DSP Major Project 1

Table of Contents

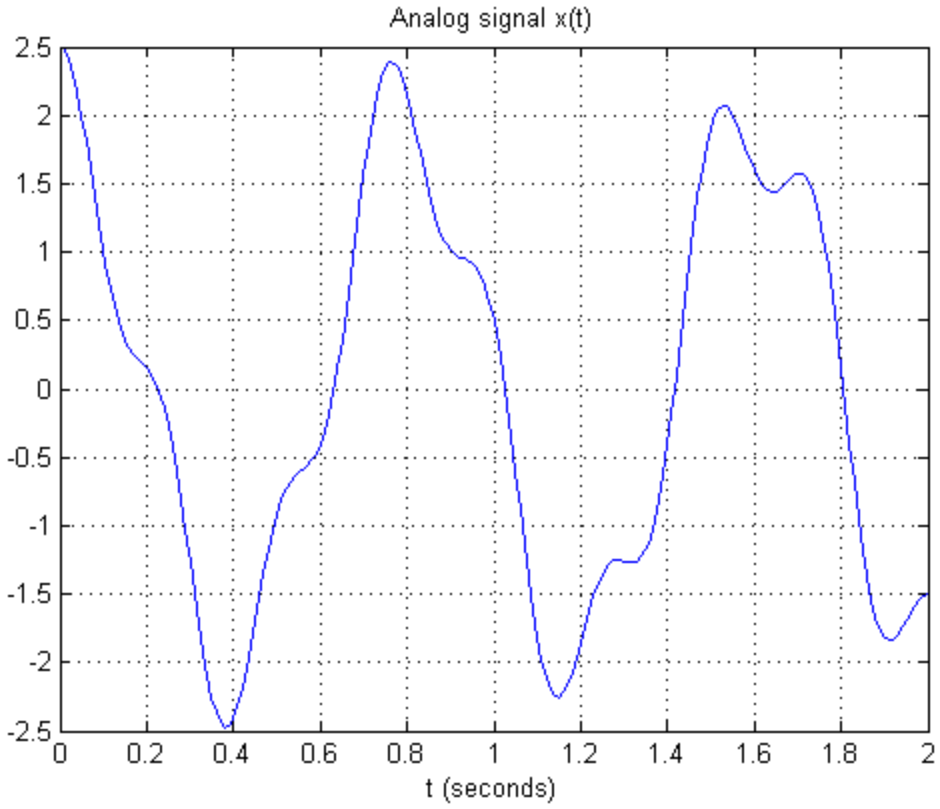
This project illustrates the concept of signal sampling and reconstruction using digital signal processing. It shows a CT signal with multiple frequency components as an example, which is sampled with a sampling rate factor and reconstructed afterwards. The project further demonstrates the concept of Nyquist theorem and sampling theorem.

Mark Anthony Cabilo, MEECE-CCO 2, MEE 1231

Continuous-Time Signal

This section creates a signal with two components of different frequencies. The signal is then plotted. To represent a CT signal, we use a small step size (1/100 or 0.001). By CT properties, any distinct value of F is periodic. So the two components are periodic by themselves. When the two components are added together, the resulting signal is still periodic (LCM of the two frequencies).

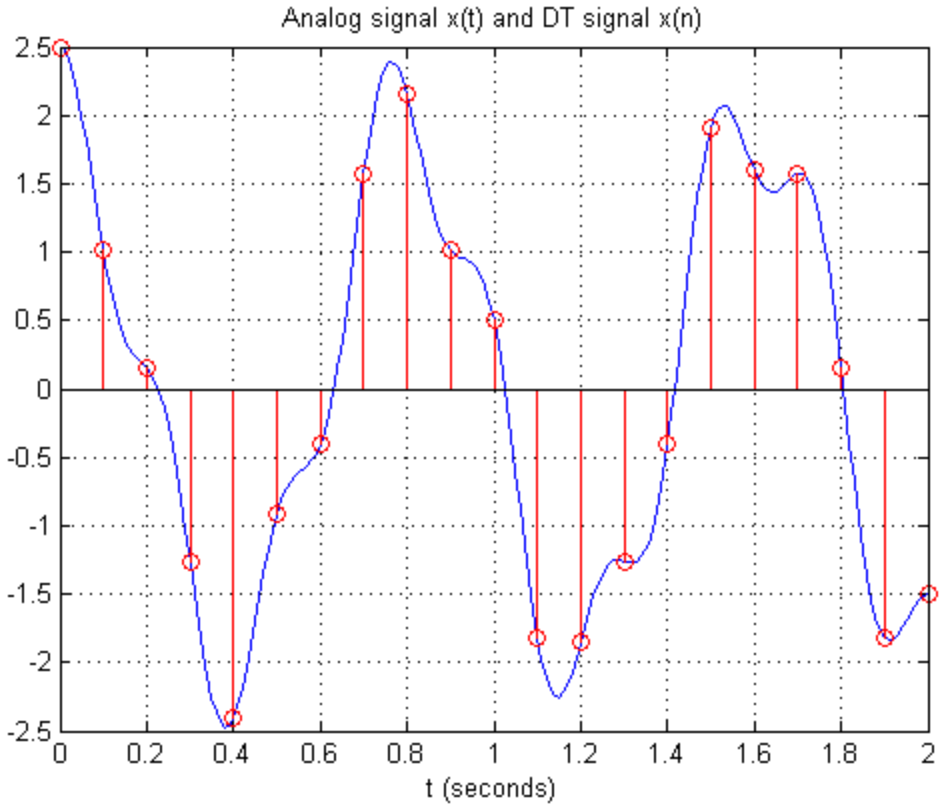
```
t = [0:1/100:2]'; %increment by 0.001 for continuous-time signal
F1 = 5/4;         % frequency of component 1 of x(t)
F2 = 4;           % frequency of component 2 of x(t)
xt = 2*cos(2*pi*F1*t) + 0.5*cos(2*pi*F2*t); % CT signal function
figure, plot(t,xt), grid % plot of the CT function
xlabel('t (seconds)'), title('Analog signal x(t)')
```



Signal sampling

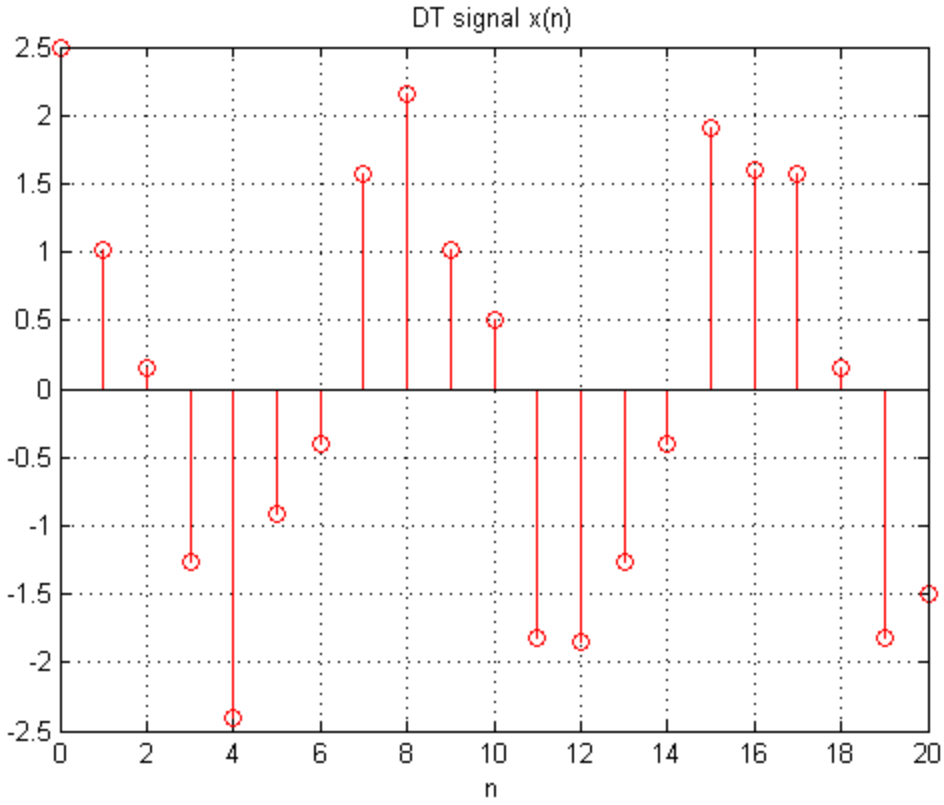
To convert the signal into a discrete-time signal, we sample. We use a 10 Hz sampling rate. This value follows the Nyquist condition; that is, the sampling frequency must be at least twice the maximum frequency of the signal. This means we can resconstruct the original CT signal after sampling it to a DT signal. We superimpose the resulting sampled signal on the original analog signal.

```
Fs = 10; % sampling rate, Hertz
T = 1/Fs; % sampling interval, seconds
nT = t(1:10:end); % time when sampling occurs
xn = xt(1:10:end); % sequence of sampled values; DT signal x(n)
hold on, stem(nT,xn,'r'); % stem plot of samples
title('Analog signal x(t) and DT signal x(n)');
```



We create a plot of the DT signal or the sequence x(n).

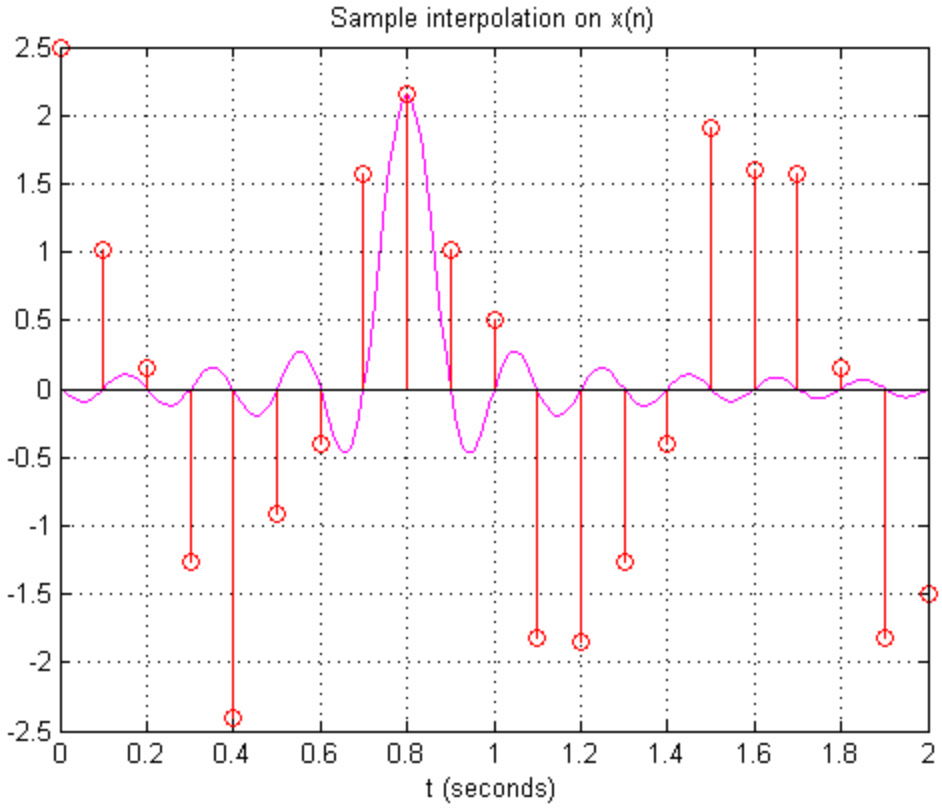
```
n = [0:length(nT)-1]'; % this is the sequence's indices
figure, stem(n,xn,'r'), grid % stem plot of sequence
% note the values on the x-axis (they are now discrete numbers or integers)
xlabel ('n'), title('DT signal x(n)')
```



Sample sinc function

As an illustration, we select n = 9 and apply the sinc function as the basis for our CT signal reconstruction. The sinc function is a interpolation function. Interpolation means to add values in between known values. This is helpful to recreate a CT signal from a DT signal.

```
tt = Fs*(t-0.8); % sinc function in Matlab is defined as sin(pi*x)/pi*x
y = xn(9)*sinc(tt); % sampling theorem (weighted sinc function)
figure, plot(t,y,'m'), grid, hold on % plot sinc function
stem(nT,xn,'r') % stem plot of DT signal
xlabel('t (seconds)'), title('Sample interpolation on x(n)');
```



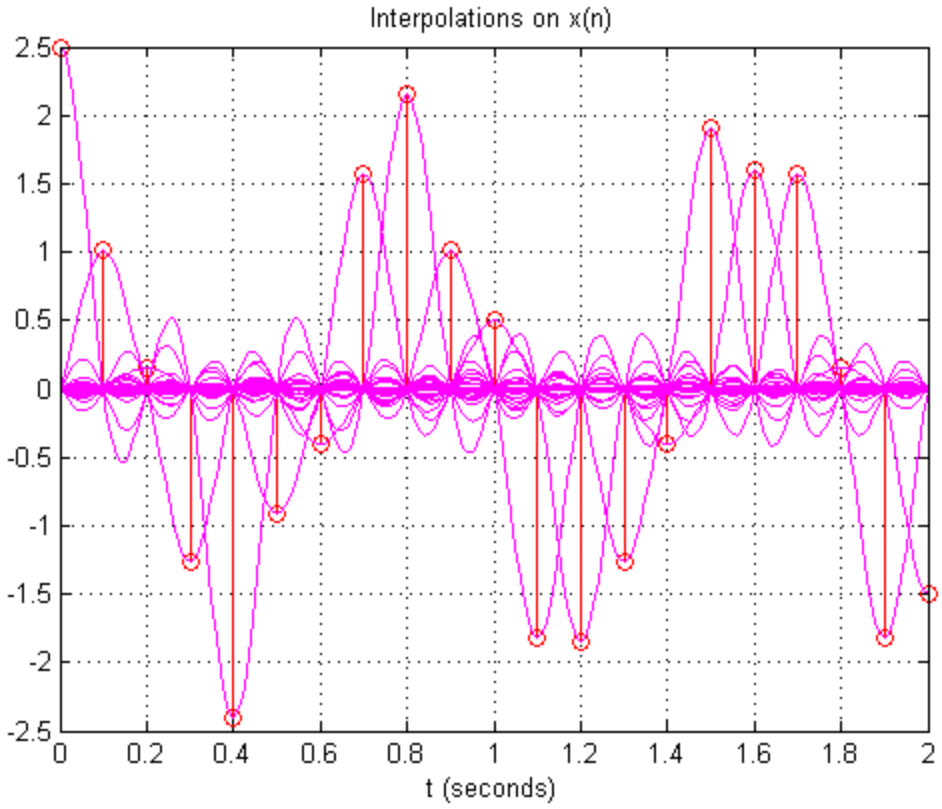
It is important to observe that this single interpolation intersects with other samples only in the x-axis. This means that a single sample's interpolation is not weighted by the other samples but only the current sample alone. This is important as the sampling theorem would require the summation of each sample's interpolation. It can be further observed that the interpolation 'dies' out as further it gets from the center (location of current sample). It can be likened to the concept of 'influence'. The discrete sample has a greater 'influence' to its neighboring time compared to the time closer to other samples. The gaps between samples are filled and interpolated due to these 'influences', the value of which are based from the sinc function itself. It has something to do with the concept or principle behind using the sinc function as an interpolation function. If we look into the derivation of the sinc function, we will find that the function is the continuous inverse Fourier transform of a rectangular pulse of width 2pi and of height 1. "The SPACE OF FUNCTIONS bandlimited in the range of frequencies (-pi to pi) is SPANNED by an INFINITE YET COUNTABLE set of sinc functions shifted by integers". This spanning effect creates the 'influences' being discussed here. A set of integer-spaced samples can then be reconstructed into a CT signal by adding the sinc functions of each sample.

Interpolation per sample

The next step in reconstructing the analog signal, we get the weighted sinc function for all the samples. We can at the same time sum them all up in preparation to the complete reconstruction in the next section.

```
sum = zeros(201,1); % this variable is used for the next section
figure, stem(nT,xn,'r'), hold on % stem plot of DT signal
for nn = 1:length(nT) % for every sample, we interpolate:
    %the addition of 0.1 is to compensate the shift in indexing
    tt = Fs*(t-nn*T+0.1); % compute for the sample-shifted sinc func input
    y = xn(nn)*sinc(tt); % weighted sinc function
```

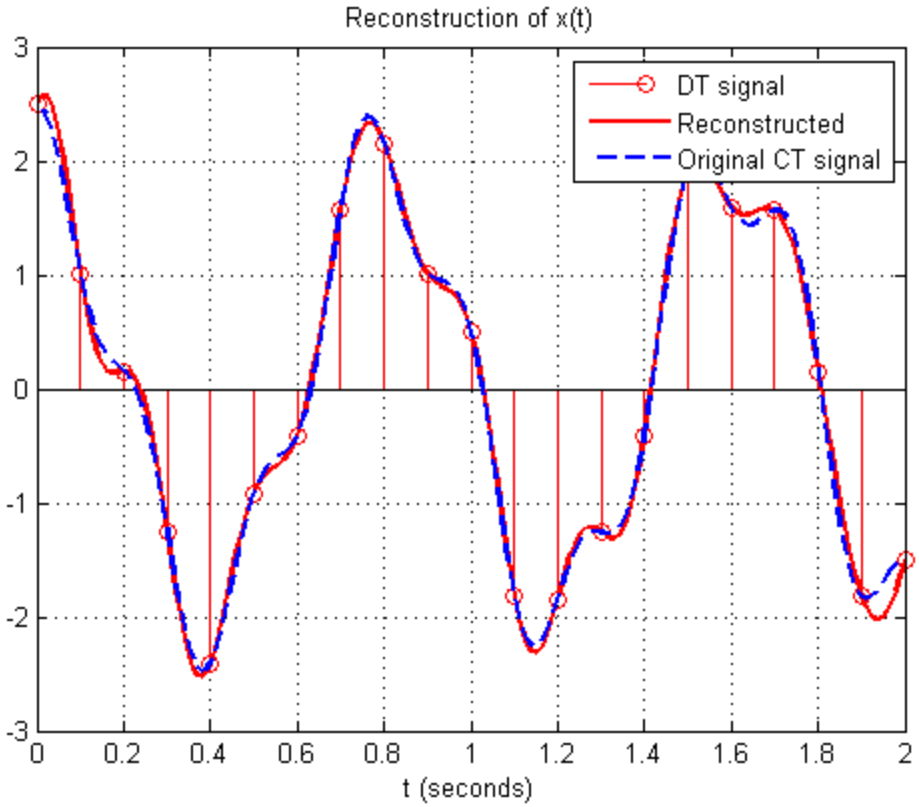
```
    % plot each sample's sinc function
    plot(t,y, 'm'), grid, hold on
    sum = sum+y; % sum up each weighted sinc function
end
xlabel('t (seconds)'), title('Interpolations on x(n)');
```



Analog signal reconstruction

To complete the reconstruction, we sum up all the weighted interpolation functions (which is according to the sampling theorem). The sum has already been computed in the previous section and is simple plotted here. The original analog signal is superimposed to compare with the reconstructed CT signal.

```
figure, stem(nT,xn,'r'), hold on % stem plot of the DT signal
plot(t,sum,'r','LineWidth',2), grid, hold on % plot of reconstructed signal
plot(t,xt, '--b', 'LineWidth', 2); % plot of original signal
xlabel('t (seconds)'), title('Reconstruction of x(t)');
legend('DT signal', 'Reconstructed', 'Original CT signal');
```



It can be observed that the reconstructed signal is not exactly the same as the original signal. However, the sampling theorem says that a CT signal can be exactly recontructed for as long as the frequency requirements are satisfied. The possible reason for this is that the chosen sampling frequency (10Hz) is not an integer multiple of Fmax(4Hz). Although Fs > 2Fmax which means no aliasing will happen, the samples will not get equal number of points per cycle of the original signal. This will later lead to inconsistent number of points to interpolate per cycle.

Published with MATLAB® R2013a