LAMPIRAN

Program Perhitungan Sistem Plant menggunakan Model Predictive Control

```
%inisialisasi parameter
P=10;%horizon predksi
M=1;%horizon kendali
N=50;%panjang model
w=0.0;%bobot
ysp=1;%output set point dari 0
timesp=1;%waktu perubahan set-point
delt=0.1;%interval waktu sampling
tfinal=6;%waktu simulasi akhir
noise=0;
%definisi waktu
tvec=0:delt:tfinal;
ksp=fix(timesp/delt);
kfinal=length(tvec);
%definisi vektor set-point
r=[zeros(1,ksp),ones(1,(kfinal-ksp))*ysp];
%///////////////definisi plant sebagai obyek LTI 'SISO'
////////////////////////
c=menu('Tulislah plant dalam:',' 1.statespace','
2.transferfunction',' 3.polezero',' 4.frquencyresponse');
if c==1 a=input('Tulis Matrix A : ');b=input('Tulis Matrix B : ');
    c=input('Tulis Matrix C : ');d=input('Tulis Matrix D :
');plant=ss(a,b,c,d);
elseif c==2
    nump=input('Tulis koefisien pembilang : ');denp=input('Tulis
koefisien penyebut : ');plant=tf(nump,denp);
elseif c==3
    zero=input('Tulis zero : ');pole=input('Tulis pole :
');K=input('Tulis Gain : ');
    plant=zpk(zero,pole,K);
elseif c==4
    resp=input('Tulis response : ');freq=input('Tulis frekuensi :
');plant=frd(resp,freq,'satuan','Hz');
end
plant=tf(plant);
%plant=s/(s*s - 1.4*s +0.45),it is continous
%define plant parameters here
% nump=[1];
% denp=[1,-1.4,0.45];
% plant=tf(nump,denp);
%discretize the plant
plant=c2d(plant,delt);
%/////////////////define model
here////////////////////////////////////
%assumption plant = model
model=plant;
% [numm,denm,tm]=tfdata(plant);
numm = get(model,'num'); numm = numm{:}; % Get numerator polynomial
denm = get(model,'den'); denm = denm{:}; % Get denominator
polynomial
numm
%define step response coefficient matrix
```

```
s=step(model,0:delt:N*delt);
%define free response i.e. Sp matrix for past control moves
for i=1:P
    for j=1:N-2
        if(i+j<=N-1)
            Sp(i,j)=s(i+j);
        else
            Sp(1,j)=0;
        end
    end
end
%define forced response i.e. Sf matrix for future and control moves
for i=1:P
    for j=1:M
        if i+1-j>0
            Sf(i,j)=s(i+1-j);
        else
            Sf(i,j)=0;
        end
    end
end
Sf
% Tentukan W matrix
W=w*eye(M,M);
%Tentukan Kmat dengan Kmat=(Sf'*Sf + W)^-1*Sf'
Kmat=inv(Sf'*Sf + W)*Sf';
% kondisi awal piant
ndenm=length(denm)-1;
nnumm=length(numm)-1;
umpast=zeros(1,nnumm);
ympast=zeros(1,ndenm);
% uu=zeros(1,kfinal);
% yy=zeros(1,kfinal);
% xinit=zeros(1,size(
% nump=[zeros(1,ndenp-nnump-1),nump]; % Pad numerator with leading
zeros
% numm=[zeros(1,ndenm-nnumm-1),numm];
uinit=0;
yinit=0;
%Inisialisasi  input vector
u=ones(1,min(P,kfinal))*uinit;
u
dist(1)=0;
y(1)=yinit;
% x(:,1)=xinit;
dup=zeros(1,N-2);
for k=1:kfinal
    [m,p]=size(Kmat);
    for i=1:p
        if k-N+i>0
            uold(i)=u(k-N+i);
        else
            uold(i)=0;
        end
```

```
    end
    dvec=dist(k)*ones(1,p);
    rvec=r(k)*ones(p,1);
    y_free=Sp*dup' + s(N)*uold'+dvec';
    E=rvec-y_free;
    delup(k)=Kmat(1,:)*E;
    if k>1
        u(k)=u(k-1)+delup(k);
    else
        u(k)=delup(k)+uinit;
    end
    %persamaan plant
    umpast=[u(k),umpast(1,1:length(umpast)-1)];
    y(k+1)=-denm(2:ndenm+1)*ympast'+numm(2:nnumm+1)*umpast';
    ympast=[y(k+1),ympast(1:length(ympast)-1)];
    %model prediction
    if k-N+1>0
        ymod(k+1)=Sf(1,1)*delup(k)+Sp(1,:)*dup'+s(N)*u(k-N+1);
    else
        ymod(k+1)=Sf(1,1)*delup(k)+Sp(1,:)*dup';
    end
    %disturbance compensation
    dist(k+1)=y(k+1)-ymod(k+1);
    %additive disturbance compensation
    %put input change into vector of past control moves
    dup=[delup(k),dup(1,1:N-3)];
end
%stairs plotting for input(zero order hold) and setpoint
[tt,uu]=stairs(tvec,u);
[ttr,rr]=stairs(tvec,r);
figure(1)
subplot(2,1,1)
plot(ttr,rr,'--',tvec,y(1:length(tvec)))
ylabel('y');
xlabel('time');
title('plant output');
subplot(2,1,2)
plot(tt,uu)
ylabel('u');
xlabel('time');
```