NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

Department of Electronics and Communication Engineering

EC4091 Digital Signal Processing Lab

Tool: MATLAB

Additional Experiment: Cycle 1

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

```
% 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])
```

2. The family of continuous time sinusoids $\cos(\Omega_0+k\Omega_s)t$, k=0,±1, ±2..... where $\Omega s=2\pi/T$ 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.

```
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
```

3. 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. $b=[b_0,b_1,.....b_M]$; $a=[a_0,a_1,......a_N]$; 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.
   a. 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).

   ```
   a=[1 -1.143 .4128 ]; b=[.0675 .149 .675];
   y=filter(b,a,.2*ones(1,50)); stem(0:49,y)
   ```

   b. 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

   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]?