MATLAB Program for the Identification of discrete nonlinear systems

The system is given by:

$$y(k) = \frac{-0.9\ y(k-1) + u(k-1)}{1 + y(k-1)^2}$$

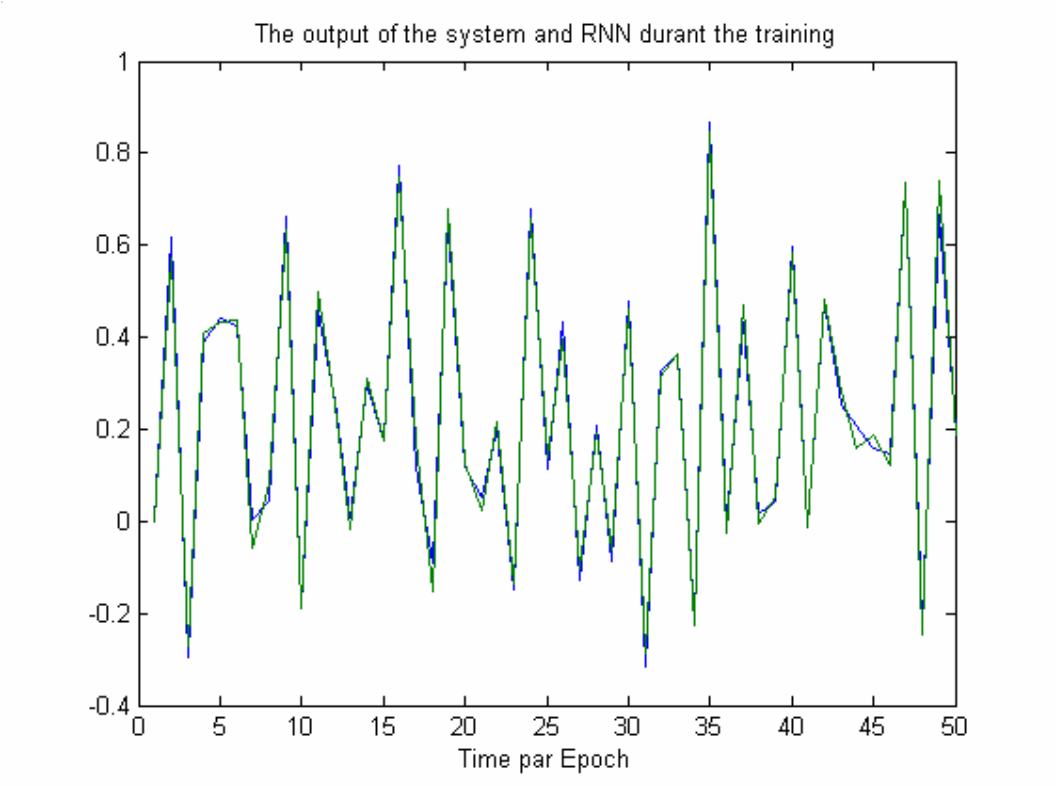
The Matlab Program:

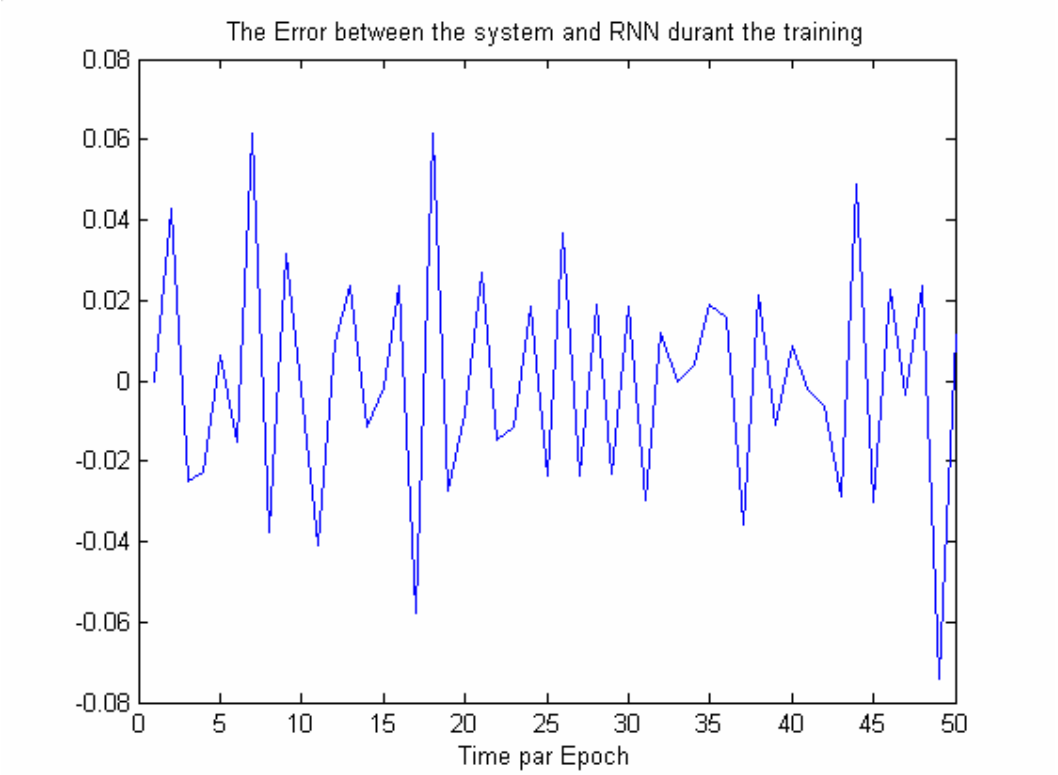
```
% Identification  of nonlinear system by FNN
% y(k) = (-0.9*y(k-1)+u(k-1))/(1+y(k-1)^2)
% R3,N1,N2,1
%
%Dr. R. HEDJAR (Dec 1, 2004)
% KSU-CCIS

clear all
N1=10;N2=10;              % N1 is the Number of Neurons in the first HDL  and N2 second HDL
nn=0.55;                  %Learning parameter

% Initilisation
W1=0.5*rand(N1,3);
W2=0.61*rand(N2+1,N1+1);
W3=0.3*rand(1,N2+1);


yp(1)=0;
rand('seed',9)
% Input
u=rand(50,1);

for l=1:500
jk=0;
% Beginning of the Epoch
for  k=2:50
     yo=[yp(k-1) u(k-1) 1]';                    %%% Forward Pass %%%%
     x1=W1*yo;
     for i=1:N1
        y1(i)=1/(1+exp(-x1(i)));                 %%% Output of the First HDL
     end

     K1=[y1';1];
     x2=W2*K1;
     for i=1:N2
        y2(i)=1/(1+exp(-x2(i)));                 %%%% Output of the second HDL
     end
     K2=[y2';1];
     x3=W3*K2;
     z(k)=x3;                                   %%%% The output of RNN
     y3(k)=1/(1+exp(-x3));
```

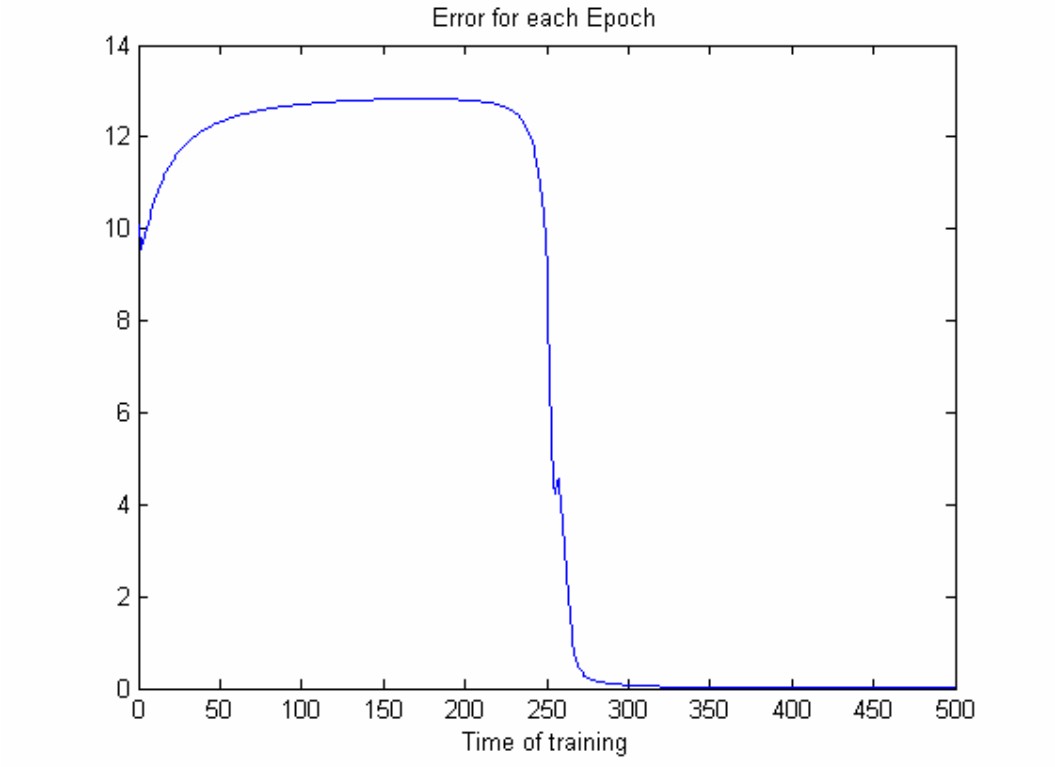
```
    yp(k)=(-0.9*yp(k-1)+u(k-1))/(1+yp(k-1)^2);                % The output
    e(k)=yp(k)-z(k);                                          %%%% Error
    jk=jk+e(k)^2;                                             %%%% Sum of Error per Epoch
                                  %%%%% Back ward Pass%%%%%%
    d3=y3(k)*(1-y3(k))*e(k);
    for j=1:N2+1,
       W3(j)=W3(j)+nn*d3*K2(j);
       d2(j)=K2(j)*(1-K2(j))*d3*W3(j);
    end

    for i=1:N2+1,
       for j=1:N1+1
          W2(i,j)=W2(i,j)+nn*d2(i)*K1(j);
       end
    end

    for i=1:N1,
       d1(i)=K1(i)*(1-K1(i))*d2*W2(:,i);
    end

    for i=1:N1,
       for j=1:3
          W1(i,j)=W1(i,j) + nn*d1(i)*yo(j);
       end
    end

 end                                  %%%%%%%%% End of the Epoch
je(l)=jk/2;                           %%%%  save the error
end                                  %%%% End of the training %%%%%%%

t=1:50;
plot(t,yp,t,z),xlabel('Time par Epoch'),title('The output of the system and RNN Durant the training')
pause
plot(t,yp-z),xlabel('Time par Epoch'),title('The Error between the system and RNN Durant the training')
pause
plot(je),xlabel('Time of training'),title('Error for each Epoch')
pause

yp(1)=0;

                  %%%%%%%%%% TEST %%%%%%%%%%%%
for  k=2:100
    u(k)=cos(2*pi*k/15);          %%  We use here another signal
    yo=[yp(k-1) u(k-1) 1]';
    x1=W1*yo;
    for i=1:N1,
       y1(i)=1/(1+exp(-x1(i)));
    end
    K1=[y1';1];
    x2=W2*K1;
```

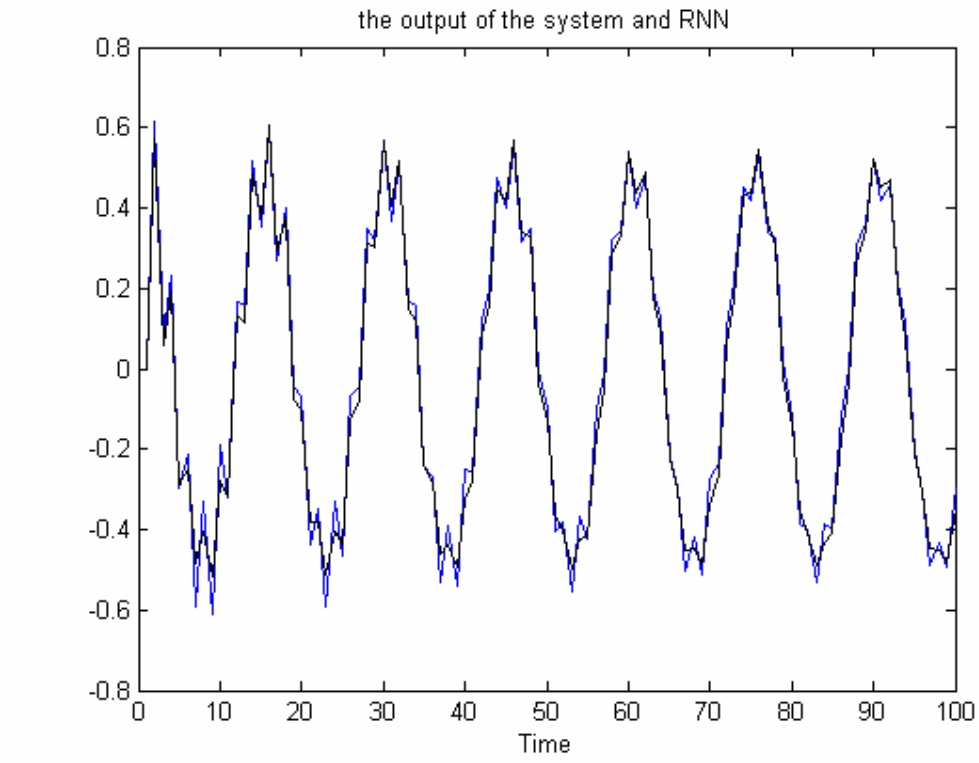
```
    for i=1:N2,
       y2(i)=1/(1+exp(-x2(i)));
    end
    K2=[y2';1];
    x3=W3*K2;
    z(k)=x3;
    y3(k)=1/(1+exp(-x3));
    yp(k)=(-0.9*yp(k-1)+u(k-1))/(1+yp(k-1)^2);
end
t=1:100;
plot(t,yp,'-',t,z,'k'),xlabel('Time'),title('the output of the system and RNN')
pause
plot(t,yp-z,'k'),xlabel('Time'),title('The Error between the output of the system and RNN')
```

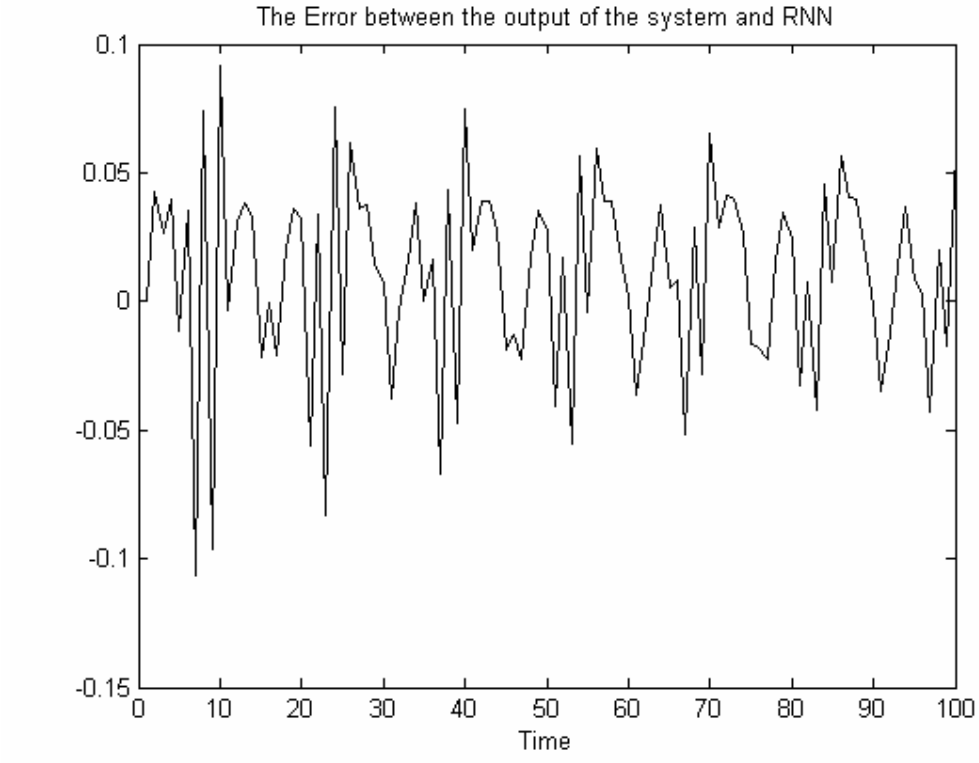
The figures are:




The Error between the system and RNN durant the training
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
0
5
10
15
20
25
30
35
40
45
50
Time par Epoch


Error for each Epoch
14
12
10
8
6
4
2
0
0
50
100
150
200
250
300
350
400
450
500
Time of training


the output of the system and RNN
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
10
20
30
40
50
60
70
80
90
100
Time


The Error between the output of the system and RNN
0.1
0.05
0
-0.05
-0.1
-0.15
0
10
20
30
40
50
60
70
80
90
100
Time