LabVIEW™

Advanced Signal Processing Toolkit

Wavelet Analysis Tools User Manual

June 2008
371533B-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 5050 9800, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511, France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466, New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the info code feedback.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

Chapter 4
Signal Processing with Discrete Wavelets

Chapter 5
Interactively Designing Discrete Wavelets

Chapter 6
Integer Wavelet Transform

Appendix A
Technical Support and Professional Services

About This Manual

This manual provides information about the wavelet analysis tools in the LabVIEW Advanced Signal Processing Toolkit, including different types of methods that you can use to perform wavelet signal processing, theoretical basis for each type of method, and application examples based on the wavelet transform-based methods.

Conventions

The following conventions appear in this manual:

» The **»** symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

 This icon denotes a note, which alerts you to important information.

bold Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic Italic text denotes variables, emphasis, a cross-reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**
- *Getting Started with LabVIEW*, available by selecting **Start» All Programs»National Instruments»LabVIEW *x.x*» LabVIEW Manuals**, where ***x.x*** is the version of LabVIEW you installed, and opening `LV_Getting_Started.pdf`. This manual also is available by navigating to the `labview\manuals` directory and opening `LV_Getting_Started.pdf`. The *LabVIEW Help* includes all the content in this manual.
- *LabVIEW Fundamentals*, available by selecting **Start»All Programs» National Instruments»LabVIEW *x.x*»LabVIEW Manuals**, where ***x.x*** is the version of LabVIEW you installed, and opening `LV_Fundamentals.pdf`. This manual also is available by navigating to the `labview\manuals` directory and opening `LV_Fundamentals.pdf`. The *LabVIEW Help* includes all the content in this manual.
- The LabVIEW Digital Filter Design Toolkit documentation

Note The following resources offer useful background information on the general concepts discussed in this documentation. These resources are provided for general informational purposes only and are not affiliated, sponsored, or endorsed by National Instruments. The content of these resources is not a representation of, may not correspond to, and does not imply current or future functionality in the Wavelet Analysis Tools or any other National Instruments product.

- Mallat, Stephane. *A Wavelet Tour of Signal Processing*. 2nd ed. San Diego, California: Academic Press, 1999.
- Qian, Shie. *Introduction to Time-Frequency and Wavelet Transforms*. Upper Saddle River, New Jersey: Prentice Hall PTR, 2001.

1

Introduction to Wavelet Signal Processing

Wavelets are functions that you can use to decompose signals, similar to how you use complex sinusoids in the Fourier transform to decompose signals. The wavelet transform computes the inner products of the analyzed signal and a family of wavelets.

In contrast with sinusoids, wavelets are localized in both the time and frequency domains, so wavelet signal processing is suitable for nonstationary signals, whose spectral content changes over time. The adaptive time-frequency resolution of wavelet signal processing enables you to perform multiresolution analysis on nonstationary signals. The properties of wavelets and the flexibility to select wavelets make wavelet signal processing a beneficial tool for feature extraction applications. Refer to the *Benefits of Wavelet Signal Processing* section of Chapter 2, *Understanding Wavelet Signal Processing*, for information about the benefits of wavelet signal processing.

This chapter describes the application areas of wavelet signal processing and provides an overview of the LabVIEW Wavelet Analysis Tools.

Wavelet Signal Processing Application Areas

You can use wavelets in a variety of signal processing applications, such as analyzing signals at different scales, reducing noise, compressing data, and extracting features of signals. This section discusses these application areas by analyzing signals and images with the Wavelet Analysis Tools.

The Wavelet Analysis Tools provide example VIs for each application area. In the **Browse** tab of the NI Example Finder, you can view these example VIs by selecting **Toolkits and Modules»Wavelet Analysis» Applications**. Refer to the *Finding Example VIs* section of this chapter for information about launching the NI Example Finder.

Multiscale Analysis

Multiscale analysis involves looking at a signal at different time and frequency scales. Wavelet transform-based multiscale analysis helps you understand both the long-term trends and the short-term variations of a signal simultaneously.

Figure 1-1 shows a multiscale analysis of a Standard & Poor's (S&P) 500 stock index during the years 1947 through 1993. The **S&P 500 Index** graph displays the monthly S&P 500 indexes. The other three graphs are the results of wavelet analysis. The **Long-Term Trend** graph is the result with a large time scale, which describes the long-term trend of the stock movement. The **Short-Term Variation** and **Medium-Term Variation** graphs describe the magnitudes of the short-term variation and medium-term variation, respectively.

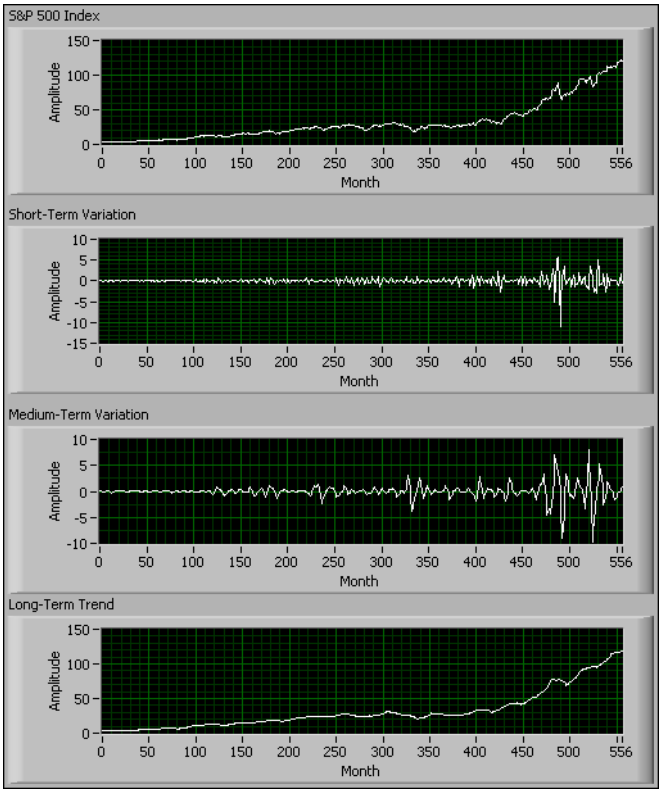

Figure 1-1. Multiscale Analysis of the S&P 500 Stock Index

In the NI Example Finder, refer to the **Multiscale Analysis VI** for more information about performing wavelet transform-based multiresolution analysis on stock indexes.

Noise Reduction

One of the most effective applications of wavelets in signal processing is denoising, or reducing noise in a signal. The wavelet transform-based method can produce much higher denoising quality than conventional methods. Furthermore, the wavelet transform-based method retains the details of a signal after denoising.

Figure 1-2 shows a signal with noise and the denoised signal using the wavelet transform-based method.

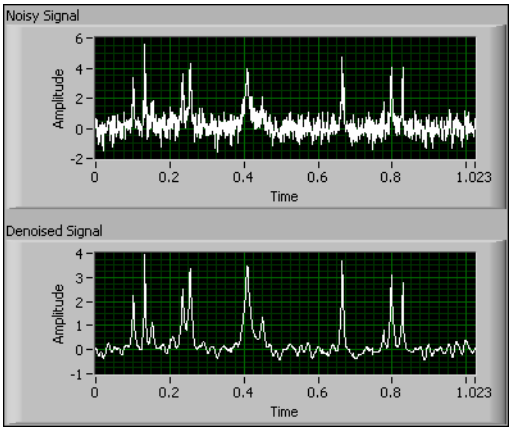

Figure 1-2. Noise Reduction

With the wavelet transform, you can reduce the noise in the signal in the **Noisy Signal** graph. The resulting signal in the **Denoised Signal** graph contains less noise and retains the details of the original signal.

In the NI Example Finder, refer to the **Noise Reduction VI** for more information about performing wavelet transform-based denoising on signals.

Compression

In many applications, storage and transmission resources limit performance. Thus, data compression has become an important topic in information theory. Usually, you can achieve compression by converting a source signal into a sparse representation, which includes a small number

of nonzero values, and then encoding the sparse representation with a low bit rate. The wavelet transform, as a time-scale representation method, generates large coefficients only around discontinuities. So the wavelet transform is a useful tool to convert signals to sparse representations.

In the NI Example Finder, refer to the **ECG Compression VI** for more information about performing wavelet transform-based compression on electrocardiogram (ECG) signals.

Feature Extraction

Extracting relevant features is a key step when you analyze and interpret signals and images. Signals and images are characterized by local features, such as peaks, edges, and breakdown points. The wavelet transform-based methods are typically useful when the target features consist of rapid changes, such as the sound caused by engine knocking. Wavelet signal processing is suitable for extracting the local features of signals because wavelets are localized in both the time and frequency domains.

Figure 1-3 shows an image and the associated edge maps detected at different levels of resolutions using the wavelet transform-based method. Conventional methods process an image at a single resolution and return a binary edge map. The wavelet transform-based method processes an image at multiple levels of resolution and returns a series of grey-level edge maps at different resolutions.

Figure 1-3. Image Edge Detection

A large level value corresponds to an edge map with low resolution. You can obtain the global profile of the image in a low-resolution edge map and the detailed texture of the image in a high-resolution edge map. You also can form a multiresolution edge detection method by examining the edge maps from the low resolution to the high resolution. With the multiresolution edge detection method, you can locate an object of interest in the image reliably and accurately, even under noisy conditions.

In the NI Example Finder, refer to the **Image Edge Detection VI** for more information about performing wavelet transform-based edge detection on image files.

Overview of LabVIEW Wavelet Analysis Tools

The Wavelet Analysis Tools provide a collection of Wavelet Analysis VIs that assist you in processing signals in the LabVIEW environment. You can use the Continuous Wavelet VIs, the Discrete Wavelet VIs, and the Wavelet Packet VIs to perform the continuous wavelet transform, the discrete wavelet transform, the undecimated wavelet transform, the integer wavelet transform, and the wavelet packet decomposition. You can use the Feature Extraction VIs to detrend and denoise a signal. You also can use these VIs to detect the peaks and edges of a signal. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about the Wavelet Analysis VIs.

The Wavelet Analysis Tools provide a collection of commonly used continuous wavelets, such as Mexican Hat, Meyer, and Morlet, and a collection of commonly used discrete wavelets, such as the Daubechies, Haar, Coiflet, and biorthogonal wavelets. Refer to the *Selecting an Appropriate Discrete Wavelet* section of Chapter 4, *Signal Processing with Discrete Wavelets*, for information about the collection of discrete wavelets. You also can create a discrete wavelet that best matches the signal you analyze using the Wavelet Design Express VI. Refer to Chapter 5, *Interactively Designing Discrete Wavelets*, for information about designing a wavelet.

The Wavelet Analysis Tools contain Express VIs that provide interfaces for signal processing and analysis. These Express VIs enable you to specify parameters and settings for an analysis and see the results immediately. For example, the Wavelet Denoise Express VI graphs both the original and denoised signals. You can see the denoised signal immediately as you select a wavelet, specify a threshold, and set other parameters. The Wavelet Analysis Tools also provide Express VIs for multiresolution analysis, wavelet design, and wavelet packet decomposition.

Finding Example VIs

The Wavelet Analysis Tools also provide example VIs that you can use and incorporate into the VIs that you create. You can modify an example VI to fit an application, or you can copy and paste from one or more examples into a VI that you create. You can find the examples using the NI Example Finder. Select **Help»Find Example** to launch the NI Example Finder. You also can select the **Examples** or **Find Examples** options on the **Getting Started** window, which appears when you launch LabVIEW, to launch the NI Example Finder.

Related Signal Processing Tools

In signal processing, you usually categorize signals into two types: stationary and nonstationary. For stationary signals, you assume that the spectral content of stationary signals does not change as a function of time, space, or some other independent variable. For nonstationary signals, you assume that the spectral content changes over time, space, or some other independent variable. For example, you might work under the assumption that an engine vibration signal is stationary when an engine is running at a constant speed and nonstationary when an engine is running up or down.

Nonstationary signals are categorized into two types according to how the spectral content changes over time: evolutionary and transient. The spectral contents of evolutionary signals change over time slowly. Evolutionary signals usually contain time-varying harmonics. The time-varying harmonics relate to the underlying periodic time-varying characteristic of the system that generates signals. Evolutionary signals also can contain time-varying broadband spectral contents. Transient signals are the short-time events in a nonstationary signal, such as peaks, edges, breakdown points, and start and end of bursts. Transient signals usually vary over time and you typically cannot predict the occurrence exactly.

The LabVIEW Advanced Signal Processing Toolkit contains the following tools and toolkit that you can use to perform signal analysis and processing:

- Wavelet Analysis Tools
- LabVIEW Time Series Analysis Tools
- LabVIEW Time Frequency Analysis Tools
- LabVIEW Digital Filter Design Toolkit

To extract the underlying information of a signal effectively, you need to choose an appropriate analysis tool based on the following suggestions:

- For stationary signals, use the Time Series Analysis Tools or the Digital Filter Design Toolkit. LabVIEW also includes an extensive set of tools for signal processing and analysis. The Time Series Analysis Tools provide VIs for preprocessing signals, estimating the statistical parameters of signals, building models of signals, and estimating the power spectrum, the high-order power spectrum, and the cepstrum of signals. The Digital Filter Toolkit provides tools for designing, analyzing, and simulating floating-point and fixed-point digital filters and tools for generating code for DSP or FPGA targets.
- For evolutionary signals, use the Time Frequency Analysis Tools, which include VIs and Express VIs for linear and quadratic time-frequency analysis methods, including the linear discrete Gabor transform and expansion, the linear adaptive transform and expansion, the quadratic Gabor spectrogram, and the quadratic adaptive spectrogram. The Time Frequency Analysis Tools also include VIs to extract features from a signal, such as the mean instantaneous frequency, the mean instantaneous bandwidth, the group delay, and the marginal integration.
- For both evolutionary signals and transient signals, use the Wavelet Analysis Tools. Refer to the *Overview of LabVIEW Wavelet Analysis Tools* section of this chapter for information about the Wavelet Analysis Tools.

2

Understanding Wavelet Signal Processing

This chapter introduces wavelets and the wavelet transform and describes the benefits of wavelet signal processing in detail.

Wavelet and Wavelet Transform

Just as the Fourier transform decomposes a signal into a family of complex sinusoids, the wavelet transform decomposes a signal into a family of wavelets. Unlike sinusoids, which are symmetric, smooth, and regular, wavelets can be either symmetric or asymmetric, sharp or smooth, regular or irregular. Figure 2-1 shows a sine wave, the db02 wavelet, and the FBI wavelet.

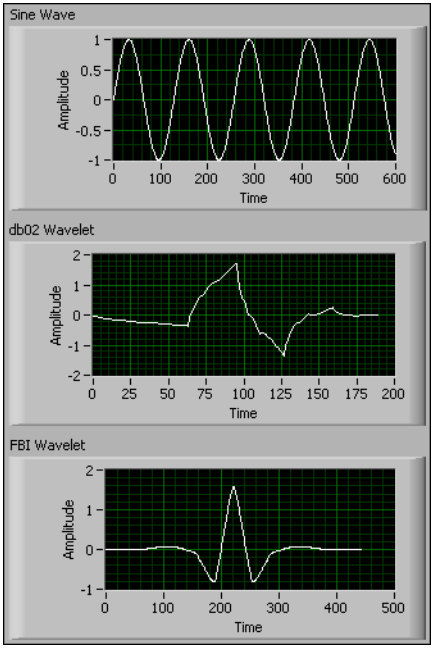

Figure 2-1. Sine Wave versus Wavelets

In Figure 2-1, you can see that the **Sine Wave** is symmetric, smooth, and regular. The **db02 Wavelet** is asymmetric, sharp, and irregular. The **FBI Wavelet** is symmetric, smooth, and regular. You also can see that a sine wave has an infinite length, whereas a wavelet has a finite length.

For different types of signals, you can select different types of wavelets that best match the features of the signal you want to analyze. Therefore, you can perform wavelet signal processing and generate reliable results about the underlying information of a signal.

The family of wavelets contains the dilated and translated versions of a prototype function. Traditionally, the prototype function is called a mother wavelet. The scale and shift of wavelets determine how the mother wavelet dilates and translates along the time or space axis. A scale factor greater than one corresponds to a dilation of the mother wavelet along the horizontal axis, and a positive shift corresponds to a translation to the right of the scaled wavelet along the horizontal axis. Figure 2-2 shows the db02 mother wavelet and the associated dilated and translated wavelets with different scale factors and shift values.

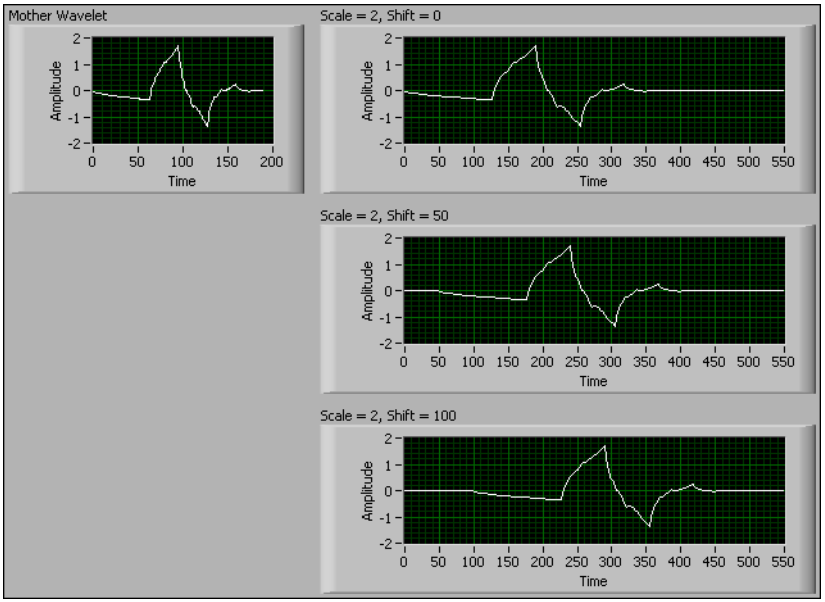

Figure 2-2. Dilations and Translations of the db02 Wavelet

The wavelet transform computes the inner products of a signal with a family of wavelets. The wavelet transform tools are categorized into continuous wavelet tools and discrete wavelet tools. Usually, you use the continuous wavelet tools for signal analysis, such as self-similarity analysis and time-frequency analysis. You use the discrete wavelet tools for both signal analysis and signal processing, such as noise reduction, data compression, peak detection and so on. Refer to Chapter 3, *Signal Processing with Continuous Wavelets*, for information about the continuous wavelet tools. Refer to Chapter 4, *Signal Processing with Discrete Wavelets*, for information about the discrete wavelet tools.

Benefits of Wavelet Signal Processing

Wavelet signal processing is different from other signal processing methods because of the unique properties of wavelets. For example, wavelets are irregular in shape and finite in length. Wavelet signal processing can represent signals sparsely, capture the transient features of signals, and enable signal analysis at multiple resolutions.

Sparse Representation

Wavelets are localized in both the time and frequency domains because wavelets have limited time duration and frequency bandwidth. The wavelet transform can represent a signal with a few coefficients because of the localization property of wavelets. Figure 2-3 shows the waveform of the Doppler signal.

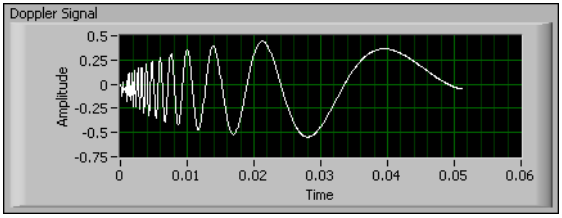

Figure 2-3. Waveform of the Doppler Signal

Figure 2-4 shows the discrete wavelet transform (DWT) coefficients of the Doppler signal. Refer to Chapter 4, *Signal Processing with Discrete Wavelets*, for more information about the DWT.

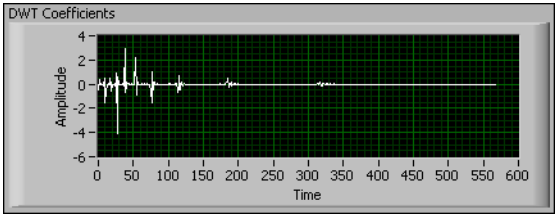

Figure 2-4. DWT Coefficients of the Doppler Signal

In Figure 2-4, most of the DWT coefficients are zero, which indicates that the wavelet transform is a useful method to represent signals sparsely and compactly. Therefore, you usually use the DWT in some signal compression applications.

Transient Feature Detection

Transient features are sudden changes or discontinuities in a signal. A transient feature can be generated by the impulsive action of a system and frequently implies a causal relationship to an event. For example, heartbeats generate peaks in an electrocardiogram (ECG) signal.

Transient features generally are not smooth and are of short duration. Because wavelets are flexible in shape and have short time durations, the wavelet signal processing methods can capture transient features precisely. Figure 2-5 shows an ECG signal and the peaks detected with the wavelet transform-based method. This method locates the peaks of the ECG signal precisely.

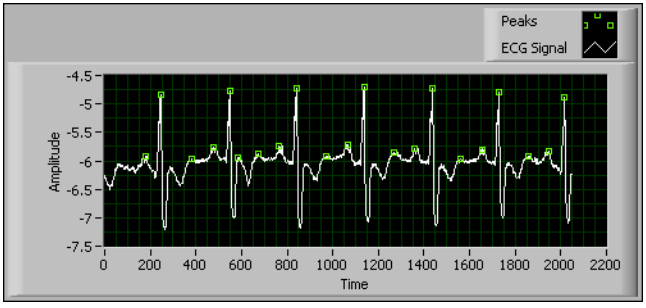

Figure 2-5. Peaks in the ECG Signal

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Wavelet Analysis»Applications» ECG QRS Complex Detection VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

The LabVIEW Wavelet Analysis Tools provide many types of wavelets, such as the Daubechies, Haar, and Coiflet wavelets. Refer to Chapter 4, *Signal Processing with Discrete Wavelets*, for information about these wavelets and applying them to the wavelet transforms.

Multiple Resolutions

Signals usually contain both low-frequency components and high-frequency components. Low-frequency components vary slowly with time and require fine frequency resolution but coarse time resolution. High-frequency components vary quickly with time and require fine time resolution but coarse frequency resolution. You need to use a multiresolution analysis (MRA) method to analyze a signal that contains both low- and high-frequency components.

Wavelet signal processing is naturally an MRA method because of the dilation process. Figure 2-6 shows the wavelets with different dilations and their corresponding power spectra.

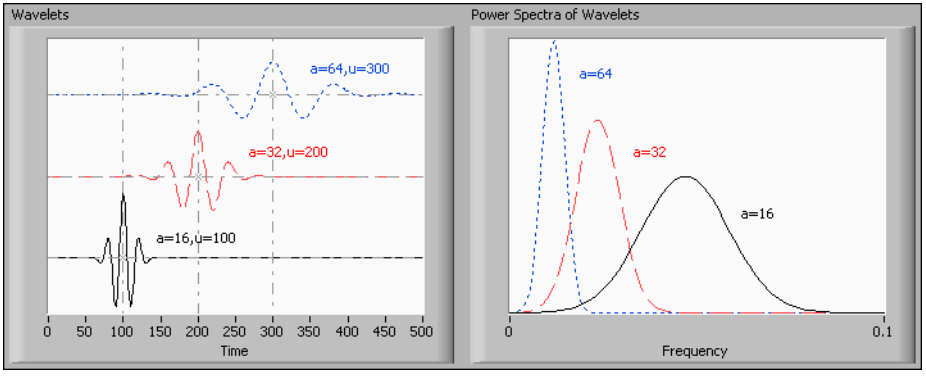

Figure 2-6. Wavelets and the Corresponding Power Spectra

The **Wavelets** graph contains three wavelets with different scales and translations. The **Power Spectra of Wavelets** graph shows the power spectra of the three wavelets, where *a* and *u* represent the scale and shift of the wavelets, respectively. Figure 2-6 shows that a wavelet with a small scale has a short time duration, a wide frequency bandwidth, and a high

central frequency. This figure also shows that a wavelet with a large scale has a long time duration, a narrow frequency bandwidth, and a low central frequency.

The time duration and frequency bandwidth determine the time and frequency resolutions of a wavelet, respectively. A long time duration means coarse time resolution. A wide frequency bandwidth means coarse frequency resolution. Figure 2-7 shows the time and frequency resolutions of the three wavelets with three boxes in the time-frequency domain. The heights and widths of the boxes represent the frequency and time resolutions of the wavelets, respectively. This figure shows that a wavelet with a small scale has fine time resolution but coarse frequency resolution and that a wavelet with a large scale has fine frequency resolution but coarse time resolution.

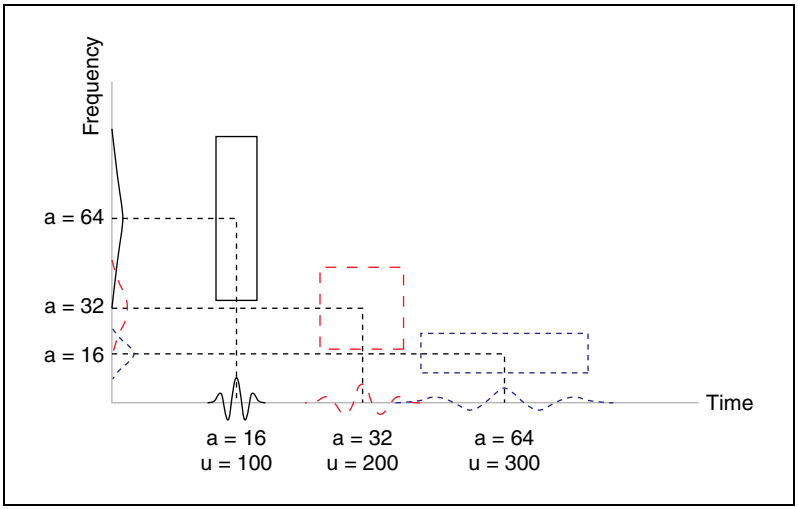

Figure 2-7. Time and Frequency Resolutions of Wavelets

The fine frequency resolution of large-scale wavelets enables you to measure the frequency of the slow variation components in a signal. The fine time resolution of small-scale wavelets enables you to detect the fast variation components in a signal. Therefore, wavelet signal processing is a useful multiresolution analysis tool. Refer to the *Discrete Wavelet Transform for Multiresolution Analysis* section of Chapter 4, *Signal Processing with Discrete Wavelets*, for information about performing multiresolution analysis.

3

Signal Processing with Continuous Wavelets

You can use continuous wavelet tools to perform wavelet transforms on signals that are defined in continuous time. Unlike discrete wavelet tools, which operate on sampled-data signals, continuous wavelet tools operate on signals that are defined for all time over a time region of interest, though the computations are done numerically in discrete time.

The LabVIEW Wavelet Analysis Tools provide two continuous wavelet tools: the continuous wavelet transform (CWT) and the analytic wavelet transform (AWT). The AWT retains both the magnitude and phase information of signals in the time-scale or time-frequency domain, whereas the CWT retains only the magnitude information. The CWT is simpler because the results of the CWT are real values if both the wavelet and the signal are real. The results of the AWT normally are complex values.

From a mathematical point of view, both the CWT and AWT add informational redundancy because the number of the resulting wavelet coefficients in the time-scale or time-frequency domain is larger than the number of time samples in the original signal. Excess redundancy generally is not desirable because more computations and more memory are required to process signals with excess redundancy. However, excess redundancy can be helpful for some applications, such as singularity and cusp extraction, time-frequency analysis of nonstationary signals, and self-similarity analysis of fractal signals.

This chapter explains both the CWT and the AWT in detail and provides an application example that uses the CWT.

Continuous Wavelet Transform

Mathematically, the CWT computes the inner products of a continuous signal with a set of continuous wavelets according to the following equation:

$$WT_{u,a} = \langle s, \psi_{u,a} \rangle = \int_{-\infty}^{\infty} s(t)\psi^*_{u,a}(t)dt$$

where

$$\psi_{u,a} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-u}{a}\right)$$

$WT_{u,a}$ is the resulting wavelet coefficients. $\psi_{u,a}$ denotes a continuous wavelet, where u is the shift factor and a is the scale factor of the wavelet. $\psi^*_{u,a}$ is the complex conjugate of $\psi_{u,a}$. For the continuous-time signal $s(t)$, the scale factor must be a positive real number, whereas the shift factor can be any real number. If the continuous wavelet $\psi_{u,a}$ meets the admissibility condition[1], you can use the computed wavelet coefficients to reconstruct the original signal $s(t)$.

However, you seldom use the above integration to compute the CWT because of the following reasons:

- The majority of real-world signals that you encounter are available as discrete-time samples. The analytical form of the signal $s(t)$ usually is not accessible.
- The closed-form solution of the integration does not exist except for very special cases.

For these reasons, you usually select a set of discrete values for the scales and shifts of the continuous wavelets and then compute the CWT numerically.

Use the WA Continuous Wavelet Transform VI to compute the CWT by specifying a set of integer values or arbitrary real positive values for the scales and a set of equal-increment values for the shifts. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about this VI.

[1] Shie Qian. *Introduction to Time-Frequency and Wavelet Transforms*. Upper Saddle River, New Jersey: Prentice Hall PTR, 2001.

Figure 3-1 shows the procedure that the WA Continuous Wavelet Transform VI follows.

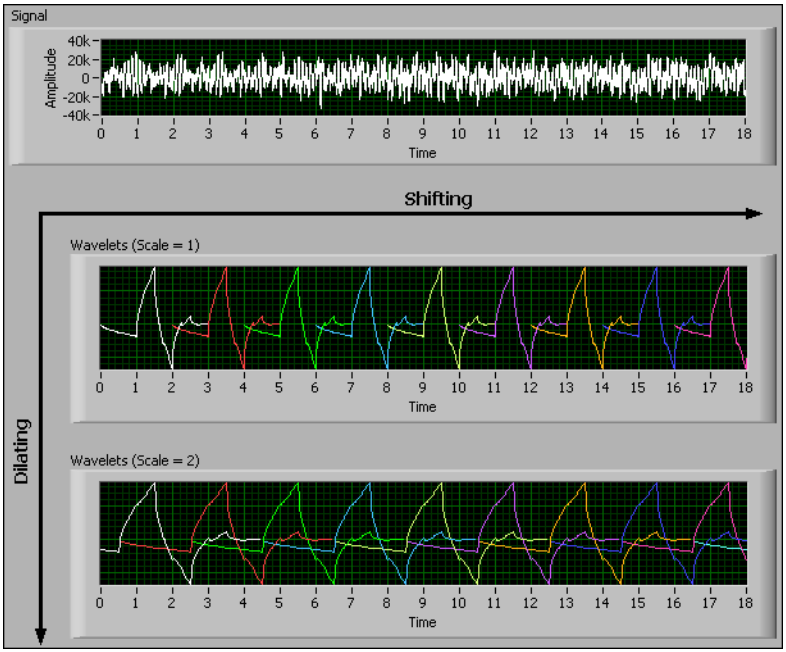

Figure 3-1. Procedure of the Continuous Wavelet Transform

The procedure involves the following steps:

1. Shifts a specified wavelet continuously along the time axis.
2. Computes the inner product of each shifted wavelet and the analyzed signal.
3. Dilates the wavelet based on the scale you specify.
4. Repeats steps 1 through 3 till the process reaches the maximum scale you specify.

The output of the CWT is the CWT coefficients, which reflect the similarity between the analyzed signal and the wavelets.

You also can compute the squares of the CWT coefficients and form a scalogram, which is analogous to the spectrogram in time-frequency analysis. In signal processing, scalograms are useful in pattern-matching applications and discontinuity detections. If a signal contains different scale characteristics over time, the scalogram can present a time-scale view

of the signal, which is more useful than the time-frequency view of that signal.

Figure 3-2 shows a test signal, the Devil's Staircase fractal signal. An important characteristic of a fractal signal is self-similarity.

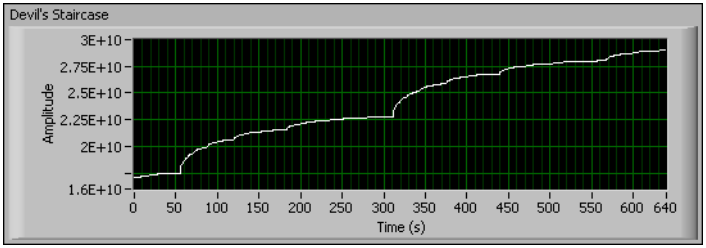

Figure 3-2. Devil's Staircase Signal

Figure 3-3 shows the scalogram and the STFT spectrogram of the fractal signal, respectively.

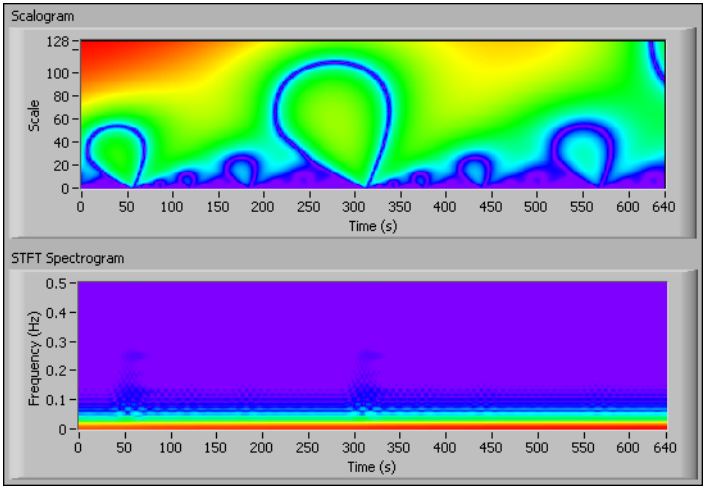

Figure 3-3. Scalogram versus STFT Spectrogram of the Devil's Staircase Signal

In Figure 3-3, you can see the self-similarity characteristic of the signal clearly in the **Scalogram** graph but not in the **STFT Spectrogram** graph. The **STFT Spectrogram** graph displays the conventional time-frequency analysis result of the signal. Refer to the *Time Frequency Analysis Tools User Manual* for more information about STFT spectrograms.

The CWT has the following general disadvantages:

- The CWT adds excess redundancy and is computationally intensive, so you usually use this transform in offline analysis applications.
- The CWT does not provide the phase information of the analyzed signal. For applications in which the phase information is useful, use the AWT. Refer to the *Analytic Wavelet Transform* section of this chapter for information about the AWT.
- You cannot reconstruct the original signal from the CWT coefficients. For applications that require signal reconstruction, use the discrete wavelet tools. Refer to Chapter 4, *Signal Processing with Discrete Wavelets*, for information about the discrete wavelet tools.

Application Example: Breakdown Point Detection

One useful CWT application is the detection of abrupt discontinuities or breakdown points in a signal. Figure 3-4 shows an example that detects the breakdown points in a noise-contaminated signal using the WA Continuous Wavelet Transform VI.

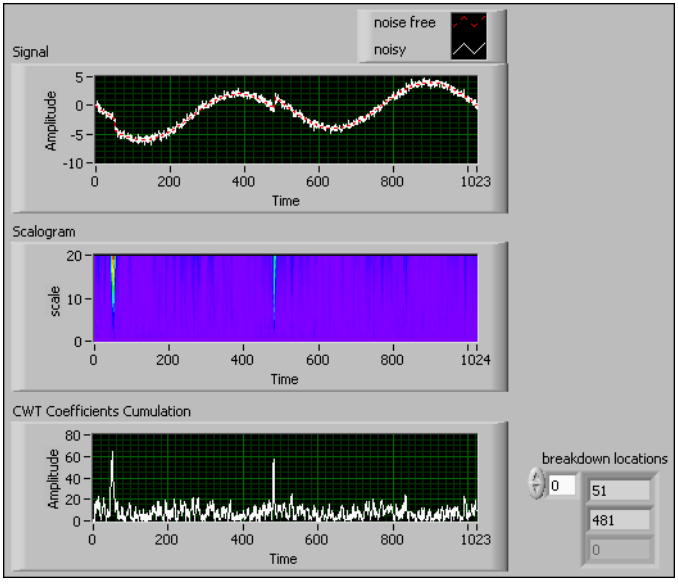

Figure 3-4. Breakdown Points in the Noise-Contaminated HeaviSine Signal

The **Signal** graph in Figure 3-4 shows the HeaviSine signal, which is a common wavelet test signal, contaminated with white noise. The HeaviSine signal is a sinusoid with two breakdown points—one at 51 and the other at 481. The CWT precisely shows the positions of the two breakdown points by doing the following steps:

1. Computes the CWT using the Haar wavelet.
2. Calculates the squares of the CWT coefficients of the signal and forms a scalogram, as shown in the **Scalogram** graph in Figure 3-4.
3. Cumulates the CWT coefficients along the **scale** axis and forms a cumulation plot, as shown in the **CWT Coefficients Cumulation** graph in Figure 3-4.
4. Detects the peak locations in the **CWT Coefficients Cumulation** graph. The peak locations are where the breakdown points exist.

Breakdown points and noise can generate large values in the resulting coefficients. Breakdown points generate large positive or negative coefficients at all scales. Noise generates positive coefficients at some scales and negative coefficients at other scales. If you accumulate the coefficients at all scales, the coefficients of breakdown points are enlarged while the coefficients of noise at different scales counteract one another. Therefore, the peaks in the **CWT Coefficients Cumulation** graph correspond only to the breakdown points.

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Wavelet Analysis»Applications» Breakdown Point Detection VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

Analytic Wavelet Transform

The AWT is a wavelet transform that provides both the magnitude and phase information of signals in the time-scale or time-frequency domain. The magnitude information returned by the AWT describes the envelopes of signals. The phase information encodes the time-related characteristics of signals, for example, the location of a cusp. You usually use the magnitude information for time-frequency analysis and phase information for applications such as instantaneous frequency estimation.

The AWT computes the inner products of the analyzed signal and a set of complex Morlet wavelets. This transform is called the analytic wavelet transform because the complex Morlet wavelets are analytic, that is, the power spectra of the Morlet wavelets are zero at negative frequencies. The resulting AWT coefficients are complex numbers. These coefficients measure the similarity between the analyzed signal and the complex Morlet wavelets. The AWT is just one type of complex continuous wavelet transform.

Use the WA Analytic Wavelet Transform VI to compute the AWT. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about this VI.

Scale and Frequency

Wavelets are functions of time and scale, so you can consider a wavelet transform as a tool that produces a time-scale representation of signals. You also can consider the time-scale representation of signals as a time-frequency representation, because wavelets with different scales measure the corresponding frequency components in the signal. The frequency of a wavelet is inversely proportional to the scale factor. Refer to the *Multiple Resolutions* section of Chapter 2, *Understanding Wavelet Signal Processing*, for information about the relationship between the scale factor and the frequency of a wavelet.

Using the WA Analytic Wavelet Transform VI, you can specify different settings for the scale factor to compute the AWT. When you set **scale sampling method** to **even scale**, this VI computes the wavelet coefficients at evenly-distributed integer scales. You usually use the **even scale** option to obtain the time-scale representation of a signal. When you set **scale sampling method** to **even freq**, this VI computes the wavelet coefficients at scales with evenly-distributed frequencies. Notice that the scales are not evenly-distributed. You usually use the **even freq** option to obtain the time-frequency representation of a signal.

Because the time and frequency resolutions of wavelets are adaptive, the AWT provides adaptive time and frequency resolutions. Conventional time-frequency analysis methods, such as the short-time Fourier transform (STFT), only provide uniform time and frequency resolutions in the whole time-frequency domain. Refer to the *Time Frequency Analysis Tools User Manual* for information about the STFT and other conventional time-frequency analysis methods.

Figure 3-5 shows a common wavelet test signal, the HypChirps signal. This signal contains two frequency components, which are hyperbolic functions over time. The frequency components change slowly at the beginning and rapidly at the end.

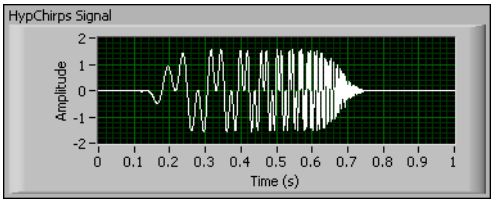

Figure 3-5. The HypChirps Signal

Figure 3-6 shows two representations of this HypChirps signal in the time-frequency domain based on the STFT method.

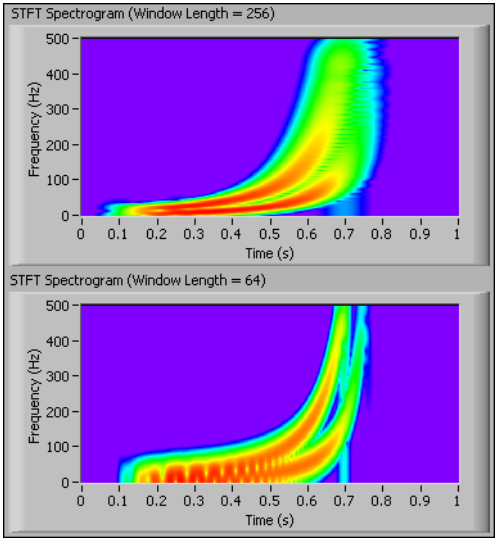

Figure 3-6. STFT Spectrograms of the HypChirps Signal

In Figure 3-6, if you use a relatively long window, 256, you obtain fine frequency resolution and coarse time resolution. Therefore, you can distinguish the frequency components of the HypChirps signal at lower frequencies with a long window. If you use a short window, 64, you obtain coarse frequency resolution and fine time resolution. Therefore, you can distinguish the frequency components of the HypChirps signal at higher frequencies with a short window. However, you cannot distinguish the

two frequency components at both low and high frequencies in either of the STFT spectrograms.

Figure 3-7 shows the tiling of the STFT-based time-frequency representation.

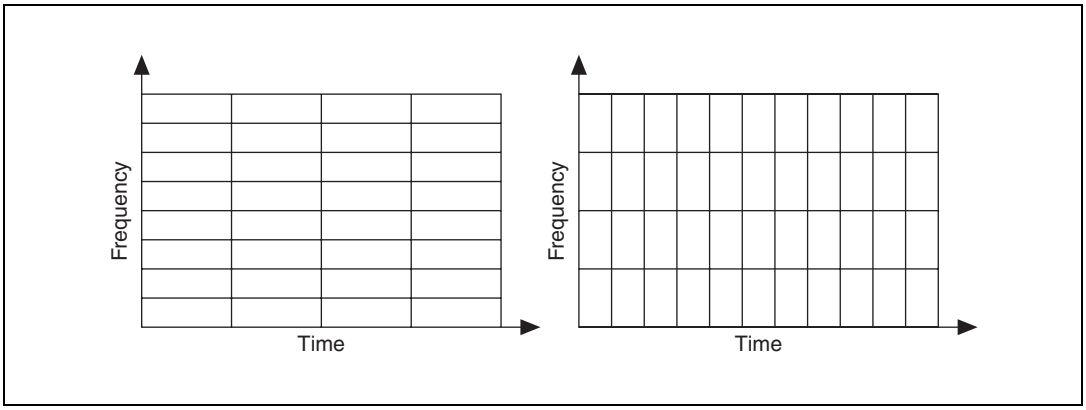

Figure 3-7. Tiling of STFT-Based Time-Frequency Representation

In Figure 3-7, you can see that the STFT spectrogram has uniform time-frequency resolution across the whole time-frequency domain. You can balance the time-frequency resolution by adjusting the window length. The left tiling diagram provides better frequency resolution in the **STFT Spectrogram (Window Length = 256)** graph of Figure 3-6. The right tiling diagram shows better time resolution in the **STFT Spectrogram (Window Length = 64)** graph of Figure 3-6. However, you cannot achieve high time resolution and frequency resolution simultaneously.

Figure 3-8 shows the AWT-based time-frequency representation of the HypChirps signal. In the **Scalogram** graph, you can distinguish the two frequency components at both low and high frequencies.

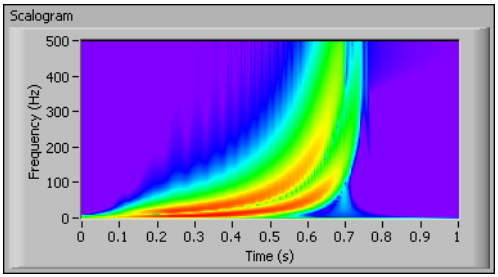

Figure 3-8. AWT-Based Scalogram of the HypChirps Signal

Figure 3-9 shows the tiling of the AWT-based time-frequency representation that provides fine frequency resolution at low frequencies and fine time resolution at high frequencies.

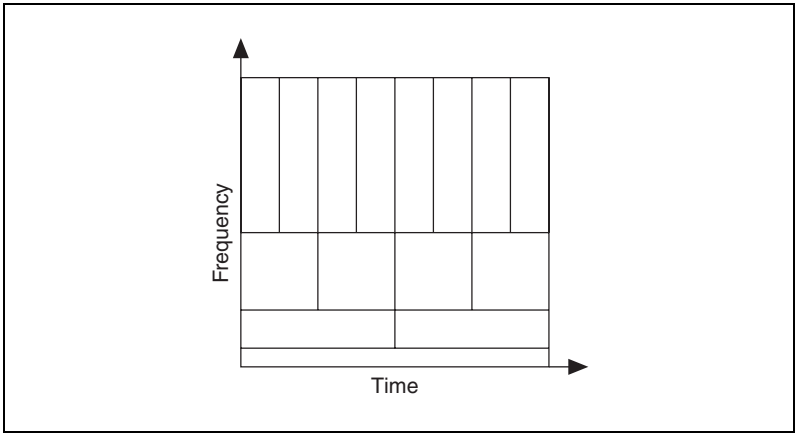

Figure 3-9. Tiling of Wavelet-Based Time-Frequency Representation

Similar to the CWT, the AWT also adds excess information redundancy and is computationally intensive. Moreover, you cannot reconstruct the original signal from the AWT coefficients.

Wavelet Normalization: Energy versus Amplitude

In wavelet analysis, wavelets at different scales often have the same energy. Because both the center frequency and the bandwidth of a wavelet are inversely proportional to the scale factor, the wavelet at a larger scale has a higher magnitude response than a wavelet at a smaller scale. Figure 3-10 shows the Fourier magnitude spectra of different wavelets with energy normalization.

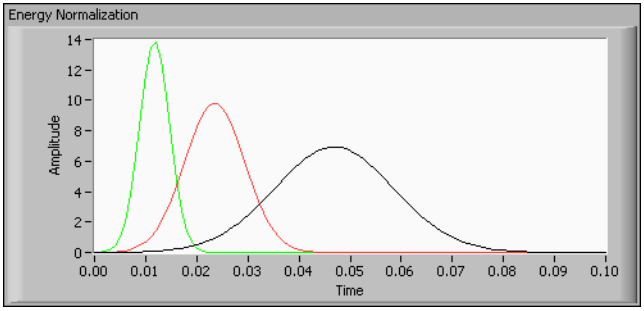

Figure 3-10. Magnitude Spectra of Wavelets with Energy Normalization

However, some real-world applications require that you use a uniform amplitude response to measure the exact amplitude of the signal components, as shown in Figure 3-11.

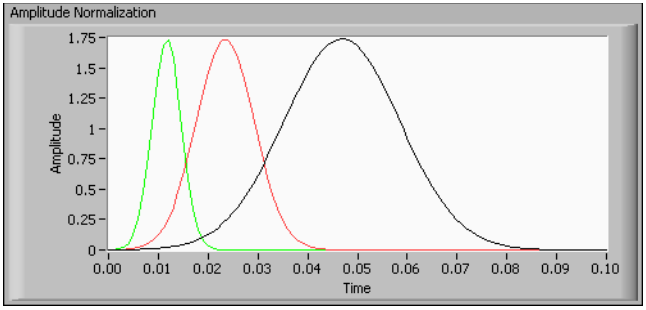

Figure 3-11. Magnitude Spectra of Wavelets with Amplitude Normalization

With the WA Analytic Wavelet Transform VI, you can analyze a signal based on amplitude normalization by selecting **amplitude** in the **normalization** list. If you set **scale sampling method** to **even freq** and set **normalization** to **amplitude**, the WA Analytic Wavelet Transform VI generates the scalogram of the HypChirps signal, as shown in Figure 3-12.

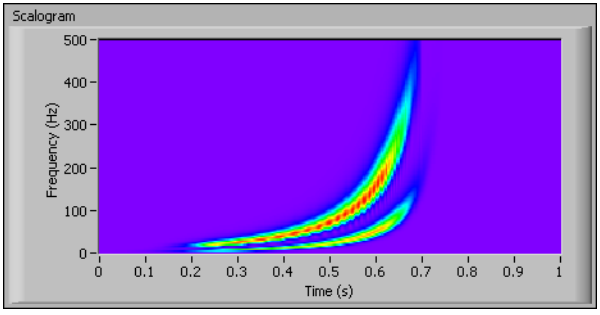

Figure 3-12. Scalogram with Amplitude Normalization

Notice that the magnitude at high frequencies (small scales) also has been enlarged. With amplitude normalization, you can obtain the precise magnitude evolution over time for each hyperbolic chirp.

4

Signal Processing with Discrete Wavelets

Although you can use numerical algorithms to compute continuous wavelet coefficients, as introduced in Chapter 3, *Signal Processing with Continuous Wavelets*, to analyze a signal, the resulting wavelet coefficients are not invertible. You cannot use those wavelet coefficients to recover the original data samples. For applications that require signal reconstruction, the LabVIEW Wavelet Analysis Tools provide the following discrete wavelet tools:

- Discrete wavelet transform (DWT)
- Wavelet packet decomposition and arbitrary path decomposition
- Undecimated wavelet transform (UWT)

You can use the discrete wavelet tools to perform signal analysis and signal processing, including multiresolution analysis, denoising, compression, edge detection, peak detection and others.

This chapter introduces the commonly used discrete wavelets and describes discrete filter banks that you use to implement the wavelet transforms. This chapter also explains each discrete wavelet tool in detail and provides application examples.

Selecting an Appropriate Discrete Wavelet

The Wavelet Analysis Tools provide the following commonly used discrete wavelets:

- Orthogonal wavelets—**Haar**, Daubechies (**db*x***), Coiflets (**coif*x***), and Symmlets (**sym*x***)
- Biorthogonal wavelets—**FBI**, and Biorthogonal (**bior*x_x***)

x indicates the order of the wavelet. The higher the order, the smoother the wavelet.

Orthogonal wavelets are suitable for applications such as signal and image compression and denoising, because the wavelet transform with orthogonal wavelets possesses the same amount of energy as that contained in the original data samples. The energy-conservative property ensures that the inverse wavelet transform does not enlarge the energy of noise suppressed in the wavelet domain. However, the filters associated with orthogonal wavelets are not linear-phase filters. Linear-phase filters maintain a constant time delay for different frequencies and are necessary in many signal and image feature extraction applications, such as peak detection and image edge detection. Biorthogonal wavelets can be linear phase and are suitable for applications that require linear-phase filters.

You also can use the Wavelet Design Express VI to design a customized wavelet. Refer to Chapter 5, *Interactively Designing Discrete Wavelets*, for information about wavelet design.

Discrete Wavelet Transform

Unlike the discrete Fourier transform, which is a discrete version of the Fourier transform, the DWT is not really a discrete version of the continuous wavelet transform. Instead, the DWT is functionally different from the continuous wavelet transform (CWT). To implement the DWT, you use discrete filter banks to compute discrete wavelet coefficients. Two-channel perfect reconstruction (PR) filter banks are a common and efficient way to implement the DWT. Figure 4-1 shows a typical two-channel PR filter bank system.

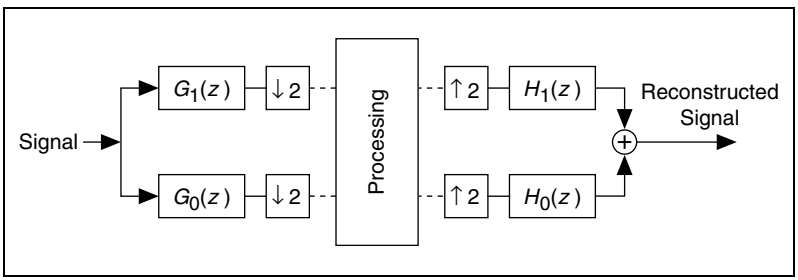

Figure 4-1. Two-Channel Perfect Reconstruction Filter Banks

The signal $X[z]$ first is filtered by a filter bank consisting of $G_0(z)$ and $G_1(z)$. The outputs of $G_0(z)$ and $G_1(z)$ then are downsampled by a factor of 2. After some processing, the modified signals are upsampled by a factor of 2 and filtered by another filter bank consisting of $H_0(z)$ and $H_1(z)$.

If no processing takes place between the two filter banks, the sum of outputs of $H_0(z)$ and $H_1(z)$ is identical to the original signal $X(z)$, except for the time delay. This system is a two-channel PR filter bank, where $G_0(z)$ and $G_1(z)$ form an analysis filter bank, and $H_0(z)$ and $H_1(z)$ form a synthesis filter bank.

Traditionally, $G_0(z)$ and $H_0(z)$ are lowpass filters, and $G_1(z)$ and $H_1(z)$ are highpass filters. The subscripts `0` and `1` represent lowpass and highpass filters, respectively. The operation $\downarrow 2$ denotes a decimation of the signal by a factor of two. Applying decimation factors to the signal ensures that the number of output samples of the two lowpass filters equal the number of original input samples $X(z)$. Therefore, no redundant information is added during the decomposition. Refer to the LabVIEW Digital Filter Design Toolkit documentation for more information about filters.

You can use the two-channel PR filter bank system and consecutively decompose the outputs of lowpass filters, as shown in Figure 4-2.

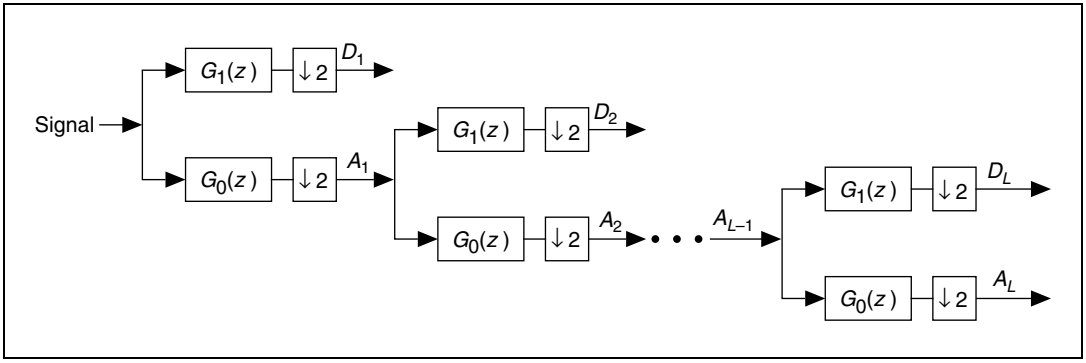

Figure 4-2. Discrete Wavelet Transform

Lowpass filters remove high-frequency fluctuations from the signal and preserve slow trends. The outputs of lowpass filters provide an approximation of the signal. Highpass filters remove the slow trends from the signal and preserve high-frequency fluctuations. The outputs of highpass filters provide detail information about the signal. The outputs of lowpass filters and highpass filters define the approximation coefficients and detail coefficients, respectively. Symbols ***A*** and ***D*** in Figure 4-2 represent the approximation and detail information, respectively.

You also can call the detail coefficients *wavelet coefficients* because detail coefficients approximate the inner products of the signal and wavelets. This manual alternately uses the terms wavelet coefficients and detail coefficients, depending on the context.

The Wavelet Analysis Tools use the subscripts `0` and `1` to describe the decomposition path, where `0` indicates lowpass filtering and `1` indicates highpass filtering. For example, D_2 in Figure 4-2 denotes the output of two cascaded filtering operations—lowpass filtering followed by highpass filtering. Therefore, you can describe this decomposition path with the sequence `01`. Similarly, D_L denotes the output of the filtering operations `000...1` in which the total number of `0` is $L–1$. The impulse response of `000...1` converges asymptotically to the mother wavelet and the impulse response of `000...0` converges to the scaling function in the wavelet transform.

The DWT is invertible, meaning that you can reconstruct the signal from the DWT coefficients with the inverse DWT. The inverse DWT also is implemented with filter banks by cascading the synthesis filter banks. Figure 4-3 shows the inverse DWT using filter banks.

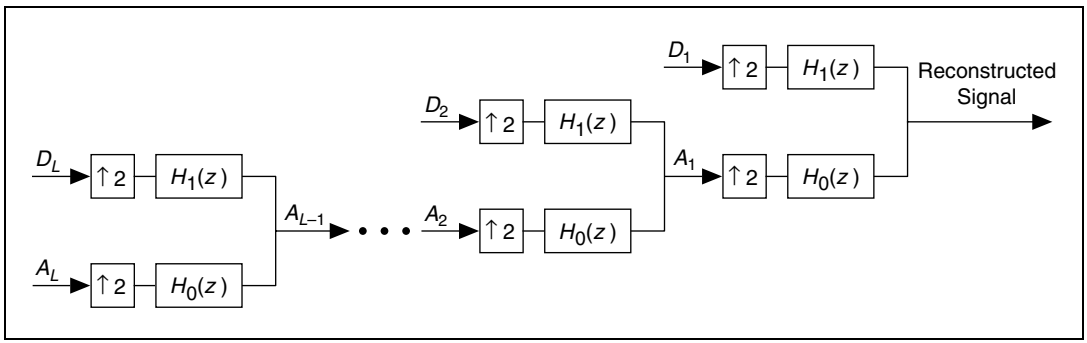

Figure 4-3. Inverse Discrete Wavelet Transform

Use the WA Discrete Wavelet Transform VI to compute the DWT of 1D and 2D signals. Use the WA Inverse Discrete Wavelet Transform VI to compute the inverse DWT of 1D and 2D signals. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about these VIs.

Discrete Wavelet Transform for Multiresolution Analysis

The DWT is well-suited for multiresolution analysis. The DWT decomposes high-frequency components of a signal with fine time resolution but coarse frequency resolution and decomposes low-frequency components with fine frequency resolution but coarse time resolution.

Figure 4-4 shows the frequency bands of the DWT for the db08 wavelet.

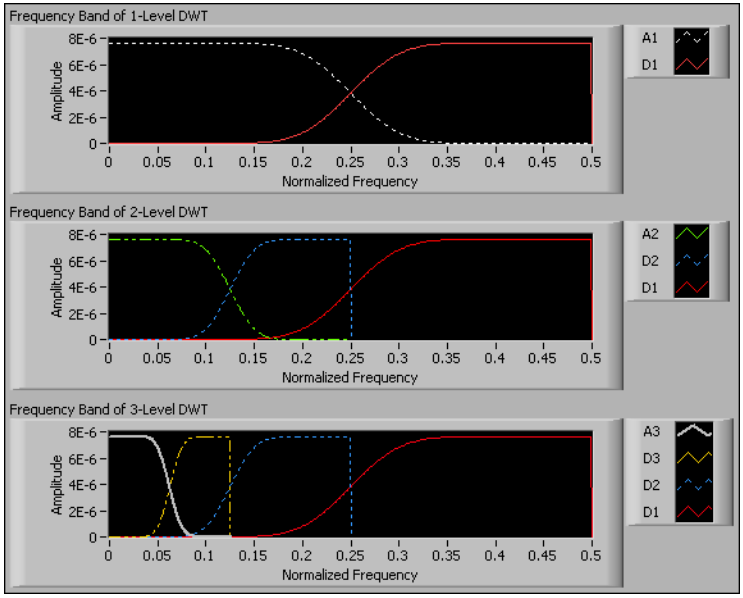

Figure 4-4. Frequency Bands of the Discrete Wavelet Transform

You can see that the central frequency and frequency bandwidth of the detail coefficients decrease by half when the decomposition level increases by one. For example, the central frequency and frequency bandwidth of D_2 are half that of D_1. You also can see that the approximation at a certain resolution contains all of the information about the signal at any coarser resolutions. For example, the frequency band of A_2 covers the frequency bands of A_3 and D_3.

DWT-based multiresolution analysis helps you better understand a signal and is useful in feature extraction applications, such as peak detection and edge detection. Multiresolution analysis also can help you remove unwanted components in the signal, such as noise and trend.

Figure 4-5 shows the multiresolution results for a signal using the DWT.

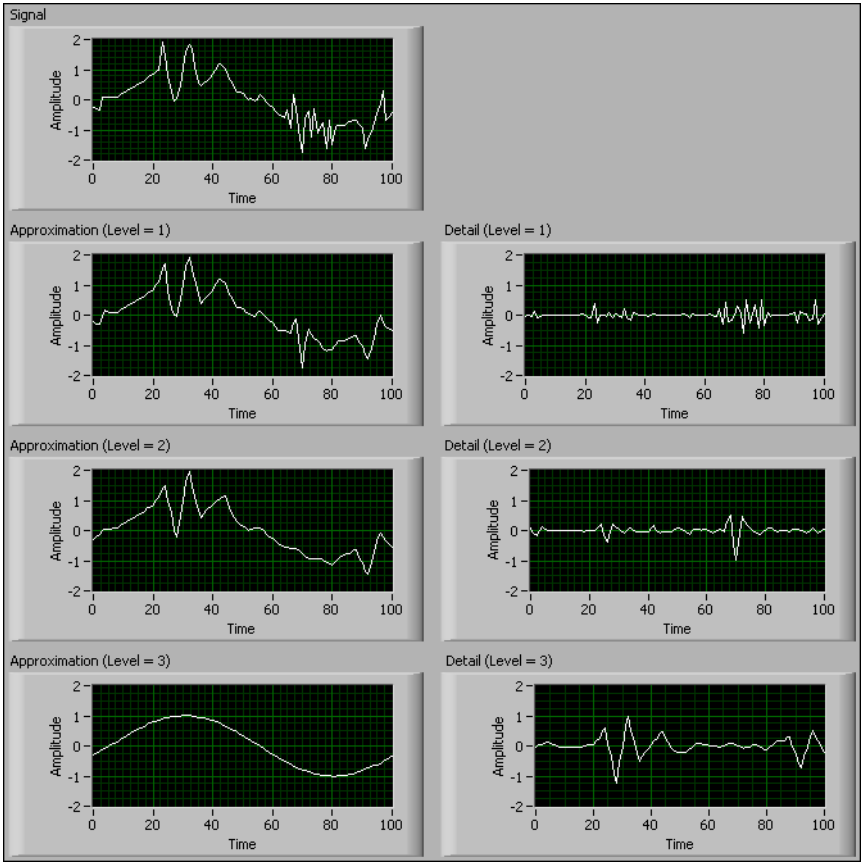

Figure 4-5. DWT-Based Multiresolution Analysis

You can see that the approximation at level 1 is the summation of the approximation and detail at level 2. The approximation at level 2 is the summation of the approximation and detail at level 3. As the level increases, you obtain lower frequency components, or large-scale approximation and detail, of the signal.

In the **Browse** tab of the NI Example Finder, you can view a multiresolution analysis example by selecting **Toolkits and Modules»Wavelet Analysis»Getting Started»Multiresolution Analysis VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

Use the Multiresolution Analysis Express VI to decompose and reconstruct a signal at different levels and with different wavelet types. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about this Express VI.

2D Signal Processing

The preceding sections introduce the DWT in 1D signal processing. Using the Wavelet Analysis Tools, you can extend the DWT to 2D signal processing.

Figure 4-6 shows the PR filter bank implementation of the 2D DWT, which applies the filter banks to both rows and columns of an image.

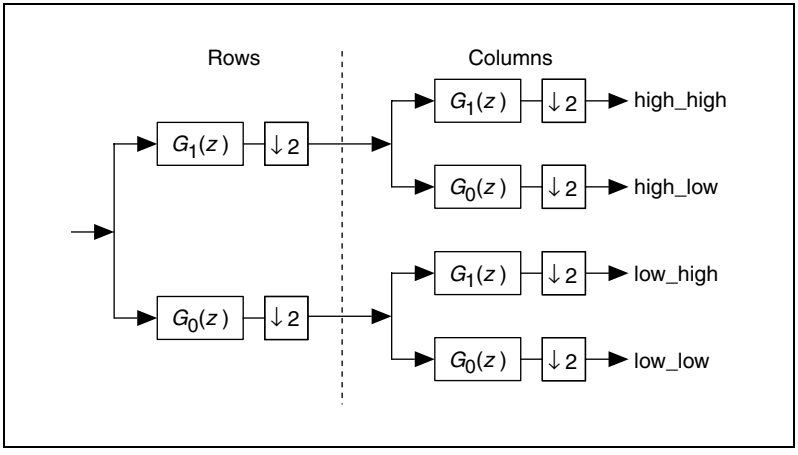

Figure 4-6. 2D Discrete Wavelet Transform

As Figure 4-6 shows, when decomposing 2D signals with two-channel PR filter banks, you process rows first and then columns. Consequently, one 2D array splits into the following four 2D arrays:

- low-low
- low-high
- high-low
- high-high

Each array is one-fourth of the size of the original 2D array.

Figure 4-7 shows an example of decomposing and reconstructing an image file with the 2D DWT and the inverse 2D DWT.

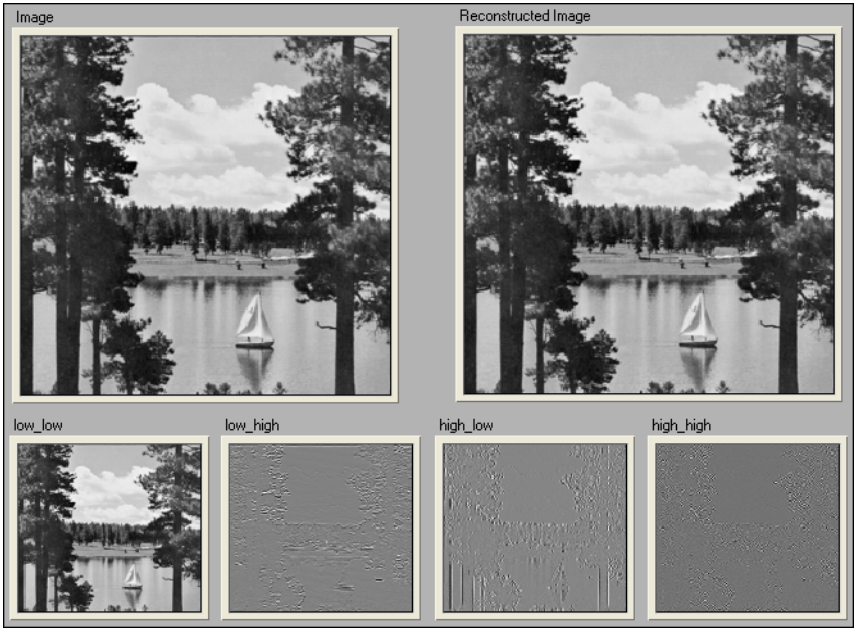

Figure 4-7. Example of 2D Discrete Wavelet Transform

The source image is decomposed into the following four sub-images:

- **low_low**—Shows an approximation of the source signal with coarse resolution.
- **low_high**—Shows the details at the discontinuities along the column direction.
- **high_low**—Shows the details at the discontinuities along the row direction.
- **high_high**—Shows the details at the discontinuities along the diagonal direction.

You can apply the decomposition iteratively to the **low_low** image to create a multi-level 2D DWT, which produces an approximation of the source signal with coarse resolution. You can determine the appropriate number of decomposition levels for a signal-processing application by evaluating the quality of the decomposition at different levels.

Use the Multiresolution Analysis 2D Express VI to decompose and reconstruct a 2D signal. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about this Express VI.

Figure 4-8 shows an example of image compression using the 2D DWT with the FBI wavelet.

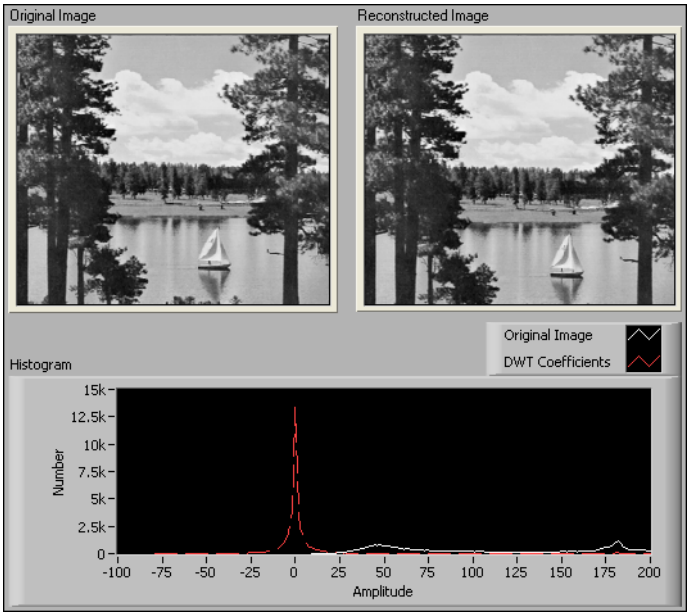

Figure 4-8. Example of Image Compression

The histogram of the **DWT Coefficients** plot shows that the majority of the DWT coefficients are small, meaning that you can use a small number of large DWT coefficients to approximate the image and achieve data compression.

Wavelet Packet Decomposition

As discussed in the *Discrete Wavelet Transform* section of this chapter, you can approximate the DWT using filter banks. When the decomposition is applied to both the approximation coefficients and the detail coefficients, the operation is called wavelet packet decomposition.

Figure 4-9 shows the wavelet packet decomposition tree.

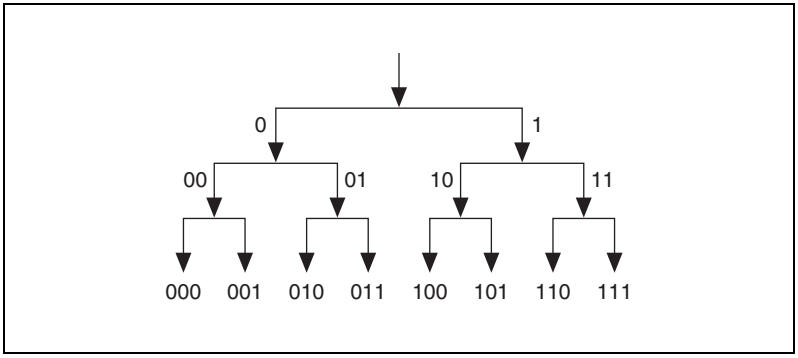

Figure 4-9. Wavelet Packet Decomposition Tree at Level Three

The numbers indicate the path of each node. The path is a combination of the characters `0` and `1`, where `0` represents lowpass filtering followed by a decimation with a factor of two, and `1` represents highpass filtering followed by a decimation with a factor of two.

Based on Figure 4-9, you can represent a signal with different sets of sequences, or different decomposition schemes, such as (`1, 01, 001, 000`), (`1, 00, 010, 011`), or (`000, 001, 010, 011, 100, 101, 110, 111`). As the decomposition level increases, the number of different decomposition schemes also increases.

The DWT is useful in compressing signals in some applications. The wavelet packet decomposition also can compress signals and provide more compression for a given level of distortion than the DWT does for some signals, such as signals composed of chirps.

For example, the wavelet packet decomposition and the DWT with the sym8 wavelet, decomposition level 4, and periodic extension are applied to the Piece Polynomial signal and the Chirps signal. Figure 4-10 shows the decomposition of the Piece Polynomial signal. The resulting histogram of the wavelet packet coefficients is similar to the histogram of the discrete wavelet coefficients, meaning that the DWT and the wavelet packet decomposition have similar compression performance for the Piece Polynomial signal.

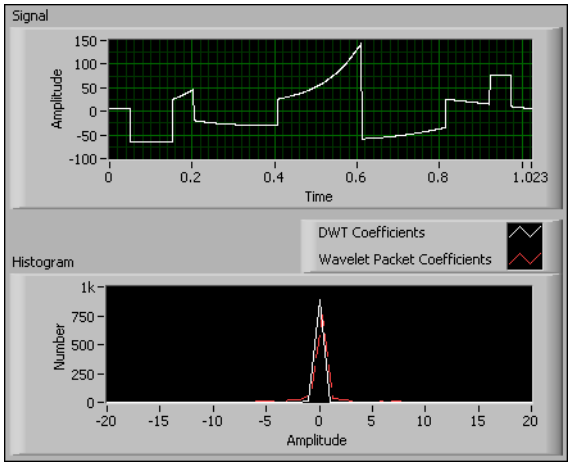

Figure 4-10. Decomposition of the Piece Polynomial Signal

Figure 4-11 shows the decomposition of the Chirps signal. The resulting histogram of the wavelet packet coefficients is more compact than the histogram of the DWT coefficients. Therefore, the wavelet packet decomposition can achieve a higher compression ratio for signals like the Chirps signal.

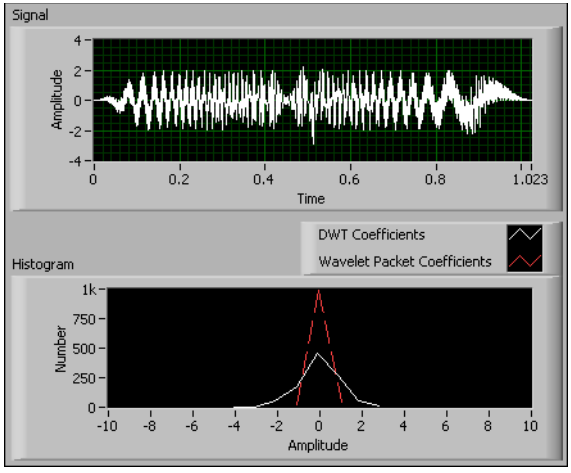

Figure 4-11. Decomposition of the Chirps Signal

Arbitrary Path Decomposition

Traditional wavelet packet decomposition iteratively applies the lowpass and highpass filters to both the approximation and the detail coefficients. The arbitrary path decomposition, as a special case of the wavelet packet decomposition, iteratively applies the lowpass and highpass filters to either the approximation or the detail coefficients at each level. You can consider arbitrary path decomposition as a band-pass filter, which you can implement by cascading filter banks. Figure 4-12 shows an example arbitrary path decomposition.

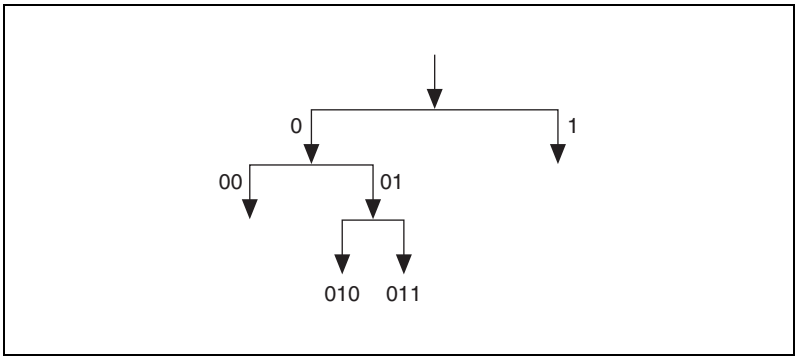

Figure 4-12. Arbitrary Path Decomposition

In this example, the decomposition path is `011`, because the signal first enters a lowpass filter `0`, then a highpass filter `1`, and finally a highpass filter `1` again. The results on the paths `1`, `00`, and `010` also can be saved for reconstruction purpose. The paths `1`, `00`, and `010` define residual paths.

Use the WA Arbitrary Path Decomposition VI and the WA Arbitrary Path Reconstruction VI to decompose and reconstruct a signal according to different paths and wavelet types. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about these two VIs.

Figure 4-13 shows an application of the arbitrary path decomposition in detecting engine knocking due to an ignition-system malfunction.

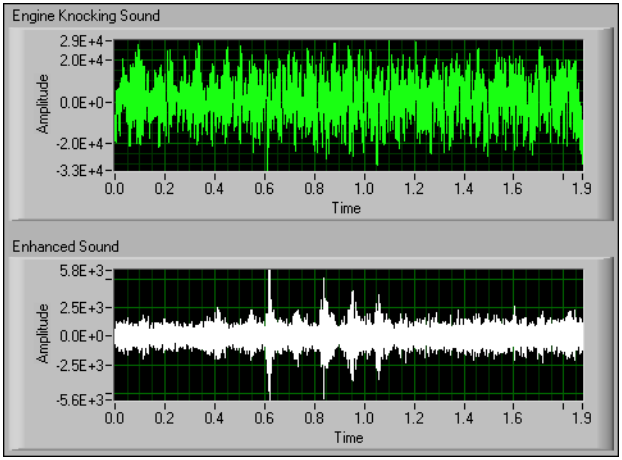

Figure 4-13. Engine Knocking Detection

This example applies the bior3_7 wavelet and the path `11` to the decomposition of the signal that the **Engine Knocking Sound** graph contains. The **Enhanced Sound** graph shows the signal reconstructed from the path `11`. The high-amplitude components around 0.6, 0.8, and 1.0 in the **Enhanced Sound** graph indicate where the ignition malfunction of the engine occurs.

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Wavelet Analysis»Applications» Engine Knocking Detection VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

Undecimated Wavelet Transform

Unlike the DWT, which downsamples the approximation coefficients and detail coefficients at each decomposition level, the UWT does not incorporate the downsampling operations. Thus, the approximation coefficients and detail coefficients at each level are the same length as the original signal. The UWT upsamples the coefficients of the lowpass and highpass filters at each level. The upsampling operation is equivalent to dilating wavelets. The resolution of the UWT coefficients decreases with increasing levels of decomposition.

Use the WA Undecimated Wavelet Transform VI and the WA Inverse Undecimated Wavelet Transform VI to decompose and reconstruct 1D or 2D signals. Refer to the *LabVIEW Help*, available by selecting **Help» Search the LabVIEW Help**, for information about these two VIs.

Benefits of Undecimated Wavelet Transform

This section describes the unique features of the UWT by comparing the UWT with the DWT.

Translation-Invariant Property

Unlike the DWT, the UWT has the translation-invariant, or shift-invariant, property. If two signals are shifted versions of each other, the UWT results for the two signals also are shifted versions of each other. The translation-invariant property is important in feature-extraction applications.

Figure 4-14 shows an example that detects discontinuities in the HeaviSine signal with both the DWT and the UWT.

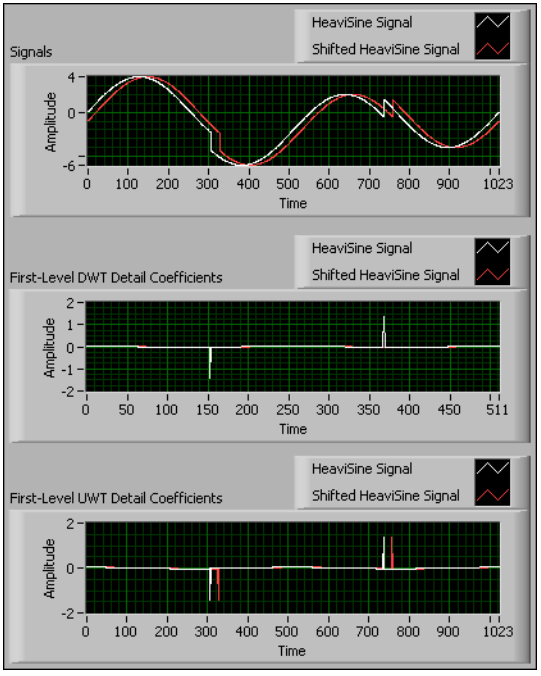

Figure 4-14. Discrete Wavelet Transform versus Undecimated Wavelet Transform

You can use the first-level detail coefficients of either the DWT or the UWT to detect the discontinuities in the HeaviSine signal by locating the peaks in the coefficients. However, if the HeaviSine signal is shifted by 21 samples, all of the first-level DWT detail coefficients become very small. Therefore, you cannot use the first-level DWT detail coefficients to detect the discontinuities in the shifted HeaviSine signal. Because of the translation-invariant property of the UWT, you can use the first-level UWT detail coefficients to detect the discontinuities of the shifted HeaviSine Signal. The first-level UWT detail coefficients of the shifted HeaviSine Signal are simply the shifted version of the first-level UWT detail coefficients of the original HeaviSine signal.

Better Denoising Capability

Denoising with the UWT also is shift-invariant. The denoising result of the UWT has a better balance between smoothness and accuracy than the DWT. The DWT-based method is more computationally efficient than the UWT-based method. However, you cannot achieve both smoothness and accuracy with the DWT-based denoising method.

Use the Wavelet Denoise Express VI or the WA Denoise VI to reduce noise in 1D signals with both the UWT-based and DWT-based methods. The UWT-based method supports both real and complex signals. The DWT-based method supports only real signals. You also can use the WA Denoise VI to reduce noise in 2D signals with the UWT-based method. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about these VIs.

The denoising procedure in the Wavelet Denoise Express VI and the WA Denoise VI involves the following steps:

1. Applies the DWT or the UWT to noise-contaminated signals to obtain the DWT coefficients or the UWT coefficients. The noise in signals usually corresponds to the coefficients with small values.
2. Selects an appropriate threshold for the DWT coefficients or the UWT coefficients to set the coefficients with small values to zero. The Wavelet Denoise Express VI and the WA Denoise VI provide methods that automatically select the thresholds. The bound of noise reduction with these methods is 3 dB. To achieve better denoising performance for a signal, you can select an appropriate threshold manually by specifying the **user defined thresholds** parameter of the WA Denoise VI.
3. Reconstructs the signal with the inverse DWT or the inverse UWT.

Figure 4-15 shows the denoising results of a noisy Doppler signal with both the DWT-based method and the UWT-based method. Both methods use the level-5 wavelet transform and the soft threshold.

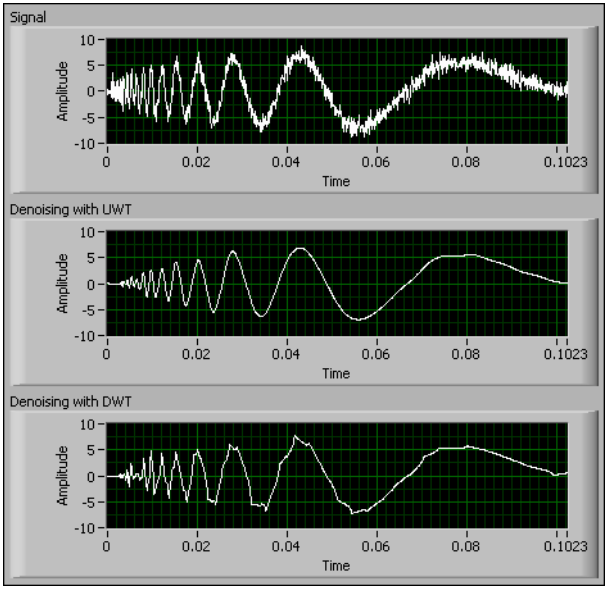

Figure 4-15. Denoising Comparison

In Figure 4-15, you can see that the UWT outperforms the DWT in signal denoising because the denoised signal in the **Denoising with UWT** graph is smoother.

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Wavelet Analysis»Getting Started» Denoise - 1D Real Signal VI**.

You also can view the follow examples that use the UWT-based method to denoise complex signals and 2D signals:

- **Toolkits and Modules»Wavelet Analysis»Getting Started» Denoise - 1D Complex Signal VI**
- **Toolkits and Modules»Wavelet Analysis»Getting Started» Denoise - Image VI**

Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

Better Peak Detection Capability

Peaks often imply important information about a signal. You can use the UWT to identify the peaks in a noise-contaminated signal.

The UWT-based peak detection method is more robust and less sensitive to noise than the DWT-based method, because the UWT-based method involves finding zero-crossings in the multiscale UWT coefficients. The UWT-based method first finds zero-crossings among the coefficients with coarse resolution and then finds zero-crossings among the coefficients with finer resolution. Finding zero-crossings among the coefficients with coarse resolution enables you to remove noise from a signal efficiently. Finding zero-crossings among the coefficients with finer resolution improves the precision with which you can find peak locations.

The WA Multiscale Peak Detection VI uses the UWT-based method. This VI detects peaks in offline and online signals. You can use this VI in the following ways:

- Once for an offline signal
- Continuously for a block of signals
- Continuously for signals from streaming data sources

Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about this VI.

Figure 4-16 shows an example that uses the WA Multiscale Peak Detection VI to detect peak in an electrocardiogram (ECG) signal. The UWT-based method locates the peaks of the ECG signal accurately, regardless of whether the peaks are sharp or rounded.

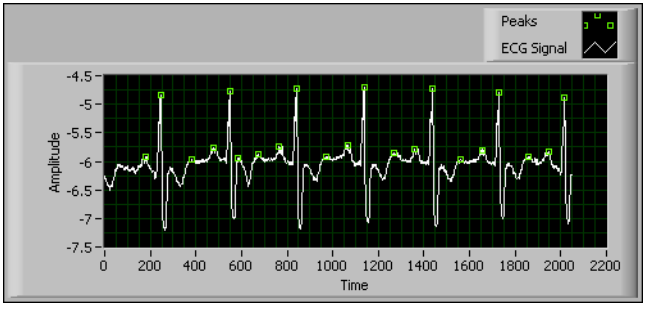

Figure 4-16. Peak Detection in a Noisy ECG Signal

In the **Browse** tab of the NI Example Finder, you can view the following examples by selecting:

- **Toolkits and Modules»Wavelet Analysis»Applications» ECG Heart Rate Monitor (Online) VI**
- **Toolkits and Modules»Wavelet Analysis»Applications» ECG QRS Complex Detection VI**
- **Toolkits and Modules»Wavelet Analysis»Getting Started» Peak Detection (Wavelet vs. Normal) VI**

Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

5

Interactively Designing Discrete Wavelets

Both the discrete wavelet transform and the inverse discrete wavelet transform are implemented using a set of cascaded two-channel perfect reconstruction (PR) filter banks. Refer to Chapter 4, *Signal Processing with Discrete Wavelets*, for more information about discrete wavelets and filter banks.

The WA Wavelet Filter VI already contains a collection of predefined wavelets, including orthogonal wavelets (Haar, Daubechies, Coiflets, Symmlets) and biorthogonal wavelets (FBI, Biorthogonal). You can apply the predefined wavelets directly to signal processing applications. If you cannot find a wavelet that best matches the signal, you can use the Wavelet Design Express VI to design a customized discrete wavelet.

The design of discrete wavelets is essentially the design of two-channel PR filter banks. This chapter describes the steps that you can follow when using the Wavelet Design Express VI to design discrete wavelets and provides an example of designing the FBI wavelet.

Figure 5-1 shows the **Configure Wavelet Design** dialog box of the Wavelet Design Express VI.

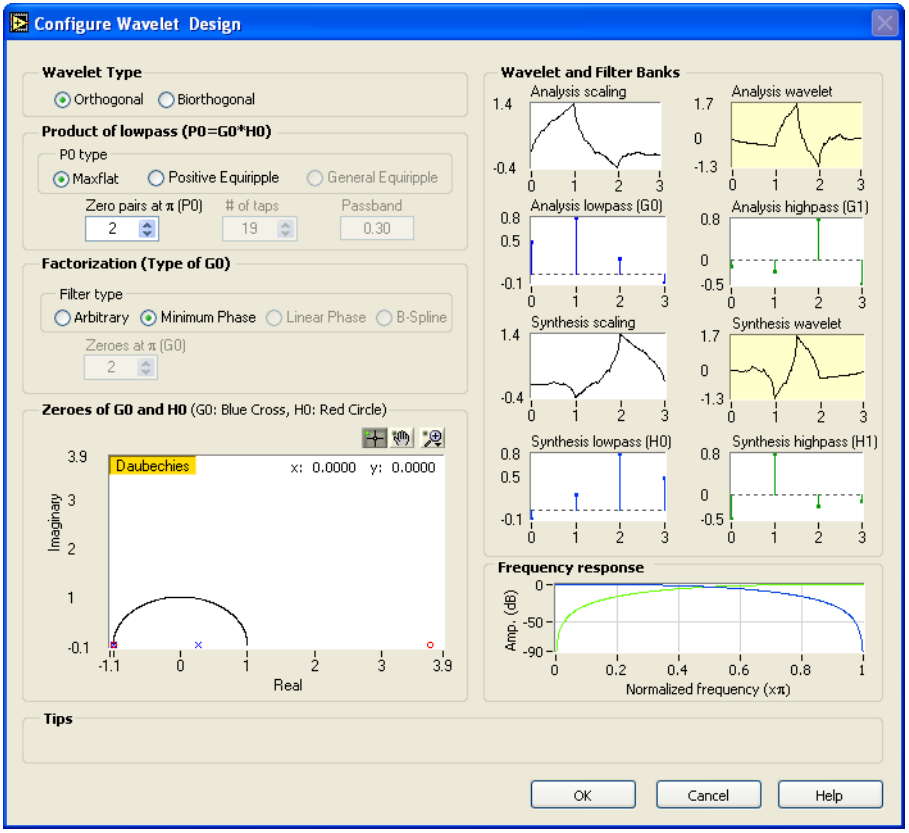

Figure 5-1. Configure Wavelet Design Dialog Box

On the left-hand side of the configuration dialog box, you can specify attributes of the wavelet that you want to design. On the right-hand side of the configuration dialog box, you can see the real-time plots of the designed wavelet. Refer to the LabVIEW Help, available by selecting **Help»Search the LabVIEW Help**, for more information about the Wavelet Design Express VI.

Figure 5-2 shows the wavelet design process.

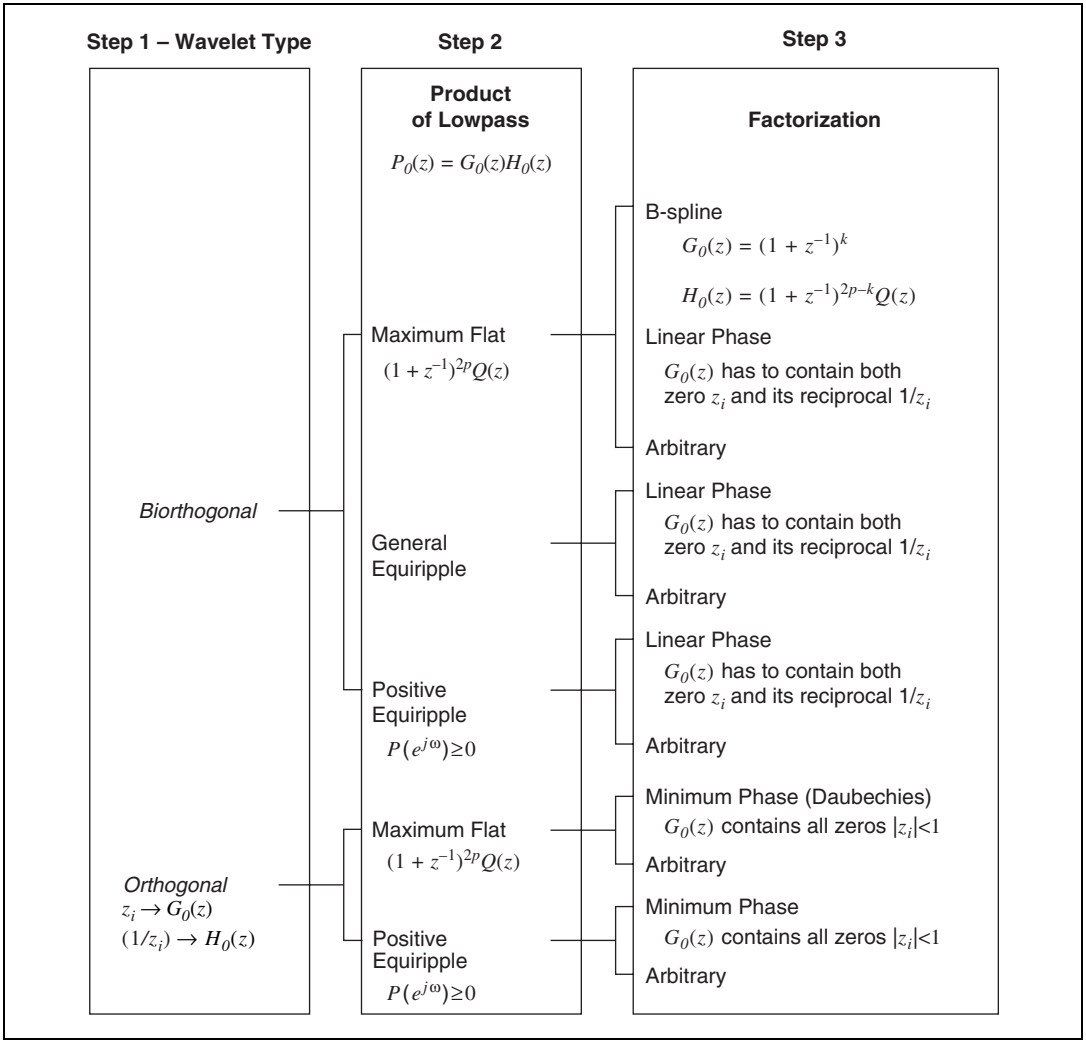

Figure 5-2. Design Procedure for Wavelets and Filter Banks

Using the Wavelet Design Express VI, you need to complete the following steps to design wavelets:

1. Select the wavelet type.
2. Design the product of lowpass filters, $P_0(z)$, where the auxiliary function $P_0(z)$ is the product of $G_0(z)$ and $H_0(z)$.
3. Select the factorization type to factorize $P_0(z)$ into $G_0(z)$ and $H_0(z)$.

After you create an analysis lowpass filter $G_0(z)$ and a synthesis lowpass filter $H_0(z)$, the Wavelet Design Express VI automatically generates the corresponding analysis highpass filter $G_1(z)$ and synthesis highpass filter $H_1(z)$.

The following sections describe each of the steps in the wavelet design process and the controls you use to complete the steps. You also can select **Help»Show Context Help** or press the <Ctrl-H> keys for more information about controls and indicators on the **Configure Wavelet Design** dialog box.

Selecting the Wavelet Type

Use the **Wavelet Type** control on the **Configure Wavelet Design** dialog box to select the wavelet type. You can choose from the following two wavelet types, **Orthogonal** (default) and **Biorthogonal**.

The wavelet transform with orthogonal wavelets is energy conserving, meaning that the total energy contained in the resulting coefficients and the energy in the original time samples are the same. This property is helpful for signal and image compression and denoising. But the filters associated with orthogonal wavelets are not linear phase. Linear phase is a helpful property for feature-extraction applications. The filters associated with biorthogonal wavelets can be linear phase.

Designing the Product $P_0(z)$

The auxiliary function $P_0(z)$ denotes the product of $G_0(z)$ and $H_0(z)$, as shown in the following equation:

$$P_0(z) = G_0(z)H_0(z)$$

You usually use one of the following three types of filters for $P_0(z)$:

- Maximally-flat
- General equiripple halfband
- Positive equiripple halfband

The maximally-flat filter is defined by the following equation:

$$P_0(z) = (1 + z^{-1})^{2p} Q(z)$$

The **Zero pairs at π (P0)** control on the **Configure Wavelet Design** dialog box specifies the value of the parameter *p*, which determines the number of zeroes placed at π on the unit circle. The more the zeroes at π, the smother the corresponding wavelet. The value of *p* also affects the transition band of the frequency response. A large value of *p* results in a narrow transition band. In the time domain, a narrower transition band implies more oscillations in the corresponding wavelet. When you specify a value for the parameter *p*, you can review the frequency response shown on the right-hand side of the **Configure Wavelet Design** dialog box.

In a general equiripple halfband filter, halfband refers to a filter in which $\omega_s + \omega_p = \pi$, where ω_s denotes the stopband frequency and ω_p denotes the passband frequency, as shown in Figure 5-3.

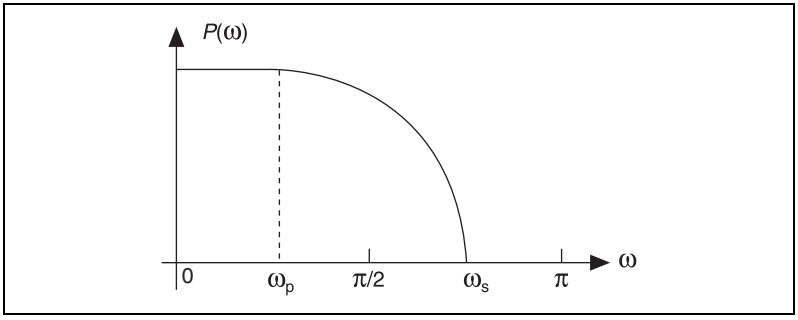

Figure 5-3. Halfband Filter

The positive equiripple halfband filter is a special case of general equiripple halfband filters. The Fourier transform of this type of filter is always nonnegative. Positive equiripple halfband filter is appropriate for orthogonal wavelets because the auxiliary function $P_0(z)$ must be nonnegative.

Use the **P_0 type** control to specify the $P_0(z)$ type. When **Wavelet Type** is set to **Orthogonal**, you can set **P_0(z)** either to **Maxflat** (default), for a maximally-flat filter, or to **Positive Equiripple**. When **Wavelet Type** is set to **Biorthogonal**, you can set **P_0(z)** to **Maxflat** (default), **Positive Equiripple**, or **General Equiripple**.

Because all filters, including $P_0(z)$, $G_0(z)$, and $H_0(z)$, are real-valued finite impulse response (FIR) filters, the zeroes of these filters are mirror-symmetric about the x-axis in the z-plane. Therefore, for any zero z_i, a corresponding complex conjugate z_i^* always exists. If z_i is complex, meaning that if z_i is located off of the x-axis, you always can find a corresponding zero on the other side of the x-axis, as shown in Figure 5-4.

As a result, you only need to see the top half of the z-plane to see all of the zeroes that are present. After you select z_i, the Wavelet Design Express VI automatically includes the complex conjugate z_i*.

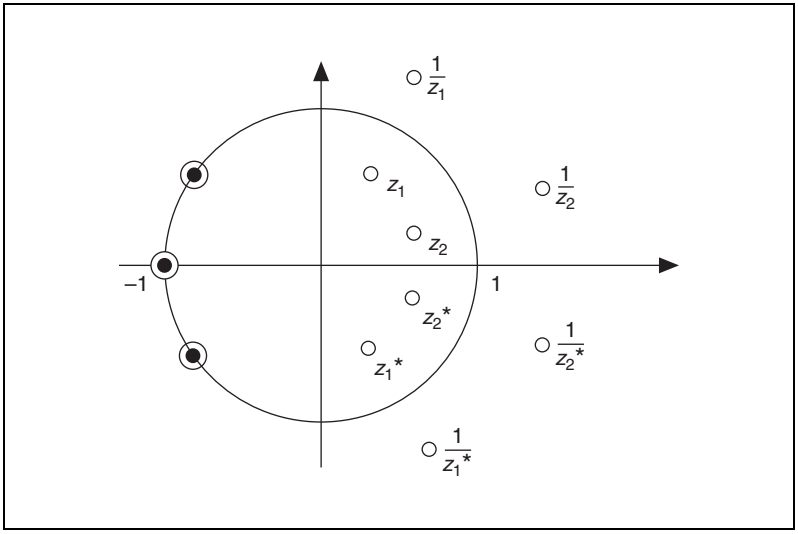

Figure 5-4. Zero Distribution of Real-Valued FIR Filters

Two parameters are associated with equiripple filters—**# of taps** and **Passband**. Use the **# of taps** control to define the number of coefficients of $P_0(z)$. Because $P_0(z)$ is a type-I FIR filter, the length of $P_0(z)$ must be odd. Use the **Passband** control to define the normalized passband frequency, ω_p, of $P_0(z)$. The value of ω_p must be less than 0.5. Longer filters improve the sharpness of the transition band and the magnitude of the attenuation in the stopband at the expense of extra computation time for implementation.

Selecting the Factorization Type for $P_0(z)$

After you determine $P_0(z)$, the next step is to specify how $P_0(z)$ is factorized into the analysis lowpass filter $G_0(z)$ and the synthesis lowpass filter $H_0(z)$, respectively. Use the **Factorization (Type of G0)** control to specify the factorization type. The factorizing process is not unique. For a given $P_0(z)$, you have the following four options for creating $G_0(z)$ and $H_0(z)$:

- **Arbitrary**—No specific constraints are associated with this filter. Figure 5-5 shows an example of arbitrary factorization. The blue crosses represent the zeroes of $G_0(z)$, and the red circles represent the

zeroes of $H_0(z)$. Click on the zero you want to select to switch the zero from that of $G_0(z)$ to that of $H_0(z)$ and vice versa.

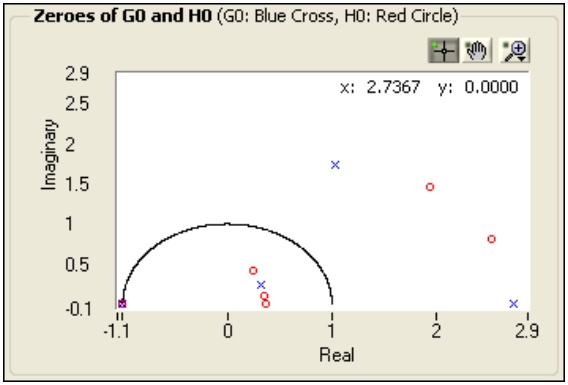

Figure 5-5. Arbitrary Filter

- **Minimum Phase**—All of the zeroes of $G_0(z)$ are contained inside the unit circle, as shown in Figure 5-6. All the zeroes of $H_0(z)$ are the reciprocal of the zeroes of $G_0(z)$. The Wavelet Design Express VI automatically generates the zeroes for $H_0(z)$ and $G_0(z)$. You cannot switch the zeroes between $G_0(z)$ and $H_0(z)$. The minimum phase filter possesses minimum phase-lag. When $P_0(z)$ is maximally-flat and $G_0(z)$ is minimum phase, the resulting wavelets are the Daubechies wavelets.

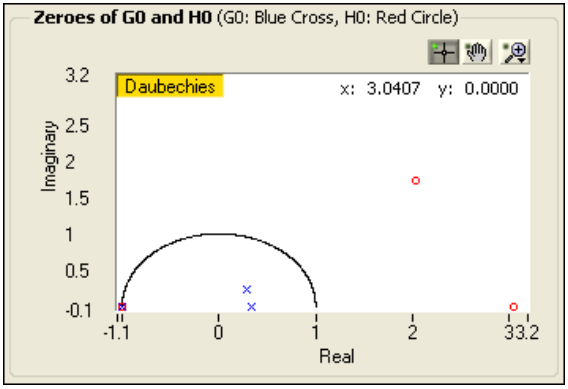

Figure 5-6. Minimum Phase Filter

- **Linear Phase**—Any zero and its reciprocal must belong to the same filter, as shown in Figure 5-7. When you switch a zero of $G_0(z)$ to that of $H_0(z)$, the reciprocal of the zero also switches to $H_0(z)$. When you switch a zero of $H_0(z)$ to that of $G_0(z)$, the reciprocal of the zero also switches to $G_0(z)$. This option is available only if the filter is biorthogonal.

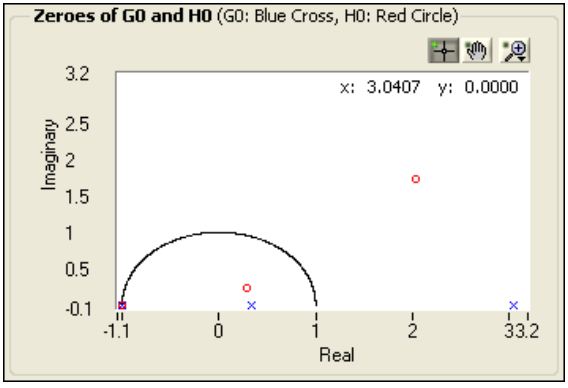

Figure 5-7. Linear Phase Filter

In the time domain, a linear phase implies that the coefficients of the filter are symmetric or antisymmetric. Linear phase filters have a constant group delay for all frequencies. This property is required in many signal and image feature-extraction applications, such as peak detection and image edge detection.

- **B-Spline**—This option is available only if **Wavelet Type** is **Biorthogonal** and **P0 type** is **Maxflat**. In this case, the analysis lowpass filter $G_0(z)$ and the synthesis lowpass filter $H_0(z)$ are defined by the following equations, respectively:

$$G_0(z) = (1 + z^{-1})^k \qquad H_0(z) = (1 + z^{-1})^{2p-k} Q(z)$$

where k is specified with the **Zeroes at π (G0)** control, and p is determined by the **Zero pairs at π (P0)** control. The Wavelet Design Express VI automatically generates the zeroes of $G_0(z)$ and $H_0(z)$ based on the settings for k and p. You cannot switch the zeroes between $G_0(z)$ and $H_0(z)$. Figure 5-8 shows an example of B-Spline factorization.

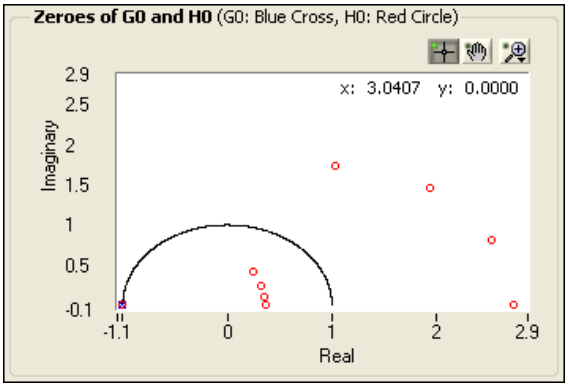

Figure 5-8. B-Spline Filter

Refer to *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about the Wavelet Design Express VI.

Example: Designing the FBI Wavelet

Different signal processing applications require different properties of wavelets. For image compression, you need a wavelet that is smooth, linear phase, and orthogonal. However, as discussed in the *Selecting the Wavelet Type* section of this chapter, you cannot achieve all those properties simultaneously. One thing you can do is to ensure smoothness (higher order) and linear phase first and then pursue near-orthogonality.

Using the Wavelet Design Express VI, you can design a wavelet with specific properties. For example, you can complete the following steps to design the FBI wavelet, which is linear phase and near-orthogonal.

1. Place the **Wavelet Design** Express VI on the block diagram. The **Configure Wavelet Design** dialog box, as shown in Figure 5-1, automatically launches.
2. Select **Biorthogonal** as the **Wavelet Type** because only biorthogonal wavelets have the linear-phase property.
3. In the **Product of lowpass (G0*H0)** section, select **Maxflat** as the **P0 type** and set the value of **Zero pairs at π (P0)** to `4`.

   When you set parameters on the left-hand side of the configuration dialog box, plots of the designed wavelet and the associated filter banks interactively appear on the right-hand side.

4. In the **Factorization (Type of G0)** section, select **Linear Phase** as the **Filter type** and set the value of **Zeroes at π (G0)** to 4 because the wavelet must be near-orthogonal, meaning that $G_0(z)$ and $H_0(z)$ have the same or almost the same amount of zeroes. By setting the value of **Zeroes at π (G0)** to 4, you can ensure that both $G_0(z)$ and $H_0(z)$ have the same amount of zeroes at π.

   Figure 5-9 shows the zeroes of $G_0(z)$ and $H_0(z)$.

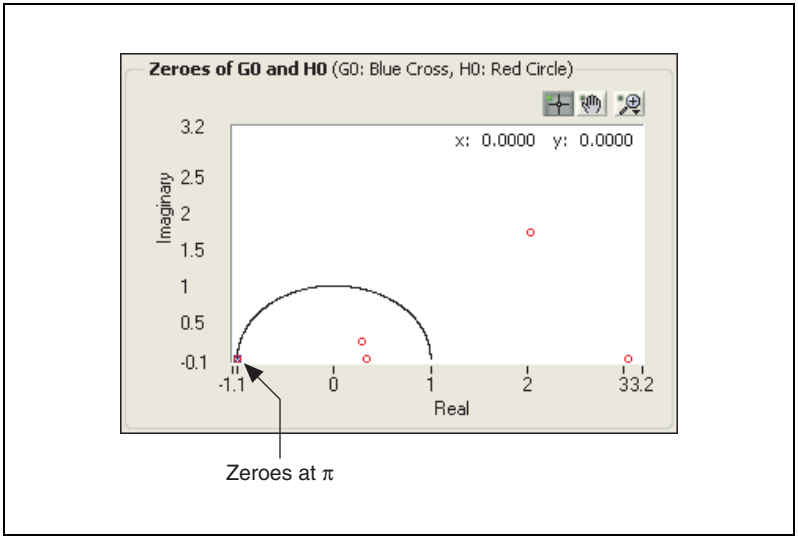

Figure 5-9. Zeroes of G0 and H0

   In Figure 5-9, notice that besides the four zeroes assigned to $H_0(z)$ at π, the **Zeroes of G0 and H0** graph also contains six more zeroes that belong to $H_0(z)$. Note that two zeroes are on the negative half plane and do not appear on this graph.

5. To make the number of zeroes of $G_0(z)$ close to that of $H_0(z)$, click either of the two zeroes (o) of $H_0(z)$ near the bottom of the **Zeroes of G0 and H0** graph and switch the two zeroes to those of $G_0(z)$. $G_0(z)$ now has six zeroes and $H_0(z)$ has eight zeroes.

   Figure 5-10 shows the design result. Notice that the analysis and synthesis scaling functions are similar, and the analysis and synthesis wavelets also are similar, which means the designed wavelet is near-orthogonal. The symmetry of the filter banks also preserves the linear-phase property.

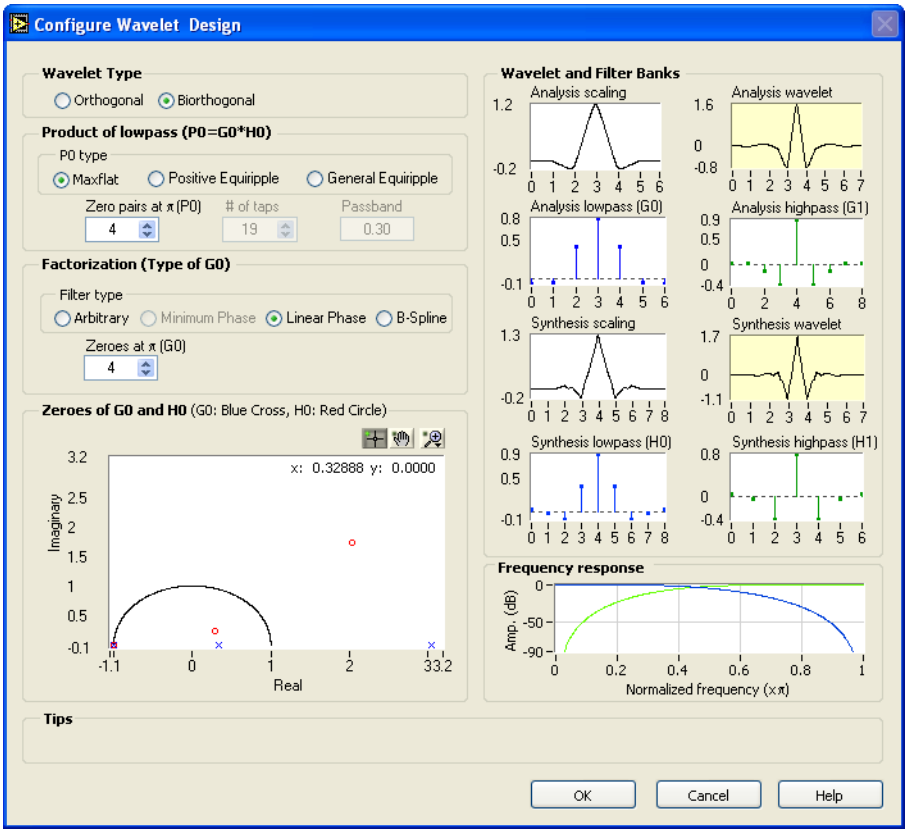

Figure 5-10. The Designed FBI Wavelet

Because of the near-orthogonality and linear-phase properties of the FBI wavelet, you can apply this wavelet to many kinds of signal and image processing, for example, image compression in JPEG2000. The FBI wavelet is called **bior4_4** because both the analysis and synthesis lowpass filters $G_0(z)$ and $H_0(z)$ have four zeroes at π. This wavelet also is known as **CDF 9, 7** because the lengths of the analysis and synthesis highpass filters are nine and seven, respectively.

6

Integer Wavelet Transform

Many signal samples you encounter in real-world applications are encoded as integers, such as the signal amplitudes encoded by analog-to-digital (A/D) converters and color intensities of pixels encoded in digital images. For integer-encoded signals, an integer wavelet transform (IWT) can be particularly efficient. The IWT is an invertible integer-to-integer wavelet analysis algorithm. You can use the IWT in the applications that you want to produce integer coefficients for integer-encoded signals. Compared with the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT), the IWT is not only computationally faster and more memory-efficient but also more suitable in lossless data-compression applications. The IWT enables you to reconstruct an integer signal perfectly from the computed integer coefficients.

Use the WA Integer Wavelet Transform VI, which implements the IWT with the lifting scheme, to decompose an integer signal or image. Use the WA Inverse Integer Wavelet Transform VI, which implements the inverse IWT with the inverse lifting scheme, to reconstruct an integer signal or image from the IWT coefficients. Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help**, for information about these VIs.

This chapter describes an application example that uses the IWT to compress an image file.

Application Example: Lossless Compression

When you apply the DWT to integer signal samples, you convert the original integer signal samples to floating-point wavelet coefficients. In signal compression applications, you typically further quantize these coefficients to an integer representation before entropy-based encoding. As a result, compression with the DWT is lossy, meaning that some information is lost when you compress a signal using the DWT, and that you typically cannot reconstruct the original signal perfectly from the coefficients of the DWT.

The IWT, however, provides lossless compression. You can use the IWT to convert integer signal samples into integer wavelet coefficients, and you can compress these integer coefficients by entropy-based encoding without further quantization. As a result, you can reconstruct the original signal perfectly from a compressed set of IWT coefficients. Figures 6-1 shows an example of lossless compression with the IWT.

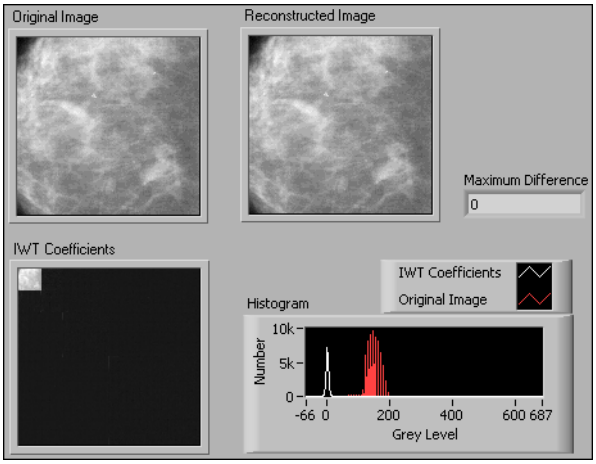

Figure 6-1. Lossless Image Compression

In the **Histogram** graph, most of the elements in the **IWT Coefficients** plot are zero, meaning that you can obtain a high compression ratio using the IWT of this image. You can reconstruct the image perfectly with the inverse IWT, as shown in the **Reconstructed Image** graph. The **Maximum Difference** value of `0` indicates that the reconstructed image retains all the information of the original image.

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Wavelet Analysis» Applications»Lossless Medical Image Compression VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Wavelet Signal Processing*, for information about launching the NI Example Finder.

A

Technical Support and Professional Services

Visit the following sections of the award-winning National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Technical support resources at ni.com/support include the following:
  - **Self-Help Technical Resources**—For answers and solutions, visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support as well as exclusive access to on demand training modules via the Services Resource Center. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

    For information about other technical support options in your area, visit ni.com/services, or contact your local office at ni.com/contact.
- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.