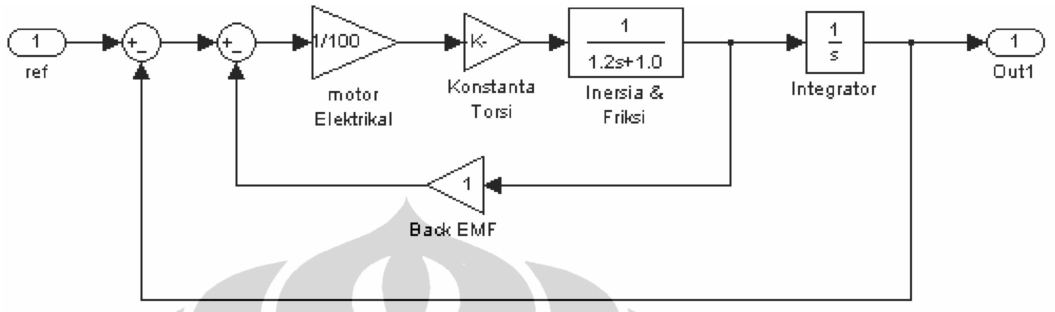
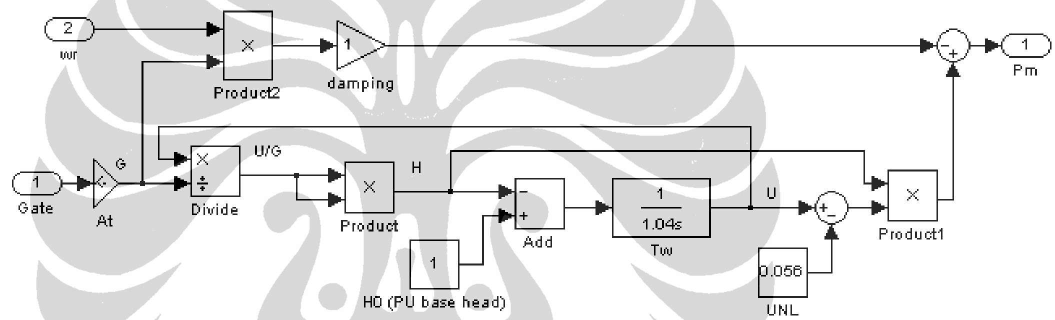
LAMPIRAN

Simulink Pemodelan sistem PLTMH

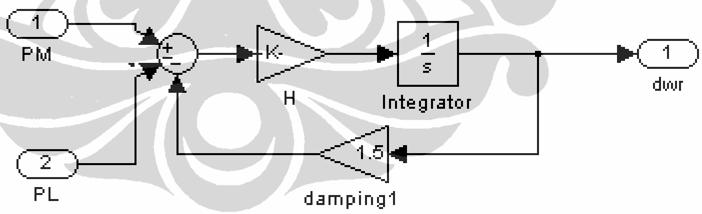
- Pemodelan dc motorservo



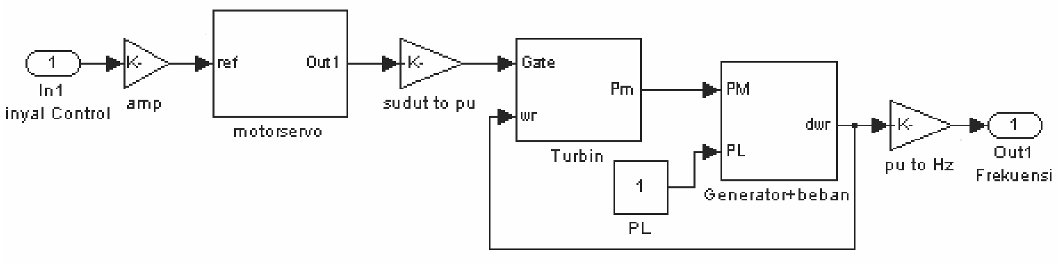
- Pemodelan Turbin air



- Pemodelan Generator (swing equation) + beban



- Sistem PLTMH



Listing Program

- **MPC *unconstraint* dengan gangguan**

```
clear
clc
%Model Linier
A=[-0.1573 0.4606 0 -0.01784;0 -2.166 0 0.1042;0 0 -9.167 -5;0 0
0.8333 0];
B=[0 -0.1667;0 0;120 0;0 0];
C=[50 0 0 0];
D=0;
x0=[0.88136;0.92352;0;19.2];
u0=19.200;
y0=44.0678;
Linier = ss(A,B,C,D); % ubah ke bentuk State Space
% Bentuk Diskrit dengan Time Sampling 3 s
dis=c2d(Linier,3);
a1 = dis.a;
b1 = dis.b(:,1);
c1 = dis.c;
d1 = dis.d;
gd = dis.b(:,2);
a=a1;
b=b1;
c=c1;
d=0;
g=gd;
% Inisialisasi simulasi
state = 4;
in = 1;  %input
out = 1; % output
xnext=zeros(state,1);
uprev=zeros(in,1);
x=x0;
% Tuning parameter pengendali MPC
Hp = 6; %prediction horizon
Hu = 1; % control horizon
q = 1; r=1; %faktor bobot
Q = q*eye(Hp*out);  %control output u/ mempercepat output mencapai
set point
R=r*eye(Hu*in);   % control sinyal kendali
% Sinyal referens
T(1:1:40)=50;
T(40:1:90)=50;
T(90:1:180)=50;
T(180:1:200)=50;
%perubahan beban
v(1:1:30)=0;
v(30:1:50)=-0.1;
v(50:1:70)=0;
v(70:1:90)=0.05;
udist=v(1:1:90); %input ganggguan
ydist = udist;

% Matrix phi, psi, theta
temp = zeros(Hp*out,Hp*state);
temp1 = zeros(state*Hp,state);
```

```
temp2 = zeros(state*Hp,in);
temp3 = zeros(state*Hp,Hu*in);
gamma = zeros(Hp*out,Hu*in);
for j=1:Hp
    gamma((j-1)*out+1:j*out,(j-1)*in+1:j*in) = d;
    temp((j-1)*out+1:j*out,(j-1)*state+1:j*state) = c;
    temp1((j-1)*state+1:j*state,:) = eye(state);
    for i=1:j
        temp1((j-1)*state+1:j*state,1:state) = temp1((j-
1)*state+1:j*state,1:state)*a;
        temp2((j-1)*state+1:j*state,1:in) = temp2((j-
1)*state+1:j*state,1:in) + a^(i-1)*b;
    end
    temp3((j-1)*state+1:j*state,1:in) = temp2((j-
1)*state+1:j*state,1:in);
end
for j=2:Hu
    for i=j:Hp
       temp3((i-1)*state+1:i*state,(j-1)*in+1:j*in) = temp3((i-
2)*state+1:(i-1)*state,(j-2)*in+1:(j-1)*in);
    end
end
phi = temp*temp1;
psi = temp*temp2;
th = temp*temp3;
%ga = temp*[g zeros(state,1) zeros(state,1);a*g g
zeros(state,1);a^2*g a*g g] %hp=3
ga = temp*[g zeros(state,1) zeros(state,1) zeros(state,1)
zeros(state,1) zeros(state,1);
         a*g g zeros(state,1) zeros(state,1) zeros(state,1)
zeros(state,1);
         a^2*g a*g g zeros(state,1) zeros(state,1) zeros(state,1);
         a^3*g a^2*g a*g g zeros(state,1) zeros(state,1);
         a^4*g a^3*g a^2*g a*g g zeros(state,1);
         a^5*g a^4*g a^3*g a^2*g a*g g];  %Hp = 6
%NILAI KMPC (KONTANTA)
kmpc=inv(th'*Q*th+R)*th'*Q;
ypd = zeros(90,1);% Plant output with output disturbance

% Perhitungan keluaran prediksi (z(k)) %
for j=1:90;
    y(j) = c*x;
    ypd(j)=y(j)+ydist(j);
    dk = ypd(j)-y(j);
   %nilai error E(k)
   epp=T(j*2:j*2+2+2+1)'-phi*x-psi*uprev-
ga*[dk;dk;dk;dk;dk;dk];%hp=6
   %epp=T(j*2:j*2+2)'-phi*x-psi*uprev-ga*[dk;dk;dk]; %hp=3
   %epp=T(j*2:j*2+2+2+2+2)'-phi*x-psi*uprev-
ga*[dk;dk;dk;dk;dk;dk;dk;dk;dk]; %hp=9

   %nilai delta u = Kmpc * E
   du=kmpc*epp;
   u(:,j)=du(1,1,:)+ uprev;
   x=a*x+b*u(:,j)+g*udist(j);
   uprev=u(:,j);
end
```

```
figure
subplot(311),plot(0:89,y(1,:),'b-',0:89,T(1:2:90*2),'r-
','LineWidth',2);
title('Keluaran Sistem');
xlabel('Time [s]');
ylabel('Frekuensi [Hz]');
legend('Output','Set Point',0);
subplot(312),plot(0:89,udist,'r-','LineWidth',2)
title('Sinyal Kendali')
xlabel('Time [s]');
ylabel('beban');
subplot(313),plot(0:89,u(1,:),'b-','LineWidth',2)
title('Sinyal Kendali')
xlabel('Time [s]');
ylabel('sinyal kendali');
```

- **MPC *unconstraint* dan observer**

```
clear
clc
%Model Linier
A=[-0.1573 0.4606 0 -0.01784;0 -2.166 0 0.1042;0 0 -9.167 -5;0 0
0.8333 0];
B=[0;0;120;0];
C=[50 0 0 0];
D=0;
x0=[0.88136;0.92352;0;19.2];
u0=19.200;
y0=44.0678;
Linier = ss(A,B,C,D); % ubah ke bentuk State Space
% Bentuk Diskrit dengan Time Sampling 3 s
dis=c2d(Linier,3);
a = dis.a
b = dis.b
c = dis.c
d = dis.d
% Inisialisasi simulasi
state = 4;
in = 1;  %input
out = 1; % output
xnext=zeros(state,1);
uprev=zeros(in,1);
x_pred=zeros(state,1);
x=x0;
% Tuning parameter pengendali MPC
Hp = 9; %prediction horizon
Hu = 1; % control horizon
q = 1; r=1; %faktor bobot
Q = q*eye(Hp*out);  %control output u/ mempercepat output mencapai
set point
R=r*eye(Hu*in);   % control sinyal kendali
% Sinyal referens
T(1:1:50)=50;
T(50:1:100)=48;
T(100:1:180)=51;
T(180:1:200)=49;
% Matrix phi, psi, theta
temp = zeros(Hp*out,Hp*state);
```

```
temp1 = zeros(state*Hp,state);
temp2 = zeros(state*Hp,in);
temp3 = zeros(state*Hp,Hu*in);
gamma = zeros(Hp*out,Hu*in);
for j=1:Hp
    gamma((j-1)*out+1:j*out,(j-1)*in+1:j*in) = d;
    temp((j-1)*out+1:j*out,(j-1)*state+1:j*state) = c;
    temp1((j-1)*state+1:j*state,:) = eye(state);
    for i=1:j
        temp1((j-1)*state+1:j*state,1:state) = temp1((j-
1)*state+1:j*state,1:state)*a;
        temp2((j-1)*state+1:j*state,1:in) = temp2((j-
1)*state+1:j*state,1:in) + a^(i-1)*b;
    end
    temp3((j-1)*state+1:j*state,1:in) = temp2((j-
1)*state+1:j*state,1:in);
end
for j=2:Hu
    for i=j:Hp
       temp3((i-1)*state+1:i*state,(j-1)*in+1:j*in) = temp3((i-
2)*state+1:(i-1)*state,(j-2)*in+1:(j-1)*in);
    end
end
phi = temp*temp1;
psi = temp*temp2;
th = temp*temp3;

%NILAI KMPC (KONTANTA)
kmpc=inv(th'*Q*th+R)*th'*Q;
%Perhitungan gain observer
%uji observability
N=obsv(a,c);
rank(N);
%poles sistem
L=eig(dis);
L1=L(1);
L2=L(2);
L3=L(3);
L4=L(4);
%poles yang diinginkan
JJ=[(L1-0.3);(L2);(L3);(L4-0.2)];
poly(JJ);
%parameter Ke (matriks gain observer)
Ke=acker(a',c',JJ)';
% Perhitungan keluaran prediksi (z(k)) %
for j=1:90;
   y1(:,j)=c*x;   %X(k)=y(k)
   ypred=c*x_pred;
   %nilai error E(k)
   %epp=T(j*2:j*2+2+2+1)'-phi*x_pred-psi*uprev;%hp=6
   %epp=T(j*2:j*2+2)'-phi*x_pred-psi*uprev; %hp=3
   epp=T(j*2:j*2+2+2+2+2)'-phi*x_pred-psi*uprev; %hp=9
   %nilai delta u = Kmpc * E
   du=kmpc*epp;
   u(:,j)=du(1,1,:)+ uprev;

   %persamaan estimasi state observer
```

```
   x_pred = (a-Ke*c)*x_pred + (b)*u(:,j) + Ke*y1(:,j);
   x_pred_data(j,:)=x_pred;
   x=a*x+b*u(:,j);
   x_data(j,:)=x;
   uprev=u(:,j);
end
figure
subplot(211),plot(0:89,y1(1,:),'b-',0:89,T(1:2:90*2),'r-
','LineWidth',2);
title('Keluaran Sistem');
xlabel('Time [s]');
ylabel('Frekuensi [Hz]');
legend('Output','Set Point',0);
subplot(212),plot(0:89,u(1,:),'b-','LineWidth',2)
title('Sinyal Kendali')
xlabel('Time [s]');
ylabel('Tegangan [Volt]');
figure;
subplot(411), plot(0:89,x_pred_data(:,1),'b--
',0:89,x_data(:,1),'r-','LineWidth',2);
xlabel('Time [seconds]');
ylabel('Frekuensi [pu]');
legend('Estimated state','Real state',0);
subplot(412), plot(0:89,x_pred_data(:,2),'b--
',0:89,x_data(:,2),'r-','LineWidth',2);
xlabel('Time [seconds]');
ylabel('PM [pu]');
legend('Estimated state','Real state',0);
subplot(413), plot(0:89,x_pred_data(:,3),'b--
',0:89,x_data(:,3),'r-','LineWidth',2);
xlabel('Time [seconds]');
ylabel('Tegangan [V]');
legend('Estimated state','Real state',0);
subplot(414), plot(0:89,x_pred_data(:,4),'b--
',0:89,x_data(:,4),'r-','LineWidth',2);
xlabel('Time [seconds]');
ylabel('Sudut [pu]');
legend('Estimated state','Real state',0);
```