Process Dynamics and Simulations: PI Controller Design

Initial Equations:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_1 \exp\left(\frac{-E_1}{RT}\right) C_A$$

$$\frac{dC_B}{dt} = \frac{-F}{V} C_B + k_1 \exp\left(\frac{-E_1}{RT}\right) C_A - k_2 \exp\left(\frac{-E_2}{RT}\right) C_B$$

$$\frac{dC_C}{dt} = \frac{-F}{V} C_C + k_2 \exp\left(\frac{-E_2}{RT}\right) C_B$$

$$\frac{dT}{dt} = \frac{UA}{\rho C_p V}(T_j - T) + \frac{F}{V}(T_0 - T) + k_1 \exp\left(\frac{-E_1}{RT}\right) C_A \frac{\Delta H_1}{\rho C_p} - k_2 \exp\left(\frac{-E_2}{RT}\right) C_B \frac{\Delta H_2}{\rho C_p}$$

$$\frac{dT_j}{dt} = \frac{F_j}{V_j}(T_{j0} - T_j) + \frac{UA}{\rho_j C_{pj} V_j}(T - T_j)$$

To linearize this system of equations the Taylor Expansion Series will be utilized. The linearization will be performed around the given steady states in the problem following equation [1].

$$\frac{dF}{dt} = f(x, u, d)$$

$$\frac{dF}{dt} \sim \frac{dF}{dx}\bigg|(x - xs) + \frac{dF}{du}\bigg|(u - us) + \frac{dF}{dd}|(d - ds) \quad \text{Equation [1]}$$

Each of the (x-xs) variable can be re-written as $\bar{x}$ or the deviation variable of that term. The left-hand side of the equation can be re-written following: $\bar{F} = (F - Fs) \rightarrow F = \bar{F} + Fs$ and then the derivative of F will only be $\bar{F}$ as Fs is a constant. Below are the linearized equations.

$$\frac{d\overline{C_A}}{dt} = \left(-\frac{F}{V} - k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)\overline{C_A} - \left(k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right)\right)\bar{T}$$

$$\frac{d\overline{C_B}}{dt} = \left(k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)\overline{C_A} + \left(-\frac{F}{V} - k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)\overline{C_B} + \left(k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right) C_{AS} - k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)\bar{T}$$

$$\frac{d\overline{C_C}}{dt} = \left(k_2 \exp\left(\frac{-E_2}{RT_s}\right)\right)\overline{C_B} + \left(-\frac{F}{V}\right)\overline{C_C} + \left(k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)\bar{T}$$

$$\frac{d\bar{\bar{T}}}{dt} = \left(-k_1 \exp\left(\frac{-E_1}{RT_s}\right)\frac{\Delta H_1}{\rho C_p}\right)\overline{C_A} + \left(-k_2 \exp\left(\frac{-E_2}{RT_s}\right)\frac{\Delta H_2}{\rho C_p}\right)\overline{C_B} + \left(\frac{UA}{\rho C_p V}\right)\overline{T_J}$$
$$+ \left(-\frac{UA}{\rho C_p V} - \frac{F}{V} - k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right) C_{AS} - k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)\bar{T}$$

$$\frac{d\overline{T_J}}{dt} = \left(-\frac{F}{V} - \frac{UA}{\rho_j C_{pj} V_j}\right)\bar{\bar{T_J}} - \left(\frac{UA}{\rho_j C_{pj} V_j}\right)\bar{T}$$

Once these equations are linearized the method of determining stability is by forming a matrix of the coefficients of each species by column and row. This is visualized in Table 1 with the values calculated out using the constant table [A3] in the appendix.

	CA	CB	CC	T	Tj
CA	-0.66697	0	0	-0.00016	0
CB	0.000307	-0.66679	0	0.000163	0
CC	0	0.00012	-0.66667	3.25E-08	0
T	0.012183	0.007172	0	-1.057	0.396825
TJ	0	0	0	0.057416	-0.25742

Table 1: Matrix of coefficients of linearized system

Once this matrix is formed, using MATLAB and the "eig" function, the eigen values of the matrix may be found. If all the eigen values of this matrix are negative, then the designated steady state of the system will be stable. If there is any positive value observed, then the system will be unstable. For reference each steady state is shown in the appendix under [A1]. Both steady states were observed to be stable as shown in Table 2. Steady state two was chosen as the steady state for further calculations as there was not an unstable steady state to choose, however the method of further calculations allows for changing the steady state value within the MATLAB code easily [A2a].

SS1	SS2
-0.66667	-0.66667
-314.244	-1.08455
-195.961	-0.22987
-1.1054	-0.66679
-0.2296	-0.66697

Table 2: Eigen Values of each steady-state

The transfer function of the linearized system is derived from re-establishing the linearized function with the additional aspect of the input and disturbance depicted in the problem statement. To do this, the previously linearized forms will be the same, but the addition of these components will be shown. When previously derived, the linearized form had these partial derivatives occur, but before this section of the problem statement they were not depicted as an input or disturbance and therefore a potentially changing variable.

$$\frac{d\overline{C_A}}{dt} = \left(-\frac{F}{V} - k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)\overline{C_A} - \left(k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right)\right)\bar{T} + 0 * \bar{T}_{jo} + 0 * \bar{T}_o$$

$$\frac{d\overline{C_B}}{dt} = \left(k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)\overline{C_A} + \left(-\frac{F}{V} - k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)\overline{C_B} + \left(k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right) C_{AS} - k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)\bar{T} + 0 * \bar{T}_{jo} + 0 * \bar{T}_o$$

$$\frac{d\overline{C_C}}{dt} = \left(k_2 \exp\left(\frac{-E_2}{RT_s}\right)\right)\overline{C_B} + \left(-\frac{F}{V}\right)\overline{C_C} + \left(k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)\bar{T} + 0 * \bar{T}_{jo} + 0 * \bar{T}_o$$

$$\frac{d\bar{\bar{T}}}{dt} = \left(-k_1 \exp\left(\frac{-E_1}{RT_s}\right)\frac{\Delta H_1}{\rho C_p}\right)\overline{C_A} + \left(-k_2 \exp\left(\frac{-E_2}{RT_s}\right)\frac{\Delta H_2}{\rho C_p}\right)\overline{C_B} + \left(\frac{UA}{\rho C_p V}\right)\bar{T}_J + \left(-\frac{UA}{\rho C_p V} - \frac{F}{V} - k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right) C_{AS} - k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)\bar{T} + 0 * \bar{T}_{jo} + \frac{F}{V} * \bar{T}_o$$

$$\frac{d\bar{T}_J}{dt} = \left(-\frac{F_j}{V_j} - \frac{UA}{\rho_j C_{pj} V_j}\right)\bar{\bar{T}}_J - \left(\frac{UA}{\rho_j C_{pj} V_j}\right)\bar{T} + \frac{F_j}{V_j} * \bar{T}_{jo} + 0 * \bar{T}_o$$

To make the further calculations shown easier to observe, the coefficients in front of each of the deviation variables will be depicted as a constant following alphabetic order and negatives will be factored in. The dCc variables were given values but were not used later.

$$a = \left(-\frac{F}{V} - k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right) \quad b = -\left(k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right)\right) \quad c = \left(k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right) \quad d = \left(-\frac{F}{V} - k_1 \exp\left(\frac{-E_1}{RT_s}\right)\right)$$

$e = \left(k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right) C_{AS} - k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)$ $f = \left(k_2 \exp\left(\frac{-E_2}{RT_s}\right)\right)$ $g = -\frac{F}{V}$ $h = \left(k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)$

$i = \left(-k_1 \exp\left(\frac{-E_1}{RT_s}\right) \frac{\Delta H_1}{\rho C_p}\right)$ $j = \left(-k_2 \exp\left(\frac{-E_2}{RT_s}\right) \frac{\Delta H_2}{\rho C_p}\right)$

$k = \left(-\frac{UA}{\rho C_p V} - \frac{F}{V} - k_1 \frac{E_1}{RT_s} \exp\left(\frac{-E_1}{RT_s}\right) C_{AS} - k_2 \frac{E_2}{RT_s} \exp\left(\frac{-E_2}{RT_s}\right) C_{BS}\right)$ $l = \left(\frac{UA}{\rho C_p V}\right)$ $m = \frac{F}{V}$

$n = \left(-\frac{F_j}{V_j} - \frac{UA}{\rho_j C_{pj} V_j}\right)$ $o = \left(\frac{UA}{\rho_j C_{pj} V_j}\right)$ $p = \frac{F_j}{V_j}$

Now the previous equations will be re-written with these constants in place and the following rules to apply for the values that will be the state, input, and disturbance variables.

$$\frac{d\overline{C_A}}{dt} = a\overline{C_A} + b\bar{y} \qquad \frac{d\overline{C_B}}{dt} = c\overline{C_A} + d\overline{C_B} + e\bar{y}$$

$$\frac{d\bar{\bar{y}}}{dt} = i\overline{C_A} + j\overline{C_B} + k\bar{y} + l\overline{T_J} + m\bar{d} \qquad \frac{d\overline{T_J}}{dt} = n\overline{T_j} + o\bar{y} + p\bar{u}$$

Once this is complete the next step will be to take the Laplace Transforms of each equation so further derivation may take place within the s domain. The Laplace transform of the derivative and each coefficient and bar term is shown below. In the first, the derivative of the term at time t = 0 is equal to 0 as the linearization is performed around the steady state and therefore the change in m at time t = 0 is 0 as the m term is at the steady state.

$$L\left\{\frac{d\bar{m}}{dt}\right\} = s * \hat{\bar{m}}(s) - \frac{d\bar{m}}{dt}(t = 0)$$

$$L\{i * \bar{m}\} = i * \hat{\bar{m}}(s)$$

$$s\widehat{\overline{C_A}}(s) = a\widehat{\overline{C_A}}(s) + b\hat{\bar{y}}(s) \qquad s\widehat{\overline{C_B}}(s) = c\widehat{\overline{C_A}}(s) + d\widehat{\overline{C_B}}(s) + e\hat{\bar{y}}(s)$$

$$s\hat{\bar{y}}(s) = i\widehat{\overline{C_A}}(s) + j\widehat{\overline{C_B}}(s) + k\hat{\bar{y}}(s) + l\hat{\overline{T_J}}(s) + m\hat{\bar{d}}(s)$$

$$s\hat{\overline{T_J}}(s) = n\hat{\overline{T_J}}(s) + o\hat{\bar{y}}(s) + p\hat{\bar{u}}(s)$$

Once each of these formulas is made, next will be getting the $\hat{\bar{y}}$ function in terms of only $\hat{\bar{u}}$ and $\hat{\bar{d}}$. This will start with isolating the $\widehat{\overline{C_A}}$ equation as a function of $\hat{\bar{y}}$ and then plugging that into $\widehat{\overline{C_B}}$, solving this for $\hat{\bar{y}}$. Then $\hat{\overline{T_J}}$ will be solved for $\hat{\bar{y}}$ and $\hat{\bar{u}}$. Once all of these are solved, each equation is plugged into the $\hat{\bar{y}}$ Equation [2]. Then $\hat{\bar{y}}$ is isolated from the other side and the terms will be separated [3].

$$\widehat{\overline{C_A}}(s) = \frac{b\hat{\bar{y}}(s)}{(s-a)} \qquad \widehat{\overline{C_B}}(s) = \frac{c\left(\frac{b\hat{\bar{y}}(s)}{(s-a)}\right) + e\hat{\bar{y}}(s)}{(s-d)} \qquad \hat{\overline{T_J}}(s) = \frac{o\hat{\bar{y}}(s) + p\hat{\bar{u}}(s)}{(s-n)}$$

$$s\hat{\bar{y}}(s) = i\frac{b\hat{\bar{y}}(s)}{(s-a)} + j\frac{c\left(\frac{b\hat{\bar{y}}(s)}{(s-a)}\right) + e\hat{\bar{y}}(s)}{(s-d)} + k\hat{\bar{y}}(s) + l\frac{o\hat{\bar{y}}(s) + p\hat{\bar{u}}(s)}{(s-n)} + m\hat{\bar{d}}(s) \quad \text{Equation [2]}$$

$$s\hat{\hat{y}}(s) = \frac{ib\hat{\hat{y}}(s)}{(s-a)} + \frac{jcb\hat{\hat{y}}(s)}{(s-d)(s-a)} + \frac{je\hat{\hat{y}}(s)}{(s-a)} + k\hat{\hat{y}}(s) + \frac{lo\hat{\hat{y}}(s)}{(s-n)} + \frac{lp\hat{\hat{u}}(s)}{(s-n)} + m\hat{\hat{d}}(s) \quad \text{Equation [3]}$$

Then all $\hat{\hat{y}}$ terms will be brought over to the left-hand side of the equation to simplify and isolate the $\hat{\hat{y}}$. The terms will all be given a common denominator to make dividing back over to the right side easier [4]. Once divided over, the values of $m\hat{\hat{d}}(s)$ are ignored as these do not pertain to the closed loop transfer function any further. The $\hat{u}(s)$ terms will be combined with the divided over terms to now form the GP transfer function before simplification [5].

$$\hat{\hat{y}}(s)\left[\frac{s(s-a)(s-d)(s-n)-i\ \ (s-d)(s-n)-jcb(s-n)-je(s-n)(s-d)-k(s-a)(s-d)(s-n)-lo(s-a)(s-d)}{(s-a)(s-d)(s-n)}\right] = \frac{lp\hat{\hat{u}}(s)}{(s-n)} + m\hat{\hat{d}}(s) \quad \text{Equation [4]}$$

$$\hat{\hat{y}}(s) = \left[\frac{(s-a)(s-d)(s-n)}{s(s-a)(s-d)(s-n)-ib(s-d)(s-n)-jcb(s-n)-je(s-n)(s-d)-k(s-a)(s-d)(s-n)-lo(s-a)(s-d)}\right]\left(\frac{lp\hat{\hat{u}}(s)}{(s-n)} + m\hat{\hat{d}}(s)\right)$$

$$\hat{\hat{y}}(s) = \left[\frac{lp(s-a)(s-d)}{s(s-a)(s-d)(s-n)-i\ \ (s-d)(s-n)-jcb(s-n)-je(s-n)(s-d)-k(s-a)(s-d)(s-n)-lo(s-a)(s-d)}\right]\hat{\hat{u}}(s) \quad \text{Equation [5]}$$

Now all the of the terms in the denominator will be foiled out and the form the following equations, then all like terms will be combined following signage of the term on the highest order polynomial. Each equation below will be the terms foiled out from left to right.

$s^4 - (d + a + n)s^3 + (nd + an)s^2 - (and)s + 0$

$0s^4 - 0s^3 - (bi)s^2 + (bin + bid)s - bind$

$0s^4 - 0s^3 - 0s^2 - (jbc)s + jbcn$

$0s^4 - (k)s^3 + (kd + ka + kn)s^2 - (knd + kan + kad)s + kand$

$0s^4 - 0s^3 - (lo)s^2 + (loa + lod)s - load$

$s^4 - (d + a + n + k)s^3 + (nd + an + ad + bi - je + kd + ka + kn - lo)s^2 - (and - bin - bid + jbc - jen - jea + knd + kan + kad - loa - lod)s + (-bind + jbcn - jean + kand - load)$

From here to simplify for the rest of the calculations, each of these terms will be reduced to A,B,C,D with A starting as the coefficient to the s^3 term and following suit. Then the simplified GP transfer function will be complete as the coefficient of the input term in Equation 6 and shown as Equation 7.

$$\hat{\hat{y}}(s) = \frac{lp(s-a)(s-d)}{s^4-As^3+Bs^2-Cs+D}\hat{\hat{u}}(s) + \frac{m(s-a)(s-d)(s-n)}{s^4-As^3+Bs^2-Cs+D}\hat{\hat{d}}(s) \qquad \text{Equation [6]}$$

$$Gp = \frac{lp(s-a)(s-d)}{s^4-As^3+Bs^2-Cs+D} \qquad \text{Equation [7]}$$

Now a block diagram, Figure 1 may be made to represent the system and develop the generalized equation for the system using the block diagram, also it is assumed Gs = 1 in this case.

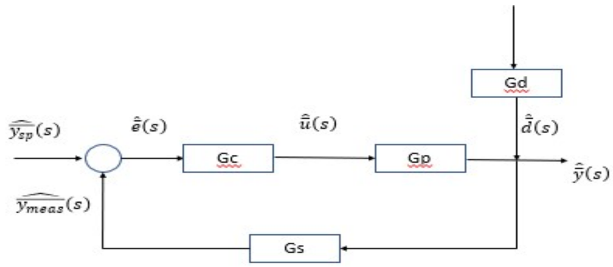

Figure 1: Block Diagram of System

$\hat{\bar{y}}(s) = Gp\hat{\bar{u}}(s) + Gd\hat{\bar{d}}(s)$

$\hat{\bar{u}}(s) = Gc\hat{\bar{e}}(s)$

$\hat{\bar{e}}(s) = \widehat{y_{sp}}(s) - \widehat{y_{meas}}(s)$

$\hat{\bar{y}}(s) = GpGc(\widehat{y_{sp}}(s) - \widehat{y_{meas}}(s)) + Gd\hat{\bar{d}}(s)$ $\qquad \widehat{y_{meas}}(s) = \hat{\bar{y}}(s)Gs$

As previously stated Gs =1 so the final equation we have before solving for $\hat{\bar{y}}(s)$ is Equation [8]. Then the $\hat{\bar{y}}(s)$ will be moved to the left-hand side of the equation and $\hat{\bar{y}}(s)$ will be isolated to give Equation [9].

$$\hat{\bar{y}}(s) = GpGc\left(\widehat{y_{sp}}(s) - \hat{\bar{y}}(s)\right) + Gd\hat{\bar{d}}(s) \qquad \text{Equation [8]}$$

$$\hat{\bar{y}}(s) = \frac{GpGc(\widehat{y_{sp}}(s))}{1+Gp} + \frac{Gd\hat{\bar{d}}(s)}{1+G} \qquad \text{Equation [9]}$$

Then the transfer function from Equation 7 will be plugged into the place of Gp, the Gd aspect will be ignored again. For the purpose of creating the closed loop transfer function and finding the range our Kc values will be effective, a P controller will be assumed and Gc will be replaced with only Kc in Equation 10.

$$\hat{\bar{y}}(s) = \frac{\frac{lp(s-a)(s-d)}{s^4-As^3+Bs^2-Cs+D}Kc}{1+\frac{lp(s-a)(s-d)}{s^4-As^3+Bs^2-Cs+D}Kc}(\widehat{y_{sp}}(s)) \qquad \text{Equation [10]}$$

A common denominator will be made for the terms in the total denominator of the total term allowing the multiplication of the reciprocal. This multiplication will cancel out the same denominator that is in the total numerator of the term. This will result in Equation 11. Then the total denominator will be simplified to create coefficients to check for stability in Equation 12.

$$\hat{\bar{y}}(s) = \frac{lp(s-a)(s-d)Kc}{s^4-As^3+Bs^2-Cs+D+lp(s-a)(s-d)Kc}(\widehat{y_{sp}}(s)) \qquad \text{Equation [11]}$$

$$\hat{\bar{y}}(s) = \frac{lp(s-a)(s-d)Kc}{s^4-As^3+(lpKc+B)s^2-(Kclpa+Kclpd \quad)s+(Kclpad+D)}(\widehat{y_{sp}}(s)) \qquad \text{Equation [12]}$$

Once at this stage, to test for stability all coefficients will be assumed to be positive so negatives will be multiplied in. Each coefficient will be set to be greater than 1 and solved to find the Kc value that will make it positive. This was performed in MATLAB as the constants were not calculated by hand. This will ensure accurate ranges as the terms may change based on the steady state chosen. The method to do this if the values were calculated by hand would be to set the terms to be greater than 0, isolate the Kc value and ensure signs are correct to keep the inequality symbol indicating the correct value of Kc either being greater than the number or less than. The method through MATLAB was to find the zeros to each term and then test those values by adding and subtracting a small amount and testing these values in the equations in [A2b]. By hand, the method would be to find these values, assume then that all the coefficients would be positive and then utilize a Reuth Array to determine the final Kc range that will make all coefficients positive and the first column of the array positive as well. The Reuth Array is described with Equation 14 as the standard polynomial structure. Table 3 is the general structure of the Reuth Array with 5 terms and Table 4 is this projects structure. To simplify the Reuth Array, each coefficient within Equation 13 will be assigned a value A' B' C' D'.

$$\hat{\hat{y}}(s) = \frac{lp(s-a)(s-d)Kc}{s^4+(-A)s^3+(lpKc+B)s^2+(-Kclpa-Kclpd-C)s+(Kclpad+D)}\left(\widehat{\overline{y_{sp}}}(s)\right) \qquad \text{Equation [13]}$$

$$a_0 s^n + a_1 s^{n-1} \dots a_{n-1}s + a_n \qquad \text{Equation [14]}$$

a_{even}	a_0	a_2	a_4
a_{odd}	a_1	a_3	$a_5 = 0$
b	$\frac{a_1a_2 - a_3a_0}{a_1}$	$\frac{a_1a_4 - a_5a_0}{a_1} = a_4$	0
c	$\frac{\left(\frac{a_1a_2 - a_3a_0}{a_1}a_3 - a_4a_1\right)}{\frac{a_1a_2 - a_3a_0}{a_1}}$	$\frac{\left(\frac{a_1a_2 - a_3a_0}{a_1}a_5 - 0a_1\right)}{\frac{a_1a_2 - a_3a_0}{a_1}} = 0$	0
d	$\frac{\frac{\left(\frac{a_1a_2 - a_3a_0}{a_1}a_3 - a_4a_1\right)}{\frac{a_1a_2 - a_3a_0}{a_1}}a_4 - 0}{\frac{\left(\frac{a_1a_2 - a_3a_0}{a_1}a_3 - a_4a_1\right)}{\frac{a_1a_2 - a_3a_0}{a_1}}} = a_4$	0	0

Table 3: Reuth Array of generalized polynomial

a_{even}	1	B'	D'
a_{odd}	A'	C'	$a_5 = 0$
b	$\frac{A'B' - C'1}{A'}$	$\frac{A'D' - 0*1}{A'} = D'$	0
c	$\frac{\left(\frac{A'B' - C'}{A'}C' - D'A'\right)}{\frac{A'B' - C'}{A'}}$	$\frac{\left(\frac{A'B' - C'1}{A'}0 - 0a_1\right)}{\frac{A'B' - C'1}{A'}} = 0$	0
d	$\frac{\frac{\left(\frac{A'B' - C'}{A'}C' - D'A'\right)}{\frac{A'B' - C'1}{A'}}D' - 0}{\frac{\left(\frac{A'B' - C'}{A'}C' - D'A'\right)}{\frac{A'B' - C'}{A'}}} = D'$	0	0

Table 4: Reuth Array of Project Polynomial

Once these values are found, the prime terms may be replaced again and the equations may be placed into MATLAB to solve the zeros using the zero function for b and the solve function for c as there will be a squared Kc term, d is the same as the D' value so it will be repeated, but it is ran through the zero function as well for continuity [A2b]. Once these terms are determined using the code, they are tested against all the equations found and a matrix is created with rows being the equations and columns being the tested values. The odd columns will be the less than values and even will be the greater than values. Table 5 displays the determined Kc values of steady-state two as well as their respective test columns.

Kc values	-30.83417484		-8.663587526		-3.141253619		-53.33100099		-19.79262264		-19.78640413		-3.141253619	
Kc1	-0.00794	0.007937	1.751634	1.767507	2.189914	2.205787	-1.7934	-1.77753	0.876234	0.876393	0.876014	0.877601	2.189914	2.205787
Kc2	-2.35743	-2.33626	-0.01059	0.010585	0.573976	0.595147	-4.73881	-4.71764	-1.17816	-1.17795	-1.17845	-1.17634	0.573976	0.595147
Kc3	-0.98098	-0.97392	-0.19845	-0.19139	-0.00353	0.00353	-1.77503	-1.76797	-0.58776	-0.58769	-0.58786	-0.58715	-0.00353	0.00353
Kc4	0.882272	0.89015	1.755631	1.76351	1.973171	1.981049	-0.00394	0.003939	1.321128	1.321207	1.321019	1.321806	1.973171	1.981049
Kc5a	0.587022	0.561131	0.288749	0.297982	0.578713	0.590429	-1198	1183.801	2.28E-08	-1.65E-08	1.19E-07	5.12E-07	0.578713	0.590429
Kc5b	0.587022	0.561131	0.288749	0.297982	0.578713	0.590429	-1198	1183.801	2.28E-08	-1.65E-08	1.19E-07	5.12E-07	0.578713	0.590429
Kc6	-0.98098	-0.97392	-0.19845	-0.19139	-0.00353	0.00353	-1.77503	-1.76797	-0.58776	-0.58769	-0.58786	-0.58715	-0.00353	0.00353

Table 5: Kc values with respective test columns

As seen in Table 5, the only values that deliver a positive column meaning all terms will be positive is column 6 and 14, with 14 being a duplicate of 6. This indicates that for steady-state two, the only restriction to observe stability is $Kc > -3.1413$ with no upper limit. This however does not mean that a large Kc would be preferable as this will make the controller very aggressive in its proportional response.

For the design of the PI controller, the use of the Explicit Euler Method will be used within MATLAB [A2c] to view how the system is changing for the linearized forms. Initially a P controller will be assumed so the TauI addition of the ubar term will be commented out. From there values of Kc will be guessed corresponding to the range determined previously. As shown in Figure 2, an initial guess of Kc = 20 was used, the behavior exhibited is an overshoot shown by surpassing the set point even when offset is still present in the system. The offset is caused by the concept from Equation [15].

$$\frac{GpGc}{1+GpG} = \frac{GpKc}{1+GpKc} \rightarrow GpKc \neq 1 + GpKc \quad \text{Equation [15]}$$

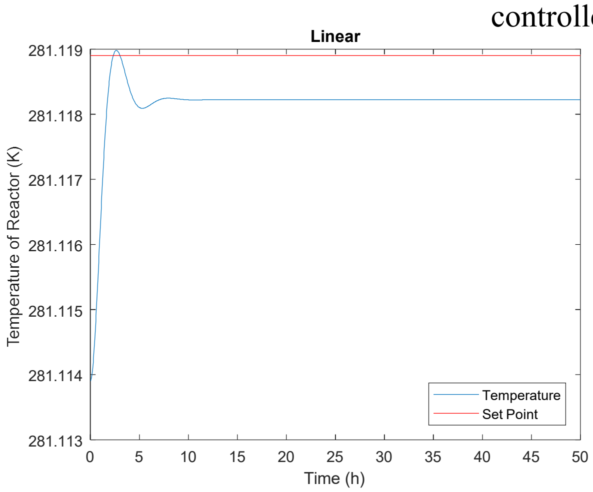

Figure 2: Kc = 20

Due to the numerator never equaling the denominator, value of 1 may never be reached and therefore the system may never stabilize at the set point. If the Kc value is large enough the initial change my be so much large than needed that it may pass the new set-point momentarily, but once it stabilizes will remain below the set-point.

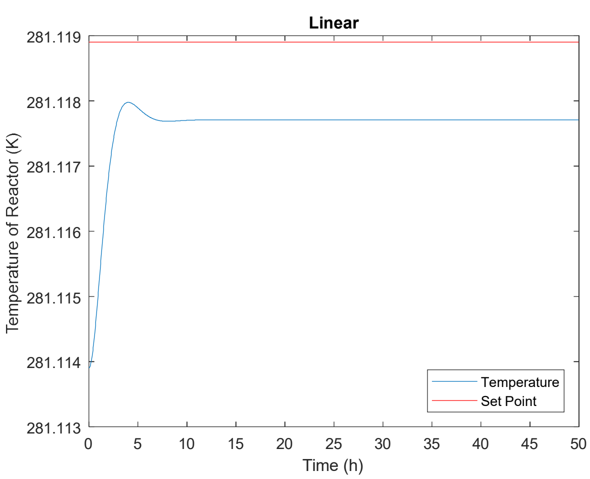

Figure 3: Kc = 10

The next attempt then is made with a Kc guess of 10 to lower the overshoot that is observed. It is assumed though that once the integral aspect of the controller is added the system will stabilize around the set-point, therefore the Kc value will need to be lowered further to prevent this overshoot.

The next guess was made at Kc = 2 because the overshoot was still occurring, so an over correction is made. This also shows what happens when a lower Kc value than optimal is chosen, the system does meet the offset set-point gradually which is trying to be achieved, but comparing Figure 4 and Figure 3, the offset is much greater in Figure 4 than Figure 3. This means that the Kc is severely overdamping the response. Therefore, the next guess should be in between the previous two.

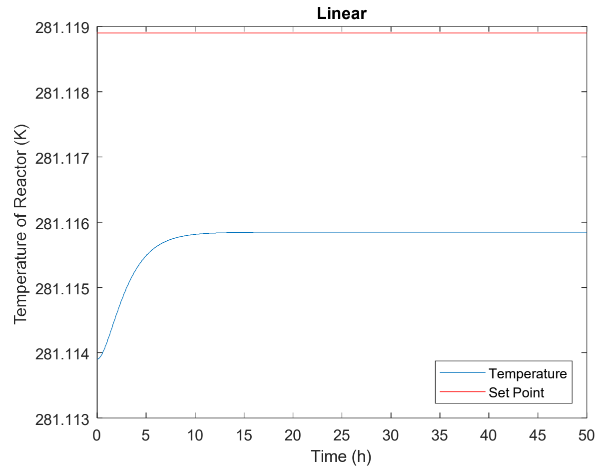

Figure 4: Kc = 2

The final guess is shown in Figure 5, a Kc value of 4 was guessed. The offset can be seen to decrease from the previous guess but not be as small as the first two guesses. However, the gradual approach is maintained in this value. This value may be assumed as the good initial setting once the PI controller tuning commences as it achieves the two important aspects of minimizing the offset and having little overshoot.

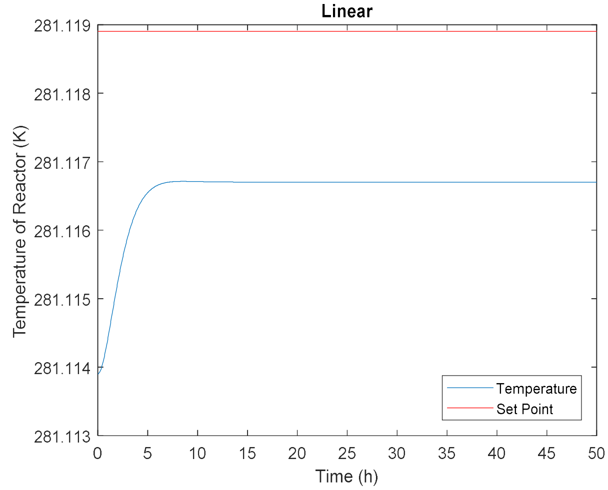

Figure 5: Kc = 4

From this point on the integral response will be implemented, this allows the system controller to overcome the offset of the P control system due to the following Equation 16.

$$\frac{GpGc}{1+GpG} = \frac{GpKc\left(1+\frac{1}{\tau_I s}\right)}{1+Gp\left(1+\frac{1}{\tau_I s}\right)} \rightarrow \frac{\left(\frac{GpKc(\tau_I s)+GpK}{\tau_I s}\right)}{\left(\frac{\tau_I s+Gp\quad(\tau_I s)+GpKc}{\tau_I s}\right)} \rightarrow \frac{GpKc(\tau_I s)+GpKc}{\tau_I s+GpKc(\tau_I s)+GpKc}$$

Equation [16]

The final form of Equation 16 if the τ_I term is minimized as much as possible and especially as it approaches 0, every term in the equation would be minimized except for the GpKc term in both the numerator and denominator which would then be 1. This would mean that with the integral response, the system should be able to achieve the set-point.

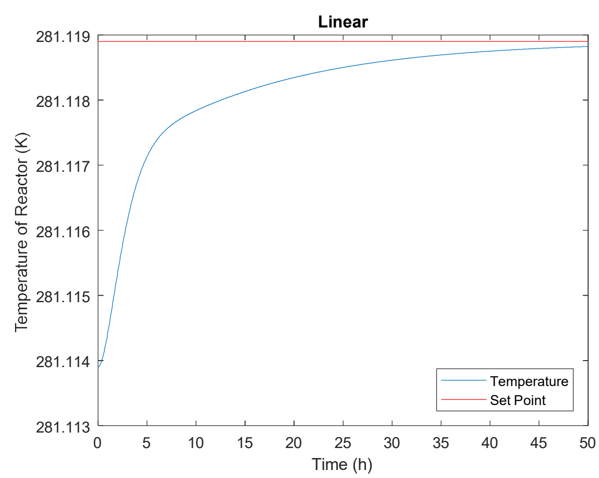

Figure 6: Kc = 4 and τ_I = 10

The process to tune the PI controller is to assume the Kc value of the P controller will hold true for initial τ_I guesses. The first guess is shown in Figure 6, and the new set point of .005 is achieved with no offset. This is achieved after 50 hours which is very poor response time and it can be seen right at the start of the plot that the slope is increasing meaning there is lag time to the response. This is understood that from the initial point the slope should be decreasing

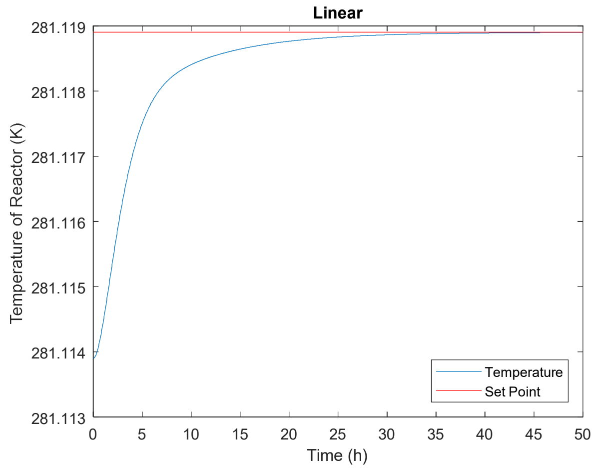

Figure 7: Kc = 4 and τ_I = 6

as it approaches the set-point as see throughout the rest of the figure. These conditions indicate that the system is now being overdamped by the τ_I value it therefore needs to be decreased.

The next guess is shown in Figure 7. This new τ_I guess of 6 significantly reduced the time to achieve set-point. The slight increase in slope is still present at the beginning and the time of set-point achieving is still too high. The τ_I should therefore be lowered once again or the Kc value may be increased instead, both examples are shown in Figures 8 and 9 respectively. As seen in Figure 8, decreasing the τ_I too far results in overshoot once again which unless the process warrants overshoot, this should be avoided. Figure 9 shows that

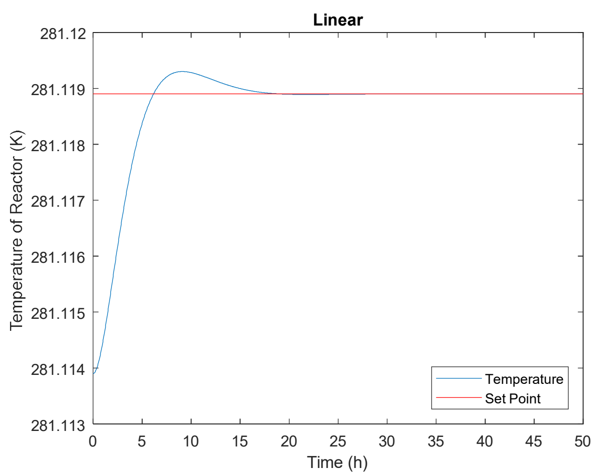

Figure 8: Kc = 4 and τ_I = 3

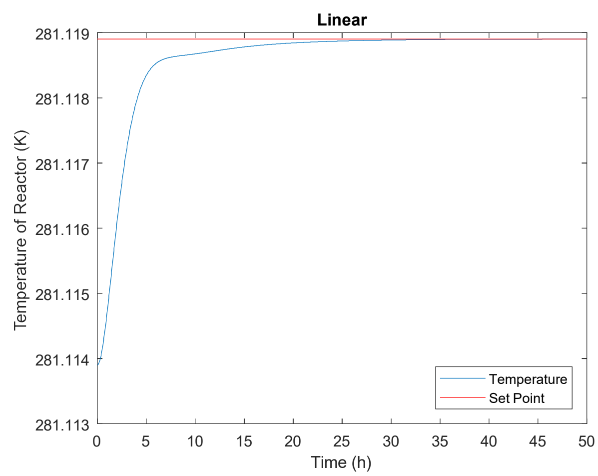

Figure 9: Kc = 6 and τ_I = 6

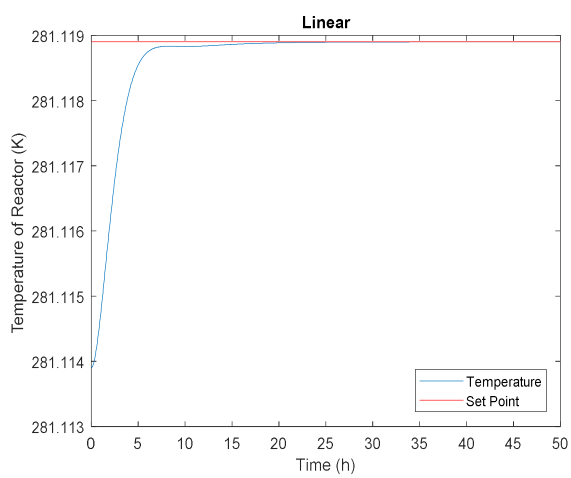

Figure 10: Kc = 6 and τ_I = 5

increasing the Kc value to 6 made the proportional response more aggressive and then approach the set point faster, the τ_I can be then decreased with the Kc remaining the same to see if it can approach from here without overshooting the set point or creating oscillations. As seen in Figure 10 the approach is very close with little oscillations. To get as close as possible, a little overshoot will be allowed to allow oscillation to reach the set-point. To do this, Kc will be increased slightly while τ_I is decreased slightly, the final result is shown in Figure 11.

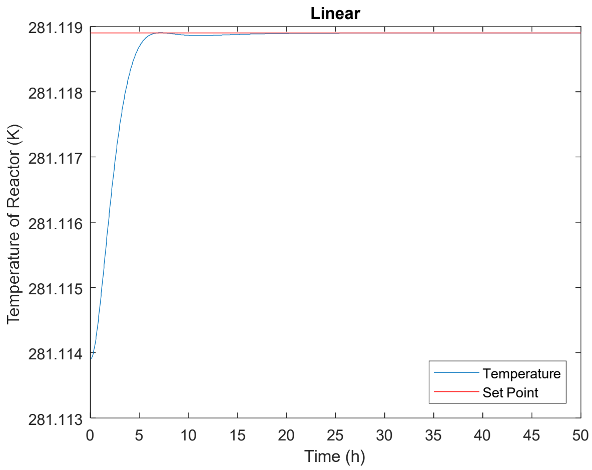

Figure 11: Kc = 6.35 and τ_I = 4.8

Figure 11 shows the final Kc and τ_I values that will be used throughout the rest of the project. These achieve a rapid approach due to the larger Kc value of 6.35, and the τ_I value has little overshoot with low oscillation away from the set point before being consistently at the set-point.

Next a set-point change of .005 will be applied and the controller will be applied to the non-linear system as well to compare how each system will react to the tuning. Figure 12 and 13 compare the linear and non-linear temperature output and jacket inlet temperature input respectively.

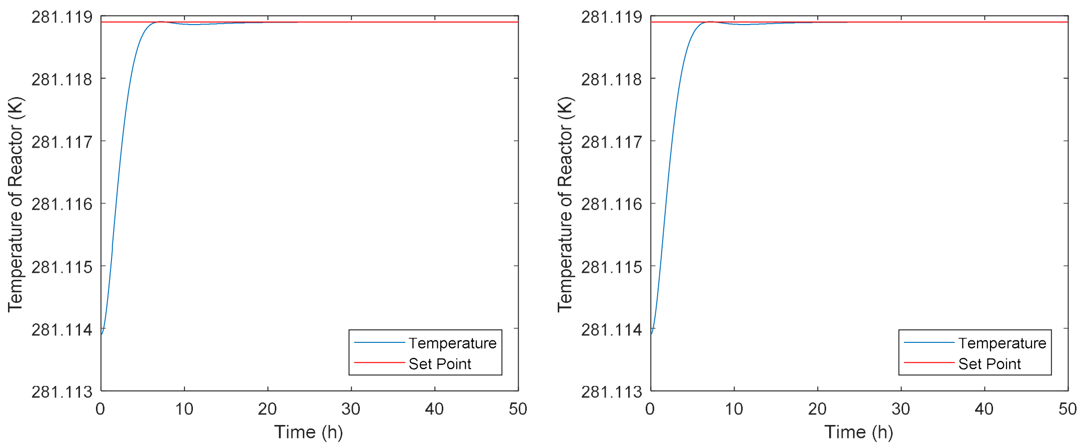

Figure 12: System Temperature Output Linear(left) Non-linear(right) with set point change of .005

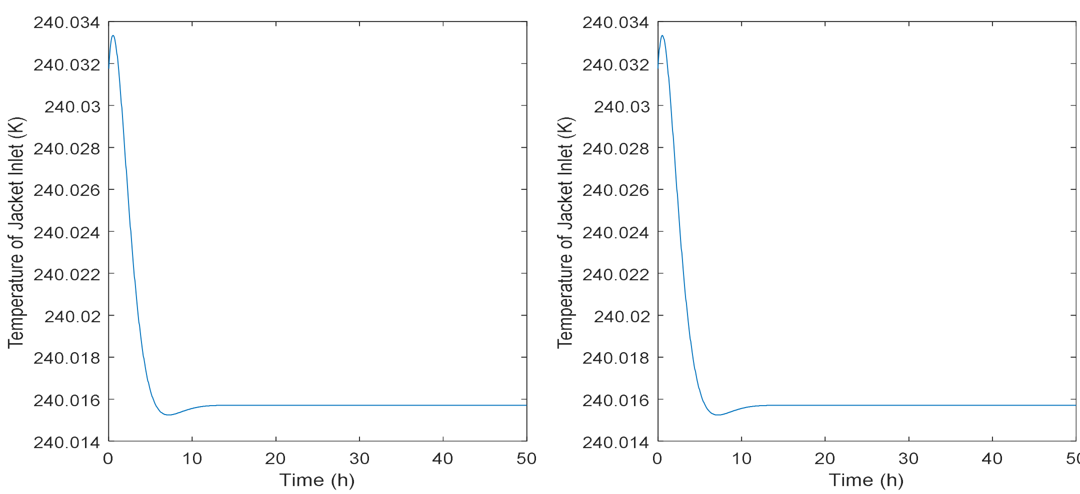

Figure 13: Jacket Inlet Temperature Linear(left) Non-LInear(right) with set point change of .005

The responses for both aspects with a set-point change of .005 are essentially the same. There are most likely minute differences throughout the graphs that may be shown by inspecting each plot y-axis variables. The same can be said about the set-point change of 1, as each of these is only a small change from the steady state of the system, there is not a very observed difference between the graphs on Figures 14 and 15. This is explained by the concept of the linearized system, being linearized around the steady-state of the system means that it will have an accurate modeling of the system so long as any set-point changes occur around the steady-state. If a large set-point change were to occur for example, that would cause a noticeable difference between the graphs. A noticeable difference is seen in the ranges of the jacket input temperature graphs of Figures 13 and 15. This is because of the magnitude difference between the set-point changes of .005 and 1. Because the set point change is 1, the change in the jacket inlet temperature from the steady state value will be explicitly what the proportional response will be. Another curious aspect is that the jacket inlet temperature increases at the beginning then decreases. This is most likely due to an initial calculation showing that it needs to increase the temperature of the coolant stream to support the increasing system temperature, but as the system temperature approaches the new set-point, it quickly recalculates to lower the coolant inlet temperature so it will not overshoot.

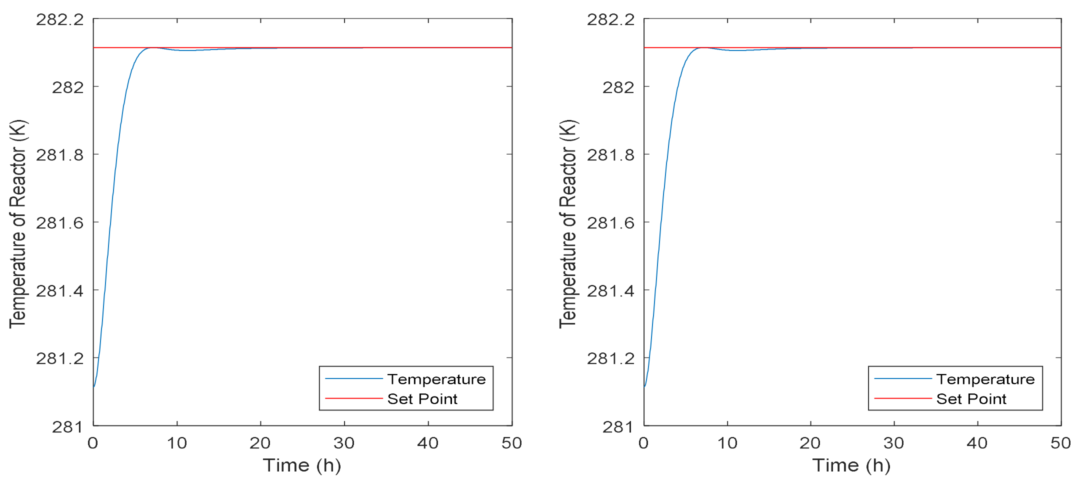

Figure 14: Linear(L) and Non-linear(R) Temperature Output with Set-Point change of 1

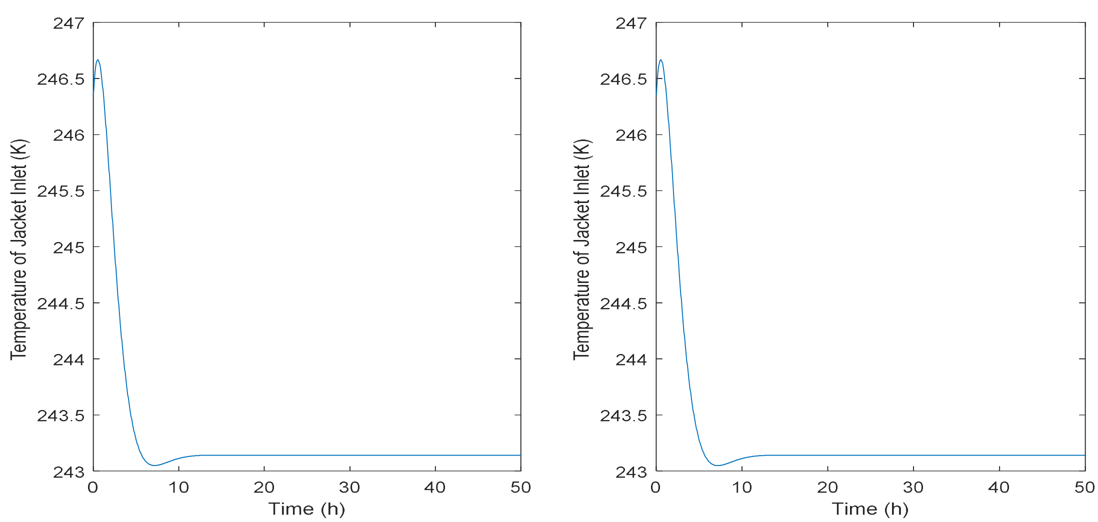

Figure 15: Linear(L) and Non-linear(R) Jacket Inlet Temperature with Set-Point change of 1

Then the system will run with a disturbance affecting the system instead of a set-point change. The set point change from the previous sections will be set to 0. Each of the disturbances will be present within the linearized system now that there is a unit step change with the disturbance. Before the disturbance was not showing an affect on the linearized system, as there was no change from the steady state disturbance value. This is what made the disturbance deviation variable in the equations is equal to 0.

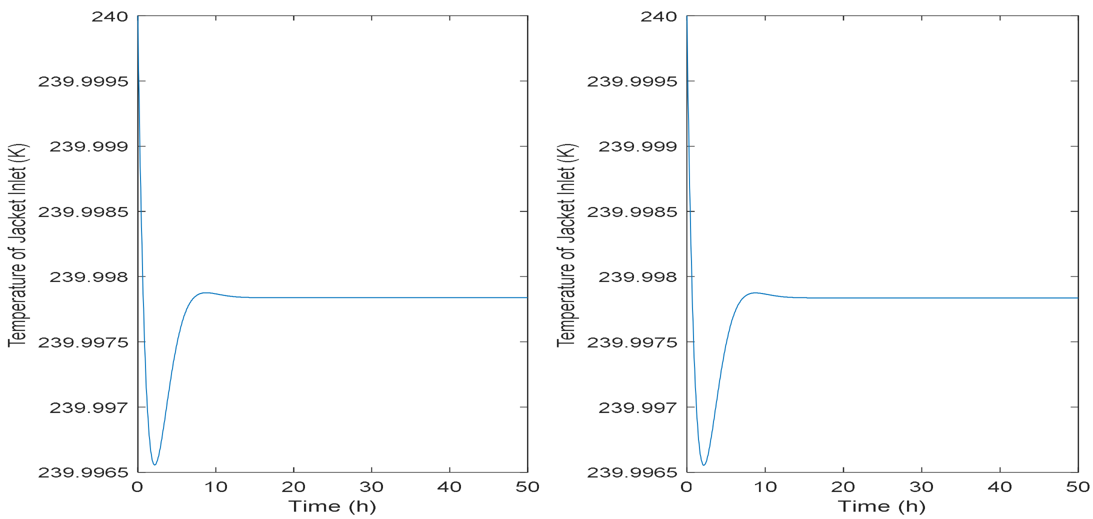

Figure 16: Temperature Ouput Linear(left) vs Non-Linear(right) Disturbance unit step .005

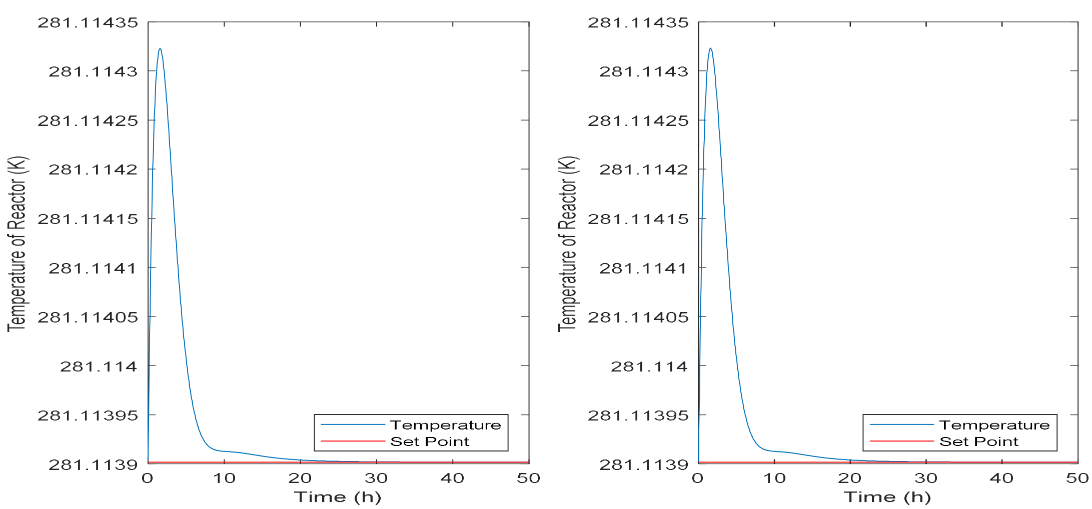

Figure 17: Jacket Input Temperature Linear(left) vs Non-Linear(right) Disturbance unit step .005

Once again, a small change from the steady-state even through the disturbance resulted in nearly identical graphs of both linear and non-linear. From these graphs the temperature of the system is corrected relatively fast, but once it gets near the set-point it slows until around 25 hours. This is bad in respect to disturbance rejection capabilities, while it responds quickly initially, the response is underdamped in that it does not re-achieve set-point in a good amount of time. The change of the jacket inlet temperatures for both linear and non-linear show an overshoot response but the significant figures show it is not a large overshoot at all, this may be indicative of why the outlet temperature effect takes so long as the controller determines the inlet temperature to be at steady state very fast instead of gradually allowing it to lower to enact faster change on the system.

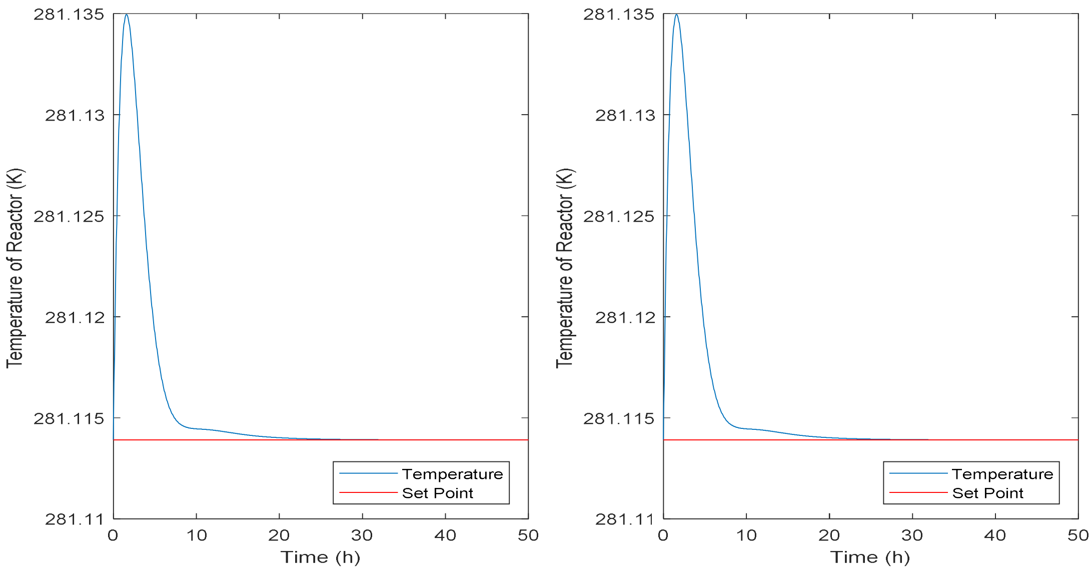

Figure 18: Temperature Ouput Linear(left) vs Non-Linear(right) Disturbance unit step .01

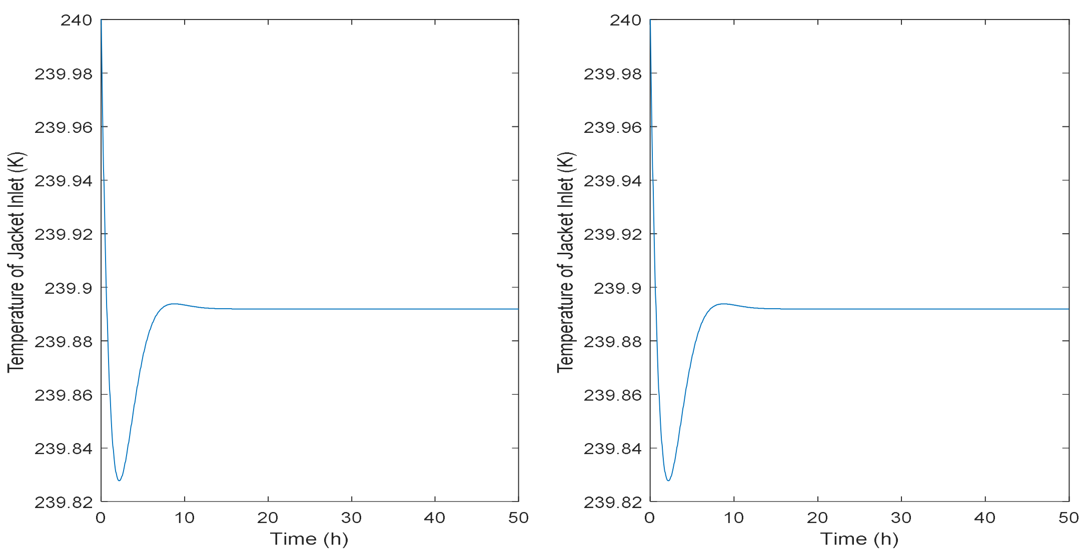

Figure 19: Jacket Input Temperature Linear(left) vs Non-Linear(right) Disturbance unit step .01

Figures 18 and 19 show the changes in temperature and jacket inlet temperature, respectively, with a disturbance unit step of .01. As can be seen again, the graphs between linear and non-linear are very similar. One noticeable difference between the change in the disturbances, is that it appears as if Figure 18 approaches the steady-state slightly faster than the smaller disturbance. The only reason this may happen is that the input response change was larger than the step change. This means that the controller responds better to larger disturbances by changing the input temperature more than when a smaller disturbance occurs. This most likely means that the proportionality with respect to disturbances is optimized around a larger disturbance. This is oddly dissimilar to the set-point change figures which appear to be around the same value even with a small change from the steady-state.

Appendix

C_A	C_B	C_C	T	T_j
0.0226301208683143	0.0150450773785251	6.96232480175316	753.722140349192	354.584864018407
6.99677782598635	0.00322159294563358	5.81073225911324e-07	281.113901760693	249.17038700984

A1: Given Steady State Values

A2: MATLAB Code for all sections

A2a: MATLAB code for Part 1

```
clear;close all;clc

%% Constants

UA = 1200;
RHO = 1200;
CP = .42;
RHOJ = 1000;
CPJ = 4.18;
K1 = 6e5;
K2 = 2e6;
E1 = 5e4;
E2 = 5.5e4;
CAO = 7;
T0 = 300;
F = 4;
FJ = 1;
TJ0 = 240;
DH1 = -2e4;
DH2 = -3e4;
V = 6;
VJ = 5;
R = 8.314;

%% Steady-State Guesses
%
% CAS = 0.0226301208683143;
% CBS = 0.0150450773785251;
% CCS = 6.96232480175316;
% TS = 753.722140349192;
% TJS = 354.584864018407;

CAS = 6.99677782598635;
CBS = 0.00322159294563358;
CCS = 5.81073225911324e-07;
TS = 281.113901760693;
TJS = 249.17038700984;

%% Vars for Matrix

Av = F/V; %Fo/V
Bv = K1*exp(-E1/(R*TS));
Cv = K2*exp(-E2/(R*TS));
Dv = E1/(R*TS^2);
```

```
Ev = E2/(R*TS^2);
Fv = UA/(RHO*CP*V);
Gv = UA/(RHOJ*CPJ*VJ);
Hv = FJ/VJ;
Iv = DH1/(RHO*CP);
Jv = DH2/(RHO*CP);

%% PART 1 Matrix vectors, Matrix and eigen vector
M1 = [(-Av-Bv),0,0,(-Bv*CAS*Dv),0];
M2 = [(Bv),(-Av-Cv),0,(Bv*Dv*CAS-Cv*Ev*CBS),0];
M3 = [0,Cv,-Av,(Cv*Ev*CBS),0];
M4 = [(-Bv*Iv),(-Cv*Jv),0,(-Fv-Av-Bv*Dv*CAS*Iv-Cv*Ev*CBS*Jv),Fv];
M5 = [0,0,0,Gv,(-Hv-Gv)];

M = [M1;M2;M3;M4;M5];

J = eig(M);
```

A2b: MATLAB code for part 3

```
%% Part 3 to find Kc vals and test potential issue values. TEST matrix will
have 14 columns
% and 7 rows each column pair corresponds to a Kc value and each row, one
% of the test equations. odd columns are KC < val even are KC > Val. So I
% have to find a column of all positive values.
a = M1(1,1);
b = M1(1,4);
c = M2(1,1);
d = M2(1,2);
e = M2(1,4);
i = M4(1,1);
j = M4(1,2);
k = M4(1,4);
l = M4(1,5);
m = Av;
n = M5(1,5);
o = M5(1,4);
p = Hv;

A = d + a + n + k;
B = n*d+a*n+a*d-b*i-j*e+k*d+k*a+k*n-l*o;
C = (a*n*d-b*i*n-b*i*d+j*b*c-j*e*n-j*e*a+k*n*d+k*a*n+k*a*d-l*o*a-l*o*d);
D = -b*i*n*d + j*b*c*n - j*e*a*n + k*a*n*d - l*o*a*d;

Kc1 = -B/(l*p)
Kc2 = -C/(l*p*a+l*p*d)
Kc3 = -D/(l*p*a*d)

x0 = [-1e10 1e10];
syms x1
fun4 = @(x) ((-A*(B+x*l*p)-(-x*l*p*a-x*l*p*d-C))/-A);
fun5 = (((-A*(B+x1*l*p)-(-x1*l*p*a-x1*l*p*d-C))/-A).*(-x1*l*p*a-x1*l*p*d-C)-
(-A)*(D+x1*l*p*a*d))./((-A*(B+x1*l*p)-(-x1*l*p*a-x1*l*p*d-C))/-A) == 0;
fun6 = @(x) (D+x*l*p*a*d);
Kc4 = fzero(fun4,x0)
Kc55 = solve(fun5,x1);
Kc5 = double(Kc55)'
```

```
Kc6 = fzero(fun6,x0)

KCT = [Kc1-.1,Kc1+.1;Kc2-.1,Kc2+.1;Kc3-.1,Kc3+.1;Kc4-.1,Kc4+.1;Kc5(1,1)-
.001,Kc5(1,1)+.001;Kc5(1,2)-.01,Kc5(1,2)+.01;Kc6-.1,Kc6+.1];
KCTT = [Kc1;Kc2;Kc3;Kc4;Kc5(1);Kc5(2);Kc6];

for jj = 1:7
    for ii = 1:2
        TEST(1,ii+2*(jj-1)) = KCT(jj,ii)*l*p +B;
        TEST(2,ii+2*(jj-1)) = KCT(jj,ii)*(-l*p*a-l*p*d)-C;
        TEST(3,ii+2*(jj-1)) = KCT(jj,ii)*(l*p*a*d)+D;
        TEST(4,ii+2*(jj-1)) = ((-A*(B+KCT(jj,ii)*l*p)-(-KCT(jj,ii)*l*p*a-
      KCT(jj,ii)*l*p*d-C))/-A);
        TEST(5,ii+2*(jj-1)) = (((-A*(B+KCT(jj,ii)*l*p)-(-KCT(jj,ii)*l*p*a-
KCT(jj,ii)*l*p*d-C))/-A)*(-KCT(jj,ii)*l*p*a-KCT(jj,ii)*l*p*d-C)-(-
A)*(D+KCT(jj,ii)*l*p*a*d))/((-A*(B+KCT(jj,ii)*l*p)-(-KCT(jj,ii)*l*p*a-
KCT(jj,ii)*l*p*d-C))/-A);
        TEST(6,ii+2*(jj-1)) = (((-A*(B+KCT(jj,ii)*l*p)-(-KCT(jj,ii)*l*p*a-
KCT(jj,ii)*l*p*d-C))/-A)*(-KCT(jj,ii)*l*p*a-KCT(jj,ii)*l*p*d-C)-(-
A)*(D+KCT(jj,ii)*l*p*a*d))/((-A*(B+KCT(jj,ii)*l*p)-(-KCT(jj,ii)*l*p*a-
KCT(jj,ii)*l*p*d-C))/-A);
        TEST(7,ii+2*(jj-1)) = KCT(jj,ii)*(l*p*a*d)+D;
    end
end
```

A2c: MATLAB code for part 4/5

```
%% Part 4/5 design the PI controller aspect by making ubar
Kc*e(s)*(1+1/taui*s
time = 0;
h = 1e-5;
finaltime = 50;

INTSTEP = floor(finaltime/h +.5);

YMATL = zeros(INTSTEP,1);
ZMATL = zeros(INTSTEP,1);
CAMATL = zeros(INTSTEP,1);
CBMATL = zeros(INTSTEP,1);
TIMEMAT = zeros(INTSTEP,1);
TJMATL = zeros(INTSTEP,1);
UMATL = zeros(INTSTEP,1);

TMATNL = zeros(INTSTEP,1);
ZMATNL = zeros(INTSTEP,1);
CAMATNL = zeros(INTSTEP,1);
CBMATNL = zeros(INTSTEP,1);
TJMATNL = zeros(INTSTEP,1);
UMATNL = zeros(INTSTEP,1);

%arbitrarily changed steady state values using random addition

TMATNL(1,1) = TS;
CAMATNL(1,1) = CAS;
CBMATNL(1,1) = CBS;
TJMATNL(1,1) = TJS;
```

```
%guessed Kc value between range and guess tauI value to find stability.
KCL = 6.35;
TAUIL = 4.8;
KCNL = 6.35;
TAUINL = 4.8;
SPC = 0;
YSPNL = TS + SPC;
YSPL = SPC;
DISTSPC = .05;
DISTNL = T0+DISTSPC;
DISTL = DISTSPC;
UMATL(1,1) = KCL*(YSPL-YMATL(1,1));
UMATNL(1,1) = KCL*(YSPNL-TMATNL(1,1))+TJ0;
for ii = 1:(INTSTEP-1)


    %LINEARIZED SYSTEM
    UMATL(ii+1,1) = KCL*(YSPL-YMATL(ii,1)) + (KCL/TAUIL)*ZMATL(ii,1);
    CAMATL(ii+1,1) = CAMATL(ii,1) + h*(a*CAMATL(ii,1) + b*YMATL(ii,1));
    CBMATL(ii+1,1) = CBMATL(ii,1) + h*(c*CAMATL(ii,1) + d*CBMATL(ii,1) +
e*YMATL(ii,1));
    YMATL(ii+1,1) = YMATL(ii,1) + h*(i*CAMATL(ii,1) + j*CBMATL(ii,1) +
k*YMATL(ii,1) + l*TJMATL(ii,1) + m*DISTL);
    TJMATL(ii+1,1) = TJMATL(ii,1) + h*(n*TJMATL(ii,1) + o*YMATL(ii,1) +
p*UMATL(ii,1));
    ZMATL(ii+1,1) = ZMATL(ii,1) + h*(YSPL-YMATL(ii,1));

    %nonlinear system
    UMATNL(ii+1,1) = KCNL*(YSPNL-TMATNL(ii,1)) + (KCNL/TAUINL)*ZMATNL(ii,1) +
TJ0;
    CAMATNL(ii+1,1) = CAMATNL(ii,1) + h*(Av*(CAO-CAMATNL(ii,1)) -
CAMATNL(ii,1)*K1*exp(-E1/(R*TMATNL(ii,1))));
    CBMATNL(ii+1,1) = CBMATNL(ii,1) + h*(CBMATNL(ii,1)*-Av +
CAMATNL(ii,1)*K1*exp(-E1/(R*TMATNL(ii,1))) -CBMATNL(ii,1)*K2*exp(-
E2/(R*TMATNL(ii,1))));
    TMATNL(ii+1,1) = TMATNL(ii,1)+ h*(Fv*(TJMATNL(ii,1) - TMATNL(ii,1)) +
Av*(DISTNL-TMATNL(ii,1)) - CAMATNL(ii,1)*K1*exp(-E1/(R*TMATNL(ii,1)))*Iv -
CBMATNL(ii,1)*K2*exp(-E2/(R*TMATNL(ii,1)))*Jv);
    TJMATNL(ii+1,1) = TJMATNL(ii,1) + h*(Hv*(UMATNL(ii,1)-TJMATNL(ii,1)) +
Gv*(TMATNL(ii,1)-TJMATNL(ii,1)));
    ZMATNL(ii+1,1) = ZMATNL(ii,1) + h*(YSPNL-TMATNL(ii,1));

    TIMEMAT(ii+1,1) = TIMEMAT(ii,1) + h;

end

subplot(1,2,1)
PLOTVEC = ones(1,length(TIMEMAT));
plot(TIMEMAT,YMATL+TS)
hold on
plot(TIMEMAT,PLOTVEC*YSPNL,'red')
%title('Linear Output')
xlabel('Time (h)')
ylabel('Temperature of Reactor (K)')
legend('Temperature','Set Point','Location','Southeast')
hold off
```

```
subplot(1,2,2)
plot(TIMEMAT,TMATNL)
hold on
plot(TIMEMAT,PLOTVEC*YSPNL,'red')
%title('Non-linear')
xlabel('Time (h)')
ylabel('Temperature of Reactor (K)')
legend('Temperature','Set Point','Location','Southeast')

figure(2)
subplot(1,2,1)
plot(TIMEMAT,UMATL+TJ0)
%title('Linear Input')
xlabel('Time (h)')
ylabel('Temperature of Jacket Inlet (K)')

subplot(1,2,2)
plot(TIMEMAT,UMATNL)
%title('Non-linear Input')
xlabel('Time (h)')
ylabel('Temperature of Jacket Inlet (K)')
```

A3

Table 1: Parameters for the CSTR.

Parameter	Description	Value	Units
UA	Overall heat transfer coefficient*Heat exchange area	1200	$\frac{kJ}{h\cdot K}$
ρ	Density of fluid in CSTR	1200	$\frac{kg}{m^3}$
C_p	Heat capacity of fluid in CSTR	0.42	$\frac{kJ}{kg\cdot K}$
ρ_j	Density of fluid in jacket	1000	$\frac{kg}{m^3}$
C_{pj}	Heat capacity of fluid in jacket	4.18	$\frac{kJ}{kg\cdot K}$
k_1	Pre-exponential factor, Reaction 1	6×10^5	$\frac{m^3\cdot h}{kmol}$
k_2	Pre-exponential factor, Reaction 2	2×10^6	$\frac{m^3\cdot h}{kmol}$
E_1	Activation energy, Reaction 1	5×10^4	$\frac{kJ}{kmol}$
E_2	Activation energy, Reaction 2	5.5×10^4	$\frac{kJ}{kmol}$
C_{A0}	Inlet stream concentration of A	7	$\frac{kmol}{m^3}$
T_0	Inlet stream temperature	300	K
F	Flow rate through CSTR	4	$\frac{m^3}{h}$
F_j	Flow rate through jacket	1	$\frac{m^3}{h}$
T_{j0}	Jacket inlet stream temperature	240	K
ΔH_1	Heat of reaction, Reaction 1	-2×10^4	$\frac{kJ}{kmol}$
ΔH_2	Heat of reaction, Reaction 2	-3×10^4	$\frac{kJ}{kmol}$
V	CSTR volume	6	m^3
V_j	Jacket volume	5	m^3
R	Gas constant	8.314	$\frac{kJ}{kmol\cdot K}$