5-14-1993

# Applications of the Wavelet Transform to Signal Analysis

Jie Chen '93
*Illinois Wesleyan University*

# Applications of the Wavelet Transform to Signal Analysis

Jie Chen [1]
Advisors: Dr. Tian-Xiao He and Dr. Melvyn W. Jeter
Illinois Wesleyan University

May 14, 1993

This report is for the Senior Research Honors.

## Abstract

Like the Fourier Transform, the Wavelet Transform decomposes signals as a superposition of simple units from which the original signals can be reconstructed. The Fourier Transform decomposes signals into sine and cosine functions of different frequencies, while the Wavelet Transform decomposes signals into wavelets. Since the Fourier Transform is a global integration transform and there is no time factor in it, it cannot effectively analyze nonstationary signals whose statistical properties change with time. In order to analyze nonstationary signals, we need to decompose signals into units that are localized in both the time and frequency domains. Using the Wavelet Transform with the B-wavelet, we wrote a program package in Mathematica to implement the decomposition and reconstruction algorithms for signal processing. A data acquisition system developed in another project is used to acquire both the synthesized signals and real voice signals. Application of the Wavelet Transform on these signals will be presented.

[1]This research is supported by a Grant-in-Aid of Research from Sigma Xi, The Scientific Research Society

# 1 Introduction

The Fourier Transform is widely used in science and engineering to analyze and process signals. It is a global integral transformation of the form:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt, \tag{1}$$

which decomposes the original signal into sine and cosine signal units of different frequencies. These units of signal are easier to analyze and process than the original complex signal. The Fourier Transform has been a powerful tool for scientists and engineers with the technique of Fast Fourier Transform (FFT.) But it is most effective in analyzing narrow-band stationary signals because of its global integration on the time axis. In order to analyze wide-band non-stationary signals, the windowed Short Time Fourier Transform (STFT) is used.

$$STFT(\tau, f) = \int_{-\infty}^{+\infty} x(t)g^*(t-\tau)e^{-j2\pi ft}dt, \tag{2}$$

where $g^*(t-\tau)$ is the complex conjugate of the shifted version of $g(t)$, a window function whose length is preset. With the window function, we can either get high time resolution and poor frequency resolution by setting a small window length, or high frequency resolution and poor time resolution by setting a large window length, but not both. Scientists have developed various techniques to overcome this difficulty, such as the use of the ambiguity function, filter banks, pyramidal decomposition, multiresolution analysis; etc. Recent studies [1, 2, 3] have found that all these different techniques can be unified under the wavelet theory. The basic Wavelet Transform has the following form:

$$W_x(a,b) = \frac{1}{\sqrt{a}} \int \psi\left(\frac{t-b}{a}\right) x(t)dt, \tag{3}$$

where $\psi(t)$ is a mother wavelet function. It acts as a window function to localize the integration. Notice that $\psi\left(\frac{t-b}{a}\right)$ is a dilated and shifted version of the mother wavelet function; $a$ is the dilation factor and $b$ is the translation

factor. In the Wavelet Transform, a one dimensional signal $x(t)$ is mapped to a two dimensional function $W_x(a, b)$.

Both STFT and WT map a one dimensional signal to a two dimensional function, and both of them are integral transforms, but they are significantly different. STFT is a time-frequency transformation. Because its window size is fixed, its frequency resolution is fixed across the whole spectrum. While a 10 Hz resolution is suitable at a high frequency of 40 KHz, it is unacceptable at a low frequency of 20 Hz. In order to extract different information from the original signal, one needs to run STFT several times with different window sizes. We use a microscope to examine details and a telescope to see landscape. In this sense, the Wavelet Transform is a microscope and telescope in one; that is, it is adaptive to the different frequency components of the signal. Since the window function $\psi\left(\frac{x-b}{a}\right)$ is a dilated and shifted version of the same mother wavelet function, it has *constant relative resolution* across all scales. It extracts information of all scales by doing one transformation. The local, detail characteristics of the signal are preserved in the small scale part of the transformation coefficients, and the global characteristics of the signal are stored in the large scale part of the transformation coefficients. Because of the adaptive window size, it is possible to get all the relevant information of all scales in one transformation.

Another way to look at the differences between STFT and WT is by using the correlation (or ambiguity) functions. In cross-correlation analysis, we define:

$$C(\tau) = \int_{-\infty}^{+\infty} f(t)g(t-\tau)dt \tag{4}$$

The value $C(\tau)$ is a measure of the similarity of the two functions: $f(t)$ and a shifted version of $g(t)$. A large value of $C(\tau)$ at $\tau$ means that $g(t)$ shifted by $\tau$ is a good approximation to $f(t)$. If $f(t)$ and $g(t)$ are inherently unsimilar, no matter what $\tau$ value is chosen, $g(t)$ will always be a "bad" approximation. In the Fourier Transform, similarities are evaluated when the original signal is compared to sine and cosine functions of different frequencies. Sine and cosine functions are global functions. If the signal has the same property, e.g., if the signal is a narrow-band stationary signal, the Fourier Transform is an efficient representation. But if the original signal contains transient components, the Fourier Transform fails to represent these local events, since the local information is spread out. Using the Wavelet Transform, the local

transient component can be matched by the small scale version of the mother wavelet, and the global component of the signal can be matched by the large scale version of the mother wavelet; thus a good similarity can be expected.

In this report, we summarize the theoretical foundations of the Discrete Wavelet Transformation with B-wavelet and apply them to solve real world problems. Some of the questions we need to discuss are:

- What function can be considered as a mother wavelet?
- What kind of signals can be analyzed by the Wavelet Transform?
- Is the Wavelet Transform unique?
- How does one calculate a Wavelet Transform?
- Is there an Inverse Wavelet Transform? And how to calculate it?

We will present some results of Daubechies [4], Mallat [5], Chan and Chui [6], and He [7] in the next section, followed by an implementation of the algorithm in a Mathematica Package. In Section 3, we discuss how to program the algorithm in Mathematica and Section 4 gives an application of the Discrete Wavelet Transform with B-wavelets on analysis of synthesized signals.

# 2 Theoretical Foundations

## 2.1 Wavelet and Multiresolution Analysis

The Wavelet Transform introduced in Section 1 is also called the Continuous Wavelet Transform (**CWT**). Another approach is the Discrete Wavelet Transform (**DWT**), where the transformation is calculated only over a discrete grid of $a$ and $b$. A convenient way to make the dilation factor $a$ of the transformation discrete is to consider $a = 2^{-j}$ where $j \in Z$, and the translation factor is adjusted accordingly, $b = 2^{-j}k$ where $j, k \in Z$. Thus

$$DWT_x(j,k) = \sqrt{2^j} \int \psi(2^j t - k)x(t)dt = \sqrt{2^j} \int \psi_{j,k}(t)x(t)dt, \qquad (5)$$

where

$$\psi(2^j t - k) = \psi_{j,k}(t)$$

This DWT definition leads to a multiresolution analysis of the signal. Let us consider signals in $L^2(\Re)$ space (i.e. finite energy signals.) First we use functions in a well understood subspace $V_0$ to approximate it. Then we construct a sequence of subspaces $(V_{2^j})_{j\in Z}$ with the following properties:

$$V_{2^j} \subset V_{2^{j+1}}, \forall j \in Z \tag{6}$$

$$x(t) \in V_{2^j} \Longleftrightarrow x(2t) \in V_{2^{j+1}}, \forall j \in Z \tag{7}$$

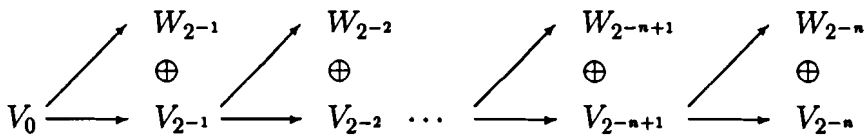
$$\overline{\bigcup_{j=-\infty}^{+\infty} V_{2^j}} = L^2(\Re) \tag{8}$$

$$\bigcap_{j=-\infty}^{+\infty} V_{2^j} = \phi \tag{9}$$

Since this sequence of spaces $(V_{2^j})_{j\in Z}$ are the coarser version of the previous one, the orthogonal complement space of $V_{2^j}$ in $V_{2^{j+1}}$ can be characterized by the $W_{2^j}$ space:

$$W_{2^j} \oplus V_{2^j} = V_{2^{j+1}}, \text{ and } W_{2^k} \perp W_{2^l}, \ \forall k \neq l. \tag{10}$$

Thus the first approximation functional space $V_0$ can be decomposed as the following:




To find a set of bases of these functional spaces is the first step in multiresolution analysis of signals. The following theorems prove that under certain conditions there exists a *scaling function* $\phi(t)$ and a *mother wavelet function* $\psi(t)$, whose dilations and translations form the bases of the functional spaces $(V_{2^j})_{j\in Z}$ and $(W_{2^j})_{j\in Z}$ respectively. They also show how the scaling function and the mother wavelet function can be constructed.

**Theorem 1 (Existence of the Scaling Function)**

*Let $(V_{2^j})_{j\in Z}$ be a multiresolution approximation of $L^2(\Re)$. There exists a unique function $\phi(t) \in L^2(\Re)$, called a scaling function, such that if we set $\phi_{j,k}(t) = \phi(2^j t - k)$, for $(j,k) \in Z^2$, then:*

$$\left(\sqrt{2^j}\phi_{j,k}(t)\right)_{k\in Z} \text{ is an orthonormal basis of } V_{2^j}.$$

**Theorem 2 (Construction of a Scaling Function)**

*Let $\phi(t)$ be a scaling function, and let $H$ be a discrete filter with impulse response $h(n)=< \phi_{-1,0}(u), \phi_{0,n}(u) >$ , let $H(\omega)$ be the Fourier Transform defined by:*

$$H(\omega) = \sum_{n=-\infty}^{+\infty} h(n)e^{-jn\omega}$$

*$H(\omega)$ satisfies the following two properties:*

1. *$H(0)=1$ and $h(n)=O(n^{-2})$ at infinity;*
2. *$| H(\omega) |^2 + | H(\omega+\pi) |^2 = 1$*

*Conversely, let $H(\omega)$ be a Fourier Transform satisfying (1) and (2), and such that $| H(\omega) | \neq 0$ for $\omega \in [0, \frac{\pi}{2}]$.*

*The function defined by*

$$\hat{\phi}(\omega) = \prod_{p=1}^{+\infty} H(2^{-p}\omega)$$

*is the Fourier Transform of a scaling function.*

**Theorem 3 (Existence and Construction of a Wavelet Function)**

*Let $\psi(t)$ be a function whose Fourier Transform is given by:*

$$\hat{\psi}(\omega) = G(\frac{\omega}{2})\hat{\phi}(\frac{\omega}{2})$$

*with $G(\omega)=e^{-j\omega}H^*(\omega+\pi)$, Then:*

$$\left(\sqrt{2^j}\psi_{j,k}(t)\right)_{k\in Z} \text{ is an orthonormal basis of } W_{2^j}$$

$$\left(\sqrt{2^j}\psi_{j,k}(t)\right)_{(j,k)\in Z^2} \text{ is an orthonormal basis of } L^2(\Re)$$

Based on the above theorems, if the original approximation space $V_0$ can be multiresolution decomposed, we can find a scaling function and a mother wavelet function whose dilation and translation form the bases of the $(V_{2^j})_{j\in Z}$ and $(W_{2^j})_{j\in Z}$ respectively. Thus we can approximate the original signal with a linear combination of these bases.

**A. Approximate a signal at a resolution $2^j$**

Let

$$A_{2^j}x = 2^{-j}\sum_{n=-\infty}^{+\infty} < x(u), \phi_{j,n}(u) > \phi_{j,n}(t), \tag{11}$$

where

$$A^d_{2^j}(x,n) =< x(u), \phi_{j,n}(u) >, n \in Z$$

are called *Discrete approximation coefficients.*

Also, we can express the signal components in $W_{2^j}$ space using the bases in $W_{2^j}$ as the following:

$$D_{2^j}x = 2^{-j}\sum_{n=-\infty}^{+\infty} < x(u), \psi_{j,n}(u) > \psi_{j,n}(t), \tag{12}$$

where

$$D^d_{2^j}(x,n) =< x(u), \psi_{j,n}(u) >, n \in Z$$

are called *Discrete detail signal coefficients.*

**B. Iterative algorithms for calculating discrete coefficients**

**Decomposition**

$$A^d_{2^j}(x,n) = \sum_{k=-\infty}^{+\infty} \tilde{h}(2n-k)A^d_{2^{j+1}}(x,k), \tag{13}$$

$$D^d_{2^j}(x,n) = \sum_{k=-\infty}^{+\infty} \tilde{g}(2n-k)A^d_{2^{j+1}}(x,k), \tag{14}$$

where

$$h(n) =< \phi_{-1,0}(u), \phi_{0,n}(u) >, \tilde{h}(n) = h(-n),$$
$$g(n) =< \psi_{-1,0}(u), \psi_{0,n}(u) >, \tilde{g}(n) = g(-n).$$

See **Appendix A** for detailed deduction of the formula.

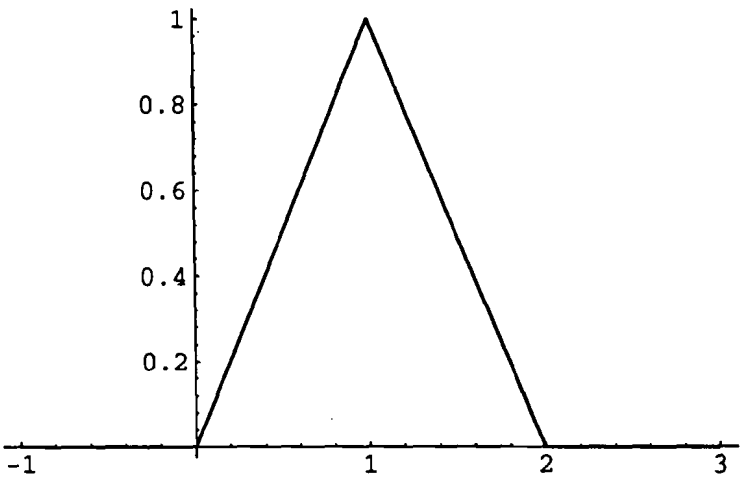



Figure 1: Second order B-spline function $\phi(x)$.

**Reconstruction**

$$A^d_{2j+1}(x,n) = 2\sum_{k=-\infty}^{+\infty} h(n-2k)A^d_{2j}(x,k) + 2\sum_{k=-\infty}^{+\infty} g(n-2k)D^d_{2j}(x,k). \tag{15}$$

## 2.2 B-Spline and B-Wavelet

We will now apply the basic theories in Section 2.1 to a special set of functions called B-spline functions. The second order B-spline function (see Figure 1) is defined as:

$$\phi(t) = \begin{cases} t & 0 \le t \le 1; \\ 2-t & 1 \le t \le 2; \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

And the higher order B-spline is the convolution from the lower order B-spline function.

According to their definitions, B-spline functions have the following property:

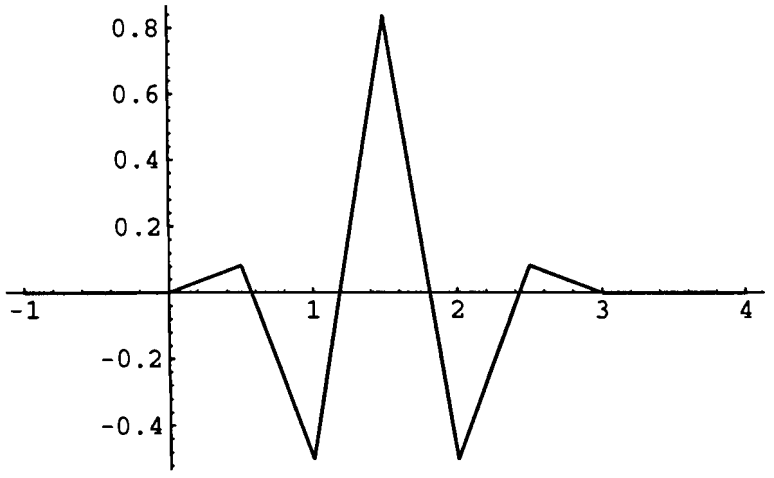



Figure 2: Second order B-wavelet function $\psi(x)$.

$$\phi(t) = \sum_{n=-\infty}^{+\infty} p_n^m \phi(2t - n) \tag{17}$$

where m is the order of the B-spline function (in this report m=2), and $p_n^m$ is defined as the following:

$$p_n^m = \begin{cases} 2^{-(m+1)} \begin{pmatrix} m \\ n \end{pmatrix} & 0 \le n \le m; \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

This property means that the functional spaces spanned by the B-spline functions satisfy the multiresolution decomposition requirements of $(V_{2^j})_{j \in Z}$. So we can use a B-spline function as a scaling function. The corresponding B-wavelet function (see Figure 2) has the form:

$$\psi(t) = \sum_{n=-\infty}^{+\infty} q_n^m \phi(2t - n); \tag{19}$$

where

$$q_n^m = \begin{cases} \frac{(-1)^n}{2^{m-1}} \sum_{j=0}^{m} \begin{pmatrix} m \\ j \end{pmatrix} \phi_{2m}(n - j + 1) & 0 \le n \le 3m - 2; \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

When using B-wavelet, although they form the bases of the space and they are orthogonal across different resolution levels, they are not orthogonal in the same level. While we use B-wavelet to decompose the signal and get the decomposition coefficients, we need to introduce another function called *dual B-wavelet* to reconstruct the original signal from these wavelet coefficients. The dual B-wavelet $\tilde{\psi}(t)$ is defined by the following:

$$\int_{-\infty}^{+\infty} \psi(t-j)\tilde{\psi}(t-l)dt = \delta_{jl}. \tag{21}$$

Hence

$$x(t) = \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} d_{j,k}\psi_{j,k}(t), \tag{22}$$

where

$$d_{j,k} = 2^j < f, \tilde{\psi}_{j,k} > .$$

The approximation and decomposition formula using B-wavelet are the following:

$$\begin{aligned} x(t) &\approx x_{2^0}(t) \\ &= g_{2^{-1}}(t) + x_{2^{-1}}(t) \\ &= g_{2^{-1}}(t) + g_{2^{-2}}(t) + x_{2^{-2}}(t) \\ &= \cdots, \end{aligned} \tag{23}$$

where

$$\begin{cases} x_{2^{-j}}(t) = \sum\limits_{k=-\infty}^{+\infty} c_{j,k}\phi_{j,k}(t), \\ g_{2^{-j}}(t) = \sum\limits_{k=-\infty}^{+\infty} d_{j,k}\psi_{j,k}(t). \end{cases} \tag{24}$$

And the coefficients have the following relationships:

$$\begin{cases} c_{j,k} = \sum\limits_{n=-\infty}^{+\infty} a_{n-2k}c_{j+1,n}, \\ d_{j,k} = \sum\limits_{n=-\infty}^{+\infty} b_{n-2k}d_{j+1,n}, \end{cases} \tag{25}$$

where $\{a_n\}$ and $\{b_n\}$ are computed from:

$$\phi(2x-l) = \sum_{n=-\infty}^{+\infty} a_{l-2n}\phi(x-n) + \sum_{n=-\infty}^{+\infty} b_{l-2n}\psi(x-n). \tag{26}$$

The reconstruction formula uses the $\{p_n^m\}$ and $\{q_n^m\}$ sequences defined above and the formula is the following:

$$c_{j+1,k}^* = \sum_{n=-\infty}^{+\infty} (p_{k-2n}^m c_{j,n}^* + q_{k-2n}^m d_{j,n}^*) \tag{27}$$

Both $\{a_n\}$ and $\{b_n\}$ are infinite sequences and their computation requires much efforts. But these filter sequences need to be computed only once. They depend only upon the selection of the scaling function and do not depend upon the signals we need to analyze. Also, $\{a_n\}$ and $\{b_n\}$ decay quite fast and we can chop off to get finite sequences.

# 3 Programming in Mathematica

## 3.1 Advantages of Programming in Mathematica

Mathematica is a sophisticated mathematics software package especially good at symbolic manipulation of math forms. But it is also a programming language. Programming in Mathematica has several advantages:

- We do not need to reinvent the wheel. Mathematica has abundant built-in math functions and list operation commands. They are carefully optimized. It is especially suitable for computational intensive problems.
- It has convenient "plot" and "sound" commands to visualize and hear the results immediately.
- It supports both **procedural** and **functional** programming style. In procedural programming, we use flow control commands to tell computer how to carry out the task step by step. In contrast, functional programming let us concentrate on what need to be done and let the computer to figure out how to carry out the tasks.

- We can use interactive mode to see the intermediate results instantly. And it is easy to group these interactive commands into a package that can be used later as an extension of the built-in packages.

## 3.2 Build a Mathematica Package

Mathematica has a rich set of built-in packages. End user simply type the command << *packagename* to load a package and all the special functions programmed in the package are available to the user. The implementation details are transparent to the user but the user can get help on the usage of the package.

When we have a set of interactive commands to carry out certain task, we can use the following steps to convert this set of commands into a Mathematica package. Please refer to **Appendix B** for code listing of the package. The following is the skeleton of a package:

```
BeginPackage["PacakgeName`"]
Needs["OtherPackage`"]

FunctionName::usage="Help message on how to use this function."

...

Begin["`Private`"]

FunctionName[clist_]:=
Block[
      Actual implementation of the function
           ]
...

End[]
EndPackage[]
```

We built our package in the following steps:

1. We first utilize the interactive nature of the Mathematica to carry out commands line by line to see each intermediate result. In this step, we want to make sure that each command works properly.

2. Then we put all the commands in a file and add a set of commands: `BeginPackage["MultiResolution`" ]`and `EndPackage[]` to create a "box" to hold the whole package. What they do is to change the `$Context` from `Global`` to a new variable name based on the name of the package.

3. The actual implementation of the functions are further encapsulated by the commands:`Begin["Private`"]` and `End[]`. The part that is included by these two commands, is not accessible out of the package. This makes the implementation of the specific algorithm local and avoids the accidental corruption of the program by outer functions.

4. The wavelet and B-spline function definitions are given in a separate file named `WDefinition.m`. This file is loaded automatically into Mathematica by putting the command `Needs["WDefinition`"]` at the beginning of the package.

5. The online help message is added to the package before the command `Begin["Private`"]`, so that it is accessible by the user.

6. Finally, the package is saved as `Wavelet.m` and it is put into the right directory that is included in `$Path`. In our case, it is put in `~/Library/Mathematica/Packages`.

# 4 Application of the Wavelet Transform to Signal Analysis

The Wavelet Transform has been used in a variety aspects, such as image coding [5], singularity detection [1], digital filtering [2], speech processing [9],etc. We are trying to use the B-wavelet transform on pitch detection of speech signals. Because of the complication of the problem, the application

details will be presented in a later article [8]. In this section we will show the validity of the approach and several test examples.

The pitch of the sound signal is its instant frequency. Detection of pitch change of a sound signal has applications in speech recognition and clinical diagnosis. Due to the large blocks of data that are needed to represent a meaningful section of the sound signal at high sample rate, we are still searching for effective ways to process it. The following five figures are the results of several simple test signals to show the validity of the approach. We decompose the original signals into subspace representations and reconstruct the original signals back from these subspace representations using the Mathematica package described in the previous section. Through this decomposition and reconstruction scheme, it is possible to process each set of coefficients individually to extract the information which is not accessible before.

In all the graphs, (a) is the original signal. (b) and (c) are the first level decomposition coefficients in $V$ and $W$ spaces respectively. (d) and (e) are the further decomposition of (b) into coarser $V$ and $W$ spaces, and so are (f) and (g) the decomposition coefficients of (d). (a') is the reconstruction of signal from coefficients in (b) and (c), since we did not make further manipulations on (b) and (c), (a') should be the same as (a). Also (b') is the reconstruction of (b) from (d) and (e).

The first one (see Figure 3) is a multiresolution decomposition of a pure sine wave. After each step of decomposition, the signal length is halved. Since we provide a pure sine wave, the coefficients in $V_{2j}$ subspaces are very regular and similar to the original signal. The coefficients in $W_{2j}$ subspaces are significantly smaller compared to their $V_{2j}$ counterparts. The coefficients of $W_{2j}$ subspaces in the boundary regions are relatively larger compared to the inner part of the coefficients.

The second one (see Figure 4) is a demonstration of the reconstruction algorithm using a pure chirp signal. We can compare (a) with (a') and (b) with (b'). Since we did not make further processing on coefficients of the individual levels, the original signal is reconstructed back as expected. Also notice the similarity of the overall shapes of the coefficients in different $V_{2j}$ subspaces in this example.

Figure 5 shows us that the Wavelet Transform can do similar "tricks" that the Fourier Transform can. Here we superimpose a pure sine wave with a white noise signal. Since their spectrum are separated, it is quite easy

to filter out the signal using the Fourier Transform. As we can see in this example, the Wavelet Transform also can separate the signal from the noise.

If the spectra of the signal and the noise are strongly overlapped, the Fourier Transform is not able to differentiate them. Figure 6 shows a chirp signal superimposed with white noise, both have a wide spectrum. In the Fourier transform result, we can only know that the signal includes a whole spectrum of different frequency components. But no time information on how the signal changes is evident from the spectrum. The result on autocorrelation analysis cannot give us useful information on the transition either. In Figure 7, the Wavelet Transform is used to analyze the same signal. It is clear that after only three steps of decomposition, the coefficients of these different subspaces reveal far more information of the original signal than the previous methods can.

To summarize, we find that the Wavelet Transform can be used to analyze stationary signals. These signals can be analyzed successfully by the Fourier Transform, the Wavelet Transform gives us an alternative view. The Wavelet Transform can effectively analyze nonstationary signals and shows superior performance than the Fourier Transform and the autocorrelation analysis.

# 5 Acknowledgements

The author would like to express his sincere thanks to Dr. Tian-Xiao He and Dr. Melvyn Jeter for their encouragement and guidance. Their enthusiastic support made this project an very enjoyable learning experience. He would also like to thank Dr. Jack Jiang of Northwestern University for his help on speech science and data acquisition; Dr. Larry Stout and Dr. Robin Sanders for their help on using NeXT computer.

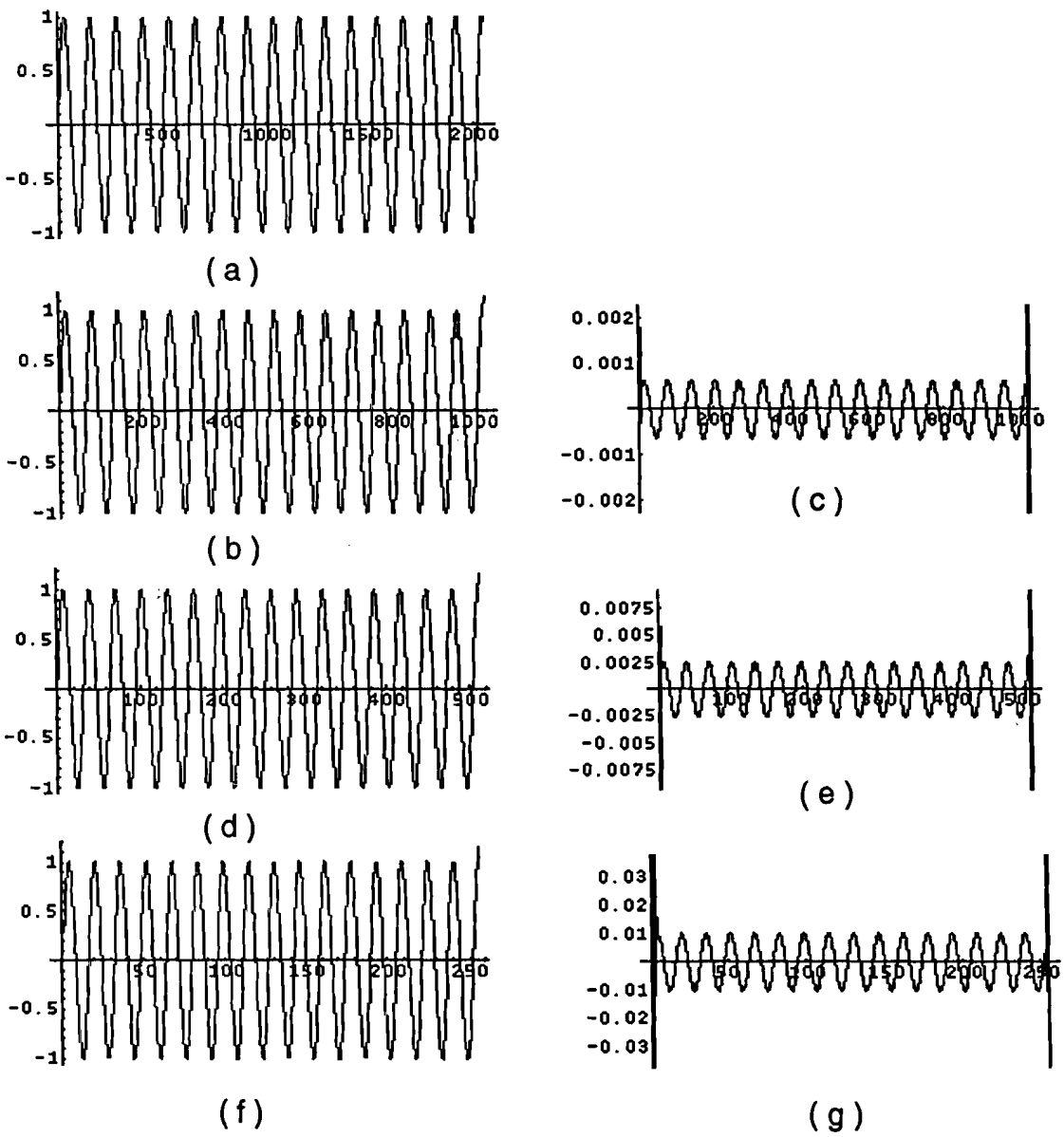



Figure 3: Multiresolution Decomposition of Pure Sine Function

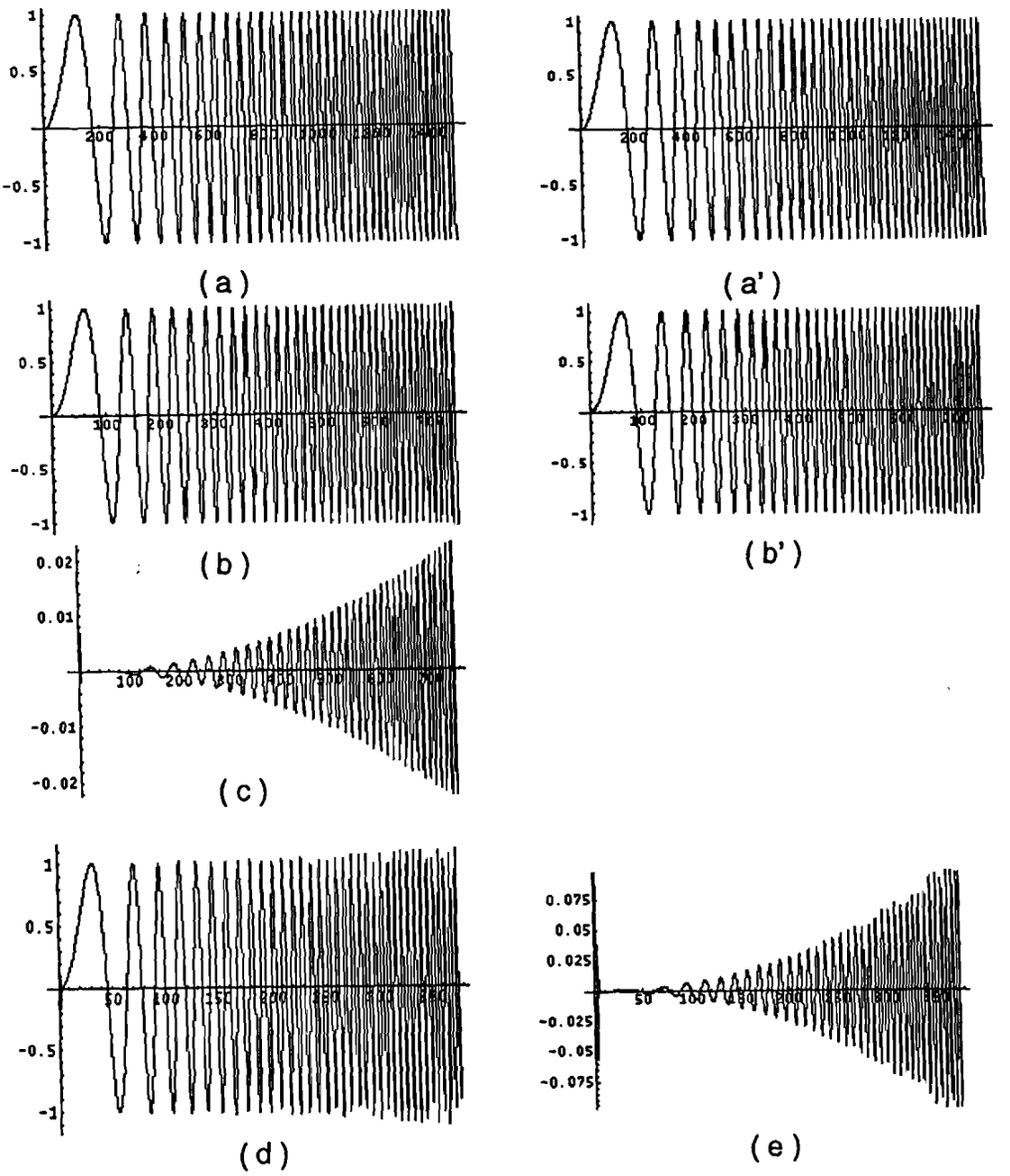



Figure 4: Decomposition and Reconstruction of Pure Chirp Function

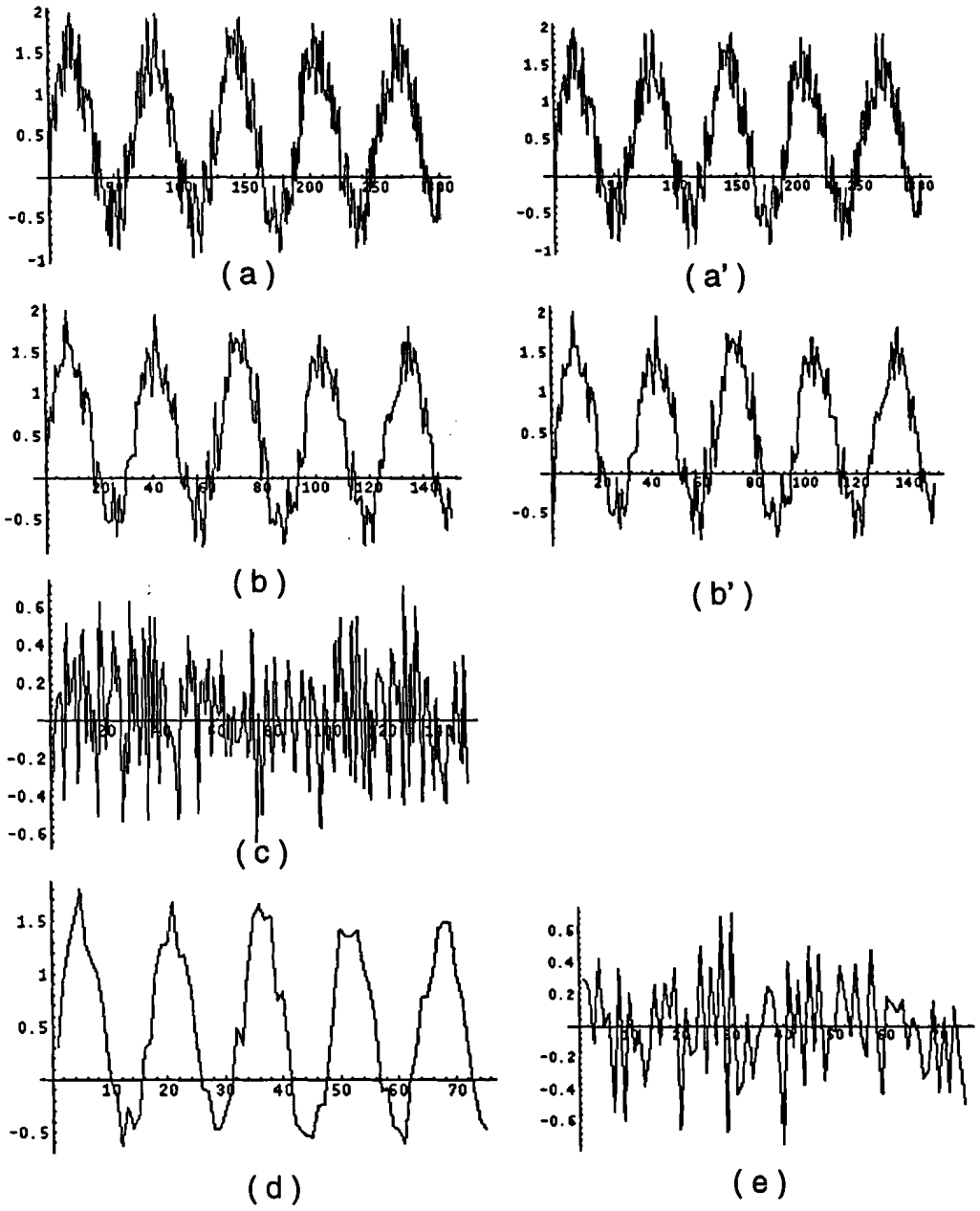
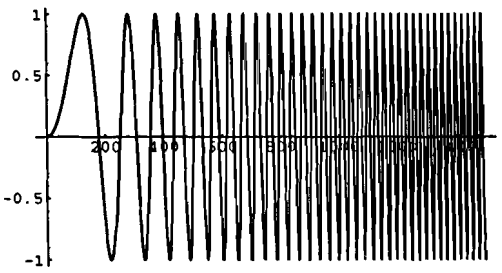
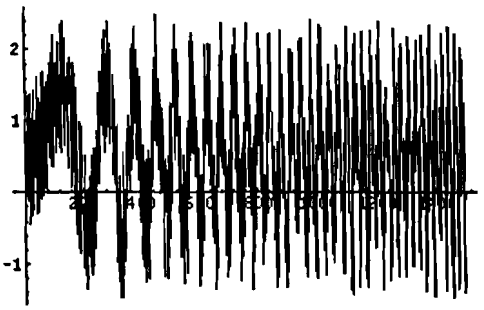


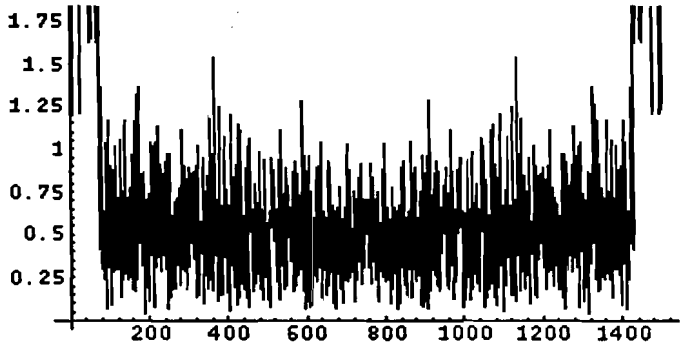

Figure 5: Decomposition and Reconstruction of Sine Wave with Noise



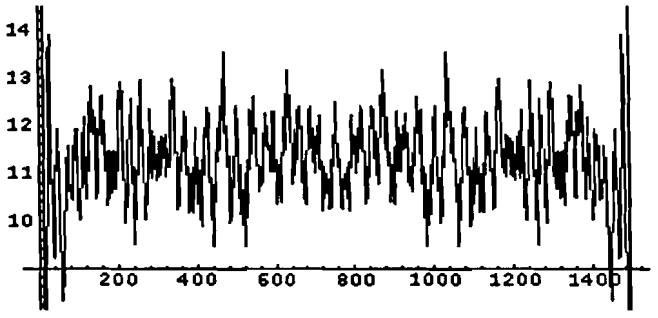

1. Pure Chirp Signal Without Noise




2. Chirp Signal with White Noice




3. Result from the Fourier Transform




4. Result from the Autocorrelation Analysis

Figure 6: Analysis of the Chirp Signal with White Noise

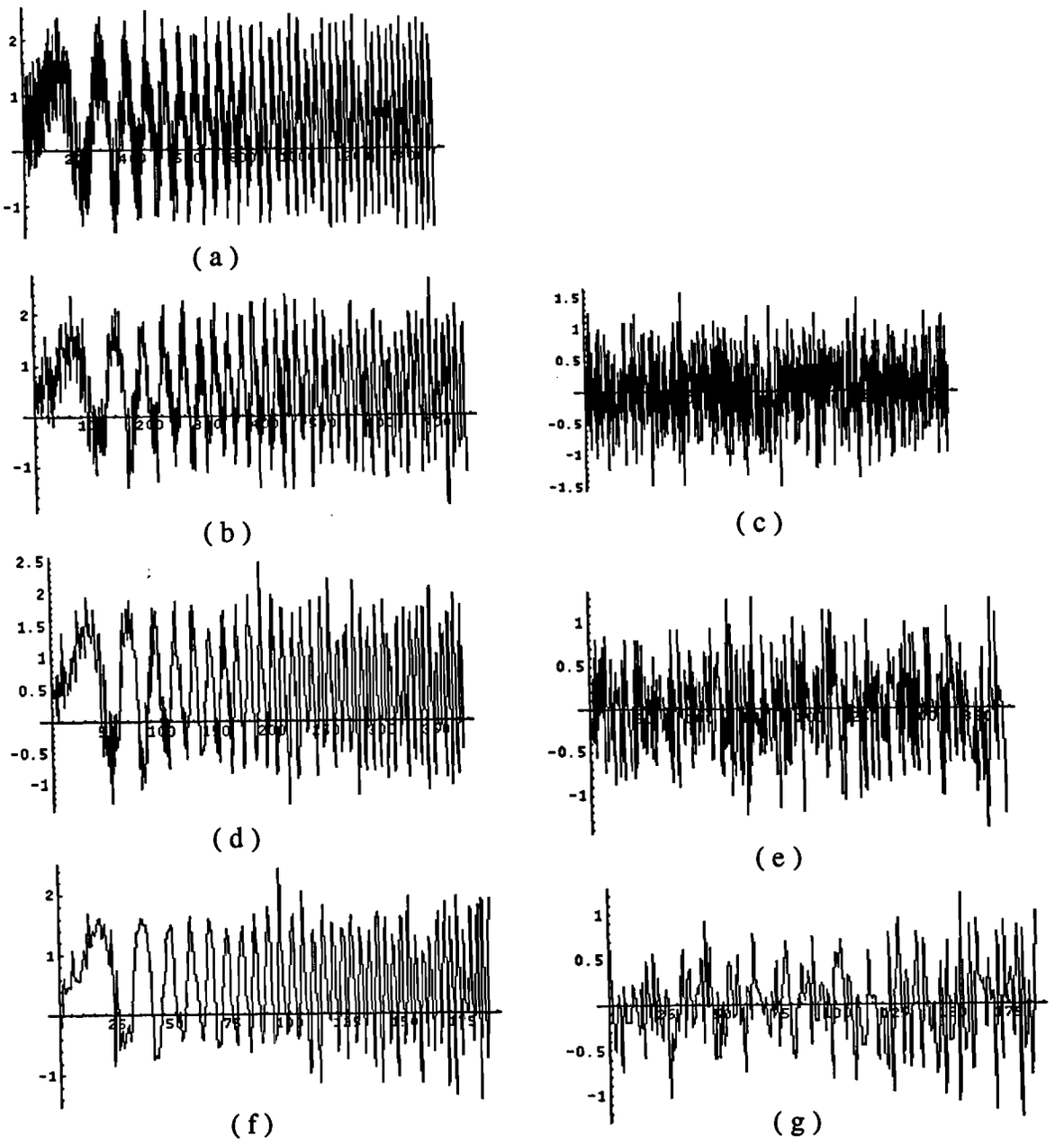



Figure 7: Decomposition of Chirp Signal with Noise

# Appendix A

## Decomposition and Reconstruction Algorithms

## A Decomposition

Since $V_{2j} \subset V_{2j+1}$,

$$\phi_{j,n}(t) = 2^{-(j+1)} \sum_{k=-\infty}^{+\infty} < \phi_{j,n}(u), \phi_{j+1,k}(u) > \phi_{j+1,k}(t). \tag{28}$$

By changing variables in the inner product integral, we have

$$2^{-(j+1)} < \phi_{j,n}(u), \phi_{j+1,k}(u) > = < \phi_{-1,0}(u), \phi_{0,(k-2n)}(u) > \tag{29}$$

Thus

$$\begin{aligned} & < x(u), \phi_{j,n}(u) > \\ = & \sum_{k=-\infty}^{+\infty} < \phi_{-1,0}(u), \phi_{0,(k-2n)}(u) > \cdot < x(u), \phi_{j+1,k}(u) > \\ = & \sum_{k=-\infty}^{+\infty} \tilde{h}(2n-k) < x(u), \phi_{j+1,k}(u) >, \end{aligned} \tag{30}$$

where $\tilde{h}(n) = h(-n)$ and $h(n) = < \phi_{-1,0}(u), \phi_{0,n}(u) >$. Therefore

$$A_{2j}^d(n) = \sum_{k=-\infty}^{+\infty} \tilde{h}(2n-k) A_{2j+1}^d(k). \tag{31}$$

Similarly,

$$D_{2j}^d(n) = \sum_{k=-\infty}^{+\infty} \tilde{g}(2n-k) D_{2j+1}^d(k), \tag{32}$$

where

$$\tilde{g}(n) = g(-n) \text{ and } g(n) = < \psi_{-1,0}(u), \psi_{0,n}(u) > .$$

# B Reconstruction

Since $V_{2j+1} = V_{2j} \oplus W_{2j}$,

$$\begin{aligned}
&\phi_{j+1,n}(x) \\
&= \quad 2^{-j} \sum_{k=-\infty}^{+\infty} < \phi_{j,k}(u), \phi_{j+1,n}(u) > \phi_{j,k}(t) \\
&+ \quad 2^{-j} \sum_{k=-\infty}^{+\infty} < \psi_{j,k}(u), \phi_{j+1,n}(u) > \psi_{j,k}(t). \qquad (33)
\end{aligned}$$

Therefore

$$\begin{aligned}
&< x(u), \phi_{j+1,n}(u) > \qquad (34) \\
&= 2 \sum_{k=-\infty}^{+\infty} h(n-2k) < x(u), \phi_{j,k}(u) > +2 \sum_{k=-\infty}^{+\infty} g(n-2k) < x(u), \psi_{j,k}(u) > .
\end{aligned}$$

Hence we have

$$A^d_{2j+1}(n) = 2 \sum_{k=-\infty}^{+\infty} h(n-2k) A^d_{2j}(k) + 2 \sum_{k=-\infty}^{+\infty} g(n-2k) D^d_{2j}(k). \qquad (35)$$

# Appendix B

## Code List of the Package

This Appendix includes the program code list of the package. It has two files: **WaveletPackage.m** and **WDefinition.m**. Detailed tutorial of Mathematica programming can be found in the book by Roman Maeder [10].

```
% WaveletPackage.m
BeginPackage["MultiResolution`"]
Needs["WDefinition`"]

CDecomposition::usage="CDecomposition[c_List] compute the
        multiresolution coefficients Ci of the subsequent
        level."

DDecomposition::usage="DDecomposition[d_List] compute the
        multiresolution coefficients Di of the subsequent
        level."

Reconstruction::usage="Resconstructon[c_List,d_List]
        reconstructs the multiresolution coefficients of
        the upper level by the coefficients Ci and Di of
        the lower level."

Begin["`Private`"]
CDecomposition[clist_]:=
Block[{app1={},app2={},c={},cc={},l=0},
        l=Length[clist];
        app1=Take[clist,21];
        app2=Take[clist,-20];
        c=Join[app2,clist,app1];
        For[j=1,j<=l/2,j++,
              AppendTo[cc,Sum[NA2[[n]]*c[[n+2j-2]],
                                        {n,1,41}]]];
              cc
          ]
```

```
DDecomposition[clist_]:=
Block[{bpp1={},bpp2={},d={},dd={},l=0},
        l=Length[clist];
        bpp1=Take[clist,21];
        bpp2=Take[clist,-20];
        d=Join[bpp2,clist,bpp1];
        For[j=1,j<=l/2,j++,
            AppendTo[dd,Sum[NB2[[n]]*d[[n+2j-2]],
                                        {n,1,41}]]];
            dd
           ]

Reconstruction[cc_,dd_]:=
Block[{c={},d={},ac={},ad={},clist={},l=0},
        l=Length[cc];
        ac=Take[cc,-1];
        ad=Take[dd,-2];
        c=Join[ac,cc];
        d=Join[ad,dd];
        c=Partition[c,1];
        d=Partition[d,1];
        c=Map[{#,0}&,c];
        d=Map[{#,0}&,d];
        c=Flatten[c];
        d=Flatten[d];
        For[j=1,j<=2 l,j++,
           AppendTo[clist,
              Sum[NP2[[n]]*c[[n+j-1]],{n,1,3}]+
              Sum[NQ2[[n]]*d[[n+j-1]],{n,1,5}] ]
            ]; (* End For *)
         clist
      ]


End[]
EndPackage[]
```

```
% WDefinition.m
% This is the support part of the package. It includes scale
% and wavelet functions definitions and decomposition,
% reconstruction coefficients

BeginPackage["WDefinition`"]

phi2[x_]:=Which[0<x<1,x, 1<=x<2, 2-x,True,0]

casi2[x_]:=1/12(phi2[2x]-6phi2[2x-1]+10phi2[2x-2]
-6phi2[2x-3]+phi2[2x-4])

h1[x_]:=1/6 x^3;
h2[x_]:=2/3 (x-1)^3+2(2-x)(x-1)^2+(2-x)^2(x-1)+1/6(2-x)^3;
h3[x_]:=1/6(x-2)^3+(x-2)^2(3-x)+2(x-2)(3-x)^2+2/3(3-x)^3;
h4[x_]:=1/6(4-x)^3;
phi4[x_]:=Which[0<x<=1,h1[x],1<x<=2,h2[x],2<x<=3,h3[x],
          3<x<=4,h4[x],True,0]

casi4[x_]:=1/(8*5040) (phi4[2x]-124 phi4[2x-1]+1167 phi4[2x-2]
       - 7904 phi4[2x-3] + 18482 phi4[2x-4] - 24264 phi4[2x-5]
       + 18482 phi4[2x-6] - 7904 phi4[2x-7] + 1167 phi4[2x-8]
       - 124 phi4[2x-9] + phi4[2x-10] )

NP2={0.5,1,0.5};

NQ2={1/12,-1/2,5/6,-1/2,1/12};

NA2=

{8.26079*10^-7, -(2.256905*10^-6), -(3.08299*10^-6),
  8.422897*10^-6, 0.000011505891, -0.000031434679,
  -0.00004294056900000001, 0.000117315818,
  0.000160256388, -0.000437828595,
  -0.0005980849830000001, 0.001633998562, 0.002232083545,
```

```
 -0.006098165652, -0.008330249198, 0.022758664048,
 0.031088913246, -0.084936490539, -0.116025403784,
 0.316987298108, 0.6830127018920001, 0.316987298108,
 -0.116025403784, -0.084936490539, 0.031088913246,
 0.022758664048, -0.008330249198, -0.006098165652,
 0.002232083545, 0.001633998562, -0.0005980849830000001,
 -0.000437828595, 0.000160256388, 0.000117315818,
 -0.00004294056900000001, -0.000031434679,
 0.000011505891, 8.422897*10^-6, -(3.08299*10^-6),
 -(2.256905*10^-6), 8.26079*10^-7};


NB2=

 {0.0, 0.0, 2.2569054*10^-6, -(6.165980000000001*10^-6),
 -(8.422897*10^-6), 0.000023011782, 0.000031434678,
 -0.000085881139, -0.000117315818, 0.000320512777,
 0.000437828595, -0.001196169967, -0.001633998561,
 0.004464167091000001, 0.006098165652, -0.016660498395,
 -0.022758664047, 0.06217782649100001, 0.084936490539,
 -0.232050807569, -0.316987298108,
 0.866025403784, -0.316987298108, -0.232050807569,
 0.084936490539, 0.06217782649100001, -0.022758664047,
 -0.016660498395, 0.006098165652, 0.004464167091000001,
 -0.001633998561, -0.001196169967, 0.000437828595,
 0.000320512777, -0.000117315818, -0.000085881139,
 0.000031434678, 0.000023011782, -(8.422897*10^-6),
 -(6.165980000000001*10^-6), 2.2569054*10^-6};

EndPackage[]
```

# References


[1] S.G.Mallat and W.L.Hwang *Singularity Detection and Processing with Wavelets* IEEE Transaction on Information Theory, Vol38,No.2, March 1992

[2] Martin Vetterli and Cormac Herley *Wavelets and Filter Banks: Theory and Design* IEEE Transactions on Signal Processing, Vol.40, No.9, September 1992

[3] R.K.Young *Wavelet Theory and Its Applications* Klumer Academic Publishes, Boston 1992

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

[5] S. G. Mallat *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation* IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.11, No.7, July 1989

[6] A.K. Chan, C.K.Chui, J.Z.Wang, Q.Liu and J.Zha *Introduction to B-wavelets and Applications to Signal Processing* Preprint, 1991

[7] T.X. He *The Spline Interpolations and B-Wavelets* J. Math. Res. Exp., To appear

[8] T.X. He, J. Chen, and J. Jiang *Application of the Wavelet Transform with B-Wavelet on pitch detection* in preparation

[9] Shubha Kadambe and G. Faye Boudreaux-Bartels *Application of the Wavelet Transform for Pitch Detection of Speech Signals* IEEE Transaction on Information Theory, Vol.38, No.2, March 1992

[10] Roman Maeder *Programming in Mathematica* Addison-Wesley Publishing Company, Inc. 1990