Author Lucky Yerimah

```
%
% Simulation file to run linear CSTR Plant using MPC
% This simulation file requires the following function files to run
%       linearmodSS to generate the linear model
%       lincstrplant for linear plant
%

clear all

% Parameters used for the linear plant

parameters = [102 294 -69.71*10^6 1.205 20.75*10^6 69.71*10^6 8314 801 3137 ...
    851 101 10.1 294 1000 4183 294 339.7022 323.7669 0.0842 0.04377 0.011];
Tr_ss = 339.7022;
Fj_ss = 0.011;

% a separate file linearmodSS contain the linear plant equations
% Parameters are passed to linearmodSS to obtain state space matrics

[A,B,C,D] = linearmodSS(parameters);
eig(A)
```

```
ans = 3×1
   -0.0407
   -0.0005
   -0.0034
```

```

sys = ss(A,B,C,D);
Gz = c2d(sys,100); % sample time is in seconds.
[Amod,Bmod,Cz,Dz] = ssdata(Gz);
Aplant = A;
Bplant = B;
tbeng = 0;
deltaT = 100;
```

MPC With Perfect Model

```
P = input('Prediction Horizon [100] ')
```

```
P = 100
```

```
M = input('Control Horizon [1] ')
```

```
M = 1
```

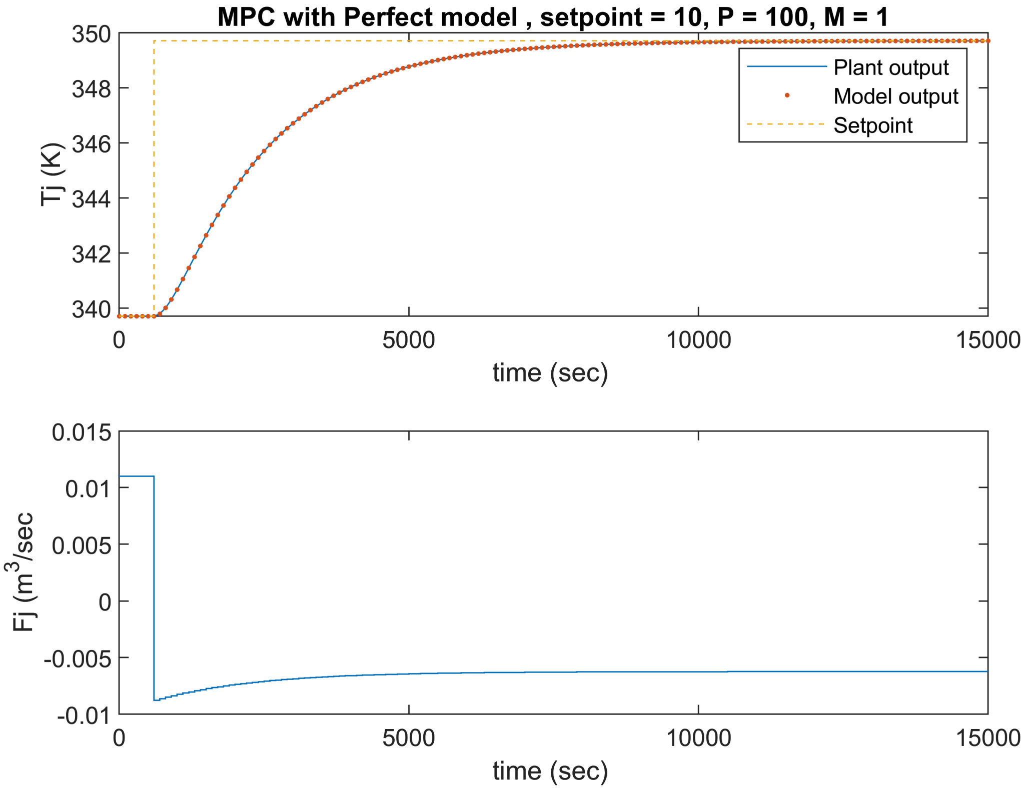
```
rsp = input('[Setpoint [0/1/10/20... degree K] ')
```

```
rsp = 10
```

```
Sf = zeros(P,M); % initial zeros for DMC matrix

H = 0;
```

```
for j=1:P
    H = H + Cz*Amod^(j-1)*Bmod;
    S(j,1) = H;
    Cphi(j,:) = Cz*Amod^(j);
end

for j=1:M
    zero_DMC = zeros(j-1,1);
    Sf(:,j) = cat(1,zero_DMC,S(1:end+1-j,1)); %DMC
end
Wu = 0;
Kmat = (Sf'*Sf+Wu)\Sf';

time = tbeng:deltaT:15000; % simulation runs for 5 hours (18000 sec)
x0 = [0;0;0];
%x0 = [0;0;0]; % initial conditions for the states
x(:,1) = x0;
y(1) = Cz*x(:,1); % measured output
xmod(:,1) = x0;
ymod(1) = Cz*xmod(:,1);
u_ini = 0;
d = y - ymod;

for k = 1:length(time)-1

    if k < 7
        r(k) = 0;
        r_sp = ones(P,1)*r(k);
        free_resp = Cphi*xmod(:,k) + S*u_ini + ones(P,1)*d(k); % free response

    else
        r(k) = rsp;
        r_sp = ones(P,1)*rsp;
         free_resp = Cphi*xmod(:,k) + S*u(k-1) + ones(P,1)*d(k);
    end

     E = r_sp - free_resp; % error term
    uf = Kmat*E; % vector of calculated control moves
    du(k) = uf(1); % implement only first move

    if k == 1
        u(k) = u_ini + du(k);
    else
        u(k) = u(k-1) + du(k);
    end


[tdummy,xdummy] = ode45(@(t,x) lincstrplant(t,x,Aplant,Bplant,u(k)), [time(k) time(k+1)], x0);
x0= xdummy(end,:)'; % assigning states variables as input for next iteration
x(:,k+1) = xdummy(end,:)';

y(k+1) = Cz*x(:,k+1); % output as last value of xnew
xmod(:,k+1) = Amod*xmod(:,k) + Bmod*u(k);
ymod(k+1) = Cz*xmod(:,k+1);
```

```
d(k+1) = y(k+1) - ymod(k+1);


end
r(k+1) = r(k);
u(k+1) = u(k);
figure(100);
subplot(2,1,1)
plot(time,y+Tr_ss,time,ymod+Tr_ss,'.')
hold on
stairs(time,r+Tr_ss,'--')
hold off
title(['MPC with Perfect model , setpoint = ', num2str(rsp), ', P = ',num2str(P), ', M = ', num
ylabel('Tj (K)')
xlabel('time (sec) ')
legend('Plant output', 'Model output', 'Setpoint')


subplot(2,1,2)
stairs(time,u+Fj_ss)
ylabel('Fj (m^3/sec')
xlabel('time (sec) ')
```

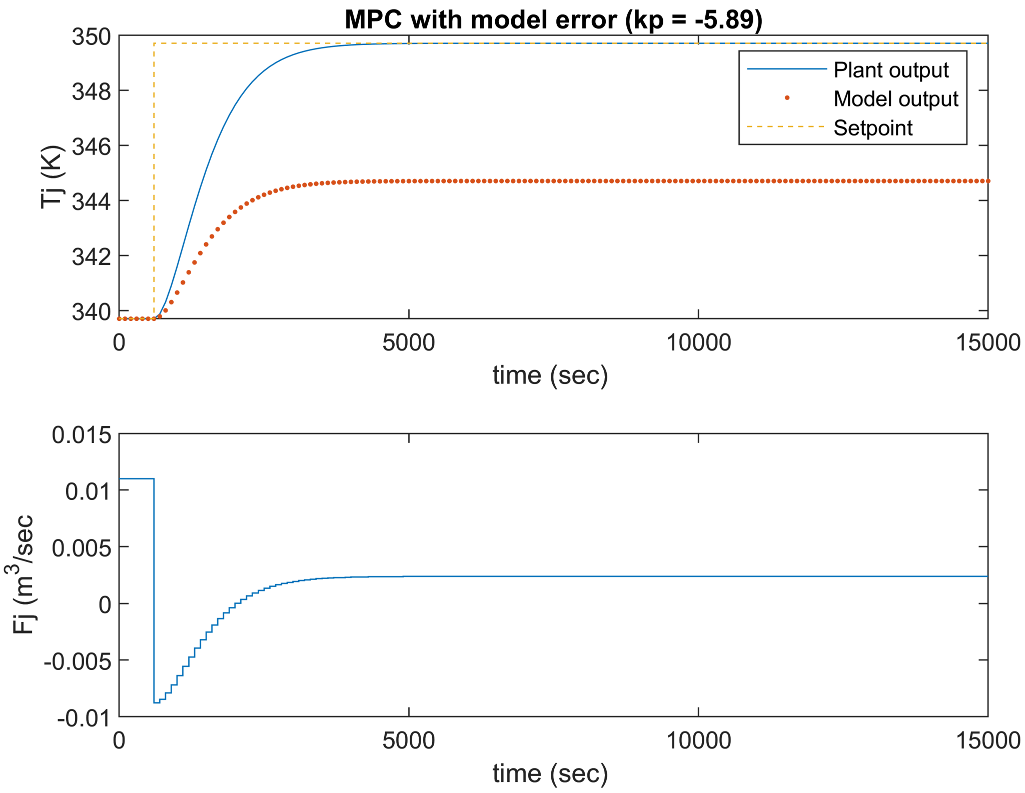


MPC With Model Error

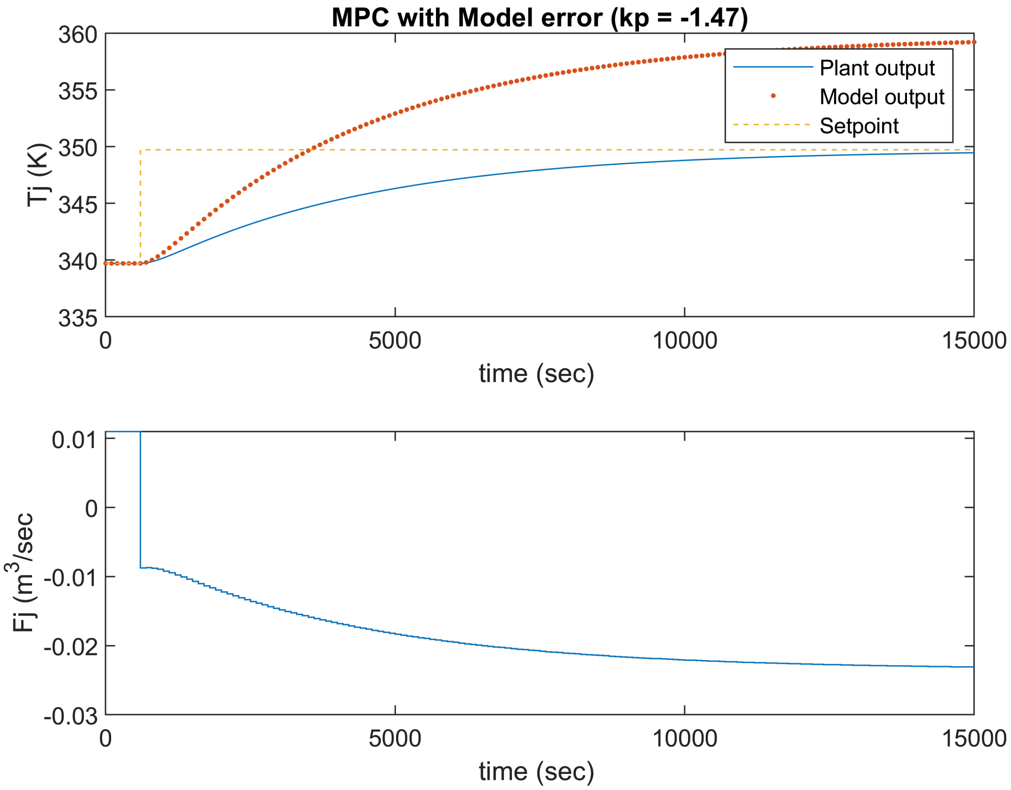
```
Bplant1 = [0 0;0 0;-5.8944 -1.4736]; % vector of plant gains
```

```
% cells are used to hold each of the plotting variables before being
% cleared
  yout = cell(2,1);
  uinput = cell(2,1);
  ymodout = cell(2,1);

for i = 1:2

x0 = [0;0;0]; % initial conditions for the states
x1(:,1) = x0;
y1(1) = Cz*x1(:,1); % measured output
xmod1(:,1) = x0;
ymod1(1) = Cz*xmod1(:,1);
d1 = y1 - ymod1;
u_ini = 0;

for k = 1:length(time)-1

    if k < 7
        r1(k) = 0;
        r_sp = zeros(P,1)*r1(k);
        free_resp = Cphi*xmod1(:,k) + S*u_ini + ones(P,1)*d1(k); % free response

    else
        r1(k) = rsp;
        r_sp = ones(P,1)*rsp;
         free_resp = Cphi*xmod1(:,k) + S*u1(k-1) + ones(P,1)*d1(k);
    end

     E1 = r_sp - free_resp; % error term
    uf = Kmat*E1; % vector of calculated control moves
    du1(k) = uf(1); % implement only first move

    if k == 1
        u1(k) = u_ini + du1(k);
    else
        u1(k) = u1(k-1) + du1(k);
    end


[tdummy,xdummy] = ode45(@(t,x) lincstrplant(t,x,Aplant,Bplant1(:,i),u1(k)), [time(k) time(k+1)]
x0= xdummy(end,:)'; % assigning states variables as input for next iteration
x1(:,k+1) = xdummy(end,:);

y1(k+1) = Cz*x1(:,k+1); % output as last value of xnew
xmod1(:,k+1) = Amod*xmod1(:,k) + Bmod*u1(k);
ymod1(k+1) = Cz*xmod1(:,k+1);
d1(k+1) = y1(k+1) - ymod1(k+1);


end

r1(k+1) = r1(k);
```

```
u1(k+1) = u1(k);

yout{i} = y1;
uinput{i} = u1;
ymodout{i} = ymod1;

clear u1; clear y1; clear ymod1;
end


figure(200);
subplot(2,1,1)
plot(time,yout{1}+Tr_ss,time,ymodout{1}+Tr_ss,'.');
hold on
stairs(time,r1+Tr_ss,'--')
hold off
title('MPC with model error (kp = -5.89)')
ylabel('Tj (K)')
xlabel('time (sec) ')
legend('Plant output', 'Model output', 'Setpoint')

subplot(2,1,2)
stairs(time,uinput{1}+Fj_ss)
ylabel('Fj (m^3/sec')
xlabel('time (sec) ')
```



```
% new figure

figure(300);
subplot(2,1,1)
plot(time,yout{2}+Tr_ss,time,ymodout{2}+Tr_ss,'.');
hold on
stairs(time,r1+Tr_ss,'--')
hold off
title('MPC with Model error (kp = -1.47)')
ylabel('Tj (K)')
xlabel('time (sec) ')
legend('Plant output', 'Model output', 'Setpoint')

subplot(2,1,2)
stairs(time,uinput{2}+Fj_ss)
ylabel('Fj (m^3/sec')
xlabel('time (sec) ')
```


MPC with Model error (kp = -1.47)
Tj (K)
time (sec)
Plant output
Model output
Setpoint
Fj (m^3/sec
time (sec)