**NATIONAL INSTITUTE OF TECHNOLOGY CALICUT**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

****

**EC 4091: DIGITAL SIGNAL PROCESSING LABORATORY**

ADDITIONAL EXPERIMENT:CYCLE 1

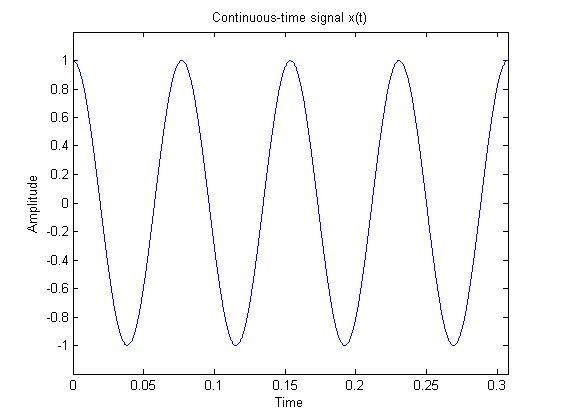
**Submitted by:**

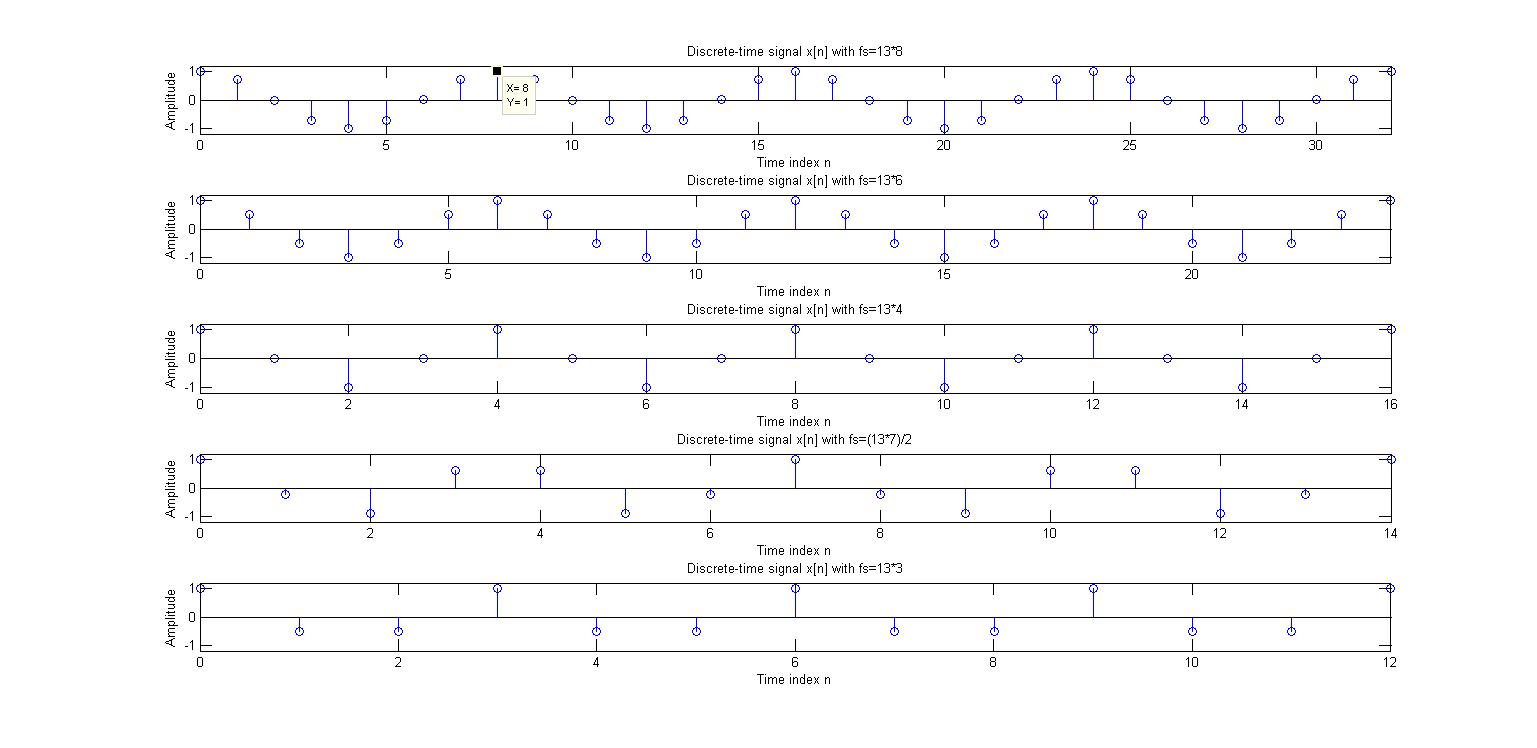
B.Veda Sree - B150734EC

B.Sravani - B150783EC

Ch.Priyanka - B150837EC

Ch.Vamsi Narayana Reddy - B150530EC





1. Sampling: Program below illustrates the sampling of a continuous time sinusoid of frequency 13 Hz. Since Matlab cannot strictly generate a continuous time signal, we simulate a continuous time signal by sampling it at a very high rate Th=0.0001 sec. A plot of the samples using plot command will then look like a continuous time signal. Since the frequency of the continuous time signal is 13Hz, the Nyquist rate is T=1/26. Run the program for several values of T, below and above the Nyquist rate such as 1/(8\*13), 1/(6\*13), 1/(4\*13), 2/(7\*13), 1/(3\*13), 2/(5\*13), 4/(9\*13), 1/(2\*13), 2/(3\*13), 4/(5\*13), 1/13, etc. . (all rational multiples of 1/13 so that the resulting discrete time signal is periodic) Which of the discrete time signals will give correct reconstruction of the continuous time signal? Mathematically determine the angular frequency in rad/sample and the fundamental period N of each of the discrete time signals and verify from the plots.

**Matlab Code:**

% Illustration of the Sampling Process

% in the Time-Domain

F = 13; %frequency=13 Hz

tmax=4/13; %display four cycles

t = 0:0.0001:tmax;%Th=0.0001

xa = cos(2\*pi\*F\*t);

subplot(211)

plot(t,xa);

xlabel('Time');ylabel('Amplitude');

title('Continuous-time signal x(t)');

axis([0 tmax -1.2 1.2])

T=input('Enter the sampling period T');

nmax=tmax/T;n = 0:nmax;

xs = cos(2\*pi\*F\*n\*T);

subplot(212); stem(n,xs);

xlabel('Time index n');ylabel('Amplitude');

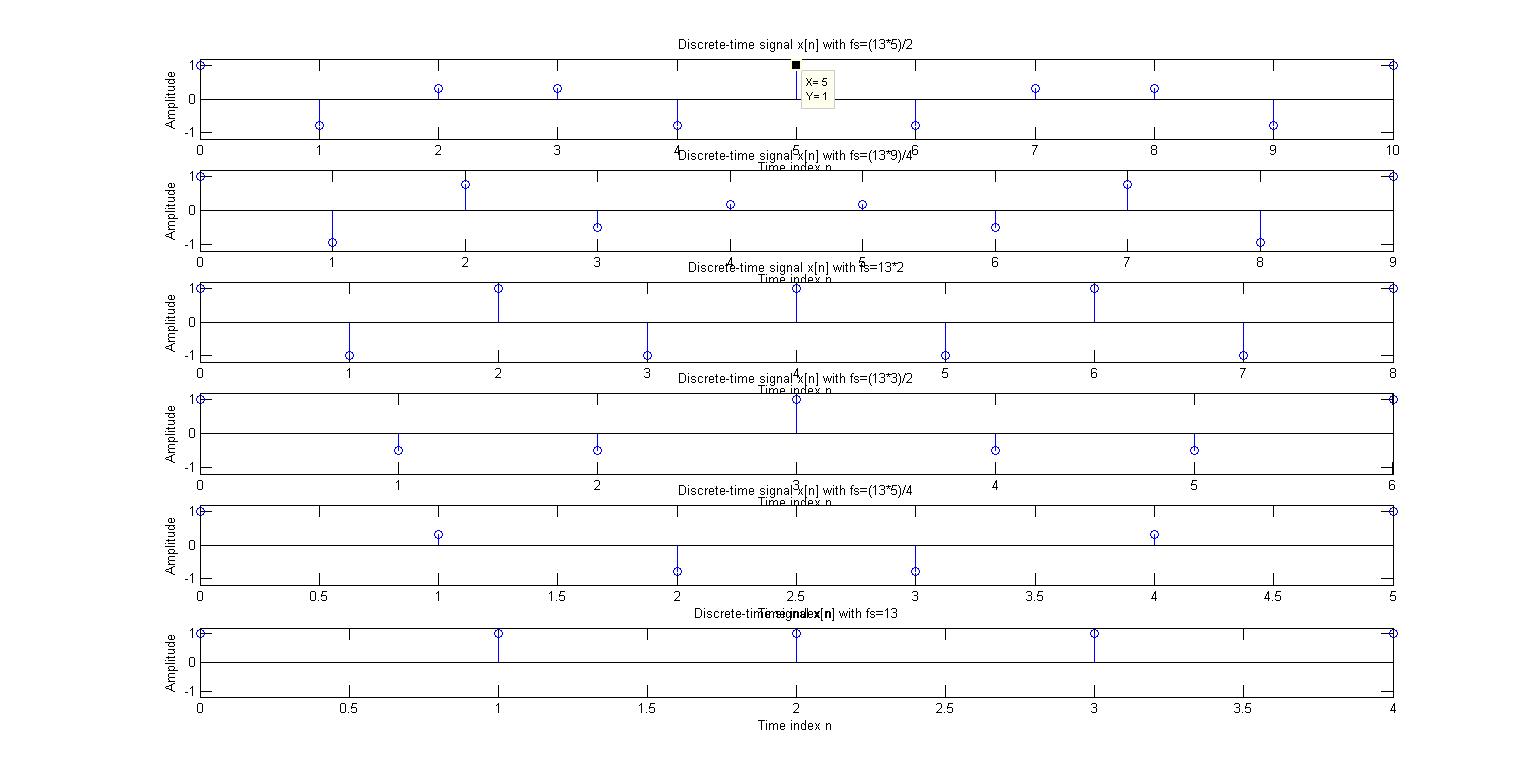
title('Discrete-time signal x[n]');

axis([0 nmax -1.2 1.2])

* Sampling Theorem states that a continuous time signal can be recovered back/reconstructed when sampling frequency fs is greater than or equal to twice the highest frequency component of message signal.

i.e., Nyquist frequency of sampling is (2\*13)Hz, f= 26Hz

The Nyquist rate here is T=1/26.



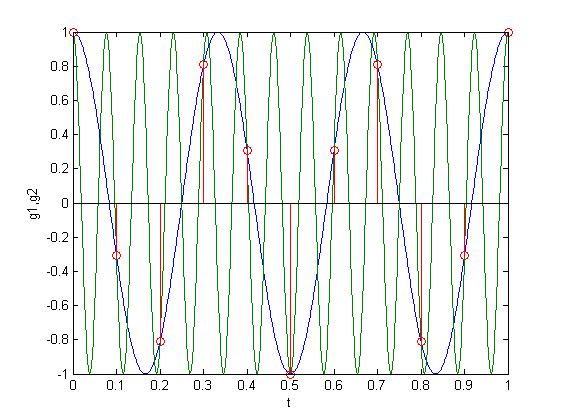
**Inferences:**

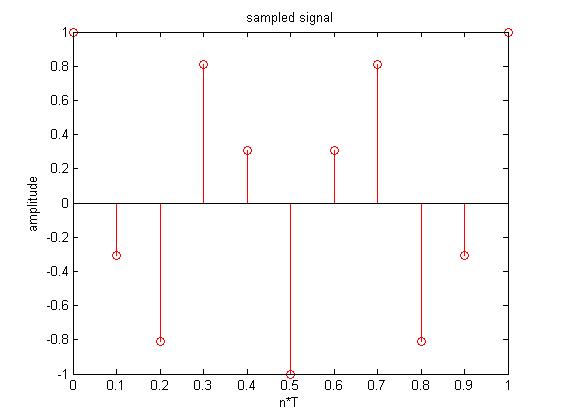
* All those discrete signals obtained by sampling the given sinusoid with sampling frequency greater than or equal to the nyquist frequency can be used to reconstruct the original sinusoid.

The discrete-time signals when sampled with Ts= 1/(8\*13), 1/(6\*13), 1/(4\*13), 2/(7\*13), 1/(3\*13), 2/(5\*13), 4/(9\*13), 1/(2\*13); have given correct reconstruction of the given sinusoid as the sampling frequency is greater than nyquist rate.

* Discrete-time angular frequency (Ω) = 2(radians/sample)
* N(samples) – Fundamental Period is the smallest positive integer for which the signal repeats.N=2π/ Ω

|  |  |  |
| --- | --- | --- |
| Ts(Sampling Time) | Ω (Angular Frequency) | N (Fundamental Period) |
| 1/(8\*13) | /4 | 8 |
| 1/(6\*13) | /3 | 6 |
| 1/(4\*13) | /2 | 4 |
| 2/(7\*13) | /7 | 7 |
| 1/(3\*13) | 2/3 | 3 |
| 2/(5\*13) | 25 | 5 |
| 4/(9\*13) | 29 | 9 |
| 1/(2\*13) |  | 2 |
| 2/(3\*13) | /3 | 3 |
| 4/(5\*13) | 2/5 | 5 |
| 1/13 |  | 1 |





1. The family of continuous time sinusoids cos(Ω0+kΩs)t , k=0,±1, ±2..... where Ωs=2π/T leads to identical sampled sequences.( Prove) This phenomenon of a continuous time sinusoid of a higher frequency acquiring the identity of a sinusoidal sequence of lower frequency after sampling is called aliasing. Consider the continuous time sinusoid g1(t)=cos(6πt). When sampled with T=0.1 sec, it will lead to cos[.6πn]. When sampled with T=0.1 sec, g2(t)=cos(26πt) will also lead to cos[.6πn]. (Verify mathematically). Run the program below to see aliasing in this case.

**Matlab Code:**

Th=.001;

tmax=1;

t=0:Th:tmax;

g1=cos(6\*pi\*t);

g2=cos(26\*pi\*t);

plot(t,[g1;g2]);hold on;

T=.1; %sampling period

nmax=tmax/T; n=0:nmax;

gn=cos(.6\*pi\*n);

stem(n\*T,gn,'r')%time axis denormalised by using n\*T so that the

%samples can be superimposed

figure

gn=cos(.6\*pi\*n);

stem(n\*T,gn,'r')%time axis denormalised by using n\*T so that the

%samples can be superimposed

**Inferences:**

* Let x(t)= cos(Ω0+kΩs)t , k=0,±1, ±2.....where Ωs=2π/T

x(t)= cos(Ω0t+k t2π/T) = cos(2π(fs\*k\*t)+ Ω0t) = cos(Ω0t)

This implies that the family of continuous time sinusoids cos(Ω0+kΩs)t , k=0,±1, ±2.....where Ωs=2π/T leads to identical sampled sequences.

* When g1(t)=cos(6πt) is sampled with T=0.1 sec, i.e., t=nT this implies, discretized signal g1[n] = cos(6πn\*0.1) = cos(0.6πn);

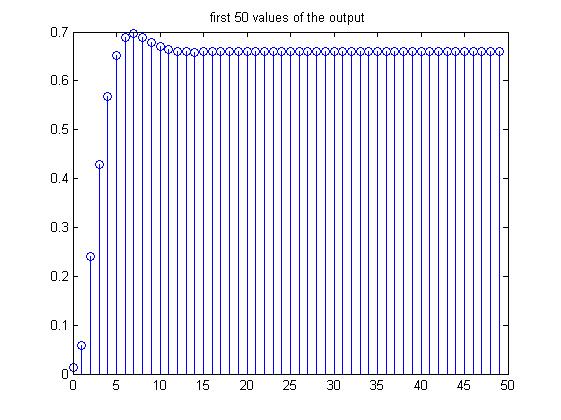
Nyquist frequency for g1(t) [cos(2π\*3\*t)] is 6Hz, i.e., Fs=10Hz >nyquist frequency. So, there will be no aliasing and the original signal can be reconstructed from the obtained discretized signal.

* When g2(t)=cos(26πt) is sampled with T=0.1 sec, i.e., t=nT this implies, discretized signal is g2[n] = cos(26πn\*0.1) = cos(2.6πn) = cos(2π+0.6πn) = cos(0.6πn) (cosine is periodic with the period 2πn).

Nyquist frequency for g1(t) [cos(2π\*13\*t)] is 26Hz, i.e., Fs=10Hz <nyquist frequency. So, there will be aliasing effect and the original signal cannot be retrieved from the obtained discretized signal.

* The sampled values of cos(26πt) are same as that of cos(6πt), when these signals are reconstructed back cos(26πt) will be wrongly retrieved.

(a)



1. The filter() function : The filter() command recursively computes the the output y(n) of an LTI system described by a difference eqn from the input x(n) and initial conditions. b=[b0,b1,.....bM ]; a=[ a0,a1,......aN]; y=filter(b,a,x) . The number of output values in y correspond to the number of input values in x. See help on filter for more details.
2. Run the code fragment below to determine the first 50 values of the output of the system described by y(n)-1.143y(n-1)+.4128y(n-2)=.0675x(n)+.1349x(n-1)+.675x(n-2) if the initial conditions are zero and x(n)=.2u(n).

**Matlab Code:**

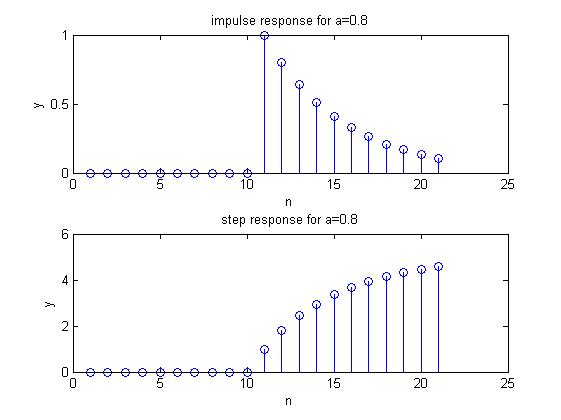
a=[1 -1.143 .4128 ]; b=[.0675 .149 .675];

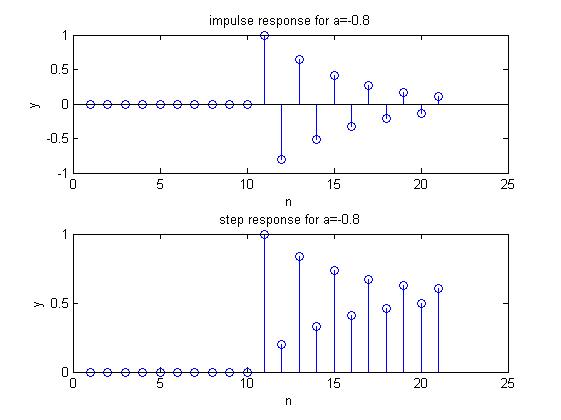
y=filter(b,a,.2\*ones(1,50));

figure

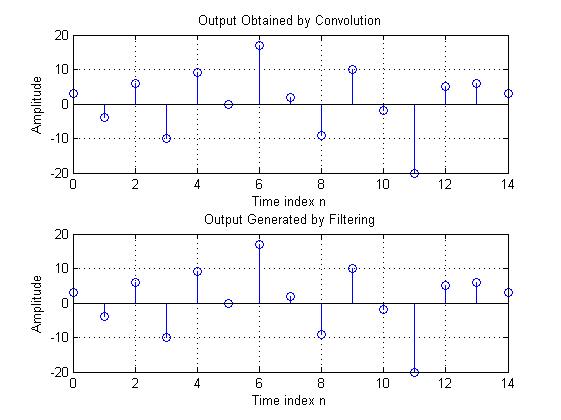
stem(0:49,y)

(b)

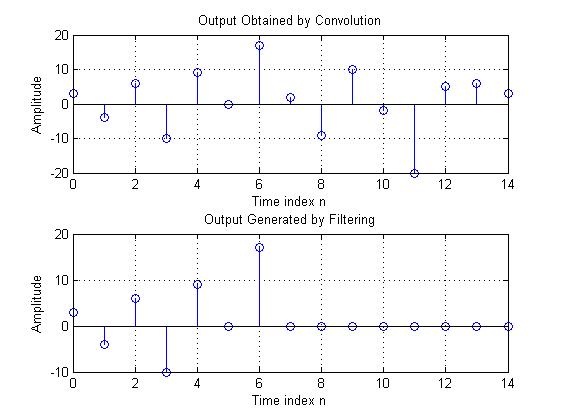




1. Using filter(), determine and stem the first 41 samples of the impulse and step response of the system described by y(n)-ay(n-1)=x(n) for a=.8 and -.8. Verify that the step response is the running sum of the impulse response



**Without zero padding**



c. Run the following program to generate output using both conv() and filter():

h = [3 2 1 -2 1 0 -4 0 3]; % impulse response

x = [1 -2 3 -4 3 2 1]; % input sequence

y = conv(h,x);

n = 0:14;

subplot(2,1,1);stem(n,y);

xlabel('Time index n'); ylabel('Amplitude');

title('Output Obtained by Convolution');grid;

x1 = [x zeros(1,8)];

y1 = filter(h,1,x1);

subplot(2,1,2);stem(n,y1);

xlabel('Time index n'); ylabel('Amplitude');

title('Output Generated by Filtering');grid;

Is there any difference between y[n] and y1[n]? What is the reason for using x1[n] obtained by zero-padding x[n] as the input for generating y1[n]?

Ans: No, there is no difference between y[n] and y1[n]. Here both convolution and filtering gives the same response. Using filter function the

number of output (y1)values produced corresponds to the number of values in input(x).

Thus if x is given as input without zero padding , filtering operation gives only first 7(length of input)values as output. These are same as the first 7 values obtained by convolution.

**Without zero padding code:**

h = [3 2 1 -2 1 0 -4 0 3]; % impulse response

x = [1 -2 3 -4 3 2 1]; % input sequence

y = conv(h,x);

n = 0:14;

subplot(2,1,1);stem(n,y);

xlabel('Time index n'); ylabel('Amplitude');

title('Output Obtained by Convolution');grid;

x1 = [x zeros(1,8)];

h1= [h zeros(1,6)];

y1 = filter(h,1,x);

subplot(2,1,2);stem((0:14),[y1 zeros(1,8)]);

xlabel('Time index n'); ylabel('Amplitude');

title('Output Generated by Filtering');grid;

**RESULT**

The MATLAB codes for each question was written and the plots were verified.