Conversion of Data in Sampled Form to Daubechies Inner Products

Robert Steacy

DMS-785-IR

August 1997

Conversion of Data in Sampled Form to Daubechies Inner Products

ROBERT STEACY
Department of Mathematics and Statistics
University of Victoria, Canada
rsteacy@Math.UVic.CA

1 Abstract

We present a method for converting sampled data of the form $f(n)$ into inner products $\langle f, \phi_{0,n} \rangle$. The method is user-friendly, easy to implement and computationally fast. It permits us to replace what Strang refers to as "a wavelet crime" [10] with a quite acceptable procedure. As a demonstration, we present some examples of denoising of a neurophysiological signal by various techniques, suggesting that conversion of sampled data to Daubechies inner products is well worth the small extra effort.

Keywords: Wavelets, Denoising, Signal Processing, Data Compression.

2 Introduction

In recent years, much attention has been devoted to wavelet analysis of one-dimensional signals such as sound recordings, electrocardiograms and electroencephalograms. Although the examples presented in this paper are neurophysiological, the technique is universally applicable. Often, the goal is to filter noise out of the signal, usually as a preliminary step to some other treatment. These techniques can be extended to two, three or more dimensions in a straightforward manner using an analogue of the multidimensional Fast Fourier Transform.

Although the emphasis in this article is denoising, it is hoped that those interested in data compression may incorporate the technique of operating on actual inner products, instead of the raw data vector, and find it to be valuable. The author has not investigated compression, as the emphasis in the medical signals has been on quality of result.

Those wishing to implement discrete wavelet transforms open a standard reference such as *Numerical Recipes in C* [7] to Section 13.10 and are presented with detailed instructions on how to calculate the discrete wavelet transform, henceforth referred to as the *DWT*, of a sampled data vector of the form $f(1), f(2), \dots f(n)$.

We shall use $n = 256$ and the first member of the Daubechies family of compactly supported wavelets (N = 2, known as Daub4), for purposes of illustration.

The output of the *DWT* algorithm is a 256-vector (v(1),...,v(256)) whose entries are the output of the *pyramid algorithm* of Mallat [6]. The algorithm and its inverse transform are accomplished by repeated matrix operations. See [7] for details.

An early, and still widely used, denoising technique is called linear denoising. One simply assumes that the data in $v(65) \dots v(256)$ is due to noise, set all 192 coefficients to zero, and reconstruct by the inverse transform. This is known colloquially as "deleting the top two scales". When applied to electrocardiograms, it results in valid features of the original signal being distorted or discarded entirely.

In Hess et al. [5] the conclusion is that among a wide variety of denoising techniques compared, the best is non-linear denoising with hard thresholding. This consists of setting to zero any of the $v(1)\ldots v(256)$ whose absolute value is below a pre-determined threshold, and reconstructing. This is the method which will be used throughout this paper. Hess et al. [5] recommend the second member of the Daubechies family of compactly supported wavelets (N = 3, known as Daub6) as the optimal Daubechies wavelet for this purpose. Later, we will directly compare Daub4 to Daub6, with and without conversion to inner products, with a view to which is most suitable for denoising the neurophysiological example.

3 "Wavelet Crime"

Now, what is the "wavelet crime" to which Strang [10] refers ? Wavelet analysis takes place in the context of a Multiresolution Analysis, henceforth referred to as an *MRA*, due to Mallat and Meyer [6]. This short discussion of *MRA*s is entirely borrowed from Daubechies [3]. It is actually from a discussion of the Haar wavelet, but it will serve quite well for illustration purposes.

A multiresolution analysis consists of a sequence of successive approximation spaces (V_j), $j \in \mathbb{Z}$, where $V_j = \{f \in \mathbb{L}^2(\mathbb{R})$; f piecewise constant on $[\ 2^j k,\ 2^j(k+1)\ )$, $k \in \mathbb{Z}\ \}$.

We assume a signal to be given by discrete (sampled) values at the finest scale possible; i.e., no finer scale information is available. For illustrative purposes, let us consider the finest scale to be V_0.

In our multiscale analysis, each of the subspaces V_j is spanned by $\phi_{j,\ell} = 2^{-\frac{j}{2}}\ \phi(2^{-j}x - \ell)$. Since $f \in V_0$,

$$f(x) = \sum_{n\in\mathbb{Z}} a_n^0\ \phi_{0,n} = \sum_{n\in\mathbb{Z}} a_n^0\ \phi(x-n)$$

We wish to construct a low-pass filter which is a projection $P_1 : V_0 \to V_1$. That is, a low-pass filter is a projection of the element of V_0 onto the next coarser scale, V_1. Because $P_1\ f \in V_1$ it can be represented by

$$P_1\ f = \sum_{\ell\in\mathbb{Z}} a_\ell^1\ \phi_{1,\ell} = \sum_{\ell\in\mathbb{Z}} a_\ell^1\ \frac{1}{\sqrt{2}}\phi(\frac{x}{2} - n)$$

So we have

$$P_1\ f = \sum_{\ell\in\mathbb{Z}} a_\ell^1\ \phi_{1,\ell} = \sum_{k\in\mathbb{Z}} a_k^0\ P_1\ \phi_{0,k}$$

by the linearity of projection.

We assume that we know the a_k^0; what we need is a way to determine the a_ℓ^1. The details of this construction are well known, and result in the pyramid algorithm.

In practice many find it more convenient to operate on the sampled values of $f(x)$ *as though they were* the coefficients a_k^0, instead of the coefficients a_k^0 themselves. By doing this, results of denoising and data compression algorithms are in terms of sampled values of $f(x)$. This is the "wavelet crime" to which Strang refers — using a technique which is explicitly developed to operate on inner products, and instead deciding to operate on the sampled values $f(1), \ldots, f(n)$ just as though they were inner products.

The rationale behind this has been primarily that it avoids what appears to be the messy task of converting data in sampled form to inner products, that it permits reconstruction to the limits of computing accuracy, and that it *appears* to work adequately well. We will now develop a simple technique for carrying out the conversion.

4 The Explicit Procedure

The explicit procedure for conversion of data in sampled form to inner products is given as Footnote 12 to Chapter 5 of Daubechies [3], which we reproduce here:

If f is given in "sampled" form, i.e., if we only know the $f(n)$, then the inner products $\langle f, \phi_{0,n}\rangle$ can be computed by a convolution (or filtering) operation, under the assumption that $f \in V_0$ to start with (components of f orthogonal to V_0 cannot be recovered). We have $f = \sum_k \langle f, \phi_{0,k}\rangle \phi_{0,k}$; hence $f(n) = \sum_k \langle f, \phi_{0,k}\rangle \phi(n-k)$. Consequently,

$$\sum_n f(n) e^{-in\xi} = \left(\sum_k \langle f, \phi_{0,k}\rangle e^{-ik\xi}\right) \cdot \left(\sum_m \phi(m) e^{-im\xi}\right) ,$$

i.e. the $\langle f, \phi_{0,k}\rangle$ are the Fourier coefficients of $\left(\sum_n f(n)e^{-in\xi}\right) \cdot \left(\sum_m \phi(m)e^{-im\xi}\right)^{-1}$.
It follows that $\langle f, \phi_{0,k}\rangle = \sum_n a_{k-n} f(n)$, where

$$a_m = (2\pi)^{-1} \int_0^{2\pi} d\xi \, e^{im\xi} \left(\sum_\ell \phi(\ell)\, e^{-i\ell\xi}\right)^{-1} .$$

So for any member of the Daubechies family of wavelets, the a_m can be found by explicit integration. For the Daub4 wavelet, Daubechies [3] gives $\phi(1) = \frac{1+\sqrt{3}}{2}$ and $\phi(2) = \frac{1-\sqrt{3}}{2}$. We now calculate the corresponding values of a_m for this wavelet.

$$\begin{aligned}
a_m &= \frac{1}{2\pi} \int_0^{2\pi} d\xi \, e^{im\xi} \left(\sum_\ell \phi(\ell)\, e^{-i\ell\xi}\right)^{-1} \\
&= \frac{1}{2\pi} \int_0^{2\pi} d\xi \, e^{im\xi} \left(\frac{1+\sqrt{3}}{2} e^{-i\xi} + \frac{1-\sqrt{3}}{2} e^{-2i\xi}\right)^{-1} \\
&= \frac{1}{2\pi} \int_0^{2\pi} d\xi \, e^{im\xi} \frac{\frac{1+\sqrt{3}}{2}\cos\xi + \frac{1-\sqrt{3}}{2}\cos 2\xi + i(\frac{1+\sqrt{3}}{2}\sin\xi + \frac{1-\sqrt{3}}{2}\sin 2\xi)}{2-\cos\xi}
\end{aligned}$$

At this point, if we replace $e^{im\xi}$ with $\cos m\xi + i \sin m\xi$ and separate into a real integral and a pure imaginary integral, we find that the pure imaginary integral is identically zero, and we are left with

$$= \frac{1}{2\pi}\int_0^{2\pi} d\xi \frac{(\frac{1+\sqrt{3}}{2}\cos\xi + \frac{1-\sqrt{3}}{2}\cos 2\xi)\cos m\xi - (\frac{1+\sqrt{3}}{2}\sin\xi + \frac{1-\sqrt{3}}{2}\sin 2\xi)\sin m\xi}{2-\cos\xi}$$

It is now easily established that for any integer value of $m \leq -2$, $a_m = 0$.

We shall explicitly calculate a_{-1} and then establish a procedure for determining a_m for $m \in \{0, 1, 2 \ldots\}$.

$$\begin{aligned}
a_{-1} &= \frac{1}{2\pi}\int_0^{2\pi} d\xi \frac{(\frac{1+\sqrt{3}}{2}\cos\xi + \frac{1-\sqrt{3}}{2}\cos 2\xi)\cos(-\xi) - (\frac{1+\sqrt{3}}{2}\sin\xi + \frac{1-\sqrt{3}}{2}\sin 2\xi)\sin(-\xi)}{2-\cos\xi} \\
&= \frac{1}{2\pi}\int_0^{2\pi} d\xi \frac{\frac{1+\sqrt{3}}{2} + \frac{1-\sqrt{3}}{2}\cos\xi\sin^2\xi + \frac{1-\sqrt{3}}{2}\cos^3\xi}{2-\cos\xi} \\
&= \frac{1}{2\pi}\int_0^{2\pi} d\xi \frac{\sqrt{3}-1}{2} + \frac{\frac{\sqrt{3}-3}{2}}{\cos\xi - 2}
\end{aligned}$$

Now $(cos\xi - 2)$ is never equal to 0, so there are no problems with our integrand being discontinuous, however the standard expression for the last integral possesses a discontinuity at $\xi = \pi$, so to avoid this we will change our limits of integration from 0 to 2π, and use $-\pi$ to π instead.

$$\begin{aligned}
&= \frac{\sqrt{3}-1}{2} - \frac{1}{2\pi}\frac{2}{\sqrt{3}}\left(\frac{\sqrt{3}-3}{2}\right)\arctan\left(\sqrt{3}\tan\frac{\xi}{2}\right)\Bigg|_{-\pi}^{\pi} \\
&= \sqrt{3}-1
\end{aligned}$$

Now we shall establish the following result, which will not only permit us to calculate the remaining a_m, but will provide a computationally rapid means of carrying out the actual conversion of sampled data into Daubechies inner products.

Theorem 1 *For the Daubechies wavelet, with any value of* $\mathrm{N} \geq 2$, *for all* a_k *with* $k \geq -1$, *we have* $\sum_{\ell=1}^{2\mathrm{N}-2} \phi(\ell)\, a_{(k+2\mathrm{N}-2-\ell)} = 0$.

Proof. There are of course only $2\mathrm{N}-2$ nonzero terms in the summation. For all $\mathrm{N} \geq 2$, for all a_k with $k \geq -1$,

$$\begin{aligned}
\sum_{\ell=1}^{2\mathrm{N}-2} \phi(\ell)\, a_{(k+2\mathrm{N}-2-\ell)} &= \frac{1}{2\pi}\int_0^{2\pi} d\xi \frac{\sum_{\ell=1}^{2\mathrm{N}-2}\phi(\ell)\, e^{i(k+2\mathrm{N}-2-\ell)\xi}}{\sum_\ell \phi(\ell)\, e^{-i\ell\xi}} \\
&= \frac{1}{2\pi}\int_0^{2\pi} d\xi\, e^{i(k+2\mathrm{N}-2)\xi} \\
&= \frac{1}{2\pi}\frac{e^{i(k+2\mathrm{N}-2)}}{i(k+2\mathrm{N}-2)}\Bigg|_0^{2\pi} \\
&= 0
\end{aligned}$$

This suggests the following computationally simple way to generate the matrix A which, when multiplied on the left by the column vector $(f(1), \ldots f(256))$, will give $(\langle f, \phi_{0,1}\rangle, \ldots \langle f, \phi_{0,256}\rangle)$ which we can call $(g(1), g(2), \ldots g(256))$.

Since the discrete wavelet transform as presented in Press et al. [7] is implemented by means of a wraparound matrix, we are already treating the signal as though it were periodic, and identifying its ends. Accordingly, it makes sense to continue this practice for the conversion from f to g, so we will use a wraparound matrix for the matrix A as well. This will mean using a_{-1} to a_{254} and not using a_{255} or a_{256}. From a computational point of view, a_{254} is approximately equal to 1.0421×10^{-146}, so most of the a_m are zero for all practical purposes.

Our theorem and the value of a_{-1}, together with the wraparound convention and the acceptance that any a_m beyond a_{254} is zero, permit the following Toeplitz matrix relation, where both A and B are 256×256:

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{254} & \ldots & \ldots & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_{-1} & a_{254} & \ldots & \ldots & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_{-1} & a_{254} & \ldots & \ldots & a_3 \\ \ldots & \ldots & \ldots & a_0 & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & a_0 & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & a_0 & \ldots & \ldots \\ a_{254} & \ldots & \ldots & \ldots & \ldots & a_1 & a_0 & a_{-1} \\ a_{-1} & a_{254} & \ldots & \ldots & \ldots & \ldots & a_1 & a_0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & \ldots & \ldots & \ldots & \ldots & \phi(2) & \phi(1) \\ \phi(1) & 0 & 0 & \ldots & \ldots & \ldots & \ldots & \phi(2) \\ \phi(2) & \phi(1) & 0 & \ldots & \ldots & \ldots & 0 & 0 \\ 0 & \phi(2) & \phi(1) & 0 & \ldots & \ldots & \ldots & 0 \\ \ldots & 0 & \phi(2) & \phi(1) & 0 & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \phi(2) & \phi(1) & 0 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \phi(2) & \phi(1) & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \phi(2) & \phi(1) & 0 \end{bmatrix}$$

(each row and column of B contain only two nonzero entries) and finally

$$AB = I$$

that is, A and B are inverses, within the limits of computing accuracy.

For those who prefer to use convolution methods instead of matrices, follow the directions in Daubechies' Footnote 12 above, with the values

$$a_m = (3\sqrt{3} - 5)(2 - \sqrt{3})^m \quad m \in \{-1, 0, 1, \ldots\} \tag{1}$$

So the procedure is as follows: express the data vector $(f(1), f(2), \ldots f(256))$ as a column vector f. The conversion of the data vector f to the Daubechies inner products is given by performing the matrix multiplication $Af = g$. Now g is given the customary treatments normally given to the data vector f. Finally, the reconstructed, denoised version of g, which we may call g^*, must be multiplied on the left by the matrix B to give f^*, the reconstructed version of f. So $Bg^* = f^*$.

5 Results

Fig. 1 shows the neurophysiological signal h. Fig. 2 shows a version of the signal f with normally distributed noise and signal-to-noise ratio of 30:1 ($\mu = 0$ and $\sigma = 0.00225$).

Using the same wavelet as in Hess et al. [5], N = 3, better known as Daub6, Figure 3 shows the denoised version obtained by operating directly on the raw data vector as though it were the inner products. Figure 4 shows the result of first converting to inner products using matrix A, denoising, and converting back using matrix B. In order to keep all comparisons as "apples to apples" as possible, the same denoising threshold of 0.007 was used for the Daub4 examples as well, even though some other value may be considered optimal.

As in Hess et al. [5], we use the following definitions of $\ell_p - norm$ and entropy-norm:

$$\ell_p - norm: \quad ||h - f||_p = \left(\sum_i |h(i) - f(i)|^p\right)^{\frac{1}{p}}, \quad p = 1, 2 \tag{2}$$

$$entropy - norm: \quad -\sum_i \frac{|h(i) - f(i)|^2}{||h - f||_2^2} \cdot \log \frac{|h(i) - f(i)|^2}{||h - f||_2^2} \tag{3}$$

Careful examination of Fig. 3 and 4 shows that there are fewer *artifacts*, i.e. visually serious features which are not present in the original signal, when operating on inner products. Comparing the ℓ_1 and ℓ_2 errors between the original signal and the two denoised versions with Gaussian noise, we find there is no significant difference. In both cases the ℓ_1 result is 0.6935, while the ℓ_2 figure is 0.0334 for Fig. 3 and 0.0379 for Fig. 4. While the ℓ_2 figure for inner products appears significantly worse, Hess et al. [5] recorded a standard deviation on 30 samples of 0.00194, so a 95% confidence interval would be about 0.008 wide. What may be significant is that there appear to be fewer visually objectionable artifacts, of the type which might interfere with diagnosis, either by machine or by human practitioner.

Now a bit of explanation as to why the Daub4 wavelet is highly suitable for electrocardiograms and electroencephalograms, and why it performs poorly on some other signals. The following is from Daubechies [3], and is taken from a discussion of the continuous wavelet transform, which is the predecessor of the discrete wavelet transform:

Definition 1 *Condition 3 (also known as Condition A — Approximation)*
For $p \in \mathbb{N}$

$$\sum_{n \in \mathbb{Z}} (-1)^n \, n^m \, \alpha_n \; = \; 0 \;\; for \; m = 0, 1, \ldots, p-1 \tag{4}$$

This condition guarantees that the polynomials $1,\ x,\ x^2,\ \ldots,\ x^{p-1}$ *are linear combinations of the translates* $\phi(x-n)$. *The first p moments of the wavelet* $\psi(x)$ *associated with* ϕ *are then zero:*

$$\int_{-\infty}^{\infty} x^m \, \psi(x) \, dx \; = \; 0 \quad m \; = \; 0, 1, \ldots, p-1 \tag{5}$$

As briefly as possible. this means that if Daub4 (N = 2, i.e. $p = 2$) is applied to a quadratic curve, there will be some loss of data if the top scale, i.e. the coefficients $(v(129), \ldots, v(256))$ are set to zero. Using Daub6, within the limits of computing accuracy, there will not be (except in the wraparound region, where the curve is no longer a quadratic). Consequently, when we use Daub6 on data in which the features of interest are not smooth quadratic curves, we are imposing a structure on it which is not present, and this is responsible for much of the artifacts. If a wavelet of very long support, such as Daub20, is used on a signal which is essentially only piecewise linear, the resulting denoised version has severe artifacts, sinusoidal in appearance, as the treatment attempts to impose a high-order polynomial structure which is simply not present. Similarly, if Daub4 is used on a signal which is highly sinusoidal, one which would be perfect for Fourier analysis, the result is highly unsatisfactory. So, since the features of interest of the electroencephalogram and the electrocardiogram are essentially piecewise linear, the author has found that Daub4 works best for them. "Best" meaning visually; from an objective point of view such as ℓ_1 or ℓ_2 error there is no significant improvement.

Essentially, a good rule of thumb for deciding whether to use Fourier analysis, or one of the wide variety of wavelets (including the Haar wavelet), is that the more the elements of the intended method *look like* the data, generally speaking, the more likely it is to be suitable.

So now let us examine Fig. 5, which is similar to Fig. 3, except that Daub4 has been used on the raw data vector instead of Daub6, and Fig. 6, which is similar to Fig. 4, only Daub4 has been used on inner products instead of Daub6. Again, the inner product version is visually better in terms of artifacts, and each of them is an improvement over the respective Daub6 versions. The ℓ_1 and ℓ_2 errors are 0.6678 and 0.0328, and 0.6796 and 0.0359 for Fig. 5 and Fig. 6 respectively. That is, ~~there is no significant difference.~~

The measure of the error which does capture what we are seeing visually is the entropy-norm. The entropy-norms of the errors for Fig. 3 and Fig. 5 are, respectively, 5.3248 and 5.2348. These are for Daub6 and Daub4 on the raw data vector. Moving to Daub6 on inner products, Fig. 4 yields an error with entropy-norm of 5.1755. Finally, using Daub4 on inner products, Fig. 6 yields an error with entropy-norm of 4.8136.

Comparing Fig. 3 with Fig. 6 is the most visually striking comparison. The combined effect of using a wavelet of suitably short support, together with operating on the inner products instead of the raw data vector, is most promising.

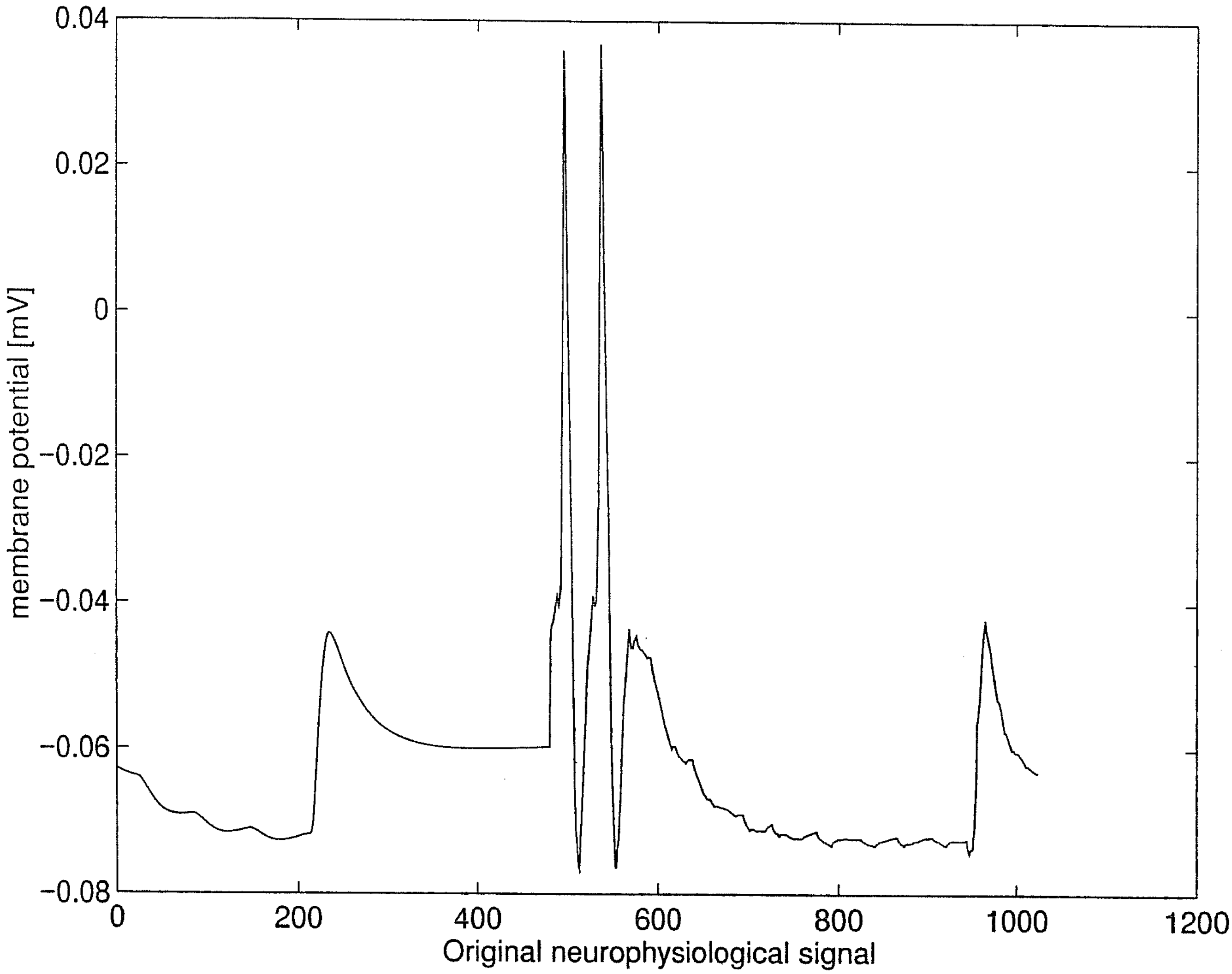

Figure 1: This figure first appeared in Hess et al.

6 Acknowledgements

I would like to thank Markus Kraft and his co-authors F. Hess, M. Richter, and H. Bockhorn for their permission to use Fig. 1 and 2. I would like to thank my academic supervisor, Reinhard Illner, for his support and encouragement of this project. I cannot forget my office-mates, Sean Bohun, Rob Bures, David Feldman and Eugene Neufeld for their many valuable suggestions. Lastly, I would like to thank Elaine Cumming, Betty Page and Georgina Smith of the Department of Mathematics and Statistics office for their technical support in producing this and other articles and presentations.

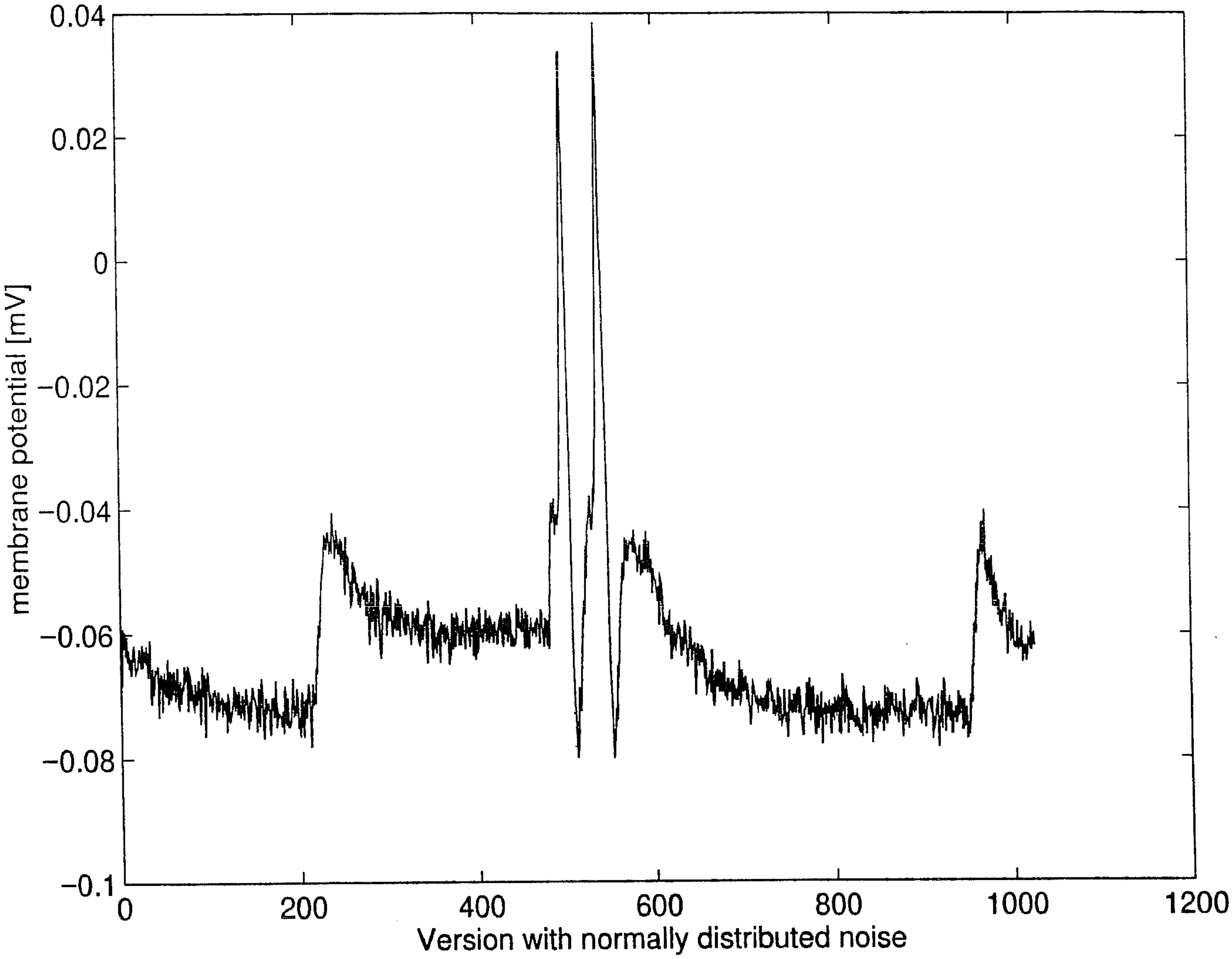

Figure 2: Noise is normally distributed with $\mu = 0$ and $\sigma = 0.00225$.

7 Appendix: Matlab Code

Here is a sample of code which could be used to generate the matrices A and B, written in C language to run in a Matlab environment. First write the Matlab file "t.m" as follows:

```
function trans = t()
B = [zeros(size(256:256))];
temp(1:2) = [(1 - sqrt(3))/2 (1 + sqrt(3))/2];
for i = 3:256
B(i,i-2:i-1) = temp(1:2);
end
B(1,255:256) = temp(1:2);
B(2,1) = temp(2);
```

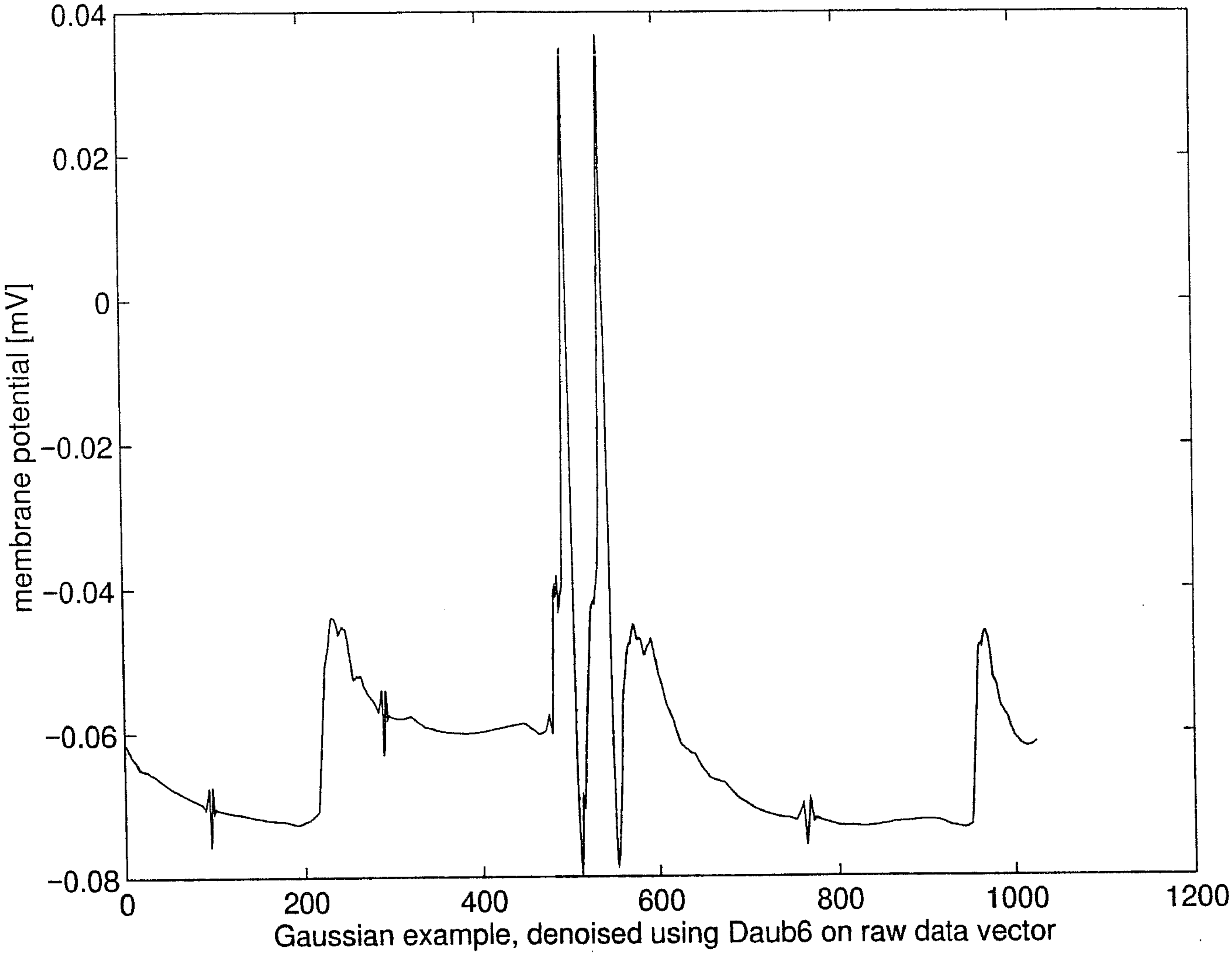

Figure 3:

```
B(2,256) = temp(1);
trans = B;
```

Then, at the start of the Matlab session, the commands

```
B = t;
A = inv(B);
```

produce the matrices A and B.

Now let us proceed to de-noise the version of the neurophysiological signal which has been treated with Gaussian noise at a signal to noise ratio of 30:1. It is stored in a Matlab file called "neuron.gauss.asc". Since it is a vector of length 1024, the program has been modified to that length.

```
fd = fopen('neuron.gauss.asc');
f = fscanf(fd,'%f');
```

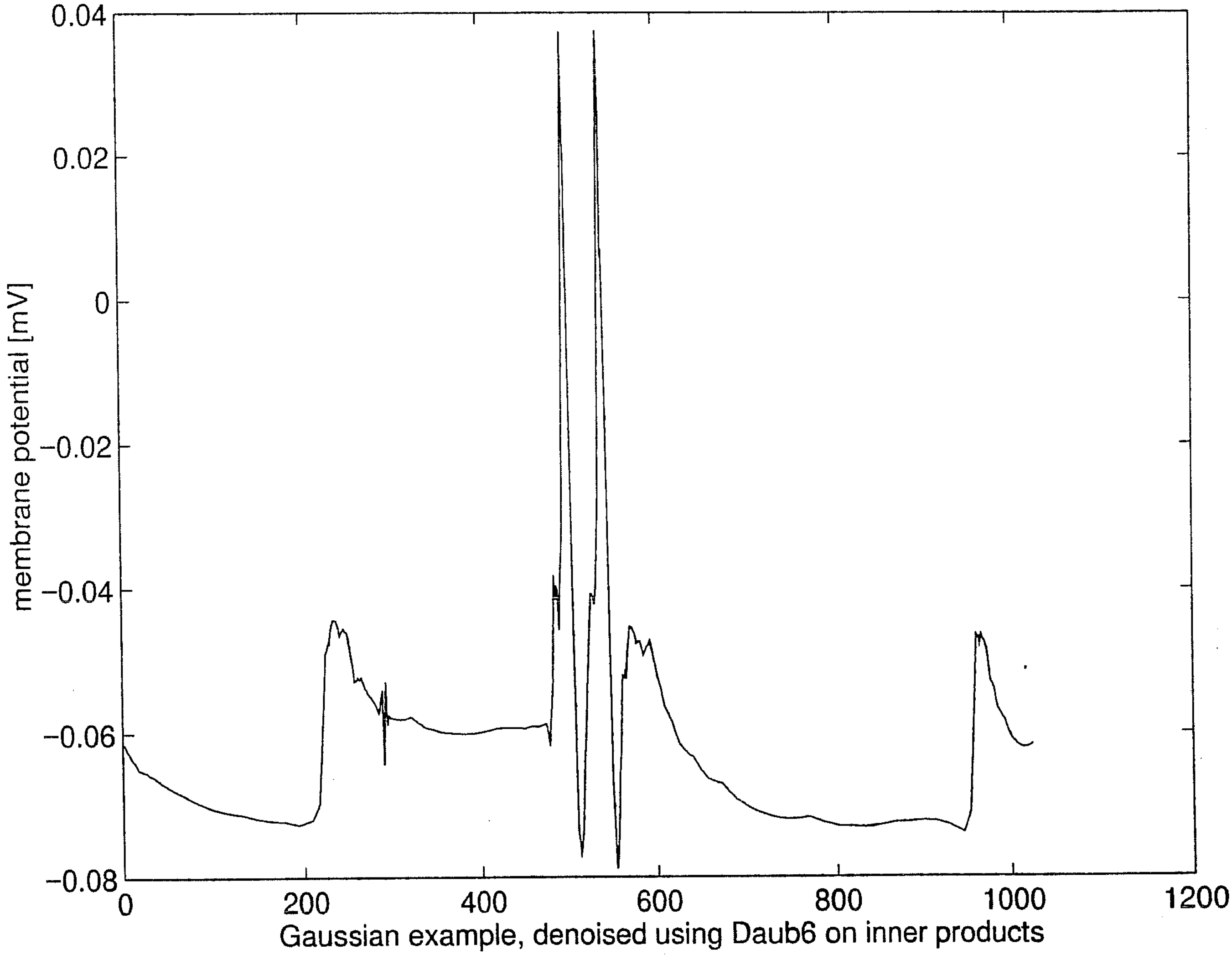

Figure 4:

```
fclose(fd);
g = A * f';
t1 = wavel(g, daub4);
%% comment:  ``wavel'' is the pyramid algorithm using the Daub4
discrete wavelet transform.  The four coefficients are stored
in a Matlab file called ``daub4.m''.
test = (abs(t1) >= .007);
%% comment:  ``test'' is a truth-valued Matlab variable.
t2 = t1.*test;
%% comment:  Matlab doesn't mind if you want to multiply real-valued
data by a truth-valued variable.  t2 is t1 with small coefficients
```

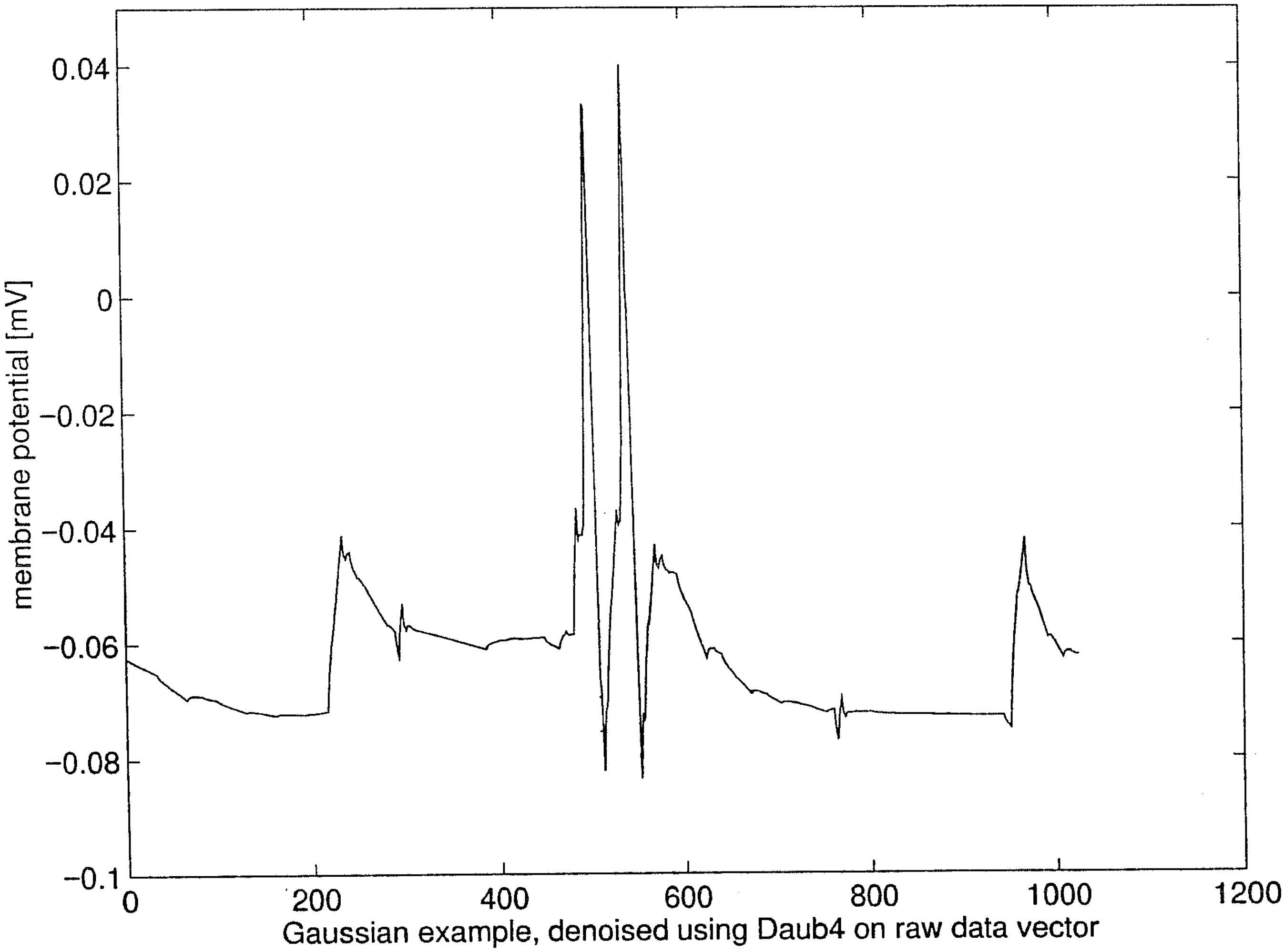

Figure 5:

set to zero. `` .* '' means pointwise multiplication of vectors, i.e. $(u_1, u_2).*(v_1, v_2) = (u_1v_1, u_2v_2)$.

```
t3 = invwavel(t2,daub4);
t4 = B*t3;
%% comment:  t4 is the de-noised version of the noisy signal f.
```

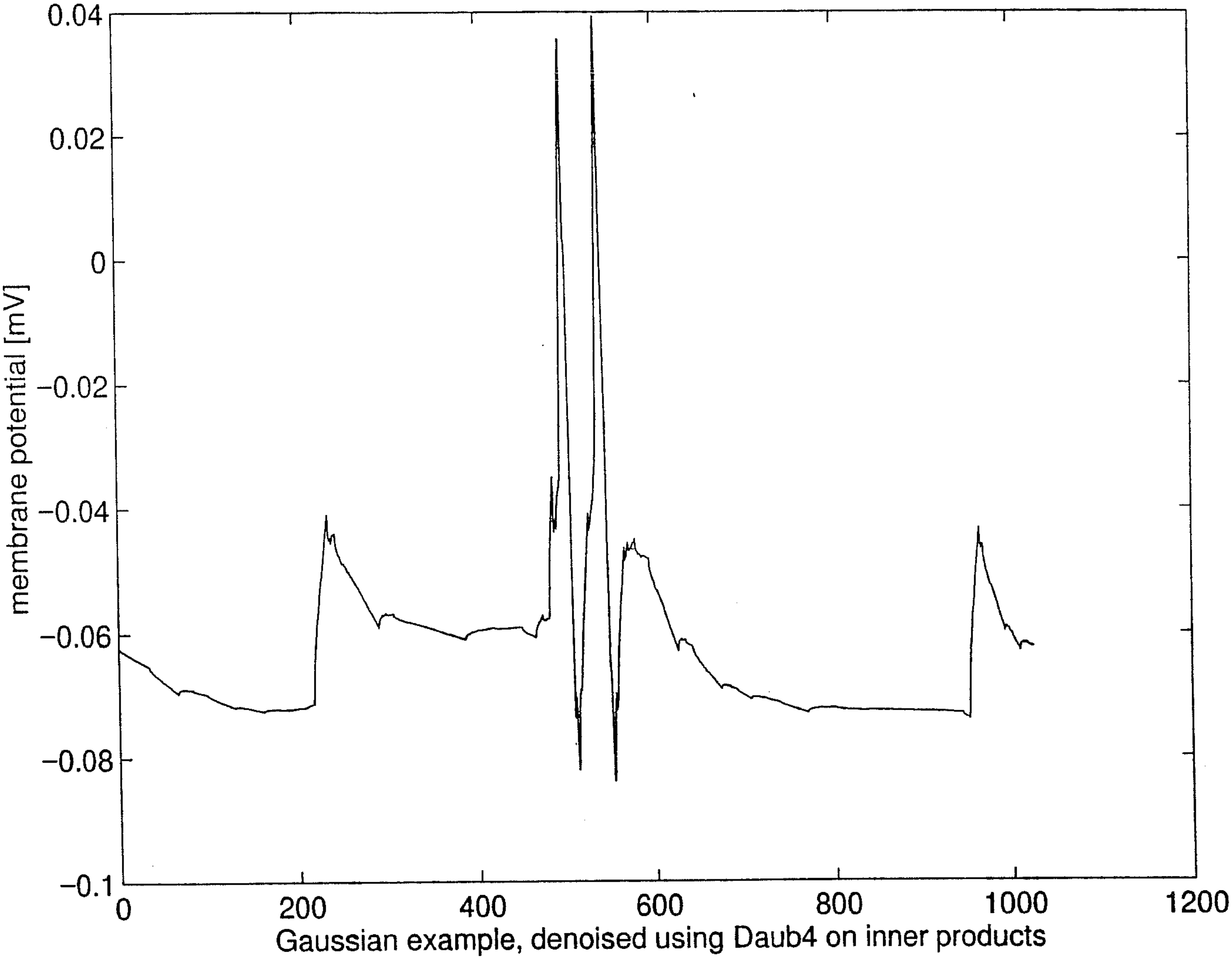

Figure 6:

8 References

References

[1] Coifman, Ronald R. & Wickerhauser, Mladen V. (1990). *Best—adapted wave packet bases*, preprint, Yale University, New Haven.

[2] Coifman, Ronald R. & Wickerhauser, Mladen V. (1992). *Entropy—based Algorithms for Best Basis Selection*. IEEE Transactions on Information Theory, **38**, 2, 713-718.

[3] Daubechies, Ingrid (1992). *Ten Lectures on Wavelets*, Philadelphia: Society for Industrial and Applied Mathematics.

[4] Daubechies, Ingrid (1988). *Orthonormal Bases of Compactly Supported Wavelets.* Communications on Pure and Applied Mathematics, **XLI**, 909-996.

[5] Hess, F., Kraft, M., Richter, M., & Bockhorn, H. (1997). *Comparison and Assessment of Various Wavelet and Wavelet Packet Based Denoising Algorithms for Noisy Data.* Progress in Industrial Mathematics at ECMI 96, edited by M. Brons, John Wiley & Sons and B.G. Teubner.

[6] Mallat, S. (1989). *A theory of multiresolution signal decomposition: the wavelet representation.* IEEE Trans. PAMI, **11**, 674-693.

[7] Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. (1992). *Numerical Recipes in C.* Cambridge University Press.

[8] Ramirez, R.W. (1985). *The FFT, fundamentals and concepts.* Englewood Cliffs, N.J.: Prentice—Hall, Inc..

[9] Scheidt, Stephen, MD. *Basic Electrocardiography* Ciba—Geigy Pharmaceuticals internal publication.

[10] Strang, G. & Nguyen, T. (1996). *Wavelets and Filter Banks.* Wellesley: Wellesley—Cambridge Press.

[11] Strang, G. (1989). *Wavelets and Dilation Equations: a Brief Introduction.* SIAM Review, **31**, 4, 614-627.

[12] Taswell, Carl (1995). *Satisficing Search Algorithms for Selecting Near—Best Bases in Adaptive Tree—Structured Wavelet Transforms*, IEEE Transactions on Signal Processing preprint.