SANDIA REPORT

SAND2010-3628
Unlimited Release
Printed June 2010

SIRHEN: a data reduction program for photonic Doppler velocimetry measurements

Tommy Ao and Daniel H. Dolan

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2010-3628
Unlimited Release
Printed June 2010

SIRHEN: a data reduction program for photonic Doppler velocimetry measurements

Tommy Ao
Daniel H. Dolan
Dynamic Material Properties

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185

Abstract

SIRHEN (Sandia InfraRed HEtrodyne aNalysis) is a program for reducing data from photonic Doppler velocimetry (PDV) measurements. SIRHEN uses the short-time Fourier transform method to extract velocity information. The program can be run in MATLAB (2008b or later) or as a Windows executable.

Acknowledgments

The authors would like to thank following people for obtaining the data used to test the SIRHEN program: Aaron Bowers and Andy Shay for building the targets, Randy Hickman and Jesse Lynch for operating the gas gun, and Sheri Payne for fielding the PDV diagnostic.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Time-resolved velocimetry is one of the primary measurements in dynamic material studies. Traditionally, optical diagnostics such as velocity interferometer system for any reflector (VISAR) [1] and Fabry-Pérot interferometer system [2] have been used to measure material velocities in shock wave experiments. Each of those diagnostics have their advantages and disadvantages.

In VISAR measurements, Doppler shifted light from a moving target is split along two different paths (a reference leg and a delay leg of an interferometer), then recombined and recorded on digitizers. The velocity of the target is measured by comparing Doppler shifted light at time t with Doppler shifted light at time $t - \tau$, where τ is the delay time. A chief advantage of VISAR is that large velocities (> 1 km/s) can be tracked with modest diagnostic bandwidth. On the other hand, VISAR uses absolute intensities to obtain velocity information which is disadvantageous as the light intensity changes substantially. Fabry-Pérot interferometry is an alternative to VISAR that is more robust to extreme changes in light intensity, but has lower optical efficiency and is more costly and complex. Fabry-Pérot systems record data on streak cameras which have limited record lengths and are susceptible to image distortions. The main advantage of the Fabry-Pérot system is its ability to measure multiple discrete velocities simultaneously and velocity dispersion over a limited range, which VISAR cannot do. However, both systems are vulnerable to abrupt velocity changes and require having an additional backup system to resolve fringe jump ambiguities.

Photonic Doppler velocimetry (PDV), also known as heterodyne velocimetry [3], is a compact displacement interferometer system that is rapidly becoming a standard diagnostic in dynamic compression research. It has many of the advantages of VISAR and the Fabry-Pérot systems, while avoiding some their disadvantages. A PDV system is essentially a fiber-based Michelson interferometer, utilizing recent advances in near-infrared ($\lambda_0 = 1550$ nm) detector technology and fast digitizers to record beat frequencies in the gigahertz range. While fluctuations in the light intensity returned from the surface are observed in the data, PDV is robust against large variations. The use of Fourier transform techniques in the analysis of PDV data enable resolving multiple discrete velocities and even velocity dispersion. Additional advantages of PDV include simple assembly and operation, readily available components, and the lack of an intrinsic delay time.

This report describes the new Sandia InfraRed HEtrodyne aNalysis program (SIRHEN; pronounced "siren") that has been developed for efficient and robust analysis of PDV data. The program was designed for easy use within Sandia's dynamic compression community.

1.1 Program summary

The SIRHEN program is completely self-contained and employs a simple graphical user interface (GUI) to guide the user through the analysis process. It is written in MATLAB and can operate on any platform (Mac OS X, Windows XP/Vista, and Linux) with MATLAB version 2008b or later. For non-MATLAB users, a compiled executable is also available for Windows XP. The SIRHEN program can be obtained by contacting Tommy Ao (tao@sandia.gov) or Dan Dolan (dhdolan@sandia.gov).

1.2 Overview of report

Chapter 2 provides a theoretical background on photonic Doppler velocimetry and the analysis procedures used in the SIRHEN program. An overview of the operation of SIRHEN and its graphical user interface are presented in Chapter 3. In Chapter 4, several example problems are presented to give the user experience with the program. Chapter 5 summarizes the program's capabilities and discusses future releases.

Chapter 2

Theoretical background

This chapter reviews theoretical concepts used in the SIRHEN program. First, the conceptual operation of photonic Doppler velocity is described. Next, the analysis procedures employed in SIRHEN are presented. Finally, various PDV configurations are discussed.

2.1 Photonic Doppler velocimetry

PDV data may be analyzed in terms of either a displacement or a velocity framework [4]. The displacement mode analysis is more fundamental and provides a helpful starting point for the discussion. However, the velocity mode analysis described in the subsequent section is the actual basis of operation for the SIRHEN program.

2.1.1 Displacement mode

Figure 2.1 shows the conceptual layout of the PDV measurement in terms as a displacement interferometer. Coherent light (wavelength $\lambda_0 = 1550$ nm) input is split into two paths, one of which reflects off a moving target at position $x(t)$. The Doppler shifted light from the moving target is combined with the unshifted reference light at a reference position x_r. The interference between the two optical frequencies create a beat frequency that is proportional to the target velocity ($f = 2v/\lambda_0$). For example, a target moving at a *constant* velocity of 1 km/s would produce a measured PDV signal with a beat frequency of 1.29 GHz.

To develop a precise relationship between target motion and measured detector signal, suppose the electric field at the detector is given as,

$$E(t) = E_0(t)\cos\phi_0(t) + E_1(t)\cos\phi_1(t), \tag{2.1}$$

where $E_i(t)$ is the field amplitude and $\phi_i(t)$ is the optical phase of each signal, and the subscripts 0 and 1 refer to the reference and target paths, respectively. The cycle averaged output intensity is

$$I(t) = I_0(t) + I_1(t) + 2\sqrt{I_0(t)I_1(t)}\cos\Phi(t), \tag{2.2}$$

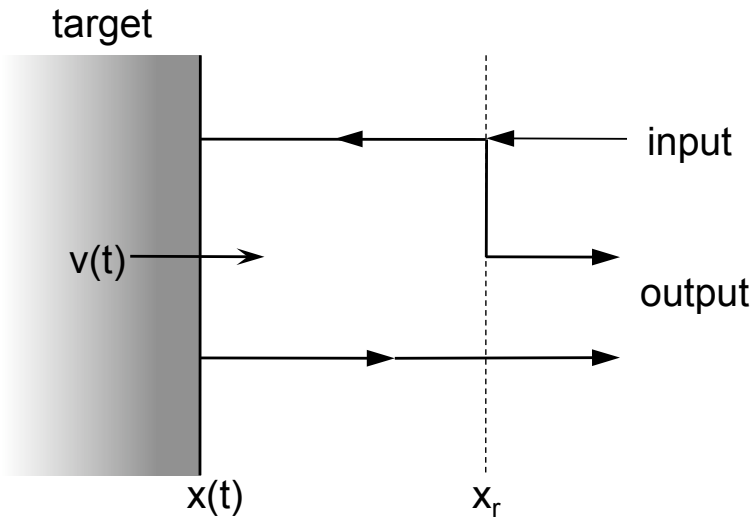

Figure 2.1. Conceptual diagram of PDV measurement.

where the optical phase difference is $\Phi(t) = \phi_1(t) - \phi_0(t)$, which describes the interference of light traveling along the two different paths. The output intensity is recorded using an optical detector that converts it to an electrical signal $s(t)$. Changes in optical phase difference lead to variations in the electrical signal which in turn relate to the motion of the reflecting target. For a monochromatic source (angular frequency $\omega_0 = 2\pi c/\lambda_0$), the input optical phase is a linear function of time,

$$\phi_{in}(t) = \omega_0 t + \delta_{in}. \tag{2.3}$$

The reference and reflected optical phases are time delayed versions of the input signal in which the reference signal time delay (T_0) is constant while the reflected signal time delay ($2T_1$) varies with the target position. The factor of 2 in the reflected signal delay is due to the round trip transit between the reference position x_r and the target postion $x(t)$. The output optical phases are given as,

$$\phi_0(t) = \phi_{in}(t - T_0), \tag{2.4}$$

$$\phi_1(t) = \phi_{in}[t - 2T_1(t))] - \delta_m, \tag{2.5}$$

where the phase change caused by reflection δ_m is assumed to be constant. Referring to Fig. 2.1, the reflection delay time is given as,

$$T_1(t) \approx \frac{x_r - x\,[t - T_1(t_i)]}{c};\ \ (v \ll c), \tag{2.6}$$

where $T_1(t_i)$ is the initial transit time. Omitting the constant time shift between $\Phi(t)$ and $x(t)$, the optical phase difference is related to the target's position relative to the reference position at time t_i,

$$\Phi(t) = \Phi(t_i) + 4\pi\frac{x(t) - x(t_i)}{\lambda_0}, \tag{2.7}$$

where $\Phi(t_i)$ is the initial phase difference, and $x(t_i) = x_r$. Thus, the electrical signal of a detector measuring the output intensity is given as,

$$s(t) = aI_0(t) + bI_1(t) + 2\sqrt{abI_0(t)I_1(t)}\cos\left[\Phi(t_i) + 4\pi\frac{x(t) - x(t_i)}{\lambda_0}\right] \tag{2.8}$$

where the constants a and b represent a collection of coupling factors from the reference and target paths, respectively and the detector's sensitivity. The target displacement can be determined by inverting Eq. 2.8, which may give ambiguous results due to the symmetry and periodicity of the cosine function; similar concerns led to the use of quadrature in VISAR [5]. To obtain velocity information, numerical differentiation is required which could result in substantial noise amplification [6].

2.1.2 Velocity mode

Numerical differentiation may be avoided using a velocity mode analysis. When the target undergoes a slow velocity change such that over a time interval τ, its instantaneous position may be approximated as

$$x(t) \approx x(\bar{t}) + \overline{v}(\bar{t})[t-\bar{t}]; \;\; (|t-\bar{t}| \leq \tau/2), \tag{2.9}$$

where $\overline{v}$ represents the average interval velocity, and $\bar{t}$ is the center of the time interval. The optical phase difference within this interval is

$$\Phi(t) \approx \Phi(t_1) - \overline{\omega}(\bar{t})[t-\bar{t}], \tag{2.10}$$

where $\overline{\omega} = 4\pi\overline{v}/\lambda_0$. The output electrical signal the detector may be given as,

$$s(t) = A\cos\left(\overline{\Phi} - \overline{\omega}[t-\bar{t}]\right), \tag{2.11}$$

where $\overline{\omega}$ represents the radial beat frequency within the signal. This frequency can be determined by calculating the electrical power spectrum using a short-time Fourier transform (STFT),

$$\widetilde{S}(\omega,\bar{t}) = \int_{-\infty}^{\infty} s(t)w(t)e^{-i\omega t}dt, \tag{2.12}$$

where $w(t)$ is a window function, discussed in the following section. At each time of interest $\bar{t}$ within an interval of time duration τ, the beat frequency is extracted from the peak of the power spectrum. The average velocity in each interval is given as

$$\overline{v} = \frac{\lambda_0}{2}\bar{f} = \frac{\lambda_0}{4\pi}\overline{\omega}, \tag{2.13}$$

where the velocity resolution is defined by how well the frequency peak can be resolved.

2.2 Analysis

In the SIRHEN program, a discrete Fourier transform (DFT) analysis is used to transform the input function $s(t)$ (detector signal in the time domain) into an output function $S(f)$ called the frequency domain representation. DFT is a specific kind of Fourier transform where the input function is discrete and whose non-zero values have a limited (finite) duration. This is precisely the type of data created by digitizing at a finite sampling rate a continuous PDV electrical signal,

$$s(t)[F_s] \longrightarrow s_n, \tag{2.14}$$

where F_s is the sampling frequency, and s_n is the discrete signal. DFTs can be computed efficiently using a fast Fourier transform (FFT) algorithm [7].

Unlike the discrete-time Fourier transform (DTFT), DFT only evaluates enough frequency components to reconstruct the finite segment that was analyzed. Using the DFT implies that the finite segment analyzed is one period of an infinitely extended periodic signal. Since that is not true of real experimental signals, a window function has to be used to reduce the artifacts in the spectrum. From Eq. 2.12, the previously described STFT analysis can be given as

$$S(f) = \sum_{n=0}^{N-1} s_n w_n e^{-i2\pi fn}, \tag{2.15}$$

where w_n is a discrete window function.

2.2.1 Window functions

Spectral leakage is an effect in frequency analysis of finite-length signals or finite-length segments of infinite signals where it appears as if some energy has "leaked" out of the original signal spectrum into other frequencies. Typically the leakage shows up as a series of frequency "side-lobes". Windows are weighting functions applied to data to reduce the spectral leakage associated with finite observation intervals.

When a signal (data) is multiplied by a window function w_n, the product is zero-valued outside the interval so all that is remains is the signal segment "viewed" through the window. The main criteria for the spectral response of a window are:

1. the width of the main-lobe,
2. the attenuation at the maximum height of a side-lobe, generally the first side lobe, and
3. the rate at which the peaks of the side-lobes fall-off.

Ideally, the spectral response of the window consists of:

1. a narrow main-lobe (corresponding to high frequency resolution),
2. a low first-side-lobe (corresponding to noise suppression), and
3. rapid fall-off of the remaining side-lobes.

A vast collection of window functions have been developed for DFT analysis [7, 8]. Four common window functions are available in the SIRHEN program: Boxcar, Hann, Hamming, and Blackman, which are shown in Fig. 2.2. The corresponding spectral response of the windows are shown in Fig. 2.3 and summarized in Table 2.1. The subsequent window descriptions use the following FFT algorithm terminology:

- N represents the width (total number of samples) of the discrete window function.
- n is an integer with values: $0 \leq n \leq N-1$.

Window	Highest Side-lobe Level (dB)	Side-lobe Fall Off (dB/oct)	Comments
Boxcar	-13	-6	sensitive to signal phase
Hann	-32	-18	preferable with low noise (1%)
Hamming	-43	-6	preferable with moderate noise (5-10%)
Blackman	-58	-18	widest main-lobe

Table 2.1. Spectral response of window functions.

Boxcar

The "Boxcar" or rectangular or Dirichlet window is the simplest window,

$$w_n = 1. \tag{2.16}$$

It extracts a segment of the signal without any other modification at all, which leads to discontinuities at the endpoints. Using a normalized coordinates with sample period $T = 1.0$ so that ω is periodic in 2π and hence identified as θ, the spectral response of the Boxcar is given as,

$$W(\theta) = \exp\left(-i\frac{N-1}{2}\theta\right)\frac{\sin\left[\frac{N}{2}\theta\right]}{\sin\left[\frac{1}{2}\theta\right]}, \tag{2.17}$$

whose magnitude is shown in Fig. 2.3(a). The width of its main-lobe is quite narrow, but its first-side-lobe is only -13 dB down from the main-lobe peak and the remaining side-lobes fall-off gradually at about -6 dB/octave.

Hann

The "Hann" or sometimes mistakenly called "Hanning" window is a raised cosine named after Julius von Hann,

$$w_n = 0.5\left[1 - \cos\left(\frac{2\pi n}{N-1}\right)\right]. \tag{2.18}$$

Not only is this window continuous, but so is its first derivative. As shown in Fig. 2.3(b), its first-side-lobe is -32 dB from the main-lobe peak and has a steeper side-lobe fall-off of about -18 dB/octave. However, its main-lobe is wider than the Boxcar's.

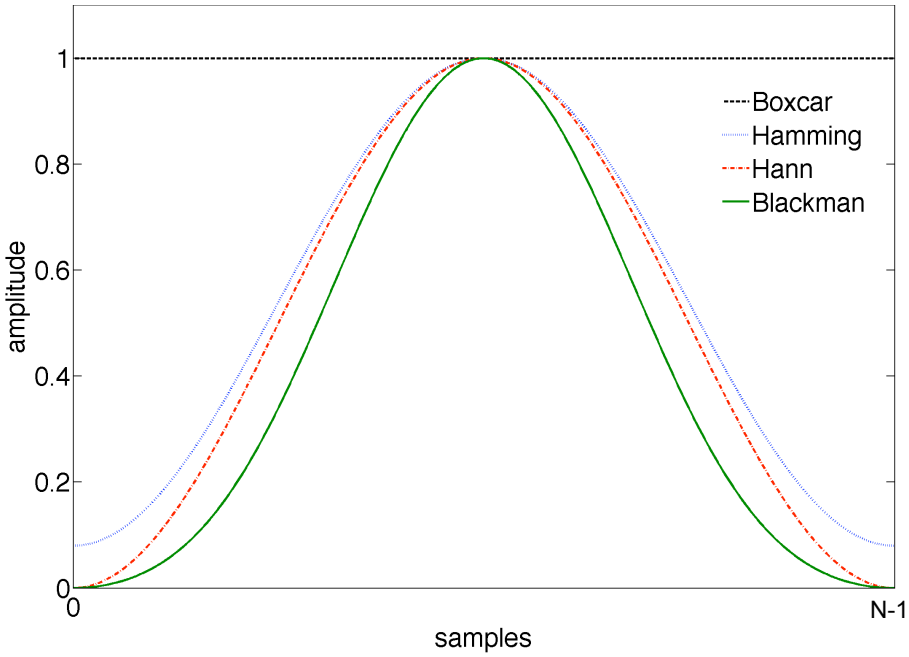

Figure 2.2. Window functions.

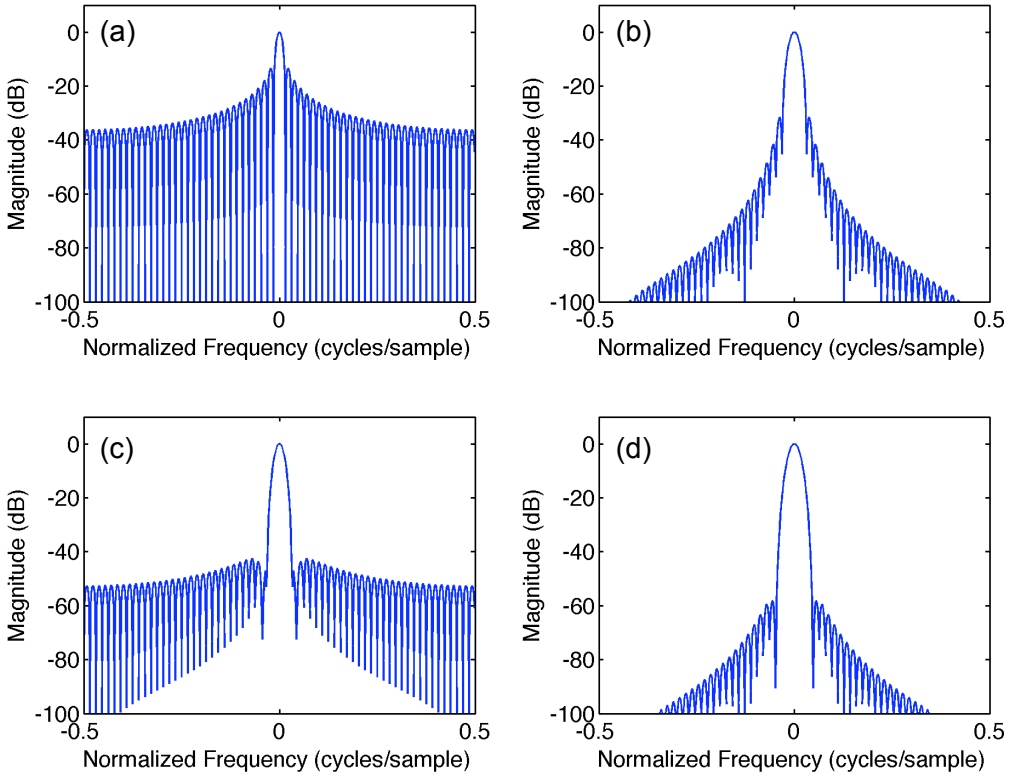

Figure 2.3. Spectral response of window functions: (a) Boxcar, (b) Hann, (c) Hamming, and (d) Blackman.

Hamming

The "Hamming" window, proposed by Richard W. Hamming, is also a raised cosine,

$$w_n = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right). \tag{2.19}$$

It is optimized to minimize the first-side-lobe (-43 dB), whose height is about one-tenth that of the Hann window's first-side-lobe, as shown in Fig. 2.3(c). The Hamming window has a main-lobe width similar to the Hann window, while it has a gradual side-lobe fall-off (-6 dB/octave) like the Boxcar window.

Blackman

R. B. Blackman developed the "Blackman" window,

$$w_n = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right). \tag{2.20}$$

As shown in Fig. 2.3(d), its first-side-lobe is down to about -58 dB and has a steep side-lobe fall-off (-18 dB/octave), but it also has the widest main-lobe of the 4 described windows.

2.2.2 Peak finding methods

After generating a power spectrum with the STFT analysis, the next stage of the SIRHEN program is to find the frequency location of the strongest peak within the power spectrum. Five peak finding methods are available in SIRHEN: maximum, Gaussian, parabola, centroid, and robust. Each method has its advantages and disadvantages, and are described in this section. A summary of the frequency range in which each peak finding method is applied is given in Table 2.2.

Method	Frequency Range	Comments
maximum	all f	velocity "hopping"
Gaussian	$f_1 \leq f \leq f_2$	computationally intensive
centroid	$f_1 \leq f \leq f_2$	computationally efficient
parabola	$f_1 \leq f \leq f_2$	fitting restricted to near peak
robust	all f	centroid method over entire data

Table 2.2. Properties of peak finding methods.

Maximum

The most obvious peak finding algorithm is to simply locate the frequency of the global maximum within the spectrum. While this method will always produce a result, the accuracy of the frequency value is dependent on the number of frequency points used in the STFT calculation. Consider the following example, a sinusoidal signal sampled at $N = 64$ points,

$$s_n = \cos\left(2\pi \frac{6n}{N-1}\right), \tag{2.21}$$

and windowed with a Hamming window function, as shown Fig. 2.4(a). This example signal has a normalized frequency of

$$\hat{f} = \frac{6}{2\pi(64-1)} = 0.095238. \tag{2.22}$$

Taking a FFT of the signal produces the power spectrum shown Fig. 2.4(b). Because of the limited number of sample points in the signal, the spectral peak only contains 3 frequency points. Using the maximum peak finding method, the normalized frequency of the peak is found to be $\hat{f} = 0.093750$, which results in peak finding error of $\sim 1.5\%$.

The procedure of appending zero value samples to the end of the signal (in this case $N_z = 448$ points and shown in Fig. 2.5(a)), otherwise known as "zero-padding", generates more frequency points in the spectrum, as shown in Fig. 2.5(b). Because appending zeros does not change the input sampling rate, F_s, the frequency span of the FFT output will remain the same and evenly distributed set of samples spread over 0 to $F_s/2$. The sample spacing of the new output must decrease to fit more samples over the same frequency range and corresponds to a resolution increase in those samples. However, it is important to note that only the frequency grid of the analysis is improved, but not the frequency peak width; the only way to do improve the latter is to increase the time duration of the block of signal that is examined. In any case, using the maximum method again on the zero-padding signal, a more accurate normalized frequency is obtained, $\hat{f} = 0.095703$, which reduces the peak finding error to $\sim 0.5\%$.

However, it should also be noted that large amounts of zero-padding slow the analysis considerably. In addition, the maximum method has problems during rapid transitions between low and high velocities as the intensities of the two different spectral peaks compete in magnitude. The maximum method tends to "hop" between the two extreme frequency values which producing a jagged velocity history. Thus, other more efficient peak finding methods that provide smoother transitions have been explored, and made available in the SIRHEN program. The following three methods (Gaussian, parabola, and centroid) uses an initial maximum method to center (f_0) and limit the data range ($f_1 \leq f \leq f_2$) for their subsequent peak finding algorithms.

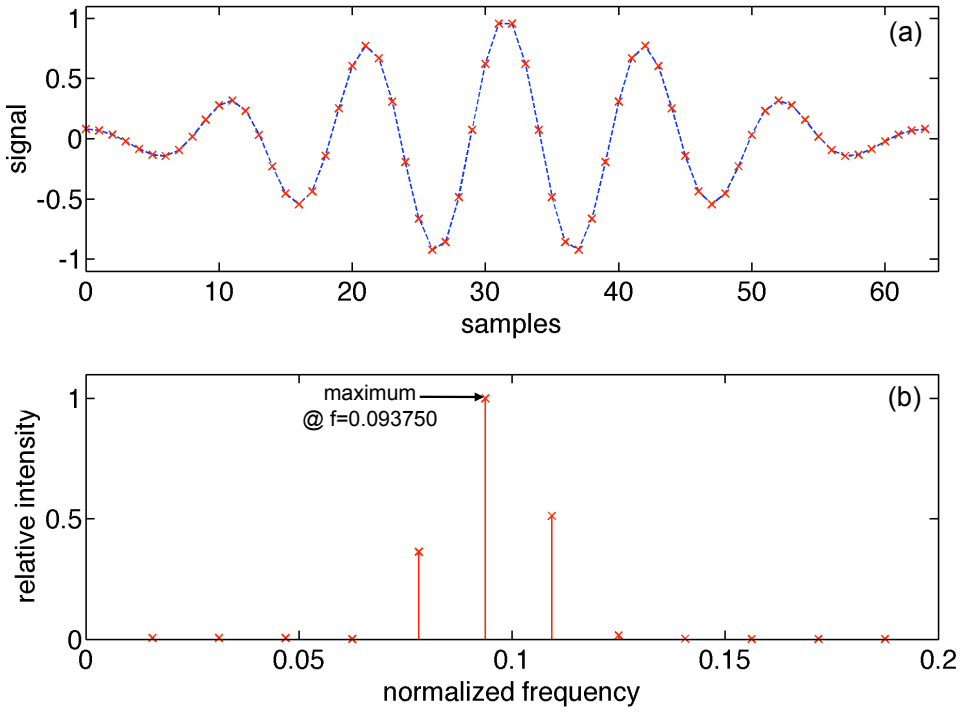

Figure 2.4. DFT analysis with no zero padding: (a) signal, and (b) spectrum.

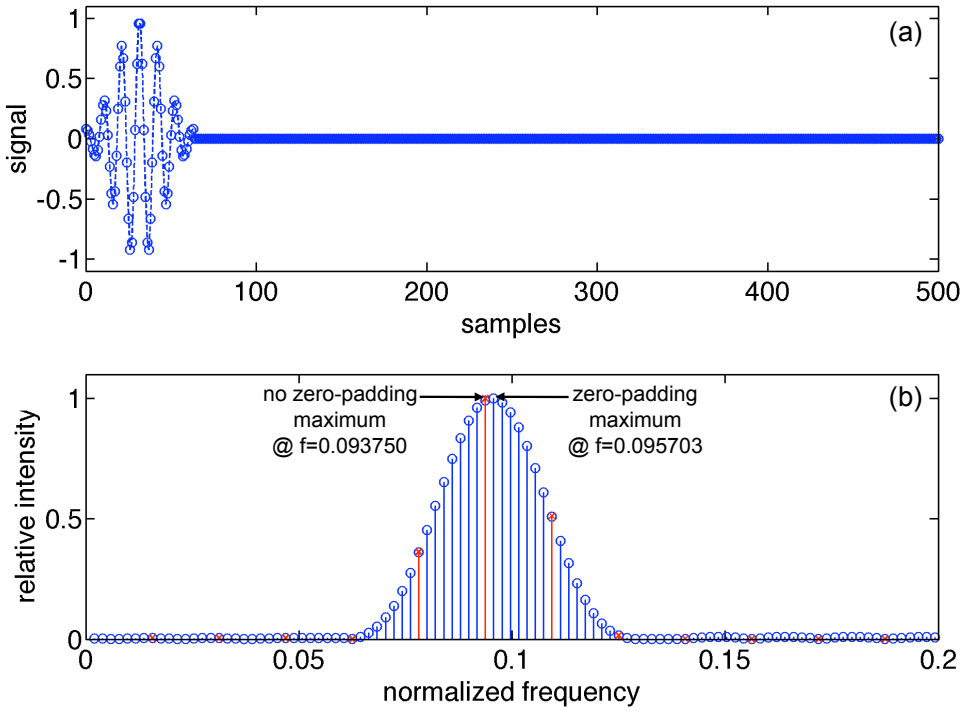

Figure 2.5. DFT analysis with zero padding: (a) signal, and (b) spectrum.

Gaussian

One of the most widely used peak finding algorithms is to fit the spectral peak with a Gaussian curve,

$$S(f) = A\exp\left(-\frac{(f-\bar{f})^2}{2\sigma^2}\right) + S_0, \tag{2.23}$$

where the fitting parameters are the amplitude A, the width σ, and the center frequecny $\bar{f}$ of the Gaussian peak, and the background signal level S_0. Even using just 3 points within the spectral peak, the Gaussian method noticeably improves peak frequency location accuracy over the maximum method. Certainly, having at least a moderate amount of zero-padding to the signal gives even better results. However, since the algorithm involves 2 linear and 2 nonlinear parameters in a least squares fitting, it may also be computationally intensive.

Parabola

Another peak finding method fits the spectral peak to a simple polynomial curve, namely a parabola,

$$S(f) = A(f-\bar{f})^2. \tag{2.24}$$

This fitting algorithm is more efficient than the Gaussian fitting since it uses the built-in polynomial fitting routine of MATLAB on a limited range of data restricted near the peak.

Centroid

Alternatively, instead of fitting the spectral peak to a curve, the profile is integrated to find its centroid location,

$$\bar{f} = \frac{\int_{f_1}^{f_2} fS(f)df}{\int_{f_1}^{f_2} S(f)df}. \tag{2.25}$$

This method tends to be less sensitive to any skewness in the spectral profile compared to the Gaussian and Parabola methods, and is very computationally efficient.

Robust

The final peak finding method, referred to as the "robust" method, uses the above centroid algorithm. However, instead of limiting the integrals to within a certain data range ($f_1 \leq f \leq f_2$), the robust method applies them over the entire signal.

2.3 PDV configurations

2.3.1 Standard

In a typical (standard) PDV configuration, a single laser light source is used to illuminate a target and provide an unshifted reference light for interference with the target light, as shown in Fig. 2.6(a). When the target is stationary, no beating within the PDV signal occurs since the reflected light from the target also remains unshifted.

The relationship between the time duration τ and characteristic peak width Δf follows the uncertainly product,

$$(\Delta f)(\tau) \geq \frac{1}{4\pi}. \tag{2.26}$$

For example, to achieve a velocity precision $\Delta v = 10$ m/s, the minimum time duration needed in the STFT analysis is $\tau = 6$ ns. For measured velocities that are reasonably large (> 1 km/s), the relative velocity precision ($\Delta v/v < 1\%$) is sufficient to investigate many dynamic material properties. However, low velocity (< 100 m/s) transients can be difficult to resolve with standard PDV since the beat period of the feature of interest may be longer than the time duration of the analysis. Also, in order to improve the poor relative velocity precision ($\Delta v/v \sim 10\%$), τ must be increased, thus sacrificing time precision.

2.3.2 Frequency-conversion

To provide optimal velocity and time precision measurements, a frequency-conversion PDV configuration has recently been developed where the reference light is preset at a slightly different frequency (wavelength) than the target light frequency. This is achieved by either using an acousto-optic (AO) frequency shifter to modify the reference light frequency, as shown in Fig. 2.6(b), or by using two separate laser sources at slightly different frequencies, as shown in Fig. 2.6(c). In this configuration, the PDV signal contains an underlying beat frequency even when the target is stationary. Thus, the low velocity features now have a shorter beat period than in the standard PDV configuration, which enables the use of a small time duration while maintaining sufficient velocity precision.

Suppose the target light before Doppler shift has wavelength λ_T, and the reference source has wavelength λ_R. The phase difference $\Phi(t) = \phi_T(t) - \phi_R(t)$, where

$$\phi_R(t) = \frac{2\pi c_0}{\lambda_R} t + \delta_R, \tag{2.27}$$

$$\phi_T(t) = \frac{2\pi c_0}{\lambda_T}[t - 2T(t)] + \delta_T - \delta_m, \tag{2.28}$$

$$T(t) \approx \frac{x_r - x(t)}{c_0}, \tag{2.29}$$

becomes

$$\Phi(t) = 2\pi c_0 \left[\frac{1}{\lambda_T} - \frac{1}{\lambda_R}\right] t + \frac{4\pi x(t)}{\lambda_T} + \delta_T - \delta_m - \delta_R - \frac{4\pi x_r}{\lambda_T}. \tag{2.30}$$

The fringe shift $F(t)$ (defined from a reference position x_0 at $t = 0$) is

$$F(t) = \left[1 - \frac{\lambda_T}{\lambda_R}\right] \frac{c_0 t}{\lambda_T} + \frac{2}{\lambda_T} \left(x(t) - x_0\right). \tag{2.31}$$

The time derivative of the fringe shift is the apparent frequency of the PDV signal,

$$\frac{dF}{dt} = \left[1 - \frac{\lambda_T}{\lambda_R}\right] \frac{c_0}{\lambda_T} + \frac{2v}{\lambda_T}. \tag{2.32}$$

Thus, when $\lambda_T > \lambda_R$ the frequency conversion acts in opposition to the Doppler shift, creating lower frequencies with increasing velocity (until crossing, at which point the frequency increases). In the opposite case ($\lambda_T < \lambda_R$), the frequency increases with positive velocity.

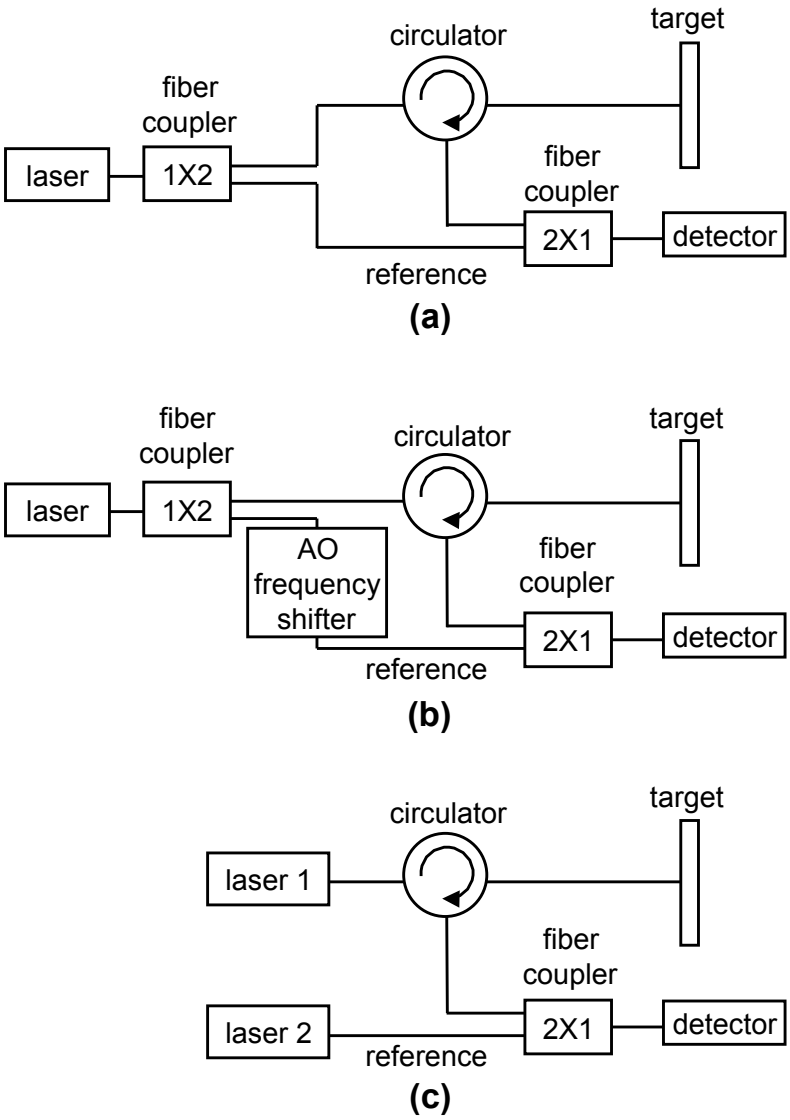

Figure 2.6. PDV configurations: (a) standard, (b) frequency-conversion with AO frequency shifter, and (c) frequency-conversion with 2 lasers.

Chapter 3

Program overview

An overview of the SIHREN program is presented in this chapter. First, program installation and execution instructions are given. Next, the program's operational stages are defined. Finally, the features of the graphical user interface are presented.

3.1 Installing and running SIRHEN

SIRHEN exists in two formats:

1. MATLAB version which runs within the MATLAB program, and

2. Windows executable version which may be used on Windows systems without the MATLAB program.

Installation of each version is different and are described separately in the following sections.

3.1.1 MATLAB version

The MATLAB version of SIRHEN is intended for MATLAB version 2008b or later. A valid MATLAB license (http://www.mathworks.com) is required. First, copy the folder `matlabSIRHEN`, which contains all files and subdirectories of the MATLAB version of SIRHEN to a local directory of the machine. Next, startup the MATLAB program. To install SIRHEN, add the `matlabSIRHEN` directory to the MATLAB path, using either the "addpath" command or the "Set path" tool on the "File" menu. Only the main folder itself, not the `private` subdirectory, should be added to the path. After installation is complete, the program can be started by typing "SIRHEN" at the command line.

3.1.2 Windows executable

The executable version of SIRHEN is intended for Windows XP. The program may operate in older versions of Windows but is not supported. Also, the executable version of SIRHEN has not been tested on Windows Vista and Windows 7 but should presumably work. After the contents of the `winexe` folder has been copied to a local directory of the machine, double click the `MCRInstaller2008b.exe` program and accept all default choices in the installation. This process installs necessary libraries and support functions for SIRHEN, and needs to be performed once for each machine where the program is to be used. When the MCR installer is complete, SIRHEN can be launched by double clicking on the `SIRHEN.exe` executable. The initial launch of the program will be somewhat slow as the various routines are unpacked for the first time, but subsequent launches should be considerably faster.

3.2 Analysis outline

Figure 3.1 outlines the analysis stages of the SIRHEN program. The PDV signal is cropped to the time range of interest for the experiment. The experiment signal is multiplied with a window function and analyzed with a short-time Fourier transform (STFT) over a certain time duration at discrete time points to generate a power spectrum of the signal. At each time point, the frequency of spectral peak is obtained with a peak finding algorithm. The peak frequencies are converted to velocity values which are collected to form a velocity history of the experiment.

An operational schematic of SIRHEN is shown in Fig. 3.2. All actions of the first stage are done within the "Selection Screen", while all actions of the second stage are done in the "Analysis Screen". The SIRHEN program opens with the Selection Screen (see Fig. 3.3) and begins with the user selecting a signal data file to import into SIRHEN with the "Load data" menu action. After the signal data is loaded, it is displayed in the top figure of the Selection Screen. Meanwhile, a coarse STFT is performed on the full signal and its STFT image is presented in the bottom figure. The signal may be shifted along the time axis with the "Shift signal" menu action. The time region of experimental interest is selected with the "Experiment region" menu action. A time region prior to the experiment region used for a reference calculation is chosen with the "Reference region" menu action.

To continue onto the second stage of operation, the user clicks on the "Next" button which replaces the Selection Screen with the Analysis Screen (see Fig. 3.4). The STFT is performed on the selected experiment region of the signal and its STFT image is displayed in the top figure of the Analysis Screen. Meanwhile, using a set of default parameters, the velocity history is calculated from the locations of the peaks of the STFT power spectrum and is presented in the bottom figure. Refinement of the velocity history is performed with the "Calculate history" menu action. The velocity history is saved as a text file with the "Export history" menu action. The STFT image is saved as a text file with the "Export STFT image" menu action. To go back to the Selection Screen, the user clicks on the "Previous" button.

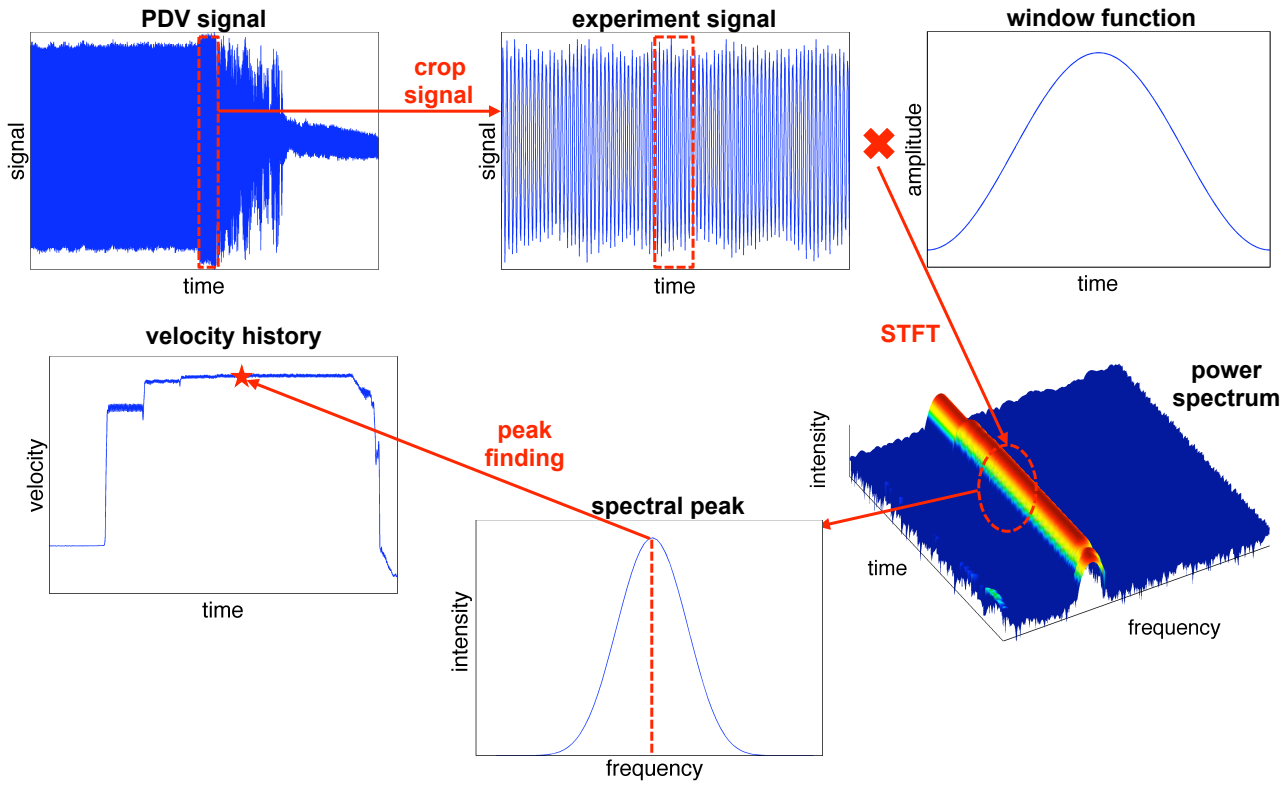

Figure 3.1. SIRHEN analysis stages.

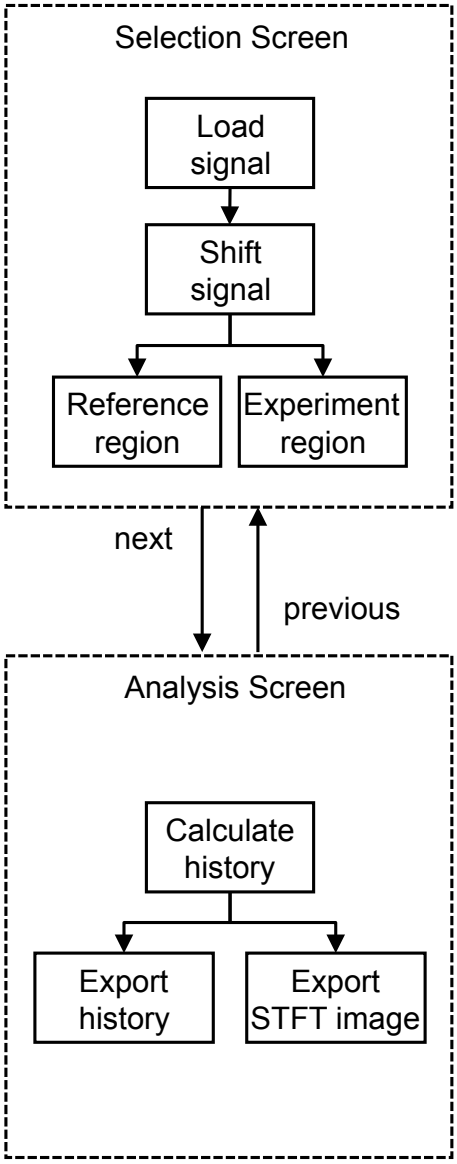

Figure 3.2. SIRHEN operational schematic.

3.3 Graphical user interface

This section describes the graphical user interface (GUI) of SIRHEN, as shown in Figs. 3.3 & 3.4. Each screen has two figures, a menu bar, and a toolbar.

3.3.1 Figures

In the Selection Screen, the top figure displays the loaded PDV signal and the bottom figure shows a STFT image over the full signal. In the Analysis Screen, the top figure displays a STFT image over the experiment region of the signal and the bottom image shows the corresponding calculated velocity history. Right clicking on the vertical axis label of the figures in the Analysis Screen allows the y-axis to be given in terms of absolute frequency, relative frequency, and velocity. Right clicking on the colorbar allows the display of the STFT image to be adjusted.

3.3.2 Menu bar

Each screen in SIRHEN has the common "Program" menu, "Help" menu, and "Options" menu. Meanwhile, the "Data" menu contains different actions for the separate screens.

Program and Help

The Program menu allows the user to restart and exit the program. "Restart" closes and relaunches the program, clearing all entries and returning the program to its default state. "Exit" closes the program. The Help menu provides general information about SIRHEN and briefly summarizes the operations that can be performed.

Options

The Options menu has the following actions:

1. General
   - Set parameters for PDV wavelength, velocity scale factor, and velocity level offset.
2. STFT
   - Set parameters for STFT calculation including number of time points, number of frequency points, overlap between time durations, window type, normalization, and DC removal.
3. Display
   - Set parameters for displaying the STFT image including scale range, color scaling, and color map.

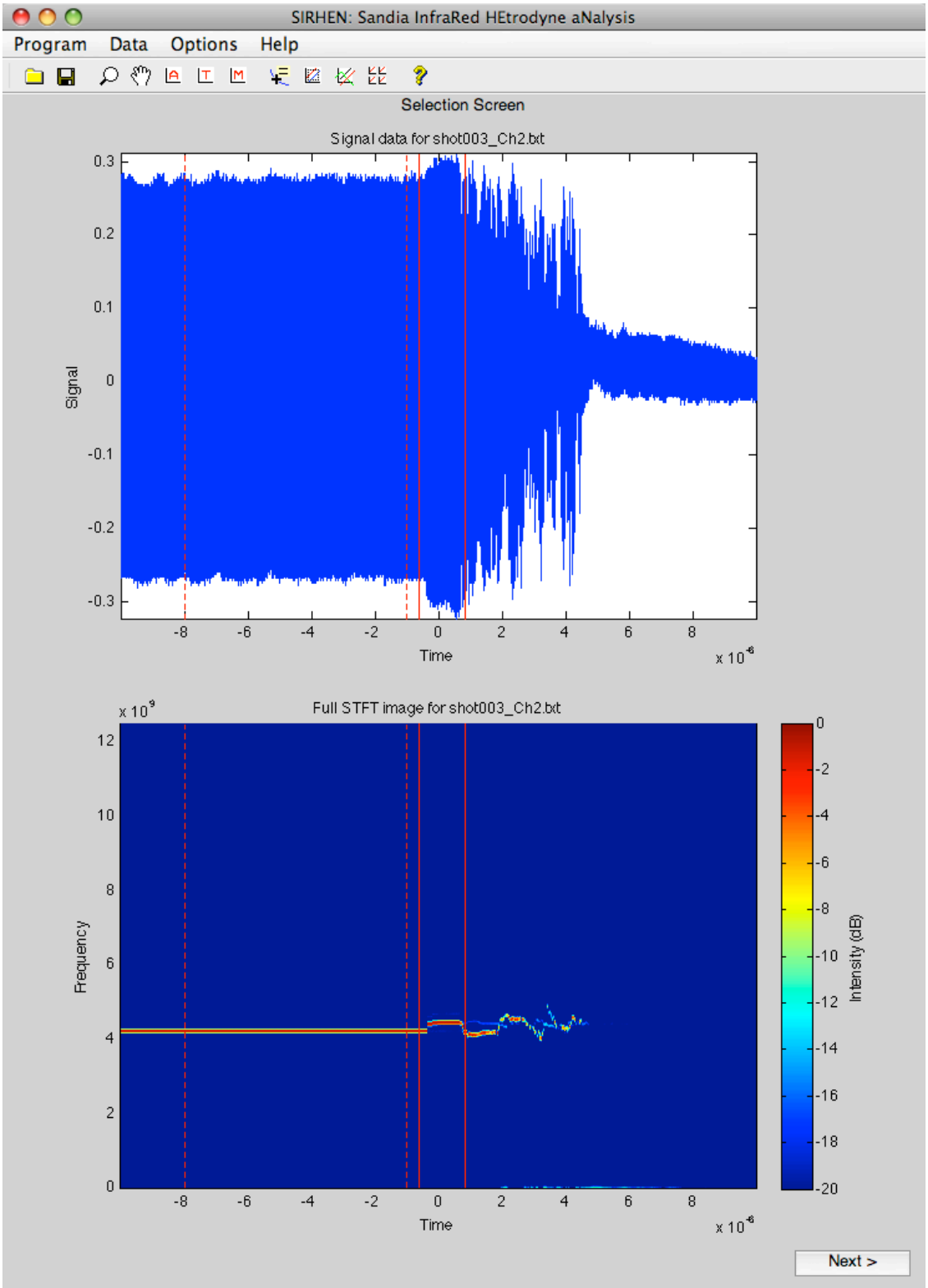

Figure 3.3. SIRHEN Selection Screen.

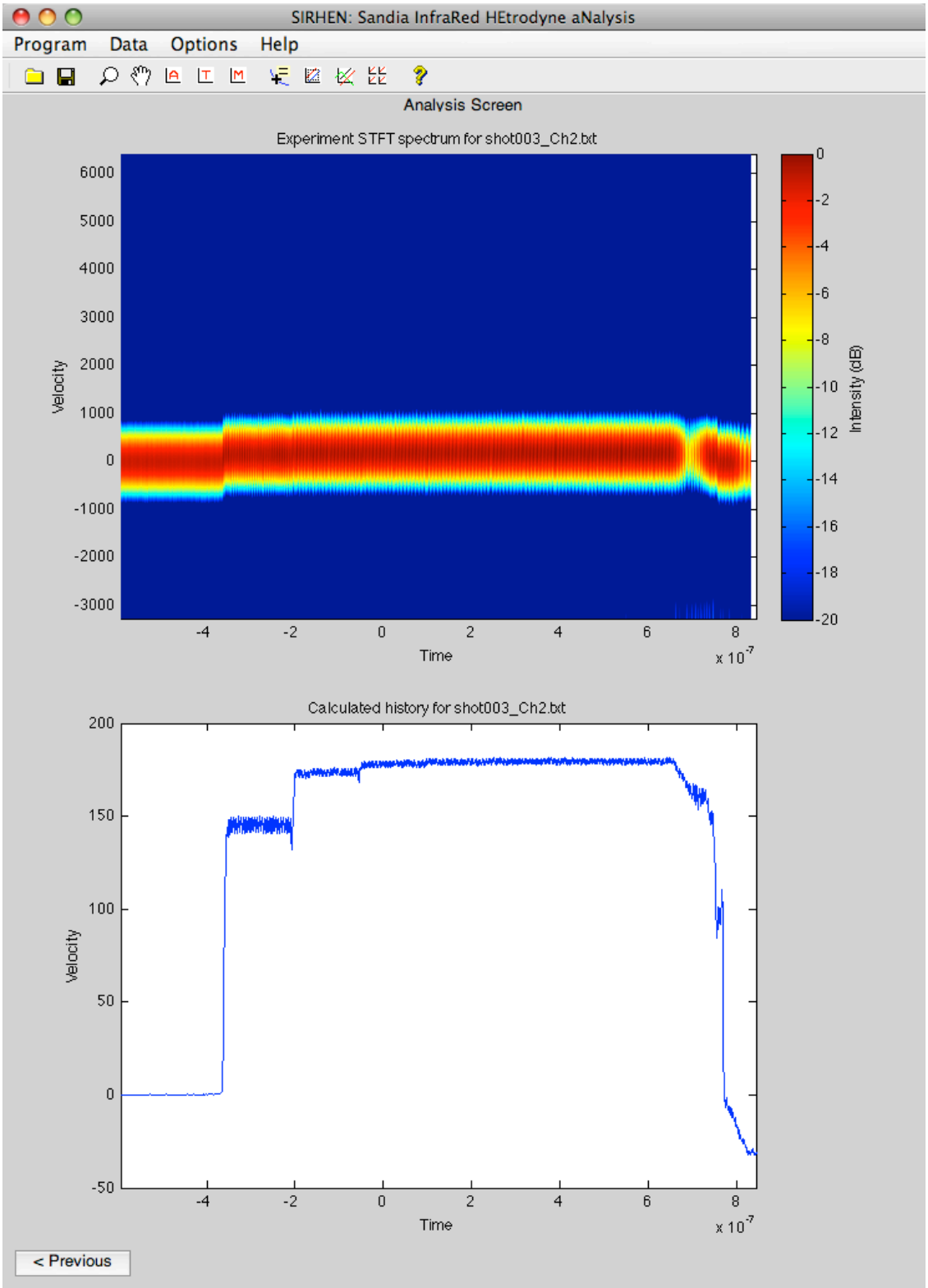

Figure 3.4. SIRHEN Analysis Screen.

Data

In the Selection Screen, the Data menu has the following actions:

1. Load signal
   - Select data file to import into the SIRHEN program. Supported file types include text files (*.txt, *.dat, *.csv), Tektronix binary files (*.wfm), and NTS binary files (*.dig).

2. Shift signal
   - Signal shifting along time axis (x-axis).

3. Reference region
   - Select time region for reference calculation.

4. Experiment region
   - Select time region of experimental interest.

In the Analysis Screen, the Data menu has the following actions:

1. Calculate history
   - Set parameters for calculating the velocity history including analysis duration, skip interval, minimum number of frequency points, and peak location method.

2. Export history
   - Save the calculated velocity history to an text file.

3. Export STFT image
   - Save the STFT image to a text file.

3.3.3 Toolbar

Each screen in SIRHEN contains a toolbar with the following actions:

1. Set working directory
   - Change the current directory using a dialog box.

2. Save figure
   - Save figure as a MATLAB *.fig file or a graphic file (*.pdf, *.jpg, etc.).

3. Zoom
   - Zoom in with mouse click or click and drag.
   - Zoom out with shift-click; double-click to restore original view.
   - Press right mouse button (control-click) for additional options.

4. Pan
   - Click and drag to pan over an axes; double-click to restore original view.
   - Press right mouse button (control-click) for additional options.

5. Auto scale axes
   - Click on axes to set auto limits.
   - Shift-click to auto scale all figure axes.

6. Tight scale axes
   - Click on axes to set tight limit.
   - Shift-click to tight scale all figure axes.

7. Manual scale axes
   - Click on axes to manually set limits.

8. Data cursor
   - Click on data to display (x,y,z) coordinates.
   - Press right mouse button (control-click) for additional options.

9. Region of interest (ROI) statistics
   - Click and drag to specify a region of interest.
   - Local statistics in this region will be displayed.

10. Overlay (x,y) data
    - Click on axes to overlay (x,y) data from a file.
    - Right-click overlays to make adjustments.

11. Clone axes
    - Click on axes to clone to another figure.

12. Toolbar help
    - Displays the above tools operations.

Chapter 4

Using SIRHEN

This chapter describes the practical use of the SIRHEN program using several example problems, which are located within the `examples` directory. The user is guided through the analysis of these examples in the following sections. The first section presents examples based on the standard PDV configuration, while the second section deals with examples using the frequency-conversion PDV configuration. The example data signals are synthesized under ideal conditions: noise-free and over-sampled by a 32-bit digitizer at 10× the Nyquist limit (20 samples/cycle).

4.1 Standard PDV examples

4.1.1 Velocity step

Example A-1 is based on an instantaneous velocity step at time $t = 0$ from an initial zero velocity to a final velocity $v_1 = 387.5$ m/s (see Fig. 4.1(a)),

$$v = \begin{cases} 0, & t < 0; \\ v_1, & t \geq 0. \end{cases} \tag{4.1}$$

The example file `exA-1.txt` contains the synthetic standard PDV signal for this velocity history, as shown in Fig. 4.1(b). Notice when there is no velocity ($t < 0$), the signal has no beat frequency and is ideally flat (under noise-free conditions). At $t = 0$, the signal immediately begins oscillating at the constant final beat frequency.

After starting the SIRHEN program, load the data file `exA-1.txt` using the Load signal action under the Data menu. The Selection Screen will display the full signal record and an initial STFT analysis of the signal. Adjust the program by setting the following Data menu parameters:

- Reference region: -Inf, -Inf;
- Experiment region: -5.00e-8, 4.00e-7.

Next, under the Options menu set the following General parameters:

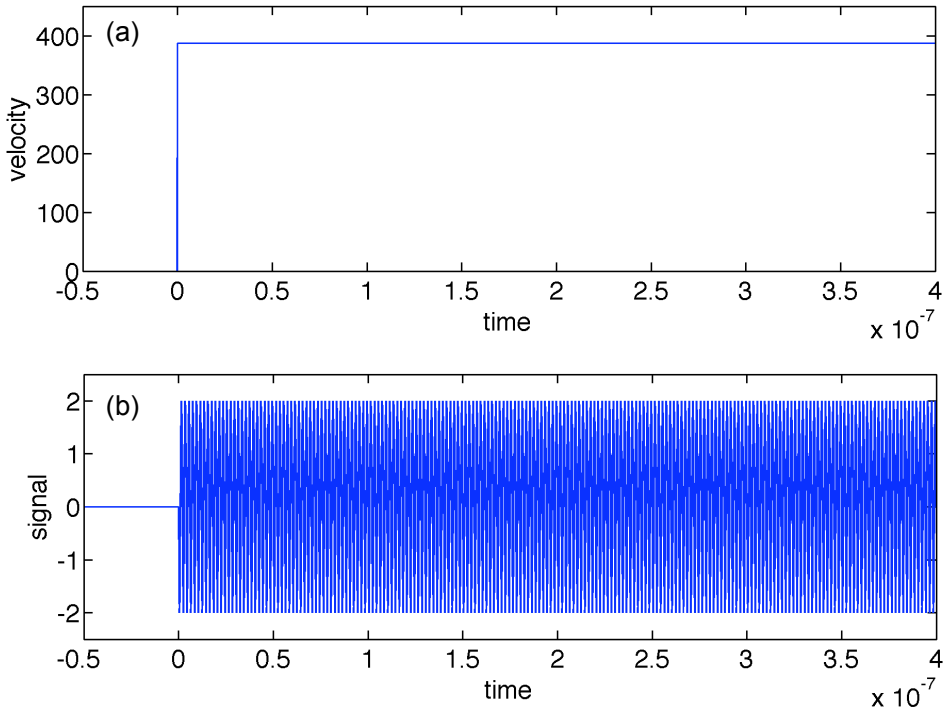

Figure 4.1. Example A-1 velocity step: (a) velocity profile, and (b) standard PDV signal.

- Wavelength: 1.550e-6,
- Velocity scale factor: 1,
- Velocity level offset: 0,

and the following STFT parameters:

- Number time points: 1024,
- Number frequency points: 1024,
- Overlap ratio: 4,
- Window type: Hamming,
- Normalization: global,
- DC removal: no,

and the following Display parameters:

- Scale range (min max): -40 0,
- Color scale: log,
- Color map: jet.

The Selection Screen should now look like Fig. 4.2. Click on the Next button to move to the Analysis Screen. Under the Data menu, set the following Calculate history parameters:

- Analysis duration: 5.0e-9,
- Skip interval: 2.0e-10,
- Minimum # frequency points: 2048,
- Peak location method: maximum.

The Analysis Screen should look like Fig. 4.3. Under the Data menu, select the Export history action to save the extracted velocity profile.

A comparison of the various peak finding methods is shown in Fig. 4.4, where in each case only the "Peak location method" parameter was changed. Within each of the subplots, the input velocity profile (dashed line) used to obtain the example signal data is also shown. The maximum method gives the correct zero initial velocity, but has noticeable ringing (oscillations) during the

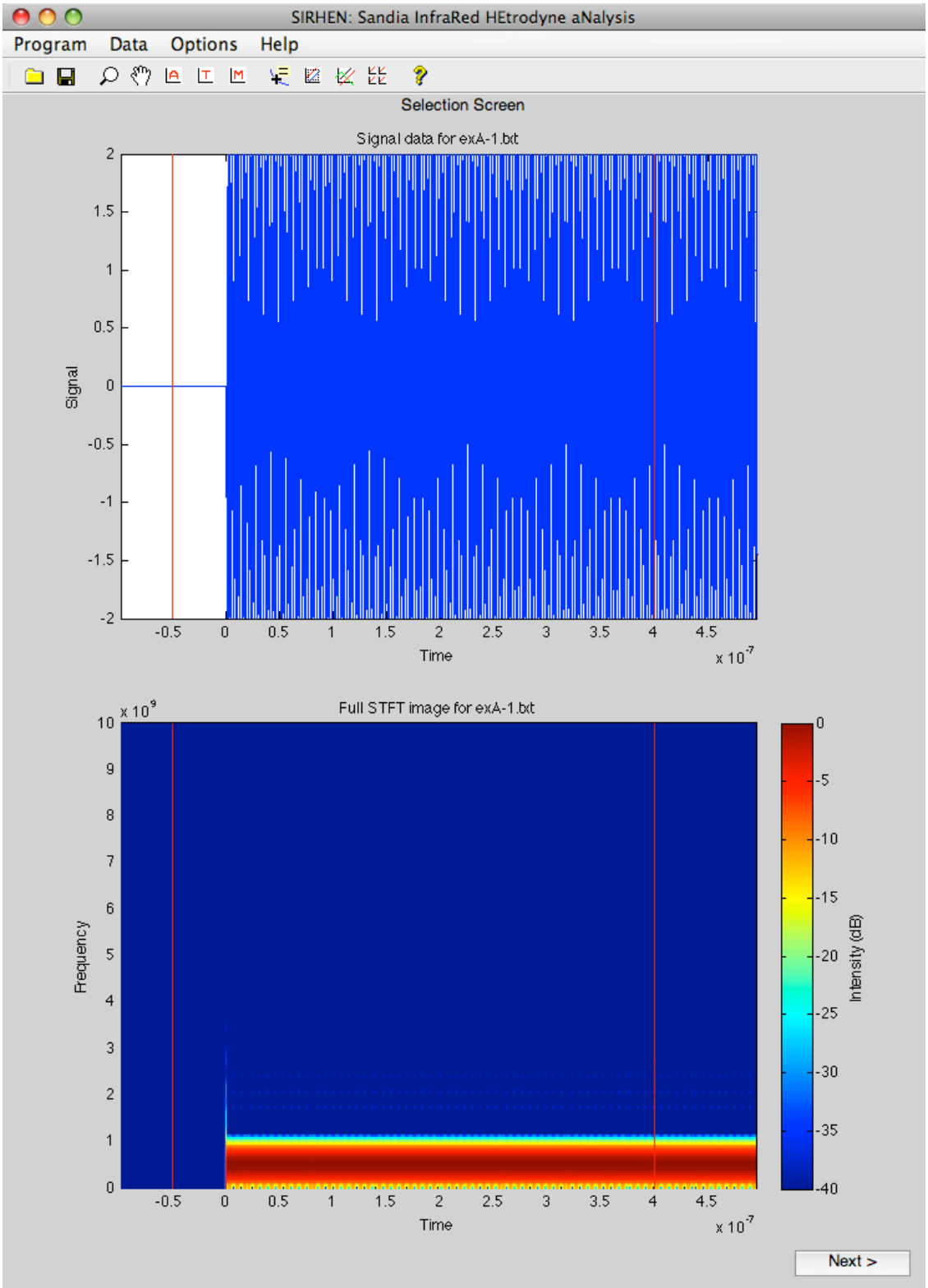

Figure 4.2. Example A-1 Selection Screen.

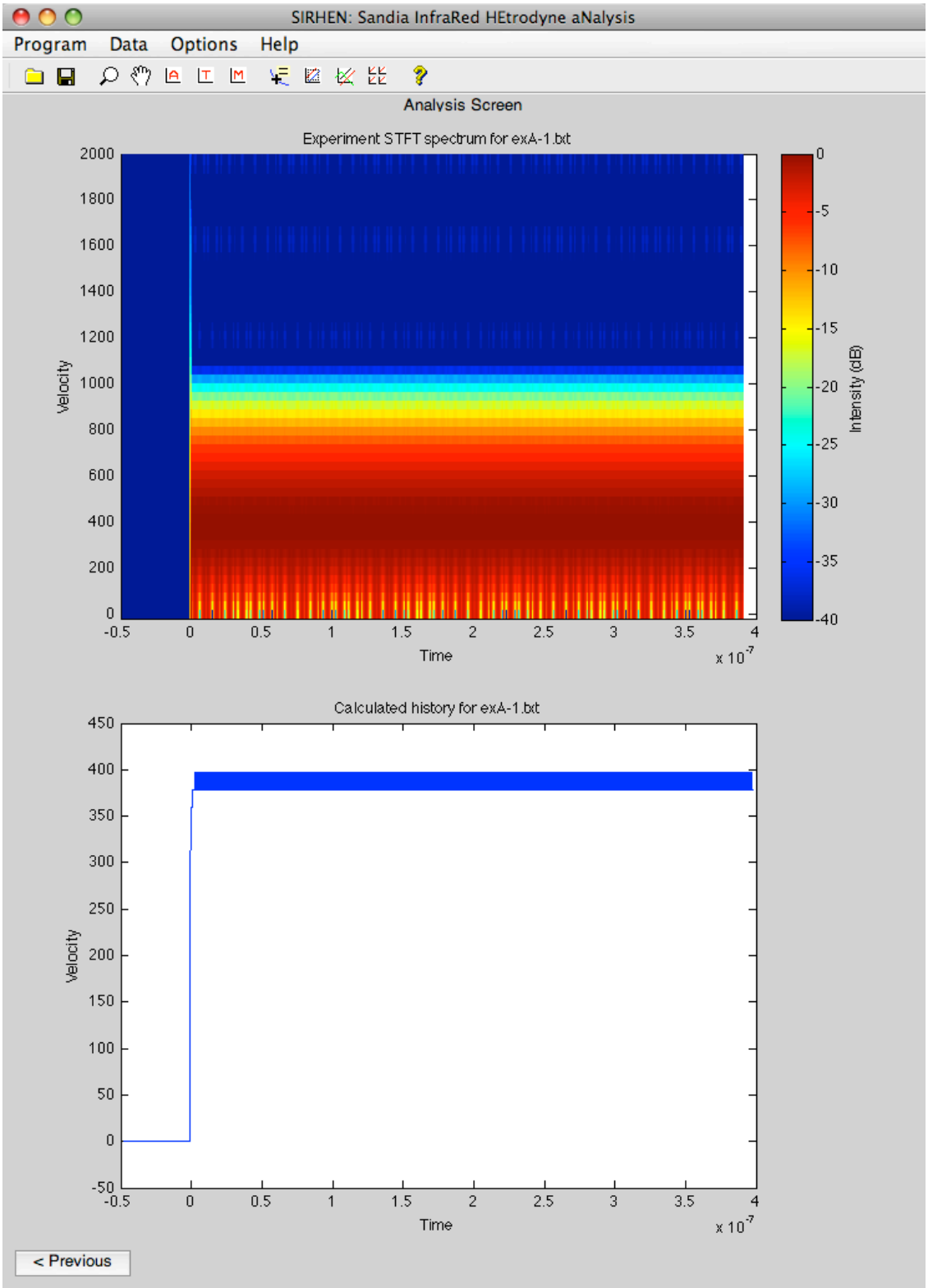

Figure 4.3. Example A-1 Analysis Screen.

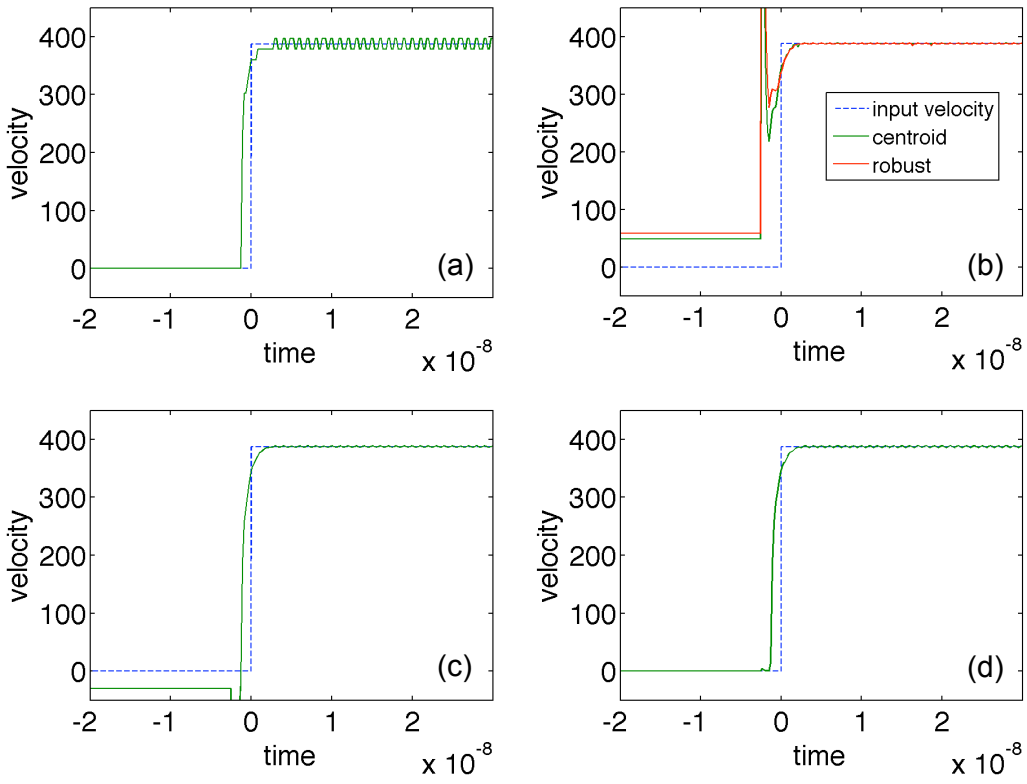

Figure 4.4. Example A-1 extracted velocities from peak finding methods (solid lines) vs input velocity (dashed line): (a) maximum, (b) centroid and robust, (c) parabola, and (d) Gaussian.

final velocity plateau. The centroid and robust methods both reduce the final velocity ringing, but each has a large positive velocity spike during the step transition and an incorrect positive finite initial velocity. The parabola method also reduces the final velocity ringing, but has a large negative velocity spike and an incorrect negative initial velocity. The Gaussian method gives the correct zero initial velocity and a relatively flat final velocity.

The risetime of the velocity step is related to the analysis duration used ($\tau = 5$ ns), thus it can be slightly improved by shrinking the analysis duration. However, the analysis duration should not be made less than the beat period of the signal which for this example is $T = 2$ ns.

4.1.2 Velocity ramp

Example A-2 is based on the following linear ramp profile (see Fig. 4.5(a)),

$$v = \begin{cases} 0, & t < 0; \\ at, & 0 \le t \le t_1; \\ v_1, & t > t_1. \end{cases} \tag{4.2}$$

The example file `exA-2.txt` contains the synthetic standard PDV signal for this velocity history under ideal conditions, as shown in Fig. 4.5(b). Load the example file `exA-2.txt`, and set the same program parameters as in the previous example. The Selection Screen should now look like Fig. 4.6.

In the Analysis Screen under the Data menu, set the following Calculate history parameters:

- Analysis duration: 15.0e-9,
- Skip interval: 2.0e-10,
- Minimum # frequency points: 2048,
- Peak location method: maximum.

The Analysis Screen should now look like Fig. 4.7. Under the Data menu, select the Export history action to save the extracted velocity profile. A comparison of the various peak finding methods is shown in Fig. 4.8, where in each case only the "Peak location method" parameter was changed. The maximum method gives the correct zero initial velocity, but produces large discrete steps within the ramp profile and an incorrect low final velocity plateau. Both the centroid and robust method reduce the discreteness of the ramp profile and give the correct final velocity, but have incorrect positive finite initial velocities and positive velocity spikes. The parabola method also reduces the discreteness of the ramp profile and gives the correct final velocity, but has an incorrect negative initial velocity and a negative velocity spike. The Gaussian method gives the correct initial velocity and final velocity, and a smooth ramp profile, but like all of the other methods has noticeable oscillations at low velocities.

The inability of all the methods to resolve the low velocity portion of the ramp is really due to the shortcoming of standard PDV, which consequently has led to the development of frequency-conversion PDV, and illustrated in the next section's examples.

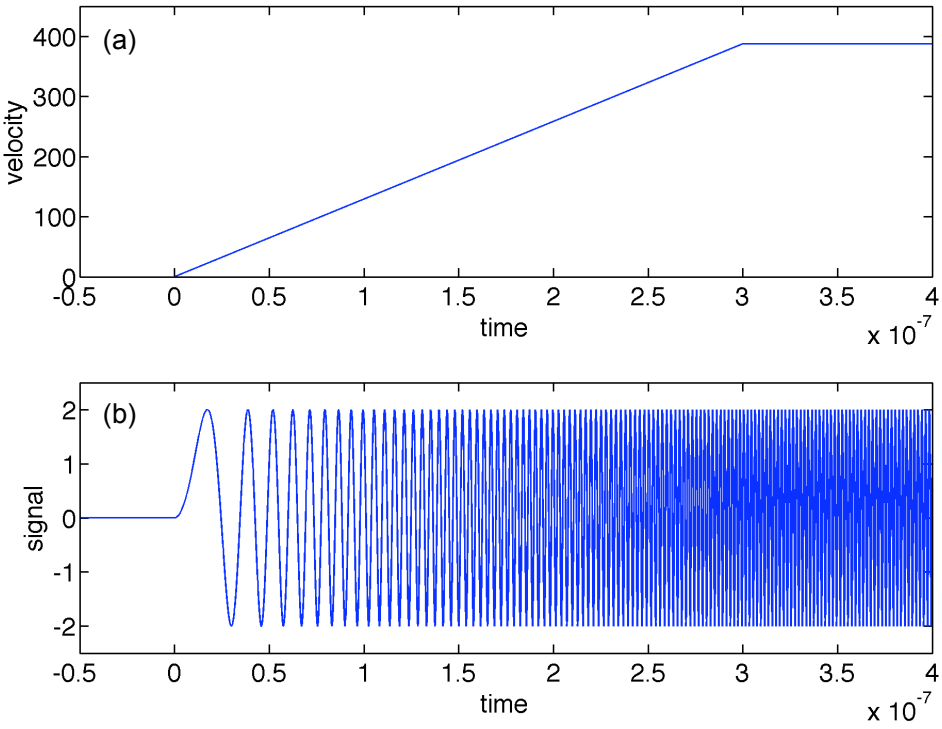

Figure 4.5. Example A-2 velocity ramp: (a) velocity profile, and (b) standard PDV signal.

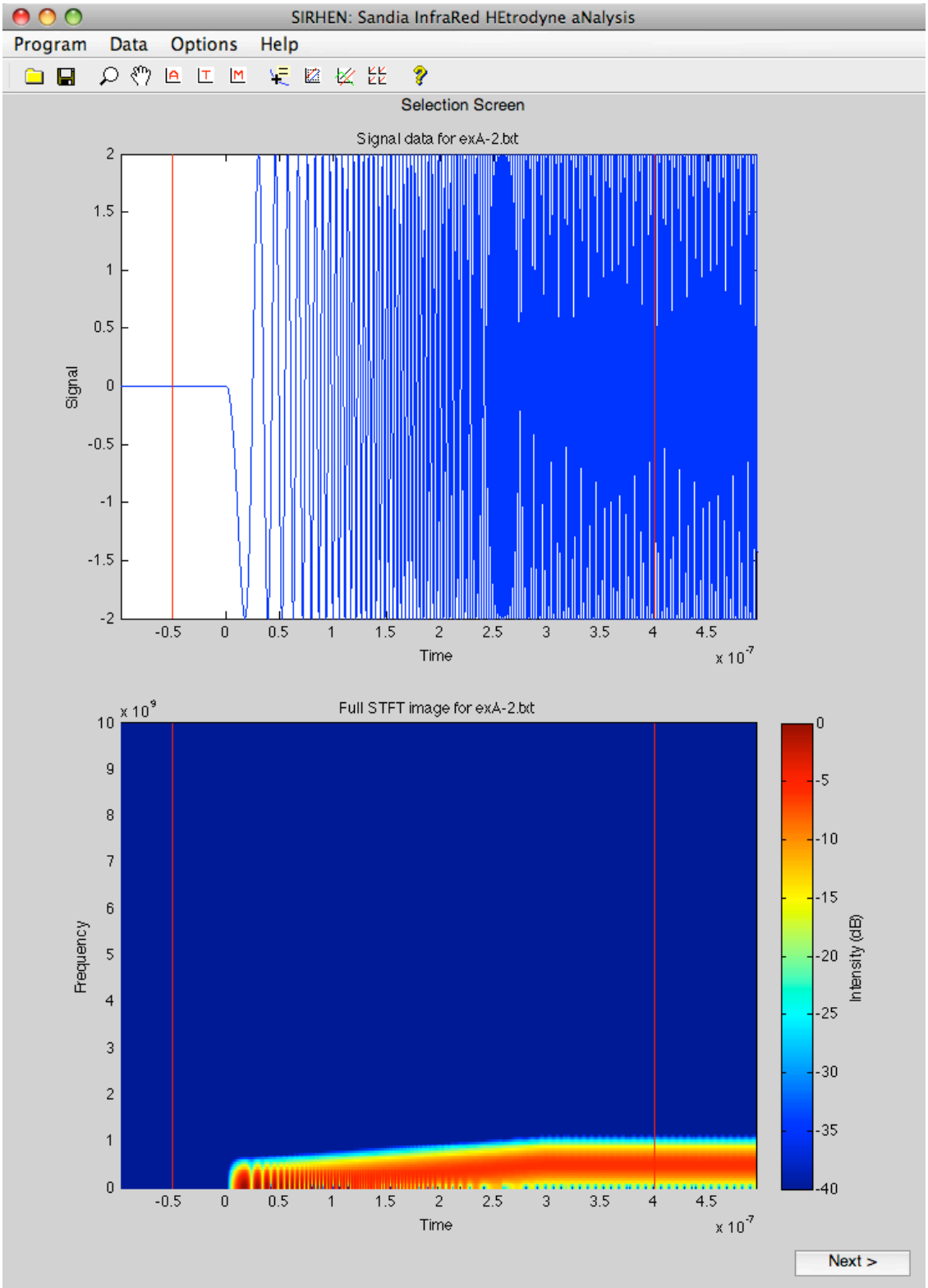

Figure 4.6. Example A-2 Selection Screen.

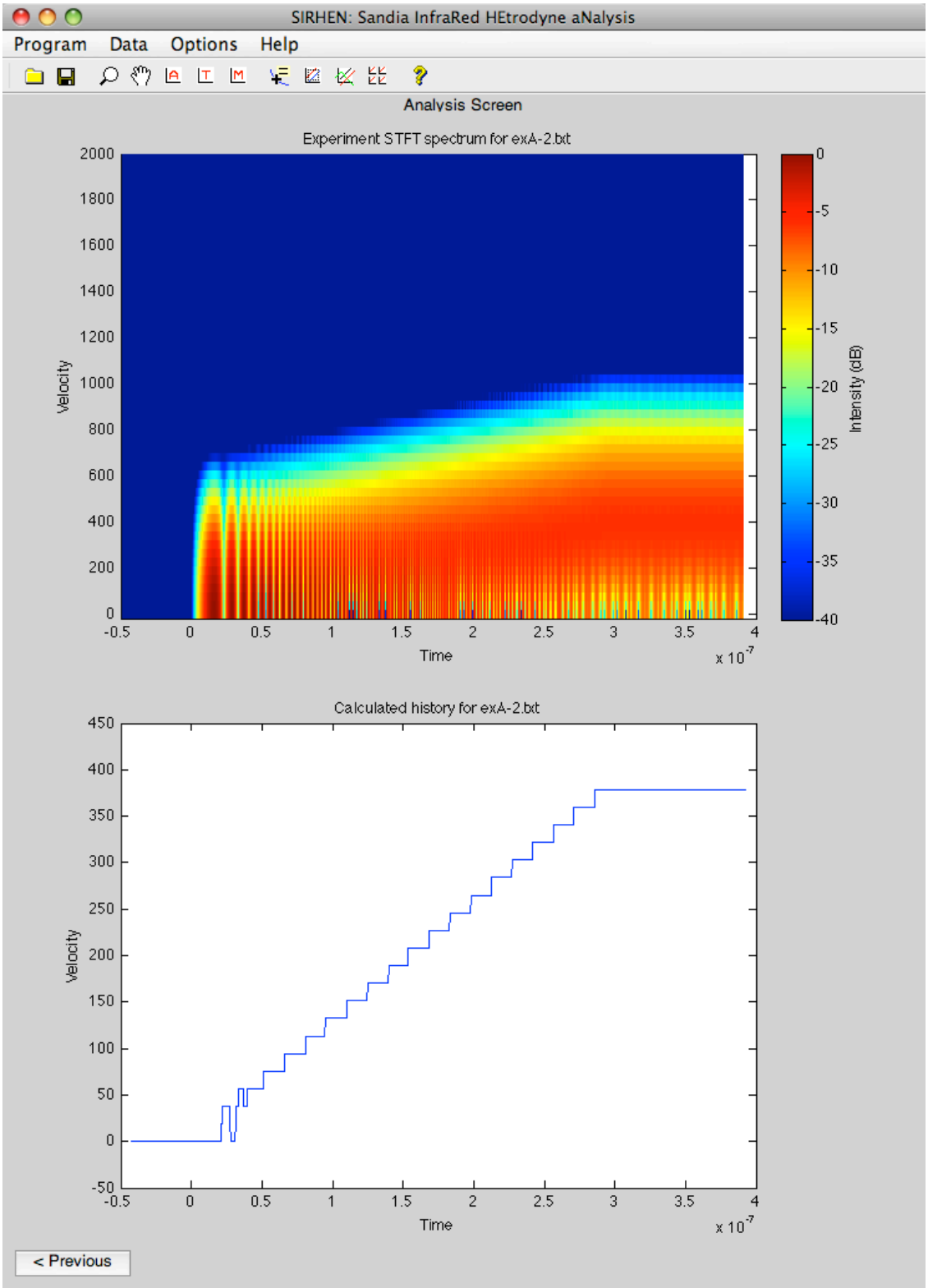

Figure 4.7. Example A-2 Analysis Screen.

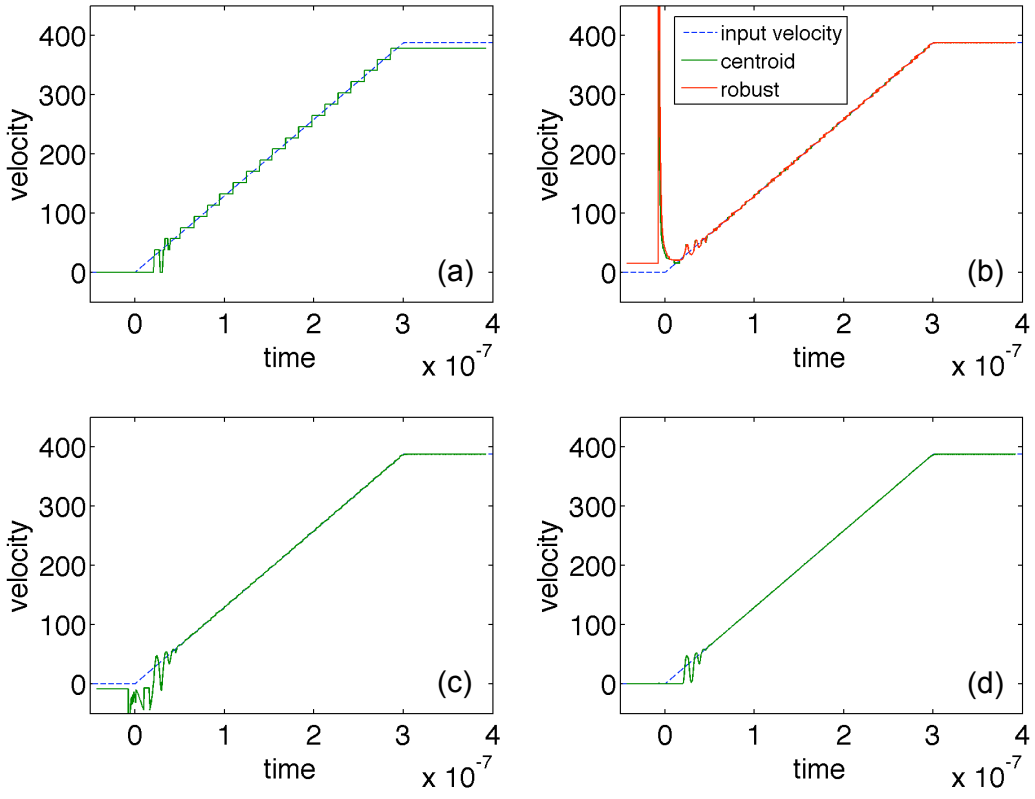

Figure 4.8. Example A-2 extracted velocities from peak finding methods (solid lines) vs input velocity (dashed line): (a) maximum, (b) centroid and robust, (c) parabola, and (d) Gaussian.

4.2 Frequency-conversion PDV examples

As in the previous section, similar velocity step and velocity ramp examples are presented here. However, an underlying beat frequency ($f_0 = 0.5$ GHz) is now contained within each of the example PDV signals.

4.2.1 Velocity step

Example B-1 is the frequency-conversion PDV velocity step example. The example file `exB-1.txt` contains the frequency-conversion PDV signal for a velocity step, as shown in Fig. 4.9(a). At times $t < 0$, the signal displays only the underlying beat frequency or "reference frequency" of 0.5 GHz. At $t = 0$, a noticeable jump in the signal's frequency is observed at the onset of target motion. Load the example file `exB-1.txt`, and in the Selection Screen adjust the program by setting the following Data menu parameters:

- Reference region: -9.00e-8, -1.00e-8;
- Experiment region: -5.00e-8, 4.00e-7.

The Selection Screen should look like Fig. 4.10 and the Analysis Screen should look like Fig. 4.11. Under the Data menu, select the Export history action to save the extracted velocity profile. A comparison of the various peak finding methods is shown in Fig. 4.12, where in each case only the "Peak location method" parameter was changed. The maximum method gives the correct final velocity, but produces some ringing around the zero initial velocity. The centroid, robust, parabola, and Gaussian methods all reduce the initial ringing and give almost identical velocity profiles.

4.2.2 Velocity ramp

Example B-2 is the frequency-conversion PDV velocity ramp example. The example file `exB-2.txt` contains the frequency-conversion PDV signal for a velocity ramp, as shown in Fig. 4.9(b). It might be less obvious than the previous step velocity example, but a careful examination does reveal a noticeable increase the signal's frequency starting at $t = 0$ due to the onset of target motion. Load the example file `exB-2.txt`, and and set the same program parameters as in the previous example. The Selection Screen should look like Fig. 4.13. In the Analysis Screen under the Data menu, set the following Calculate history parameters:

- Analysis duration: 15.0e-9,
- Skip interval: 2.0e-10,
- Minimum # frequency points: 2048,

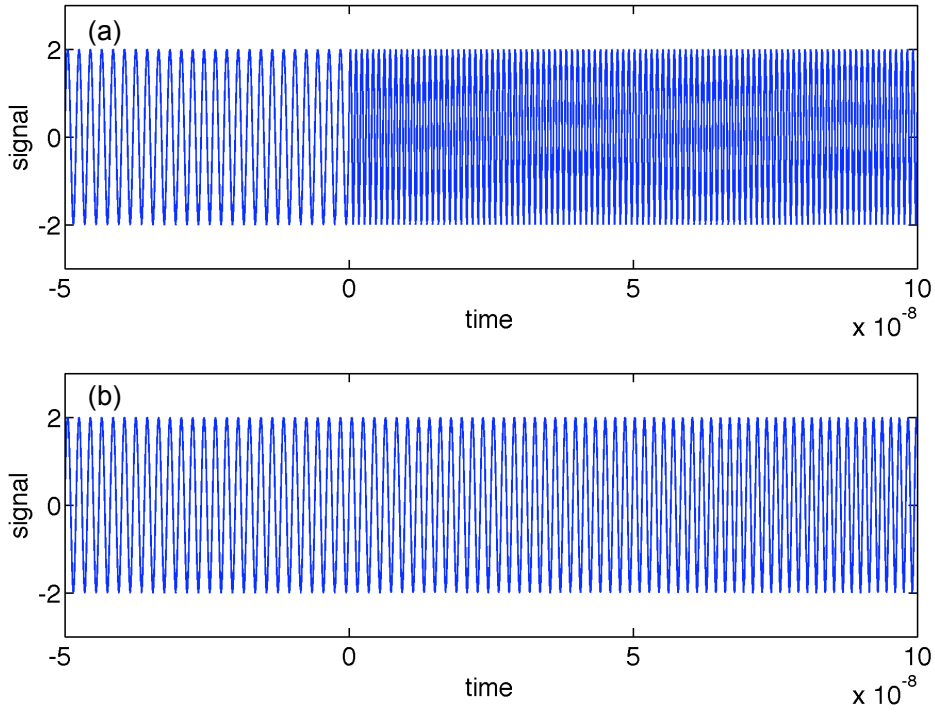

Figure 4.9. Examples B-1 and B-2: (a) B-1, frequency-conversion PDV signal of velocity step, (b) B-2, frequency-conversion PDV signal of velocity ramp.

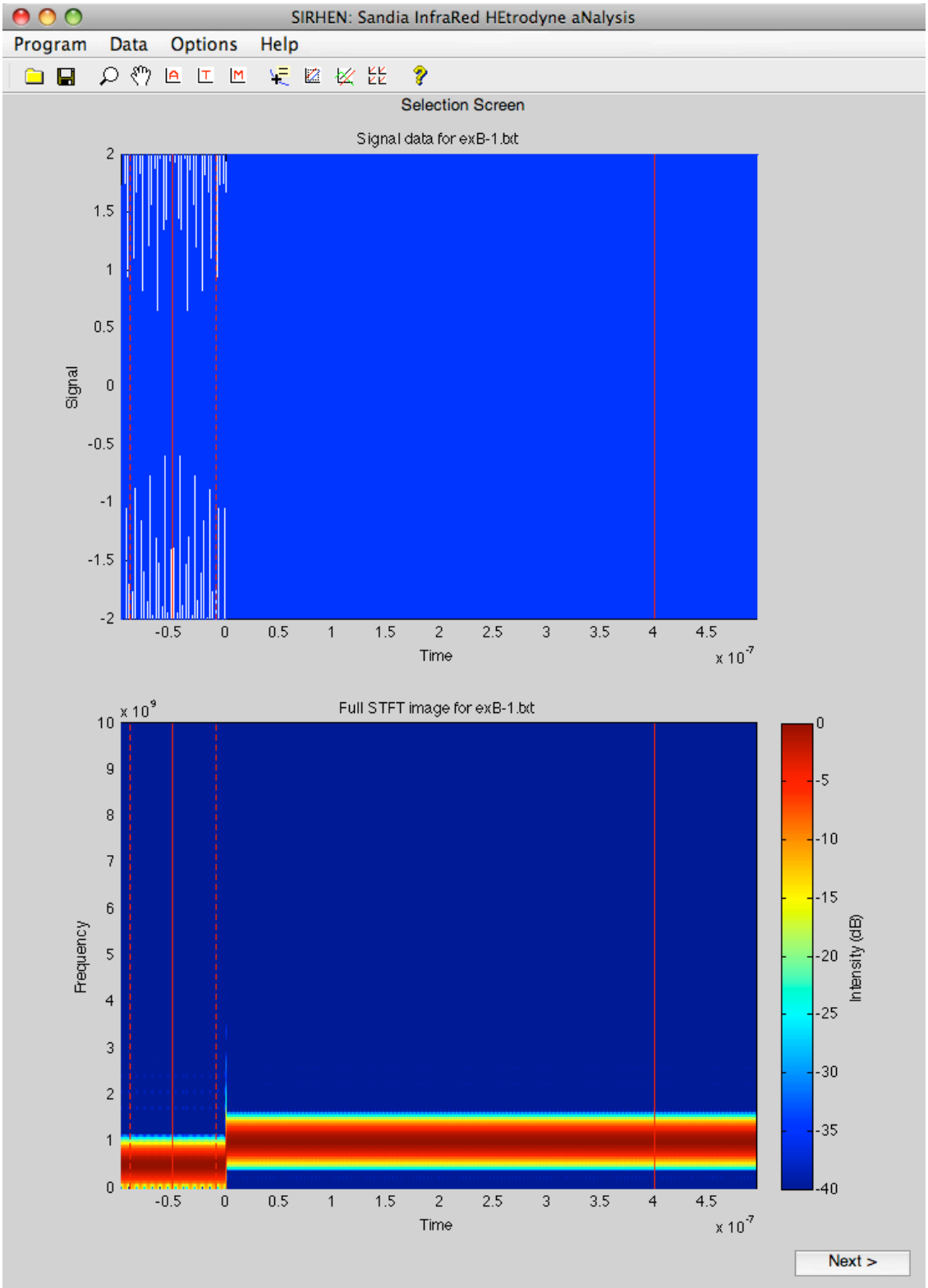

Figure 4.10. Example B-1 Selection Screen.

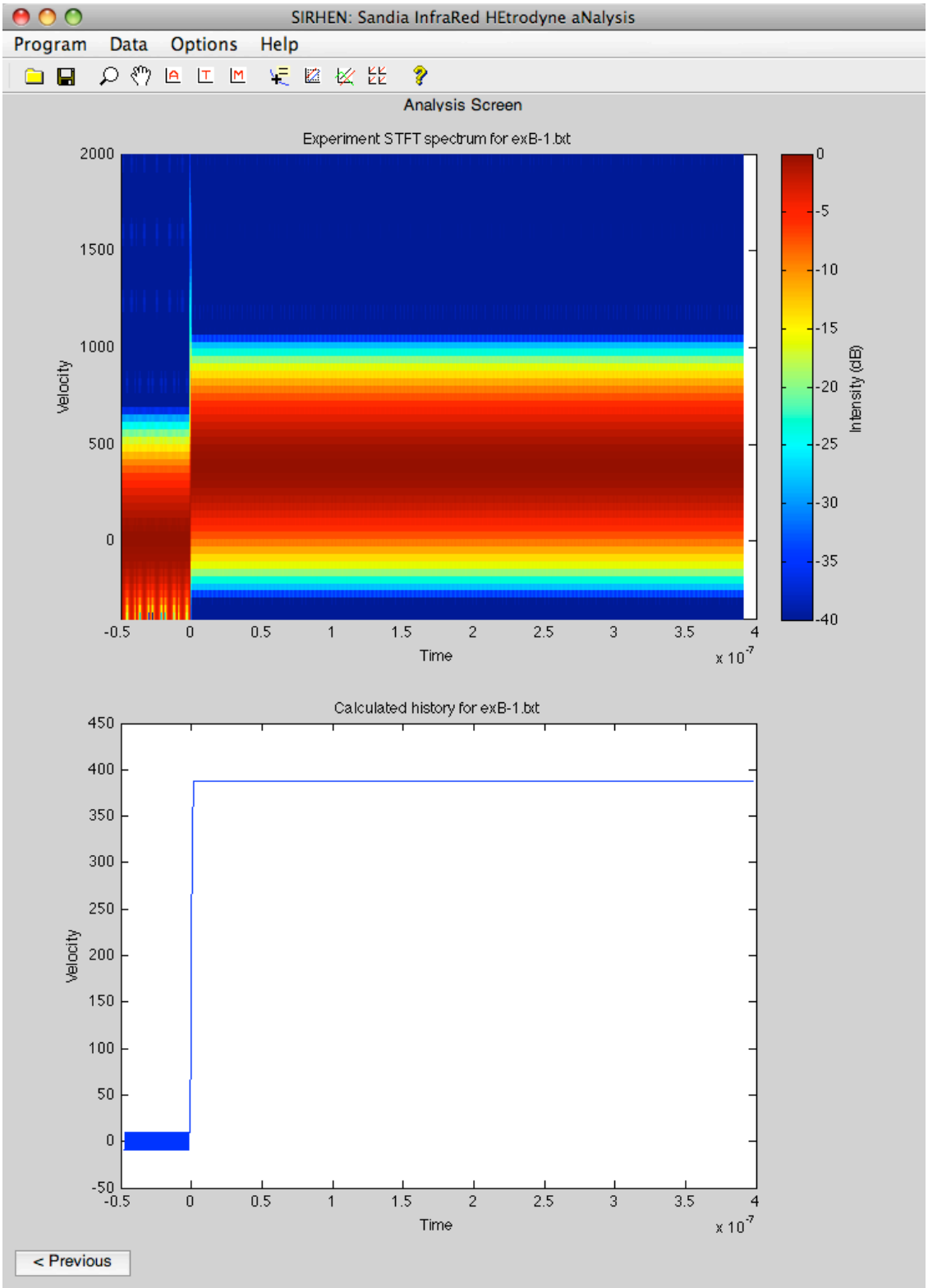

Figure 4.11. Example B-1 Analysis Screen.

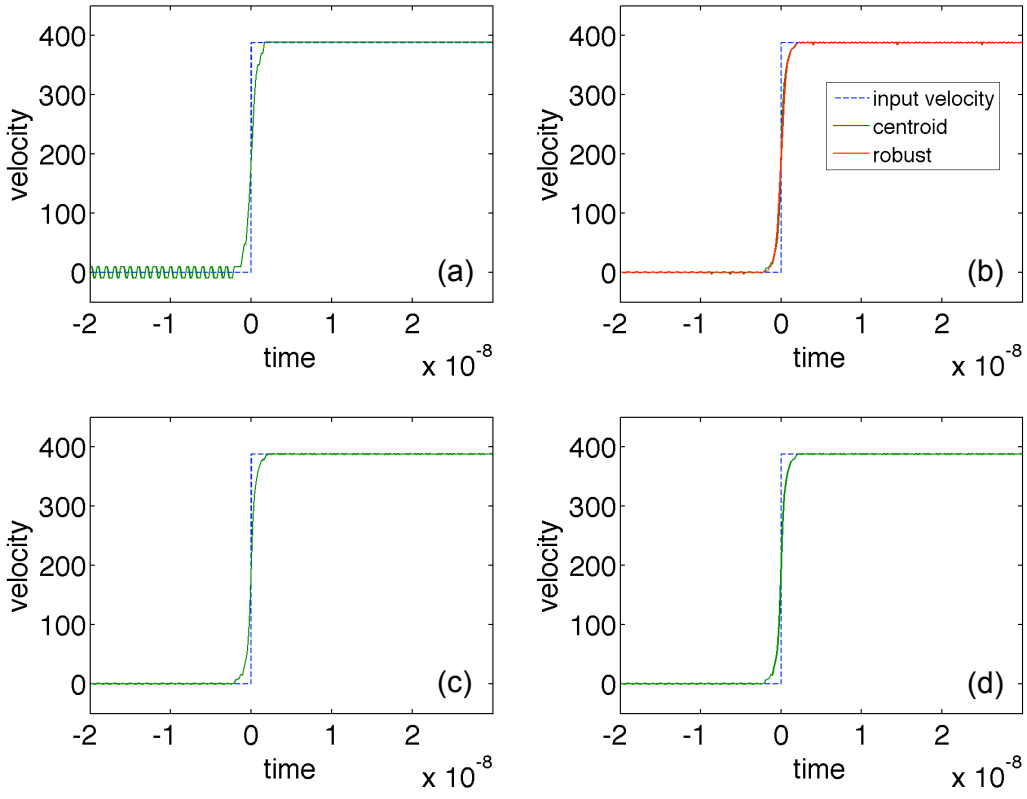

Figure 4.12. Example B-1 extracted velocities from peak finding methods (solid lines) vs input velocity (dashed line): (a) maximum, (b) centroid and robust, (c) parabola, and (d) Gaussian.

- Peak location method: maximum.

The Analysis Screen should now look like Fig. 4.14. Under the Data menu, select the Export history action to save the extracted velocity profile. A comparison of the various peak finding methods is shown in Fig. 4.15, where in each case only the "Peak location method" parameter was changed. The maximum method gives a correct final velocity, but has an incorrect negative initial velocity and discrete steps within the ramp profile. The centroid, robust, parabola, and Gaussian methods all reduce the discreteness of the ramp profile, and have the correct initial zero velocity. Finally, the oscillations at the foot of the ramp observed in the earlier standard PDV example are eliminated in the frequency-conversion PDV example.

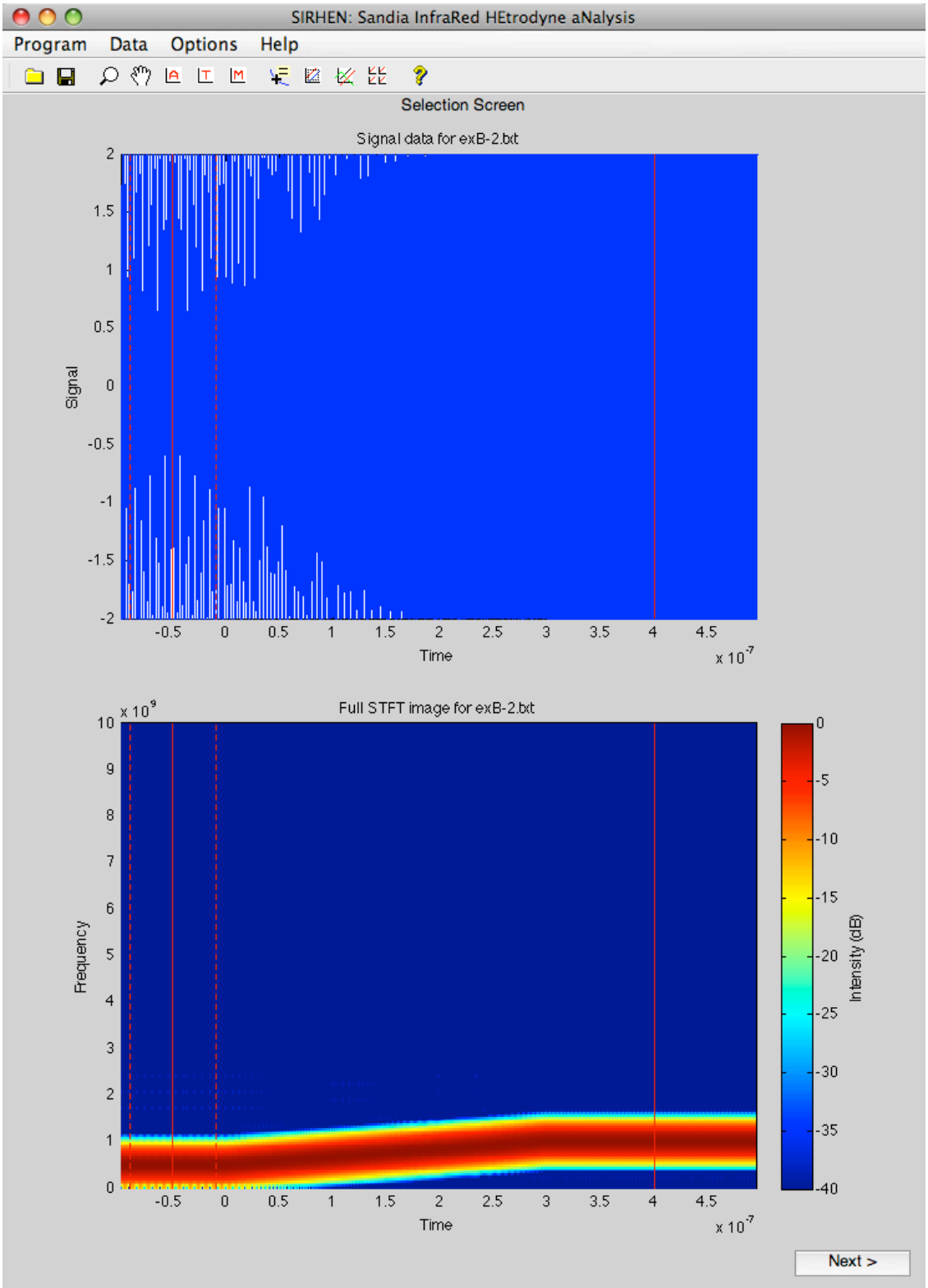

Figure 4.13. Example B-2 Selection Screen.

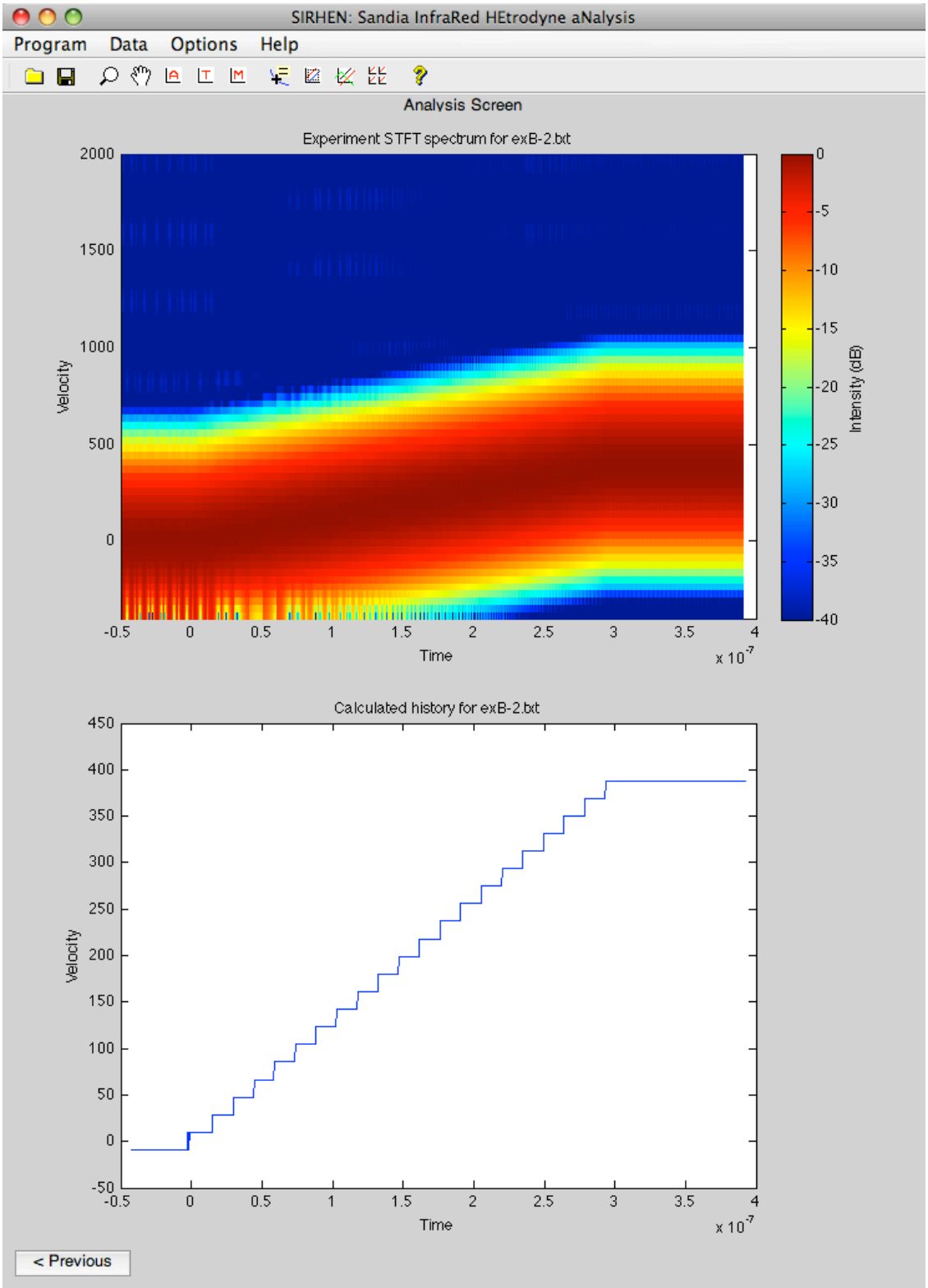

Figure 4.14. Example B-2 Analysis Screen.

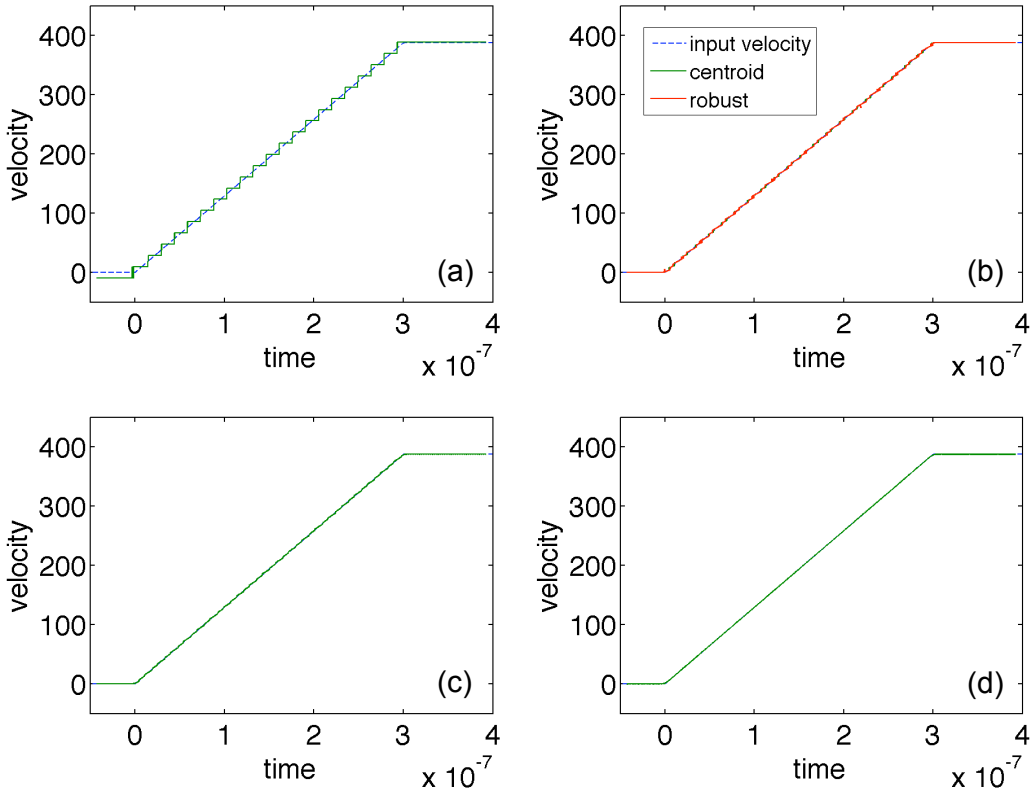

Figure 4.15. Example B-2 extracted velocities from peak finding methods (solid lines) vs input velocity (dashed line): (a) maximum, (b) centroid and robust, (c) parabola, and (d) Gaussian.

Chapter 5

Summary

5.1 Program features

The SIRHEN program reduces a photonic Doppler velocimetry (PDV) signal into a velocity history. SIRHEN accepts a single data file in either text or binary format. Users may specify a reference time range (over which an underlying beat frequency is characterized) and an experiment range (over which the analysis is performed). A short-time Fourier transform analysis is used to generate the frequency spectrum of the PDV signal. The velocity history is extracted from the frequency spectrum using a peak finding method. Data generated by SIRHEN can be exported to a text file for post-processing or saved in various graphical formats.

5.2 Future releases

Users should contact Tommy Ao (tao@sandia.gov) and/or Dan Dolan (dhdolan@sandia.gov) with questions, bug reports, and feature requests. Bug fixes will be made as necessary and will be distributed by email. No scheduled updates to SIRHEN is planned at this time, but new releases will be considered based on users' feedback.

References

[1] L. M. Barker and R. E. Hollenbach, "Laser interferometer for measuring high velocities of any reflecting surface," *Journal of Applied Physics*, vol. 43, p. 4669, November 1972.

[2] C. F. McMillan, D. R. Goosman, N. L. Parker, L. L. Steinmetz, H. H. Chau, T. Huen, R. K. Whipkey, and S. J. Perry, "Velocimetry of fast surfaces using Fabry-Perot interferometry," *Review of Scientific Instruments*, vol. 59, p. 1, 1988.

[3] O. T. Strand, D. R. Goosman, C. Martinez, and T. L. Whitworth, "Compact system for high-speed velocimetry using hetrodyne techniques," *Review of Scientific Instruments*, vol. 77, p. 083108, August 2006.

[4] B. J. Jensen, D. B. Holtkamp, P. A. Rigg, and D. H. Dolan, "Accuracy limits and window corrections for photon Doppler velocimetry," *Journal of Applied Physics*, vol. 101, p. 013523, August 2007.

[5] W. F. Hemsing, "Velocity sensing interferometer (VISAR) modification," *Review of Scientific Instruments*, vol. 50, no. 1, pp. 73–78, 1979.

[6] D. H. Dolan, "THRIVE: a data reduction program for three-phase PDV/PDI and VISAR meaurements," Sandia report SAND2008-3871, Sandia National Laboratories, June 2008.

[7] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Prentice Hall, 1989.

[8] F. Harris, "On the use of windows for hamonic analysis with the discrete Fourier transform," in *Proceedings of the IEEE*, vol. 66, pp. 51–83, IEEE, January 1978.

DISTRIBUTION:

1	MS	0425	S. C. Jones, 545
1	MS	0826	W. M. Trott, 1512
1	MS	0836	M. R. Baer, 1500
10	MS	1106	T. Ao, 1646
5	MS	1133	M. U. Anderson, 5916
5	MS	1133	J. Podsednik, 5916
1	MS	1157	K. J. Fleming, 54341
1	MS	1186	M. Herrmann, 1640
1	MS	1189	M. P. Desjarlais, 1640
1	MS	1193	R. G. Hacking, 16561
1	MS	1193	S. L. Payne, 16561
1	MS	1195	J. R. Asay, 1646
1	MS	1195	C. S. Alexander, 1646
1	MS	1195	J.-P. Davis, 1646
5	MS	1195	D. H. Dolan, 1646
1	MS	1195	D. G. Dalton, 1646
1	MS	1195	M. D. Furnish, 1646
1	MS	1195	R. J. Hickman, 1646
1	MS	1195	M. D. Knudson, 1646
1	MS	1195	B. V. Oliver, 1656
1	MS	1195	S. Root, 1646
1	MS	1195	W. D. Reinhart, 1646
1	MS	1195	J. L. Wise, 1646
1	MS	1205	C. A. Hall, 5902
1	MS	1454	M. D. Willis, 2552
1	MS	9042	T. J. Vogler, 8246
1	MS	0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories