Using Comb Filter and Adaptive Line Enhancer to Enhance the Audio Quality of a Recording

Yevgen Solodkyy

1 Introduction

In this project I attempted to improve the audio quality of the video linked (`https://www.youtube.com/watch?time_continue=14&v=PB_8iOCm92c`). The audio is corrupted by a strong power line tone, as well as white noise in the background, resulting in the signal having a fairly low SNR, as shown in Figure 1. I was also curious to see how much the SNR improved if I was able to remove the noise. Though the SNR is an important quantitative metric, my measure of the outcome was the more subjective perceived audio quality. I chose to approach this problem by first eliminating the harmonic tone with a comb filer and later using an adaptive line enhancer (ALE) to remove white background noise. This, in my estimation, would result in improved SNR and audible quality of the signal. The comb filter output is denoted as $g(n)$, and adaptive line enhancer output as $y(n)$.

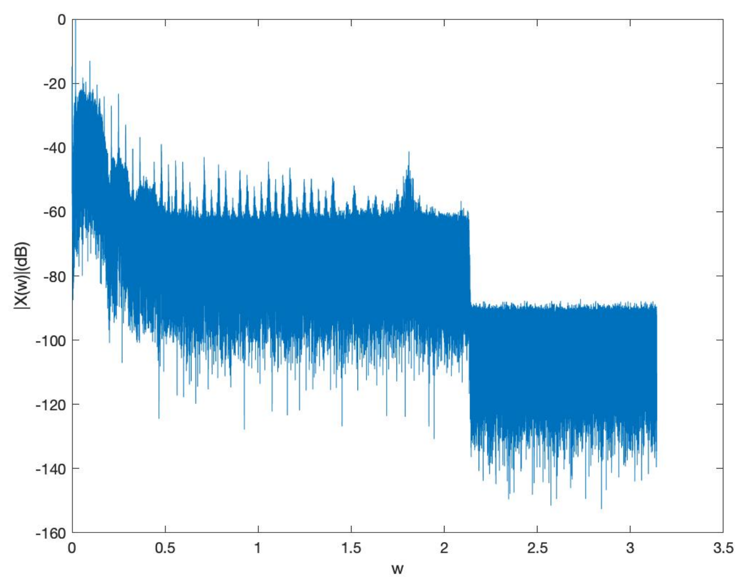

Figure 1: Original signal $|X(w)|$ dB

The power line noise, also known as the Mains Hum, is a result of power line interference that manifests itself in the form of a tone, whose fundamental frequency is around double the power line frequency of 50 Hz or 60 Hz, depending on the country. In the frequency domain the harmonic note is represented by peaks at the fundamental frequency f_0 and its odd integer harmonics.

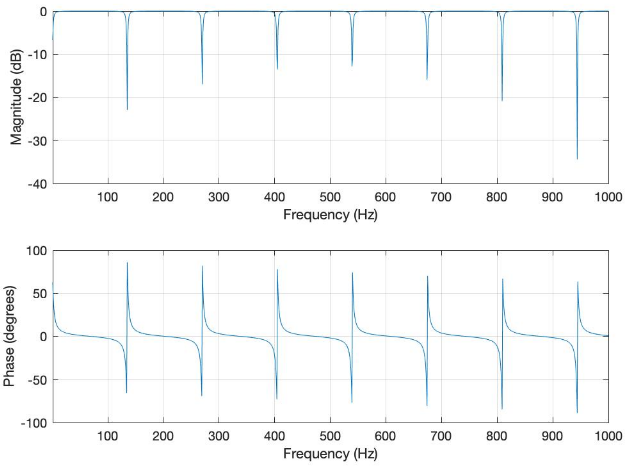

Figure 2: Original comb filter magnitude response

1.1 Comb Filter

The standard approach removing harmonic a tone noise is to use a comb filter, which filters out the fundamental and harmonic frequencies of the note using a series of notches, as shown in the Figure 2. An IIR comb filter is described by the difference equation

$$g[n] = \alpha g[n-K] + bx[n] - bx[n-K] \tag{1}$$

where K is the order of the filter. The filter will have $K+1$ evenly spaced notches on the unit circle ranging from 0 to 2π. The transfer function $H(z)$ for the filter is given as

$$G(z) = \alpha G(z)z^{-K} + b\left(1 - z^{-K}\right)X(z)$$

$$H(z) = \frac{G(z)}{X(z)} = b\frac{1 - z^{-K}}{1 - \alpha z^{-K}} \tag{2}$$

1.2 Adaptive Line Enhancer

To remove the background white noise I wanted to try using Adaptive Noise Cancelling with the LMS algorithm, since eliminating undesired noise is a classical Adaptive Signal Processing problem. A typical noise cancelling application requires the use of a reference noise signal. Since there was no reference noise available, I used the Adaptive Line Enhancer (ALE), which is a modification of the Adaptive Line Canceller, with the external reference signal replaced by a delayed version of the input signal $x(n) = s(n-L) + v(n-L)$, where $s(n)$ is the periodic signal $v(n)$ is broadband noise, and L is the delay (measured in samples). Its role is to suppress broadband noise and enhance, or highlight, narrowband or sinusoidal signals. The ALE is ensured to perform properly when $v(n-L)$ in the delayed signal $x(n)$ is not correlated with $v(n)$, while $s(n-L)$ and $s(n)$ are still correlated. Since periodic signals have periodic correlation, the correlation remains after L delay intervals, while wideband noise loses it correlation due to the same delay [5].

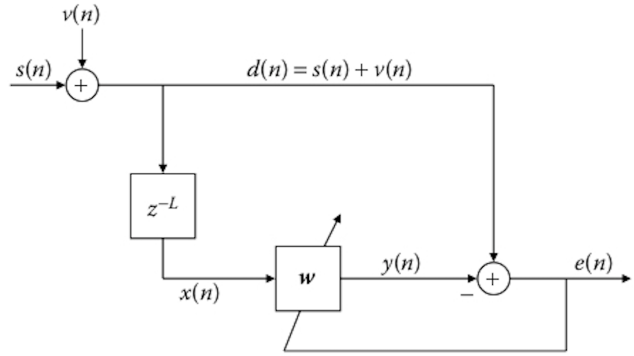

Figure 3: The Adaptive Line Enhancer
[4]

2 Procedure

2.1 Removing the Power Line Tone

Since the fundamental frequency of the tone is twice that of the power line frequency, I expected the fundamental frequency of the tone to be somewhere in the range of 110-140Hz. Additionally, given the volume of the tone in the audio, it should correspond to the samples with the largest magnitude in the frequency magnitude representation of the signal $|X(f)|$. The sample at $f = 134.7$Hz had the greatest magnitude and, in fact, due to the large magnitude difference compared to other frequencies, it actually obscured much of the rest of the signal content.

As shown in Figure 4, at a first glance the tone appeared to be represented by a single narrow peak, which led me to assume that the noise was of very narrow bandwidth. To keep the bandwidth of the filter no wider than necessary in order to avoid removing desired frequency content, I used a filter with a base frequency of $f_b = 134.7$Hz and default bandwidth settings $bw = 1.7454 \times 10^{-4}$ Hz (Figure 2). With the sampling frequency $Fs = 41000Hz$, this resulted in a filter of order $K = floor(Fs/fo) = 327$.

Playing the filtered output $g(n)$ confirmed that the main tone had been removed, but there were also sporadic tones in the output, which suggested that the filtering did work as intended. This was confirmed by the plot of the output $|G(f)|$ in Figure 5, which showed that the filtered signal had retained much of the noise. This indicated that (1) the filter may not be functioning properly, (2) because the signal had such low SNR, I might have misidentified the fundamental frequency of the tone, or (3) there might be several harmonic tones with different fundamental frequencies that I had failed to identify.

I recorded and plotted all the peaks to see if there should be any pattern that might help me identify any relationships between the frequencies of the peaks. Specifically, I was looking to see if they would fall in a single line (single tone), or multiple lines (multiple tones).

As shown in Figure 6, the plot looked close enough to a straight line to indicate that there was a single fundamental frequency. To get a better sense of the numbers, I calculated Δf's between the peaks. For the linear region the mode $\Delta f = 269$ Hz, which was close to twice the fundamental frequency f_o=134.7Hz that I had initially identified. In fact, for the the linear region $\Delta fave/2 = 134.69$ Hz. Additionally the frequency corresponding to each peak divided by $\Delta fave/2$ resulted in an odd integer, which indicated that they were odd integer harmonics $f_k = kf_o$ of the same fundamental frequency. Table 1 lists the recorded frequencies f_k of peaks, Δf's between the peaks, and the quotients $k = f_k/\Delta fave/2$.

With the confirmation that I had initially identified the frequency fo of the tone correctly, and that there were no other harmonic tones present in the signal, I concluded the filter had functioned as specified, removing the large peak at $f = 134.7$Hz, but it didn't have large enough bandwidth to eliminate all the harmonics content around the fundamental frequency. A plot of the input $|X(f)|$ between 130Hz and 150Hz (Figure 7) revealed that the bandwidth of the noise ranged from about 133.8Hz to 135.8Hz, having a bandwidth of 2Hz, while the larger-magnitude part of the noise with a bandwidth of approximately

Table 1: Frequency peaks from the initial output $|G(f)|$

frequency peaks f (Hz)	deltas between peaks (Hz)	f/delta$_{a}ve/2$
73.71	0	0.547256663
93	19.29	0.690474423
134.6	41.6	0.999331799
437	302.4	3.244487341
518	81	3.84586829
673	155	4.996658995
783.7	110.7	5.818546291
873.6	89.9	6.4860049
940	66.4	6.978988789
1039	99	7.714009949
1212	173	8.998440864
1481	269	10.99561957
1750	269	12.99279828
2019	269	14.98997698
2288	269	16.98715569
2559	271	18.99918331
2799	240	20.78105279
3096	297	22.98611627
3365	269	24.98329497
3634	269	26.98047368
3903	269	28.97765239
4172	269	30.97483109
4442	270	32.97943426
4714	272	34.99888633

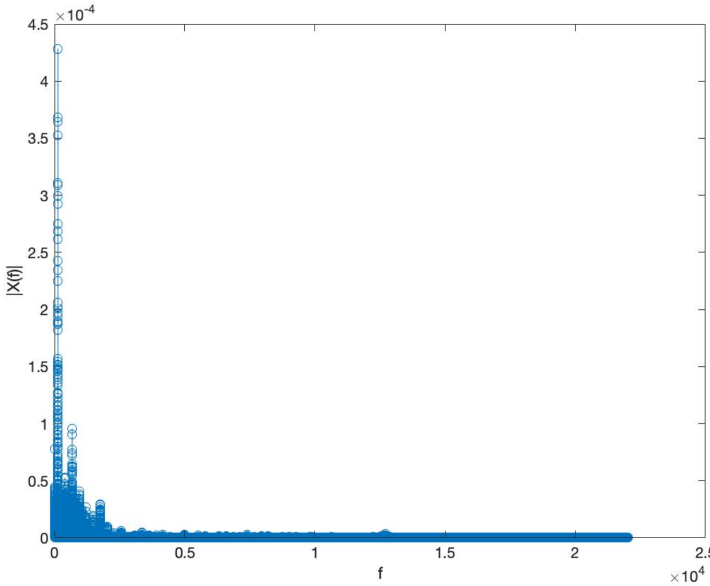

Figure 4: Input signal frequency magnitude $|X(f)|$

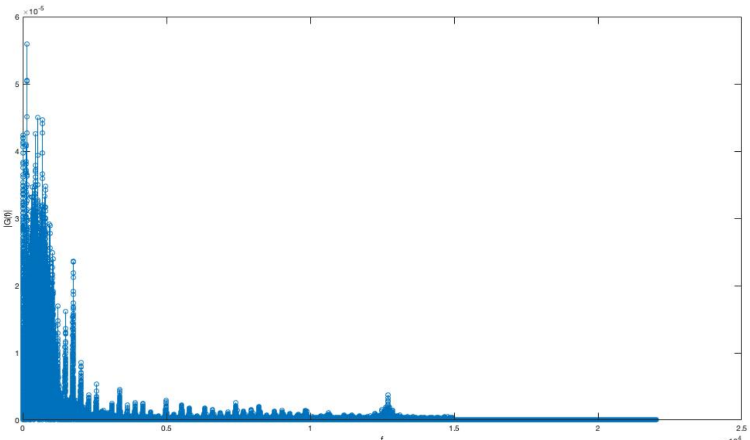

Figure 5: Initial comb filter output $|G(f)|$ with unfiltered peaks

1.2Hz ranged from roughly from 134.3Hz to 135.5Hz. The base frequency $fb = 134.7$ Hz that I had initially specified for the comb filter was right in the middle of that range.

In order to avoid having to use multiple filters, since the maximum allowed Matlab comb filter bandwidth is $0 < bw < 1$, I redesigned the filter to have a base frequency $fb = (134.4 + 135.4)/2$ Hz $= 134.9$Hz and $bw = .97$ (Figure 9). As shown in Figure 8, the redesigned filter eliminated all harmonic tone noise in the region of the fundamental frequency fo. Additionally, Figure 10 shows that all harmonics were also successfully eliminated, in contrast to Figure 5. Removing the harmonic tone noticeably improved the SNR of the signal, as shown in Figure 11. A slightly smoother sounding output resulted with a filter using $fb = 134.8$Hz and $bw = .97$, which I kept as the final filter values.

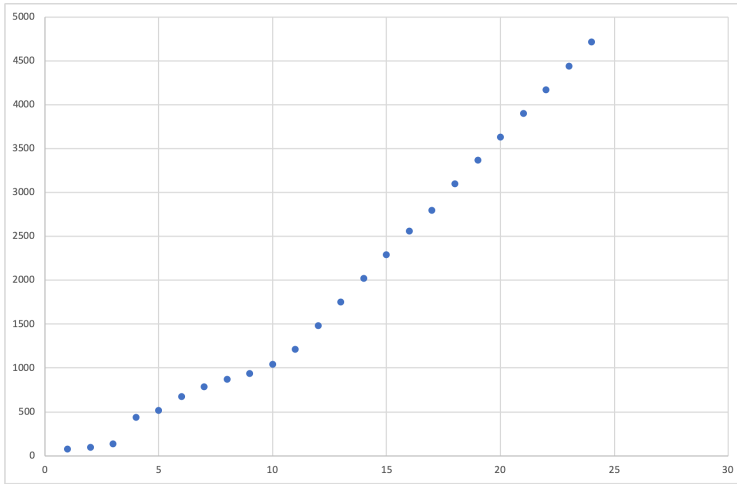

Figure 6: Recorded frequency peaks in output $|G(f)|$

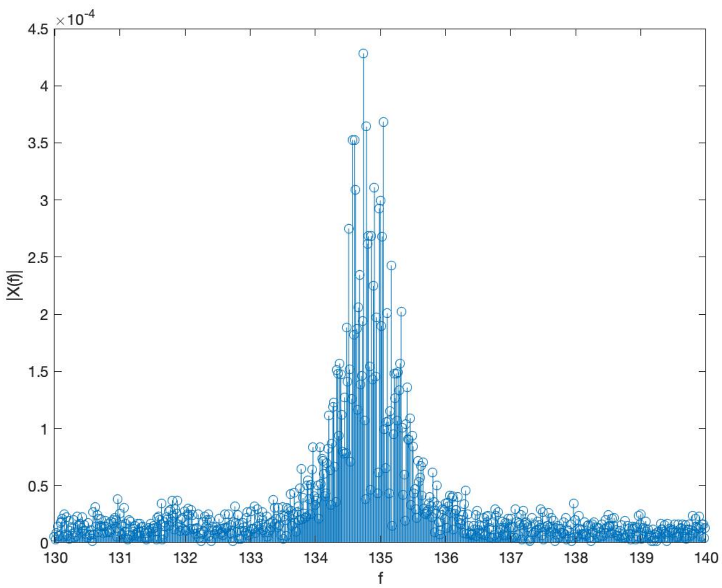

Figure 7: $|X(f)|$ noise bandwidth

2.2 Removing White Noise with ALE

The performance characteristics of the ALE are driven by the delay L and filter size M. By the Fourier Transform time-scaling property(wide spectrum in the frequency domain corresponds to a short duration in the time domain), the delay value L need not be large to remove white noise, since white noise is by definition broadband. The ALE will try to remove any frequency whose correlation length is smaller than the delay used in the reference signal. The ALE was implemented using the DSP library NLMS algorithm to make the filter more robust to input magnitude variations. I was able to achieve a combination of effective background noise reduction and good sound quality with values $L = 3$, $M = 64$, and mu = .00001. The very small learning rate was rather surprising, as it was expected to be around .005 $<= mu <=$ .01. The intent was to use the smallest possible delay L. The value $L = 3$ was chosen based on the autocorrelation of the signal, where the autocorrelation was high enough for the signal not

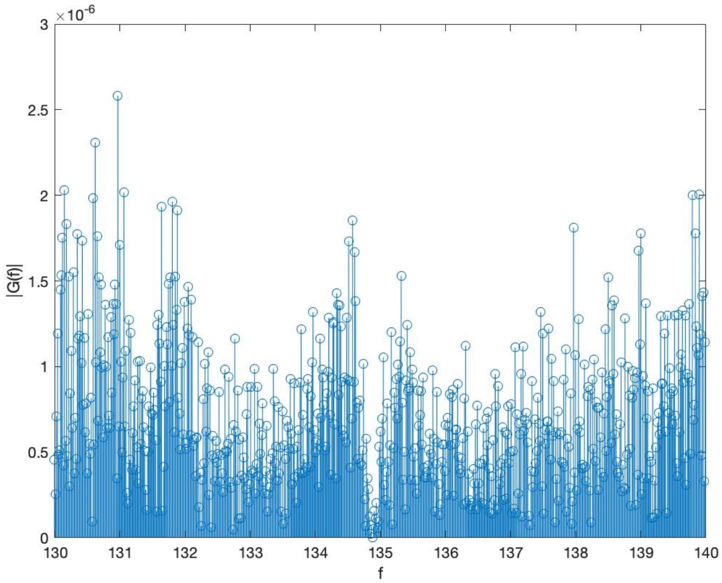

Figure 8: output $|G(f)|$ with noise removed

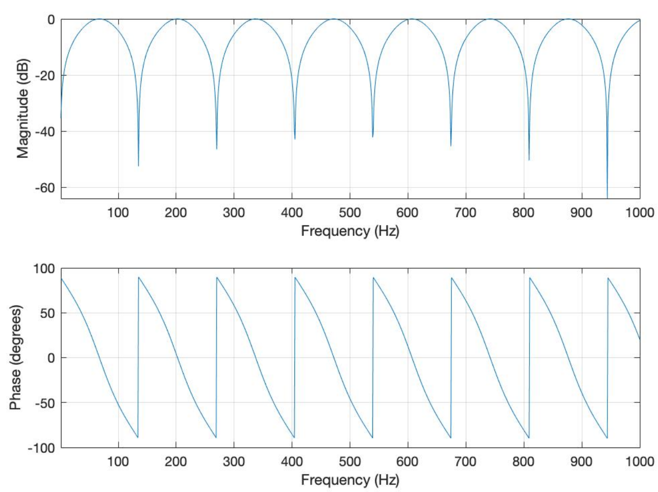

Figure 9: Final comb filter magnitude response plot

to be distorted. Compared to Figure 11, Figure 14 shows substantial reduction in noise across the entire spectrum. It additionally shows greater suppression for higher frequencies, which are more audible as background noise.

The major obstacle in removing the background noise using the ALE turned out to be the fact that the initial assumption that the background noise was all white noise was incorrect. After several unsuccessful attempts to use the ALE to eliminate loud buzzing and hissing noise, a closer look at the frequency content of the last few seconds of the output (Figure 15), where both music and speech were absent, revealed regularly occurring peaks, indicating that the background noise also had a harmonic component to it. Unlike the power line harmonic tone of $bw < 2Hz$ removed by the comb filter, these harmonics appeared to have a bandwidth of around 138Hz. This meant that the background noise had periodic

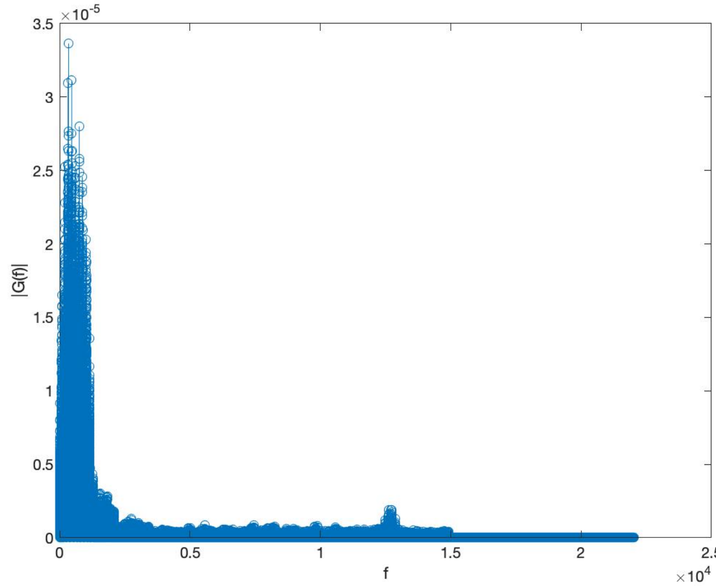

Figure 10: Final comb filter output $|G(f)|$

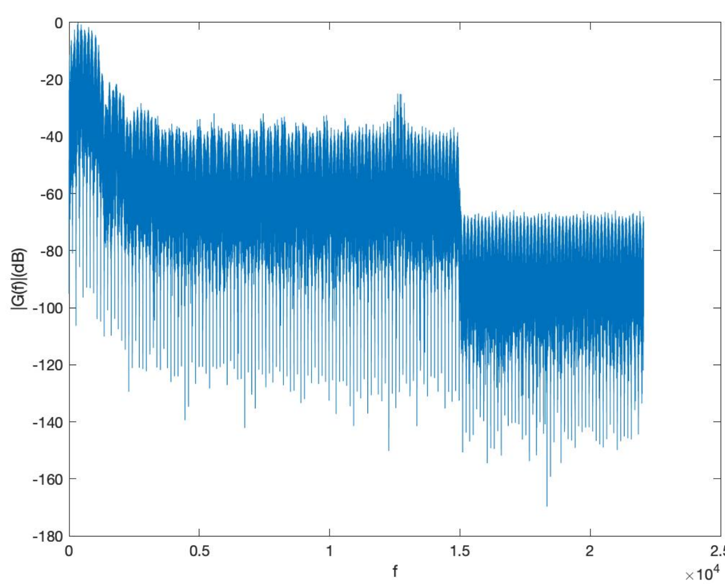

Figure 11: Final comb filter output $|G(f)|$ dB

correlation, which made the ALE ineffective at removing the correlated part of the noise. Using a larger delay L was considered as an option to attempt to remove the harmonics, but it would come at a cost of distorting the signal. As a result, although I was able to achieve reduction in background noise, the harmonic elements of the noise remained in the output. Comparing Figure 15 to Figure 16 shows a difference in the magnitude of the frequencies without much change in the frequency content, indicating a uniform reduction across the spectrum. This can be attributed to whine noise reduction in the signal.

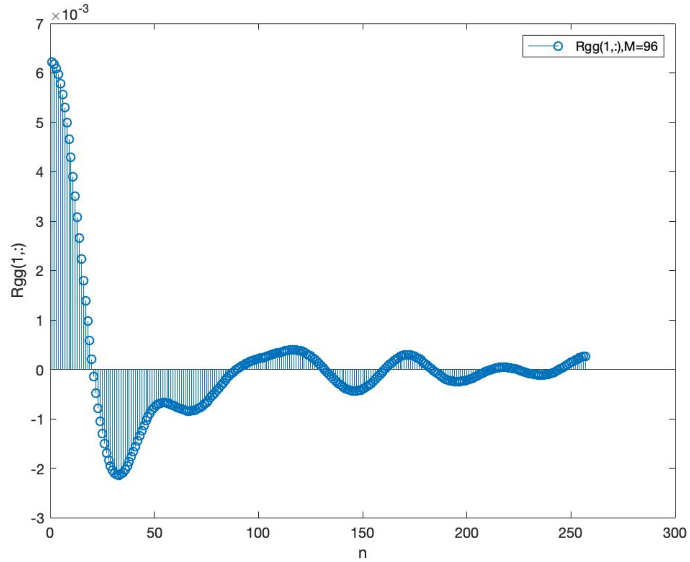

Figure 12: Autocorrelation of $g(n)$

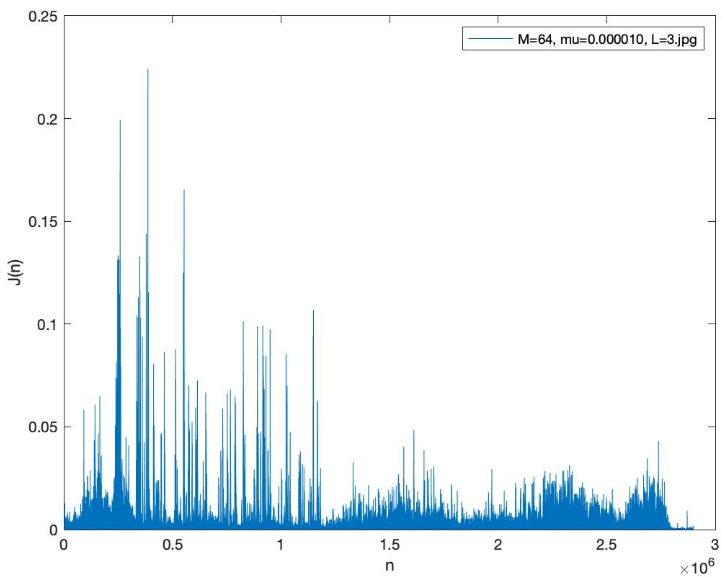

Figure 13: NLMS error curve

3 Abandoned Approach: Removing Harmonic Tone with ALE

My initial idea for the project was to eliminate the harmonic tone using an ALE. An ALE using the NLMS algorithm filter of size $M = 256$ and learning rate $mu = 0.01$ and a delay $L = 112$ substantially suppressed the tone, which was very encouraging. Figure 17 shows substantial improvement in the SNR of the signal. In fact, the figure shows that, unlike a comb filter, the ALE suppressed the fundamental frequency of the tone and its harmonics without actually eliminating them. This may serve as a viable approach to the

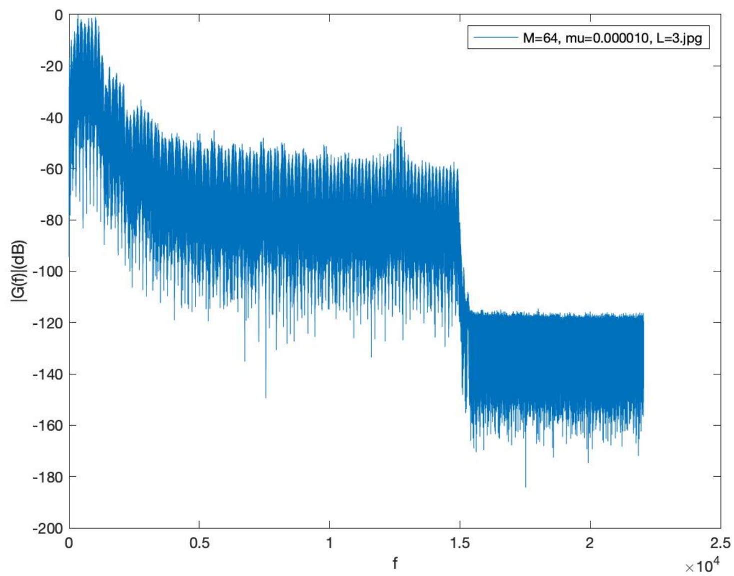

Figure 14: The ALE output $|Y(f)|$ dB

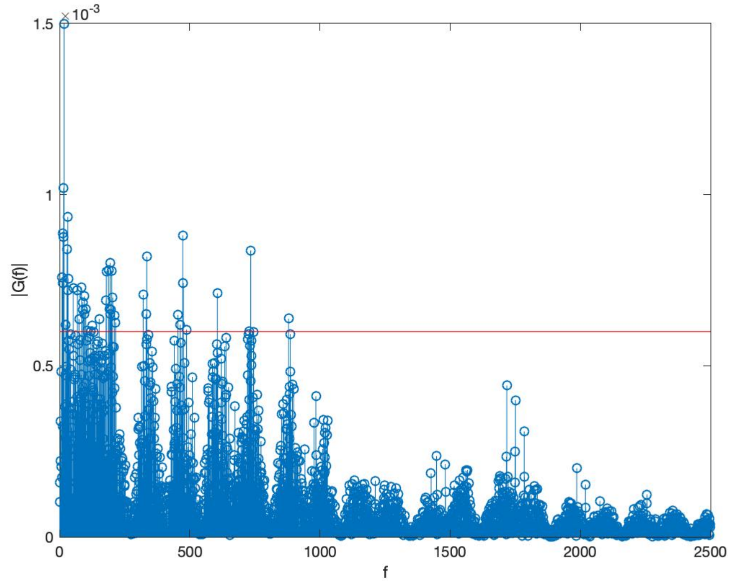

Figure 15: FFT of last few seconds of $g(n)$

problem if the desired outcome is to bring the rest of the signal to the foreground without necessarily eliminating the noise completely, but rather suppressing it. This, however, demonstrates one disadvantage of this method, as subsequent attempts to remove white background noise by passing it through another ALE resulted in amplification of the suppressed harmonics, which pointed to an interesting property of the adaptive noise canceller in general: The adaptive algorithm essentially functions as a linear predictor that estimates the periodic content of the signal. Then depending on what we select as the output of

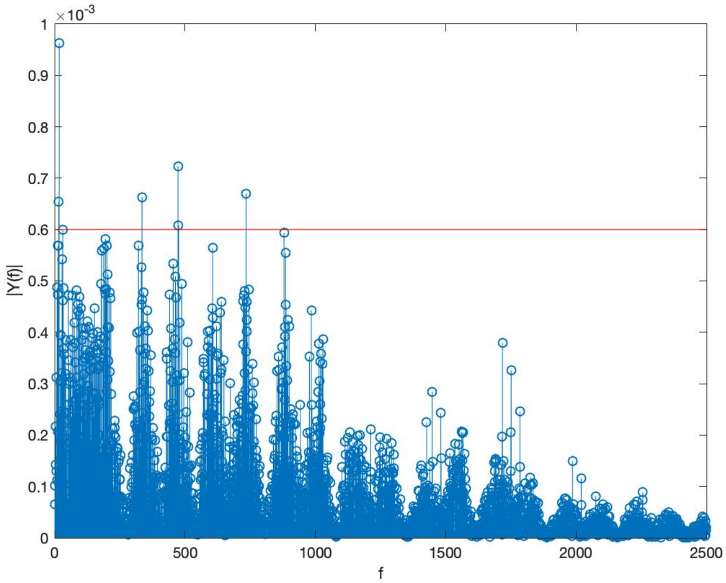

Figure 16: FFT of last few seconds of ALE output $y(n)$

interest, we can choose $e(n)$ to suppress the periodic noise in the input, or $y(n)$ to suppress the random noise. Since $e(n)$ was the selected output for this application, it had to be amplified by a factor of 3 to obtain volume comparable to the input.

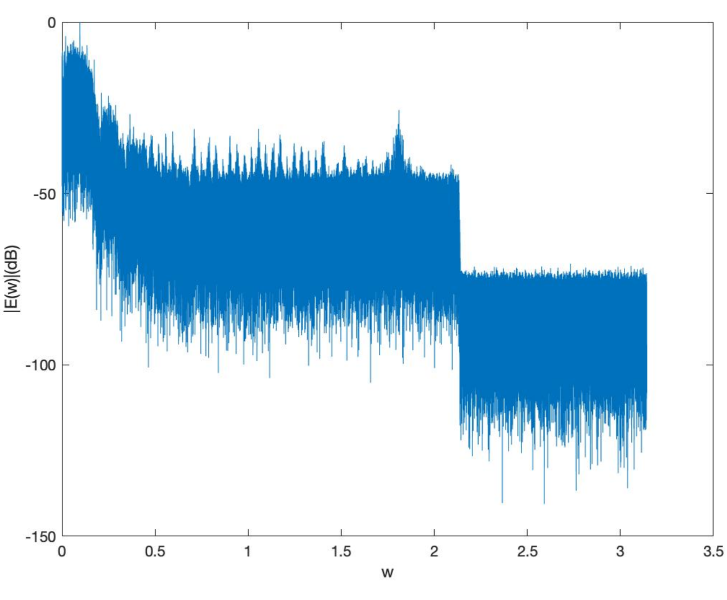

Figure 17: ALE tone canceller output $|E(f)|$

4 Conclusion

My objective for the project was to improve the audio quality in the linked video. Since the harmonic note was the predominant source of noise, I initially focused on removing it, and thought it would be a very interesting exercise to use Adaptive Signal Processing techniques to do it. As mentioned in the section above, although I was able to remove the tone, the quality of the rest of the signal suffered as well. As a result, I decided to treat the results as a demonstration of the concept of being able to remove a tone with an ALE, and shifted my focus to removing the tone with a comb filter instead.

Once I was able to properly tune the comb filter, it seemed like a very straight forward way of removing the harmonic tone. If fact, the bigger lesson learned was to investigate why a particular approach to solving a problem doesn't work while it should. It also demonstrated the contrast between the comb filter and the adaptive filter. Once I properly identified the frequency of the noise, I was able to remove it without distorting any other part of the signal.

Next since I wanted to do better than just remove the tone, after referencing a couple of papers discussing using an ALE as a means of removing background noise in communications system, I returned to trying using it for this problem. The results showed the effectiveness of the ALE at removing white broadband noise. Additionally, it showed the ALE was ineffective at removing non-white noise in the form of popping or buzzing noise, which was the harmonic part of the noise seen in Figure 15 due to its periodic correlation as a result of harmonics. From the sound of the noise, it could be reasoned that the corresponding frequencies are in the vicinity of the original harmonic tone frequency $f0 = 134.7Hz$. However, given the very large number of peaks, it proved practically impossible to identify the exact peaks responsible for the tones.

Selecting a larger filter size M did appear to help reduce the harmonic noise in the background but at the cost of deteriorating the stereo effect of the signal and making the output sound muffled. Filter size $M = 32 or 64$ was chosen as the best compromise between the reduction of background noise and keeping the desired portion of the signal relatively intact. A larger filter size M allows the ALE to more effectively realize a narrow-band LPF, which kills more higher frequency contents. For purely white background noise, I believe the ALE could utilize a smaller filter size and still deliver desired results.

Given the very large bandwidth of the harmonics present in the background noise, it would be infeasible to try to construct a comb filter to match the bandwidth or to use a series of comb filters, since the number of filters required would be impractically large. Trying to remove the highest energy peaks produced no results. Moreover, since the peaks occupy about the same frequency spectrum as the desired part of the signal, it may be impossible to use a non-comb filter without also removing parts of the signal indented to be preserved and improved. As a result, it might be more appropriate to use space methods as mentioned in the following section.

5 Further Methods to Explore

An interesting method that may prove effective in eliminating random background noise in the signal is applying Singular Value Decomposition to the Henkel Matrix of the input [3]. It is used to divide the signal space into desired signal and noise, where the singular values corresponding to noise can be set to zero, thus eliminating the noise. I came across the method after unsuccessfully attempting to apply PCA, as described in Sebastian Raschka's Python Python for Machine Learning book, to the Fourier Transform of the data matrix $F[X(n)]$ of the input signal . While the results were far from acceptable, I was encouraged by the fact that I could recover a playable audio signal, which led me to researching whether there was a similar, more robust method. If like PCA this method relies on there being a distinct, if not sharp, transition between the singular values of noise and signal so that a distinction between them could be made, then for this particular application, the method would still require the use of a comb filter to eliminate the harmonic tone, as given the magnitude of the tone, the rest of the signal content might seem like noise in terms of magnitude.

References

[1] Jonathan Cedarleaf, Steve Philbert, and Arvind Ramanathan. *NOISE CANCELLATION USING LEAST MEAN SQUARES ADAPTIVE FILTER*. University of Rochester, Department of Electrical and Computer Engineering.

[2] R. Lyons. *Understanding Digital Signal Processing (3nd Edition)*. Prentice Hall, 2010.

[3] Rahim Mahmoudvand, Mohammad Zokaei, and Shahid Beheshti. On the singular values of the hankel matrix with application in singular spectrum analysis. *Chilean Journal of Statistics*, 3(1):43–56, 2012.

[4] Alexander D. Poularikas. *Adaptive Filtering*. Routledge, 2014.

[5] Tianshuang Qiu and Ying Guo. *Signal Processing and Data Analysis*. De Gruyter Textbook, 2018.

[6] Ali O. Abid Noor Roshahliza M. Ramli and Salina Abdul Samad. A review of adaptive line enhancers for noise cancellation. *Australian Journal of Basic and Applied Sciences*, 6(6):337–352, 2012.

[7] M.A.Josephine Sathya and Dr.S.P.Victor. Noise reduction techniques and algorithms for speech signal processing. *Australian Journal of Basic and Applied Sciences*, 6(1):317–322, 2015.