Hybrid Wavelet-Based Algorithms with Fast Reconstruction Features

Ileana-Diana Nicolae
Dept. of Computers and Information Technology
University of Craiova
Craiova, Romania
nicolae_ileana@software.ucv.ro

Petre-Marian Nicolae* and Marian-Ştefan Nicolae+
Dept. of Electr., Energetic and Aero-Spatial Engineering
University of Craiova
Craiova, Romania
* pnicolae@elth.ucv.ro , + nmarianstefan@yahoo.com

Abstract— **The paper is concerned with various algorithms used for the analysis based on the Discrete Wavelet Transform (DWT) of (non)stationary regimes involving almost sinusoidal waves and for data communication applications respectively. Different techniques are employed to perform analysis based on (de)compositions using two different trees: an unbalanced 10 level tree and a 6 level binary tree respectively. Our hybrid algorithms (for filters of length 4 and 6) are described and discussed. Their usability is demonstrated both for high and low power applications, a significant advantage being related to their fast "exact reconstruction" property. Considering the results and other important criteria (run time, memory consumptions), practical recommendations are made with respect to the selection of a proper reliable and fast DWT algorithm depending on the real applicability scenario.**

Keywords-discrete wavelet transform; hybrid algorithms; signals reconstruction; digital transmissions.

I. INTRODUCTION

Fourier Transforms (FT) are very helpful in signal processing, but they capture global features. They evaluate harmonic components of the entire signal, being obtained by dot-producting of the whole signal. Therefore local features can get lost and, if signal is not stationary (features change with time or in space) then this is not captured by FT. On the contrary, the wavelet transform can provides frequency information locally [9]. Wavelets have found beneficial applicability in various aspects of wireless communication systems design including channel modeling, transceiver design, data representation, data compression, source and channel coding, interference mitigation, signal de-noising and energy efficient networking [14].

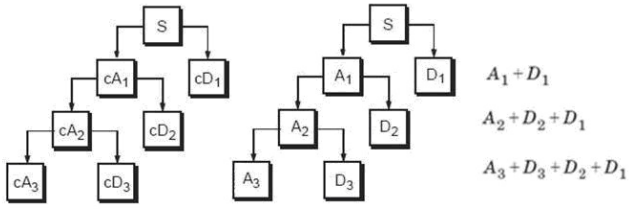
The DWT (Discrete Wavelet Transform) analyzes the signal at different frequency bands with different resolutions by decomposing the signal into an approximation containing coarse and detailed information. DWT employs two sets of functions, known as scaling and wavelet functions, which are associated with low pass and high pass filters. The decomposition of the signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal. The original signal $S[n]$ is first passed through a half-band high pass filter $g[n]$ and a half-band low pass filter $h[n]$. A half-band low pass filter removes all frequencies that are above half of the highest frequency, while a half-band high pass filter removes all frequencies that are below half of the highest frequency of the signal. The low pass filtering halves the resolution, but leaves the scale unchanged. The signal is then sub-sampled by two since half of the number of samples is redundant, according to the Nyquist's rule [14].

When the DWT is used to analyze periodic and almost sinusoidal waveforms, firstly the original waveform S is decomposed in approximations and details. Afterward successive decompositions of the approximations are made, with no further decomposition of the details (Fig. 1), based on an unbalanced tree [5]. When sets of almost random data corresponding to adjacent equally spaced frequency bands are analyzed, the wavelet packet transform is used. It is just like the wavelet transform except that it decomposes even the high frequency bands which are kept intact in the wavelet transform, using a balanced tree [12].

As Wavelets are widely applied in many scientific areas, huge efforts were done to deduce better, faster algorithms. Unfortunately, these efforts are disjointed among several disciplines [15], too few researches being interested (as this paper is) in multidisciplinary applications. Because a lot of large scientific problems require adaptivity, a collaborative framework must be developed, this involving mathematical techniques, computational methods and software [15].

A very difficult problem related to wavelet analysis is the providing of "exact reconstruction" feature, an extended study on reconstruction errors being made in [16], but no general solutions could be found.

Some of our research activities were recently dedicated to the DWT analysis of waveforms from power applications, in order to get more accurate and faster diagnosis/ evaluation methods employing specialized functions from MATLB that implement DWT [5], [6], or using our original algorithms for filters of length 4 - [8]. Our previous efforts were not concerned yet with the "exact reconstruction" property of the studied algorithms. A step forward is made with this paper, presenting our newly implemented algorithms with filters of length 6 dedicated to (non)stationary regimes. We also introduce our original hybrid algorithms, able to provide



(a) (b)

Figure 1. Signal decomposition in approximations and details (a) and its re-composition (b).

 ISBN: 978-1-61208-199-1

supplementary data, enabling our newly conceived original reconstruction functions to provide a "fast perfect reconstruction property". New research interests were revealed for us with this paper, as it contains our first studies on modern wavelet-based transmission techniques.

After a short introduction, we dedicated a section to the operational context in which we conceived and tested our algorithms and Matlab specialized functions and we decided what DWT technique is more convenient. Few mathematical fundaments are provided in Section III, to explain how the filters from our algorithms were calculated. Details on our direct and inverse algorithms with filters of length 4 or 6, using (or not) the interpolation to evaluate the missing right-end components are provided by Section IV, along with the presentation of some reconstruction-related problems. Section V is dedicated to our original hybrid algorithms, which are usable in all regimes and make possible the "exact reconstruction" property in difficult contexts. A comparative study is made relative to our algorithms' usability for the simulation and implementing of an orthogonal frequency division multiplexing system in Section VI. In Section VII, metrics are provided, that reveal the superiority of our algorithms. Conclusions and directions for our future work are presented in the end.

II. OPERATIONAL CONTEXT

The first application was implemented on a desktop (with the processor frequency of 2.4 GHz and 2GB of RAM) running Matlab vers. 7.1 under Windows XP, personalized to run for „best performances". Our data acquisition system supplies 560 samples/period per channel. The studied signal (Fig. 2), representing a phase current has distortions and is seriously affected beginning with its 5-th period.

The power quality indices considered to perform a comparison between various calculation methods are: the "node-zero" current (I_{j0}), the "non-zero node" current (I_{jn}) and the current's RMS value (I_{ef}) respectively. For an umbalanced tree with j levels and a currently analyzed number of 2^N samples, denoting by $a^{(0)}$ the approximation vector from the ultimate decomposition level (e.g. cA_3 in Fig. 1) and by $d_l^{(n)}$ the detail vector for the level l and a non-zero node (e.g. cD_i in Fig. 1), the following expression for the current's RMS value was used [11]:

$$I_{ef} = \sqrt{\frac{1}{2^N}\sum_{k=0}^{2^{N-j+1}-1}\left(a^{(0)}(k)\right)^2 + \frac{1}{2^N}\sum_{l=1}^{j}\sum_{k=0}^{2^{N-l+1}-1}\left(d_l^{(n)}(k)\right)^2} = \quad (1)$$
$$= \sqrt{I_{j0}^{\,2} + I_{jn}^{\,2}}$$

I_{j0} denotes the RMS value for the band with the lowest frequency j_0. I_{jn} represent the sets of RMS values for higher frequency bands.

The results depicted by Fig. 3 were obtained with the following methods: our original functions dwm4 and dw4i [8] implementing a DWT algorithm where different assumptions are made relative to the right border of the signal with finite length (as described in Section IV), dwt (from Matlab), considering all the values for the option "dwtmode"

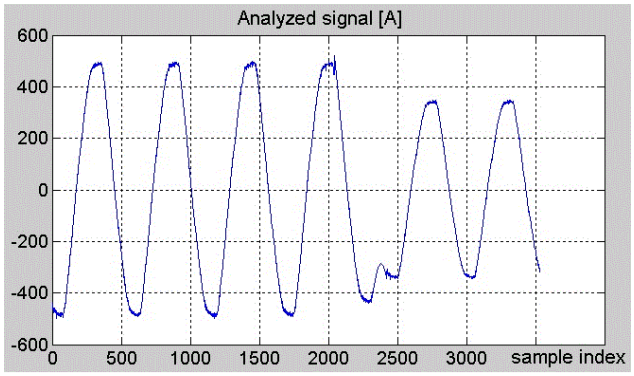


Figure 2. The analyzed signal.

that provide distinct treatments of the "finite signal's boundaries problem" [7], called with the same 4-length filter as that used by dwm4 and dw4i respectively (see Section III). The calculated filter's coefficients were found as being identical to those used by default by Matlab when dwt is called with the parameter "db2".

The analysis made with the Fast Fourier Transform (FFT) of the considered signal revealed differences lower then 1% for the power quality indices calculated with dwm4. It is why a dotted horisontal line was placed in Fig. 3 to represent the value calculated with dwm4 at the figures for stationary regime. For similar reasons dots were placed for the nonstationary regime following the value yielded by dw4i.

We applied a "shortening" procedure to the decomposition vectors obtained with dwt, when "dwtmode" was set to any of the values from the set: {"sym", "symw", "asym", "asymw", "zpd", "spd", "sp0" , "ppd", "per"}, as the internal algorithm of dwt artificially adds new components to vectors resulted from the DWT decomposition trying to rich the "exact reconstruction" and "global power preservation" properties.

We calculated the relative differences between the power quality indices with:

$$\begin{aligned} diff\_s &= (val\_s - val\_dwm4\_s)/val\_dwm4\_s\, x100; \\ diff\_n &= (val\_n - val\_dw4i\_n)/val\_dw4i\_n\, x100 \end{aligned} \quad (2)$$

where *"n"* or "*s"* denotes "nonstationary" or "stationary".

Their analysis revealed that in stationary regime, good results are provided by dwm4 and dwt with the following options for dwtmode: "ppd", "per", "asymw" and "spd". On the other hand, in nonstationary regime, dw4i is a good option while dwt can be called assigning to dwtmode the value:
- "asym" if errors of over 20% are accepted for I_{jn};
- "asymw" if errors over 10% are accepted ;
- "zpd" , as in the case of "asym", but it is not recommended in applications where more consecutive segments are analyzed to assume that the currently analyzed segment is zero beyond its borders.

The explanation for the high differences between the calculated power quality indices consists in the way of treating the "edge effect", as depicted by Figures 4 and 5. The arrows are used to mark the incorrect handling of the borders, that not only affects the calculated indices, but results also into the detection of "fake faults".

 ISBN: 978-1-61208-199-1

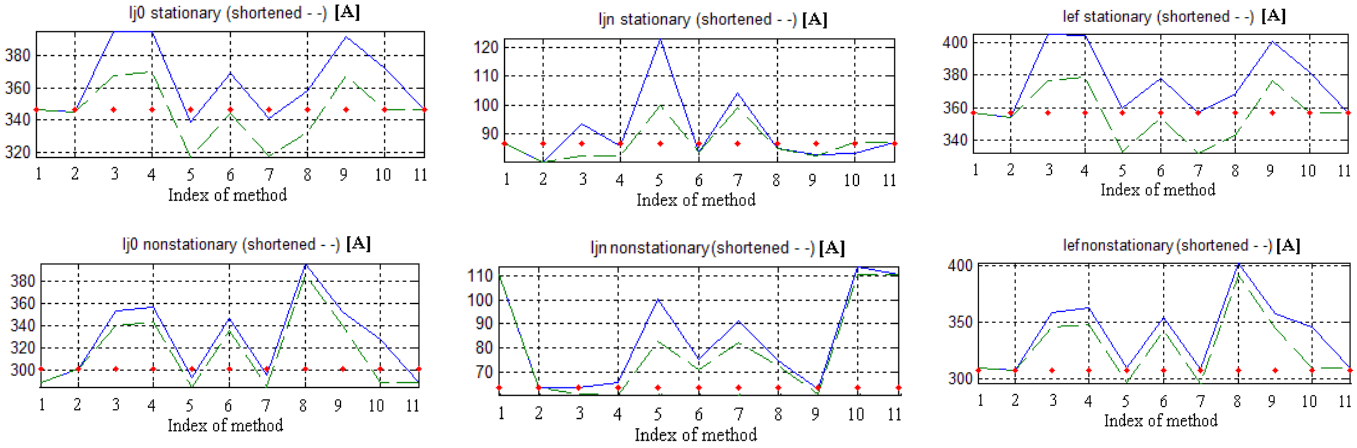


Figure 3. Power quality indices calculated with various methods. Indices for methods: 1 – dwm4; 2 – dw4i; 3…11 , dwt called with different values of dwtmode: 3- 'sym'; 4 -'symw'; 5 - 'asym'; 6 – 'asymw'; 7 – 'zpd'; 8 – 'spd'; 9 – 'sp0'; 10- 'ppd'; 11 – 'per'.

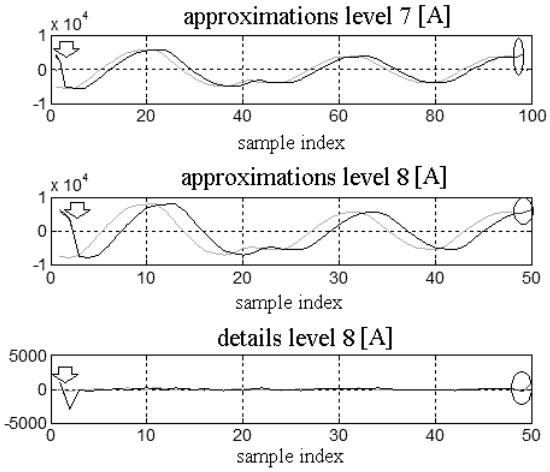


Figure 4. Nonstationary, dwt called with "ppd".

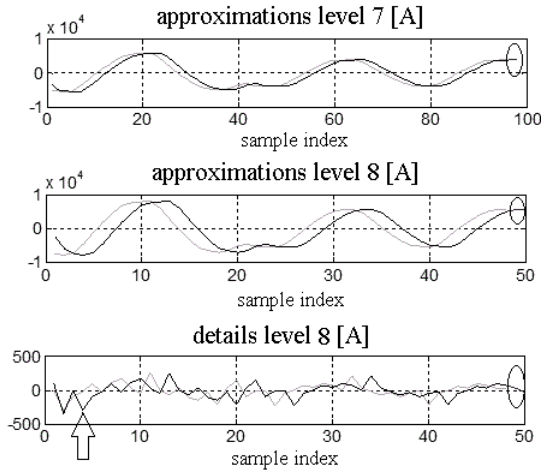


Figure 5. Nonstationary, dwt called with "asymw".

The darker waveforms correspond to the use of dwt whilst the lighter ones correspond to dw4i. The ellipses surround the components introduced artificially by dwt.

III. MATHEMATICAL FUNDAMENTS

A major problem in the development of wavelets was the search for scaling functions that are compactly supported, orthogonal, and continuous. Efforts were made to find the low-pass filter h, or equivalently, the Fourier series $H(\omega)=\sum_{k=0}^{N} h_k e^{-ik\omega}$. This trig polynomial $H(\omega)$ is often called the symbol of scaling function ϕ, which has support on $[0,N]$ [1], [4]. To ensure orthogonality, H must satisfy

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 = 1,\ \omega \in \Re. \qquad (3)$$

This is true for both DWT and the continuous settings. To create a good lowpass filter, $H(\omega)$ must have vanishing derivatives at $\omega=\pi$, which forces the graph of $|H(\omega)|$ to be flat near $\omega=\pi$. Starting from the statement: „if $H(\omega)$ is the Fourier series of a low pass filter, then $G(\pi)=H(\omega+\pi)$ is the Fourier series of a high pass filter" , considering that the orthogonality constraints of G are the same as those imposed to H, and supplementary imposing that $H'(\pi)=0$ (condition introduced by Daubechy to flatten more H), in [1] and [3], the values for the filters h and g are calculated as :

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}};\ h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}};\ h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}};\ h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}; \qquad (4)$$

$$g_0 = h_3;\ g_1 = -h_2;\ g_2 = h_1;\ g_3 = -h_0. \qquad (5)$$

The filters used to reconstruct the signal s from the approximation/detail vectors obtained with the filters h and g can be expressed as [10]:

$$lh_0 = h_2;\ lh_1 = g_2;\ lh_2 = h_0;\ lh_3 = g_0; \qquad (6)$$

$$lg_0 = h_3;\ lg_1 = g_3;\ lg_2 = h_1;\ lg_3 = g_1. \qquad (7)$$

For a filter of length 6, the orthogonality conditions are [3]:

$$\sum_{k=0}^{5} h^2{}_k = 1; h_0h_2 + h_1h_3 + h_2h_4 + h_3h_5 = 0; h_0h_4 + h_1h_5 = 0 \cdot \quad (8)$$

The low-pass conditions are:

$$h_0 + h_1 + h_2 + h_3 + h_4 + h_5 = 1; h_0 - h_1 + h_2 - h_3 + h_4 - h_5 = 0. \quad (9)$$

Two derivative-related conditions are imposed now:

$$H'(\pi) = 0;\ H''(\pi) = 0 \cdot \quad (10)$$

Considering the above, the numerical values of the components from h were calculated as being [1]: h_0=0.3327; h_1=0.8069; h_2=0.460; h_3= - 0.135 ; h_4= - 0.085; h_5=0.035.

For the algorithm used in this paper, the high-pass filter g is selected as: $g = (h_5, -h_4, h_3, -h_2, h_1, -h_0)$, another possible solution being $(h_5, h_4, h_3, h_2, h_1, h_0)$ [3].

For the signal reconstruction, one can use:

$$lh_0 = h_4;\ lh_1 = g_4;\ lh_2 = h_2;\ lh_3 = g_2;\ lh_4 = h_0;\ lh_5 = g_0;$$
$$lg_0 = h_5;\ lg_1 = g_5;\ lg_2 = h_3;\ lg_3 = g_3;\ lg_4 = h_1;\ lg_5 = g_1. \quad (11)$$

In general, for a filter of even length $h=(h_0, \ldots h_L)$, the general system (GS) to be solved relies on $(L+1)/2$ orthogonality conditions, the conditions $H(0)=\sqrt{2}$, $H(\pi)=0$, and $(L-1)/2$ derivative conditions $H^{(m)}(\pi)=0$ [3].

IV. ALGORITHMS

A. Distinct algorithms for stationary and nonstationary waves

The algorithm used for the determination of approximation/detail vectors when performing a DWT decomposition (filter of length 4) of a signal s relies on [10]:

$$a_i = s(i) * h_0 + s(i+1) * h_1 + s(i+2) * h_2 + s(i+3) * h_3;$$
$$d_i = s(i) * g_0 + s(i+1) * g_1 + s(i+2) * g_2 + s(i+3) * g_3 \quad (12)$$

where the vector s contains the signal's discrete values, a and d denote the approximation/detail vectors, obtained when s is decomposed using the low-pass filter $h=(h_0\ h_1\ h_2\ h_3)$ and the high-pass filter $g=(g_0\ g_1\ g_2\ g_3)$ respectively.

When a filter of length 6 is used, two additional terms (corresponding to the last two components of the filters) appear at both a_i and d_i respectively:

$$a_i = \ldots + s(i+4) * h_4 + s(i+5) * h_5;$$
$$d_i = \ldots + s(i+4) * g_4 + s(i+5) * g_5. \quad (13)$$

For stationary waves, one can consider that the missing values beyond the right edge of the currently analysed data segment are equal to those from the left edge. So:
- for dwm4: $s(n+1) = s(1)$ and $s(n+2)= s(2)$;
- for dwm6 (our function with noninterpolated vectors and filter of length 6): $s(n+i)=s(i)$, with $i=1\ldots4$.

For nonstationary waves, our functions implement spline interpolation to evaluate the needed "$s(n+i)$" components using the rightmost components of s, as follows:
- for dw4i:
yi=spline([*n*-2 *n*-1 *n*], [*s*(*n*-2) *s*(*n*-1) *s*(*n*)],[*n*-2 *n*-1 *n n*+1 *n*+2]) . *s*(*n*+1) will be evaluated as *yi*(4) and *s*(*n*+2) will be evaluated as *yi*(5);
- for dw6i (our function with interpolated vectors and filter of length 6):
xi=spline([*n*-2 *n*-1 *n*], [*s*(*n*-2) *s*(*n*-1) *s*(*n*)],[*n*-2 *n*-1 *n n*+1 *n*+2]);
yi=spline([*n n*+1 *n*+2], [*s*(*n*) *xi*(4) *xi*(5)],[*n n*+1 *n*+2 *n*+3 *n*+4]).
s(*n*+*i*) will be evaluated as *yi*(1+*i*) for *i*=1…4.

B. Reconstruction algorithms

Using our functions id4m (or id6m), the reconstruction of the original signal s when its decomposition was done with dw4m (or dw6m) generates "almost zero" reconstruction errors, as no interpolations were involved (Fig. 6 and 7).

In the reconstruction algorithm corresponding to id4m, *temp* denotes the reconstructed signal (which is either the approximation vector from the level k, or the original signal, if $k=1$), a and d are the approximation/detail vectors from level $k+1$ and *lh*/*lg* are the low/high pass reconstruction filters. Starting from $j=3$ (the first 2 components have to be evaluated differently, because they use components with indices below the vectors' left boundaries) sets as instructions as those reproduced below are executed:

*temp(j)=a(i)*lh0+d(i)*lh1+a(i+1)*lh2+d(i+1)*lh3; j=j+1;*
*temp(j)=a(i)*lg0+d(i)*lg1+a(i+1)*lg2+d(i+1)*lg3.*

The left-most 2 components are calculated considering that *a(0)=a(length(a))* and *d(0)=d(length(d))*.

A similar algorithm is used by *idm6*, but two additional terms appear in each of the instructions (for the first line: *a(i+2)*lh4 + d(i+2)*lh5* and for the second line *a(i+2)*lg4 + d(i+2)*lg5*) .

Firstly, we conceived two functions in order to reconstruct the original signal from its corresponding approximation/detail vectors obtained with dw4i or dw6i. The missing components to the left were evaluated through spline interpolations. Unacceptable errors were obtained near the signal's left border (errors were accumulated during 2 interpolation steps - one to the right followed by another to the left). Additional operations for errors' removing should make the algorithms unusable for time-critical applications.

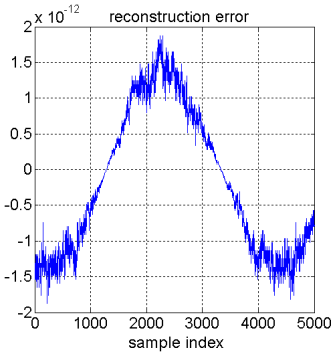


Figure 6. Stationary. Reconstruction error generated by id4m.

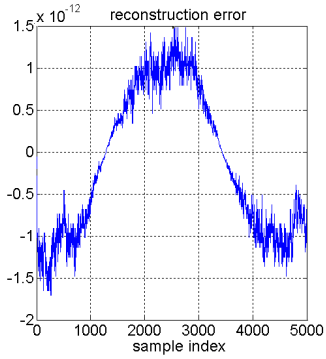


Figure 7. Nonstationary. Reconstruction error generated by id4m.

Then we applied id4m (or id6m) to reconstruct the signal from its corresponding approximation/detail vectors obtained with dw4i or dw6i. The errors were diminished (as the second interpolation was no longer performed), but still were not driven up to an acceptable level. On the other hand, when idwt (function provided by Matlab) was used to reconstruct the signal previously decomposed with dwt (providing the same value for the option dwtmode both to dwt and to idwt respectively), acceptable reconstruction errors (at most 0.1 in absolute value) were obtained for all regimes and values for the option dwtmode (Fig. 8). An exception was noticed when using the option dwtmode="per" (Fig. 9).

V. A HYBRID ALGORITHM

The reconstruction errors generated by our "idm like" functions (when applied on interpolated vectors) have something in common: they are unacceptable high for the first components of the re-constructed signal and are almost 0 afterward (order of magnitude 10^{-12}). This behavior is related to the indices of the components affected by interpolations. From Figs. 10 and 11 one can see that, except for their last components, the approximation/detail vectors (denoted by *A* and *D*) are calculated using only components from the decomposed signal (*Y*), which were never submitted to any interpolating process. For example, when d4h is used, *A*(1) and *D*(1) are calculated using only *Y*(*i*), *i*=1…4. The components affected by interpolation (either beginning with the current decomposition level or because they are calculated based on components affected by interpolations made at previous decomposition levels) are marked with *.

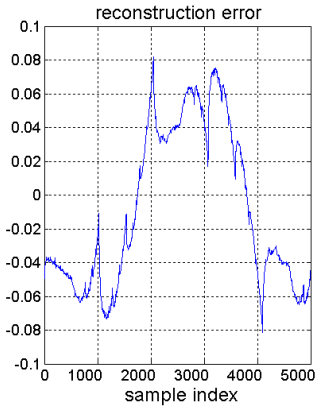


Figure 8. Filter of length 4. Reconstruction error, nonstationary regime, dwtmode="asymw".

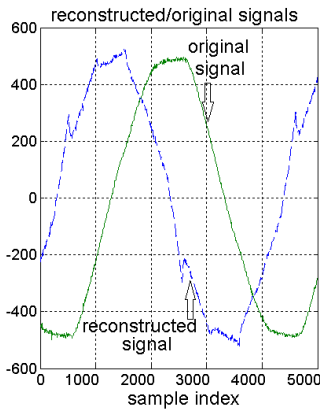


Figure 9. Filter of length 4. Reconstructed and original signals, stationary regime, dwtmode="per".

Y (1)	Y (2)	Y (3)	Y (4)	Y (5)	Y (6)	…	Y (n-3)	Y (n-2)	Y (n-1)	Y (n)	Y (1)	Y (2)
									Y* (n-1)	Y* (n)	Y* (1)	Y* (2)
A(1), D(1)												
		A(2), D(2)										
						…						
							A(n/2-1), D(n/2-1)					
							A*(n/2-1), D*(n/2-1)					
									A(n/2), D(n/2)			
									A*(n/2), D*(n/2)			

Figure 10. Conceptual schema for d4h.

…	Y (n-7)	Y (n-6)	Y (n-5)	Y (n-4)	Y (n-3)	Y (n-2)	Y (n-1)	Y (n)	Y (1)	Y (2)	Y (3)	Y (4)
					Y* (n-3)	Y* (n-2)	Y* (n-1)	Y* (n)	Y* (1)	Y* (2)	Y* (3)	Y* (4)
	A(n/2-3),D(n/2-3)											
	A*(n/2-3), D*(n/2-3)											
			A(n/2-2), D(n/2-2)									
			A*(n/2-2), D*(n/2-2)									
					A(n/2-1), D(n/2-1)							
					A*(n/2-1), D*(n/2-1)							
							A(n/2), D(n/2)					
							A*(n/2), D*(n/2)					

Figure 11. Conceptual schema for d6h.

Therefore, we conceived and tested two hybrid algorithms: d4h and d6h. d4h will be called with an instruction like: *[Cm a d]=d4h(y,n,v)*. The input parameters are: the noninterpolated version of the decomposed signal *y*, its length *n* and a vector *v* containing the last 2 components from the interpolated version of *y* (except for the first level of decomposition). The result of the decomposition performed with d4h is the vector *Cm*, whose first half represents the approximation vector and second half represents the detail vector (noninterpolated versions). The components of approximation and details affected by interpolation in the current level are provided as the vectors *a* and *d*, both having 2 components.

When a d4h-based decomposition is done for all the 10 levels, the following structures are calculated:

- 10 approximation vectors and 10 detail vectors (the noninterpolated versions). For example *cAm_k* is the approximation vector and *cDm_k* is the detail vector for the level *k* ;
- the matrix containing the final components of the approximation vectors affected by interpolation *a(2 x 10)*. For example *a(1,7)* and *a(2,7)* can be used to get the interpolated version of *cAm_7*, using two simple instructions:

cAm_7(n-1)=a(1,7); cAm_7(n)=a(2,7);

- the matrix containing the final components of the detail vectors affected by interpolation *d(2,10)*.

In a similar manner, d6h will be called with an instruction like: *[Cm a d]=d6h(y,n,v)*, but *v, a* and *d* will have 4 components instead of 2.

For the reconstruction, the noninterpolated versions of the approximation/detail vectors are used as input data to the functions that provide the signal reconstruction (id4m and id6m), according to a sequence of calls as follows:

*cAmr_9=id4m(cAm_10,cDm_10, 2*length(cAm_10));*
*cAmr_8=id4m(cAmr_9,cDm_9, 2*length(cAm_9));*
…
*cAmr_1= id4m(cAmr_2,cDm_2, 2*length(cAm_2));*
y_reconstructed=id4m(CAmr_1,cDm_1).

VI. APPLYING THE ALGORITHMS FOR RANDOM SETS OF DATA IN COMMUNICATIONS

Conventional OFDM (Orthogonal Frequency Division Multiplexing) systems use (I)FFT to multiplex the signals and transmit them simultaneously over a number of subcarriers. These systems employ guard intervals or cyclic prefixes (CP) so that the delay spread of the channel becomes longer than the channel impulse response. CP reduces the power efficiency, data throughput and the spectral containment of the channels.

An alternative method is known as „DWT-OFDM", that uses the Discrete Wavelet Transform to replace the IFFT and FFT blocks, and provides a better spectral containment of the channels as CP is no longer provided [12]. A typical block diagram of an OFDM system is shown in Fig. 12, where the inverse and forward transform blocks can be FFT-based or DWT-based OFDM [13].

For a DWT-OFDM system, the binary data *d* is firstly processed by a constellation mapping. A common solution for the mapping of *d* into OFDM symbols is the 16 QAM digital modulator (DM), which yields OFDM complex symbols X_m.

The analyzed scenario considers 64 channels of binary data, processed by DM in sets of 996 bits that are mapped into 166 complex symbols per set. We used the Matlab function „qammod" to obtain the symbols, that can have integer real and imaginary parts, belonging to the set {-7, -5, -3, -1, 1, 3, 5, 7}.

To obtain the time representations of both signals to be assembled in order to get the signal *y* (one for the real and the other for the imaginary parts), the 64 vectors (one per channel) formed from the corresponding symbols' real/imaginary parts were used as bottom level nodes of a binary tree. Fig. 13 depicts a 3-level tree. The number of levels for our tree was calculated as $\log_2(64)=6$. The tree can be used to apply either the inverse DWT transform („bottom-up") or the forward DWT transform („up-down"), because each channel provides data with a distinct frequency range and the band width is entirely covered by the channels' adjacent distinct frequency ranges.

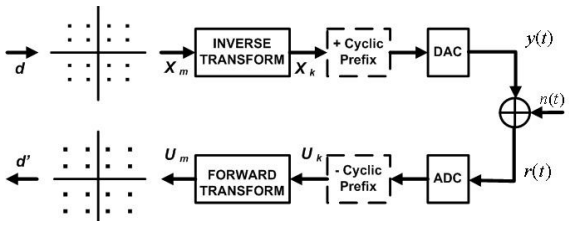


Figure 12. Schematic of an OFDM transceiver

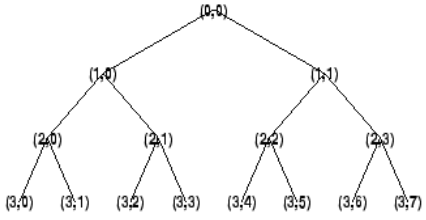


Figure 13. Example of a tree used to apply the inverse and forward DWT transforms

We firstly employed our function id4m. For example if we denote by v_{ij} the data from a node, then *v21=id4m(v32,v33,length(v32))*.

Fig. 14 depicts the first 50 discrete time values (from a total of 10624) of the signal y_r obtained from the real components , the processed data relying on 64x996 randomly generated binary data.

On the receiver side, the received data are submitted to a forward transform. The same tree can be used, with instructions like: *[v32 v33]=d4m(v21, length(v21))*.

Finally, the vectors obtained after the applying of the forward transform on the first level („received data") were compared to the data that were submitted to the inverse transform („transmitted data"). Our algorithms provided the „exact reconstruction" property. Maximum absolute differences of 10^{-13} were obtained between the transmitted and received data for all channels.

Following the same technique, but using the Matlab functions idwt, respectively dwt with the same filters and all values for the option dwtmode, considerable differences were revealed between the transmitted and received data (see the maximal differences in Fig. 15 and arrows in Fig. 16). Similar results were obtained for the imaginary components.

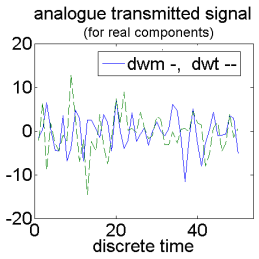


Fig. 14. The first values of the signal obtained from real components

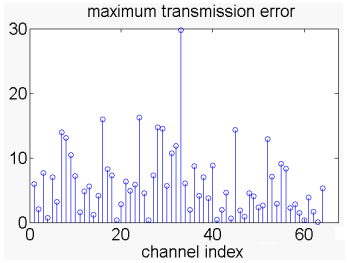


Fig. 15. Maximum transmission errors per channel (real components) when idwt/dwt were used (dwtmode='asymw')

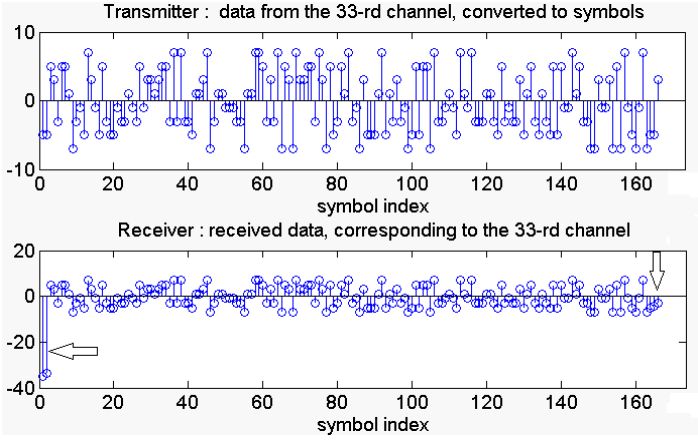


Fig. 16. Symbols corresponding to transmitted data ,real components, from the 33-rd channel (up) and received data (down), when idwt/dwt were used (dwtmode='asymw')

VII. METRICS

When an n – sized vector is analyzed with our hybrid algorithms using an unbalanced tree with l levels, the number of memory locations (*NML*) required to store the vectors yielded by decomposition and to restore the original signal can be calculated as the sum of :

- length of the approximation vector from the last level = $n/2^l$;
- sum of all details vectors' lengths $\sum_{1}^{l} n/2^i$;
- *2x2x l* (for d4h) or *2x4x l* (for d6h) = *NML* from the arrays used to store the values resulted through interpolations required for the estimated values from the right edge.

For a d4h-based decomposition in l=l0 levels, *NML*=*n*+*40*. For 3 periods (Fig. 2), n=12288, but n is usually larger, as usually more than 16 periods are analyzed. Therefore the additional memory requirements involved by the storing of the decomposition vectors required for the analyzed vector reconstruction are negligible as compared to the case when the analyzed vector is stored. The storing of decomposition vectors is preferred instead the original signal's storing, as the procedure to "exactly" reconstruct the original signal requires a runtime that is significantly smaller (25% from that required to determine the decomposition vectors with hybrid algorithms). Table I presents mean decomposition runtimes in this scenario.

At stationary regimes, dwm performs almost 2 times faster than dwt used with "asymw", with less memory consumption. idwt operates correctly only over "un-shortened" vectors, this making it unusable for the evaluation of power quality indices. At nonstationary regimes, dwi is slower than dwt used with "asymw", but provides accurate power quality indices and correct fault detection, with no "boundary effects". When no apriori information is available relative to the (non) stationary nature, it is indicated to use dwh as it provides vectors for all regimes (stationary or not), along with the data required to "exactly" reconstruct the signal.

In the communication application, (i)dwm and (i)dwt exhibited comparable runtimes and memory consumptions, but d4m and id4m are better options, as only they provided "exact reconstructions".

VIII. CONCLUSION AND FUTURE WORK

The Matlab toolkit for DWT analysis is not always the best option for a proper analysis of waveforms from power systems. It might introduce artificial energies of decomposition vectors, unacceptably revealed in the values of indices used for power quality analysis, supplementary memory consumption, detection of fake faults, supplementary run-times, conditioned recovery procedures (providing very poor results for the option "dwtmode"= "per"). Our algorithms do not generate longer decomposition vectors, detect correctly the moment when a fault occurs, can calculate the RMS value in nonstationary regimes for which FFT is hard to apply in real-time restrictions and exhibited good run-times in the analyzed scenarios. They can be used both in stationary and nonstationary regimes, providing a fast "exact reconstruction" method.

In communication applications, when data submitted to analysis have a random nature and can take values from a restricted set of integer values with a relative modest variation, the algorithms d4m and id4m are very good options, providing "exact reconstructions" and good runtimes, whilst the Matlab functions dwt/idwt exhibited unacceptable transmission errors.

Our future work will focus on the exploring the abilities of our algorithms in more applicability domains and their adaptation to artificial intelligence techniques.

TABLE I. MEAN RUNIMES FOR 1500 DECOMPOSITION OF 3 PERIODS

Filter length	Mean runtimes [sec]					
	asymw	dwm	dwi	dwh	idwt	reconstruction (using data provided by hybrid algorithms)
4	0.0024	0.0011	0.0031	0.0056	0.0030	0.0014
6	0.0028	0.0015	0.0054	0.0078	0.0036	0.0019

ACKNOWLEDGMENT

The work was supported by the Romanian Programe PN II "Partnership in Priority Domains", the grant "SECENGES".

REFERENCES

[1] P. Van Fleet, Discrete Wavelet Transformations: An Elemententary Approach with Applications, Wiley-Interscience, 2009.

[2] D. Percival and A. Walden, Wavelet Methods for Time Series Analysis, Cambridge , Cambridge University Press, 2006 .

[3] V.F. Patrick, Daubechies Filters, PREP 2006, Wavelet Workshop, 2006, available on line at http://cam.mathlab.stthomas.edu/ wavelets/pdffiles/UST06/Lecture6.pdf <retrieved: March, 2012>.

[4] C. Bénéteau, Haddad, C., D. Ruch and P. Van Fleet, Classical Theory and Daubechies Waveletes, PREP 2008, Wavelet Workshop, 2008, available on line at http://cam.mathlab.stthomas.edu / wavelets/pdffiles/UST08/Lecture7.pdf, <retrieved: March, 2012>.

[5] I.D. Nicolae and P.M. Nicolae, Using Wavelet transform for the evaluation of power quality in distorting regimes, Acta Electrotechnica, vol. 52, no .5, pp. 331-338, 2011.

[6] I.D. Nicolae and P.M. Nicolae, Using discrete Wavelet transform to evaluate power quality at highly distorted three-phase systems, Proceed. of the 11-th Int. Conf. on Electrical Power Quality and Utilization (EPQU 11), pp: 1 – 6, 17-19 Oct. 2011, Lisboa, Portugal, doi: 10.1109/ EPQU.2011. 6128825.

[7] http://www.mathworks.com /help/toolbox/wavelet/ug/f8-25097.html, <retrieved: March, 2012>.

[8] I.D. Nicolae and P.M. Nicolae, Real-time analysis using Discrete Wavelet Transform in power systems, EPE-PEMC 2012, in press.

[9] R. Ashis, Transforms, Fourier and Wavelets, available at http://www.cs.cornell.edu/ courses/cs5540/2010sp/lectures/Lec5.Transforms.pdf , <retrieved: March, 2012>.

[10] I. Kaplan, The Daubechies D4 Wavelet Transform, available at http://www.bearcave.com/software/java/wavelets/daubechies/index.htm, <retrieved: March, 2012>.

[11] Morsi, W.G. and El-Hawary, M.E., Wavelet Packet Transform-Based Power Quality Indices for Balanced ad Unbalanced Three-Phase Syst. under Stationary and Nonstationary Operating Conditions, IEEE Trans. on Power Delivery, vol. 24, no. 4, pp. 2300-2310, 2009.

[12] R. Dilmirghani and M. Ghavami, „Wavelet Vs Fourier Based UWB Systems", 18t-h IEEE Intern. Symposium on Personal, Indoor and Mobile Radio Communications, pp.1-5, 2007.

[13] S. Baig, F.Farrukh and M. J. Mughal, Discrete Wavelet Transforms - Algorithms and Applications, Intech, 2011.

[14] M. K. Lakshmanan and H. Nikookar, A Review of Wavelets for Digital Wireless Comm., Wireless Personal Communications, vol. 37, No. 3-4, 387-420, DOI: 10.1007/s11277-006-9077-y.

[15] G. Latu, Data Structure Design and Algorithms for Wavelet-Based Applications, available at http://icps.u-strasbg.fr/people/latu/ public_html/wavelet/course_slide.pdf, 2010, <retrieved: March, 2012>.

[16] T. K. Sarkar, M. Salazar-Palma and M.C. Wicks, Wavelet Applications in Engineering Electromagnetics, Artech House, 2002.

 ISBN: 978-1-61208-199-1