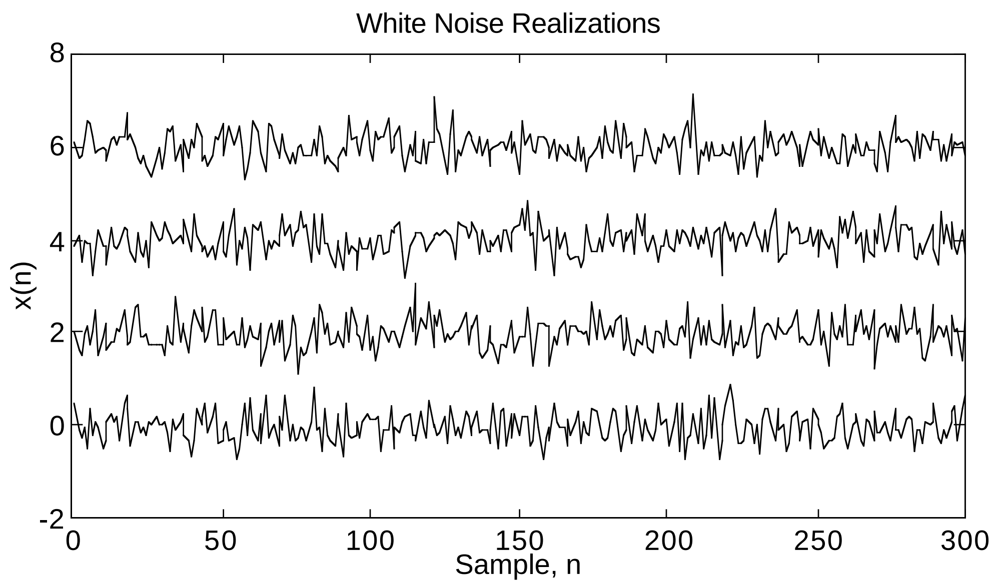
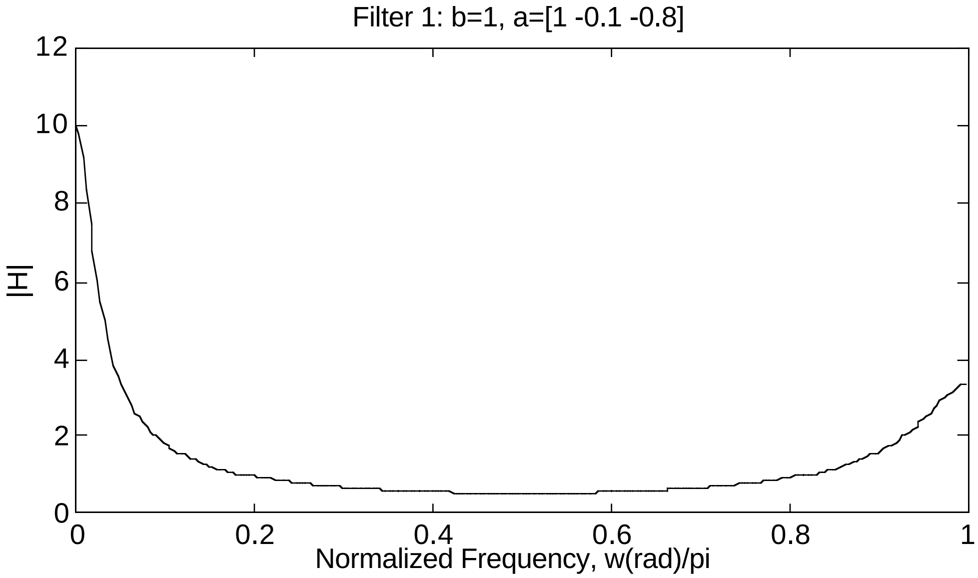
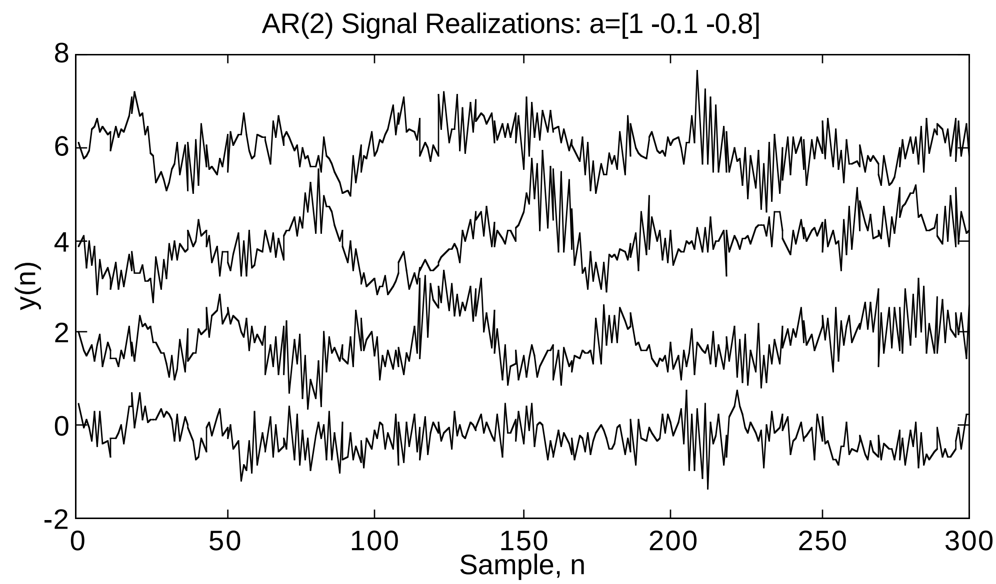
```
% EE254                                                    Prof. Essam Marouf
% DSP II                                                   Matlab Handout #1
% ----------------------------------------------------------------------------------------------------
% Models of Random Signals
% ---------------------------------
%
% 1) White Gaussian Noise Model
% --------------------------------------
clear, close all
N=300;                                 % number of samples
n=[1:N];                               % discrete-time sample index
%
% Generate four 1xN realizations of a white noise vector (zero mean, standard dev = 0.3)
% and store in the rows of matrix X
sigma_v=0.3;
for k=1:4,
 x(k,:)=sigma_v*randn(size(n));        % each new call to rand reseeds the random
                                       % number generator
end
% plot the four signal realizatons displaced 2 unit along the vertical axis
plot(n,x(1,:),n,x(2,:)+2,n,x(3,:)+4,n,x(4,:)+6),
xlabel('Sample, n'), ylabel('x(n)'),
title('White Noise Realizations')
pause
```

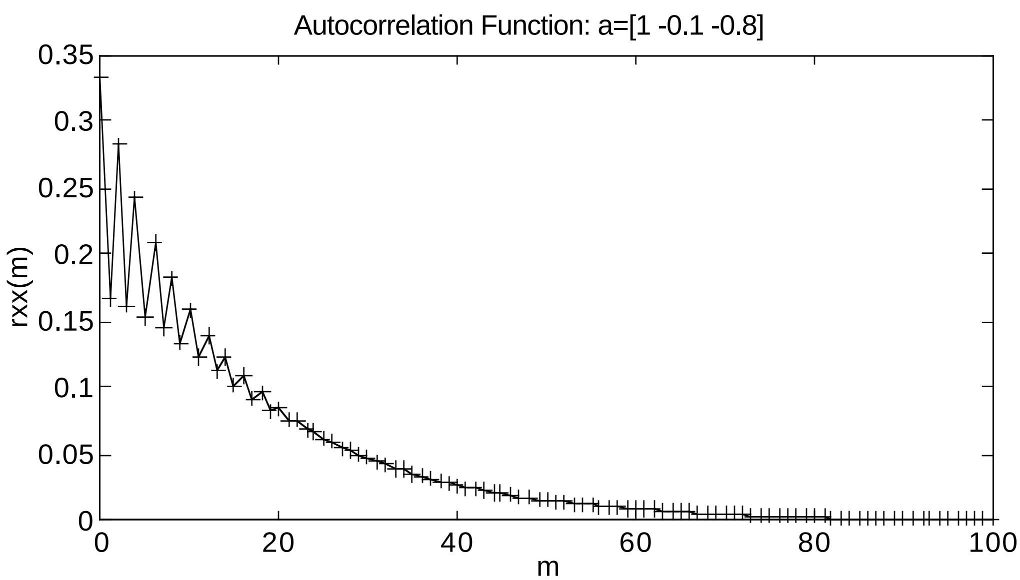


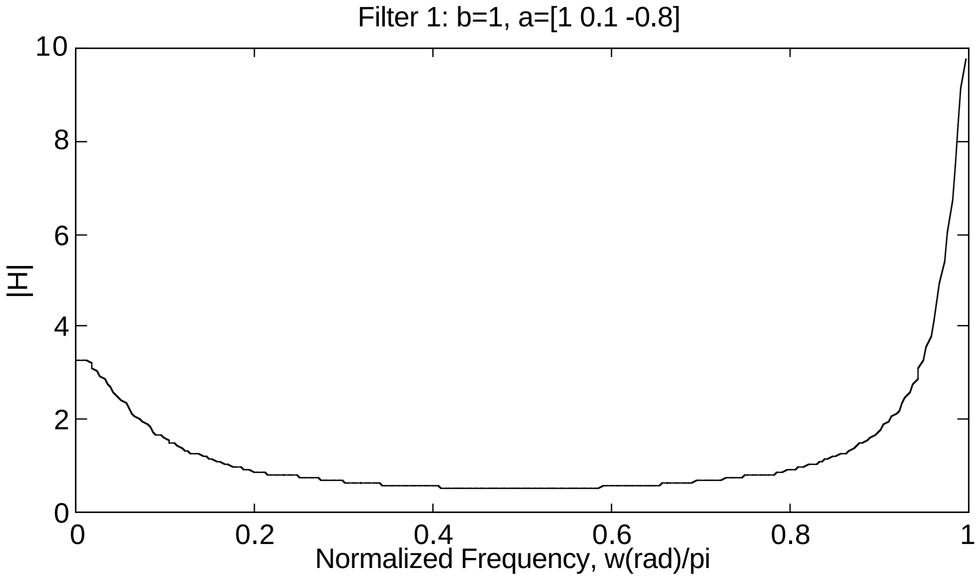

```
% 2) AR(2) Signal Model: Filter 1
% -------------------------------------
a=[1 -0.1 -0.8];                       % All-pole IIR Filter Coefficients
b=1;
c=roots(a)
        c =
             0.9458
            -0.8458
% This filter emphasizes low frequencies at its output; this may be
% seen from a plot of its amplitude response:
nn=256;
[H,w]=freqz(b,a,nn);                   % 0 <= w <= pi (digital frequency)
plot (w/pi, abs(H),'-')                % plot of |H| (amplitude response)
xlabel('Normalized Frequency, w(rad)/pi'),
ylabel('|H|'), title ('Filter 1: b=1, a=[1 -0.1 -0.8]')
```

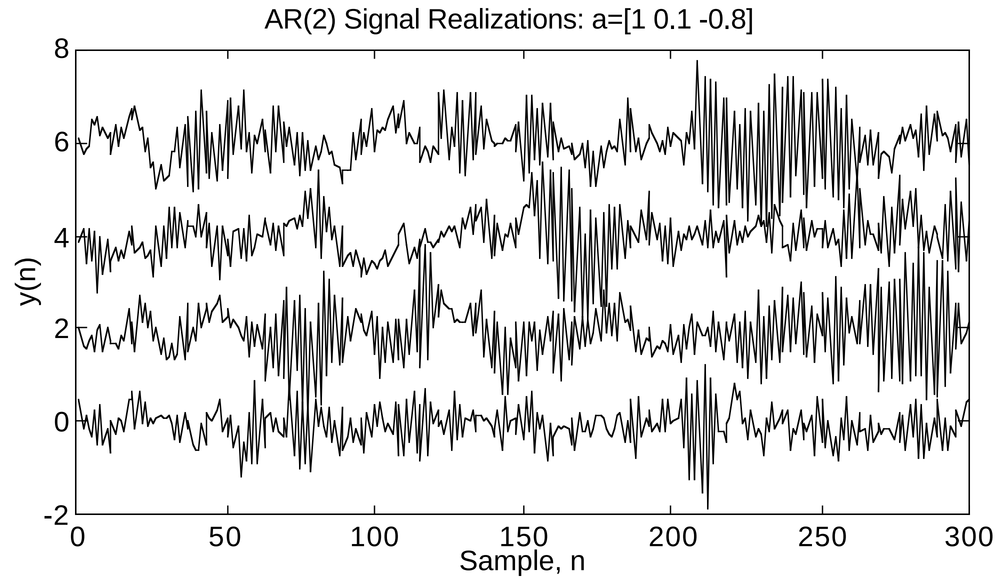
```
pause
%
% The AR(2) signal models are generated by filtering the white noise
% through the filter above
for k=1:4,
y1(k,:)=filter(b,a,x(k,:));
end
% plot the four signal realizatons displaced 2 unit along the vertical axis
plot(n,y1(1,:),n,y1(2,:)+2,n,y1(3,:)+4,n,y1(4,:)+6),
xlabel('Sample, n'), ylabel('y(n)'),
title('AR(2) Signal Realizations: a=[1 -0.1 -0.8]')
pause
% Autocorrelation Function rxx(m) (see lecture notes)
% -----------------------------------------------------------
rxx(1)=sigma_v^2/ (1- a(3)^2- (a(2)^2* (1-a(3)) / (1+a(3))) );
% rxx(1) is really rxx(0), a(2) is a(1), ...etc; Matlab indices start from 1
rxx(2)=rxx(1)*(-a(2))/(1+a(3));
NN=101;
for m=3:NN;
  rxx(m)=-a(2)*rxx(m-1)-a(3)*rxx(m-2);
end
plot([0:NN-1], rxx,'+',[0:NN-1], rxx,'-')
xlabel('m'), ylabel('rxx(m)'),
title('Autocorrelation Function: a=[1 -0.1 -0.8]') ,pause
%
%  AR(2) Signal Model: Filter 2
%  ----------------------------------
a=[1 0.1 -0.8];
b=1;
c=roots(a)
c =
     -0.9458
      0.8458
% Again, two real roots.
% This filter emphasizes high frequencies at its output; this may be
% seen from a plot of its amplitude response:

% ...............Code is the same as before; omitted for brevity

%  AR(2) Signal Model: Filter 3
%  -----------------------------------
a=[1 -0.975 0.95];
b=1;
c=roots(a)
c =
   0.4875 + 0.8440i
   0.4875 - 0.8440i
% Complex conjugate pole pair (Bandpass filter).
% ...............Code is the same as before; omitted for brevity
```

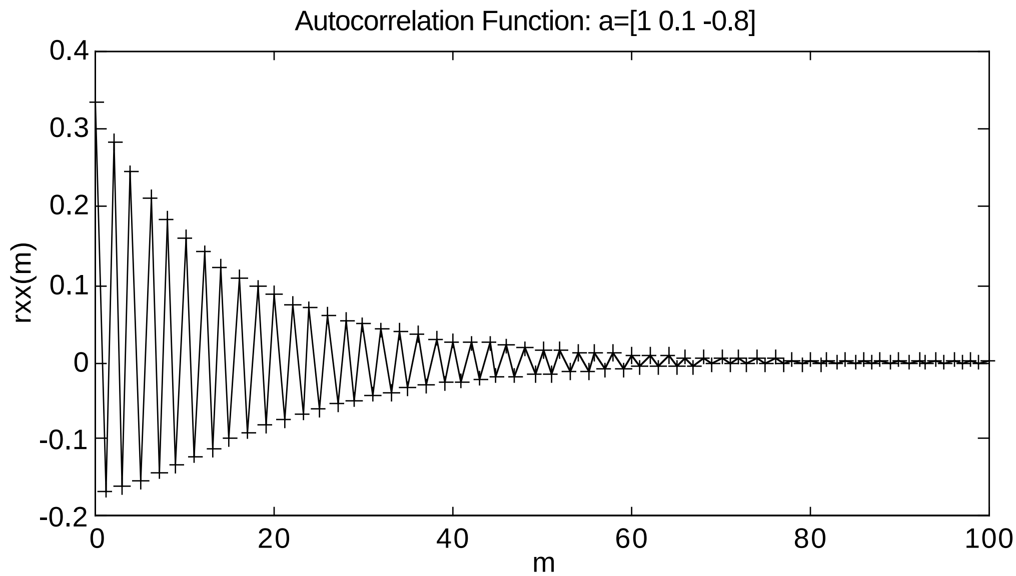

Filter 1: b=1, a=[1 -0.1 -0.8]
|H|
0
2
4
6
8
10
12
0
0.2
0.4
0.6
0.8
1
Normalized Frequency, w(rad)/pi

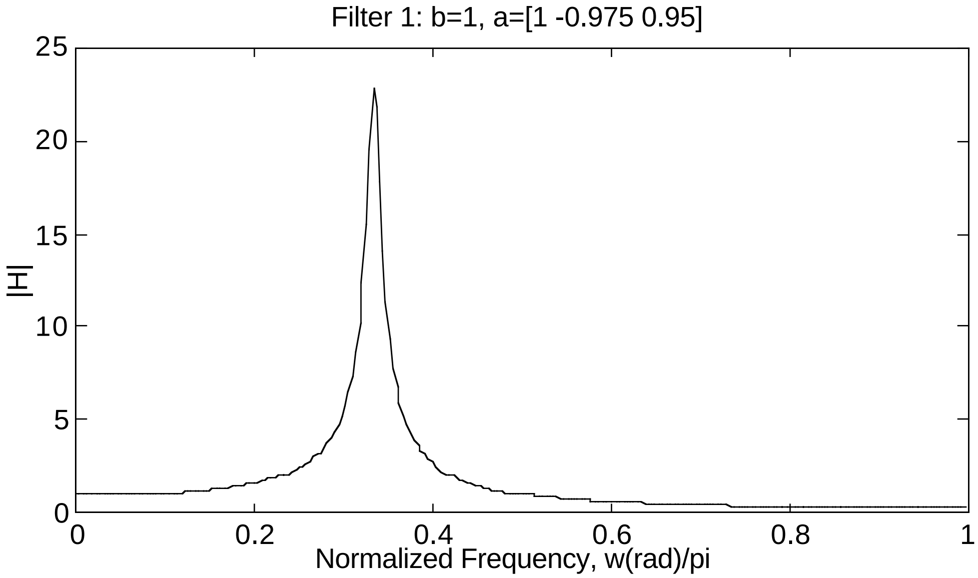

AR(2) Signal Realizations: a=[1 -0.1 -0.8]
y(n)
-2
0
2
4
6
8
0
50
100
150
200
250
300
Sample, n

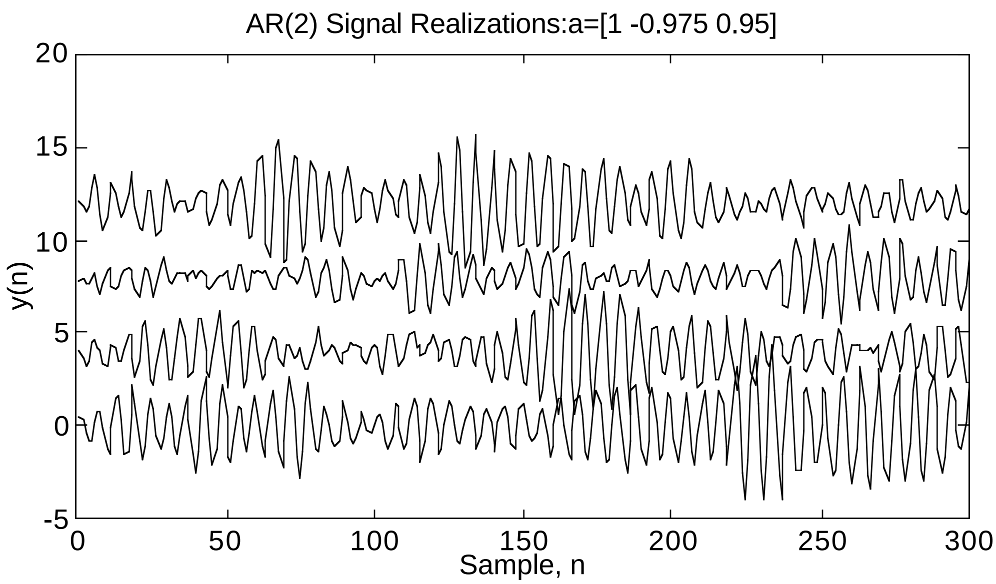

Autocorrelation Function: a=[1 -0.1 -0.8]
rxx(m)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0
20
40
60
80
100
m

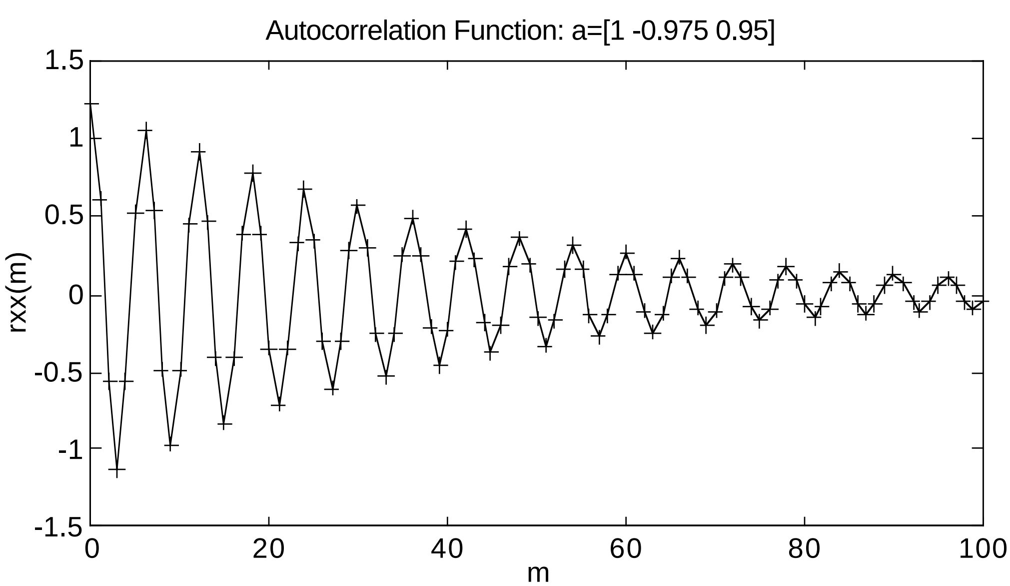

Filter 1: b=1, a=[1 0.1 -0.8]
|H|
0
2
4
6
8
10
0
0.2
0.4
0.6
0.8
1
Normalized Frequency, w(rad)/pi


AR(2) Signal Realizations: a=[1 0.1 -0.8]
y(n)
-2
0
2
4
6
8
0
50
100
150
200
250
300
Sample, n


Autocorrelation Function: a=[1 0.1 -0.8]
rxx(m)
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
m


Filter 1: b=1, a=[1 -0.975 0.95]
|H|
Normalized Frequency, w(rad)/pi
0
5
10
15
20
25
0.2
0.4
0.6
0.8
1


AR(2) Signal Realizations:a=[1 -0.975 0.95]
y(n)
Sample, n
-5
0
5
10
15
20
50
100
150
200
250
300


Autocorrelation Function: a=[1 -0.975 0.95]
rxx(m)
m
-1.5
-1
-0.5
0
0.5
1
1.5
20
40
60
80
100