**Chbe 4400 Project 1**

Group 1

Jiyuan Ding

Junxian Wu

Zongzhe Xue

1. Balances

Mass Balance:

(1)

Linearization for later use:

Subtract steady state:

(2)

If Steady State:

Since outflow mass must equal to inflow mass, we have:

Energy balance:

(3)

Linearization:

At steady state:

Finally:

(4)

Equation for h:

(5)

At steady state:

(6)

2. Operating temperature plot:

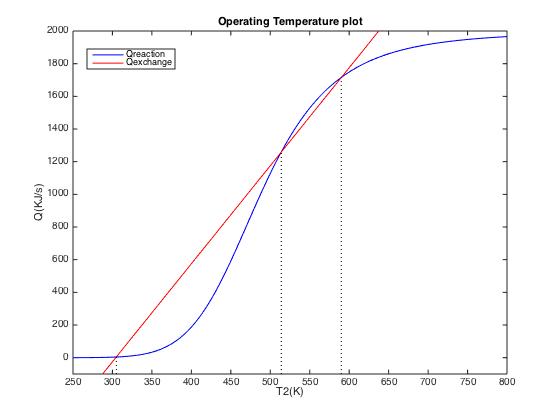
Use mass balance to substitute C2:

Plug C2 in the Qreaction:

(7)

(8)

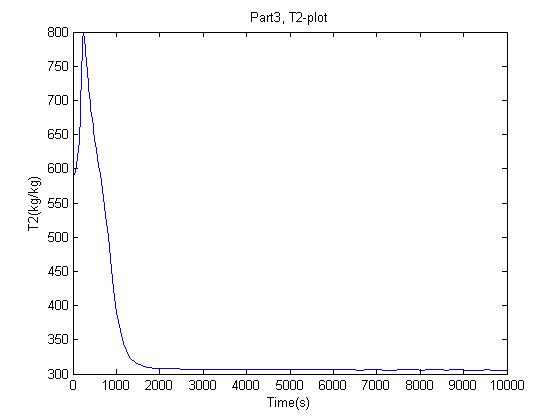
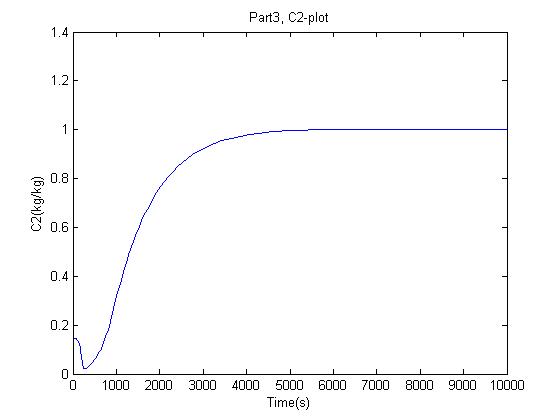
Evaluate eq (7) and (8) in Matlab, we get the following graph:

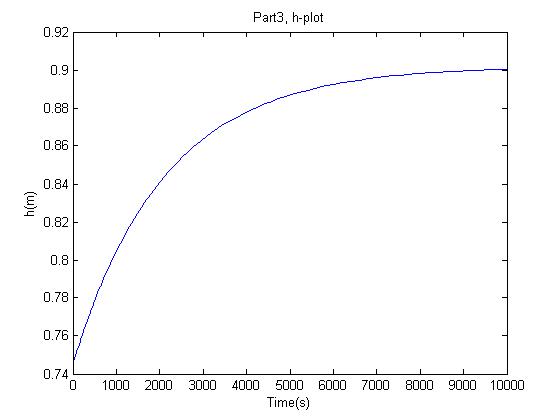


The highest operating temperature is about 590K. This system has three possible steady state operating temperature. However, only the left and right one are stable. If a disturbance causes the system to be a bit colder than the middle temperature, heat exchanger will remove more heat than the reaction can generate. This will cause the temperature to drop down to the left point. Likewise, the temperature will shift to the right if the system because a bit hotter than the middle point. The left and right points will always shift back to themselves whenever there’s a small disturbance.

3. Feed rate change response

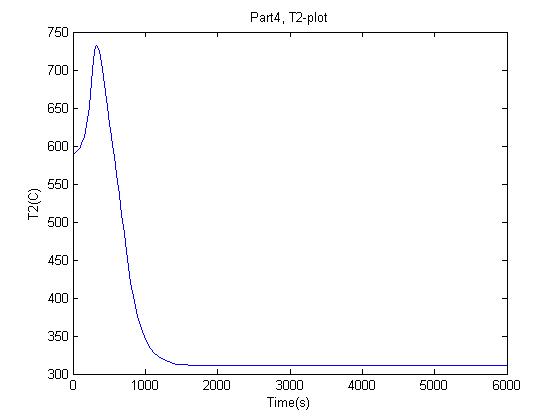
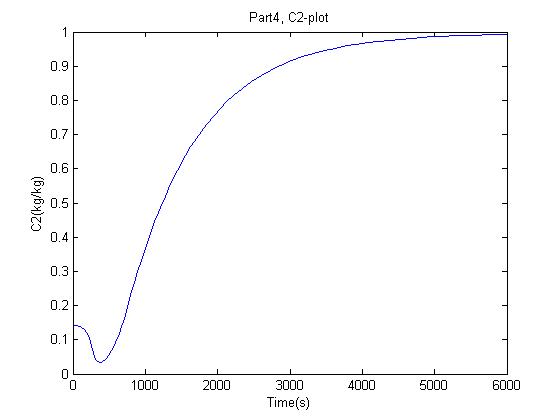
By defining the nonlinear equations (1), (3) and (5) in Matlab and feeding them to an ode45 solver, the response of outlet concentration, temperature and liquid level was computed and represented in the following three images. The liquid level will increase from ~0.74m to 0.90m. The concentration will rise from ~0.15 to 1. The outlet temperature will decrease from 590 to 306K. Matlab codes can be seen in Appendix.





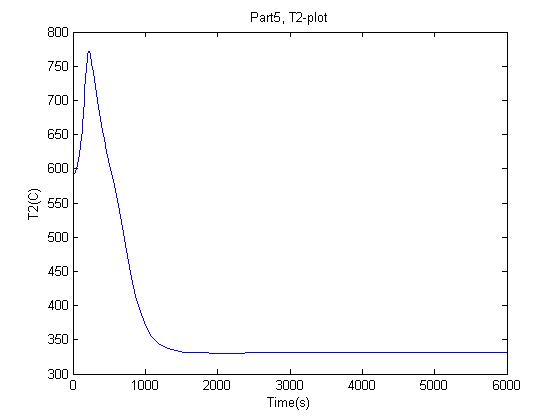
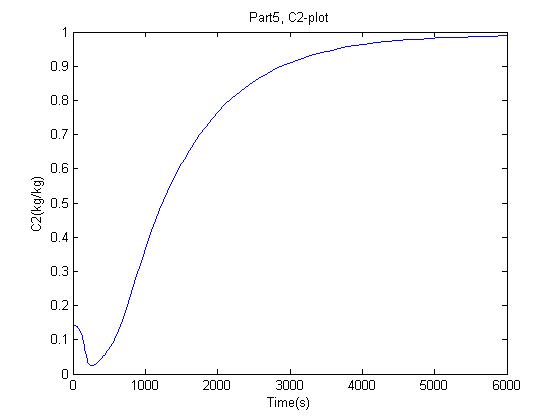
4. Feed temperature change response

Similarly, the outlet concentration and temperature response to a feed temperature change was computed and shown in the following two images. The concentration will increase from ~0.15 to 1. The temperature will drop from 590 to 311.



5. Cooling temperature change response

With the same method as part 3 and 4, the outlet concentration and temperature response are shown in the following two images. The concentration will rise from ~0.15 to 0.99. The temperature will drop from 590 to 331K.



6. State space representation:

State variables:

Input variables: [u]=[

Rearranging equation (2), (4) and (6), we get the following state-space representation:

= + [][]

7. Simulink

An array of all the transfer functions was obtained by doing the following operation:

G is a 3x4 array, each array item represents a specific transfer function.

Assume s goes to 0 and multiply the input impulse as 0.1/s, then we can calculate the change of steady state in a long time. Then the results were compared to Part 3-5.

Compared to part 3:

For T2, as s goes to 0:

T2’ = -191.099K

T2 = 590 - 191.099 = 398.901K

Which is close enough to the answer in Part 3 as 310.

For C2 and h, the results were deflected much from reasonable value. The massive balance around these two variables could be wrong in somewhere.

Compared to part 4:

For C2 and T2, the results were deflected much from reasonable values. The massive balance around these two variables could be wrong in somewhere.

Compared to part 5:

For T2 and C2, the results were deflected much from reasonable values. The massive balance around these variables could be wrong in somewhere.

Appendix:

Complete code repository can be see at: <https://github.com/axeisghost/CHBE4400Project1>

Heat of reaction vs. Heat of exchange:

k0 = 32; Rc = 8.31; Ea = 42.\*10.^3; Cvpm = 2.34e-5;

C1 = 1; Tc = 295; CP = 1.;

dHrxn = 2.\*10.^3; T1 = 350;

F1 = 10.^(-3);

F2 = F1;

den = 10.^3; UA = 5;

Cvp = 0.5.\*Cvpm;

h1 = (F2.^2)./((Cvp.^2).\*den.\*9.8);

Vt = 1.33 .\* h1;

T2 = (250:900);

QR = Vt.\*den.\*dHrxn.\*((k0.\*C1)./((exp(Ea./(Rc.\*T2)))+k0.\*Vt./F1));

QE = -UA.\*(Tc - T2)-F1.\*den.\*CP.\*T1+F2.\*den.\*CP.\*T2;

plot(T2, QR, 'b');

hold on;

plot(T2, QE, 'r');

title('Operating Temperature plot');

legend('Qreaction', 'Qexchange');

ylabel('Q(KJ/s)');

xlabel('T2(K)');

xlim([250 800]);

ylim([-100 2000]);

for ind = (1:length(T2))

if (abs(QR(ind) - QE(ind)) <= 1)

plot([T2(ind) T2(ind)], [-100 QR(ind)], 'k:');

ans = T2(ind);

C2 = C1./(1+((k0.\*exp(-Ea./Rc./T2(ind)).\*Vt./F1)));

end

end

Cooling temperature change:

function re = CoolingTempChange(t, input)

Vt0 = 1.5; %m3

A = 1.33; %m2

Cvpm = 2.34e-5;

Pvp = 9.;

F1ss = 1e-3; %m3/s

F2ss = F1ss;

C1ss = 1.; %kg/kg

T1ss = 350.; %k

ro = 1.e3; %kg/m3

Cp = 1; %kj/kg/k

Hrx = 2.e3; %j/kg

k0 = 32.; %1/s

R = 8.31; %J/k

Ea = 42.\*10.^3; %j

Tc = 295.; %k

UA = 5; %kj/k/s

Fcmax = 90.; %L/s

Pc = 9.; %psi

C2ss = C1ss.\*0.2;

g= 9.8;% m/s2

%secondary parameters:

Cvp = (Pvp - 3)./12.\*Cvpm;

Fc = (Pc - 3)./12.\*Fcmax;

F1 = F1ss;

F2 = sqrt(input(1).\*ro.\*g) .\* Cvp;

Vt = input(1) .\* A;

C2 = input(2);

T2 = input(3);

T1 = T1ss;

Tc = Tc .\* 1.1;

%-------------------------------------

re(1,1) = (F1 - F2)./A;

re(2,1) = F1.\*C1ss./Vt - F2.\*C2./Vt-k0.\*exp(-Ea./R./T2).\*C2;

re(3,1) = F1.\*T1./Vt - F2.\*T2./Vt + UA.\*(Tc - T2)./Vt./Cp./ro + Hrx.\*k0.\*exp(-Ea./R./T2).\*C2./Cp;

end

Flow rate change:

function re = FlowRateChange(t, input)

Vt0 = 1.5; %m3

A = 1.33; %m2

Cvpm = 2.34e-5;

Pvp = 9.;

F1ss = 1e-3; %m3/s

F2ss = F1ss;

C1ss = 1.; %kg/kg

T1ss = 350.; %k

ro = 1.e3; %kg/m3

Cp = 1; %kj/kg/k

Hrx = 2.e3; %j/kg

k0 = 32.; %1/s

R = 8.31; %J/k

Ea = 42.\*10.^3; %j

Tc = 295.; %k

UA = 5; %kj/k/s

Fcmax = 90.; %L/s

Pc = 9.; %psi

% C2ss = C1ss.\*0.2;

g= 9.8;% m/s2

%secondary parameters:

Cvp = (Pvp - 3)./12.\*Cvpm;

% Fc = (Pc - 3)./12.\*Fcmax;

F1 = F1ss .\* 1.1;

F2 = sqrt(input(1).\*ro.\*g) .\* Cvp;

Vt = input(1) .\* A;

C2 = input(2);

T2 = input(3);

T1 = T1ss;

%-------------------------------------

re(1,1) = (F1 - F2)./A;

re(2,1) = F1.\*C1ss./Vt - F2.\*C2./Vt-k0.\*exp(-Ea./R./T2).\*C2;

re(3,1) = F1.\*T1./Vt - F2.\*T2./Vt + UA.\*(Tc - T2)./Vt./Cp./ro + Hrx.\*k0.\*exp(-Ea./R./T2).\*C2./Cp;

end

input temperature change:

function re = InputTempChange(t, input)

Vt0 = 1.5; %m3

A = 1.33; %m2

Cvpm = 2.34e-5;

Pvp = 9.;

F1ss = 1e-3; %m3/s

F2ss = F1ss;

C1ss = 1.; %kg/kg

T1ss = 350.; %k

ro = 1.e3; %kg/m3

Cp = 1; %kj/kg/k

Hrx = 2.e3; %j/kg

k0 = 32.; %1/s

R = 8.31; %J/k

Ea = 42.\*10.^3; %j

Tc = 295.; %k

UA = 5; %kj/k/s

Fcmax = 90.; %L/s

Pc = 9.; %psi

C2ss = C1ss.\*0.2;

g= 9.8;% m/s2

%secondary parameters:

Cvp = (Pvp - 3)./12.\*Cvpm;

Fc = (Pc - 3)./12.\*Fcmax;

F1 = F1ss;

F2 = sqrt(input(1).\*ro.\*g) .\* Cvp;

Vt = input(1) .\* A;

C2 = input(2);

T2 = input(3);

T1 = T1ss .\* 1.1;

%-------------------------------------

re(1,1) = (F1 - F2)./A;

re(2,1) = F1.\*C1ss./Vt - F2.\*C2./Vt-k0.\*exp(-Ea./R./T2).\*C2;

re(3,1) = F1.\*T1./Vt - F2.\*T2./Vt + UA.\*(Tc - T2)./Vt./Cp./ro + Hrx.\*k0.\*exp(-Ea./R./T2).\*C2./Cp;

end

Solving part 3:

[t1, ans1] = ode45('FlowRateChange', [0 10000], [0.7454, 0.1420, 590]);

plot(t1, ans1(:,2),'b');

title('Part3, C2-plot');

ylabel('C2(kg/kg)');

xlabel('Time(s)');

figure;

plot(t1, ans1(:,3),'b');

title('Part3, T2-plot');

ylabel('T2(kg/kg)');

xlabel('Time(s)');

figure;

plot(t1, ans1(:,1),'b');

title('Part3, h-plot');

ylabel('h(m)');

xlabel('Time(s)');

Solving part 4:

[t1, ans1] = ode45('InputTempChange', [0 6000], [0.7454, 0.1420, 590]);

plot(t1, ans1(:,2),'b');

title('Part4, C2-plot');

xlabel('Time(s)');

ylabel('C2(kg/kg)');

figure;

plot(t1, ans1(:,3),'b');

title('Part4, T2-plot');

xlabel('Time(s)');

ylabel('T2(C)');

solving part 5:

[t1, ans1] = ode45('CoolingTempChange', [0 6000], [0.7454, 0.1420, 590]);

plot(t1, ans1(:,2), 'b');

title('Part5, C2-plot');

xlabel('Time(s)');

ylabel('C2(kg/kg)');

figure;

plot(t1, ans1(:,3),'b');

title('Part5, T2-plot');

xlabel('Time(s)');

ylabel('T2(C)');

solving for transfer function with state space:

%project1 parameters

A = 1.33; %m2

Cvpm = 2.34e-5;

Pvp = 9;

F1ss = 1e-3; %m3/s

F2ss = F1ss;

C1ss = 1; %kg/kg

T1ss = 350; %k

ro = 1e3; %kg/m3

Cp = 1;

Hrx = 2e3; %kj/kg

k0 = 32; %1/s

R = 8.31./1000; %kJ/k

Ea = 42; %kj

Tcss = 295; %k

UA = 5; %kj/k/s

Fcmax = 90; %L/s

Pc = 9; %psi

T2ss = 590; %from 'fucking balance'

g= 9.8;% m/s2

%secondary parameters:

Cvp = (Pvp - 3)./12.\*Cvpm;

Fc = (Pc - 3)./12.\*Fcmax;

hss = (F2ss.^2)./((Cvp.^2).\*ro.\*g);

Vtss = A.\*hss;

C2ss=C1ss./(1+(k0.\*exp(-Ea./(R.\*T2ss)).\*Vtss)./F1ss);

%State space:

% x = [T2p;C2p;F2p]

% xdot = [dT2p/dt;dC2p/dt;dF2p/dt]

a11=-Cvp.\*sqrt(ro.\*g)./sqrt(hss)+UA./(ro.\*Cp.\*A.\*hss)-Hrx.\*k0.\*Ea.\*exp(-Ea./(R.\*T2ss)).\*C2ss./ ...

(Cp.\*R.\*T2ss); %T2'

a12=Hrx.\*k0./Cp.\*exp(-Ea./(R.\*T2ss)); %C2'

a13=-T1ss.\*F1ss./(A.\*(hss.^2))+Cvp.\*sqrt(ro.\*g).\*T2ss./(2.\*(hss.^(1.5)))+UA.\*Tcss./(ro.\*Cp.\*A.\*(hss.^2))...

-UA.\*T2ss./(ro.\*Cp.\*A.\*(hss.^2)); %h'

a21=k0.\*Ea./(R.\*T2ss).\*exp(-Ea./(R.\*T2ss)).\*C2ss; %T2' for dc2'/dt

a22=-Cvp.\*sqrt(ro.\*g.\*hss)./(hss.\*A)-k0.\*exp(-Ea./(R.\*T2ss)); %C2'

a23=C2ss.\*Cvp.\*sqrt(ro.\*g)./(2.\*A.\*(hss.^1.5))-F1ss.\*C1ss./(hss.^2.\*A); %h'

a31=0;%T2' for dh'/dt

a32=0; %C2'

a33=-Cvp.\*sqrt(ro.\*g)./(2.\*A.\*sqrt(hss)); %h'

%Matrix A:

A1=[a11, a12, a13; a21, a22, a23; a31, a32, a33];

% U = [T1p;C1p;F1p;Tcp];

b11=F1ss./(A.\*hss); %T1p for dT2p/dt

b12=0; %C1p

b13=T1ss./(A.\*hss); %F1p

b14=-UA./(ro.\*Cp.\*A.\*hss); %Tcp

b21=0;%T1p for dC2p/dt

b22=F1ss./(A.\*hss);%C1p

b23=C1ss./(A.\*hss);%F1p

b24=0;%Tcp

b31=0;%T1p for dF2p/dt

b32=0;%C1p

b33=1./A;%F1p

b34=0;%Tcp

%Matrix B:

B = [b11,b12,b13,b14;b21,b22,b23,b24;b31,b32,b33,b34];

%Selector matrix C:

%C = [1 1 1;1 1 1;1 1 1; 1 1 1];

syms s;

%Identity matrix

I = eye(3,3);

%Transfer function

G = vpa(inv((s\*I-A1))\*B,3);

%test response here:

%U = [10; 0; 0; 0];

%Y = C\*G\*U;