Annals of the University of Craiova, Electrical Engineering series, No. 38, 2014; ISSN 1842-4805

Improving Efficiency of DWT Analysis through Faster Interpolation Methods and Multithreading Techniques

Ileana-Diana Nicolae[*], Petre-Marian Nicolae[†], Marian-Ştefan Nicolae[†] and Andreea Chiva[*]
[*]Department of Computers and Information Technology, nicolae_ileana@software.ucv.ro, chivaandreea@gmail.com
[†]Department of Electrical, Energetic and Aero-Spatial Engineering, pnicolae@elth.ucv.ro, snicolae@elth.ucv.ro
University of Craiova, Craiova, Romania

Abstract— **The paper deals with techniques to accelerate the DWT power quality analysis. A brief description of a non-spline interpolation schematic is provided firstly. It is used for the evaluation of the first components near the right border of the signal. Comparisons were made between power quality indices, shape of details waveforms and runtimes, considering data acquired on stand, corresponding to a driving system using chopper and DC motor and respectively waveforms acquired from the secondary winding of the excitation system used to supply a generator. FFT and 3 distinct original functions were used for comparison. Similar results were generated by all methods. The spline-free interpolation technique, which used mean slopes, proved to be a better option when the analyzed signal's periodicity is questionable, because it exhibited halved runtimes as compared to the spline interpolation technique. Two multi-threading techniques, relying on a horizontal and respectively on an interleaved schematic, were afterward analyzed considering their performances relative to the DWT decomposition and respectively re-composition of single-phase currents or voltages. The analyzed filters had different lengths: 4, 6 and 8. 10 sets of 300 consecutive decompositions of signals consisting in 36864 samples were used as input data. Our Java programs implementing the horizontal technique revealed the diminishing of runtimes for all analyzed filters during the decompositions by 16%...23%, the technique being more efficient for shorter filters. Another multi-threading technique, relying on an interleaving schematic, was considered for the reconstruction of signals. It proved to be non-effective, revealing increases of runtimes in the range 19%...23%. Similar results were obtained with the horizontal technique. Finally a multithreading technique, relying on a vertical schematic is presented. Using it, savings of 45% of runtimes were revealed by tests made on three-phase systems.**

I. INTRODUCTION

DWT 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 which are associated with low pass and high pass filters. The decomposition of the signal into different frequency bands is 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 $h[n]$. $h[n]$ removes all frequencies that are above half of the highest frequency, while $g[n]$ removes all frequencies that are below that frequency [1]. Afterward successive decompositions of the approximations are made, with no further decomposition of the details. Fig.1 depicts a 3 levels decomposition [2]. cA_i denotes the approximation vector whilst cD_i denotes the detail vector on the i-th level. The most significant frequencies from the original signal appear with high magnitudes in that specific region of the DWT signal including them, with the preservation of their time localization.

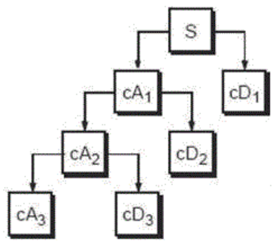

Fig.1. A decomposition on 4 levels using DWT.

Continuous efforts have been done to find more efficient methods for power quality analysis [3]-[6].

For the analysis of quasi-stationary regimes, corresponding to highly distorted waveforms with small variations of magnitude and harmonic contents in three-phase systems, DWT is a valuable tool. The Fast Fourier Transform (FFT) can be used for stationary regimes in order to provide "reference data" for the values obtained for power quality indices when DWT-based theories are used [7]…[9].

This paper represents an improved and substantially extended version of the paper [9].

II. ANALYZED DATA AND ANALYSIS METHODS

A. Analyzed data

The first 2 sets of analyzed data, acquired on stand using a data acquisition system with a sampling frequency of 19200 Hz, correspond to a driving system using chopper and DC motor. The first set corresponds to unfiltered currents. A partial representation is depicted by Fig. 2. The second set (partially reproduced by Fig. 3) contains currents with a reduced harmonic content and the corresponding voltages. The third set of data is defined by sets of currents and voltages acquired from the secondary winding of the excitation system used to supply the main generator from

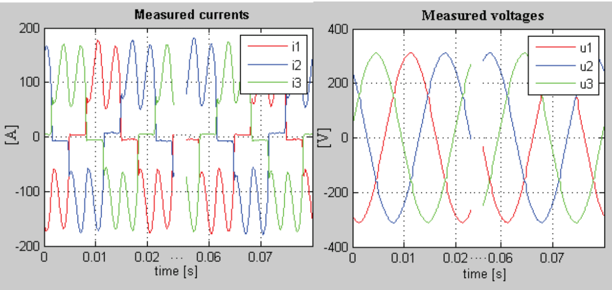

Fig. 2. First set of analyzed data (partial representation)

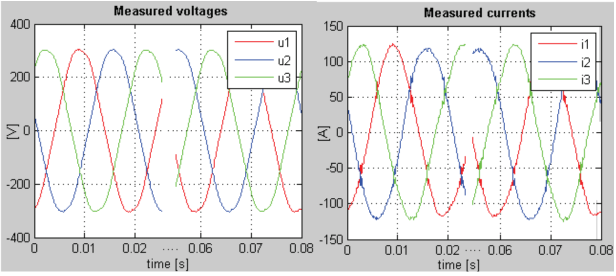

Fig. 3. Second set of analyzed data (partial representation).

a power plant. The first 200 samples of the analyzed three-phase systems of voltages and currents are depicted by Figs. 4 and 5 [9], [10].

B. Analysis methods

Firstly an original program, based on Fast Fourier Transform, was used to analyze all sets of data. Our previous studies [9],[10] were concerned with 10 levels DWT-based decompositions, using Daubechy filters of length 8, implemented as original functions called "dwm" and "dwi" respectively. dwm assumes that the analyzed signal is periodic, whilst dwi performs interpolations relying on

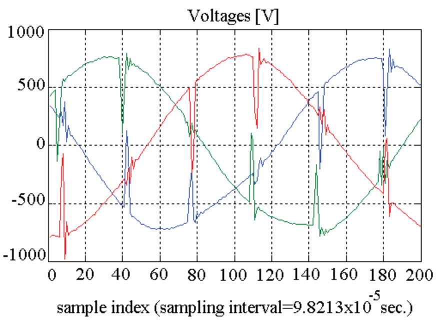

Fig. 4. 3rd set of data: voltages.

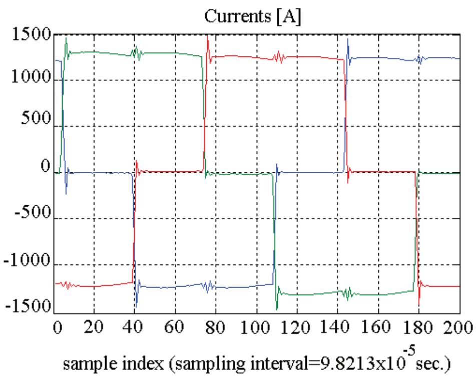

Fig. 5. 3rd set of data: currents.

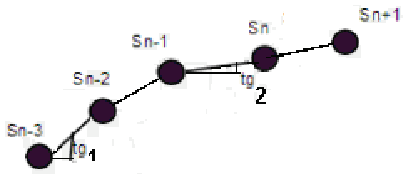

Fig. 6. Non-spline interpolation schematic.

spline techniques provided by the MATLAB library , being also applicable to non-periodic signals.

The filters' coefficients are calculated as described in [11], [12] and details on their implementation are given in [13].

The spline interpolations involve significant runtimes. This made us consider a different, non-spline, interpolation schematic in order to accelerate the decomposition process. This schematic [9],[13] is depicted by Fig. 6.

In this case, the estimation of s_{n+1} when it is separated by "*step*" seconds from s_n requires the following calculations:

$tg1=(s_{n-2}-s_{n-3})/step$;

$tg2=(s_n-s_{n-1})/step$;

$mean\_tg=(tg1+tg2)/2$;

$s_{n+1}= s_n+mean\_tg \times step$.

III. ACCELERATING DWT DECOMPOSITIONS WITH SPLINE-FREE INTERPOLATION SCHEMATICS

A. Qualitative results

The reference [9] provides in detail the values of node-zero, non-zero and RMS for all phase voltages and currents calculated for all sets of data using fft, dwm, dwi and dwns (the function which performs non-spline interpolations). The differences between the values calculated with dwm and respectively those calculated with fft were calculated with:

$$diff\_dwm = (val\_dwm - val\_fft) / val\_fft \cdot 100 \qquad (1)$$

Similar formulas were used for the values generated by dwi and dwns respectively.

Synthetic data are gathered by Table I [9]. They were gathered to provide a global picture over the percent differences between values calculated with both functions using interpolations and those calculated when no interpolation is performed. Each cell contains values mediated over all phases from the same set. The differences dwns versus dwm and respectively dwi versus dwm were found

TABLE I
SYNTHETIC DATA GENERATED AS AVERAGE OVER ALL PHASES FOR PERCENT RELATIVE DIFFERENCES BETWEEN VALUES GENERATED WITH DIFFERENT DWT BASED FUNCTIONS

Set index / Power Quality index	1st set		2nd set		3rd set	
	dwns vs dwm [%]	dwi vs dwm [%]	dwns vs dwm [%]	dwi vs dwm [%]	dwns vs dwm [%]	dwi vs dwm [%]
I_{ef}	0.05	0.06	0.04	0.07	-0.03	0.05
I_{j0}	4.64	4.93	3.83	5.53	0.07	0.05
I_n	0.05	0.06	0.06	0.07	-1.21	0.09
U_{ef}	0.40	-0.24	0.09	0.13	0.04	0.01
U_{j0}	-2.36	0.04	-1.00	-0.01	0.06	0.02
U_n	-0.01	-0.20	0.09	0.14	-0.56	0.00

to be very close, revealing no systematic pattern: positive, negative or even zero mean values of differences were obtained with no correlation to the phase number or type of power quality index [9],[10].

B. Analysis of waveforms shape

The quantitative evaluations are in good agreement with the conclusions drawn from the shape-related analysis of the decomposition vectors generated with different functions, as depicted by Figs. 7...10 [9]. In these figures cyan is used for waveforms generated by dwi, red is used for waveforms generated by dwns and dark blue is used for waveforms generated by dwm.

As one can see, no shape-related pattern could be derived relative neither to the decomposition level, nor to the phase or type of waveform (voltage versus current) [9].

C. Analysis of runtimes

300 consecutive decompositions over all 10 levels were performed for data from Figs. 2 and 3. The analysis of runtimes obtained in the programming environment MATLAB revealed mean runtimes of a single phase full

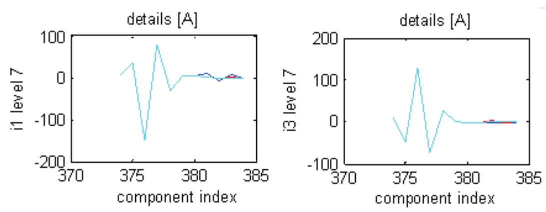

Fig. 7. 1st set of data. Almost identical details were generated by all functions, so almost zero differences are generated.

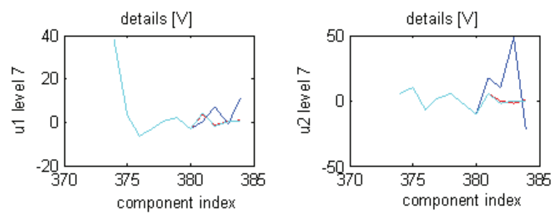

Fig. 8. 1st set of data. dwm generates higher values compared to dwi and dwns, which are almost identical.

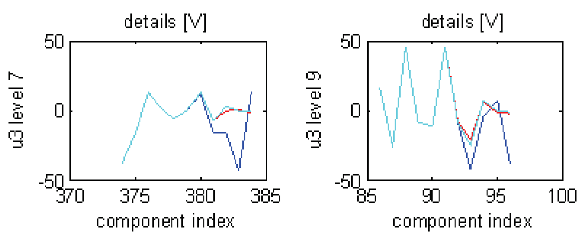

Fig. 9. 2nd set of data. dwm generates lower values compared to dwi and dwns, which are almost identical.

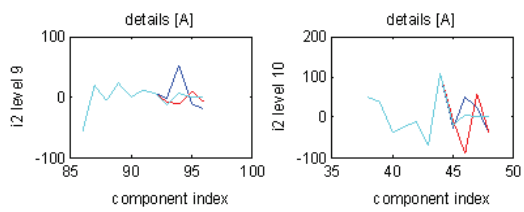

Fig. 10. 2nd set of data. No systematic rule can be deduced relative to differences between waveforms generated with different functions.

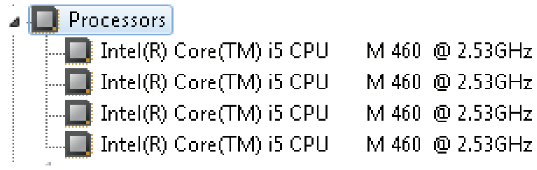

Fig. 11. Configuration of the employed machine.

decomposition equal to 3.85 msec when using dwns, of 7.67 msec when using dwi and respectively of 2.96 msec when using dwm [9]. The small runtimes and the results of shape and accuracy analysis allow us to recommend the use of the new function – dwns, as a faster alternative, providing a similar degree of accuracy both for non-stationary and quasi-stationary regimes respectively. The runtime required by dwns is reduced to a half as compared to the case when spline interpolations are used and is only approximately 30% higher than then that required by dwm, which is interpolation free [9].

IV. MULTITHREADING IMPLEMENTATIONS FOR THE DWT ANALYSIS OF A SINGLE PHASE

A. Horizontal schematics

When multi-threading techniques are used, the configuration of machine is crucial for the accomplished performances. The results presented in this paper were obtained on a computer with a processor which has the configuration from Fig. 11, with original Java programs. The available quantity of installed RAM memory was of 4 GB. Discouraging results were obtained when the same programs were executed on a dual core configuration.

When only a single phase must be analyzed at a time, a possibility to improve the execution's efficiency through the runtime diminishing resides in implementing a multi-threading technique based on a horizontal schematic. Fig. 12 depicts the operating principle for an implementation with two threads.

The main idea consists in dividing the vector to be decomposed (v) , whose length is l, in n parts, where n is the number of threads which will be used to perform in parallel the decomposition of v. Each thread will decompose l/n components.

All threads used for decomposition access in a synchronized manner a common variable *no_of_ended_threads*, used to store the number of threads which accomplished their task (decomposition of the vector's partitions assigned

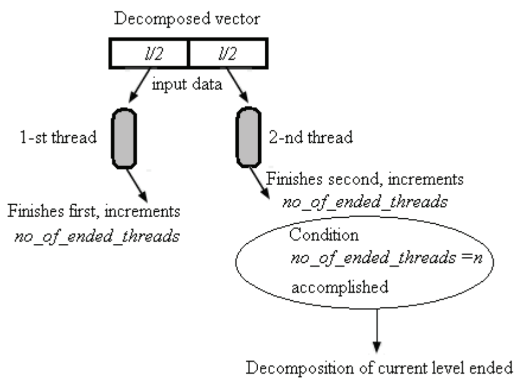

Fig. 12. Horizontal schematic used as principle by the multi-threading technique used during the decomposition for a single phase.

to them). This variable is incremented each time a thread ends the decomposition stage. Each time *no_of_ended_threads* is incremented, the condition *C1*, "*no_of_ended_threads =n*" is tested, to see if the whole vector was decomposed. If *C1* is true, the application's main thread is unblocked and the execution moves toward the following decomposition level.

B. Interleaved schematics

A good solution for saving memory in archives is to store only the vectors yielded by the DWT decomposition. If the original signal (used for their generation) is needed, it can be reconstructed considering the above mentioned vectors, using the inverse functions.

When the decomposition was made with a dwm function, the reconstruction relies on our inverse functions that use the following principle [13]:

$$\begin{aligned} &rec_j = a_i \cdot lh_0 + d_i \cdot lh_1 + \ldots + a_{i+1f-1} \cdot lh_{l-2} + d_{i+1f-1} \cdot lh_{lf-1}; \\ &j = j+1; \\ &rec_j = a_i \cdot lg_0 + d_i \cdot lg_1 + \ldots + a_{i+1f-1} \cdot lg_{lf-2} + d_{i+1f-1} \cdot lg_{lf-1}. \end{aligned} \quad (2)$$

Here the vector *rec* is used to represent the reconstructed signal for the level *i*, *a* and *d* represent the approximation and detail vectors for the lower level and *lf* represents the filter's length.

Firstly we tested the horizontal schematic presented above. The vectors *a* and *d* were divided in equal parts, assigned in pairs to threads. Because instead of reducing the runtimes we obtained their increase, we tested another schematic, depicted by Fig. 13 for a filter of length *lf* and an implementation with 2 threads.

Our interleaved horizontal schematic used to implement a multi-threading technique for reconstruction involves a particular division of the reconstruction-related tasks, described below. If *nr_threads* is the number of threads executed in parallel for the reconstruction associated to the level *i*, then the first thread will reconstruct the components 1,2,*nr_threads*+1,*nr_threads*+2, 2x*nr_threads*+1, 2x*nr_threads*+2, ... kx*nr_threads*+1, kx*nr_threads*+2 etc. The second thread will reconstruct the components 3,4, *nr_threads*+3, *nr_threads*+4, 2x*nr_threads*+3, 2x*nr_threads*+4 , ... kx*nr_threads*+3, kx*nr_threads*+4 etc.

Fig. 14 depicts an example of execution, to reveal the random order in which the threads are scheduled, along with the indices revealing which components from the reconstructed vector (*i*) and respectively from the approximation vector (*j*) are processed at that moment.

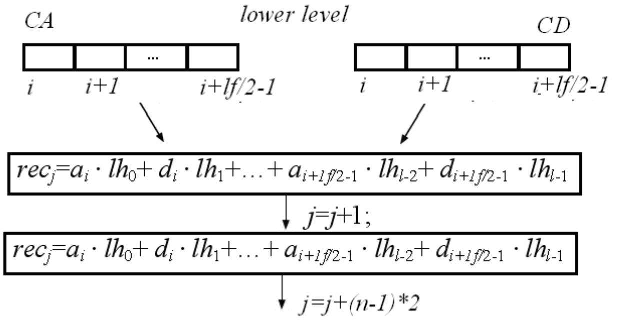

Fig. 13. Interleaved schematic used as principle by the multi-threading technique used during the re-composition for a single phase.

```
Thread 1 i = 2201 j=4407
Thread 1 i = 2203 j=4411
Thread 2 i = 3364 j=6733
Thread 1 i = 2205 j=4415
Thread 2 i = 3366 j=6737
Thread 1 i = 2207 j=4419
Thread 2 i = 3368 j=6741
Thread 1 i = 2209 j=4423
```

Fig. 14. Example of execution in interleaved implementation

V. Tests with Horizontal and Interleaved Multithreading Techniques

A. Accelerating the decomposition by using multi-threading techniques based on horizontal schematics

The shape of waveform has no influence over the runtime and therefore for any data the runtimes will be the same, when the same number of calculation points and machine architecture are considered [9].

10 sets of 300 consecutive decompositions of a phase current (represented by a signal with 36864 components) were performed with our original Java programs.

Figs. 15-17 represent the cumulated runtimes when dwm was used, whilst Figs. 18-20 represent similar results yielded by the use of dwns.

Tables II and III gather synthetic data with respect to the calculated runtimes when horizontal multithreading techniques are applied for DWT decompositions. *R* represents the ratio (mean cumulated runtime – multithreading)/ (mean cumulated runtime – monothread) and *W* is used to denote " the worst scenario" . It is calculated as (maximum cumulated runtime – multithreading)/ (minimum cumulated runtime – monothread).

B. Results of multithreading implementation for reconstruction

None of the above described multithreading techniques could decrease the runtimes. Table IV gathers synthetic data with respect to the calculated runtimes when interleaving multithreading techniques are applied for DWT re-compositions. Similar results were obtained when the horizontal schematic was applied.

TABLE II.
SYNTHETIC DATA FOR DECOMPOSITIONS WITH DWM

	Filter length		
	4	6	8
Mean cumulated runtimes – multihreading [sec]	1.0698	1.1635	1.1459
Mean cumulated runtimes – monothread [sec]	1.3900	1.5075	1.4223
R [p.u.]	0.7696	0.7718	0.8057
W [p.u.]	0.8952	0.9251	0.9040

TABLE III.
SYNTHETIC DATA FOR DECOMPOSITIONS WITH DWNS

	Filter length		
	4	6	8
Mean cumulated runtimes – multihreading [sec]	1.1118	1.2436	1.3350
Mean cumulated runtimes – monothread [sec]	1.4590	1.5927	1.5864
R [p.u.]	0.7620	0.7808	0.8415
W [p.u.]	0.9011	0.8875	0.8188

TABLE IV.
SYNTHETIC DATA FOR RE-COMPOSITIONS

	Filter length		
	4	6	8
Mean cumulated runtimes – multihreading [sec]	2.9239	2.9483	1.8861
Mean cumulated runtimes – monothread [sec]	2.4129	2.4817	1.5438
R [p.u.]	1.2118	1.1880	1.2217

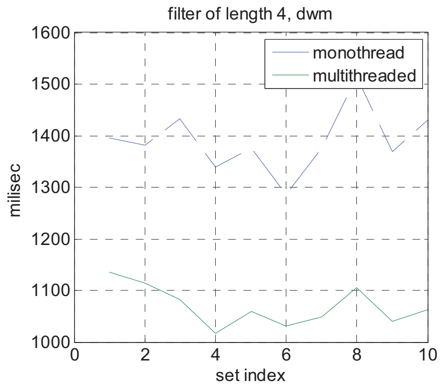

Fig. 15. Runtimes for multithreading decomposition with a filter of length 4, using dwm

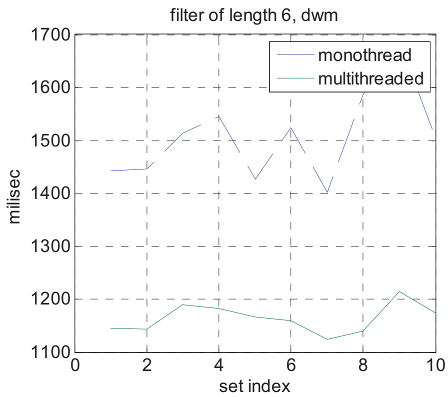

Fig. 16. Runtimes for multithreading decomposition with a filter of length 6, using dwm

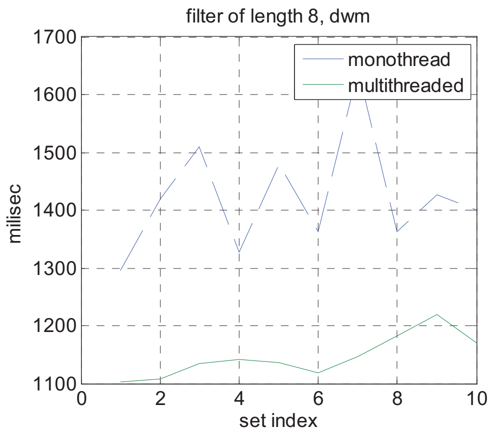

Fig. 17. Runtimes for multithreading decomposition with a filter of length 8, using dwm

When implementing the multithreading techniques, in all cases a preliminary evaluation was performed. According to its results, for the horizontal schematic the multithreading was used only for the first level.

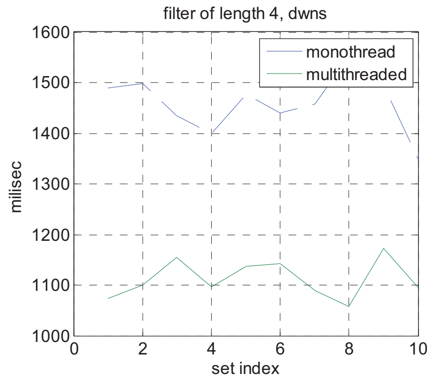

Fig. 18. Runtimes for multithreading decomposition with a filter of length 4, using dwns

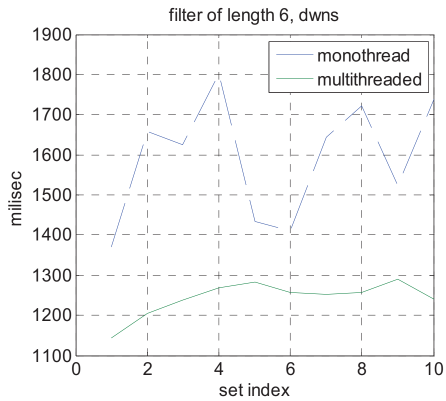

Fig. 19. Runtimes for multithreading decomposition with a filter of length 6, using dwns

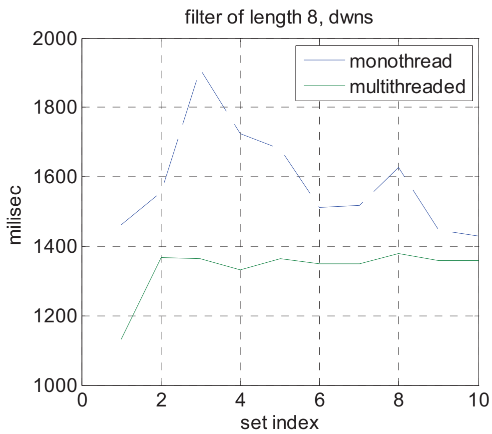

Fig. 20. Runtimes for multithreading decomposition with a filter of length 8, using dwns

The choice of using multi-threading only for the 1st level was imposed by the effect of the decomposed segment's length over the multithreading efficiency, considering the decimation phenomenon (the decomposed vector's length is reduced by a factor of 2 with each level).

VI. VERTICAL MULTITHREADING SCHEMATIC FOR DWT DECOMPOSITION AT THREE-PHASE SYSTEMS

In [9] we presented our results related to the possibility of accelerating the wavelet decompositions for three-phase systems.

Our starting point was to perform in parallel the decomposition of all currents with that of all voltages, over

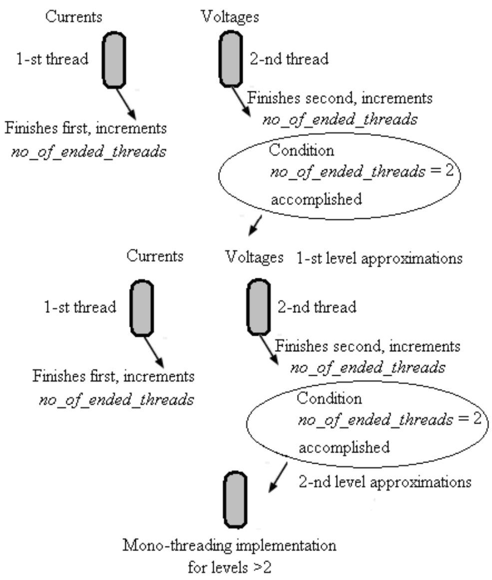

Fig. 20. Vertical schematic used for the implementation of multithreading at three-phase systems.

some decomposition levels [9], [14]. The waveforms depicted by Fig. 2 were used for the simulations, but the shape of waveforms does not influence the runtime-related efficiency. A vertical schematic was used for the implementation of the multithreading technique (Fig. 20).

The decomposition was made with dwm for a filter of length 8, and a number of calculation points equal to 12288.The mean runtime per set when using the multithreading technique for the first two levels of decomposition and a mono-threading technique for the last 8 levels was of 2267.3 msec, whilst that corresponding to the mono-threading, classical implementation for all decomposition levels, was of 4084.3 msec. For reasons explained in the previous section, only the first 2 levels were decomposed in a multithreading manner. The runtime was clearly reduced when using multi-threading with 45% [9].

VII. CONCLUSIONS

When performing a DWT analysis for quasi-stationary or non-stationary regimes, interpolations are required. The analyzed function (dwns) relying on an interpolation schematic based on averaged slopes corresponding to points from the analyzed data segment's right edge was tested from 3 points of view: accuracy when applied to the evaluation of power quality indices, shapes of decomposition vectors and runtimes. Savings of 50 % were obtained for the runtimes required by the non-spline interpolation based function (dwns) as compared to those when spline interpolations were used, under similar accuracy-related performances and shapes of waveforms corresponding to the decomposition vectors. Moreover, the runtime required by dwns is only approximately 30% higher than then that required by dwm, which is interpolation free. This makes dwns a better option for analysis when the nature of the analyzed regime cannot be defined with certitude as perfect stationary.

For decompositions of single phase quantities, the horizontal and interleaved schematics implemented under the form of multithreading techniques provide runtime savings of around 20%, decreasing with the filter's length. The number of calculation points is crucial for multithreading efficiency – too few calculation points result in the depreciation of this parameter, because the overhead associated to the implementation of multithreading becomes too high relative to the runtime saving offered by multithreading. For re-composition none of the multithreading techniques used with success for decomposition could reduce the runtime. It means that the algorithm itself has a great influence over the multithreading efficiency.

Our original vertical multi-threading implementation of the DWT decomposition for the first 2 levels when the interpolations-free function (dwm) is used exhibits runtime savings of 45% as compared to the classic mono-threading implementation.

ACKNOWLEDGMENT

This paper got partial support from the grant no. 448 / CEOSE under the frame of "Research projects in partnership between universities/RD institutes and companies" and respectively from the grant number POSDRU/174/1.3/S/149155.

REFERENCES

[1] M. K. Lakshmanan , and H. Nikookar, "A Review of Wavelets for Digital Wireless Communications", *Wireless Personal Communications*, vol. 37, No. 3-4, pp. 387-420 , 2006.

[2] S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1998, New York.

[3] R. Zolfaghari, Y. Shrivastava and V.G. Agelidis, "Spectral analysis techniques for estimating power quality indices", in *Proc. of the* 14th International Conference on Harmonics and Quality of Power, pp.1-8, 2010.

[4] M. Apráiz, J. Barros and R.I. Diego, "A laboratory for time-frequency analysis of power quality disturbances using wavelets", in *Proc. of the* 14th International Conference on Harmonics and Quality of Power, pp.1-5, 2010.

[5] D. Purcaru, A. Purcaru and C. Gordan, "Interface with Simultaneous Sampling of Analog Inputs for Power Quality Measurement: Testing Programs and Experimental Results", *Annals of the University of Craiova, Series Automation, Computers, Electronics and Mechatronics*, Vol. 9 (36), No.2, pp. 33-42 , 2012.

[6] D. Purcaru, C. Gordan and I. Purcaru, "An Application of the PC104 Interface with Synchronized Sampling of Analog Inputs in Energetic Systems", *Journal of Electrical and Electronics Engineering*, Vol. 6, No. 1, pp. 121-124, 2013.

[7] I.D. Nicolae, and P.M. Nicolae, "Using Wavelet Transform for the Evaluation of Power Quality in Distorting Regimes", *Acta Electrotehnica*, vol. 52, no.5, pp. 331-338, 2011.

[8] I.D. Nicolae and P.M. Nicolae, "Using wavelet transform for power systems", *R*evue Roumaine de Sciences Techniques – Électrotechnique et Énergetique, 57 (2), p. 172–182, 2012.

[9] I.D. Nicolae, A. Chiva, P.M. Nicolae and M.Ş. Nicolae, "Techniques to Accelerate Power Quality Analysis Based on DWT" , International Conference on Applied and Theoretical Electricity, in *Proceedings of the IEEE Int. Conference on Applied and Theoretical Electricity , ICATE 2014*, paper no. 4.17, 2014.

[10] I.D. Nicolae, P.M. Nicolae and D.L. Popa, "Using Discrete Wavelet Transform to Analyze Quasi-Stationary Three-Phase Systems", *Proceedings of the International Conference on Harmonics and Quality of Power , ICHQP 2014*, pp. 102 – 106, 2014.

[11] P. V. Fleet, "*Discrete Wavelet Transformations: An Elementary Approach with Applications*", Wiley-Interscience, 2009.

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

[13] I.D. Nicolae, and P.M. Nicolae, "Performances Evaluation of a Class of Original Discrete Wavelet Transform Based Hybrid Algorithms", *Proceedings of the International Conference on Harmonics and Quality of Power , ICHQP 2012*, pp. 727 – 732, 2012.

[14] A. Chiva, "Software using parallel computing for wavelet functions implementation", *Diploma project*, unpublished, 2014.