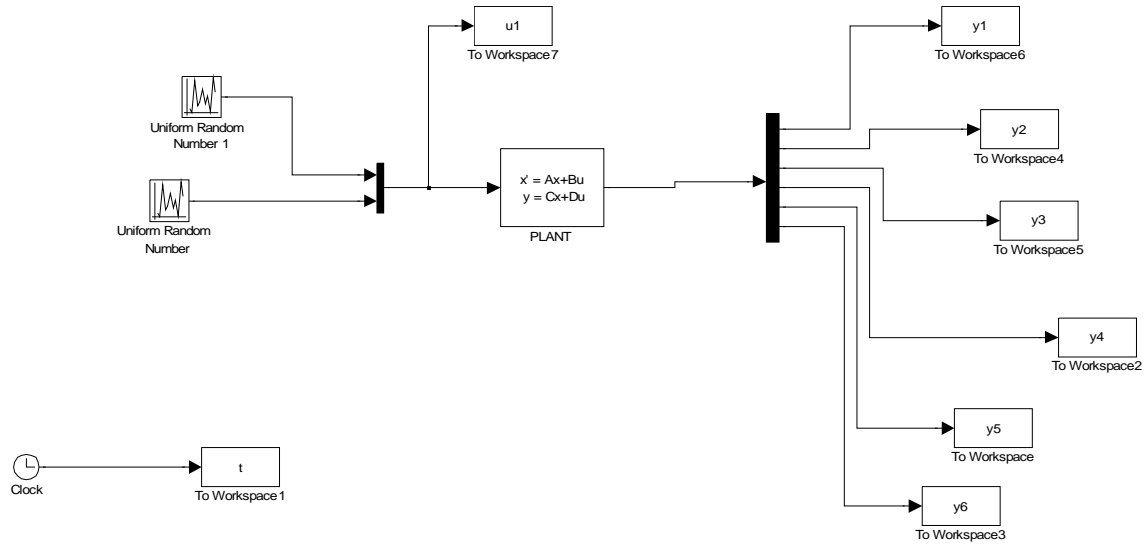
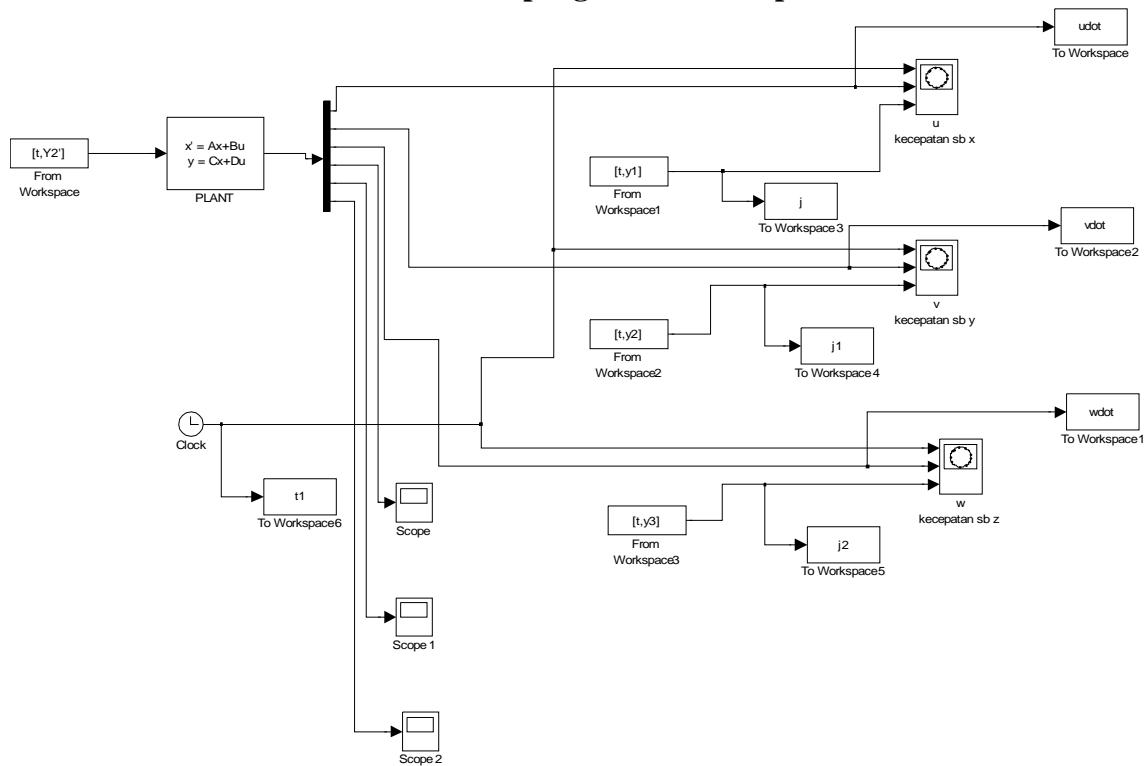


LAMPIRAN A

***** Model pengambilan data latih*****



***** Model hasil pengontrolan kecepatan*****



Input data dinamika kapal selam

```
u = 12.86;v = 8.86;w = 5.86;p = 0.35;q = 1.79;r = 1.80;pi = 5;theta = 5;B = 1800;
M = diag([215,265,265,40,80,80]);
D = diag([70+100*u , 100+200*v ,100+200*w , 30+50*p , 50+100*q , 50+100*r]);
g = [0;0;0;0.02*B*theta*pi;0.02*B*theta;0];
C = [0 -185*r 185*q 0 80*w -80*v ; 185*r 0 -185*p -80*w 0 30*u ; -185*q 185*p 0 80*v -30*u 0 ; 0
80*w -80*v 0 50*r+r^2 -80*q ; -80*w 0 30*u -50*r-30*r^2 0 40*p ; 80*v -30*u 0 80*q -40*p 0];
```

%persamaan dinamika Vppm:(jurnal neural network augmented identification of underwater vehicle models)

```
%V dot(ppm) = inv(M)*{T - C(Vppm)*Vppm - D(Vppm)*Vppm - g}
%T =[100 100 100 0 0 0]'
```

%persamaan dinamika sistem dalam bentuk state-space:

```
x = -C-D;
Appm = inv(M)*x;
Bppm = [0.4651 0;0.3774 0;0.3774 0;0 -22.5;0 -2.25;0 0];
Cppm = diag([1,1,1,0,0,0]);
Dppm = zeros(6,2);
```

Pengontrol JST#####

% ambil data latih dari workspace

```
P = [u1';y1';y2';y3']; % data latih
T = u1'; % target pelatihan
```

% jaringan saraf tiruan propagasi balik %

```
% 5 input dengan 4 hidden layer,2 output dengan fungsi aktivasi purelin
% hidden layer-1 :10 neuron,fungsi aktivasi tansig
% hidden layer-2 :30 neuron,fungsi aktivasi tansig
% hidden layer-3 :20 neuron,fungsi aktivasi logsig
% hidden layer-4 : 5 neuron,fungsi aktivasi logsig
% P dan T adalah vektor masukan dan keluaran dari jaringan saraf tiruan
```

```
net = newff(minmax(P),[10 20 5 30
2],{'tansig','tansig','logsig','logsig','purelin'},'traingdx','learngdm
');
[Y1,e,perf]=sim(net,P,[],[],T); % simulasi keluaran awal jaringan
sebelum training,Y1=hasil simulasi, e = error
```

% parameter training jaringan:

```
net.trainParam.epochs = 20000;
net.trainParam.goal = 1e-4;
net.trainParam.mc = 0.001;
net.trainParam.lr = 0.01;
```

% training jaringan

```
[net,tr,Y2,E,Pf,Af] = train(net,P,T,[],[]);%Y2=hasil training jaringan,
E = error jaringan setelah training
```

```
% simpan hasil training
save hasil_training;
```

#####Transformasi kecepatan ke posisi#####

```
% Transformasi dari kecepatan menjadi posisi dengan menggunakan data
dari hasil pelatihan JST
```

```
fk = [-0.2773 0.8967 -0.9363;0.8972 0.6194 3.0297;-1.1249 0.2656 -
0.2773]; %faktor pengali untuk transformasi kecepatan ke posisi
PoK = fk*[udot';vdot';wdot'];
```

```
%Transformasi dari kecepatan menjadi posisi dengan menggunakan data
dari
%target kecepatan yang ingin dicapai
```

```
PoT = fk*[j';j1';j2'];
```

Plot hasil pelatihan JST#####

```
% u1 adalah input ke plant sekaligus target pelatihan jaringan saraf
tiruan yang ingin dicapai
% Y1 adalah keluaran awal dari jaringan saraf tiruan(sebelum dilakukan
training)
% Y2 adalah hasil akhir training jaringan
```

```
subplot(211),plot(t,u1,'x',t,Y1,'o'),title('Keluaran Awal JST')
hold
subplot(212),plot(t,u1,'x',t,Y2,'o'),title('Hasil Training
JST'),xlabel('waktu')
```

#####Plot hasil pengontrolan kecepatan#####

```
subplot(311), plot(t1,udot,'b',t1,j,'r',t2,us,'g'),title('HASIL
PENGONTROLAN KECEPATAN')
xlabel('kecepatan sumbu x')
hold
subplot(312),plot(t1,vdot,'b',t1,j1,'r',t2,vs,'g')
xlabel('kecepatan sumbu y')
hold
subplot(313),plot(t1,wdot,'b',t1,j2,'r',t2,ws,'g')
xlabel('kecepatan sumbu z')
```

#####Plot hasil pengontrolan posisi#####

```
pok1=PoK(1,:);  
pok2=PoK(2,:);  
pok3=PoK(3,:);  
pot1=PoT(1,:);  
pot2=PoT(2,:);  
pot3=PoT(3,:);
```

```
pok11=PoK1(1,:);  
pok22=PoK1(2,:);  
pok33=PoK1(3,:);
```

```
subplot(311),plot(t1,pot1,'r',t1,pok1,'b',t2,pok11,'g'),title('HASIL  
PENGONTROLAN POSISI')  
xlabel('posisi sumbu x')  
subplot(312),plot(t1,pot2,'r',t1,pok2,'b',t2,pok22,'g')  
xlabel('posisi sumbu y')  
subplot(313),plot(t1,pot3,'r',t1,pok3,'b',t2,pok33,'g')  
xlabel('posisi sumbu z')
```