# ECE 417 Lab Exercise # 1
# Signals and Systems: Key Matlab DSP commands

In this lab, you will work on simple Matlab exercises designed to familiarize you with common commands used in digital signal processing. You will learn to generate sinusoidal signals and design FIR filters using the *firpm* routine. You will learn to load speech signals, listen to them, and do some simple speech analysis. You will also learn to load images, view them, and do some simple manipulations. Most of the Matlab programming will be provided to you, and you will re-use common commands with possibly different parameters.

The first thing you need to do is to note the last three non-zero digits of your UIC ID #, say $i, j, k$. You will be required to use the number $N_{ID} = ijk = 100 * i + 10 * j + k$ in many lab exercises.

## 1 Discrete-time signals and systems: DTFT display and FIR filter design

### 1.1 Generation of sum of two sinusoidal sequences

Define frequencies: $\omega_1 = 0.08 * \pi + 0.30 * N_{ID} * \pi/1000$ and $\omega_2 = \pi - \omega_1$. Generate the sequence $x[n] = \cos(\omega_1 n) + 0.5\cos(\omega_2 n)$ using program Plab1_1_SinusoidalSignals.m

```
% Plab1_1.m Generation of sum of two sinusoidal sequences
%
% Plab1_1.m Generation of sum of two sinusoidal sequences
% Plotting the DFT and DTFT
clf;
NID = input('Type in N_ID = ');
w1 = 0.15*pi + 0.2*NID*pi/1000;
w2 = pi - w1;
disp('Frequency w1 = '), disp(w1);
disp('Frequency w2 = '), disp(w2);
Nsamples = 101;
n = 1:Nsamples;
```

```
x = 2*cos(w1*n) + cos(w2*n);

Xfft = fft(x);
Xmax1 =max(abs(Xfft));
w = (n-1)*2*pi/Nsamples;
[Xdtft, wgrid] = freqz(x,1,Nsamples);
Xmax2 =max(abs(Xdtft));
stem(n-1, x(n));
xlabel('Sample number n');ylabel('Signal value x');
axis([0 Nsamples -2 2]);
grid;
title('Sum of sinusoids');
disp('PRESS RETURN for magnitude of DFT and DTFT of x');
pause
Xdirect = zeros(1,Nsamples);%direct computation of DTFT at Nsamples points
for k=1:Nsamples
    tem1 = exp(1i*pi*(n-1)*(k-1)/Nsamples);
    Xdirect(k) = sum(x.*tem1);
end
w = (n-1)*pi/Nsamples;
plot(w,abs(Xdirect));
xlabel('frequency \omega');ylabel('DTFT magnitude');
title('Directly computed DTFT versus frequency \omega');
pause
subplot(3,1,1)
stem(n-1, abs(Xfft));%Plot the magnitude of DFTof x
xlabel('frequency index k');ylabel('DFT magnitude');
axis([0 Nsamples-1 0 1.2*Xmax1]);
grid;
title('DFT versus index k');
subplot(3,1,2)
stem(w, abs(Xfft));%Plot the magnitude of DFTof x
xlabel('frequency');ylabel('DFT magnitude');
title('DFT versus discretized frequency \omega');
axis([0 2*pi*(Nsamples-1)/Nsamples 0 1.2*Xmax1]);
grid;
```

```
subplot(3,1,3)
plot(wgrid, abs(Xdtft));%Plot the magnitude of DTFTof x
xlabel('frequency \omega');ylabel('DTFT magnitude');
title('DTFT versus frequency \omega');
axis([0 2*pi*(Nsamples-1)/Nsamples 0 1.2*Xmax1]);
grid;
```

Q1.1: What are the values of $\omega_1$ and $\omega_2$ for the sinusoids you generated?

Q1.2: Compare the last three plots you obtained and describe their relation.

### 1.2 Design a filter using firpm

In Matlab, type "help firpm" to understand the input format for the command to design a FIR filter using the Parks-McClellan program. Run Plab1_2_FIRFilterDe

```
% Plab1_2.m Linear-Phase FIR filter design
%
% Plab1_2_ECE417.m Linear-Phase FIR filter design
%
clf;
NID = input('Type in N_ID = ');
fp=0.31+0.14*NID/1000;
fs=1-fp;
Nord=10;% choose even order
h = firpm(Nord, [0 fp fs 1], [1 1 0 0], [1 1]);
[H, w] = freqz(h,1,100);
Hmax=max(abs(H));
plot(w, abs(H));%Plot the magnitude response of filter
xlabel('frequency');ylabel('DFT magnitude');
title('Magnitude of H versus frequency');
axis([0 pi 0 1.2*Hmax]);
grid;
disp('PRESS RETURN for pole-zero plot of H');
pause
```

```
roots(h)
zplane(h);
```

Q1.3: Display the stem plot of $h$, with a proper shift, such that the phase is zero in the passband. Also plot the frequency response using the command *freqz*.

Q1.4: Show the plot of the poles and zeros of $H(z)$ using the *zplane* command.

Q1.5: Use the *firpm* command to design a bandpass filter of minimum even order such that it meets the requirements:
first stopband $0 \leq \omega \leq 0.25\pi$ with peak magnitude deviation $\leq 0.025$
passband $0.35\pi \leq \omega \leq 0.65\pi$ with peak magnitude deviation $\leq 0.1$
second stopband $0.75\pi \leq \omega \leq \pi$ with peak magnitude deviation $\leq 0.05$
Plot the frequency response using the command *freqz*. What is the minumum filter order that you needed?

# 2 Speech signal Display and Analysis

## 2.1 Speeech Signal display and analysis

Display the speech signal *ece417_audio.wav* stored as a wav-file. Identify (as closely as you can) the start-point and end-point indexes of the word-string "E C E Four Seventeen". Run Plab1_3_Speech_Display_Analysis.m and listen to the sounds as noted below.

```
% Plab1_3_ECE417.m Updated 2020-09-06
% Display and examine a speech signal
[y,Fs] = audioread('ece417_audio.wav');
fprintf('Sampling frequency = %.1f Hz\nSampling Period = %.2f microsec\n'
clf
soundsc(y,Fs)
plot(y);
title('Speech signal');
xlabel('Time sample index n')
ylabel('Amplitude y[n]')
pause
```

```
plot(y(12501:19000)); %Plot of the second E sound in ECE
title('Sound "E"');
xlabel('Time sample index n')
ylabel('Amplitude y[n]')
soundsc(y(12501:19000),Fs); % "E"
pause

plot(y(15210:15620)); %Plot of short segment the second E sound in ECE
title('A few pitch periods of Sound "E"');
xlabel('Time sample index n')
ylabel('Amplitude y[n]')
pause

h=firpm(20,[0 0.01 0.15 1.0],[0 0 1 1],[1 1]); %highpass filter IR to remo
yf = conv(y,h); %highpass filter output y with DC content removed
soundsc(10*y(5601:7600),Fs); % "s"

plot(yf(5601:7600))
title('Speech segment with sound "s" with DC content removed');
xlabel('Time sample index n')
ylabel('Amplitude y[n]')
pause

subplot(2,1,1)
y1 = y(12501:19000); %voiced segment of y
[Y1,W]=freqz(y1); % Y1 is DTFT of y1
plot(W,abs(Y1));
xlabel('Frequency \omega')
ylabel('Magnitude of DTFT')
title('Voiced sound frequency content');
subplot(2,1,2)
y2 = yf(5601:7600); %unvoiced segment of y with DC removed
[Y2,W]=freqz(y2); % Y2 is DTFT of y2
plot(W,abs(Y2));
xlabel('Frequency \omega')
```

```
ylabel('Magnitude of DTFT')
title('Unvoiced sound frequency content');
```

Q1.6 What is the sampling rate of the speech signal?

Q1.7 Compare the frequency content of the voiced and unvoiced parts of the utterance “s” and “E” described above.

Q1.8 Determine the pitch of the utterance “E” from the plot of a short segment given in the Matlab code.

### 2.2 Speech signal denoising

In this part you will perform speech processing to remove noise from a speech signal *'noisy_speech.wav'*. The noisy speech is created by adding noise to the clean signal *'ece417_audio.wav'* used in part 2.1. The noise in the noisy speech is primarily added in the frequency region $[0.25\pi \;\; \pi]$. In this exercise you will remove the noise from (or denoise) the noisy speech using an FIR filter.

Design a suitable FIR filter of order 30 using *firpm* to denoise the speech. Choose appropriate care-bands and desired filter magnitude.

Write Matlab code to read the noisy speech file, and process it with the denoising filter you designed. Show the code in your report.

Q1.9 Listen and compare the denoised speech to the noisy speech and report the results. Also listen and compare denoised speech to the clean original speech *'ece417_audio.wav'* that you used in section 2.1 earlier. Report your observations.

Lab exercise written by R. Ansari. Updated Fall 2023.