

LabVIEW™

**Joint Time-Frequency
Analysis Toolkit
Reference Manual**

Internet Support

E-mail: support@natinst.com

FTP Site: <ftp.natinst.com>

Web Address: <http://www.natinst.com>

Bulletin Board Support

BBS United States: 512 794 5422

BBS United Kingdom: 01635 551422

BBS France: 01 48 65 15 59

Fax-on-Demand Support

512 418 1111

Telephone Support (USA)

Tel: 512 795 8248

Fax: 512 794 5678

International Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 288 3336,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 09 725 725 11,
France 01 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, Israel 03 6120092, Italy 02 413091,
Japan 03 5472 2970, Korea 02 596 7456, Mexico 5 520 2635, Netherlands 0348 433466, Norway 32 84 84 00,
Singapore 2265886, Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51, Taiwan 02 377 1200,
United Kingdom 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway Austin, Texas 78730-5039 USA Tel: 512 794 0100

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 manual 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 THEREFORE 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

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

LabVIEW™, natinst.com™, National Instruments™, and NI-DAQ™ are trademarks of National Instruments Corporation. Product and company names listed are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

About This Manual

Organization of This Manual	ix
Conventions Used in This Manual	x
Related Documentation	x
Customer Communication	xi

Chapter 1

Overview of the Joint Time-Frequency Analysis Toolkit

Package Contents	1-1
Installation	1-1
Windows 95/NT	1-1
Windows 3.x	1-2
Power Macintosh	1-2
Getting Help	1-2

Chapter 2

The Need for Joint Time-Frequency Analysis

Review of the Classical Fourier Transform	2-1
The Need for JTFA	2-4
Basic Approaches to JTFA	2-6

Chapter 3

Joint Time-Frequency Analysis Algorithms

Linear JTFA Algorithms	3-1
Gabor Expansion and STFT	3-1
Adaptive Representation and Adaptive Transform	3-2
Quadratic JTFA Algorithms	3-3
STFT Spectrogram	3-3
Wigner-Ville Distribution and Pseudo Wigner-Ville Distribution	3-5
Cohen's Class	3-8
Choi-Williams Distribution	3-9
Cone-Shaped Distribution	3-10
Gabor Spectrogram	3-11
Adaptive Spectrogram	3-13

Chapter 4 Joint Time-Frequency Analysis VIs

Adaptive Transform.....	4-1
Inverse Adaptive Transform.....	4-2
Short-Time Fourier Transform.....	4-3
Gabor Expansion.....	4-4
2D STFT.....	4-4
2D Gabor Expansion.....	4-6
Fast Dual.....	4-7
Normalized Gaussian Window Function.....	4-8
Adaptive Spectrogram.....	4-9
Cohen’s Class.....	4-10
CWD (Choi–Williams Distribution).....	4-10
Cone-Shaped Distribution.....	4-11
STFT Spectrogram.....	4-12
PWVD (Pseudo Wigner–Ville Distribution).....	4-13
Gabor Spectrogram (Gabor Expansion-Based Spectrogram).....	4-14
Time-Frequency Distribution Series.....	4-15

Chapter 5 Joint Time-Frequency Analysis Applications

Linear Algorithm Examples.....	5-1
Denoise.....	5-1
Image Analysis.....	5-3
Time-Dependent Spectrum Analysis Examples.....	5-5
Online STFT Spectrogram Analyzer.....	5-6
Setting NI DAQ.....	5-6
Setting the Analysis Window.....	5-7
Acquiring Data.....	5-7
Saving Data.....	5-7
Offline Joint Time-Frequency Analyzer.....	5-8
Changing Spectrogram Display.....	5-9
Inputting Data.....	5-9
Saving Results.....	5-9
Switching Between Conventional Power and Instantaneous Spectrum.....	5-9
Frequency Zooming.....	5-10
Applying the Pre-Emphasis Filter.....	5-10
Setting Time Parameters.....	5-11

Selecting the JTFA Method	5-11
STFT Spectrogram	5-11
Gabor Spectrogram.....	5-12
Adaptive Spectrogram	5-12
Pseudo Wigner–Ville Distribution	5-13
Choi–Williams Distribution	5-13
Cone-Shaped Distribution	5-14

Chapter 6

Frequently Asked Questions

Appendix A

References

Appendix B

Error Codes

Appendix C

Customer Communication

Glossary

Index

Figures

Figure 2-1.	Basis Functions Used for Fourier Transform	2-2
Figure 2-2.	Seismic Signal	2-2
Figure 2-3.	ECG Signal.....	2-3
Figure 2-4.	Speech Signal	2-3
Figure 2-5.	Ionized Impulse Signal.....	2-5
Figure 2-6.	Reconstructed Signal.....	2-5
Figure 3-1.	STFT Spectrogram with a Narrowband Hanning Window for the Three-Tone Test Signal.....	3-4
Figure 3-2.	STFT-based Spectrogram with a Wideband Hanning Window for the Three-Tone Test Signal.....	3-4
Figure 3-3.	Wigner–Ville Distribution for the Three-Tone Test Signal	3-6
Figure 3-4.	Pseudo Wigner–Ville Distribution (with Gaussian Window $w[m]$) for the Three-Tone Test Signal.....	3-7

Figure 3-5.	Pseudo Wigner–Ville Distribution for the Three-Tone Test Signal	3-8
Figure 3-6.	CWD ($\alpha = 1$) for the Three-Tone Test Signal	3-9
Figure 3-7.	Cone-shaped Distribution ($\alpha = 1$) for the Three-Tone Test Signal	3-10
Figure 3-8.	Gabor Spectrogram (Order Four) for the Three-Tone Test Signal	3-12
Figure 3-9.	Adaptive Spectrogram for the Three Gaussian Tones	3-13
Figure 5-1.	Gabor Expansion Based Denoise	5-2
Figure 5-2.	2D STFT for Image Analysis	5-3
Figure 5-3.	Subimage Frequency Contents	5-4
Figure 5-4.	2D STFT for the Image Analysis in Figure 5-2	5-5
Figure 5-5.	Online STFT Spectrogram Analyzer Panel	5-6
Figure 5-6.	Offline Joint Time-Frequency Analyzer	5-8
Figure 5-7.	Instantaneous Spectrum Display	5-10
Figure 6-1.	STFT Spectrogram (Hanning Window)	6-4
Figure 6-2.	Gabor Spectrogram (Order Four)	6-5

Tables

Table 6-1.	Quadratic JTFA Algorithms	6-2
Table B-1.	Error Codes	B-1

About This Manual

This manual describes the LabVIEW Joint Time-Frequency Analysis Toolkit package. You can use the LabVIEW Joint Time-Frequency Analysis Toolkit to analyze time and frequency behavior of a signal simultaneously.

Organization of This Manual


This manual is organized as follows:

- Chapter 1, *Overview of the Joint Time-Frequency Analysis Toolkit*, lists the contents of the LabVIEW Joint Time-Frequency Analysis Toolkit and describes the installation procedure.
- Chapter 2, *The Need for Joint Time-Frequency Analysis*, explains the need for and the approaches to joint time-frequency analysis.
- Chapter 3, *Joint Time-Frequency Analysis Algorithms*, describes the algorithms used by the joint time-frequency analysis VIs. The JTFA algorithms implemented in this package fall into two categories: linear and quadratic.
- Chapter 4, *Joint Time-Frequency Analysis VIs*, describes the joint time-frequency analysis VIs.
- Chapter 5, *Joint Time-Frequency Analysis Applications*, introduces some JTFA applications. Because JTFA is relatively new, it is less known among practicing engineers and scientists, unlike the well-known Fourier analysis. The examples in this chapter reveal only the potential of JTFA—the power of JTFA has not been fully explored. We present these examples so that you can learn and apply JTFA to your own applications.
- Chapter 6, *Frequently Asked Questions*, addresses some questions frequently asked by JTFA users.
- Appendix A, *References*, lists reference material that contains more information on the theory and algorithms implemented in the JTFA VIs.
- Appendix B, *Error Codes*, lists the error codes returned by the joint time-frequency analysis VIs.

- Appendix C, *Customer Communication*, contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation.
- The *Glossary* contains an alphabetical list and description of terms used in this manual, including acronyms, abbreviations, metric prefixes, mnemonics, and symbols.
- The *Index* contains an alphabetical list of key terms and topics in this manual, including the page where you can find each one.

Conventions Used in This Manual

The following conventions are used in this manual:

bold	Bold text denotes the names of menus, menu items, parameters, dialog box buttons or options, and icons.
<i>italic</i>	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept.
<i>bold italic</i>	Bold italic text denotes a note.
monospace	Text in this font denotes text that you should literally enter from the keyboard. This font also is used for the proper names of disk drives, paths, directories, filenames, and extensions.
	This icon to the left of bold italicized text denotes a note, which alerts you to important information.

Related Documentation

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

- *LabVIEW User Manual*
- *LabVIEW Function and VI Reference Manual*
- *LabVIEW Online Reference*, available by selecting **Help»Online Reference**

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and technical support forms for you to complete. These forms are in Appendix C, *Customer Communication*, at the end of this manual.

Overview of the Joint Time-Frequency Analysis Toolkit

This chapter lists the contents of the LabVIEW Joint Time-Frequency Analysis Toolkit and describes the installation procedure.

Package Contents

Your LabVIEW Joint Time-Frequency Analysis Toolkit contains the following materials:

- Joint Time-Frequency Analysis Toolkit CD or diskettes
- *LabVIEW Joint Time-Frequency Analysis Toolkit Reference Manual*

Installation

The following sections contain instructions for installing the LabVIEW Joint Time-Frequency Analysis Toolkit on the Windows and Power Macintosh platforms.

Windows 95/NT

Complete the following steps to install the toolkit.

1. Launch Windows 95 or Windows NT.
2. If you have a CD-ROM drive, insert the LabVIEW Joint Time-Frequency Analysis Toolkit CD. If you need to install from diskettes, insert disk 1 of the Joint Time-Frequency Analysis Toolkit into the 3.5-inch disk drive.

3. From the **Start** menu, choose **Run** and type `A:\setup.exe`. If you are installing the toolkit from a different drive, specify the correct drive.
4. Follow the instructions on your screen.

Once you have completed the on-screen installation instructions, you are ready to use the LabVIEW Joint Time-Frequency Analysis Toolkit.

Windows 3.x

Complete the following steps to install the toolkit.

1. Launch Windows.
2. If you have a CD-ROM drive, insert the LabVIEW Joint Time-Frequency Analysis Toolkit CD. If you need to install from diskettes, insert disk 1 of the Joint Time-Frequency Analysis Toolkit into the 3.5-inch disk drive.
3. From the File Manager, run `SETUP.EXE`.
4. Follow the instructions on your screen.

Once you have completed the on-screen installation instructions, you are ready to use the LabVIEW Joint Time-Frequency Analysis Toolkit.

Power Macintosh

Complete the following steps to install the toolkit.

1. If you have a CD-ROM drive, insert the LabVIEW Joint Time-Frequency Analysis Toolkit CD. If you need to install from diskettes, insert disk 1 of the Joint Time-Frequency Analysis Toolkit into the 3.5-inch disk drive.
2. Double-click on the **JTFA Installer** icon.
3. Follow the instructions on your screen.

Once you have completed the on-screen installation instructions, you are ready to use the LabVIEW Joint Time-Frequency Analysis Toolkit.

Getting Help

Press `<Ctrl-H>` to access the online help.

The Need for Joint Time-Frequency Analysis

This chapter explains the need for and the approaches to joint time-frequency analysis.

Review of the Classical Fourier Transform

From a mathematical point of view, we can describe a given signal in many different ways. For instance, we can write the signal as a function of time, which shows how the signal amplitude changes over time. Alternatively, we can write the signal as a function of frequency, which tells us how frequently the amplitude changes. The bridge between time and frequency representations is the *Fourier transform*, first introduced by Jean Baptiste Joseph Fourier in 1807.

During the study of heat propagation and diffusion, Fourier found that a series of harmonically related sinusoids was useful in representing the temperature distribution throughout a body. Later he claimed that any periodic signal could be represented by such a series, and any aperiodic signal could be represented as a weighted integral of sinusoids.¹ By the 1820s, Fourier's revolutionary claims were proved mathematically by S.D. Poisson, A.L. Cauchy, and P.L. Dirichlet. Since then, the Fourier transform has become one of the most important signal analysis methods.

By applying the Fourier transformation, we can easily decompose any signal as a weighted sum of sinusoid functions as shown in Figure 2-1. Consequently, we can process either the signal time waveform or its corresponding set of sinusoid functions, depending on which form is more convenient. In addition to being linear, the Fourier transform provides a feasible way of computing the power spectrum for a signal. Because the power spectrum usually has a simpler pattern than the time waveform, it often serves as the fingerprint of the analyzed signal. In signal processing, we often want to represent certain attributes of the signal explicitly.

¹ Fourier's original paper had never been published because of J.L. Lagrange's vehement objections. Lagrange, a very important mathematician during that time, argued that trigonometric series were of very limited use.

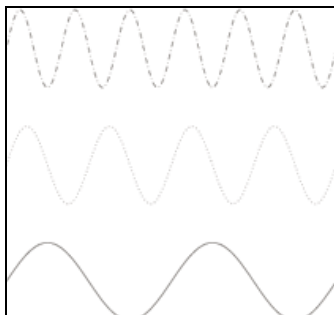


Figure 2-1. Basis Functions Used for Fourier Transform

Although the Fourier transform has been widely recognized in many disciplines, it possesses certain disadvantages that prevent its use in many important applications.

Figures 2-2 and 2-3 depict two common signals: seismic and ECG (electrocardiogram). Unlike the sinusoid functions used as the basis of the Fourier transform (see Figure 2-1) that extend over the entire time domain, the seismic signal only lives for a very short period, and the ECG basically consists of isolated bursts. Using the Fourier series to represent those signals, we need an infinite number of sinusoid functions that can cancel each other to achieve the near-zero points. Therefore, the classical Fourier series cannot economically represent these applications.

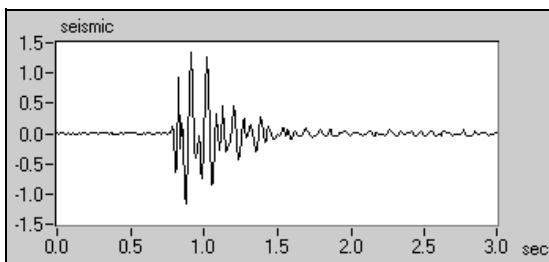


Figure 2-2. Seismic Signal

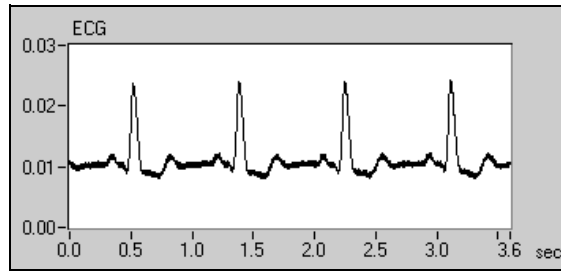


Figure 2-3. ECG Signal

As mentioned earlier, another important application of the Fourier transform is spectral analysis. Figure 2-4 illustrates a word, *hood*, spoken by a five-year-old boy. The bottom plot depicts the time waveform. The upper right plot depicts the conventional power spectrum. The conventional power spectrum shows that the word *hood* contains three main frequency tones; however, the power spectrum alone does not clearly indicate how those frequencies evolve over time. Obviously, the frequency tones of a speech signal are not constant. If they were, speech could not be used as a means of communication. Despite the fact that the frequency contents of most signals change with time, the classical Fourier theory only allows us to analyze a signal in the time domain or in the frequency domain.

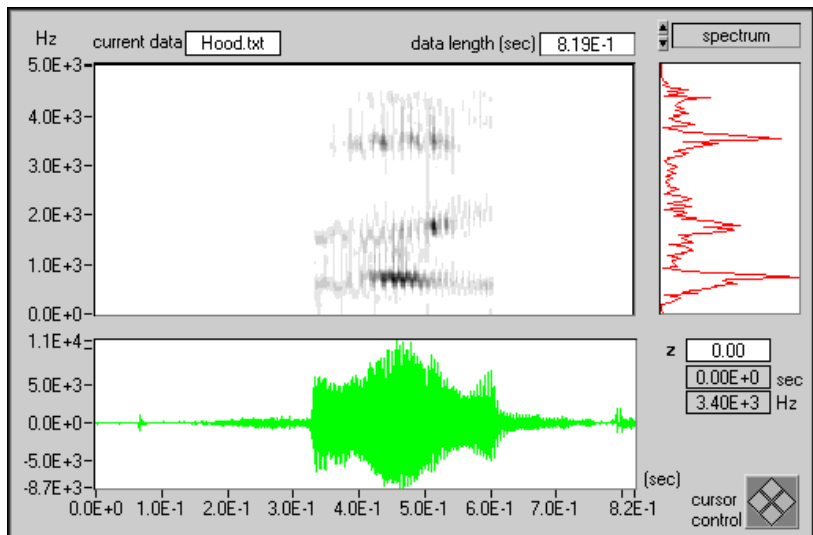


Figure 2-4. Speech Signal

The Need for JTFA

The middle plot of Figure 2-4 is a time-dependent spectrum, which plots the signal's energy as a function of both time and frequency. As shown, the time-dependent spectrum clearly reveals the pattern of the formants. From the formants, we see how the frequency changes. The relative brightness levels of the plot show the intensity of the frequencies. In this example, the JTFA (Joint Time-Frequency Analysis) enables us to better understand the mechanism of human speech.

Another important motivation for applying JTFA is the detection of noise-corrupted signals. In general, random noise tends to spread evenly across the time and frequency domains. On the other hand, the signal usually concentrates in a relatively short time period or narrow frequency band. If we can convert the noise-corrupted signal to the joint time-frequency domain, the local (or regional) Signal-to-Noise Ratio (SNR) can be improved substantially.

Figure 2-5 depicts the impulse signal received by the U.S. Department of Energy ALEXIS/BLACKBEARD satellite.¹ After passing through dispersive media such as the ionosphere, the impulse signal becomes the nonlinear chirp signal. As shown in Figure 2-5, random noise dominates both the time waveform and power spectrum. Neither indicate the existence of the impulse signal. However, from the time-dependent spectrum, we can immediately identify the presence of the chirp-type signal arching across the joint time-frequency domain. The horizontal lines correspond to radio carrier signals that basically remain unchanged over time.

¹ Data courtesy of Non-Proliferation and International Security Division, Los Alamos National Laboratory.

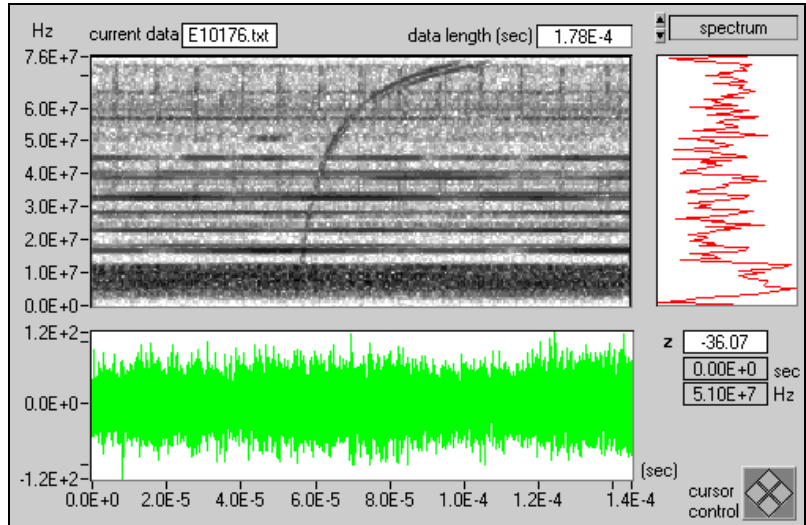


Figure 2-5. Ionized Impulse Signal

Based on the joint time-frequency representation, we can further mask the desired signal, as shown by the top plot in Figure 2-6. Then, we can apply the inverse transformation to recover the noiseless time waveform. The lower plot of Figure 2-6 illustrates the noisy and reconstructed signals. When the SNR is very low, as with many satellite signals, JTFA might offer the only opportunity to detect the signal of interest.

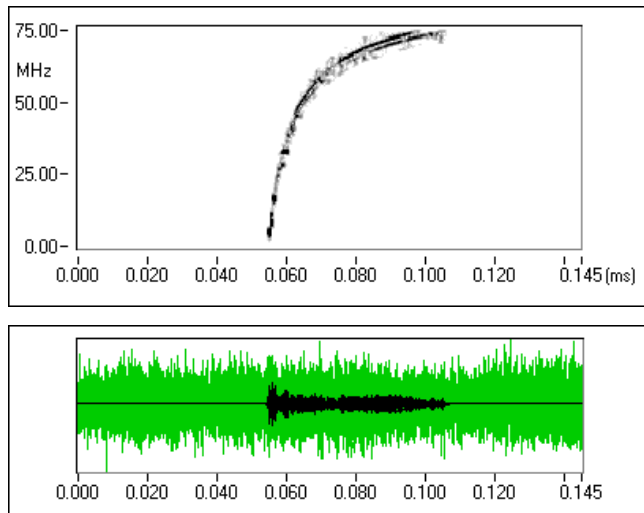


Figure 2-6. Reconstructed Signal

Basic Approaches to JTFA

The development of JTFA began more than half of a century ago. The most straightforward approach of characterizing a signal's frequency as a function of time was to divide the signal into several blocks that could be overlapped. Then, the Fourier transform was applied to every individual block of data to indicate the frequency contents of each. This process has become known as the short-time Fourier transform (STFT) and roughly reflects how frequency contents change over time. The size of the blocks determines the time accuracy. The smaller the block, the better the time resolution. However, frequency resolution is inversely proportional to the size of a block. While the small block yields good time resolution, it also deteriorates the frequency resolution and vice versa. Traditionally, this phenomena is known as the window effect.

From the concept of expansion and series, Dennis Gabor,¹ a Hungarian-born British physicist, suggested expanding a signal into a set of weighted frequency modulated Gaussian functions. Because the Gaussian function is concentrated in both the time and frequency domains, the weights describe the signal behavior in local time and frequency. We know the resulting presentation as the *Gabor expansion*. In fact, we can consider the Gabor expansion the inverse of the STFT. However, this inverse relationship was not clear during Gabor's lifetime and not well understood until the 1980s. At present, both the theory and implementation of the Gabor expansion and STFT are mature enough to be applied to solving real application problems.

As the linear JTFA develops, the quadratic JTFA (time-dependent spectrum) attracts great attention. The simplest time-dependent spectrum is the square of the STFT, which was named the STFT based spectrogram or the STFT spectrogram. As mentioned earlier, the STFT spectrogram suffers from the window effect. A more elegant method is the WVD (*Wigner-Ville Distribution*), which was originally developed in the context of quantum mechanics by a Hungarian-born American physicist, Eugene P. Wigner.² The WVD has high resolution as well as many other useful properties for signal analysis, but it suffers from crossterm interference. To reduce crossterm interference, we can use two proven algorithms: *Cohen's class* and the Gabor expansion based spectrogram (also known as the *Gabor spectrogram*). Scientists at National Instruments developed the

¹ In 1970, Gabor (1900–1979) was awarded the Nobel Physics Prize for his discovery of the principles underlying the science of holography.

² Wigner's pioneering application of group theory to an atomic nucleus established a method for discovering and applying the principles of symmetry to the behavior of physical phenomena. In 1963, he was awarded the Nobel Physics Prize.

Gabor spectrogram in the early 1990s. Based on the conventional Gabor expansion and the WVD, the adaptive representation based spectrogram—the adaptive spectrogram—was also introduced by scientists at National Instruments.

Unlike the linear JTFA method, the quadratic JTFA method is not unique. This toolkit contains the following quadratic algorithms:

- adaptive spectrogram
- Cohen's class
 - Choi–Williams distribution
 - cone-shaped distribution
 - STFT spectrogram
 - Wigner–Ville distribution
- Gabor spectrogram

Which method should you use? Often, the choice is application dependent. Through these methods, you can process those signals that the conventional Fourier transform cannot handle.

Joint Time-Frequency Analysis Algorithms

This chapter describes the algorithms used by the joint time-frequency analysis VIs. The JTFA algorithms implemented in this package fall into two categories: linear and quadratic. For more detailed information on a particular algorithm, consult Qian's (1996) and Cohen's (1995) works.

Linear JTFA Algorithms

Linear JTFA includes the following methods:

- Gabor expansion (considered the inverse STFT)
- STFT (short-time Fourier transform used for computing the Gabor coefficients)
- adaptive representation (considered the inverse adaptive transform)
- adaptive transform

Gabor Expansion and STFT

The Gabor expansion represents a signal $s[i]$ as the weighted sum of the frequency-modulated and time-shifted function $h[i]$:

$$s[i] = \sum_m \sum_{n=0}^{N-1} C_{m,n} h[i - m\Delta M] e^{j2\pi ni/N} \quad (3-1)$$

where the Gabor coefficients $C_{m,n}$ are computed by the STFT:

$$C_{m,n} = STFT[m\Delta M, n] = \sum_{i=0} s[i] \gamma^*[i - m\Delta M] e^{-j2\pi ni/N}$$

where N denotes the number of frequency bins, and ΔM denotes the time sampling interval. You can use any function as $\gamma[i]$, as long as its dual function $h[i]$ exists. For the perfect reconstruction, the oversampling rate, $N/\Delta M$, must be greater than or equal to one. For a given $h[i]$ (or $\gamma[i]$), use the FastDual VI to compute the corresponding dual function.

If the STFT is not used for computing the Gabor coefficient $C_{m,n}$, there are no restrictions for $\gamma[i]$ or the ratio $N/\Delta M$.

Adaptive Representation and Adaptive Transform

In the Gabor expansion, the elementary functions $h[i-m\Delta M]e^{j2\pi ni/N}$ are time-shifted and frequency-modulated versions of the single prototype function $h[i]$ (see equation (3-1)). To better match the analyzed signal, the adaptive representation was developed to decompose the signal $s[i]$ as a sum of weighted linear adaptive chirp modulated Gaussian functions:

$$s[i] = \sum_{k=0}^{D-1} A_k h_k[i] \quad (3-2)$$

where the linear chirp modulated Gaussian function $h_k[i]$ is defined by

$$h_k[i] = (\alpha_k \pi)^{-0.25} \exp \left\{ -\frac{[i - i_k]^2}{2\alpha_k} + j \left(2\pi f_k [i - i_k] + \frac{\beta_k}{2} [i - i_k]^2 \right) \right\}$$

which has four-tuple parameters $(\alpha_k, i_k, f_k, \beta_k)$. Therefore, the adaptive representation is more flexible than the elementary function used in the Gabor expansion.

The parameter D in equation (3-2) denotes the total number of elementary functions used by $h_k[i]$. A_k is the weight of each individual $h_k[i]$, as computed by the adaptive transform.

Scientists at National Instruments and Mallat and Zhang independently developed the adaptive representation (also known as the *matching pursuit*). The adaptive methods in this toolkit were implemented with the adaptive oriented orthogonal projective decomposition algorithm. The source code for this algorithm was developed by Professor Qinye Yin and Zhifang Ni at Xi'an Jiaotong University, China (Yin 1997).

Quadratic JTFA Algorithms

The quadratic JTFA algorithms include the following:

- STFT spectrogram
- Pseudo Wigner–Ville distribution (PWVD)
- Cohen’s class
- Choi–Williams distribution (CWD)
- cone-shaped distribution
- Gabor spectrogram
- adaptive spectrogram

STFT Spectrogram

The STFT-based spectrogram is defined as the square of the STFT:

$$SP[m\Delta M, n] = \left| \sum_{i=0}^{N-1} s[i]\gamma[i - m\Delta M]e^{-j2\pi ni/N} \right|^2$$

where N denotes the number of frequency bins, and ΔM denotes the time sampling interval. The STFT based spectrogram is simple and fast but suffers from the window effect.

Figures 3-1 and 3-2 illustrate the window effect of the STFT spectrogram. With a narrowband window (Figure 3-1), the time-dependent spectrum has high frequency resolution but poor time resolution. With a wideband window (Figure 3-2), the time-dependent spectrum has poor frequency resolution but high time resolution.

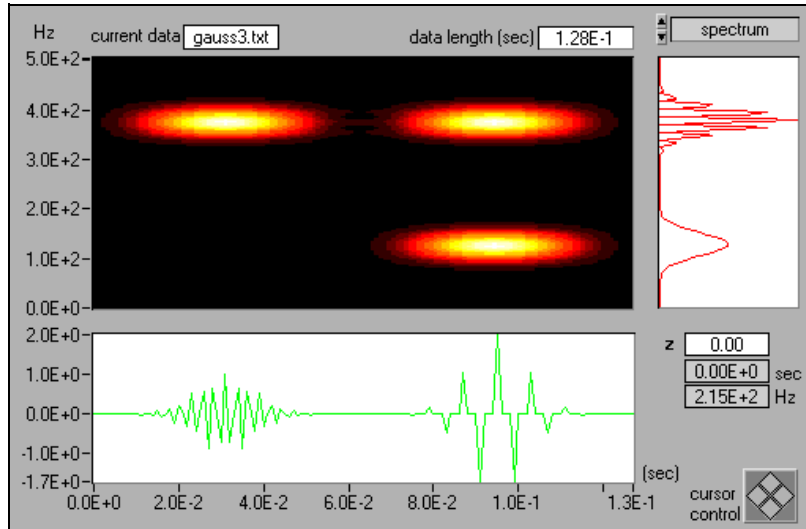


Figure 3-1. STFT Spectrogram with a Narrowband Hanning Window for the Three-Tone Test Signal

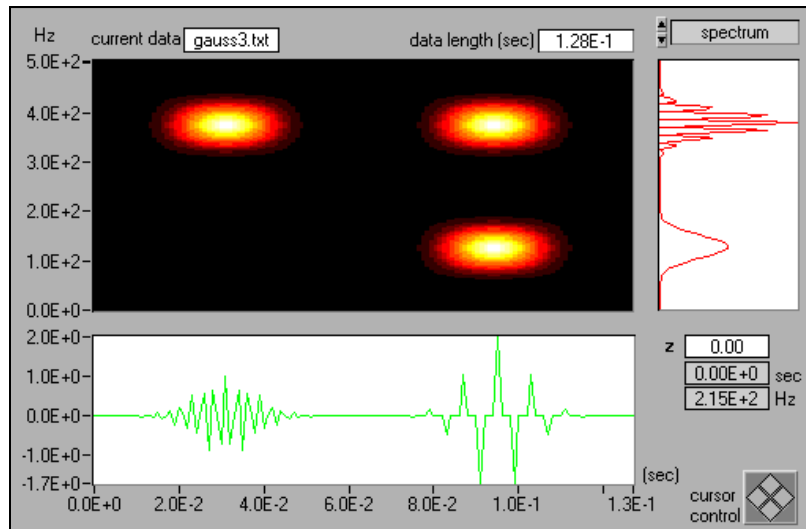


Figure 3-2. STFT-based Spectrogram with a Wideband Hanning Window for the Three-Tone Test Signal

Wigner–Ville Distribution and Pseudo Wigner–Ville Distribution

For a signal $s[i]$, the Wigner–Ville distribution (WVD) is

$$WVD[i, k] = \sum_{m=-L/2}^{L/2} R[i, m] e^{-j2\pi km/L}$$

where the function $R[i, m]$ is the *instantaneous correlation* given by¹

$$R[i, m] = z[i + m]z^*[i - m]$$

The WVD can also be computed by

$$WVD[i, k] = \sum_{m=-L/2}^{L/2} \Re[i, m] e^{j2\pi km/L}$$

where

$$\Re[i, m] = Z[i + m]Z^*[i - m]$$

and $Z[k]$ denotes the Fourier transform of $z[i]$. The Wigner–Ville distribution is simple and fast. It has the best joint time-frequency resolution of all known quadratic JTFA algorithms. However, if the analyzed signal contains more than one component, the WVD method suffers from crossterm interference.

Figure 3-3 depicts the WVD of the three-tone test signal. Three real signal terms are centered at (0.03 sec, 400 Hz), (0.09 sec, 100 Hz), and (0.09 sec, 400 Hz). Three crossterms exist, labeled in Figure 3-3, as 1, 2, and 3.

Autoterms at (0.03 sec, 400 Hz) and (0.09 sec, 400 Hz), which have different time centers, cause crossterm 1. Autoterms at (0.09 sec, 100 Hz) and (0.09 sec, 400 Hz), which have different frequency centers, cause crossterm 3. Autoterms at (0.03 sec, 400 Hz) and (0.09 sec, 100 Hz) create crossterm 2. The crossterm reflects the correlation between a pair of corresponding autoterms, always sits halfway between two corresponding autoterms, and oscillates frequently. Although its magnitude can be very large, its average usually is limited.

¹ $z[i]$ is the analytical or interpolated form of $s[i]$. See Qian (1996) for more details.

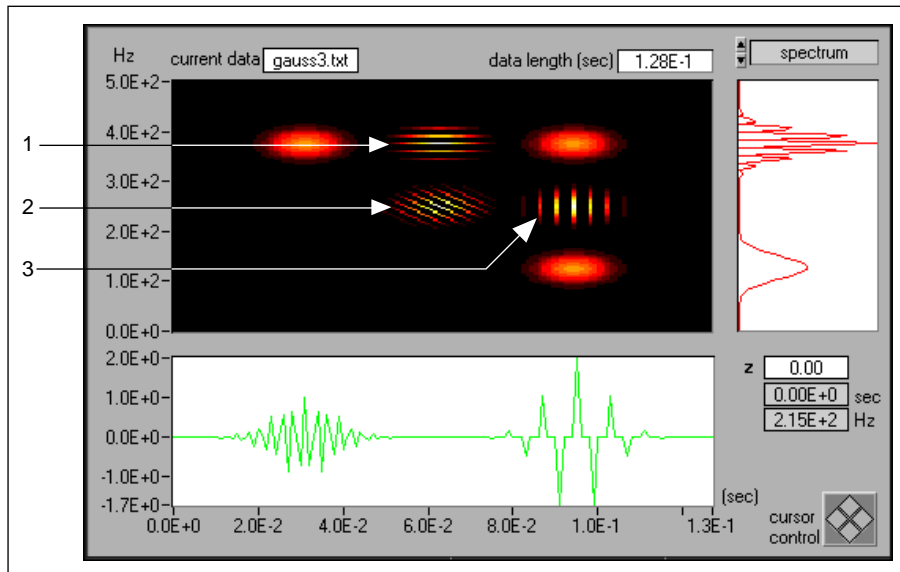


Figure 3-3. Wigner–Ville Distribution for the Three-Tone Test Signal

To alleviate the crossterm interference, we can assign different weights to the instantaneous correlation $R[i, m]$ to suppress the less important parts and enhance the fundamental parts.

Traditionally, two methods exist for applying the weighting function to the instantaneous correlation $R[i, m]$. The first is in the time domain:

$$PWVD[i, k] = \sum_{m=-L/2}^{L/2} w[m]R[i, m]e^{-j2\pi km/L} \quad (3-3)$$

which is called the Pseudo Wigner–Ville distribution (PWVD). PWVD effectively suppresses crossterms corresponding to a pair of autoterms with different time centers, such as crossterms 1 and 2 in Figure 3-3. Figure 3-4 illustrates the PWVD with the Gaussian window function $w[m]$. Compared with the WVD in Figure 3-3, the PWVD successfully eliminates crossterms 1 and 2.

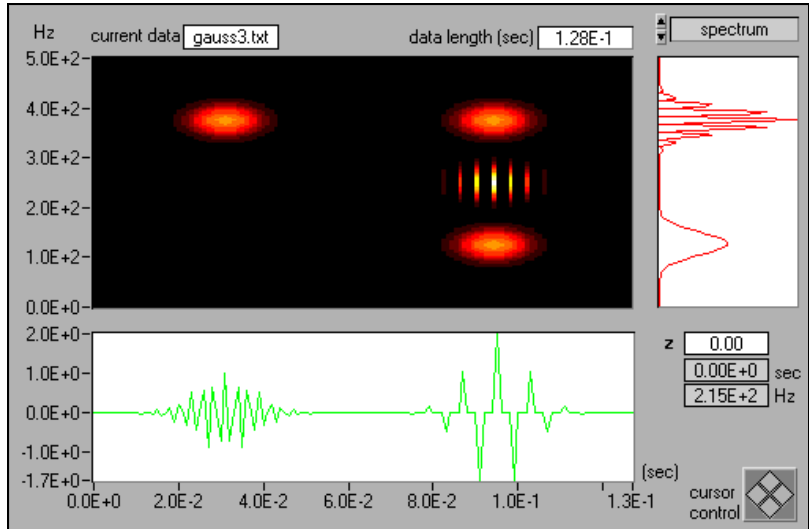


Figure 3-4. Pseudo Wigner–Ville Distribution (with Gaussian Window $w[m]$) for the Three-Tone Test Signal

In the second method, we assign weights to the instantaneous correlation $R[i, m]$ in the frequency domain:

$$WVD[i, k] = \sum_{m=-L/2}^{L/2} H[m] \Re[i, m] e^{j2\pi km/L} \quad (3-4)$$

This weighting function effectively suppresses crossterms corresponding to a pair of autoterms with different frequencies, such as crossterms 2 and 3 in Figure 3-3. Figure 3-5 illustrates the PWVD with the Gaussian window function $H[m]$, and, compared with the WVD in Figure 3-3, the PWVD successfully eliminates crossterms 2 and 3.

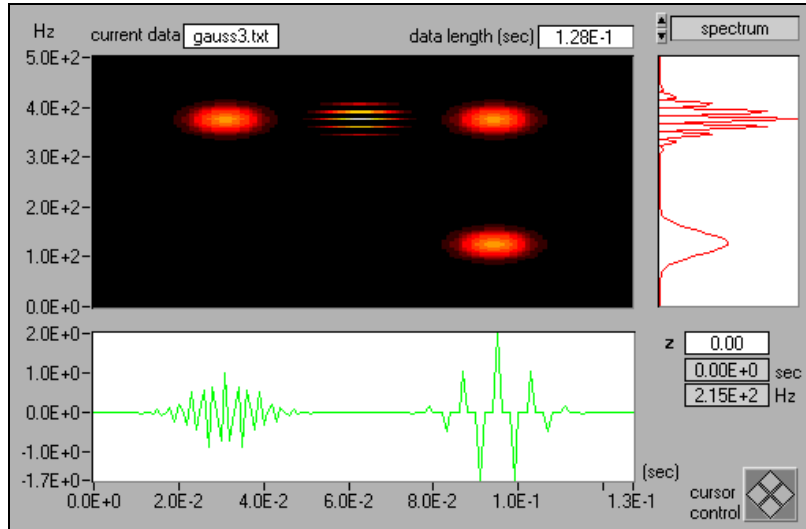


Figure 3-5. Pseudo Wigner–Ville Distribution for the Three-Tone Test Signal

Notice that equation (3-4) is equivalent to

$$PWVD[i, k] = \sum_{m=-L/2}^{L/2} \left(\sum_n h[n]R[i-n, m] \right) e^{-j2\pi km/L} \quad (3-5)$$

where $h[n]$ is the inverse Fourier transform of $H[m]$ in equation (3-4).

Cohen’s Class

Combining the two approaches from equations (3-3) and (3-5),

$$C[i, k] = \sum_{m=-L/2}^{L/2} \sum_n \Phi[n, m]R[i-n, m]e^{-j2\pi km/L} \quad (3-6)$$

where $\Phi[i, m]$ denotes the kernel function. The window functions $w[m]$ in equation (3-3) and $h[m]$ in equation (3-5) are special cases of $\Phi[i, m]$ in equation (3-6).

In 1966, Leon Cohen developed the representation $C[i, k]$ in equation (3-6), so it is traditionally known as *Cohen’s class*. Compared with the PWVD in equation (3-3) or (3-5), the Cohen’s class method is more general and flexible. Most quadratic equations known so far, such as the STFT spectrogram, WVD, PWVD, CWD (Choi–Williams distribution), and the cone-shaped distribution, belong to Cohen’s class.

Choi–Williams Distribution

When the kernel function in equation (3-6) is defined by

$$\Phi[i, m] = \sqrt{\frac{\alpha}{4\pi m^2}} e^{-\alpha i^2 / (4m^2)} \quad (3-7)$$

the yield is the Choi–Williams distribution (CWD). By adjusting the parameter α in equation (3-7), you balance crossterm interference and time-frequency resolution. The greater α , the less smoothing. Figure 3-6 illustrates the CWD for $\alpha = 1$. The CWD can effectively suppress the crossterm caused by two autoterms with different time and frequency centers, such as term 2 in Figure 3-3. However, the CWD method cannot reduce those crossterms that correspond to autoterms with the same time center (term 3) or the same frequency center (term 1). Furthermore, the computation speed of the CWD is very slow.

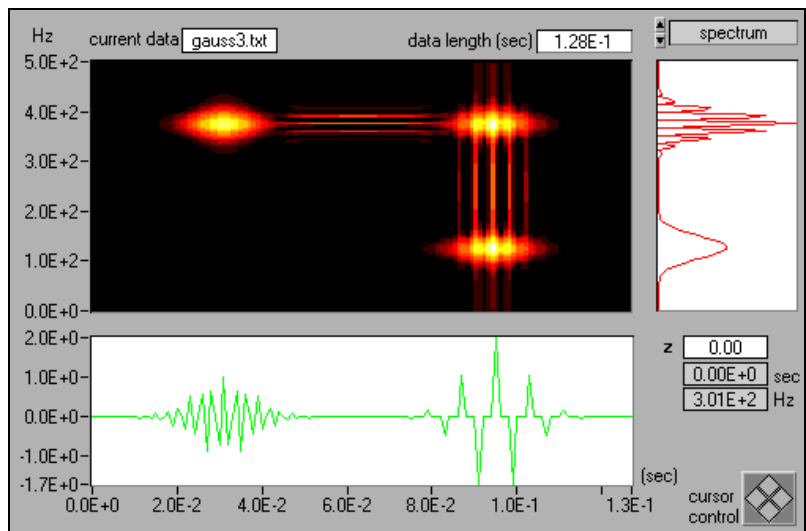


Figure 3-6. CWD ($\alpha = 1$) for the Three-Tone Test Signal

Cone-Shaped Distribution

When the kernel function in equation (3-6) is defined by

$$\Phi[i, m] = \begin{cases} \frac{\alpha m^2}{c} & \text{for } i < |m| \\ 0 & \text{otherwise} \end{cases} \quad (3-8)$$

the yield is the cone-shaped distribution. In this package, the constant c is set to 500. By adjusting the parameter α in equation (3-8), you can balance crossterm interference and time-frequency resolution. The greater α , the less smoothing. Figure 3-7 illustrates the cone-shaped distribution for $\alpha = 1$. The cone-shaped distribution effectively suppresses terms 2 and 3 from Figure 3-3, but it cannot reduce the crossterms corresponding to autoterms with the same frequency center (term 1). The cone-shaped distribution is faster than the CWD method.

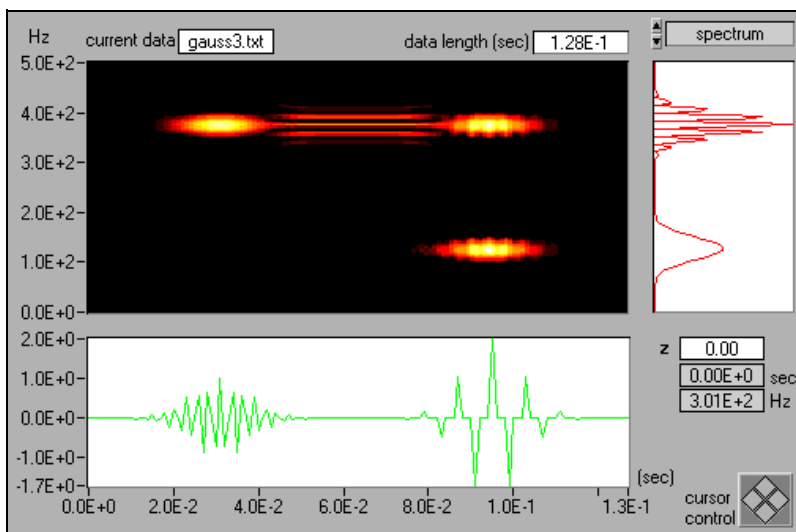


Figure 3-7. Cone-shaped Distribution ($\alpha = 1$) for the Three-Tone Test Signal

Gabor Spectrogram

In addition to applying the pseudo Wigner–Ville distribution’s window method, we can apply the Gabor expansion to a signal to identify the significance of each term to the signal’s energy at point $[i, k]$. We can then preserve those terms that have major contributions at point $[i, k]$ and remove those terms that have a negligible influence on the signal’s energy. Because it is a Gabor expansion based spectrogram, the resulting method is the Gabor spectrogram. The Gabor spectrogram is defined by

$$GS_D[i, k] = \sum_{|m-m'|+|n-n'| \leq D} C_{m,n} C_{m',n'} WVD_{h,h'}[i, k]$$

where $WVD_{h,h'}[i, k]$ denotes the cross WVD of frequency modulated Gaussian functions. The order of the Gabor spectrogram, D , controls the degree of smoothing. For $D = 0$, $GS_0[i, k]$ is non-negative and similar to the STFT spectrogram. As D goes to infinity, the Gabor spectrogram converges to the WVD.

A lower-order Gabor spectrogram has less crossterm interference but lower resolution. A higher order Gabor spectrogram has better resolution but more crossterms. Moreover, the higher the order, the longer the computation time. For best results, choose an order of three to five. The Gabor spectrogram has better resolution than the STFT spectrogram and much less crossterm interference than the cone-shaped, Choi–Williams, or Wigner–Ville distributions.

Figure 3-8 illustrates the fourth order Gabor spectrogram for the three-tone test signal. It possesses high time-frequency resolution and does not have the crossterm interference appearing in the cone-shaped, Choi–Williams, or Wigner–Ville distributions. The computational speed of the fourth order Gabor spectrogram is slower than the STFT spectrogram and WVD but faster than the CWD or cone-shaped distribution.

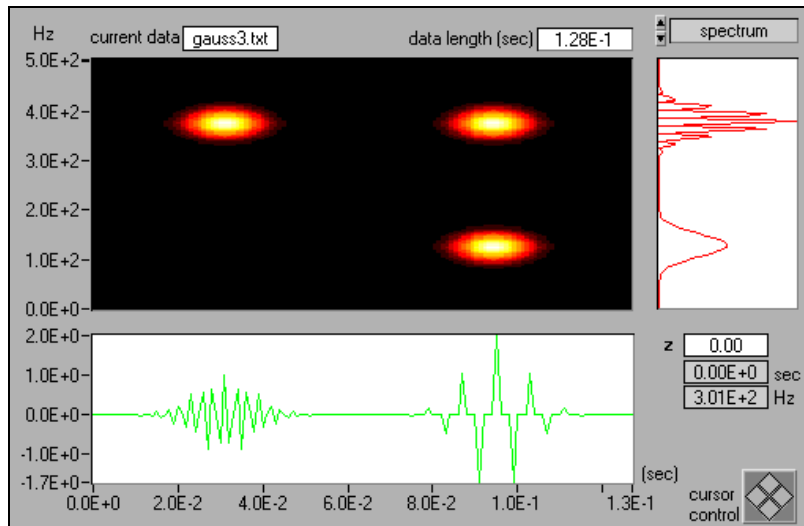


Figure 3-8. Gabor Spectrogram (Order Four) for the Three-Tone Test Signal

Scientists at National Instruments developed the Gabor spectrogram method, winning the *EDS (Electronic Design News) 1992 Software Award* and the *1993 R&D 100* award.

Adaptive Spectrogram

The adaptive spectrogram method is an adaptive representation based spectrogram (see equation (3-2)) computed by

$$AS[i, n] = 2 \sum_{k=0}^{D-1} |A_k|^2 \exp \left\{ -\frac{[i-i_k]^2}{\alpha_k} - (2\pi)^2 \alpha_k [n-f_k - \beta_k i]^2 \right\} \quad (3-9)$$

The adaptive spectrogram achieves the best joint time-frequency resolution if the analyzed signal is a sum of linear chirp modulated Gaussian functions. For example, Figure 3-9 shows that the adaptive spectrogram effectively describes the three-tone test signal. Unfortunately, the computation speed of the adaptive spectrogram increases exponentially with the analyzed data size.

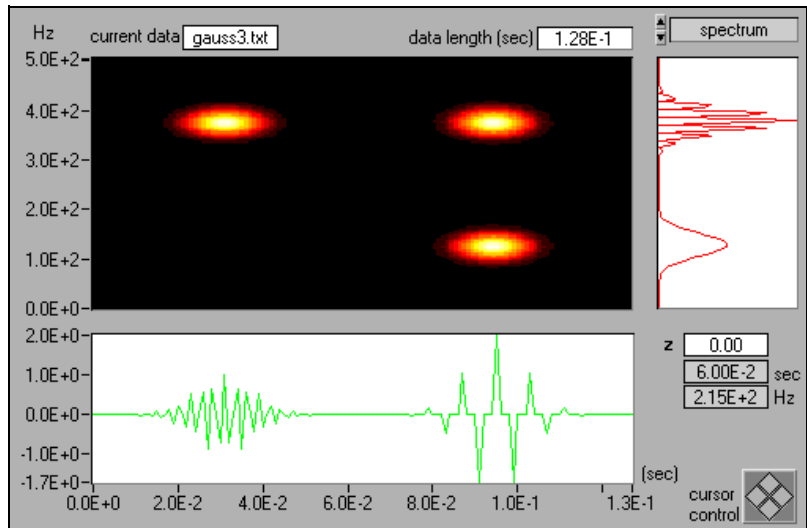


Figure 3-9. Adaptive Spectrogram for the Three Gaussian Tones

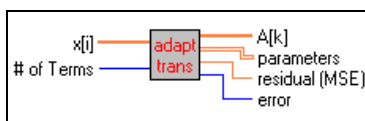
Scientists at National Instruments uniquely developed the adaptive spectrogram included in this package.

Joint Time-Frequency Analysis VIs

This chapter describes the joint time-frequency analysis VIs.

Adaptive Transform

Computes the coefficients of the adaptive representation.



[CDB]

$x[i]$ is the time waveform, either a real-valued or analytical signal.

[I32]

of Terms determines the maximum number of elementary functions used for the adaptive representation. The more elementary functions, the more accurate the presentation; however, computation time increases as the number of elementary functions increases.

[CDB]

$A[k]$ is a 1D array indicating the weight of each elementary function $h_k[i]$.

[DBL]

parameters is a 2D array indicating four-tuple parameters of elementary functions $h_k[i]$:

$$h_k[i] = (\alpha_k \pi)^{-0.25} \exp \left\{ -\frac{[i - i_k]^2}{2\alpha_k} + j \left(2\pi f_k [i - i_k] + \frac{\beta_k}{2} [i - i_k]^2 \right) \right\} \quad (4-1)$$

Column	Parameter	k^{th} Elementary Function
0	i_k	time center
1	α_k	variance
2	f_k range: [0, 0.5]	normalized center frequency
3	β_k	frequency changing rate

[DBL]

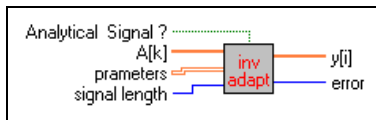
residual (MSE) indicates the MSE (Mean Square Error) of the signal $\mathbf{x}[\mathbf{i}]$ and the adaptive representation. The MSE reduces as the number of elementary functions increases.

[32]

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Inverse Adaptive Transform

Reconstructs the time waveform based on the adaptive representation.



[TF]

Analytical Signal? determines if the reconstructed signal is an analytical signal.

[1D]

A[k] is a 1D array indicating the weight of each elementary function $h_k[i]$.

[2D]

parameters is a 2D array indicating the four-tuple parameters of the elementary function $h_k[i]$ from equation (4-1).

Column	Parameter	k^{th} Elementary Function
0	i_k	time center
1	α_k	variance
2	f_k range: [0, 0.5]	normalized center frequency
3	β_k	frequency changing rate

I32

signal length controls the length of the reconstructed signal.

[CDB]

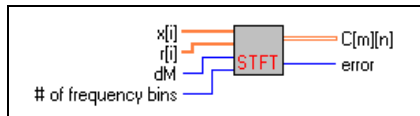
y[i] is the reconstructed time waveform.

I32

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Short-Time Fourier Transform

Computes the Gabor coefficients of the one-dimensional Gabor expansion, or the Gabor transform.

**[CDB]**

x[i] is the time waveform, either a real-valued or general complex signal.

[DBL]

r[i] is the analysis window function.

I32

dM is the Gabor time sampling interval. The length of the analysis function **r[i]** must be evenly divisible by **dM**.

I32

of frequency bins determines the number of columns of **C[m][n]**. It must be a power of two.

[CDB]

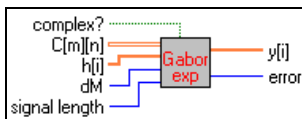
C[m][n] is the Gabor coefficient.

I32

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Gabor Expansion

Represents a signal $s[i]$ as the weighted sum of the frequency modulated function $\mathbf{h}[i]$.



complex? determines if the reconstructed signal is a complex value.



C[m][n] is the Gabor coefficient.



h[i] is the synthesis window function.



dM is the Gabor time sampling interval. The length of the synthesis function $\mathbf{h}[i]$ must be evenly divisible by **dM**.



signal length controls the length of the reconstructed signal.



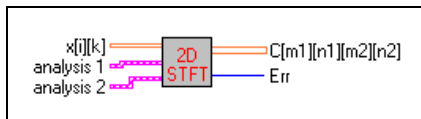
y[i] is the reconstructed time waveform.

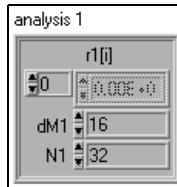
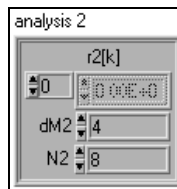


error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

2D STFT

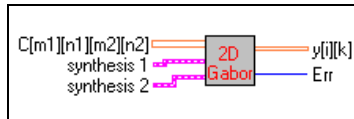
Computes the Gabor coefficients of a two-dimensional signal Gabor expansion.



[DBL] $\mathbf{x}[i][k]$ is a two-dimensional signal.**[EOL]****analysis 1** is the cluster for the analysis window function of the row.**[DBL]** $\mathbf{r1}[i]$ is the analysis function for the row of $\mathbf{x}[i][k]$.**[I32]****dM1** is the Gabor time sampling interval. The length of the analysis function $\mathbf{r1}[i]$ must be evenly divisible by **dM1**.**[I32]****N1** determines the number of elements for the second index $n1$ of the Gabor coefficients. It must be a power of two.**[EOL]****analysis 2** is the cluster for the analysis window function of the column.**[DBL]** $\mathbf{r2}[k]$ is the analysis window function for the column of $\mathbf{x}[i][k]$.**[I32]****dM2** is the Gabor time sampling interval. The length of the analysis function $\mathbf{r2}[i]$ must be evenly divisible by **dM2**.**[I32]****N2** determines the number of elements for the fourth index $n2$ of the Gabor coefficients. It must be a power of two.**[CDB]** $\mathbf{C}[m1][n1][m2][n2]$ is the Gabor coefficient.**[I32]****Err** indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

2D Gabor Expansion

Computes the Gabor expansion for a two-dimensional signal.

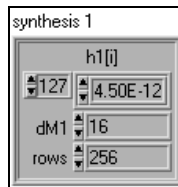


C

$C[m1][n1][m2][n2]$ is the Gabor coefficient.

Synthesis 1

synthesis 1 is the cluster for the synthesis window function of the row.



h1

$h1[i]$ is the synthesis function for the row of $x[i][k]$.

dM1

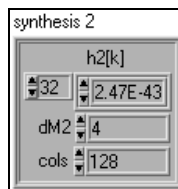
dM1 is the Gabor time sampling interval. The length of the analysis function $h1[i]$ must be evenly divisible by **dM1**.

N1

N1 determines the number of elements for the second index **n1** of the Gabor coefficients. It must be a power of two.

Synthesis 2

synthesis 2 is the cluster for the analysis window function of the column.



h2

$h2[k]$ is the synthesis window function for the column of $x[i][k]$.

dM2

dM2 is the Gabor time sampling interval. The length of the synthesis function $h2[i]$ must be evenly divisible by **dM2**.

N2

N2 determines the number of elements for the fourth index **n2** of the Gabor coefficients. It must be a power of two.

[DBL]

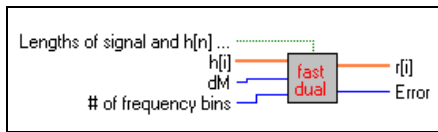
$y[i][k]$ is the reconstructed two-dimensional signal.

[I32]

Err indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Fast Dual

Computes the dual function of a given function $h[i]$.

**[TF]**

Lengths of signal and $h[i]$ are the same? (T) determines if lengths of the signal and the window function $h[i]$ are the same. If the length of $h[i]$ is the same as the analyzed signal, the dual function solution of the given function $h[i]$ is much broader. However, if the length of the signal is very long, the dual function solution may not be realistic.

[DBL]

$h[i]$ is the analysis or synthesis window function.

[I32]

dM is the Gabor time sampling interval. The length of $h[i]$ must be evenly divisible by dM .

[I32]

of frequency bins controls the number of frequency bins of the resulting STFT and Gabor expansion. It must be a power of two. The length of $h[i]$ must be evenly divisible by **# of frequency bins**. The ratio of **# of frequency bins** to dM is the oversampling rate. For stable reconstruction, the oversampling rate must be greater than or equal to 1:

$$\text{oversampling} = \frac{\text{the number of frequency bins}}{dM} \geq 1$$

[DBL]

$r[i]$ is the dual function of $h[i]$. $r[i]$ and $h[i]$ constitute a pair of analysis and synthesis functions for the STFT (Gabor transform) and Gabor expansion (considered the inverse Gabor transform).

[I32]

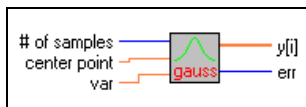
Error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions (–20083 no solution; 20001 rank deficiency).

Normalized Gaussian Window Function

Computes the unit energy Gaussian window function defined by

$$y[i] = (\pi\sigma^2)^{-1/4} e^{-\frac{(i-t_0)^2}{2\sigma^2}} \quad (4-2)$$

which is optimally concentrated in time and frequency domains simultaneously.



I32

of samples determines the length of the normalized Gaussian window function $y[i]$.

DBL

center point determines the center point t_0 in equation (4-2).

DBL

var determines the variance of the normalized Gaussian function σ^2 in equation (4-2). If the Normalized Gaussian Window Function VI is used for the STFT or Gabor expansion, the variance is

$$\sigma^2 = \frac{\text{the number of frequency bins}}{2\pi} \times dM$$

In this case, the MSE of the dual function is minimum, as computed by the FastDaul VI and the Normalized Gaussian Window Function VI. The resulting representation is known as the orthogonal-like Gabor expansion.

DBL

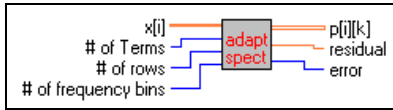
$y[i]$ is the resulting normalized Gaussian window function.

I32

err indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Adaptive Spectrogram

Computes the adaptive representation based spectrogram.



[CDB]

$x[i]$ is the time waveform, either a real-valued or analytical signal.

[I32]

of Terms determines the maximum number of elementary functions used for the adaptive representation. The more elementary functions, the more accurate the presentation. Unfortunately, computation time increases as the number of elementary functions increases.

[I32]

of rows determines the maximum number of rows of the spectrogram $p[i][k]$.

[I32]

of frequency bins determines the number of columns of the spectrogram $p[i][k]$ by

$$\text{number of columns} = \frac{\text{number of frequency bins}}{2} + 1 \quad (4-3)$$

of frequency bins must be a power of two.

[DBL]

$p[i][k]$ is the adaptive spectrogram.

[DBL]

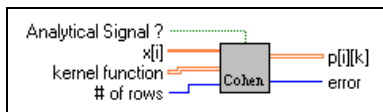
residual (MSE) indicates the MSE (Mean Square Error) of the signal $x[i]$ and the adaptive representation. The MSE reduces as the number of elementary functions increases.

[I32]

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Cohen's Class

Computes the general Cohen's class.



TF

Analytical Signal? determines whether to process the analytical signal.

[DBL]

$x[i]$ is the time waveform.

[DBL]

kernel function is the first quarter of a user-defined two-dimensional kernel function. The number of rows of the kernel function determines the number of distinct frequencies (the number of columns) of the spectrogram $p[i][k]$. It must be a power of two.

I32

of rows determines the maximum number of rows of the spectrogram $p[i][k]$.

[DBL]

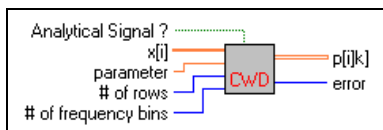
$p[i][k]$ is the Cohen's class.

I32

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

CWD (Choi–Williams Distribution)

Computes the Choi–Williams distribution.



TF

Analytical Signal? determines whether to process the analytical signal.

[DBL]

$x[i]$ is the time waveform.

[DBL]

parameter controls the resolution and crossterm interference. **parameter** must be greater than zero. Decreasing **parameter** suppresses the crossterm interference in the resulting spectrogram. Unfortunately, decreasing **parameter** also reduces the resolution.

[I32]

of rows determines the maximum number of rows of the spectrogram $p[i][k]$.

[I32]

of frequency bins determines the number of columns of the spectrogram $p[i][k]$ by

$$\text{number of columns} = \frac{\text{number of frequency bins}}{2} \quad (4-4)$$

of frequency bins must be a power of two.

[DBL]

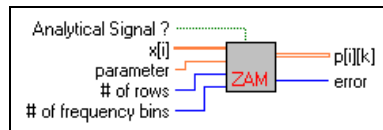
$p[i][k]$ is the Choi–Williams distribution.

[I32]

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Cone-Shaped Distribution

Computes the cone-shaped kernel distribution.

**[TF]**

Analytical Signal? determines whether to process the analytical signal.

[DBL]

$x[i]$ is the time waveform.

[DBL]

parameter controls the resolution and crossterm interference. **parameter** must be greater than zero. Decreasing **parameter** suppresses the crossterm interference in the resulting spectrogram. Unfortunately, decreasing **parameter** also reduces the resolution.

[I32]

of rows determines the maximum number of rows of the spectrogram $p[i][k]$.

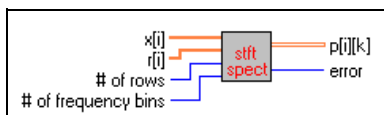
[I32]

of frequency bins determines the number of columns of the spectrogram $p[i][k]$ from equation (4-4). It must be a power of two.

[DBL] $p[i][k]$ is the cone-shaped distribution.**[I32]****error** indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

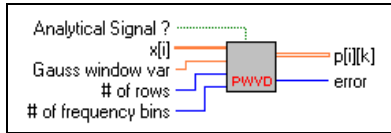
STFT Spectrogram

Computes the STFT based spectrogram.

**[DBL]** $x[i]$ is the time waveform.**[DBL]** $r[i]$ is the analysis window function.**[I32]****# of rows** determines the maximum number of rows of the spectrogram $p[i][k]$.**[I32]****# of frequency bins** determines the number of columns of the spectrogram $p[i][k]$ from equation (4-3). It must be a power of two.**[DBL]** $p[i][k]$ is the STFT spectrogram.**[I32]****error** indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

PWVD (Pseudo Wigner–Ville Distribution)

Computes the windowed Wigner–Ville distribution.



TF

Analytical Signal? determines whether to process the analytical signal.

[DBL]

$x[i]$ is the time waveform.

[DBL]

Gauss window var controls the resolution and crossterm interference. It must be greater than zero. The greater the variance, the better the resolution but the more severe the interference (the opposite is also true). This process suppresses only those crossterms corresponding to two autoterms with different time centers.

I32

of rows determines the maximum number of rows of the spectrogram $p[i][k]$.

I32

of frequency bins determines the number of columns of the spectrogram $p[i][k]$ from equation (4-4). It must be a power of two.

[DBL]

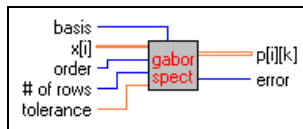
$p[i][k]$ is the pseudo Wigner–Ville distribution.

I32

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

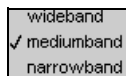
Gabor Spectrogram (Gabor Expansion-Based Spectrogram)

Computes the Gabor expansion based spectrogram with given analysis functions.



I32

basis selects the analysis and synthesis functions used for the Gabor spectrogram. Three type of functions are provided:



The default is *mediumband*.

DOB

x[i] is the time waveform, either a real-valued or complex signal.

I32

order balances the resolution and crossterm interference. **order** must be greater than or equal to zero. The greater the order, the better the resolution but the more severe the interference (the opposite is also true). When **order** is set to zero, the Gabor spectrogram is non-negative. As **order** gets larger, the Gabor spectrogram converges to the Wigner–Ville distribution.

Computation time is related to the **order**. The higher the **order**, the longer the computation time. For most applications, choose an **order** between three and five.

I32

of rows determines the maximum number of rows of the spectrogram **p[i][k]**.

DBL

tolerance controls the precision of the resulting Gabor spectrogram. The smaller the tolerance, the more computation time required. The default value is 10^{-2} .

DOB

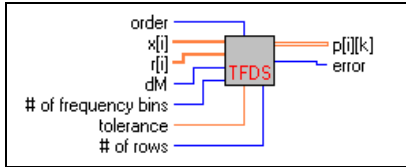
p[i][k] is the Gabor spectrogram.

I32

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Time-Frequency Distribution Series

Computes the Gabor expansion based spectrogram, which is the engine of the GaborSpectrogram VI.



I32

order balances the resolution and crossterm interference. It must be greater than or equal to zero. The larger the **order**, the better the resolution but the more severe the interference (the opposite is also true). When **order** is set to zero, the Gabor spectrogram is non-negative. As **order** increases, the Gabor spectrogram converges to the Wigner–Ville distribution.

Computing time is related to **order**. The higher the **order**, the longer the computation time. For most applications, choose an **order** between three and five.

[CDB]

x[i] is the time waveform, either a real-valued or complex signal.

[CDB]

r[i] is the analysis function. **r[i]** must be the dual function of the normalized Gaussian function in equation (4-2).

I32

dM is the Gabor time sampling interval. The length of the analysis function **r[i]** must be evenly divisible by **dM**.

I32

of frequency bins controls the number of columns of **p[i][k]**. When the analyzed signal **x[i]** is a real value, the number of columns of **p[i][k]** is determined by equation (4-3). When **x[i]** is complex, the number of columns of **p[i][k]** is equal to **# of frequency bins**.

The number of frequency bins must be a power of two. The length of the analysis function **r[i]** must be evenly divisible by **# of frequency bins**.

The ratio of **# of frequency bins** to **dM** is the oversampling rate, which must be greater than or equal to 1:

$$\text{oversampling} = \frac{\text{the number of frequency bins}}{dM} \geq 1$$

DBL

tolerance controls the precision of the resulting Gabor spectrogram. The smaller the tolerance, the more computation time required. The default value is 10^{-2} .

I32

of rows determines the maximum number of rows of the spectrogram $p[i][k]$.

[DBL]

$p[i][k]$ is the Gabor spectrogram.

I32

error indicates a JTFA VI error. See Appendix B, *Error Codes*, for a list of JTFA error codes and their descriptions.

Joint Time-Frequency Analysis Applications

This chapter introduces some JTFA applications. Because JTFA is relatively new, it is less known among practicing engineers and scientists, unlike the well-known Fourier analysis. The examples in this chapter reveal only the potential of JTFA—the power of JTFA has not been fully explored. We present these examples so you can learn and apply JTFA to your own applications.

Unlike the traditional analysis in which you can analyze a signal only in the time domain or frequency domain, JTFA defines a signal in the joint time and frequency domain. Consequently, you can better understand and process the signal in which you are interested.

The examples in this chapter demonstrate VIs based on linear or quadratic algorithms. You can find all examples in `Examples.11b`.

Linear Algorithm Examples

The following two linear algorithm examples demonstrate noise reduction (denoise) and image analysis.

Denoise

Noise reduction is a powerful JTFA application. In general, random noise evenly distributes over the entire joint time-frequency domain because it is not limited to a short time period or narrow frequency band. A signal's joint time-frequency representation always concentrates in a relatively small region. As a result, the regional SNR in the joint time-frequency domain can be much higher than that in either the time or frequency domain. In other words, the noise-corrupted signal is easier to recognize from the joint time-frequency domain. After identifying the signal component, you can apply a mask to filter out the signal components and take the inverse transform to obtain the noise-free time waveform.

Figure 5-1 illustrates the Gabor expansion based denoise example. By adjusting the noise control knob, you vary the noise level. In Figure 5-1, the original SNR is -0.37 db. From examining the time waveform, power spectrum, and Gabor coefficients, you may notice that the signal is much easier to recognize from the Gabor coefficients than from either the time waveform or the spectrum. The following procedure describes the Gabor expansion based denoise process:

1. Take the STFT with respect to the noise signal $x[n]$.
2. Mask the signal STFT (or Gabor coefficients) from the joint time-frequency domain.
3. To get the noise-reduced time waveform $x_1[n]$, compute the Gabor expansion of the signal's Gabor coefficients obtained in step 2.
4. Repeat steps 1 through 3. The number of iterations depends on the original SNR. The lower the SNR, the more iterations.

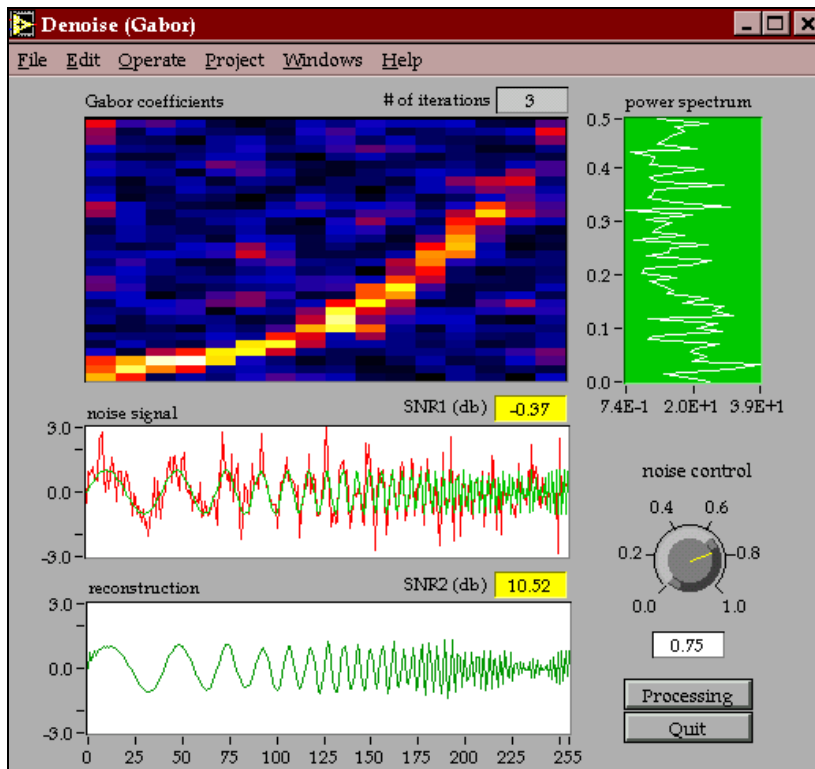


Figure 5-1. Gabor Expansion Based Denoise

In this example, we have three iterations, and the resulting SNR is 10.52 db. The bottom plot depicts the time waveform based on the prominent Gabor coefficients selected from the joint time-frequency domain. Notice an improvement of approximately 11 db—an impossible improvement using traditional techniques.

Scientists at National Instruments and Hughs Aircraft jointly developed this iterative denoise method.

Image Analysis

By applying the 2D STFT method, you can decompose a two-dimensional image into several subimages, as shown in Figure 5-2. Figure 5-3 describes the corresponding frequency contents of each individual subimage. If you adjust the analysis window functions, you can have each subimage represent only the information in which you are interested.

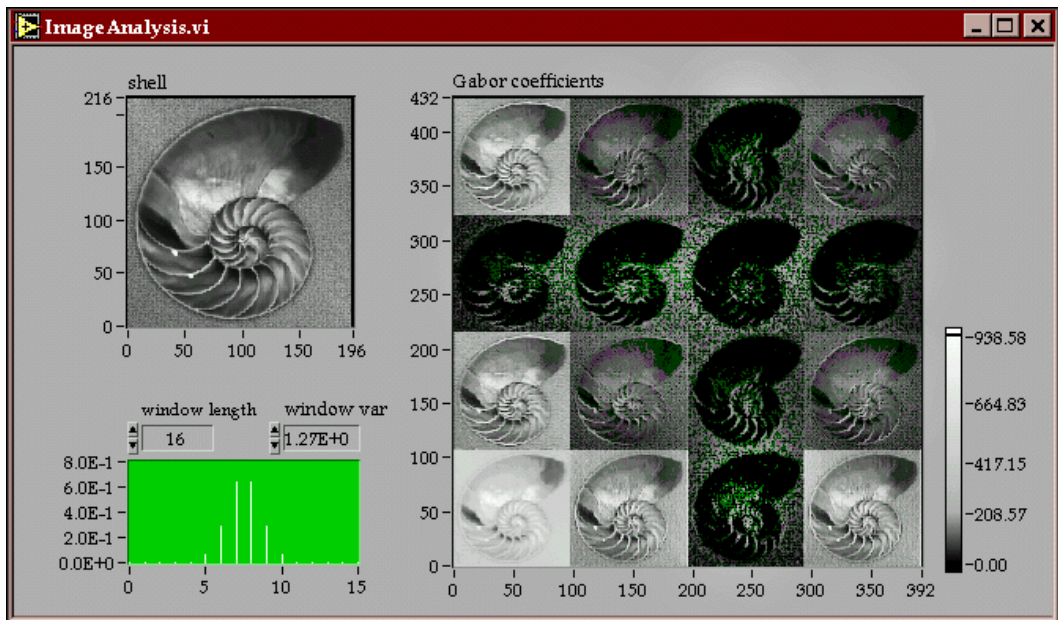


Figure 5-2. 2D STFT for Image Analysis

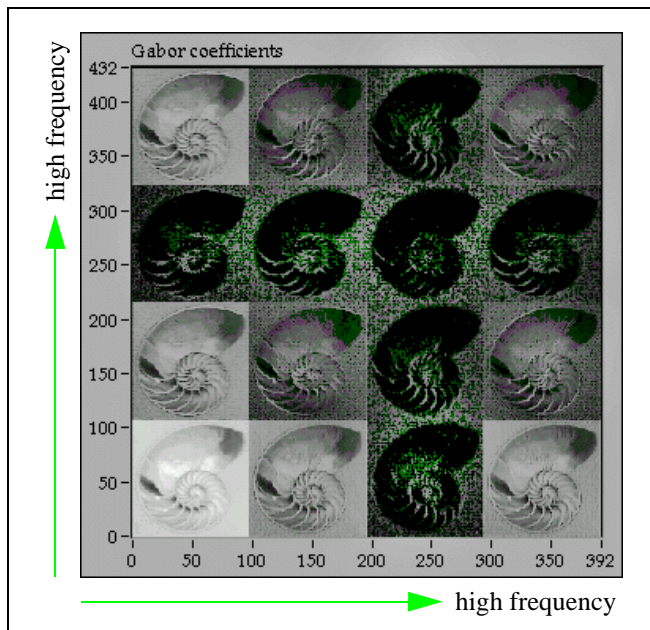


Figure 5-3. Subimage Frequency Contents

Figure 5-4 illustrates the 2D STFT VI used for the image analysis plotted in Figure 5-2. In this example, the row analysis window, $\mathbf{r1}[i]$, and column analysis function, $\mathbf{r2}[i]$, are the same. Moreover, $\mathbf{N1} = \mathbf{N2} = 4$ and $\mathbf{dM1} = \mathbf{dM2} = 2$. In other words, the oversampling rate is 2.

The relationship between the analysis windows and the resulting subimage is determined as follows:

- The ratio of the number of rows (between the original image and the new image) is equal to the oversampling rate $\mathbf{N1}/\mathbf{dM1}$. In this example, 2.
- The number of vertical subimages is equal to $\mathbf{N1}$.
- The ratio of the number of columns (between the original image and the new image) is equal to the oversampling rate $\mathbf{N2}/\mathbf{dM2}$. In this example, 2.
- The number of horizontal subimages is equal to $\mathbf{N2}$.
- The total number of subimages is equal to the product of the number of vertical subimages and the number of horizontal subimages. In this example, 16.

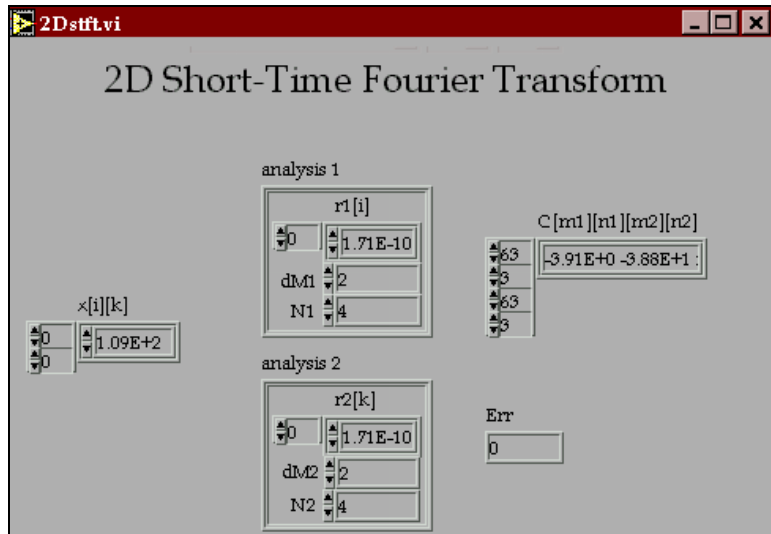


Figure 5-4. 2D STFT for the Image Analysis in Figure 5-2

Time-Dependent Spectrum Analysis Examples

A primary motivation of JTFA is to discover how the power spectrum of a signal changes over time. While classical algorithms such as the square of the Fourier transform indicate only the average of a signal's power spectrum, JTFA algorithms allow you to examine the instantaneous spectrum. Consequently, you have a better understanding of the nature of the signal in which you are interested.

For your convenience, this package provides a comprehensive combination of online and offline joint time-frequency analyzers. Using these analyzers, you can perform rather sophisticated time-dependent spectrum analysis. Because each algorithm has advantages as well as disadvantages, you should select an algorithm based on the application. The simplest and fastest algorithm is the STFT spectrogram, which is suitable for online analysis. If the frequency contents of a signal change rapidly, consider one of the other algorithms included in this toolkit.

Online STFT Spectrogram Analyzer

The **Online Analyzer** allows you to collect real data and perform online analysis. Figure 5-5 illustrates the front panel of the **Online Analyzer**. The bottom plot displays the time waveform. The top plot displays the corresponding STFT spectrogram. The following sections describe how to manipulate controls and read indicators on the **Online Analyzer** panel.

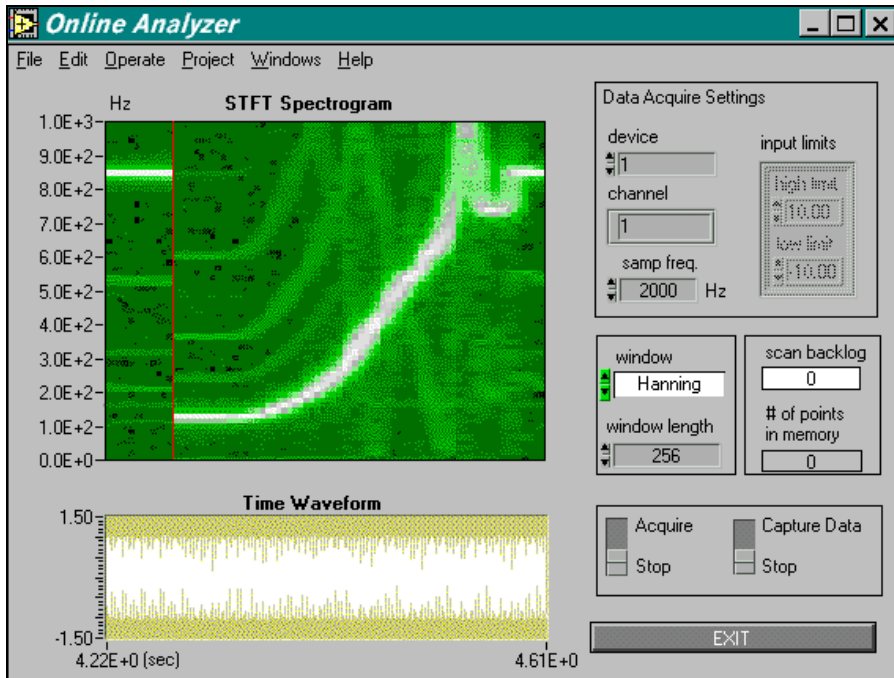
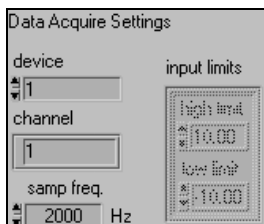


Figure 5-5. Online STFT Spectrogram Analyzer Panel

Setting NI DAQ



To properly perform online analysis, the **Online Analyzer** needs to know the **device** number, **channel** number, and **input limits**. Choose a sampling frequency (**samp freq.**) based on the application. The maximum **samp freq.** depends on the DAQ card as well as the computer that you use. **scan backlog** (see Figure 5-5) indicates whether **samp freq.** is adequate. If the **scan backlog** value keeps increasing, reduce **samp freq.**. For more information on DAQ settings, consult your NI-DAQ manual.

Setting the Analysis Window



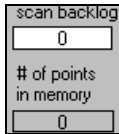
You can choose one of four types of windows: **rectangular**, **Blackman**, **Hamming**, or **Hanning**. The maximum **window length** is 512.

Acquiring Data



To start collecting data with NI DAQ, set the left switch to **Acquire**. **scan backlog** (see Figure 5-5) indicates whether the system can keep up with the incoming samples. You must move the right switch to **Capture Data** to preserve the acquired data.

Saving Data



To save data, you need to move the right switch to **Capture Data**. The indicator **# of points in memory** displays the number of data points in memory. When you toggle **Capture Data** to **Stop**, **# of points in memory** is replaced by the indicator **total # of points saved**, which displays the total number data points saved in memory. The captured data remains in temporary memory until you stop acquiring data. When you finish collecting data, a dialog box prompts you to save the data to a text file. If you select **Discard**, the memory is cleared, and the data is lost. If you select **Yes**, the data in memory is saved to a text file designated by you.

Offline Joint Time-Frequency Analyzer

If the signal's frequency contents change rapidly, the STFT spectrogram is not adequate, and you need an **Offline Analyzer**, such as the one shown in Figure 5-6. The bottom plot illustrates the time waveform of the analyzed signal. The right plot displays the classical power spectrum or instantaneous spectrum. The upper-left plot shows the spectrogram (time-dependent spectrum).

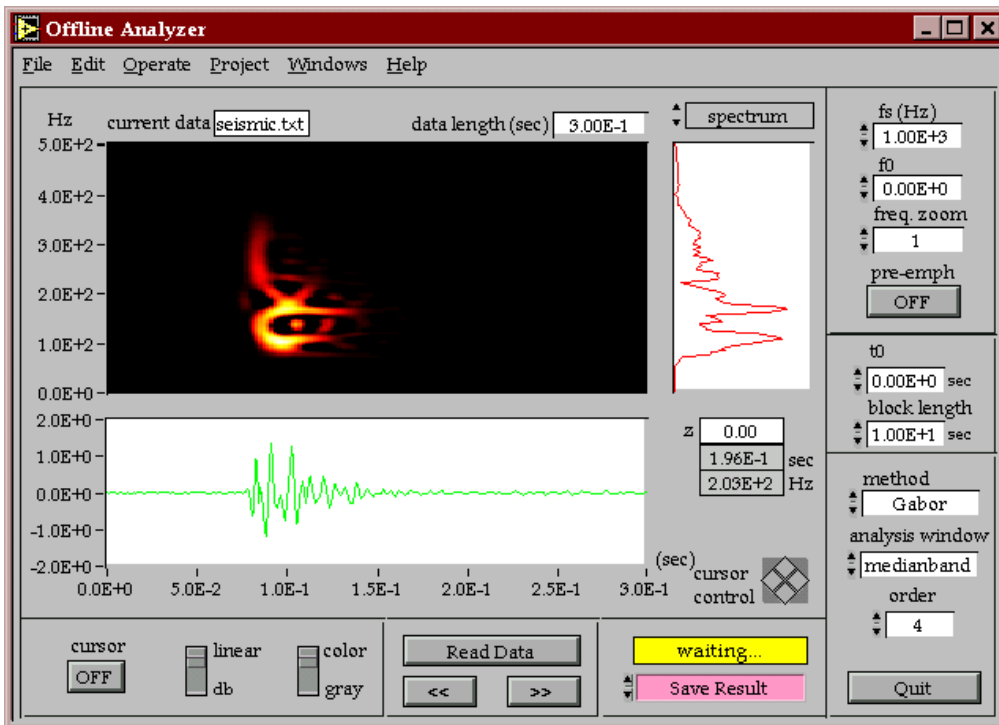


Figure 5-6. Offline Joint Time-Frequency Analyzer

Because it demonstrates each individual quadratic JTFA algorithm included in this toolkit, the **Offline Analyzer** helps you select the algorithm that best fits your application.

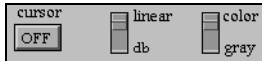


Note

This instrument was designed for demonstration purposes only. For most real applications, you need to build your own JTFA instrument using the VIs included in this toolkit.

The following sections explain how to manipulate controls and read indicators in the **Offline Analyzer**.

Changing Spectrogram Display



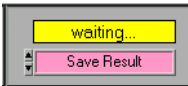
There are three options for the spectrogram display. By pressing the **cursor** button, you turn the cursor on and off. By flipping the **linear-db** switch, you select linear or db display. By flipping the **color-gray** switch, you control the color table of the spectrogram.

Inputting Data



Press the **Read Data** button to select the data file that you want to analyze. The data file must be a one-dimensional text file. If the file contains x-index, remove it with any word processor before analyzing the data. The buttons << and >> move you to the previous or next block of data, respectively.

Saving Results



By selecting the **Save Result** button, you can save any displayed data, such as the **time waveform**, **power spectrum**, and **spectrogram**, as a text file.

All spectrograms display only the non-negative points. The **Offline Analyzer** automatically truncates negative points to zero. If you use log scale, the displayed spectrogram is further normalized. However, real spectrograms, except for the STFT and the adaptive spectrograms, might contain negative values. **Save Result** saves the real spectrogram without truncating or normalizing.

Switching Between Conventional Power and Instantaneous Spectrum



Press the **spectrum** button to switch the spectrum display between a conventional power spectrum and an instantaneous spectrum. When you select **instant spectrum**, as shown in Figure 5-7, the cursor controls the time instant. A group of indicators below the spectrum display the cursor position.

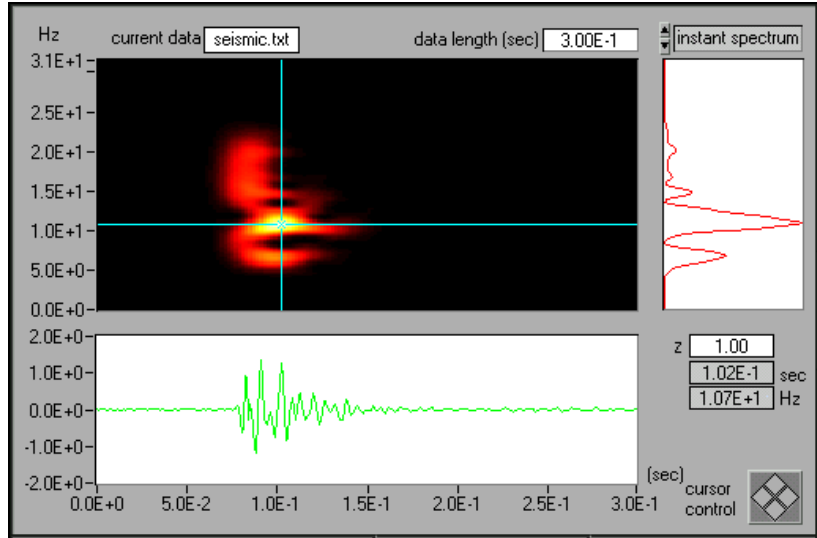
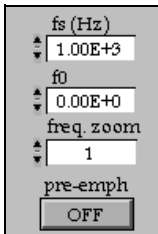


Figure 5-7. Instantaneous Spectrum Display

Frequency Zooming

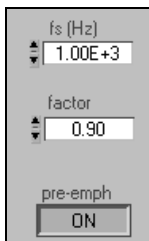
The control **fs** determines the frequency range to display. The highest frequency is equal to $fs/2$. By using the **freq. zoom** control, you can examine the signal in the frequency domain. The frequency range displayed is equal to $fs/(2 * \text{zoom factor})$. The maximum zoom factor is limited to 16, so the smallest frequency range is $fs/32$.

The **f0** parameter determines the start frequency to be displayed and must be greater than or equal to zero. Moreover, $f0 + fs/(2 * \text{zoom factor}) < fs/2$. If **f0** is out of the valid range, the selection is ignored.

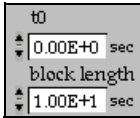


Applying the Pre-Emphasis Filter

Press the **pre-emph** button to apply the pre-emphasis filter to the import signal. The pre-emphasis filter suppresses DC and enhances high frequency components. The larger the control **factor**, the less the DC remaining. The control **factor** ranges from 0 to 1.



Setting Time Parameters



Time parameters allow you to process the part of signal in which you are most interested. **t0** controls the start point of the analyzed signal. If **t0** is out of range, it reverts to zero. The control **block length** determines the length of the signal to be processed. If **block length** is larger than the signal length, it is ignored.

Selecting the JTFA Method

The **Offline Analyzer** contains the following quadratic JTFA algorithms. Use the list box **method** to specify any of them.

- adaptive spectrogram
- Cohen's class
- CWD (Choi–Williams distribution)
- cone-shaped distribution
- PWVD (Pseudo Wigner–Ville distribution)
- Gabor spectrogram
- STFT spectrogram



Note

Generally, you should start with the STFT spectrogram because it is fast and simple.

STFT Spectrogram

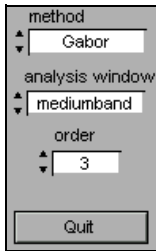


To run the STFT spectrogram, you need to input values for **window selector** and **window length**. The **Offline Analyzer** provides four window types: **rectangular**, **Blackman**, **Hamming**, and **Hanning**. **Window length** must be less than 256. If you enter a length greater than 256, the length automatically truncates to 256. The longer the window, the higher the frequency resolution but the poorer the time resolution and vice versa. Hence, we traditionally consider the long window as narrowband and the short window as wideband.

Adjust the window length and type such that the resulting STFT spectrogram achieves the best compromise between time and frequency resolution. You can use the resulting **window length** as a reference for the Gabor spectrogram (discussed in the next section).

If you cannot achieve satisfactory resolution with the STFT spectrogram, try the Gabor spectrogram or one of the other methods explained later in this chapter.

Gabor Spectrogram



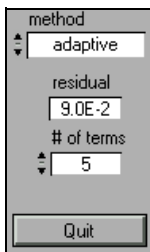
If the time-frequency resolution of the STFT spectrogram is not satisfactory, try the Gabor spectrogram next. This method requires more computation time than the STFT spectrogram, but the Gabor spectrogram has better time-frequency resolution.

To process the Gabor spectrogram, you need to set the **order** and **analysis window** parameters. **order** controls the resolution and crossterm interference. The higher the order, the better the time-frequency resolution. As the order goes to infinity, the Gabor spectrogram converges on the Wigner–Ville distribution. Notice that as the order increases, crossterms become more obvious. Furthermore, computation time is proportional to the order selected. Set **order** from 3 to 5 to achieve the best compromise between resolution and crossterm interference.

In general, the Gabor spectrogram is less sensitive to window length than the STFT spectrogram. Hence, there are only three analysis window selections: **wideband**, **mediumband**, and **narrowband**. If you adjusted the window length using the STFT method, use that value to determine which analysis window you should select for the Gabor spectrogram.

STFT Spectrogram Window Length	Analysis Window for Gabor Spectrogram
1 ~ 32	wideband
33 ~ 96	mediumband
96 ~ 256	narrowband

Adaptive Spectrogram



If you can consider a signal as the sum of linear chirp functions (with different Gaussian envelopes), use the adaptive spectrogram to achieve the best time-frequency resolution.

of terms determines the number of linear chirp functions used to approximate the analyzed signal. The more elementary linear chirp functions, the more accurate the approximation and the smaller the **residual**. Unfortunately, the more elementary functions you specify, the longer the computation time. Usually, you should start with a small number

of terms. Increase **# of terms** until **residual** is satisfied. The residual is computed as

$$residual = \frac{\sum_n |s[n] - a[n]|^2}{\sum_n |s[n]|^2}$$

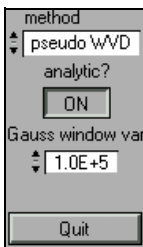
where $a[n]$ denotes the approximation. If the approximation is equal to the original signal $s[n]$, **residual** reduces to zero.

Pseudo Wigner–Ville Distribution

The pseudo Wigner–Ville distribution is fast and provides high time-frequency resolution. However, it may suffer from serious crossterm interference if the analyzed signal consists of multiple components.

You can lessen the crossterm interference two ways. First, you can take the pseudo Wigner–Ville distribution with respect to the analytical function (set **analytic?** to **ON**). However, this approach destroys the low frequency portion of the signal’s time-dependent spectrum.

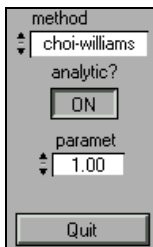
Second, you can reduce **Gauss window var** to eliminate the crossterm caused by a pair of autoterms separated in time. However, reducing **Gauss window var** deteriorates time-frequency resolution.



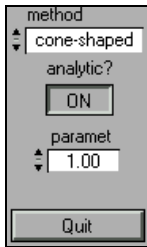
Choi–Williams Distribution

The Choi–Williams distribution is designed to reduce crossterm interference while preserving as many useful Wigner–Ville distribution properties as possible. Like the pseudo Wigner–Ville distribution, you can take the Choi–Williams distribution with respect to the analytical function (set **analytic?** to **ON**). The resulting spectrogram has reduced crossterm interference.

You can also lessen crossterm interference by setting the control **paramet**. In general, the smaller the **paramet** value, the less crossterm interference but the poorer the time-frequency resolution. **paramet** defaults to a value of 1.



Cone-Shaped Distribution



The cone-shaped distribution is another type of time-dependent spectrum designed to reduce crossterm interference. Like the pseudo Wigner–Ville distribution and the Choi–Williams distribution, you can also take the cone-shaped distribution with respect to the analytical function (set **analytic?** to **ON**). The resulting spectrogram has reduced crossterm interference.

You can also lessen crossterm interference by setting the control **paramet**. In general, the smaller the **paramet** value, the less crossterm interference but the poorer the time-frequency resolution. **paramet** defaults to a value of 1.

Frequently Asked Questions

This chapter addresses some questions frequently asked by JTFA users.

What Is the Difference between Linear and Quadratic JTFA Methods?

This package includes both linear and quadratic methods. Linear transforms include the adaptive representation, adaptive transform (for computing the adaptive representation coefficients), Gabor expansion, and STFT (short-time Fourier transform used to compute the Gabor coefficients). Quadratic methods include the adaptive spectrogram, Cohen's class, Choi-Williams Distribution (CWD), cone-shaped distribution, Pseudo Wigner-Ville Distribution (PWVD), Gabor spectrogram (Gabor expansion based spectrogram), and STFT spectrogram.

If we consider the linear JTFA as the evolution of the conventional Fourier transform, the quadratic JTFA is the counterpart of the standard power spectrum. The difference between linear and quadratic methods is that the linear transform is invertible. As with the fast-Fourier transform, you can reconstruct the original signal based on the Gabor coefficients. Thus, the linear transform is suitable for signal processing, such as time-varying filtering. In general, the quadratic form is not reversible. You cannot restore the original time waveform from the time-dependent spectrum. However, the quadratic JTFA describes the energy distribution of the signal in the joint time-frequency domain, which is useful for signal analysis.

Which Quadratic JTFA Algorithms Should I Use?

Each quadratic JTFA algorithm has its own advantages and disadvantages. You should select a particular algorithm based on the application. Generally speaking, no algorithm is superior to all others in all applications.

Table 6-1. Quadratic JTFA Algorithms

Method	Resolution and Crossterm Description	Speed
adaptive spectrogram	extremely high resolution when a signal is made up of linear chirps no crossterms non-negative	slow
Choi–Williams distribution (CWD)	less crossterm than PWVD	very slow
cone-shaped distribution	less crossterm interference than PWVD or CWD	slow
Gabor spectrogram	good resolution robust minor crossterms	moderate
pseudo Wigner–Ville distribution (PWVD)	extremely high resolution for a few types of signals severe crossterms	fast
STFT spectrogram	poor resolution robust non-negative	fast

If the frequency contents of the analyzed signal do not change rapidly, try the STFT spectrogram first. You can apply a relatively long window function to obtain a good frequency resolution with tolerable time resolution deterioration. Because the STFT spectrogram is fast, it is suitable for online analysis.

The other algorithms generally have better joint time and frequency resolution than the STFT spectrogram, but they require more computation time, which is suitable for offline analysis only. Table 6-1 summarizes the advantages and disadvantages of all quadratic algorithms provided in this package. If you need a higher resolution, use the third or fourth order Gabor spectrogram to reduce crossterm interference and achieve faster processing speeds.

We recommend that you use the following procedure when analyzing a signal.

1. Begin with the STFT and determine which analysis window is best: wide, median, or narrow band.
2. Are you satisfied with the result? If you are satisfied, use STFT. If not, continue with step 3.
3. If the STFT does not produce satisfactory results, try the Gabor spectrogram. Independent of the analysis window used, the Gabor spectrogram converges to the Winger-Ville distribution as the order increases. If the order is low, the type of analysis window influences the Gabor spectrogram (although the effect is not as large as that on the STFT). Select your Gabor spectrogram **analysis window** based on the window information obtained in the first step.
4. Increase the **order** until the crossterm interference becomes obvious. For most applications, an order of three to five should be adequate.
5. To examine detailed features, reduce the data **block length** and increase the **freq. zoom**.

Can I Measure a Signal's Energy in the Joint Time-Frequency Domain Point-to-Point?

This question addresses a fundamental issue in the joint time-frequency analysis area. Except for a few special cases, the answer is no. As far as we are aware, no algorithm can meaningfully measure a signal's energy point-to-point in the joint time-frequency domain.

Roughly speaking, the result $P(t,w)$ of all quadratic JTFA algorithms indicates a certain type of weighted average energy near the point (t,f) . Some algorithms take an average over a larger area, such as the STFT spectrogram. In this case, the time-frequency resolution is poor, but it is always greater than or equal to zero. Some methods cause heavy weights on a small number of points, such as the high-order Gabor spectrogram, which yields better time-frequency resolution. In this case, some points may approach negativity, which is not acceptable for certain applications. In short, every algorithm has advantages and disadvantages.

As an example, Figure 6-1 shows an STFT spectrogram with a test signal containing 10 sine cycles at 10 Hz. Although the signal starts at $t = 1$ s and ends at $t = 2$ s, the STFT spectrogram clearly shows something before $t = 1$ s and after $t = 2$ s, as indicated by the arrows. Moreover, the time-dependent spectrum indicates that the signal does not contain only 10 Hz but that it possesses a certain bandwidth.

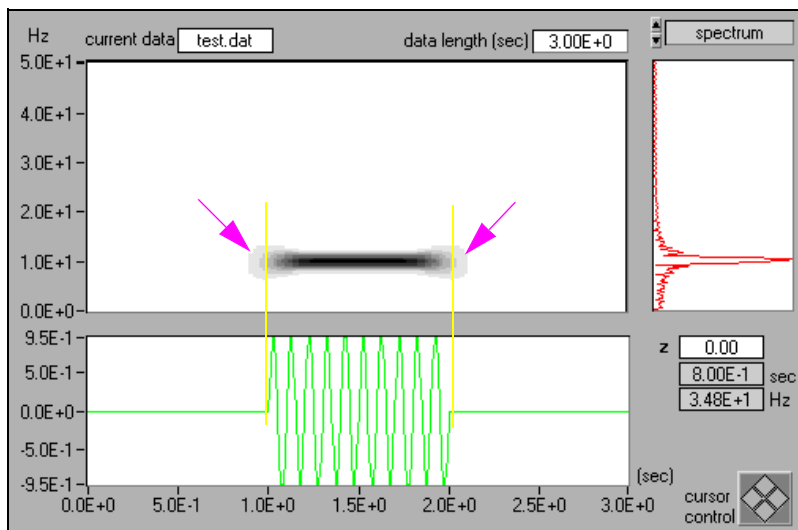


Figure 6-1. STFT Spectrogram (Hanning Window)

By applying other methods, you can substantially suppress the energy outside 1 to 2 s and 10 Hz, thereby achieving a near point-to-point measurement. In Figure 6-2, we used the Gabor spectrogram. As shown, most of the signal's energy is between 1 to 2 s and 10 Hz. The higher the order, the higher the concentration and the closer we get to achieving a point-to-point measurement. On the other hand, the higher order Gabor spectrogram produces negative values, which might be difficult to accept from the classical energy definition point of view. Moreover, the Gabor spectrogram generally requires more computation time than the STFT spectrogram.

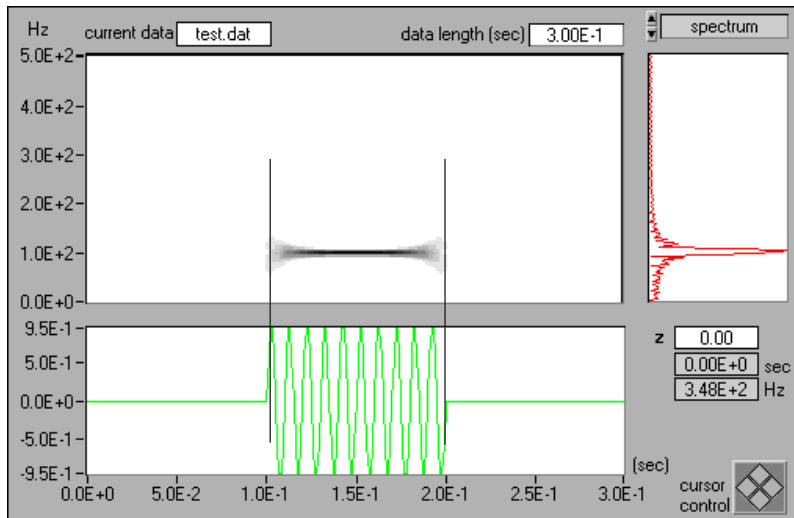


Figure 6-2. Gabor Spectrogram (Order Four)

What Can I Do If the Time-dependent Spectrum Shows a Line Only at DC?

If the analyzed signal is non-negative, such as an ECG or the stock index, or if it contains a large DC offset, the resulting time-dependent spectrum is dominated by a single line in the vicinity of DC, and you might not be able to see more interesting frequency patterns. To suppress the DC component, you have to apply certain types of preprocessing. However, the methods for removing the DC components (or detrending) are application dependent. No general method works in all cases. Common techniques of detrending include lowpass filtering and curve fitting; however, a more promising technique is the wavelet transform. Refer to the WFBD (Wavelet and Filter Bank Design) manual for information on wavelet-based detrending.

Can I Use Other Software to Plot the Time-Dependent Spectrum?

Yes. Save the time-dependent spectrum to a text file. The resulting text file contains only *Z* values and does not retain the time- and frequency-axis information. The time and frequency axis can be determined as follows.

While **t0** and **f0** are shown on the front panel of **Offline Analyzer**, the time increment Δt is computed by

$$\Delta t = \frac{\text{time span}}{\text{number of rows}}$$

and the frequency increment Δf is determined by

$$\Delta f = \frac{\text{sampling frequency}}{2 \times \text{zoom factor} \times 128}$$

References

This appendix lists reference material that contains more information on the theory and algorithms implemented in the JTFA VIs.

Choi, H. and W. J. Williams. 1989. Improved time-frequency representation of multicomponent signals using exponential kernels. *IEEE Trans. Acoustics, Speech, Signal Processing* vol. 37.6:862–871.

Cohen, L. 1966. Generalized phase-space distribution functions. *J. Math. Phys.* 7:781–806.

———. 1989. Time-frequency distribution: A review. *Proc. IEEE* vol. 77.7:941–981.

———. 1995. *Time-frequency analysis*. Englewood Cliffs, NJ: Prentice Hall.

Qian, S. and J. M. Morris. 1992. Wigner distribution decomposition and crossterm deleted representation. *Signal Processing* vol. 25. 2 (May):125–144.

Qian, S. and D. Chen. 1993. Discrete Gabor transform. *IEEE Trans. Signal Processing* vol. 41.7:2429–2439.

———. 1994. Decomposition of the Wigner-Ville distribution and time-frequency distribution series. *IEEE Trans. Signal Processing* vol. 42.10:2836–2841.

———. 1996. *Joint time-frequency analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Wexler, J. and S. Raz. 1990. Discrete Gabor expansions. *Signal Processing* vol. 21. 3 (November):207–221.

Yin, Q., Z. Ni, S. Qian, and D. Chen. 1997. Adaptive oriented orthogonal projective decomposition. *Journal of Electronics (Chinese)* vol. 25.4:52–58.

Xia, X. G. and S. Qian. An iterative algorithm for time-varying filtering in the discrete Gabor transform domain. Appearing on *IEEE Trans. on Circuit and Systems*.

Zhao, Y., L. E. Atlas, and R. J. Marks. 1990. The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals. *IEEE Trans. Acoustics, Speech, Signal Processing* vol. 38.7:1084–1091.

Error Codes

This appendix lists the error codes returned by the joint time-frequency analysis VIs.

Table B-1. Error Codes

Code	Error Name	Description
-2080	GabordMErr	The Gabor time sampling interval is not evenly divided by the window length. The oversampling rate N/dM is less than one.
-2081	GaborNErr	The number of frequency bins is not a power of two. The number of frequency bins is not evenly divided by the window length. The oversampling rate N/dM is less than one.
-2082	GaborOversamplErr	The oversampling rate N/dM is less than one.
-2083	JTFANoSolutionErr	The dual function does not exist.
-2084	JTFWindowErr	The window length is not an integral multiple of the number of frequency bins N .
-2085	JTFAParametErr	The order of the Gabor spectrogram is less than zero. The number of terms of the adaptive transform is less than zero.
-2086	JTFATimeSamplErr	The Gabor coefficient array is empty.
-2087	JTFAHilbertErr	The sample length is not equal to the length of the corresponding Hilbert transform.
-2088	JTFADecimatErr	The time decimation is greater than the sample length.
2001	RankDeficient	The matrix is near ill condition.



Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation. When you contact us, we need the information on the Technical Support Form and the configuration form, if your manual contains one, about your system configuration to answer your questions as quickly as possible.

National Instruments has technical assistance through electronic, fax, and telephone systems to quickly provide the information you need. Our electronic services include a bulletin board service, an FTP site, a fax-on-demand system, and e-mail support. If you have a hardware or software problem, first try the electronic support systems. If the information available on these systems does not answer your questions, we offer fax and telephone support through our technical support centers, which are staffed by applications engineers.

Electronic Services

Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call 512 795 6990. You can access these services at:

United States: 512 794 5422

Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422

Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 01 48 65 15 59

Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

FTP Support

To access our FTP site, log on to our Internet host, `ftp.natinst.com`, as anonymous and use your Internet address, such as `joesmith@anywhere.com`, as your password. The support files and documents are located in the `/support` directories.

Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at 512 418 1111.

E-Mail Support (Currently USA Only)

You can submit technical support questions to the applications engineering team through e-mail at the Internet address listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

support@natinst.com

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the list below to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.

Country	Telephone	Fax
Australia	03 9879 5166	03 9879 6277
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Brazil	011 288 3336	011 288 8528
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Québec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	01 48 14 24 24	01 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Israel	03 6120092	03 6120095
Italy	02 413091	02 41309215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	5 520 2635	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
United Kingdom	01635 523545	01635 523154
United States	512 795 8248	512 794 5678

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

Fax (____) _____ Phone (____) _____

Computer brand _____ Model _____ Processor _____

Operating system (include version number) _____

Clock speed _____ MHz RAM _____ MB Display adapter _____

Mouse ____ yes ____ no Other adapters installed _____

Hard disk capacity _____ MB Brand _____

Instruments used _____

National Instruments hardware product model _____ Revision _____

Configuration _____

National Instruments software product _____ Version _____

Configuration _____

The problem is: _____

List any error messages: _____

The following steps reproduce the problem: _____

LabVIEW Joint Time-Frequency Analysis Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

DAQ hardware _____

Interrupt level of hardware _____

DMA channels of hardware _____

Base I/O address of hardware _____

Programming choice _____

LabVIEW version _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Other Products

Computer make and model _____

Microprocessor _____

Clock frequency or speed _____

Type of video board installed _____

Operating system version _____

Operating system mode _____

Programming language _____

Programming language version _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: *LabVIEW™ Joint Time-Frequency Analysis Toolkit Reference Manual*

Edition Date: February 1998

Part Number: 320544D-01

Please comment on the completeness, clarity, and organization of the manual.

If you find errors in the manual, please record the page numbers and describe the errors.

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

E-Mail Address _____

Phone (____) _____ Fax (____) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway
Austin, Texas 78730-5039

Fax to: Technical Publications
National Instruments Corporation
512 794 5678

Glossary

Prefix	Meanings	Value
n-	nano-	10^{-9}
μ -	micro-	10^{-6}
m-	milli-	10^{-3}

Numbers/Symbols

°	Degrees.
%	Percent.
∞	Infinity.
π	Pi.
1D	One-dimensional.
2D	Two-dimensional.

A

A/D	Analog/digital.
ANSI	American National Standards Institute.
array	Ordered, indexed set of data elements of the same type.
ASCII	American Standard Code for Information Interchange.

B

basis	A core or fundamental function.
block diagram	Pictorial description or representation of a program or algorithm. In G, the block diagram is the source code for the VI. It consists of executable icons called nodes as well as wires that carry data between the nodes.
Boolean controls and indicators	Front panel objects used to manipulate and display Boolean (TRUE or FALSE) data. Several styles are available, such as switches, buttons, and LEDs.

C

chart	<i>See</i> scope chart, strip chart, and sweep chart.
check box	Small square box in a dialog box that can be selected or cleared. Check boxes generally are associated with multiple options that can be set. More than one check box can be selected.
CIN	<i>See</i> code interface node.
cluster	Set of ordered, unindexed data elements of any data type including numeric, Boolean, string, array, or cluster. The elements must be all controls or all indicators.
code interface node	CIN. Special block diagram node through which you can link conventional, text-based code to a VI.
continuous run	Execution mode in which a VI is run repeatedly until the operator stops it. You enable it by clicking the Continuous Run button.
control	Front panel object for entering data to a VI interactively or to a subVI programmatically.
CWD	Choi–Williams Distribution.

D

DAQ	<i>See</i> data acquisition.
data acquisition	DAQ. Process of acquiring data, typically from A/D or digital input plug-in boards.
data type	Format for information. In BridgeVIEW, acceptable data types for tag configuration are analog, discrete, bit array, and string. In LabVIEW, acceptable data types for most functions are numeric, array, string, and cluster.
datalog file	File that stores data as a sequence of records of a single, arbitrary data type that you specify when you create the file. While all the records in a datalog file must be of a single type, that type can be complex; for instance, you can specify that each record is a cluster containing a string, a number, and an array.
developer	<i>See</i> system developer.
DFT	Discrete Fourier Transform.
DGT	Discrete Gabor Transform.
dialog box	Window that appears when an application needs further information to carry out a command.

E

error message	Indication of a software or hardware malfunction, or an unacceptable data entry attempt.
---------------	--

F

FFT	Fast Fourier transform.
formula node	Node that executes formulas you enter as text. Formula nodes are especially useful for lengthy formulas that would be cumbersome to build in block diagram form.
frame	Segment of time domain data.

front panel Interactive user interface of a VI. Modeled after the front panel of physical instruments, it is composed of switches, slides, meters, graphs, charts, gauges, LEDs, or other controls or indicators.

function Built-in execution element, comparable to an operator, function, or statement in a conventional language.

G

G Graphical programming language used to develop LabVIEW and BridgeVIEW applications.

global variable Non-reentrant subVI with local memory that uses an uninitialized shift register to store data from one execution to the next. The memory of copies of these subVIs is shared and thus can be used to pass global data between them.

GPIO General Purpose Interface Bus. Common name for the communications interface system defined in ANSI/IEEE Standard 488.1-1987 and ANSI/IEEE Standard 488.2-1987.

H

Help Online instructions that explain how to use a Windows application. The **Help** menu displays specific Help topics. Pressing <F1> displays a list of Help topics.

Help window Special window that displays the names and locations of the terminals for a function or subVI, the description of controls and indicators, the values of universal constants, and descriptions and data types of control attributes.

Hz Hertz. Cycles per second.

I

I/O Input/output. Transfer of data to or from a computer system involving communications channels, operator input devices, and/or data acquisition and control interfaces.

icon Graphical representation of a node on a block diagram.

IEEE	Institute for Electrical and Electronic Engineers.
indicator	Front panel object that displays output.
J	
JTFA	Joint Time-Frequency Analysis.
L	
LabVIEW	Laboratory Virtual Instrument Engineering Workbench. Program development application based on the programming language G used commonly for test and measurement purposes.
local variable	Variable that enables you to read or write to one of the controls or indicators on the front panel of your VI.
M	
matrix	Two-dimensional array.
MB	Megabytes of memory.
multicomponent signal	A signal containing significant energy concentrated around more than one distinct and separate frequency.
N	
node	Program execution element. Nodes are analogous to statements, operators, functions, and subroutines in conventional programming languages. In a block diagram, nodes include functions, structures, and subVIs.
nonstationary signal	Signal whose frequency content changes within a captured frame.
numeric controls and indicators	Front panel objects used to manipulate and display or input and output numeric data.

O

object	Generic term for any item on the front panel or block diagram, including controls, nodes, wires, and imported pictures.
oversampling rate	Ratio between the number of Gabor coefficients and the number of test samples.

P

plot	Graphical representation of an array of data shown either in a graph or a chart.
pop up	To call a special menu by right-clicking (Windows) or command-clicking (Macintosh) an object.
pop-up menu	Menu accessed by right-clicking (Windows) or command-clicking (Macintosh) an object. Menu options pertain to that object.
preemphasis	Filtering before processing.
PWVD	Pseudo Wigner–Ville Distribution.

R

reentrant execution	Mode in which calls to multiple instances of a subVI can execute in parallel with distinct and separate data storage.
representation	Subtype of the numeric data type. Representations include signed and unsigned byte, word, and long integers, as well as single-, double-, and extended-precision floating-point numbers, both real and complex.

S

sampling rate	Rate at which a continuous waveform is digitized.
scope chart	Numeric indicator modeled on the operation of an oscilloscope.
spectral changes	Changes in the frequency content of a signal.

spectrogram	A display of the energy distribution of a signal with one axis being time and the other being frequency.
STFT	Short-time Fourier transform.
string controls and indicators	Front panel objects used to manipulate and display or input and output text.
strip chart	Numeric plotting indicator modeled after a paper strip chart recorder, which scrolls as it plots data.
subVI	VI used in the block diagram of another VI. It is comparable to a subroutine.
sweep chart	Numeric indicator modeled on the operation of an oscilloscope. It is similar to a scope chart, except that a line sweeps across the display to separate old data from new data.
system developer	Creator of the application software to be executed.

T

temporal	Of or relating to time domain.
two-dimensional	Having two dimensions, such as an array with both rows and columns.

V

VI	<i>See</i> virtual instrument.
VI library	Special file that contains a collection of related VIs for a specific use.
virtual instrument	VI. Program in the graphical programming language G; so-called because it models the appearance and function of a physical instrument.
Virtual Instrument Software Architecture	Single interface library for controlling GPIB, VXI, RS-232, and other types of instruments.

W

waveform chart Indicator that plots data points at a certain rate.

WVD Wigner–Ville Distribution.

Index

A

- adaptive representation, 3-2
- adaptive spectrogram, 2-7, 3-13, 4-9, 6-2
 - demonstrating with Offline Joint Time-Frequency Analyzer, 5-12 to 5-13
- adaptive transform, 3-2, 4-1 to 4-2
- algorithms, 3-1 to 3-13
 - adaptive representation, 3-2
 - adaptive spectrogram, 3-13
 - adaptive transform, 3-2
 - Choi–Williams distribution (CWD), 3-9
 - Cohen’s class, 3-8
 - cone-shaped distribution, 3-10
 - Gabor expansion, 3-1 to 3-2
 - Gabor spectrogram, 3-11 to 3-12
 - linear vs. quadratic, 6-1
 - point-to-point measurement of a signal, 6-4 to 6-5
 - pseudo Wigner–Ville distribution (PWVD), 3-5 to 3-8
 - selecting, 6-2 to 6-3
 - STFT, 3-1 to 3-2
 - STFT spectrogram, 3-3 to 3-4
 - Wigner–Ville distribution (WVD), 3-5 to 3-8
- applications, 5-1 to 5-14
 - denoise, 5-1 to 5-3
 - image analysis, 5-3 to 5-5
 - linear algorithm application examples, 5-1 to 5-5
 - Offline Joint Time-Frequency Analyzer, 5-8 to 5-14
 - applying pre-emphasis filter, 5-10
 - changing spectrogram display, 5-9
 - frequency zooming, 5-10
 - inputting data, 5-9
 - saving results, 5-9

- selecting a JTFA method, 5-11 to 5-14
 - setting time parameters, 5-11
 - switching between conventional power and instantaneous spectrum, 5-9 to 5-10
- Online STFT Spectrogram Analyzer, 5-6 to 5-7
 - acquiring data, 5-7
 - saving data, 5-7
 - setting analysis window, 5-7
 - setting NI DAQ, 5-6
 - time-dependent spectrum analysis examples, 5-5 to 5-14
- approaches to JTFA, 2-6 to 2-7
- Atlas, L. E., A-2

B

- bibliography, A-1
- bulletin board support, C-1

C

- Chen, D., A-1
- Choi, H., A-1
- Choi–Williams distribution (CWD), 3-9, 4-10 to 4-11, 6-2
 - demonstrating with Offline Joint Time-Frequency Analyzer, 5-13
- Classical Fourier Transform
 - See* Fourier Transform.
- Cohen, L., A-1
- Cohen, Leon, 3-8
- Cohen’s class, 2-6, 3-8, 4-10
- cone-shaped distribution, 3-10, 4-11 to 4-12, 6-2
 - demonstrating with Offline Joint Time-Frequency Analyzer, 5-14

crossterm interference, 2-6, 3-5 to 3-8,
3-9, 3-11
customer communication, xi, C-1 to C-7

D

DC component, 6-5
denoise, 5-1 to 5-3
documentation
 conventions used in this manual, x
 organization of this manual, ix
 related documentation, x

E

electronic support services, C-1 to C-2
error codes, B-1

F

Fast Dual, 4-7
Fourier, Jean Baptiste Joseph, 2-1
Fourier transform
 review of, 2-1 to 2-3
frequently asked questions, 6-1 to 6-6
FTP support, C-1

G

Gabor, Dennis, 2-6
Gabor expansion, 2-6, 3-1 to 3-2, 4-4
Gabor expansion, 2D, 4-6 to 4-7
Gabor spectrogram, 2-6, 3-11 to 3-12,
4-14, 6-2
 demonstrating with Offline Joint
 Time-Frequency Analyzer, 5-12
getting help, 1-2

H

help, 1-2

I

image analysis, 5-3 to 5-5
installation, 1-1 to 1-2
 Power Macintosh, 1-2
 Windows 3.1, 1-2
 Windows 95 and Windows NT, 1-1
inverse adaptive transform, 4-2 to 4-3

J

Joint Time-Frequency Analysis

See JTFA.

JTFA

 algorithms, 3-1 to 3-13
 adaptive representation, 3-2
 adaptive spectrogram, 3-13
 adaptive transform, 3-2
 Choi–Williams distribution
 (CWD), 3-9
 Cohen’s class, 3-8
 cone-shaped distribution, 3-10
 Gabor expansion, 3-1 to 3-2
 Gabor spectrogram, 3-11 to 3-12
 linear, 3-1 to 3-2
 pseudo Wigner–Ville distribution
 (PWVD), 3-5 to 3-8
 quadratic, 3-3 to 3-13, 6-2
 STFT, 3-1 to 3-2
 STFT spectrogram, 3-3 to 3-4
 Wigner–Ville distribution (WVD),
 3-5 to 3-8
 applications, 5-1 to 5-14
 denoise, 5-1 to 5-3
 image analysis, 5-3 to 5-5

Offline Joint Time-Frequency Analyzer, 5-8 to 5-14
 Online STFT Spectrogram Analyzer, 5-6 to 5-7
 approaches to, 2-6 to 2-7
 need for, 2-4 to 2-5
 VIs, 4-1 to 4-16
 2D Gabor Expansion, 4-6 to 4-7
 2D STFT, 4-4 to 4-5
 adaptive spectrogram, 4-9
 adaptive transform, 4-1 to 4-2
 Choi–Williams distribution (CWD), 4-10
 Cohen’s class, 4-10
 cone-shaped distribution, 4-11 to 4-12
 Fast Dual, 4-7
 Gabor expansion, 4-4
 Gabor spectrogram, 4-14
 inverse adaptive transform, 4-2
 Normalized Gaussian Window Function, 4-8
 pseudo Wigner–Ville distribution (PWVD), 4-13
 short-time Fourier transform (STFT), 4-3
 STFT spectrogram, 4-12
 Time-Frequency Distribution Series, 4-15 to 4-16
 JTFA toolkit, 1-1

L

linear algorithm application examples, 5-1 to 5-5
 linear JTFA, 2-6
 linear JTFA algorithms, 3-1 to 3-2
 linear vs. quadratic methods, 6-1

M

manual
 See documentation.
 Marks, R. J., A-2
 measuring signal energy, 6-4 to 6-5
 Morris, J. M., A-1

N

noise reduction, 5-1 to 5-3
 Normalized Gaussian Window Function, 4-8

O

Offline Joint Time-Frequency Analyzer, 5-8 to 5-14
 applying pre-emphasis filter, 5-10
 changing spectrogram display, 5-9
 frequency zooming, 5-10
 inputting data, 5-9
 saving results, 5-9
 selecting the JTFA method, 5-11 to 5-14
 adaptive spectrogram, 5-12 to 5-13
 Choi–Williams distribution, 5-13
 cone-shaped distribution, 5-14
 Gabor spectrogram, 5-12
 pseudo Wigner–Ville distribution, 5-13
 STFT spectrogram, 5-11
 setting time parameters, 5-11
 switching between conventional power and instantaneous spectrum, 5-9 to 5-10
 Online STFT Spectrogram Analyzer, 5-6 to 5-7
 acquiring data, 5-7
 saving data, 5-7
 setting analysis window, 5-7
 setting NI DAQ, 5-6

P

- package contents, 1-1
- point-to-point measurement, 6-4 to 6-5
- Power Macintosh installation, 1-2
- pseudo Wigner–Ville distribution (PWVD),
 - 3-5 to 3-8, 4-13, 6-2
 - demonstrating with Offline Joint Time-Frequency Analyzer, 5-13

Q

- Qian, S., A-1
- quadratic JTFA, 2-6
- quadratic JTFA algorithms, 3-3 to 3-13
- quadratic vs. linear methods, 6-1

R

- Raz, S., A-1
- references, A-1

S

- saving time-dependent spectrums to a text file, 6-6
- selecting an algorithm, 6-2 to 6-3
- short-time Fourier transform (STFT), 2-6,
 - 3-1 to 3-2, 4-3
- signal energy measurements, 6-4 to 6-5
- STFT spectrogram, 2-6, 3-3 to 3-4, 4-12, 6-2
 - demonstrating with Offline Joint Time-Frequency Analyzer, 5-11
- STFT, 2D, 4-4 to 4-5
- suppressing the DC component, 6-5

T

- 2D Gabor Expansion, 4-6 to 4-7
- 2D STFT, 4-4 to 4-5
- technical support, C-1 to C-2
- time-dependent spectrum
 - plotting with other software, 6-6
 - saving to a text file, 6-6
- time-dependent spectrum analysis examples,
 - 5-5 to 5-14
- Time-Frequency Distribution Series,
 - 4-15 to 4-16

V

- VIs
 - 2D Gabor Expansion, 4-6 to 4-7
 - 2D STFT, 4-4 to 4-5
 - adaptive spectrogram, 4-9
 - adaptive transform, 4-1 to 4-2
 - Choi–Williams distribution (CWD), 4-10
 - Cohen’s class, 4-10
 - cone-shaped distribution, 4-11
 - Fast Dual, 4-7
 - Gabor expansion, 4-4
 - Gabor spectrogram, 4-14
 - inverse adaptive transform, 4-2
 - Normalized Gaussian Window Function, 4-8
 - pseudo Wigner–Ville distribution (PWVD), 4-13
 - short-time Fourier transform (STFT), 4-3
 - STFT spectrogram, 4-12
 - Time-Frequency Distribution Series,
 - 4-15 to 4-16

W

Wexler, J., A-1
Wigner, Eugene P., 2-6
Wigner–Ville distribution (WVD), 2-6,
3-5 to 3-8
Williams, W. J., A-1
Windows 3.1 installation, 1-2
Windows 95 and Windows NT
installation, 1-1
works cited, A-1

X

Xia, X. G., A-2

Y

Yin, Q., A-1

Z

Zhao, Y., A-2