ADVANCED SIGNAL ANALYSIS AND PROCESSING LAB

Lab sheet. No: 03

NAME: Pedasingu Rajesh **Roll. No:** EE21M019

Q1. Using MATLAB, find the odd and even part of the signal given below:

(a) x1[n] = [5 4 6 7 3 2]

(b) x2[n] = sin(2πfn) + cos(πfn)

AIM: To generate and plot the even and odd part of the given signal.

Short Theory: A signal x[n] can be decomposed into its even, xe, and odd, xo, parts. By,

$$xe(n) = \frac{x(n) + x(-n)}{2}$$

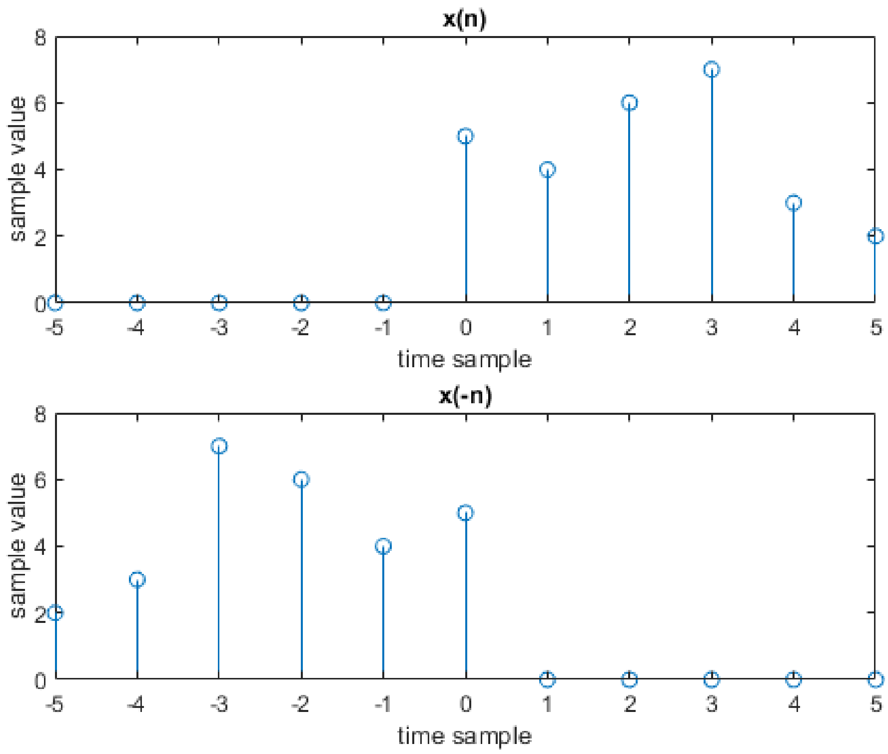
$$xo(n) = \frac{x(n) - x(-n)}{2}$$

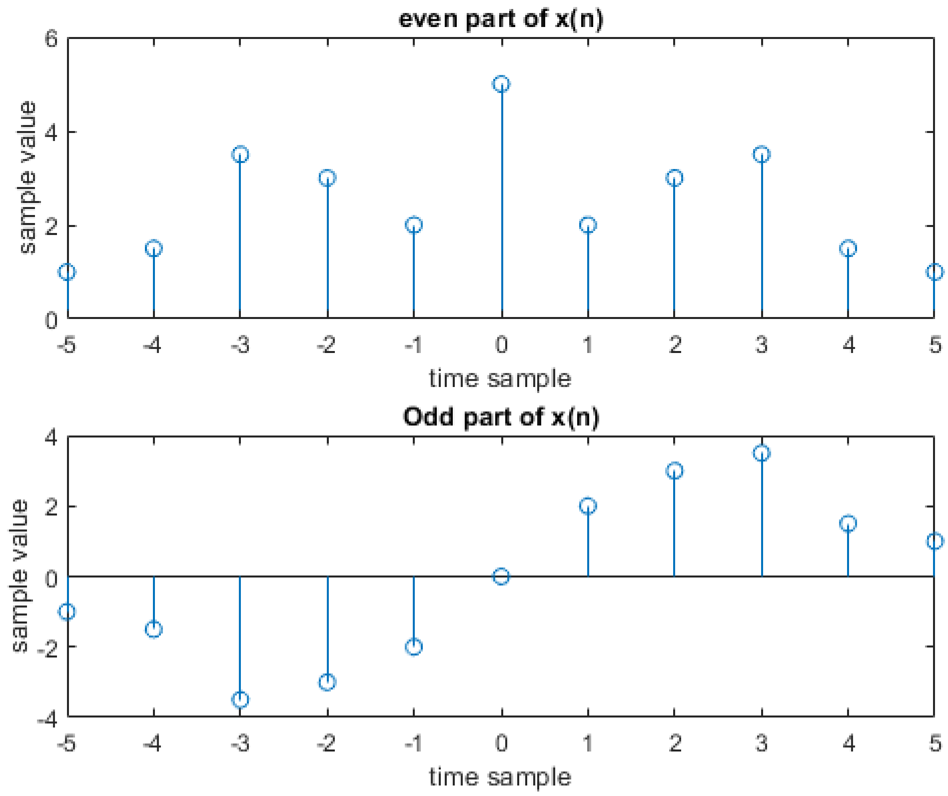
Key Commands:

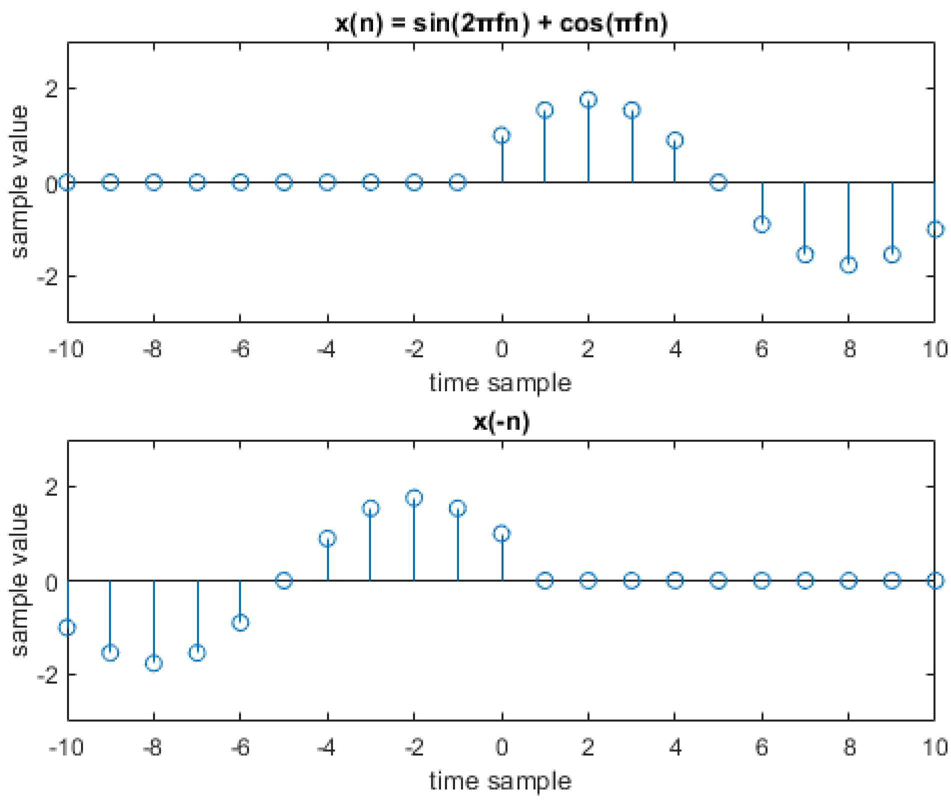
Stem % stem(X , Y) plots the data sequence, Y , at values specified by X .

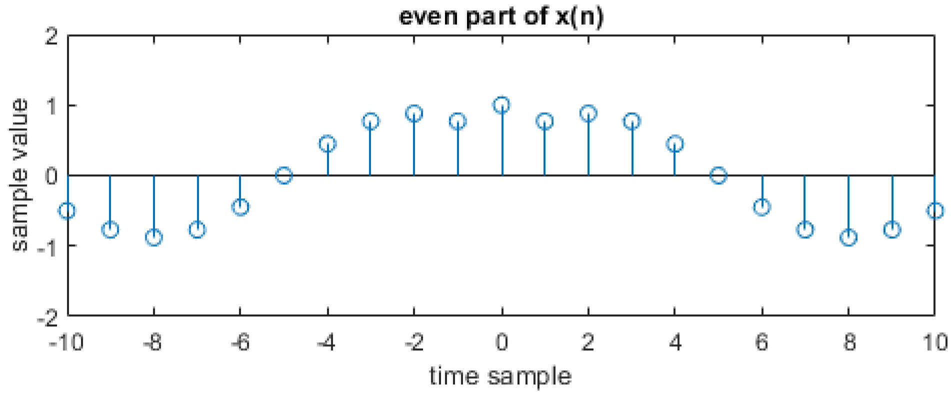
Zeros % zeros(n) returns an n-by-n matrix of zeros.

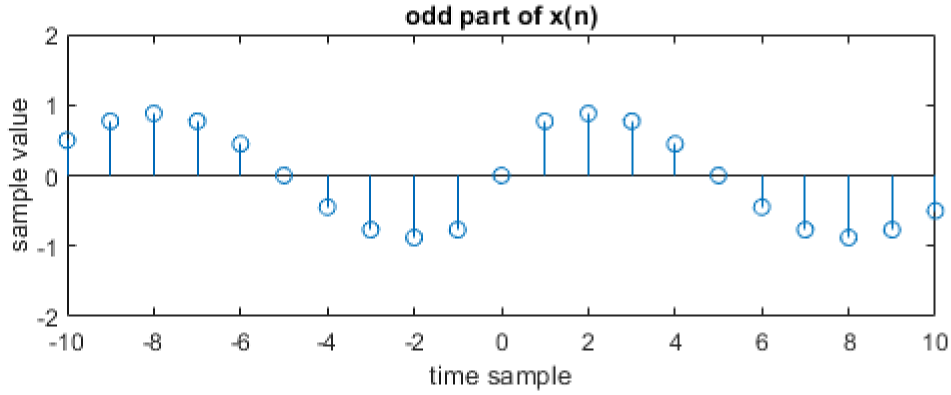
Plots:




even part of x(n)
sample value
time sample
Odd part of x(n)
sample value
time sample


x(n) = sin(2πfn) + cos(πfn)
sample value
time sample
x(-n)
sample value
time sample





Inferences/comments:

1)To find the even part and odd part of the signal we must need the position of zero index of a sequence.

2) even part of the any signal is symmetric around vertical axis.

3) odd part of the signal is symmetric about origin.

Q2. Using MATLAB, state whether the given system is linear and/or time invariant.

Use x1[n] = u[n] − u[n − 10] and/or x2[n] = n $0 \le n \le 10$

=0   Otherwise as an input signals.

(a) y1[n] = x[n − 3] ∗ x[n − 2]
(b) y2[n] = x[n + 2]
(c) y3[n] = sin(x[n])

AIM: To generate and plot the given responses of a signals and check the linearity and time invariance conditions using given input signals.

Short Theory:

Linearity:

A system S is

1.**homogeneity**: if, for any input x(n) and any number a,

$$\boldsymbol{S\{ax(n)\} = aS\{x(n)\}}$$

2.**additive**: if for any two inputs x1(n) and x2(n),

$$\boldsymbol{S\{x1(n) + x1(n)\} = S\{x1(n)\} + S\{x2(n)\}}$$

A system that is both additive and homogeneous is called linear. In other words, S is linear if, for any two inputs x1(n) and x2(n) and any two numbers a1 and a2,

$$\boldsymbol{S\{a1x1(n) + \mathrm{a2}x1(n)\} = \mathrm{a1}S\{x1(n)\} + \mathrm{a2}S\{x2(n)\}}$$

Time Invariant

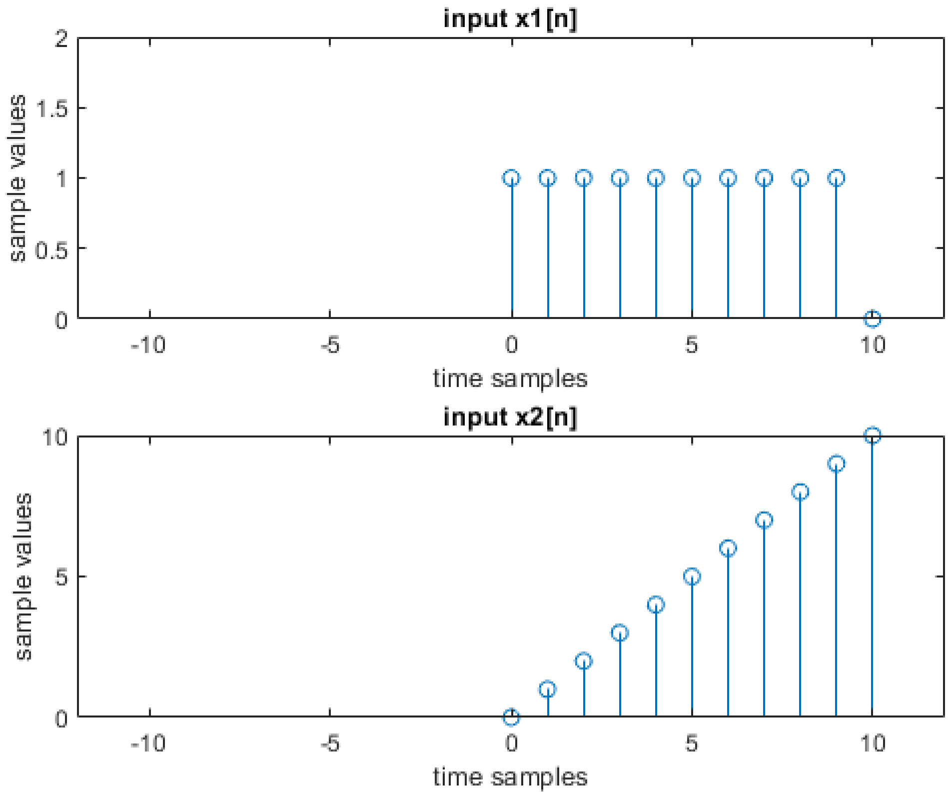
A system S is time-invariant if, for any input x(n) and any fixed time n1, the output $S\{x(n - n1)\}$ is equal to y(n − n1), where y(n) is the output due to x(n), i.e., y(n) = $S\{x(n)\}$.

Key Commands:

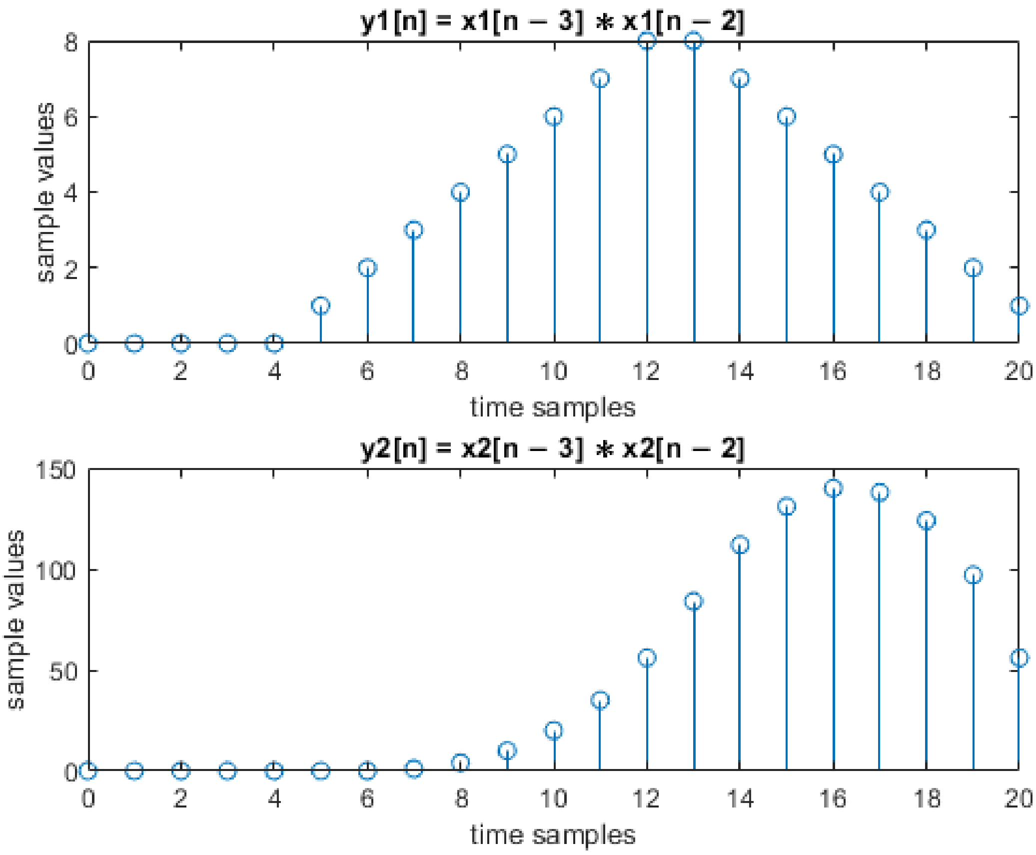
Conv %it convolves the 2 sequences

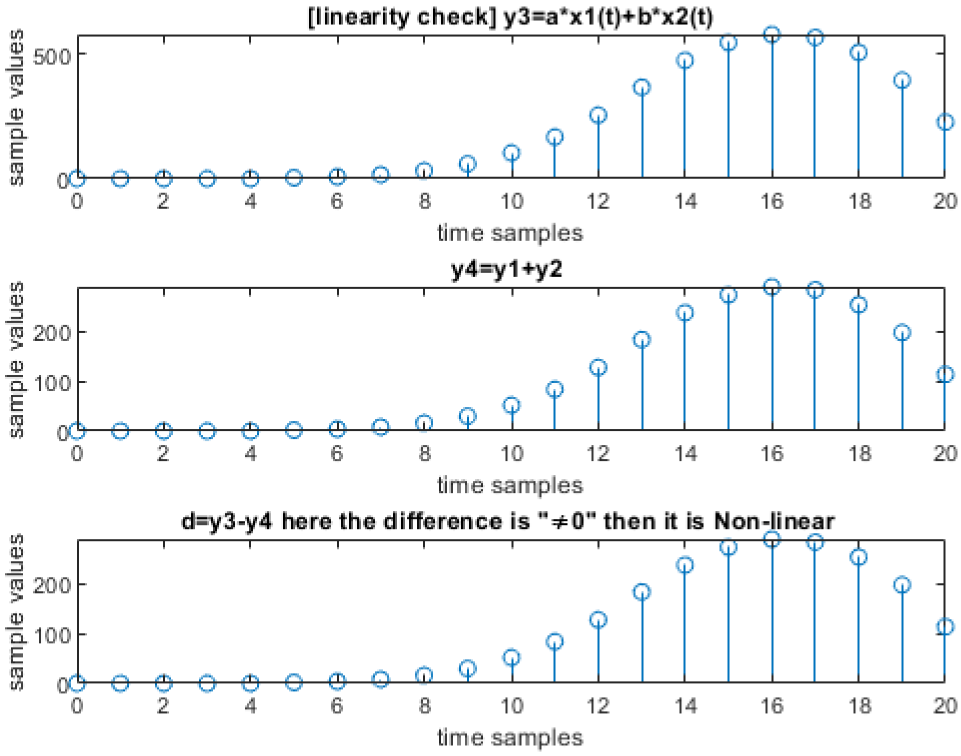
Stem % stem(X , Y) plots the data sequence, Y , at values specified by X .

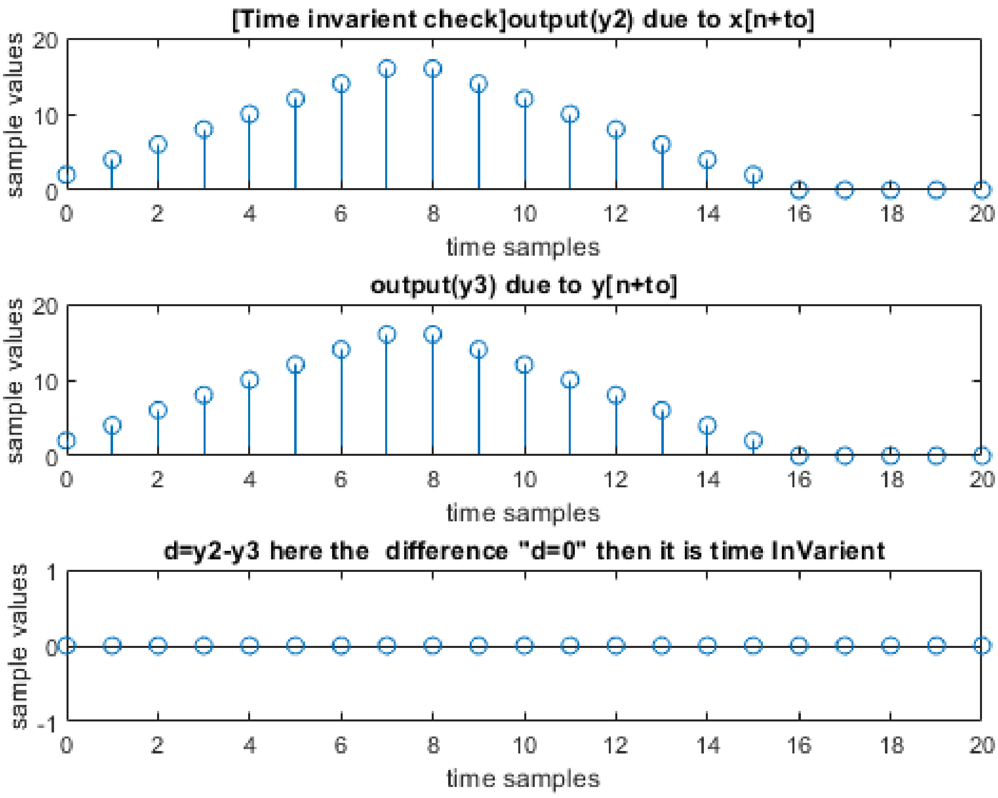
Plots:


input x1[n]
sample values
time samples
input x2[n]
sample values
time samples

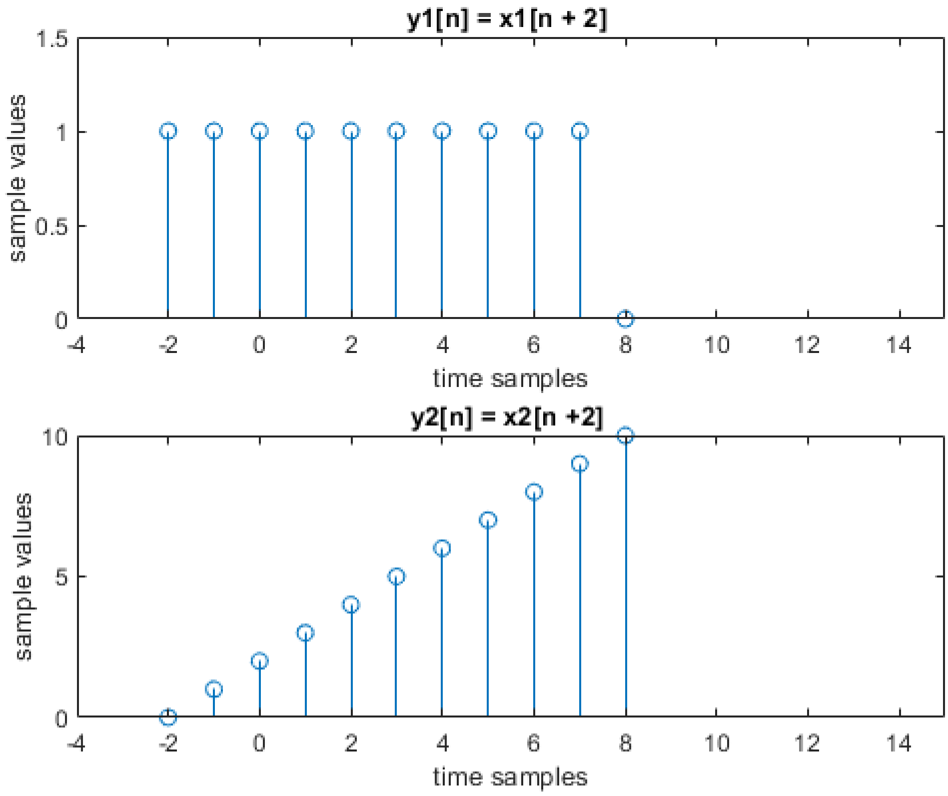
$$y1[n] = x[n-3] * x[n-2]$$

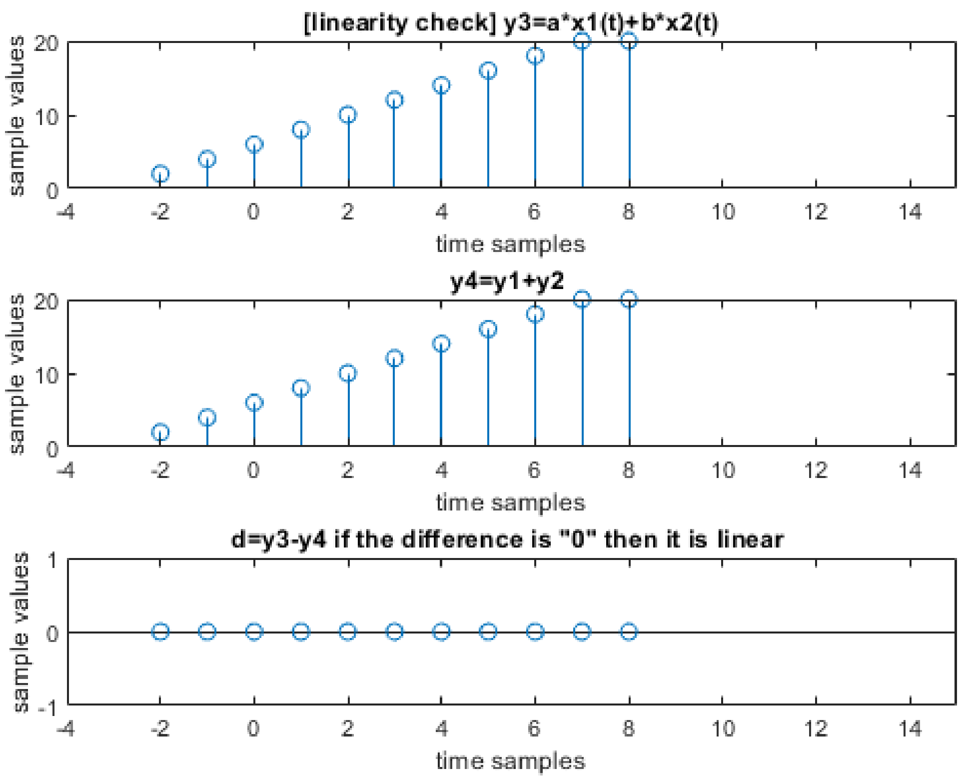

y1[n] = x1[n − 3] ∗ x1[n − 2]
sample values
time samples
y2[n] = x2[n − 3] ∗ x2[n − 2]
sample values
time samples

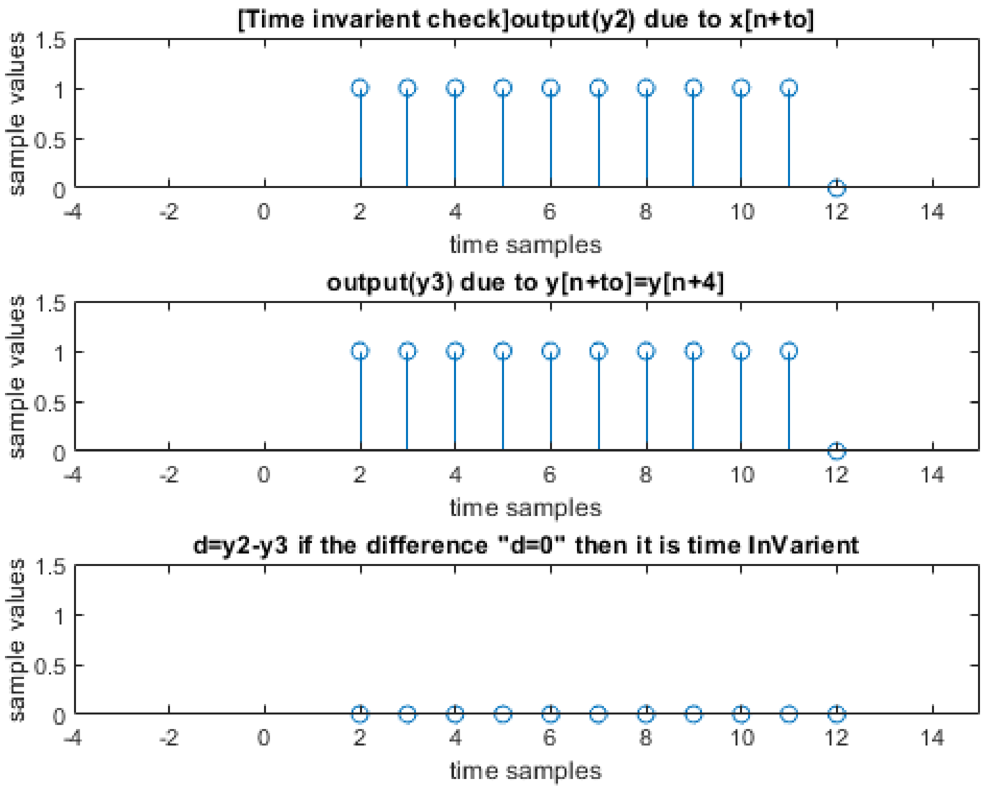

[linearity check] y3=a*x1(t)+b*x2(t)
sample values
time samples
y4=y1+y2
sample values
time samples
d=y3-y4 here the difference is "≠0" then it is Non-linear
sample values
time samples


[Time invarient check]output(y2) due to x[n+to]
sample values
time samples
output(y3) due to y[n+to]
sample values
time samples
d=y2-y3 here the difference "d=0" then it is time InVarient
sample values
time samples

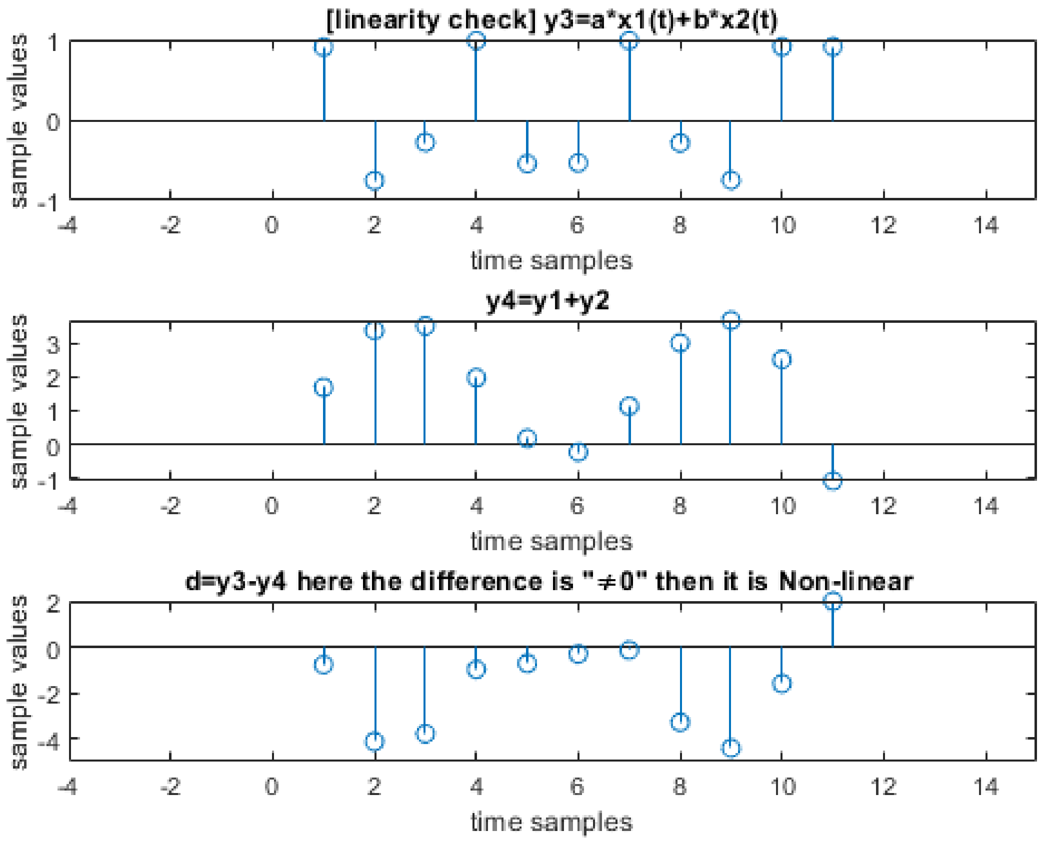
y2[n] = x[n + 2]

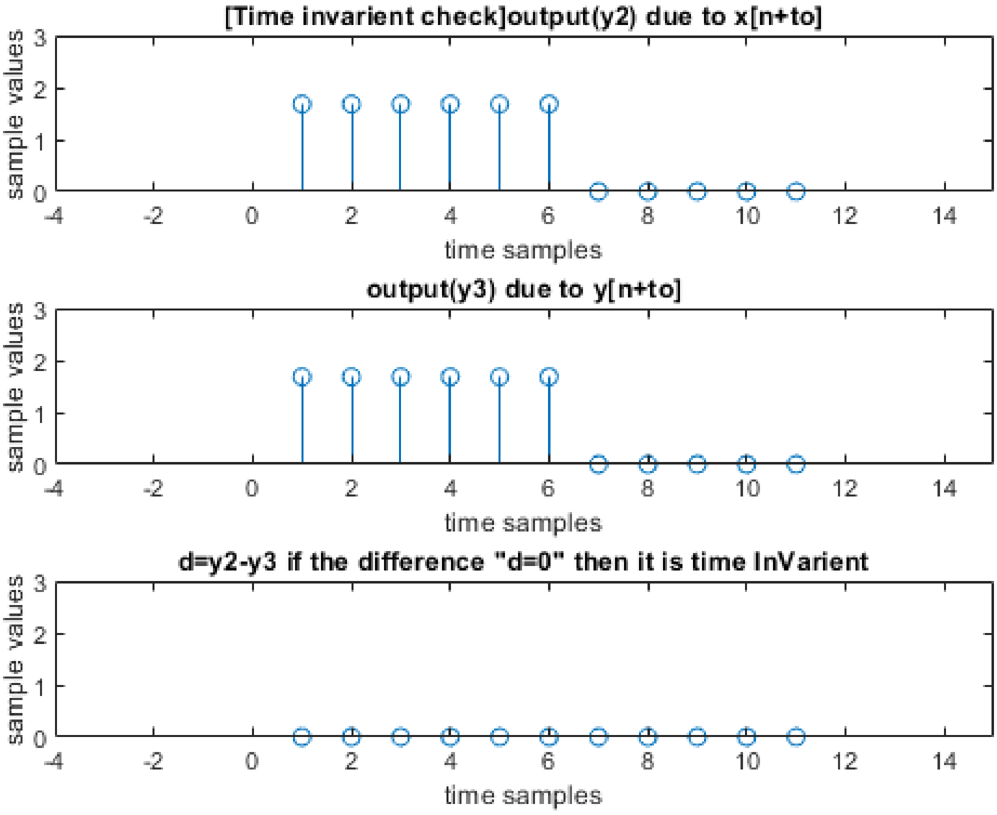

y1[n] = x1[n + 2]
sample values
time samples
y2[n] = x2[n +2]
sample values
time samples


[linearity check] y3=a*x1(t)+b*x2(t)
sample values
time samples
y4=y1+y2
sample values
time samples
d=y3-y4 if the difference is "0" then it is linear
sample values
time samples


[Time invarient check]output(y2) due to x[n+to]
sample values
time samples
output(y3) due to y[n+to]=y[n+4]
sample values
time samples
d=y2-y3 if the difference "d=0" then it is time InVarient
sample values
time samples

y3[n] = sin(x[n])


[linearity check] y3=a*x1(t)+b*x2(t)
sample values
time samples
y4=y1+y2
sample values
time samples
d=y3-y4 here the difference is "≠0" then it is Non-linear
sample values
time samples



Inferences/comments:

1) y1[n] = x[n − 3] ∗ x[n − 2] ,it is a non linear and time invariant system.

   y2[n] = x[n + 2] it is a linear and time invariant system.

   y3[n] = sin(x[n]) it is a non linear and time invariant system.

2) From the above plots ,To check the linearity, we first plotted the linear combination of input and then plotted the linear combination of output and then plotted the difference. The difference between them is zero or equal then it is Linear system.

3. Given the impulse response of the system, plot the impulse response and state whether the system is causal and/or stable.

(a) h1[k] = 1 – k   0 ≤ k ≤ 3
= 0 Otherwise

(b) h2[k] = sin(2πk/50) −10 ≤ k ≤ 30
=0   Otherwise

(c) h3[n] = sin(2πfn)u[n + 2]

AIM: To generate and plot the given signals and check the causality and stability conditions.

Short Theory:

Causal system

A system is said to be causal if its output depends upon present and past inputs, and does not depend upon future input. h[n]=0 for n<0 then it is a causal system.

stability

A discrete system is stable if and only if the its impulse response sequence is h[n] is absolutely summable.

$$S = \sum_{n=-\infty}^{\infty} h[n] < \infty$$

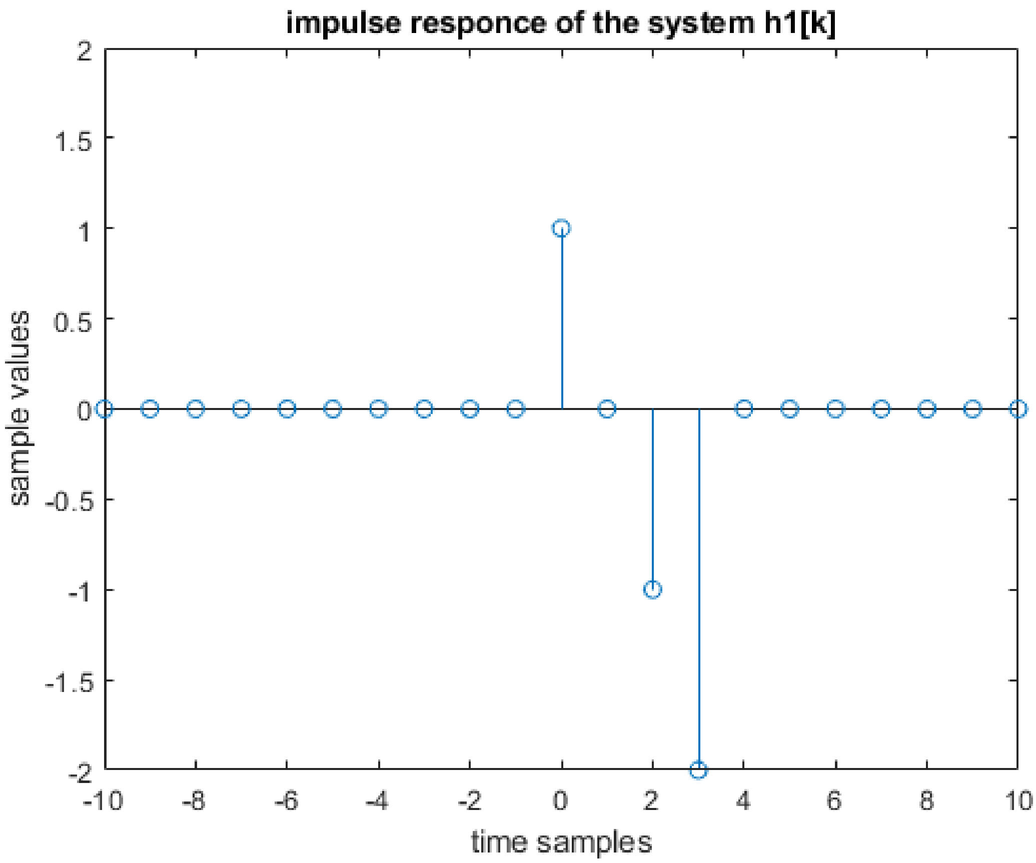
Key Commands:

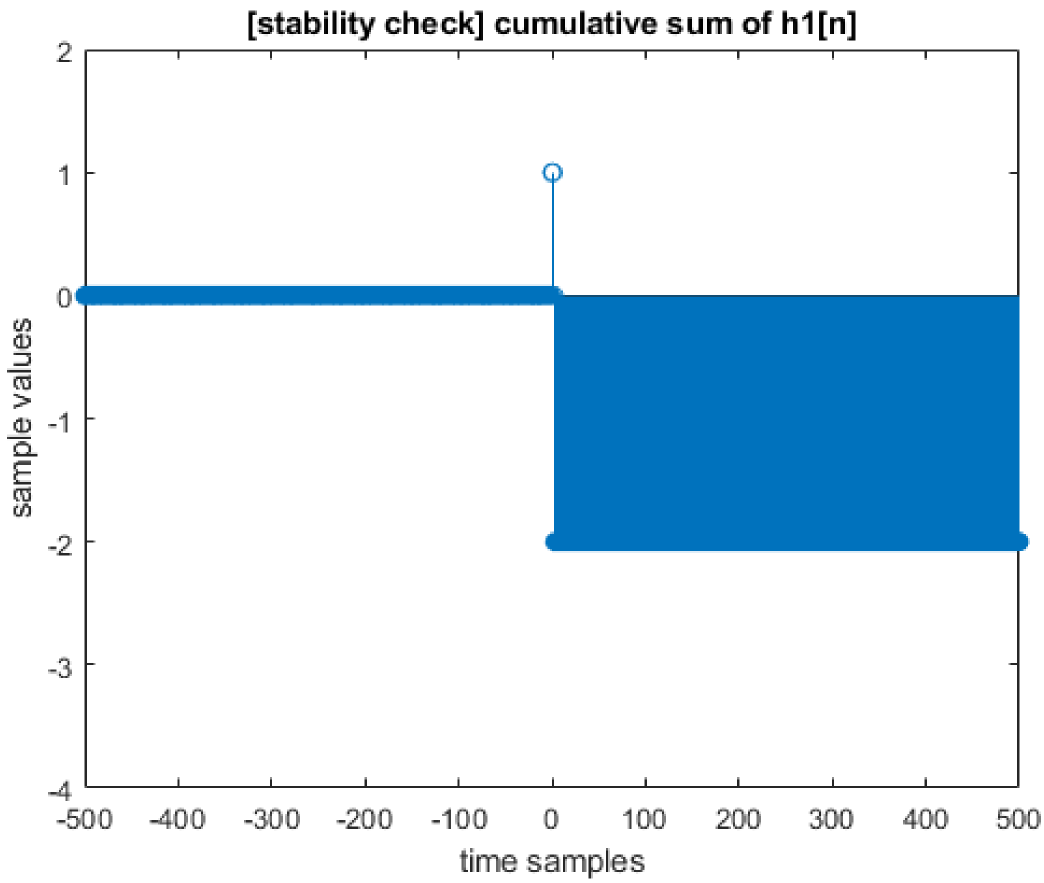
Stem % stem(X , Y) plots the data sequence, Y , at values specified by X .

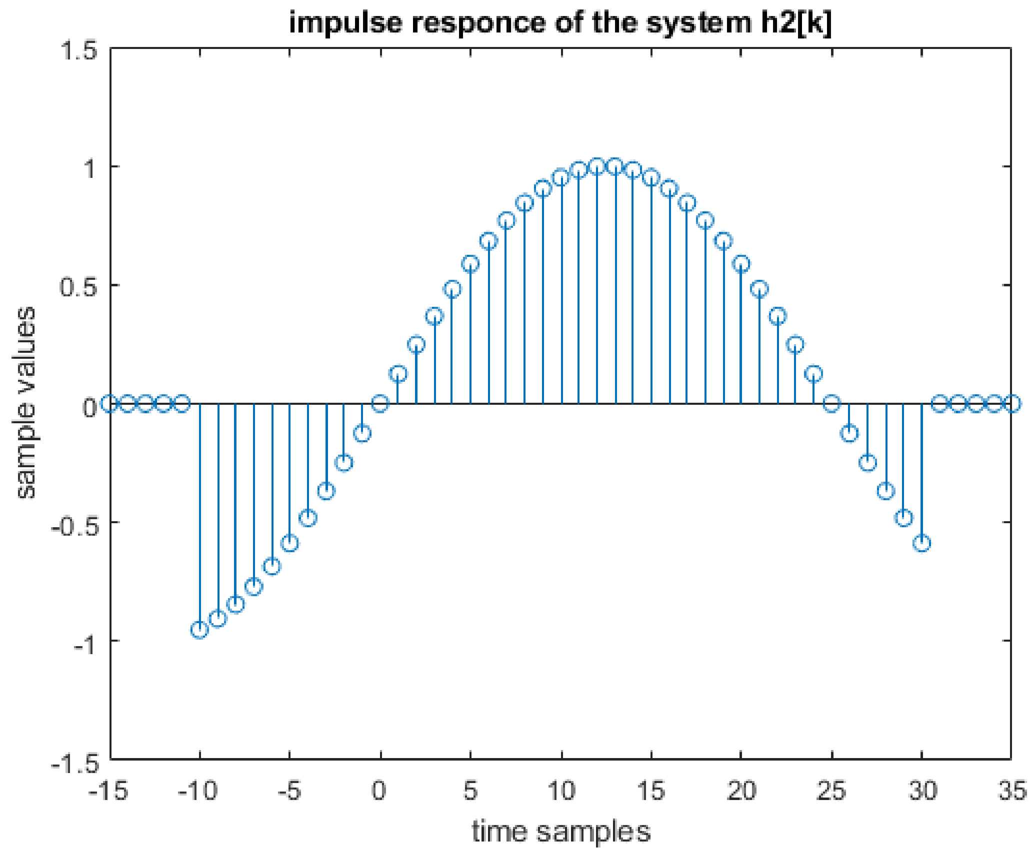
Cumsum %cumsum(A) returns the cumulative sum of A

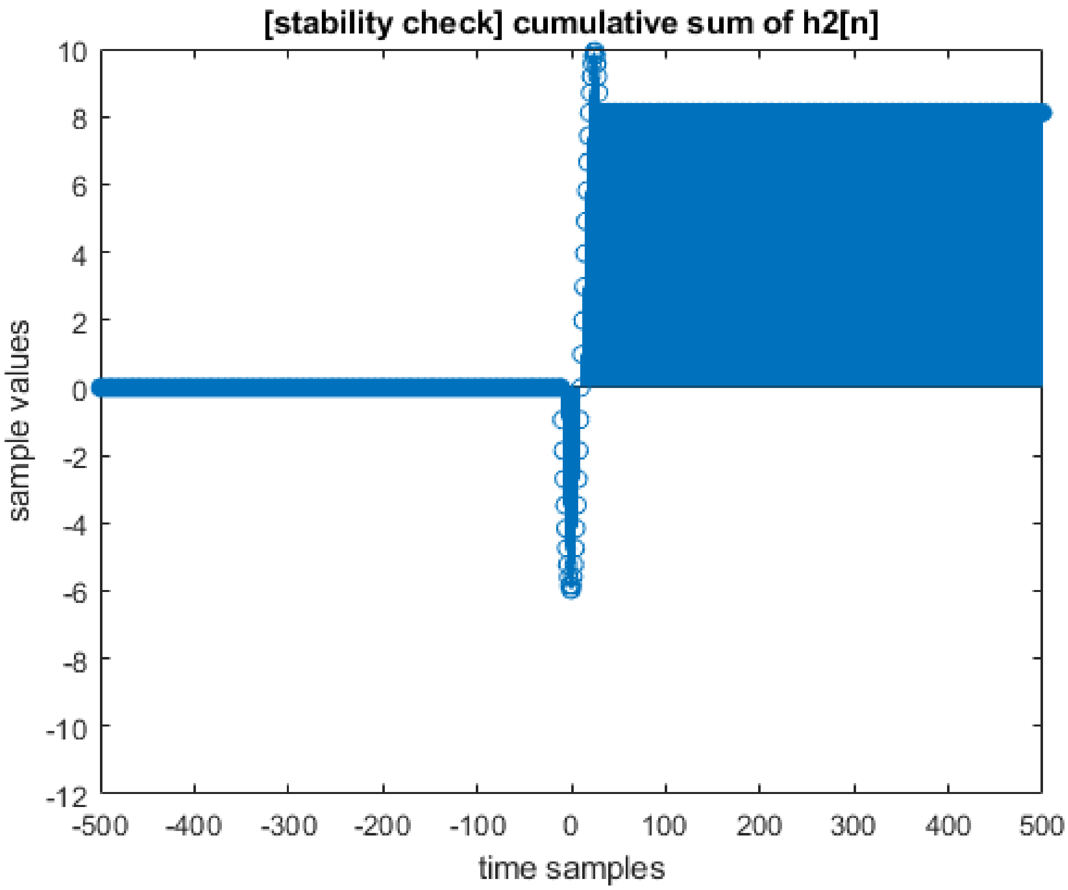
1)sin(x) %this command gives the values of sin(x)

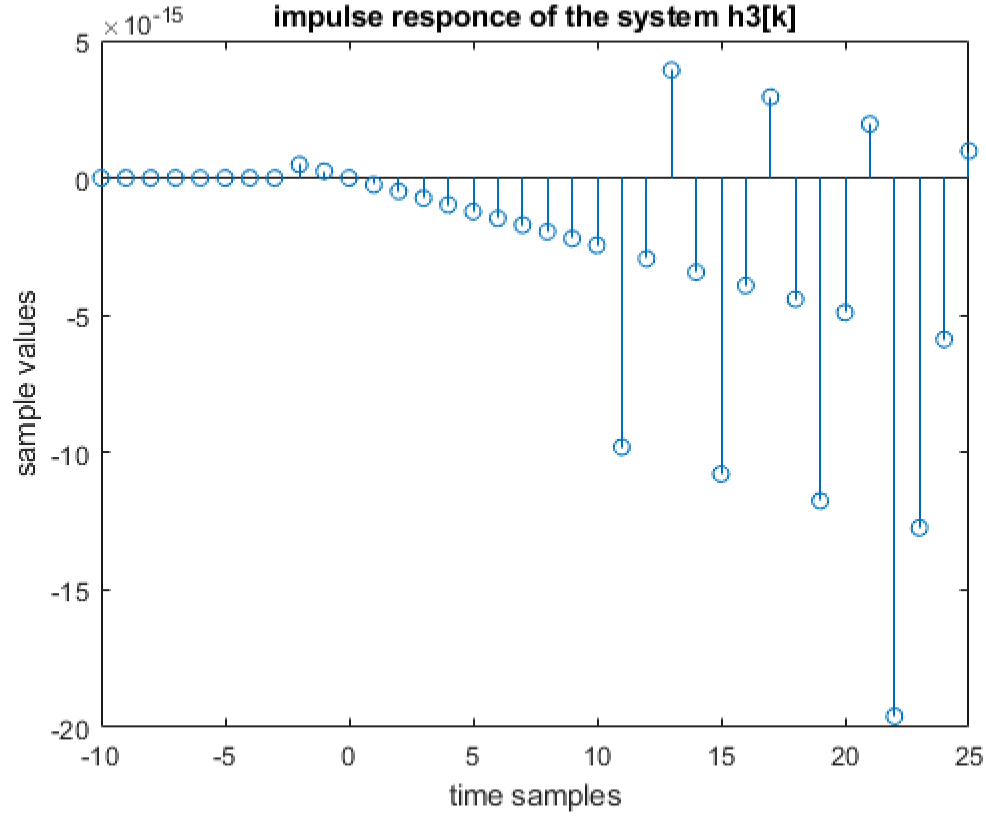
Plots:

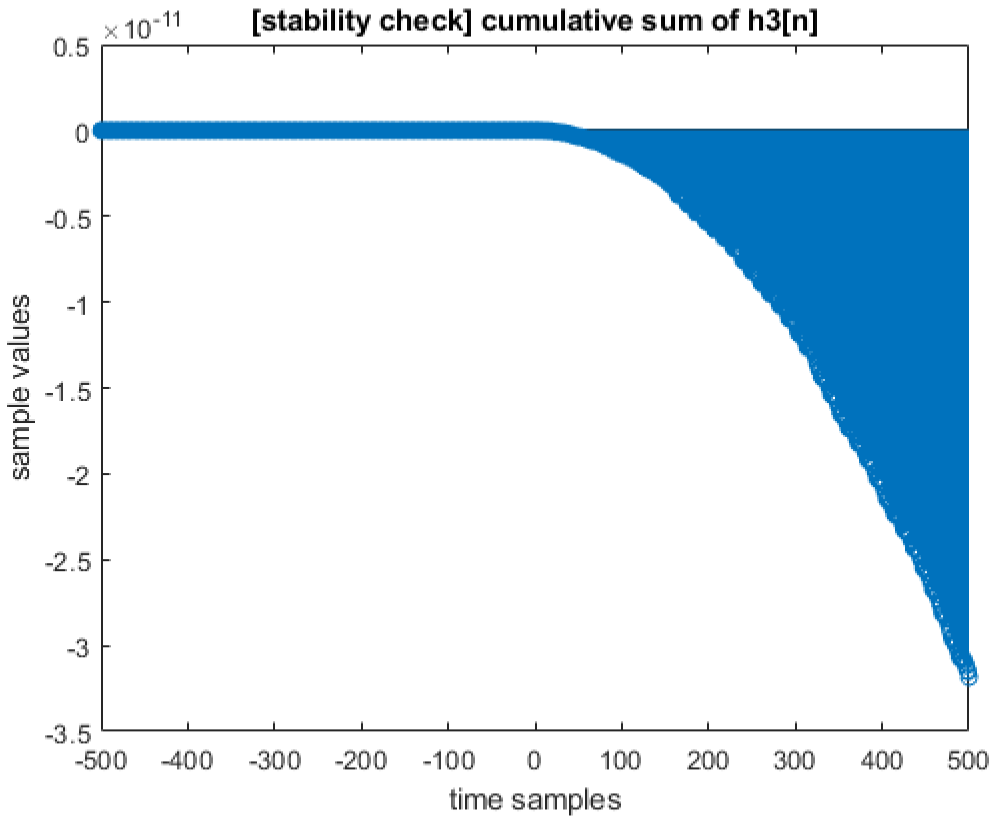

impulse responce of the system h1[k]
sample values
time samples
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-10
-8
-6
-4
-2
0
2
4
6
8
10


[stability check] cumulative sum of h1[n]
sample values
time samples
2
1
0
-1
-2
-3
-4
-500
-400
-300
-200
-100
0
100
200
300
400
500


impulse responce of the system h2[k]
sample values
time samples
1.5
1
0.5
0
-0.5
-1
-1.5
-15
-10
-5
0
5
10
15
20
25
30
35


[stability check] cumulative sum of h2[n]
sample values
time samples
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-500
-400
-300
-200
-100
0
100
200
300
400
500


impulse responce of the system h3[k]
×10^-15
5
0
-5
-10
-15
-20
sample values
-10
-5
0
5
10
15
20
25
time samples


[stability check] cumulative sum of h3[n]
×10^-11
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
sample values
-500
-400
-300
-200
-100
0
100
200
300
400
500
time samples

Desktop ▸ dd

Editor - C:\Users\Rajesh\Desktop\dd\Q3.m

Q3.m

```matlab
for i=-500:500
        if i<0
            pl=pl+1;
          if c(pl)~=0
            display("h1[k]-->it is not causal system")

          break;
          end
        else

             display("h1[k]-->it is causal system")
           break;
        end
```

Command Window

```
    "h1[k]-->it is causal system"

    "h2[k]-->it is not causal system"

    "h3[k]-->it is not causal system"

fx >>
```

UTF-8 | script | Ln 8 Col 1

Inferences/comments:

1) h1[k] is a causal system and stable.

   h2[k] is a non causal and stable system.

   h3[k] is a non causal and unstable system.

2) To find the stability of a system , I plotted the cumulative sum of the h[k] .we can observe that the cumulative plots of h1[k] & h2[k] reaches to constant value after k reaches to large value. So those are stable systems. In case of h3[k] system, the cumulative plot growing exponentially with k value. So this system is unstable.

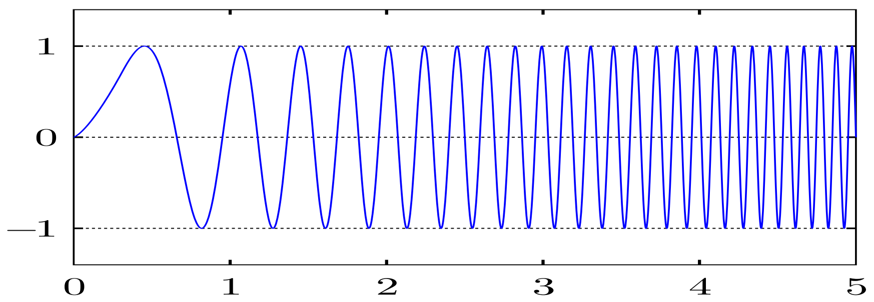
Experimental Exercises:

Q1. Generate and play the following basic signals:
(a) sinusoidal signal of length 2 seconds with frequency 500Hz, sampled at 22100Hz.
(b) square wave of length 2 seconds with appropriate frequency sampled at 22100Hz
(c) chirp signal by using MATLAB chirp function
(d) dual tone signal by adding two sinusoidal signals of length 0.5 seconds and different frequencies.

AIM: To generate and play the basic signals sinusoidal, square of length 2sec with a frequency f=500 Hz, fs=22100 Hz and chirp signal and dual tone signal of 0.5 sec with different frequencies.

Short Theory:

A chirp is a signal in which the frequency increases (up-chirp) or decreases (down-chirp) with time.



Dual tone signal can be create by adding to signals with different frequencies.

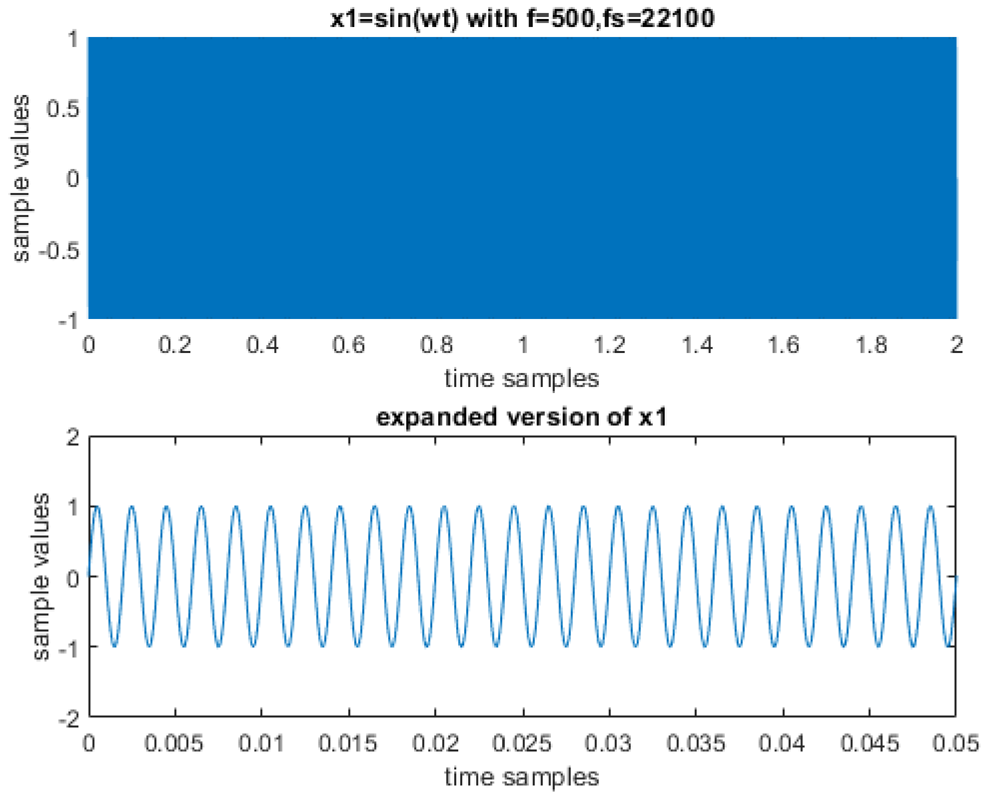
Key Commands:

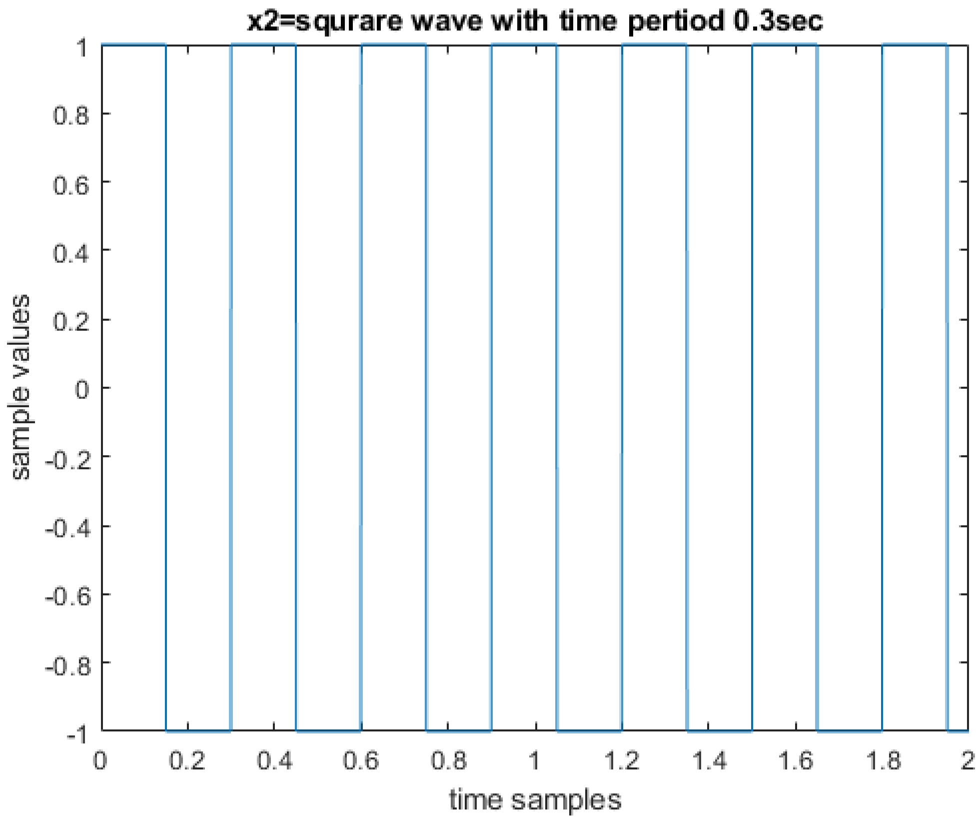
sin % it gives the values of sin(x)

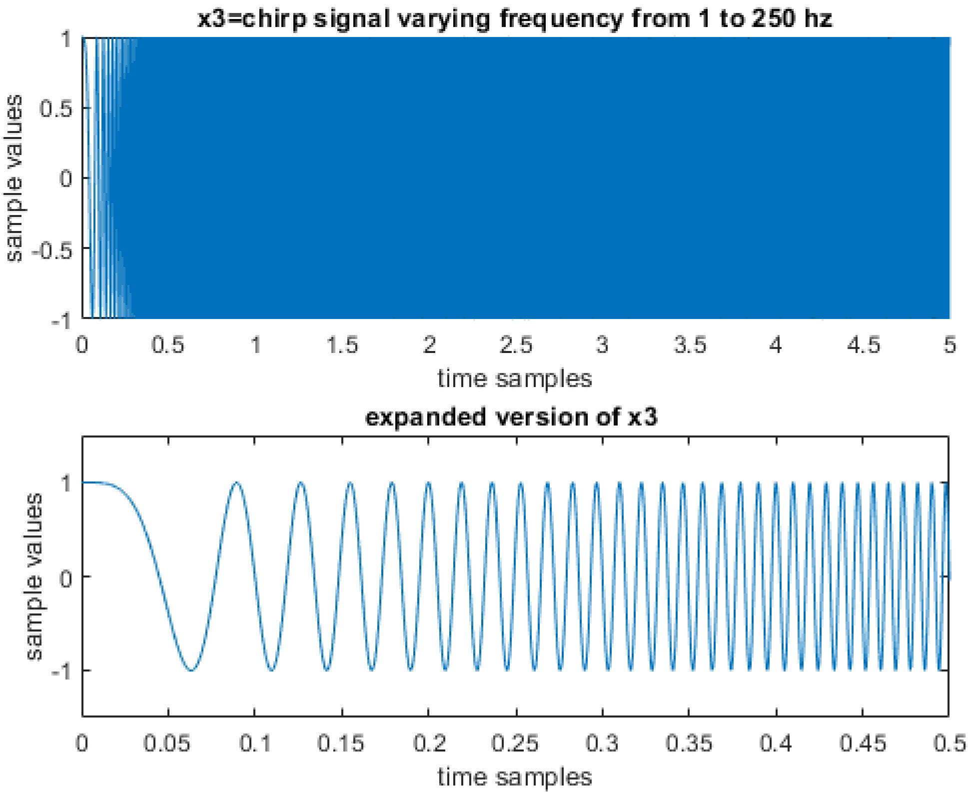
Chirp % chirp(t,f0,t1,f1) generates samples of a linear swept-frequency cosine signal at the time instances defined in array t. The instantaneous frequency at time 0 is f0 and the instantaneous frequency at time t1 is f1.

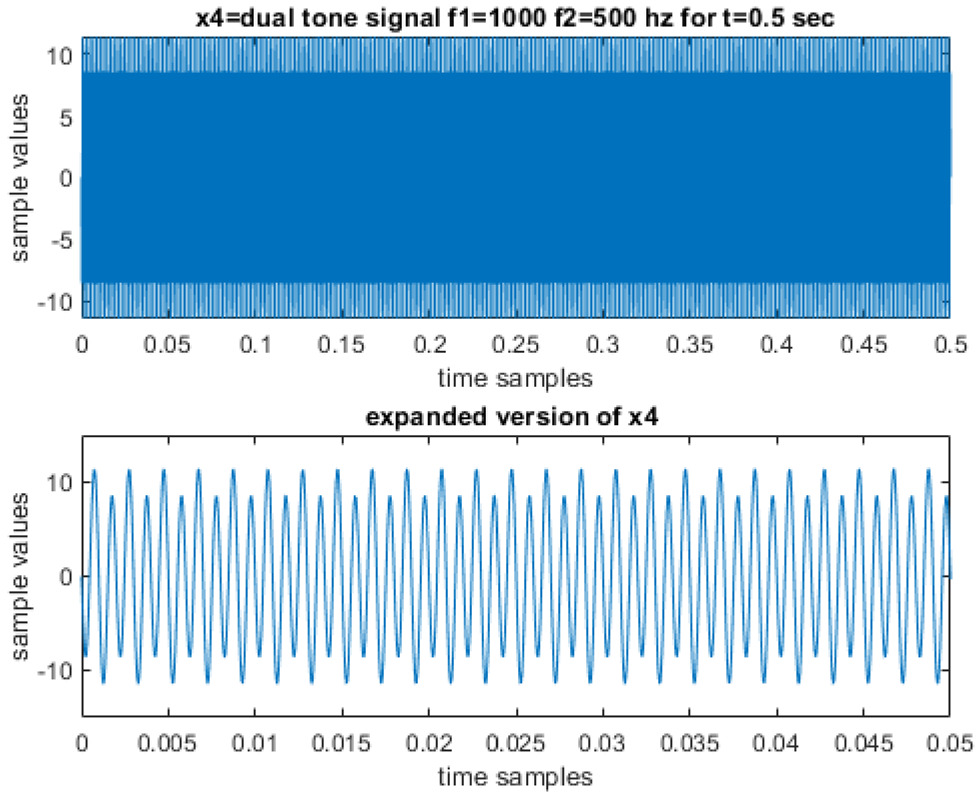
Square % square(t) generates a square wave with period 2π for the elements of the time array t.

Plots:


x1=sin(wt) with f=500,fs=22100
sample values
time samples
expanded version of x1
sample values
time samples


x2=squrare wave with time pertiod 0.3sec
sample values
time samples


x3=chirp signal varying frequency from 1 to 250 hz
sample values
time samples
expanded version of x3
sample values
time samples


x4=dual tone signal f1=1000 f2=500 hz for t=0.5 sec
sample values
time samples
expanded version of x4
sample values
time samples

Inferences/comments:

1) Sinewave **:** the sound produced by a sine wave is continuous .
2) Square wave **:** The produced by square wave is discontinuous .
3) Chirp signal **:**The frequency is gradually increases ,then the sound intensity also increases. This kind of sound can be use in alarms.
4) We can observe that if we generate any of the above signals by give the frequency bellow hearing range of humans (f<20hz) we can not clearly feel those audio signals.

1) sampling frequency (**fs**) variations.

Fs_1 < fs

If we listen the signal with Less than the sampling frequency ,it feels like fast forward the audio, I,e we can hear the complete audio in less time. It compress the whole whole audio with respect to time. Here we are ignoring the some of the samples in original audio signal.

Fs_2 > fs

If we listen the signal with more than sampling frequency, it feel like a audio playing in slow motion , I,e. we can hear the complete audio in more time. It expands the whole audio with respect to time. here we adding more samples to the original audio.

Q2. Use audiorecorder and audioplayer functions of MATLAB to record and play the audio activity of your surroundings. Take alternate samples out of your signals and play with same sampling frequency. Write, in brief, your observations.

AIM: To record and play the audio of surrounding by using audiorecorder and audioplayer functions of MATLAB.

Short Theory:

In this experiment we record the sound using microphone with a some sampling frequency and we need to take out the alternative samples and play those samples with the same recorded sampling frequency.

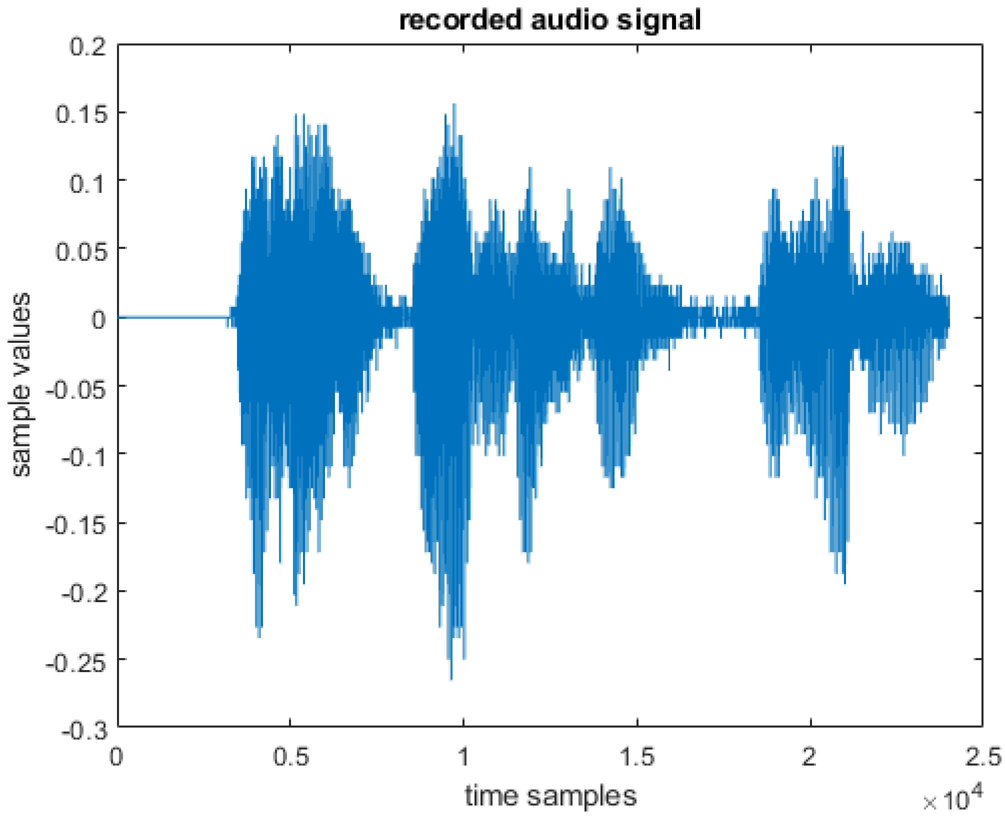
Key Commands:

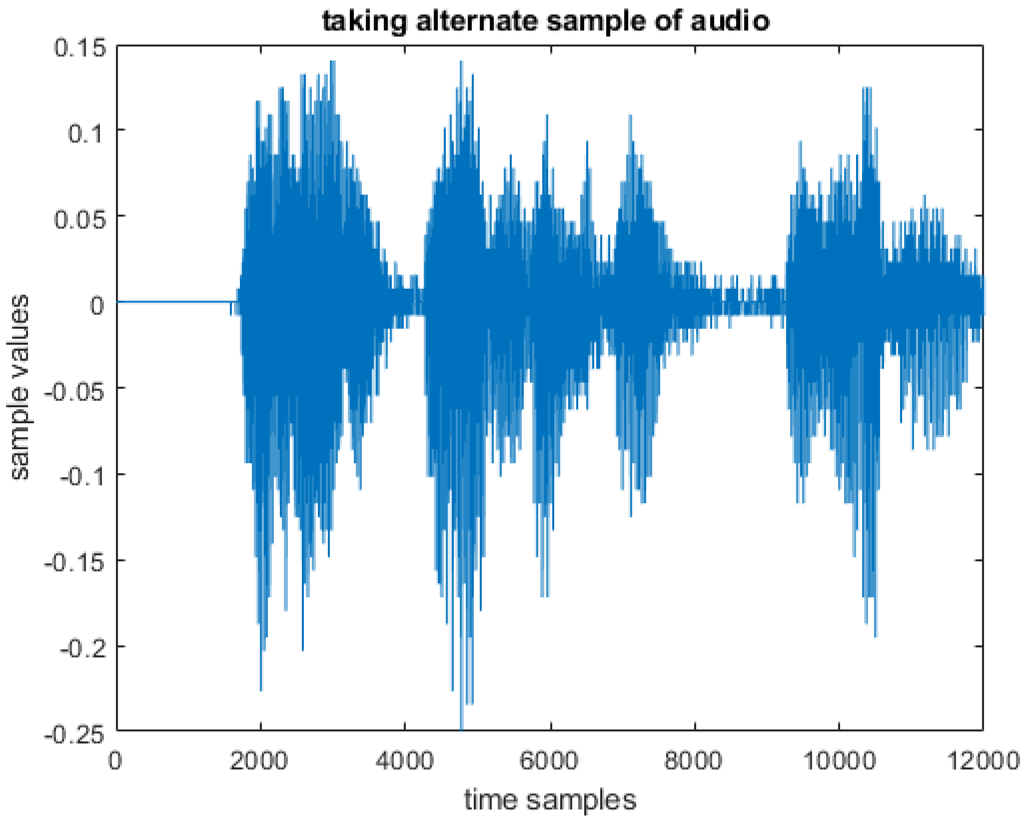
Sound % sound(y,Fs) sends audio signal y to the speaker at sample rate Fs.

Audiorecorder %audiorecorder(Fs,nBits,NumChannels) sets the sample rate Fs (in hertz), the bits per sample nBits, and the number of channels nChannels.

Audioplayer % audioplayer(Y,Fs) creates an audioplayer object for signal Y, using sample rate Fs.

Plots:





Inferences/comments:

1). Let say we record the audio with fs sampling frequency (Fs) → sound(y,Fs)

Then if you listen the audio by collecting all the alternating samples with a sampling frequency Fs/2, still we can clearly observe the sound. → sound(x1,fs/2)

if you listen the audio of alternating samples with a sampling frequency Fs then we can not hear the original audio properly.it feel like a fast forward audio.→ sound(x1,fs)

2) The difference between play() and sound() commands in MATLAB. Both the commands used to play the audio signals but we can not have the full control over the audio using sound() command. But we have the flexibility to play the sound [like play, pause, continue, stop] using play command.