

A PACKAGE TOOL FOR GENERAL-PURPOSE SIGNAL DENOISING

Filipe C. da C. B. Diniz and Sergio L. Netto

Electrical Engineering Program/COPPE
Federal University of Rio de Janeiro
P.O.Box 68504, Rio de Janeiro, RJ, 21941-972, Brazil

ABSTRACT

This paper describes a broad package composed of signal denoising techniques in a uniform framework. The tool aims at simplifying the general tuning of the denoising techniques, by allowing complete visualization of the results and of the parameter influence on the overall performance. The methods implemented include spectral subtraction, Wiener filtering, Wavelet transform, and adaptive filtering. Performance evaluation for each method can be assessed through signal-to-noise ratio, crossed correlation, and amplitude reduction criteria. Examples included illustrate the broad applicability for the package tool.

1. INTRODUCTION

In many applications, the signal of interest is corrupted by a large amount of additive noise, which makes it necessary some sort of signal pre-processing. The modern techniques for signal denoising are greatly influenced by the actual application. Many parameters are involved in the process and the way they interfere depends on the characteristics of the signal and the noise itself.

This work describes a software package that includes several denoising techniques in a unified and easy-to-use framework. The tool provides a friendly graphic user interface (GUI) and makes all these methods available to the practitioner in a uniform setup. For instance, with such package one can easily visualize the effect of varying any parameter on a denoising algorithm, and fine tune the algorithm to the application at hand. The methods included are spectral subtraction, Wiener filtering, Wavelet transform, and adaptive filtering.

The remaining of this paper is organized as follows: Section 2 introduces the package capabilities for the spectral subtraction and Wiener filtering methods; Sections 3 and 4 describe the tool for the Wavelet and adaptive filtering approaches, respectively; each of these previous sections includes a brief description of the corresponding algorithm along with all functionalities of the developed software; Section 5 discusses some performance evaluation criteria for the filtering algorithms made available within the package tool; Section 6 presents a case study of denoising in which the package has been applied; Section 7 concludes the paper emphasizing its main contributions.

2. SPECTRAL SUBTRACTION AND WIENER FILTERING

The power spectrum density (PSD), $S_x(z)$ for a corrupted signal can be expressed as

$$S_x(z) = S_s(z) + S_n(z) \quad (1)$$

where the right-hand terms correspond to the PSD of the clean signal and of the noise, respectively. Hence, one may want to determine the PSD for the original signal just by subtracting the noise PSD from the corrupted signal PSD. This is the basic idea behind the spectral subtraction denoising algorithm [1].

In practice, the noise PSD may be estimated from a time interval or from a frequency range of the corrupted signal where the original signal is absent. After the subtraction, the original signal can be obtained (except for a constant) from its PSD estimate, if some properties like stationarity and ergodicity are imposed within windows of K samples. In such cases, the PSD $S_x(z)$ can be estimated by the periodogram $|X(z)|^2$ of each window. The sampled version of this periodogram is then related to the the square of the DFT magnitude.

One can also consider the filtering operation that minimizes the mean square error (MSE) between the estimated spectrum and the original one. Such approach is commonly known as the Wiener filter, whose transfer function is determined by [2]

$$H(z) = \frac{S_s(z)}{S_x(z)} = \frac{S_s(z)}{S_s(z) + S_n(z)} \quad (2)$$

which can be rewritten as a function of the signal-to-noise ratio (SNR), as given by

$$H(z) = \frac{\frac{S_s(z)}{S_s(z)}}{\frac{S_s(z)}{S_s(z)} + \frac{S_n(z)}{S_s(z)}} = \frac{1}{1 + \frac{1}{SNR(z)}} \quad (3)$$

This relation indicates that the Wiener filter acts more intensively in the frequency ranges where the SNR is lower, and vice-versa.

To obtain $H(z)$ from the noisy signal and noise PSD's, $S_s(\omega)$ and $S_n(\omega)$, respectively, by the same assumptions made for the spectral subtraction, one can write that

$$H(z) = \frac{S_x(z) - S_n(z)}{S_x(z)} = \frac{|X(z)|^2 - |N(z)|^2}{|X(z)|^2} \quad (4)$$

This expression also indicates the close relationship between the spectral subtraction and Wiener filtering methods.

2.1. Software capabilities

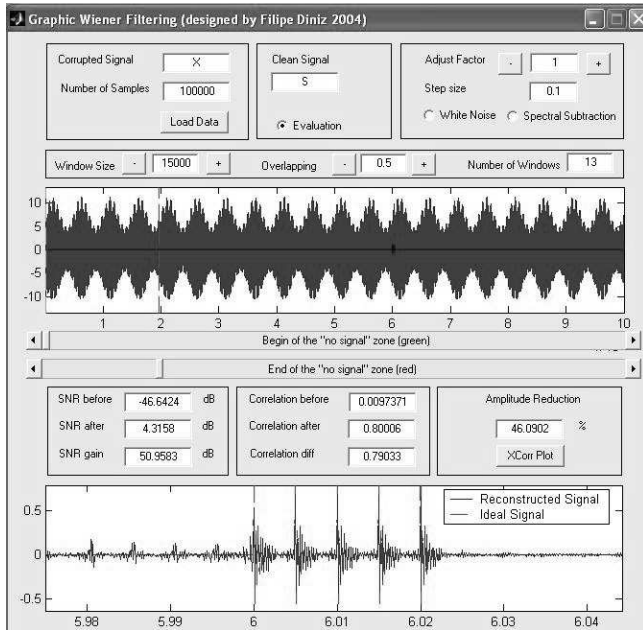


Fig. 1. GUI for denoising with spectral subtraction and Wiener filtering.

The three GUI's described in this work contain some components in common:

- i **Edit box for loading the corrupted signal.** That signal is originally stored in a Matlab ¹ variable and can be plotted. As soon as it is loaded, by entering the name of the variable and pressing **Load**, the total number of samples is displayed.
- ii **Edit box for loading the original signal.** This signal is, of course, unavailable in a real situation but it is crucial for determining the performance of the system. When this signal is available, as in a synthetic signal, it should be activated by setting the radiobutton **Evaluation**.
- iii **Performance evaluation criteria.** The criteria described in Section 5 are displayed to the user if the radiobutton **Evaluation** is marked. The SNR is calculated before the denoise, after the denoise and the difference between these two values is displayed. The same occurs to the crossed correlation and the amplitude reduction.
- iv **Plot for the original (ideal) signal and the processed signal.** This plot is updated as soon as any parameter value is changed.

¹Matlab is a trademark of the Mathworks Inc. [3]

The others components are specific of each tool. For the current GUI, they are:

- i **Windowing control.** It is possible to set the number of windows and their overlapping factor. Such windows are used in the estimation of signal PSD.
- ii **White noise radiobutton.** If the radiobutton **White Noise** is marked, the PSD estimation is made constant.
- iii **Spectral subtraction radiobutton.** If the radiobutton **Spectral subtraction** is marked, the denoising method is changed from Wiener filter to spectral subtraction.
- iv **“No signal” zone control.** Using the two sliders, it is possible to adjust the interval where the presence of signal is not significant and, hence, is formed basically by noise. Such interval is then used to estimate the noise PSD.
- v **Adjust factor control.** The PSD estimation may be affected by a small number of samples. To adjust the PSD value manually, the edit box **Adjust factor** can be employed by the user.

3. WAVELET TRANSFORM

If one wants to analyze an impulsive signal (very concentrated in time), the Fourier transform is probably no more a good choice, as its base functions have infinite support. A simple alternative is supplied by the short-time Fourier transform (STFT), which is the Fourier transform of the signal multiplied by a moving constant-sized window. Each sample of the Fourier transform can be viewed as a sample of the output of a filter. The entire transform may be implemented by a filter bank. It is easy to notice that the number of samples of the STFT would be multiplied by the number of filters in the bank, which makes it necessary the decimation of each output.

However, as the size of the window is constant, the low-frequency components estimation may be harmed. As an evolution of this transform, the constant window turned into a varying-size window (which depends on the center frequency). So, for lower frequencies, there would be larger windows. Besides that, the decimation factor would also have to change depending on the center frequency. These two modifications in the STFT lead to the Wavelet transform [4]. In such approach, the desired signal presents significant components only within given bands of the wavelet domain, while the noise tends to spread all over. By properly eliminating or reducing the components outside the signal range, one can then significantly reduce the noise influence. Such procedure is commonly referred to as the shrinkage stage. The original signal is determined by applying the corresponding inverse Wavelet transform, thus returning to the time domain.

3.1. Software capabilities

The specific capabilities for the wavelet tool include:

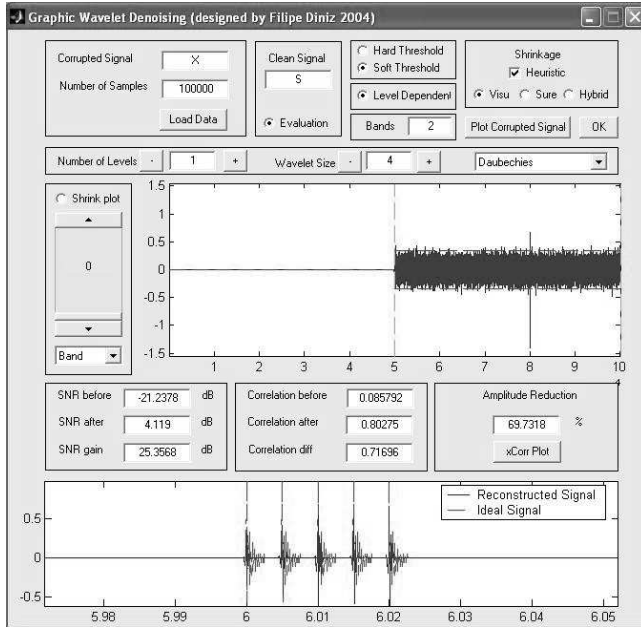


Fig. 2. Interface for denoise with Wavelet transform.

- i **Wavelet-mother control.** The user can select the wavelet family from the set: daubechies, symlets, coiflets and biorthogonal. The number of levels of decomposition and the size of the wavelet can also be selected.
- ii **Shrinkage method control.** The method for setting the shrinkage threshold can be selected among these options: Heuristic (where the user can choose the bands of interest in the **Bands** edit box), VisuShrink (also known as minimax [5]), SureShrink and Hybrid (which the best method between the last two is selected depending on the signal).
- iii **Thresholding control.** The user can define if the shrinkage would be made using hard or soft thresholding.
- iv **Manual thresholding.** After denoising, the user can manually select any band in the combo-box **Band** and, using the slider, put the threshold at any height.
- v **Wavelet bands plot.** This plot shows the wavelet bands separated by vertical dashed green lines. The horizontal dashed red lines indicate the thresholds for each wavelet band. If the user wants to verify the effect of the shrinkage under the current configuration, the **Shrink** radiobutton must be marked.

4. ADAPTIVE FILTERING

An adaptive filter [6] is a system whose behavior changes in order to minimize some norm of the error between its output and some reference signal. For denoising, as illustrated in Figure 3, the general behavior of the adaptive filter can be understood as follows: There is a signal of interest $s(k)$

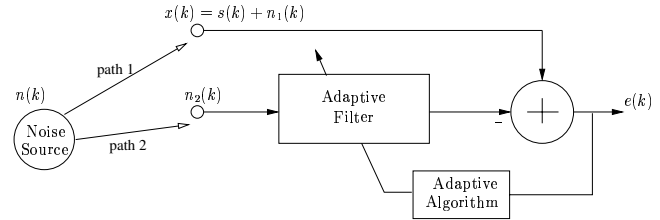


Fig. 3. Signal denoise configuration using adaptive filtering.

which must be estimated and a noise source $n(k)$. Consider the use of two sensors, one of them detecting the desired signal along with some version $n_1(k)$ of the noise, and the other detecting only a correlated version $n_2(k)$ of the noise. The central idea behind adaptive filtering is to use this second sensor to eliminate the noise component from the first one. In that manner, the denoising filter acts just like a vaccine that employs a sample of the disease to eliminate, or at least greatly reduce, its own effect.

4.1. Software capabilities

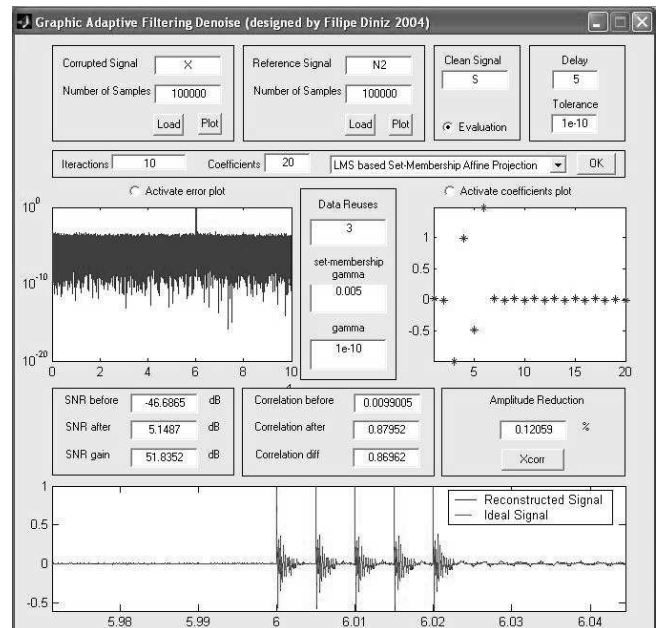


Fig. 4. Interface for denoise with adaptive filtering.

- i **Reference signal control.** This signal can be loaded in the same way as the noisy signal presented before in Section 2.
- ii **Algorithm control.** The user can, through a combo-box, select the algorithm: Least Mean Squares (LMS), Normalized LMS, Recursive Least Squares, and LMS-based set-membership affine projection [7]. The tolerance in the MSE can also be determined.

- iii **Filter characteristics.** The user can set the number of coefficients and the delay for the reference signal.
- iv **Number of iterations.** The number of iterations can be set by the user.
- v **Plots.** The plots for MSE (in dB) and for the coefficient values can be enabled and disabled.

5. PERFORMANCE EVALUATION CRITERIA

Three criteria were used to evaluate the performance of a specific denoise tool. In practice, it is essential to use more than one criterium in order to have a full idea of the impact of varying the parameters. For instance, in some cases, the SNR may not carry meaningful information on the waveform of the processed signal, whereas the crossed correlation may not retain information on the amplitude of the output signal, and so on. Hence the three following criteria are jointly used as in [8].

5.1. Signal-to-noise ratio

This well known measure of how much the signal of interest is corrupted is calculated by the ratio between the signal and noise powers, that is

$$SNR = 10 \cdot \log \left\{ \frac{\sum_{k=1}^L s^2(k)}{\sum_{k=1}^L [x(k) - s(k)]^2} \right\} \quad (5)$$

where $x(k)$ is the corrupted signal, formed by adding the original signal $s(k)$ to some additive noise $n(k)$, and L is the total length of that signal. The unit is dB.

5.2. Crossed Correlation

This measure is used in its normalized form described by

$$R = \frac{1}{L} \cdot \frac{\sum_{n=1}^L \left\{ [Y(n) - E[Y]] \cdot [S(n) - E[S]] \right\}}{\sigma_Y \cdot \sigma_S} \quad (6)$$

where $E[\cdot]$ denotes the expected value operator, σ_Y and σ_S denote standard deviations from the processed signal and the ideal signal and the value varies within $-1 \leq R \leq 1$.

5.3. Amplitude reduction

This measure is the percentage difference between the highest peak contained in the original signal and the highest peak in the processed signal. It denotes the amount of signal peak that is lost through the denoising process.

6. PRACTICAL EXAMPLE

The package tool has been applied to eliminate noise from the measurements of partial discharges in electrical transformers [8]. Such electrical discharges allow a complete

check-up on the “health” of a given transformer. This analysis can indicate if a certain device should be replaced or not, greatly reducing maintenance costs. The main problem in measuring such discharges derives from the fact that signal collecting must be made while the transformer is fully operational. In addition, nearby transformers or other large electrical devices can generate great amount of noise.

In this experiment, a clean electrical discharge was obtained in a well insulated device, and subsequently corrupted by noise. Such corrupted signal was then used to test the performance of each method in this specific problem. The results from the denoising stage of the measuring process are illustrated in Figures 1, 2, and 4, for the respective methods presented in Sections 2, 3, and 4. In each figure, one can see the result best achieved by each method. More than making a direct comparison between the several methods, these figures allows one to verify all capabilities made available to the general practitioner.

7. CONCLUSION

This paper described a general toolbox for signal denoising. The toolbox included the implementation of four denoising techniques, namely: spectral subtraction, Wiener filtering, Wavelet transform, and adaptive filtering. For each advanced algorithm, a friendly and powerful GUI is provided, and all its capabilities described. The presented toolbox makes the application of each of the above methods a simple mouse-oriented operation, easily monitored by a positive feedback through graphics and proper performance measurements. The toolbox can be complemented by, for instance, a independent component analysis module as obtained from [9]. All material here described, including .m and .fig scripts and demo files, can be found in <http://www.lps.ufrj.br/~sergioln/denoising>.

8. REFERENCES

- [1] S. J. Godsill and P. J. W. Rayner, *Digital Audio Restoration*, Springer, 1998.
- [2] S. Haykin, *Modern Filters*, MacMillan, 1989.
- [3] Mathworks site, <http://www.mathworks.com>.
- [4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 2000.
- [5] D. L. Donoho and I. M. Johnstone, “Threshold selection for wavelet shrinkage of noisy data,” *Proceedings of the 16th Annual Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 24a–25a, November 1994.
- [6] P. S. R. Diniz, *Adaptive Filtering*, Kluwer Academic Publishers, 2002.
- [7] S. Werner and P. S. R. Diniz, “Set-membership affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 8, August 2001.
- [8] L. Satish and B. Nazneen, “Wavelet-based denoising of partial discharge signals buried in excessive noise and interference,” *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 10, no. 2, pp. 354–367, April 2003.
- [9] A. Hyvriinen, J. Karhunen, and E. Oja, <http://www.cis.hut.fi/projects/ica/book>.