**Registration#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

University of Engineering & Technology Lahore, FSD Campus

Experiment # 3

Title: Introduction to Sampling, reconstruction, and aliasing

Equipment Required: Personal computer (PC) with windows operating system and MATLAB software

Theory:

The course discusses discrete-time signal processing (DSP) in general that can be applied to any discrete-time sequences. Since MATLAB works on a digital system, we can only *simulate* the sampling of a continuous signal to produce a discrete time signal (i.e. we must approximate the continuous signal by a discrete time signal obtained with a high sampling rate). The same comment applies for A/D conversion and D/A conversions. But visual and audio representations of the aliasing are still possible.

In this lab, we will learn how to convert speech signal into discrete-time sequences using analog to-digital converter (ADC) available in the sound cards of every computer. A sound card can be regarded as a combination of an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC). Any sound card has at least the following inputs:

* Line-In
* Mic Input
* Speaker Output

Your sound card’s ADC samples the signal coming from Line-In and Mic Inputs, whereas the DAC converts samples to an analog signal and sends it to playback devices such as your laptop’s speaker. In this lab, we will learn how to sample a speech signal using MATLAB.

**Task 1**

Read the following webpage and write a MATLAB script to sample your own voice at 11025 samples per second and 8-bit width samples.

<https://www.mathworks.com/help/matlab/ref/audiorecorder.html>

**Task 2**

Play the recorded voice on your computer’s speakers using sound command in MATLAB at a DAC playback rate of 8000 samples per second with 8 bits per sample.

<https://www.mathworks.com/help/matlab/ref/sound.html>

**Questions**

1. Do you hear the same voice as yours? If not, why the voice played back on the speaker is different?

**Task 3**

1. *x*(*t*)=sin(2(*π/3)f t* )sampled at a frequency of *f s* produces the discrete time signal and reconstruct it by using zero order hold method. By varying the value of fs, it is possible to illustrate the effect of fs in DAC.
2. Vary the frequency fs and plot four different plots. Use *subplot* to put four plots on one screen. Explain why these plots are different to each other.
3. Can you predict the best fs to reconstruct the signal?
4. Write Zero order hold method drawbacks? Mention any other method to reconstruct the signal on MATLAB?

Script Help

f = ; % signal frequency

t = -(1/f):1/(4\*f):(1/f);

sig = sin(2\*(pi/3)\*f\*t);

fs = ; % sampling frequency

t\_full = -(1/f):1/(4\*fs):1/f;

sig\_full = sin(2\*(pi/3)\*(f)\*t\_full);

subplot(311); plot(t,sig,'b-','LineWidth',1); ylabel('Amplitude'); title('Signal to sample')

hold on

plot(t\_full,sig\_full,'rx','LineWidth',2,'MarkerSize',10); ylabel('Amplitude'); %title('Discrete time points kept')

%%% RECONSTRUCTION%%%

subplot(312); plot(t\_full,sig\_full,'rx','LineWidth',2,'MarkerSize',10); title('The sample points with a flat line (zero order) drawn between them'); hold on

subplot(312); stairs(t\_full,sig\_full,'b','LineWidth',2); ylabel('Amplitude')

subplot(313); stairs(t\_full,sig\_full,'b','LineWidth',2); ylabel('Amplitude'); title('The reconstructed signal with the sample points removed')

**Task 4**

Solve Example 1.4.2 by J.G. Proakis and D.G. Manolakis, Digital Signal Processing Principles, Algorithms, and Applications.

Use *subplot* to put all plots on one screen.

Why part (d) plot is similar to original signal?