Lab 4: Signal Analysis Using the Discrete Fourier Transform

Professor: Dr. Mohamed Elamien
Names: Abdurahman Butt and Erion Keka
Student Numbers: 400435085 and 400435050
Submission Date: 2024-12-05

Part 1:

Step 2: Upon listening to the audio clip, we hear a constant high-pitched frequency.

Step 3:

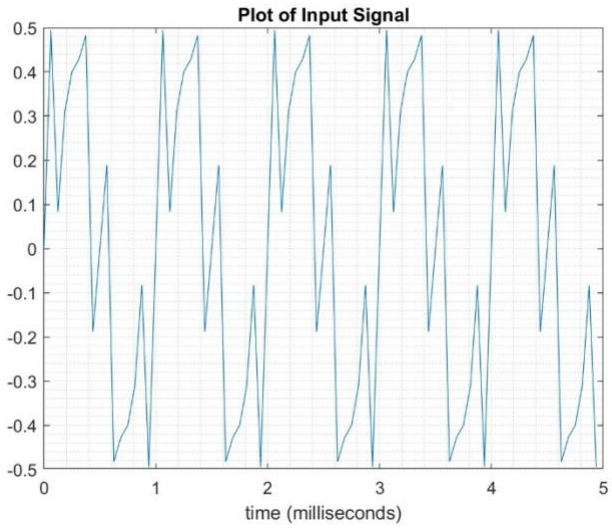
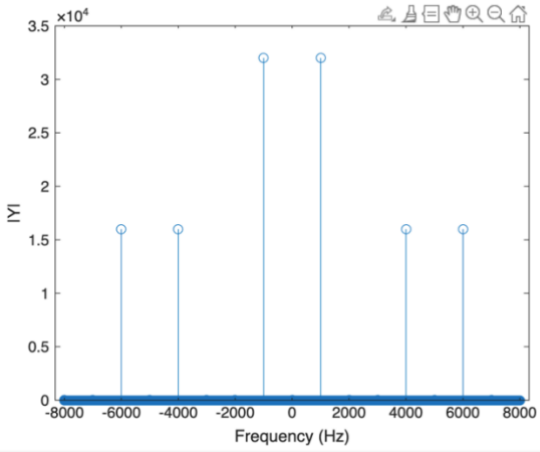


Figure 1: Plot from Step 3

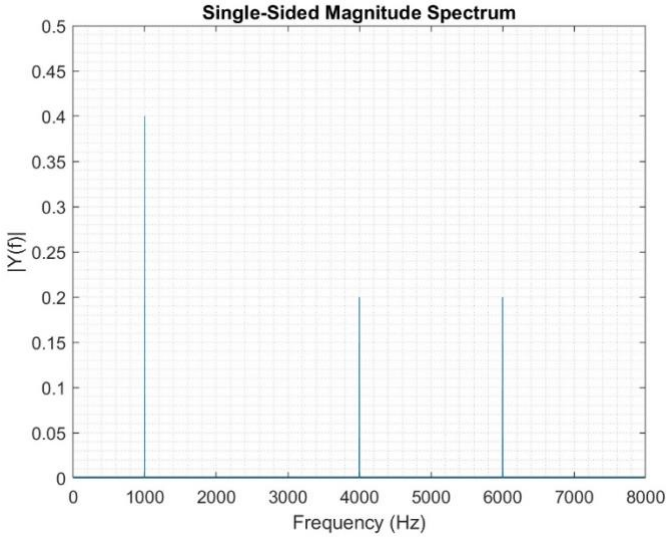
Step 4: Based the graph generated in Question 2, we can note that the output is periodic where one cycle occurs every 1mS. Within one cycle, we can see that there are three peaks meaning that there are three dominant frequencies. To conclude, we can estimate that there are three sinusoids that create the signal.

Step 5:

```
[signal, Fs] = audioread("tones.wav");
L = length(signal)'
T = 1/Fs;
t = [0:L-1]*T;

Y = fft(signal);
Z = fftshift(Y);


LY = length(Y);
f = (-LY/2:LY/2-1)/LY*Fs;

stem(f,abs(Z))
xlabel 'Frequency (Hz)'
ylabel '|Y|'
grid off;
```



Step 6:



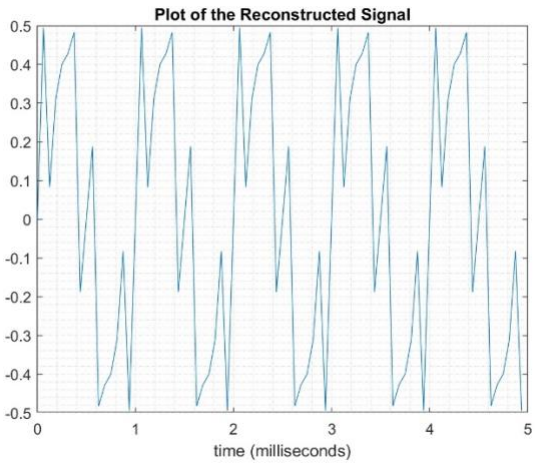
Step 7: From the graph results obtained in Step 6, we have the following frequencies and magnitudes:

a. Frequency: $f_1 = 1000$ Hz, Magnitude: $|Y(f_1)| = 0.4$
b. Frequency: $f_2 = 4000$ Hz, Magnitude: $|Y(f_2)| = 0.2$
c. Frequency: $f_3 = 6000$ Hz, Magnitude: $|Y(f_3)| = 0.2$

Step 8: Making use of the code shown below and comparing the output to the results of Part 3, we can see that the new output is the same as the output in Part 3.

Step2.m

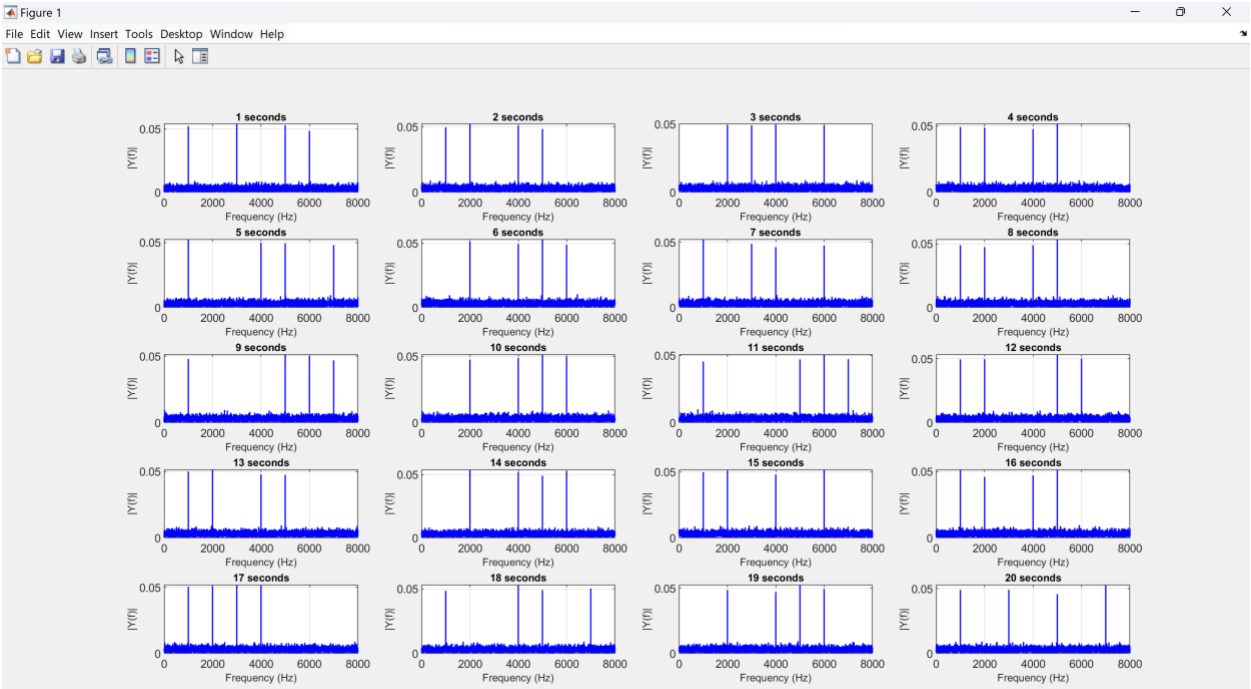
```
% Read in the signal from the audio file
[signal, Fs] = audioread("tones.wav");
L = length(signal);
T = 1/Fs;
t = [0:L-1]*T;

% Plot the signal for t_plot msec
t_plot = 5;
msec_per_sec = 1000;

% Adding Signals
signal1 = 0.4*sin(2*pi*1000*t);
signal2 = 0.2*sin(2*pi*4000*t);
signal3 = 0.2*sin(2*pi*6000*t);

outputsignal = signal1+signal2+signal3;

numSamples = t_plot*Fs/msec_per_sec;
plot(msec_per_sec*t(1:numSamples), signal(1:numSamples))
title('Plot of the Reconstructed Signal')
xlabel('time (milliseconds)')
grid('minor');
```

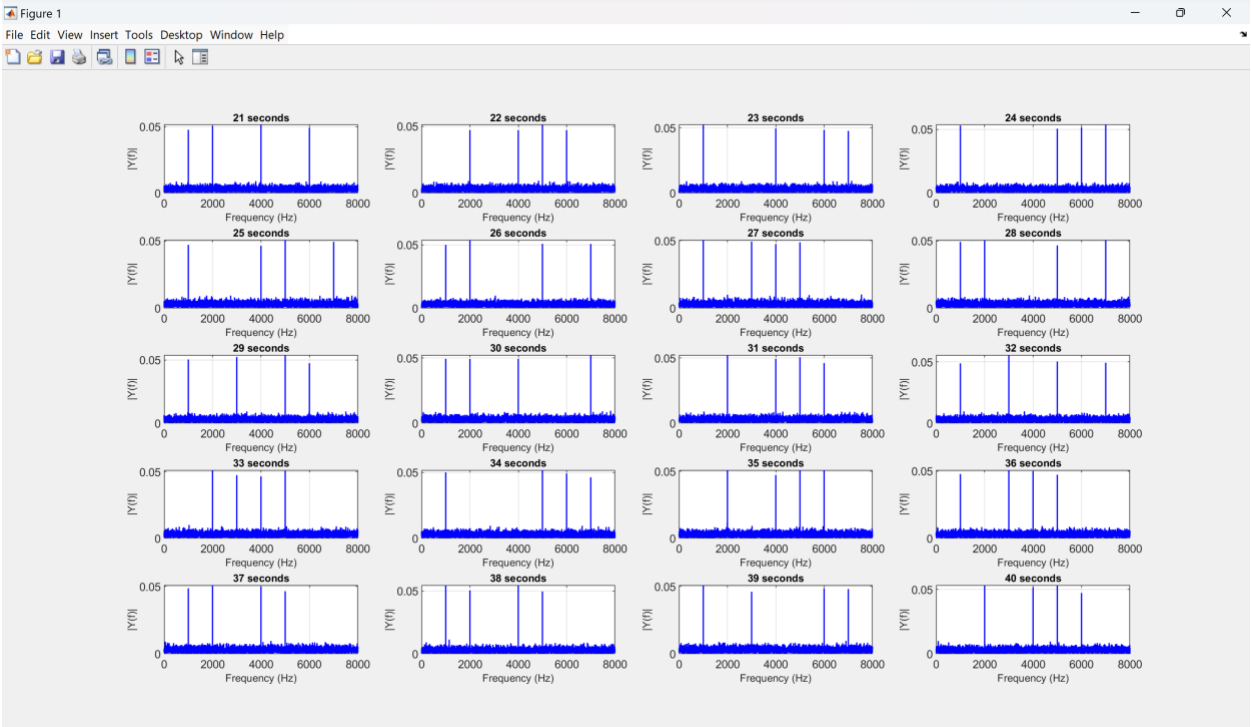


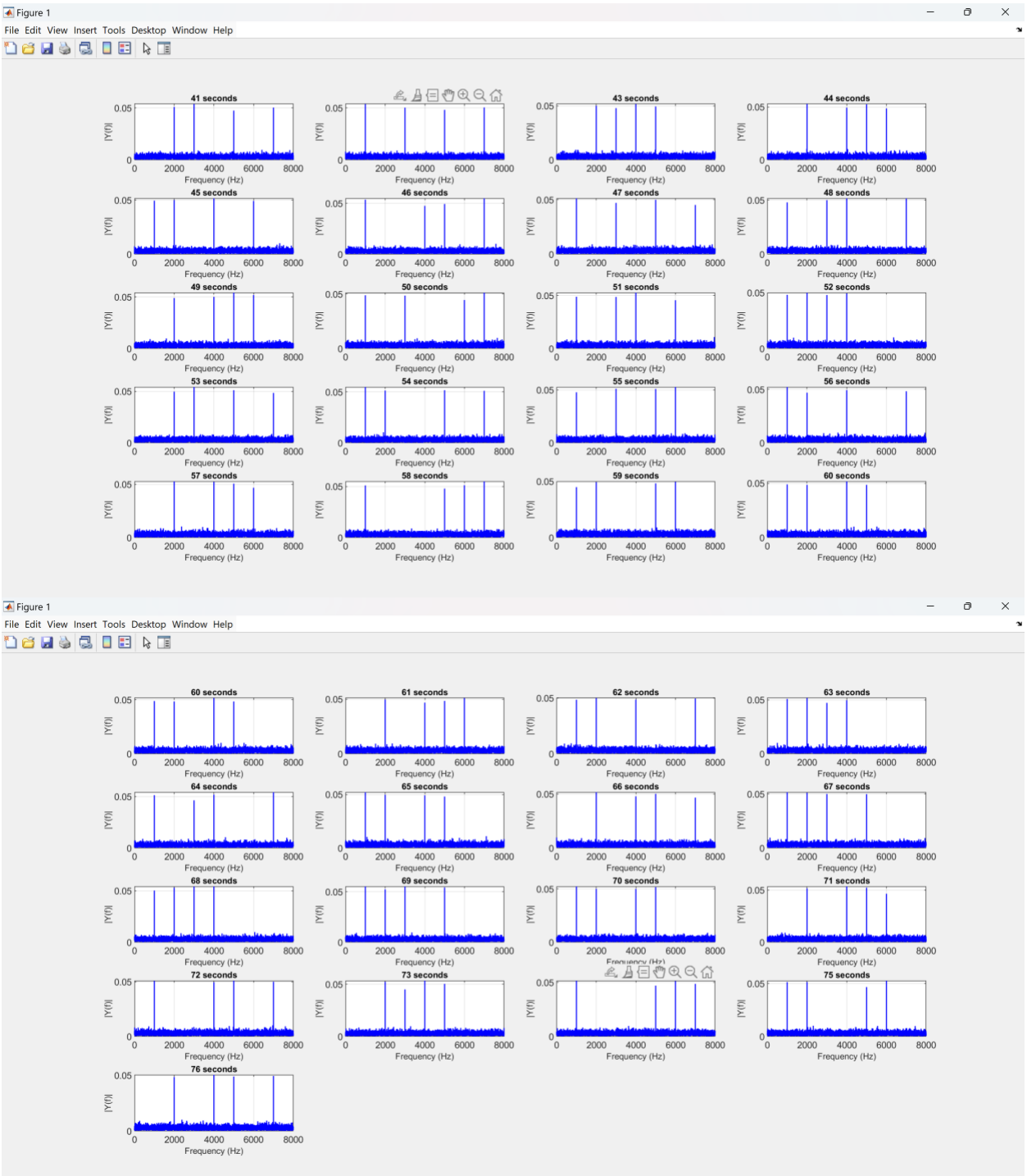
Part 2:

Step 2: Upon listening to the secret message audio clip, we hear a constant static noise.

Step 3-4:







Using the codebook as a reference, we have determined that the code is as follows:

“NEVER LET THE FEAR OF STRIKING OUT KEEP YOU FROM PLAYING THE GAME. BABE RUTH.”

Code for Part 2:

```
[audioSignal, samplingRate] = audioread('SecretMessage.wav');
signalLength = length(audioSignal);
timeStep = 1 / samplingRate;
k=1;

for i = 60:81 %change this to view different interval of audio clip (must be less than 20)
    start = 1 + (i-1)*samplingRate;
    ending = i * samplingRate;

    window_signal = audioSignal(start:ending);

    fftResult = fft(window_signal) / samplingRate;

    [~,frequencyIndices] = maxk(abs(fftResult(1:samplingRate/2+1)),4);
    dominant_freqs = frequencyIndices-1;

    frequency_axis = samplingRate/2*linspace(0,1,samplingRate/2+1);
    subplot(5,4,k);
    plot(frequency_axis,2*abs(fftResult(1:samplingRate/2+1)),'b-','LineWidth',1);
    title(sprintf('%d seconds', i), 'FontSize', 8);
    xlabel('Frequency (Hz)','FontSize',8);
    ylabel('|Y(f)|','FontSize',8);
    xlim([0,samplingRate/2]);
    grid on;
    k = k+1;
end
```