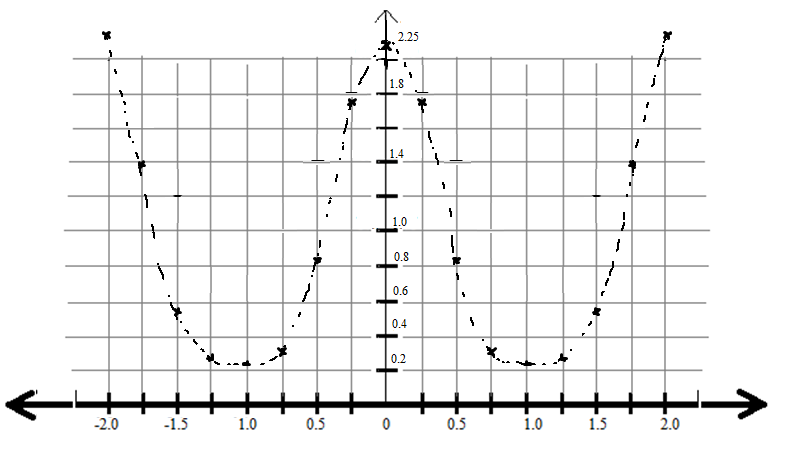
**Lab 4: Digital Filters**

**Problem 1(Single echo) :**

A single echo FIR filter is defined by the difference equation

* Positive constant
* Integer parameter R
* System Sampling period Ts

1. Theory
2. Write down the transfer function H(z)
3. Calculate and plot (with pen and paper) the square of the amplitude response within the frequency interval from 0 to ws. Choose R = 3 and for this purpose.

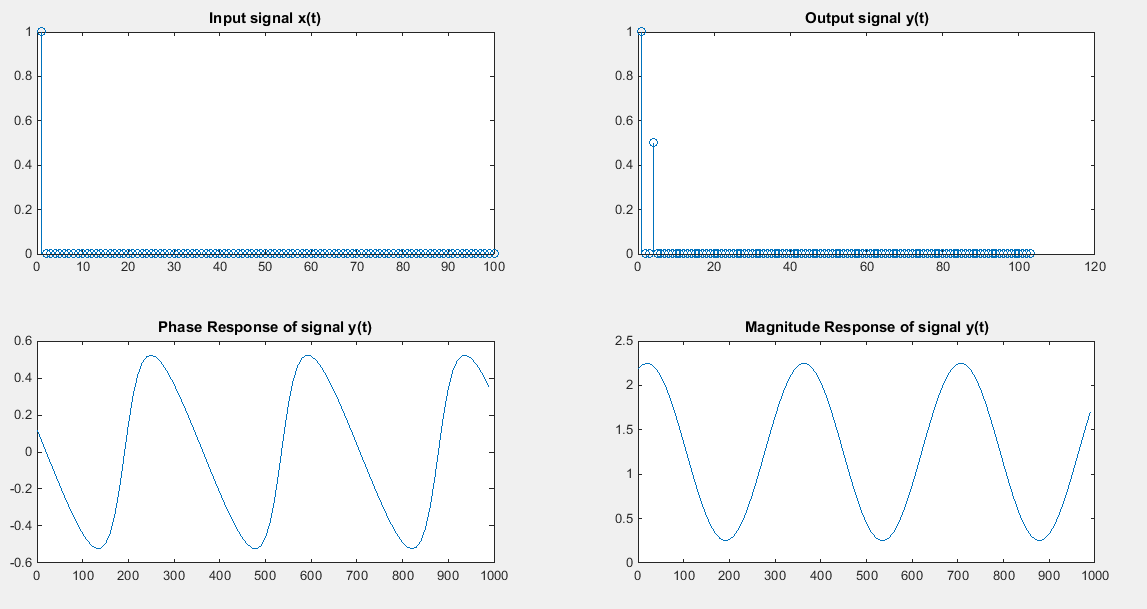


1. Matlab Implementation
2. Write a Matlab function implementing a single echo filter that takes the vector x of input values as well as the parameters and R as input arguments and that returns the output vector y.

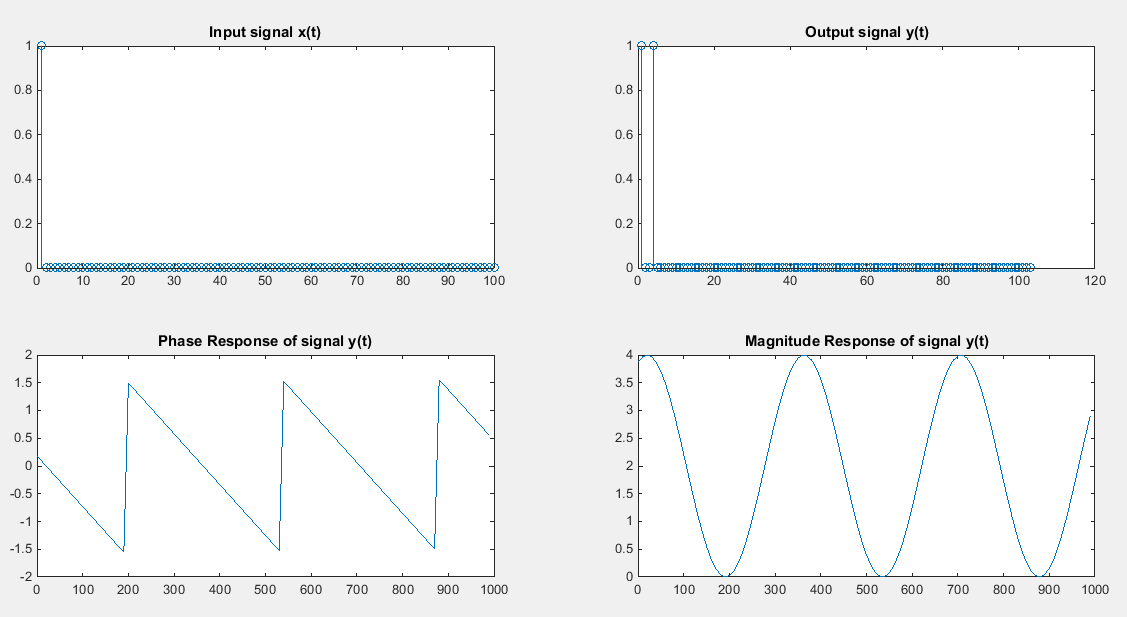
|  |
| --- |
| function[] = Test(u)    alpha = 1/2;  R = 3;  N = length(u);  dt = 0.001;  T = dt\*N;  Ts = T/N;  f = 1/T;  fs = 1/Ts;  F = 0:f:fs-f;      x = [u zeros(1,R)];  y = x;    for i=1:length(u)  L = i-R;  if L>0  y(i) = u(i) + alpha\*u(i-R);  else  y(i) = u(i);  end  end  H = fftshift(fft(y));  H = [H H];  H = H(length(u)/2:1.5\*length(u)-1);  subplot(2,2,1);  stem(u);  title('Input signal x(t)');  subplot(2,2,2);  stem(y);  title('Output signal y(t)');  subplot(2,2,3)  plot(F,abs(phase(H)));  title('Phase Response of signal y(t)');  subplot(2,2,4)  plot(F,abs(power(H,2)));  title('Magnitude Response of signal y(t)');  end |

1. Set and R = 3. Test your function using a discrete as input. Use MATLAB to calculate the frequency response as the DFT of the impulse response h [n] and plot its magnitude and phase between 0 Hz and fs. Additionally choose and explain the observed phase response in this case.

Alpha = ½,R = 3



With alpha = 1,R = 3



* "Linear Phase" refers to the condition where the phase response of the filter is a linear .This results in the delay through the filter being the same at all frequencies. Therefore, the filter does not cause "phase distortion" or "delay distortion".

1. Apply the echo filter to a sound file(eg., any .wav file). Choose R so that

Td as a starting point . Vary and R and observe the respective effects.

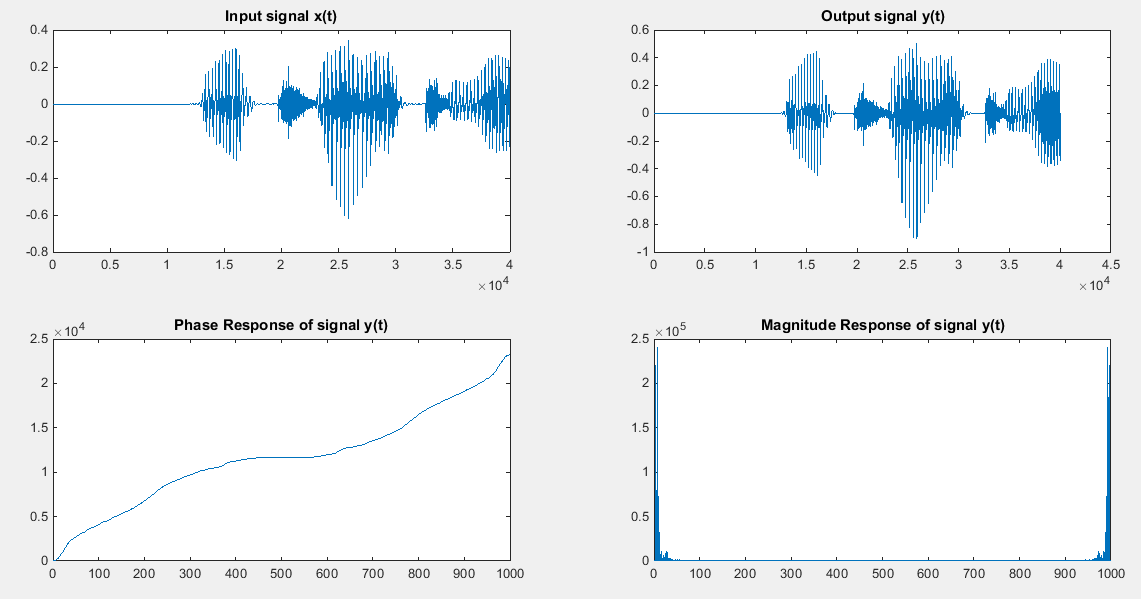
**filename = 'C:\Users\Muhammad Afaque Khan\Songs\song.wav';**

**N = 1;**

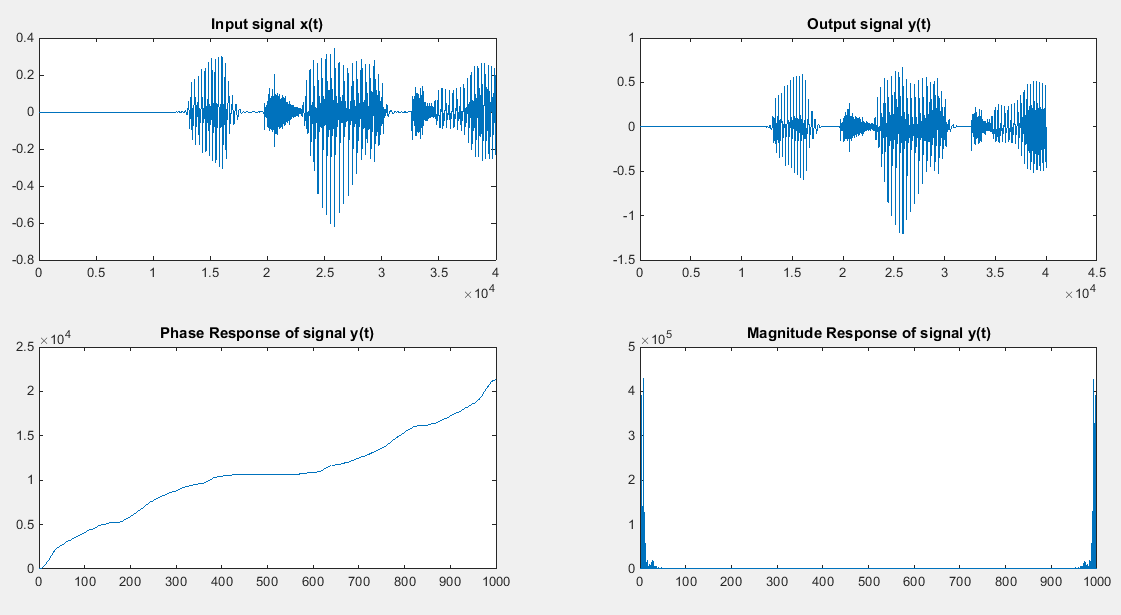
**[ ~, Fs ] = wavread(filename,N);**

|  |
| --- |
| **Fs = 48000 Hz**  **Ts = 2.08\*10e-5** |

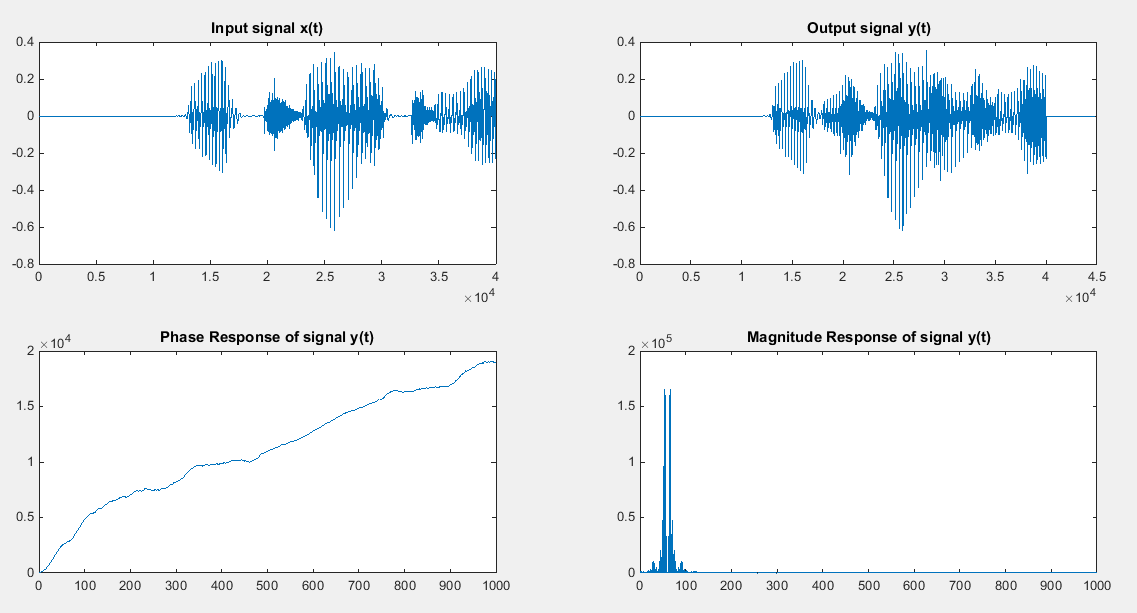
**R = 3,a = 1/2**

****

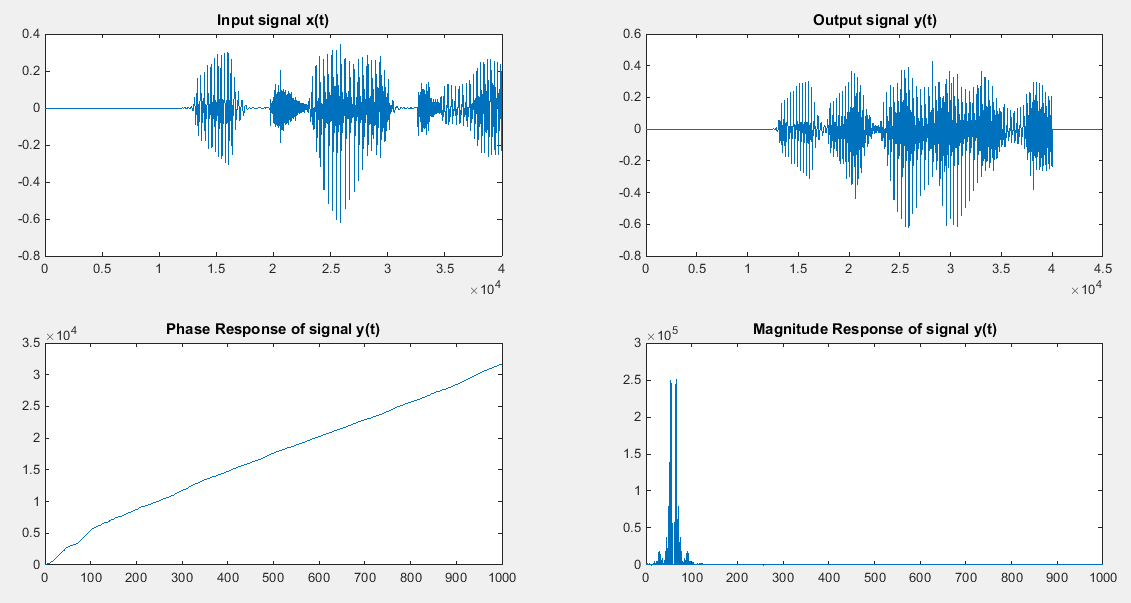
**R =3,a = 1**

****

**R = 4800, a = 1/2**

****

**R = 4800 a = 1**

****

**Problem 2(Multiple echoes):**

An N-echo FIR filter is defined by the difference equation

1. Theory
2. Write down the transfer function H (z) and simplify the expression using geometric sum. Does H (z) have poles in the complex z plane?

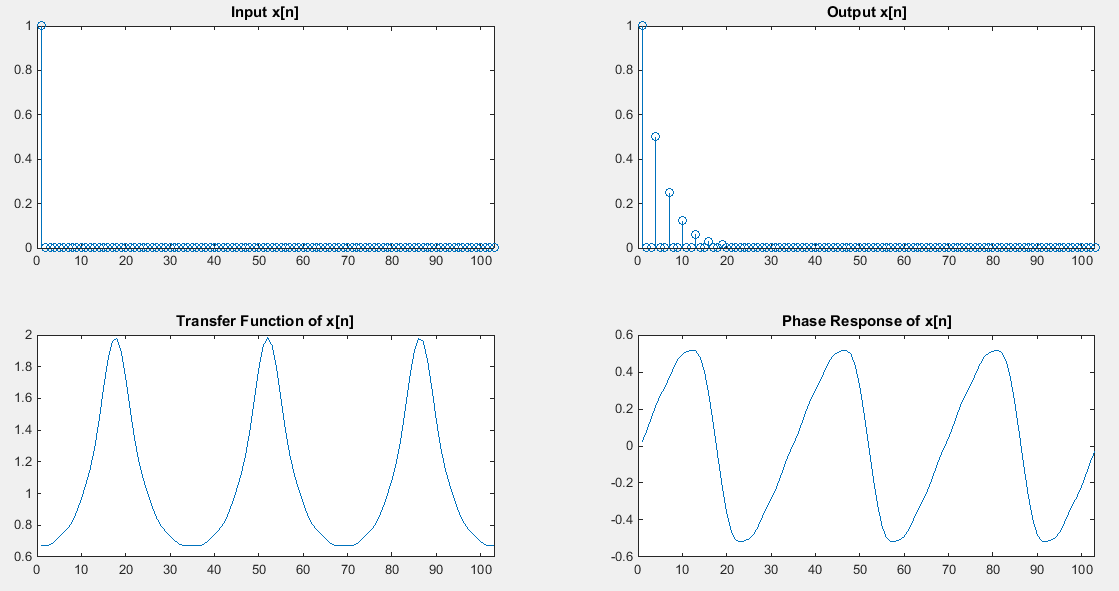
* When <1 the geometric sum converges which means that the transfer function H(Z) does not have poles in the complex Z-plane.

1. Use the result of part i) to write down the difference equation of a recursive filter describing the same N-echo filter.

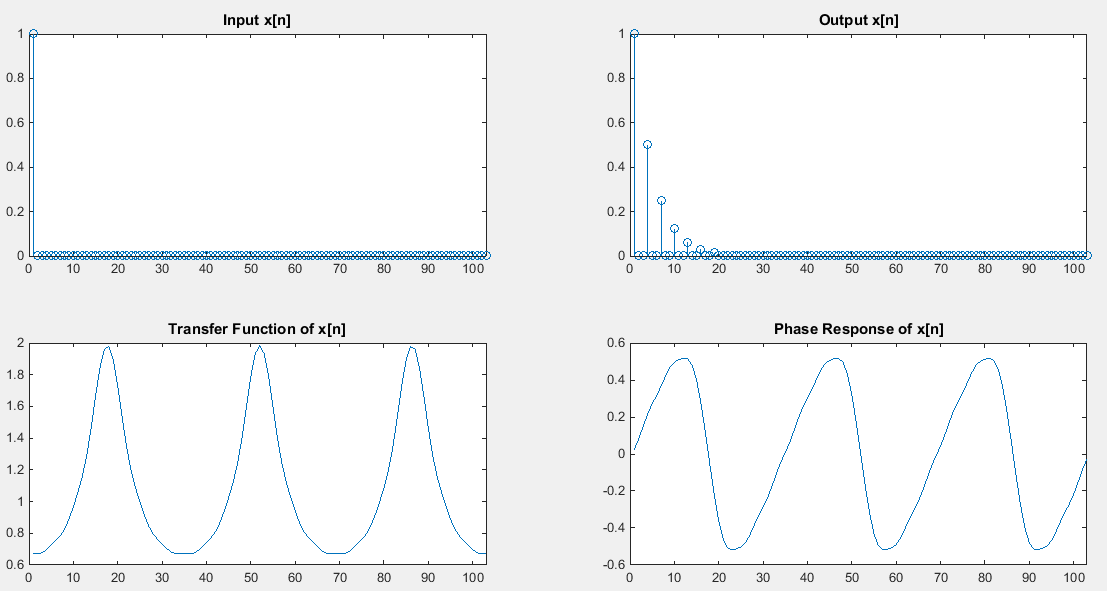
* Recursive difference equation describes an IIR filter

1. MATLAB implementation
2. Implement both the recursive and non-recursive filters for N= 6 and verify that they yield the same impulse response.

**NON RECURSIVE, a=1/2,R=3**

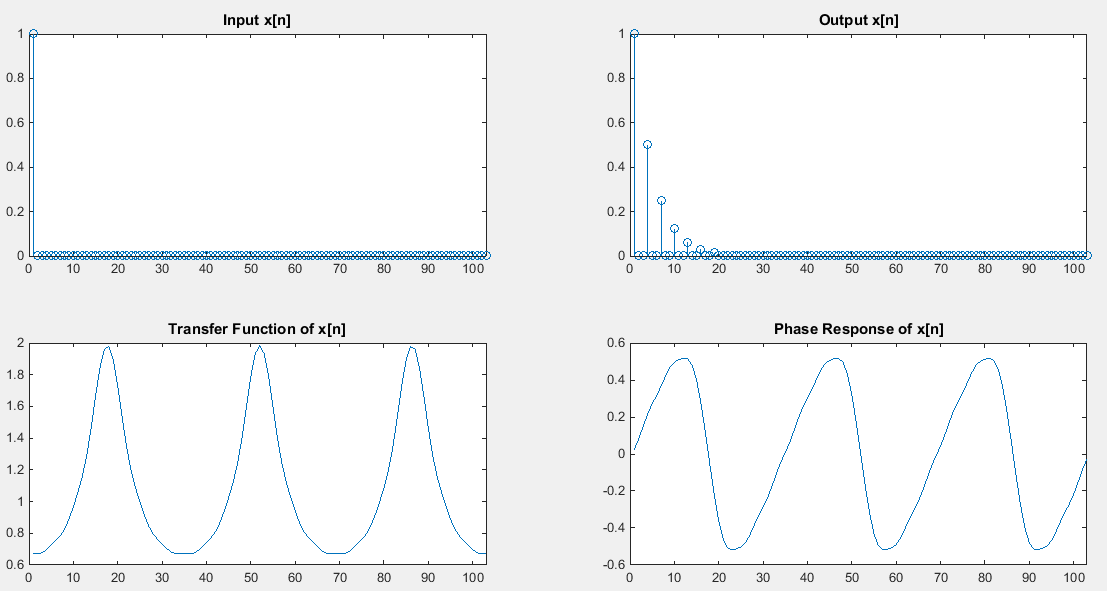


**RECURSIVE ,a=1/2,R = 3**



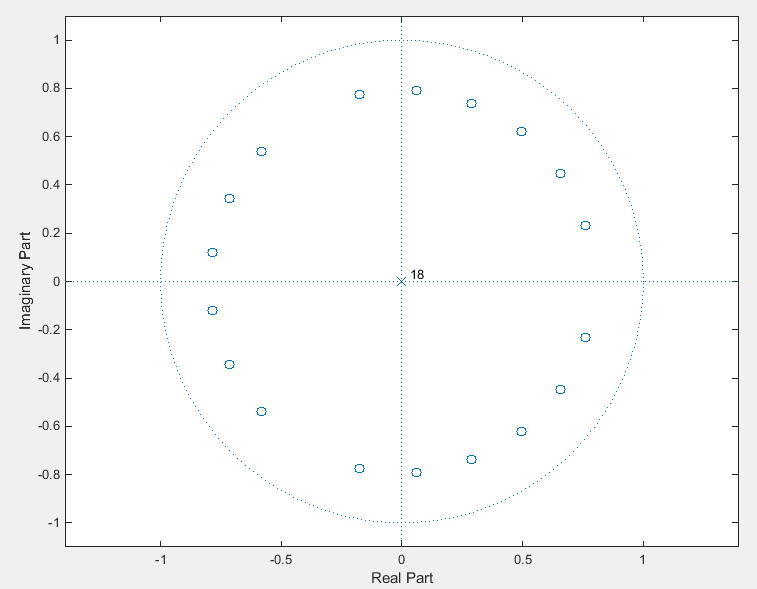


* Use MATLAB to calculate the DFT of the impulse response and plot the amplitude and phase responses(Hint: If you do this correctly, you should be able to see why this type of filter is also called comb filter).

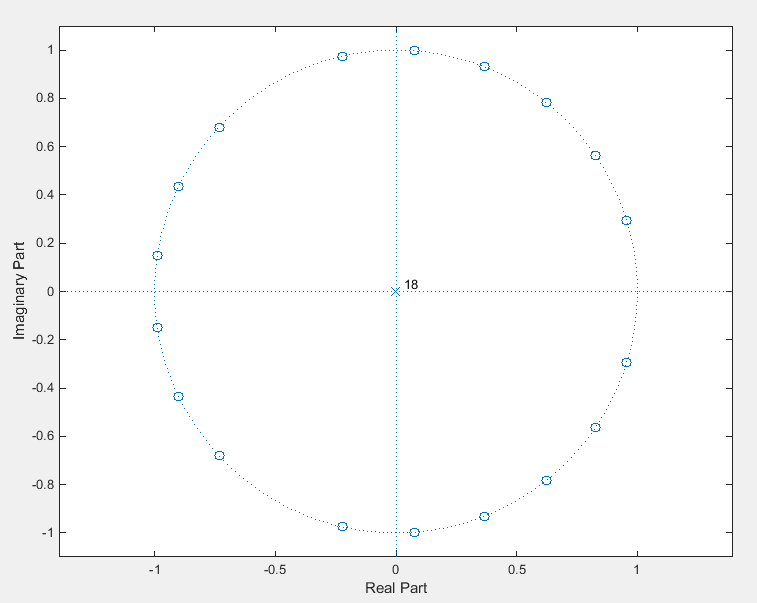


**Z-PLANE**

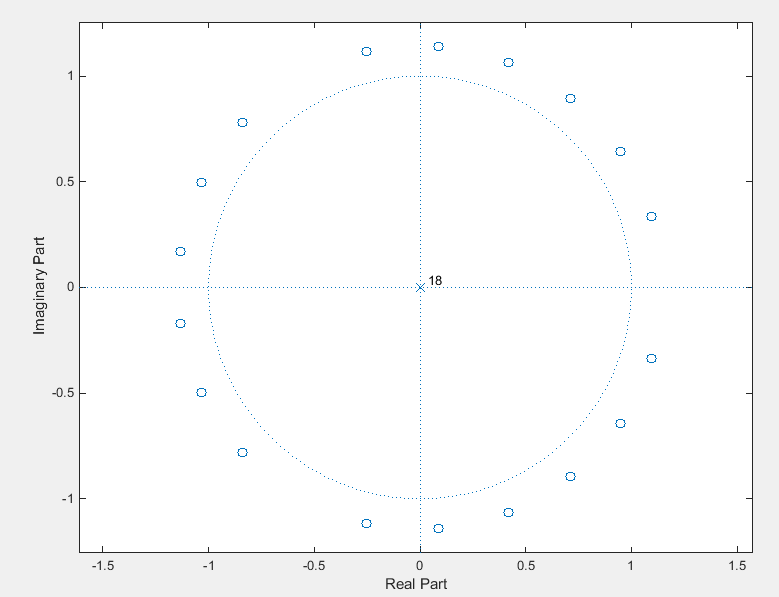
**RECURSIVE: R =3,a=0.5**



**RECURSIVE: R = 3,a = 1**



**RECURSIVE : R = 3,a=1.5**

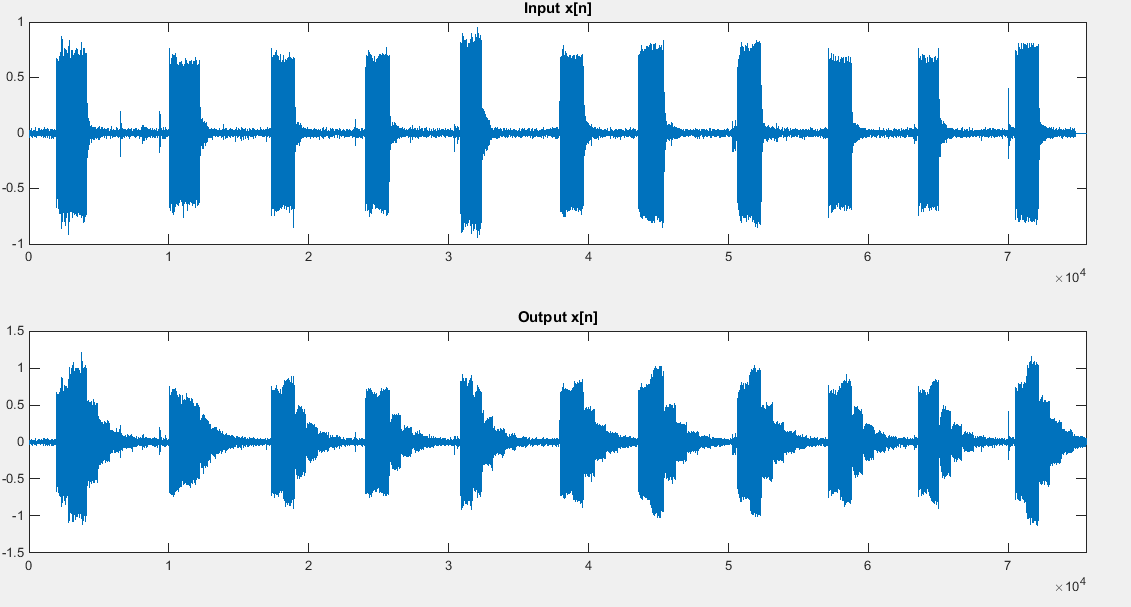


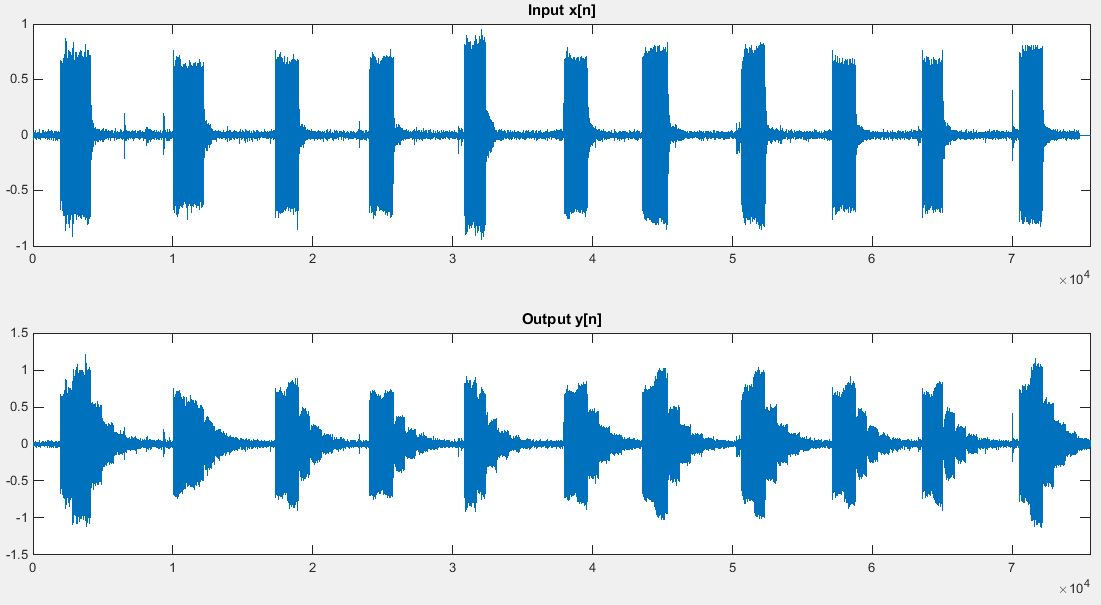
1. Apply both the recursive and non-recursive filters to a sound file. Choose R so that Td = 100ms as a starting point. Vary and R, particularly testing the behavior for . Explain the different behaviors of the recursive and non-recursive implementation in this case.

|  |
| --- |
| **N = 1;**  **[ ~, Fs ] = wavread(filename,N);**  **Fs = 8192** |
| **NON-RECURSIVE**  function[y] = Test(x)    N = 6;  R = 3;  a = 1/2;  x = [x zeros(1,R)];  y = x;  for n=0:length(y)  for k=1:N  L = n-(k\*R);  if L>0  y(n) = y(n) + (a.^k)\*x(n-R\*k);  end    end  end  H = fftshift(fft(y));  subplot(2,2,1);  stem(x);  title('Input x[n]');  xlim([0 length(y)]);  subplot(2,2,2);  stem(y);  title('Output x[n]');  xlim([0 length(y)]);  subplot(2,2,3);  plot(abs(H));  title('Transfer Function of x[n]');  xlim([0 length(y)]);  subplot(2,2,4);  plot(phase(H));  title('Phase Response of x[n]');  xlim([0 length(y)]);    end | | |

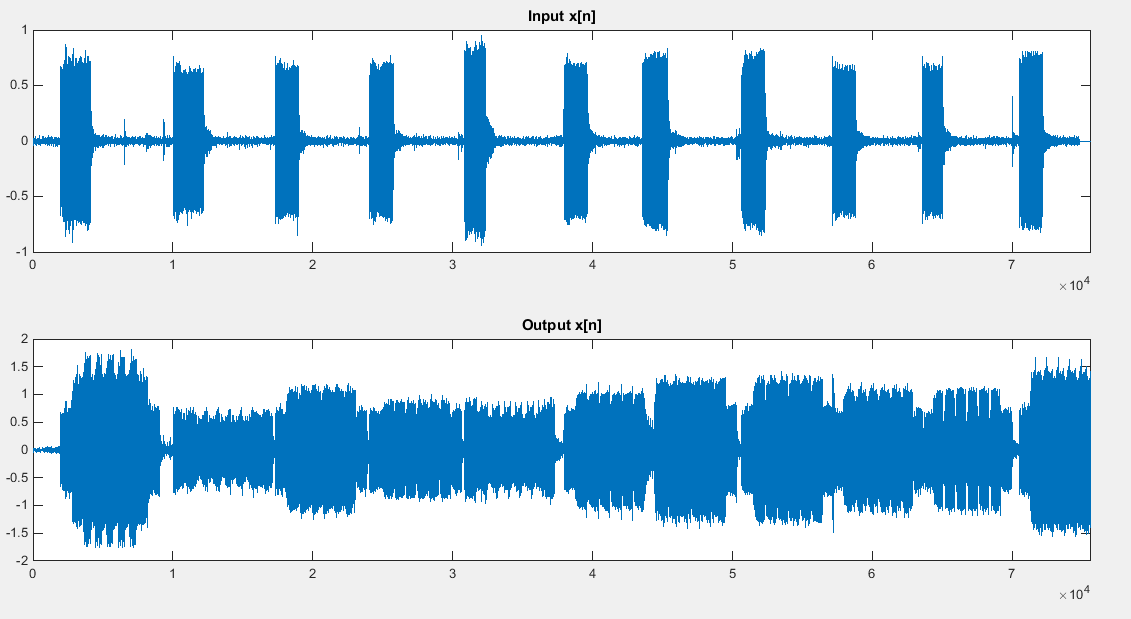
|  |
| --- |
| **RECURSIVE**  function[y] = Test(x)    N = 6;  R = 3;  a = 1/2;  x = [x zeros(1,R)];  y = x;  for n=0:length(y)  L = n-(R\*(N+1));  if L>0  y(n) = y(n) - (a.^(N+1))\*x(n-R\*(N+1));  end  if n>R  y(n) = y(n) + a\*y(n-R);  end  end  H = fftshift(fft(y));  subplot(2,2,1);  stem(x);  title('Input x[n]');  xlim([0 length(y)]);  subplot(2,2,2);  stem(y);  title('Output x[n]');  xlim([0 length(y)]);  subplot(2,2,3);  plot(abs(H));  title('Transfer Function of x[n]');  xlim([0 length(y)]);  subplot(2,2,4);  plot(phase(H));  title('Phase Response of x[n]');  xlim([0 length(y)]);    end |

**RECURSVE R = 819,a = 0.5**

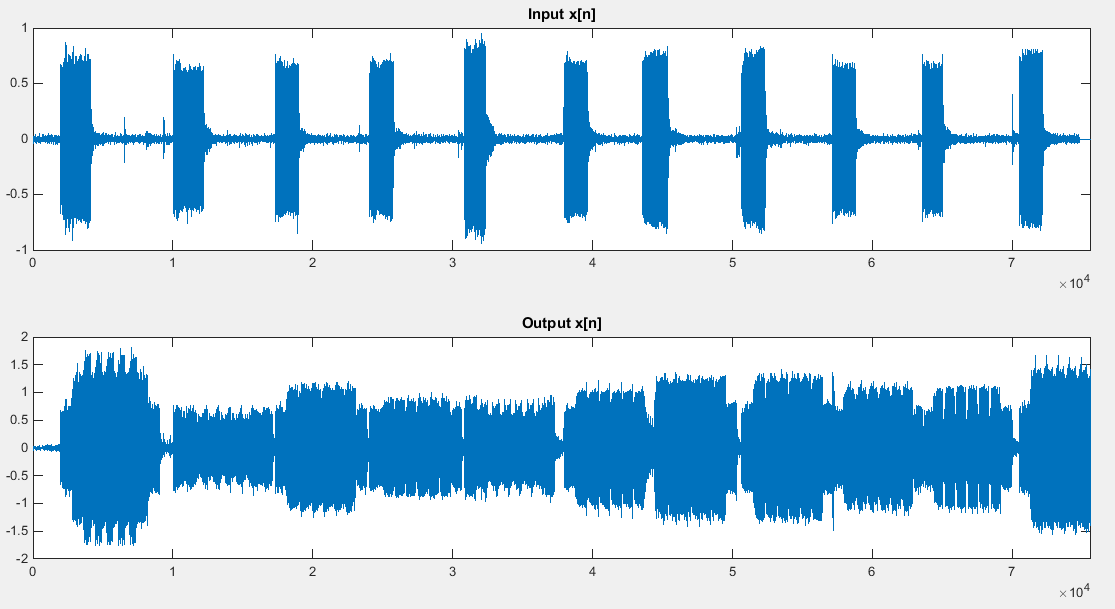


**Non-Recursive : R = 819,a = 0.5**

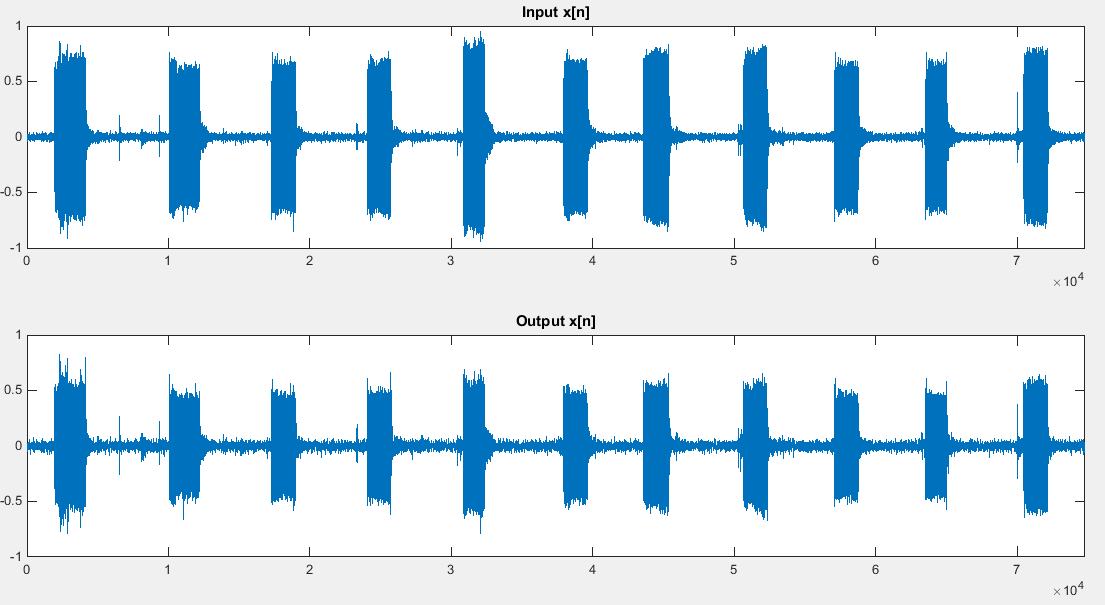
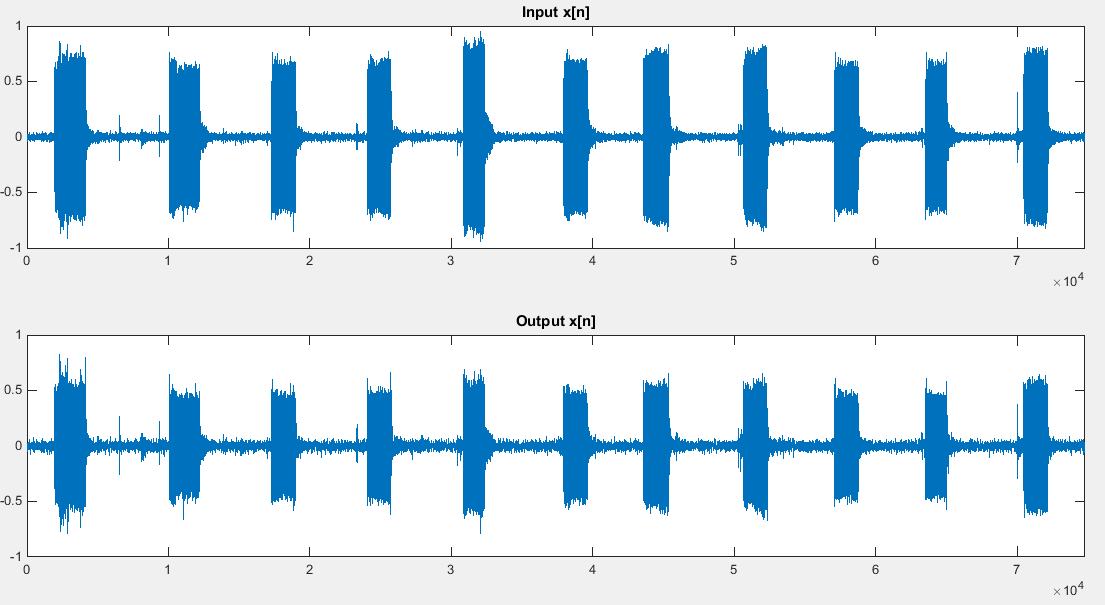
**RECURSIVE : R = 819,a = 1**



**NON-RECURSIVE : R = 819,a = 1**



**NON-RECURSIVE : R = 3,a =1 / 2 RECURUSIVE : R = 3,a =1 / 2**

 ****

* Recursive and Non-Recursive filter give the same output for same R and alpha
* Increasing R increases the delay in the filter resulting in the echo
* Decreasing alpha <1 gives an output with a exponentially decreasing echo.