Filter Design Project
CSE 3313-001

Sean-Michael Woerner

1001229459

December 3, 2020

Table of Contents

Introduction

The purpose of this project is to design a filter that removes unwanted noise from an audio file. This will be accomplished by incorporating the paper design of a Butterworth lowpass filter. This deigned filter will then be implemented in MATLAB and used to filter a given audio file, comparing it to the unfiltered audio file.

This process will consist of the following steps:

1. Audio Frequency Analysis
2. Filter Design
3. Filter Implementation

Procedure

1. Audio Frequency Analysis

a) Use the audioread() MATLAB command to provide the sampling frequency ‘Fs’ of the file and read the contents of ‘noisyaudio.wav’ into an array.

b) Use the MATLAB fft() command to find the DFT of the audio sample.

c) Form a frequency axis for plotting the DFT. It should have the same number of elements as the input audio sample and span from -Fs/2 to Fs/2.

d) Plot the magnitude (abs) of the DFT vs frequency with ω=0 at the center of the plot. This can be accomplished using the fftshift() command. The plot should show energy in the speech frequencies from about 100 Hz to 2 kHz. The plot should also show significant high-frequency noise from about 2.5 kHz to 5.5 kHz. The goal will be to design a filter

that will remove this high-frequency noise occurring above 2.5 kHz, while having minimal impact on the speech frequencies up to around 2 KHz.

e) Plot the normalized log plot of the DFT, where the max value of the DFT is 0 dB.

2. Filter Design

f) Based on the DFT of the signal, decide on a digital frequency ωp for the end of the passband and a digital frequency ωs for the beginning of the stop band.

g) Assume no more than 1dB attenuation in the passband. Based on 1e above, decide on the appropriate minimum attenuation for the stopband so that the noise is attenuated sufficiently.

h) Use the Butterworth filter design procedure in the lecture for bilinear transformation to design an analog filter according to the specification on the digital filter that you found in 2a and 2b above. This will involve finding the filter order, N, and the Butterworth cutoff frequency, Ωc. Describe where you are meeting filter requirements and where you are exceeding filter requirements in choosing Ωc.

i) Find the equivalent ωc for the digital filter based on 2h above.

j) Plot the logarithmic gain of the frequency response of your analog filter using the equation below.

$$|H_a(s)| = 20\log_{10}\left(\sqrt{\frac{1}{1+\left(\frac{j\Omega}{j\Omega_c}\right)^{2N}}}\right)$$

3. Filter Implementation

a) The MATLAB butter() command will design a digital filter for us based on our desired digital cutoff frequency. Use the MATLAB butter() command to find the numerator and denominator(b and a) polynomial coefficients for a digital Butterworth lowpass filter with the final cutoff frequency Ωc and order N you found in step 2. In this command, the cutoff frequency is specified as a percentage of $Fs/2$. So, if Fs is 2000 Hz and we need the cutoff frequency to be 500 Hz, use:

$$\mathrm{Wn} = \frac{\Omega_c}{\frac{F_s}{2}} = 0.5$$

in the butter() command for cutoff frequency.

b) Use the a and b filter coefficients you found in 3a and the MATLAB filter() command to create a filtered version of the original noisy audio sample.

c) Calculate the DFT if the filtered audio sample and plot the magnitude.

d) Compare the unfiltered audio and filtered audio DFT plots. Did the filter reduce the noise in the stop band frequencies and keep the desired speech in the passband frequencies?

e) Use the sound() MATLAB command to listen to the original and filtered audio. What difference do you hear?

f) Write the new filtered audio file to 'filteredaudio.wav' using the audiowrite() MATLAB command.

g) BONUS POINTS: What movie is the audio from?

Results

Initial Overview

The original audio file sounds like dialog between two people. It is rather difficult to listen to however, due to loud unwanted noise in the background. The original audio file's DFT magnitude is shown below.

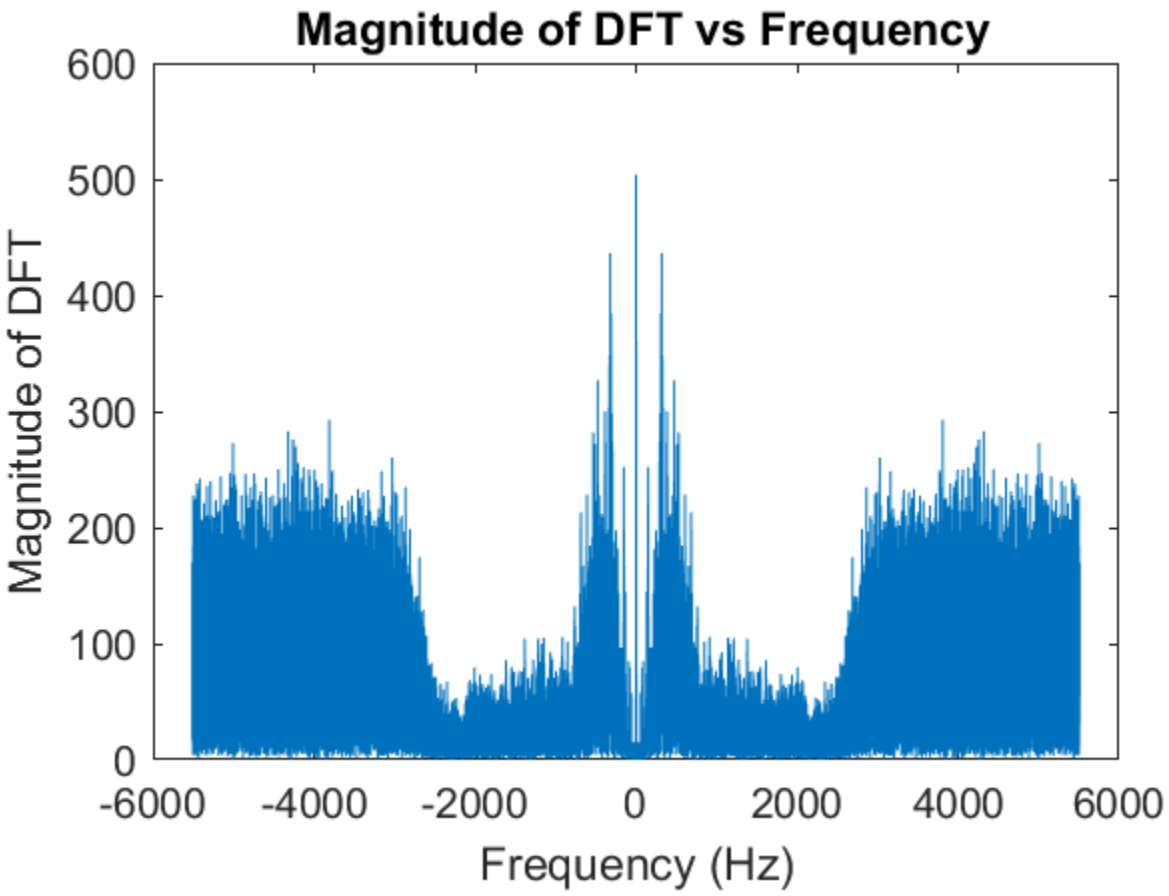

Figure 1a – Magnitude of DFT vs Frequency

One can see that between 100 Hz to 2 kHz, there appears to be moderate sound energy in the form of human speech. From 2.5 kHz to 5.5 kHz however, there seems to be significantly high audio energy.

A normalized Log Plot of the same DFT is shown below.

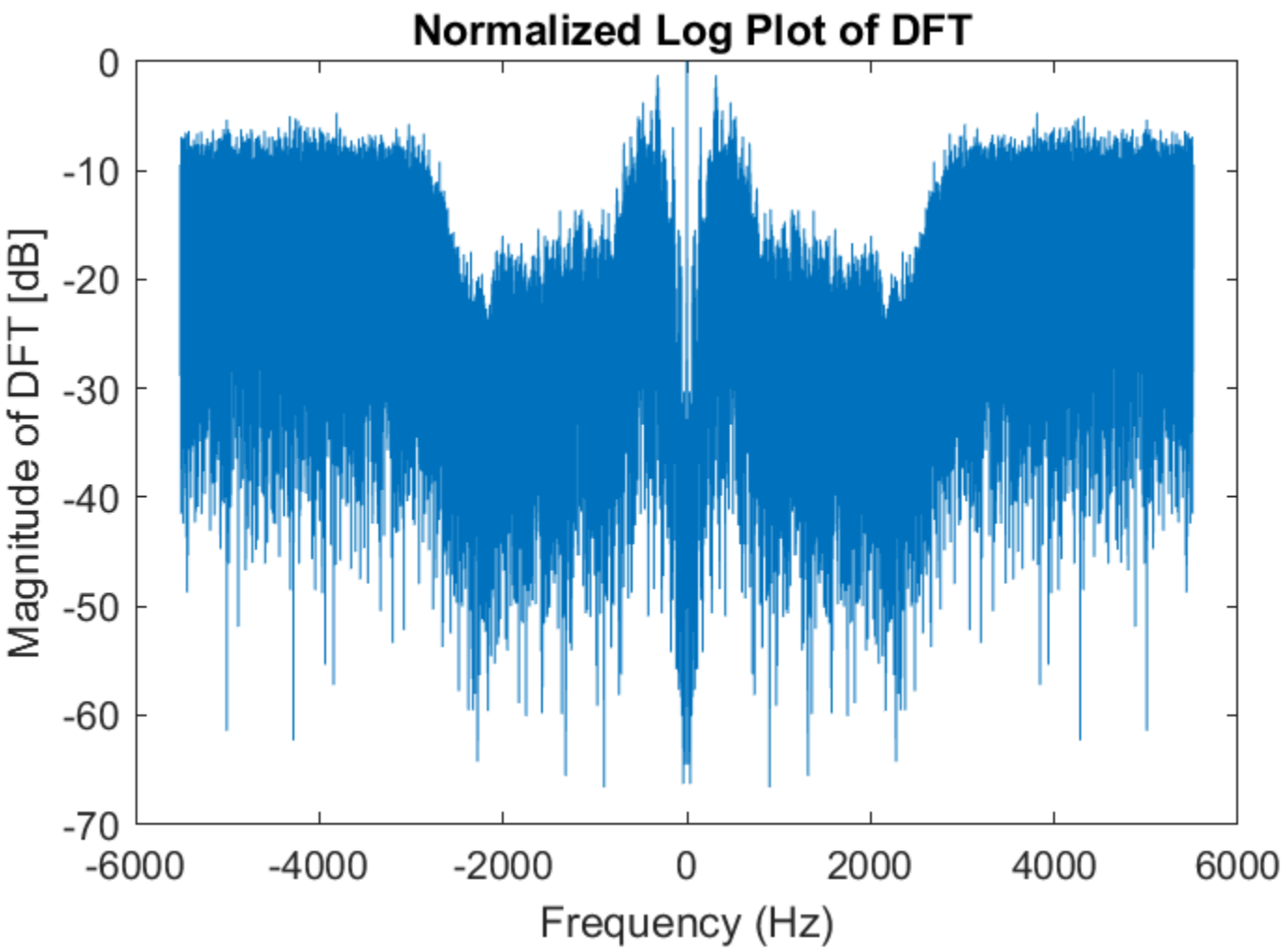

Figure 1b – Normalized Log Plot of DFT

The sound energy from this range of frequencies is the source of the unwanted noise, thus beginning the filtering process.

Filtering

Based on the DFT of the original audio file, the end of the passband ωp was selected as 1.9 kHz and the beginning of the stopband ωs was selected as 2.5 kHz. Following the normalized log plot

of the DFT, the minimum attenuation was selected as 0.5 to allow appropriate attenuation of the noise.

Using the Butterworth Procedure along with the values found above, the filter order 'N' was selected as 29, rounded from:

```
args = (10.^(60dB/10) - 1) / (10.^(1dB/10) - 1);
N = (log10(args)) / (2*log10(2500/1900));
```

The Butterworth cutoff frequency 'Ωc' was selected as ~1.9448 kHz, calculated from:

```
Ωc = 1900/(10^(1dB/10)-1)^(1/(2*N)); % with 1900 as Wp
```

All filter requirements are met with the selections of these values.

The equivalent ωc is 2arctan(ΩcT / 2).

The resulting logarithmic gain of the frequency response is shown below.

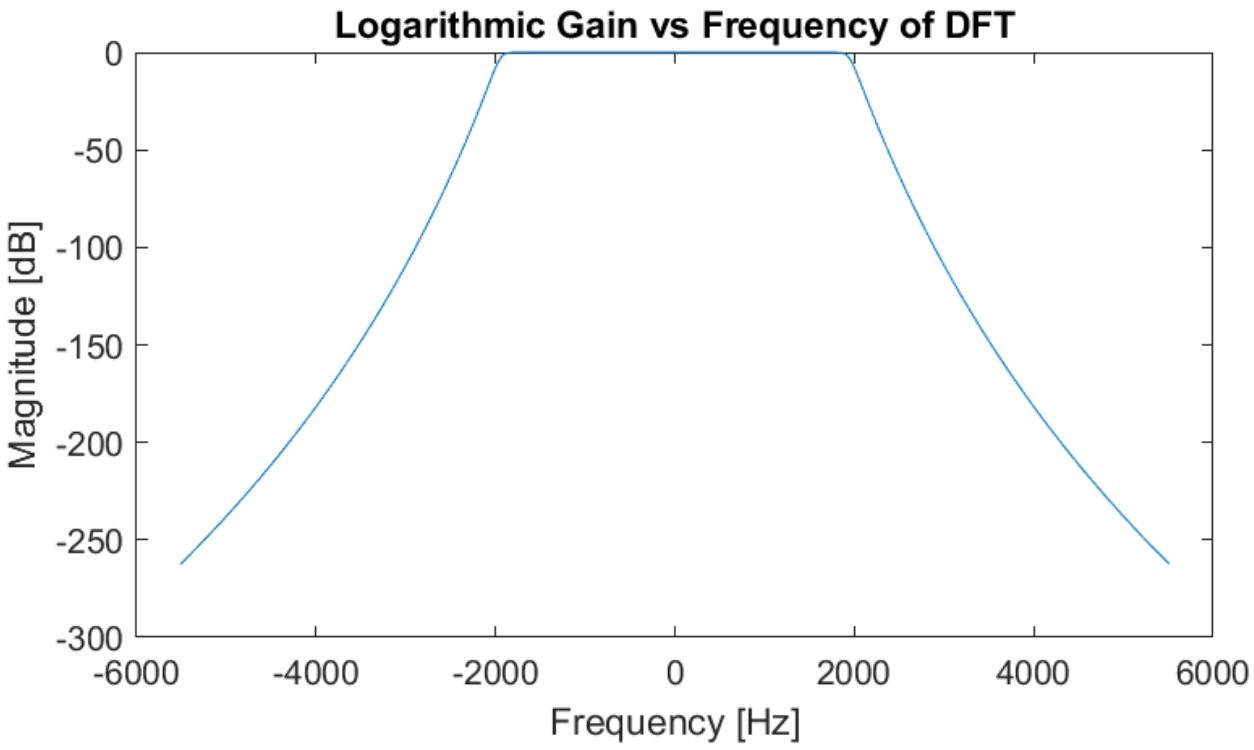

Figure 1c – Logarithmic Gain vs Frequency of DFT

Implementation

Before creating the Butterworth filter, Wn must be found. Using the cutoff frequency Ω*c*, Wn comes out to be 0.3528. After creating the Butterworth filter and applying to the original audio, a resultant, new audio file is formed along with a new calculated DFT.

The magnitude of the new DFT vs frequency is shown below.

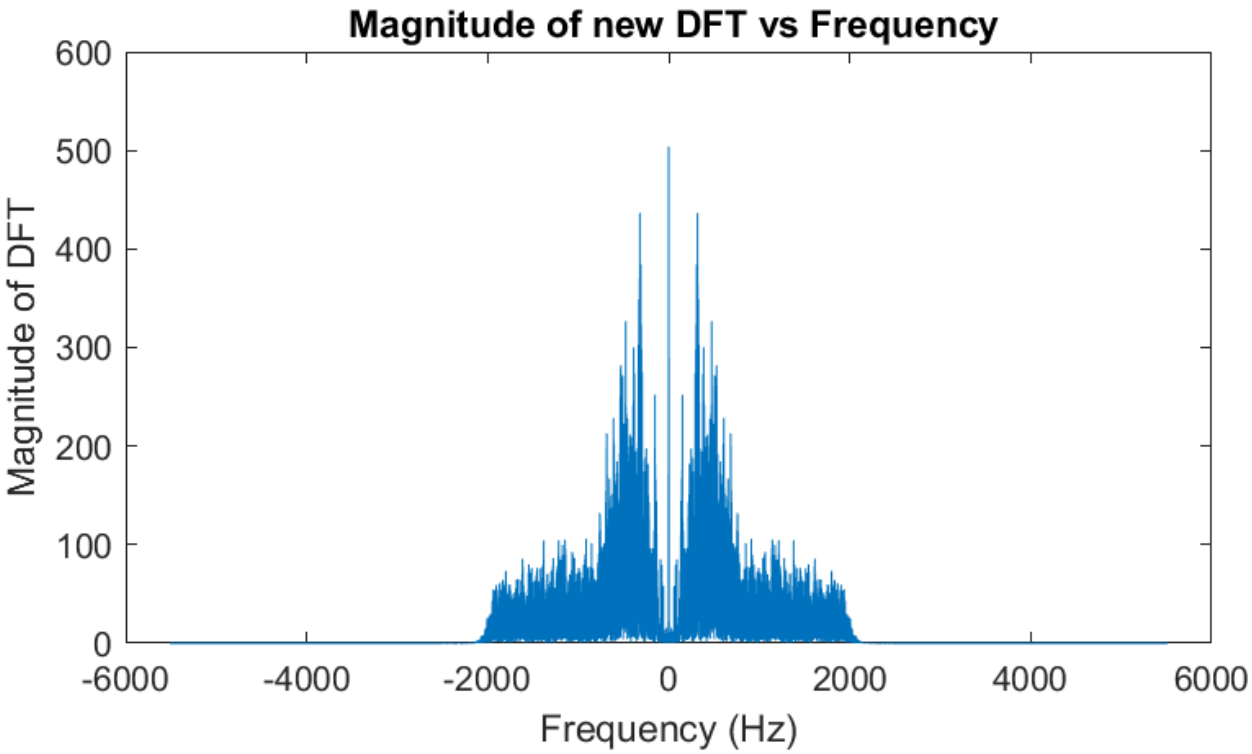

Figure 1d – Magnitude of new DFT vs Frequency

At this point, we can compare the magnitude of the original audio's DFT with the magnitude of the new audio's DFT. If filtered correctly, there should be a noticeable difference in sound energy between 2.5 kHz and 5.5 kHz.

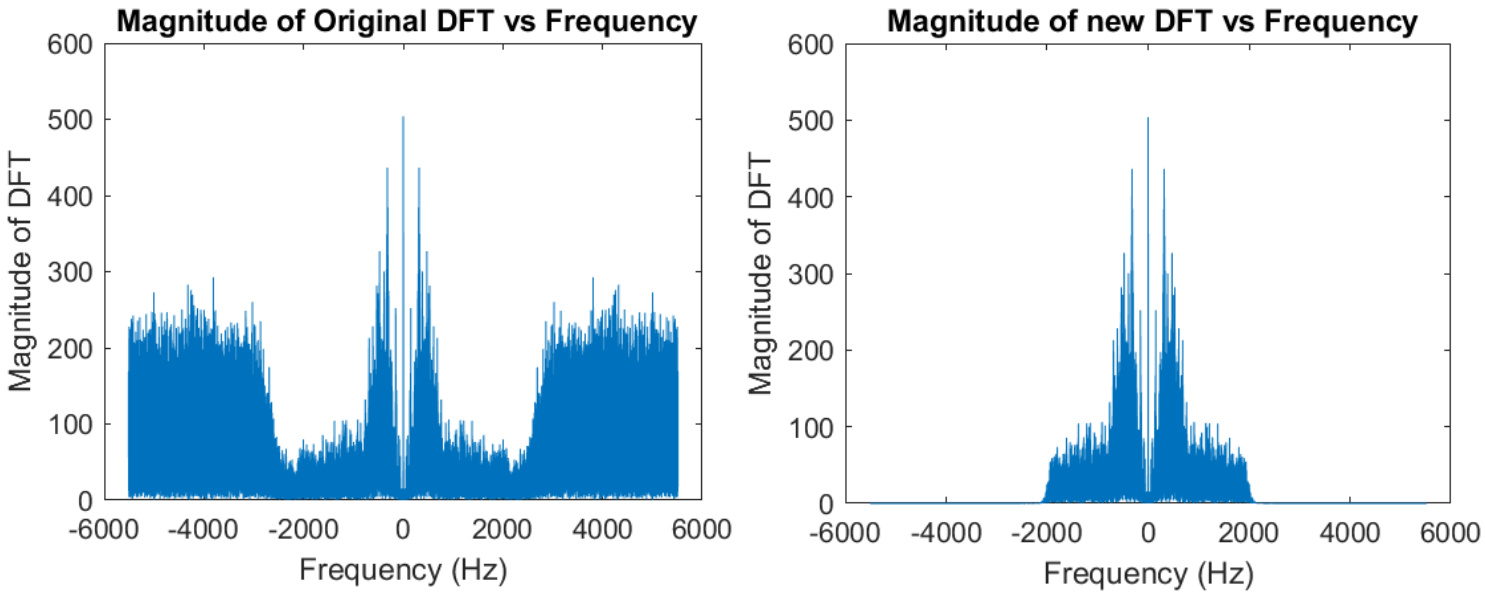

Figure 2 – Comparison of Original and New DFT Magnitude vs Frequency

As shown, the sound energy for the filtered audio has now been cut off from 2.5 kHz to 5.5 kHz. This should also translate to how the new audio file will now sound compared to the original.

Conclusion

After listening to the original and filtered audio files, one can conclude that the use of a Butterworth filter has successfully filtered out unwanted noise in a given audio file, while keeping desired speech audio.

The new audio file, called “filteredaudio.wav”, plays a filtered version of the original audio file.

To hear the comparison of both audio samples, run FilterDesignProject.m on MATLAB

Plots

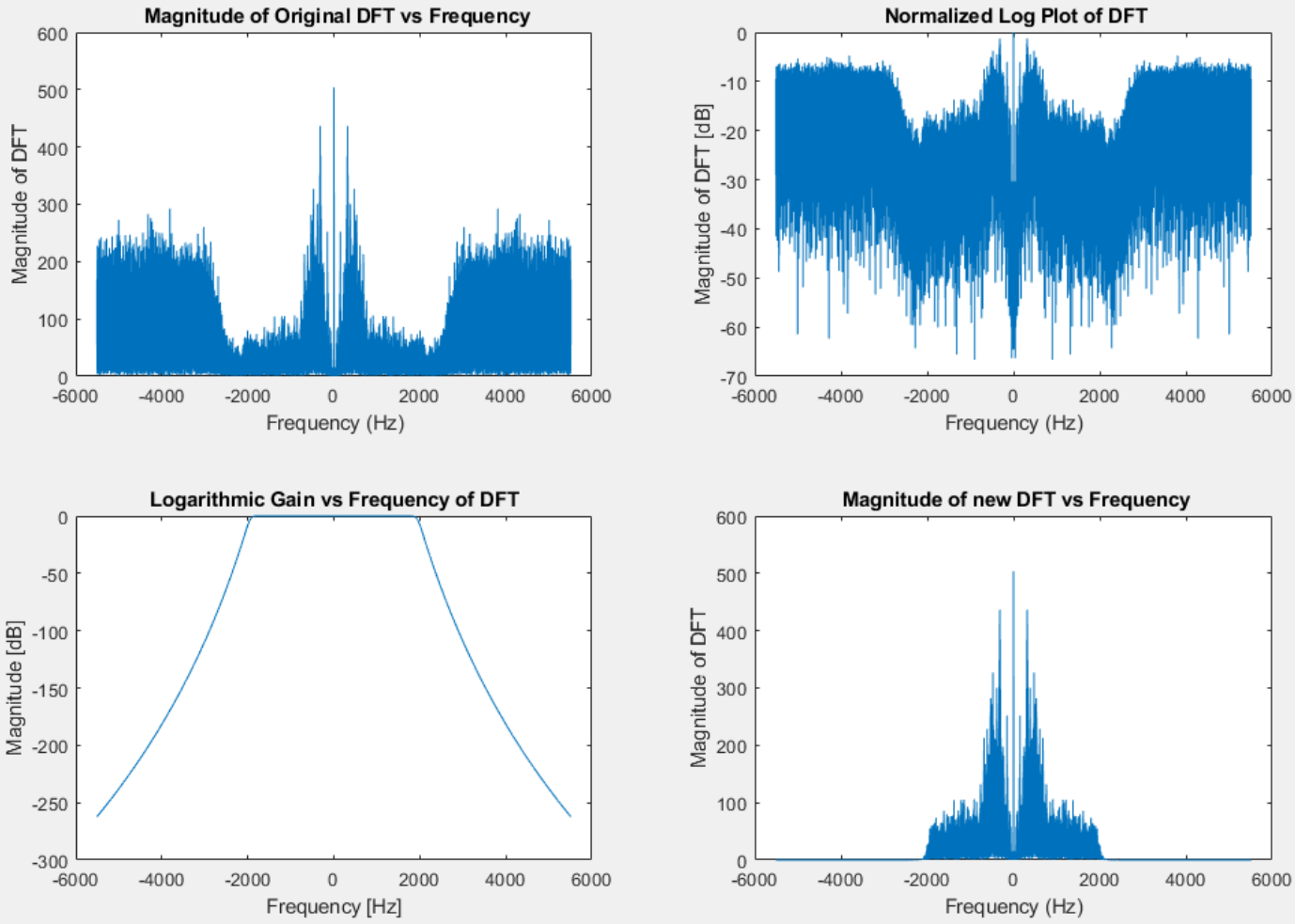

Figure 1 – Canvas of all generated plots

EXTRA CREDIT

The audio is from a movie called Airplane! (1980).

Directors: Jim Abrahams, David Zucker, Jerry Zucker

Transcription of the audio file:

Rumack:

Captain, how soon can you land?

Captain Oveur:

I can't tell.

Rumack:

You can tell me. I'm a doctor.

Captain Oveur:

No. I mean I'm just not sure.

Rumack:

Well, can't you take a guess?

Captain Oveur:

Well, not for another two hours.

Rumack:

You can't take a guess for another two hours?

References

Movie Transcription:

https://www.quotes.net/mquote/1147