Discrete Time System Simulation

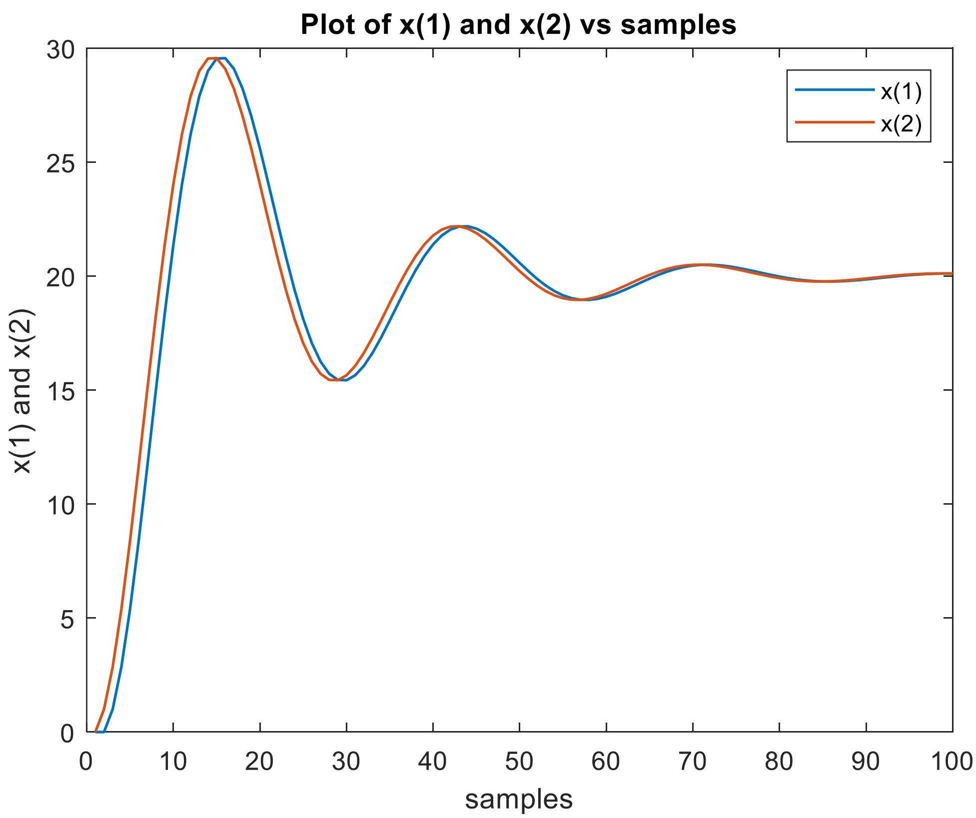
The System is defined by the following set of equations and simulated for 100 time samples. Then a Gaussian process noice is added with each component having a covariance of 0.1 i.e., Q=0.1*I*

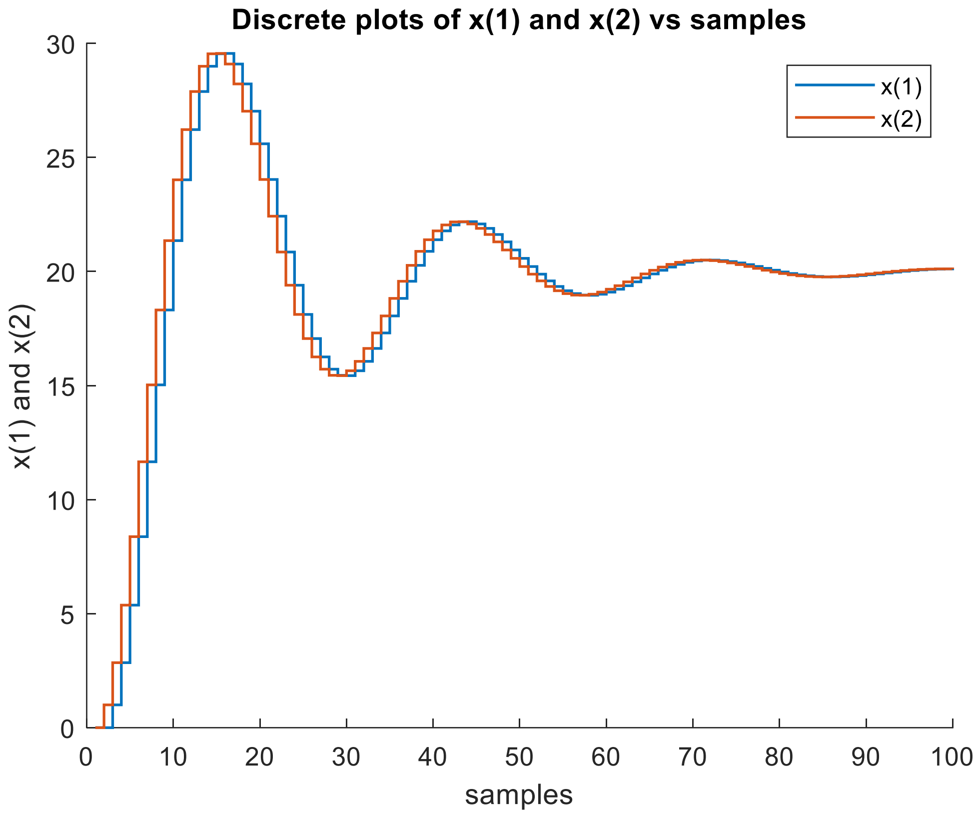
$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ -0.9 & 1.85 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k$$

a. MATLAB Code

```
function discretesystemsimulation
x(1,2)=0;
u=1;
for k=1:1:100
    x(k+1,1)=x(k,2);
    x(k+1,2)=-0.9*x(k,1)+1.85*x(k,2)+u;
end
t=1:1:100;
figure(1)
plot(t,x(1:100,1),t,x(1:100,2))
figure(2)
hold on
stairs(t,x(1:100,1))
stairs(t,x(1:100,2))
end
```

Plots

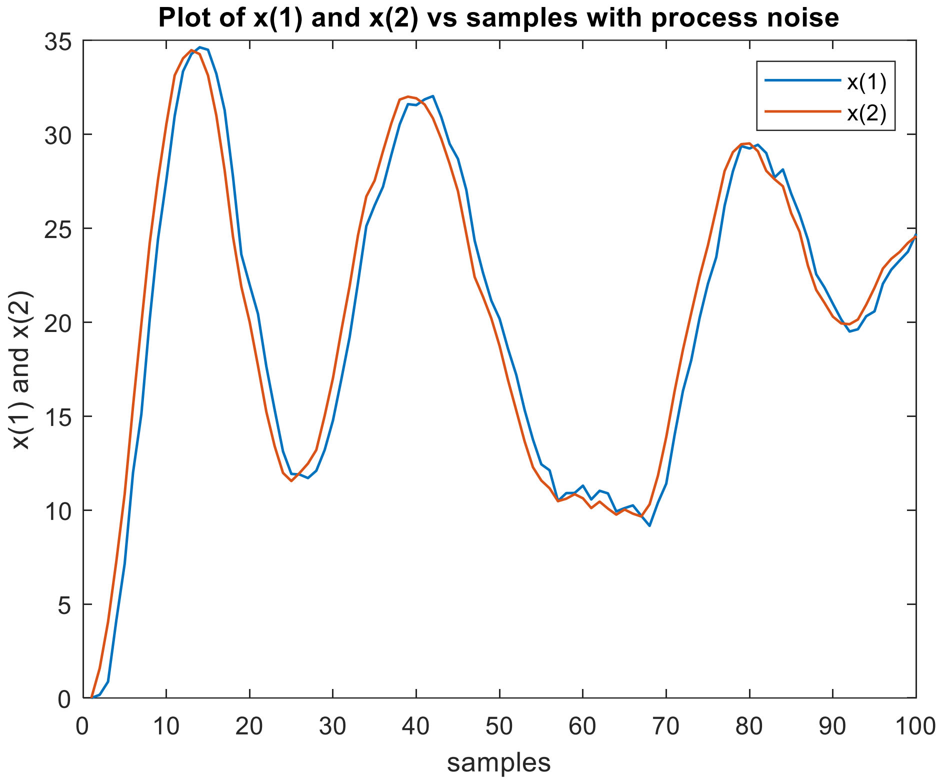


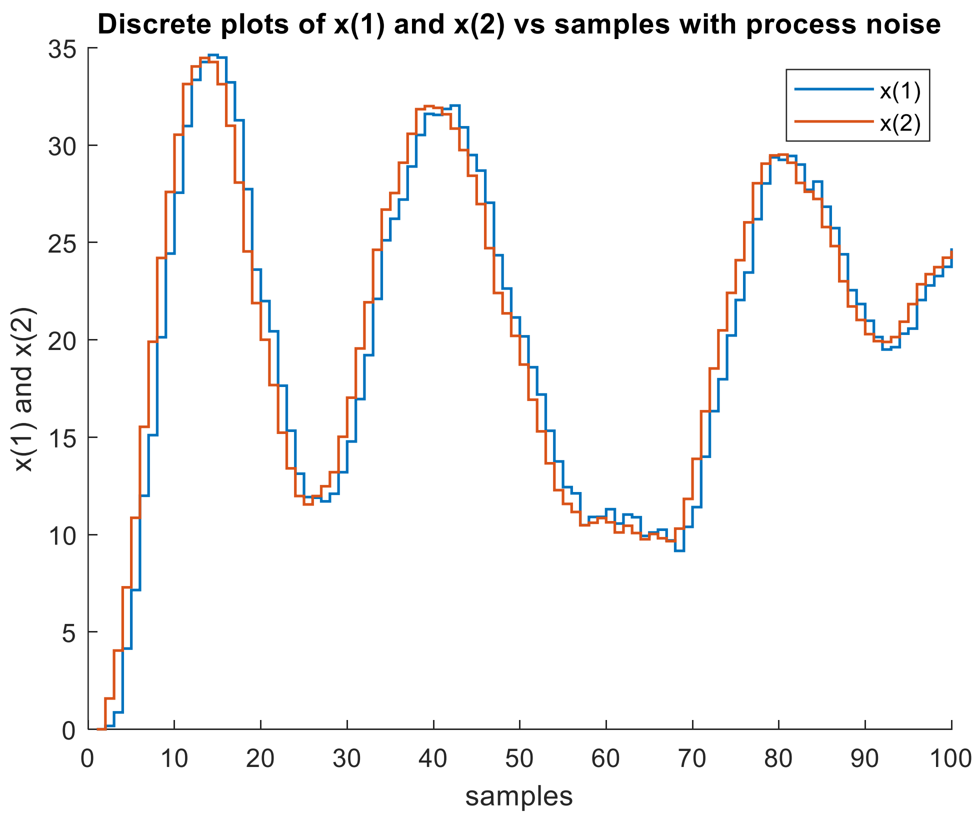


b. MATLAB code

```
function discretesystemsimulationwithnoise
x(1,2)=0;
u=1;
for k=1:1:100
    mu = [0 0];
    sigma = [0.1 0; 0 0.1];
    R = chol(sigma);
    z = repmat(mu,1,1) + randn(1,2)*R;
    x(k+1,1)=x(k,2)+z(1,1);
    x(k+1,2)=-0.9*x(k,1)+1.85*x(k,2)+u+z(1,2);
end
t=1:1:100;
figure(1)
plot(t,x(1:100,1),t,x(1:100,2))
figure(2)
hold on
stairs(t,x(1:100,1))
stairs(t,x(1:100,2))
end
```

Plots


Plot of x(1) and x(2) vs samples with process noise
x(1)
x(2)
x(1) and x(2)
samples
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60
70
80
90
100


Discrete plots of x(1) and x(2) vs samples with process noise
x(1)
x(2)
x(1) and x(2)
samples
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60
70
80
90
100

Recursive Least Squares System Identification

a. MATLAB code

```
function systemidentificationwithRLS
%fixed and initial values
cov=0.1; Pk1=10000*eye(3); thetak1=zeros(3,1);

%retrieving data
filename='Input Output data.xls';
T=readtable(filename);
uin=str2double(T{3:603,3});
yout=str2double(T{3:603,5});

%estimating system parameters
for k=3:601
    hk1=[-yout(k-1);-yout(k-2);uin(k-2)];
    Pk1=Pk1-Pk1*hk1*(inv(transpose(hk1)*Pk1*hk1+cov))*(transpose(hk1))*Pk1;
    thetak1=thetak1+Pk1*(hk1/cov)*(yout(k)-(transpose(hk1))*thetak1);
end
Sys=thetak1
end
```

Output

```
Sys =

   -1.9000

    0.9500

    0.2000
```

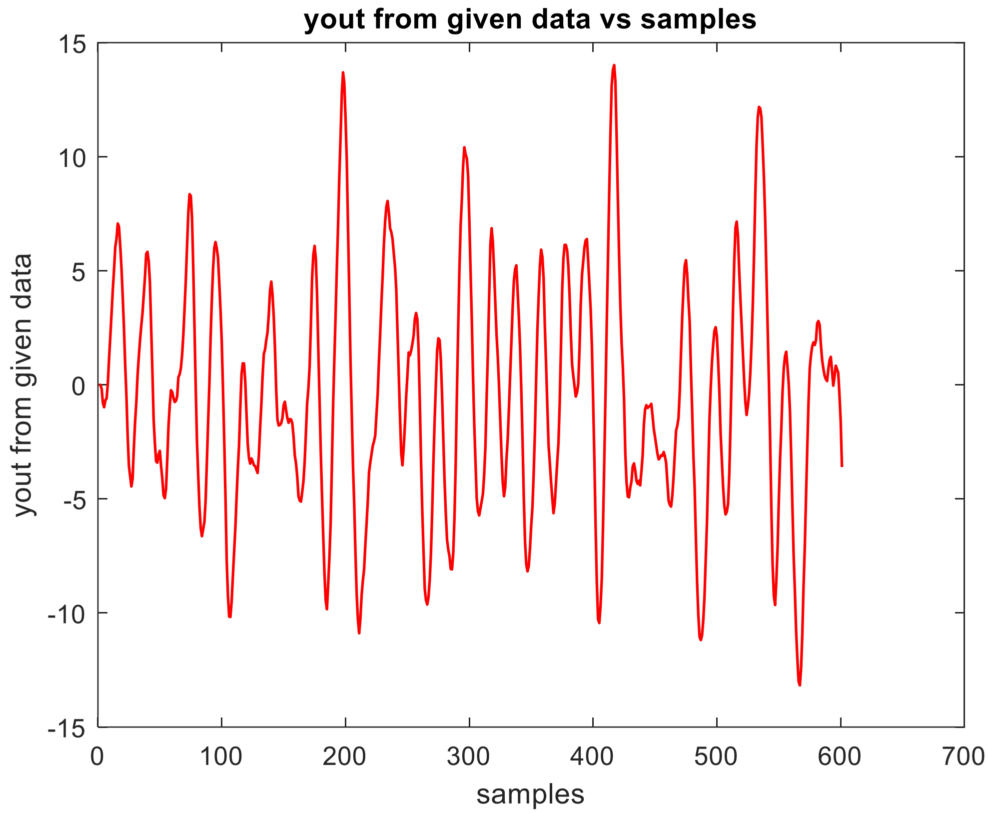
b. MATLAB Code

```
function ComparisonofEstimatedandActualOutputs
%fixed and initial values
cov=0.1; Pk1=1e6*eye(3); thetak1=zeros(3,1); Yhat=zeros(601,1); xp=zeros(2,1);

%retrieving data
filename='Input Output data.xls';
T=readtable(filename);
uin=str2double(T{3:603,3});
yout=str2double(T{3:603,5});

%estimating system parameters
for k=3:601
    hk1=[-yout(k-1);-yout(k-2);uin(k-2)];
    Pk1=Pk1-Pk1*hk1*(inv(transpose(hk1)*Pk1*hk1+cov))*(transpose(hk1))*Pk1;
    thetak1=thetak1+Pk1*(hk1/cov)*(yout(k)-(transpose(hk1))*thetak1);
end

%output from estimated parameters
for k=1:601
```

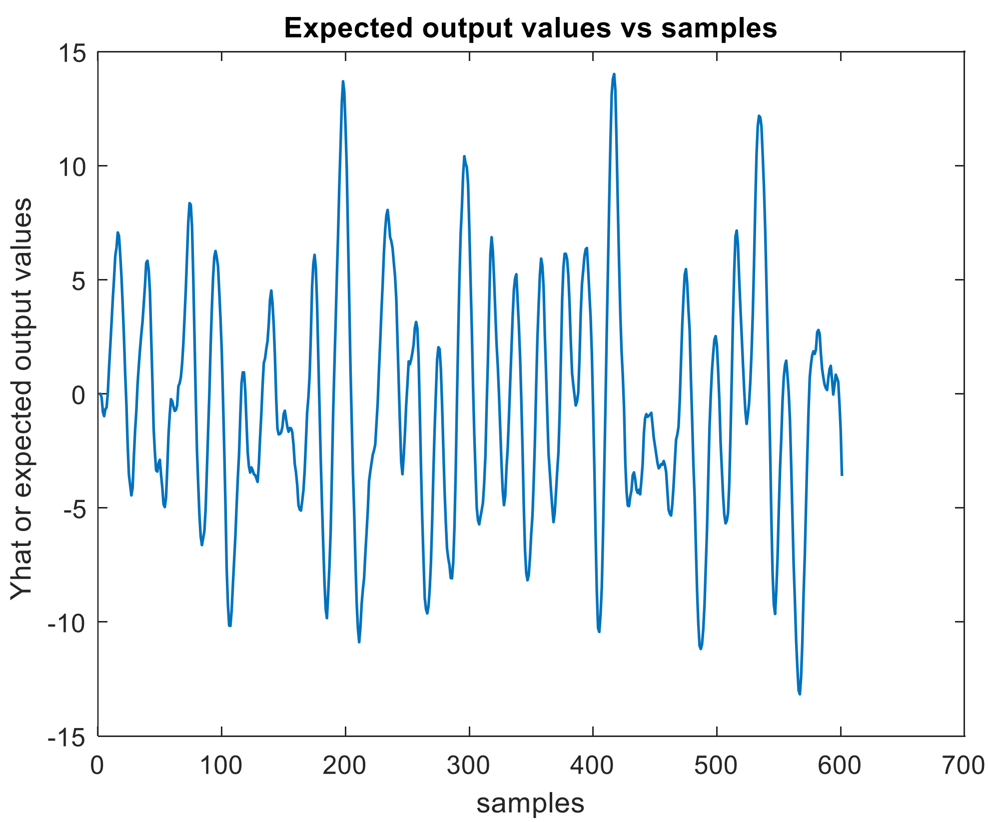
```
    Yhat(k)=[thetak1(3) 0]*xp;
    xp=[0 1;-thetak1(2) -thetak1(1)]*xp+[0;1]*uin(k);
end

%comparison plots
figure(1)
plot(1:601,yout,'r')
figure(2)
plot(1:601,Yhat,'')
figure(3)
subplot(1,2,1)
plot(1:601,yout,'r')
subplot(1,2,2)
plot(1:601,Yhat,'')
figure(4)
subplot(2,1,1)
plot(1:601,yout,'r')
subplot(2,1,2)
plot(1:601,Yhat,'')
figure(5)
hold on
plot(1:601,yout,'r','linewidth',1)
plot(1:601,Yhat,'k:','linewidth',2)
end
```

Note: I have drawn so many comparison plots to ensure that I didn’t miss out on anything and to show better that both the plots are same. The plots are given on the next page.

Plots

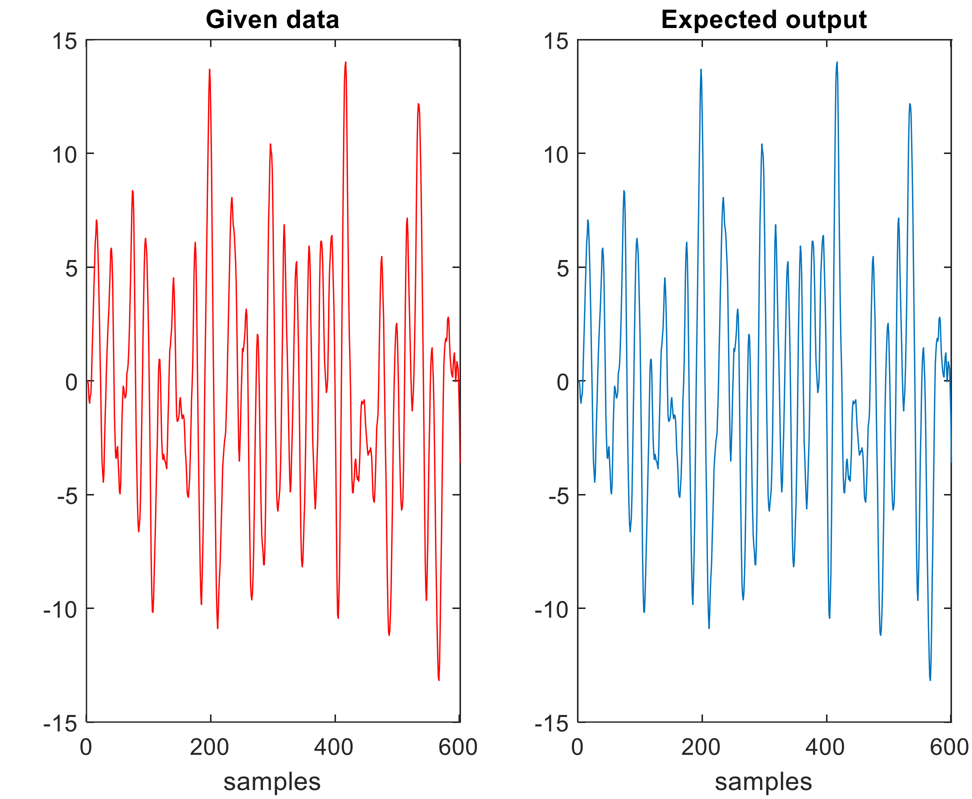


yout from given data vs samples
15
10
5
0
-5
-10
-15
0
100
200
300
400
500
600
700
samples
yout from given data

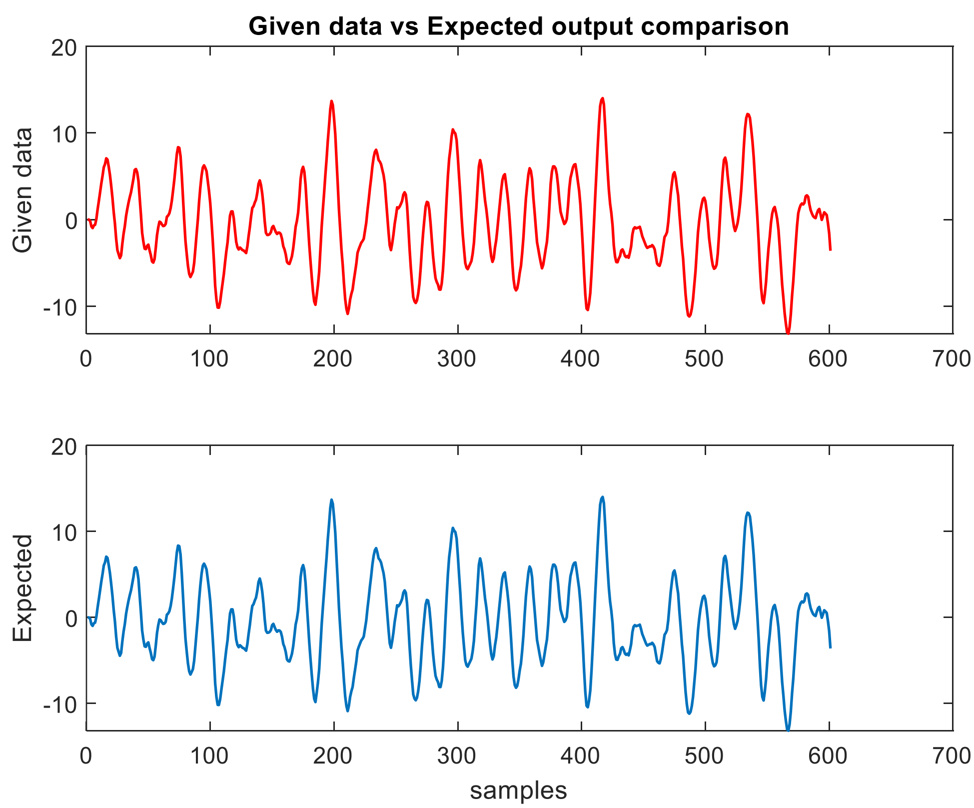


Expected output values vs samples
15
10
5
0
-5
-10
-15
0
100
200
300
400
500
600
700
samples
Yhat or expected output values

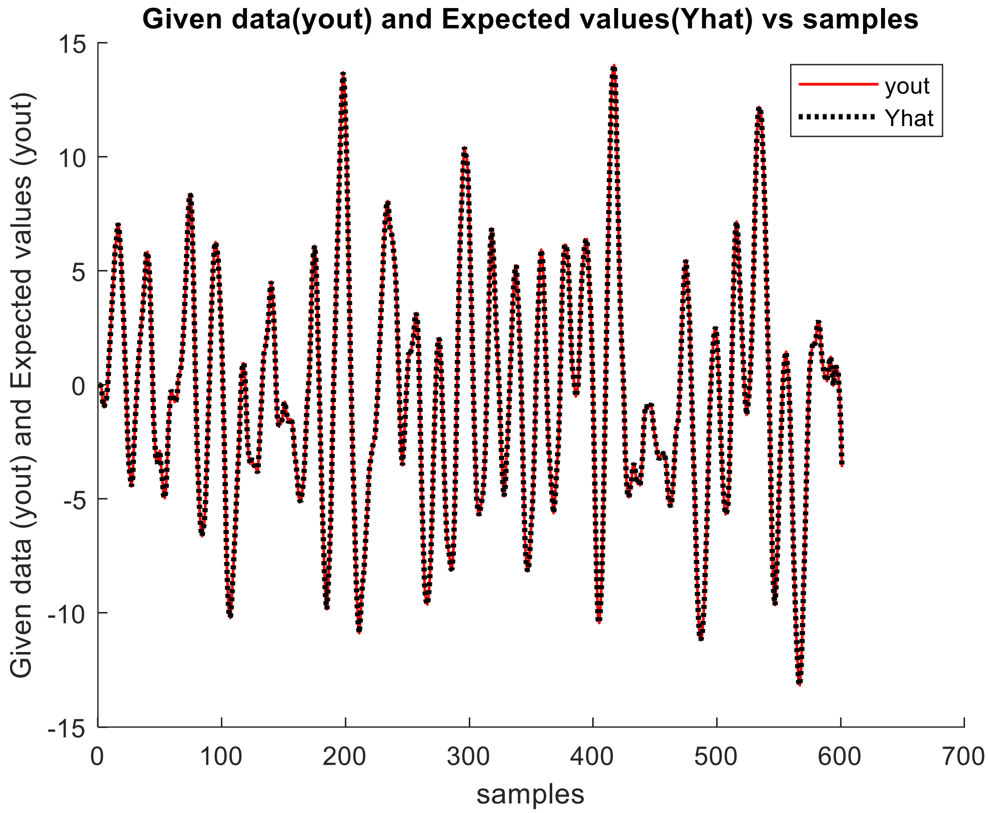
Side by Side Comparison



Top Bottom Comparison



Comparison on the same axes



References

1. Intelligent Control Systems – Dr. Frank L. Lewis (Professor, Electrical Engineering, The University of Texas at Arlington)