Wave Whiz Guide

BACKGROUND

Most real-world systems are complex. Some examples of complex systems include global climate (a physical system), the human body (a physiological system), and the economy (a social system). A complex system comprises a large number of intricately interacting components. It is therefore difficult to examine all dependencies and relationships between the numerous elements within a complex system. Consequently, a complex system is often studied by analyzing its behaviour as a whole, rather than analyzing the behaviours of and relationships amongst its myriad parts[1-3].

A signal or time series emanating from a complex system represents the overall behaviour of the system. For example, an electrocardiogram (ECG) signal characterizes the functioning of a cardiovascular system and a composite stock index series describes the performance of a financial market. Therefore, time series analysis promises to throw light on the behaviour, functioning, and performance of various real-world complex systems[4-7].

The concept of scaling is quite universal to complex systems and the time series that these systems produce[1-7]. Scaling means that these systems and signals behave differently over different timescales. For example, a stock price index tends to show higher volatility (high variability) over shorter timescales (minutes, hours, days) and greater stability (low variability and a trend) over longer timescales (months, years).

The multiscale wavelet analysis (WA) is a computational technique rooted in mathematics and digital signal processing (DSP) that offers a natural framework for studying the behaviour of complex time serial data over various timescales[4,8-15]. Essentially, the WA breaks down (decomposes) a time series (of N samples) into several time-domain sub-series (each containing N samples) that exhibit high-frequency (small timescale) to low-frequency (large timescale) phenomena. The WA can also be used to form a multiresolution analysis (MRA), whereby the resulting sub-series are linearly additive in nature. That is, adding all the MRA sub-series results in a perfect reconstruction of the original signal. Therefore, wavelets can be effectively used for accomplishing time series analysis tasks like pattern recognition, denoising (DNG), trend/cycle detection, and multiscale variance (VAR) analysis.

Percival et al. have proposed a robust wavelet algorithm, which they refer to as the maximal overlap discrete wavelet transform (MODWT)[14], for analyzing time series data. The MODWT has several advantages over the traditional discrete wavelet transform (DWT), for example, it can analyze signals that have arbitrary sample sizes (not restricted to sample sizes that are multiples of 2^J, where, $J = 1, 2, \ldots, J_{Max}$), it produces wavelet sub-series that are temporally better aligned with features in the original signal, and it provides a more efficient estimate of multiscale VAR. Various research studies that have analyzed complex real-world time series data conclude that the MODWT is indeed a superior and effective signal analysis tool[4,16-18].

The MODWT-based WA is accomplished via Mallat's pyramidal algorithm (Fig. 1)[12-14]. Simplistically, a given discrete time signal *Y(t)* is convolved with a predefined wavelet filter *$h_1(t)$* and scaling filter *$g_1(t)$* to obtain level-one *(J = 1)* wavelet coefficients *$W_1(t)$* and scaling coefficients *$V_1(t)$* respectively. For level-two *(J = 2)*, *$V_1(t)$* is convolved with *$h_2(t)$* and *$g_2(t)$* (upscaled versions of *$h_1(t)$* and *$g_1(t)$*) to obtain *$W_2(t)$* and *$V_2(t)$*. The above procedure is repeated on scaling coefficients *$V_J(t)$* for *J = 1, 2, …, J_{Max}*. A level *J_{Max}* WA of *Y(t)* yields the following sub-series: *[$W_1(t)$, $W_2(t)$, …, $W_{JMax}(t)$, $V_{JMax}(t)$]*. Here, *$W_J(t)$* are the wavelet coefficients while *$V_{JMax}(t)$* are the scaling coefficients.

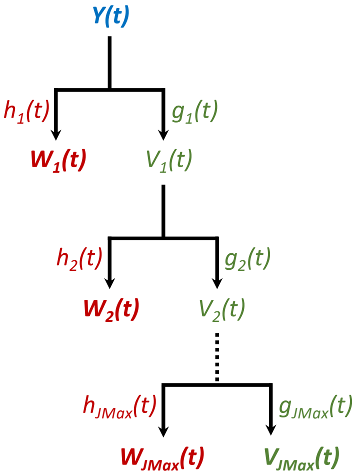

Fig. 1: Mallat's Pyramidal Algorithm

A deconvolution is carried out using the original signal *Y(t)* and WA sub-series *[$W_J(t)$, $V_{JMax}(t)$]* to deconstruct *Y(t)* into MODWT-based MRA sub-series *[$D_1(t)$, $D_2(t)$, …, $D_{JMax}(t)$, $S_{JMax}(t)$]*. Here, *$D_J(t)$* are referred to as wavelet details while *$S_{JMax}(t)$* is referred to as the wavelet smooth. Please note that a sample-by-sample addition of the MRA details and smooth, *[$D_J(t) + S_{JMax}(t)$]*, returns a perfect reconstruction of the original signal *Y(t)*.

For a signal *Y(t)*, the MODWT-based wavelet coefficients *$W_J(t)$* can be used for assessing multiscale VAR while the additive MRA sub-series *[$D_J(t) + S_{JMax}(t)$]* can be used for signal DNG.

WAVE WHIZ SUMMARY

Wave Whiz (Wavelet Wizard) is a powerful, versatile, and fully interactive iOS App that can robustly compute the MODWT of one-dimensional (1D) signals utilizing 25 different wavelet and scaling filters (for example, Haar, Daubechies DB4, and Least Asymmetric LA8).

Any 1D signal can be loaded into the App from a digital signal file or via an input console where the signal values can be typed in or pasted.

The App allows users to undertake a MODWT-based WA for decomposing an input signal into wavelet and scaling coefficients. The App can also accomplish a wavelet MRA to deconstruct an input signal into linearly additive wavelet details and smooth. Additionally, users can apply a hard thresholding method on the MRA details for DNG a signal. Furthermore, users can estimate multiscale VAR of a signal via the various wavelet coefficients. Finally, the App provides a dictionary (DICT) feature, whereby users can review and examine all the 25 wavelet/scaling filters and their respective squared gain functions (SGFs).

A user-friendly tab menu in conjunction with a smart control panel enables users to undertake various MODWT-based analyses. Users can eyeball all wavelet sub-series and wavelet-related data via fully interactive zoomable and scrollable charts. These are high-fidelity charts that are updated in real-time as various analysis parameters like type of filter and number of decomposition levels are changed/adjusted via the smart control panel.

Users can save all wavelet analysis data locally on their devices and/or export it through email, iCloud, or third-party Apps like Dropbox. This feature allows users to seamlessly transfer all processed data from the App to other devices/platforms where they can review it and perform further analyses as required. Moreover, this functionality also enables users to share their wavelet analysis data with their colleagues/peers.

The free version of the App includes 10 tester files that users can load, visualize, and analyze. Users can also save and/or export all processed information using the free version of the App.

In Wave Whiz's paid version, users can load/analyze their own signals and save/export all results.

WAVE WHIZ DETAILS

Data Format: Wave Whiz can load, read, and parse tab-delimited and/or comma separated values (CSV) numeric data. That is, the App can work with any discrete 1D signal presented to it in a tabular form, whereby the delimiter is a tab and/or a comma.

The App expects the input data to consist of either one or two columns. If the input data is a single column of numeric values, then the App will automatically generate the associated time (*T*) values using an initial value of 0 and a ΔT of 1. Suppose, the following single column, containing five samples or numeric values ($N = 5$), is presented to the App:

0.5377
1.8339
-2.2588
0.8622
0.3188

The App will consider the above input as the first five signal values and generate the associated *T* values as follows:

0	0.5377
1	1.8339

2	-2.2588
3	0.8622
4	0.3188

If the input data is in the form of two columns, then the App will consider the first column as *T* and the second column as signal values. A valid two-column five-sample input signal (*N = 5*) will look like the following:

0.0000	-0.2050
0.0100	-0.1241
0.0200	1.4897
0.0300	1.4090
0.0400	1.4172

Please note that in the above example, the values of *T* are monotonically increasing and $\Delta T = 0.01$ (sampling frequency is 100 Hz if *T* is measured in seconds). For any input data that has a two-column format, the App will check and ensure that the *T* (first column) values are always increasing and contain no repeated values. In case, these conditions are not met, the App will throw a data parsing error.

Moreover, the App will also throw an error for any data formats that it does not recognize and/or cannot parse.

The App will accept any signal with a valid format (as described above) as long as the number of samples in this signal are in the 2-70000 range ($2 \leq N \leq 70000$). The App will check *N* for all incoming signals and throw an error if the above condition is not met. The lower limit of *N = 2* is chosen based on the fact that at least two samples are required to compute the MODWT. The upper limit of *N = 70000* has been chosen for optimizing the App's performance.

Tester Files: The free version of the App ships with 10 tester files. These are TXT files that contain various real-world signals ranging from stock price volatility to geophysical time series to physiological recordings like the ECG. Users can load any of the tester signals into the App by tapping at the "Tap to load data…" button at the top-center of the screen and selecting from the list of tester files. Users are encouraged to carry out various MODWT-based analyses on these tester signals to get a feel of the App.

User Data: In Wave Whiz's paid version, users can upload and analyze their own data. User data can be inputted into the App: (i) Via tab-delimited/CSV files with a .TXT/.CSV extension and (ii) Via a data input console where users can type in or paste signal values.

To load data from a user file, tap on the "Tap to load data…" button, then tap on the browse button on the right to browse and load the desired signal file. The App will load data from all correctly formatted files that have a .TXT and/or .CSV extension.

For typing or pasting user data, tap on the “Tap to load data…” button, then tap on the console button on the right to open the data input console. Enter signal values into the data input console as per the correct format and click on the “Done” button.

Please note that the App will check all incoming user data and proceed with the analysis only if the data format is correct (see Data Format section for details).

MODWT Filters: Wave Whiz supports all 25 MODWT filters described in Percival et al.’s book “Wavelet methods for time series analysis”[14]. Users can undertake various MODWT-based analyses using any of these filters. A complete list of the MODWT filters that Wave Whiz supports is presented below:

Extremal Phase (Haar and Daubechies)

1.	Haar	(2-tap)
2.	DB4	(4-tap)
3.	DB6	(6-tap)
4.	DB8	(8-tap)
5.	DB10	(10-tap)
6.	DB12	(12-tap)
7.	DB14	(14-tap)
8.	DB16	(16-tap)
9.	DB18	(18-tap)
10.	DB20	(20-tap)

Least Asymmetric

11.	LA8	(8-tap)
12.	LA10	(10-tap)
13.	LA12	(12-tap)
14.	LA14	(14-tap)
15.	LA16	(16-tap)
16.	LA18	(18-tap)
17.	LA20	(20-tap)

Best Localized

18.	BL14	(14-tap)
19.	BL18	(18-tap)
20.	BL20	(20-tap)

Coiflet

21.	CF6	(6-tap)
22.	CF12	(12-tap)
23.	CF18	(18-tap)
24.	CF24	(24-tap)
25.	CF30	(30-tap)

Wavelet Analysis (WA): Within the WA tab, users can carry out a level J_{Max} MODWT-based decomposition of a given signal *Y(t)*. Users can choose the desired wavelet filter and decomposition levels (*J*) via pickers inside the smart control panel that can be invoked by tapping on the blue sandwich button at the top-left corner of the screen.

On the decomposition level picker, J_{Max} updates dynamically based on the number of samples (*N*) in the input signal. For *N* samples, $J_{Max} = floor(log_2(N))$. Therefore, if $N = 100$ for a given signal, then $J_{Max} = 6$ on the decomposition level picker.

Inside the smart control panel, users can also choose any of the two boundary conditions, namely, Periodic or Reflection, via a segmented button. For Periodic boundary condition, the signal is analyzed in its original form without any extension. For Reflection boundary condition, the signal is first extended to the right by mirroring and then analyzed. The Reflection boundary condition mitigates boundary distortion of the wavelet sub-series that may arise due to the convolution operation.

Raw data is plotted at the top of the App screen (Plot 1). The various wavelet sub-series $[V_{JMax}(t), W_{JMax}(t), W_{JMax-1}(t), \ldots, W_2(t), W_1(t)]$ are plotted below Plot 1.

All plots are synchronized on the *T-axis* and can be horizontally zoomed and scrolled. Moreover, various WA sub-series plots (Plot 2 onwards) can be vertically scrolled for enabling their visualization with respect to the raw plot (Plot 1).

Multiresolution Analysis (MRA): The functionality of the MRA tab is very similar to the WA tab. The main difference is that the MRA sub-series $[S_{JMax}(t), D_{JMax}(t), D_{JMax-1}(t), \ldots, D_2(t), D_1(t)]$ are linearly additive. That is, $Y(t) = [S_{JMax}(t) + D_J(t)]$.

Denoising (DNG): Within the DNG tab, users can denoise a given signal *Y(t)* by employing the additive property of the MODWT-based MRA. The type of filter, number of decomposition levels (*J*), and boundary condition can be chosen via the smart control panel (as described in the WA and MRA sections).

A slider control in the 0-100% range (with an increment of 1%) is dynamically created inside the smart control panel for each detail $D_J(t)$. Denoising is accomplished via hard thresholding whereby $D_J(t)$ values are set to zero based on the corresponding slider (threshold) values. A slider value of 0% will retain all values of the associated wavelet detail whereas a slider value of 100% will set all values of the associated detail to zero.

After thresholding, the denoised signal is synthesized via the sum of the denoised MRA sub-series, that is, $Y_{Denoised}(t) = [S_{JMax}(t) + D_{J,Denoised}(t)]$.

The denoised signal $Y_{Denoised}(t)$ is plotted in red color over the blue-colored raw signal *Y(t)* (Plot 1). The various denoised sub-series, $D_{J,Denoised}(t)$, are plotted below Plot 1. All plots zoom and scroll in the same manner as described in the WA and MRA sections.

Variance (VAR): Users can carry out a multiscale analysis of variance (ANOVA) within the VAR tab of Wave Whiz. Multiscale VAR is computed from WA coefficients $W_J(t)$ as the sum of squares of these coefficients.

The App supports two methods, namely, Biased and Unbiased, for the estimation of multiscale VAR.

In a Biased estimation of multiscale VAR, the computation is made on the entire length (*N*) of the wavelet coefficients $W_J(t)$. This method is referred to as Biased because the boundary distortion of $W_J(t)$ due to the convolution process is included in the estimate of VAR.

In an Unbiased estimation of multiscale VAR, $W_J(t)$ boundary coefficients that are affected (distorted) due to the convolution process are excluded from the calculation[14].

The type of filter, number of decomposition levels (*J*), and boundary condition can be chosen via the smart control panel (as described in the WA, MRA, and DNG sections). Inside the smart control panel, users can also choose any of the VAR estimation methods, namely, Biased or Unbiased, via a segmented button.

The sampling rate (ΔT) of the input time series ($Y(t)$) is important with regards to multiscale VAR. For a given decomposition level (or scale) *J*, the value of VAR characterizes variance at a scale equal to ($2^{J-1}*\Delta T$). Suppose a signal has a $\Delta T = 0.02$, then, *scale 1* $= 2^0*0.02 = (0.02)$, *scale 2* $= 2^1*0.02 = (0.04)$, etc. The multiscale VAR is computed and reported for each of these scales (*scale 1*, *scale 2*, etc.).

Wave Whiz employs a logarithmic (*log-log*) plot to display multiscale VAR (scale versus VAR) of a given signal $Y(t)$. The logarithmic plot offers a clear visualization of the multiscale VAR and its changes with respect to various scales.

The VAR tab screen of Wave Whiz displays two plots: (i) Plot 1 is the plot of the raw signal $Y(t)$ and (ii) Plot 2 is the logarithmic plot of the multiscale wavelet VAR (scale versus variance). Plot 1 is horizontally zoomable and scrollable.

Dictionary (DICT): In the DICT tab, users can visualize and review any of the 25 MODWT filters $g_J(t)$ and $h_J(t)$ that the App supports at ten scales or decomposition levels ($J = 1, 2, \ldots, 10$). Users can also review/visualize the associated SGFs for all MODWT filters at these ten scales.

The type of filter and number of decomposition levels (*J*) can be chosen via the smart control panel (as described in the WA, MRA, DNG, and VAR sections).

The DICT tab screen of Wave Whiz displays four plots: (i) Plot 1 (top-left) is the plot of the scaling filter $g_J(t)$ at a given scale *J*, (ii) Plot 2 (top-right) is the plot of the wavelet filter $h_J(t)$ at a given scale *J*, (iii) Plot 3 (bottom-left) is the associated SGF plot of the scaling filter $g_J(t)$ at a given scale *J*, and (iv) Plot 4 (bottom-right) is the SGF plot of the associated wavelet filter $h_J(t)$ at a given scale *J*. All these four plots are horizontally zoomable and scrollable.

Results Summary: For each tab, Wave Whiz generates a tabular summary of all analyses displaying important parameters like input signal sample size *N*, sampling rate ΔT, filter length *L*, type of MODWT analysis, VAR values, and filter coefficients. To access the results summary for each tab, click on the right-facing blue arrow at the top-right corner of the screen.

Exporting Processed Data: For each tab, users can export all MODWT-based data and/or save it on their devices locally. All data is exported in CSV format via .TXT files. Click on the export button in top-right corner of the results summary screen and choose the desired TXT files that need to be exported.

Interactive Charting: Wave Whiz generates fully-interactive charts (plots) for all data. All charts for the WA, MRA, and DNG tabs are synchronized on the *T-axis*. Users can pinch the charts horizontally to zoom-in and zoom-out. Users can also scroll the zoomed charts horizontally in both forward and backward directions. Double-tapping on any of the synchronized charts brings all charts to their original un-zoomed state.

Additionally, for the WA, MRA, and DNG tabs, the wavelet sub-series plots (Plot 2 onwards) can be vertically scrolled up and down for enabling their visualization with respect to the raw plot (Plot 1). The vertical scroll feature works best when these plots are scrolled via the extreme left or right edges of the screen.

All charts for the DICT tab and the input signal chart for the VAR tab are horizontally zoomable and scrollable.

The interactive charting feature of the App allows complete and comprehensive visualization and review of all data.

Smart Control Panel: Wave Whiz employs a smart control panel that is movable and vertically pinchable to reduce and increase its height. This menu is also tappable whereby each double tap (on its top bar) reduces and increases its height. The smart control panel thus never comes in the way of visualizing processed data on charts while various parameters are changed and/or fine-tuned.

To access the smart control panel, tap on the blue sandwich button at the top-left corner of the screen for each tab. All controls for each tab are provided inside the smart control panel.

DISCLAIMER

The App Wave Whiz is distributed and sold without any warranties. Any risks associated with the use of the App lies with the user(s) and/or their organization(s). The developer, Saiftron Inc., takes no responsibility for any damage or harm (for example, loss of profit, loss of data, flawed decision-making, etc.) that the use of this App may cause.

REFERENCES

1. Bar-Yam Y. *Dynamics of complex systems.* Reading, Mass.: Addison-Wesley; 1997.
2. Fellman PV, Bar-Yam Y, Minai A. *Conflict and complexity : countering terrorism, insurgency, ethnic and regional violence.* New York: Springer; 2015.
3. Fieguth P. An introduction to complex systems society, ecology, and nonlinear dynamics. Cham, Switzerland: Springer; 2017: http://dx.doi.org/10.1007/978-3-319-44606-6 MIT Access Only.
4. Ahmad S. *A Temporal Pattern Identification and Summarization Method for Complex Time Serial Data*, University of Surrey; 2007.
5. Box GEP, Jenkins GM, Reinsel GC, Ljung GM. *Time series analysis : forecasting and control.* Fifth edition / ed. Hoboken, New Jersey: John Wiley & Sons, Inc.; 2016.
6. Rojas I, Pomares H, Valenzuela O. Advances in time series analysis and forecasting selected contributions from ITISE 2016. *Contributions to statistics,*. Cham: Springer; 2017: http://dx.doi.org/10.1007/978-3-319-55789-2 MIT Access Only.
7. Mandelbrot BB, Hudson RL. *The (Mis)Behaviour of Markets — A Fractal View of Risk, Ruin and Reward.* Profile Books; 2009.
8. Daubechies I. *Ten lectures on wavelets.* Philadelphia, Pa.: Society for Industrial and Applied Mathematics; 1992.
9. Daubechies I. *Different perspectives on wavelets : American Mathematical Society short course, January 11-12, 1993, San Antonio, Texas.* Providence, R.I.: American Mathematical Society; 1993.
10. Gençay R, Selçuk F, Whitcher B. *An introduction to wavelets and other filtering methods in finance and economics.* San Diego, Calif.: Academic Press; 2002.
11. Hubbard BB. *The world according to wavelets : the story of a mathematical technique in the making.* 2nd ed. Wellesley, Mass: A.K. Peters; 1998.
12. Mallat SG. *A wavelet tour of signal processing.* 2nd ed. San Diego: Academic Press; 1999.
13. Mallat SG. *A wavelet tour of signal processing : the sparse way.* 3rd ed. Amsterdam ; Boston: Elsevier/Academic Press; 2009.
14. Percival DB, Walden AT. *Wavelet methods for time series analysis.* Cambridge ; New York: Cambridge University Press; 2000.
15. Zhang Z, Moore JC, Grinsted A. Haar Wavelet Analysis of Climatic Time Series. *International Journal of Wavelets, Multiresolution and Information Processing.* 2014;12(02):11.
16. Ahmad S, Bolic M, Dajani H, Groza V, Batkin I, Rajan S. Measurement of Heart Rate Variability Using an Oscillometric Blood Pressure Monitor. *IEEE Transactions on Instrumentation and Measurement.* 2010;59(10):16.
17. Khalighi S, Sousa T, Oliveira D, Pires G, Nunes U. Efficient feature selection for sleep staging based on maximal overlap discrete wavelet transform and SVM. *Conf Proc IEEE Eng Med Biol Soc.* 2011;2011:3306-3309.
18. Seo Y, Choi Y, Choi J. River Stage Modeling by Combining Maximal Overlap Discrete Wavelet Transform, Support Vector Machines and Genetic Algorithm. *Water.* 2017;9(7):24.