Signal Denoising Using Wavelets

Project Report

Author:

Rami Cohen (rc@tx.technion.ac.il)

http://tx.technion.ac.il/~rc

Department of Electrical Engineering

Technion, Israel Institute of Technology

Winter 2011/12

February 2012

Abstract

One of the fields where wavelets have been successfully applied is data analysis. Beginning in the 1990s, wavelets have been found to be a powerful tool for removing noise from a variety of signals (denoising). They allow to analyse the noise level separately at each wavelet scale and to adapt the denoising algorithm accordingly.

Wavelet thresholding methods for noise removal, in which the wavelet coefficients are thresholded in order to remove their noisy part, were first introduced by Donoho in 1993. The theoretical justifications and arguments in their favour are highly compelling. These methods do not require any particular assumptions about the nature of the signal, permits discontinuities and spatial variation in the signal, and exploits the spatially adaptive multiresolution of the wavelet transform.

This project report presents and discusses two specific thresholding methods. Their main features and limitations are discussed. Two signals contaminated with additive Gaussian additive noise (AWGN) are used for performance evaluation and simulations results are provided.

Contents

1 Introduction

It is well known to any scientist and engineer who work with a real world data that signals do not exist without noise, which may be negligible (i.e. high SNR) under certain conditions. However, there are many cases in which the noise corrupts the signals in a significant manner, and it must be removed from the data in order to proceed with further data analysis. The process of noise removal is generally referred to as *signal denoising* or simply *denoising*. Example of a noisy signal and its denoised version can be seen in Figure 1. It can be seen that the noise adds high-frequency components to the original signal which is smooth. This is a characteristic effect of noise.

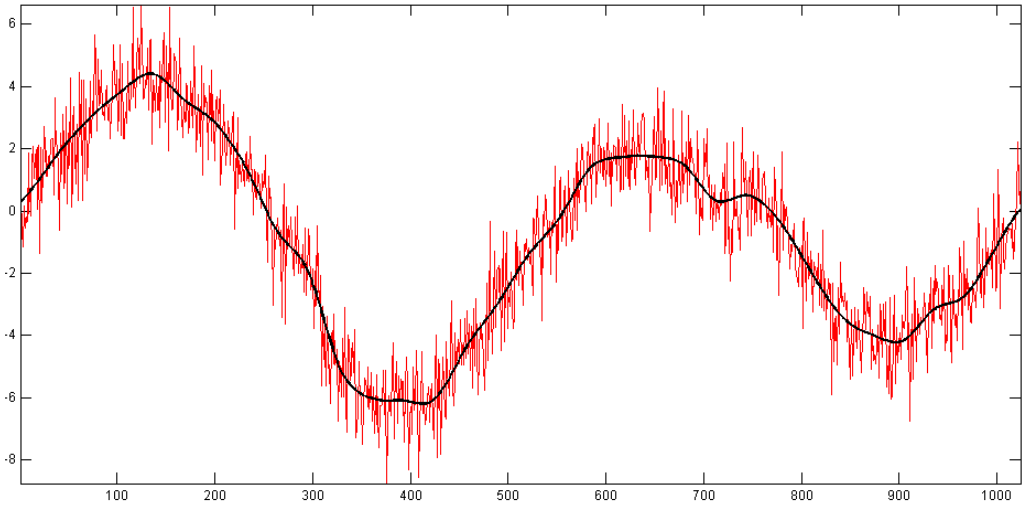


Figure 1: Noisy sine and its denoised version (solid line)

Although the term "signal denoising" is general, it is usually devoted to the recovery of a digital signal that has been contaminated by additive white Gaussian noise (AWGN), rather than other types of noise (e.g., non-additive noise, Poisson/Laplace noise, etc.). This report will concentrate on the case of AWGN.

The optimization criterion according to which the performance of a denoising algorithm is measured is usually taken to be mean squared error (MSE)-based, between the original signal (if exists) and its reconstructed version. This common criterion is used mostly due its computational simplicity. Moreover, it usually leads to expressions which can be dealt with analytically. However, this criterion may be inappropriate for some tasks in which the criterion is perceptual quality driven, though perceptual quality assessment itself is a difficult problem, especially in the absence of the original signal.

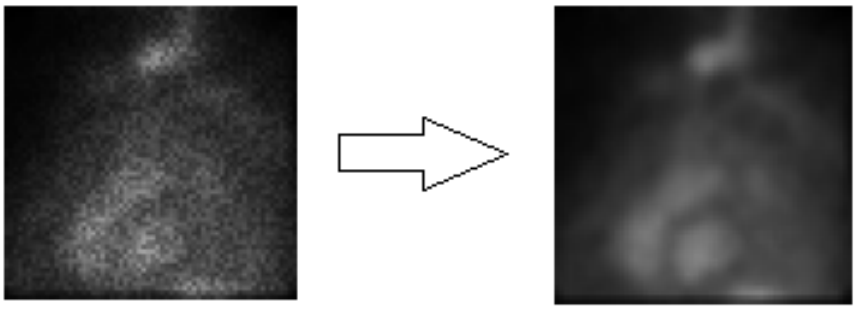


Figure 2: Typical result of LPF as a technique for denoising

There is a wide range of applications in which denoising is important. Examples are medical image/signal analysis, data mining, radio astronomy and there are many more. Each application has its special requirements. For example, noise removal in medical signals requires specific care, since denoising which involves smoothing of the noisy signal (e.g., using low-pass filter) may cause the lose of fine details, as can be seen in Figure 2.

There are many approaches in the literature for the task of denoising, which can be roughly divided into two categories: denoising in the original signal domain (e.g., time or space) and denoising in the transform domain (e.g., Fourier or wavelet transform).

The development of wavelet transforms over the last two decades revolutionized modern signal and image processing, especially in the field of signal denoising. During the 1990s, the field was dominated by *wavelet shrinkage* and *wavelet thresholding* methods (to be introduced later), including several papers by Donoho et. al [1–5]. Review of most of these methods is given in [6].

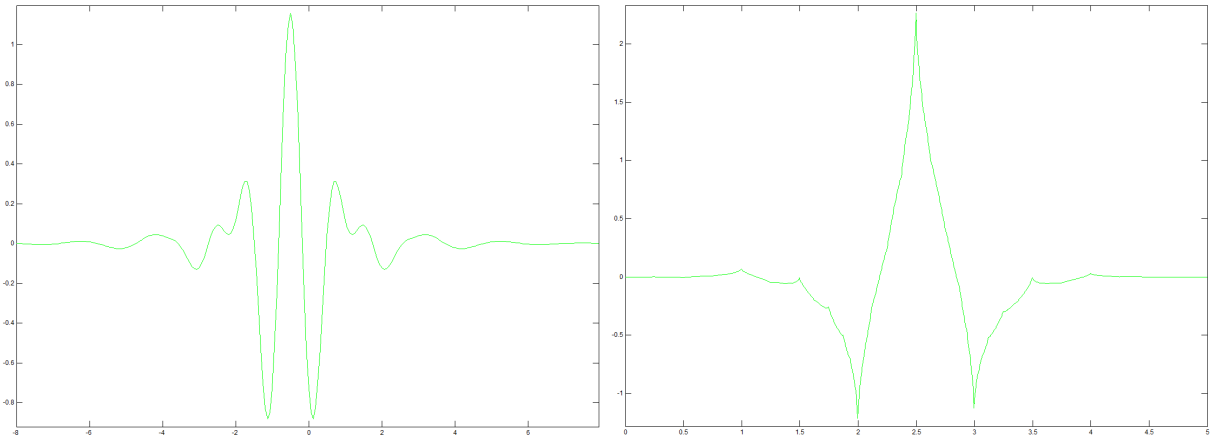
In this report two deoising methods are reviewed and implemented. The first one is *SureShrink*, introduced by Donoho and Johnstone in [3]. The second one was introduced by Cai and Silverman in [7], and is called *NeighBlock*.

This report is organized as follows. First, wavelet transform and Mallat's algorithm are reviewed briefly in Section 2. Later, the denoising problem is presented in Section 3 and two denoising algorithms are described in Section 4. Finally, discussion of the algorithms is given in Section 5, and conclusions are given in Section 6.

2 Wavelet transform

In this section, the wavelet transform and its implementation for discrete signals are reviewed briefly. This review is not intended by any means to be rigorous, and its sole purpose is to describe the tools which will be used later. For a more detailed discussion of wavelet transform, the reader is referred (for example) to [8–10].

A wavelet is a wave-like oscillation with an amplitude that starts out at zero, increases, and then decreases back to zero. Unlike the sines used in Fourier transform for decomposition of a signal, wavelets are generally much more concentrated in time. They usually provide an analysis of the signal which is localized in both time and frequency, whereas Fourier transform is localized only in frequency. Examples for wavelets are given in Figure 3.



(a) Meyer wavelet (b) Coiflet wavelet

Figure 3: Wavelets - examples

Given a *mother wavelet* $\psi(t)$ (which can be considered simply as a basis function of L^2), the continuous wavelet transform (CWT) of a function $x(t)$ (assuming that $x \in L^2$) is defined as:

$$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t)\, dt \tag{2.1}$$

The *scale* or *dilation* parameter a corresponds to frequency information ($a \sim \text{frequency}^{-1}$), and the *translation* parameter b relates to the location of the wavelet function as it is shifted through the signal, so it corresponds to the time information in the transform. The integral in (2.1) can be seen as a convolution operation of the signal and a basis function $\psi(t)$ (up to dilations and translations). It should be emphasized that, unlike Fourier transform in which

the basis function is $e^{i\omega t}$, there are many (in fact, infinite) possible choices of $\psi(t)$.

In practice, the transform which is used is the discrete wavelet transform (DWT) which transforms discrete (digital) signals to discrete coefficients in the wavelet domain. This transform is essentially a sampled version of CWT. Instead of working with $a, b \in \mathbb{R}$, the values of $X(a, b)$ are calculated over a discrete grid:

$$a = 2^{-j}, b = k \cdot 2^{-j}, \qquad j, k \in \mathbb{Z} \tag{2.2}$$

where this type of discretization is called *dyadic dilation* and *dyadic position*, respectively.

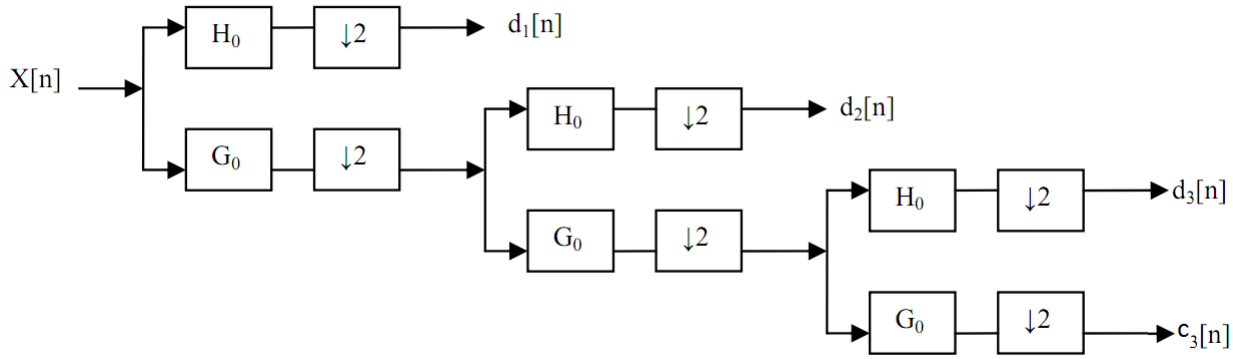
2.1 Mallat's algorithm

In the case of DWT, assuming that the length of the signal satisfies $N = 2^J$ for some positive J, the transform can be computed efficiently, using Mallat's algorithm [11], which has a complexity of $O(N)$. Essentially the algorithm is a fast hierarchical scheme for deriving the required inner products (which appear in (2.1), as a function of a and b) using a set of consecutive low and high pass filters, followed by a decimation. This results in a decomposition of the signal into different *scales* which can be considered as different frequency bands.

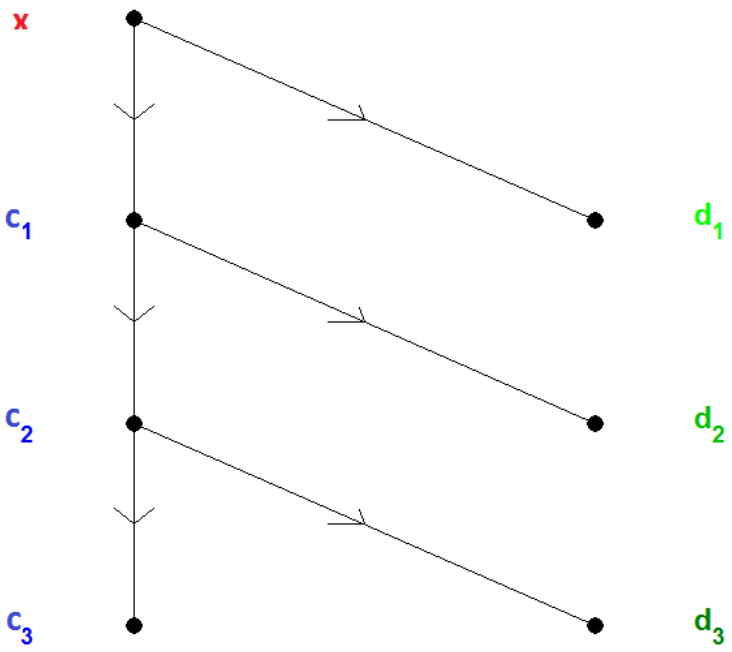
The low-pass (LP) and high-pass (HP) filters used in this algorithm are determined according to the mother wavelet in use. The outputs of the LP filters are referred to as *approximation coefficients* and the outputs of the HP filters are referred to as *detail coefficients*. Demonstration of the process of 3-level decomposition of a signal can be seen in Figure 4.

At each decomposition level, the filters produce signals spanning only half the frequency band. This doubles the frequency resolution as the uncertainty in frequency is reduced by half. Thus, the output of each filter can be down-sampled (decimated) by a factor of 2 in accordance with Nyquist's theorem. Using this approach, the time resolution becomes arbitrarily good at high frequencies, while the frequency resolution becomes arbitrarily good at low frequencies. The number of decomposition levels should be determined by the user (according to the characters of the signal to be analysed). Typical number is $3 - 5$.

The DWT of the original signal is obtained by concatenating all the coefficients starting from the last level of decomposition. From now on, we will adopt the notation c_{jk} for the k^{th} value of the approximation coefficients of level j, where d_{jk} will have similar meaning for the detail coefficients of level j. Example of the coefficients produced by Mallat's algorithm is given in Figure 5. As can be easily seen in this figure, the detail coefficients tend to have high value in the noisy parts of the signal.



(a) Mallat's algorithm demonstration - 3-level decomposition of a signal. H_0 is an HPF and G_0 is an LPF. $c_j[n]$ denote the approximation coefficients and $d_j[n]$ denote the detail coefficients (adapted from [12])



(b) Visualization of a three-level decomposition of a signal as a wavelet tree

Figure 4: Mallat's algorithm

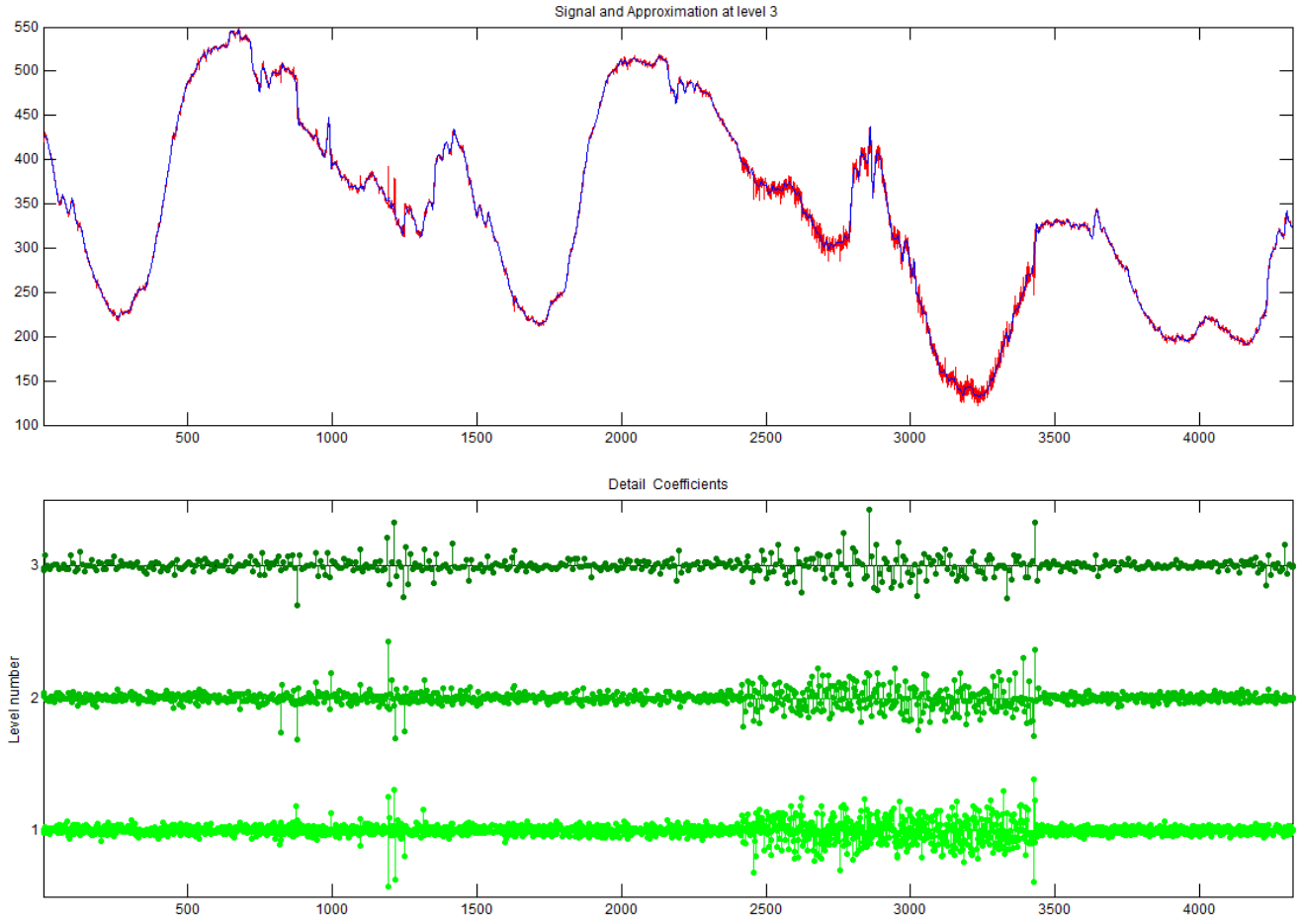


Figure 5: Mallat's algorithm example (the mother wavelet is Symlet). The signal $X[n]$ and its approximation (smooth curve) $c_3[n]$) are shown in the upper figure, and the three detail coefficients are depicted in the lower one

3 Problem formulation

A common posing of the denoising problem is as follows. Suppose that there are n noisy samples of a function f:

$$y_i = f(t_i) + \sigma\varepsilon_i, \quad i = 1...n \tag{3.1}$$

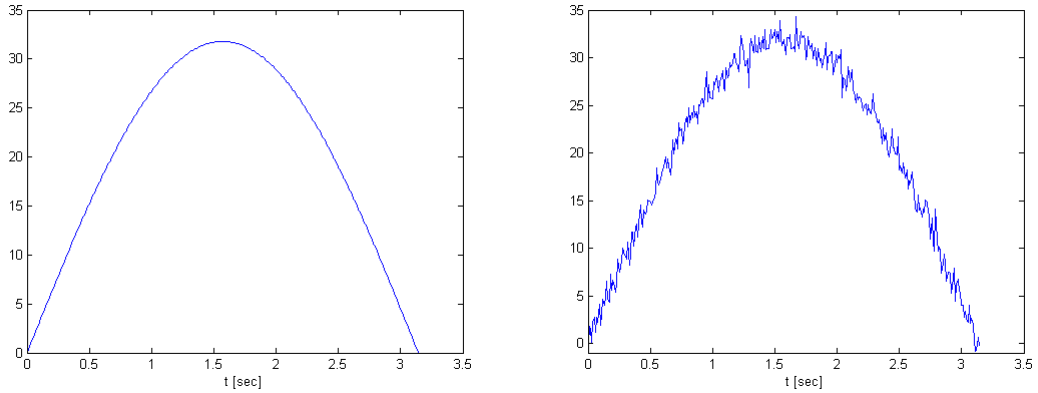
where ε_i are iid $N(0,1)$ and the noise level σ may be known or unknown. Example can be seen in Figure 6. The goal is to recover the underlying function f from the noisy data, $\mathbf{y} = (y_1, ..., y_n)$, with small error, when the criterion is the mean squared error. In other words, it is needed to find a function $\hat{f}$ which satisfies:

$$\hat{f} = \min_{\hat{f}} \left\| \hat{f} - f \right\|_2 \tag{3.2}$$

where $\hat{f} = \hat{f}(\mathbf{y})$. It should be clear that in practice the function f is unknown, so the MSE is usually estimated.

Note that relation (3.1) is not genetal, since the noise may be non-additive and the relation between the observed signal and the original signal may be stochastic. Nevertheless, (3.1) is a good model for many practical situations.

It is assumed in what follows, without loss of generality, that t_i are within the unit interval $[0, 1]$. Furthermore, for simplicity, it is assumed that these sample points are equally spaced and that $n = 2^J$ for some $J \in \mathbb{N}$. These assumptions allow to perform both the DWT and the IWDT using Mallat's fast algorithm. In the following section, two methods for the (approximate) solution of (3.2) are discussed.



(a) Clean signal f (b) y, which is f with AWGN (SNR=20dB)

Figure 6: Signal f and its noisy version y

4 Wavelet thresholding

The orthogonality of DWT (assuming that orthogonal wavelets are used with periodic boundary conditions) leads to the feature that white noise is transformed into white noise. Hence, if y_{jk} (where j denotes the decomposition level and k is the index of the coefficient in this level) are the wavelet coefficients of y_i, the transform of (3.1) to the wavelet domain is:

$$y_{jk} = w_{jk} + \sigma\tilde{\varepsilon}_{jk} \tag{4.1}$$

where w_{jk} are the (clean) wavelet coefficients of $(f(t_i))$, consisting of the approximation and detail coefficients (see 2.1) and $\tilde{\varepsilon}_{jk}$ are $iid\ N(0,1)$. That is, the wavelet coefficients of the observed signal can themselves be considered as a noisy version of the wavelet coefficients of the original signal. In a first view, (4.1) looks like no improvement over (3.1), but it is not true, since the analysis of the signal in the wavelet domain has some advantages.

The coefficients of the wavelet transform are usually sparse. That is, most of the coefficients in a noiseless wavelet transform are effectively zero (as can be seen, for example, in Figure 5). Therefore, we may reformulate the problem of recovering f as one of recovering the coefficients of f which are relatively "stronger" than the Gaussian white noise background. That is, coefficients with small magnitude can be considered as pure noise and should be set to zero. The approach in which each coefficient is compared with a threshold in order to decide whether it constitute a desirable part of the original signal or not, is called *wavelet thresholding*.

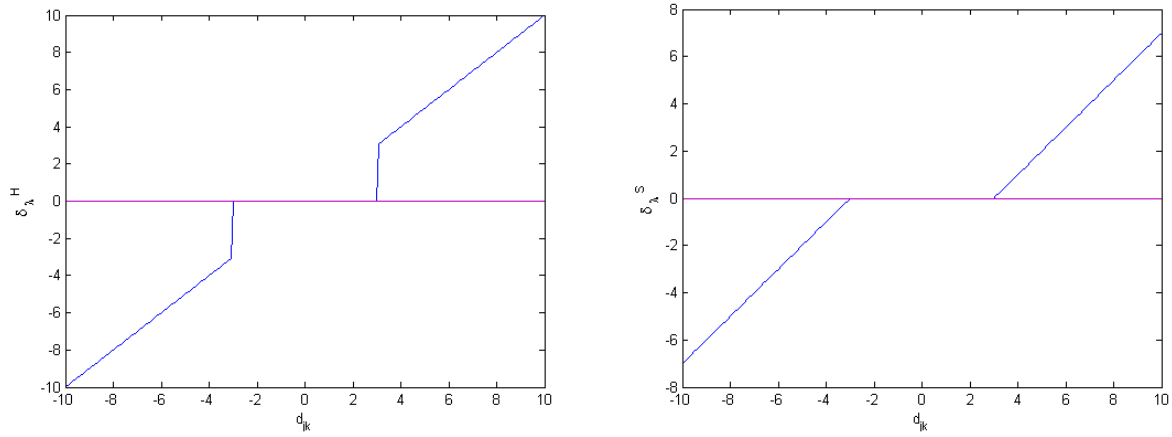
The thresholding of the wavelet coefficients is usually applied only to the detail coefficients d_{jk} of $\mathbf{y}$ rather than to the approximation coefficients c_{jk}, since the latter ones represent 'low-frequency' terms that usually contain important components of the signal, and are less affected by the noise. The thresholding extracts the significant coefficients by setting to zero the coefficients which their absolute value is below a certain threshold level, which is to be denoted by λ. This value can generally be a function of the decomposition/resolution level j and the index k, i.e.: $\lambda = \lambda(j,k)$, but usually it is function of j only: $\lambda = \lambda(j)$. In the latter case the threshold is called *level-dependent threshold*.

The thresholded wavelet coefficients are obtained using either *hard* or *soft* thresholding rule given respectively by:

$$\delta_\lambda^H(d_{jk}) = \begin{cases} 0, & \text{if } |d_{jk}| \le \lambda \\ d_{jk}, & \text{if } |d_{jk}| > \lambda \end{cases}$$

$$\delta_\lambda^S(d_{jk}) = \begin{cases} 0, & \text{if } |d_{jk}| \leq \lambda \\ d_{jk} - \lambda, & \text{if } d_{jk} > \lambda \\ d_{jk} + \lambda, & \text{if } d_{jk} < -\lambda \end{cases}$$

where, as noted before, λ can generally be a function of j and k. The hard thresholding rule is usually referred to simply as *wavelet thresholding*, whereas the soft thresholding rule is usually referred to as *wavelet shrinkage*, since it "shrinks" the coefficients with high amplitude towards zero. The thresholding rules are depicted in Figure 7.



(a) Hard-thresholding (b) Soft-thresholding

Figure 7: Thresholding, $\lambda = 3$

Most of the algorithms using the thresholding approach try to estimate the optimal value of λ. However, the first step in these algorithms usually involves the estimation of the noise level σ. Assuming simply that σ is proportional to the standard deviation of the coefficients is clearly not a good estimator, unless f is reasonably flat. A popular estimate of the noise level σ was proposed by Donoho and Johnstone [2]. This is based on the last level of the detail coefficients, according to the *median absolute deviation*:

$$\hat{\sigma} = \frac{\text{median}\left(\{|d_{J-1}, k|\} : k = 0, 1, ..., 2^{J-1} - 1\right)}{0.6745} \tag{4.2}$$

where the factor in the denominator is the *scale factor* which depends on the distribution of d_{jk}, and is equal to 0.6745 for a normally distributed data. In this report it will be assumed that σ is known, and for convenience it is taken to be 1.

In the following pages, two methods which take the approach of wavelet thresholding are described and discussed. These methods assume no particular parametric structure of f, and they are appropriate for estimating relatively regular functions.

4.1 SureShrink

Donoho and Johnstone [3] proposed a scheme which fits a level-dependent threshold λ_j to the wavelet coefficients y_{jk} from (4.1). Their scheme is based on *Stein's unbiased risk estimate* (SURE) [13], which is explained as follows.

Assume that $X_1, ..., X_s$ are *iid* $N(\mu_i, 1)$ $(i = 1, ..., s)$. The problem is to estimate the mean vector $\mu = (\mu_1, ..., \mu_s)$ from $\mathbf{x} = (X_1, ..., X_s)$ with minimum l_2 *risk*, that is, to find an estimator $\hat{\mu}$ which satisfies:

$$\hat{\mu} = \min_{\hat{\mu}} \|\mu - \hat{\mu}\|_2 \tag{4.3}$$

The exact risk in (4.3) is in practice unknown, since the original μ is in unknown. In order to get an estimate of the risk without the need to know μ, Stein showed that for any estimator of μ which can be written as $\hat{\mu}(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$ where $g : \mathbb{R}^s \to \mathbb{R}^s$ and g is weakly differentiable, the l_2 risk can be estimated unbiasedly:

$$\text{SURE}(\hat{\mu}) = \mathrm{E}_\mu \|\mu - \hat{\mu}\|_2^2 = s + \mathrm{E}_\mu \left\{ \|g(\mathbf{x})\|_2^2 + 2\nabla \cdot g(\mathbf{x}) \right\} \tag{4.4}$$

where $\nabla \cdot g(\mathbf{x})$ is the divergence of g:

$$\nabla \cdot g(\mathbf{x}) = \sum_{i=1}^{s} \frac{\partial g_i}{\partial x_i}$$

Consider w_{jk} from (4.1) as $\mathbf{x}$. Using the expression of the soft thresholding rule δ_λ^S, it can be seen that $\delta_\lambda^S(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$, where:

$$\|g(\mathbf{x})\|_2^2 = \sum_{i=1}^{s} \left[\min\left(|X_i|, \lambda\right)\right]^2, \quad \nabla \cdot g(\mathbf{x}) = -\sum_{i=1}^{s} 1_{[-\lambda,\lambda]}(X_i) \tag{4.5}$$

($1_A(x)$ is the indicator function for set A). Now it is possible to get an explicit expression for SURE:

$$\text{SURE}(\lambda; \mathbf{x}) = s + \sum_{i=1}^{s} \left[\min\left(|X_i|, \lambda\right)\right]^2 - 2 \cdot \#\left(i : |X_i| < \lambda\right) \tag{4.6}$$

(# denotes cardinality). The obvious advantage of the expression in (4.6) is that the unknown μ does not appear. It is clear that the threshold λ which appears in (4.6) should be chosen so SURE is minimized:

$$\lambda^* = \text{argmin}_{0 \le \lambda \le \sqrt{2 \log s}} \text{SURE}(\lambda; \mathbf{x}) \tag{4.7}$$

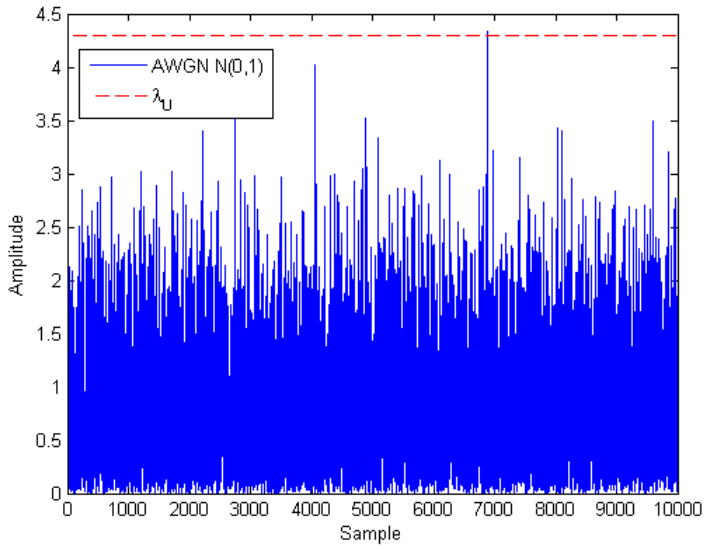


Figure 8: $s = 10,000$ *iid* normally distributed samples and the universal bound $\lambda_U = \sqrt{2\log s} = 4.29$

Before proceeding, it should be noted that the search for the optimal λ in (4.7) does not take into account values which are greater than $\lambda_U = \sqrt{2\log s}$ (known as the *universal bound*). The reason is as follows. According to theorem proved in [2], if $X_i, ..., X_s$ are *iid* $N(0,1)$, then:

$$\Pr\left\{\max_{1\leq i\leq s}|X_i| > \sqrt{2\log s}\right\} \sim \frac{1}{\sqrt{\pi\log s}} \underset{s\to\infty}{\to} 0 \tag{4.8}$$

This is depicted in Figure 8.

The upper bound on $|X_i|$ from (4.8) can serve as a universal threshold, since coefficients with magnitude less than λ_U are, with high probability, noise. As noted before, the wavelet transform of many noiseless objects is filled with many zero coefficients (i.e., *sparsity* of the wavelet transform). After contamination with noise, these coefficients become non-zero. If these noisy coefficients are not set to zero, the reconstruction will have an annoying visual appearance. Hence, limiting λ by $\sqrt{2\log s}$, ensures, with high probability, that every sample in the wavelet transform in which the underlying sample is zero will be estimated as zero. Therefore, λ values which are greater than λ_u should not be taken into account in the optimization problem of (4.7).

Setting λ (for all levels) simply to the universal bound $\lambda_U = \sqrt{2\log s}$ provides good results with virtually no computational complexity. This method was proposed by Donoho

and Johnstone in [2] and is known as *VisuShrink*. However, this method is usually too general and is not adapted to the data, and as noted in [3], λ^* from (4.7) is more adaptive to unknown smoothness than λ_U.

The optimization problem in (4.7) can be solved in a straight forward manner. If X_i are ordered in order of increasing $|X_i|$ (this can be done in $O(s \log s)$, using merge sort, for example), it is clear from (4.6) that on intervals of λ that lie between two values of $|X_i|$, SURE is strictly increasing. Therefore, λ^* must be equal to one of the values $|X_i|$, which correspond to the absolute values of y_{jk} (for some j). Hence, SURE should be calculated for no more than s coefficients, resulting in an additional complexity of $O(s)$. The resulting total complexity is $O(s \log s)$.

Example for the solution of (4.7) is given in Figure 9. In this example, μ is of dimension $s = 128$, consists of 16 consecutive 4's followed by all zeros, where AWGN of variance 1 was added. Using the SURE profile, and looking for minimum point, a threshold is obtained. It can be seen that SURE follows the actual loss quite closely. The estimate of μ is given in Figure 14b, where the shrinkage effect is quite noticeable.

However, as noted in [3], selecting λ^* simply according to (4.7) is too naive. This is because of the common sparse nature of the wavelet coefficients. In the case of significant sparsity, the noise contributed to the SURE profile by the many coordinates at which the signal is zero is much more significant than the information contributed to the SURE profile by the few non-zero coordinates. Hence, *hybrid* scheme is proposed, which is called *SureShrink*.

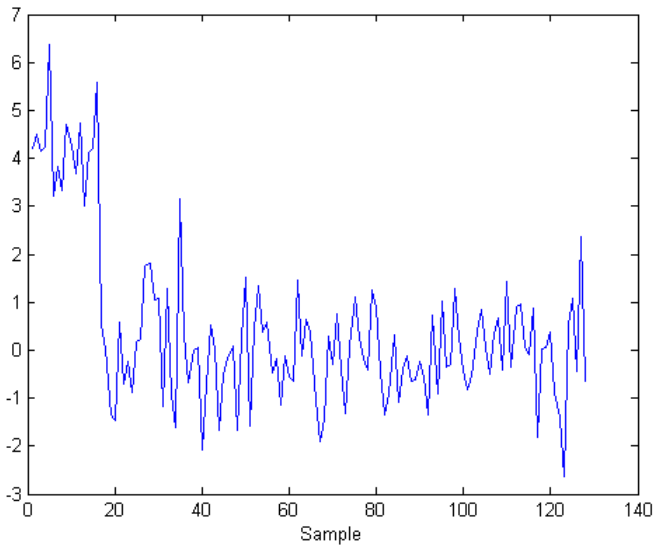
According to this scheme, the threshold value in the sparse case should be set to λ_U (universal bound) so with high probability a pure noise sample is correctly thresholded to zero. This is desirable, since there are many such samples in the sparse case. If the wavelet coefficients are not sparsely represented, SURE is supposed to provide good estimation of the loss, and λ^* is used.

In order to decide whether to use λ^* or λ_U, a measure of the sparsity of the coefficients is needed. In [3], the following measure is suggested:

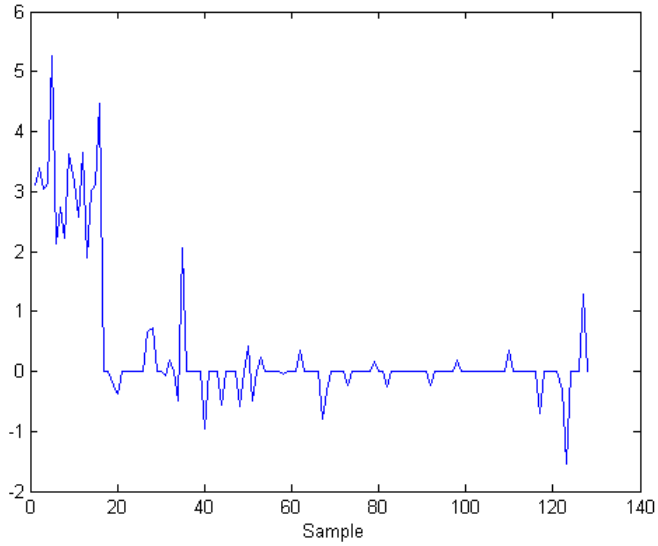
$$\nu_s(\mathbf{x}) = \frac{s^{-1} \cdot \sum_{i=1}^{s} (x_i^2 - 1)}{s^{-1/2} \cdot \log_2^{3/2}(s)} = s^{-1/2} \cdot \frac{\sum_{i=1}^{s} (x_i^2 - 1)}{\log_2^{3/2}(s)} \tag{4.9}$$

where $\mathbf{x}$ is considered sparse if $\nu_s(\mathbf{x}) \leq 1$ and non-sparse otherwise. Finally, thresholding of (noisy) data $\mathbf{x} \in \mathbb{R}^s$ using SureShrink is decided as follows:

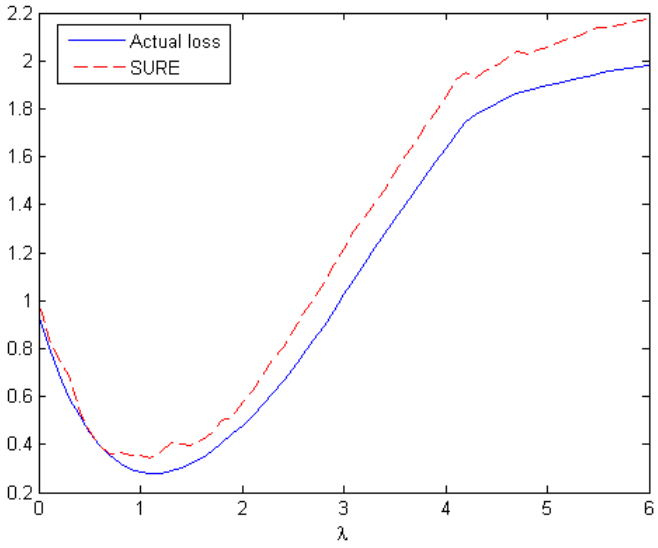
$$\lambda_{\text{SureShrink}}(\mathbf{x}) = \begin{cases} \sqrt{2 \log s}, & \text{if } \nu_s(\mathbf{x}) \leq 1 \\ \lambda^*, & \text{otherwise} \end{cases} \tag{4.10}$$



(a) μ, noisy data



(b) $\hat{\mu}$ (estimate of μ) selected according to SURE



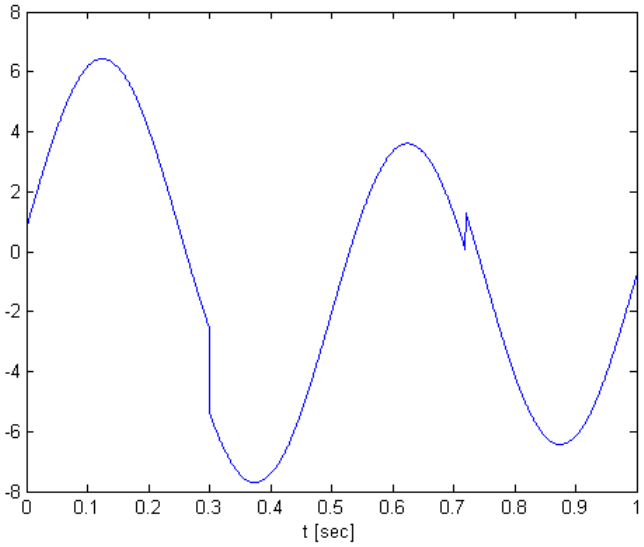
(c) SURE versus actual loss

Figure 9: Choice of λ using SURE

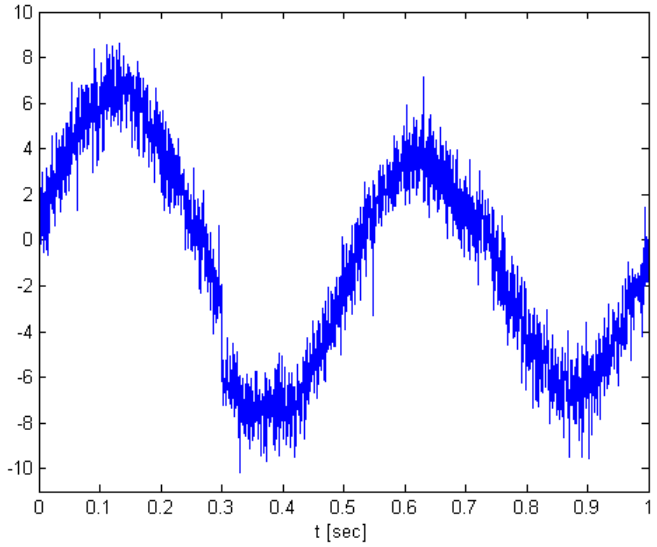
In our case, each $\mathbf{x}$ is actually one of the detail coefficients $\mathbf{d}_j$, where j corresponds to the decomposition level. That is, λ is level-dependent: $\lambda = \lambda(j)$.

The signals (Doppler and HeaviSine) which were used for simulation purposes, with their noisy versions, are depicted in Figure 10. The noise used in the simulations is AWGN with variance 1. The signals were chosen in accordance with [3] in order to represent spatially nonhomogeneous phenomena. The wavelet which will be used is Daubechies with $N = 8$, and $j_{max} = 5$ decomposition levels will be used. The results of SureShrink are depicted in Figures 11 and 12.

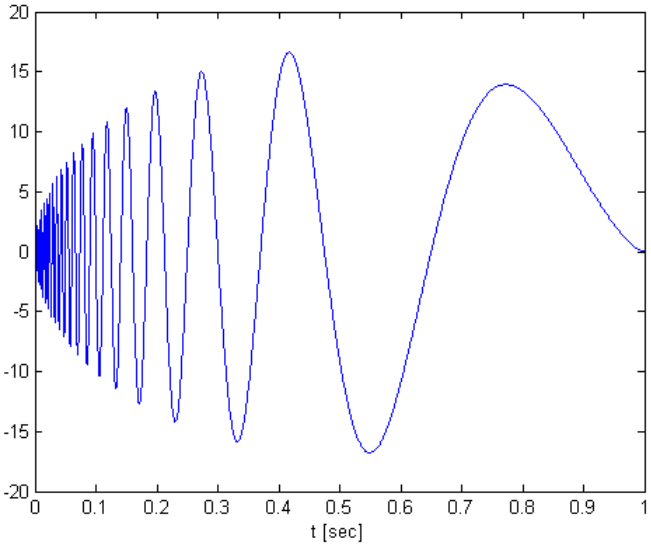
As can be seen in the results, SureShrink performs well in removing the noise from both signals. The reconstructed Doppler signal is very similar to the original one. However, SureShrink failed to reconstruct the discontinuities of HeaviSine, since they are masked by the noise, but there are some indications to the existence of these continuities in the reconstructed signal.



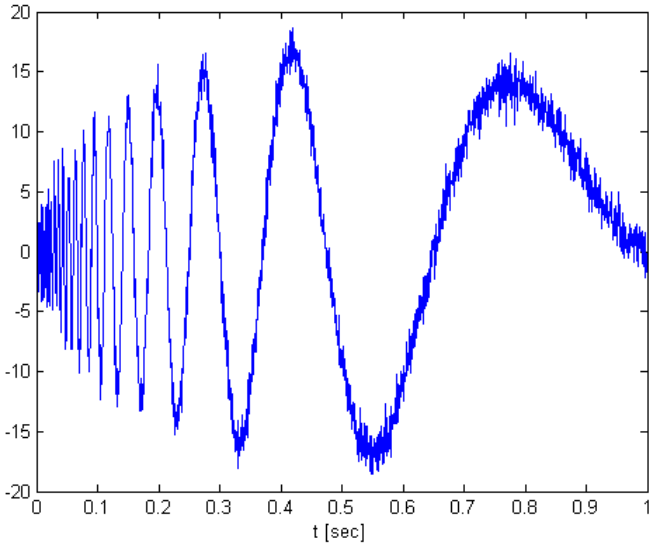
(a) HeaviSine



(b) HeaviSine with noise $N(0,1)$



(c) Doppler



(d) Doppler with noise $N(0,1)$

Figure 10: The signals used in this report

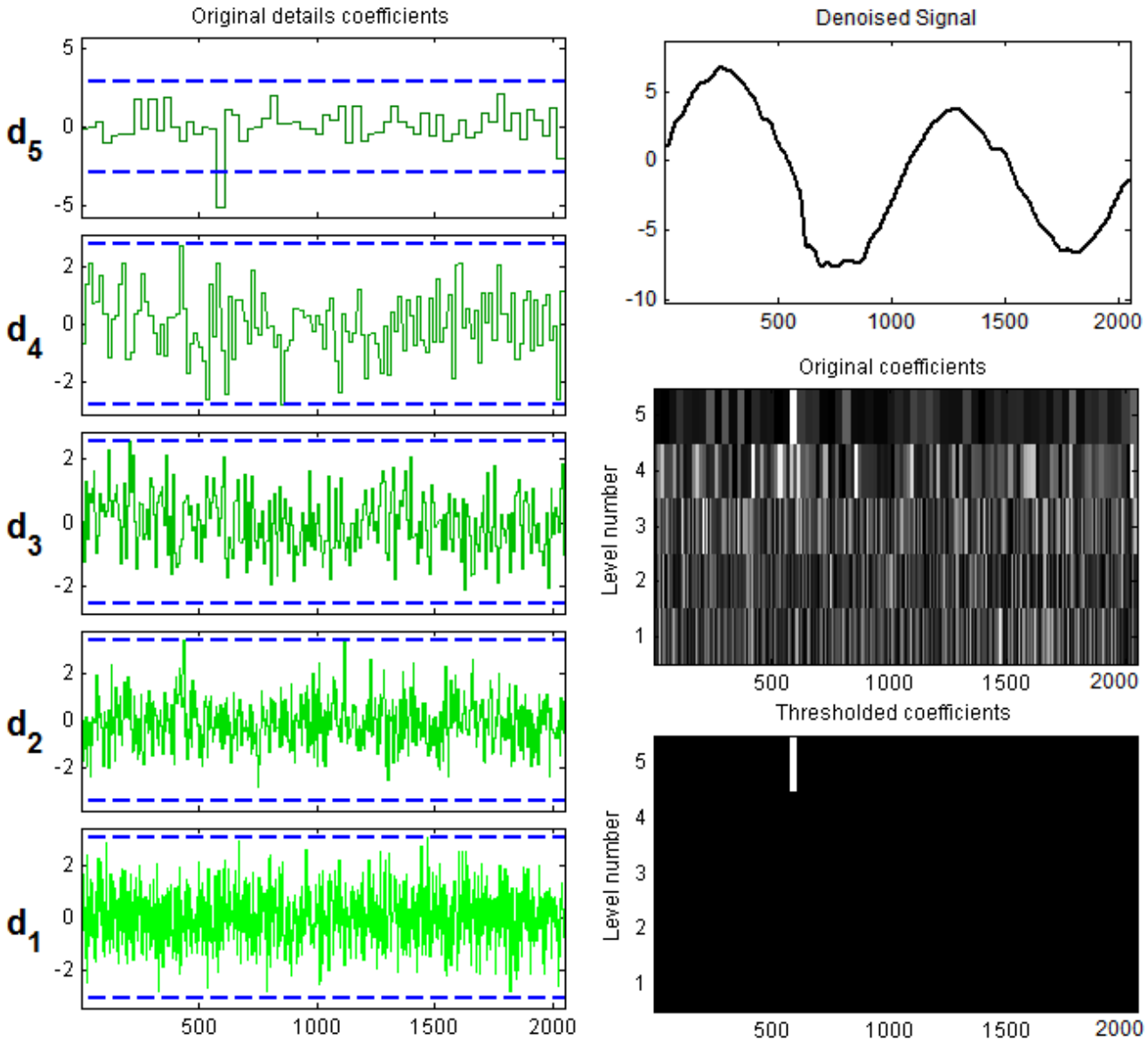


Figure 11: Denoising of HeaviSine signal using SureShrink. Scaleograms of the detail coefficient of HeaviSine signal before and after denoising appear in the right (brighter - stronger amplitude). The dashed lines in the left side of the figure denote the thresholds λ_j of the detail coefficients $\mathbf{d}_j$.

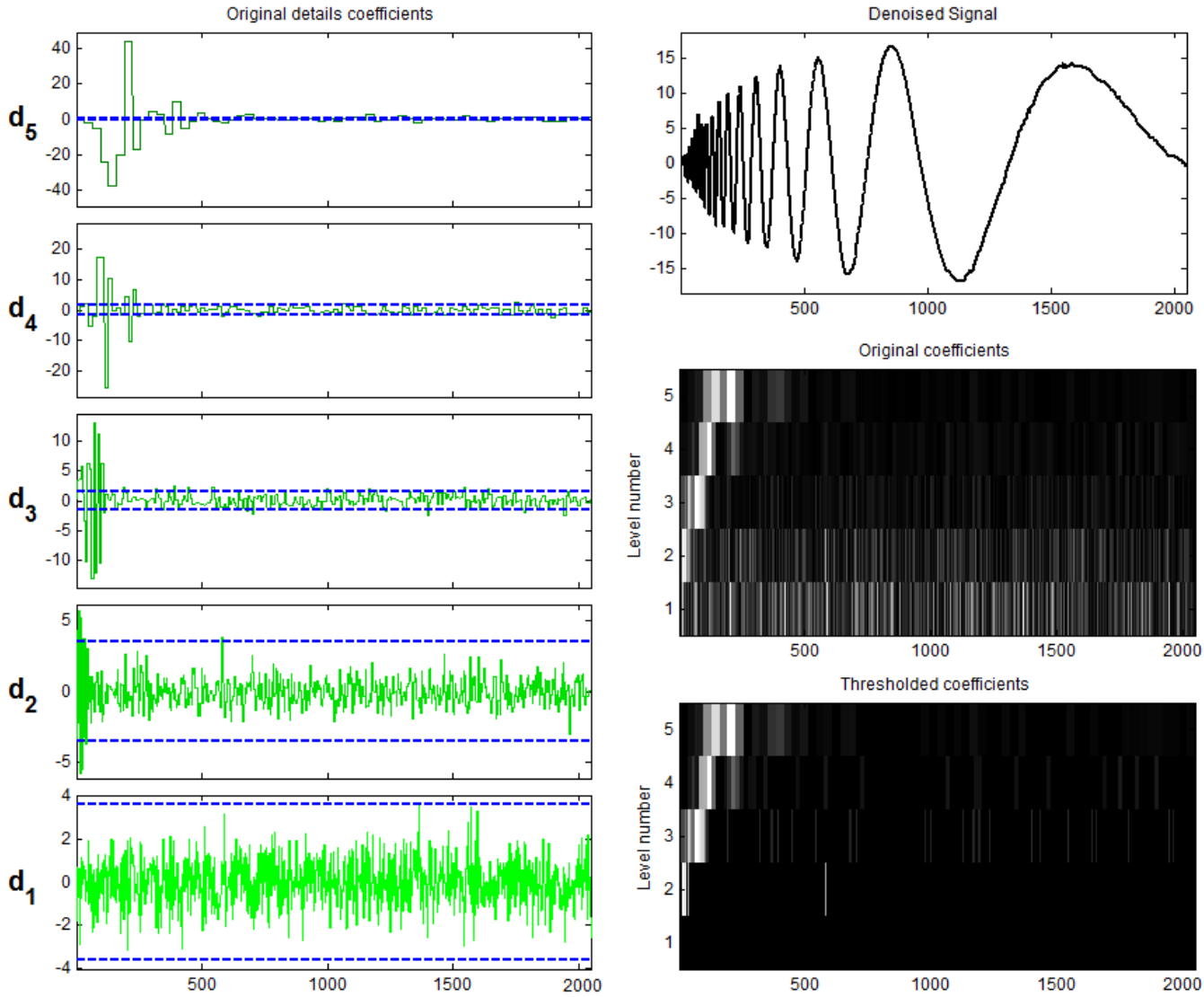


Figure 12: Denoising of Doppler signal using SureShrink. Scaleograms of the detail coefficient of Doppler signal before and after denoising appear in the right (brighter - stronger amplitude). The dashed lines in the left side of the figure denote the thresholds λ_j of the detail coefficients $\mathbf{d}_j$.

4.2 NeighBlock

SureShrink thresholding procedure described previously achieves adaptivity through sparsity considerations. However, it does not take into account the local neighbourhood of each coefficient, and it is decided per resolution level. Therefore, it may remove too many terms from the wavelet coefficients (i.e., resulting in a biased estimator).

A possible way to increase estimation precision is by utilising information about neighbouring wavelet coefficients. For example, the threshold level can be chosen per group (blocks) of coefficients, according to their local properties, where each group contains a subset of the coefficients in the current resolution level.

The basic motivation of working with blocks is simple: if neighbouring coefficients contain important part of the signal, then it is likely that the current coefficient is also important and so a lower threshold should be used. This yields a *local* trade-off between signal and noise.

Cai and Silverman [7] proposed a method called *NeighBlock*, which incorporates information on neighbouring coefficients into the decision making. This method is consisted of the following steps. At each resolution level j, the wavelet coefficients are grouped into disjoint blocks (jb) of length $L_0 = \left[\frac{\log n}{2}\right]$. Then, each block is extended by an amount of $L_1 = \max(1, [L_0/2])$ in each direction, forming overlapping blocks (jB) of length $L = L_0 + 2L_1$. Now, the coefficients in each disjoint block jb are estimated according to:

$$\hat{d}_{jk}^{(jb)} = \max\left(0, \frac{S_{(jB)}^2 - \lambda_C L}{S_{(jB)}^2}\right) d_{jk}^{(jb)} \tag{4.11}$$

where S_i^2 denotes the l_2 energy of the i^{th} block (sum of squared coefficients), and λ_C is chosen as the solution for the equation $\lambda_C - \log \lambda_C = 3$, which is $\lambda_C = 4.505$. The choice of this value is analogous to the choice of the universal bound (see page 12), and is explained as follows.

Let $X_i, ..., X_s$ be iid $N(0,1)$ and $L = \log s$. Divide X_i into blocks of length L. Then, the maximal l_2 energy of the blocks satisfies [14]:

$$\Pr\left\{\max_b S_b^2 > \lambda_C L\right\} \underset{s\to\infty}{\to} 0 \tag{4.12}$$

where the value $\lambda_C = 4.505$ is the smallest constant satisfying (4.12). Hence, with this choice of λ_C, pure noise should be removed completely, with high probability. It should be noted that the upper bound $\lambda_C L$ on the energy of the blocks is somewhat loose, as demonstrated in Figure 13.

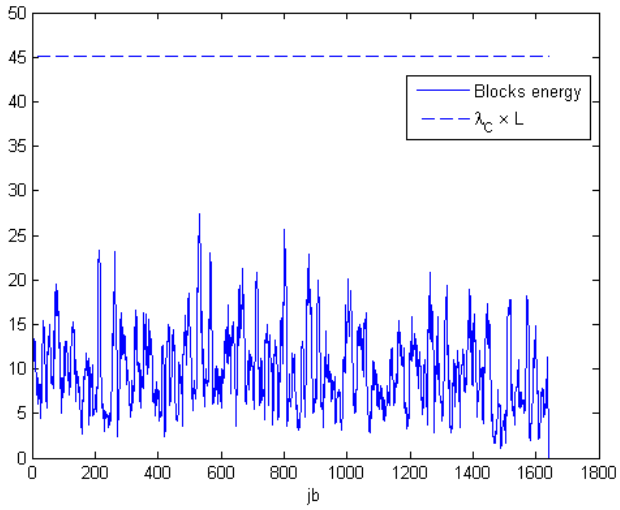
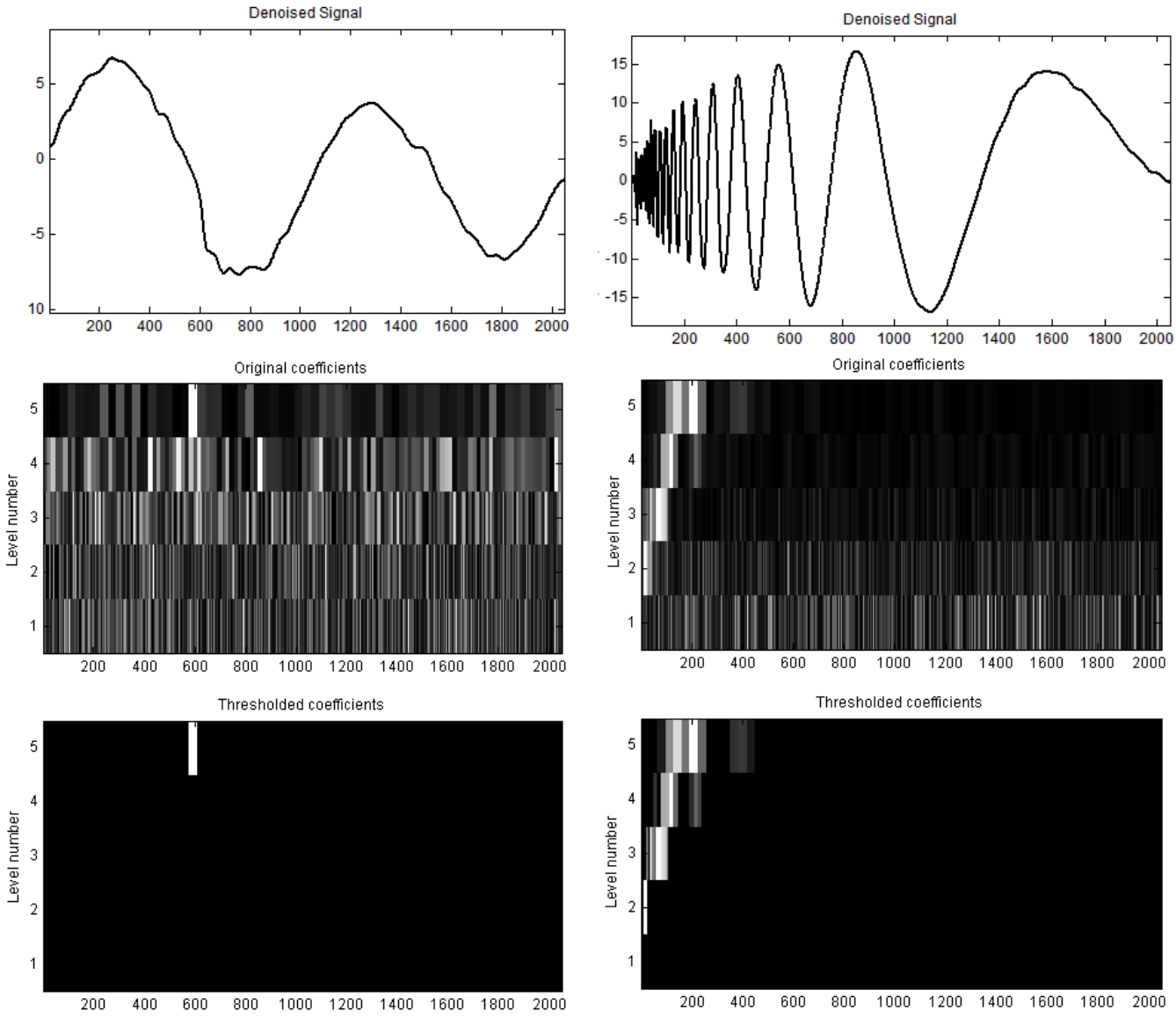


Figure 13: $s = 2^{14}$ iid samples from standard normal distribution, divided into blocks of size $\mathrm{L} = [\log s] = 10$

The results of this method can be seen in Figure 14. Comparing with SureShrink, NeighBlock appears to perform the denoising task slightly better in the case of the Doppler signal. In the case of HeaviSine, SureShrink seems to provide better results.

It should be noted that Cai and Silverman also suggest (in [7]) another method which takes into account a smaller local neighbourhood of each coefficient. This method is called *NeighCoeff*, and it follows the same steps as NeighBlock, but with $L_0 = L_1 = 1, L = 3$ and $\lambda = \frac{2}{3} \log n$. This value of λ is smaller than λ_U. The difference comparing to NeighBlock is that each individual coefficient is shrunk by an amount that depends on the coefficient and on its immediate neighbours, while NeighBlock uses neighbouring coefficients outside the block of current interest. Simulation results in [6, 7] show that these methods have similar performance.



(a) HeaviSine (b) Doppler

Figure 14: Denoising using NeighBlock

5 Discussion

5.1 Performance

The results of NeighBlock are somewhat better than those of SureShrink in the case of the Doppler signal, whereas SureShrink gives slightly better results in the case of HeaviSine signal. This can be explained by the significant sparsity of the detail coefficients (from level 3) in the case of Doppler, as can be seen in Figure 12. In this case, SureShrink uses the universal bound λ_U which is less adapted to the data than the block dependent threshold approach of NeighBlock, which will zero most of the coefficients.

In the case of HeaviSine signal (Figure 11), the details coefficients are not sparsely represented, so SureShrink uses λ^* which better fits the data than λ_U. The magnitude of the coefficients in each level does not change too fast, so the adaptive approach of NeighBlock is less effective in this case.

5.2 Adaptivity

It is clear from the previous description of the algorithms that NeighBlock is, in principle, more adapted to the data than SureShrink, since it takes into account the neighbourhood of each coefficient (*intra* correlation), and the threshold can be different for each group of coefficients. This is not the case in SureShrink, in which the threshold is constant for a specific level. However, the adaptivity feature of NeighBlock suffers from few drawbacks.

First, the block size is constant, once the signal length n is given. In case that the significant coefficients are distributed in smaller or larger blocks, the obtained threshold may be inaccurate. Moreover, it might be a better idea to assign different block sizes to different parts of each coefficients level, especially when the noise magnitude varies over the signal. Example for signal with interval-dependent noise can be seen in Figure 15.

Although SureShrink is not adaptive to the data as NeighBlock is, it tries to determine in advance whether the coefficients are sparsely represented, and if it is indeed the case, the universal bound is used instead of λ^*. However, using λ_U as a threshold in the case of sparse representation does not take into account the actual values of the coefficients, which are quite important.

In both methods, the correlation between different resolution levels (*inter* correlation) is not taken into account. However, it makes sense that indication of noise in one level can serve as a prediction for the next one. Approaches which take this correlation into account can be found, for example, in [15, 16].

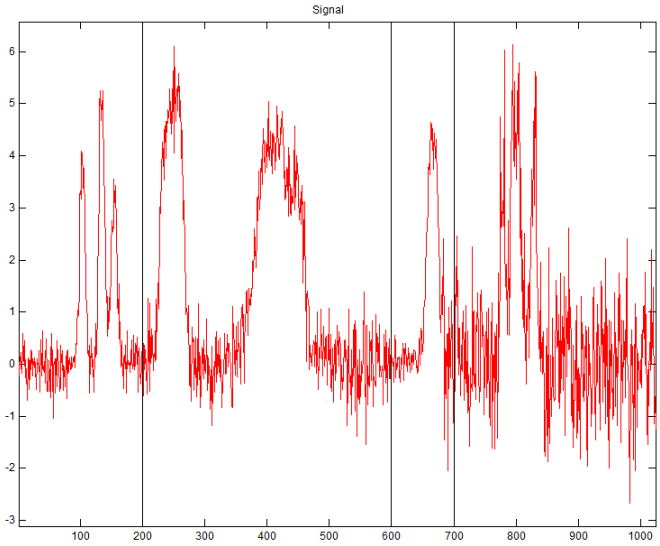


Figure 15: Different noise levels in each interval, delimited by vertical lines

5.3 Computational complexity

In SureShrink it is first needed to calculate SURE $(|X_i|)$ for each coefficient X_i in order to solve the optimization problem (4.7). This involves the arrangement of X_i in an increasing $|X_i|$ values, which requires as much as $O(n \log n)$ calculations (where n is the length of the signal). Then, the threshold is applied to each decomposition level, resulting in $O(n)$ additional operations (it is assumed that the number of decomposition levels is much smaller than n so it is not taken into account). Finally, SureShrink complexity is $O(n \log n)$.

NeighBlock introduces a lower complexity. This algorithm divides each decomposition level to blocks with a constant size, and then calculates the energy of each block. The complexity of these calculations is $O(n)$ and it is also the complexity of this algorithm.

5.4 Choice of wavelet and number of resolution levels

Both methods do not address the important issues of the wavelet choice and the desired number of resolution level. For example, the wavelet should be chosen such that it is able to characterize well the signal. This can be decided according to the correlation between the wavelet and the signal.

Moreover, the choice of the number of the decomposition levels should be decided with care. For example, if the noise is hardly noticeable, we may need more levels, in order to get the fine details part of the signal. It may be even not needed to apply the methods above

on each level. It makes sense to work on different resolution level in somewhat different way, especially when there are many decomposition levels.

5.5 Smoothness

Both methods above assume that the functions to be recovered have some degree of smoothness. In general, they are assumed to belong to *Sobolev space* (for some $m, p \geq 2, C$):

$$W_p^m(C) = \left\{ f : \|f\|_p^p + \left\| \frac{d^m}{dt^m} f \right\|_p^p \leq C^p \right\}$$

where the parameter m is the degree of differentiability and C is a quantitative limit on the mth derivative. Actually, it can be proved (see [3], [5]) that both methods are supposed to perform well even for functions which belong to *Besov space*, which is more general than Sobolev space. Moreover, they are supposed to perform better than linear threshold methods.

However, there are cases in which the functions do not satisfy the requirement of smoothness, such as functions with many discontinuities, unbounded derivatives, etc. In such cases, some prior knowledge of the structure of the signal is probably needed and these methods may not work properly.

Better approach is these cases is to use *parametric* methods, or to assume some form of prior distributions and to use Bayesian methods (some of them are reviewed in [6]). However, these methods are very model-sensitive, and for relatively smooth functions the methods reviewed in this report provide better performance.

6 Conclusions

Wavelets serve as a powerful tool for the task of signal denoising. The ability to decompose a signal into different scales is very important for denoising, and it improves the analysis of the signal significantly.

Two methods of wavelet shrinkage were reviewed in this report. Whereas classical denoising methods of a signal remove high frequencies (i.e., smoothing of the signal) which are usually associated with noise, shrinkage methods attempt to remove whatever noise is present and retain whatever signal is present regardless of the frequency content of the signal. The latter methods has been theoretically proven to be nearly optimal when the smoothness of the signal to be recovered is unknown. In effect, no alternative procedure can perform better without some a priori knowledge on the smoothness class of the signal.

The methods that were presented in this paper are relatively simple. They use the principle of thresholding, without any complicated assumptions on the structure of the signal. That is, they are non-parametric methods, so no particular model of the signal is needed to be assumed. In addition, they have low computational complexity.

Both methods were implemented and simulation results were presented. The signals that were used are non-stationary ones which usually pose a problem for classical methods. Nevertheless, the methods presented in this report deal quite well with these signals.

Even from the small number of simulations in this report, it is clear that there is no single "best" wavelet-based denoising method. The methods and their parameters should be chosen according to the signals in hand.

References

[1] David L. Donoho and Iain M. Johnstone. Minimax estimation via wavelet shrinkage. Technical report, 1992. 3

[2] David L. Donoho and Jain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. 3, 10, 12, 13

[3] David L. Donoho and Iain M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, pages 1200–1224, 1995. 3, 11, 13, 15, 24

[4] David L. Donoho, Iain M. Johnstone, Gerard Kerkyacharian, and Dominique Picard. Wavelet shrinkage: asymptopia. *Journal of the Royal Statistical Society, Ser. B*, pages 371–394, 1995. 3

[5] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613 –627, May 1995. 3, 24

[6] Anestis Antoniadis, Jeremie Bigot, and Theofanis Sapatinas. Wavelet estimators in nonparametric regression: A comparative simulation study. *Journal of Statistical Software*, 6(6):1–83, June 2001. 3, 20, 24

[7] T.T. Cai and B.W. Silverman. Incorporating information on neighbouring coefficients into wavelet estimation. *Sankhya, Series A, 63*, 2001. 3, 19, 20

[8] S. G. Mallat. *A wavelet tour of signal processing: the sparse way.* 3rd edition, Academic Press, 2009. 4

[9] Shalom Raz. Mixed representation and their applications, lecture notes, Technion - Israel Institute of Technology, 2011. 4

[10] M. Steinbuch and M.J.G. van de Molengraft. Wavelet theory and applications: A literature study. Technical report, Eindhoven University of Technology, 2005. 4

[11] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989. 5

[12] S. Tsai. Wavelet transform and denoising. Master's thesis, Chapter 4, pp. 35-42, 2002. 6

[13] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):pp. 1135–1151, 1981. 11

[14] T. Tony Cai, Keywords James-stein, Estimator Adaptivity, Wavelet Block, and Thresholding Nonparametric. Adaptive wavelet estimation: A block thresholding and oracle inequality approach. *The Annals of Statistics*, 27:898–924, 1998. 19

[15] Lei Zhang and P. Bao. Denoising by spatial correlation thresholding. *IEEE Transactions on Circuits and Systems for Video Technology on*, 13(6):535 – 538, June 2003. 22

[16] James S. Walker and Ying-Jui Chen. Image denoising using tree-based wavelet subband correlations and shrinkage. *Opt. Eng. 39*, 2000. 22