PROJECT: Data Handling, Dynamics, and PI Controller of a Chemical Reactor

Name: Slava Ermolaev

Date: March 25, 2024

Table of Contents

1. Process Model

The dynamic system investigated is a jacketed continuously stirred tank reactor (CSTR) shown in Figure A-1 in the Appendix. Reactions that happen inside the reactor are the following

$$A \rightarrow B \rightarrow C$$

A → B is highly exothermic while B → C produces a negligible amount of heat. Both reactions are first order and follow an Arrhenius rate law given by equations E-1 and E2 in the Appendix. The reactor is fed with species A in an inert solvent at temperature T_0, concentration C_{A0}, and volumetric flow rate F_0. The effluent stream exits the reactor at concentrations C_A, C_B, C_C, and temperature T. Temperature and compositions in the reactor are assumed to be uniform and have constant density ρ, liquid hold-up volume V, heat of reactions ΔH_1 and ΔH_2 for reactions 1 and 2, respectively, and heat capacity C_p. The jacket is fed with cooling water at temperature T_{j0} and flow rate F_{j0}. The overall heat transfer coefficient and heat transfer area between the tank and the jacket are U and A_{ht}, respectively. The effluent leaves the jacket at temperature T_j. The temperature inside the jacket is assumed to be uniform.

The dimensionless differential equations of mass and energy balances that describe the dynamics of the process are given below

$$\frac{dx_1}{d\theta} = 1 - x_1 - Da_1 exp(-\overline{E_1}/x_3)x_1 \tag{1}$$

$$\frac{dx_2}{d\theta} = -x_2 + Da_1 exp(-\bar{E}_1/x_3)x_1 - Da_2 exp(-\bar{E}_2/x_3)x_2 \tag{2}$$

$$\frac{dx_3}{d\theta} = x_{30} - x_3 + Da_1' exp(-\bar{E}_1/x_3)x_1 + Da_2' exp(-\bar{E}_2/x_3)x_2 - \bar{U}(x_3 - x_4) \tag{3}$$

$$\frac{dx_4}{d\theta} = \epsilon_1\epsilon_2(x_{40} - x_4) + \bar{U}\epsilon_1\epsilon_3(x_3 - x_4) \tag{4}$$

where

$x_1 = C_A/C_{A0}$:	dimensionless concentration of species A in the reactor
$x_2 = C_B/C_{A0}$:	dimensionless concentration of species B in the reactor
$x_3 = T/T_{ref}$:	dimensionless temperature of the reactor
$x_4 = T_j/T_{ref}$:	dimensionless temperature of the jacket
$x_{30} = T_0/T_{ref}$:	dimensionless temperature of reactor inlet stream
$x_{40} = T_{j0}/T_{ref}$:	dimensionless temperature of jacket inlet stream
$\theta = F_0 t/V$:	dimensionless time

T_{ref} is the reference temperature and the remaining parameters (Da_1, Da_2, Da_1', Da_2', $\bar{E}_1$, $\bar{E}_2$, $\bar{U}$, ϵ_1, ϵ_2, and ϵ_3) are dimensionless process parameters.

2. Mass and Energy Balance

The mass balance equations of species A and B around the reactor are given below

$$V\frac{dC_A}{dt} = F_0C_{A0} - F_0C_A - Vk_{10}e^{-E_1/RT}C_A \tag{5}$$

$$V\frac{dC_B}{dt} = -F_0C_B + Vk_{10}e^{-E_1/RT}C_A - Vk_{20}e^{-E_2/RT}C_B \tag{6}$$

The energy balance equation around the reactor accounting for the inlet and outlet streams energy transfer, heat transfer to the jacket, and exothermic reaction enthalpy is given below

$$\rho C_pV\frac{dT}{dt} = \rho C_pF_0(T_0 - T) + \Delta H_1Vk10e^{-E_1/RT}C_A + \Delta H_2Vk_{20}e^{-E_2/RT}C_B - UA_j(T - T_j) \tag{7}$$

The energy balance around the jacket accounting for the inlet and outlet streams energy transfer and heat transfer to the jacket is given below

$$\rho C_pV\frac{dT_j}{dt} = \rho \mathrm{F}_{j0}\,C_p(T_{j0} - T_j) + UA_j(T - Tj) \tag{8}$$

Solving for the differential term on the left, the Equations 5-8 become

$$\frac{dC_A}{dt} = \frac{F_0C_{A0}}{V} - \frac{F_0C_A}{V} - k_{10}e^{-E_1/RT}C_A \tag{9}$$

$$\frac{dC_B}{dt} = \frac{-F_0C_B}{V} + k_{10}e^{-E_1/RT}C_A - k_{20}e^{-E_2/RT}C_B \tag{10}$$

$$\frac{dT}{dt} = \frac{F_0}{V}(T_0 - T) + \frac{\Delta H_1}{\rho C_p}k_{10}e^{-E_1/RT}C_A + \frac{\Delta H_2}{\rho C_p}Vk_{20}e^{-E_2/RT}C_B - \frac{UA_j}{\rho C_pV}(T - T_j) \tag{11}$$

$$\frac{dT_j}{dt} = \frac{F_{j0}(T_{j0} - T_j)}{V} + \frac{UA_j(T - T_j)}{\rho C_pV} \tag{12}$$

3. Three Steady-States of the Nonlinear Algebraic System

Process parameters defined in Section 1 were initialized with values given in Table 1.

Table 1: Set of Parameters for Steady-State Determination

Parameter	Value
Da_1	10^6
Da_2	10^7
Da_1'	1.5×10^6
Da_2'	0
x_{30}	0.025

x_{40}	0.025
$\bar{E}_1$	1.0066
$\bar{E}_2$	1.0532
$\bar{U}$	8
ϵ_1	12.5
ϵ_2	40
ϵ_3	1

Three guesses for four x values were chosen, and MATLAB's `fsolve` function was used to solve for the steady state of the Equations 1-4 such that the derivative terms are set to zero. Table 2 below displays three steady state sets of values of x.

Table 2: Three steady-state sets of x values

Steady State	Initial Guess $[x_1\ x_2\ x_3\ x_4]$	x_1	x_2	x_3	x_4
1	[0 0 0 0]	1	0	0.0250	0.0250
2	[0.75 0.75 0.75 0.75]	0.7877	0.0908	0.0665	0.0319
3	[1 1 1 1]	0	0	0.2206	0.0576

4. System Linearization and Determination of Unsteady Steady-State

To linearize the system of nonlinear differential equations, Jacobian was constructed using MATLAB `diff` function. The Equations 1-4 were redefined as the following

$$f_1 = \frac{dx_1}{d\theta}, \qquad f_2 = \frac{dx_2}{d\theta}, \qquad f_3 = \frac{dx_3}{d\theta}, \qquad f_4 = \frac{dx_4}{d\theta}$$

And the Jacobian matrix is defined as the following

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_4} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial x_1} & \cdots & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \tag{13}$$

Using the Jacobian, eigenvalues were calculated substituting steady-state X values from Table 2 and using `eig` function in MATLAB. Eigenvalues, λ, are defined using the following equation

$$(J - \lambda I)X = 0 \tag{14}$$

The calculated eigenvalues for the corresponding steady states in Table 2 are displayed in Table 3

Table 3: Eigenvalues, λ, for Each Steady State in Table 2

Steady State	λ_1	λ_2	λ_3	λ_4
1	-1	-1	-7.6494	-601.3506
2	-2.3384	-0.9725	64.3083	-601.2037
3	-8.4506×10^4	-1.0408×10^4	-7.6692	-601.3549

When all four eigenvalues for each steady state are negative, the system is stable. When at least one is positive, the system is unstable. Therefore, steady state #2 is the unstable one while steady states 1 and 3 are both stable.

5. Block Diagram of the Closed-loop System

The block diagram of the closed-loop system is depicted in Figure 1 where G_c, G_p, and G_d represent control, process, and disturbance transfer functions, respectively, y is the measured and controlled temperature of the reactor, y_{sp} is the desired reactor temperature, e is the error difference between y and y_{sp}, u is the temperature of the jacket inlet stream as manipulated input, and d is the temperature of the reactor inlet stream as disturbance.

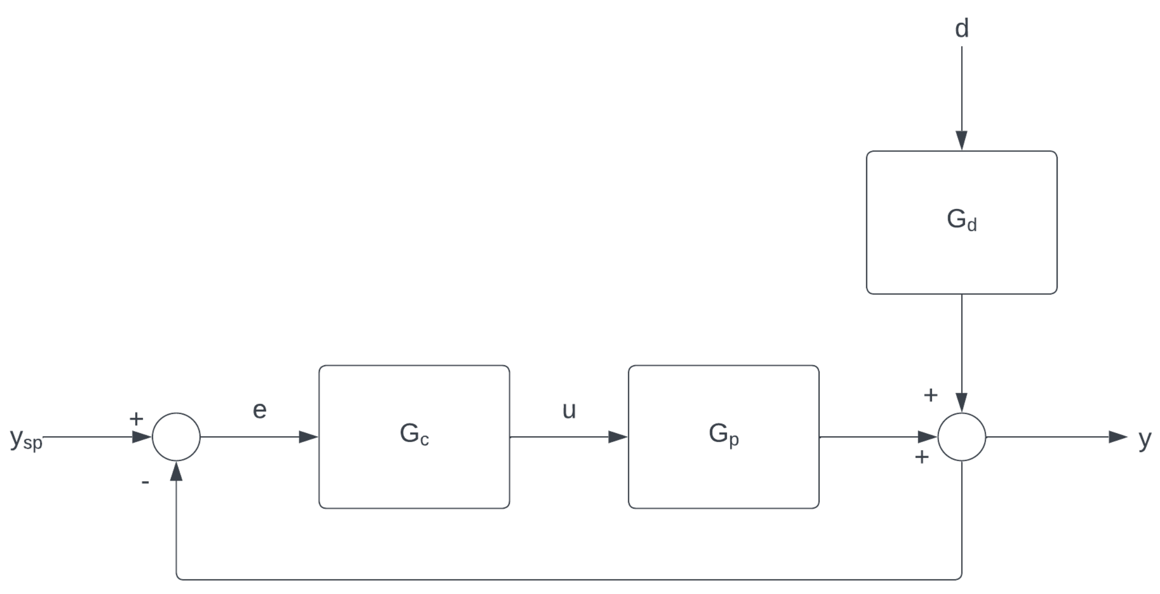

Figure 1: Block Diagram of the Closed-loop System with Disturbance

6. Transfer Function of the Linearized System

Using the unsteady steady state #2 X values based on the results from Table 2, the following Jacobian was calculated

$$J = \begin{bmatrix} -1.2696 & 0 & -48.2678 & 0 \\ 0.2696 & -2.3384 & 19.3623 & 0 \\ 0.4044 & 0 & 63.4016 & 8 \\ 0 & 0 & 100 & -600 \end{bmatrix} \tag{15}$$

Using the derived Jacobian in Equation 15, definition of Jacobian in Equation 13 and integrating each partial derivative of the Jacobian with respect to its x with the goal of obtaining equations similar to Equations 1-4, the following system of linear differential equations describing the unsteady steady state system was derived

$$\frac{dx_1}{d\theta} = -1.2696x_1 - 48.2678x_3 \tag{16}$$

$$\frac{dx_2}{d\theta} = 0.2696x_1 - 2.3384x_2 + 19.3623x_3 \tag{17}$$

$$\frac{dx_3}{d\theta} = d + 0.4044x_1 + 63.4016x_3 + 8x_4 \tag{18}$$

$$\frac{dx_4}{d\theta} = \epsilon_1\epsilon_2 u + 100x_3 - 600x_4 \tag{19}$$

where d and u are disturbance and controlled parameters, respectively given by the following

$$d = x_{30} - x_{30s}, \qquad u = x_{40} - x_{40s} \tag{20}$$

where subscript s denotes steady-state parameter.

Taking the Laplace transform of Equations 16-19, the following equations were obtained

$$\mathrm{sx}_1(\mathrm{s}) = -1.2696x_1(s) - 48.2678x_3(s) \tag{21}$$

$$sx_2(s) = 0.2696x_1(s) - 2.3384x_2(s) + 19.3623x_3(s) \tag{22}$$

$$sx_3(s) = d(s) + 0.4044x_1(s) + 63.4016x_3(s) + 8x_4(s) \tag{23}$$

$$sx_4(s) = \epsilon_1\epsilon_2 u(s) + 100x_3(s) - 600x_4(s) \tag{24}$$

Given that the output *y* in Figure 1 is the temperature of the reactor, which is given by $x_3(s)$, then $y(s)=x_3(s)$. Solving Equations 21-24 for $y(s)$ and factorizing polynomials using Wolfram Alpha gives the following

$$\begin{aligned} y = &\frac{8\epsilon_1\epsilon_2(s + 1.2696)}{(s - 64.3082)(s + 0.9725)(s + 601.204)}u \\ &+ \frac{(s + 1.2696)(s + 600)}{(s - 64.3082)(s + 0.9725)(s + 601.204)}d \end{aligned} \tag{25}$$

Separating Equation 25 into transfer functions G_p and G_d gives the following

$$G_p = \frac{8\epsilon_1\epsilon_2(s + 1.2696)}{(s - 64.3082)(s + 0.9725)(s + 601.204)} \tag{26}$$

$$G_d = \frac{(s + 1.2696)(s + 600)}{(s - 64.3082)(s + 0.9725)(s + 601.204)} \tag{27}$$

7. Proportional-Integral (PI) Controller for Linear System Stability

For Proportional-Integral (PI) controller, the transfer function is given by the following equation

$$G_c = K_c\left(1 + \frac{1}{\tau_I s}\right) \tag{28}$$

where K_c is the proportional gain, τ_I is the time integral constant.

Using G_p and G_d derived in Equation 26 and 27, respectively, the closed-loop transfer function for the system without disturbance depicted in Figure 1 was computed as follows

$$G = \frac{G_p G_c}{1 + G_p G_c} \tag{29}$$

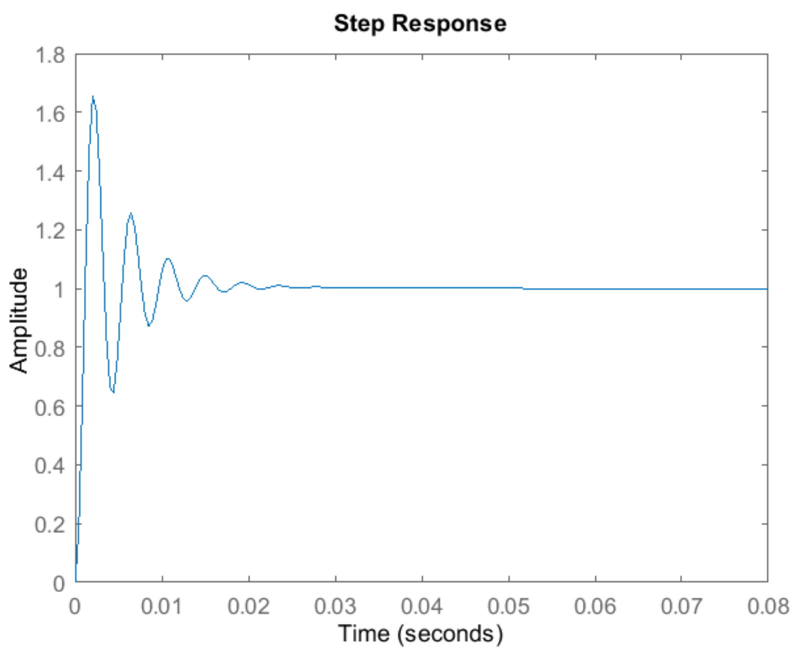

Figure 2: Step response of the most optimal PI controller using $K_c = 570, \tau_I = 0.013$, which has the minimum settling time

Using iterative method and determining using `stepinfo(G).SettlingTime` function in MATLAB, the transfer function with the minimum settling time was determined to be the most optimal, and its most optimal K_c and τ_I parameters were determined to be the following.

$$K_c = 570, \qquad \tau_I = 0.013$$

Therefore, the optimized PI controller is the following

$$G_c = 570\left(1 + \frac{1}{0.013s}\right) \tag{30}$$

Figure 2 displays the graph of the step response of the system with the optimized PI controller.

8. Applying PI controller to Linear and Nonlinear System

Using the transfer function for the closed loop system in Equation 29, the following equation can be used to calculate the controlled output response to setpoint change in the reactor temperature Y_{sp}

$$Y = \frac{G_p G_c}{1 + G_p G_c} Y_{sp} \tag{31}$$

Using the optimized PI controller in Equation 30 and G_p as defined in Equation 26, the controlled output for the linear system can be computed by using Equation 31. Similarly, by using Figure 1 block diagram, the manipulated input, u, can be calculated using the following

$$u = G_c e \tag{32}$$

where $e = Y_{sp} - Y$

Therefore, using the optimized G_c for the linearized system in Equation 30 and controlled output from Equation 31, the manipulated input response can be computed using Equation 32.

For the nonlinear system, the Equations 1-4 were defined in terms of deviation variables from the unsteady steady state. x_3 in Equation 3 was defined as the controlled output y, x_{30} was defined as the disturbance, x_{40} was defined as the manipulated input u. The `ode45` MATLAB function was used to integrate nonlinear differential equations.

Step responses of both manipulated input and controlled output for the linear and nonlinear system with set-point change of 0.005 are depicted in Figure 3. Both linear and nonlinear systems display oscillatory behavior for a small time, but all eventually stabilize. The amplitude of controlled output of both systems stabilizes at around the set-point value of 0.005 for large times. The amplitude of the manipulated input stabilizes around 0 for large times. The amplitude of oscillations for nonlinear system is a little larger than for linear system. While the linear system on the plot seems to be steady, nonlinear system displays small ripples that can be seen in controlled output for times between 0.06 and 0.07 seconds as well as after 0.09 seconds. Nevertheless, these small ripples do not affect the overall stability, and the system returns to its steady-state set-point value.

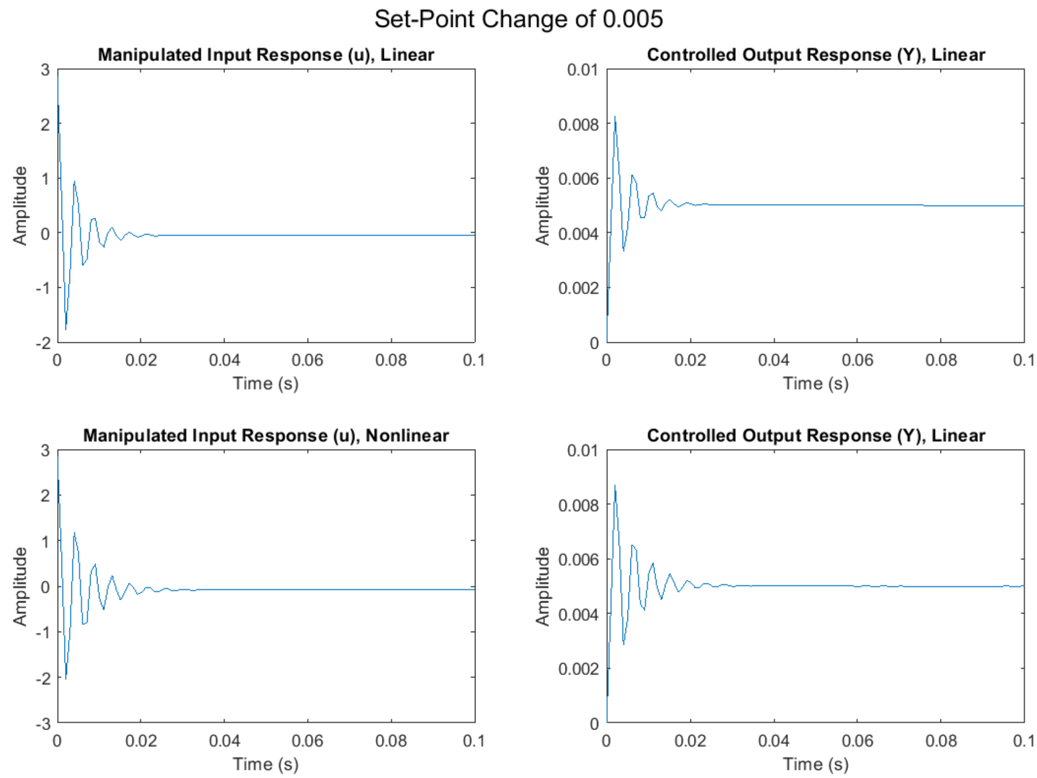

Figure 3: Step responses of linear and nonlinear system to set-point change of 0.005

The controlled output amplitude represents the dimensionless temperature of the reactor, and the manipulated input amplitude represents dimensionless temperature of the jacket inlet stream. As the controlled output initially increases from 0, the manipulated input decreases, which reflects the increasing temperature of the reactor and decreasing temperature of the jacket inlet stream, which is reasonable because the temperature of the reactor overshoots the set-point of 0.005 and this the jacket inlet temperature is decreased to counteract this effect and vice versa.

The same methods described above were used for determining linear and nonlinear step responses for the set-point change of 0.01. The step response plots for both linear and nonlinear systems for the set-point change of 0.01 are displayed in Figure 4.

For both linear and nonlinear systems, the overall amplitude is higher compared to Figure 3. The linear systems in Figures 3 and 4 exhibit the same number of oscillations before stabilization. However, the nonlinear system in Figure 4 exhibits significantly more oscillations compared to nonlinear system in Figure 3. The overall behavior and correlation of controlled output and manipulated input is similar to what was described previously; as the controlled output increases, manipulated input decreases, which is reasonable according to dynamics of the system. The main conclusion, which can be derived from both systems with set-point changes of 0.005 and 0.01, is that the system is stable and settles to its set-point value for large times eliminating the offset.

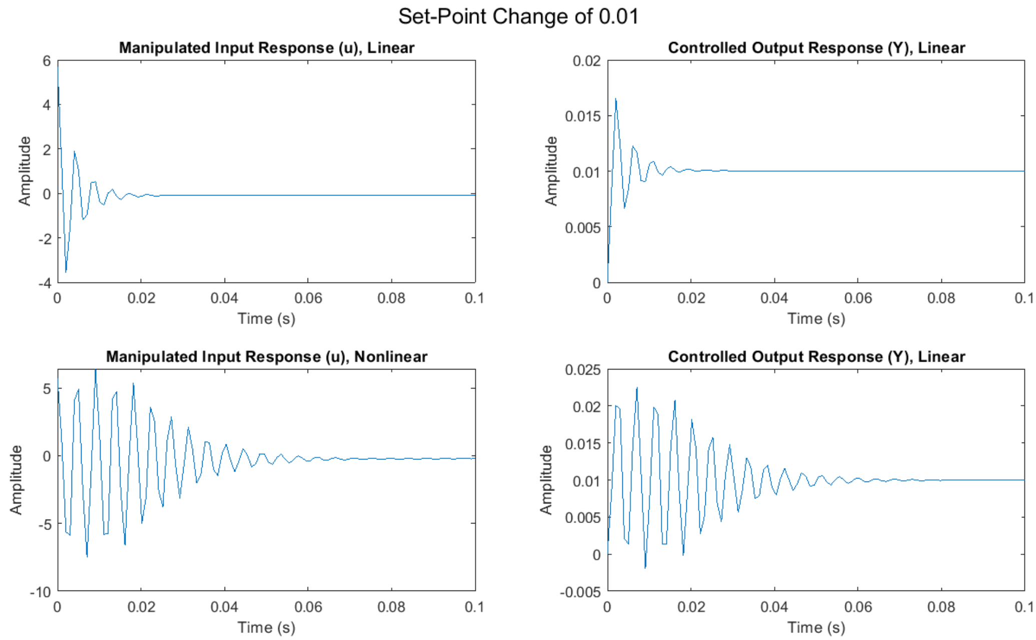

Figure 4: Step responses of linear and nonlinear system to set-point change of 0.01

9. Examining Disturbance Rejection Capabilities of the Controller

To study how the changes in reactor inlet temperature that act as a disturbance, G_d, affect the system, d is set to a desired magnitude of 0.001 and 0.002, while the setpoint of the reactor temperature Y_{sp} is set to 0. The effect is studied for both linear and nonlinear systems.

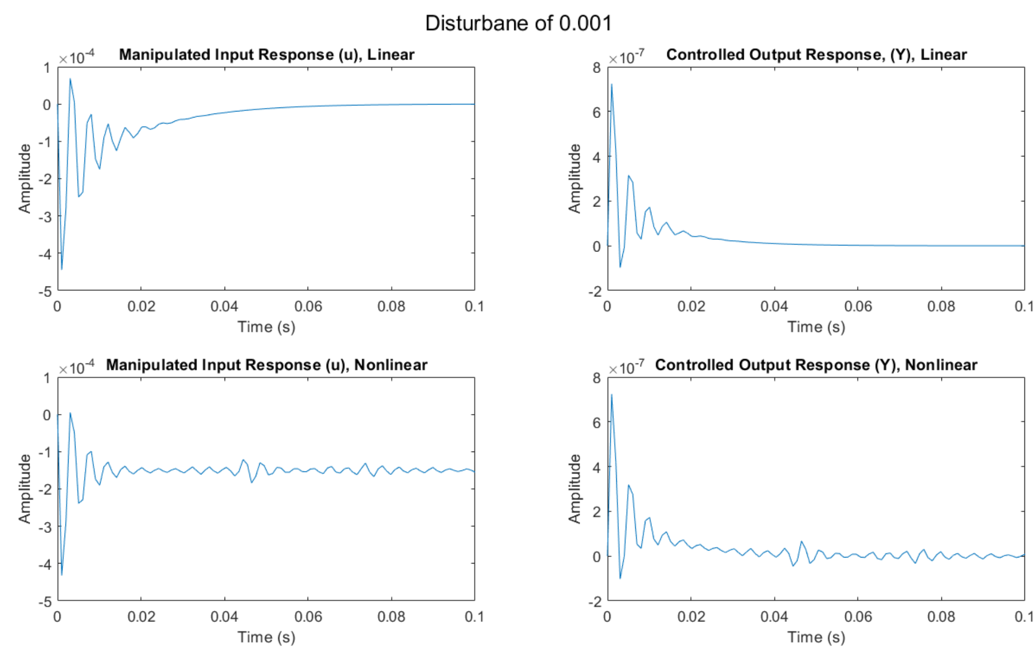

Figure 5: Step responses of linear and nonlinear system to disturbance of 0.001

Figure 5 shows input and output responses of linear and nonlinear systems. Manipulated input in both systems initially decreases and then increases and begins to oscillate around an equilibrium value. However, for linear system, input response eventually plateaus back to 0 unlike for nonlinear input response, which stays and oscillates around value of -1.5×10^{-4}. Controlled output responses for both linear and nonlinear systems in Figure 5 look very similar. Initially they both increase, which reflects positive disturbance, increasing temperature of reactor inlet, which in turn increases reactor temperature. As the controller decreases the temperature of the jacket, decreasing manipulated input response, the output response begins to decrease, oscillating and then exponentially decaying back to 0. The only observable difference between linear and nonlinear output responses is oscillatory decay of nonlinear system, while linear system exhibits smooth decay.

Figure 6 shows step responses for disturbance magnitude of 0.002.

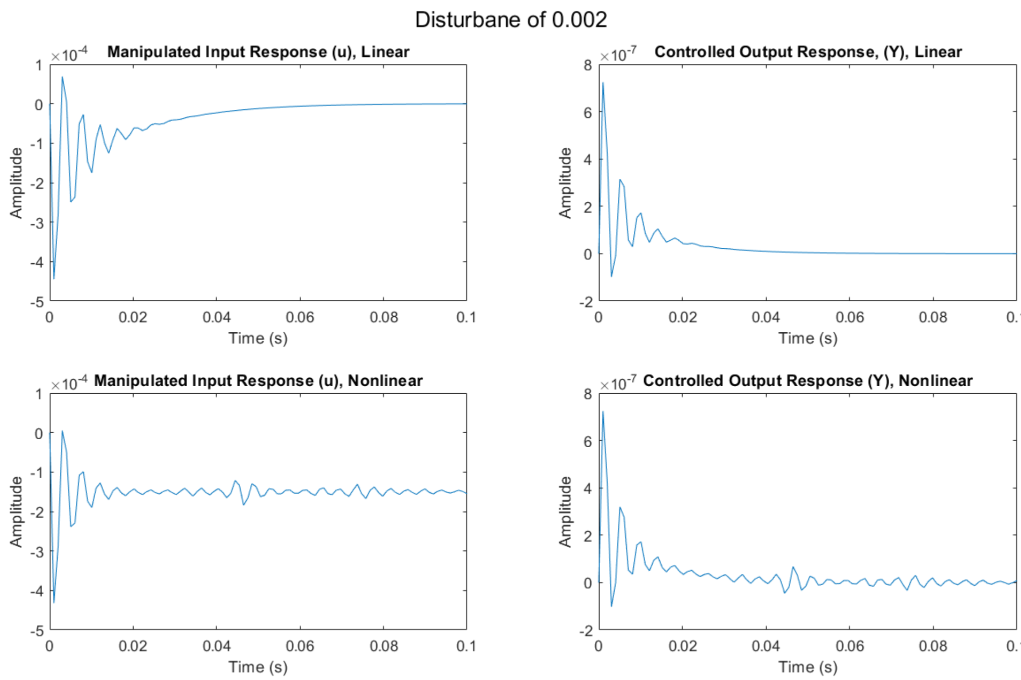

Figure 6: Step responses of linear and nonlinear system to disturbance of 0.002

No significant differences observed between Figure 6 and 5. Disturbance magnitude of 0.002 has the same effect on step responses as disturbance of 0.001. It can be concluded that the control successfully rejects disturbances of magnitudes 0.001 and 0.002 stabilizing the system at set point output of 0.

10. Application of PI Controller when $Da_2' = 10^5$

PI controller was applied to the same nonlinear system in the unsteady steady state using parameters given in Table 1 except Da_2' was changed from 0 to 10^5. Da_2' parameter represents heat released by exothermic reaction B → C. In previous sections heat was assumed to be negligent. The responses were investigated for the system with two sets of conditions: 1) set point of 0.005 and no disturbance; 2) set point of 0 and disturbance of 0.001. The responses are depicted in Figure 7.

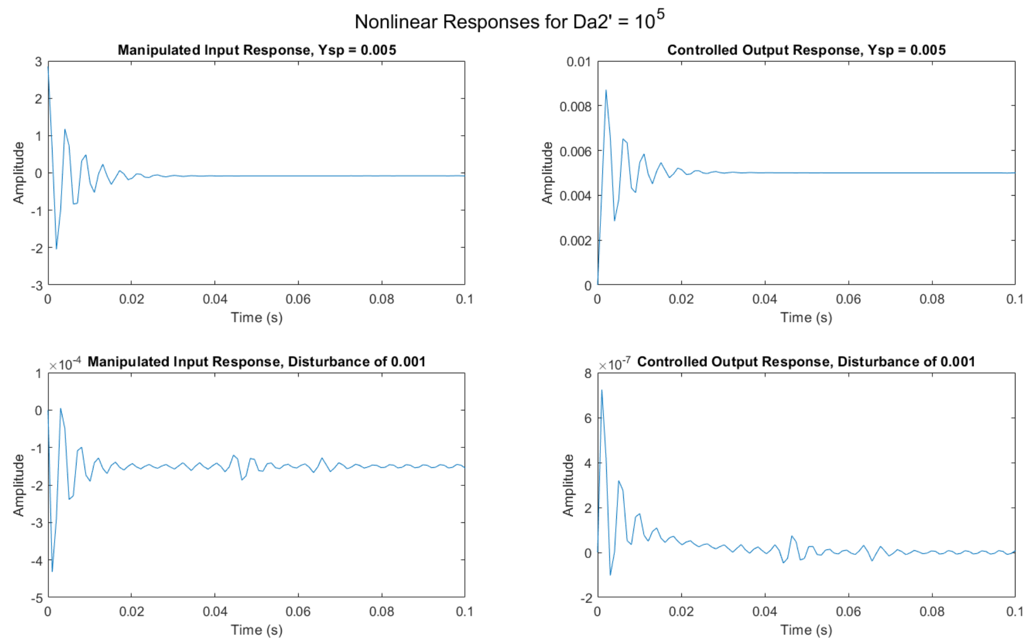

Figure 7: Step responses of nonlinear system to set point of 0.005 and disturbance of 0.001

Results in Figure 7 are very similar to results in Figure 6 for nonlinear system. No significant differences are observed. It can be concluded that the system is stable under PI controller regardless of the magnitude of heat of exothermic reactions.

11. Conclusion

Overall, it can be concluded that the system is stable under PI controller in unsteady steady state encountering different changes in conditions such as changes in set point, disturbance, and heat of exothermic reactions. PI controller proves to be an effective controller that offsets any changes in controller output back to the stable value.

12.Appendix

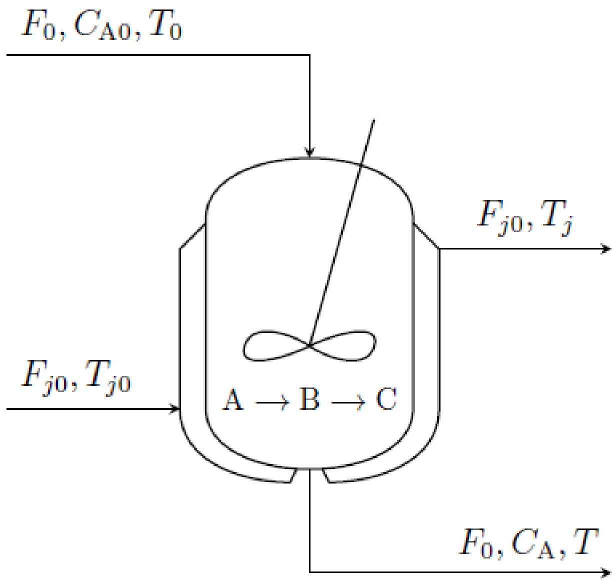

Figure A-1: Jacketed continuously stirred tank reactor (CSTR)

Arrhenius rate law of two reactions happening inside CSTR

$$r_1 = k_{10}e^{-E_1/RT}C_A \qquad (E-1)$$

$$r_2 = k_{20}e^{-E_2/RT}C_B \qquad (E-2)$$

where k_{10} and k_{20} are the pre-exponential factors in units per time, E_1 and E_2 are activation energies, R is the gas constant, C_A and C_B are reactants' concentrations in the tank, and T is temperature in the reactor.

MATLAB Script

```
clear all %#ok
close all
clc

%% Q1 For mass and energy balance derivation see report

%% Q2 Steady State

X_ss = zeros(3,4);

X_ss(1,:) = fsolve(@process, [0 0 0 0]);
X_ss(2,:) = fsolve(@process, [0.75 0.75 0.75 0.75]);
X_ss(3,:) = fsolve(@process, [1 1 1 1]);

%% Q3
X = sym('x', [1 4]); syms(X, 'real');

% Create Jacobian
```

```
J = sym('dFdx', [4 4]);
F = process(X);

for i = 1:4
    for j = 1:4
        J(i,j) = diff(F(i),X(j));
    end
end

eigenvals = zeros(3,4);
for i = 1:3
    eigenvals(i,:) = eig(double(subs(J, [x1 x2 x3 x4], X_ss(i,:))));
end

%% Q4 For Block Diagram see report

%% Q5 Operating around unsteady steady-state
X = X_ss(2,:);
J = double(subs(J, [x1 x2 x3 x4], X));

%% Q6 Designing PI controller


s = tf('s');
e1 = 12.5;
e2 = 40;
Gp = 8*e1*e2*(s+1.2696)/(s-64.3082)/(s+0.9725)/(s+601.204);
Gd = (s+1.2696)*(s+600)/(s-64.3082)/(s+0.9725)/(s+601.204);

% t_min = 1;
% %  Iterate through different Kc and tauI values
% for Kc = 50:5:600
%     for tauI = 0.001:0.001:0.1
%         Gc = Kc*(1+1/(tauI*s));
%         G = Gp*Gc/(1+Gp*Gc);
%         t = stepinfo(G).SettlingTime;
%         Kc/1000
%         if t < t_min
%             t_min = t;
%             Kc_optimum = Kc;
%             tauI_optimum = tauI;
%         end
%     end
% end
Kc_optimum = 570;
tauI_optimum = 0.013;

Gc = Kc_optimum*(1+1/(tauI_optimum*s));
G = Gp*Gc/(1+Gp*Gc); % Set transfer function

figure(1)
hold on
step(G,0.1)

%% Q7 Applying PI Controller to Linear and Nonlinear System
```

```
%  Ysp = 0.005 for Linear system
Ysp = 0.005;
Da2prime = 0;
%  No disturbance
x30 = 0;
Y = Ysp*G;
u = Gc*(Ysp-Y);
t = linspace(0,0.1);
Y_lin = step(Y,t);
u_lin = step(u,t);

%  Ysp = 0.005 for Nonlinear system
[t,Y_nonlin] =
ode45(@(t,X_nonlin)DeviationVarProcess(X_nonlin,Ysp,x30,Kc_optimum,tauI_optimum,X,Da2
prime),t,zeros(5,1));
%u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*(Ysp - Y_nonlin(:,3)));
u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*Y_nonlin(:,5));

%  Plots for Ysp = 0.005

figure(2)
hold on
sgtitle('Set-Point Change of 0.005')
subplot(2,2,1)
plot(t,u_lin)
title('Manipulated Input Response (u), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,2)
plot(t,Y_lin)
title('Controlled Output Response (Y), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,3);
plot(t,u_nonlin);
title('Manipulated Input Response (u), Nonlinear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,4);
plot(t,Y_nonlin(:,3));
title('Controlled Output Response (Y), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

%  Ysp = 0.01 for Linear system
Ysp = 0.01;
%  No disturbance
x30 = 0;
Y = Ysp*G;
u = Gc*(Ysp-Y);
t = linspace(0,0.1);
```

```
Y_lin = step(Y,t);
u_lin = step(u,t);

%  Ysp = 0.01 for Nonlinear system
[t,Y_nonlin] =
ode45(@(t,X_nonlin)DeviationVarProcess(X_nonlin,Ysp,x30,Kc_optimum,tauI_optimum,X,Da2
prime),t,zeros(5,1));
%u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*(Ysp - Y_nonlin(:,3)));
u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*Y_nonlin(:,5));

figure(3)
hold on
sgtitle('Set-Point Change of 0.01')
subplot(2,2,1)
plot(t,u_lin)
title('Manipulated Input Response (u), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,2)
plot(t,Y_lin)
title('Controlled Output Response (Y), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,3);
plot(t,u_nonlin);
title('Manipulated Input Response (u), Nonlinear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,4);
plot(t,Y_nonlin(:,3));
title('Controlled Output Response (Y), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

%% Q8 Distrubance rejection capabilities of the controller

%  For disturbance of 0.001
x30 = 0.001;
Ysp = 0;
t = linspace(0,0.1);

%  Get step responses for linear system
G = x30*Gd/(1+Gp*Gc); % Transfer function for Ysp = 0
Y_lin = step(G,t);
u_lin = step(Gc,t).*(Ysp-Y_lin);

%  Get step responses for nonlinear system
[t,Y_nonlin] =
ode45(@(t,Y_nonlin)DeviationVarProcess(Y_nonlin,Ysp,x30,Kc_optimum,tauI_optimum,X,Da2
prime),t,zeros(5,1));
u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*Y_nonlin(:,5));
```

```
%  Plot linear and nonlinear responses
figure(4)
hold on
sgtitle('Disturbane of 0.001')
subplot(2,2,1)
plot(t,u_lin)
title('Manipulated Input Response (u), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,2)
plot(t, Y_lin)
title('Controlled Output Response, (Y), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,3)
plot(t,u_nonlin)
title('Manipulated Input Response (u), Nonlinear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,4)
plot(t,Y_nonlin(:,3))
title('Controlled Output Response (Y), Nonlinear')
xlabel('Time (s)')
ylabel('Amplitude')

%  For disturbance of 0.002
%  Linear
x30 = 0.001;
G = x30*Gd/(1+Gp*Gc);
Y_lin = step(G,t);
u_lin = step(Gc,t).*(Ysp-Y_lin);

%  Nonlinear
[t,Y_nonlin] =
ode45(@(t,Y_nonlin)DeviationVarProcess(Y_nonlin,Ysp,x30,Kc_optimum,tauI_optimum,X,Da2
prime),t,zeros(5,1));
u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*Y_nonlin(:,5));

%  Plot linear and nonlinear responses
figure(5)
hold on
sgtitle('Disturbane of 0.002')
subplot(2,2,1)
plot(t,u_lin)
title('Manipulated Input Response (u), Linear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,2)
plot(t, Y_lin)
title('Controlled Output Response, (Y), Linear')
xlabel('Time (s)')
```

```
ylabel('Amplitude')

subplot(2,2,3)
plot(t,u_nonlin)
title('Manipulated Input Response (u), Nonlinear')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,4)
plot(t,Y_nonlin(:,3))
title('Controlled Output Response (Y), Nonlinear')
xlabel('Time (s)')
ylabel('Amplitude')

%% Q9 Applying PI controller to nonlinear process with Da2prime = 10^5
Da2prime = 10^5;

%  Set point change of 0.005 and no disturbance
Ysp = 0.005;
x30 = 0;
[t,Y_nonlin] =
ode45(@(t,Y_nonlin)DeviationVarProcess(Y_nonlin,Ysp,x30,Kc_optimum,tauI_optimum,X,Da2
prime),t,zeros(5,1));
u_nonlin = Kc_optimum*(Ysp-Y_nonlin(:,3) + 1/tauI_optimum*Y_nonlin(:,5));

figure(6)
hold on
sgtitle('Nonlinear Responses for Da2'' = 10^5')

subplot(2,2,1)
plot(t,u_nonlin)
title('Manipulated Input Response, Ysp = 0.005')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,2)
plot(t,Y_nonlin(:,3))
title('Controlled Output Response, Ysp = 0.005')
xlabel('Time (s)')
ylabel('Amplitude')

%  Disturbance of 0.001 with no set point change
Ysp = 0;
x30 = 0.001;
[t,Y_nonlin] =
ode45(@(t,Y_nonlin)DeviationVarProcess(Y_nonlin,Ysp,x30,Kc_optimum,tauI_optimum,X,Da2
prime),t,zeros(5,1));
u_nonlin = Kc_optimum*(Ysp - Y_nonlin(:,3) + 1/tauI_optimum*Y_nonlin(:,5));

subplot(2,2,3)
plot(t,u_nonlin)
title('Manipulated Input Response, Disturbance of 0.001')
xlabel('Time (s)')
ylabel('Amplitude')
```

```
subplot(2,2,4)
plot(t,Y_nonlin(:,3))
title('Controlled Output Response, Disturbance of 0.001')
xlabel('Time (s)')
ylabel('Amplitude')

%% Process Function
function F = process(X)

Da1 = 10^6;
Da2 = 10^7;
Da1prime = 1.5*10^6;
Da2prime = 0;
x30 = 0.025;
x40 = 0.025;
E1 = 1.0066;
E2 = 1.0532;
U = 8;
e1 = 12.5;
e2 = 40;
e3 = 1;

F(1) = 1 - X(1) - Da1*exp(-E1/X(3))*X(1);
F(2) = -X(2) + Da1*exp(-E1/X(3))*X(1)-Da2*exp(-E2/X(3))*X(2);
F(3) = x30 - X(3) + Da1prime*exp(-E1/X(3))*X(1) + Da2prime*exp(-E2/X(3))*X(2) -
U*(X(3)-X(4));
F(4) = e1*e2*(x40 - X(4)) + U*e1*e3*(X(3) - X(4));

end

function F = DeviationVarProcess(X,Ysp,x30,Kc,tauI,X_ss,Da2prime)

Da1 = 10^6;
Da2 = 10^7;
Da1prime = 1.5*10^6;
E1 = 1.0066;
E2 = 1.0532;
U = 8;
e1 = 12.5;
e2 = 40;
e3 = 1;

F = zeros(4,1);
F(1) = -X(1) - Da1*exp(-E1/(X(3)+X_ss(3)))*(X(1)+X_ss(1)) + Da1*exp(-
E1/X_ss(3))*X_ss(1);
F(2) = -X(2) + Da1*exp(-E1/(X(3)+X_ss(3)))*(X(1)+X_ss(1)) - Da1*exp(-
E1/X_ss(3))*X_ss(1) - Da2*exp(-E2/(X(3)+X_ss(3)))*(X(2)+X_ss(2)) + Da2*exp(-
E2/X_ss(3))*X_ss(2);
F(3) = x30 - X(3) + Da1prime*exp(-E1/(X(3)+X_ss(3)))*(X(1)+X_ss(1)) - Da1prime*exp(-
E1/X_ss(3))*X_ss(1) + Da2prime*exp(-E2/(X(3)+X_ss(3)))*(X(2)+X_ss(2)) -
Da2prime*exp(-E2/X_ss(3))*X_ss(2) - U*(X(3) - X(4));
F(4) = e1*e2*(Kc*(Ysp - X(3)+1/tauI*X(5)) - X(4)) + U*e1*e3*(X(3)-X(4));
F(5) = Ysp - X(3);

end
```