ELECENG 3TP3 – Signals and Systems

Lab 4: Signal Analysis Using the Discrete Fourier Transform

Instructor: Dr. Kiruba

Prepared and submitted by
Marryam Kamal – kamalm18 – 400446997

December 5th, 2024

Table of Contents

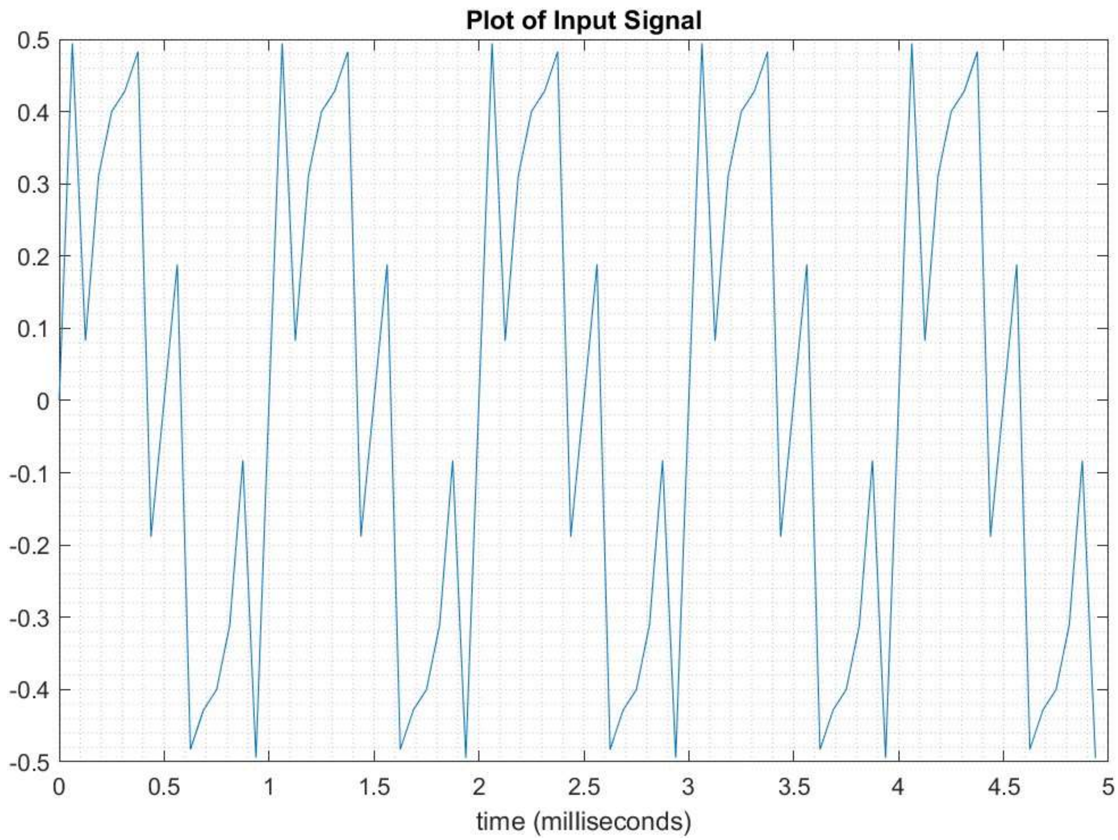
Part 1 – tones.wav 3
Part 2 – SecretMessage.wav..8

Part 1 – tones.wav

1) Qualitative description of what's heard in the audio file

Just listening to the tone, I can hear a constant sound. Though it's obvious that it's not a pure frequency due to how distorted it sounds.

2) Waveform of the signal



3) Estimating the sinusoids and their frequencies

Based on the graph, I can conclude that the signal consists of 3 different frequencies because there are 3 peaks in the signal that repeat periodically. Though without a Fourier transform, it's difficult to estimate exactly what these frequencies are.

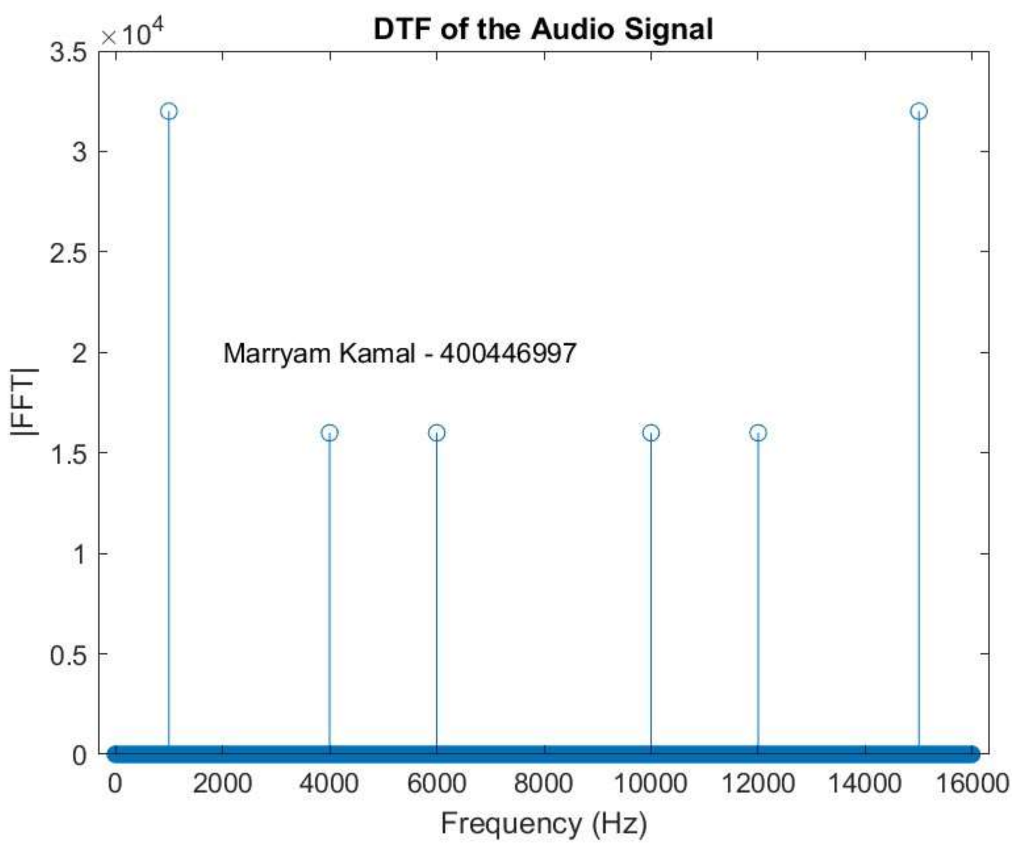
4) Discrete Fourier transforms of the signal

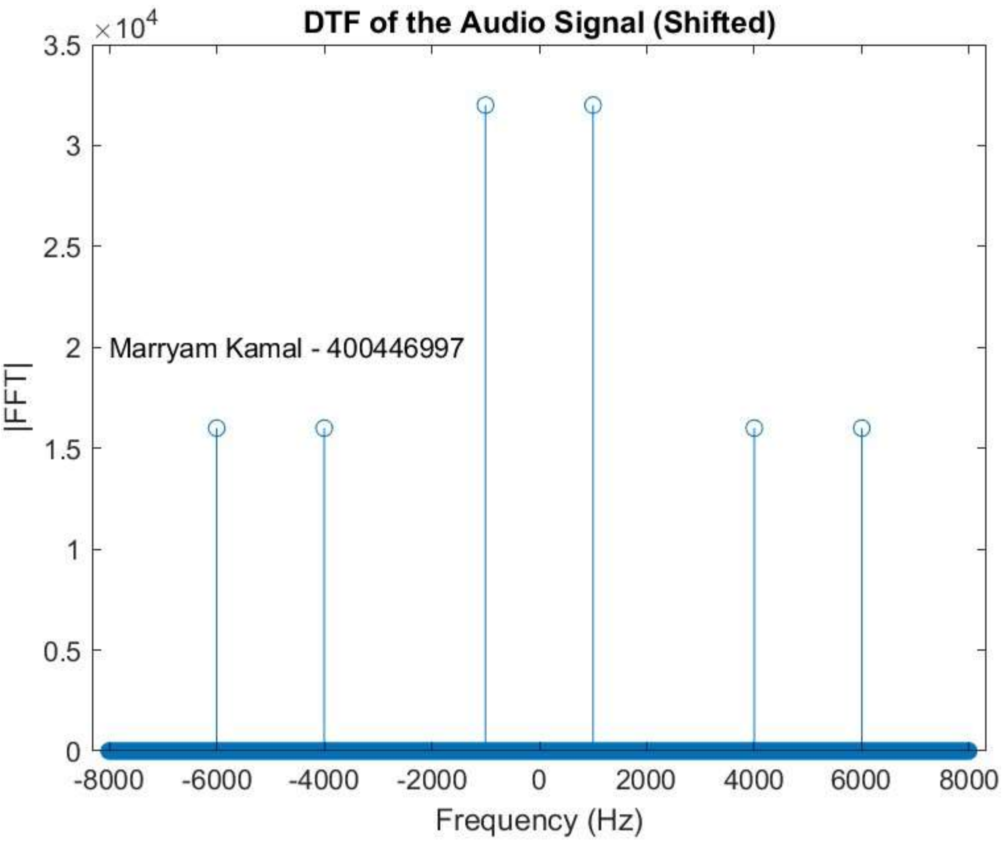
Code:

```
[signal, Fs] = audioread("tones.wav");
L = length(signal);

% Raw Signal
fftSignal = abs(fft(signal)); % Compute FFT
f = (0:L-1) * (Fs/L); % Frequency vector (unshifted)

% Plotting
```

```
stem(f, fftSignal);
title('DTF of the Audio Signal');
xlabel('Frequency (Hz)');
ylabel('|FFT|');
text(2000, 20000,'Marryam Kamal - 400446997', 'FontSize', 10)
exportgraphics(gcf, 'tones_magnitude.jpg');

% Shifted Signal
fftShifted = abs(fftshift(fft(signal))); % Center FFT around zero
fShifted = (-L/2:L/2-1) * (Fs/L); % Frequency vector (shifted)

% Plotting
stem(fShifted, fftShifted);
title('DTF of the Audio Signal (Shifted)');
xlabel('Frequency (Hz)');
ylabel('|FFT|');
text(-8000, 20000,'Marryam Kamal - 400446997', 'FontSize', 10)
exportgraphics(gcf, 'tones_magnitude_shifted.jpg');
```

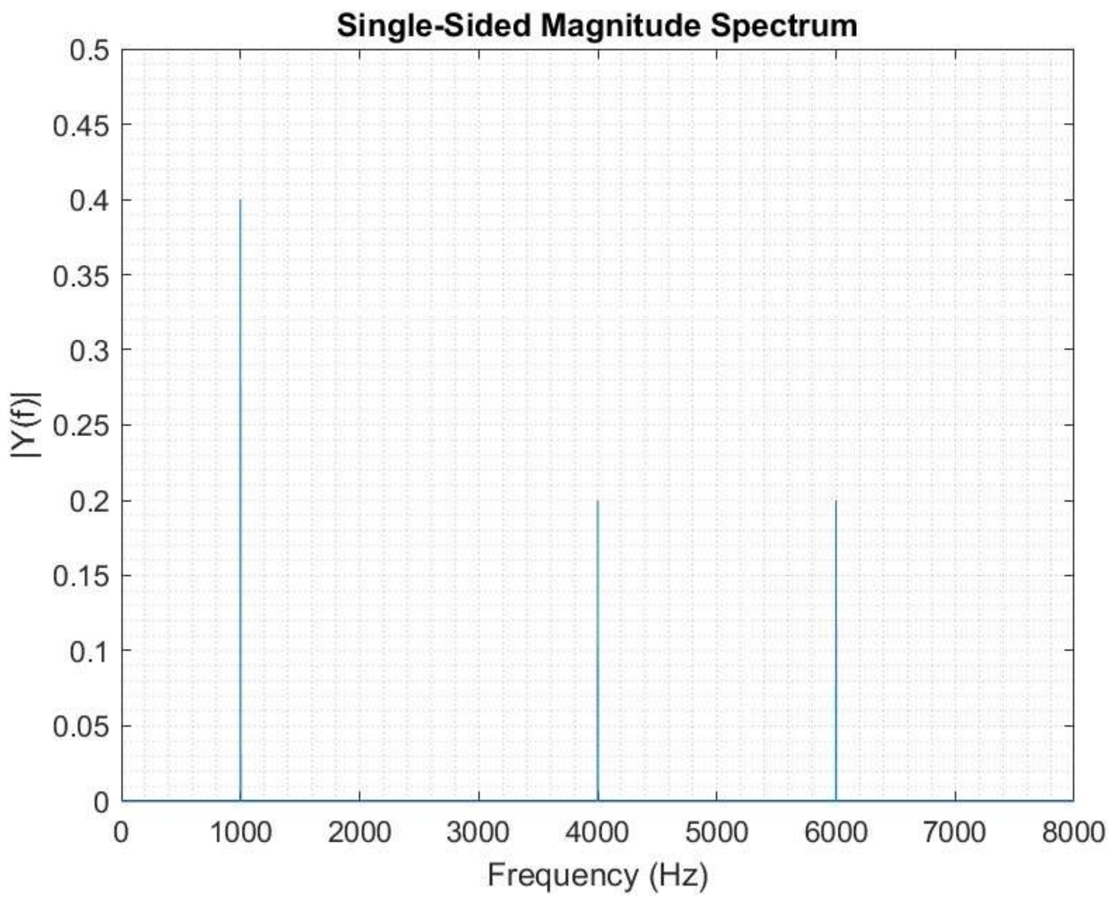
Plots:





Both the unshifted and shifted graphs are representations of the discrete Fourier transforms of the signal, but fftshift changes how the frequencies are displayed. By moving the zero-frequency component to the center, it allows us to see exactly how many frequencies are present in the signal.

5) Single-sided magnitude spectrum



6) Frequencies and magnitudes in the audio signal

Sinusoid	Frequency	Magnitude
Sinusoid 1	1 kHz	0.4
Sinusoid 2	4 kHz	0.2
Sinusoid 3	6 kHz	0.2

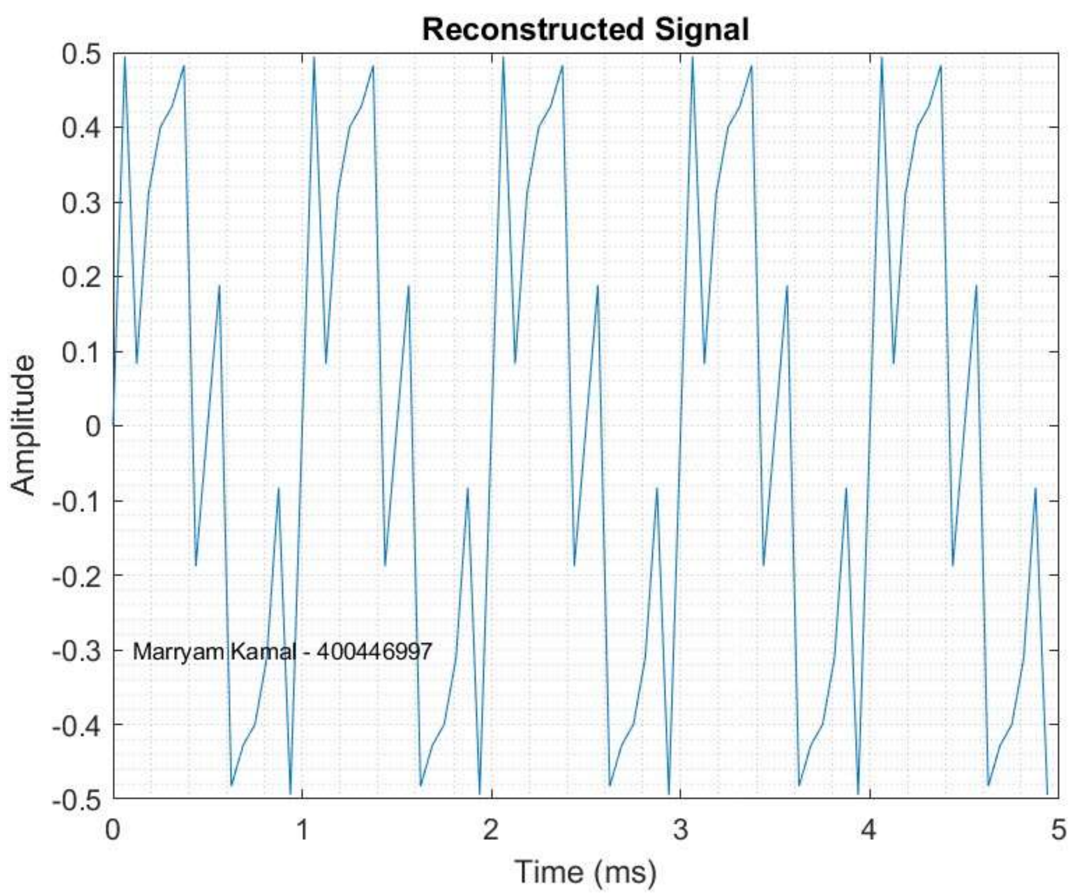
7) Regenerating the original signal from the calculated frequencies

Code:

```
% Define parameters from previous part
[signal, Fs] = audioread("tones.wav");
frequencies = [1000, 4000, 6000];
magnitudes = [0.4, 0.2, 0.2];

% Time vector for 5 msec
L = length(signal);
T = 1/Fs;
t = [0:L-1]*T;
t_plot = 5;
msec_per_sec = 1000;
numSamples = t_plot*Fs/msec_per_sec;
```

```
% Generate the signal
new_signal = zeros(size(t));
for i = 1:length(frequencies)
    new_signal = new_signal + magnitudes(i)*sin(2*pi*frequencies(i)*t);
end

% Plotting
plot(msec_per_sec*t(1:numSamples), new_signal(1:numSamples));
title('Reconstructed Signal');
xlabel('Time (ms)');
ylabel('Amplitude');
grid('minor');
text(0.1, -0.3,'Marryam Kamal - 400446997', 'FontSize', 8);
exportgraphics(gcf, 'recreated_signal.jpg');
```

Plot:



As expected, the regenerated plot matches perfectly with the original audio signal prior to the Fourier transform.

Part 2 – SecretMessage.wav

1) Qualitative description of what's heard in the audio file

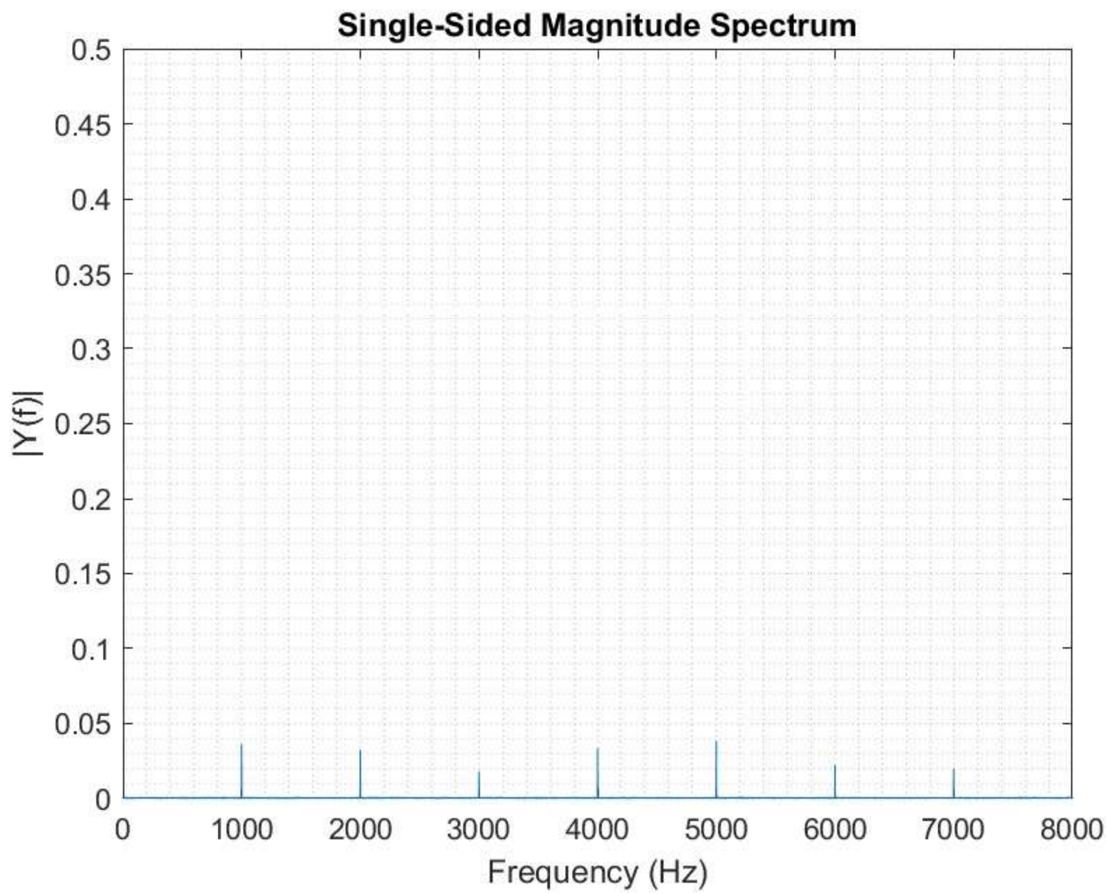
In the audio file, I heard a lot of static noise with a faint tone in the background. This tone appeared to change every second, however, there isn't enough information to decipher each tone separately.

2) Frequencies in the audio file

Code:

```
% Take the DFT
[signal, Fs] = audioread("SecretMessage.wav");
L = length(signal);
Y = fft(signal)/L;
f = Fs/2*linspace(0,1,L/2+1);

% Plot the single-sided magnitude spectrum.
plot(f,2*abs(Y(1:L/2+1)));
title('Single-Sided Magnitude Spectrum');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
axis([0 Fs/2 0 .5]);
grid('minor');
exportgraphics(gcf, 'secret_message.jpg');
```

Plot:



Frequencies used in the signal:

- 1000 Hz
- 2000 Hz
- 3000 Hz
- 4000 Hz
- 5000 Hz
- 6000 Hz
- 7000 Hz

3) Decoding the message

Code:

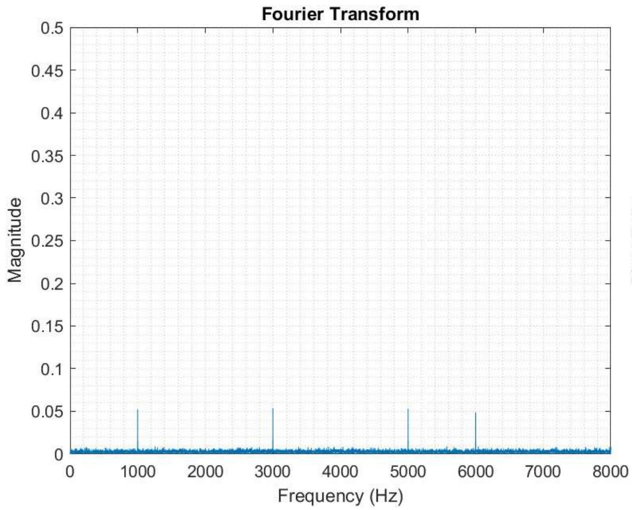
```
function peaks_cell = get_peak_frequencies(audio_file)
    % Read the audio file
    [signal, Fs] = audioread(audio_file);

    peaks_cell = {};
    N = floor(length(signal)/Fs);

    % Loop over each second of the audio
    for n = 1:N
        % Break the signal into the corresponding second
```

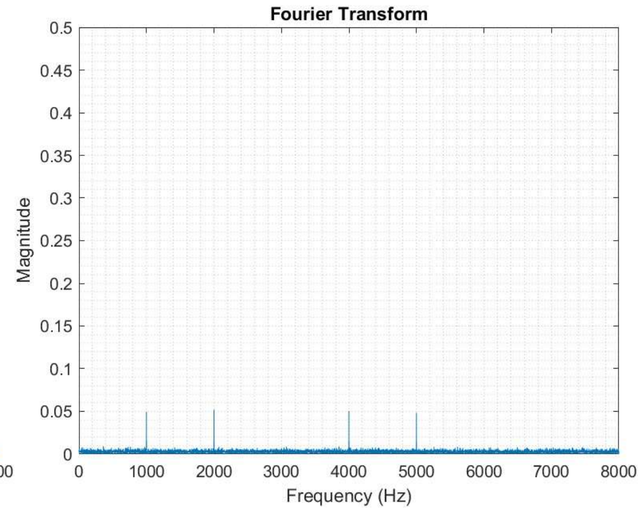
```
        start_sample = (n - 1)*Fs + 1;
        end_sample = n * Fs;
        signal_second = signal(start_sample:end_sample);
        L = length(signal_second);

        % Take the DFT
        Y = fft(signal_second)/L;
        f = Fs/2*linspace(0, 1, L/2 + 1);
        magnitude = 2*abs(Y(1:L/2 + 1));

        % Find the peaks in the magnitude spectrum
        [peaks, locs] = findpeaks(magnitude);
        valid_locs = locs(peaks > 0.01); % Only keep peaks with magnitude > 0.01
        peaks_cell{n} = f(valid_locs); % Store the frequencies of the valid peaks
    end

    % Display the peak frequencies for each second
    for n = 1:N
        fprintf('Second %d: ', n);
        fprintf('%.f Hz ', peaks_cell{n});
        fprintf('\n');
    end
end

peak_frequencies = get_peak_frequencies('SecretMessage.wav');
```

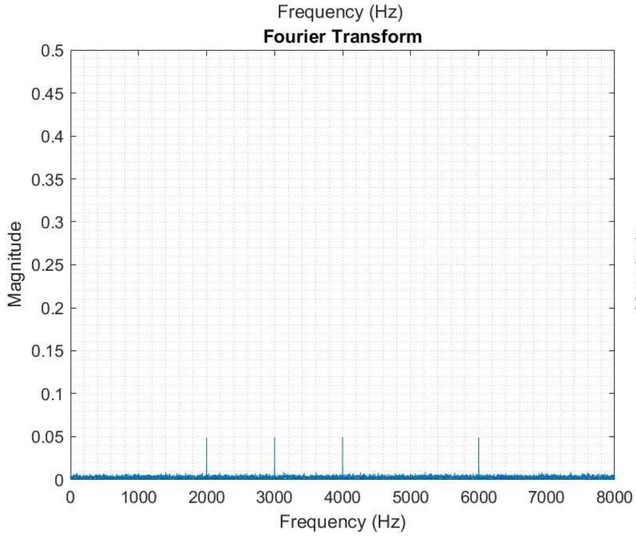
Output:

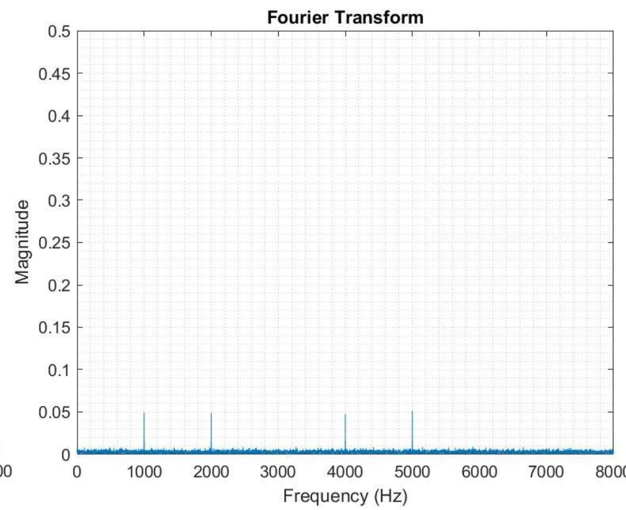
```
Second 1: 1000 Hz 3000 Hz 5000 Hz 6000 Hz
Second 2: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 3: 2000 Hz 3000 Hz 4000 Hz 6000 Hz
Second 4: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 5: 1000 Hz 4000 Hz 5000 Hz 7000 Hz
Second 6: 2000 Hz 4000 Hz 5000 Hz 6000 Hz 6447 Hz
Second 7: 1000 Hz 3000 Hz 4000 Hz 6000 Hz
Second 8: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 9: 1000 Hz 5000 Hz 6000 Hz 7000 Hz
Second 10: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 11: 1000 Hz 5000 Hz 6000 Hz 7000 Hz
Second 12: 1000 Hz 2000 Hz 5000 Hz 6000 Hz
Second 13: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 14: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 15: 1000 Hz 2000 Hz 4000 Hz 6000 Hz
Second 16: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 17: 1000 Hz 2000 Hz 3000 Hz 4000 Hz
Second 18: 1000 Hz 4000 Hz 5000 Hz 7000 Hz
Second 19: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 20: 1000 Hz 3000 Hz 5000 Hz 7000 Hz
Second 21: 1000 Hz 2000 Hz 4000 Hz 6000 Hz
Second 22: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 23: 1000 Hz 4000 Hz 6000 Hz 7000 Hz
Second 24: 1000 Hz 5000 Hz 6000 Hz 7000 Hz
Second 25: 1000 Hz 4000 Hz 5000 Hz 7000 Hz
Second 26: 1000 Hz 2000 Hz 5000 Hz 7000 Hz
Second 27: 1000 Hz 3000 Hz 4000 Hz 5000 Hz 7544 Hz
Second 28: 1000 Hz 2000 Hz 5000 Hz 7000 Hz
Second 29: 1000 Hz 3000 Hz 5000 Hz 6000 Hz
Second 30: 1000 Hz 2000 Hz 4000 Hz 7000 Hz
Second 31: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 32: 1000 Hz 3000 Hz 5000 Hz 7000 Hz
Second 33: 1031 Hz 2000 Hz 3000 Hz 4000 Hz 5000 Hz
Second 34: 1000 Hz 5000 Hz 6000 Hz 7000 Hz
Second 35: 1994 Hz 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 36: 1000 Hz 3000 Hz 4000 Hz 5000 Hz
Second 37: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 38: 1000 Hz 1143 Hz 2000 Hz 4000 Hz 5000 Hz
Second 39: 1000 Hz 3000 Hz 6000 Hz 7000 Hz
Second 40: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 41: 2000 Hz 3000 Hz 5000 Hz 7000 Hz
Second 42: 1000 Hz 3000 Hz 5000 Hz 7000 Hz
Second 43: 2000 Hz 3000 Hz 4000 Hz 5000 Hz
Second 44: 1986 Hz 2000 Hz 4000 Hz 4046 Hz 5000 Hz 6000 Hz
Second 45: 1000 Hz 2000 Hz 4000 Hz 6000 Hz 7315 Hz
Second 46: 1000 Hz 4000 Hz 5000 Hz 7000 Hz 7590 Hz
Second 47: 1000 Hz 3000 Hz 5000 Hz 7000 Hz
Second 48: 1000 Hz 3000 Hz 4000 Hz 7000 Hz
Second 49: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 50: 1000 Hz 3000 Hz 6000 Hz 7000 Hz
Second 51: 1000 Hz 3000 Hz 4000 Hz 6000 Hz 7957 Hz
Second 52: 1000 Hz 2000 Hz 3000 Hz 4000 Hz 6364 Hz
Second 53: 2000 Hz 3000 Hz 5000 Hz 7000 Hz
Second 54: 1000 Hz 1915 Hz 2000 Hz 5000 Hz 7000 Hz
Second 55: 1000 Hz 3000 Hz 5000 Hz 6000 Hz
Second 56: 1000 Hz 2000 Hz 4000 Hz 7000 Hz
Second 57: 2000 Hz 2373 Hz 4000 Hz 5000 Hz 6000 Hz
Second 58: 1000 Hz 5000 Hz 6000 Hz 7000 Hz
Second 59: 1000 Hz 2000 Hz 5000 Hz 6000 Hz
Second 60: 1000 Hz 2000 Hz 4000 Hz 4361 Hz 5000 Hz 5517 Hz
Second 61: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 62: 1000 Hz 2000 Hz 4000 Hz 7000 Hz
Second 63: 504 Hz 1000 Hz 2000 Hz 3000 Hz 4000 Hz
Second 64: 1000 Hz 3000 Hz 4000 Hz 4582 Hz 7000 Hz 7672 Hz
Second 65: 1000 Hz 1081 Hz 2000 Hz 4000 Hz 5000 Hz 7105 Hz
Second 66: 2000 Hz 4000 Hz 5000 Hz 7000 Hz
Second 67: 1000 Hz 2000 Hz 3000 Hz 5000 Hz
Second 68: 1000 Hz 2000 Hz 3000 Hz 4000 Hz
Second 69: 1000 Hz 2000 Hz 3000 Hz 5000 Hz
Second 70: 1000 Hz 2000 Hz 4000 Hz 5000 Hz
Second 71: 2000 Hz 4000 Hz 5000 Hz 6000 Hz
Second 72: 1000 Hz 4000 Hz 5000 Hz 6964 Hz 7000 Hz
Second 73: 2000 Hz 3000 Hz 4000 Hz 5000 Hz
Second 74: 1000 Hz 5000 Hz 6000 Hz 7000 Hz
Second 75: 1000 Hz 2000 Hz 2314 Hz 5000 Hz 6000 Hz
Second 76: 2000 Hz 2389 Hz 4000 Hz 5000 Hz 7000 Hz
```

Plots: First 4 seconds of the audio. A similar graphing process was used as part 1 and repeated for all 76 seconds of the audio. To make things simpler, I condensed all the frequencies into a vector instead of graphs for easy readability (displayed above).









Final decoded message:

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