EE092IU
DIGITAL SIGNAL PROCESSING LABORATORY

LAB 1

SAMPLING AND RECONSTRUCTION OF ANALOG SIGNALS

Student name: ………………………………..

Student ID: ………………………………..

Class: ………………………………..

Date: ………………………………..

LAB OBJECTIVES

The objective of this lab is to explore the concepts of sampling and reconstruction of analog signals. Specifically, we will simulate the sampling process of an analog signal using MATLAB, investigate the effect of sampling in the time and frequency domains, and introduce the concept of aliasing.

FUNDAMENTAL DISCUSSION

An analog-to-digital converter (ADC) converts an analog signal to a digital form. An ADC produces a stream of binary numbers from analog signals by taking the samples of the analog signal and digitizing the amplitude at these discrete times. Prior to the ADC conversion, an analog filter called the prefilter or antialiasing filter is applied to the analog signal in order to deal with an effect known as aliasing. Aliasing causes multiple continuous time signals to yield the exact same sampled discrete time signal.

In this lab we focus our attention in the process of sampling and how to avoid the problem of aliasing. During sampling, an analog signal $x_a(t)$ is periodically measured every T_s seconds:

$$t = nT_s, \qquad n = 0,1,2 \ldots \tag{1}$$

where T_s is called the sampling period, and is the fixed time interval between samples (here we assume a uniform sampling rate that does not change with time.) The inverse of Ts is called the sampling frequency, that is, the samples per second:

$$f_s = {}^{1}\!/_{T_s} \tag{2}$$

When sampling an analog signal, we must sample fast enough (i.e. be sure f_s is sufficiently high), so that the samples are a good representation of the original analog signal. If the sampling frequency fs is not fast enough then too much information is lost, and it becomes impossible to reconstruct our original analog signal using a digital-to-analog converter (DAC). However, if you do sample fast enough, then, theoretically, it is possible to exactly reconstruct the original signal.

Sampling theorem:

For accurate representation of signal $x_a(t)$ by its time samples $x[nT]$, two conditions must be met:

1) The signal $x_a(t)$ must be bandlimited, that is, its frequency content (spectrum) must be limited to contain frequencies up to some maximum frequency f_{max} and no frequencies beyond that, and

2) the sampling rate f_s must be chosen to be at least twice the maximum frequency f_{max} , that is

$$f_s \geq 2f_{max} \tag{3}$$

According to the sampling theorem, before sampling we must make sure the signal is bandlimited (this is the function of the analog prefilter) and that the sampling frequency is at least twice the maximum frequency.

The traditional Nyquist sampling theorem presented above is true for real-valued (i.e. not complex), lowpass (i.e. baseband) signals. For complex lowpass signals, the sampling theorem states that the ultimate minimum sampling rate to avoid aliasing is actually $f_s = \beta$, where β is the double-sided bandwidth. For bandpass signals, things get even more interesting. You can subsample (i.e. sample below the Nyquist rate) to achieve frequency translation to lower frequencies and recover the original signal.

Example 1:

Consider the following continuous time signal:

$$x_a(t) = 3sin(2\pi \times 0.5t) + 2sin(2\pi \times 2.5t) \tag{4}$$

The signal contains two frequency components at $f_1 = 0.5Hz$ and $f_2 = 2.5Hz$. We explore the effect of aliasing by sampling $x_a(t)$ at $f_s = 2.5f_{max} = 6.25Hz$. This sampling frequency meets the sampling theorem requirement. In MATLAB we explore time-domain by making the following plots:

- The original time signal $x_a(t)$
- The sampled signal $x[n]$ (discrete time signal) with $f_s = 2.5f_{max} = 6.25Hz$
- The sampled signal superimposed on top of the reconstructed continuous time-signal

Result:

Figure 1 shows the original continuous time signal $x_a(t)$ in the first plot. Second plot shows the sampled discrete time signal $x[n]$ with $f_s = 2.5f_{max} = 6.25Hz$. Third plot shows the sampled discrete time signal superimposed on top of the reconstructed time-domain signal.

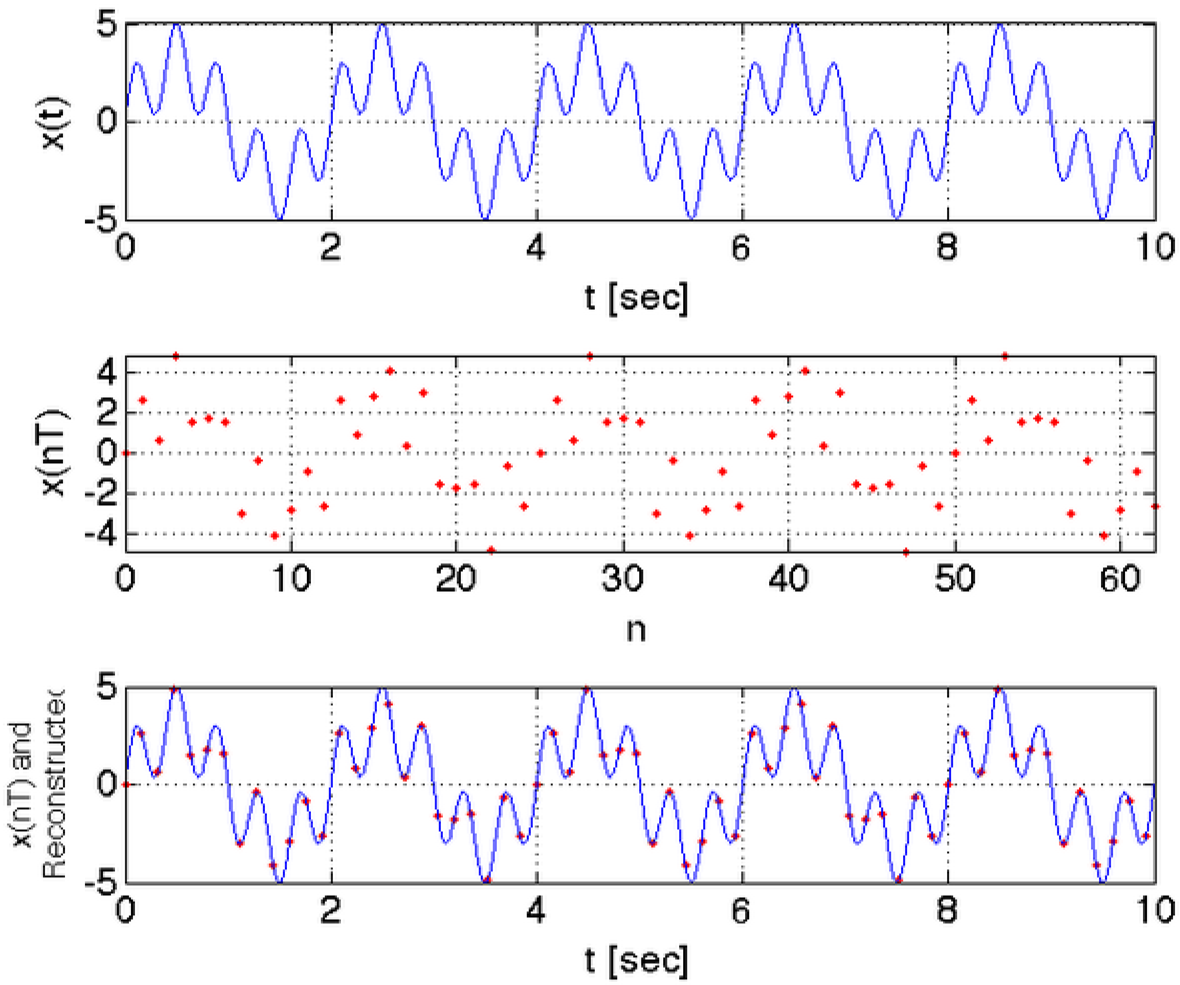

Figure 1. Simulation Result of Example 1

Example 2:

Consider the same continuous time signal as in Example 1:

$$x_a(t) = 3sin(2\pi \times 0.5t) + 2sin(2\pi \times 2.5t) \quad (5)$$

The signal contains two frequency components at $f_1 = 0.5Hz$ and $f_2 = 2.5Hz$. We explore the effect of aliasing by sampling $x_a(t)$ at $f_s = 1Hz$. This sampling frequency does not meet the sampling theorem requirement. In MATLAB we explore both time-domain and frequency domains by making the following plots:

- The original time signal $x_a(t)$
- The sampled signal $x[n]$ (discrete time signal) with $f_s = f_{max}/2.5 = 1Hz$
- The sampled signal superimposed on top of the original continuous time-signal
- The sampled signal superimposed on top of the reconstructed continuous time signal

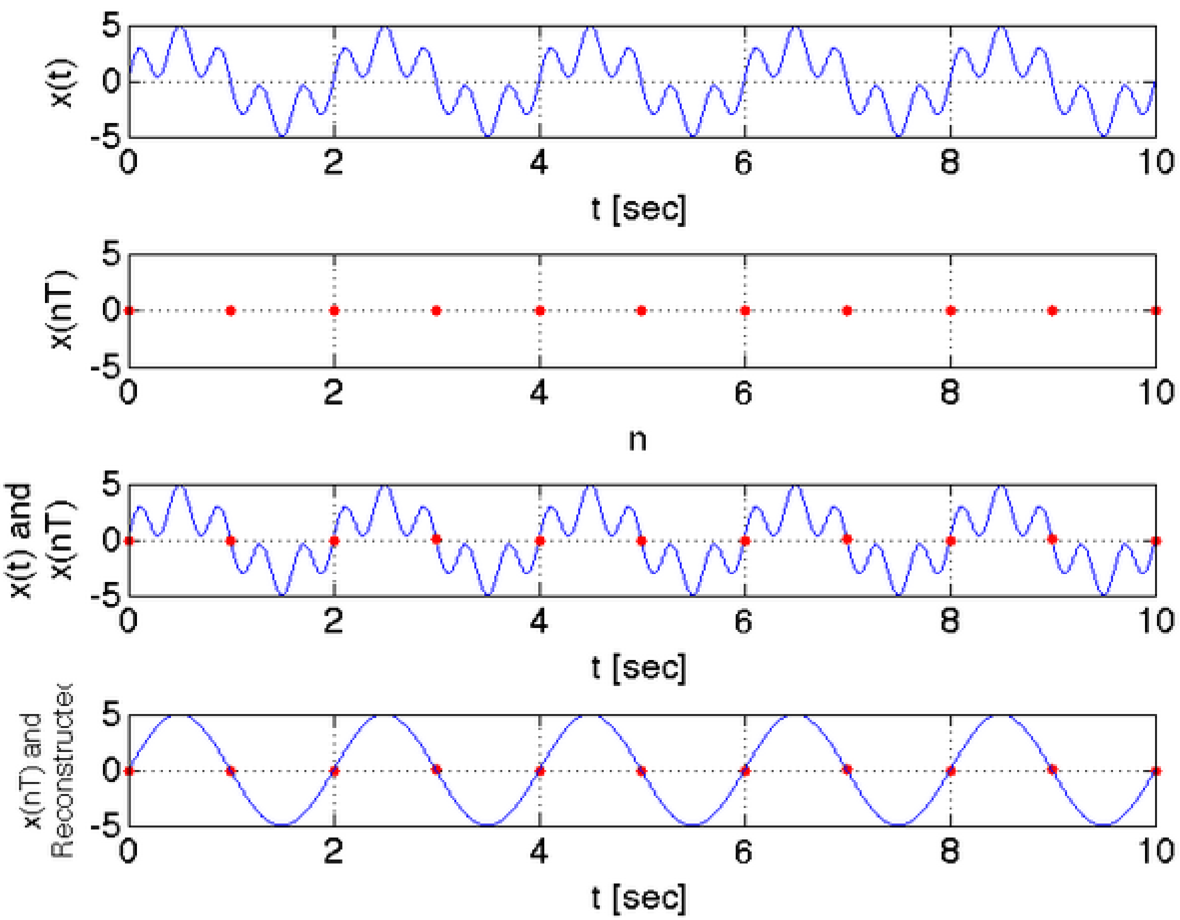

Figure 2. Simulation Result of Example 2

LAB PROCEDURE

Problem 1: Let $x(n) = \{1, -2,4,6-5,8,10\}$, with $n = 0$ at $x = -5$. Plot the given signal using ***stem*** function. Perform the signal manipulation technique as the following order to test your function, then plot those 4 signals using ***subplot*** function:

a) $x_1[n] = 3x(n+2) + x(n-4) - 2x(n)$

b) $x_2[n] = 5x(5+n) + 4x(n+4) + 3x(n)$

c) $x_3[n] = x(n+4)x(n-1) + x(2-n)x(n)$

d) $x_4[n] = 2x(n) + cos(0.1\pi n)x(n+2), -10 \le n \le 10$

Problem 2:

a) Given the signal $x(t) = cos(2\pi t) + cos(8\pi t) + cos(12\pi t)$, where t is in milliseconds. Show that if this signal is sampled at a rate of $f_s = 5kHz$, it will be aliased with the following signal, in the sense that their sample values will be the same:

$$x_a(t) = 3cos(2\pi t)$$

On the same graph, plot t $x(t)$ and $x_a(t)$ versus t in the range $0 \leq t \leq 2\ msec$. To this plot, add the time samples $x(nT_s)$ and verify that $x(t)$ and $x_a(t)$ intersect precisely at these samples.

b) Restart part (a) with $f_s = 10kHz$. In this case, determine the signal $x_a(t)$ with which $x(t)$ is aliased.

Problem 3:

Consider the two sinusoids:

$$x_1(t) = cos(\Omega_0 t) \qquad -\infty \leq t \leq \infty$$

$$x_2(t) = cos[(\Omega_0 + \Omega_s)t] \qquad -\infty \leq t \leq \infty$$

Show that if we sample these signals using $T_s = \frac{2\pi}{\Omega_s}$, we cannot differentiate the sampled signals. Do this by first proving the equation $x_1(nT_s) = x_2(nT_s)$ (Remember to include it in your report). Then, write a script file to input any 2 angular frequencies, and create a 3-by-1 plot where the first 2 plots are the given signals, and the final plot is the sampled signals overlap each other.

Problem 4:

Consider a continuous time domain signal:

$$x_a(t) = 2.5cos(2\pi \times f_1 t) - 1.5sin(2\pi \times f_2 t) + cos(2\pi \times f_3 t) + 0.5cos(2\pi \times f_4 t)$$

a. Write a script file to input the 4 component frequencies, then plot out the $x_a(t)$

b. Suppose we sample $x_a(t)$ with a sampling frequency $f_s = 5f_{max}$ (f_{max} is the maximum frequency automatically find out by the script). Write a MATLAB program to create a 3-by-1 plot like Example 1 where you plot the original continuous time signal $x_a(t)$, the sampled discrete time signal $x[n]$, and the sampled discrete time signal superimposed over the reconstructed signal.

c. Suppose we sample $x_a(t)$ with a sampling frequency $f_s = 0.5f_{max}$. Write a MATLAB program to create a 4-by-1 plot like Example 2 where you plot the original continuous time signal $x_a(t)$, the sampled discrete time signal $x[n]$, the sampled discrete time signal superimposed over the original continuous time signal, and the sampled discrete time signal superimposed over the reconstructed signal (by replacing $f_1 = f_2 = f_3 = f_4 = f_s$).