Rithika V Dandu, HW-5

BMEn 8101: Homework Set #5 **Instructor: Professor Hubert Lim**

Due by 11:59PM on March 29, 2024

Three of the problems below will be selected for grading. Problem 7 will be graded for sure. Each graded problem will be worth 10 points, so total of 30 points.

Please submit your homework in digital form to **BMEN8101Lim@gmail.com**.

Problem 1: 10.12

Problem 2: 10.19
Explain what each method will do to the frequency resolution of the STFT of x[n] (i.e., $X_r[k]$)?

Problem 3: 10.22

Problem 4: 10.32 (a), (b), (c), (d)
Please explain how you determined your answer choices.

Problem 5:
Using Matlab, please plot the STFT magnitude (i.e., spectrogram) for the both the clean and noisy speech signals from HW#4 (SigNoise and SigOG). Try both the rectangular and Hamming windows with varying lengths, L, and explain how these different windows affect the spectrogram. By comparing the spectrograms for the clean versus noisy speech, do any particular parameters improve visualization of the speech components over the noise components? Make sure to use a small enough frame shift value, R, and sufficient zero padding (so increase DFT to N points) to improve visualization of the plots. Does zero padding improve visualization of the speech components from the noisy background? Provide sufficient plots to justify your answers.

Problem 6:
From Problem 5, please reconstruct your STFT signals back into the time domain. Show that you selected appropriate overlapping windows based on the mathematical constraint shown in class and that the reconstructed signal matches the original signal. Please zoom up on your signals in several locations to show they are similar. Also zoom in at the start and end of the reconstructed signal and describe any distortion that may be caused by a lack of sufficiently overlapping windows to satisfy the constraint. Please look at pages 7-11 in the **STFT Windowing Reconstruction** document on CANVAS for different rules/constraints for each type of window (i.e., how to select R, M, L and N for your windows).

Problem 7:
You need to denoise a time-varying noisy signal using STFT (download **"tvNoisySpeech.mat"** from CANVAS; it contains a noisy signal and the original signal as well as the sampling frequency (fs) for the signals). This will require submission of multiple figures and justifications of what you tried and how well they worked. You will also email your best denoised signal as sound wav files to the TA (**BMEN8101Lim@gmail.com**). Please name this name: **HW5_StudentName_STFT.wav**. Try different window types and lengths to help you improve the denoising of the signal. Because the type of noise is changing over time, you will need to make an algorithm to denoise the signal differently across different time segments. I would suggest that you first visualize the signal using STFT as you did in Problem 5 to determine the type of noise in each segment. Then run your algorithm to denoise the signal over time using STFT (e.g., making a bandstop filter that shifts its frequency range with each proceeding segment

or using spectral subtraction that eliminates different frequency components across segments). Remember to follow the STFT Flow Chart presented in class (e.g., windowing constraint, zeropadding, etc.).

NOTE: Please create your own matlab code to do Problems 5 to 7 above. You can use Matlab's spectrogram function and the ones provided on the CANVAS website to help you understand and implement your own STFT code.

A nice link for more info on various/numerous topics relating to DSP. The link goes directly to reconstructions with the STFT but you can navigate the site to view other topics.
https://ccrma.stanford.edu/~jos/sasp/Overlap_Add_OLA_STFT_Processing.html

Problem -1 : 10.12

$x[n] \to$ sinusoidal

L pt HW $\to V_1(w)$ larger peak =?

RW $\to V_2(w)$

$$v[n] = w[n]\, x[n] \;\to\; V'[w] = W[w] * X[w]$$

$V_1(w) \to$ hamming

$V_2(w) \to$ rectangular

The peaks will not be of the same height

FT of a window $W(w)$ at $w = 0$

$$W(e^0) = \sum_{n=0}^{L-1} w[n]\cdot 1 = \sum_{n=0}^{L-1} w[n]$$

Fig 7.29

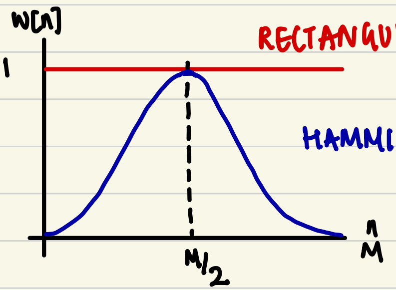

Fig 7.30

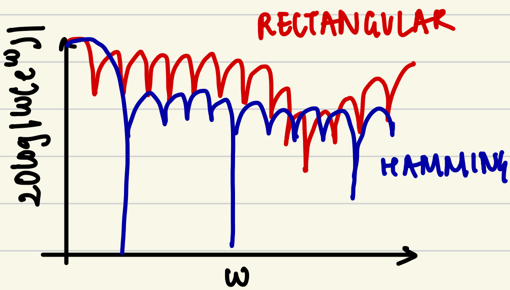

Fig 7.29 and Fig 7.30 show the time and frequency domain representations of the windows

From 7.29, we can see that $\sum_{n=0}^{L-1} w_R[n] > \sum_{n=0}^{L-1} w_H[n]$

$\Rightarrow W_R(w) > W_H(w)$ when $w = 0$

This will mean that $|V_2| > |V_1|$

10.19

Problem 2: $N=128,\ L=128,\ R=128$

Method-1: $N=256 \quad L=128 \quad R=128$

by increasing the points we are simply taking more samples. Frequency resolution is <u>NOT</u> improved.

Method-2: $N=256 \quad L=256 \quad R=128$

increasing the window length narrows the main lobe, improving frequency resolution.

Method-3: $N=128 \quad L=128 \quad R=64$

Reducing R improves time resolution and reduces smearing but this does not improve frequency resolution.

Method-4: $N=128 \quad L=64 \quad R=128$

decreasing window length widens the main lobe and decreases frequency resolution.

Method-5: $N=128 \quad L=128 \quad R=128 \qquad w_R[n]$

ignoring side lobe leakage, resolution is improved.
if there is no side lobe leakage,

main lobe widths → hanning $8\pi/M$
main lobe widths → rectangular $4\pi/M+1$

Rect has narrower lobe ⇒ better frequency resolution

PROBLEM-3 10.22

main lobe widths from table 7.2

Rectangular:

$$\frac{4\pi}{M+1} = \frac{4\pi}{256} = \frac{\pi}{64}$$

Hamming:

$$\frac{8\pi}{M} = \frac{8\pi}{255} \approx \pi/32$$

$x_1[n] \rightarrow$

$$\Delta_1 = \pi/32 - \pi/64 = \pi/64$$

This is better resolved by a rectangular window for $x_1[n]$

$x_2[n] \rightarrow$ $\Delta_2 = \frac{11\pi}{32} - \frac{\pi}{4} = \frac{3\pi}{32}$

larger than both main lobes of hamming and rectangular → now checking for the side lobes

As for the side lobes:

$20\log(1/0.017) = -35.39\text{dB}$ → difference between amplitudes of higher and lower frequencies

For Rectangular: $-13 - 21 = -34$ dB

For Hamming: $-41 - 53 = -94$ dB

peak side lobe amplitude

(peak approximation error

The separation at the side lobes is better for Hanning, for signal $x_2[n]$
so it is better to use Hanning for $x_2[n]$

$x_3[n]$ $\Delta_3 = \frac{257\pi}{1024} - \frac{256\pi}{1024} = \frac{\pi}{1024}$

This is far less than the main lobe width for either signal, thus resolving is not possible with either window.

Problem-4: 10.32

a) Rectangular window function was used for (a) and (c) as we can identify sidelobes that are not present in the other two.

Ans: a, c

b) (a),(b) } have the same frequency resolution
(c),(d) } as the lines are of similar width at given frequencies.

Ans: a, b and c, d

c) (c) and (d) both have broad frequency resolution. (c) takes lesser space to transition at the ends though.

Ans: (c)

d) In (b) we can observe that the fuzzy region is about ≈ 400 samples long which is the window length.

Ans: 400 samples

Problem 5.

For direct comparison, I did one set of plots with the same R, L values across both types of input signal and types of windows.

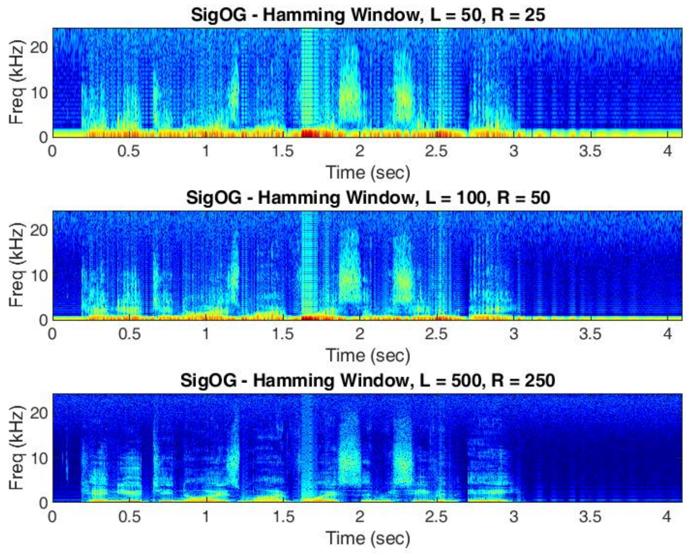

Fig 1: Hamming Window for SigOG

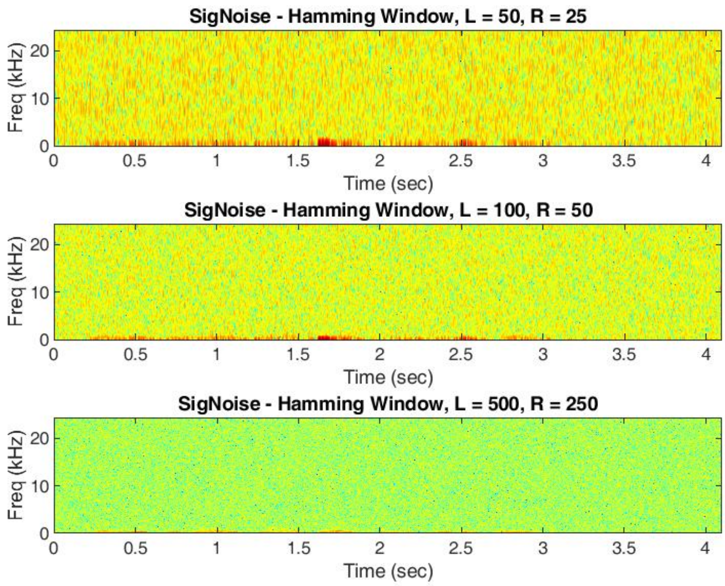

Fig 2: Hamming window for SigNoise

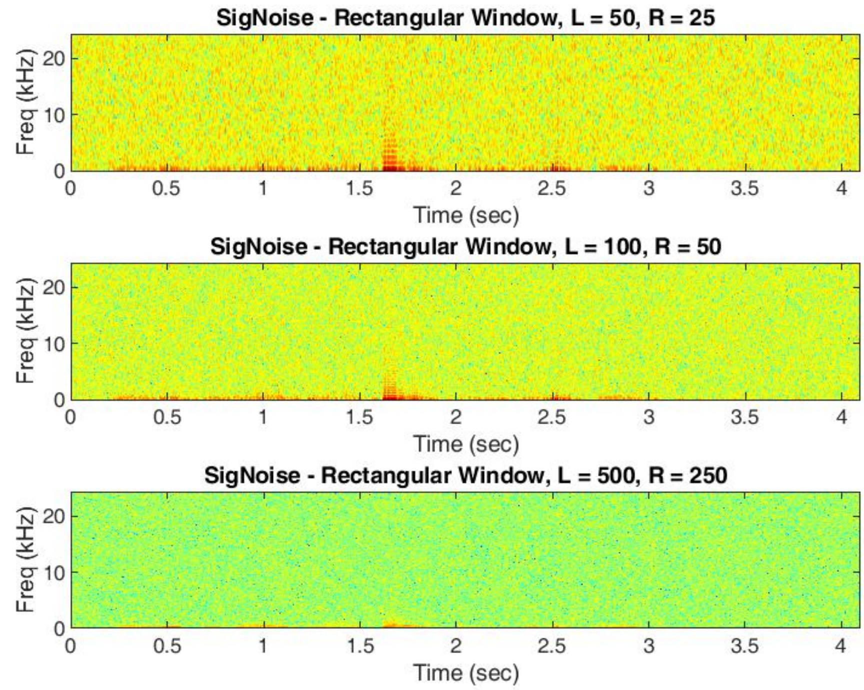

Fig 3: Rectangular Window for SigNoise

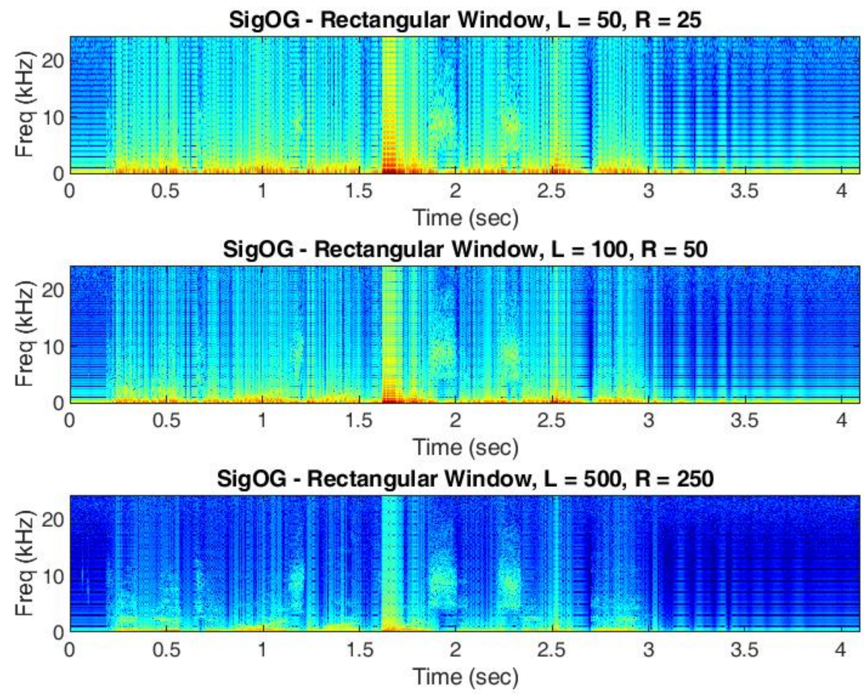

Fig 4: Rectangular Window for SigOG

From Figs 1-4, we can say that:

1. Short windows provide better time resolution as they capture changes in the signal in shorter periods better. On the flip side, they provide poorer frequency resolution as they have wide main lobes, causing the energy to be spread over a larger range of frequencies.
2. Longer windows provide better frequency resolution because the main lobe of the window is narrow. They provide poor time resolution, though, as changing signal content is not readily captured.

This is easily noticed in the plots of SigOG - where increasing the window length provides narrower horizontal ‘bands’ in the spectrogram along given frequencies, suggesting that the frequency resolution is improving. Smaller window lengths have broader bands, implying that a given band is spread over a small range of frequencies, suggesting poor frequency resolution. This is the case for both Hamming and Rectangular windows.

The overlap size also plays a role in the visualization of plots. I have elaborated on this in Problem 6, but R and L values that satisfy the COLA condition give us the ideal signal.

We can also compare Hamming and Rectangular windows:

1. Hamming Windows have a smoother distribution of energy around the main lobe. The frequency content is more defined with less spillage into other frequencies.
    a. They have lesser spectral leakage.
    b. They have wider main lobes.
    c. They have smaller side lobes.
2. Rectangular windows have a boxier appearance, with abrupt transitions between segment. The side lobes are higher in comparison as well.
    a. They have more spectral leakage.
    b. They have narrow main lobes.
    c. They have larger side lobes.

Zero padding a signal gives us better visualization of the signal without improving frequency resolution. Zooming in on the zero padded signal’s spectrogram did not show me better frequency bands with the same parameters ($L = 100$, $R = 50$) but it qualitatively looked better

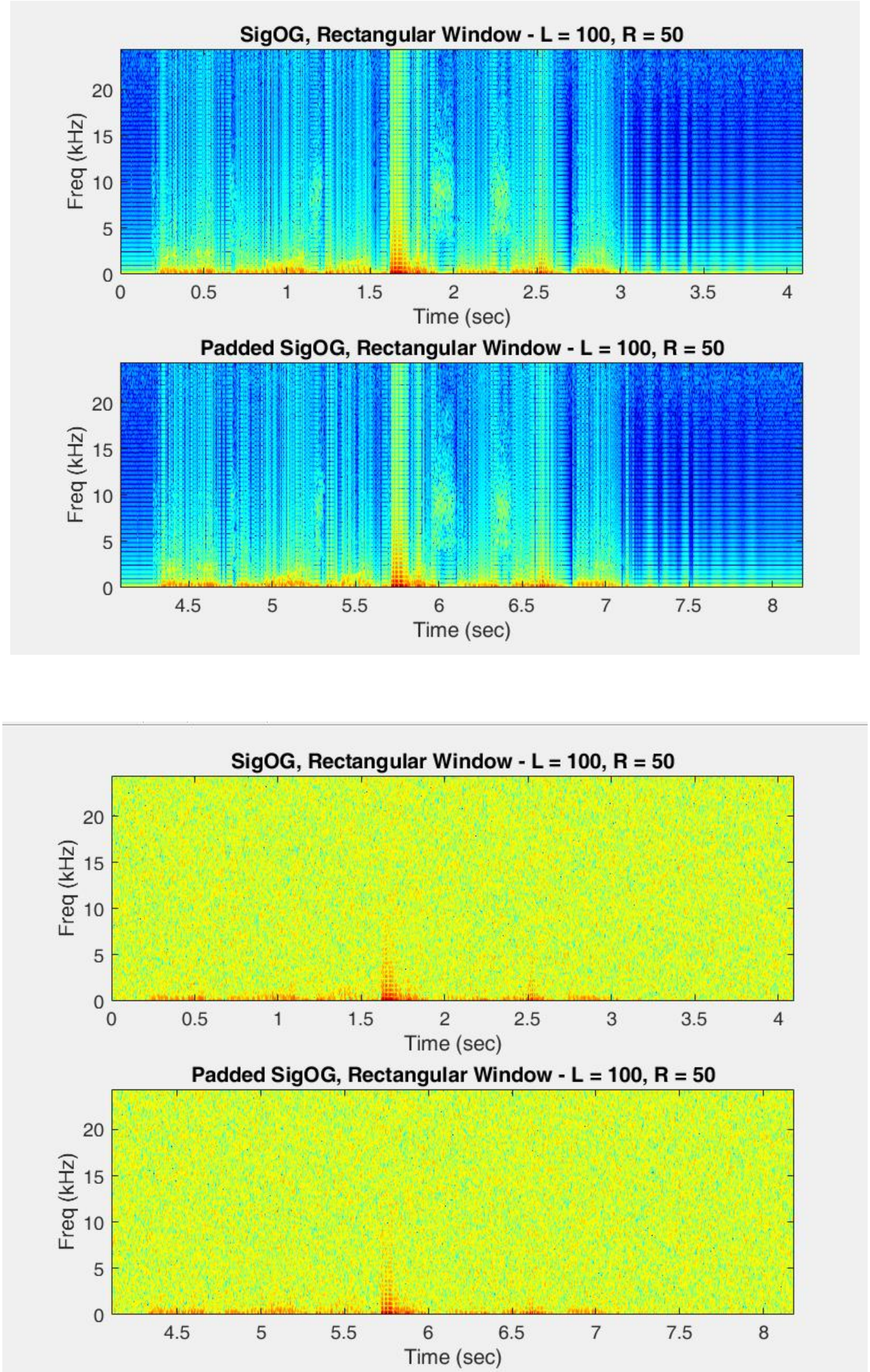

Fig 5: Zero padded signal with Rectangular Window

Problem 6.

I first reconstructed my signal using a Hamming window with L = 500, R = 250.

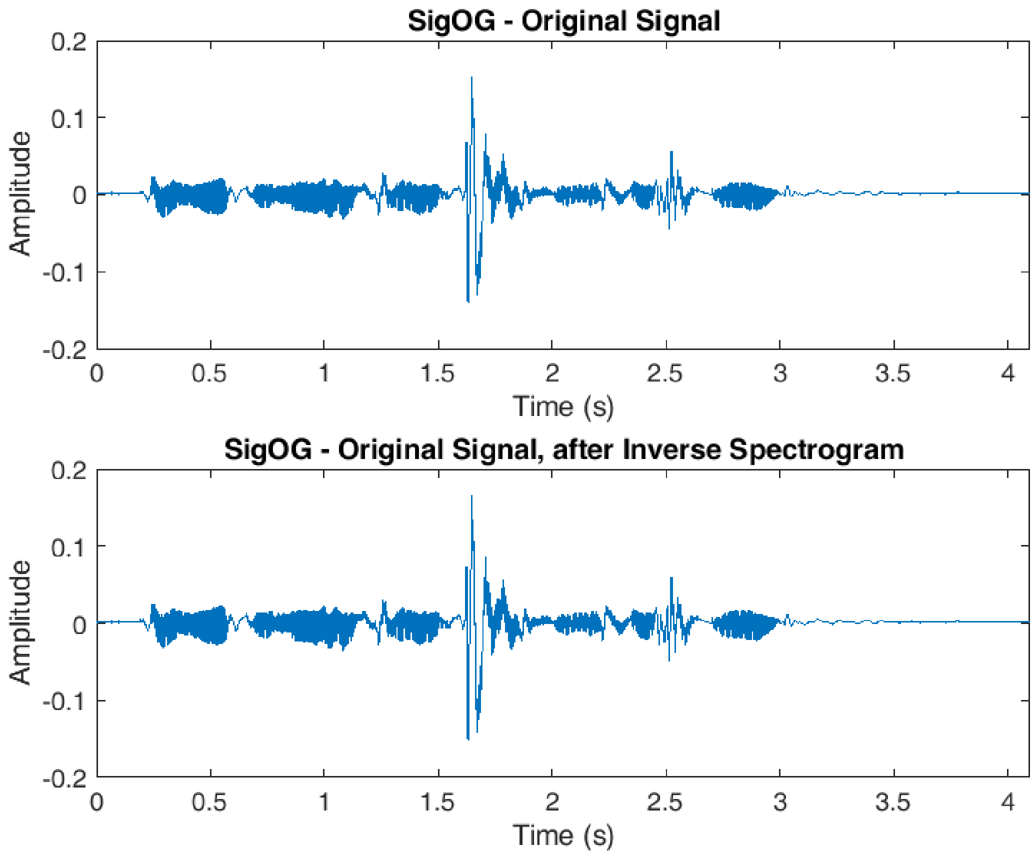

Fig 6: SigOG compared to the reconstructed SigOG

The signals look exactly the same under these conditions, and zooming in on various parts also yielded the same signal except for right at the beginning. The ends of the signals looked identical. This is possibly because the signal was reconstructed with ideal parameters R = L/2.

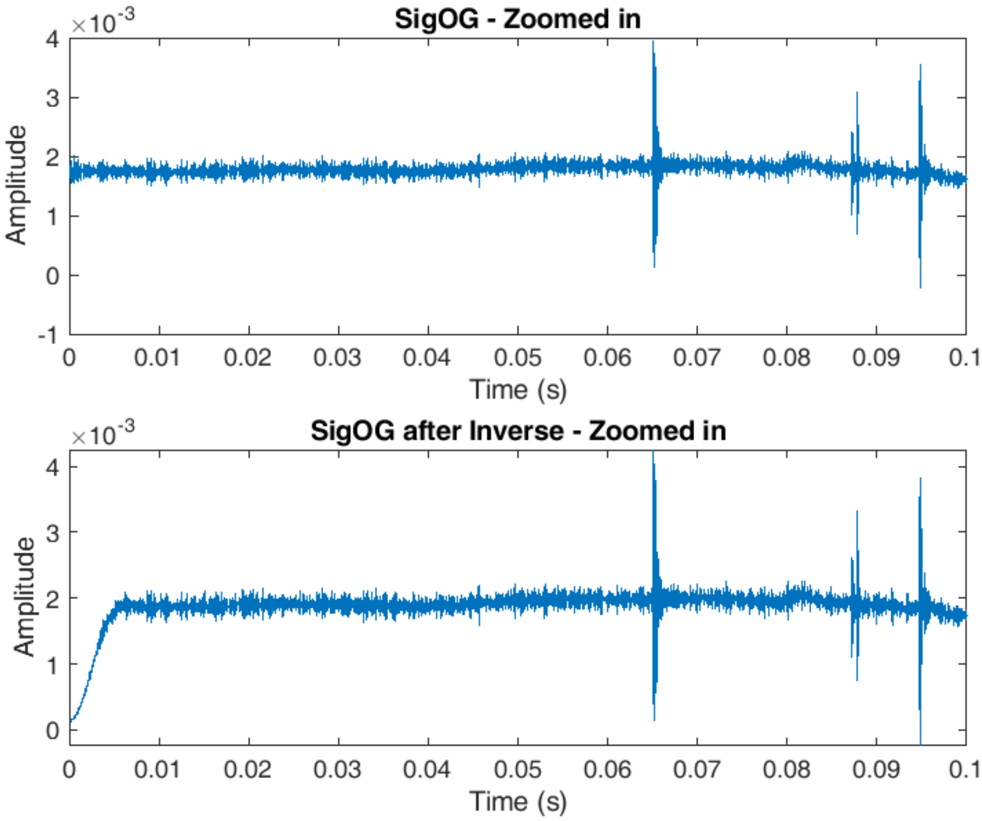

Fig 7: Distortion observed right at the beginning of the reconstructed SigOG

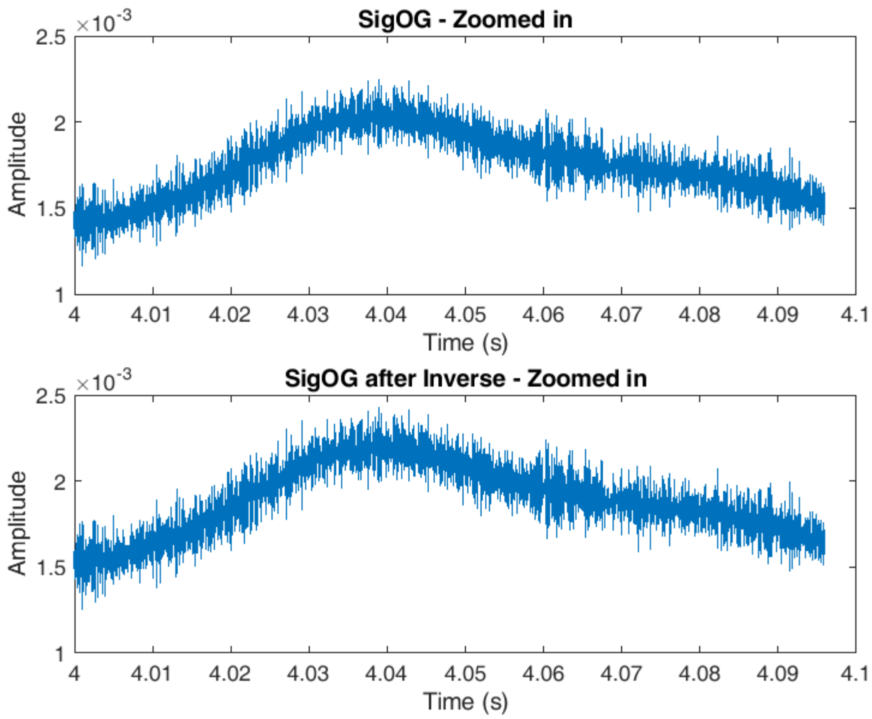

Fig 8: Ends of the signals SigOG and reconstructed version

I changed the window overlap size to violate COLA by a small margin, and used a rectangular window to generate and display more distortions. The following plots were generated using L = 128, R = 70

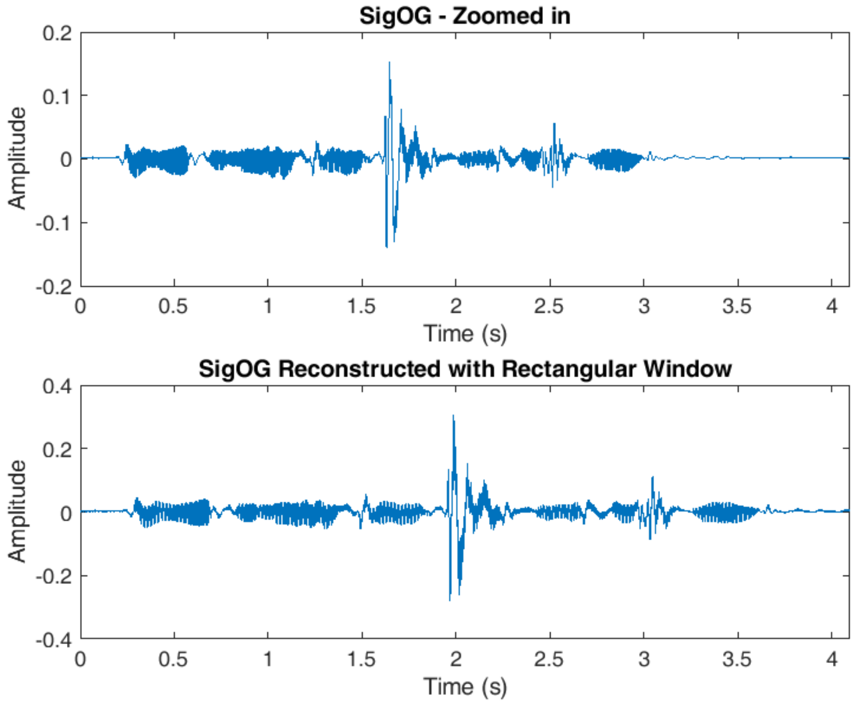

Fig 9: Overall signal and its reconstruction with a rectangular window

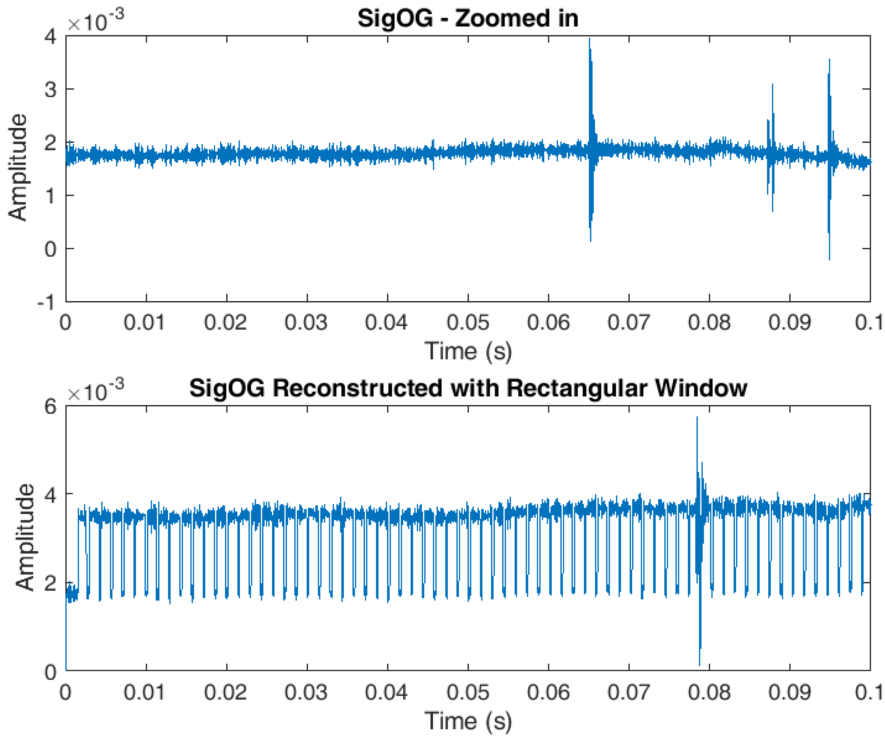

Fig 10: Beginning of the signals, distortion observed in reconstruction

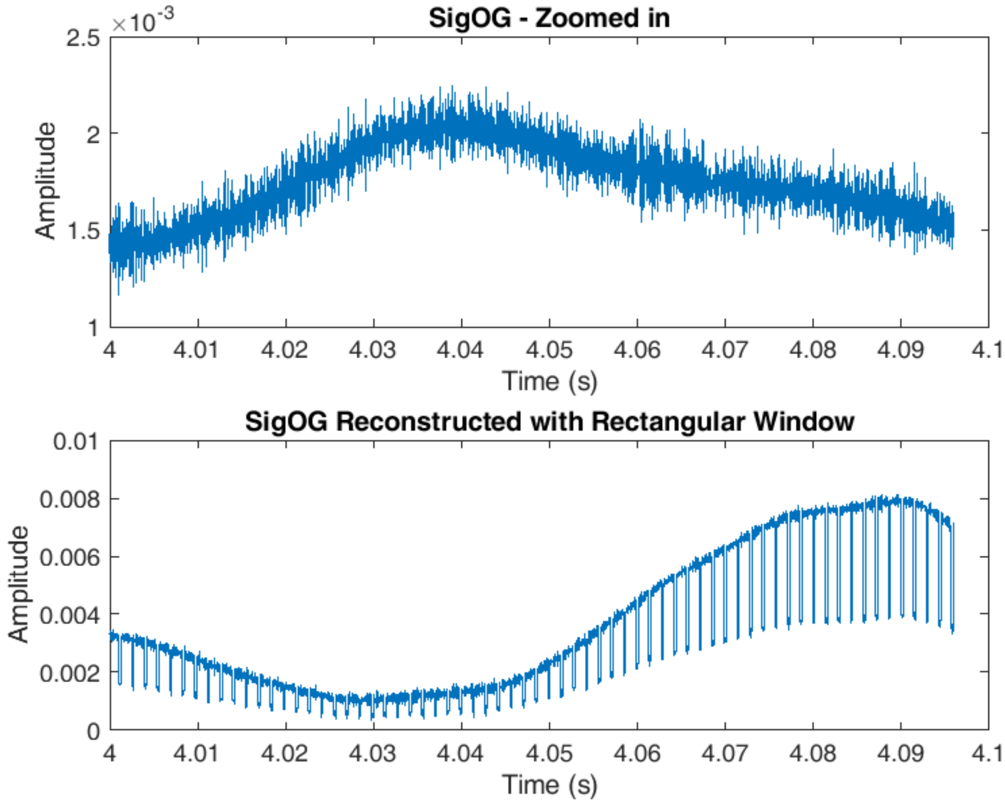

Fig 11: Ends of both signals, distortion observed in the reconstruction

The mathematical constraint that needs to be followed here is the condition of overlap addon (COLA). Violations of COLA can cause significant distortions which we have observed a little of above. This is demonstrated further below.

The conditions were:

1. Cola violating: L = 128, R = 80
2. Cola violating: L = 128, R = 100

For a hamming window, and were plotted compared to the COLA satisfying condition of L = 128, R = 64. If the length of the window is even, then the overlap should be ideally half of that. If odd, it must be (L-1)/2. The following plots were generated using a Hamming window.

Violating COLA causes inconsistencies in the signal reconstruction process, which is why we can observe distortion, delays and amplitude modulation in our reconstructed signals that violate COLA.

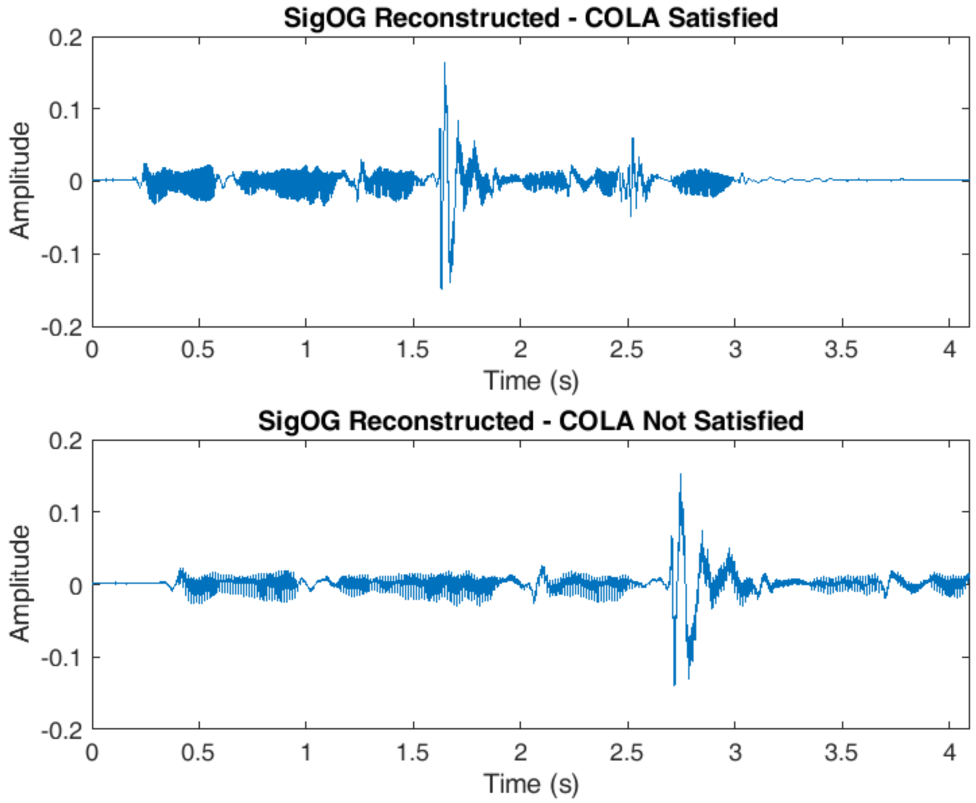

Fig 12: Depiction of distortion and delay from violating COLA, L = 128, R = 80

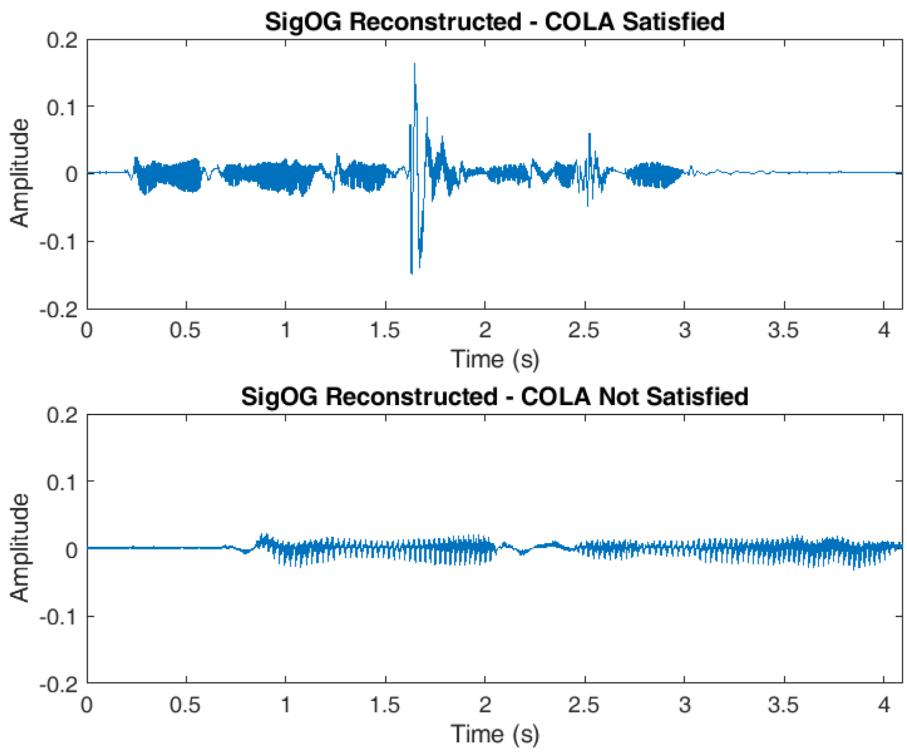

Fig 13: Another depiction of distortion, delay and amplitude modulation from violating COLA, L = 128, R = 100

Problem 7

The workflow involved: STFT with Hamming window and Bandstop filtering, Low pass filtering, signal fitting and removal.

For this problem, I tried denoising the signal using windowing, but did not find L and R values or windows that could sufficiently denoise the signal by themselves even with the bandstop filtering. In addition to STFT, I utilized methods that were done for HW4 to attempt to entirely get rid of the noise.

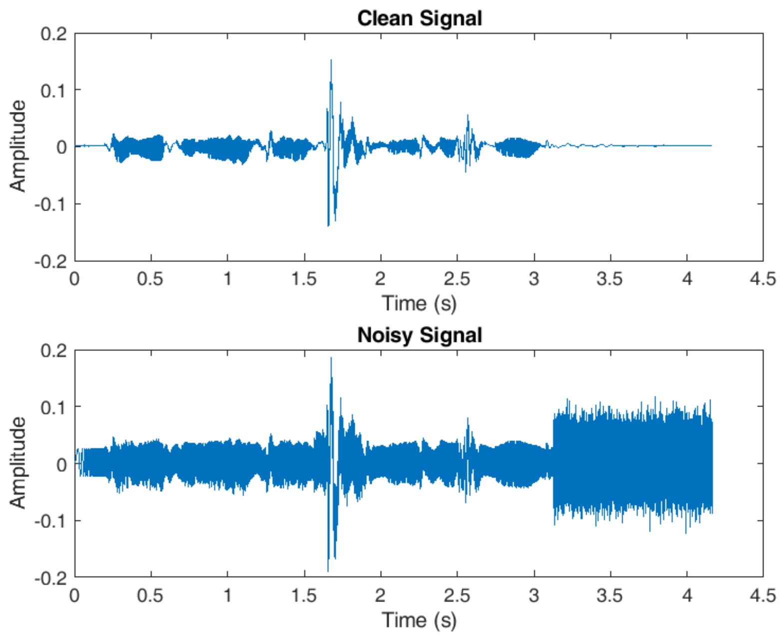

Fig 14: Amplitude-Time plots of the Clean and Noisy Signals

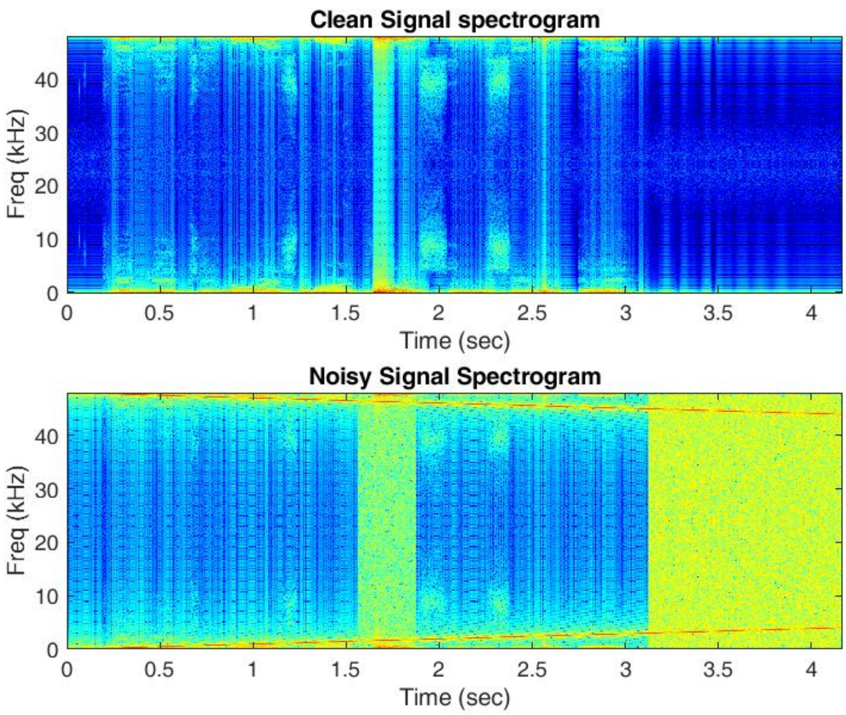

Fig 15: Spectrograms of the Clean and Noisy Signals

We can see from the spectrogram (and also by listening to the clip) that multiple types of noise exist. The green area in the spectrogram is the white noise that is present intermediately and at the very end of the clip. The orange bands are from the noise that increases in frequency as the clip plays.

I performed STFT with COLA Satisfying conditions with a Hamming window. L = 500, R = 250. The spectrogram and the reconstructed signal are shown below.

I modified the program for inversespectrogram and called it inv7.m. The modifications included a band stop filter that adapts to the frequency segments and removes noise accordingly. Specifically, at every frame, it identifies all frequency values above a certain threshold and, with a bandstop filter, filters noise between fmin (800Hz) and fmax (fs/2) using a butterworth low pass filter. This filter is still applied within the function, and filters at the level of each individual frame. The frames are then recombined as done in the regular invspectrogram.m function.

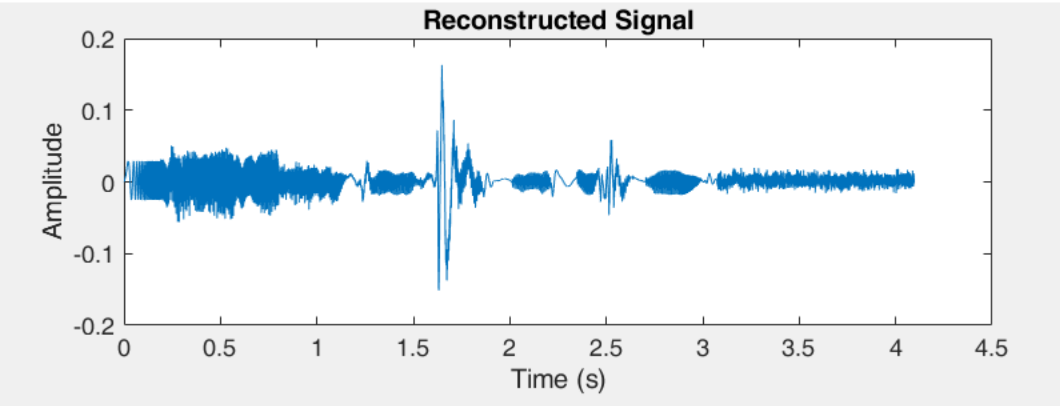

Fig 16: Reconstructed signal from STFT with Hamming window

The last component involved using two low pass filters for the entire signal. The filters were IIR Least Square filters with orders of 1000. The filter parameters were as follows:

1. Filter 1
   a. Wpass = 1
   b. Wstop = 1
   c. Fpass = 1000
   d. Fstop = 1200
2. Filter 2
   a. Wpass = 1
   b. Wstop = 1
   c. Fpass = 500

d. Fstop = 750

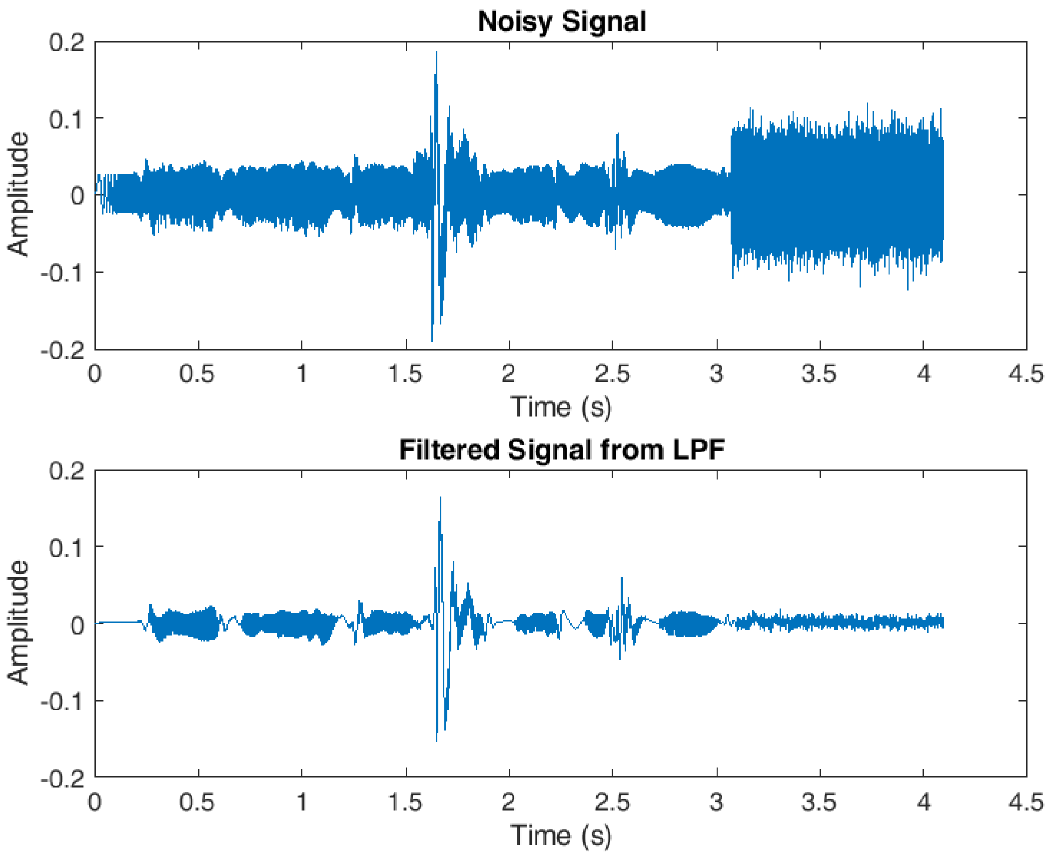

Fig 17: Filtered signal from the two LPF after Hamming Window

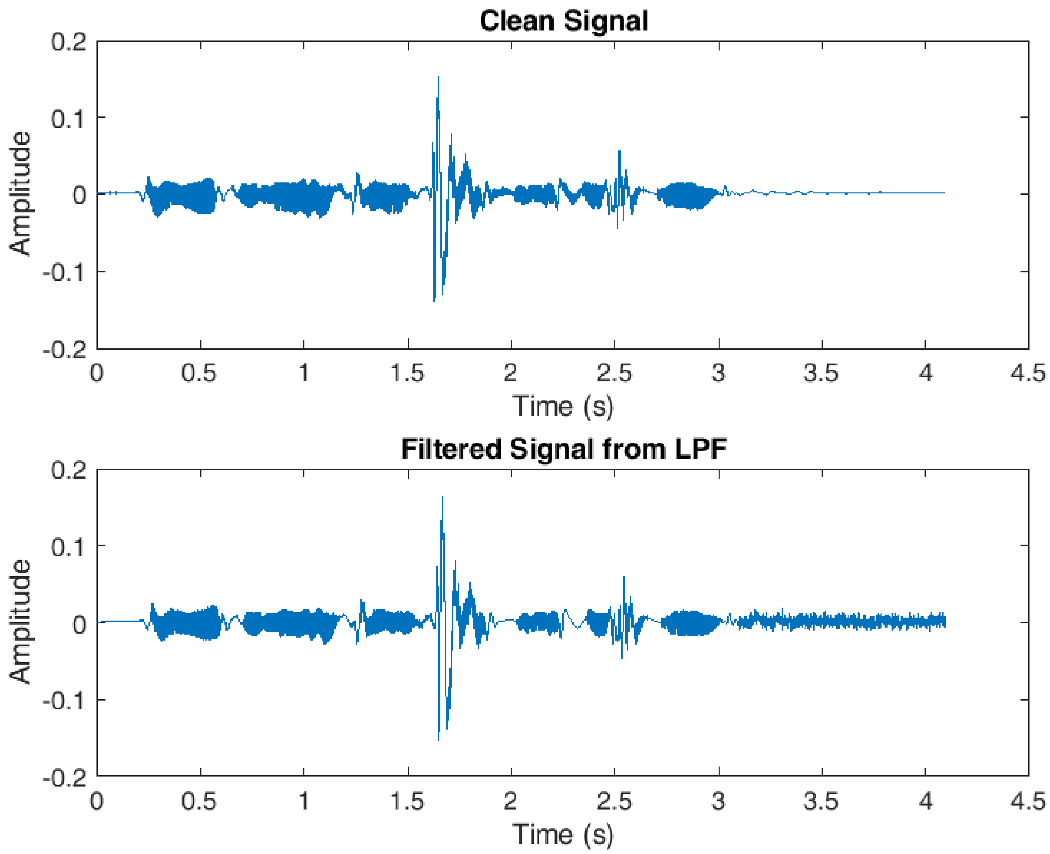

Fig 18: Comparison of the real signal and the filtered output after all the aforementioned steps have been completed

In summary, the final output summitted as ‘HW5_RithikaVarma_STFT.wav’ is fairly denoised. There are faint traces of the chirp and noise at their onsets, but are mostly removed from the signal.

<u>Two other things I tried:</u>

I also attempted to ‘fit’ the noise that increases in frequency and remove it prior to doing the Hamming window, which is in the main Prob7.m document but is commented out because it involves prior knowledge of the clean signal to estimate the frequency increase. This is similar to fitting 60Hz noise from HW4. - fitting a chirp and removing it from the main signal. Plotting the spectrogram for removing the chirp showed that it was successful.

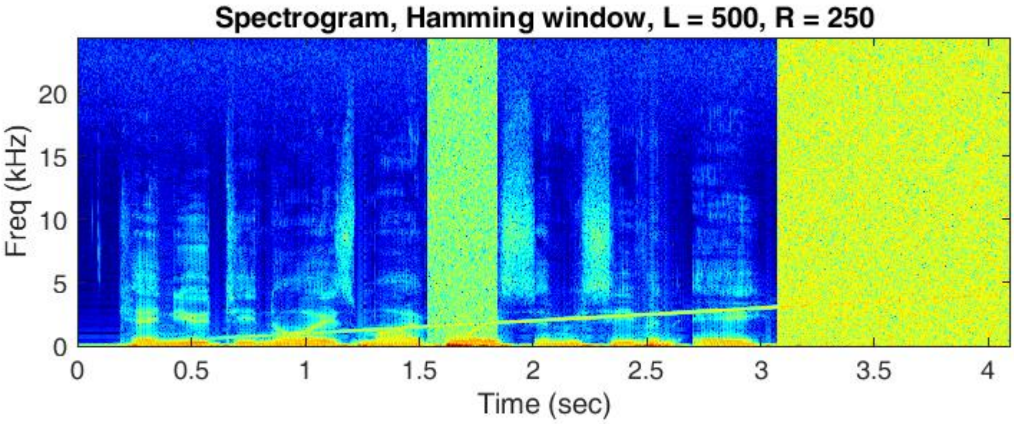

Fig 19: Chirp removal using signal fitting and removal (commented out)

Another thing I tried was directly using different thresholds for spectrally filtering for different time segments of the signal like in HW4, but I wasn't successful at extracting something meaningful from it.