Math 260 Project: Wiener Filter in Frequency Domain

Adi Pall

Nov. 2020

1 Description

1.1 Overview

The `wien_f` function implements the Wiener filter in the frequency domain. It takes in a noisy signal, the number of samples to window over per FFT, and the number of samples to shift the window per FFT. Optionally, the type of window to use can be specified, and is rectangular by default. Similar to the time domain implementation in SciPy, the signal variance is used for an estimate of the noise. In this implementation, the variance is related to signal power, $|E_k|^2$. The variance can translate to the frequency domain by an application of Parseval's theorem:

$$\sum |e(n)|^2 = N \sum |E_k|^2$$

Or, when N is sufficiently large,

$$N\sigma_e^2 = N^2 |E_k|^2$$

The variance of a noisy signal is found using a helper `get_stats` function for the same size windows used by the Wiener filter. This function finds the variance in each window, sliding the window over by one sample each time. An array of variances is returned, and the filter averages all of these together to produce an estimate of the noise power. This estimate is a constant for the whole signal, even though FFT and filtering is performed separately for each window.

The noise power estimate $|N'|^2$ is used to determine the optimal Wiener filter, and the filter is applied to the combined signal in the frequency domain:

$$S' = \frac{|C|^2 - |N'|^2}{|C|^2} C$$

Taking windows can improve the Power Spectral Density (PSD) estimate, and this becomes important for long signals. However, leakage can occur whereby frequency components from adjacent windows affects the spectral power for the window in consideration. This leakage can be minimized by using a window that gradually attenuates frequencies on the edge of the window. `wien_f` supports use of the Hanning window for this reason.

1.2 Challenges

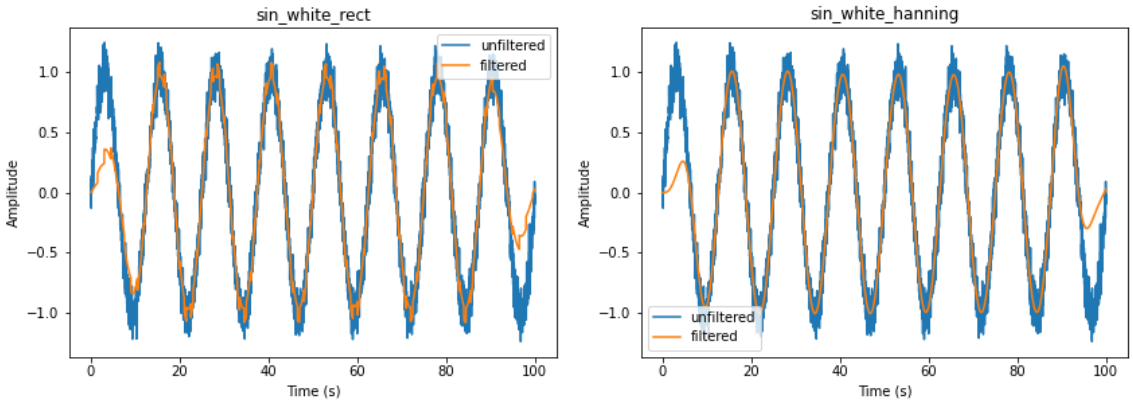
The goal of this project was to create a fairly robust algorithm by which different types of noise could be handled. To begin, a hyperbolic tangent was used to fit the PSD and the noise region was considered to be where the gradient is zero. However, this method was not sufficient for time-varying white noise. Therefore, through research it was determined that the best way to estimate noise power would be using the variance.

To improve the PSD estimate and filtering, windowing was implemented. This reduced the resolution of the filtered signal, thereby diminishing the accuracy of the filter. To overcome this, overlapping windows were used. Windowing introduced some scaling issues, which was countered by normalizing the PSD by the window size. Additionally, after filtering, the inverse transform of the output signal is scaled by an "overlap factor" which is the ratio of the window shift to window size.

There were some miscellaneous issues to overcome as well. For example, if the number of samples is not divisible by the window size, there will be samples left over. Another issue highlighted during the development process was that maintaining phase information for both the filter and the combined signal is crucial for successful filtering. Finally, there were small regions where the variance (noise estimate) exceeded the actual combined signal power. This resulted in negative amplitudes at those frequencies, and showed up as wild fluctuations in the time domain. Forcing these to be zero avoided this problem with no visible negative consequences.

2 Discussion

The Wiener filter was tested with both generated signals and audio samples. White noise and time-dependent white noise were added manually to the signals. The impact of using a rectangular versus Hanning window was also compared. The filter's success was determined by graphical comparison between the noisy signal and the filtered signal. Three examples are provided below in Figures 1-3, but many more are contained in the `/demo/` folder. The first observation to be made is that for time-varying white noise the filters allow more noise to pass over time, which was expected since a constant variance is used for the noise estimate, across all windows. With further development, this should be modified.



(a) Rectangular Window

(b) Hanning Window

Figure 1: Visibly smoother signal extracted with sinusoidal signal using Hanning window

Figure 1 shows that the Hanning window performs better for the sinusoidal signal. The windows have a more nuanced effect on the impulse signal. For example, the Hanning window let pass more white noise than the rectangular window, but the Hanning window performed better than the rectangular window in the actual impulse waveform region with time-varying noise. However, for both sinusoidal and impulse signals, the Hanning window provided a more uniformly filtered signal. The lack of uniformity seen in some places with the rectangular window is attribute to the "leakage" discussed earlier. For example, in Figure 2, there are small spikes in the filtered signal in a symmetric pattern around the impulse waveform.

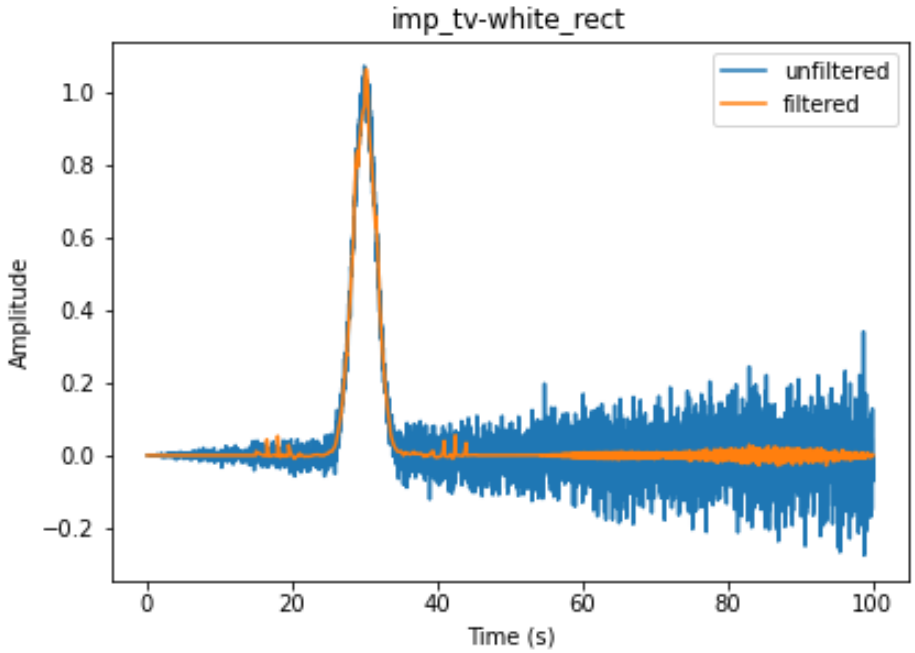
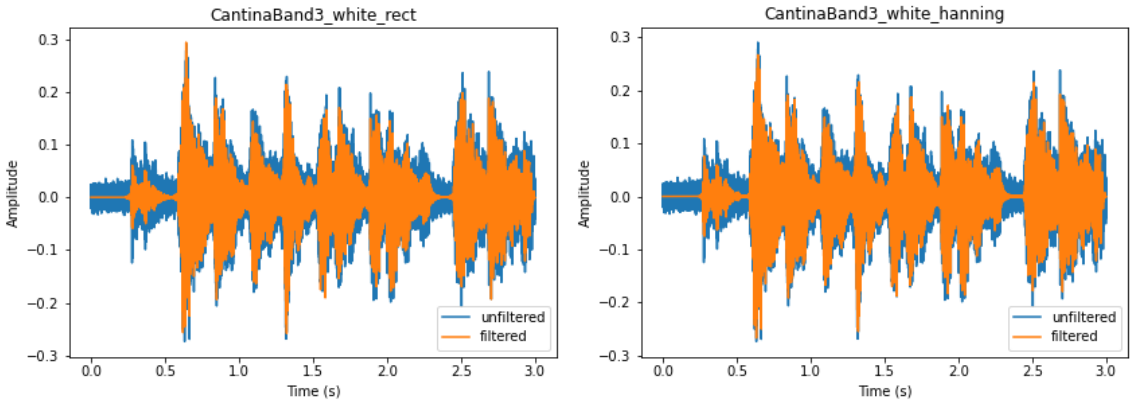


Figure 2: "Leakage" effects around the impulse waveform

It was gratifying to test the filter on audio signals, because the added white and time-varying white noise could be heard clearly. White noise presented as static in the background. Since the time-varying white noise was linearly increasing, this static is not heard at the beginning, but becomes apparent near the end. This static provided a tangible marker for determining success of the filter. Overall, both windows were able to de-noise the white and time-varying white noise with no discernible difference when listening to the output. There was some noticeable difference when comparing the signals graphically (Figure 3). When comparing the audio before adding noise to the output audio after adding noise & filtering, the output audio sounds slightly muffled.



(a) Rectangular Window

(b) Hanning Window

Figure 3: Slight difference between the two window types

References

[1] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. (2007). Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press. ISBN: 0521880688

[2] Robertson, Neil, "Evaluate Window Functions for the Discrete Fourier Transform" https://www.dsprelated.com/showarticle/1211.php

[3] Welch, Peter D., "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms", IEEE Trans. Audio and Electroacoustics, vol AU-15, pp. 70 – 73, June 1967.