Una Drakulić_april 2019

INTELIGENTNI SISTEMI_MATERIJAL ZA AUDITORNE VJEŽBE

STATIČKE NEURONSKE MREŽE

-nemaju povratne veze (nema povratka u više slojeve)
-kreiranje neuronske mreže se vrši naredbom:

```
net=newff(PR, [S1, S2, ..., SN], {'TF1','TF2',..'TFN'},'BFT');
```

gdje je:
PR – min i max vrijednost funkcije
S – velicina sloja
TF – aktivacijska funkcija sloja
BTF – algoritam učenja, metoda učenja (treniranje)

Naredbe za treniranje nuronske mreže:

npr. neke vrijednosti za parametre
1) net.performFcn='sse'; (funkcija performasni sse)
2) net.trainParam.goal=30; (ukupna kvadratna pogrška izlaza)
3) net.trainParam.show=10; (frekvencija prikaza rezultata u kvadratnom prozoru)
4) net.trainParam.epochs=1000; (maximalni broj epoha, iteracija-vrijeme treniranja neuronske mreže)
5) net.trainParam.me=0.95; (momentna konstanta)
6) net.trainParam.lr=0.05; (brzina učenja);

Naredba za treniranje mreže:

```
net1=train(net,x,y);
```

gdje je:
net- kreirana mreža
x – ulaz
y – izlaz

Nakon toga treba prikazati istreniranu mrežu metodom simulacije:

```
ytest=sim(net1,x);
```

gdje je:
net1 – istrenirana mreža
x - ulaz

PRIMJER 1: Kreirati funkciju humps, $y=(1/((x-3)^2+0.01) + 1/((x-9)^2+0.04)) – 6$

```
x=0:0.05:2;
y=humps(x);
P=x;
T=y;
plot(P,T,'x');
grid ON

%kreiranje neuronske mreže
net=newff([0 2],[5,1],{'tansig','purelin'},'traingd');

%definiranje parametara mreže
net.trainParam.show=50;
net.trainParam.lr=0.05;
net.trainParam.epochs=1000;
net.trainParam.goal=1e-3; %greška tolerancije

%treniranje neuronske mreže
net1=train(net,P,T);

%pohrana vrijednosti - simulacija
a=sim(net1,P);

%iscrtavanje rezultata i poredenje
plot(P,a,'r',P,T,'g');
legend('neuronska mreza','humps funkcija');
```

PRIMJER 2: Aproksimirati signal sa šumom, ako je zadani signal: $e^{-0.1x2}\sin(5\sin(3pi))$;

```
x=0:0.1:5;
T0=exp(-0.1*P.^2).*sin(5*sin(3*pi)); % signal
T=T0+0.05*randn(size(P)); % signal sa šumom

net=newff(minmax(P),[30 1],{'tansig','purelin'},'traincgf');

%definiranje parametara mreže
net.trainParam.show=100;
net.trainParam.epochs=2000;
net.trainParam.goal=2e-4;

%treniranje neuronske mreže
[net,tr]=train(net,P,T);

figure(1)
plot(P,T0.P,T,':r','linewidth',2);
legend('signal','signal sa šumom');

figure(2)
```

```
plot(P,T,':r',P,a,'b','linewidth',2);
legend('signal sa šumom','neuronska mreza');
figure(3)
plot(P,T0,P,a,':g','linewidth',2);
legend('signal','neurosnka mreza');
```

PEIMJER 3:Aproksimirati signal sa šumom, ako je zadani signal: $f(x)=e^{-2\sin^2}(x)\sin(5x)$;

```
x=-4:0.01:4;
T0=exp(-2*sin(x).^2).*sin(5x); % signal
T=T0+0.05*randn(size(P)); % signal sa šumom

net=newff([-4 4],[30 1],{'tansig','purelin'},'traincgf');

%definiranje parametara mreže
net.trainParam.show=100;
net.trainParam.epochs=2000;
net.trainParam.goal=2e-4;

%treniranje neuronske mreže
[net,tr]=train(net,P,T);

figure(1)
plot(P,T0.P,T,':r','linewidth',2);
legend('signal','signal sa šumom');

figure(2)
plot(P,T,':r',P,a,'b','linewidth',2);
legend('signal sa šumom','neuronska mreza');
figure(3)
plot(P,T0,P,a,':g','linewidth',2);
legend('signal','neurosnka mreza');
```

PRIMJER 4: Za zadanu funkciju x=5sin(2pi i), istrenirati neuronsku mrežu ako je i=0:10

```
i=0:0.1:10;
x=5*sin(2*pi*i);
plot(i,x);
figure
pause

P=[i];
T=[x;
net=newff([0 10],[20 1],{'tansig','purelin','trainlm');
izlaz=sim(net,i);
plot(i,izlaz,'*r');
hold ON

plot(i,x);
hold ON
```

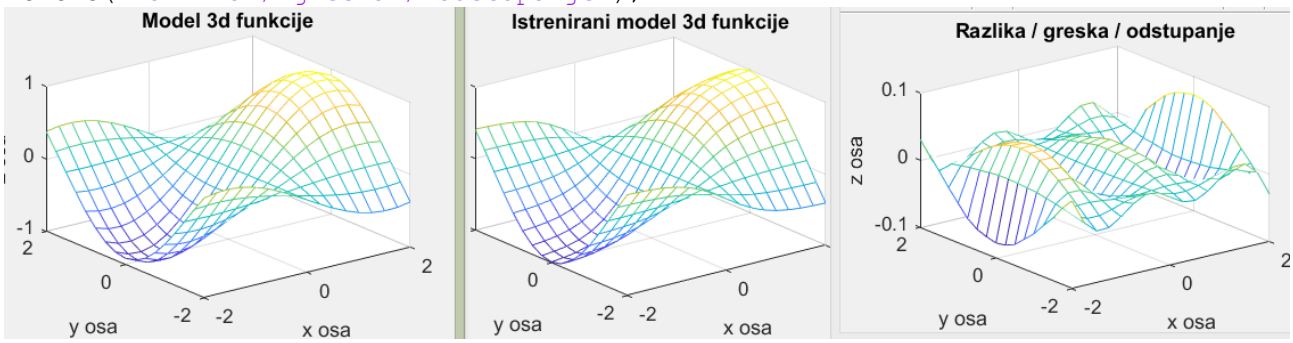
```
pause

fprintf('Pocetak treniranja \n');
net.trainParam.epochs=3000;
net.trainParam.show=10;
net.trainParam.time=Inf;
net.PerformFcn='sse';

tic
net=train(net,P,T);
toc

figure
izlaz=sim(net,i);
plot(i,izlaz,'g*');
hold ON

plot(i,x);

gensim(net,0.1);
```

PRIMJER 5: Predstaviti model 3D funkcije Z=cosx*siny definiranu na opsegu x=[-2,2], y=[-2,2]

a)Iscrtati funkciju
b)Kreirati neuronsku mrežu za funkciju pod a) ako je prvi sloj mreže na vrijednosti 25, drugi 17. Koristiti funkcije tansig i purelin. Algoritam treniranja je trainlm. Rezultat prikazati svakih 50 iteracija ako je brzina učenja 0.05, a tolerancija greške 3000.

```
x=-2:0.25:2;
y=-2:0.25:2;

z=cos(x)'*sin(y);
figure(1)
mesh(x,y,z);
xlabel('x osa');
ylabel('y osa');
zlabel('z osa');
title('Model 3d funkcije');
pause

P=[x;y];
T=z;
net=newff([-2 2;-2 2],[2 17],{'tansig','purelin'},'trainlm');
net.trainParam.show=50;
net.trainParam.lr=0.05;
net.trainParam.epochs=3000;
net.trainParam.goal=1e-3;

net1=train(net,P,T);
a=sim(net1,P);

figure(2)
mesh(x,y,a);
xlabel('x osa');
ylabel('y osa');
```

```
zlabel('z osa');
title('Istrenirani model 3d funkcije');
pause

figure(3)
mesh(x,y,a-z);
xlabel('x osa');
ylabel('y osa');
zlabel('z osa');
title('Razlika / greska / odstupanje');
```




```
pause
```

PRIMJER 6: Kreirati običnu neuronsku mrežu koja će oponašati funkciju

$10s^2 + 106s + 60 / 4s^3 + 14s^2 + 106s + 60$

```
P=[y (:,1)];
T=[y (:,2)];

minulaz=min(P);
maxulau=max(P);
minizlaz=min(T);
maxizlaz=max(T);

p=2*(P-minulaz)./(maxulaz-minulaz)-1;
t=2*(T-minizlaz)./(maxizlaz-minizlaz)-1;

net=newff([-1 1], [60 1], {'tansig','purelin'},'trainlm');

net.trainParam.epochs=1000;
net.trainParam.show=100;
net.trainParam.time=2e-9;
net.performFcn='sse';

tic
net=train(net,p',t');
toc

izalz=sim(net,p);
izlaz=(izlaz+1)*(maxizlaz-minizlaz).*2+maxizlaz;

figure
plot(p',izlaz,'r');
```

```
title('Podaci dobiveni sa treniranom neuronskom mrežom');
xlabel('x osa');
ylabel('y osa');

gensim(net,0.1);
```

PRIMJER 7: Kreirati neuronsku mrežu koja će oponašati inverznu dimamiku sistema:

$25s^2 + 50s + 50 / 15s^3 + s^2 + 50s + 50$

```
N=4;
P=y(:,1):
T=y(:,2);

minulaz=min(P);
maxulaz=max(P);
minizlaz=min(T);
maxizlaz=max(T);

net=newff([zeros(2*N,1)-1   zeros(2*N,1)+1], [15 5 1], {'tansig','tansig','purelin'},'trainlm');

net.trainParam.epochs=2000;
net.trainParam.goal=2e-9;
net.trainParam.show=10;
net.trainParam.time=Inf;
net.performFcn='sse';

P=2*(P-minulaz)./(maxulaz-minulaz)-1;
T=2*(T-minizlaz)./(maxizalz-minizlaz)-1;

vel=length(T);
ulaz=zeros(2*N, vel-N);
izlaz=zeros(1, vel-N);

for k=N:vel-1
        k=flipud(T(k-N+1:k+1));
        p=flipud(P(k-N+1:k-1));
        ulaz(:,k)=[t;p];
        izlaz(k)=p(k);
end

tic
net=train(net,ulaz,izlaz);
toc

izlaz=sim(net,ulaz);
izlaz=(izlaz+1)*(maxulaz-minulaz)./2+minulaz;

gensim(net);
```

PRIMJER 8: Kreirati inverznu neuronsku mrežu za zadanu funkciju: $1 / s^2 + s + 2$ ili $1/ (s+1)^2 +1$

```
N=4;
brojepoha=500;

minulaz= -3;
maxulaz= 3;
minizlaz= -2;
maxizlaz= 2;

net=newff([zeros(2*N,1)-1   zeros(2*N,1)+1], [15 5 1], {'tansig','tansig','purelin'},'trainlm');

net.trainParam.epochs=brojepoha;
net.trainParam.goal=0;
net.trainParam.show=10;
net.trainParam.time=Inf;
net.performFcn='sse';

fprintf('Opseg ulaza je: (% g, % g) \n' , minizlaz, maxizlaz);
fprintf('Pocetak treniranja');

P=t(:,1);
T=t(:,2);

P=2*(P-minulaz)./(maxulaz-minulaz)-1;
T=2*(T-minizlaz)./(maxizalz-minizlaz)-1;

vel=length(T);

ulaz=zeros(2*N, vel-N);
izlaz=zeros(1, vel-N);

for k=N:vel-1
        k=flipud(T(k-N+1:k+1));
        p=flipud(P(k-N+1:k-1));  % okreće niz, zadnji dio stavlja na prvi
        ulaz(:,k)=[t;p];
        izlaz(k)=p(k);
end

fprintf('Pocetak treniranja');

tic
net=train(net,ulaz,izlaz);
toc

izlaz=sim(net,ulaz);
izlaz=(izlaz+1)*(maxulaz-minulaz)./2+minulaz;
```

```
figure
plot(izlaz,'r');
title('Podaci dobiveni sa treniranom mrežom');

gensim(net, 0.1);
```

PRIMJER 8: Kreirati Elmanovu neuronsku mrežu za zadanu funkciju na intervalu 0-5, brojem iteracija 1500 i greškom 10^{-9}. Algoritam treniranja je traingdx.

$y=e^{-0.2x2}\sin(5*\sin 3x)$

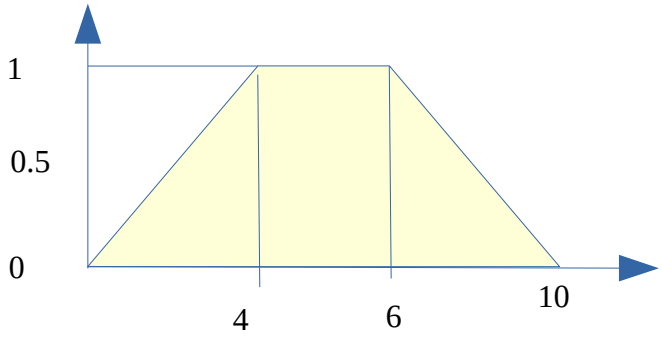
```
p=0:0.01:5;
t=exp(-0.1*p.^2).*sin(5*sin(3*p);

P=p;
T=t;

Pseq=con2seq(P); % pretvaranje niza u polje, uzalz
Tseq=con2seq(T);

net=newelm(minmax(P), [30 10 1], {'tansig','tansig','purelin'},'traingdx');

net.trainParam.epochs=1500;   % 8000
net.trainParam.show=50;
net.trainParam.goal=0.000000001;  % 0.00001

tic
[net,tr]=train(net,Pseq,Tseq);
toc

a=sim(net,Pseq);
b=cat(2,a{:});  % prikaz polja

time=1:length(p);
plot(time,t,'b- -',time,b,'r- -');
title('Rezultat treniranje mreže);
xlabel('vrijeme');
ylabel('izlazne vrijednosti');
legend('signal','elmanova mreza');
```

PRIMJER 9: Kreirati Elmanovu neuronsku mrežu za zadanu funkciju, za x na intervalu 0-5, broj iteracija 3000, greška 10^{-4}, algoritam treniranja traingdx. y=5*sin((2 pi x)/5)

```
p=0:0.01:5;
t=5*(sin(2*pi*p)./5);
P=p;
T=t;
Pseq=con2seq(P); % pretvaranje niza u polje, uzalz
```

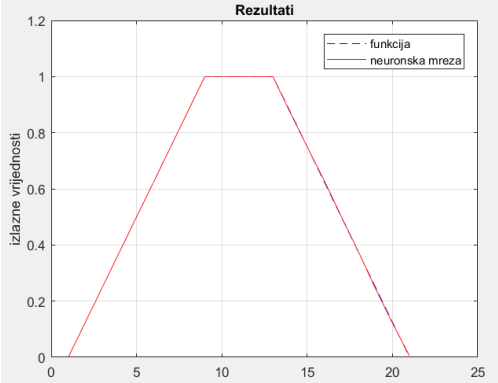
```
Tseq=con2seq(T);

net=newelm(minmax(P), [30 10 1], {'tansig','tansig','purelin'},'traingdx');

net.trainParam.epochs=3000;
net.trainParam.show=50;
net.trainParam.goal=0.0001;

tic
[net,tr]=train(net,Pseq,Tseq);
toc

a=sim(net,Pseq);
b=cat(2,a{:});  % prikaz polja

time=1:length(p);
plot(time,t,'b- -',time,b,'r- -');
title('Rezultat treniranje mreže);
xlabel('vrijeme');
ylabel('izlazne vrijednosti');
legend('signal','elmanova mreza');
```

PRIMJER 10: Kreirati Elmanovu neuronsku mrežu na intervalu 0-10, broj iteracija 3000, greška 10^{-4} i algoritam učenja je traingdx.




```
p=0:0.5:10;
t=(1/4).*(p.*(p<=4)+4*((p>4) & (p<=6)) + (10-p).*((p>6) & (p<=10)));
P=p;
T=t;
Pseq=con2seq(P);
Tseq=con2seq(T);
net=newelm(minmax(P),[40 1],{'tansig','purelin'}, 'traingdx');
net.trainParam.epochs=2500;
net.trainParam.show=50;
net.trainParam.goal=1e-7;
tic
[net,tr]=train(net,Pseq,Tseq);
toc
a=sim(net,Pseq);
b=cat(2,a{:});
time=1:length(p);
plot(time,t,'b--',time,b,'r')
```

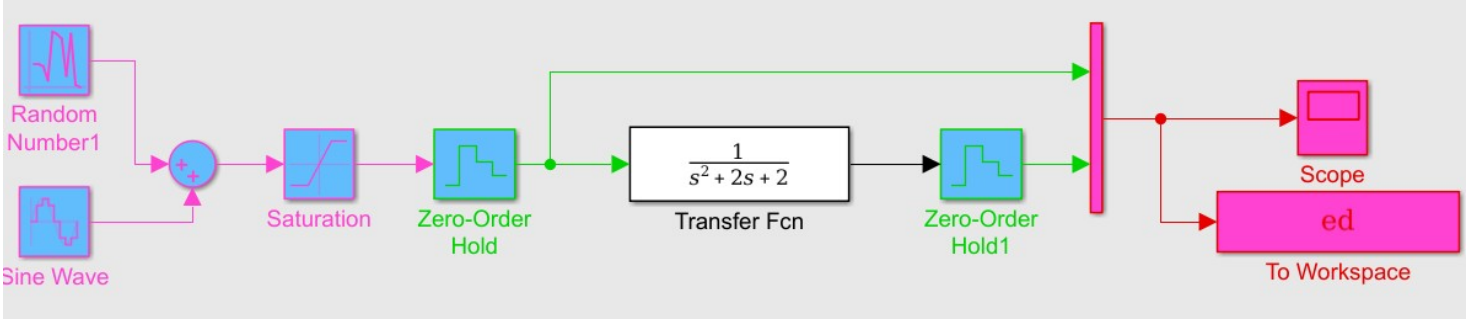
```
grid on
title('Rezultati');
xlabel('vrijeme');
ylabel('izlazne vrijednosti')
legend('funkcija','neuronska mreza');
```

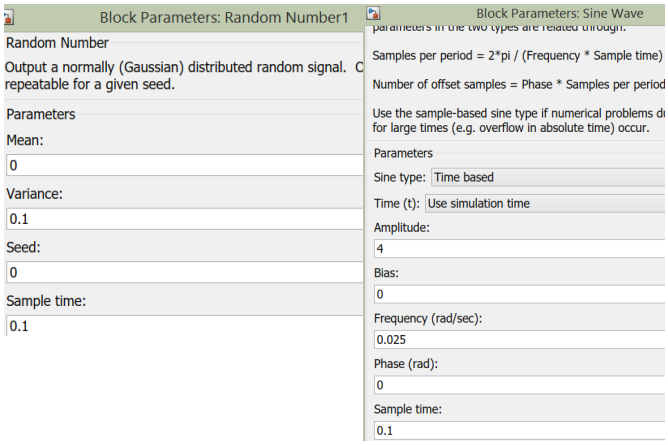


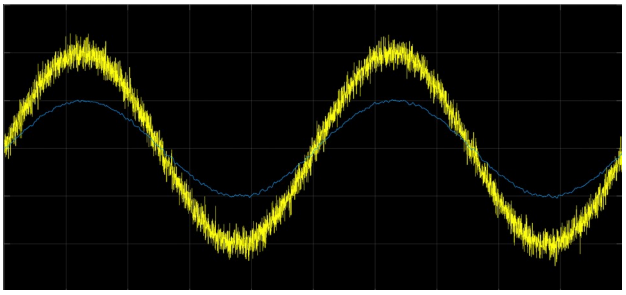
INVERZNA DINAMIKA

a) ulazni signal

vrijeme simulacije t=500



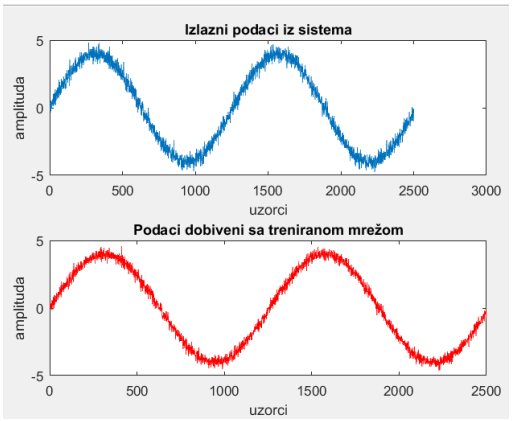


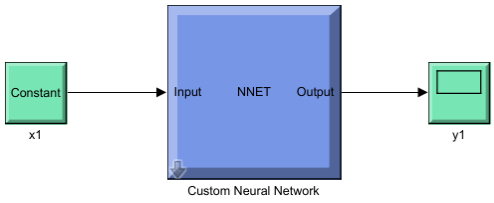


b) Matlab kod

```
N=4;
% Priprema podataka za treniranje mreze
fprintf('Priprema podataka za treniranje mreze...\n');
P= ed(:, 1);
T = ed(:, 2);
minulaz = min(P);
maxulaz = max(P); %opseg ulaznih vrijednosti
minizlaz = min(T);
maxizlaz = max(T);%opseg moguceg izlaza

net = newff([zeros(2*N,1)-1 zeros(2*N,1)+1], ...
[15 5 1],{'tansig','tansig', 'purelin'}, 'trainlm');

net.trainParam.epochs = 2000;
net.trainParam.goal = 2e-9;
net.trainParam.show = 10;
net.trainParam.time = Inf;

fprintf('Opseg ulaza mreze je: [%g, %g].\n', minizlaz, maxizlaz);

% normiranje ulaza i izlaza na opseg [-1, 1]
P= 2 * (P - minulaz) ./ (maxulaz - minulaz) - 1;
T = 2 * (T - minizlaz) ./ (maxizlaz - minizlaz) - 1;
```

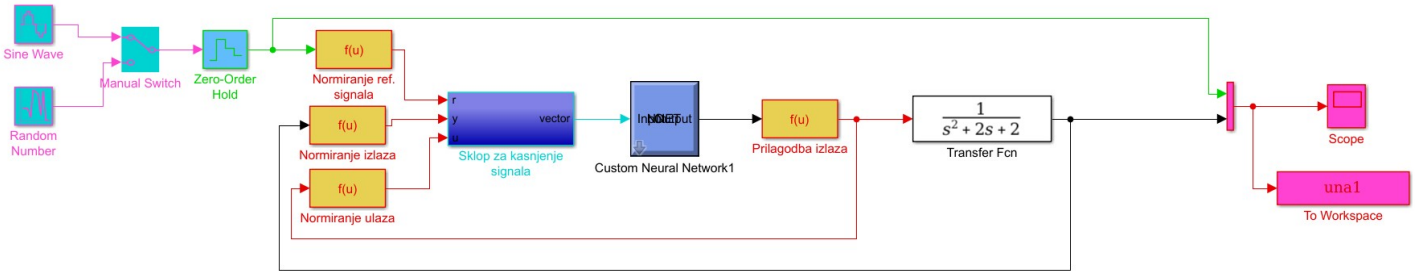
```
vel = length(T);
ulaz = zeros(2*N, vel-N);
izlaz = zeros(1, vel-N);

for k = N : vel-1
    t = flipud( T(k-N+1:k+1) );
    p = flipud( P(k-N+1:k-1) );
    ulaz(:,k) = [t; p];
    izlaz(k) = P(k);
end;
% Treniranje...
fprintf('Po?etak treniranja\n');
tic
net = train(net, ulaz, izlaz);
toc

izlaz=sim(net,ulaz);
izlaz=(izlaz+1)*(maxulaz-minulaz)./2 +minulaz;

figure
subplot(2,1,1),plot(ed(:,1));
title('Izlazni podaci iz sistema');
xlabel('uzorci')
ylabel('amplituda')
subplot(2,1,2),plot(izlaz,'r')
title('Podaci dobiveni sa treniranom mrežom');
xlabel('uzorci')
ylabel('amplituda')
gensim(net,0.1)
```

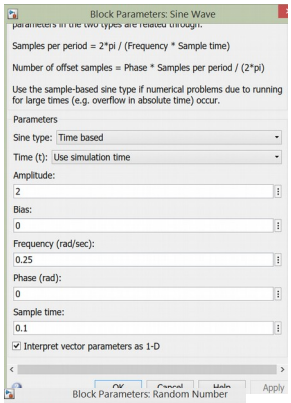


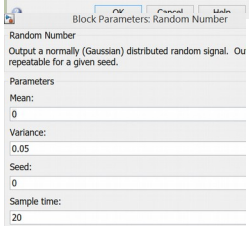


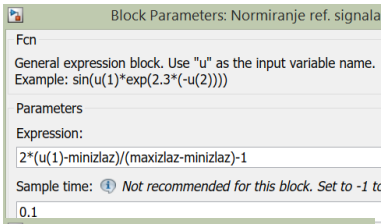
c) završni model

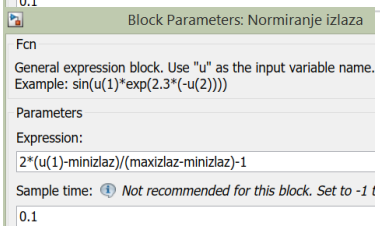
t=500

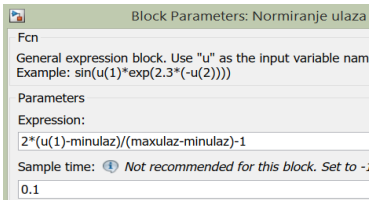


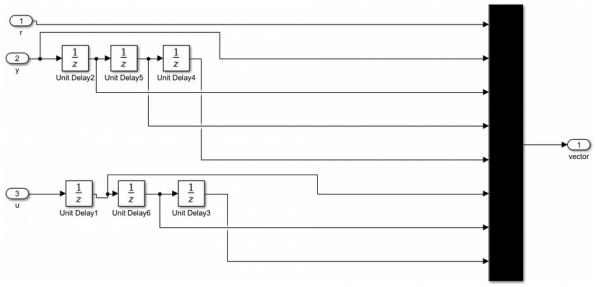


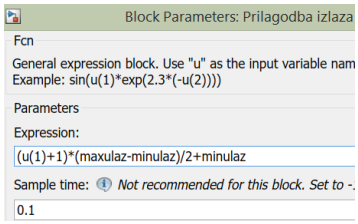


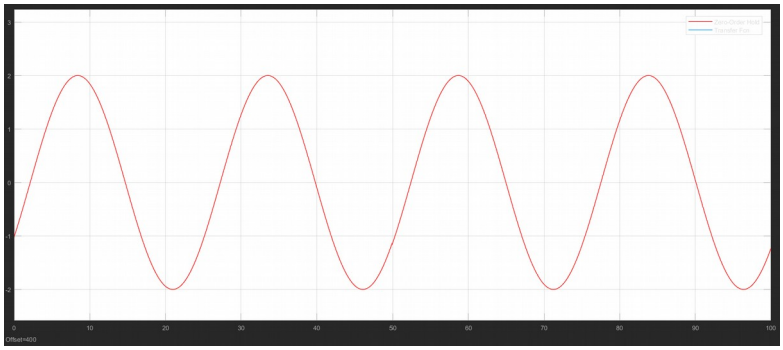








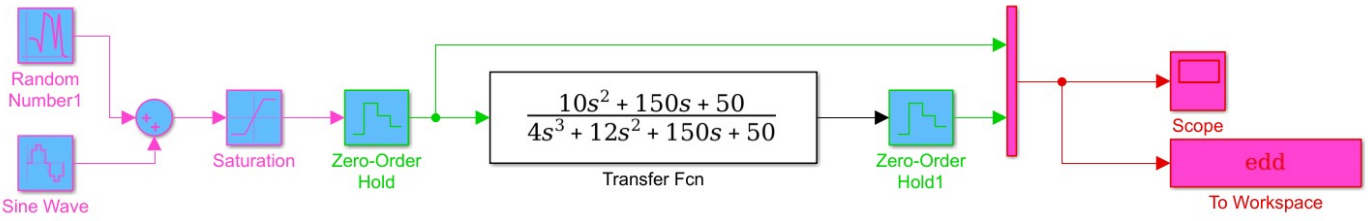


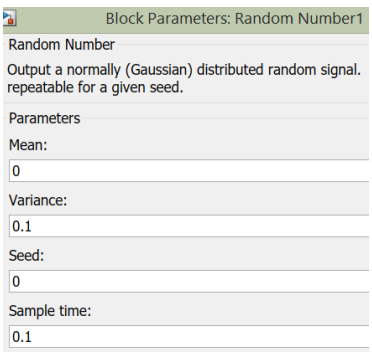


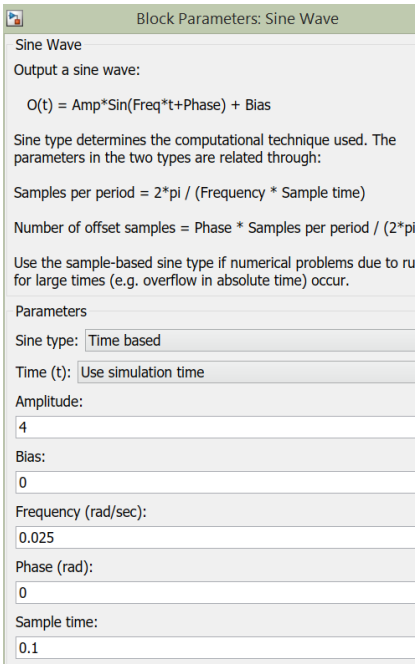
DINAMIČKA NEURONSKA MREŽA - NARKS

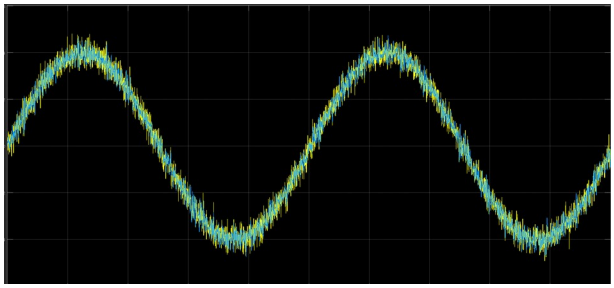
a) ulazni signal

vrijeme simulacije t=500







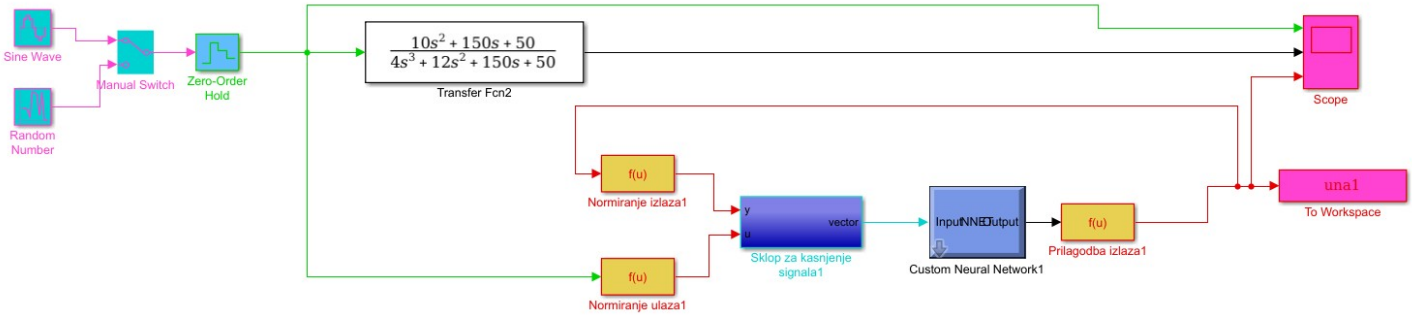


b) Matlab kod

```
P= edd(:, 1);
T = edd(:, 2);
vel = length(P);
minulaz = min(P);
maxulaz = max(P); %opseg ulaznih vrijednosti
minizlaz = min(T);
maxizlaz = max(T);%opseg moguceg izlaza

% normiranje ulaza i izlaza na opseg [-1, 1]
p= 2 * (P - minulaz) ./ (maxulaz - minulaz) - 1;
t = 2 * (T - minizlaz) ./ (maxizlaz - minizlaz) - 1;
N=4;
p=p;
t=t;
for k = N+1 : vel
    t1 = flipud( t(k-N:k-1) );
    p1 = flipud( p(k-N:k-1) );
    ulaz(:,k) = [t1; p1];
    izlaz(k) = t(k)';
end

ulaz
```

```
izlaz

net = newff([-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1 1],[15 1],{'tansig',
'purelin'}, 'trainlm');

net.trainParam.epochs = 2000;
net.trainParam.goal = 2e-4;
net.trainParam.show = 300;
net.trainParam.time = Inf;
net.performFcn='sse';

% Treniranje...
fprintf('Po?etak treniranja\n');
tic
net = train(net, ulaz, izlaz);
toc

izlaz=sim(net,ulaz);
izlaz=(izlaz+1)*(maxizlaz-minizlaz)./2 +minizlaz;

figure
subplot(2,1,1),plot(edd(:,1));
title('Izlazni podaci iz sistema');
xlabel('uzorci')
ylabel('amplituda')
subplot(2,1,2),plot(izlaz,'r')
title('Podaci dobiveni sa treniranom mrežom');
xlabel('uzorci')
ylabel('amplituda')
gensim(net,0.1)
```

c) Zadnji model



2*(u(1)-minizlaz)/(maxizlaz-minizlaz)-1
2*(u(1)-minulaz)/(maxulaz-minulaz)-1
(u(1)+1)*(maxizlaz-minizlaz)/2+minizlaz