Multiwavelets and EP Denoising

T. L. Berkopec[a], J. D. Lakey[b], M. C. Pereyra[c] and N. Tymes[c]

[a] MZA Associates Coorporation, Albuquerque, NM 87106.
[b] Department of Mathematical Sciences, New Mexico State University, Las Cruces, NM 88003.
[c] Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131.

ABSTRACT

In this paper we report on the performance of Lemarie uniwavelets and biwavelets for solving the ill-posed inverse problem of recovering the derivative of a noisy signal. The noise under consideration can be white gaussian or other noises like Tukey noise. The denoising procedures utilized are wavelet and biwavelet Efromovich-Pinsker (EP) estimators, which have been shown to be universal for both estimating the function and its derivative.

Keywords: Lemarié wavelets and multiwavelets, denoising, vaguelettes, EP estimator, block thresholding, inverse problems

1. INTRODUCTION

There are different approaches to statistical linear inverse problems, see the discussion in, for example, Abramovich-Silverman.[1] The problem under consideration amounts to estimating an unknown signal f from some data Kf, where K is some known linear operator. Moreover the observed data Y has been corrupted by an additive noise. More precisely, in the frame of an equidistant regression model,

$$Y_i = (Kf)(i/n) + \sigma\epsilon_i, \qquad i = 0, 1, 2, \cdots, n. \tag{1}$$

The random variables ϵ_i are independently identically distributed realizations of a random variable ϵ with zero mean noise and unit variance. The positive constant σ defines the standard deviation of the additive noise.

In this paper we want to discuss the case when K is the integration operator, thus the problem is to recover the derivative of a given noisy signal. The denoising methods are wavelet and biwavelet robust block threshold methods proposed by Efromovich et al..[2,3] This procedure was suggested by Efromovich and Pinsker[4,5] for a sharp optimal denoising based on using a trigonometric basis, therefore the name: EP estimator. One of the important features of this method is that it does not care about the underlying distribution of the noise. Thus, the noise under consideration need not always be a white Gaussian noise.

The classical methods utilized to solve such problems are *singular value decompositions* (SVD) which expand into a basis of eigenfunctions of the selfadjoint operator K^*K. The basis depends exclusively on K and ignores any information about the signal f. Donoho[6] introduced the *wavelet-vaguelette decomposition* (WVD) method to overcome some of the drawbacks of the singular value decomposition. In this method the noisy data is decomposed in a vaguelette basis, the coefficients are thresholded using a hard or soft threshold rule; and then they are mapped back into the original space using the wavelet basis. The problem of doing this is that the vaguelette coefficients of the white noise are no longer independent. Abramovich and Silverman[1] proposed the *vaguelette-wavelet decomposition* (VWD) method which performs a wavelet decomposition on the observed noisy data (hence this time white noise remains white), then maps thresholded coefficients using the vaguelettes. T. Cai[7] used a block threshold and the vaguelette-wavelet decomposition method, he showed that the estimator has a higher degree of adaptivity than the standard term-by-term threshold methods and it attains the exact optimal rates of convergence over a range of Besov spaces. All these methods apply to a variety of problems including convolution with homogeneous kernels (like Radon transform), but conditions on K are necessary to ensure that the transform or the inverse transform under K of the

Further author information:
T.L.B.: E-mail: berkopec@mza.com
J.D.L.: E-mail: jlakey@nmsu.edu
M.C.P.: E-mail: crisp@math.unm.edu
N.T.:E-mail: ntymes@math.unm.edu

wavelet basis is indeed a vaguelette system. The benchmark application is the estimation of the derivative of a noisy signal. Cai[7] shows that his derivative estimator is spacially adaptive and attains the local adaptive minimax rate for estimating the derivative at a point. In all these methods, the optimal threshold for reconstructing the signal is different than the optimal threshold for reconstructing the derivative.

We use an Efromovich-Pinsker (EP) block estimator and biorthogonal bases of biwavelets or wavelets that have the property that their derivatives are again biorthogonal (bi)wavelet bases, therefore the name *vaguelette-vaguelette decomposition* (VVD) method. The wavelets were constructed by Lemarié,[8] the generalization to multiwavelets is due to Lakey et al.[9]; we call them Lemarié (multi)wavelets. The biwavelets used generalize the well known Donovan-Geronimo-Hardin-Massopust[10] (DGHM) biwavelets, all the functions involved (wavelets and scaling functions) have very short support and symmetry plus enough regularity/approximation order, see Lakey and Pereyra.[11] Being a vaguelette-vaguelette method, the white noise will be correlated after performing a (bi)wavelet transform. This is a problem even in the case of orthogonal multiwavelets since a certain prefiltering is necessary that already contaminates the empirical multiwavelet coefficients with a nonstationary correlated noise. Different prefilters and different multiwavelets generate different noise distributions as very well describe Strela and Walden.[12] To overcome these difficulties and justify the use of multiwavelets to statisticians, denoising should be robust and self-learning, that is minimal or no knowledge at all about the discrete biwavelet transform (DBWT) or the prefilter is necessary to use the estimator. We report in some experiments using the (bi)wavelet EP estimator for the particular problem of recovering the derivative of a noisy signal which is well adapted to this class of (bi)wavelets. The (bi)wavelet EP estimator is optimal for both the function and its derivative, thus it is a *universal* estimator.[5]

There has been some work in the area of denoising using multiwavelets. It has been shown that in certain contexts multiwavelets could perform better than uniwavelets (due to more flexibility in the trade-off between support length, regularity and symmetries). Strela and Walden[12] did extensive denoising experiments and report that multiwavelets generally outperform scalar wavelets for image and signal denoising, and they suggest that a combination of Chui-Lian (CL) biwavelets and repeated row preprocessing is a good general method (they only considered white noise and they did not consider inverse problems). Efromovich et al.[2,3] reported on intensive Monte Carlo studies comparing the performance of term by term threshold for uniwavelets (symmlet 8), vector threshold for multiwavelets (CL, DGHM), and EP denoising for biwavelets (CL, Lemarié). The results showed that in terms of accuracy of denoising, the EP biwavelet estimator outperformed the threshold estimators. In terms of data compression, the EP estimator significantly outperformed the threshold estimates. There was no clear winner between uniwavelet and biwavelet threshold estimators. This conclusion is supported by other known numerical studies. The noise used was Gaussian normal.

The situation changes dramatically if the noise is not Gaussian. Some experiments with Tukey noise (a mixture of two zero mean normal variables) were studied by Efromovich et al..[2,3] Here the robustness of the biwavelet EP estimator shined.

Despite these encouraging results, multiwavelet denoising remains technically more difficult. Is the additional effort associated to multiwavelets worthwhile? The results in Efromovich et al.[2,3] seem to indicate that multiwavelets can be an attractive tool in statistical applications, in particular for denoising non-Gaussian noise and solving inverse problems. In this paper we want to compare the performance of the EP estimator for uniwavelets and biwavelets for the particular problem of recovery of the derivative of a noisy signal. We expect that the better trade-off between short support, symmetries and regularity that is gained with the multiwavelets will make it a better estimator. We think the particular biwavelets used will be specially well adapted to boundary problems, see Lakey and Pereyra[13] and Sec. 4.10 in Efromovich.[5]

The paper is organized as follows. In the first two sections we record the Lemarié biwavelets and some of their properties, as well as the analogue calculations required for the simpler case of the uniwavelets. In the third section we recall the Efromovich-Pinsker block estimator for (bi)wavelets. In the fourth section we discuss the problem of recovering the derivative from a noisy signal. In the last sections we report on some preliminary experiments performed.

2. BIWAVELETS

We will use the notation $\frac{1}{2}\Phi(\frac{x}{2}) = \sum C_k \Phi(x-k)$ and $\frac{1}{2}\Psi(\frac{x}{2}) = \sum D_k \Phi(x-k)$ to denote the fundamental scaling and wavelet equations. The *scaling vector* is $\Phi(x) = [\phi^1(x), \phi^2(x)]^T$ and the wavelet vector is $\Psi(x) = [\psi^1(x), \psi^2(x)]^T$.

The coefficients C_k and D_k are two-by-two matrices. The low-pass, or scaling filter is a matrix with polynomial entries $H(z) = \sum C_k z^k$. The high-pass, or wavelet filter is another matrix with polynomial entries $F(z) = \sum D_k z^k$.

2.1. Filters and Biorthogonality

We are interested in biorthogonal biwavelets and multiresolution analyses. In that case there will be a second pair of scaling and wavelet vectors, $\tilde{\Phi}$ and $\tilde{\Psi}$, satisfying the fundamental scaling and wavelet equations, $\frac{1}{2}\tilde{\Phi}(\frac{x}{2}) = \sum \tilde{C}_k \tilde{\Phi}(x-k)$ and $\frac{1}{2}\tilde{\Psi}(\frac{x}{2}) = \sum \tilde{D}_k \tilde{\Phi}(x-k)$. The corresponding low-pass and high-pass filters are $\tilde{H}(z) = \sum \tilde{C}_k z^k$, and $\tilde{F}(z) = \sum \tilde{D}_k z^k$. It is well-known that the conditions that the scaling functions and wavelets give rise to biorthogonal MRAs can be encoded in terms of the filters, see Strela.[14] These conditions are summarized in Table 2.1. They are best checked with the aid of a symbolic computational tool such as Maple. Here A^* denotes the conjugate transpose of the matrix A. In the uniwavelet case the conditions are identical except that in that case we have ordinary multiplication, $I = 1$ and A^* is just the complex conjugate of the number A.

Table 2.1: Conditions of Biorthogonality
$H(z)\tilde{H}^*(z) + H(-z)\tilde{H}^*(-z) = I$
$F(z)\tilde{F}^*(z) + F(-z)\tilde{F}^*(-z) = I$
$H(z)\tilde{F}^*(z) + H(-z)\tilde{F}^*(-z) = 0$
$F(z)\tilde{H}^*(z) + F(-z)\tilde{H}^*(-z) = 0$

Once these conditions are satisfied the following reconstruction formula and Riesz basis estimates hold for square integrable functions,

$$f = \sum_{j,k} \langle f, \tilde{\Psi}_{j,k} \rangle \Psi_{j,k}, \qquad A \sum_{j,k} \|\langle f, \tilde{\Psi}_{j,k} \rangle\|^2 \leq \|f\|^2 \leq B \sum_{j,k} \|\langle f, \tilde{\Psi}_{j,k} \rangle\|^2.$$

Where as usual $\Psi_{j,k}(t) = 2^{-j/2}\Psi(2^{-j}t - k)$, $\langle f, \Psi_{j,k} \rangle = [\langle f, \psi^1_{j,k} \rangle, \langle f, \psi^2_{j,k} \rangle]$, and $\|[a,b]\|^2 = |a|^2 + |b|^2$.

2.2. Lemarié Multiwavelets

Lemarié[8] showed that given a pair of biorthogonal scalar MRAs with sufficient regularity, one could build from these another pair connected to the initial pair by differentiation and integration. In his thesis, Strela[14] found a technique that amounts to the multiwavelet version of Lemarié's technique. In the scalar case the trick is to add a zero to the dual low-pass filter at $z = -1$. This is equivalent to multiplying the dual low-pass filter by $\frac{1+z}{2}$. In the multiwavelet setup the low-pass filter is vector-valued and the analogous procedure amounts to adding an eigenvalue zero to the low-pass filter at $z = -1$ in an appropriate manner. As is typical of multiwavelets, there is some flexibility in carrying out this extension of Lemarié's method. The method is encoded in Strela's *two-scale transform*.[14] It amounts to finding a suitable *transition matrix* M and then forming the new biorthogonal low pass filters (H_-, H_+), as shown in Table 2.2, which also shows the high pass filters (F_-, F_+) and the corresponding differential relations at the level of the new scaling functions (Φ^-, Φ^+), and new wavelets (Ψ^-, Ψ^+), see Lakey et al..[9]

Table 2.2: Smoothened and roughened biorthogonal MRAs		
smoothened scaling	$H_+(z) = \frac{1}{2}M^*(z^2)\tilde{H}(z)M^{*-1}(z)$	$\frac{d}{dx}\Phi^+(x) = -T_{M^*}\tilde{\Phi}(x)$
roughened scaling	$H_-(z) = 2M^{-1}(z^2)H(z)M(z)$	$\frac{d}{dx}\Phi(x) = T_M\Phi^-(x)$
smoothened wavelet	$F_+(z) = \frac{1}{2}\tilde{F}(z)M^{*-1}(z)$	$\frac{d}{dx}\Psi^+(x) = -\tilde{\Psi}(x)$
roughened wavelet	$F_-(z) = 2F(z)M(z)$	$\frac{d}{dx}\Psi(x) = \Psi^-(x)$

It is a simple matter to check that the conditions of biorthogonality are preserved. *So at the level of wavelets this process really amounts to integrating or differentiating the wavelet components!* The operator T_M is an operator-valued matrix whose entries are polynomials in the shift operator. Its symbol is M.

2.3. HM Biwavelets

We will consider a family of biwavelets (denoted HM) introduced by Hardin and Marasovich[15] that generalize the DGHM[10] biwavelets. When one imposes among the HM family the constraint that the scaling functions and wavelets should be symmetric, as we do here, the coefficients for the HM scaling and wavelet filters will depend on a parameter s. Furthermore, the parameter $\tilde{s}$ that is used to denote the biorthogonal multiresolution analysis (MRA) will be related to the parameter s for the original MRA by $\tilde{s} = \dfrac{1+2s}{5s-2}$. Continuity and approximation properties of the scaling/wavelet vectors depend on the values of s. The low-pass or scaling filter, is a matrix with polynomial entries $H_s(z) = \sum C_k(s) z^k$ where the C_k are the two-by-two scaling matrices with entries depending on the parameter s. The corresponding high-pass or wavelet filter has the form $F_s(z) = \sum D_k(s) z^k$. The filters for the scaling and wavelet functions in the multiresolution analysis biorthogonal to that generated by $H_s(z)$ have the form $H_{\tilde{s}}(z)$ and $F_{\tilde{s}}(z)$ respectively. To build differentiation into this scale of biorthogonal multiresolution analyses, the filter coefficients have been chosen with a normalization slightly different from that given by Hardin and Marosovich.[15] The coefficients are listed in Table 1 in Appendix A. In general, those functions are Lipschitz continuous of order α if $|s| < 2^{-\alpha}$, but can be discontinuous for $|s| \geq 1$. The orthogonal cases correspond to $s = 1$ and $s = -1/5$: the DGHM[10] case. In what follows we shall always assume that $|s| \leq 1/2$.

2.4. Smoothed and roughened HM filters

The recipe for smoothing and roughening the HM filters yields the corresponding coefficients for the Lemarié smoothened and roughened scaling and multiwavelets; which are shown in Table 2 in Appendix A (here k is running from $k = -2$ in the top row to $k = 2$ in the bottom row). The transition matrix in our case is $M(z) = \begin{bmatrix} 0 & 2\sqrt{2} \\ 1-z & -z-1 \end{bmatrix}$. Our choice of filters will allow us to choose the same M for all values of the parameter s.

The first components of the smoothed and roughened filters are symmetric about $x = 0$ whereas the second components are both antisymmetric about $x = 0$. It is not obvious from the filter equations that the scaling supports should all have length two: this ultimately depends on some subtle cancellation properties among the components. The details can be found in Lakey and Pereyra.[11]

We would like to emphasize that the smoothing transition is applied to the $\tilde{s}$ filter while the roughening transition is applied to the s filter.

3. LEMARIÉ UNIWAVELETS

Let us record in Table 3 the uniwavelet analogue of Table 2.2, where $M(z) = 1 - z$, and the filters are scalar-valued.

Table 3: Smoothened and roughened biorthogonal scalar MRAs

smoothened scaling	$H_+(z) = \frac{1+z}{2z}\tilde{H}(z)$	$\frac{d}{dx}\phi^+(x) = \tilde{\phi}(x+1) - \tilde{\phi}(x)$
roughened scaling	$H_-(z) = \frac{2}{1+z}H(z)$	$\frac{d}{dx}\phi(x) = \phi_-(x) - \phi^-(x-1)$
smoothened wavelet	$F_+(z) = \frac{z}{2(z-1)}\tilde{F}(z)$	$\frac{d}{dx}\psi^+(x) = -\tilde{\psi}(x)$
roughened wavelet	$F_-(z) = 2(1-z)F(z)$	$\frac{d}{dx}\psi(x) = \psi^-(x)$

We will record the scalar filters as polynomials (in z^n where $n \in Z$) whose coefficients are the data utilized by the wavelet toolbox. Remember also that z is in the unit disc, therefore, $\overline{z} = z^{-1}$. To fix notation, the biorthogonal scaling functions are denoted $(\phi, \tilde{\phi})$, and the corresponding low-pass filters by $(H, \tilde{H})$; the biorthogonal wavelets are denoted $(\psi, \tilde{\psi})$, and the corresponding high-pass filters by $(F, \tilde{F})$. The high pass filters $(F, \tilde{F})$ are found from the low-pass filter $(H, \tilde{H})$ by the usual conjugate flip, more precisely: $F(z) = z\overline{\tilde{H}(-z)}$, $\quad \tilde{F}(z) = z\overline{H(-z)}$. Notice that we could compute the high-pass filters (F_-, F_+) using the conjugate flip trick and we will obtain the same filters as recorded in Table 3 up to a factor of 4: more precisely,

$$F_-(z) = \frac{(1-z)}{2}F(z), \quad \frac{d}{dx}\psi(x) = 4\psi^-(x); \quad F_+(z) = \frac{2z}{(z-1)}\tilde{F}(z), \quad \frac{d}{dx}\psi^+(x) = -4\psi(x). \tag{2}$$

These last formulas were the ones obtained by Lemarié,[8] and these are the formulas we will use when utilizing MATLAB, since MATLAB computes scalar high pass filters from the low pass filters by the conjugate flip trick. We will use $\tilde{H}$ and H_+ as the decomposition filters and H and H_- as the reconstruction filters.

We have chosen the biorthogonal symmetric spline, Bior3.1, with parameter $\tilde{N} = 3$, $N = 1$; as described in Daubechies[16] p.271-278. This particular class has the virtue that the coefficients are very simple dyadic fractions. This biorthogonal wavelets are symmetric and have relatively small support and enough smoothness so that we can differentiate and still get a continuous function. We record in Table 3 in Appendix A the low-pass filters and coefficients.

We will also use the orthogonal Symmlet 8 for which we have not yet computed the smoothened and roughened filters. One could apply this recipe to any known wavelet as long as it has enough smoothness. In particular it is straightforward how to apply it to other members of the Bior$(\tilde{N}, N)$ family, or to the Daubechies' wavelets (or symmlets) whose filter polynomials are explicitely known.[16]

4. EP ESTIMATOR

Efromovich and Pinsker[4] introduced the EP block estimator. This is a data-driven estimator that is sharp minimax over Sobolev and analytic functions classes. It is very simple to implement since no optimization problem must be solved. Efromovich et al.[2,3] introduced an EP block estimator for biwavelets, asymptotic optimality of the estimator over a class of functions that includes the Besov class $B^{\alpha}_{\infty,\infty}$ was established, and some numerical simulations were presented. In particular an adaptive method of recovering the derivative of a noisy signal based on a vaguelette-vaguelette decomposition was developed. The basis used consists of the Lemarié biwavelets recorded in the previous section. In this note we want to compare the EP block estimators defined for Lemarié wavelets and biwavelets in the context of recovering the derivative of a noisy signal.

Let $\{\psi_i(t), i = 1, 2, \ldots\}$ be an orthonormal basis (trigonometric, wavelet, etc) and $f(t) = \sum_{i=1}^{\infty} \theta_i \psi_i(t)$, where θ_i are the corresponding coefficients that can be estimated by empirical coefficients $\tilde{\theta}_i$ based on the noisy equidistant observations

$$Y_i = f(i/n) + \sigma\epsilon_i, \qquad i = 0, 1, \ldots, n.$$

It is assumed that the empirical coefficients are unbiased estimates, i.e. $E(\tilde{\theta}_i) = \theta_i$. The random variables ϵ_i are independently identically distributed realizations of a random variable ϵ with zero mean noise and unit variance.

Let us consider a block smoothing oracle

$$\tilde{f}^*(t) = \sum_{m=1}^{\infty} \lambda_m \sum_{i \in T_m} \tilde{\theta}_i \psi_i(t).$$

Here the blocks T_m are consecutive non-overlapping blocks of consecutive positive integer numbers, and the weights $0 \le \lambda_m \le 1$ may depend on f. The notion of an "oracle" is familiar in the curve estimation literature and it is a procedure that knows both the data and the underlying signal f; a discussion about different types of oracles can be found in Efromovich[5] Sec. 3.2, and in Mallat[17] Sec. 10.2. Thus, the oracle can use the information about an underlying signal to choose optimal weights. It is not hard to see that an optimal weight λ^*_m that minimizes the mean square error $E\{\sum_{i \in T_m} (\lambda_m \tilde{\theta}_i - \theta_i)^2\}$ (and thus by Parseval identity it also minimizes the mean integrated squared error of $\tilde{f}^*$) is

$$\lambda^*_m = \frac{\sum_{i \in T_m} \theta_i^2}{\sum_{i \in T_m} \theta_i^2 + \sum_{i \in T_m} \mathrm{Var}(\tilde{\theta}_i)}. \tag{3}$$

The optimal weight λ^*_m depends on the mean value of the squared coefficients from the block. This is the key point of any block procedure, because the mean of the squared coefficients may be estimated better than a single one. The larger the block T_m, the better the estimation of λ^*_m. On the other hand, the larger a block is, the farther λ^*_m is from the optimal individual shrinkage. Hence the choice of blocks is a trade-off between mimicking optimal singular shrinkage and the better accuracy of estimating λ^*_m.[5]

EP denoising is based on directly mimicking (3). Assume for a moment that the variances $\mathrm{Var}(\tilde{\theta}_i)$ are known. Then the only unknown functional of f is the sum of the square of the coefficients $\sum_{i \in T_m} \theta_i^2$. An unbiased estimate

of this functional is $\left[\sum_{i\in T_m}\tilde{\theta}_i^2 - \sum_{i\in T_m}\mathrm{Var}(\tilde{\theta}_i)\right]$, and this is the estimate used. Note that the functional is not always positive so a threshold should be used to get a positive estimate. We will use a hard threshold, but a soft one could be used as well. This yields the following estimate for the weights,

$$\hat{\lambda}_m = \frac{\sum_{i\in T_m}\tilde{\theta}_i^2 - \sum_{i\in T_m}\mathrm{Var}(\tilde{\theta}_i)}{\sum_{i\in T_m}\tilde{\theta}_i^2}\, I\left(\sum_{i\in T_m}\tilde{\theta}_i^2 > c_m \sum_{i\in T_m}\mathrm{Var}(\tilde{\theta}_i)\right), \tag{4}$$

where $c_m > 1$ may depend on the number of data points, and $I(\cdot)$ is the indicator function.

The asymptotic properties of the EP denoising can be found in Efromovich and Pinsker[4] and in Efromovich[5] Ch. 7. Note that $\hat{\lambda}_m$ depends only on the variance of the underlying noise but not on its distribution. On the other hand threshold procedures are based on the underlying distribution of the noise. This explains the robustness of the EP denoising.

4.1. EP denoising for wavelets

When the basis considered is a biorthogonal basis of wavelets or biwavelets one considers consecutive blocks of equal length for each scale j, the blocks might depend on the number of data points $n = 2^l$. The blocks $T_m^{n,j}$!!a have length $L_{j,n}$. There is a wide variety of blocks that imply rate optimality; for instance blocks with logarithmic length have very attractive global and pointwise properties as was pointed out by Cai,[7] see also Sec. 7.4 in Efromovich.[5]

In the case of wavelets suppose the function f can be approximated by an inhomogeneous wavelet expansion

$$f_J(t) = \sum_{k=0}^{n/2^J-1} s_{J,k}\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_{k=0}^{n/2^j-1} d_{j,k}\psi_{j,k}(t). \tag{5}$$

Let the empirical wavelet coefficients be $\hat{s}_{J,k}$, $\hat{d}_{j,k}$. Then the wavelet EP adaptive estimator that mimicks the corresponding oracle is defined by

$$\begin{aligned}\hat{f}_J^*(t) \;=\; & \sum_{k=0}^{n/2^J-1}\hat{s}_{J,k}\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_{m=1}^{n/(2^j L_{j,n})}\left\{\left(\frac{\sum_{k\in T_m^{n,j}}\hat{d}_{j,k}^2 - \nu_m^{n,j}}{\sum_{k\in T_m^{n,j}}\hat{d}_{j,k}^2}\right)\right. \\ & \left.\times\, I\left(\sum_{k\in T_m^{n,j}}\hat{d}_{j,k}^2 > c_m^{n,j}\nu_m^{n,j}\right)\right\}\sum_{k\in T_m^{n,j}}\hat{d}_{j,k}\psi_{j,k}(t).\end{aligned} \tag{6}$$

Where $\nu_m^{n,j} = \sum_{k\in T_m^{n,j}}\mathrm{Var}(\hat{d}_{j,k})$. Notice that $\frac{n}{L_{n,j}}\nu_m^{n,j}$ is the average variance of the noise for the block $T_m^{n,j}$. Which can be calculated empirically using a learning process and Montecarlo simulations.

4.2. EP denoising for biwavelets

In the case of biwavelets suppose the function f can be approximated by an inhomogeneous biwavelet expansion

$$f_J(t) = \sum_{k=0}^{n/2^J-1} S_{J,k}\Phi_{J,k}(t) + \sum_{j=1}^{J}\sum_{k=0}^{n/2^j-1} D_{j,k}\Psi_{j,k}(t), \tag{7}$$

where here $S_{j,k} = [s_{j,k}^1, s_{j,k}^2]$ and $D_{j,k} = [d_{j,k}^1, d_{j,k}^2]$ are the approximation and detail coefficients in the biwavelet decomposition. Let the empirical biwavelet coefficients be $\hat{S}_{J,k}$, $\hat{D}_{j,k}$. Then the biwavelet EP adaptive estimator that mimicks the corresponding oracle is defined by

$$\begin{aligned}\hat{f}_J^*(t) \;=\; & \sum_{k=0}^{n/2^J-1}\hat{S}_{J,k}\Phi_{J,k}(t) + \sum_{j=1}^{J}\sum_{m=1}^{n/(2^j L_{j,n})}\left\{\left(\frac{\sum_{k\in T_m^{n,j}}\hat{D}_{j,k}\hat{D}_{j,k}^T - \mathcal{V}_m^{n,j}}{\sum_{k\in T_m^{n,j}}\hat{D}_{j,k}\hat{D}_{j,k}^T}\right)\right. \\ & \left.\times\, I\left(\sum_{k\in T_m^{n,j}}\hat{D}_{j,k}\hat{D}_{j,k}^T > c_m^{n,j}\mathcal{V}_m^{n,j}\right)\right\}\sum_{k\in T_m^{n,j}}\hat{D}_{j,k}\Psi_{j,k}(t),\end{aligned} \tag{8}$$

where $\mathcal{V}_m^{n,j} = \sum_{k \in T_m^{n,j}} \left(\mathrm{Var}(\tilde{d}_{j,k}^1) + \mathrm{Var}(\tilde{d}_{j,k}^2) \right)$. Here the blocks consist of consecutive arrays of the form $[d_{j,m}^1, d_{j,m}^2]$. Notice that $\frac{n}{2L_{n,j}} \mathcal{V}_m^{n,j}$ is the average variance of the noise for the block $T_m^{n,j}$ and can be calculated empirically using a learning process and Montecarlo simulations, see Efromovich et al.[2,3] Blocks are chosen as small as possible to take advantage of the short support and symmetry of the biwavelets.

4.3. Estimating the derivative

To apply a wavelet based EP denoising procedure for recovering of the derivative from a noisy signal, the (bi)wavelets used are those for which the Lemarié smoothed and roughened biorthogonal system can be found, as described in the first sections. The idea for the recovery of the derivative from noisy observations is simple. If we differentiate the approximation (5) and use the formulas for the derivatives of the scaling and wavelets that appear in Table 3 and in (2), we can write the derivative f_J' of f_J as

$$f_J'(t) = \sum_{k=0}^{n/2^J-1} 2^{-J} s_{J,k} (\phi_{J,k}^-(t) - \phi_{J,k+1}^-(t)) + \sum_{j=1}^{J} \sum_{k=0}^{n/2^j-1} 2^{-J} d_{j,k} 4 \psi_{j,k}^-(t).$$

To construct an estimator $(\hat{f}')_J$ for the derivative we just plug the estimated coefficients for $\hat{f}_J$ in the above equation. Numerically it means that we can use the inverse wavelet transform with the filters (H_-, F_-) corresponding to the roughened wavelet and scaling functions (ϕ^-, ψ^-) to recover the derivative.

Similarly for the biwavelets,

$$f_J'(t) = \sum_{k=0}^{n/2^J-1} 2^{-J} S_{J,k} T_M \Phi_{J,k}^-(t) + \sum_{j=1}^{J} \sum_{k=0}^{n/2^j-1} 2^{-J} D_{j,k} \Psi_{j,k}^-(t),$$

where here $T_M \Phi(t) = \left(2\sqrt{2}\phi^2(t), \phi^1(t) - \phi^1(t-1) - \phi^2(t) - \phi^2(t-1) \right)^T$ is the shift matrix operator with symbol M.

Under certain technical assumptions one can show that the EP wavelet and biwavelet estimators and their derivatives are rate minimax. More precisely,

$$\sup_{f \in HKP_\alpha} E\{ \int_0^1 \left(\hat{f}_l(t) - f(t) \right)^2 dt \} \leq C n^{-2\alpha/(2\alpha+1)},$$

and

$$\sup_{f \in HKP_\alpha} E\{ \int_0^1 \left((\hat{f}')_l(t) - f'(t) \right)^2 dt \} \leq C n^{-2(\alpha-1)/(2\alpha+1)};$$

where the class of functions $HKP_\alpha \subset L^2[0,1]$ is the Hall-Kerkyacharian-Picard function class.[18,19] A function in this class is the sum of a smooth function g from the Besov space $B_{\infty,\infty}^\alpha$ and an irregular function b (for example a piecewise polynomial with a finite number of discontinuities, this includes the "chirp" and "doppler" signals familiar in the wavelet literature). The details can be found in Efromovich.[3] The class of noises allowed by the theorem includes, for example, Tukey noise, which was used in the simulations.

This theory explains the universality of the signal and the derivative estimators. In practice we might not implement the Lemarié differentiation but instead we will calculate a numerical derivative. For smooth signals it is known that the numerical derivative and the Lemarié derivatives are very close and in the limit they coincide with the continuous derivative of the function, see Lakey and Pereyra.[11,13]

5. DESCRIPTION OF EXPERIMENTS

Now we can explain the particular EP denoising procedure suggested for biwavelets and samples up to several thousands. Signals and uniwavelets procedures used are supported by the wavelet package of MATLAB. Known multiwavelet denoising procedures are supported by Strela's software available at *http://math.dartmouth.edu/strela*, and described in Strela and Walden.[12]

We are considering samples of size $n = 1024$. In the case of biwavelets, six resolution scales are used in (7), that is $J = 6$. For the scale functions and the two coarsest scales of wavelet functions, the length of the blocks is 1; i.e. each

coefficient of these scales is smoothed individually. Also for the coarsest scales, $j = 5, 6$, the coefficients $c_m^{n,j}$ are all the same and equal to 1. For the finer scales the blocks are increased. For the fourth scale the blocks are single arrays $[d_{4,m}^1, d_{4,m}^2]$, $m = 1, \ldots, n/2^4$. For the third scale the blocks include two adjoint single arrays, for the second scale they include three adjoint arrays, and for the finest scale they include four adjoint arrays. For these scales $c_m^{n,j} = 5$, $j = 1, 2, 3, 4$. Strela's software allows to calculate the (noise) variances $\mathrm{Var}(\tilde{\theta}_i)$ exactly. But we would like the method to be self-learning, that is, no information about the prefilter or the discrete biwavelet transform (DBWT) should be used. In other words, the EP estimator considers the software as a "black-box". If the parameter σ in (1) is known then a Monte Carlo learning procedure can be used to estimate the variances, see Efromovich et al..[2,3] To estimate the parameter σ (which is assumed known in Strela's software) any robust method known for uniwavelets can be used. All these methods are based on analyzing wavelet coefficients on the finest scale. For multiwavelets the only difference is that the multiwavelet coefficients should be rescaled using the results of the learning process. Then, if an underlying signal is zero, the variance of the rescaled empirical coefficients is the wished σ^2.

In the case of the wavelets the procedure used for the EP denoising is similar.

Figures 1 and 2 illustrate the case of equidistant regression on $[0, 1]$. The first figure shows a noisy signal. The white noise is extremely low, $\sigma = 0.03$, so there is no problem identifying the underlying signal. The next diagram shows the numerical derivative of the noisy signal, and illustrates the ill-posedness nature of the problem: a small deviation in the signal causes a huge deviation in its derivative. The third diagram illustrates how the EP estimator together with the HM biwavelets perform. First the biwavelet EP estimator filters the signal and the reconstructed signal is presented. The recovered derivative is shown in the last diagram. As we mentioned before, the same software is used for denoising the signal and recovering its derivative. The same coefficients multiplied by a known factor are used for reconstructing the derivative, but with different biwavelets and biscaling functions. Recall that threshold uniwavelet estimators use different threshold levels for denoising an underlying signal and recovering the derivative, also a special software is needed for recovering the derivative. The first two diagrams in Figure 2 show the signal denoised using the EP estimator for the orthogonal wavelet Symmlet 8, and its derivative recovered numerically. The last two diagrams show the signal denoised using the universal threshold for the symmelet 8 wavelet(a wideused method supported by MATLAB), and its numerical derivative.

Figures 3 and 4 are similar to Figures 1 and 2. The integral of the familiar spacially inhomogeneous signal "bumps" is observed in Tukey noise with a standard deviation $\sigma = 0.03$ Remember that a Tukey random ξ variable is created by a mixture of two zero mean normal variables. In particular here $\xi = (u\epsilon_1 + (1-u)4\epsilon_2)/(.85)^{1/2}$ where ϵ_1, ϵ_2 and u are independent random variables, the first two being standard normal and the last one Bernoulli with probability $P(u = 1) = .9$. This noise is a classical example in the theory of robust estimation, see Sec. 4.6 in Efromovich.[3] In the first two diagrams in Figure 3, the noisy signal is shown together with its numerical derivative. In the last diagrams the HM biwavelet EP estimator is used for recovering the signal, the derivative is recovered using the HM/Lemarié biwavelets. In Figure 4 the Symmlet 8 EP estimator is used to recover the signal and its derivative is found numerically. In the the last diagrams the universal threshold estimate is used with Symmlet 8 to recover the signal and the corresponding derivative is found numerically.

6. CONCLUSION

The EP block threshold procedure is a robust method. We hope to have illustrated this fact both solving a classical inverse problem and departing from the traditional white Gaussian noise frame. Both the wavelet and biwavelet estimators should perform quite well, since there is an asymptotic theory supporting the method. Nevertheless we expected the biwavelets to do numerically a better job because of the better trade-off between short supports, symmetry and smoothness. The few experiments we have done indicate that when the signal is contaminated with white noise with small standard deviation σ, the biwavelet EP estimator performs well consistently. The wavelet Bior3.1 EP estimator does a poor job and so far we don't understand the reason (large Riesz bounds? non-optimal EP parameters?). On the other hand the orthogonal wavelet Symmlet 8 EP estimator sometimes does a better job -in terms of mean square errors- than the biwavelet estimator but sometimes does much worst; it seems to oversmooth the signal and therefore its derivative too. For the Tukey noise the biwavelet EP estimator did a remarkable job; much better than the Symmlet 8 EP and universal estimators. Overall the EP estimators did better than the Universal threshold, in particular the biwavelet estimator seem to work remarkably well, specially for the Tukey noise examples. So far these are just impressions based in a limited number of experiments. Extensive Monte Carlo studies are under way to better compare EP estimators for biwavelets and wavelets.

ACKNOWLEDGMENTS

We would like to acknowledge Prof. S. Efromovich who provided us with an unpublished manuscript and who was very generous with his time and ideas.

REFERENCES

1. F. Abramovich, B.W. Silverman, "Wavelet decomposition approaches to statistical inverse problems", *Biometrika* **85**, pp. 115-129, 1998.
2. S. Efromovich, J. Lakey, C. Pereyra, N. Tymes, "Data-driven and optimal denoising of a signal and recovery of its derivative using multiwavelets", Technical report, University of New Mexico, 1998.
3. S. Efromovich, "Multiwavelets and denoising", Preprint 2001.
4. S. Efromovich, M. Pinsker, "A learning algorithm for nonparametric filtering", *Automation and Remote Control* **24**, pp. 1434-1440, 1984.
5. S. Efromovich, *Non-parametric curve estimation: methods, theory and applications*, Springer, New York, 1999.
6. D.L. Donoho, "Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition", *Appl. Comput. Harm. Anal.* **2**, pp. 101-126, 1995.
7. T.T. Cai, "On adaptive wavelet estimation of a derivative and other related linear inverse problems", Technical report, Dept. of Statistics, Purdue Univ, 1999.
8. P.-G. Lemarié-Rieusset, "Analyses multi-resolution non orthogonales, commutation entre projecteurs et derivation et ondelettes vecteur a divergence nulle", *Rev. Mat. Iberoamericana* **8**, pp. 222-237, 1992.
9. J. Lakey, P. Massopust, C. Pereyra, "Divergence-free multiwavelets", *Approximation Theory IX*, C. Chui and L. Schumaker, eds., pp 161-68, Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 1998.
10. G. Donovan, J. Geronimo, D. Hardin, P. Massopust, "Construction of orthogonal wavelets using fractal functions", *SIAM J. of Math. Anal.*, **27**, pp. 1158-1192, 1996.
11. J. Lakey, C. Pereyra, "Divergence-free multiwavelets on rectangular domains", in *Wavelet analysis and multiresolution methods*, Xe ed. *Lecture Notes in Pure and Applied Mathematics* **212**, pp. 203-240, Dekker, New York, 2000.
12. V. Strela, A.T. Walden, " Signal and image denoising via wavelet thresholding: orthogonal and biorthogonal, scalar and multiple wavelet transforms", *Non-linear and nonstationary signal processing'*, W.J. Fitzgerald, R.L. Smith, A.T. Walden, and P.C. Young, eds. Cambridge University Press, Cambridge, 2001.
13. J. Lakey, C. Pereyra, "Multiwavelets on the interval and divergence-free wavelets", in *Wavelet Applications in Signal and Image Processing VII*, Aldroubi et al. eds. *Proc. of the SPIE* **3813**, pp. 162-173 (1999).
14. V. Strela, , PhD Thesis, MIT 1995.
15. D. Hardin, Marasovich, "Biorthogonal multiwavelets on $[-1, 1]$", Preprint 1998.
16. I. Daubechies, *Ten lectures on wavelets*, SIAM, Philadelphia, 1992.
17. S. Mallat, *A wavelet tour of signal processing*, University Press, New York, 1999.
18. P. Hall, G. Kerkyacharian, D. Picard, "Block threshold rules for curve estimation using kernel and wavelet methods", *Ann. Statist.* **26**, pp. 922-942 (1998).
19. P. Hall, G. Kerkyacharian, D. Picard, "On the minimax optimality of block thresholded wavelet estimators", *Statistica Sinica* **9**, pp. 33-50 (1999).

APPENDIX A. TABLES OF FILTER COEFFICIENTS

Table 1: HM scaling and wavelet coefficients

j	scaling coefficient $C_j(s)$	wavelet coefficient $D_j(s)$
-2	$\frac{1}{24}\begin{bmatrix} 0 & -(1+2s)\sqrt{2} \\ 0 & 0 \end{bmatrix}$	$\frac{1}{24}\begin{bmatrix} 0 & -(1+2s)\sqrt{2} \\ 0 & -(2+4s) \end{bmatrix}$
-1	$\frac{1}{24}\begin{bmatrix} -2+8s & (5-2s)\sqrt{2} \\ 0 & 0 \end{bmatrix}$	$\frac{1}{24}\begin{bmatrix} -2+8s & (5-2s)\sqrt{2} \\ \sqrt{2}(-2+8s) & 10-4s \end{bmatrix}$
0	$\frac{1}{24}\begin{bmatrix} 12 & (5-2s)\sqrt{2} \\ 0 & 8+4s \end{bmatrix}$	$\frac{1}{24}\begin{bmatrix} -12 & (5-2s)\sqrt{2} \\ 0 & 4s-10 \end{bmatrix}$
1	$\frac{1}{24}\begin{bmatrix} -2+8s & -(1+2s)\sqrt{2} \\ 8\sqrt{2}(1-s) & 8+4s \end{bmatrix}$	$\frac{1}{24}\begin{bmatrix} -2+8s & -(1+2s)\sqrt{2} \\ \sqrt{2}(2-8s) & 2+4s \end{bmatrix}$

Table 2: Smoothed and roughened scaling and biwavelet HM/Lemarié coefficients

smooth scaling $C_k^+(s)$	smooth wavelet $D_k^+(s)$	rough scaling $C_k^-(s)$	rough wavelet $D_k^-(s)$
$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} -\frac{1}{2}-s & \frac{1}{2}+s \\ -\frac{1}{2}-s & \frac{1}{2}+s \end{bmatrix}$	$\frac{1+2s}{12}\begin{bmatrix} -\sqrt{2} & \sqrt{2} \\ -2 & 2 \end{bmatrix}$
$\begin{bmatrix} \frac{1}{4} & \frac{(1-s)}{4(5s-2)} \\ \frac{3}{8} & \frac{(4-s)}{8(5s-2)} \end{bmatrix}$	$\begin{bmatrix} \frac{\sqrt{2}}{32} & \frac{\sqrt{2}}{32}\frac{2+s}{5s-2} \\ \frac{1}{16} & \frac{1}{16}\frac{s+2}{(5s-2)} \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} 3 & -4+10s \\ 3 & -4+10s \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} 6\sqrt{2} & \sqrt{2}(20s-8) \\ 12 & 40s-16 \end{bmatrix}$
$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{4} \end{bmatrix}$	$\begin{bmatrix} 0 & -\frac{1}{16}\sqrt{2} \\ -\frac{1}{8} & 0 \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} 7+2s & 0 \\ 0 & 7+2s \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} 0 & \sqrt{2}(4s-34) \\ 8s-20 & 0 \end{bmatrix}$
$\begin{bmatrix} \frac{1}{4} & \frac{(s-1)}{4(5s-2)} \\ -\frac{3}{8} & \frac{(4-s)}{8(5s-2)} \end{bmatrix}$	$\begin{bmatrix} -\frac{\sqrt{2}}{32} & \frac{\sqrt{2}}{32}\frac{s+2}{5s-2} \\ \frac{1}{16} & -\frac{1}{16}\frac{s+2}{(5s-2)} \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} 3 & 4-10s \\ -3 & 10s-4 \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} -6\sqrt{2} & \sqrt{2}(20s-8) \\ 12 & 16-40s \end{bmatrix}$
$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\frac{1}{12}\begin{bmatrix} -\frac{1}{2}-s & -\frac{1}{2}-s \\ \frac{1}{2}+s & \frac{1}{2}+s \end{bmatrix}$	$\frac{(1+2s)}{12}\begin{bmatrix} \sqrt{2} & \sqrt{2} \\ -2 & -2 \end{bmatrix}$

Table 3: Bior3.1/Lemarié low-pass filters and coefficients

low-pass dec.	$\tilde{H}(z) = -(4z)^{-1}(1+z)(1-4z+z^2)$	$(0,-1/4,3/4,3/4,-1/4)$
low-pas rec.	$H(z) = (8z)^{-1}(1+z)^3$	$(0,1/8,3/8,3/8,1/8)$
smoothed low-pass dec.	$H_+(z) = -(8z^2)^{-1}(1+z)^2(1-4z+z^2)$	$(-1/8,1/4,3/4,1/4,-1/8)$
roughened low-pass rec.	$H_-(z) = (4z)^{-1}(1+z)^2$	$(0,1/4,1/2,1/4,0)$

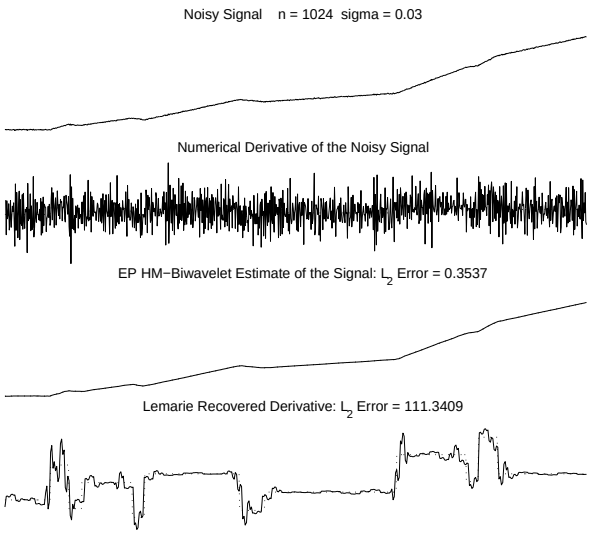

Figure 1. Recovering the derivative. The underlying derivative is the blocks signal. The noise is normal with standard deviation 0.03. HM/Lemarié biwavelets are used for the biwavelet EP denoising of the signal and the reconstruction of the derivative.

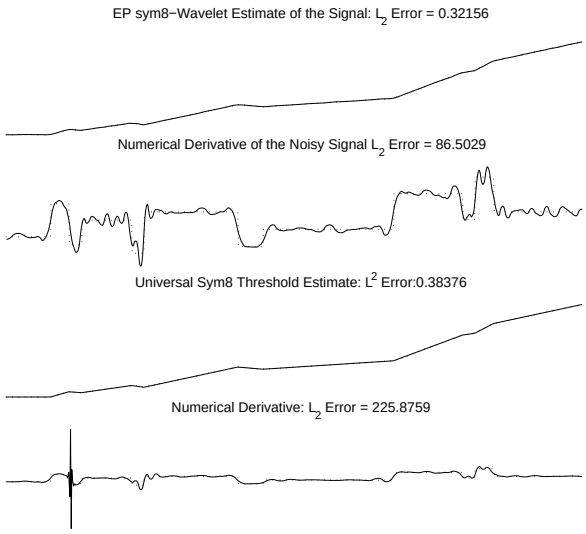

Figure 2. Same signal and white noise as in Figure 1. In the first two diagrams Symmlet 8 orthogonal wavelet EP denoising of the signal and its derivative recovered numerically. In the last two diagrams Symmlet 8 wavelets are used for denoising with a universal threshold procedure and the derivative is recovered numerically.

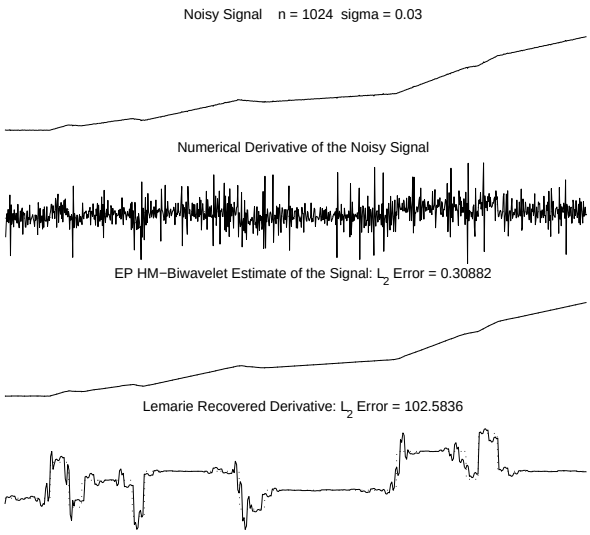

Figure 3. Recovering the derivative. The underlying derivative is the bumps signal. The noise is Tukey with parameter $\sigma = 0.03$. In the last two diagrams HM/Lemarié biwavelets are used for the biwavelet EP denoising and reconstruction of the derivative.

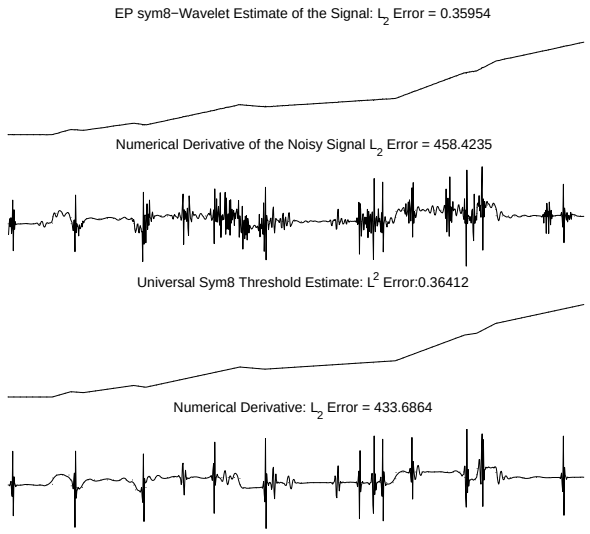

Figure 4. Same signal and Tukey noise as in Figure 3. Symmlet 8 wavelets is used for the wavelet EP denoising and reconstruction of the derivative. Symmlet 8 wavelets are used for denoising with a universal threshold procedure and the derivative is recovered numerically.

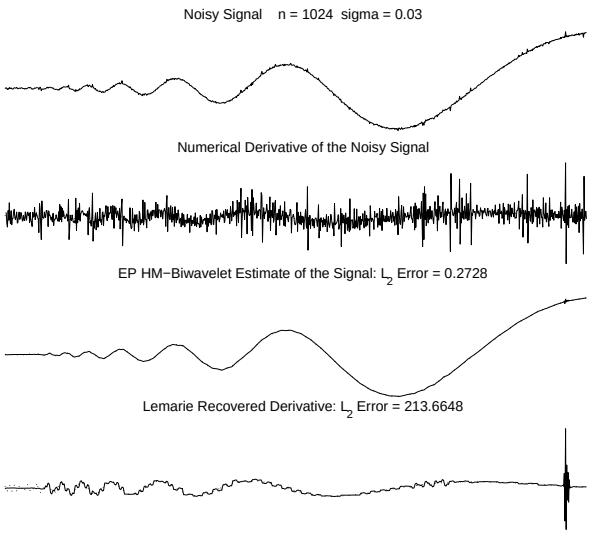

Figure 5. Recovering the derivative. The underlying derivative is the doppler signal. The noise is Tukey with parameter $\sigma = 0.03$. In the last two diagrams HM/Lemarié biwavelets are used for the biwavelet EP denoising and reconstruction of the derivative.

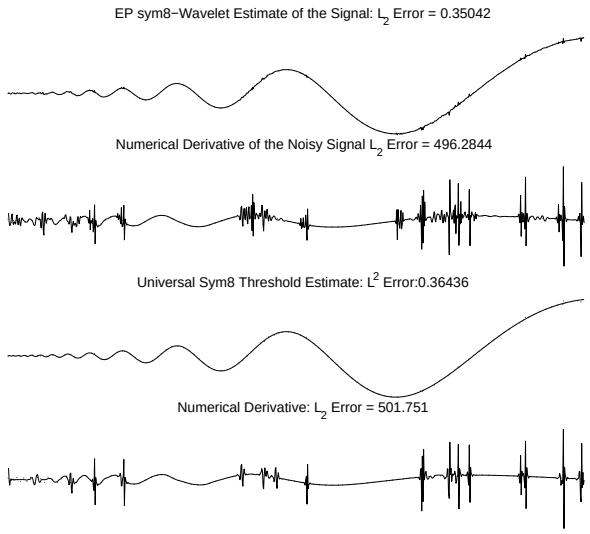

Figure 6. Same signal and Tukey noise as in Figure 5. Symmlet 8 wavelets is used for the wavelet EP denoising and reconstruction of the derivative. Symmlet 8 wavelets are used for denoising with a universal threshold procedure and the derivative is recovered numerically.