COMP ENG 4TL4 – DIGITAL SIGNAL PROCESSING

Lab #1: Introduction to Digital Signals using MATLAB

Objectives:

1. To gain experience in creating, sampling, quantizing and analyzing digital signals within MATLAB.
2. To become more familiar with MATLAB programming.

Assessment:

- Your grade for this lab will be based on your ability to create and work with digital signals within MATLAB, and on your reporting of the results. The report should contain any mathematical calculations or derivations carried out, MATLAB plots of results with brief descriptions, the MATLAB code with which you obtained your results, and answers to specific questions below.
- Clearly label all plots and their axes (points for style will be deducted otherwise)
- DON'T COPY OTHERS BLINDLY!!
- Please attend the lab section to which you have been assigned.
- You should complete this lab with one lab partner. If there are an odd number of students, then one group of three will be created by the TA.
- Each pair of students should complete one lab report together, which is to be submitted one week from the date of the lab. Those students in the "At Home" section must submit their reports one week from the day on which they demonstrated their lab to Jeff Bondy.

Pre-lab:

- Carefully read through this lab description, so that you know what is required.
- Read through the lecture notes (and bring them with you) so that you know how to answer the questions.
- Familiarize yourself with the MATLAB commands that may be required for this lab – see the list at the end of this lab description for some hints.

1. Point sampling of a sinusoid:

The general form of a continuous-time sinusoid is:

$$x(t) = A\cos(2\pi ft + \phi).$$

a. Create a Matlab script that point-samples $x(t)$ at a sampling frequency f_s of 100 Hz, for $t = 0$ to 2 seconds. Using the MATLAB function `stem()`, plot the discrete-time sequence $x[n]$ versus sample number n obtained for each of the following sets of parameters:

	Amplitude - A	Frequency – f (Hz)	Phase - ϕ (rad.)
A	5	10	0
B	5	25	0
C	5	40	0
D	5	60	0
E	5	40	$\pi/2$
F	5	60	$\pi/2$

b. Visually determine the period of each of the discrete-time signals. Compare the discrete time signals' period and waveform with those of the continuous time signals. Explain the differences in the periods using your knowledge of the Nyquist sampling theorem.

c. Explain the effect of the phase change in cases C–F.

2. Working with unit impulses and unit steps

a. Create the unit impulse signal $\delta[n - 16]$ and the unit step signal $u[n - 12]$ for samples $n = 1, 2,\ldots, 30$, and plot both using the function `stem()`.

b. Plot the difference signal $x_1[n] = u[n - 14] - u[n - 15]$. Interpret this signal as $\delta[n - k]$ and find the value of k from the plot.

c. Plot another difference signal $x_2[n] = u[n - 9] - u[n - 16]$. Determine the width of the resulting rectangular "pulse". Express this signal as a sum of unit impulses.

3. Complex signals

Generate the following complex-valued discrete-time signal:

$$x[n] = Ae^{j\omega n}, \quad \text{for } n = 1, 2, \ldots, 40; A = 1; \text{ and } \omega = \pi/10.$$

a. Plot $x[n]$ in the complex-real plane (i.e., $\mathcal{I}m\{x[n]\}$ versus $\mathcal{R}e\{x[n]\}$), using the MATLAB command `plot()`.

b. Extract the real and imaginary parts of signal using the functions `real()` and `imag()` and plot them versus the sample number n using `stem()`. Use `subplot(2,1,1)` and `subplot(2,1,2)` commands prior to each `stem()` command to create plots of real and imaginary parts placed on the same figure but different axes.

c. How are the real and imaginary components related? Justify the phase difference between the real and imaginary components. Justify the number of cycles of the real and imaginary parts that appear.

d. Generate another figure with separate stem plots of the magnitude and phase of $x[n]$ versus sample number n. Does the plot of the magnitude make sense? What is the slope of the phase versus sample number curve, and how does this relate to the parameters used to generate $x[n]$?

4. Quantization of a speech signal

a. Load the supplied speech signal into MATLAB using the command:

```
[y,fs,nbits] = wavread('oilyrag.wav');
```

b. Plot the speech waveform `y` using the `plot()` command, and plot a histogram of amplitude values using the command `hist(y,50)`. Describe the waveform and the shape of the histogram in terms of: i) the number of bits `nbits` at which the .wav file was quantized and ii) the typical p.d.f. of speech.

c. Listen to `y` using the `soundsc()` command. Describe the sound quality.

d. Write a MATLAB script for a 3-bit rounding uniform quantizer.

e. Scale the speech signal `y` so that no saturation ("peak clipping") will occur when it is passed through the quantizer – call this new variable `y_scaled`.

f. Quantize the scaled signal `y_scaled` (call the quantized signal `y3bit`) and repeat parts b and c above, but for the newly quantized signal, comparing the new results to the results from parts b and c. In addition, plot the waveform and histogram for the error `e = y_scaled – y3bits`, and describe this in terms of the standard statistical model of quantization noise.

g. Rescale `y` to greatly exceed the maximum scale range of your 3-bit quantizer, such that substantial peak clipping will occur – call this new variable `y_pclip`. Quantize this rescaled signal (call it `y3bit_pclip`) and repeat part f above, but for the peak clipped quantized signal. Again, compare the new results to the previous results.

Potentially useful MATLAB commands

Note that this is not an exhaustive list! You are not required to incorporate all of these in your scripts.

help <topic>	helpwin	figure	plot	stem
hist	subplot	hold on	xlabel	ylabel
legend	title	function	clear	close
clc	zeros	ones	cos	exp
abs	round	max	min	find
if	for	end	real	imag
angle	unwrap	phase	wavread	wavwrite
soundsc				