SYMBOLIC ANALYSIS OF ELECTRONIC CIRCUITS USING WAVELET TRANSFORM

A. A. Al-Habib and Prof. F. M. Al-Naima
Department of Computer Engineering
College of Engineering, Saddam University,
P. O. Box 64940, Jadriya, Baghdad, IRAQ
email:sadeng@uruklink.net
Fax: 7786242

ABSTRACT
In recent years, symbolic analysis has become a well-established technique in circuit analysis and design. The symbolic expression of network characteristics offers convenience for frequency response analysis, sensitivity computation, and fault diagnosis. The aim of the paper is to present a method for symbolic analysis that depends on the use of the wavelet transform (WT) as a tool to accelerate the solution of the problem as compared with the numerical interpolation method that is based on the use of the fast Fourier transform (FFT).

التحليل الرمزى للدوائر الإلكترونية باستخدام تحويل الموجبة

لاصبح التحليل الرمزى للدوائر في السنوات الأخيرة تجربة موثوقة في تحليل وتصميم الدوائر الإلكترونية. وتوفير هذه العناية الترميزية للمواد والشبكات وسماحية مانا من خلال تحليل الاسمية والفحوصات السلبية في التوصيف والإرشادات والقياسيات المعنوية، إن الهدف من هذا البحث هو تقديم طريقة للتحليل الرمزى للدوائر والذي يعتمد على تحويل الموجبة كوسنزا لتسريع حل المسائل المقاوسة مع القدرة على الإعداد وأنماطها.

المرجع:
الدكتور فوزي سعد مهندس لlesai الدفاعي
المهندس: علي عبد الرازق الطالبي
اسم جامعة الدوائر - كلية الهندسة - جامعة كوردستان
العنوان: 64040 - الدوارة
بغداد - جمهورية العراق
email: sadeng@uruklink.net
Fax: 7786242
1. INTRODUCTION:
It is obvious that the methods of symbolic analysis can be divided mainly into two categories. These are the topological and numerical methods [1]. Each one of these methods has its own advantages and disadvantages. For instance, in topological methods the number of elements represented as symbol is large but the circuits that can be handled is small, while in numerical methods fairly large networks can be handled but the number of symbolic variables should not exceed 10. The direct application of numerical interpolation method can be used to solve problems of system matrix size of 30 and about 10 elements only represented as variables beside the complex frequency's" [2,3].

Many algorithms have been developed and from these deterministic and flow graph methods appear to be favoured in terms of flexibility and efficiency [3]. All approaches suffer from restrictions inherent to the problem, the escalation of computer time and memory requirements with increase in circuit size. The serious limitation of such methods, in practice, is the rapidly increasing amount of computations required as the number of symbols to be handled increase [3]. This will, in fact, increase the time required to solve the linear system equation of the circuit.

The numerical interpolation method for obtaining the symbolic analysis suffers from serious limitation in practice, which is the rapidly increasing amount of computations required as the number of symbols to be handled increases. This, in fact, reflects the amount of time required to perform the analysis. For this reason, it is useful to find an approach to minimize the computations required by the numerical interpolation as minimum as possible. The usual numerical interpolation method is based on the use of the FFT. One way to reduce the computations required by the numerical interpolation is to search for a transform that will perform the required task, besides minimizing the computations, and hence, reduces the required time to perform the analysis as compared to the FFT. As an example, the Hardy Transform (HT) could be used to replace the FFT for the symbolic analysis and a comparison could be made between the HT and the FFT to see which is better from the point of view of reducing the required computation, and hence the time of doing the analysis. One other promising transform that may replace the FFT is the Wavelet Transform (WT) [4,5]. A new approach to minimize the computations and the time required is the neural network approach to the interpolation problem that allows to get the solution in a real time [6].

The method proposed in this paper tries to reduce the time required by the numerical interpolation method to solve the system equation by using the wavelet transform.

2. NUMERICAL INTERPOLATION METHOD FOR SYMBOLIC ANALYSIS

Numerical interpolation methods are based on the theory and implementation of numerical methods for generating symbolic functions of networks. They seem to have a lower computational cost than other well-known symbolic analysis algorithms such as a parameter extraction method. The following discussion will introduce the idea of using interpolation in finding network transfer functions using the Discrete Fourier Transform (DFT) [3,7,8,9].
2.1 POLYNOMIAL INTERPOLATION

First, we find N+1 points by evaluating the function:

\[ P_n(x) = \text{def} \left[ A(x) \right] \]  \hspace{1cm} (1)

at \( x_0, x_1, \ldots, x_N \), where N is the maximum power of x. Now, there are N+1 distinct points \( (x_i, y_i = P_n(x_i)), i = 0, 1, \ldots, N \). Both \( x_i \) and \( y_i \) may be real or complex numbers. We wish to find the coefficients of the polynomial:

\[ P_n(x) = \sum_{n=0}^{N} a_n x^n \]  \hspace{1cm} (2)

such that the polynomial passes through the given points.

Inserting \( x_i \) into the polynomial (2), we obtain the set of equations:

\[ a_0 + a_i x_i + a_2 x_i^2 + \cdots + a_N x_i^N = y_i, \]  \hspace{1cm} (3)

with unknowns \( a_0, a_1, a_2, \ldots,a_N \). Since there are \( N+1 \) unknown coefficients and the same number of equations, we can write the matrix equation:

\[ \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} \]  \hspace{1cm} (4)

Or:

\[ [X][a] = [y] \]  \hspace{1cm} (5)

The solution of (5) provides the unknown coefficients.

As we have the choice of selecting the points \( x_i \), the question arises as to what the choice should be in order to obtain the best possible result. It can be shown that the interpolation with real \( x_i \) is in general, numerically unstable [7].

2.2 THE USE OF THE DISCRETE FOURIER TRANSFORM IN INTERPOLATION

We will derive this interpolation by introducing first a special symbol for the matrix \( X \) in (5):

\[ X = \left[ x_i^n \right] \]  \hspace{1cm} (6)

where the index \( i \) and the exponent \( n \) run from \( 0 \) to \( N \). If we choose the set of points \( x_i \) to be uniformly spaced on the unit circle in the complex plane, then these points are:

\[ x_0 = 1, \quad x_i = \exp \left( \frac{2\pi i k}{N+1} \right), \quad k = 1, 2, \ldots, N. \]  \hspace{1cm} (7)

Introduce the substitution:

\[ w = \exp \left( \frac{2\pi i}{N+1} \right) \]  \hspace{1cm} (8)

Then:

\[ x_i = w^k \]  \hspace{1cm} (9)

And:

\[ X = \left[ w^{kn} \right] \]  \hspace{1cm} (10)
It can be shown that [3],

$$X^T = \frac{1}{N+1} \sum_{\omega=0}^{N-1} e^{j \omega \frac{\pi}{N}} \chi$$  \hspace{1cm} (11)$$

Where $X^T$ denotes the transpose conjugate matrix and $\chi$ runs from 0 to $N$.

The solution of (5) with the points defined by (7) is:

$$A = X^{-1} Y = \frac{1}{N+1} \sum_{\omega=0}^{N-1} e^{j \omega \frac{\pi}{N}} Y$$  \hspace{1cm} (12)$$

or:

$$a_n = \frac{1}{N+1} \sum_{\omega=0}^{N-1} p_{\omega} e^{j \omega n}$$ \hspace{1cm} (13)$$

$n = 0, 1, 2, \ldots, N$.

The original polynomial in (2), evaluated at $a_n$, can be written as:

$$y_\omega = \sum_{n=0}^{N-1} a_n e^{j n \omega}$$  \hspace{1cm} (14)$$

Equation (13) and (14) represent the solution of one another. They are called the Discrete Fourier Transform (DFT) pair.

To improve the speed of the method, one can use a fast algorithm in interpolation. Algorithms that reduce the computational cost of DFT are, in general called the Fast Fourier Transform (FFT). The DFT has been studied extensively. It can be programmed in a very efficient way, particularly when $N+1=2^m$, $m$ being a positive integer. The number of operations required in this case is $m(N+1)$ [3,5].

3. THE USE OF THE WAVELET TRANSFORM (DWT)

In this section, the use of the Discrete Wavelet Transform (DWT) will be illustrated. Before this, the DWT must be briefly explained.

3.1 THE WAVELET TRANSFORM

Like the FFT, the Discrete Wavelet Transform (DWT) is a fast linear operation that operates on a data vector whose length is an integer power of two, transforming it into a numerically different vector of the same length. Also, like the FFT, the WT is invertible and in fact orthogonal, that is, the inverse transform when viewed as a big matrix, is simply the transpose of the transform. Both FFT and DWT, therefore, can be viewed as a rotation in space, from the input space (or time) domain, where the basis functions are the unit vectors $\delta_\omega$, or Dirac delta functions in the continuum limit, to a different domain. For the FFT, this new domain has basis functions that are the familiar sines and cosines. In the wavelet domain, the basis functions are somewhat more complicated and have the fanciful names "mother functions" and "wavelets" [10].

Of course, there are an infinity of possible bases for function space, almost all of them uninteresting. What makes the wavelet basis interesting is that, unlike sines and cosines, individual wavelet functions are quite localized in space; simultaneously, like sines and cosines, individual wavelet functions are quite localized in frequency or (more precisely) characteristic scale. The particular kind of dual localization achieved by wavelets renders large classes of functions and operators sparse, or sparse to some high accuracy, when transformed into the wavelet domain. Analogously with the Fourier domain, where a class of computations, like convolutions, becomes computationally fast, there is a large class of computations (those that can take the advantage of sparsity) that become computationally fast in the wavelet domain [4,7,10].
Unlike sines and cosines, which define a unique Fourier transform, there is no single unique set of wavelets; in fact, there are infinitely many possible sets. Roughly, the different sets of wavelets make different trade-offs between how compactly they are localized in space and how smooth they are.

3.2 DAUBECHIES WAVELET FILTER COEFFICIENTS

A particular set of wavelets is specified by a particular set of numbers called wavelet filter coefficients. Here, we will largely restrict ourselves to wavelet filters in a class discovered by Daubechies. This class includes members ranging from highly localized to highly smooth. The simplest (and most localized) member, often called DAUBECHIES, has only four coefficients, $c_0, c_1, c_2,$ and $c_3$ [4,7,10].

Consider the following transformation matrix acting on a column vector of data to its right:

$\begin{pmatrix}
-\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\
-\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\
-\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\
-\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4}
\end{pmatrix}
\begin{pmatrix}
X_0 \\
X_1 \\
X_2 \\
X_3
\end{pmatrix}
$

Here, blank entries signify zeroes. Note the structure of this matrix. The first row generates one component of the data convoluted with the filter coefficients $c_0, c_1, c_2,$ and $c_3$. Likewise, the third, fifth, and other odd rows. If the even rows followed this pattern offset by one, then matrix would be a circulant, that is, an ordinary convolution that could be done by FFT methods. (Note how the last two rows wrap around like convolutions with periodic boundary conditions.) Instead of convolution with $c_0, c_1, c_2,$ and $c_3$, however, the even rows perform a different convolution, with coefficients $c_3, c_2, c_1,$ and $-c_0$. The action of the matrix, overall, is thus to perform two related convolutions, then to decimate each of them by half (throw away half the values), and interleave the remaining halves.

It is useful to think of the filter $c_0, c_1, c_2,$ and $c_3$ as being a smoothing filter called H, something like a moving average of four points. Then, because of the minus sign, the filter $c_1, -c_2, c_3,$ and $-c_0$, call it G, is not a smoothing filter. In fact, the G's are chosen so as to make $G$ yield, insofar as possible, a zero response to a sufficiently smooth data vector. This is done by requiring the sequence $c_0, -c_2, c_3,$ and $-c_0$ to have a certain number of vanishing moments. When this is the case for $p$ moments (starting with the zeroth), a set of wavelets is said to satisfy an approximate condition of order $p$. This result in the output of $H$, decimated by half, accurately representing the data’s “smooth” information. The output of $G$, also decimated, is referred to as the data’s “detail” information [10].

For such a characterization to be useful, it must be possible to reconstruct the original data vector of length $N$ from its $N/2$ smooth or $p$-components and its $N/2$ detail or $d$-components. That is effected by requiring the matrix (15) to be orthogonal, so that its inverse is just the transposed matrix.
Now, since

\[ WW^T = I \quad \text{(17)} \]

where \( I \) is the identity matrix, one sees immediately that matrix (16) is the inverse of matrix (15) if and only if the following two equations hold:

\[
\begin{align*}
    c_0^2 + c_1^2 + c_2^2 + c_3^2 &= 1 \\
    c_0c_1 + c_2c_3 &= 0
\end{align*}
\]

(18)

If additionally, we require the approximation condition of order \( p=2 \), then two additional relations are required:

\[
\begin{align*}
    c_0^2 + c_1^2 + c_2^2 + c_3^2 &= 1 \\
    c_0c_1 + c_2c_3 &= 0
\end{align*}
\]

(19)

Equations (18) and (19) are 4 equations for the 4 unknowns \( c_0, c_1, c_3, \) and \( c_4 \), first recognized and solved by Daubechies. The unique solution (up to a left-right reversal) is:

\[
\begin{align*}
    c_0 &= \left(1 + \sqrt{3}\right)/4\sqrt{2} \\
    c_1 &= \left(3 + \sqrt{3}\right)/4\sqrt{2} \\
    c_2 &= \left(3 - \sqrt{3}\right)/4\sqrt{2} \\
    c_3 &= \left(1 - \sqrt{3}\right)/4\sqrt{2}
\end{align*}
\]

(20)

In fact, DAUB4 is only the most compact of a sequence of wavelet sets. If we have six coefficients instead of four, there would be three orthogonality requirements in equation (18) (with offsets of zero, two and four), and we could require the vanishing of \( p=3 \) moments in equation (19). In this case, DAUB6, the solution coefficients can also be expressed in closed form:

\[
\begin{align*}
    c_0 &= \left(1 + \sqrt{10 + 3 + 2\sqrt{10}}\right)/16\sqrt{2} \\
    c_1 &= \left(5 + \sqrt{10 + 3 + 2\sqrt{10}}\right)/16\sqrt{2} \\
    c_2 &= \left(10 - 2\sqrt{10 - 2\sqrt{10 + 3 + 2\sqrt{10}}}\right)/16\sqrt{2} \\
    c_3 &= \left(10 - 2\sqrt{10 + 2\sqrt{10 + 3 + 2\sqrt{10}}}\right)/16\sqrt{2} \\
    c_4 &= \left(5 + \sqrt{10 - 3\sqrt{10 + 3 + 2\sqrt{10}}}\right)/16\sqrt{2} \\
    c_5 &= \left(1 + \sqrt{10 - \sqrt{3 + 2\sqrt{10}}}\right)/16\sqrt{2}
\end{align*}
\]

(21)

For higher \( p \), up to 10, Daubechies has tabulated the coefficients numerically. The number of coefficients increases by two each time \( p \) is increased by one.

### 3.3 The Discrete Wavelet Transform (DWT)

The DWT consists of applying a wavelet coefficient matrix like (15) hierarchically, first to the full data vector of length \( N \), then to the “smooth” vector of length \( N/2 \), then to the “smoothed” vector of length \( N/4 \), and so on until only a trivial number of “smooth... smooth” components (usually 2) remain. The procedure is sometimes called a pyramidal algorithm (or Mallat’s pyramidal algorithm), for obvious reasons. The output of the DWT consists of these remaining components and all the “detail” components that were accumulated along the way. A diagram should make the procedure clear:
3.4 THE USE OF DWT FOR FAST SOLUTION OF LINEAR SYSTEMS

One of the most interesting, and promising, wavelet applications is linear algebra \[16\]. The basic idea is to think of integral operator (that is, a large matrix) as a digital image. Suppose that the operator compresses well under a two-dimensional wavelet transform, i.e., that a large function of its wavelet coefficients are so small, as to be negligible. Then any system involving the operator becomes a sparse system in the wavelet basis. In other words, to solve:

\[ A \cdot x = b \]  \hspace{1cm} (23)

we first wavelet-transform the operator \( A \) and the right-hand side \( b \) by:

\[ \tilde{A} \equiv W \cdot A \cdot W^T \quad \tilde{b} \equiv W \cdot b \]  \hspace{1cm} (24)

where \( W \) represents the one-dimensional wavelet transform, then solve:

\[ \tilde{A} \cdot \tilde{x} = \tilde{b} \]  \hspace{1cm} (25)

which is a sparse system in the wavelet basis, and hence, this property can be used to solve this system in a faster way than usual, by using methods for solving the sparse systems, so that we can obtain the results almost in a real-time manner.

Finally, transform to the answer by the inverse wavelet transform:

\[ x = W^T \cdot \tilde{x} \]  \hspace{1cm} (26)

The results will appear with a high accuracy as compared with the use of other transforms to perform the same task.

The method discussed above was implemented and verified for solving numerical linear systems in a fast way. It is
The Wavelet Transform and Harmonic Analysis

3.5 The Wavelet Matrices

Symbolic matrices will be shown here. The tableau is simplified to obtain the reduced form. A tableau is simplified by putting a number of numbers in each box. The symbols are then solved out and placed in the tableau. This problem is solved by working out the symbols and applying the reduced form. The table is simplified by putting the symbols on the table and working out the symbols. The table will be obtained when performing the operations and applying the simplified form. The tableau will be obtained when performing the operations and applying the simplified form.
\[ w(x) = \left( e^{ix} - e^{-ix} \right) / 2ix \]  

(30)

4. APPLICATION EXAMPLES

This section presents some examples of using the previously mentioned algorithm that depends on the use of the DWT as compared to the use of the FFT from the point of view of reducing the amount of calculations and hence the execution time. The software required to perform this task is written using the language of the MATLAB package.

For the purpose of fair comparison, two versions of the symbolic analysis programs were written, one uses the FFT (called SAAFFT: Symbolic Analyzer Using Fast Fourier Transform) and the other uses the DWT (called SAUDWT: Symbolic Analyzer Using Discrete Wavelet Transform) in numerical interpolation. Also, the circuits were used in both programs to perform the symbolic analysis. The results are obtained using a Pentium II microprocessor that operates on 233 MHz frequency and with 16 MB RAM memory.

EXAMPLE 1: Consider the RC ladder circuit shown in Fig. 1. It is desired to find the voltage transfer function \( V_o/V_i \). This circuit contains passive elements only with 10 symbolic variables. The description of the circuit was input to the program in a SPICE-like format.

The analysis of this circuit using program SAUDWT and SAAFFT yields the same transfer function but with different times of execution. The result is as shown below:

**THE NUMERATOR IS:**

\[ \frac{C_3 + C_2 + C_4 + C_5 + R_1 + R_3 + R_2 + R_4 + R_5}{s^2 + (R_1 + R_3 + R_2 + R_4 + R_5) s + (R_1 + R_3 + R_2 + R_4 + R_5)} \]

**THE DENOMINATOR IS:**

\[ (R_1 + R_3 + R_2 + R_4 + R_5) s^3 + (R_1 + R_3 + R_2 + R_4 + R_5) s^2 + (R_1 + R_3 + R_2 + R_4 + R_5) s + (R_1 + R_3 + R_2 + R_4 + R_5) \]
Fig. 1 Circuit of example 1

\[
\begin{align*}
(C_4 R_4 + C_5 R_5 + R_2 C_2 + C_1 R_1 \\
+ R_3 C_3 + R_1 C_1 + R_2 C_2 + C_1 R_1 \\
+ R_2 C_3 + R_1 C_3 + R_2 C_2 + R_3 C_3 \\
+ R_4 C_5 + C_3 R_5 + R_2 C_2) & \quad s + 1
\end{align*}
\]

The analysis of this circuit using programs SAUDWT and SAUFFT yields the same transfer function but with different times of execution. The result is as shown below:

**THE NUMERATOR IS...**

\[g_{n1} g_{n2} g_{n3} g_{n4}\]

**THE DENOMINATOR IS...**

\[
\begin{align*}
C_3 & \quad C_2 & \quad C_3 & \quad s^4 + C_3 & \quad C_1 & \quad C_2 & \quad C_3 & \quad g_{n4} & \quad s^3 \\
+ (g_{n3} g_{n2} C_1 & \quad C_4 & + C_1 & \quad C_2 & \quad g_{n4} & \quad g_{n3}) & \quad s^2 \\
+ (g_{n2} g_{n1} C_3 & + C_1 & \quad g_{n3} & \quad g_{n2}) & \quad s + g_{n2} g_{n1} g_{n3} & \quad g_{n4}
\end{align*}
\]

**EXAMPLE 2:** Consider the circuit shown in Fig. 2. The circuit contains eight symbolic variables, which are \( C_1, C_2, C_3, C_4, g_{n1}, g_{n2}, g_{n3}, \) and \( g_{n4}, \) where the \( g_n \)'s are the transconductances of the OTA (Operational Transconductance Amplifier) devices. The active devices are modeled using the Miller-Neurath equivalent circuit.

**TIME OF EXECUTION OF PROGRAM SAUDWT:**

TIME = 4 SECONDS.

**TIME OF EXECUTION OF PROGRAM SAUFFT:**

TIME = 10 SECONDS.

**EXAMPLE 3:** Consider the circuit shown in Fig. 3. The circuit contains 11 symbolic elements and 4 OPAMP's (Operational Amplifiers). After analyzing this circuit using the two programs, the result was:
THE NUMERATOR IS...

\[-R_{10} R_7 R_4 R_5 C_2 R_0 R_3 C_1 R_1 s^2 - R_{15}(R_7 R_4 R_5 C_2 R_0 R_3 C_1 R_1 s^2 - R_{15} R_7 R_4 R_5 C_2 R_0 R_3 C_1 R_1 s^2)\]

THE DENOMINATOR IS...

\[R_3 R_5 R_3 R_2 R_1 C_2 R_1 C_1 R_1 s^2 + R_7 R_3 R_5 R_2 R_1 C_2 R_1 s + R_7 R_5 R_3 R_2 R_1 s^2\]

EXECUTION TIME:

PROGRAM ONE SADWDT:
TIME=10 SECONDS.

PROGRAM TWO SALEFFT:
TIME=25 SECONDS.

5. PERFORMANCE COMPARISON BETWEEN THE FFT AND THE DWT

Fig. 4 shows a simple comparison between the performance of the FFT and the DWT for their use in the symbolic analysis. From the figure, we can see that for small number of symbolic variables, the performance of the two transforms is almost the same. At large number of symbolic variables, however, the difference becomes very clear between the two transforms. Also, one can see that the DWT continues in providing the analysis for large number of symbolic variables with excellent time, while in FFT, the time increases rapidly with increasing the symbolic variables and it fails at certain number of symbolic variables to provide the required results. It should be mentioned that these results (those shown in Fig.4) are taken for a certain act of circuits.
and applied to programs SAUDWT and SAUFFT for the purpose of fair comparison. Of course, not only the number of symbolic elements affects the required time of execution, but also the configuration of the circuit, that is the number of nodes and branches. The figure shows the results up to about 26 symbolic variables and circuits with larger number of variables can also be analyzed with the program SAUDWT only.

6. CONCLUDING COMMENTS

The application of wavelets is still new. The subject is developing fast and many questions remain to be answered. For example, What is the best choice of wavelet to use for a particular problem? How far does the harmonic wavelet’s computational simplicity compensate for its slow rate of decay in the x-domain (proportional to x’)? For condition monitoring, the DWT (using families of orthogonal wavelets) will be competing with time-frequency methods using the Short-Time Fourier Transform (STFT) and the Wigner-Ville distribution [4,10]. Orthogonal wavelets give fast algorithms and there is no redundancy: N data points give N wavelet amplitudes. Instead of a signal’s mean-square being given by the
Number of symbolic variables
Fig. 4 Comparison in the timing performance between the DWT and the FFT transforms when used in the symbolic analysis.

Most of the usefulness of wavelets rests on the fact that wavelet transforms can usefully be severely truncated, that is, turned into sparse expansions. The case of Fourier transforms is different: FFTs are ordinarily used without truncation, to compute fast convolutions, for example. This works because the convolution operator is particularly simple in Fourier basis \[4,5,10\].

Harmonic wavelets can be described by a simple analytical formula, they are compact in the frequency domain, and are described by a complex function. Dilatation wavelets cannot be expressed in functional form, they are compact in the x-domain.

REFERENCES


