LAB 2

SAMPLING THEOREM & SIGNAL RECONSTRUCTION

OBJECTIVES

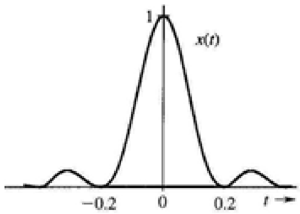
- Use the sampling theorem for signal sampling and signal reconstruction
- Understand sampling theorem, aliasing and how to minimize error due to aliasing.

INTRODUCTION

In most practical applications, signals exist in the continuous form. For the digital systems to process continuous signals, these signals are converted into discrete signals by sampling. The more samples the better reconstructed representation of the continuous signal. On the contrary, more samples require more processing time and storage space. An optimum sampling rate can preserve all information of the original signal while the minimum number of samples will reduce the processing time and power. The sampling theorem is used to predict this minimum sampling rate required for best signal reconstruction. According to the sampling theorem, the sampling rate (number of samples per seconds) must be equal or greater than twice the bandwidth of the continuous signal (the highest frequency in the continuous signal). When the sampling rate is exactly twice the bandwidth then the sampling rate is called the **Nyquist** rate. If the sampling rate is below the Nyquist rate, then the recovered continuous signal from the samples will not be 100% identical to the original signal due to under- sampling. This results in spectrum distortion which is known as aliasing. Aliasing causes the high frequency components to appear as low frequency components. For example, if the highest frequency in a signal $x(t)$ is 1000 Hz and the sampling rate is 1800 samples per second instead of the Nyquist rate 2000 samples per second; then the 1000 Hz will appear as 800 Hz in the reconstructed signal. When sampling rate, f_s, is 1800 samples per second then according to the sampling theorem the maximum possible frequency, f_{max}, is 900 Hz (half the sampling rate). So, the 1000 Hz in the signal $x(t)$ will appear, f_a, as (900 - (1000-900) = 800 Hz. The general formula for the apparent frequency due to under-sampling is given here.

$$f_a = |\langle f_o + {}^{f_s}/_{2} \rangle_{f_s} - \tfrac{f_s}{2}| \qquad (1)$$

where $\langle a \rangle_b$ is the *modulo operation a* modulo *b.* If 1100 Hz is present in the signal $x(t)$, that was sampled at sampling rate 2000 then it will appear as| $<1100+2000/2>_{2000}$ - 2000/2 | = |100 – 1000| = 900 Hz. The signal $x(t)$ in Fig. 1a has a spectrum $X(\omega)$ as shown in Fig. 1b. Spectrum $X(\omega)$ has bandwidth of 5 Hz (10π rad/sec), so the Nyquist sampling rate f_s is 10 samples/second ($\omega_s = 2\pi 10 = 20\pi$).



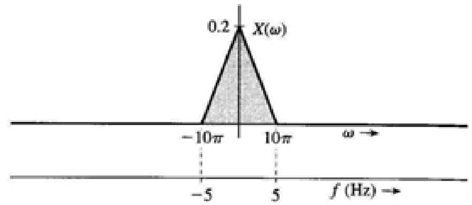


Fig. 1a Fig. 1b

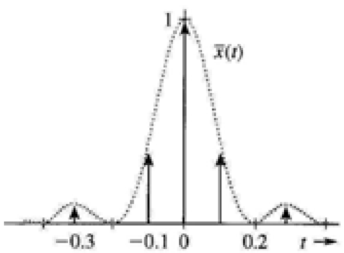
Sampling is established by multiplying the continuous signal x(t) by a train of impulses separated by sampling interval $T = 1/f_s$ seconds. Sampled signal can be expressed mathematically:

$$\bar{x}(t) = x(nT) = x(t)\sum_{n=-\infty}^{n=\infty}\delta(t-nT) \qquad (2)$$

Using the exponential Fourier series the above periodic signal can be expressed as:

$$\bar{x}(t) = x(nT) = x(t)\frac{1}{T}\sum_{n=-\infty}^{n=\infty}e^{jn\omega_s t} \qquad (3)$$

The sampled signal $x(nT)$ and its spectrum are shown in Fig. 2a and 2b, respectively.



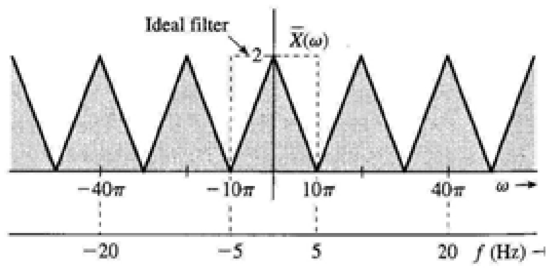


Fig. 2a Fig. 2b

Sampling in the time domain is equivalent to the duplication of the spectrum of the original signal centered around $n\omega_s$ for $n = 0, 1, 2, \ldots$ This is true because of the frequency shifting property of the Fourier transform. Note that the frequency shifting property of the Fourier transform which is phase shifting of the signal $x(t)$ in the time domain is equivalent to the frequency shifting of its spectrum $X(\omega)$ in the frequency domain.

The mathematical expression of the spectrum $\overline{X(\omega)}$ of equation 3 using the frequency shifting property is

$$\bar{X}(\omega) = \frac{1}{T}\sum_{n=-\infty}^{n=\infty}X(\omega - n\omega_s) \qquad (4)$$

In order to recover the original signal $x(t)$ from samples of $x(nT)$, the samples are passed through a low pass filter with cutoff frequency equal half the sampling rate f_s. For the above example the cutoff frequency is 5Hz which is equivalent to 10π rad/sec. The ideal low pass filter is shown in Fig. 2b by the doted rectangular. The spectrum of the low pass filter's output is the spectrum of the original signal. So, the output of the low pass filter will be the continuous signal $x(t)$. The mathematical expression of reconstructed original signal $x(t)$ from its samples $x(nT)$ using low pass filter is

$$X(\omega) = H(\omega)\,\overline{X}(\omega) \tag{5}$$

Where $H(\omega)$ is the frequency response of the low pass filter in the frequency domain. The mathematical expression of filtering in the time domain is the convolution of the sampled signal $x(nT)$ with the impulse response $h(t)$ of the signal:

$$x(t) = h(t) * \textstyle\sum_n x(nT)\delta(t - nT) \tag{6}$$

The convolution property shows that when a signal such as $h(t)$ is convolved with an impulse $\delta(t-nT)$ then the outcome of the convolution is $h(t - nT)$, so equation 6 can be expressed as.

$$x(t) = \sum_n x(nT)h(t - nT) \tag{7}$$

The transfer function $H(\omega)$ of the low pass filter is a rectangular function: $rect(\omega/20\pi)$ as shown in Fig. 3a, and its inverse Fourier transform $h(t)$ is the sinc function sinc(2π5t) as shown in Fig. 3b.

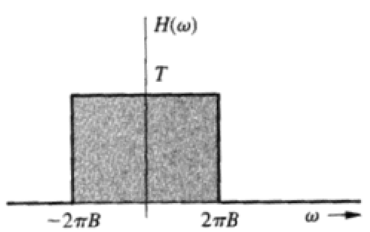


Fig. 3a

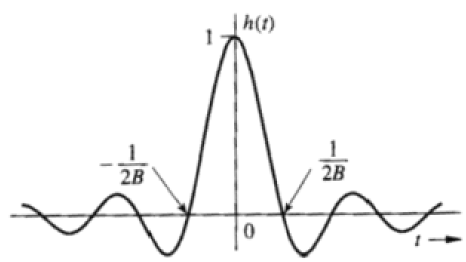


Fig. 3b

With an ideal rectangular low pass filter the output of the filter, Equation (7), is expressed mathematically as:

$$x(t) = \sum_n x(nT)\mathrm{sinc}(2\pi B(t - nT) \tag{8}$$

The above equation shows that the original signal can be reconstructed from the samples by adding sinc functions located at the time locations of the samples and with amplitude equal the values of the corresponding samples, see Fig. 4.

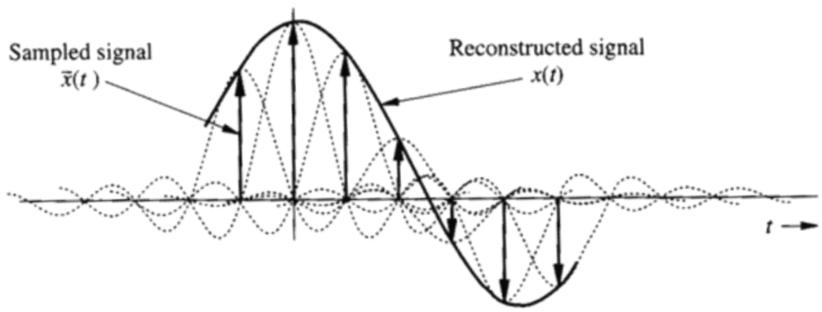


Fig. 4

One can think of resampling as interpolation of the signal's data points between samples by using the samples and the sinc functions. Other method for interpolation is to use linear interpolation where the signal's data points between samples are approximated by a linear line connecting adjacent samples. In this lab, both methods will be analyzed.

Sampling and the Discrete Fourier Transform

The spectrum of a continuous signal is calculated by the integral formula of the Fourier transform (FT).

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

However, in digital systems the continuous signal is sampled to be a discrete signal and then the spectrum is calculated by the discrete Fourier transform (DFT) formula.

$$X[\omega_o k] = \sum_{n=0}^{\text{N-1}} x[nT]e^{-jk\omega_o nT}$$

The DFT equation is obtained from FT equation by replacing t by nT where T is the sampling interval in the time domain, and replacing ω by $k\omega_o$ where ω_o is the sampling interval in the frequency domain, and the integral is converted to summation.

The terms T and ω_o are fixed quantities and determined by the application. The product $T\omega_o$ equals $2\pi T/T_o$ where T_o is the time duration of the signal that is sampled. It can be simplified further to $2\pi/N$ where N is the number of the samples in the sampled signal. The final form of the DFT equation is

$$X[k] = \sum_{\text{n=0}}^{\text{N-1}} x[n]e^{-j2\pi kn/N}$$

Here, $x[n]$ is the samples of the continuous signal $x(t)$. The number of samples N is determined by the sampling rate and the length of the signal $x(t)$ (window). For example, if the sampling rate is 8000 samples per second, then a signal $x(t)$ windowed by 2 seconds length will generate a discrete signal $x[n]$ of length 16,000 samples. Note, the sampling rate must be equal or larger than twice the highest frequency in the signal $x(t)$.

In practice the DFT is calculated faster by efficient algorithm that is called fast Fourier transform (FFT). The frequency resolution of the DFT will be the reciprocal of the time length of the signal (window). For example, a signal of length 2 second will have a discrete Fourier transform with frequency resolution equal 0.5 Hz (1/2 seconds).

The term k in the DFT equation represents the harmonic frequency. For example, for $k=3$, $X[3]$ represents the strength of the third harmonic. If the frequency resolution is 0.5 Hz then $X[3]$ is the strength of the 1.5 Hz. For $k = 0$, $X[0]$ is the dc component in the signal. The highest frequency component in the signal $x(t)$ will be half the sampling rate.

To summarize, the discrete Fourier transform can be linked to the continuous Fourier transform by the rules below:

1. The sampling rate must be larger than twice the bandwidth of the signal.
2. The frequency resolution of the FFT equal the reciprocal of the signal length (window size)
3. The number of samples will produce frequency components that is equal to half the number of samples.
4. The lowest frequency is DC ($k = 0$), and the highest frequency is half of the sampling rate.

Spectrogram Function in Matlab

The spectrogram function in Matlab can plot the magnitude spectrum $|X[k]|$ of a signal in the y-axis verses time in the x-axis. The spectrogram function calculates the DFT of the signal for specified time duration. The signal is divided to small time intervals and then the DFT for each interval is calculated. The Fourier transform for each interval will reveal the frequency components present in this interval. The strengths of the different frequency components are coded by color. The more yellowish the color the stronger is the frequency component in the signal.

If you type in Matlab >> help spectrogram then you will get detail description of the different use of the function. In this lap we will use the following format:

```
spectrogram(X, WINDOW, NOVERLAP, NFFT, Fs)
```

X: The signal.
WINDOW: number of samples in the time duration that will be used to find the spectrum of the signal during this time.
NOVERLAP: Number of samples that overlap with the previous time duration.
NFFT: The samples of the frequency components from the calculated Fourier transform.
Fs: The sampling rate of the signal.
The following example for the signal $y(t)$ demonstrates the use of the spectrogram function in MATLAB to show the spectrum of the signal.

```
clear all
Fs=8000;                                   % sampling frequency
Wind = 0.1;                                % time interval for calculating the Fourier transform (FT) is 0.1 second
NumSampWind = Fs*Wind;                     % number of samples in the window for calculating the Fourier Transform
t1=0:1/Fs:3;                               % time scale from zero to 3 seconds
t=0:1/Fs:1;                                % time scale from zero to 1 seconds
ZeroTime = zeros(1, length(t)-1);
t2=[ZeroTime t ZeroTime];                  % first and third seconds are zeros
Freq = [0:10:4000];                        % frequency at which the Fourier transform will be calculated
Sig = 3*sin(2*pi*500*t1) + sin(2*pi*2000*t2);   % sin(2π500t) for 0<t<3 and sin(2π2000t) for 1<t<2
spectrogram(Sig, NumSampWind, 64, Freq, Fs, 'yaxis')   % calculate FT and plot spectrum where time in x-axis and % frequency in y-axis
```

PRELAB EXERCISES

A continuous signal $x(t)$ defined as

$$x(t) = 3 + \cos(4\pi t - \pi/4) + 2\cos(8\pi t) \quad \text{where } 0 < t < 1\text{s}$$

Answer the following:

1) What is the Nyquist sampling rate? ___________

2) What is the fundamental frequency f_o? ___________

3) According to the sampling theorem if the above signal was sampled at a rate of 6 samples/sec, then the constructed signal from the samples will be:

$$x_s(t) = 3 + \cos(2\pi 2t) + 2\cos(2\pi f_a t)$$

What is the apparent frequency f_a? ___________

LAB EXERCISES

TASK 1: Sampling and Signal Reconstruction Using Sinc Interpolation

In this task you will sample the signal $x(t)$ at a rate higher than the Nyquist rate and then reconstruct the continuous signal $x(t)$ from its samples x_n. The continuous baseband signal $x(t)$ is shown below.

$$x(t) = 3 + cos(2\pi 2t) + 2cos(2\pi 4t) \qquad 0 < t < 1 \text{ second}$$

1) All signals processed in the computer are discrete signals, meaning they are digitally processed. However, a discrete signal of a high sampling rate (sampling rate > 20*Nyquist rate) can be considered a true representation of the original signal $x(t)$. You can verify this by plotting the sampled signal against t using the code below.

```
fs = 100;               % sampling rate 100 samples per second
t = 0:1/fs: 1;          %100 samples interval is 0.01 second (1/fs)
x = 3 + cos(2*pi*2*t) + 2*cos(2*pi*4*t);
plot(t, x);
```

2) What is the Bandwidth B in Hz of the signal $x(t)$? Sample the signal using Nyquist rate to obtain x_n.

```
fs= ? ;            % replace the question mark by the Nyquist rate
ts=0:1/fs:1;
xn= 3 + cos(2*pi*2*ts)+2*cos(2*pi*4*ts);
```

3) Plot the sampled signal, x_n, against the sampling time using MATLAB **stem** function.

>> stem(ts, xn);

4) Use MATLAB **plot** function to plot the sampled signal, x_n, against the sampling time.

>> plot(ts, xn);

The plot function in MATLAB connects a line between samples, so it uses linear interpolation. Does the plot look like the original signal in part 1?

5) According to the sampling theorem, we should be able to reconstruct the original signal $x(t)$ from the sampled signal $x[n]$, x_n, if the sampling rate equal or higher than the *Nyquist* rate.

$$x_s(t) = \sum_{n=0}^{N-1} x(nT) \text{ sinc}\left(\frac{\pi(t-nT)}{T}\right) \qquad (9)$$

Where T is the sampling interval and N is the number of samples. The MATLAB code that implements Eq. 9 is shown below and is named **`sincInterpolation(xn, fs, Rate)`**. Write this code in Matlab and save it in your folder.

<u>MATALB Code for `sincInterpolation(xn, fs)`</u>

```
function [xHat fsH] = sincInterpolation(xn, fs, Rate)
nx = length(xn);           % number of samples in the original siganl
Ts = 1/fs;                 % Oriqninal sampling interval
lx = Ts*(nx-1);            % length of original signal in seconds
ts = 0:1/fs:lx;            % Discrete time intervals of the original sampling
fsH = Rate*fs;               % A new higher sampling rate
T = 1/fsH;                 % New sampling interval
t=0:1/fsH:lx;              % New discrete time intervals = 1/fsH
xHat = 0;                  % initializing the new reconstructed signal xHat
% sinc interpolation with amplitude equal the sample and shifted one interval
for n=1:nx
    m = n-1;                 % because Matlab does not have index of value zero
    xHat = xHat + xn(n)*sinc((t-m*Ts)/Ts);
end
stem(ts, xn)                % plot the original samples.
hold on
plot(t,xHat)               % plot the reconstructed signal, with delay.
```

This function will use x_n samples and the sampling rate f_s that you pass to it to reconstruct the continuous signal $x(t)$ using Eq. 9 above. It will increase the sampling rate by a factor of 10.

```
>> [xHat fsH] = sincInterpolation(xn, fs, 10);
```

Are the plots of $x(t)$ of Part 1 and $x_s(t)$ of Part 5 the same? How many samples in the estimated signal $x_s(t)$? Are you amazed of the accuracy of estimating all the points of $x_s(t)$ from the small samples in x_n? If you are not amazed, then call your instructor to discuss the signal reconstruction.

TASK 2: <u>Under-sampling and Aliasing Effects</u>

1) Sample the signal $x(t)$ at a rate of $f_s = 6$ which is below the Nyquist rate.

```
fs= 6 ;
ts=0:1/fs:1;
xn= 3 + cos(2*pi*2*ts)+ 2*cos(2*pi*4*ts);
```

Use the MATLAB function **`sincInterpolation()`** to reconstruct the signal from the samples.

2) What is the apparent frequency due to under-sampling and aliasing effects?

3) Plot the constructed signal $x_s(t)$.

4) Does the constructed signal look the same as the original signal $x(t)$? If not, then explain the reason that causes the difference.

5) What is/are the frequencies present in the constructed signal $x_s(t)$. Plot the signal

$$x_{sa}(t) = 3 + cos(2\pi 2t) + 2cos(2\pi f_a t) \qquad 0 < t < 1 \text{ second}$$

where f_a is the apparent frequency of the 4 Hz due to under-sampling.

```
fs = 100;           % sampling rate 100 samples per second
t = 0:1/fs: 1;      %100 samples interval is 0.01 second (1/fs)
xsa = 3 + cos(2*pi*2*t)+2*cos(2*pi*fa*t);
plot(t, xsa);
```

Does the plot of $x_{sa}(t)$ look the same as the plot of the origninal signal $x(t)$ or the signal reconstructed in part 1?

TASK 3: <u>Impact of Under-sampling and Aliasing on Sound</u>

1) In this part we will try to hear tones of different frequencies sampled at 4200 samples per second to demonstrate the effect of aliasing due to undersampling. Run the code below and you should hear a tone of 1600 Hz.

```
f0=1600;                  % the frequency of the tone
fs = 4200;                % sampling rate
ts = 0:1/fs:2;            % samples for 2 second with sampling interval = 1/fs
x=0.8*sin(2*pi*f0*ts);    % samples of the continuous tone signal
sound(x, fs)
```

Change f0 to the following frequencies 1800 Hz and 1900 Hz. Do the pitches of the tones of the 1600 Hz, 1800 Hz, and 1900 Hz increases as the frequency increases? Note the following, the sampling rate of 4200 still above the Nyquist rate for the 1900 Hz.

2) Now change the frequency to 2300 Hz and 2400 Hz and hear the tones. Does the pitch of the tones keep increasing in comparison to the 1900 Hz? If not then compare them to the 1600, 1800, 1900 and determine which two frequencies sound the same. Which of the two frequencies 2300 or 2400 sounds like the 1800Hz? This is a demonstration of aliasing.

TASK 4: <u>Sampling and Reconstruction of Bandpass Signal</u>

Download the Bird song from the course website to your current Matlab folder. The Bird song file is a stereo file which means it has n rows and two columns. For the tasks below create a new audio file that has only one column as follow

```
>> MonoSig = SteroSig(:,1);
```

1) Use the spectrogram function to find the spectrum of the Bird song. Plot the spectrogram in your report

2) Down sample the Bird song audio file by a factor of 2 and 7. For each down sample make sure to plot the spectrogram and listen to the downsampled file with the appropriate sampling rate. Comment on the spectrogram and the quality of the sound.

3) Sample the Bird Song with the lowest sampling rate such that you can reconstruct the original signal from the samples without any distortion such as aliasing.

4) Reconstruct the Bird song from the few samples in task 3. Basically, estimate the 44100 samples/second from the available samples/second you obtained in task 3. Plot the spectrogram and play the audio signal. Comment on the constructed signal with respect to the original signal.

LAB REPORT FORMAT

Submit the solutions to the problems chronologically with a front Title Page and indicate corresponding Problem Number on Top Left Corner of Each Problem/Page.

(1) Title Page (2) Introduction (3) Results (Include answers to all exercises) (4) Conclusion