LAMPIRAN A

Kolom_loopterbuka.m

```
% Menentukan hubungan sistem loop terbuka
%
Kolom_model
%
% nilai singular dari G
omega = logspace(-4,2,100) ;
olp_g = frsp(G,omega) ;
olp_g82 = frsp(G82,omega) ;
figure(1)
vplot('liv,lm',vsvd(olp_g),'-',vsvd(olp_g82),'--')
grid
title('Singular Value Plots of G and G_4')
xlabel('Frequency (rad/min)')
ylabel('Magnitude')
%
% membentuk fungsi pembobotan
Kolom_bobot
%
omega = logspace(-4,2,100) ;
Wm_g = frsp(Wm1,omega) ;
figure(2)
vplot('liv,lm',Wm_g,'r-'), grid
title('Model frequency response')
xlabel('Frequency (rad/min)')
ylabel('Magnitude')
%
omega = logspace(-4,2,100) ;
wn_g = frsp(wn,omega) ;
figure(3)
vplot('liv,lm',wn_g,'m-'), grid
title('Sensor Noise Weight')
xlabel('Frequency (rad/min)')
ylabel('Magnitude')
omega = logspace(-6,2,100) ;
wp_g = frsp(wp,omega) ;
wpi_g = minv(wp_g) ;
figure(4)
vplot('liv,lm',wpi_g,'r-'), grid
title('Inverse of Performance Weighting Function')
xlabel('Frequency (rad/min)')
ylabel('Magnitude')
omega = logspace(-4,4,100) ;
wu_g = frsp(wu,omega) ;
figure(5)
vplot('liv,lm',wu_g,'r-'), grid
title('Control Action Weighting Function')
xlabel('Frequency (rad/min)')
```

```
ylabel('Magnitude')
%
% koneksi loop terbuka dengan fungsi pembobotan
systemnames = ' G W_Delta Wm Wn Wp Wu ' ;
inputvar = '[ pert{2}; ref{2}; noise{2}; control{2} ]' ;
outputvar = '[ W_Delta; Wp; Wu; -G-Wn; ref ]' ;
input_to_G = '[ pert+control ]' ;
input_to_W_Delta = '[ control ]' ;
input_to_Wm =  '[ ref ]' ;
input_to_Wn = '[ noise ]' ;
input_to_Wp = '[ G-Wm ]' ;
input_to_Wu = '[ control ]' ;
sysoutname = 'sys_ic' ;
cleanupsysic = 'yes' ;
sysic
```

kolom_model.m

```
% membentuk model distilasi kolom orde 6 dengan
konfigurasi LV
% Variabel yang didapat adalah:
% A         C         G4       Si      Uinit
% B         D         G        So      Xinit
% dimana:
%
% G4 - Model distilasi kolom orde 82 yang telah diskalakan
%
% G  - Model distilasi kolom orde 6 ,
%
% A,B,C,D - matriks dari G
% Si and So - penskalaan input and outputs
%    Note: Unscaled model is G4u = mmult(minv(So), G4,
minv(Si));
% Uinit - nilai nominal (steady state) untuk input
% Xinit - nilai nominal (steady state)  untuk  state (G4
gives the
%         deviation from this nominal state)
%-------------------------------------------
% Mencari Model
%-------------------------------------------
% Mensimulasikan model LV selama 5000 menit :
[t,x] = ode15s(@cola_lv,[0 5000],0.5*ones(1,82)');
Xinit = x(sel(size(x),1,1),:)';
%---------------------------------------
% Linierisasi model, G4u
%---------------------------------------
Ls = 2.70629; Vs = 3.20629; Fs = 1.0; zFs = 0.5;
Uinit = [Ls Vs Fs zFs]';
[A,B,C,D] = cola_lin(@cola_lv_lin,Xinit',Uinit');
G4u =  pck(A,B,C,D);
%---------------------------------------
```

```
% Model yang telah diskalakan, G4
% ------------------------------------
% Skala
Du = diag([1 1]);                  % max inputs (scalings)
Dd = diag([0.2 0.1]);              % max disturbances
(scalings)
De = diag([0.01 0.01]);            % max output errors
(scalings)
% This implies the folling in terms of the scaled model
G4:
   % Units for inputs (L,V):         1 = 1 kmol/min = F
(the feed rate)
   % Units for disturbance 1 (F):    1 = 0.2 kmol/min (20%
change)
   % Units for disturbance 2 (z_f): 1 = 0.1 mole fraction
units
   %                                        (20% change)
   % Units for outputs 1 and 2 (y_D and x_B): 1 = 0.01
mole
   %  fraction units
% Model yang telah diskalakan, G4:
Si = daug(Du,Dd); So = minv(De); % introduce scaling
matrices
G4 = mmult(So, G4u, Si);
[A82,B82,C82,D82] = unpck(G4);
A = A82; B = B82(:,1:2); C = C82; D = D82(:,1:2);
G82 = pck(A,B,C,D);
%
% Reduksi Model G
[syssb,hsig] = sysbal(G4);
G4_6 = hankmr(syssb,hsig,6,'d');
[A6,B6,C6,D6] = unpck(G4_6);
A = A6; B = B6(:,1:2); C = C6; D = D6(:,1:2);
G = pck(A,B,C,D);
%
clear t;clear x; clear G4u; clear G4_6;
clear Ls; clear Vs; clear Fs; clear zFs;
clear A6; clear B6; clear C6; clear D6;
clear Dd, clear De; clear Du;
clear hsig; clear sys1; clear syss; clear syssb; clear
sysu;
```

kolom_bobot.m

```
% Fungsi pembobotan untuk sistem kolom distilasi
%
% Bobot ketidakpastian
nuW_Delta1 = [2.2138 15.9537 27.6702  4.9050];
dnW_Delta1 = [1.      8.3412 21.2393 22.6705];
```

```
gainW_Delta1 = 1;
w_Delta1 = nd2sys(nuW_Delta1,dnW_Delta1,gainW_Delta1);
%
nuW_Delta2 = [2.2138 15.9537 27.6702  4.9050];
dnW_Delta2 = [1.      8.3412 21.2393 22.6705];
gainW_Delta2 = 1;
w_Delta2 = nd2sys(nuW_Delta2,dnW_Delta2,gainW_Delta2);
%
W_Delta = daug(w_Delta1,w_Delta2);
%
% model
nuWm1 = 1;
dnWm1 = [6.0^2  2*0.8*6.0  1];
gainWm1 = 1.0;
wm1 = nd2sys(nuWm1,dnWm1,gainWm1);
nuWm2 = 1;
dnWm2 = [6.0^2  2*0.8*6.0  1];
gainWm2 = 1.0;
wm2 = nd2sys(nuWm2,dnWm2,gainWm2);
wm12 = 0.;
wm21 = 0.;
Wm1 = sbs(wm1,wm12);
Wm2 = sbs(wm21,wm2);
Wm = abv(Wm1,Wm2);
%
% Bobot Kinerja
tol = 10^(-4);
nuWp = [9.5 3];
dnWp = [9.5 tol];
gainWp = 0.55;
wp = nd2sys(nuWp,dnWp,gainWp);
wp12 = 0.3;
wp21 = 0.3;
Wp1 = sbs(wp,wp12);
Wp2 = sbs(wp21,wp);
Wp = abv(Wp1,Wp2);
%
% Bobot Aksi Kontrol
nuWu = [1  1];
dnWu = [0.01   1];
gainWu = 8.7*10^(-1);
wu = nd2sys(nuWu,dnWu,gainWu);
Wu = daug(wu,wu);
%
% Shaping Filter untuk Noise
nuWn = [1 0];
dnWn = [1 1];
gainWn = 10^(-2);
wn = nd2sys(nuWn,dnWn,gainWn);
Wn = daug(wn,wn);
```

kolom_timedelay.m

```
% Pendekatan dari ketidakpastian time delay dengan
pertubasi multiplicative tidak terstruktur
%
hold off
omega = logspace(-2,2,200);
% 3rd order approximation
W_Delta = nd2sys([2.2138 15.9537 27.6702  4.9050], ...
                 [1.      8.3412 21.2393 22.6705]);
W_Delta_g = frsp(W_Delta,omega);
%
for k = 0.8:0.05:1.2;
    for tau = 0:0.1:1.0;
        for i = 1:200
            om = omega(i);
            pert(i) = sqrt((k*cos(om*tau)-1)^2 +
(k*sin(om*tau))^2);
        end
        magg = vpck(pert',omega');
        vplot('liv,m',W_Delta_g,'r-',magg,'c--'), grid on
        hold on
    end
end
xlabel('Frequency (rad/min)')
ylabel('Magnitude')
temp1 = 'Approximation of uncertain time delay';
temp2 = ' by multiplicative perturbation';
title([temp1 temp2])
```

kolom_ms.m

```
% Mu-synthesis of the Distillation Column system
%
DK_DEF_NAME = 'dk_col';
dkit
K_mu = k_dk4col;
K = K_mu;
```

dk_col.m

```
% Nominal plant interconnection structure
NOMINAL_DK = sys_ic;

% Number of measurements
NMEAS_DK = 4;

% Number of control inputs
```

```
NCONT_DK = 2;

% Block structure for mu calculation
BLK_DK = [1 1;1 1;4 4];

% Frequency response range
OMEGA_DK = logspace(-3,3,100);

AUTOINFO_DK = [1 4 1 4*ones(1,size(BLK_DK,1))];

NAME_DK = 'col';
%-------------------- end of dk_col ----------------------
---------%
```

mu_col.m

```
% mu-analysis of the distillation column
% closed-loop system
%
clp_ic = starp(sys_ic,K);
omega = logspace(-3,3,100);
clp_g = frsp(clp_ic,omega);
%
% nominal performance
nom_perf = sel(clp_g,[3:6],[3:6]);
figure(1)
vplot('liv,m',vnorm(nom_perf),'r-')
grid
title('Nominal performance')
xlabel('Frequency (rad/min)')
%
% robust stability
rob_stab = sel(clp_g,[1:2],[1:2]);
% Complex perturbations
blkrs = [1 1;1 1];
rbnds = mu(rob_stab,blkrs);
figure(2)
vplot('liv,lm',sel(rbnds,1,1),'r-',sel(rbnds,1,2),'c--')
grid
title('Robust stability')
xlabel('Frequency (rad/min)')
ylabel('mu')
disp(' ')
disp(['mu-robust stability: ' ...
     num2str(pkvnorm(sel(rbnds,1,1)))])
disp(' ')
%
% robust performance
rob_perf = sel(clp_g,[1:6],[1:6]);
blks = [blkrs;4 4];
```

```
rpbnds = mu(rob_perf,blks);
figure(3)
vplot('liv,m',sel(rpbnds,1,1),'r-',sel(rpbnds,1,2),'c--')
grid
title('Robust performance')
xlabel('Frequency (rad/min)')
ylabel('mu')
disp(' ')
disp(['mu-robust performance: ' ...
     num2str(pkvnorm(sel(rpbnds,1,1)))])
disp(' ')
```

kolom_parasimulink.m

```
% Parameter untuk simulasi dengan simulink
%
% Mensimulasikan model LV selama 20000menit:
[t,x] = ode15s(@cola_lv,[0 20000],0.5*ones(1,82)');
lengthx = size(x); Xinit = x(lengthx(1),:)';
%
% Nominal inputs
LT = 2.70629;                          % Reflux
VB = 3.20629;                          % Boilup
D = 0.5;                               % Distillate
B = 0.5;                               % Bottom products
F = 1.0;                               % Feedrate
zF = 0.5;                              % Feed composition
qF = 1.0;                              % Feed liquid
fraction
Uinit = [ LT VB D B F zF qF]';

clear x; clear t; clear LT; clear VB; clear D; clear B;
clear F; clear zF; clear qF;
%
% Menskalakan kontroler
S = daug(So,So);
K_u = mmult(K,S);
%
% Unpack the controller
[Ak,Bk,Ck,Dk] = unpck(K_u);
%
% Noise shaping filters for the unscaled plant
kf1 = 0.01/So(1,1); Tf11 = 1;   Tf12 = 1;
kf2 = 0.01/So(2,2); Tf21 = 1;   Tf22 = 1;

save kolom_parasimulink
```

SIMULINK Model of the Distillation Column System