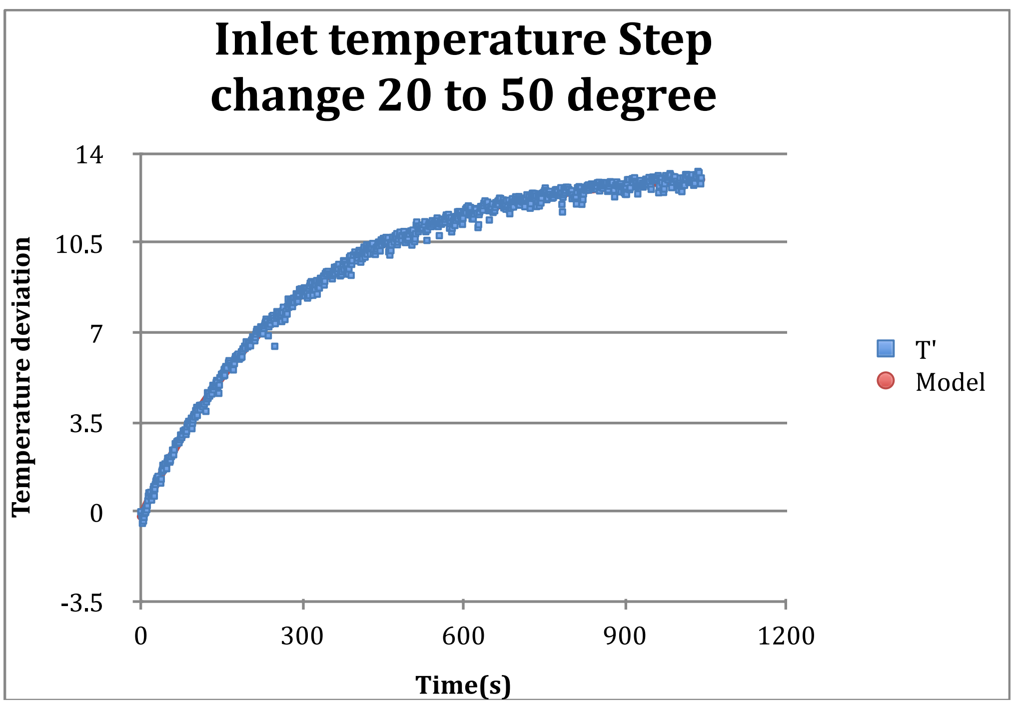
Both PI and PID controller deliver better disturbance rejection than open loop. The difference between the two is marginal.

Appendix A – Open Loop Curve Fitting Data

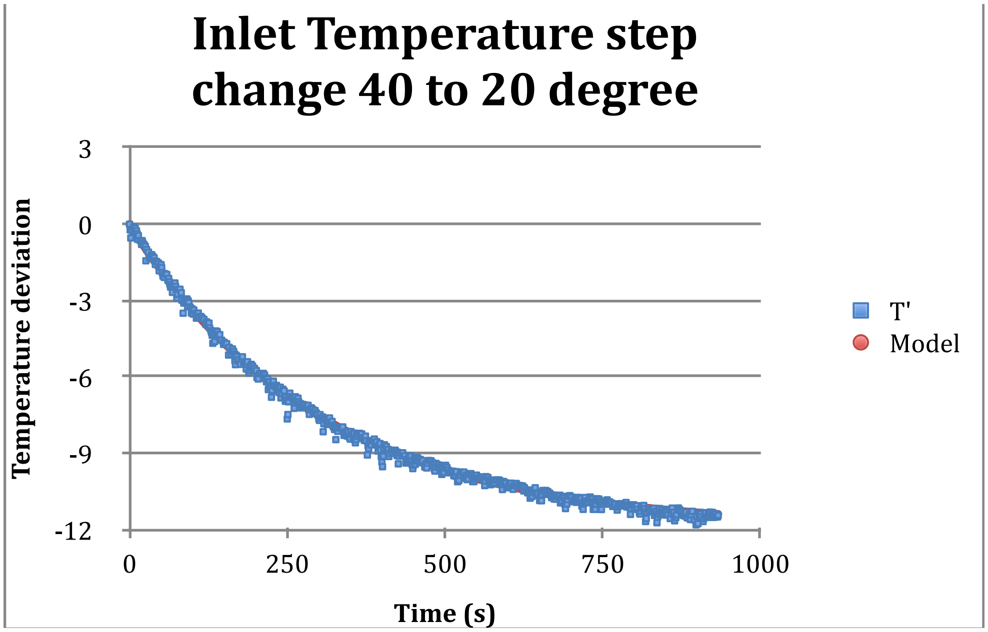


Kp = 0.42 taud=2.22 taup 326

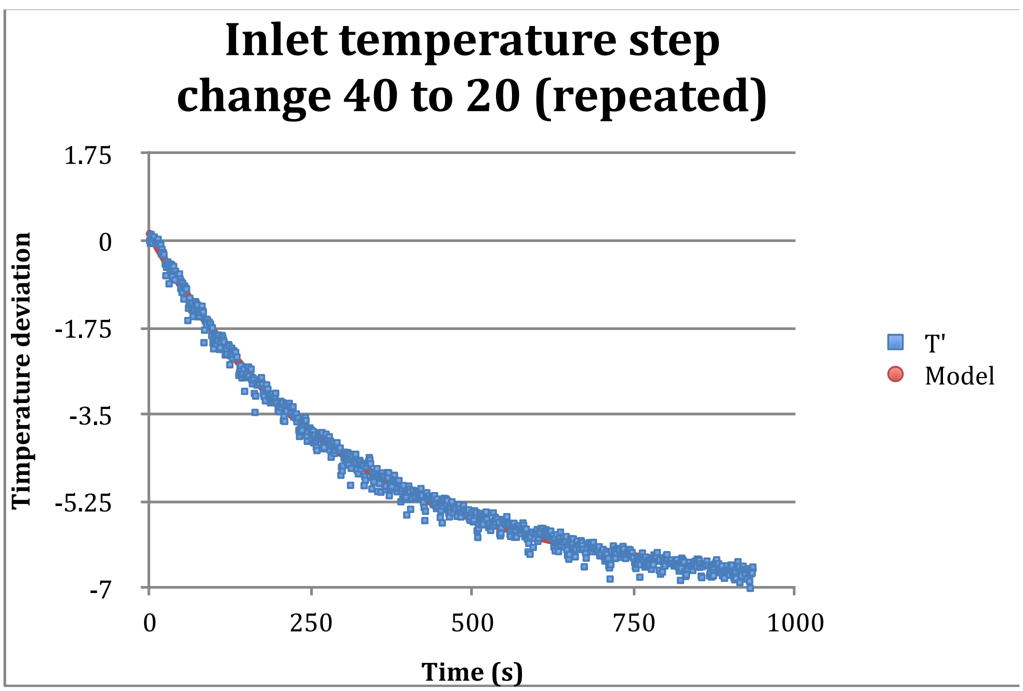




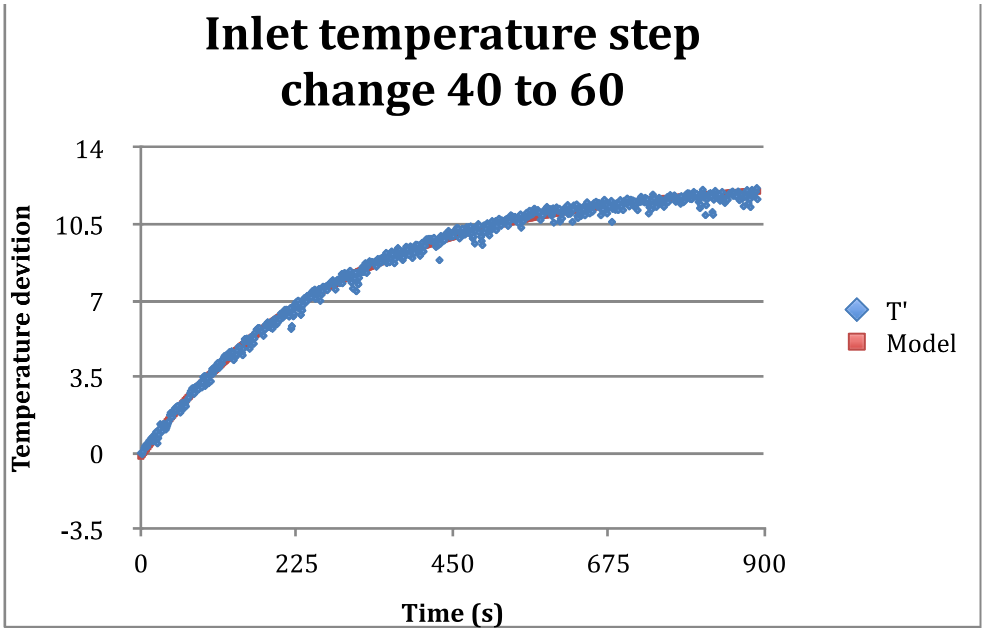
Kp= 0.44 taud = 4.011 taup = 293



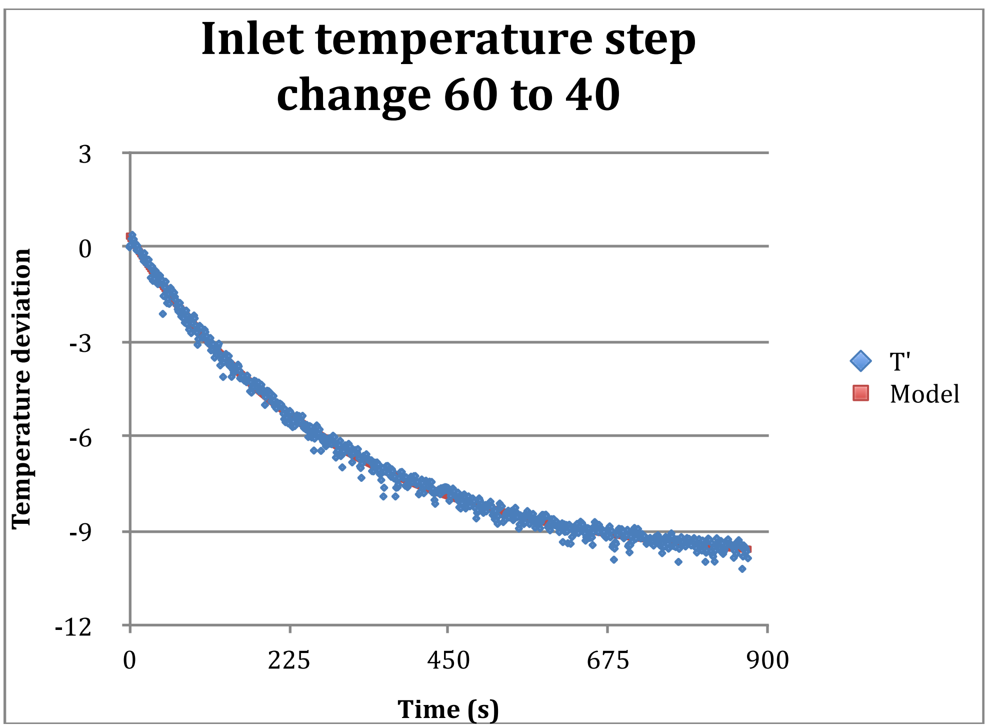
Kp=16.22 taud=0.000878 taup =300



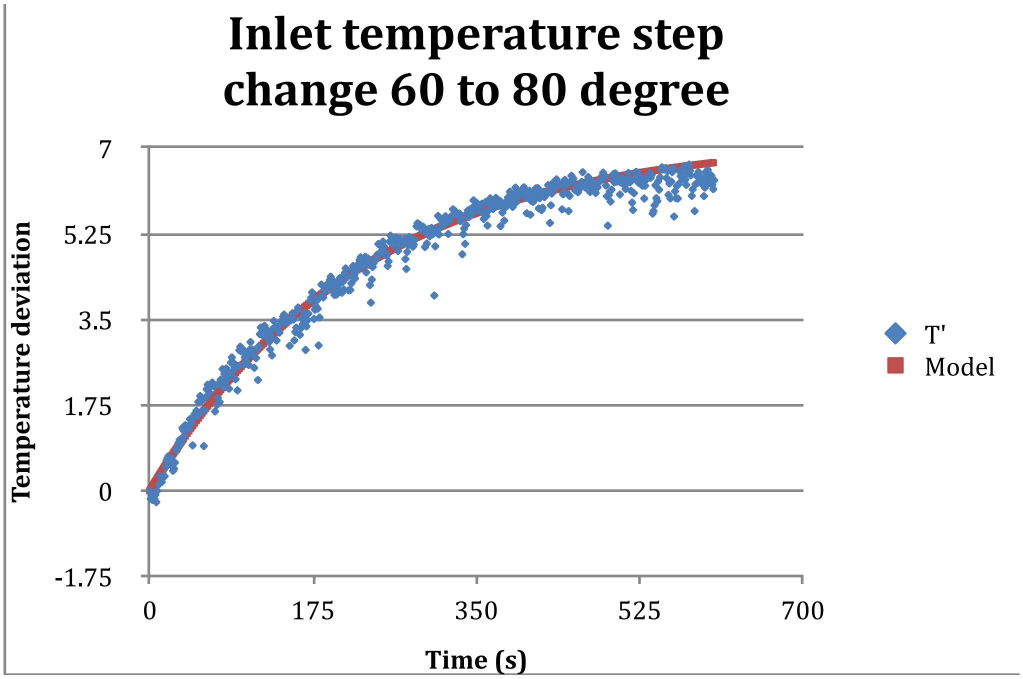
Kp = 0.3473 taud = 6.02 taup



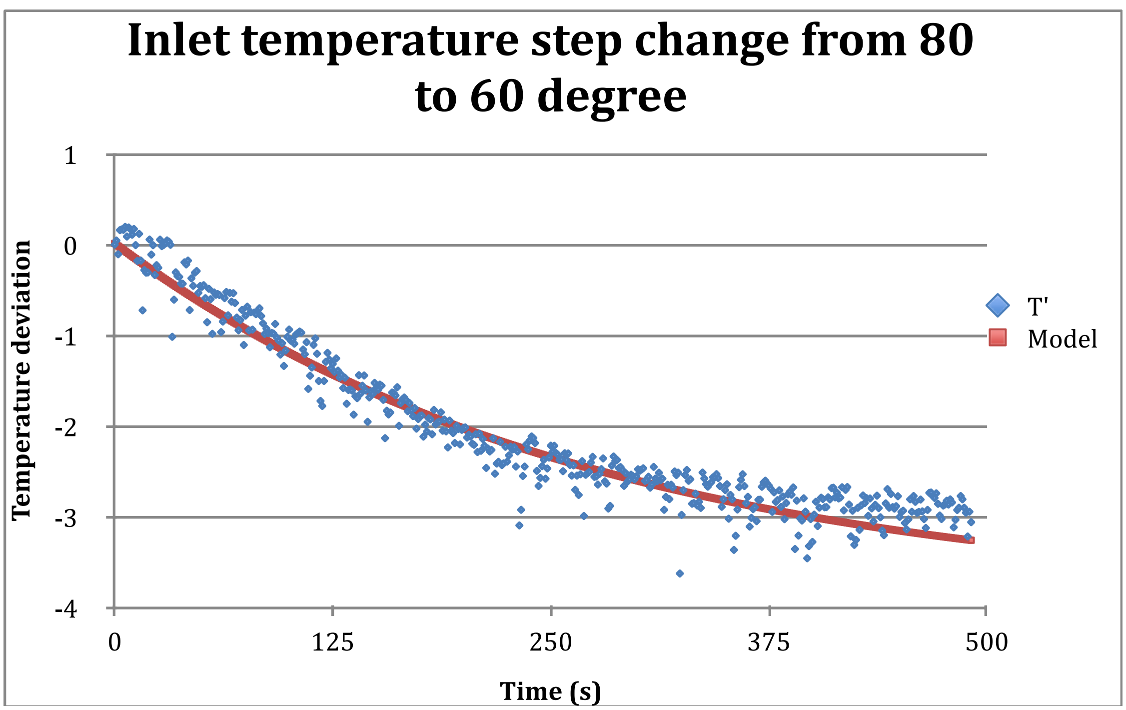
Kp=0.6303 taud=3.1175 taup=287.88



Kp=0.5046 taud=9.930

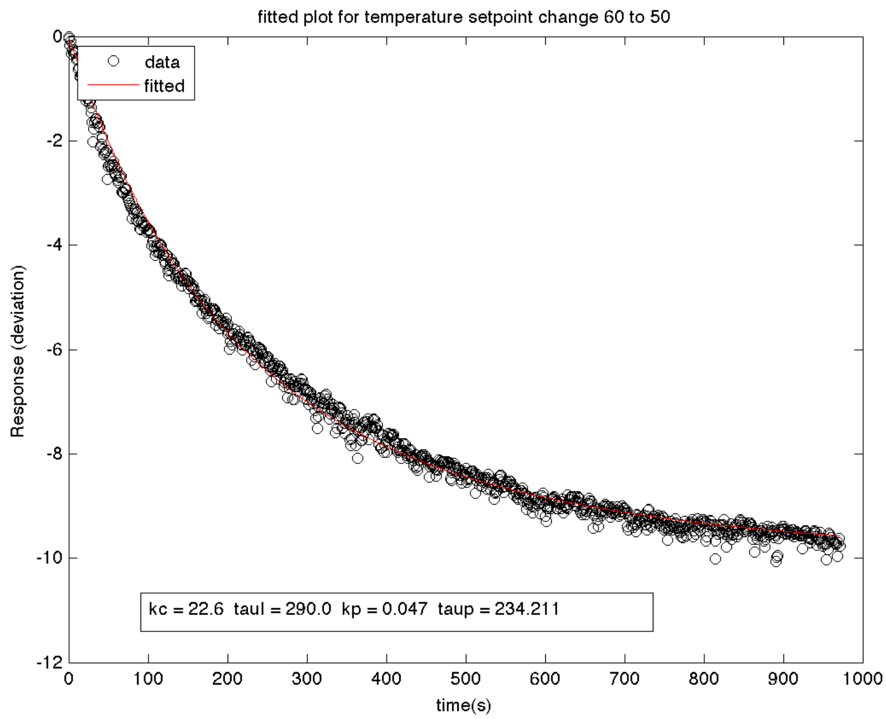


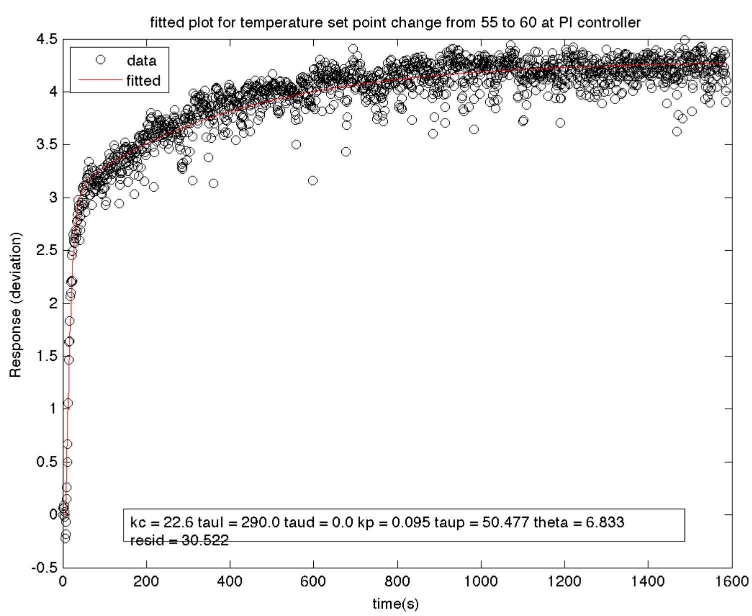
Kp=0.3594 taud=0.1022 taup

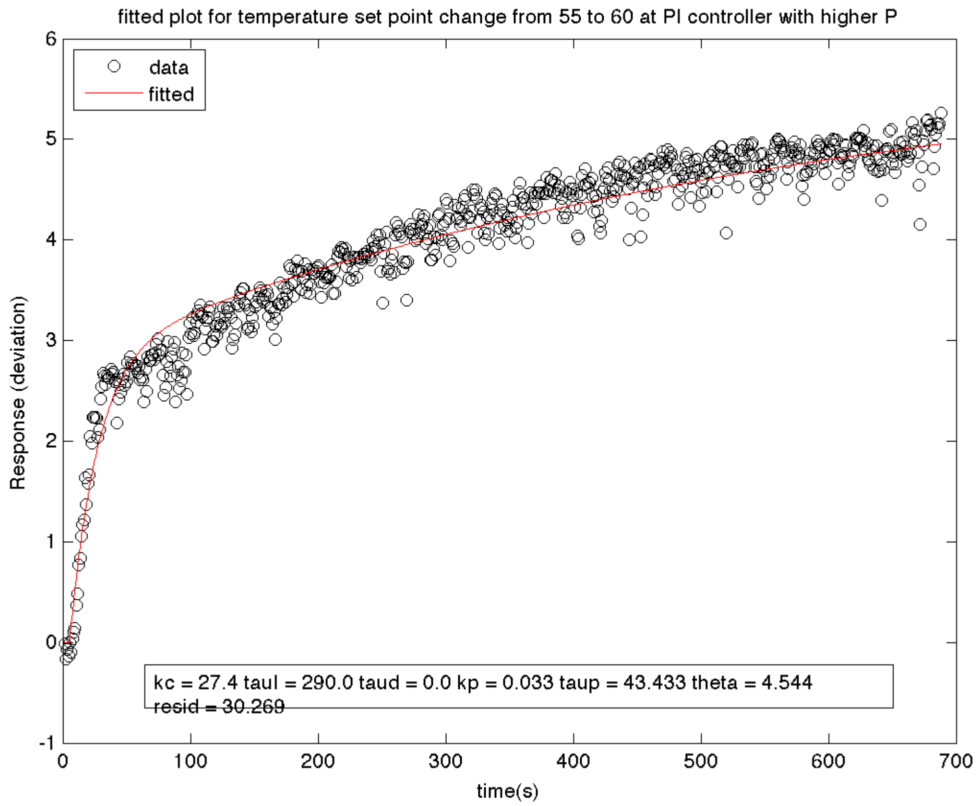


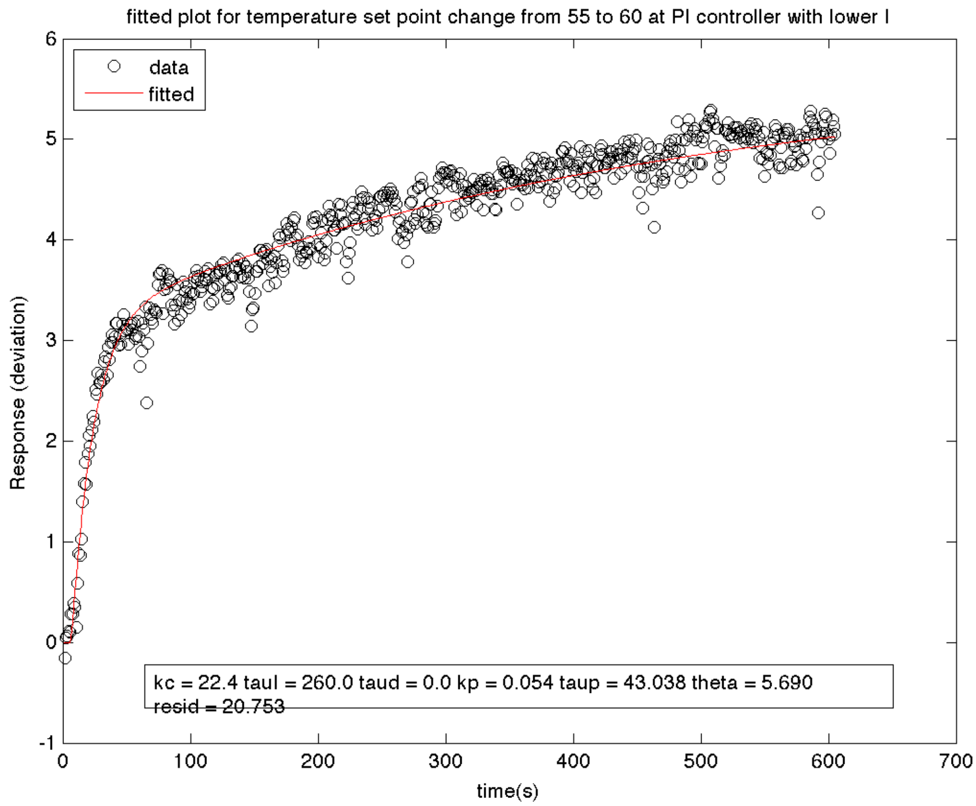
Kp=0.1949 taud=0.9745 taup=271.47

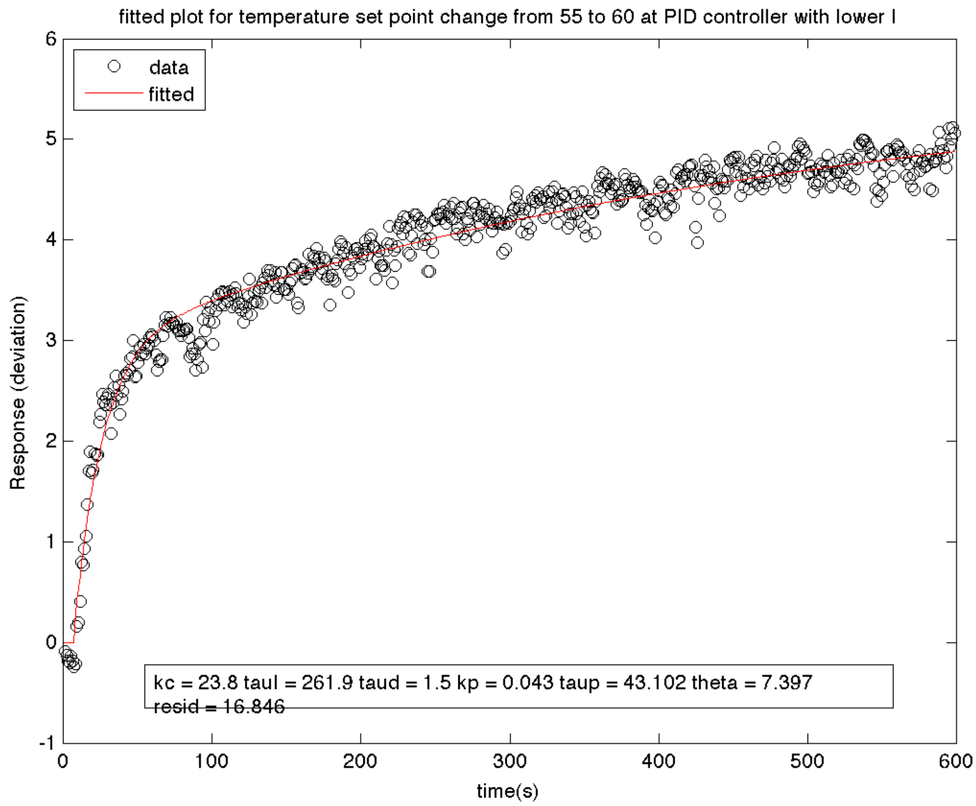
APPENDIX B – CLOSE LOOP CURVE FITTED DATA

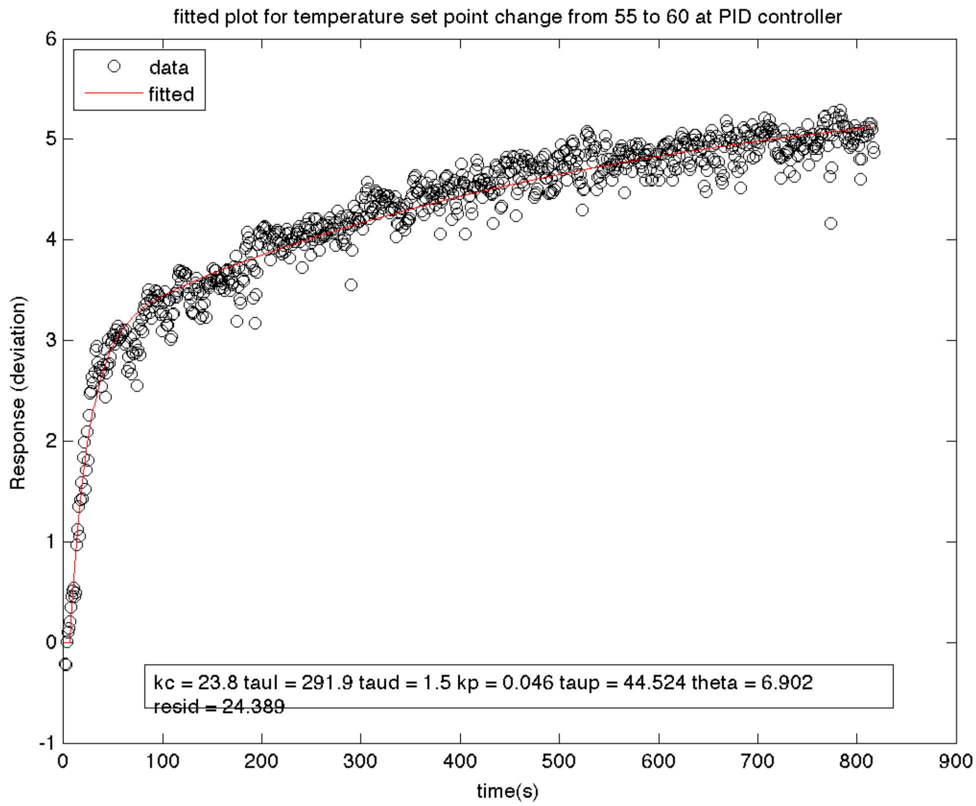


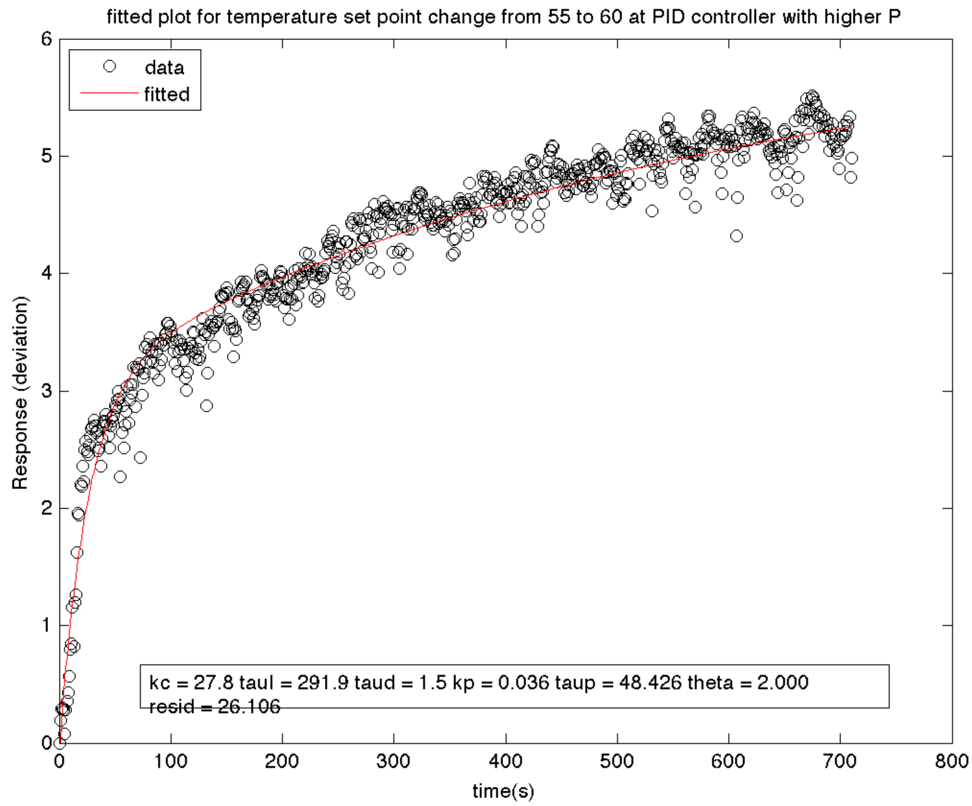



fitted plot for temperature set point change from 55 to 60 at PI controller with higher P
data
fitted
Response (deviation)
time(s)
kc = 27.4 taul = 290.0 taud = 0.0 kp = 0.033 taup = 43.433 theta = 4.544
resid = 30.269
6
5
4
3
2
1
0
-1
0
100
200
300
400
500
600
700


fitted plot for temperature set point change from 55 to 60 at PI controller with lower I
data
fitted
Response (deviation)
time(s)
kc = 22.4 taul = 260.0 taud = 0.0 kp = 0.054 taup = 43.038 theta = 5.690
resid = 20.753
6
5
4
3
2
1
0
-1
0
100
200
300
400
500
600
700


fitted plot for temperature set point change from 55 to 60 at PID controller with lower I
data
fitted
kc = 23.8 taul = 261.9 taud = 1.5 kp = 0.043 taup = 43.102 theta = 7.397
resid = 16.846
Response (deviation)
time(s)
6
5
4
3
2
1
0
-1
0
100
200
300
400
500
600


fitted plot for temperature set point change from 55 to 60 at PID controller
data
fitted
Response (deviation)
time(s)
kc = 23.8 taul = 291.9 taud = 1.5 kp = 0.046 taup = 44.524 theta = 6.902
resid = 24.389


fitted plot for temperature set point change from 55 to 60 at PID controller with higher P
data
fitted
kc = 27.8 taul = 291.9 taud = 1.5 kp = 0.036 taup = 48.426 theta = 2.000
resid = 26.106
Response (deviation)
time(s)
0
1
2
3
4
5
6
0
100
200
300
400
500
600
700
800

APPENDIX C – NUMERICAL STABILITY ANALYSIS

```
% CSTH PI stability Analysis
 clc
clear all

Gp=tf([0.4 1],[290 1])
 Kclb=17.75;
 Kcub=22.582;
 tauIlb=263;
 tauIub=317;
Kc=Kclb+(Kcub-Kclb).*rand(300,1);
tauI=tauIlb+(tauIub-tauIlb).*rand(300,1)
Stab=[];
N_Stab=[];
for i=1:length(Kc)
   Gc=pid(Kc(i),Kc(i).*tauI(i).^-1,0)

   D=1+Gc*Gp
   s=roots(D.num{:});
   if s<0
       Stab=[Stab;Kc(i), tauI(i)];
   else
       N_Stab=[N_Stab;Kc(i), tauI(i)];
   end
end
figure(1)
plot(Stab(:,1),Stab(:,2),'rx')
hold on
xlabel('Kc')
ylabel('tauI')
legend('stable')
title('CSTH PI Controller Stability for Kp=0.4 taup=290')
if size(N_Stab)>0
plot(N_Stab(:,1),N_Stab(:,2),'bo')
end
hold off
```

```
% % CSTH PID stability Analysis


Gp=tf([0.4 1],[290 1])
 Kclb=18.73;
 Kcub=23.828;
 tauIlb=265;
 tauIub=319;
 taudlb=0;
 taudub=2.96;
Kc=Kclb+(Kcub-Kclb).*rand(300,1);
tauI=tauIlb+(tauIub-tauIlb).*rand(300,1)
taud=taudlb+(taudub-taudlb).*rand(300,1)
Stab=[];
N_Stab=[];
for i=1:length(Kc)
   Gc=pid(Kc(i),Kc(i).*tauI(i).^-1,Kc(i).*taud(i))

   D=1+Gc*Gp
   s=roots(D.num{:});
   if s<0
       Stab=[Stab;Kc(i), tauI(i),taud(i)];
   else
       N_Stab=[N_Stab;Kc(i), tauI(i),taud(i)];
   end
end
figure(2)
plot(Stab(:,1),Stab(:,2),'rx')
hold on
xlabel('Kc')
ylabel('tauI')
legend('stable')
title('CSTH PID Controller Stability for Kp=0.4 taup=290')
if size(N_Stab)>0
  plot(N_Stab(:,1),N_Stab(:,2),'bo')
end
```

```
hold off
figure(3)
plot(Stab(:,1),Stab(:,3),'rx')
hold on
xlabel('Kc')
ylabel('taud')
legend('stable')
title('CSTH PID Controller Stability for Kp=0.4 taup=290')
if size(N_Stab)>0
  plot(N_Stab(:,1),N_Stab(:,3),'bo')
end
hold off
```

Appendix D – Matlab Curve Fitting For PID And First Order Plus Time Delay Model

Model Function

```
function X=PIDPTD(x,M,kc,tauI,taud,t)

kp=x(1);
taup=x(2);
theta=x(3);
if theta < 0
    theta = 0
    x(3)= 0
end

den=[taup 1];

Process=tf(kp,den,'InputDelay',theta);
PID=pid(kc,kc./tauI,kc.*taud);
Model=feedback(PID*Process,1);
opt=stepDataOptions('StepAmplitude',M);
[X,t]=step(Model,t,opt);
```

Script

```
% optimize parameter x =[kp ; taup; theta ]
%reset
t=[]
x=[]
fitted=[]
response=[]
i=6
kc=[22.65 22.4 27.4 23.82 23.82 27.82]
tauI=[290 260 290 291.89 261.89 291.89]
taud=[ 0 0 0 1.51 1.51 1.51]
M=[4.3 5.7 5.9 5.7 5.7 6.1]
name={'fitted plot for temperature set point change from 55 to 60 at PI controller' ...
      'fitted plot for temperature set point change from 55 to 60 at PI controller with lower I'...
      'fitted plot for temperature set point change from 55 to 60 at PI controller with higher P'...
      'fitted plot for temperature set point change from 55 to 60 at PID controller'...
      'fitted plot for temperature set point change from 55 to 60 at PID controller with lower I'...
      'fitted plot for temperature set point change from 55 to 60 at PID controller with higher P'}

%input file
response=PID_HI_P
%title of the plot
name=char(name(i));
%what is the step change and control parameters
M=M(i);
kc=kc(i);
tauI=tauI(i);
taud=taud(i);
%initial guess
x0= [0.02;50;10];
```

```
%rest of the code
t=(0:1:(length(response)-1))';
fun=@(x,t)PIDPTD(x,M,kc,tauI,taud,t);
[x,resnorm(i),residual] = lsqcurvefit(fun,x0,t,response);

fitted = PIDPTD(x,M,kc,tauI,taud,t);
PID_HI_P_fitted=fitted
figure
plot(t,response,'ok',t,fitted,'-r')

title(name)
legend('data', 'fitted','Location','northwest')
xlabel('time(s)')
ylabel('Response (deviation)')
annotation('textbox',[0.2 0.15 0.65 0.05],'String',['kc = ' num2str(kc,'%.1f') '  tauI = ' num2str(tauI,'%.1f') ...
    '  taud = ' num2str(taud,'%.1f') '  kp = ' num2str(x(1),'%.3f') '  taup = ' num2str(x(2),'%.3f') ...
    ' theta = ' num2str(x(3),'%.3f') ' resid = ' num2str(resnorm(i),'%.3f')])
print(name,'-dpng')
```