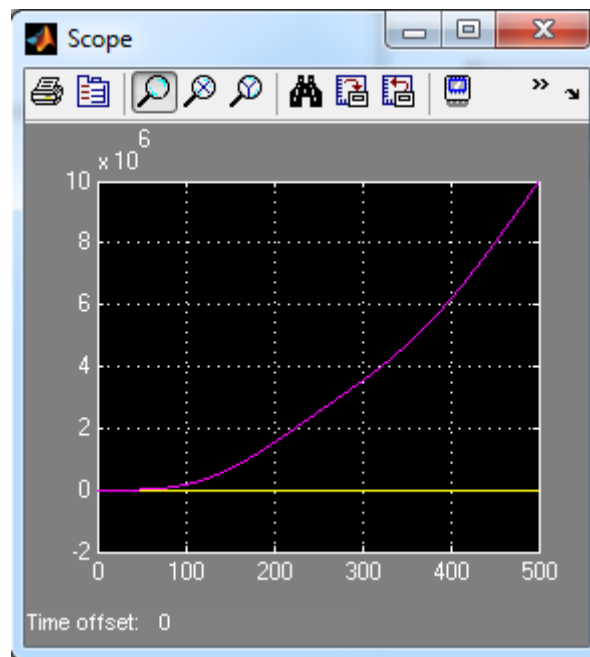
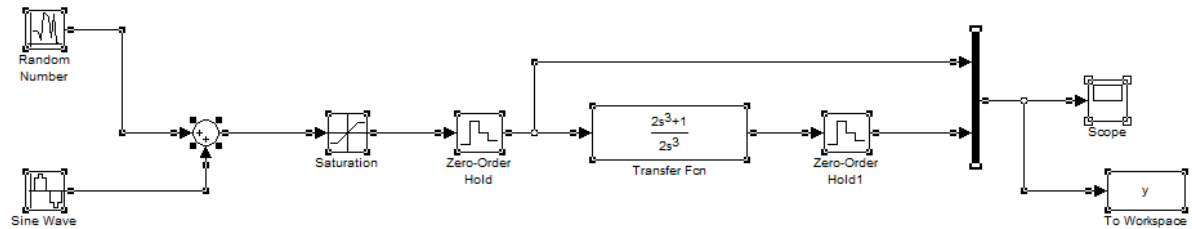


ZADATAK 1:

ZaZadatak1.mdl:



Zadatak1.m:

```
N=4;
fprintf('Priprema podataka za treniranje mreze...\n');
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;
fprintf('Opseg ulaza mreze je: [%g, %g].\n', minizlaz,maxizlaz);

P= 2 * (P - minulaz) ./ (maxulaz - minulaz) - 1;
T = 2 * (T - minizlaz) ./ (maxizlaz - minizlaz) - 1;
vel = length(T);
ulaz = zeros(2*N, vel-N);
```

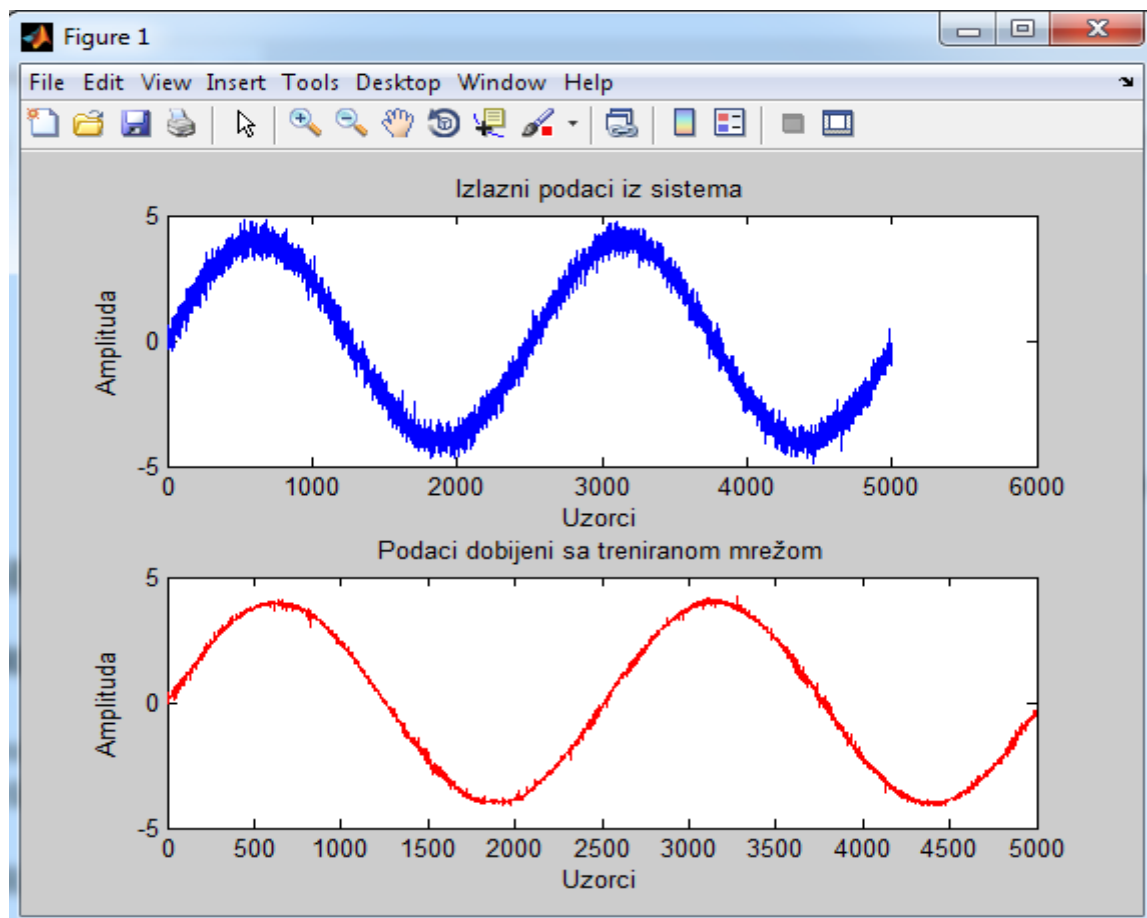
```

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;

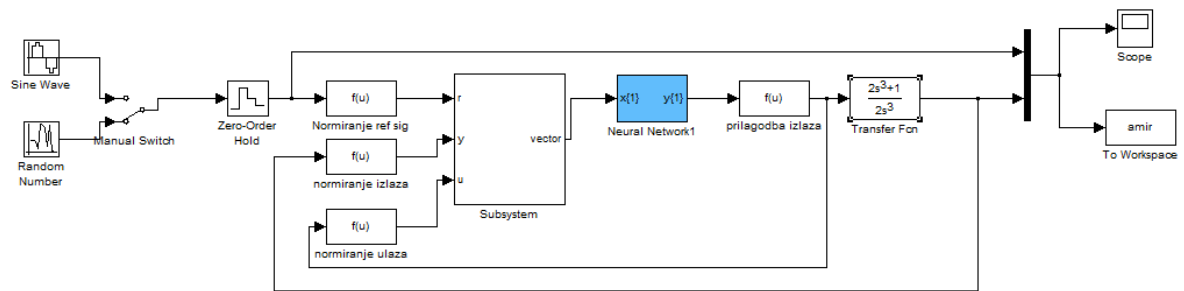
fprintf('Pocetak treniranja\n');
tic
net = train(net, ulaz, izlaz);
toc
izlaz=sim(net,ulaz);
izlaz=(izlaz+1)*(maxulaz-minulaz)./2 +minulaz;
figure
subplot(2,1,1);
plot(y(:,1));
title('Izlazni podaci iz sistema');
xlabel('Uzorci')
ylabel('Amplituda')
subplot(2,1,2);
plot(izlaz,'r');
title('Podaci dobijeni sa treniranom mrežom');
xlabel('Uzorci')
ylabel('Amplituda')
gensim(net,0.1)

```

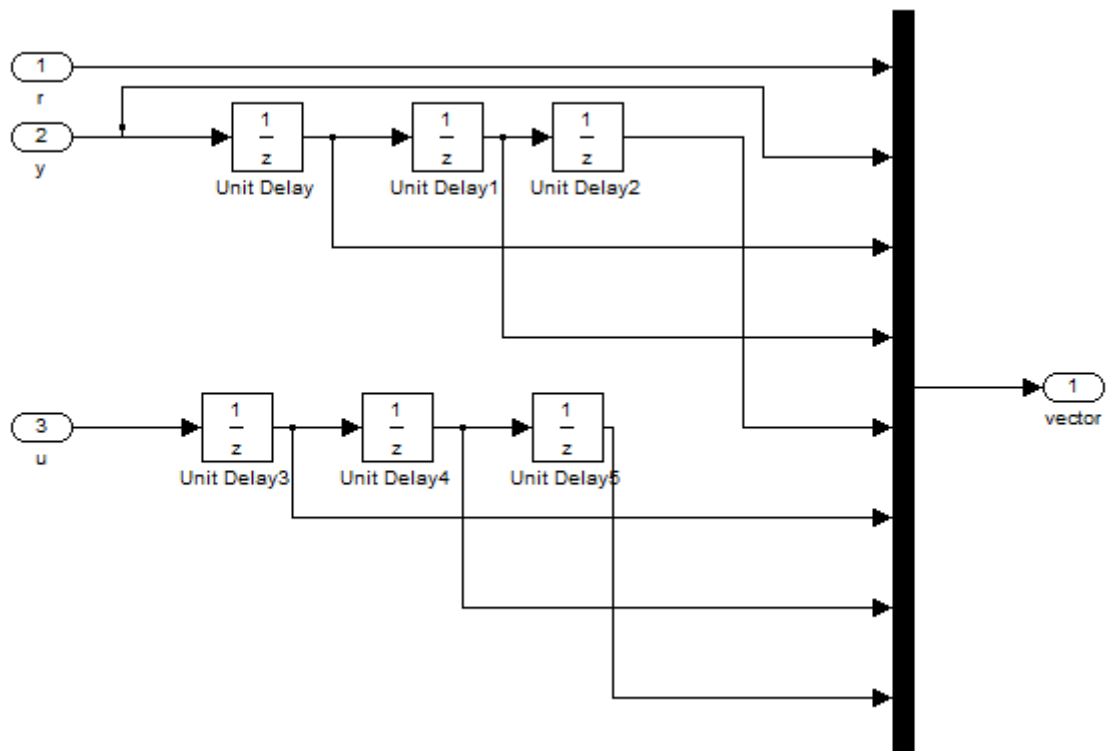
Rješenje:



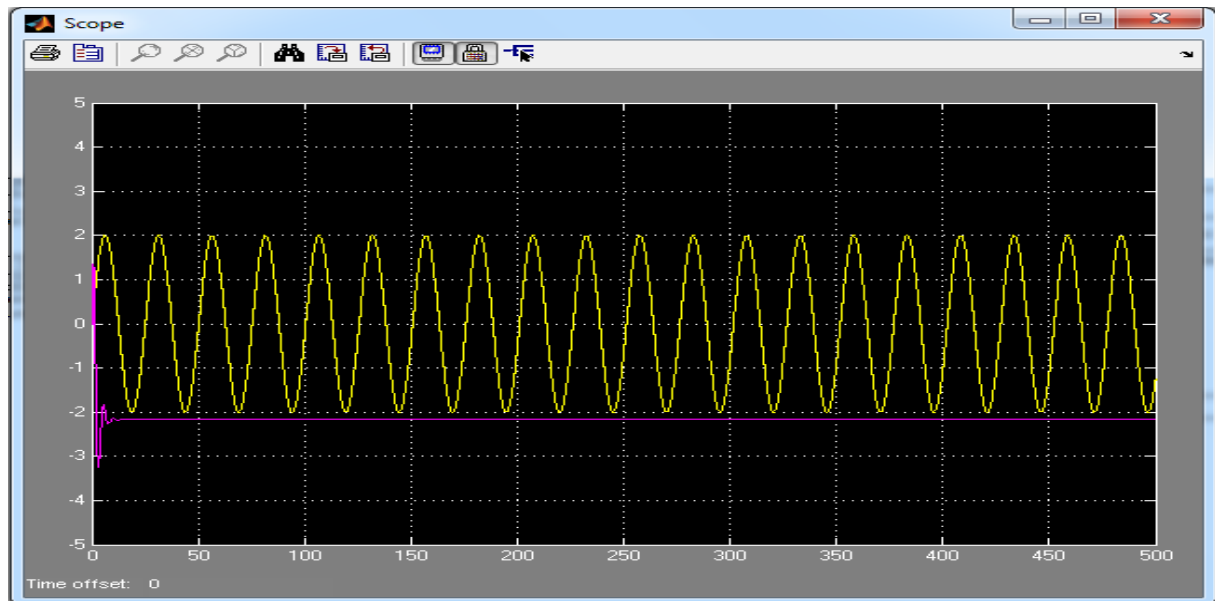
ZaZadatak_1.mdl:



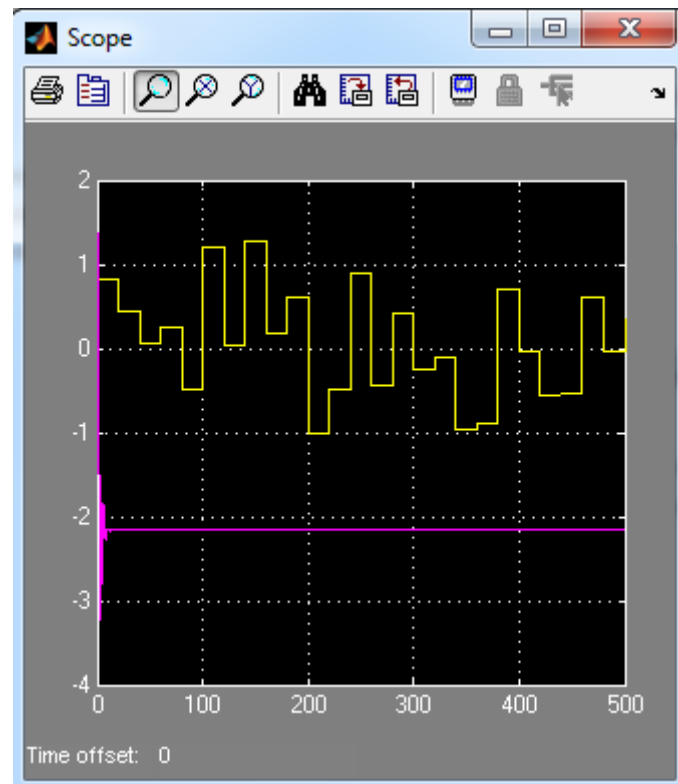
Unutar bloka „Subsystem“:



Kada je na Manual switch „Sine wave“:



Kada je na Manual switch „Random number“:



ZADATAK 2:

Zadatak2.m:

```
P=[0 4.3 7 10 11 13 13.5 15];
T=[0 1 1 1.5 0.4 0.4 0.7 0];
```

```
Pseq=con2seq(P);
```

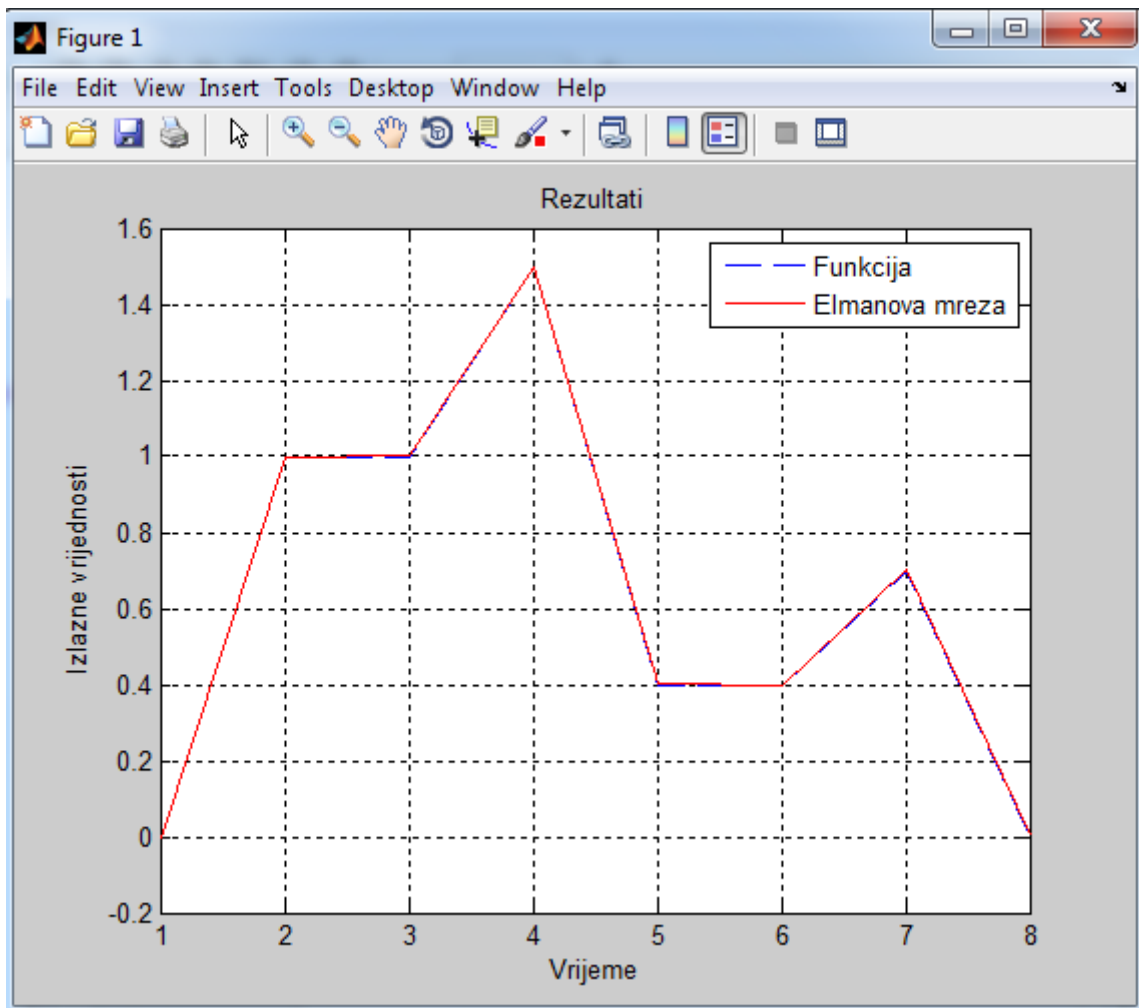
```

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', 'Elmanova mreza');

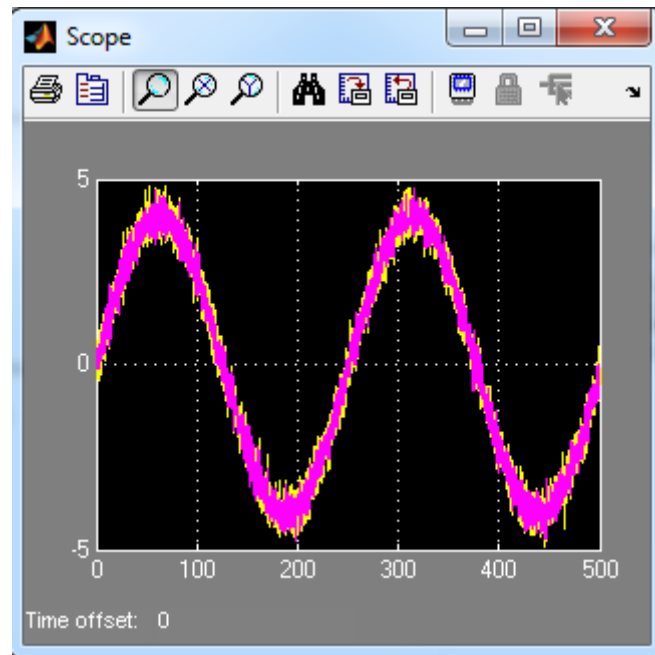
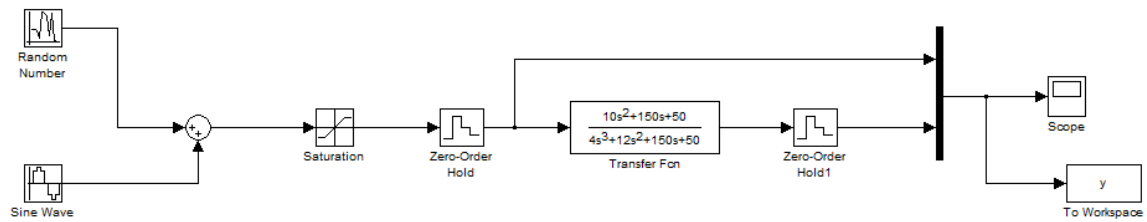
```

Rješenje:



ZADATAK 3:

ZaZadatak3.mdl:



Zadatak3.m:

```
P= y(:,1);
T = y(:,2);
vel = length(P);
minulaz = min(P);
maxulaz = max(P);
minizlaz = min(T);
maxizlaz = max(T);

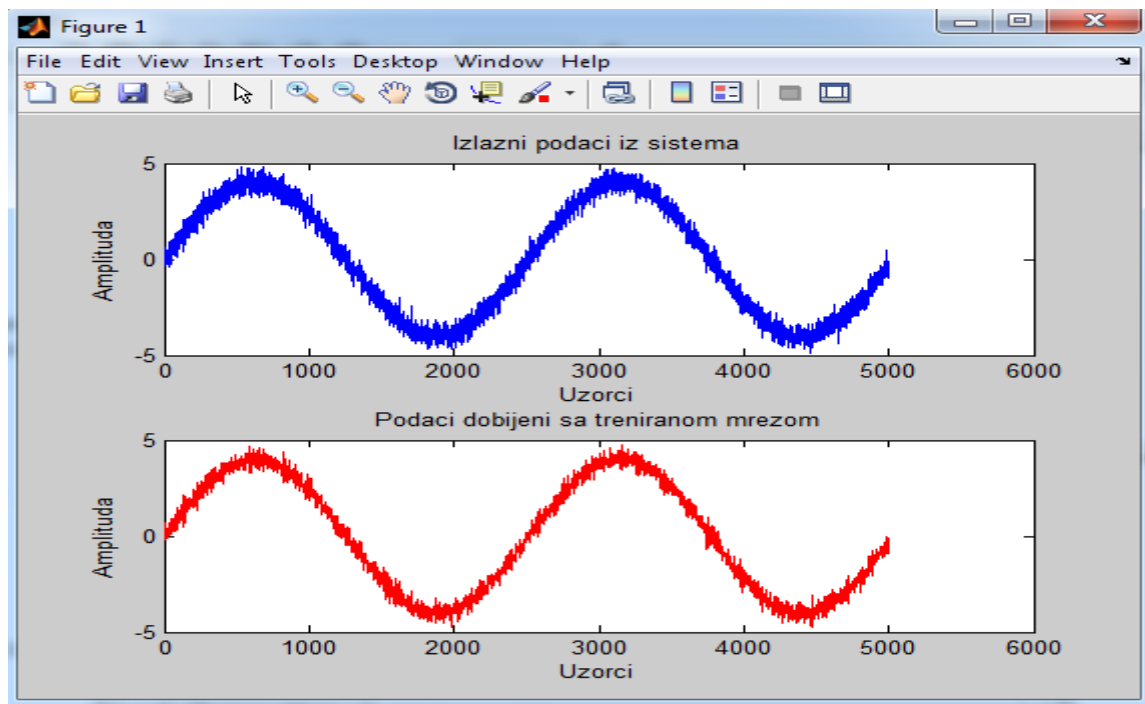
p= 2 * (P - minulaz) ./ (maxulaz - minulaz) - 1;
t = 2 * (T - minizlaz) ./ (maxizlaz - minizlaz) - 1;
N=4;
p=p;
t=t;
for k = N+1 : vel
    t1 = flipud( t(k-N:k-1) );
    p1 = flipud( p(k-N:k-1) );
    ulaz(:,k) = [t1; p1];
    izlaz(k) = t(k)';
end
ulaz
izlaz
```

```

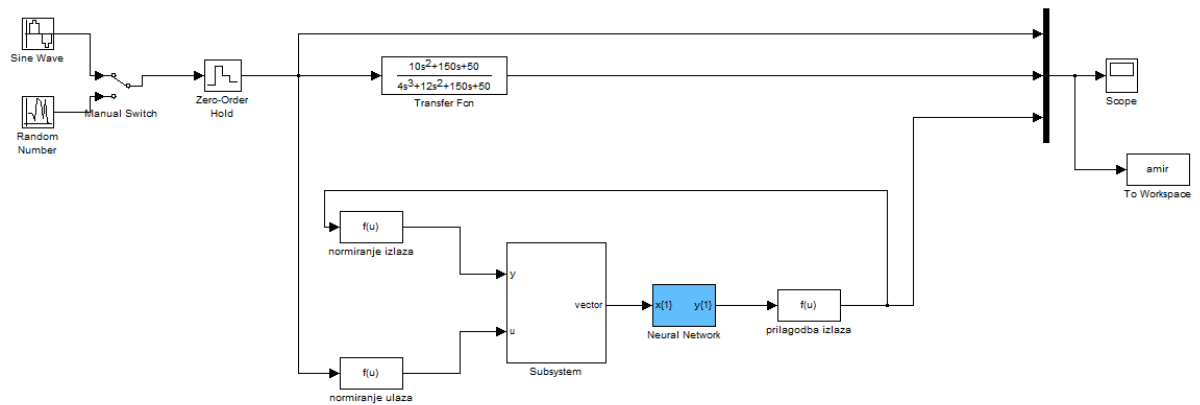
net = newff([-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1 1],[15
1],{'tansig','purelin'}, 'trainlm');
net.trainParam.epochs = 2000;
net.trainParam.goal = 2e-4;
net.trainParam.show = 300;
net.trainParam.time = Inf;
net.performFcn='sse';
fprintf('Pocetak treniranja\n');
tic
net = train(net, ulaz, izlaz);
toc
izlaz=sim(net,ulaz);
izlaz=(izlaz+1)*(maxizlaz-minizlaz)./2 +minizlaz;
figure
subplot(2,1,1);
plot(y(:,1));
title('Izlazni podaci iz sistema');
xlabel('Uzorci')
ylabel('Amplituda')
subplot(2,1,2);
plot(izlaz,'r');
title('Podaci dobijeni sa treniranom mrezom');
xlabel('Uzorci')
ylabel('Amplituda')
gensim(net,0.1)

```

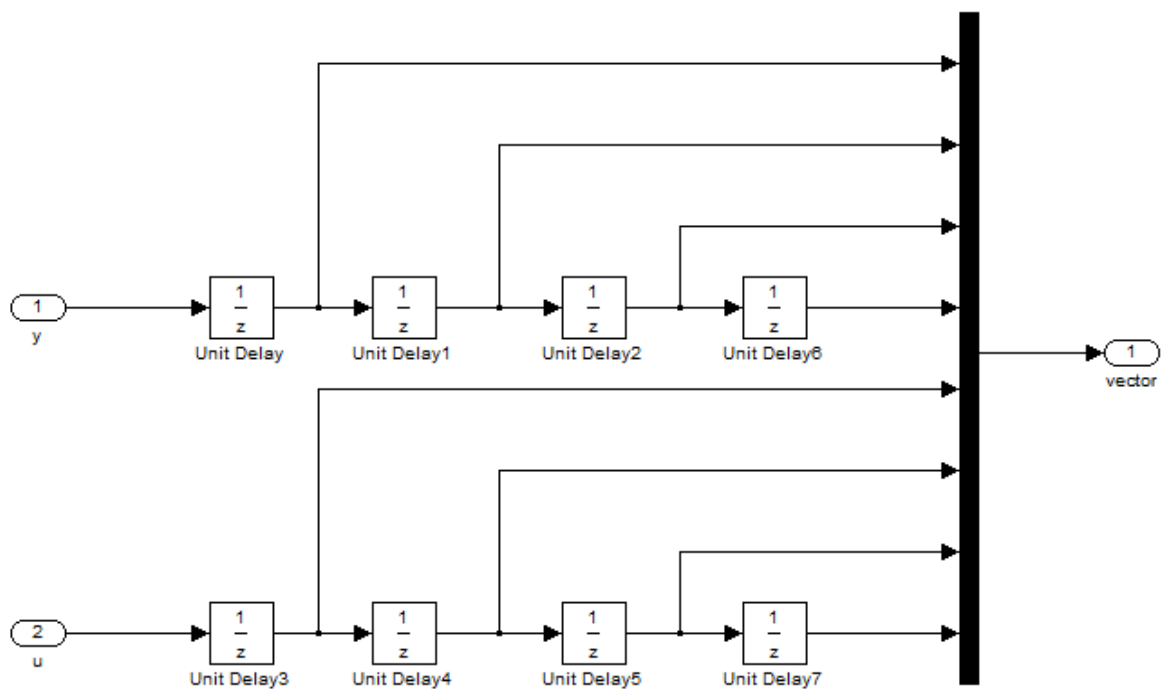
Rješenje:



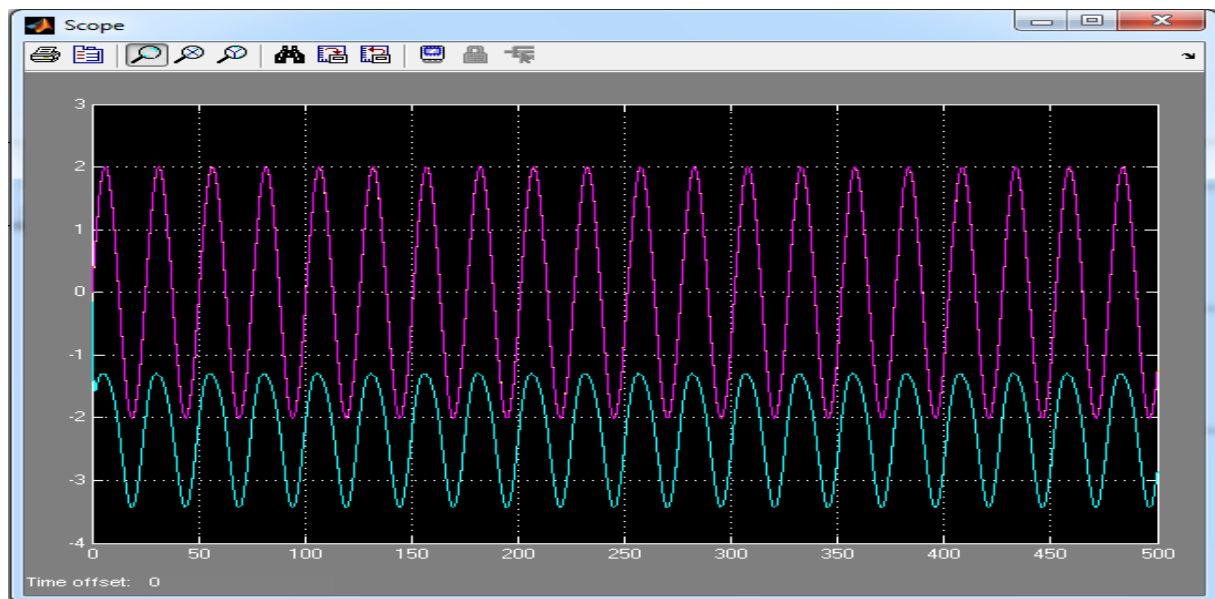
ZaZadatak_3.mdl:



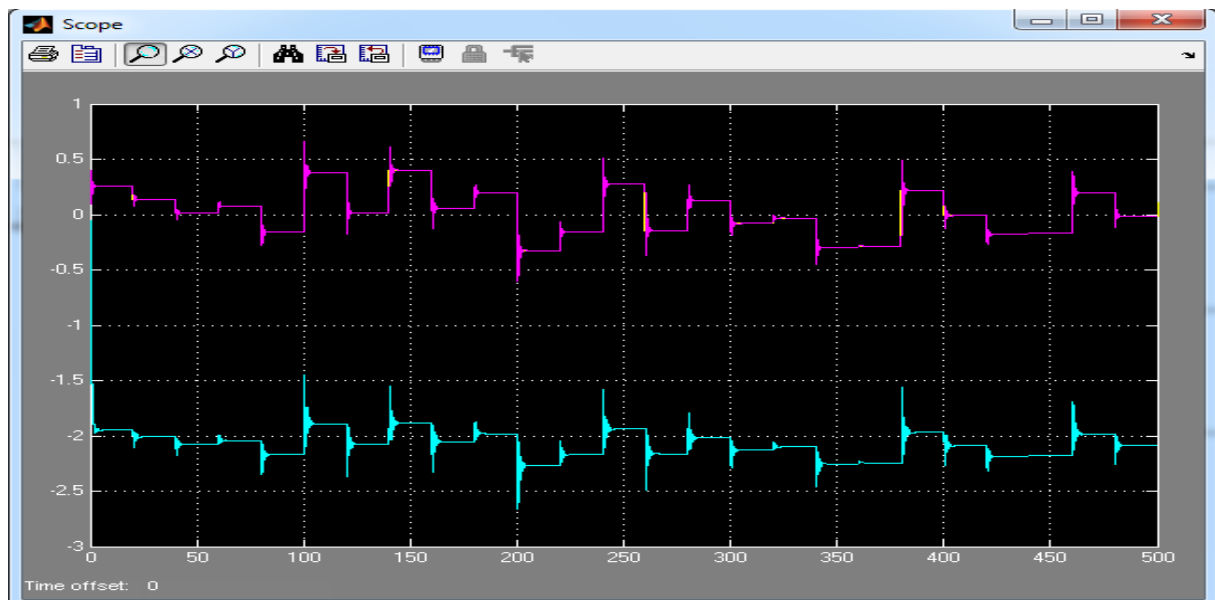
Unutar bloka „Subsystem“:



Kada je Manual Switch na „Sine wave“:



Kada je Manual switch na „Random number“:



ZADATAK 4:

Vještački neuron predstavlja jednostavni procesni element koji obavlja relativno jednostavnu matematičku funkciju. Vještački neuroni imaju jednostavnu strukturu i imaju slične funkcije kao i biološki neuroni. Tijelo neurona se naziva čvor ili jedinica. Arhitekturu vještačke neuronske mreže predstavlja specifično uređenje i povezivanje neurona u obliku mreže. Po arhitekturi, neuronske mreže se razlikuju prema broju neuronskih slojeva. Obično svaki sloj prima ulaze iz prethodnog sloja, a svoje izlaze šalje narednom sloju. Model vještačke neuronske mreže sastoji se od ulaznog, izlaznog i skrivenog sloja. Funkcije pojedinih slojeva odgovaraju njihovom nazivu. Ulazni slojevi služe da prime signale iz okoline i distribuiraju ih u unutrašnjost mreže. Skriveni slojevi te signale preuzimaju, procesiraju ih i prenose do izlaznih slojeva koji ih predaju okolini (izlaz).

Aktivacione funkcije neurona na skrivenim slojevima su potrebne da bi mreža bila u stanju da nauči nelinearne funkcije. Bez nelinearnosti, neuroni skrivenih slojeva ne bi imali veće mogućnosti od obične perceptronske mreže (koja se sastoji samo od ulaza i izlaza), jer se kombinovanjem linearnih funkcija ponovo dobija linearna funkcija. Preko težinskih koeficijenata se ostvaruju sinapse, kojima biološki neuroni regulišu pogodnost određene putanje između aksona i dendrita kod vještačkih neurona.

Osnovne komponente neuronskih mreža: neuron, dendriti, sinapsa (spoj neurona i dendrita), sinaptička težina i akson.