

Wavelet Library for Modelica

Jianbo Gao*, Yang Ji**, Johann Bals**, Ralph Kennel*

*Technische Universitaet Muenchen, Arcisstr. 21, 80333 Munich, Germany
michael.gao@tum.de, ralph.kennel@tum.de

** German Aerospace Center, Muenchner Str. 20, 82234 Wessling, Germany
yang.ji@dlr.de, johann.bals@dlr.de

Abstract

Wavelet analysis is being widely used in different fields for signal processing to increase efficiency and flexibility. A wavelet library has been a standard component in many simulation programs. However, wavelet analysis has not yet been included in Modelica as a standard component. To fill this blank, a comprehensive wavelet library has been developed for Modelica. This library includes fifteen commonly used wavelet families. It can carry out continuous transform, forward and inverse discrete transforms, and multi-level decomposition and reconstruction in one-dimensional space. In addition, special application tools for multi-resolution analysis and wavelet denoising are provided. Moreover, some examples are given to provide the users a quick start point to build up their own algorithms. This library was programmed and tested according to the Modelica language specification 3.2 under the Dymola platform version 2013. The test results prove the functionalities of the library.

Keywords: Modelica; Dymola; wavelet; Modelica library

1 Introduction

Although the wavelet transform has only been established for several decades [1][2], it has already been applied in many fields because of its superior properties in signal processing. Besides practical applications, the wavelet transform is also a versatile tool for different simulation problems [3][4][5][6]. So far the wavelet transform has become a standard toolbox in many free and commercial simulation programs, such as Simulink™, Mathematica™, Maple™, Vis-Sim™ and many others.

Modelica, as a comprehensive multi-physical simulation tool, has not yet included a library for carrying out wavelet transform. However, there exists demand for using wavelet transform within Modelica. In

2005, Bünte and his colleagues analyzed the simulation data of vehicle steering dynamics generated in Modelica with the wavelet transform [7]. Because of the lack of wavelet transform in Modelica, the analysis had to be done with other software. In the paper of Ji [8] in 2010 about a Modelica signal analysis tool towards the design of more electric aircrafts the authors expressed the wish to do the analysis using wavelet transform directly within Modelica.

The first application for the failure analysis in the power system using the prototype of the Modelica wavelet library was already introduced in 2012 [9]. After the successful development, the final Modelica wavelet library is introduced in this article. It includes the following features.

- Fifteen wavelet families;
- Continuous transform;
- Discrete forward and inverse transforms;
- Multi-level decomposition and reconstruction;
- Multi-resolution analysis;
- Denoising and data compression;
- User friendly graphic user interface (GUI);
- Importing simulation data from the hard disk;
- Fully open source under Modelica License 2.

In the first release version, the library has the following limitations.

- Only one-dimensional transformations are implemented.
- Only post-processing is possible. The algorithms cannot execute in real time applications.
- The data to be analyzed must be have equidistant time grids.
- The library is developed and tested under Dymola 2013 demo version.

2 Wavelet transform

The basic knowledge about wavelet theory is required in order to use the wavelet library. In this section, a very brief description about the wavelet the-

ory will be given [1][2]. The theoretical part of [9] is roughly repeated here but with improvements for better understanding of common readers.

2.1 Definition

The wavelet transform can be considered as a further development of the Fourier transform, or more precisely, of the short time Fourier transform (STFT) [10]. Using the STFT, people try to localize the signal changing by selecting suitable time windows. This transformation, however, has its inherent drawbacks. The most significant one is its limit in time-frequency resolution due to the uncertainty principle. The wavelet transform overcomes this problem. This transformation is defined as [1]:

$$[W_{\psi}f](a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \overline{\psi\left(\frac{t-b}{a}\right)} f(t) dt. \quad (1)$$

This equation defines the wavelet transform of the function $f(t)$ using wavelet function ψ at scale a and position b . The bar above function ψ stands for conjugation. For given a and b , the transformation result is a single real number, named wavelet coefficient. The wavelet function must satisfy some conditions to ensure that it is an orthonormal function:

- (1) Zero mean: $\int_{\mathbb{R}} \psi(t) dt = 0$;
- (2) Unit square norm: $\int_{\mathbb{R}} |\psi(t)|^2 dt = 1$;
- (3) Compact support: $\psi(t) = 0$, for $|t| > C$, where C is a positive real number.
- (4) Dyadic orthogonality: $\langle \psi_{jk}, \psi_{lm} \rangle = \delta_{jl} \delta_{km}$,
for $\psi_{jk} = 2^{\frac{j}{2}} \psi(2^j x - k)$. Here $\langle \cdot, \cdot \rangle$ means inner product of two functions.

The wavelet function only represents the signal components that have zero mean, or high frequency components. For the zero and low frequency components, a scaling function, which has unit mean value, is required. The wavelet and the scaling functions with the dyadic scaling and translation parameters build a complete orthogonal basis in the Hilbert space.

The precise mathematical description of orthonormality is easily found in almost every book about the wavelet transform, e.g. [1] and [2], and will not be repeated here.

From this definition it is known that the wavelet transform is the integral of the multiplication of the signal to be studied, f , with a wavelet function, ψ . It has the same form as the STFT. However, not like the STFT, where only sine and cosine functions are used for the transformation, the wavelet transform uses different wavelet functions, which can be se-

lected according to the specific problems from a principally unlimited set.

Parameter a defines the width and height of the wavelet function ψ keeping its unit square norm. If a makes ψ narrower, the wavelet represents fast changes and the transformation focuses on the high frequency components of the signal. Parameter, b , shifts the wavelet function along the time axis, so that the transform helps us to observe signals at different locations. Using different values a and b , it is possible to observe the signal at different positions in the time domain and in different frequency ranges with only one transformation.

Two forms of wavelet transform are available: Continuous wavelet transform (CWT) and discrete wavelet transform (DWT). In CWT both scale and position parameters are continuous real values. The transformation result, i.e. the wavelet coefficient, is therefore also continuous. CWT expresses the signal changes in a continuous manner. It is more suitable for the visual examination. However, the transform result contains redundant information and the transform requires large calculation efforts. The general CWT has no corresponding inverse transformation.

2.2 Discrete wavelet transform (DWT)

In the DWT only discrete values of the scale and location parameters are used. The values are determined according to point (4) of the wavelet properties. The transformation result, i.e. the wavelet coefficient, is therefore discrete.

As an example, the following figure shows the form of the third order Daubechies scaling and wavelet functions [1].

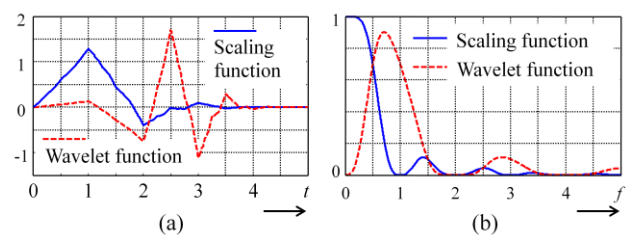


Figure 1: The third order Daubechies scaling and wavelet functions (a) with their Fourier transforms (b)

From the Fourier transforms it can be seen that the scaling function mainly covers the lower frequency range while the wavelet function stretches in a higher frequency range. This coincides with the aforementioned description. From this point of view, DWT is actually the division of the time signal into different frequency bands. Thus, it is straightforward to understand that the calculation of DWT is realized

using filter banks. In inverse DWT the calculation is similar. This process can be illustrated with Figure 2.

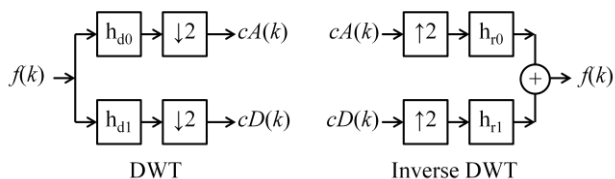


Figure 2: DWT and inverse DWT calculation using filter banks

DWT transforms the original sequence in two new series:

- (1) the approximation coefficients, $cA(k)$, representing the low frequency components, obtained using the low pass filter for decomposition, h_{d0} , and
- (2) the detail coefficients, $cD(k)$, representing the high frequency components, obtained using the high pass filter for decomposition, h_{d1} .

The symbol $\downarrow 2$ means down-sampling. The operation is to delete one from every two adjacent coefficients, in order to remove the redundant information. The inverse DWT carries out the reversed operation. The operator, $\uparrow 2$, expands a coefficient series by inserting a zero between every two adjacent elements. After that the two series are operated with two filters, respectively, and added together to get the original signal.

2.3 Multi-resolution analysis

Considering the DWT process shown in Figure 2, sequence, $cA(k)$, which represents the low frequency components, can be further divided into a lower frequency part and a higher frequency part. This process can be repeated and a series of coefficient sequences representing different frequency ranges will be obtained, as shown in the following figure:

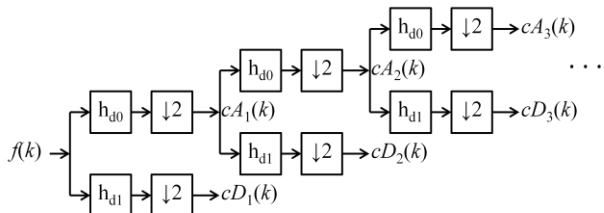


Figure 3: Wavelet multi-level decomposition

This process is named as wavelet multi-level decomposition. The output of this operation, cD_1, cD_2, \dots, cD_n and cA_n , are different levels of DWT coefficients, representing the signal components from higher to lower frequencies. This analysis provides a convenient tool to observe different frequency components of the signal depending on time.

Similar to the calculation process of inverse DWT, if the algorithm goes in an inverse direction of Figure 3, we will get the original signal using the wavelet coefficients of different levels. This process is called signal reconstruction.

Applying multi-level decomposition, we can project a signal into a sequence of nested subspaces. We are then able to observe the original signal in different subspaces that contain different details of the signal. This realizes the multi-resolution analysis (MRA) of the signal using wavelet transform.

3 Realization

The wavelet library is developed according to the Modelica Language Specification 3.2 with the simulation environment software, Dymola 2013 (32-bit) demo version. Although the Dymola demo version has limitations on the complexity of simulation models, it does not impose any limitations in programming.

3.1 Library structure

The wavelet library structure is sketched in Figure 4.

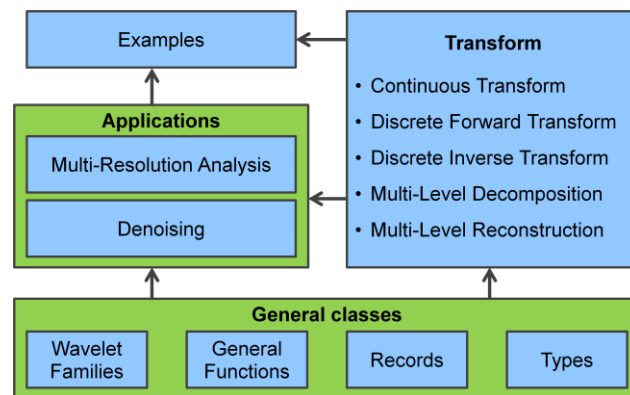


Figure 4: Wavelet library structure

The whole library is composed of eight packages and 64 single classes, plus an external C-file to support generating Meyer wavelets. All of these elements are summarized in four parts as shown in the figure.

The central part of this library is the package “Transform”, which includes the functions carrying out different wavelet transformations.

The part “General classes” provides a fundament for realizing the wavelet transforms and other operations, as shown in the bottom part of the above figure. Four packages are included in this part. They are

- “Wavelet Families” for generating wavelet filter banks and functions,
- “General Functions” for the basic data operation and processing,
- “Records” for the definition of data structures and supporting GUI, and
- “Types” to define enumeration parameters.

Two wavelet applications – MRA and denoising – are provided. It should be noted that the wavelet denoising package can also be applied for the data compression since these two operations use the same concept and algorithm [2].

In addition, the package “Examples” consists of functions and models for six examples to provide the users a quick introduction to the wavelet library.

3.2 Wavelet library release

The functional part of this library consists of only two files:

- “Wavelet.mo” as the main library package to include all Modelica codes, and
- “fft_c.c” as an external C-function to calculate Fourier transform for generating Meyer wavelets.

Two example data files, testSignal1.mat and testSignal2.mat, are delivered, although they could be very easily generated by running the two example models in the library.

In addition, a “Help” folder including all descriptions of this library is also a part of the delivery.

3.3 Required components

Besides the Modelica standard libraries, the full performance of this library relies on two further libraries, **Modelica_LinearSystem2** and **Plot3D**. The former one is a free library available from Modelica Association. The latter one is delivered with Dymola, and is only used for showing CWT results. In the free demo version of Dymola, Plot3D works but is limited to two-dimensional images.

3.4 Wavelet families

Fifteen wavelet families are available in this library.

- Haar
- Daubechies , up to the 20-th order
- Symlets , up to the 20-th order
- Coiflets , up to the 5-th order
- Biorthogonal spline, 15 variations
- Reverse biorthogonal spline, 15 variations
- Discrete Meyer
- Meyer

- Gaussian , up to the 8-th order
- Mexican hat
- Morlet
- Complex Gaussian , up to the 8-th order
- Complex Morlet
- Complex Shannon
- Complex frequency B-Spline , unlimited order

The package “Families” consists of fifteen functions for the wavelets listed above. Two more functions that are directly related to wavelet families are also included: “scalingWaveFunc” for generating the wavelet and scaling functions of a specific wavelet, and “wavFunc” providing a common entry to access all fifteen wavelets.

3.5 Wavelet transform

Five variations of one-dimensional wavelet transform have been implemented in the library. The related functions are included in the package “Transform”.

CWT is provided with two variations for more flexibility. One CWT variation accepts a wavelet name while the other one accepts a wavelet function as the input parameter. According to the wavelet theory, no inverse transform exists for general CWT.

For the discrete version, both forward and inverse wavelet transforms are possible. They are realized with two Modelica functions, respectively.

In the discrete domain, the multi-level data decomposition and the reconstruction are carried out by actually applying the discrete wavelet transform and inverse transform in a cascaded manner. The algorithms are implemented by two Modelica functions, “wavDec” and “wavRec”, respectively.

In addition, two more functions are included in this package, too, since they are closely related to the transforms. Function “wavRec1” reconstructs the data using only the wavelet coefficients of one specific level. This is actually a subset of multi-level reconstruction. The other function “wavCoef1” is used to extract the wavelet coefficients of a certain decomposition level. These two operations are specially useful for wavelet applications, such as MRA and denoising.

3.6 Applications

The wavelet library provides two applications, the MRA and the denoising, which are supposed to be the mostly used functionalities for processing the simulation data.

The principle of the wavelet MRA has been described in section 2.3. With this tool the user is able to divide the signal into different scales, or frequency regions, and observe the signal with different resolution. Using different Modelica functions, users can get the MRA results either in the numeric form or in the graphic form with curves. In addition, this tool provides the flexibility to tune the coefficients of every single level, so that the user can intentionally strengthen or suppress the information in some certain levels. An example of MRA is to be given later in this article.

Like MRA, the wavelet denoising is carried out based on wavelet multi-level decomposition and reconstruction. The wavelet denoising is suitable only for data with high signal-to-noise ratio. Based on this condition, the wavelet coefficients representing the noise have smaller values compared with those representing useful information. By removing these small coefficients, the noise can be eliminated from the data. The threshold to separate the noise and the information coefficients is the key to this algorithm. It can be selected by the user or automatically estimated by the tool. It has to be noted that, if automatic estimation is used, the noise in the data must have Gaussian distribution with zero mean and unit standard deviation.

As an extra possibility, data compression can be realized by the denoising application tool since these two operations are actually based on the same algorithm. For more information about this topic, relevant literatures, such as [1] and [2], can be referred to.

3.7 Graphical user interface

A graphical user interface (GUI) is important for a user-friendly environment in this library. The GUI is mainly realized by using Modelica “Types” combining the simulation environment software for inputs and diagrams for outputs.

By combining the GUIs for input and output, a very simple but convenient interactive operation is possible. It is also possible to realize more complex and user-friendly interactive performance using specific scripts written by users.

3.8 User’s guide

A complete User’s Guide in PDF format with a detailed description of every single class in this library is delivered in the release version. In addition, a “Help” folder is included in the delivery package. This folder contains HTML files with hyper-links.

Within the library, no separate package serving as a User’s Guide is provided since all related descriptions about the Modelica classes, including Packages, Models, Functions, Types and Records, have already been embedded in the library. These descriptions can be easily accessed in the simulation environment software. Hyper-links are included in the description texts for quick movement among the library to access the information from different classes.

4 Testing the library

The algorithms of this wavelet library were completely tested with the black-box method. Testing vectors have been defined for each algorithm. The function outputs were checked visually and numerically. The numeric check was carried out with the assistance of MATLAB because most algorithms written in the Modelica wavelet library are included in the MATLAB wavelet toolbox. For these algorithms, the same input parameters were given to the functions of the Modelica wavelet library and the corresponding commands in MATLAB. The calculation results of both tools were compared. An algorithm of the library is considered correct if the discrepancy is below a certain limit.

All testing results were recorded and summarized. As an example, Table 1 shows the summary of the testing results of the function “Wavelet.Families.wavCoiflets”, which is used to generate the wavelet filters associated with Coiflets wavelets.

Table 1: Testing recording for Wavelet.Families.wavCoiflets

order	F	lod	hid	lor	hir	Result
1	-	-	-	-	-	Error = 0
2	-	-	-	-	-	Error = 0
3	-	-	-	-	-	Max. error < 1e-16
4	-	-	-	-	-	Max. error < 1e-16
5	-	-	-	-	-	Error = 0

The first six columns of the table records the input and output values of the function. Since the data amount of the output values is too large, the data are stored in a separate file. The last column “Result”, records the errors of the outputs obtained by the Modelica function under test compared with the data calculated by the same algorithm written in MATLAB.

All other functions in the library were tested with the same process and altogether 133 numeric testing results were recorded. The errors are illustrated in Figure 5. Since the values are shown in logarithm axis,

the first 58 points with zero value are not plotted. The testing showed that the more than half of the tests have the errors below $2e-16$, which is around the digital error of a double precision floating number. Other errors with the values up to $5e-14$ might be caused by the precision of the direct constant numbers and difference in details of the coding in the algorithms.

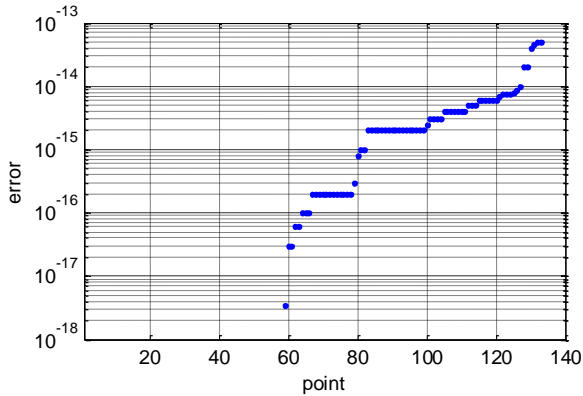


Figure 5: Error distribution of the testing results

In addition, visual checks were carried out for all functions that have graphical inputs and outputs. All results coincided with the specifications. For the functions outputting curves and images, the graphic data are actually numerical results and also tested numerically.

5 Examples

In this section, we take two examples to partially show the usage and functionalities of the new Modelica Wavelet Library in the simulation environment, Dymola.

5.1 The signal under study

The signal under study is defined in equation (2).

$$\begin{aligned}
 y &= y1 + y2 \\
 y1 &= \begin{cases} \sin(2\pi 10t), & t < 0.5 \\ 0.2 \sin(2\pi 10t), & t \geq 0.5 \end{cases} \\
 y2 &= \begin{cases} 0.2 \sin(2\pi 50t), & t < 0.5 \\ \sin(2\pi 50t), & t \geq 0.5 \end{cases}
 \end{aligned} \quad (2)$$

It is a time-variant signal containing two frequencies, 10 and 50 Hz. At time $t = 0.5$ s, the magnitudes of the two frequency components alternate. Signal y is generated with the Modelica model “testSignal2” in the package “Examples” of the wavelet library. It is shown in Figure 6.

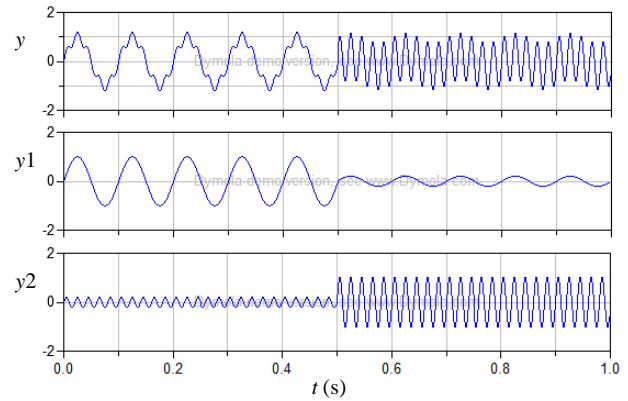


Figure 6: The signal under study for the examples

Since the wavelet transform can only be correctly performed with an equidistant time grid, the simulation data should be stored with an equidistant time grid. Otherwise, the data have to be converted to equidistant time grid with the library function “interpL” before applying wavelet transform. In the examples, this is ensured by setting the sampling frequency “fs” to a none-zero positive value.

5.2 Wavelet multi-resolution analysis

Firstly we carry out an MRA using the wavelet library to observe the signal around the transient time, $t = 0.5$ s with different resolutions. This is done by executing the function “fileDataMRA” in the package “Examples”. Most of the input parameters can be left at default values except the following ones:

- Time range for analysis is set as $t_0 = 0.25$ and $t_1 = 0.75$ since we are only interested in the transient region;
- Sampling frequency $fs = 480$, so that the highest frequency in the testing data is 240 Hz ($fs/2$);
- $Nd = 7$ in the wavelet definition “wd” to select the 7-th order Daubechies wavelet;
- $decLevel = 3$ in the MRA parameters “mraParameters” for carrying out a three level wavelet decomposition; and
- $rA = 0$ in “mraParameters” to remove the lowest frequency components in the reconstructed signal.

After execution, we get eight curves displayed in two diagrams in Dymola. In the first diagram, the original signal and the data in the approximation level and three detail levels are displayed. They are shown here in Figure 7.

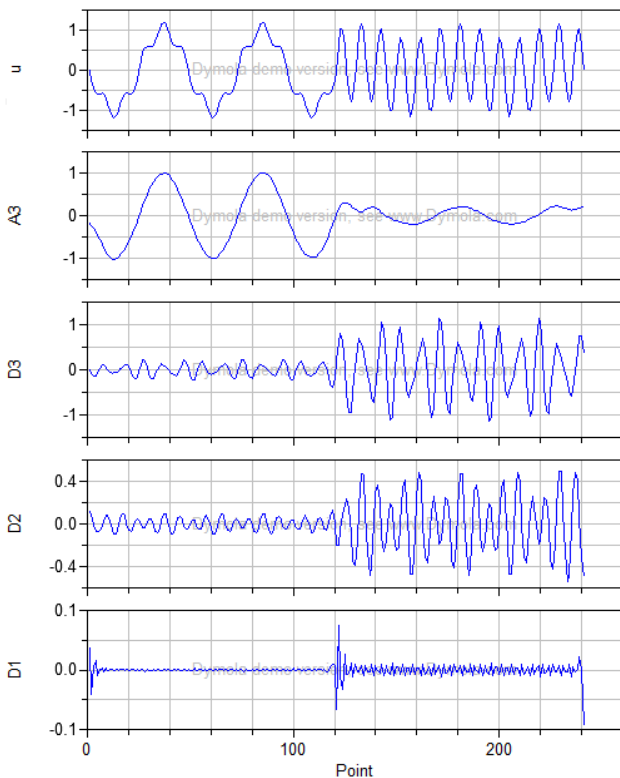


Figure 7: The original signal and decomposed data

The x-axis of the diagram shown in Dymola is given in data point. There are 240 data points; and the symbol of the original data is shown as “u” instead of “y”.

The wavelet decomposition in this MRA separates the original signal into four time series containing different frequency components. This is illustrated with Figure 8.

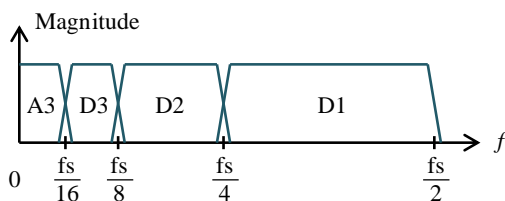


Figure 8: Separation of the frequency components with three-level wavelet multi-resolution analysis

Since the sampling frequency is $f_s = 480$ Hz, we know that the 10 Hz component of the original signal is projected into level A3 and the 50 Hz component falls in D3. From Figure 7 we also see some oscillations in level D2. This is because the wavelet decomposition does not realize perfect frequency separation. Furthermore, some peaks in level D1 represent the frequency transient and edge effect, which imply fast changing events.

The second half of the MRA in this example is data reconstruction after the tuning of the wavelet coefficients. This is shown with four curves in the second

diagram, which is not repeated in this article. Since we have set $rA = 0$, meaning to remove the information in the approximation level, which mainly contains the 10 Hz component, the reconstructed data mainly contains the 50 Hz component, as shown Figure 9. We can see that this signal is almost identical to the middle part (0.25 – 0.75 s) of signal y2 in Figure 6.

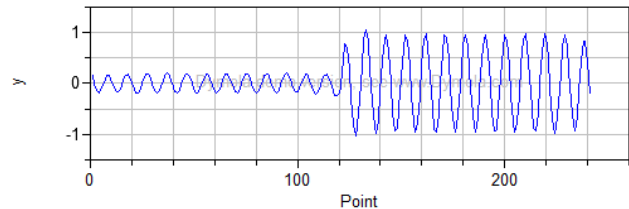


Figure 9: The reconstructed data after removing the approximation coefficients in wavelet MRA

5.3 Continuous Wavelet Transform

The second example demonstrates CWT for the same signal as shown in Figure 6. By executing the function “fileData_cwtn” without changing any default parameters we can get the CWT result illustrated with a 2-dimensional image or a 3-dimensional surface, depending on the version of Dymola used. The image with pseudo-colour is shown in Figure 10.

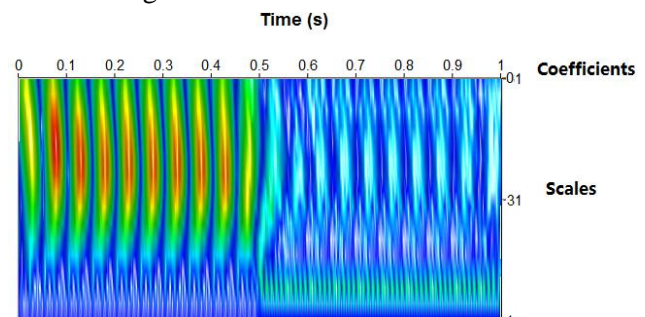


Figure 10: Continuous wavelet transform of the signal under study

The coefficients are shown with warmer colours (yellow-orange-red) for larger magnitudes and colder colours (green-light blue-dark blue) for smaller magnitudes. The horizontal axis represents the time from 0 to 1 second. The vertical axis represents the wavelet scales from 1 to 64. The relationship between scales and frequencies is determined by both the sampling frequency of the data and the wavelet used. By observing this CWT image we can draw several conclusions:

- The frequency transient happens at $t = 0.5$ s.
- Both low frequency components (here 10 Hz) and high frequency components (here 50 Hz) are present throughout the time span of the signal.

- Before 0.5 s low frequency components are much stronger than the high frequency components. This reverses after 0.5 s.

6 Conclusion

This article describes a new wavelet library developed for Modelica. This library contains fifteen wavelet families commonly used in research and engineering applications, continuous wavelet transform, discrete wavelet transform both in forward and inverse directions, two application tools and several examples. In addition, the library provides graphic user interfaces for some functions to support a user-friendly workflow. Moreover, it includes documentation embedded in the library, plus an external user's manual.

During and after the development work, black-box testing was carried out for all algorithms. The testing results have proved the correctness of the library.

The first release of this Modelica wavelet library is limited to one-dimensional wavelet transforms for post-processing of the simulation data. Several wavelet calculations, such as wavelet packet and two-dimensional transform, are not yet included. This could be gradually added in the future. Nevertheless, the introduction of this wavelet library will definitely enrich the functionality of Modelica and contribute much to the Modelica society.

Acknowledgement

The authors thank the support from the Clean-Sky Joint Undertaking through the Grant Agreement No. 296369 (Project MoMoLib) within the Seventh Framework Programme of the European Union.

References

- [1] I. Daubechies. Ten Lectures on Wavelets. SIAM 1992.
- [2] S. Mallat (2009): A wavelet tour of signal processing - the sparse way. Amsterdam: Elsevier.
- [3] X. Zeng, S. Huang, et. al. Frequency domain wavelet method for high-speed circuit simulation. IEEE International Symposium on Circuits and Systems. ISCAS 2002, pp II-233-236, 2002.
- [4] Q. Wang and F. Jiang. Simulation for fault of transmission lines based on discrete wavelet transform. International Conference on Electrical and Control Engineering. ICECE 2011, pp 2554-2556, 16-18 Sept. 2011.
- [5] G. Ala, M.L. Di Silvestre, et. al. Wavelet-based efficient simulation of electromagnetic transients in a lightning protection system. IEEE Transactions on Magnetics. ITM 2003, vol 39, no 3, pp 1257-1260, May 2003.
- [6] W. Xu, Y. Cui and J. Liu. Simulation Study of Detecting Pulse Signal Based on Wavelet transform. 8th International Conference on Electronic Measurement and Instruments, 2007. ICEMI '07, pp II-479-483, Aug. 16-18 2007.
- [7] T. Bunte, A. Sahin and N. Bajcinca. Inversion of Vehicle Steering Dynamics with Modelica / Dymola. Proceedings of the 4th International Modelica Conference, pp 319-328. Hamburg, Germany, March 7-8, 2005.
- [8] Y. Ji and J. Bals. A Modelica signal analysis tool towards design of More Electric Aircraft. Proceedings of 2010 International Conference on Information and Applied Electronics, pp 152 - 156. Chendu, China, June 9-11 2010.
- [9] J. Gao, Y. Ji, J. Bals and R. Kennel. Fault Detection of Power Electronic Circuit using Wavelet Analysis in Modelica. Proceedings of the 9th International Modelica Conference, pp 513 - 521. Munich, Germany, Sept. 3-5, 2012.
- [10] E. Jacobsen and R. Lyons. The sliding DFT. Signal Processing Magazine, vol. 20, issue 2, pp. 74-80, March 2003.