SIGNALS AND SYSTEMS LABORATORY 2:
Sampling and Frequency Analysis

INTRODUCTION

In this lab, you will learn the basic and intermediate functions of MATLAB and apply them to the theory of Sampling and Analysis. Our objectives in this lab are to

1. Understand MATLAB
   a. Create a sinusoidal wave
   b. Compute the Discrete Fourier Transform
   c. Visualize Data in time domain and frequency domain
   d. Short-Time Fourier Transform (STFT)
   e. Creating Scripts with Matlab to sample and analyze a short speed signal

2. Practical Applications:
   a. Dual-Tone Multi-Frequency (DTMF) Signal Detection

SIGNALS AND SYSTEMS LABORATORY 1:
Sampling and Frequency Analysis

Assignments will be collected in Lab #3. Work in group is allowed.

Experiment #1: Discrete Fourier Transform

A common use of Fourier transforms is to find the frequency components of a signal buried in a noisy time domain signal. In this exercise we will create a sound signal sampled at 1000 Hz containing 50 Hz and 120 Hz and corrupt it with some zero-mean random noise.

- Use SIN function to create a 5 second sound signal with 50Hz and 120 Hz

  t = 0:0.001:5; number of sampling points at 1000Hz for 5 seconds

  x = sin(2*pi*50*t)+sin(2*pi*120*t); Create a sound signal with 50Hz and 120Hz

- Use RANDN to create random noise to be added to the sound signal.
  y = x + 2*randn(size(t));
- Use PLOT command to plot the corrupted sound signal

```
plot(1000*t(1:50),y(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
```

- It is difficult to identify the frequency components by looking at the original signal. Using discrete Fourier transform to convert to the signal to frequency domain. We will perform a 512-point fast Fourier transform (FFT), calculate and plot the power spectrum.

  1. Calculate FFT
     Y = fft(y,512);
  2. Calculate power spectrum (a measurement of the power at various frequencies)
     Power spectrum = Y.* conj(Y) / 512;
  3. Graph the first 257 points (the other 255 points are redundant) on a meaningful frequency axis:
     ```
     f = 1000*(0:256)/512;
     plot(f,Pyy(1:257))
     title('Frequency content of y')
     xlabel('frequency (Hz)')
     ```

- Use specgram command to displays a spectrogram of the data.

Experiment #1: **Practical Applications**

Part 1: Dual-Tone Multi-Frequency (DTMF) Signal Detection

Dual-tone Multi-Frequency (DTMF) signaling is the basis for voice communications control and is widely used worldwide in modern telephony to dial numbers and configure switchboards. It is also used in systems such as in voice mail, electronic mail and telephone banking. This exercise will make use of the Goertzel function as a part of a DFT-based DTMF detection algorithm. In this exercise, we will

- Generating DTMF Tones
- Playing DTMF Tones
- Estimating DTMF Tones with the Goertzel Algorithm
- Detecting DTMF Tones

Part 2: Simple voice recognition application
The objective of the following exercise is to develop a digital voice recognition program utilizing MatLab that will be able to identify people uniquely by their voice. The characteristic properties of one's voice is a function/product of many parameters including the dynamics of sound as it passes through the pharynx, the vibration of the larynx, the shape of the mouth and how sound reverberates off of the palette. Due to the uniqueness complexity of one's voice, a voice has fingerprint qualities, i.e. ***no two people's voice patterns are exactly alike.*** One simple method is to make use of the FFT to map the frequency fingerprint of a person's voice, in order to systematically determine if another sample is or is not from the same subject.

We will gather the FFT fingerprint from your voice and compares it to others in order to determine if the subject's voice matches. This exercise is divided into five sections:

- Record sound signal: Record the voice of the subject, checks/converts the data to mono from stereo.
- Filter: Filters the raw data with high pass, low pass, bandpass, bandstop, or no filter.
- FFT: Compute the spectral density using FFT.
- Peak finder: Analyzes filtered data and picks out peaks via a complex downward scan of the FFT up-to a preset threshold value.
- Peak Compare: Determine the peaks. The peaks are then compared against archived voice fingerprints.