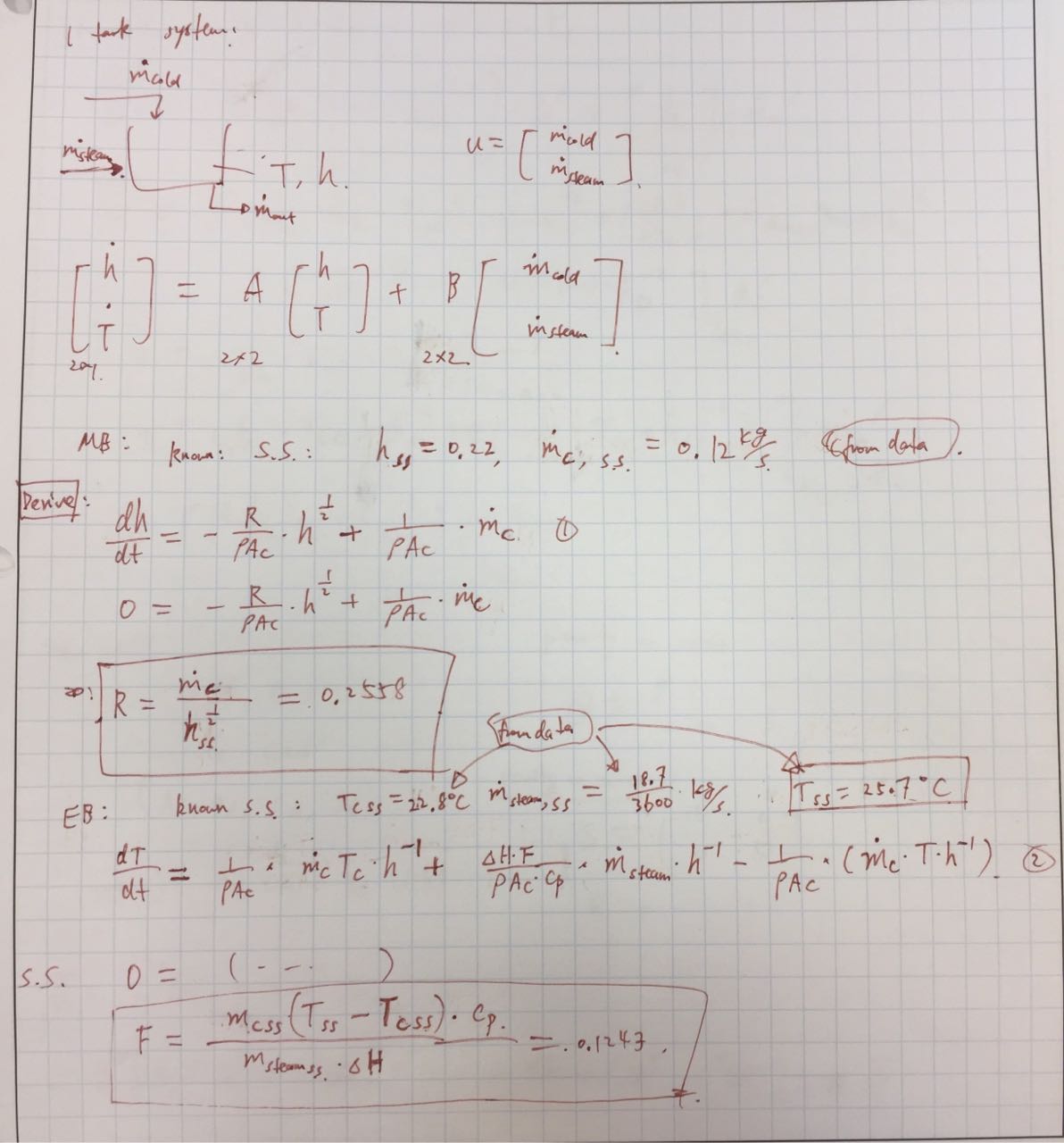
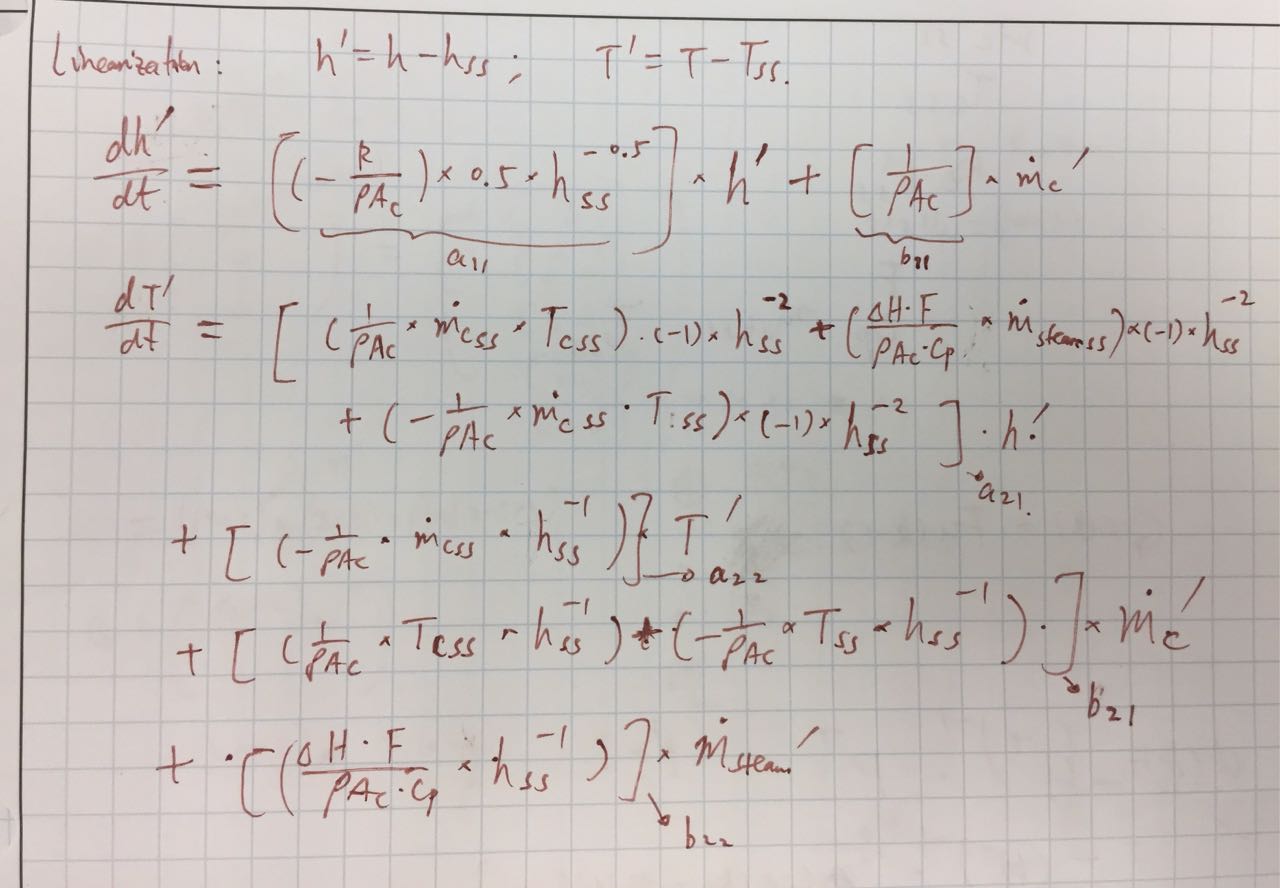
In this part of experiment, A MPC controller was designed with certain constraints and optimization criteria for one-tank system AND controlled response of the discretized model for certain setpoint changes was simulated.

The **linearized discrete time state space representation** is derived as shown below. 



The response of discretized model for the set point changes (level +0.05 m, temperature + 5 C) is shown as the following figures. From the figure, it is illustrated that the response of the system to the set point change finally stabilizes to steady state value. Although for steam flow rate, deviation does not go back to 0, it goes back and stabilizes at 10^-5 magnitude which is close enough to 0. Consequently, the conclusion can be made that the MPC is well-designed for this one tank system.



Figure 7. The response of discretized model for the set point change of states ( level +0.05 m, temperature + 5 C)

Matlab code for MPC:

See ‘PartI\_Q6.m’ file in package in the given link: <https://github.com/Haihan-W/UALabDataAnalysis/tree/master/LAB_MPC_Controller>

(3.1). main code:

clear all

%tolerance

options.StepTolerance = 1e-10;

rho=1000;

D\_T=0.145;

Ac=0.25\*pi()\*(D\_T)^2;

delta\_H=2100;

cp=4.1855;

%SS inputs

mcss=7.12/60;

Tcss=23;

msteamss=18.36/3600;

%SS states

hss=0.22;

Tss=25.7;

F = (mcss\*Tss-mcss\*Tcss)/(msteamss\*delta\_H/cp);

R=mcss/(hss.^0.5);

A=zeros(2);

B=zeros(2);

% f1=dh1'/dt=a11\*h1'+b11\*mh'

A(1,1)=(-R/(rho\*Ac))\*0.5\*hss^-0.5;

B(1,1)=(1/(rho\*Ac));

%f4=dT1'/dt=a41\*h1'+a44\*T1'+b41\*mh'+b43\*msteam

A(2,1)=((mcss\*(Tss-Tcss)-msteamss\*F\*delta\_H/cp)/rho/Ac/hss^2)

A(2,2)=(-1/(rho\*Ac)\* mcss\*hss^-1);

B(2,1)=(1/(rho\*Ac)\*Tcss\*hss^-1)+(-1/(rho\*Ac)\*Tss\*hss^-1);

B(2,2)=(delta\_H\*F/(rho\*Ac\*cp)\*hss^-1);

% sampling time = 1 s

C=eye(2)

D=0

sys=ss(A,B,C,D)

sysd = c2d(sys,1)

Ad=sysd.a;

Bd=sysd.b;

Cd=sysd.c;

Dd=sysd.d;

Q=[1,0;0,1];

S=[16,0;0,5];

Rm=0;

Qbar=dlyap(Ad',Cd'\*Q\*Cd);

% N=5( horizon)

N=20

Fm=zeros(2\*N,2)

Fm(1:2,1:2)=S;

G=zeros(2\*N,2);

for i=1:2:2\*N

G(i:i+1,1:2)=Bd'\*Qbar\*Ad^((i+1)/2);

end

H=zeros(2\*N);

for i=1:2:2\*N

for j=1:2:2\*N

if i==j

H(i:i+1,j:j+1)=Bd'\*Qbar\*Bd+R+2\*S;

elseif i==1 && j==3

H(i:i+1,j:j+1)=Bd'\*Ad'\*Qbar\*Bd-S;

elseif j==1 && i==3

H(i:i+1,j:j+1)=(Bd'\*Ad'\*Qbar\*Bd-S)';

elseif i< j %upper diagonal

power=(j+1)/2-1-((i+1)/2-1)

H(i:i+1,j:j+1)=Bd'\*(Ad')^(power)\*Qbar\*Bd;

elseif i>j

power=(i+1)/2-1-((j+1)/2-1)

H(i:i+1,j:j+1)=(Bd'\*(Ad')^(power)\*Qbar\*Bd)';

end

end

end

H=(H+H')/2

% constraint

LB=zeros(2\*N,1)

UB= zeros(2\*N,1)

for i=1:N

UB(i\*2-1,1)=9.2;

UB(i\*2,1)=30;

end

delta\_u1max=12;

delta\_u2max=2;

w=zeros(2\*N)

for i=1:2\*N

if i<= N

w(i,2\*i-1)=1

if i>1

w(i,(i-1)\*2-1)=-1

end

end

if i>N

w(i,2\*(i-20))=1

if i>N+1

w(i,(i-20-1)\*2)=-1

end

end

end

I=eye(2\*N)

Am=[I;-I;w;-w]

% bm

bm1=zeros(2\*N,1)

bm2=zeros(2\*N,1)

bm3=zeros(2\*N,1)

bm4=zeros(2\*N,1)

u1m=9.2

u2m=30

for i=1:2:2\*N

bm1(i,1)= u1m;

bm1(i+1,1)=u2m;

end

for i=1:2\*N

if i<=N

bm3(i,1)=delta\_u1max;

else

bm3(i,1)=delta\_u2max;

end

end

bm4=bm3;

x(:,1)=[0.05;5];

u(:,1)=[0,0];

kend=600

for k=2:kend

%x(:,k)=Ad\*x(:,k-1)+Bd\*u(:,k-1);

options = odeset('AbsTol',1e-10,'RelTol',1e-10);

initialu=u(:,k-1);

initialu(1,1)=u(1,k-1)+mcss;

initialu(2,1)=u(2,k-1)+msteamss;

initialx=x(:,k-1);

initialx(1,1)=x(1,k-1)+hss;

initialx(2,1)=x(2,k-1)+Tss;

[t,Tandh]=ode45(@Q6ode, [k-2:1:k+100], initialx ,options,R,F, initialu);

x(1,k)=Tandh(2,1)-hss;

x(2,k)=Tandh(2,2)-Tss;

bm3(1,1)=delta\_u1max+u(1,k-1);

bm3(N+1,1)=delta\_u2max+u(2,k-1);

bm4(1,1)=delta\_u1max-u(1,k-1);

bm4(N+1,1)=delta\_u2max-u(2,k-1);

bm=[bm1;bm2;bm3;bm4];

Fquad=G\*x(:,k)-Fm\*u(:,k-1);

u\_mpc=quadprog(H,Fquad,Am,bm,[],[],LB,UB);

u(1,k)=u\_mpc(1,1);

u(2,k)=u\_mpc(2,1);

end

figure(1)

subplot(2,2,1)

plot(1:kend,x(1,:))

xlabel('k')

ylabel('h'' [kg/s]')

title('h'' vs time')

subplot(2,2,2)

plot(1:kend,x(2,:))

xlabel('k')

ylabel('T'' [C]')

title('T'' vs time')

subplot(2,2,3)

plot(1:kend,u(1,:))

xlabel('k')

ylabel('mcold'' [kg/s]')

title('mcold'' vs time')

subplot(2,2,4)

plot(1:kend,u(2,:))

xlabel('k')

ylabel('msteam'' [kg/s]')

title('msteam'' vs time')

(3.2). odefunction:

function dydt=T\_ode\_dynamic(t,Tandh,R,F,inpvec)

h=Tandh(1)

T=Tandh(2)

%tolerance

options.StepTolerance = 1e-10;

rho=1000;

D\_T=0.145;

Ac=0.25\*pi()\*(D\_T)^2;

delta\_H=2100;

cp=4.1855;

Tc=23;

hss=0.22;

Tss=25.7;

msteamss=18.36/3600;

mcss=7.12/60;

%input vector

mc=inpvec(1,1)

msteam=inpvec(2,1)

dydt(1)=-R/(rho\*Ac)\*h^0.5+1/(rho\*Ac)\*mc

dydt(2)=(mc\*Tc/(rho\*Ac)+msteam\*delta\_H\*F/(rho\*Ac\*cp))\*h^-1 -mc/(rho\*Ac)\*(T\*h^-1)

dydt=[dydt(1);dydt(2)]

end