NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

FEATURE EXTRACTION OF INTRA-PULSE MODULATED RADAR SIGNALS USING TIME-FREQUENCY ANALYSIS

by

Ioannis Moraitakis

September 1999

Thesis Advisor:	Monique P. Fargues
Co-Advisor:	Ralph D. Hippenstiel
Second Reader:	Jovan Lebaric

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1999

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
FEATURE EXTRACTION OF INTRA-PULSE MODULATED RADAR SIGNALS USING TIME-FREQUENCY ANALYSIS

5. FUNDING NUMBERS

6. AUTHOR(S) Ioannis Moraitakis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis applies time-frequency transformations to radar signals. Specifically, it considers the feasibility of applying time-frequency transformations to extract the intra-pulse modulation parameters of radar signals. In this work, we consider radar signals with analog pulse compression; specifically linear or hyperbolic intra-pulse modulation. Several time-frequency transformations are investigated to identify which one gives the most accurate image representation for signals in noisy environments. Next, image processing techniques are applied in conjunction with an adaptive curve fitting method, for the hyperbolic modulation scheme, to extract the parameters of the frequency equation. Results show that for the linear chip case the frequency equation can be estimated with small error down to SNR equal to –10dB. The proposed method for the hyperbolic chirp modulation is less immune to noise degradation and it can be used down to SNR level equal to 2dB.

14. SUBJECT TERMS
Time-frequency representations, pulse compression, radon transform.

15. NUMBER OF PAGES
117

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFI- CATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

THIS PAGE INTENTIONALLY LEFT BLANK

FEATURE EXTRACTION OF INTRA-PULSE MODULATED RADAR SIGNALS USING TIME-FREQUENCY ANALYSIS

Ioannis Moraitakis
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

September 1999

Author: ______________________
Ioannis Moraitakis

Approved by: ______________________
Monique P. Fargues, Thesis Advisor

Ralph D. Hippenstiel
Ralph D. Hippenstiel, Co-Advisor

____
Jovan Lebaric, Second Reader

____
Jeffrey B. Knorr, Chairman
Department of Electrical andComputer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis applies time-frequency transformations to radar signals. Specifically, it considers the feasibility of applying time-frequency transformations to extract the intra-pulse modulation parameters of radar signals. In this work, we consider radar signals with analog pulse compression; specifically linear or hyperbolic intra-pulse modulation. Several time-frequency transformations are investigated to identify which one gives the most accurate image representation for signals in noisy environments. Next, image processing techniques are applied in conjunction with an adaptive curve fitting method, for the hyperbolic modulation scheme, to extract the parameters of the frequency equation. Results show that for the linear chip case the frequency equation can be estimated with small error down to SNR equal to –10dB. The proposed method for the hyperbolic chirp modulation is less immune to noise degradation and it can be used down to SNR level equal to 2dB.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank my thesis advisors Prof. Monique P. Fargues, Prof. Ralph Hippenstiel and Prof. Jovan Lebaric for their help and patience. Special thanks to Prof. Ioannis Markos Roussos from the Hamline University of Saint Paul, Minnesota, for his help with Euclidean geometry.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The analysis of a signal in both the time and frequency domain gives a more complete description than that provided by individual analysis in either domain. This advantage was recognized a long time ago but the complexity of the joint time-frequency algorithms made their implementations an extremely slow operation due to computing power limitations. However, as computers become more powerful, joint time-frequency analysis may be applied in a larger number of real-world problems.

A. OBJECTIVE

This thesis investigates the application of time-frequency transformations to the extraction of intra-pulse modulation parameters from radar signals. This information can be a useful tool to identify the specific radar type and help with jamming techniques, if so desired. Intra-pulse modulation parameters are usually extracted using hardware schemes which are well suited to extract the instantaneous frequency at high SNR levels.

Our study considers the problem from a different angle and investigates the application of time-frequency and a basic image processing technique to extract the information. Joint time-frequency transformations can be implemented inexpensively and give accurate results in medium to high SNR levels. We consider radar signals with analog pulse compression; specifically linear or hyperbolic intra-pulse modulation. Several time-frequency transformations are investigated to identify that which gives the most accurate image representation for signals in noisy environments. Next, image processing techniques are applied used in conjunction with an adaptive curve fitting method, to extract the parameters of the frequency equation. The type of modulation, as well as, the start and stop time of the pulse under investigation was assumed to be known.

B. THESIS ORGANIZATION

Radar signals with large time bandwidth products are introduced in chapter II. Chapter III presents an overview of the time frequency methods considered. Chapter IV discusses the basic idea behind the Radon transform and its application in detecting line parameters from a noisy image. Chapters V and VI present the complete methods and the simulations conducted to test the schemes derived to extract the intra-pulse modulation parameters for linear and hyperbolic modulation. Finally, conclusions and recommendations for further research are presented in chapter VII.

II. RADAR SIGNALS WITH LARGE TIME BANWIDTH PRODUCTS

Ideally we would like a radar to produce good range resolution as well as large detection range. Unfortunately, these requirements are contradictory since the first requires a small pulse period, while the second requires a large pulse duration. The problem can be solved with a technique that uses intra-pulse modulation or, as it is better known, pulse compression. This chapter present the basic ideas behind pulse compression.

A. RESOLUTION

A very important aspect of the radar is its resolution properties. By this we mean the ability to detect multiple targets that are close together. Generally, there are four types of resolution:

- Range resolution
- Horizontal (azimuth) cross-range resolution
- Vertical (elevation) cross-range resolution
- Doppler frequency resolution.

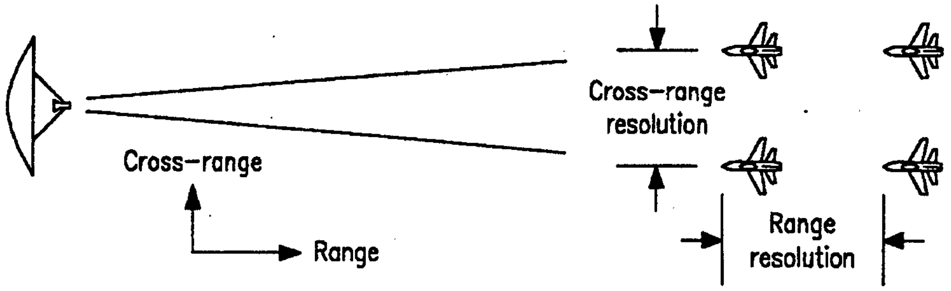

Figure 1: Range and Cross-Range Resolution From [1]

1. Range Resolution

Range resolution is the ability to distinguish between targets, which are at the same angular position but are at different distances to the radar. This resolution is directly proportional to the radar signal bandwidth: the larger the bandwidth is, the better the range becomes. In order to resolve two targets, the basic criterion is that they must be separated by at least the range equivalent of the width of the processed echo pulse as illustrated in Figure 2.

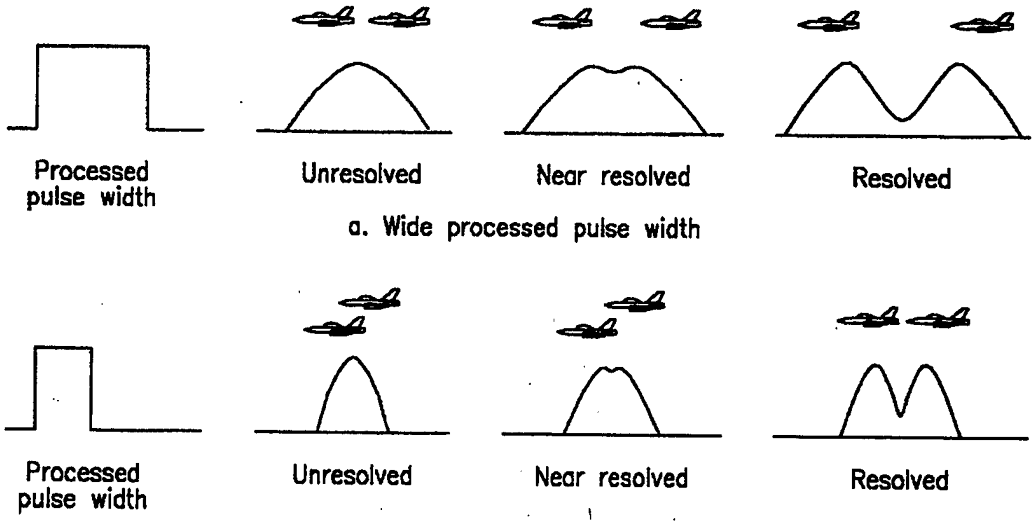

Figure 2: Range Resolution and Processed Pulse Width. From Ref. [1]

2. Horizontal and Vertical Cross-Range Resolution.

Cross-range resolution is defined as the ability to distinguish between targets that have the same distance from the radar but are at a different azimuth (horizontal) and/or different elevation (vertical). This resolution is inversely proportional to the antenna beamwidth. That means the smaller the beamwidth is, the better the cross-range resolution is. The basic constraint is that the targets must be separated by at least the antenna beamwidth. Figure 3 shows the effect of the antenna's beamwidth to the horizontal cross-range resolution.

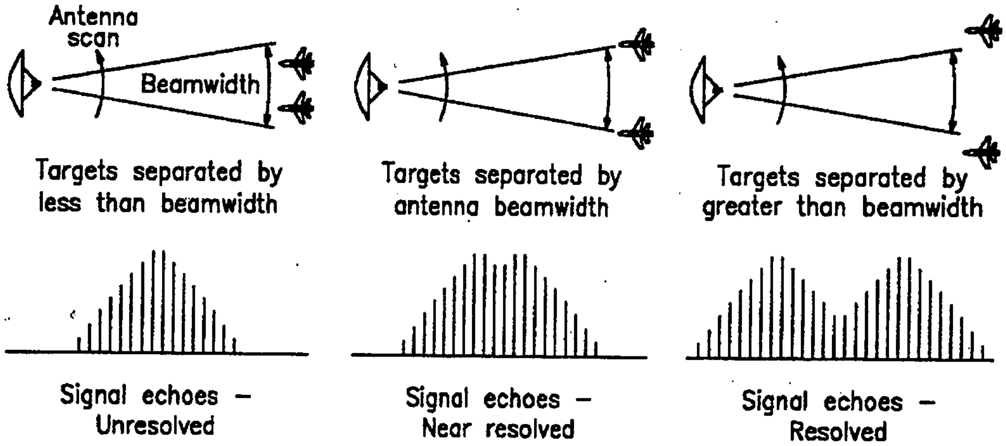
Antenna scan
Beamwidth
Targets separated by less than beamwidth
Targets separated by antenna beamwidth
Targets separated by greater than beamwidth
Signal echoes – Unresolved
Signal echoes – Near resolved
Signal echoes – Resolved

Figure 3: Antenna Beamwidth and Horizontal Cross-Range Resolution. From Ref. [1]

3. Doppler Resolution

Radar doppler resolution is defined as the ability to distinguish between targets that have the same range, azimuth and elevation but have different radial velocities. The basic criterion is that the Doppler frequencies of the targets must be separated by at least one cycle over the time of observation.

B. PULSE COMPRESSION

A radar should have a large a detection range. If we assume that we use the best techniques for antenna and signal processing gain, the only way to increase the range is by increasing the energy of the transmitted signal. There are two ways to increase the energy; by increasing its amplitude (power) of the pulse, or by increasing the pulse duration. Since all transmitters have peak power limitations, the most common choice is to increase the pulse duration, which is in contradiction to the need for narrow pulses to ensure good range resolution. The technique that uses long pulses with high bandwidth is called pulse compression.

1. Analog Pulse Compression

This technique uses modulated radio frequency signals. Let's assume that we have a finite pulse duration radar signal with a constant radio frequency defined as:

$$c(t) = r(t) \cdot s(t), \tag{II-1}$$

where r(t) is a pulse and s(t) is a sinusoid function. Let C(f), R(f) and S(f) be the Fourier transform of the time domain signals, then:

$$C(f) = R(f) * S(f), \tag{II-2}$$

where the operator (*) denotes the convolution operation. Thus, the spectrum of the total signal has the form of a sinc function centered at the frequency of the sinusoid with a null to null bandwidth equal to the double of the reciprocal of the width of the pulse in the time domain, as illustrated in Figure 4.

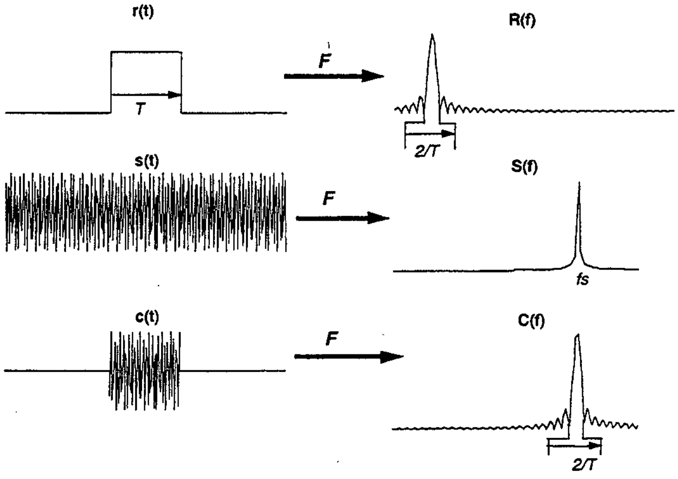

Figure 4: Bandwidth of a CW pulse.

The Fourier transform S(f) is not a delta function when the radio signal $s(t)$ is a modulated sinusoid. This in turns affects the bandwidth of the transmitted signal and gives a null to null bandwidth greater than 2/T, as shown in Figure 5.

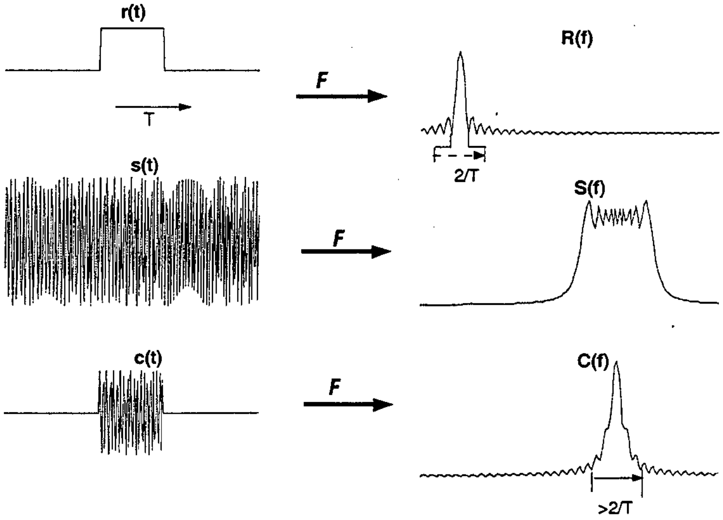

Figure 5: Bandwidth of a linear modulated pulse.

Next we will examine the two most widely used types of pulse intra-modulation; linear and hyperbolic modulations.

a) Linear Modulation

In this case the frequency of the radio signal inside the pulse changes linearly with a slope k from an initial frequency f_0 which leads to:

$$f(t) = f_0 + k \cdot t\,. \tag{II-3}$$

Where f_0 is the initial frequency at t=0.Thus, the transmitted signal c(t) is given as:

$$c(t) = rect\,(\frac{t}{T}) \cdot \exp(\,j2\pi \int fdt\,) = rect\,(\frac{t}{T}) \cdot e^{j2\pi(f_0 t + \frac{k}{2}t^2)}\;. \tag{II-4}$$

b) Hyperbolic Modulation

In this case, the signal inside the pulse is modulated following a hyperbolic equation. The resulting frequency equation is given as:

$$f(t) = \frac{a}{t} + b\,. \tag{II-5}$$

Note that the above time varying frequency f(t) is not finite for t=0. Thus, the above definition is modified by introducing a time shift t_0 to ensure a finite frequency value for t=0, which leads to:

$$f(t) = \frac{a}{t + t_0} + b\,. \tag{II-6}$$

Thus, the instantaneous frequency *f(t)* can be viewed as being a part of the hyperbola defined from the equation (II-5) with an initial value of $f_0 = \frac{a}{t_0} + b$.

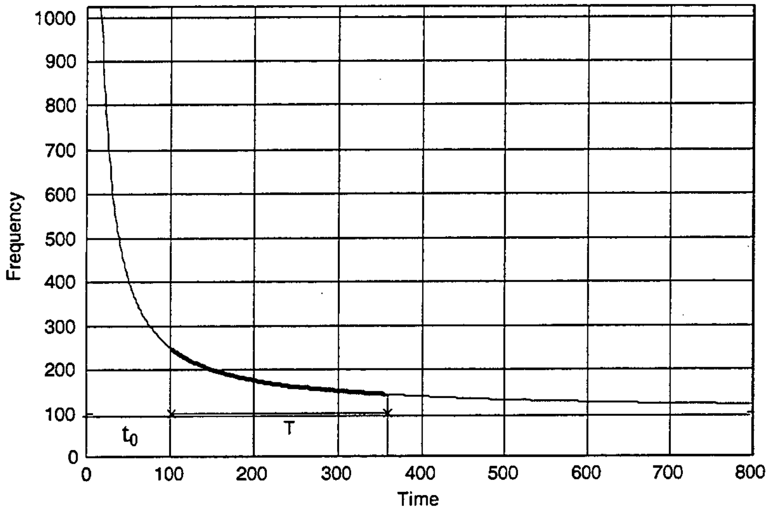

Figure 6: Hyperbolic frequency modulation.

As a result, the equation of the transmitted signal is given by:

$$c(t) = rect\,(\frac{t}{T}) \cdot \exp(\,j2\pi \int fdt\,) = rect\,(\frac{t}{T}) \cdot e^{j2\pi(a \cdot \log(t+t_0)+bt)}. \qquad \text{(II-7)}$$

2. Digital Pulse Compression

Digital pulse compression involves phase-coded waveforms. These are usually biphase modulated sinusoids with the two possible phases to be at 0° and 180°. The overall digital waveform consists of an array of N subpulses, each one with an initial phase of 0° or 180° and a time length τ_S. The width of the total waveform is τ_E. Figure 7 shows a biphase coded signal using a 13-bit biphase sequence. In this figure a "+" corresponds to a subpulse with an initial phase 0° and a "-" corresponds to a subpulse with an initial phase 180°.

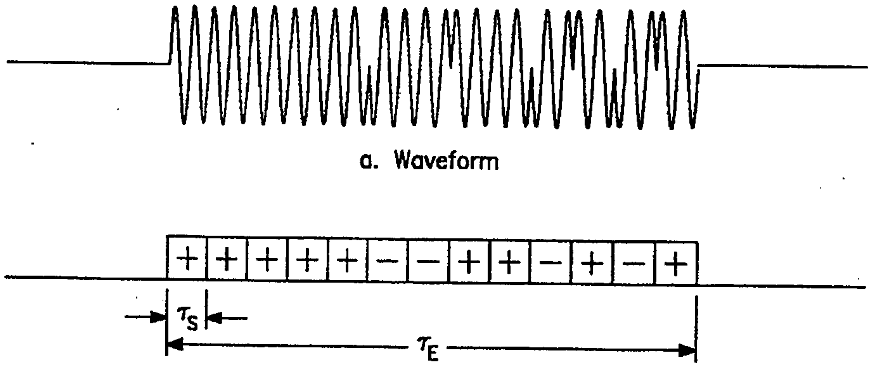

Figure 7: Phase coded waveform. From Ref.[1]

The codes that are used in the phase coded waveforms should have a delta-like autocorrelation function, to allow pulse compression to take place without windowing the signal. The two most widely used codes are the Barker codes and the Pseudorandom codes [1].

III. TIME-FREQUENCY REPRESENTATIONS

Generally, a signal can be represented using different types of decompositions. The two most widely used representations are the time domain and the frequency domain representation. The first shows how the amplitude of the signal changes with respect to time while the second shows how often these changes occur. These representations are uniquely related to each other via the Fourier transform.

A. FOURIER ANALYSIS

1. Fourier Series

The basic idea behind the Fourier decomposition is to express a periodic signal as a weighted sum of sinusoids. Let's assume that we have a signal x(t) with period T. Provided that it satisfies the Dirichlet's conditions, then it can represented as follows:

$$x(t) = \sum_{k=-\infty}^{k=\infty} a_k \cdot e^{jk(2\pi/T)t}, \qquad \text{(III-1)}$$

where

$$a_k = \frac{1}{T}\int_T x(t) \cdot e^{-jk(2\pi/T)t} dt . \qquad \text{(III-2)}$$

The frequency 1/T is called the fundamental frequency. The equation (III-1) gives the general case where the type of basis functions are complex exponentials.

2. Fourier Transform

By analogy with equation (III-2), the Fourier transform *X(f)* of a continuous signal x(t) is given by:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt . \tag{III-3}$$

The plot of the magnitude squared of the Fourier transform is proportional to the spectrum of the signal x(t). The original signal can be recovered from its Fourier transform with the inverse Fourier transform:

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{-j2\pi ft} df . \tag{III-4}$$

The fast discrete version of Fourier transform revolutionized the signal processing area. However, there exist many signals which cannot be described accurately with this transformation. Although the spectrum indicates the frequencies and intensities of the signal, it provides no information regarding the variations of the signal characteristics as a function of time. Thus, the Fourier transform does not analyze non stationary signals very accurately. For example, consider two signals $x_1(t)$ and $x_2(t)$, observed during the same time period T, as shown in Figure 8. The first one is the sum of two pure tones while the second one consists of the first tone for the first half time and then switches to the second tone for the rest of its duration. Their Fourier transform magnitudes look almost identical, however they are not exactly the same since the Fourier transform is a reversible operation. Figure 8 shows that it is not possible to extract specific information regarding the frequency variations as a function of time. Unfortunately, most signals in real life are not stationary, such as for example speech, music, vibration signals and many more. Analyzing non-stationary signals requires a tool which allows to represent the variations of the frequency content as a function of time; i.e., a joint time-frequency representation.

Several techniques have been developed for this type of representation. In this thesis, we will deal with two major classes: atomic decompositions and energy distributions.

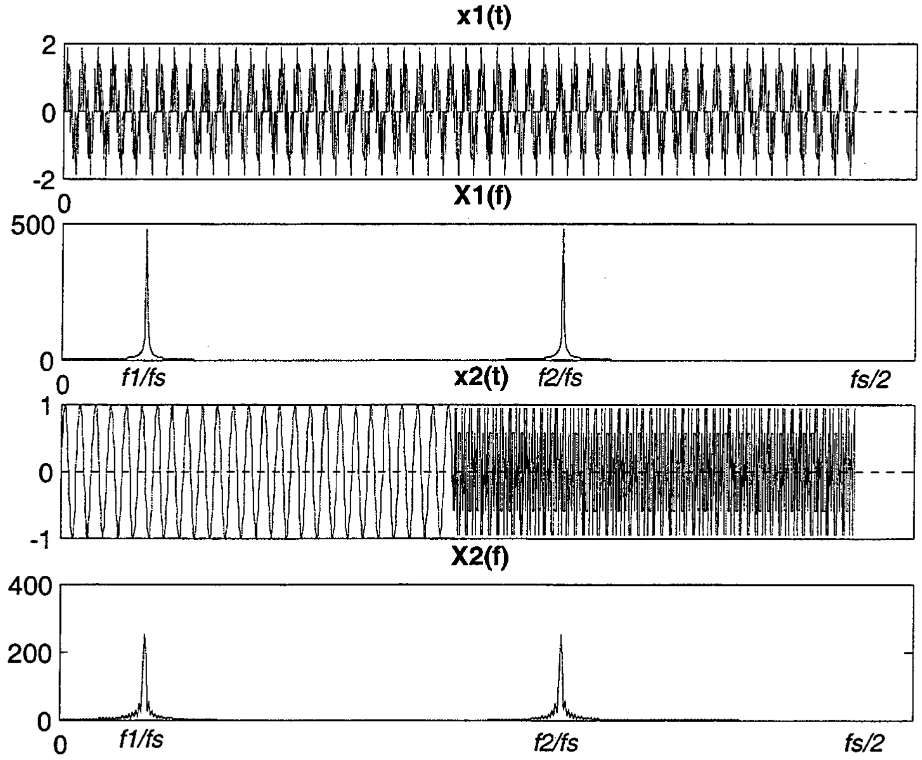

Figure 8: Fourier transform of two signals.

B. ATOMIC DECOMPOSITIONS

The Fourier transform can be considered as the projection of a signal into an infinite set of sinusoids. The problem with sinusoids is that they are not localized in time, although they are perfectly localized in frequency. The basic idea behind the atomic decomposition is to decompose a signal into a set of atoms (i.e. functions) that are well localized both in time and in frequency. These types of decompositions are called atomic decompositions or linear time-frequency representations.

1. Short-Time Fourier Transform

This method is an extension of the Fourier transform, where localization in time is introduced by windowing the sinusoidal functions used in the decomposition. Thus, the Sort-Time Fourier Transform (STFT) is given by:

$$S(\tau, f) = \int_{-\infty}^{\infty} x(t) \cdot g^{*}(t-\tau) \cdot e^{-j2\pi f t} dt \,, \qquad \text{(III-5)}$$

where $g(t)$ is the window function. Note that the STFT is an invertible operation, provided that the window is of finite energy, i.e.,

$$x(t) = \frac{1}{E_h} \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(\tau, f) \cdot g(t-\tau) \cdot e^{j2\pi\tau f} d\tau \cdot df \,, \qquad \text{(III-6)}$$

where $E_h = \int_{-\infty}^{\infty} |g(t)|^2 dt$. The STFT operation can be considered as a set of successive Fourier transforms applied to windowed segments of the signal, as illustrated in Figure 9. Thus, g(t) can be viewed as a sliding window which allows for segmentation of the original signal.

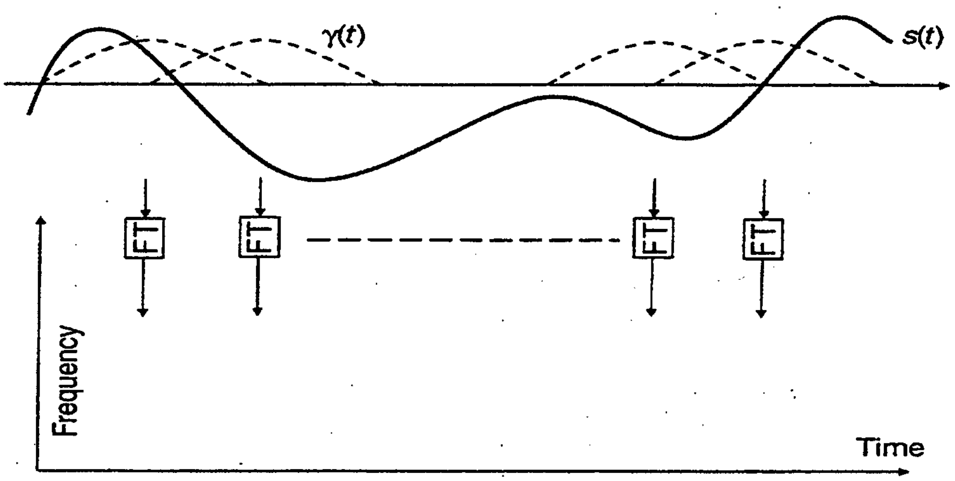

Figure 9: Short-Time Fourier Transform. From Ref [2]

The STFT of *x(t)* can also be seen as the expansion of the signal into a set of atomic functions of the form:

$$a_{\tau,f}(t) = g(t-\tau)\cdot e^{j2\pi ft}\,. \qquad \text{(III-7)}$$

The above function can be recognized as a windowed complex sinusoid, where the window type and duration affects the resolution of the STFT in time and in frequency. The plot of the squared magnitude of the STFT is called the spectrogram.

2. Gabor Expansion

In 1946, Dennis Gabor introduced the decomposition of a signal into a weighted sum of functions that are both localized in time and frequency. The Gabor expansion is defined as:

$$x(t) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} C_{m,n}\cdot h_{m,n}(t)\,, \qquad \text{(III-8)}$$

where:

$$h_{m,n}(t) = h(t - mT) \cdot \exp(jn\Omega t)\,. \qquad \text{(III-9)}$$

The coefficients $C_{m,n}$ are called the Gabor coefficients, while T and Ω are the time and frequency sampling steps. Initially, Gabor selected h(t) to be Gaussian, since it is optimally concentrated in the joint time-frequency domain. However, h(t) can be any function. Comparing equations (III-8) and (III-6) shows that the Gabor expansion is similar to the discrete version of the inverse STFT. In fact, the Gabor expansion is the generalization of the discrete inverse STFT which is used as a fast algorithm for its computation [2,9].

3. Wavelet Analysis

a) Continuous wavelet transform

We saw that the STFT of x(t) is the decomposition of the signal into a set of atoms of the form (III-7) which are two-parameter functions of a complex sinusoid. The continuous wavelet transform is a more general case, where the basis function is not a complex sinusoid but a function with specific properties of the form:

$$\Psi_{a,\tau}(t) = \Psi\left(\frac{t-\tau}{a}\right). \qquad \text{(III-10)}$$

The parameters a and τ are the scale and translation factor, respectively. The function $\Psi(t)$ is called the mother wavelet. This basis function must meet two important criteria; 1) it must have finite duration, and 2) it must have a zero average value. Some of the most widely used wavelets are the Haar, Daubechies, Coiflet, Symmlet which are shown in Figure 10.

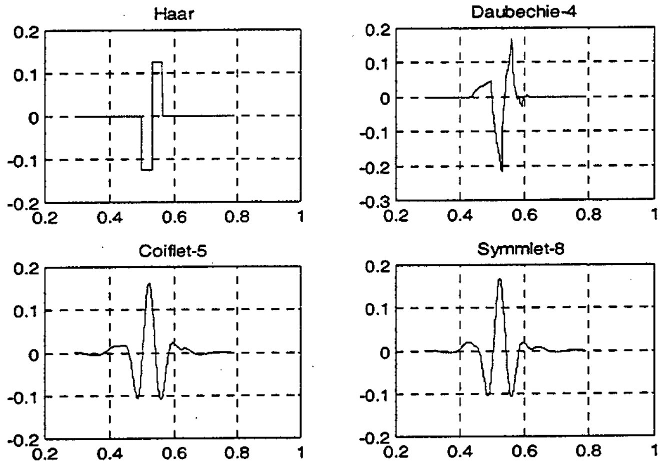

Figure 10: Different types of wavelets. From [12]

The continuous wavelet transform (CWT) of x(t) is defined as the integral of the signal multiplied by scaled and shifted versions of the mother wavelet function ψ(t) [6]:

$$C(\tau,a)=\frac{1}{\sqrt{\alpha}}\cdot\int_{-\infty}^{\infty}x(t)\cdot\Psi^{*}(\frac{t-\tau}{a})dt\,, \qquad \text{(III-11)}$$

where the factor $\frac{1}{\sqrt{\alpha}}$ normalizes the transformation. Comparing equation (III-11) with equation (III-5) shows that we can relate the scale (or dilation) factor *a* to the frequency *f*. The difference in the CWT is that the time and frequency resolution are also controlled by the factor *a* instead of the window function only, as is the case for the STFT. Specifically, small values of *a* mean that the wavelet is contracted in time, thus expanded in frequency. Therefore, the time resolution is good while the frequency resolution is poor. Similarly the wavelet is expanded in time and contracted in frequency when a takes large values. In this case, the time resolution becomes poor as the frequency resolution improves. This unique characteristic of the wavelet analysis is known as the multiresolution capability in the time-frequency plane. Figure 11 shows the tiling of the time-frequency plane for the wavelet analysis and for the STFT.

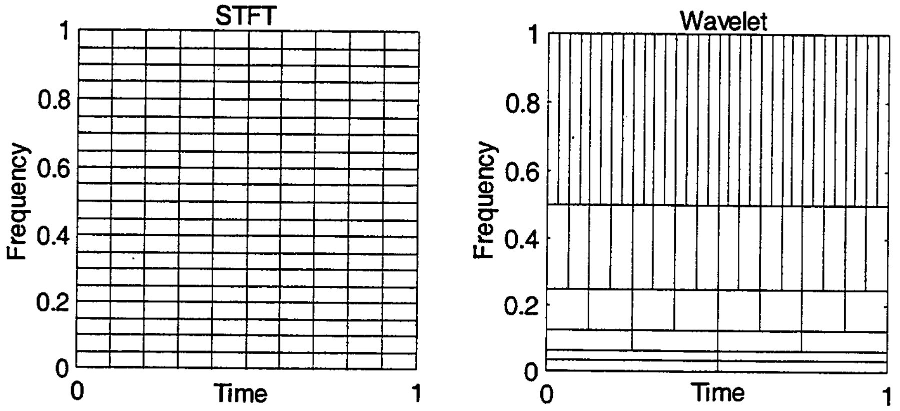

Figure 11: STFT vs. Wavelet time-frequency resolution.

Equation (III-11) can also be written in the form:

$$C(a,\tau) = x(t) * \bar{\Psi}_a(t,\tau), \qquad \text{(III-12)}$$

where $*$ denotes convolution and

$$\bar{\Psi}_a(t) = \frac{1}{\sqrt{a}} \Psi^*(\frac{-t}{a}). \qquad \text{(III-13)}$$

It can be shown that $\bar{\Psi}_a(t)$ is the impulse response of a band-pass filter [4]. Thus, equation (III-12) shows that the CWT of x(t) can be computed by filtering the signal through a series of band-pass proportional filters. The plot of the square magnitude of *C(a, τ)* is called the scalogram, by analogy with the STFT spectrogram. The CWT is a reversible operation, which means that the original signal x(t) can be derived from its transform *C(a, τ)* by [4]:

$$x(t) = \frac{1}{q_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} C(\tau, a) \cdot \frac{1}{a^2 \sqrt{a}} \cdot \psi(\frac{t-\tau}{a}) d\tau da \,, \qquad \text{(III-14)}$$

where

$$q_\psi = \int_0^{+\infty} \frac{|\hat{\Psi}(\omega)|^2}{\omega} d\omega \,, \qquad \text{(III-15)}$$

and $\hat{\Psi}(\omega)$ is the Fourier transform of $\Psi(t)$. Equation (III-14) implies that *C(a, τ)* needs to be known for the whole range of *a* in $[0, \infty]$, in order to recover the signal in time domain from the CWT. When the CWT is known for values *a<a₀* only, a scaling function which contains the information for the range $a \geq a_0$ needs to be introduced. The scaling function ϕ(t) is the impulse response of a lowpass filter. If we define:

$$\phi_s(t) = \frac{1}{\sqrt{s}} \cdot \phi(\frac{t}{s}), \qquad \text{(III-16)}$$

then, equation (III-14) can be written as:

$$x(t) = \frac{1}{q_\psi} \int_0^{a_0} C(a,\tau) * \Psi_a(t) \frac{da}{a^2} + \frac{1}{q_\psi \cdot a_0} \cdot L(a_0,\tau) * \phi_{a_0}(t), \qquad \text{(III-17)}$$

where the term $L(a,\tau)$ is called the low frequency approximation of *x(t)* at scale *a* and is given by:

$$L(a,\tau) = x(t) * \phi_a^*(t,-\tau). \qquad \text{(III-18)}$$

b) Discrete wavelet transform

The discrete wavelet transform (DWT) can be derived from the continuous wavelet transform expression using the discrete versions of a, τ and t. Specifically, the DWT of a discrete signal x(n) is defined as:

$$C(a,b) = \sum_{n=1}^{N} \frac{1}{\sqrt{a}} x(n) \Psi^* (\frac{n-b}{a}). \qquad \text{(III-19)}$$

In practice we further restrict the factors *a* and *b* to

$$a = 2^j, \;\; b = k \cdot 2^j \;, \qquad \text{(III-20)}$$

where k and j are integers. Using equation (III-20), equation (III-19) can be written as:

$$C_{j,k} = \sum_{n=1}^{N} \frac{1}{\sqrt{2^j}} x(n) \Psi^* (2^{-j} n - k). \qquad \text{(III-21)}$$

Equation (III-21) shows that the DWT is decimated by a factor of two at each successive scale j. Specifically, the maximal values of j and k are $\log_2(N)$, and $N2^{-j}$

for a given value of k, for a signal length equal to N. The signal x(n) can be recovered from the DWT using the discrete version of equation (III-17) which yields [4]:

$$x[n]=\frac{\ln 2}{q_{\psi}}\sum_{j=1}^{j_0}2^{-j}C_{j,k}\otimes\Psi_j(n)+\frac{2^{-j_0}}{q_{\psi}}L_{j_0,k}\otimes\phi_{j_0}(n), \qquad \text{(III-22)}$$

where the operator ⊗ denotes the circular convolution. Equation (III-22) shows that the signal can be decomposed in the coefficients $C_{j,k}$ and $L_{j0,k}$ using a pair of low-pass and high-pass filters. This scheme was first proposed by Mallat and uses the quadrature mirror filters (QMF) theory [4]. Specifically, the signal x(n) is fed into a pair of low-pass (LP) and high-pass (HP) filters which cover the frequency range from 0 to $f_s/2$, as shown in Figure 12. The LP filter output contains the approximation of the signal and corresponds to the first term of equation (III-22), while the output of the HP filter gives the details of the signal and corresponds to the second term of equation (III-22). The filter outputs can be decimated by a factor of two, since each filter covers only half of the original frequency range.

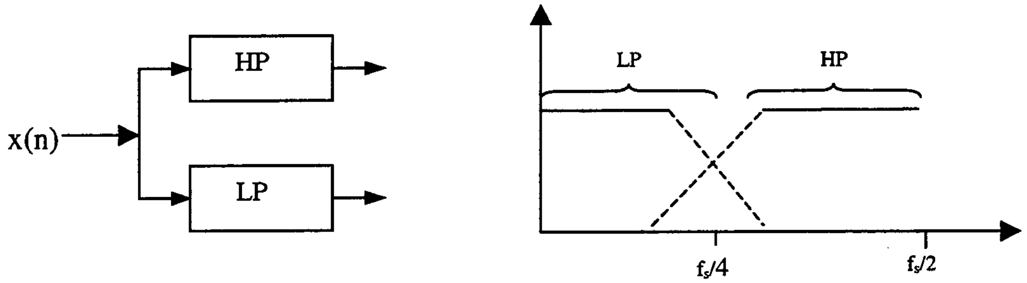

Figure 12: Quadrature mirror filters

This scheme continues at the next level by decomposing the approximation into second stage details and approximations. The whole procedure can continue recursively until the output of the filter reaches the minimum number of samples equal to 1. Thus, the maximum number of decomposition stages is $\log_2(N)$ for a signal of length N. Figure 13 shows a three-stage decomposition tree. The original signal can now be written as a combination of details and approximations of various levels. A few examples are:

- $X= A_1+D_1$
- $X= A_3+D_3+D_2+D_1$.

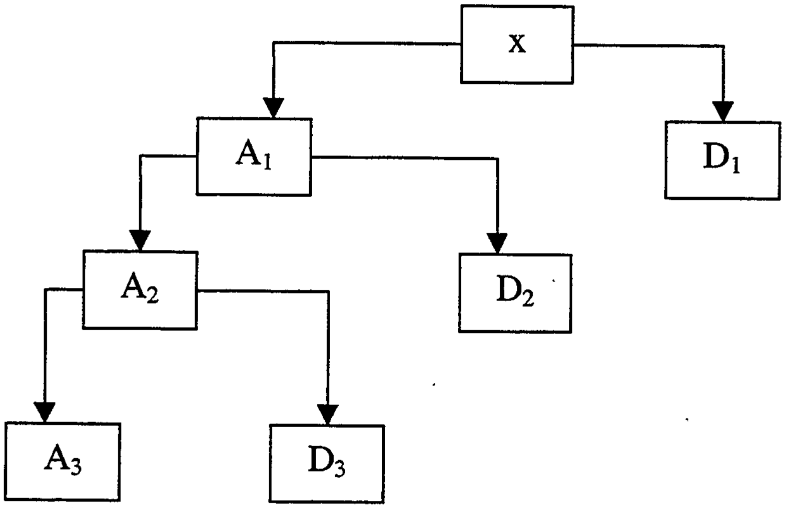

Figure 13: Three-stage wavelet decomposition tree.

Note that the detail D_1 occupies a large band of the overall frequency band when the signal is decomposed using a two-stage decomposition $X=A_2+D_2+D_1$, as shown in Figure 14. Further, note that the region covered by D_1 is not decomposed again at later stages of the decomposition, which prevents from zooming on smaller frequency bands in the region [$f_s/4$, $f_s/2$] . Further decompositions on the HP region are considered in the wavelet packet analysis, which can be viewed as a generalization of the wavelet transform.

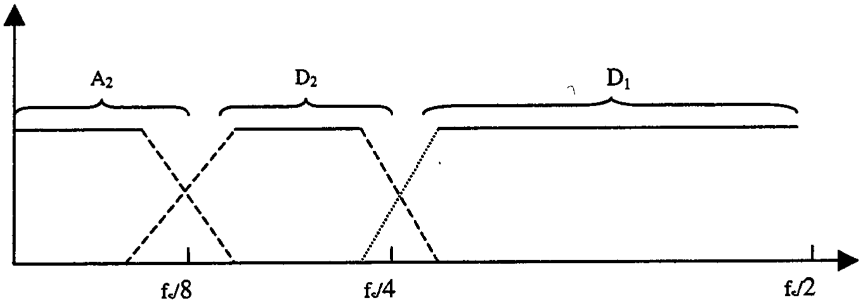

Figure 14: Spectral coverage for a two-stage wavelet decomposition.

c) Wavelet Packet Analysis

The wavelet packet decomposition is an extension of the wavelet transform where both details and approximations are decomposed further to the higher level, thereby allowing flexibility in the decomposition. This scheme leads to $N\log_2N$ possible decompositions for a signal of length N, where any complete level of the tree forms a complete orthogonal basis. Figure 15 shows the decomposition tree of a wavelet packet at depth j=3. A few possible representations of the signal are:

- $X=A_1+AD_2+DD_2$
- $X=AAA_3+DAA_3+DA_2+AD_2+ADD_3+DDD_3$.

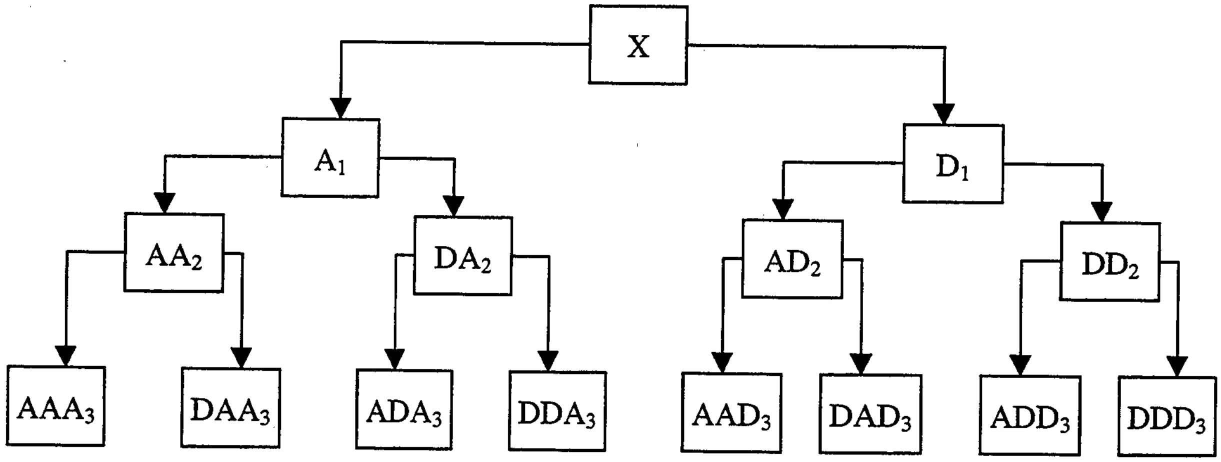

Figure 15:Wavelet packet decomposition at level 3.

4. Cosine Packet Transform

While the wavelet packet (WPT) performs a multi-resolution decomposition by partitioning the frequency axis, the cosine packet transform (CPT) performs a multi-resolution decomposition by partitioning the time axis. As a result, the CPT performs better for narrowband signals. Next, we briefly discuss the discrete cosine transform (DCT) and the local cosine transform (LCT) and show how they are used to perform the CPT.

a) Discrete Cosine Transform IV

The DCT is the inner product of the signal with a windowed cosine (also called blocked cosine). Specifically, the Discrete Cosine Transform-iv (DCT-IV) of a signal x(n) with length N is defined as:

$$Xc(k) = \sqrt{\frac{2}{N}} \cdot \sum_n x(n) \cdot \cos[\frac{\pi(k+1/2)n}{N}], \text{ k: integer.} \qquad \text{(III-23)}$$

Figure 16 shows two blocked cosine functions for N=512 and k=1 and 2.

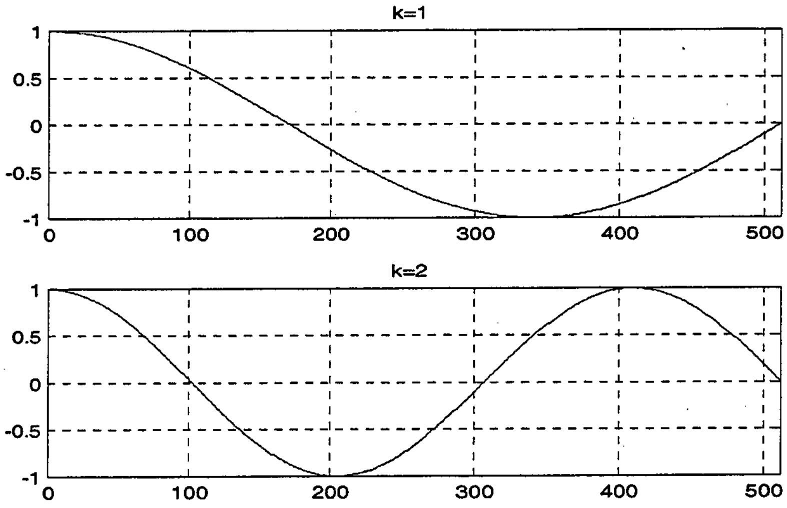

Figure 16: Blocked cosines, k=1 and k=2.

The DCT-iv is closely related to the discrete Fourier transform (DFT) by the following relation:

$$Xc(k) = \sqrt{\frac{2}{N}} \cdot X_f(k) \cdot \exp(\frac{-j \cdot \pi \cdot k}{2N}). \qquad \text{(III-24)}$$

Such a relationship shows that the DCT is a very fast algorithm, as it takes advantage of fast DFT algorithms.

b) Local Cosine Transform

As described earlier, the basis of the DCT is the blocked cosine. Note that the rectangular window used in the DCT may give undesirable sidelobes in the frequency domain. A solution to this problem is to select a Gaussian shaped window to reduce this effect. In practice, the window used has the general form [7] :

$$r(t) = \exp(j \cdot p(t)) \cdot \sin(\varphi(t)) . \qquad \text{(III-25)}$$

The effect of the window, also called bell, is to localize the blocked cosine both in time and in frequency better. Figure 17 shows the local cosine for *p(t)=0* and *φ(t)=π/4[1+sin(πt)].*

c) Cosine packet transform

The cosine packet transform partitions the time axis following the same concept as that used for the frequency axis in the wavelet packet decomposition. The local cosine transform is applied to each time segment of the signal.

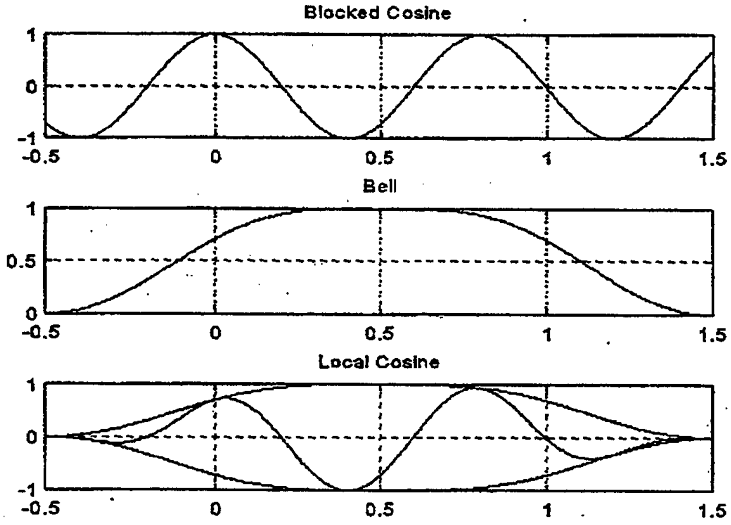

Figure 17: Local cosine. From Ref [4]

The resulting tree structure is the same as that obtained with the wavelet packet. However, the CPT time resolution improves while the frequency resolution worsens, as the decomposition level increases.

5. Best Basis and Matching Pursuit

Recall that we can decompose a discrete signal of length N using wavelets or local cosine packets with the maximum of $j=\log_2 N$ stages. The time frequency atoms that participate in the full decomposition form an orthogonal basis. Wickerhauser and Coifman developed an algorithm named 'Best Basis' which finds a complete set of orthogonal basis functions that minimizes a user-specified information cost criterion [7].

Another type of decomposition called the matching pursuit was proposed by Mallat and Zhang [8]. The matching pursuit offers much more flexibility in the type of decomposition than the WP decomposition does, as it is not restricted to orthonormal basis functions. In this algorithm, the signal is decomposed adaptively into a set of functions, not necessarily orthonormal. Specifically, for given dictionary $D=\{g_\gamma\}_{\gamma\in\Gamma}$, of P vectors (atoms), the algorithm starts by projecting the signal s(*n*) onto each function of the dictionary *D* and computing the residue *RS* . Thus,

$$S=\left\langle S,g_{\gamma_0}\right\rangle g_{\gamma_0}+RS\,. \tag{III-26}$$

At each iteration, the algorithm selects the vector g_{γ_0} with maximum inner product with the signal *S* and smallest $\|RS\|$. At the next iteration, the residue *RS* gets projected into the other functions of the dictionary to determine the second decomposition vector. The procedure continues until the residue is zero or its norm is small enough. The main drawback in this scheme is that the decomposition usually has an error due to the last residue, unless the signal can be decomposed exactly using the given dictionary in a finite number of steps. In addition, the computational load is higher than that of wavelet packet-based decompositions.

C. ENERGY DISTRIBUTIONS

So far we discussed atomic decompositions which lead to linear time-frequency representations. Energy distributions distribute the energy of the signal in both time and frequency and are viewed as quadratic transformations of the signal. The basic idea is that the signal preserves its energy in both time or frequency representation. According to Parseval's relation:

$$E_s = \int_{-\infty}^{\infty} |s(t)|^2 \cdot dt = \int_{-\infty}^{\infty} |S(f)|^2 \cdot df \, . \qquad \text{(III-27)}$$

If we consider the values $|s(t)|^2$ and $|S(f)|^2$ as energy densities, then it would be convenient to find a joint time frequency energy density $\rho_s(t,f)$ such that [9]:

$$E_s = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_s(t,f) \cdot dt \cdot df \, . \qquad \text{(III-28)}$$

This chapter considers one such class of distributions: the Cohen's class. Distributions in this class are covariant by translation in time and frequency [10]. The general expression describing the Cohen's class of energy distributions is given by:

$$C_s(t,f,\varphi) = \iint \int_{-\infty}^{\infty} e^{j2\pi\xi(x-t)} \cdot \varphi(\xi,\tau) \cdot s(x+\tau/2) \cdot s^*(x-\tau/2) \cdot e^{-j2\pi f\tau} \cdot d\xi \cdot dx \cdot d\tau \; , \qquad \text{(III-29)}$$

where $\varphi(\xi,\tau)$ is called the parameterization function.

1. The Wigner-Ville Distribution

The most popular distribution in the Cohen class is the Wigner-Ville distribution, which is defined as:

$$W_s(t,f) = \int_{-\infty}^{\infty} s(t+\tau/2)\cdot s^*(t-\tau/2)\cdot e^{-j2\pi f\tau}d\tau\,, \qquad \text{(III-30)}$$

or equivalently as:

$$W_s(t,f) = \int_{-\infty}^{\infty} S(f+\xi/2)\cdot S^*(f-\xi/2)\cdot e^{-j2\pi\xi t}d\xi\,. \qquad \text{(III-31)}$$

Note that equation (III-30) is derived from equation (III-29) when $\varphi(\xi,\tau)=1$. The Wigner-Ville distribution has many useful properties. For example, it allows for perfect localization of linear chirps, with frequency variation defined as $f(t)=at+f_0$ and described as:

$$s(t) = e^{j2\pi(\frac{a}{2}t^2+f_0 t)}\,. \qquad \text{(III-32)}$$

The WV distribution of a linear chirp is given by:

$$W_s(t,f) = \delta(f-(f_0+\frac{a}{4}t))\,. \qquad \text{(III-33)}$$

Further details on the Wigner-Ville distribution properties may be found in references [2,9]. Note that the main disadvantage of the WVD is the interference terms that appear when the signal is not linearly modulated or more than one signal is present. These interference terms are due to the bilinear nature of the WVD. Figure 18 shows the WVD and the effect of the cross terms on four different signals. The effect is negligible in the noise-free linear chirp scenario only. Cross terms show as a line between the 2 true components when the signal is composed of two parallel chirps.

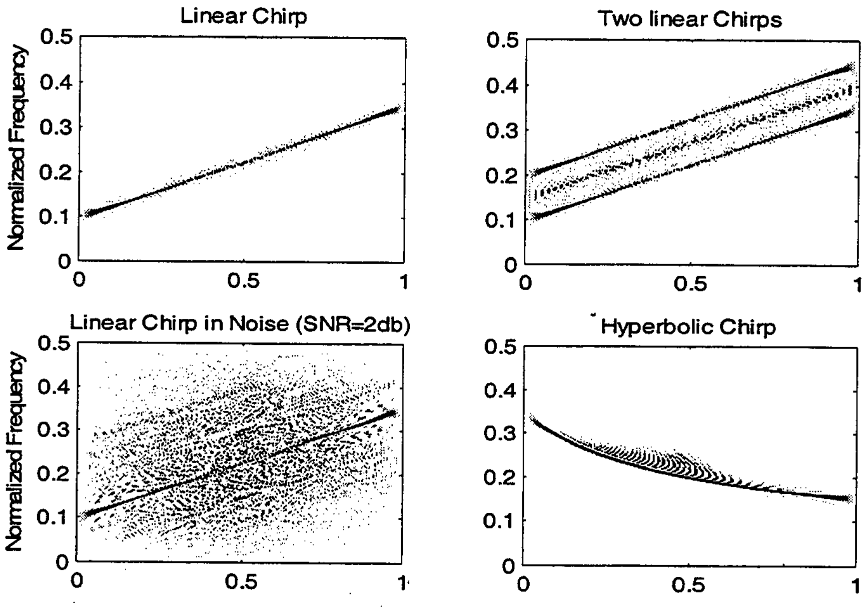

Figure 18: Wigner-Ville distribution,time is normalized over the pulse duration.

Another potential drawback of the WV distribution is the presence of spectral aliasing when the signal is real and sampled near its Nyquist frequency. However, this problem can be avoided by using the analytical version of the signal or by oversampling it by at least a factor of two.

2. Pseudo Wigner-Ville Distribution

Equation (III-31) implies that we must integrate from $\tau = -\infty$ to $\tau = \infty$, which is a problem in practice [9]. To overcome this problem, a windowed version of the WVD called the Pseudo Wigner-Ville Distribution (PWVD) is defined as:

$$PW_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \cdot s(t+\tau/2) \cdot s^*(t-\tau/2) \cdot e^{-j2\pi f\tau} d\tau\,, \qquad \text{(III-34)}$$

or equivalently:

$$PW_s(t,f) = \int_{-\infty}^{\infty} H(f-\xi) \cdot W_s(t,\xi) d\xi\,, \qquad \text{(III-35)}$$

where h(t) is a time window. Note that the interference terms have a strong oscillatory component [2]. Windowing in the time domain actually results in some frequency domain smoothing, thereby reducing the interferences, as shown in Figure 19. However, note that applying a time window to minimize the interference terms results in worsening of the frequency resolution. In addition, the PWVD looses many of the valuable WVD properties [2].

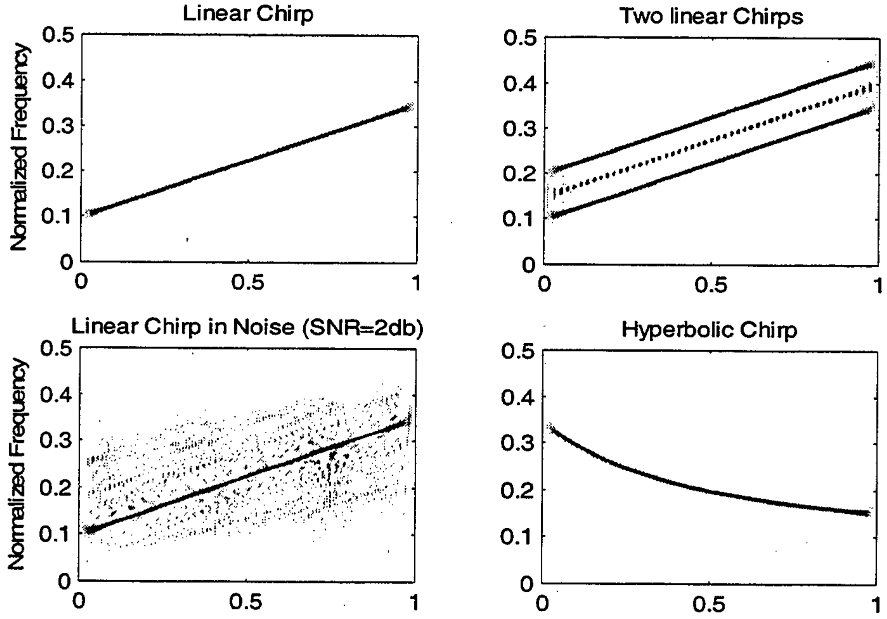

Figure 19: Pseudo Wigner-Ville Distribution, time is normalized over the pulse duration.

3. Smoothed Pseudo Wigner-Ville Distribution

The PWVD provides an attenuation of the interference terms by smoothing in the frequency domain. Interference terms can be further reduced by smoothing in the time domain. The resulting distribution is called the Smoothed Pseudo Wigner-Ville Distribution (SPWVD) and is described as follows:

$$SPW_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \int_{-\infty}^{\infty} g(s-\tau)\cdot s(t+\tau/2)\cdot s^*(t-\tau/2)\cdot e^{-j2\pi f\tau} d\tau, \quad \text{(III-36)}$$

where g(t) is a window in the time domain. Note that the SPWVD becomes the PWVD when $g(t)=\delta(t)$. The drawback of this method is that the frequency resolution further degrades. The effects of the SPWVD are shown in Figure 20 which shows that the cross terms have almost been eliminated while the frequency resolution has been degraded.

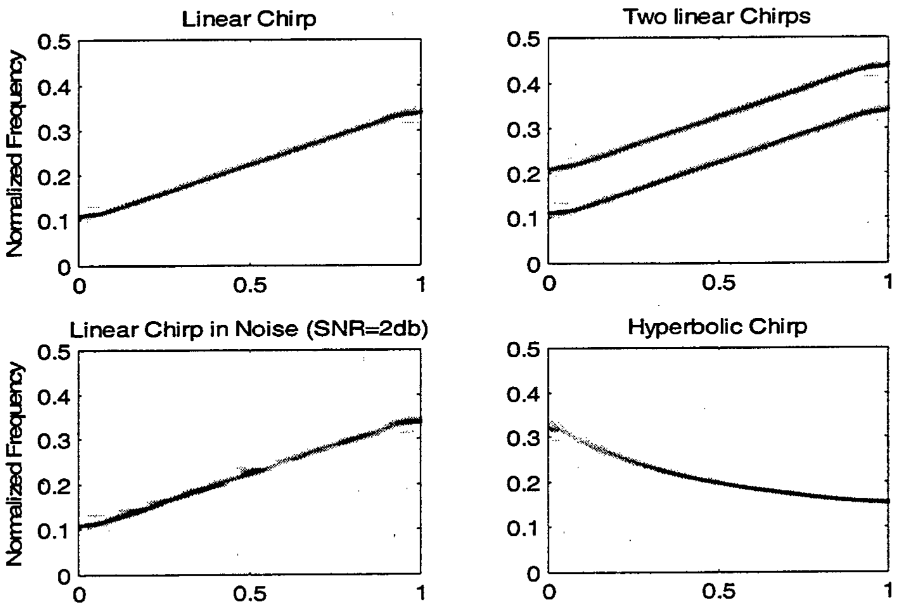

Figure 20: Smoothed Pseudo Wigner-Ville Distribution,time is normalized over the pulse duration

4. The Reassignment Method

We see that the WVD resolution worsens when we try to suppress the interference terms. The reassignment method was introduced in an attempt to improve the interpretation of the transformation [9]. This scheme assumes that the energy distribution in the time-frequency plane resembles a mass distribution and moves each value of the time-frequency plane located at a point (t,f) to another point (t',f'), which is the center of gravity of the energy distribution in the area of (t,f). The result is a very focused representation with high intensity since the value at the point (t',f') is the sum of all the neighboring values. The reassignment method can be applied to most energy distributions as well as to the spectrogram. However, the computational load is quite high. Figure 21 shows the hyperbolic chirp used in Figure 18 to 20 as represented for the reassigned spectrogram distribution, the reassigned PWVD and the reassigned SPWVD. Note the quality of the focusing in all three methods and the lack of cross terms in the PWVD and SPWVD.

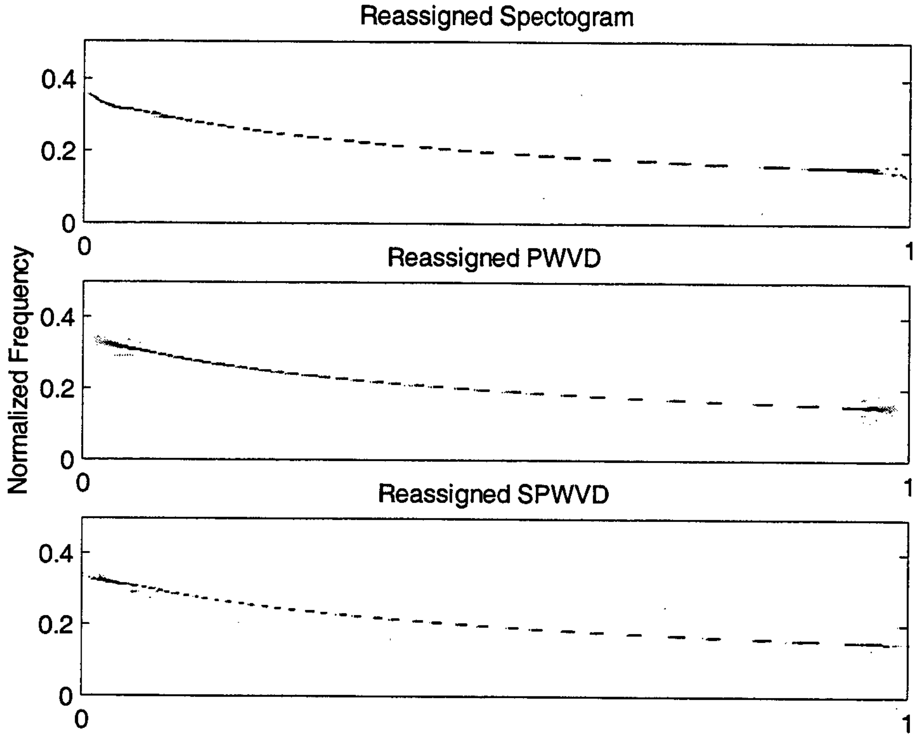

Figure 21: Reassignment method applied to a hyperbolic chirp test signal, time is normalized over the pulse duration.

IV. THE RADON TRANSFORM

Earlier chapter presented several techniques to construct an image representation of a signal in the time-frequency plane. The next step is to extract the information from this image to recover the modulation parameters for both linear and hyperbolic chirps. An important tool in image processing is the Radon transform which is used to extract line information. We will briefly review and then apply this transform to extract linear chirp parameters [13].

A. INTRODUCTION

Let's assume we have an arbitrary function f(x,y) in a subspace of R^2. The two-dimensional Radon transform is defined as the projection or line integral of the function f(x,y) along all possible lines L [13]. Mathematically, the transformation is described as:

$$R = \int_L f(x(s,L), y(s,L))ds\,. \qquad \text{(IV-1)}$$

Recall that the equation of a line in polar coordinates is given by:

$$\rho = x \cdot \cos(\vartheta) + y \cdot \sin(\vartheta)\,, \qquad \text{(IV-2)}$$

where ρ and ϑ represent the distance from the origin and the angle measured counterclockwise from the x axis respectively, as shown in Figure 22. Now suppose we use another coordinate system with axes rotated by the angle ϑ. The new x-axis lies on the line with associated orthogonal direction "s". The two cartesian systems *xoy* and *ρos* are related to each other via the following relation:

$$\begin{bmatrix}\rho \\ s\end{bmatrix} = \begin{bmatrix}\cos\vartheta & \sin\vartheta \\ -\sin\vartheta & \cos\vartheta\end{bmatrix} \cdot \begin{bmatrix}x \\ y\end{bmatrix}. \qquad \text{(IV-3)}$$

Equation (IV-1) can be rewritten as:

$$R(\rho,\vartheta) = \int_{-\infty}^{+\infty} f(\rho\cos\vartheta - s\sin\vartheta, \rho\sin\vartheta + s\cos\vartheta)\cdot ds \qquad \text{(IV-4)}$$

The above equation shows that the Radon transform translates a two-dimensional function of the variables (x,y) to one with variables (ρ,ϑ). Thus, the Radon transform of an image taken at a specific angle ϑ is the projection of the image onto the line which forms an angle ϑ with the x-axis.

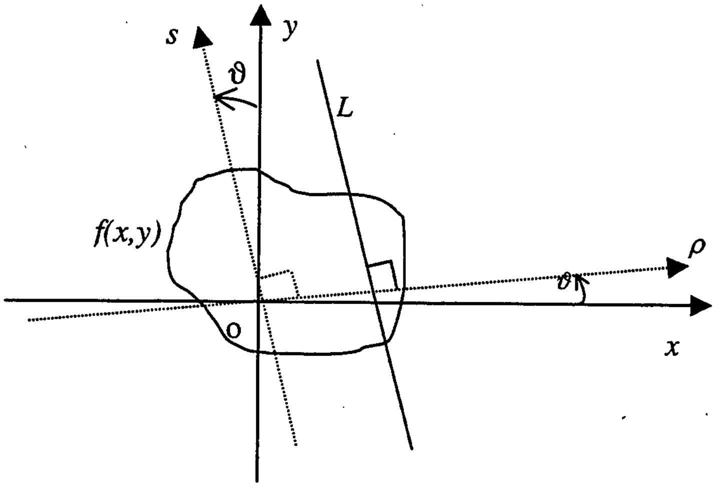

Figure 22: Two- dimensional Radon transform.

The Radon transform of a single line with a slope angle φ for the specific angle $\vartheta = \dfrac{\pi}{2} + \varphi$ is a single point with intensity equal to the sum of the intensities at each point on the line. This property allows detection of lines in an image. In addition, the transform is also robust to noise degradations.

B. LINE PARAMETER IDENTIFICATIONS

Line parameters can be obtained using the Radon transform as follows: Assume the image under investigation contains a line which has an angle φ with the x-axis, as shown in Figure 23. The equation of the line can be defined in terms of its slope *a* and initial offset value *b*.

$$y = a \cdot x + b, \quad a = \tan(\varphi). \tag{IV-5}$$

The Radon transform R(ρ,ϑ) of the image for angles 0° to 180° is maximum when the projection of the line has a minimum area. Thus, it is maximum when the line is perpendicular to the projection line, i.e., when $\vartheta=\varphi+90^o$, which leads to

$$a = \tan(\vartheta - 90^o). \tag{IV-6}$$

Now, if we take the Radon transform of the image for the specific angle $\vartheta=\varphi+90^o$, we can estimate the offset parameter *b* from the position of its maximum value along the axis ρ. This position is indicated in Figure 23 as “C:”. Next, the offset parameter *b* can be computed as:

$$b = \frac{N_y}{2} + \frac{OC - \frac{N_x}{2}\sin(\vartheta - 90^o)}{\cos(\vartheta - 90^o)} = \frac{N_y}{2} + \frac{OC + \frac{N_x}{2}\cos(\vartheta)}{\sin(\vartheta)}, \tag{IV-7}$$

where N_y is the length of the vertical side of the image, while N is the length of the horizontal side (in this case N_x=N_y=256). Note that the distance OC can be positive or negative and is actually negative in Figure 23.

Thus, the equation of the line can be determined by first applying the Radon transform of the image for angles between 0° to 180°, then finding the coordinates of the maximum point $(\vartheta_{max}, \rho_{max})$, as shown in Figure 24, and finally using equations (IV-6) and (IV-7) using $OC=\rho_{max}$ and $\vartheta=\vartheta_{max}$. Note that the accuracy of the estimates depends on the resolution of the image and the size of the angle step selected for the Radon transform. Unfortunately, any attempt to increase the resolution (size) or decrease the angle step results in increasing the computation load.

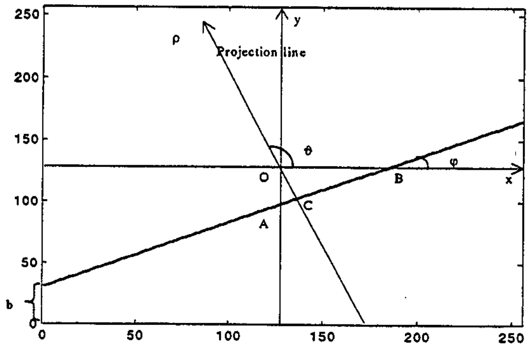

Figure 23: Geometry of the Radon transform for an image containing a single line.

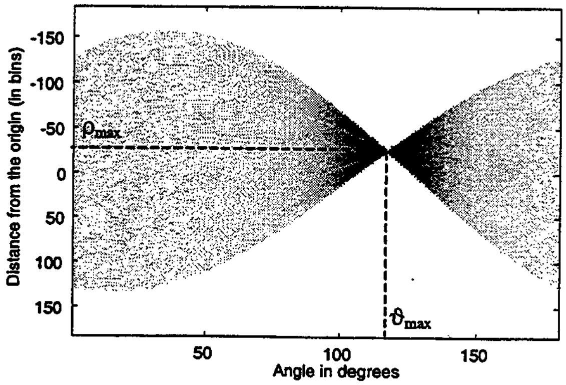

Figure 24: Radon transform for the line shown in Figure 23 for $\vartheta=0^o...180^o$, 1^0 increment.

To minimize the computational load we apply the Radon transform in two stages. First, we scan the image from 0° to 179° in steps of 2°. We determine the angle ϑ_{1m} that corresponds to the column that includes the point with the highest intensity. Next, we scan the image from the angle ϑ_{1m}-1° to ϑ_{1m}+1° in steps of 0.1°. Using this implementation allows the maximum line slope error to decrease to ±0.05° without having to cover the whole range of angles at that rate.

THIS PAGE INTENTIONALLY LEFT BLANK

V. LINEARLY MODULATED CHIRPS

This chapter presents the scheme used to estimate linear chirp parameters. This chapter is divided in four sections. The first part presents the method used to generate the signals. The second section presents the time-frequency transformations considered in this study from which the linear chirp parameters are extracted. The third section describes the image processing technique applied to extract the features from the time frequency image. Finally, the last section compares the results obtained for each time frequency representation considered and their robustness to noise degradations.

A. SIGNAL GENERATION

The radar signals considered in this study are synthetic and consist of a train of several linearly modulated pulses, as shown in Figure 25.

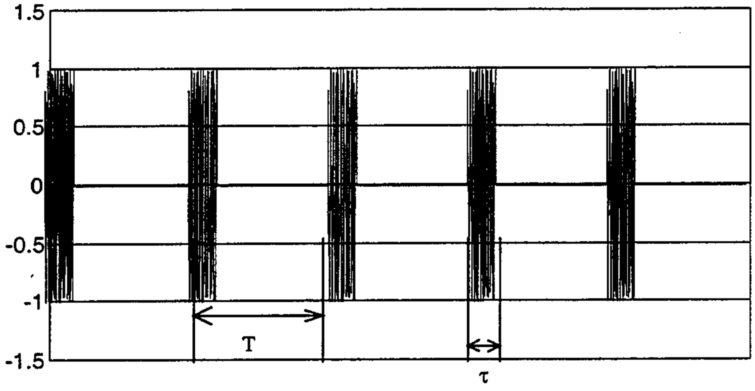

Figure 25: Synthetic radar signal, linear chirp modulation.

First, we assume that we can isolate one single pulse of duration τ. This extraction can easily be done with an energy detector in medium to high SNR levels. Thus, the received signal has an instantaneous frequency f(t) defined as:

$$f(t) = f_0 + k \cdot t \tag{V-1}$$

Some of the time-frequency transformations considered in the study use the analytical version derived from the real received signal. In such cases, the analytical signal is computed with the Hilbert transform. The analytical noise-free linear chirp signal s(t) is given as:

$$s(t) = e^{j2\pi(f_0 t + \frac{k}{2}t^2)} \tag{V-2}$$

The signal is assumed to be sampled a rate of 512 [samples/pulse length]. Recall that the sampling frequency for a real signal must be equal to at least twice its highest frequency component. Thus, the signal may be heterodyned down to a lower frequency if needed. The discrete real part of the signal is given as:

$$s[n] = \mathrm{Re}\left(e^{j2\pi(\frac{f_0}{f_s}\cdot n + \frac{k}{2f_s^2}\cdot n^2)}\right) = \cos(2\pi(\frac{f_0}{f_s}\cdot n + \frac{k}{2f_s^2}\cdot n^2)), \qquad \text{n=1,..,512} \tag{ V-3}$$

We can rewrite the above equation as:

$$s[n] = \cos(2\pi(f_{n0}\cdot n + \frac{a}{2}\cdot n^2)) \quad where \quad f_{n0} = \frac{f_0}{f_s}, \quad a = \frac{k}{{f_s}^2}\text{ , n=1,..,512} \tag{V-4}$$

The terms a and f_{n0} represent the normalized slope and the normalized starting frequency respectively, with respect to the sampling frequency f_s. Thus, the normalized frequency equation for the linear chirp discrete signal is given by:

$$f_n = f_{n0} + a \cdot n \ , \ \text{n=1,..,512.} \qquad \text{(V-5)}$$

Multiple linear chirp signals trials were generated by randomly selecting both the initial frequency and the slope. Note that sampling constraints need to be satisfied to avoid aliasing in the resulting discrete linear chirp. For example, the parameter a needs to be selected so that the final instantaneous frequency doesn't exceed 0.5 for a given initial frequency $f_{n0.}$ Thus, aliasing may be avoided by selecting both the initial and final normalized instantaneous frequencies within the range [0, 0.5], and computing the corresponding slope parameter a. Initial and final frequencies are selected from a uniform distribution defined in the range [0,0.5]. Next, the chirp is corrupted by additive white Gaussian noise. Analytical expressions needed to compute some of the time-frequency transformations are obtained with the Hilbert transform of the noisy signal. Real signal expressions were used to compute the wavelet-based decompositions.

B. SIMULATION SET UP AND FEATURE EXTRACTION

Eleven different time-frequency and wavelet-based representations were considered in this study. The goal was to select a small number of transformations leading to the best "image quality" from that set. The representations considered were:

-Wavelet packet best basis

-Cosine packet best basis

-Wavelet pursuit

-Cosine pursuit

-Wigner-Ville distribution

-Pseudo Wigner-Ville distribution

-Reassigned Pseudo Wigner-Ville distribution

-Smoothed Pseudo Wigner-Ville distribution

-Reassigned Smoothed Pseudo Wigner-Ville distribution

-Spectogram distribution

-Reassigned spectogram distribution

Wavelet and cosine decompositions were implemented with the software "Wavelab 7.01" software [12], while all others used the "Time-Frequency Toolbox" [11]. Figure 26 shows the resulting images obtained for a linear chirp with SNR=10dB, when no energy thresholding is applied to the images. A few comments are in order regarding the selection of the transform parameters.

1. Number of Atoms

Recall that defining a line in a plane requires two points only. However, these two points must be perfectly localized in both time and frequency, and must be immune from any noise interference. Theoretically, we could use two atoms only from an atomic decomposition to define the linear frequency equation. However, this doesn't hold in practice as the atoms are not perfectly localized. Note that a larger number of atoms may better represent the line trend. However, some of the atoms may represent noise contributions for noisy signals. Therefore, selecting the number of atoms to represent a linear chirp in noise requires a trade-off between these two issues: fewer atoms to denoise the signal and more atoms to improve the resolution. We selected 10 atoms per decomposition for the atomic decompositions used in our study after running several trial cases.

2. Maximum Decomposition Level

Next, the maximum decomposition level was set to 7, as simulations showed no advantage in going to higher levels.

3. Wavelet Type

The mother wavelet function was selected after several trials among the readily provided functions in the Wavelab software [12]. Finally, we selected Daubechie-IV since it gave the best time-frequency representation for the types of signal considered.

4. Image Thresholding

No image intensity thresholding was applied to the time-frequency images, as simulations showed that it this step worsened the results for low SNR levels. Note that implicit denoising is actually performed by selecting a small number of atoms for the atomic decompositions.

5. Window Length

The PWV, SPWV and Reassigned SPWVD transformations use a frequency window which has a Hamming (N/4) time domain expression, and a Hamming (N/10) for time smoothing window. The analysis window selected for the spectrogram is Hamming (N/4), where N is the length of the signal (512 bins).

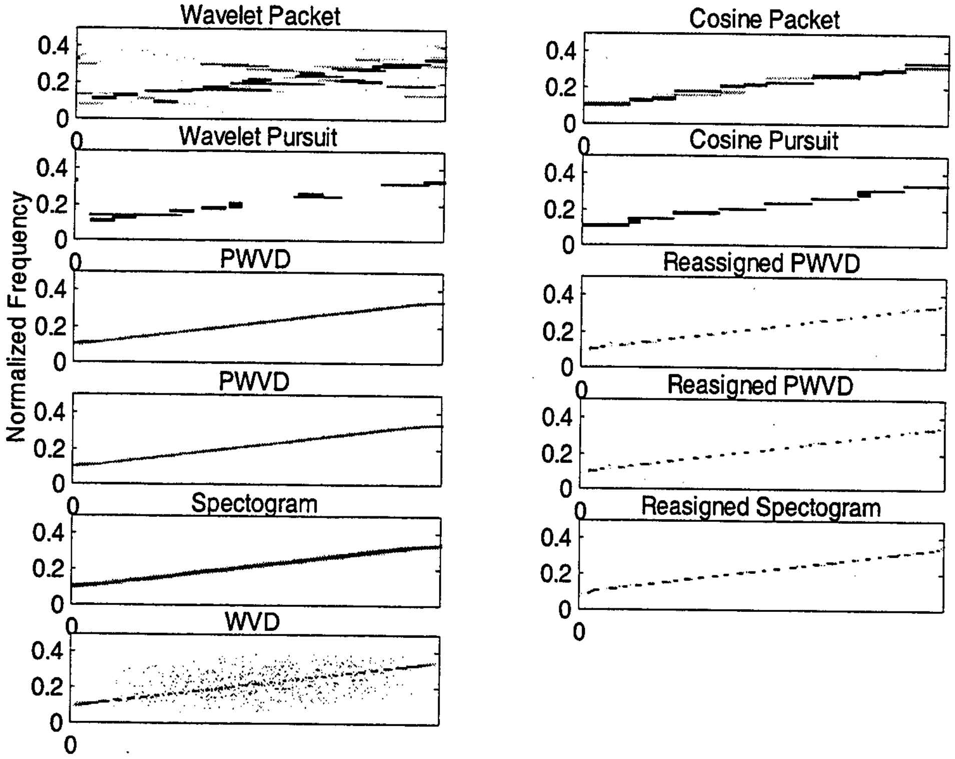

Figure 26: Various time frequency representations, linear chirp, SNR=10dB.

6. Radon Transformation Implementation Issues

The Radon transform is selected to extract the line equation from the time-frequency image, as described earlier in Section III. We use a two-stage implementation with final degree increment 0.1^0. The size of the resulting image for each time frequency representation is set to 256x256 points, as the Radon transform for larger images would be too computationally expensive for a MATLAB implementation. One hundred randomly generated linear chirps for a given SNR level were generated, and the SNR varied in the range –10dB to 10dB. Next, the chirp parameters were estimated from the images generated by each of the eleven time-frequency transformations considered. Performance comparisons are presented next.

C. PERFORMANCE COMPARISONS

1. Evaluation of Time-frequency Representations

Recall that the maximum theoretical slope error is ±0.05° since the final step angle in the radon transform is 0.1°, as discussed in Section IV. However, we also have to add noise effects, and quantization errors introduced by the finite resolution in the image. Figures 27 to 29 present the mean and the standard deviation of the absolute slope and offset errors as a function of the SNR level for all time-frequency transformations considered in the study.

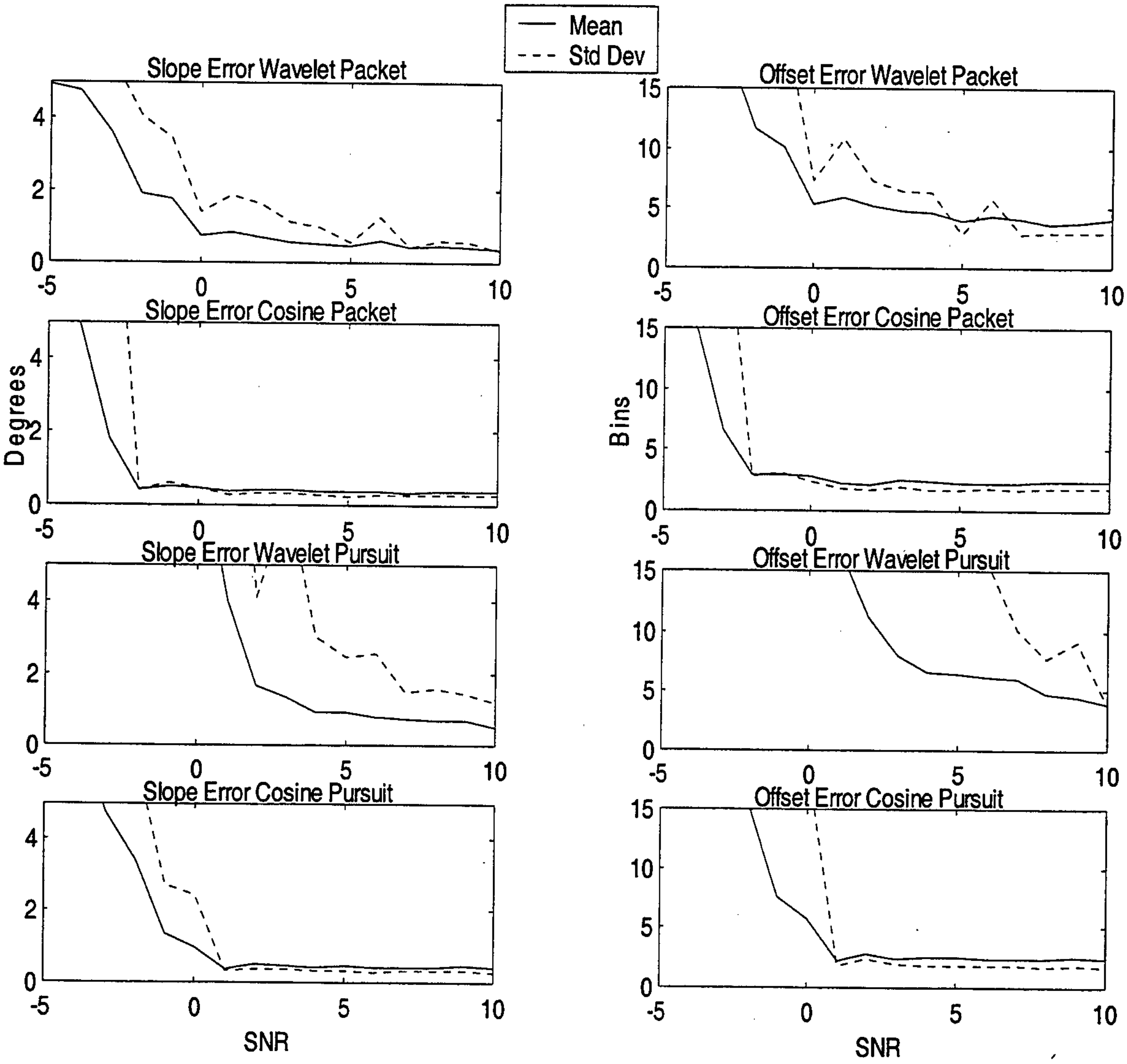

Figure 27: Linear chirp signal; Slope and offset error for the wavelet packet and cosine packet decompositions, wavelet and cosine pursuit decompositions (SNR in dB.)

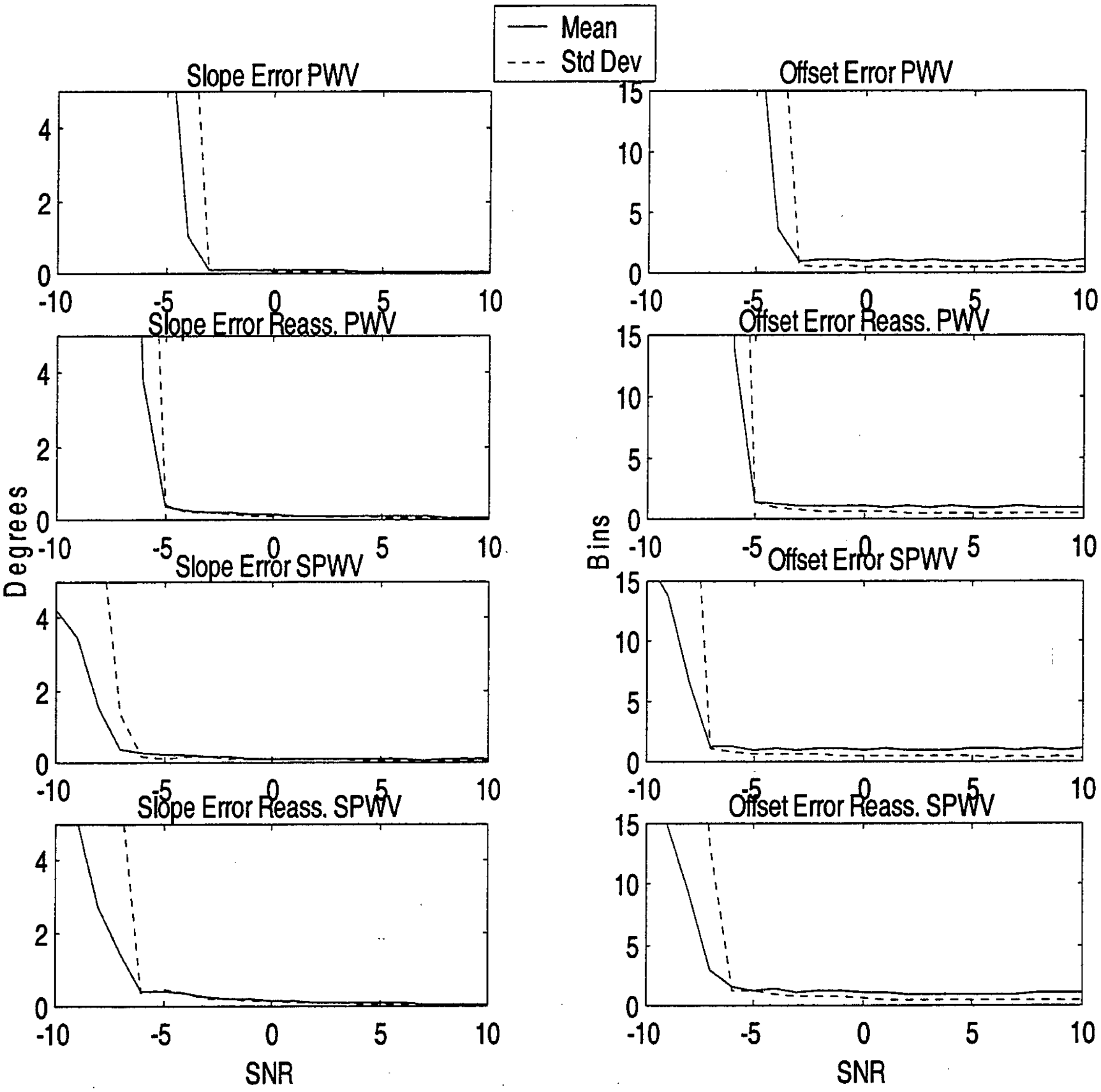

Figure 28: Linear chirp signal; Slope and offset error for the PWV, reassigned PWV, SPWV and reassigned SPWV decompositions (SNR in dB.)

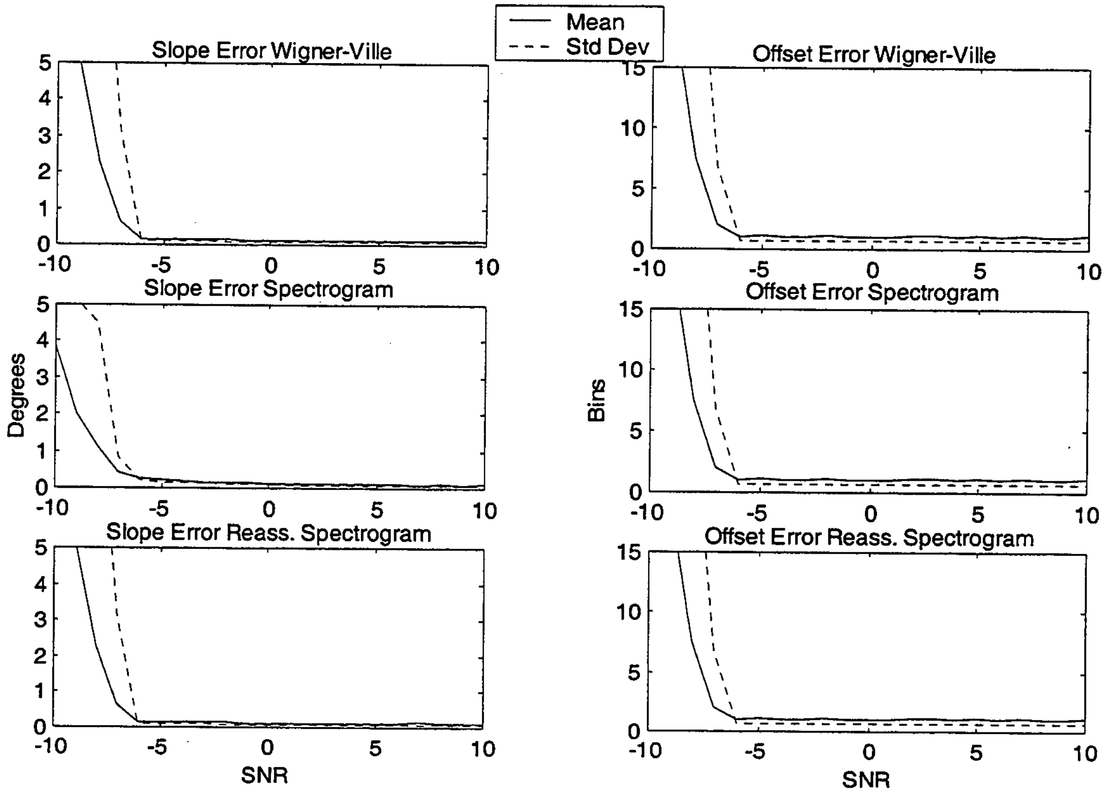

Figure 29: Linear chirp signal; Slope and offset error for the Wigner-Ville decomposition, Spectrogram, and reassigned Spectrogram (SNR in dB.)

A few comments are in order:

1. Results show that the energy distributions perform better than the atomic decompositions. This is to be expected as they provide a more accurate "image" in the time-frequency plane. Note that atoms cannot be well localized in both time and frequency, as would be required to represent linear chirps accurately. The best-basis cosine packet decomposition gives the best results followed by the cosine pursuit scheme for atomic decompositions.

2. Most of the energy distributions have slope errors very close to the theoretical value of 0.05° for medium to high SNR levels.
3. The Wigner-Ville distribution has a very good and almost stable performance for SNR's in the range of –6dB to 10dB. The smoothing time window present in the Pseudo Wigner-Ville distribution improves the estimation in higher SNR but shrinks the effective range. The Smoothed Pseudo Wigner-Ville distribution has a slightly worst performance at high SNR but also has the widest effective range. The smoothing in time and in frequency eliminates the interference terms almost completely so that the representation is more immune to the noise than other transformations are. However, frequency smoothing results in lowered performance at high SNR levels.
4. Results show that the reassign method usually improves the performance at high SNR levels, as it forces the representations to be more "focused". Unfortunately, applying the reassign method in low SNR levels worsens the performances. This is to be expected as the presence of noise close to the line moves the local center of gravity of the distribution away from its theoretical value.

Table 1 lists the mean absolute error for the frequency slope and offset parameters at SNR equal to 10dB for all transformations considered for a "rough" comparison of the transformation performances. In addition, Table 1 lists the SNR level at which the errors suddenly increase, indicating the SNR level at which using a specific distribution may become more questionable.

	Mean of absolute slope error @ SNR=10dB (in degrees)	Mean of absolute offset error @ SNR=10dB (in bins)	Breaking point (in dB)
Wavelet Packet	0.375	4	0
Cosine Packet	0.5149	3.98	0
Wavelet Pursuit	0.37	2.35	4
Cosine. Pursuit	0.4297	2.35	0
WV	0.1198	1.08	-7
PWV	0.0623	1.02	-4
Reassigned PWV	0.0729	1	-5
SPWV	0.0792	1.02	-7
Reassigned SPWV	0.0563	1.02	-7
Spectrogram	0.098	1.77	-7
Reassigned Spectrogram	0.12	1.78	-7

Table 1: Mean absolute errors for frequency slope and offset. 100 trials, SNR=10dB, performance breaking points for all time-frequency methods considered.

2. Error Evaluation

Recall the received signal was sampled at a rate of 512 [samples/pulse period] while the image size was set to 256x256bins in order to speed up the computations. However, the errors presented were corrected to correspond to an image with size 512x256, where the time and frequency axes have 512 and 256 bins respectively. Next, the goal is to investigate how the estimated errors relate to the frequency error obtained for the frequency expression given in equation (V-1).

Recall that the normalized frequency expression is given by:

$$f_n = f_{n0} + a \cdot n ,$$

and the slope angle ϑ measured in the image is:

$$\tan(\vartheta) = \frac{a \cdot N \cdot 2N_f}{N_t} , \qquad \text{(V-6)}$$

where a is the frequency slope parameter, N is the length of the signal, and N_f and N_t represent the number of bins in the frequency and time axis respectively. In our case N_f=N/2 and N_t=N. Thus, equation (V-6) simplifies to:

$$\tan(\vartheta) = a \cdot N . \qquad \text{(V-7)}$$

Recall that

$$s[n] = \cos(2\pi(f_{n0} \cdot n + \frac{a}{2} \cdot n^2)),\ where \quad f_{n0} = \frac{f_0}{f_s}, \quad a = \frac{k}{f_s^{\ 2}} \text{ , n=1,..,512}$$

Let's assume that the linear chirp has frequency slope k_0 and that ϑ_0 is the corresponding angle, which leads to:

$$\frac{k_0}{f_s^{\ 2}} = a_0 = \frac{\tan(\vartheta_0)}{N} . \qquad \text{(V-8)}$$

Now, let's assume we estimate an angle equal to ϑ_{est} instead of ϑ_0 due to errors in the estimation process. Thus,

$$\vartheta_{est} = \vartheta_o + \Delta\vartheta , \qquad \text{(V-9)}$$

where $\Delta\vartheta$ is the angle error. As a result, the resulting estimated frequency slope k_{est} expression is given by:

$$\begin{aligned} k_{est} &= f_s^{\,2} \cdot a_{est} = f_s^{\,2} \cdot \frac{\tan(\vartheta_0 + \Delta\vartheta)}{N} \\ &= \frac{f_s^{\,2}}{N} \cdot \frac{\tan(\vartheta_0) + \tan(\Delta\vartheta)}{1 - \tan(\vartheta_0) \cdot \tan(\Delta\vartheta)} \\ &= \frac{k_0 \cdot f_s^{\,2} \cdot N + f_s^4 \cdot \tan(\Delta\vartheta)}{f_s^{\,2} \cdot N - k_0 \cdot N^2 \cdot \tan(\Delta\vartheta)}. \end{aligned} \tag{V-10}$$

Thus, the final frequency slope error is:

$$\Delta k = k_0 - k_{est} = \frac{(k_0^2 \cdot N^2 + f_s^4) \cdot \tan(\Delta\vartheta)}{k_0 \cdot N^2 \cdot \tan(\Delta\vartheta) - f_s^2 \cdot N} . \tag{V-11}$$

The slope frequency error Δk is expressed in terms of the angle error, but also depends on the sampling frequency, and the length of the signal N.

If we assume again that f_0 is the initial offset value corresponding to equation (V-1), then the normalized offset frequency is derived from equation (V-4) as $f_{n0}=f_0/f_s$. The value f_{b0} measured is expressed in bins and is related to f_{n0} by:

$$f_{b0} = f_{n0} \cdot 2 \cdot N_f \tag{V-12}$$

Assume the estimated initial offset value f_{best} contains an error Δf_b. Thus,

$$f_{best} = f_{b0} + \Delta f_b \ , \tag{V-13}$$

and the estimated value for f_0 according to equations (V-4) and (V-1) is given by:

$$f_{0est} = f_s \cdot \frac{f_{b0} + \Delta f_b}{2N_f} = f_0 + f_s \cdot \frac{\Delta f_b}{2N_f} , \tag{V-14}$$

When N_f=N/2 , the resulting error is:

$$\Delta f_0 = f_0 - f_{0est} = -f_s \cdot \frac{\Delta f_b}{N} . \tag{V-15}$$

Equations (V-11) and (V-15) provide the relations between errors measured in the time-frequency image representation of size N/2xN, where N is the length of the signal, and the corresponding errors in the initial time-varying frequency expression given in Equation (V-1). Note that Δf_b is measured in bins.

3. Multi-pulse Processing

Up to this point, all results were derived by extracting the frequency modulation parameters from one noisy radar pulse. Performances improve when using more than one pulse. Assume we isolate five radar pulses, where each pulse contains the same transmitted signal at a given SNR level. Note that the signal information gets mapped to the same location of the time-frequency plane, while the noise contributions scatter to different location in each trial. Thus, averaging multiple time-frequency images improves the quality of the signal information.

Thus, averaging the contribution of five pulses leads to the results shown in Figure 30 obtained for the Smoothed Pseudo Wigner-Ville transformation. One hundred realizations per SNR level are used, and SNR levels varied from –20 to 0dB. Note that there is no need to consider higher SNR levels as the error curves already flatten for SNR=0dB. Results show a significant improvement over using one pulse only. Further improvements may be obtained by considering a larger number of pulses. However, this will result in a direct increase of the computation time.

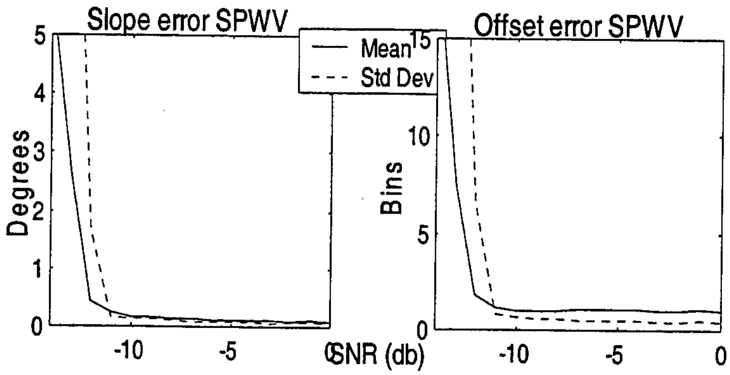

Figure 30: Slope and offset error of SPWV distribution; five realizations.

4. Examples

We apply the above results to a specific radar using pulse compression readily available to us: the SPS-40B radar. The Radar Set AN/SPS-40B is used for the detection ranging and tracking of air targets at long range in American and foreign navy services [24]. In this pulse mode, the pulse length is τ=60μsec. The transmitted waveform inside the pulse is a linear modulated down-chirp with a bandwidth of 2MHz centered at a frequency that can be varied manually by the operator from 402.5 to 447.5 MHz.

Let's assume that the radar operates at 437.5 MHz, i.e., the chirp starts with a frequency 438.5MHz and decays linearly to 436.5 MHz. Our ESM (Electronic Support Measures) receives a series of noisy pulses from this radar. Further, assume we can isolate one pulse perfectly and that the estimated received signal is centered at 438MHz with a bandwidth equal to 2MHz. Given that the duration of the pulse is 60μsec and if we use 512 [samples/pulse duration], the sampling frequency should be set at:

$$f_s = \frac{512}{60} MHz = 8.53 MHz\,. \tag{V-16}$$

This sampling frequency does not satisfy the Nyquist rate. Thus, we can either increase the sampling frequency or heterodyne to a lower center frequency. The first option has the drawback of increasing the number of samples to deal with, and the computational load of the estimation schemes considered. Heterodyning the signal down to the baseband can be accomplished by multiplying the received signal with a cosine function, and using a lowpass filter to extract the information. Assume the heterodyning frequency is selected as 440MHz. Thus, the resulting chirp signal is an up-chirp at the frequency 440-437.5=2.5MHz with bandwidth equal to 2MHz after heterodyning and filtering. The theoretical time-frequency representation is shown in Figure 31. Thus, the instantaneous frequency of the heterodyned signal is given by:

$$f(t) = 1.5 \cdot 10^6 + 3.33 \cdot 10^{10} \cdot t\,. \tag{V-17}$$

The heterodyned signal may be sampled with a frequency f_s=8.53Mhz, corresponding to 512 samples/pulse duration.

Assume we use the SPWV distribution and the chirp parameter estimation procedure described above in Section IV-B. Figure 28 shows that the mean slope error is 0.09^0 and the mean offset error is 1 bin when the SNR is equal to 5dB. Using equations (V-11) and (V-15) leads to:

$|\Delta k|=2.610^8$ and $|\Delta f_0|=1.6710^4$, meaning that the normalized errors for the frequency slope and offset are equal to $\frac{|\Delta k|}{3.33 \cdot 10^{10}} = 0.008$ or 0.8% and $\frac{|\Delta f_0|}{1.2 \cdot 10^{6}} = 0.0104$ or 1.04% respectively for the heterodyned signal.

Note that the normalized slope error remains the same for the original received signal, as the signal bandwidth has not been changed by the heterodyning process, while the normalized frequency offset error becomes equal to $1.6710^4/438.510^6$, i.e., 0.0038%.

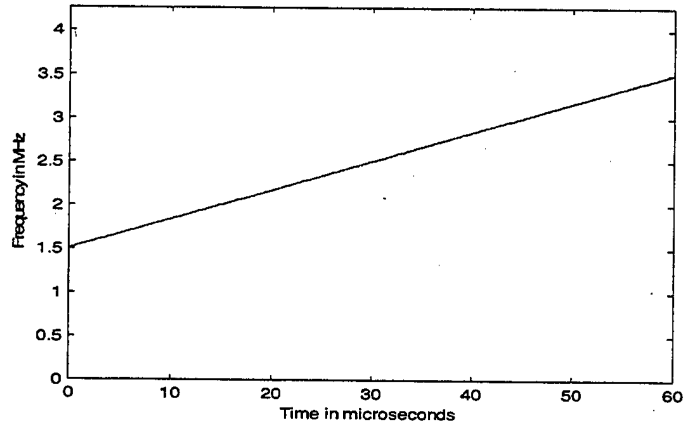

Figure 31: Instantaneous frequency for the heterodyned signal.

VI. HYPERBOLIC MODULATED CHIRPS

This section considers the estimation of hyperbolic chirp parameters. As before, the starting point is the time-frequency image representation of the information. However, generalizations of the Radon transform to hyperbolic line types were not used. Instead, we consider an iterative procedure to estimate the chirp parameters.

Note that we assume we know the type of modulation transmitted, as we do not address issues related to distinguishing between linear and hyperbolic modulation types. Such classification issues are left for future work.

A. SIGNAL GENERATION

1. Introduction

Assume that we can isolate individual pulses of duration τ=512 samples. Thus, a received signal with hyperbolic modulation frequency is given by:

$$x(t) = \cos(2\pi(A\ln(t+t_0) + B\cdot t)), \tag{VI-1}$$

where

$$f(t) = \frac{A}{t+t_0} + B. \tag{VI-2}$$

The analytical signal s(t) obtained from x(t) with a Hilbert transform is given by:

$$x(t) = e^{j\cdot 2\pi(A\ln(t+t_0)+Bt)}. \tag{VI-3}$$

The corresponding discrete signal x*[n]* is given by:

$$x[n] = e^{j2\pi(A\cdot\ln(\frac{n}{f_s}+\frac{n_0}{f_s})+B\frac{n}{f_s})} = e^{j2\pi(A\cdot\ln(n+n_0)-A\cdot\ln(f_s)+\frac{B}{f_s}\cdot n)}, \tag{VI-4}$$

or finally

$$x[n] = K \cdot e^{j \cdot 2 \cdot \pi (a \cdot \ln(n+n_0)+bn)}, \tag{VI-5}$$

where:

$$a = A, \;\; b = \frac{B}{f_s}, \;\; n_0 = f_s \cdot t_0, \text{ and } K = e^{j2\pi A \ln(f_s)} \; . \tag{VI-6}$$

The normalized instantaneous frequency expression is given by:

$$f_n[n] = \frac{a}{n+n_0} + b . \tag{VI-7}$$

Recall that we need to select the parameters *a, b* and n_0 such that the range of the normalized frequency $f_n[n]$ is between 0 and 0.5 to prevent aliasing, which leads to the following ranges:

$$\alpha \in [0,\infty], \quad n_0 \in [0,\infty], \;\; b \in [-\infty, 0.5] \tag{VI-8}$$

A valid selection for *a, b* and n_0 is obtained by selecting one parameter in its allowed range, and then the other two so that they also fall in their allowed ranges. In the simulation, we first select a value for *b* from a uniform distribution in the region [0, 0.5]. The starting frequency at *n=0*, $f_n(0)$,is selected from a uniform distribution in the region [*b*, 0.5]. The final frequency $f_n(N)$, for *n=N,* where N is the signal length, can be selected in the range [b,f_n(0)]. Defining *b*, $f_n(0)$, and $f_n(N)$ leads to the following values for *a* and n_0:

$$a = \frac{(f_n(0)-b) \cdot (b + f_n(0)) \cdot N}{(f_n(0) - f_n(N))} \tag{VI-9}$$

$$n_0 = \frac{(b + f_n(N)) \cdot N}{(f_n(0) - f_n(N))}. \qquad \text{(VI-10)}$$

Note that this random selection can sometimes lead to a frequency equation with a very steep slope at the beginning, as shown in Figure 32. Such a behavior is undesirable because none of the time-frequency representations gives good resolution in the area of the steep slope, and this modulation type is not typical in radar applications.

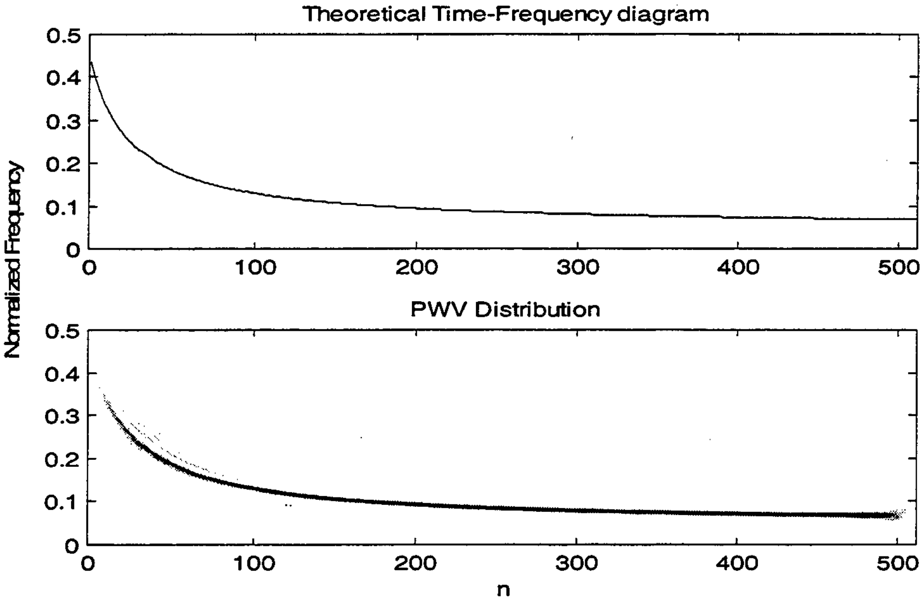

Figure 32: Normalized frequency for a=10, n_0=25, b=0.05.

2. Hyperbolic Line Parameter Constraints

We wish to restrict our random selection of the signal parameters to those leading to chirps without steep slopes, and need to automate the process so that we can run multiple trials to study the scheme robustness to noise degradations. Thus, we define a figure of merit that describes the amount of curvature present in a given hyperbolic line.

The parameter selected to characterize the curvature of the chirp is the distance d defined as the maximum distance between any point of the curve and its cord.

Note the parameter b shifts the hyperbolic chirp up or down without affecting the shape, thereby its curvature. Therefore, we assume that $b=0$ for simplicity. In such a case, the chord associated to the hyperbolic chirp $f_n[n] = \frac{a}{n+n_0}$ is a line passing through the points $(0, \frac{a}{n_0})$ and $(N, \frac{a}{N+n_0})$ with equation:

$$\frac{f - \frac{a}{n_0}}{n-0} = \frac{\frac{a}{N+n_0} - \frac{a}{n_0}}{N-0}.$$

The resulting chord equation is given by:

$$a \cdot n + n_0 \cdot (N+n_0) \cdot f - a \cdot (N+n_0) = 0, \qquad \text{(VI-11)}$$

which leads to:

$$f = \frac{-a}{n_0 \cdot (N+n_0)} n + \frac{a}{n_0}. \qquad \text{(VI-12)}$$

Recall that the gradient of a curve at a given point is a line that passes through that point and has a slope equal to the derivative of the curve at that point. Thus, the gradient at any point of the hyperbolic line is:

$$A = f' = \frac{-a}{(n+n_0)^2}. \qquad \text{(VI-13)}$$

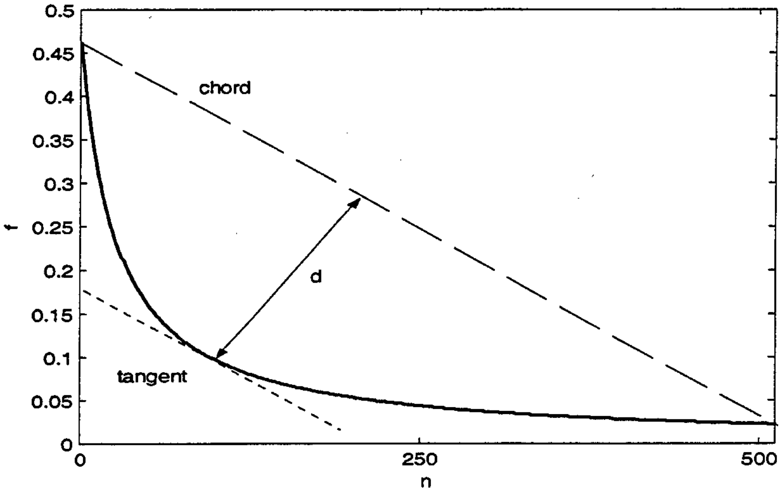

Figure 33: Maximum distance between a hyperbolic line and its chord. a=12, n0=25, b=0.

The distance d is maximum at the point $n=\xi$ where its gradient is parallel to the chord. As a result, the gradient slope at the point $n=\xi$ is equal to the slope of the chord. Thus the position of the point $n=\xi$ can be estimated from (VI-11) and (V-13) as:

$$\frac{-a}{(\xi+n_0)^2}=\frac{-a}{n_0\cdot(N+n_0)},$$

which yields the coordinates of the point ξ as:

$$\xi=\sqrt{n_0(N+n_0)}-n_0, \tag{VI-14}$$

$$f(\xi)=\frac{a}{\sqrt{n_0(N+n_0)}}. \tag{VI-15}$$

Recall the distance d between a point (x_1,y_1) and a line with equation $a_1\cdot x+a_2\cdot y+a_3=0$ is given by the equation:

$$d = \frac{| a_1 \cdot x_1 + a_2 \cdot y_1 + a_3 |}{\sqrt{a_1{}^2 + a_2{}^2}}. \tag{VI-16}$$

Thus, the distance between the chord and the point (ξ,f(ξ)) can be computed using equations (VI-11), (VI-14), (VI-15), and (VI-16), which leads to:

$$d = \frac{\left| a \cdot \xi + n_0 \cdot (N + n_0) \cdot f(\xi) - a \cdot (N + n_0) \right|}{\sqrt{a^2 + n_0^2 (N + n_0)^2}},$$

and:

$$d = \frac{|a| \cdot \left| (N + 2 \cdot n_0) - 2 \cdot \sqrt{n_0 (N + n_0)} \right|}{\sqrt{a^2 + n_0^2 (N + n_0)^2}}. \tag{VI-17}$$

The distance d can be viewed as a figure of merit for the curvature of the hyperbolic line. A high value of d means the curvature is high and the time-frequency representation near the time origin is poor. However, very small values of d represent cases for which the hyperbolic curve is almost indistinguishable from a straight line. Thus, we restricted the chirp signals generated so that the distance d is in the region $\left[\frac{20}{1024} \cdot f_s , \frac{80}{1024} \cdot f_s \right]$ to avoid such cases.

Next, additive zero-mean white Gaussian noise is added to generate the noisy chirp with a specific SNR level.

B. FEATURE EXTRACTION

1. Introduction

The signal time-frequency representation can be obtained with any of the energy distributions discussed earlier in Section III. However, note that the atomic decompositions are not as well suited as the energy distributions for the task at hand, as they do not describe the hyperbolic line curvature accurately. Therefore, we only consider energy distributions in this chapter.

At this point the task is to extract the three unknowns parameters (a,b,n_0) for a given time-frequency representation. The basic Radon transform can no longer be applied, as it is defined for straight lines only. The Radon transform was extended to detect functions of arbitrary shape [17], however the computational load is significantly higher.

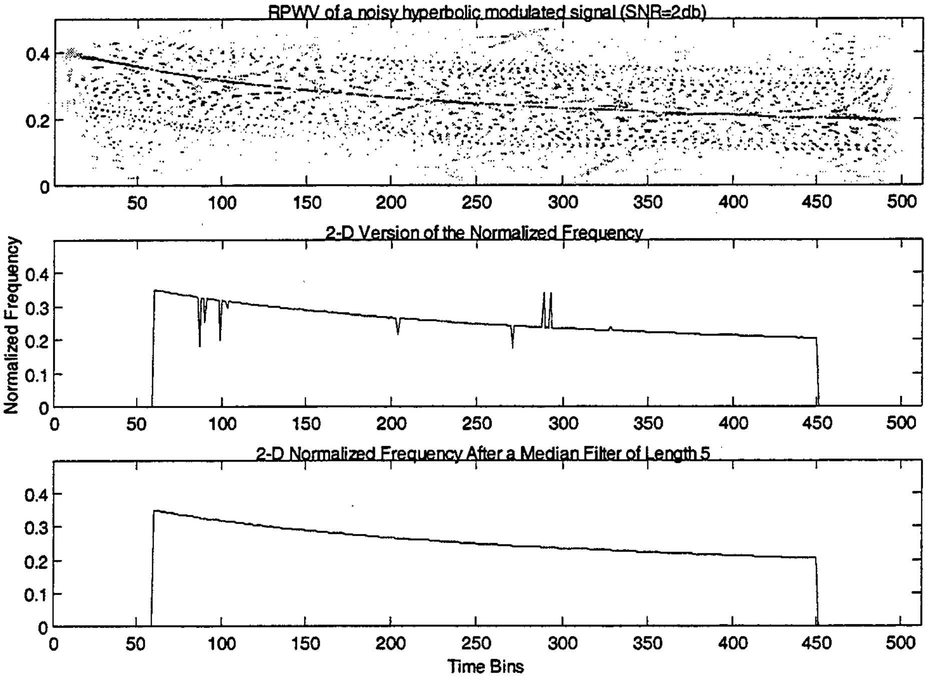

Figure 34: Extraction of the instantaneous normalized frequency expression from the time-frequency image. Before and after median smoothing, L=5.

The method considered here approaches the problem quite differently. It is an adaptive procedure which fixes one of the three unknown parameters at each iteration, and estimates the other two. The resulting scheme is presented next.

2. Method Description

The instantaneous frequency expression for the hyperbolic chirp is extracted from the two-dimensional image by selecting the peak values obtained at each time bin of the image. The result is a vector containing the frequency values for each time bin, as shown in Figure 34.

Note that the 2-D instantaneous frequency approximation is very accurate at high SNR levels, and degrades as the SNR level decreases. In noisy environments, the pixel with the highest energy obtained at a given time bin may be an outlier, resulting in spikes in the instantaneous frequency estimate, as illustrated in the middle plot of Figure 34. Such outliers can be smoothed out with a median filter. Simulations showed that a median filter of length 5 smoothed out potential "spikes" without loss of resolution.

We selected the three energy transformations leading to the best time-frequency image quality for the linear chirp case, and restricted our hyperbolic chirp analysis to those. The distributions selected are:

- Pseudo Wigner-Ville distribution,
- Reassigned Pseudo Wigner-Ville distribution,
- Reassigned Smoothed Pseudo Wigner-Ville distribution.

We set the dimension of the image at 512x512 bins to increase the image resolution and reduce quantization errors. In addition, simulations showed that these three energy distributions do not necessarily perform well at the beginning and end of the image. As a result, we only consider the image from time bin 60 to time bin 450.

The second step in the proposed scheme approximates the instantaneous frequency adaptively with a hyperbolic line by minimizing the squared error between the information contained in the image and a theoretical hyperbolic curve expression. Note that the problem to be solved is non-linear, due to the specific frequency law to be approximated, as we estimate the parameters *a, b* and n_0 given in equation (VI-7). We first tried to solve the problem using a classical nonlinear least square iteration scheme provided with the function "lsqnonlin" from the MATLAB optimization toolbox [15]. Simulations showed that the algorithm converged to different local minimum, depending on the initial values selected. However, this problem can be resolved using a two-step procedure as follows.

Assume we wish to approximate the hyperbolic curve given in equation (VI-7) with a function of the form:

$$y(n) = \frac{a}{n+n_0}. \tag{VI-18}$$

If we assume our estimation values obtained from the image to be equal to $y_{est}(n)$, n=1,...N, then we wish to find a and n_0 so that:

$$y_{est}(n) - \frac{a}{n+n_0} = 0, \quad \text{for n=1...N.} \tag{VI-19}$$

The above set of equation forms a linear system of N equation with two unknowns which can be solved using a least-square method. Next, assume we have an estimate of the parameter b, b_{est}, which contribution is subtracted from the frequency equation given in (VI-7), resulting in:

$$f_n[n] - b_{est} = \frac{a}{n+n_0} + b - b_{est}. \tag{VI-20}$$

Thus, the problem becomes to fit the data f[n] by finding the parameters *(a, n_0)* which best fit a curve of the form *$a/(n+n_0)$* in a least squares sense. The set of estimated parameters has a mean-square error e_i. At this point, the problem becomes to update the parameter b, and re-estimate corresponding values for a and n_0 so that the error function expressed as a function of b is convex with a strong minimum. The location of the minimum value for the error function is obtained for b_{est} and the best estimated values of a and n_0.

Even though we could not formally prove that the shape of the error function as a function of b is convex, we observed the same type of convex shape for the error function over 150 randomly generated hyperbolic chirps. Figure 35 plots the normalized errors and the mean values for *a, b,* and n_0.

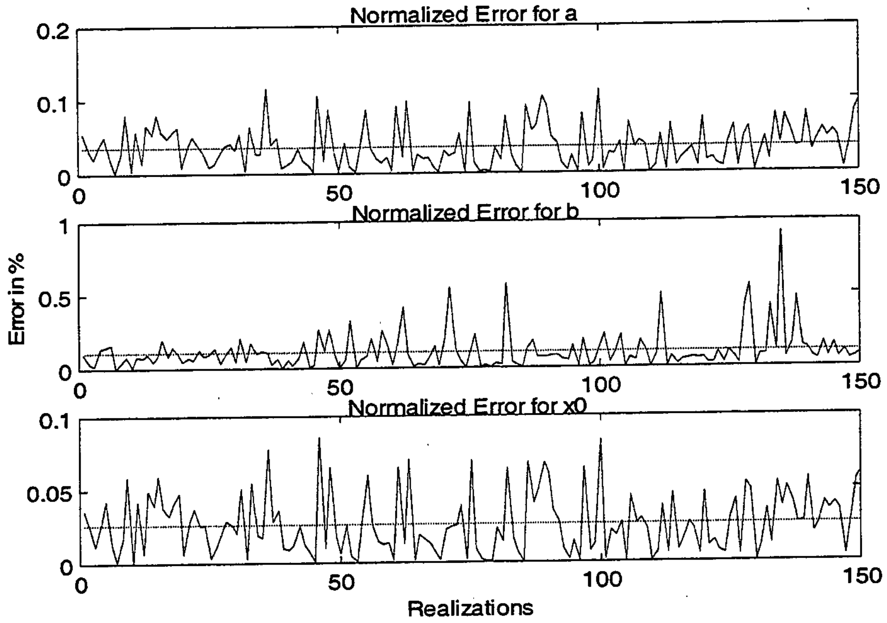

Figure 35: Normalized errors obtained for a, b, n_0 , 150 randomly generated hyperbolic chirps

The mean and standard deviation of the error is:

	Mean	Std Deviation
a	0.0361%	0.000287
b	0.1092%	0.0013
n_0	0.0262%	0.000207

Note that the mean error for b is slightly higher than that of the other two unknowns. This is to be expected, as *b* is restricted to the range [0, 0.5]. Therefore, very small error values may correspond to large normalized errors.

We adaptively estimate the value for *b*, by taking advantage of the convex shape of the error function. First, we restrict the search to a specific region of *b*. Next, we select five values of *b* equally spread within that range, and compute the other two parameters and the resulting mean-square error. Next, we restrict the *b* range to that including the minimum location by using the information provided by the mean-square errors, and repeat the process. This iterative process can be performed forever, as we have no knowledge of the minimum mean square error. In practice, we restricted the number of iterations to 10, as the values of *b* were restricted to small range [0,0.5] in our simulations to keep the computational load under control. Theoretically, the range of *b* can set as large as we want. The algorithm can converge in any area of *b* but it will require a larger number of iterations to preserve the same accuracy as the range of *b* expands, resulting in a computational load increase.

C. SIMULATION RESULTS

A few comments are in order before discussing the simulation results:

1. The scheme considered above is not as robust to noise degradations as the linear chirp scheme described in Section V. This is to be expected as a relatively clear and undistorted time-frequency image is required to extract the normalized frequency information.

2. The iterative scheme finds the set of a, b and n_0 which minimize the mean-square error. However, relatively large error values in the parameter estimates may correspond to small error in the actual hyperbolic curves. Figure 36 shows true and estimated hyperbolic curves. The true parameter values are: a=65, b=0.1 and n_0=240, while the estimated parameter values are: a=84, b=0.08 and n_0=294. The normalized

errors are 29%, 20% and 22% for *a*, *b* and n_0 respectively, even though the two curves are hardly different.

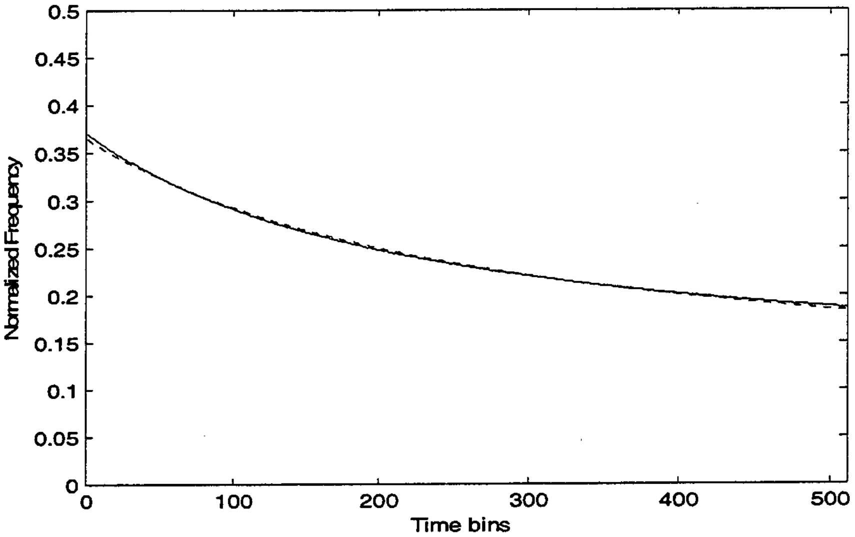

Figure 36: Two hyperbolic lines. Solid line: true hyperbolic curve (a=65, b=0.1 and n_0=240, dotted line: estimated hyperbolic curve (a=84, b=0.08 and n_0=294).

Hyperbolic chirps were generated by randomly selecting the parameters a, b, and n_0 in the allowed ranges mentioned earlier. Next, additive white Gaussian noise was added to generate SNR levels between 0 and 20dB, in steps of 2dB. One hundred realizations were generated for a given SNR level. Figures 37 to 39 plot the mean and the standard deviation for the normalized errors for (a, b, n_0) as a function of the SNR level, where the normalized error is defined as:

$$norm\ error = \frac{|\,true\,value - estimated\,value\,|}{true\,value} \cdot 100\ \%$$

Note that the above definition may lead to large errors when the parameter true values are very close to zero, due to the denominator. Thus, we restricted our simulations to cases where b is in the range [0.025, 0.5]. The other two parameters a and n_0 are selected using the method described in section (VI-A).

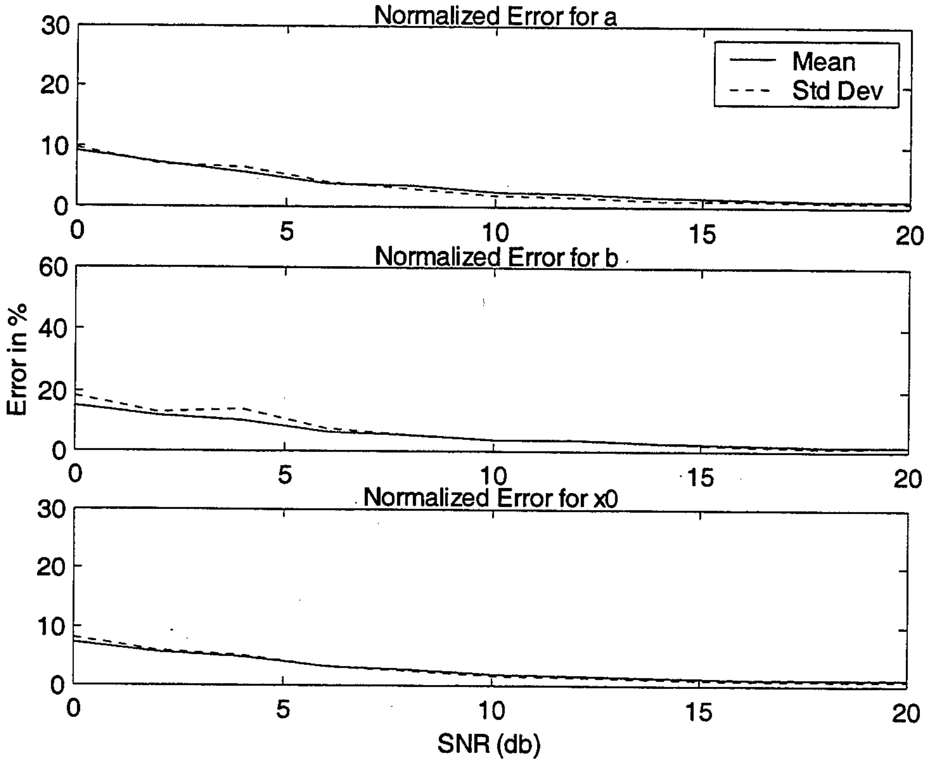

Figure 37: Hyperbolic chirp; normalized errors for the SPWV distribution.

Figure 37 to 39 show the normalized error mean-square and corresponding standard deviation for (a,b,n_0) obtained using the Smooth Pseudo Wigner-Ville, the Reassigned Pseudo Wigner-Ville, and the Reassigned Smoothed Pseudo Wigner-Ville distribution respectively.

Results show that the normalized errors decrease to zero as the SNR level increases. They also show that the SPWV is the scheme most robust to noise degradations out of the three considered.

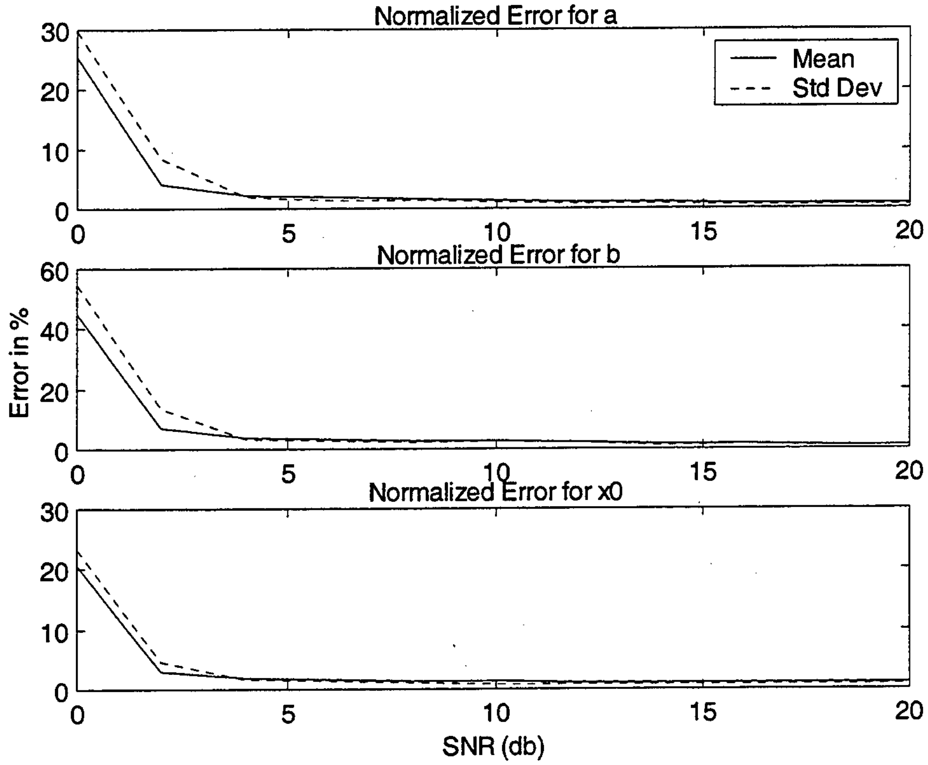

Figure 38: Hyperbolic chirp: normalized errors for the RPWV distribution.

Results also show that the reassigned methods perform better than the SPWV for high SNRs. This is to be expected, as they provide a more focused image by finding the center of gravity of the energy distribution for each time instance, resulting in a better image quality. However, the reassignment process worsens the image quality, as the noise level increases, resulting in the estimation process breaking down. Simulations show that the RPWV performs slightly better than the RSPWV, especially for SNR's in the range of 2 to 5dB.

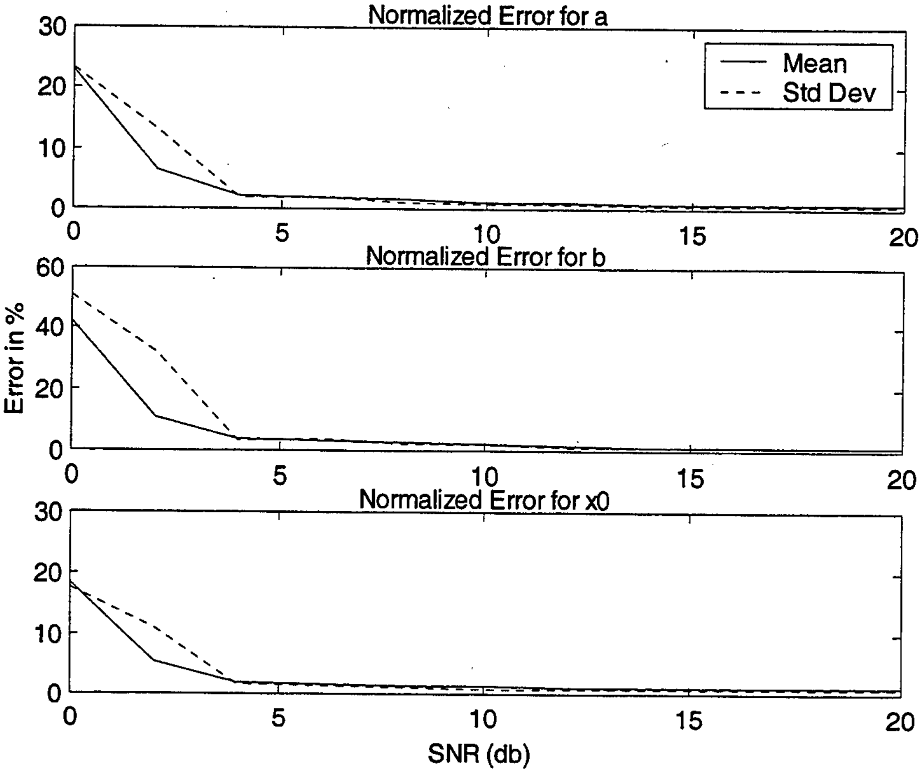

Figure 39: Hyperbolic chirp: normalized errors for the RSPWV distribution.

The continuous parameters A, B and t_0 in equation (VI-2) are related to the parameters of the discrete signal expression via equation (VI-6). Thus, the normalized errors estimated for the discrete values *a, b* and n_0 are identical to those obtained with the continuous parameters.

VII. CONCLUSIONS

This study considered the application of time-frequency analysis techniques to extract intra-pulse modulation parameters for radar signals using analog pulse compression. Two specific types of intra-pulse modulation were considered: linear and hyperbolic modulations were investigated.

The estimation procedures use the image obtained from the two-dimensional time-frequency representation of the radar pulse. Eleven types of time-frequency representations, and their robustness to noise were considered in this study. The type of modulation, as well as, the start and stop time of the pulse under investigation was assumed to be known.

Linear chirp parameters were extracted by applying the Radon transform to the time-frequency image. Results show that energy distributions performed better, as they produced better focused images. Results show performance differences between the energy distributions to be minimal. Results also show that the Smoothed Pseudo Wigner-Ville distribution has the best performance in low SNR environments, while the reassignment scheme improves the performances for high SNR's. The main drawback behind these two schemes is their computational load requirements.

Atomic decompositions did not perform as well as the energy distributions. The main reason is that the atoms used in the study were not well matched to the types of signals under consideration. The types of atoms used were not well localized in both time and frequency, as would be required to represent signals with linear modulation.

However, the cosine packet decomposition and cosine packet pursuit algorithm gave good results for medium to high SNR levels.

The Radon transform as an image processing technique accurately extracts the line parameters, even in noisy environments. The combination of the energy distributions and the Radon transform can lead to small detection errors and almost constant performances for SNRs as low as –7dB. At such a low SNR the detection is limited by the ability to find accurately the time limits of the pulse. The resulting overall error for this scheme depends directly on the size of the image and the step size of the projection angle applied in the Radon transform step. Selecting these two parameters requires a trade-off between the desired accuracy and the computation time.

Results also showed that the estimation performances increase when processing more than one pulse at a time. This improvement results from the fact that the noise gets mapped into different locations of the time-frequency image, while the signal location remains stable. Simulations showed that the detection SNR threshold dropped to –10dB when processing five pulses simultaneously. However, the computation time increases dramatically.

Hyperbolic chirp parameters were extracted from the time-frequency image using an iterative procedure. The proposed scheme is a two-step procedure. We tested the scheme using basic hyperbolic curves, and simulations showed the estimation error to be in the range 0.02% to 0.1%.

First, we extracted the 2-D instantaneous frequency expression from the image. Next, we approximated the unknown parameters adaptively. This scheme requires a very good estimate of the instantaneous frequency expression as an initial estimate, thereby restricting its application to medium to high SNR levels.

The reassignment method was shown to be the best scheme at high SNR level, resulting in 1% normalized errors for the unknown chirp parameters *a, b* and n_0. In addition, results show that the Reassigned Pseudo Wigner-Ville distribution performs better for SNR down to 2dB. However, errors increase rapidly for lower SNRs.

Simulations show that energy distributions without the reassignment step, such as the Smoothed Pseudo Wigner-Ville (SPWV) distribution, do not break down suddenly as the SNR goes down. For example, normalized errors for the SPWV are in the range 8%-15% for SNR=0dB and decrease smoothly to 1-2% for SNR=20dB.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

```
function [ofs,slope]=radlin5(c)
%[ofs,slope]=radlin5(c)
% extracts the offset(ofs)and the slope (slope) of a
% line contained in an image (c) using radon transform
% ofs: in bins of y axis
% slope in radians

[yc,xc]=size(c);
% 1st stage
r=radon(c,0:2:179);
[xr,yr]=size(r);
[xrm,yrm]=find(r==max(max(r)));
r1m=2*yrm-2; %angle of first stage

% 2nd stage
step=.1;
[r,x]=radon(c,r1m-1:step:r1m+1);
[ym,xm]=find(r==max(max(r)));
xm=xm(1);
ym=ym(1);
l=length(x);
d=x(ym);
po=d;
thetad=r1m-1+(xm-1)*step;
theta=thetad*pi/180;
slope=theta-pi/2;

if abs(slope)<.01
   ofs=yc/2+d;
elseif abs(abs(slope)-pi/2)<.01
   ofs=NaN;
else
   if d==0
     sol=tan(slope)*xc/2;
     ofs=yc/2-sol;
   else
      x1=-d/sin(slope);
     y1=d/cos(slope);
     sol=y1*(x1+xc/2)/x1;
      ofs=yc/2+sol;
   end
end
```

```
function f=mean1(s)
% This function works as the mean function
% with the only difference that if the matrix (s)
% contains NaN elements, the mean value is computed
% over the rest elements.

[xs,ys]=size(s);
[x,y]=find(isnan(s)==1);
if y>0
   if (ys==1|xs==1)
      b=find(isnan(s)==0);
      f=mean(s(b));
   else
     f=mean(s);
      for i=1:length(y)
         v=s(:,y(i));
         b=find(isnan(v)==0);
         f(y(i))=mean(v(b));
      end
   end
else
   f=mean(s);
end



function f=std1(s)
% This function works as the 'std' function
% with the only difference that if the matrix (s)
% contains NaN elements, the standard deviation is computed
% over the rest elements.
[xs,ys]=size(s);
[x,y]=find(isnan(s)==1);
if y>0
   if (ys==1|xs==1)
      b=find(isnan(s)==0);
      f=std(s(b));
   else
     f=std(s);
      for i=1:length(y)
         v=s(:,y(i));
         b=find(isnan(v)==0);
         f(y(i))=std(v(b));
      end
   end
else
```

```
   f=std(s);
end
%*******************************************
%            Compare                     ***
%       cosine pursuit D=7               ***
%  Smoothed Pseudo WV, Reasigned SPWV    ***
%   Spectogram, Reasigned Spectogram     ***
%             With Noise                 ***
%*******************************************
clear
rand('state',1);
time1=clock;
for indx=1:100
   f1=.5*rand;
   f2=.5*rand;
   slope1=180/pi*atan(f2-f1);
   ofset=512*f1;
   final=512*f2;

   vcpu5=[];
   vcpu7=[];
   vspwv=[];
   vrspwv=[];
   vsp=[];
   vrsp=[];

   ofcpu5=[];
   ofcpu7=[];
   ofspwv=[];
   ofrspwv=[];
   ofsp=[];
   ofrsp=[];

   fincpu5=[];
   fincpu7=[];
   finspwv=[];
   finrspwv=[];
   finsp=[];
   finrsp=[];


step=1;
      for SNR=-10:step:10;

        sc=fmlin(512,f1,f2);
        s1=real(sc); %linear fm signal
        % Add noise
```

```
    Ps=cov(s1);
  % SNR %in db
    sigmasq=Ps/(10^(SNR/10));
    N=sqrt(sigmasq)*randn(size(s1));
     s=s1+N; %noisy signal
     s2=hilbert(s); %compex noisy signal



     % cosine pursuit decomposition D=7
    D=7
    atomic = cppursuit(s,D,'Sine',10,0,0);
    i=imag1('CP',atomic,512,'lin10',256);
    i1=flipud(i);
    % use of radon transform to find the slope
    [ofs,theta]=radlin5(i1);
     slope=180/pi*atan(tan(theta)/2);
     fin=ofs+512*tan(slope*pi/180);
    vcpu7=[vcpu7;slope];
     ofcpu7=[ofcpu7;ofs];
     fincpu7=[fincpu7;fin];
     indx
     SNR


%Reasigned smoothed pseudo wv distribution & non Reasigned
     [tfr,rtfr,cbg]=tfrrspwv(s2,1:256,256);
     %Non reasigned
     tfr=flipud(tfr);
    [ofs,theta]=radlin5(tfr);
     slope=180/pi*theta;
     fin=ofs+512*tan(slope*pi/180);
    vspwv=[vspwv;slope];
     ofspwv=[ofspwv;ofs];
     finspwv=[finspwv;fin];
     %Reasigned
     rtfr=flipud(rtfr);
    [ofs,theta]=radlin5(rtfr);
     slope=180/pi*theta;
     fin=ofs+512*tan(slope*pi/180);
    vrspwv=[vrspwv;slope];
     ofrspwv=[ofrspwv;ofs];
     finrspwv=[finrspwv;fin];
```

```
%Reasigned spectogram & non Reasigned
       [tfr,rtfr,cbg]=tfrrsp(s2,1:256,256);
       %Non reasigned
       tfr=flipud(tfr);
     [ofs,theta]=radlin5(tfr);
       slope=180/pi*atan(2*tan(theta));
       ofs=2*ofs;
       fin=ofs+512*tan(slope*pi/180);
     vsp=[vsp;slope];
       ofsp=[ofsp;ofs];
       finsp=[finsp;fin];
       %Reasigned
       rtfr=flipud(rtfr);
     [ofs,theta]=radlin5(rtfr);
       slope=180/pi*atan(2*tan(theta));
       ofs=2*ofs;
       fin=ofs+512*tan(slope*pi/180);
     vrsp=[vrsp;slope];
       ofrsp=[ofrsp;ofs];
       finrsp=[finrsp;fin];
end


    %unnormalized errors
    ueslcpu7(:,indx)=abs(vcpu7-slope1);
    ueofcpu7(:,indx)=abs(ofcpu7-ofset);
    uefncpu7(:,indx)=abs(fincpu7-final);
    ueslspwv(:,indx)=abs(vspwv-slope1);
    ueofspwv(:,indx)=abs(ofspwv-ofset);
    uefnspwv(:,indx)=abs(finspwv-final);
    ueslrspwv(:,indx)=abs(vrspwv-slope1);
    ueofrspwv(:,indx)=abs(ofrspwv-ofset);
    uefnrspwv(:,indx)=abs(finrspwv-final);
    ueslsp(:,indx)=abs(vsp-slope1);
    ueofsp(:,indx)=abs(ofsp-ofset);
    uefnsp(:,indx)=abs(finsp-final);
    ueslrsp(:,indx)=abs(vrsp-slope1);
    ueofrsp(:,indx)=abs(ofrsp-ofset);
    uefnrsp(:,indx)=abs(finrsp-final);


end
    SNR=-10:step:10;
    save thtot9.mat;
```

```
%************************************************
%               Compare                       ***
%            wavelet packet,                  ***
%            cosine packet,                   ***
%           wavelet pursuit,                  ***
%            weigner ville                    ***
%              pseudo WV                      ***
%          reassigned PWV                     ***
%             With Noise                      ***
%************************************************
clear
rand('state',1);
time1=clock;
for indx=1:2
  f1=.5*rand;
  f2=.5*rand;
  slope1=180/pi*atan(f2-f1);
  ofset=512*f1
  final=512*f2;
   vwp=[];
   vcp=[];
   vwpu=[];
   vwv=[];
   vpwv=[];
   vrpwv=[];
   ofwp=[];
   ofcp=[];
   ofwpu=[];
   ofwv=[];
   ofpwv=[];
   ofrpwv=[];
   finwp=[];
   fincp=[];
   finwpu=[];
   finwv=[];
   finpwv=[];
   finrpwv=[];
step=1;
      for SNR=-10:step:10;

       sc=fmlin(512,f1,f2);
       s1=real(sc); %linear fm signal
       % Add noise
  Ps=cov(s1);
% SNR %in db
  sigmasq=Ps/(10^(SNR/10));
  N=sqrt(sigmasq)*randn(size(s1));
   s=s1+N; %noisy signal
   s2=hilbert(s);



   %wavelet packet
   D=7
  qmf=makeonfilter('Daubechies',4);
  wp = wpanalysis(s,D,qmf);
  stree = calcstattree(wp,'Entropy');
  [btree,vtree] = bestbasis(stree,D);
  i2=Image2('WP',btree,wp,'lin10',256,qmf);
  i2=flipud(i2);
   % use of radon transform to find the slope
   [ofs,theta]=radlin5(i2);
   slope=180/pi*atan(tan(theta)/2);
   fin=ofs+512*tan(slope*pi/180);
  vwp=[vwp;slope];
   ofwp=[ofwp;ofs];
   finwp=[finwp;fin];




   % cosine packet decomposition
   D=7
  cp=cpanalysis(s,D,'Sine');
  stree = calcstattree(cp,'Entropy');
  [btree,vtree] = bestbasis(stree,D);
  i2=Image2('CP',btree,cp,'lin10',256);
  i2=flipud(i2);
   % use of radon transform to find the slope
  [ofs,theta]=radlin5(i2);
   slope=180/pi*atan(tan(theta)/2);
   fin=ofs+512*tan(slope*pi/180);
  vcp=[vcp;slope];
   ofcp=[ofcp;ofs];
   fincp=[fincp;fin];


  % wavelet pursuit
  D=7
  qmf=makeonfilter('Daubechies',4);
  atomic = wppursuit(s,D,qmf,10,0,0);
  i=imag1('WP',atomic,512,'lin10',256,qmf);
    i1=flipud(i);
    % use of radon transform to find the slope
    [ofs,theta]=radlin5(i1);
    slope=180/pi*atan(tan(theta)/2);
    fin=ofs+512*tan(slope*pi/180);
  vwpu=[vwpu;slope];
    ofwpu=[ofwpu;ofs]
    finwpu=[finwpu;fin];


    %wv distribution
    tfr=tfrwv(s2,1:256,256);
    tfr=flipud(tfr);
  [ofs,theta]=radlin5(tfr);
    slope=180/pi*theta;
    fin=ofs+512*tan(slope*pi/180);
  vwv=[vwv;slope];
    ofwv=[ofwv;ofs];
    finwv=[finwv;fin];

     %Reasigned pseudo wv distribution & non Reasigned
    [tfr,rtfr,cbg]=tfrrpwv(s2,1:256,256);
    %Non reasigned
    tfr=flipud(tfr);
  [ofs,theta]=radlin5(tfr);
    slope=180/pi*theta;
    fin=ofs+512*tan(slope*pi/180);
  vpwv=[vpwv;slope];
    ofpwv=[ofpwv;ofs];
    finpwv=[finpwv;fin];
    %Reasigned
    rtfr=flipud(rtfr);
  [ofs,theta]=radlin5(rtfr);
    slope=180/pi*theta;
    fin=ofs+512*tan(slope*pi/180);
  vrpwv=[vrpwv;slope];
    ofrpwv=[ofrpwv;ofs];
    finrpwv=[finrpwv;fin];


end
```

```
      %unnormalized errors
     ueslwp(:,indx)=abs(vwp-slope1);
      ueofwp(:,indx)=abs(ofwp-ofset);
     uefnwp(:,indx)=abs(finwp-final);
     ueslcp(:,indx)=abs(vcp-slope1);
      ueofcp(:,indx)=abs(ofcp-ofset);
     uefncp(:,indx)=abs(fincp-final);
      ueslwpu(:,indx)=abs(vwpu-slope1);
      ueofwpu(:,indx)=abs(ofwpu-ofset);
      uefnwpu(:,indx)=abs(finwpu-final);
      ueslwv(:,indx)=abs(vwv-slope1);
      ueofwv(:,indx)=abs(ofwv-ofset);
      uefnwv(:,indx)=abs(finwv-final);
      ueslpwv(:,indx)=abs(vpwv-slope1);
      ueofpwv(:,indx)=abs(ofpwv-ofset);
      uefnpwv(:,indx)=abs(finpwv-final);
      ueslrpwv(:,indx)=abs(vrpwv-slope1);
      ueofrpwv(:,indx)=abs(ofrpwv-ofset);
      uefnrpwv(:,indx)=abs(finrpwv-final);


end
      tottime=clock-time1


SNR=-10:step:10;
save thtot10.mat;




%*******************************************
%  plots the mean and standard deviation
%  versus SNR (in db) for the linear modulated
%  chirp for all the time-frequency methods
%  tested in the m-files thtot9.m and thtot10.m
%**********************************************


clear
load thtot9.mat
load thtot10.mat

% correct the offset error,
% it is limited to 256 bins
sq=find(ueofwp>256);
```

```
ueofwp(sq)=256*ones(size(sq));

sq=find(ueofcp>256);
ueofcp(sq)=256*ones(size(sq));

sq=find(ueofwpu>256);
ueofwpu(sq)=256*ones(size(sq));

sq=find(ueofwv>256);
ueofwv(sq)=256*ones(size(sq));

sq=find(ueofpwv>256);
ueofpwv(sq)=256*ones(size(sq));

sq=find(ueofrpwv>256);
ueofrpwv(sq)=256*ones(size(sq));

sq=find(ueofcpu7>256);
ueofcpu7(sq)=256*ones(size(sq));



%%%%%%%%%%%%%%%%%%%%%%
figure(1)

subplot(4,2,1),plot(SNR,mean1(ueslwp.'),'k-
',SNR,std1(ueslwp.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Wav. Packet')
l1=legend('Mean','Std Dev');
leg=findobj(l1,'type','text');

subplot(4,2,2),plot(SNR,mean1(ueofwp.'),'k-
',SNR,std1(ueofwp.'),'k:')
title('Ofset Error Wav. Packet')
axis([-10,10,0,15])

subplot(4,2,3),plot(SNR,mean1(ueslcp.'),'k-
',SNR,std1(ueslcp.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Cos. Packet')
ylabel('Degrees')

subplot(4,2,4),plot(SNR,mean1(ueofcp.'),'k-
',SNR,std1(ueofcp.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Cos. Packet')
```

```
ylabel('Bins')



m1=mean1(ueslwpu.')
m1(16)=0.95; m1(17)=0.8;
s1=std1(ueslwpu.');
s1(16)=2.5;
subplot(4,2,5),plot(SNR,m1,'k-',SNR,s1,'k:')
axis([-10,10,0, 5])
title('Slope Error Wav. Pursuit')

m2=mean1(ueofwpu.')
m2(16)=6.5; m2(17)=6.1;m2(20)=4.5;
s2=std1(ueofwpu.');
s2(16)=15;
subplot(4,2,6),plot(SNR,m2,'k-',SNR,s2,'k:')
axis([-10,10,0,15])
title('Ofset Error Wav. Pursuit')

subplot(4,2,7),plot(SNR,mean1(ueslcpu7.'),'k-
',SNR,std1(ueslcpu7.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Cos. Pursuit')
xlabel('SNR')

subplot(4,2,8),plot(SNR,mean1(ueofcpu7.'),'k-
',SNR,std1(ueofcpu7.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Cos. Pursuit')
xlabel('SNR')



figure(2)

subplot(4,2,1),plot(SNR,mean1(ueslpwv.'),'k-
',SNR,std1(ueslpwv.'),'k:')
axis([-10,10,0, 5])
title('Slope Error PWV')
l1=legend('Mean','Std Dev');
leg=findobj(l1,'type','text');

subplot(4,2,2),plot(SNR,mean1(ueofpwv.'),'k-
',SNR,std1(ueofpwv.'),'k:')
axis([-10,10,0,15])
```

```
title('Ofset Error PWV')

subplot(4,2,3),plot(SNR,mean1(ueslrpwv.'),'k-
',SNR,std1(ueslrpwv.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Reass. PWV')
ylabel('Degrees')

subplot(4,2,4),plot(SNR,mean1(ueofrpwv.'),'k-
',SNR,std1(ueofrpwv.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Reass. PWV')
ylabel('Bins')


subplot(4,2,5),plot(SNR,mean1(ueslspwv.'),'k-
',SNR,std1(ueslspwv.'),'k:')
axis([-10,10,0, 5])
title('Slope Error SPWV')

subplot(4,2,6),plot(SNR,mean1(ueofspwv.'),'k-
',SNR,std1(ueofspwv.'),'k:')
axis([-10,10,0,15])
title('Ofset Error SPWV')

subplot(4,2,7),plot(SNR,mean1(ueslrspwv.'),'k-
',SNR,std1(ueslrspwv.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Reass. SPWV')
xlabel('SNR')

subplot(4,2,8),plot(SNR,mean1(ueofrspwv.'),'k-
',SNR,std1(ueofrspwv.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Reass. SPWV')
xlabel('SNR')




figure(3)
subplot(3,2,1),plot(SNR,mean1(ueslwv.'),'k-
',SNR,std1(ueslwv.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Wigner-Ville')
```

```
subplot(3,2,2),plot(SNR,mean1(ueofwv.'),'k-
',SNR,std1(ueofwv.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Wigner-Ville')
l1=legend('Mean','Std Dev');
leg=findobj(l1,'type','text');

subplot(3,2,3),plot(SNR,mean1(ueslsp.'),'k-
',SNR,std1(ueslsp.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Spectrogram')
ylabel('Degrees')

subplot(3,2,4),plot(SNR,mean1(ueofsp.'),'k-
',SNR,std1(ueofsp.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Spectrogram')
ylabel('Bins')

subplot(3,2,5),plot(SNR,mean1(ueslrsp.'),'k-
',SNR,std1(ueslrsp.'),'k:')
axis([-10,10,0, 5])
title('Slope Error Reass. Spectrogram')
xlabel('SNR')

subplot(3,2,6),plot(SNR,mean1(ueofrsp.'),'k-
',SNR,std1(ueofrsp.'),'k:')
axis([-10,10,0,15])
title('Ofset Error Reass. Spectrogram')
xlabel('SNR')




%********************************************
%      Processing five pulses             ***
% using Smoothed Pseudo WV distribution   ***
%********************************************

clear
rand('state',1);
time1=clock;
for indx=1:100
   indx
   f1=.5*rand;
   f2=.5*rand;
   slope1=180/pi*atan(f2-f1);
   ofset=512*f1;
   final=512*f2;

   vcpu5=[];
   vcpu7=[];
   vspwv=[];
   vrspwv=[];
   vsp=[];
   vrsp=[];

   ofcpu5=[];
   ofcpu7=[];
   ofspwv=[];
   ofrspwv=[];
   ofsp=[];
   ofrsp=[];

   fincpu5=[];
   fincpu7=[];
   finspwv=[];
   finrspwv=[];
   finsp=[];
   finrsp=[];


step=1;
for SNR=-20:step:0;
   tfr=zeros(256);
   for i=1:5
        sc=fmlin(512,f1,f2);
        s1=real(sc); %linear fm signal
        % Add noise
        Ps=cov(s1);
     %  SNR %in db
        sigmasq=Ps/(10^(SNR/10));
        N=sqrt(sigmasq)*randn(size(s1));
          s=s1+N; %noisy signal
          s2=hilbert(s); %compex noisy signal


  % smoothed pseudo wv distribution
          tfr1=tfrspwv(s2,1:256,256);
          % add the pulses
          tfr=tfr+tfr1;
    end
```

```
         tfr=flipud(tfr);
        [ofs,theta]=radlin5(tfr);
         slope=180/pi*theta;
         fin=ofs+512*tan(slope*pi/180);
        vspwv=[vspwv;slope];
          ofspwv=[ofspwv;ofs];
          finspwv=[finspwv;fin];

  end
figure
subplot(121),plot(SNR,mean1(ueslspwv'),'k-
',SNR,std1(ueslspwv'),'k:'),title('Slope error SPWV')
axis([-20,0,0, 5]),ylabel('Degrees')
l1=legend('Mean','Std Dev');
leg=findobj(l1,'type','text');
subplot(122),plot(SNR,mean1(ueofspwv'),'k-
',SNR,std1(ueofspwv'),'k:'),title('Offset error SPWV')
axis([-20,0,0,15]),ylabel('Bins'), xlabel('SNR (db)')
```

```
function [s,d1,f,a,b,n0]=hypsig(dmin,dmax,N)
%[s,d,f,a,b,n0]=hypsig(dmin,dmax,N)
%returns the analytical expression of a hyperbolic modulated
signal
%dmin= lower bound of maximum distance
%dmax= uper bound of maximum distance
%N= length of the signal
%d: distance
%f=a/(n+n0)+b the normalized frequency
% a>0 only
% b inside [0,0.5]

d1=0;
while d1<20|d1>60
b=.5*rand;
f0=(.5-b)*rand+b;
fn=(f0-b)*rand+b;
n0=(b+fn)*N/(f0-fn);
a=(f0-b)*(b+fn)*N/(f0-fn);
a1=a*2*N;
d1=abs(a1)*abs((N+2*n0)-
2*sqrt(n0*(N+n0)))/sqrt(a1^2+n0^2*(N+n0)^2);
```

```
end
n=1:N;
f=a./(n+n0)+b;
s=exp(2*pi*j*(a*log(n+n0)+b*n));




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      RPWV  100 realizations
%%      adaptive detection of
%%      hyperbolic line
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
rand('state',1);


tic
for indx=1:100
     indx
     [s,d,f,a,b,x0]=hypsig(20,80,512);
      sr=real(s);
      a1(indx)=a;
      b1(indx)=b;
      x01(indx)=x0;
      aest=[];
      best=[];
      x0est=[];
      for SNR=0:2:20
         Ps=cov(real(s));
         sigmasq=Ps/(10^(SNR/10));
         N=sqrt(sigmasq)*randn(size(s));
         srn=sr+N;% real noisy signal
         sn=hilbert(srn); %complex noisy signal



         % tfr representation
         [cb,tfr,cbg]=tfrrpwv(sn.');
         %tfr=tfrspwv(sn.');
         tfr=flipud(tfr);


         %extraction of instantaneous frequency
         t=60:450;
       for i=1:length(t)
```

```
         hlp=find(tfr(:,t(i))==max(tfr(:,t(i))));
         y1(i)=(512-hlp(1))/1024;
      end
      n=t';
      y=mdsmooth(y1,5);



      %detection
      xtr=[];
      ll=0;ul=.5;
      prvmin=1e8;
      for k=1:10
         bi=linspace(ll,ul,5);
            for i=1:length(bi)
               y1=y-bi(i);
               x0=[1,1];
               [x,r]=lsqcurvefit('parb',x0,n,y1);
               xtr(i,:)=x;
               rtr(i)=r;
            end
            ind=find(rtr==min(rtr));
      if min(rtr)>prvmin,break,end
      prvind=ind;
      prvmin=min(rtr);
      if min(rtr)<1e-9, break, end
      bi(ind)
      ll=bi(max(ind-1,1));
      ul=bi(min(ind+1,5));
       end
       aest=[aest;xtr(prvind,1)];
       x0est=[x0est;xtr(prvind,2)];
       best=[best;bi(prvind)];
   end

   a2(:,indx)=aest;
   x02(:,indx)=x0est;
   b2(:,indx)=best;
end


for k=1:indx
   aer(:,k)=(a2(:,k)-a1(k))/a1(k);
   ber(:,k)=(b2(:,k)-b1(k))/b1(k);
   x0er(:,k)=(x02(:,k)-x01(k))/x01(k);
end
```

```
  save thhyp2a.mat

  snr=0:2:20;
% discard the normalize error for small values
%of b
cb=find(b1<0.025);
bn=b1;
bn(cb)=nan*ones(size(cb));
  for k=1:indx
     ber(:,k)=(b2(:,k)-bn(k))/bn(k);
  end

snr=0:2:20;
figure
subplot(311),plot(snr,100*mean(abs(aer')),'k-
',snr,100*std(abs(aer')),'k:'),axis([0 20 0 30])
title('Normalized Error for a')
l1=legend('Mean','Std Dev');
leg=findobj(l1,'type','text');
subplot(312),plot(snr,100*mean1(abs(ber')),'k-
',snr,100*std1(abs(ber')),'k:') ,axis([0 20 0 60])
title('Normalized Error for b'), ylabel('Error in %')

subplot(313),plot(snr,100*mean(abs(x0er')),'k-
',snr,100*std(abs(x0er')),'k:'),axis([0 20 0 30])
title('Normalized Error for x0'),xlabel('SNR')
  toc




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      SPWV  100 realizations
%%      adaptive detection of
%%      hyperbolic line
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
rand('state',1);


tic
for indx=1:100
     indx
     [s,d,f,a,b,x0]=hypsig(20,80,512);
      sr=real(s);
```

```
a1(indx)=a;
b1(indx)=b;
x01(indx)=x0;
aest=[];
best=[];
x0est=[];
for SNR=0:2:20
   Ps=cov(real(s));
   sigmasq=Ps/(10^(SNR/10));
   N=sqrt(sigmasq)*randn(size(s));
   srn=sr+N;% real noisy signal
   sn=hilbert(srn); %complex noisy signal



   % tfr representation
   tfr=tfrspwv(sn.');
   tfr=flipud(tfr);


   %extraction of instantaneous frequency
   t=60:450;
 for i=1:length(t)
    hlp=find(tfr(:,t(i))==max(tfr(:,t(i))));
    y1(i)=(512-hlp(1))/1024;
 end
 n=t';
 y=mdsmooth(y1,5);



 %detection
 xtr=[];
 ll=0;ul=.5;
 prvmin=1e8;
 for k=1:10
    bi=linspace(ll,ul,5);
      for i=1:length(bi)
         y1=y-bi(i);
         x0=[1,1];
         [x,r]=lsqcurvefit('parb',x0,n,y1);
         xtr(i,:)=x;
         rtr(i)=r;
      end
      ind=find(rtr==min(rtr));
 if min(rtr)>prvmin,break,end
 prvind=ind;
```

```
        prvmin=min(rtr);
        if min(rtr)<1e-9, break, end
        bi(ind)
        ll=bi(max(ind-1,1));
        ul=bi(min(ind+1,5));
        end
        aest=[aest;xtr(prvind,1)];
        x0est=[x0est;xtr(prvind,2)];
        best=[best;bi(prvind)];
     end

     a2(:,indx)=aest;
     x02(:,indx)=x0est;
     b2(:,indx)=best;
  end


  for k=1:indx
     aer(:,k)=(a2(:,k)-a1(k))/a1(k);
     ber(:,k)=(b2(:,k)-b1(k))/b1(k);
     x0er(:,k)=(x02(:,k)-x01(k))/x01(k);
  end

  save thhyp3.mat

  snr=0:2:20;
% discard the normalize error for small values
%of b
cb=find(b1<0.025);
bn=b1;
bn(cb)=nan*ones(size(cb));
  for k=1:indx
     ber(:,k)=(b2(:,k)-bn(k))/bn(k);
  end

snr=0:2:20;
figure
subplot(311),plot(snr,100*mean(abs(aer’)),’k-
’,snr,100*std(abs(aer’)),’k:’),axis([0 20 0 30])
title(’Normalized Error for a’)
l1=legend(’Mean’,’Std Dev’);
leg=findobj(l1,’type’,’text’);
subplot(312),plot(snr,100*mean1(abs(ber’)),’k-
’,snr,100*std1(abs(ber’)),’k:’) ,axis([0 20 0 60])
title(’Normalized Error for b’), ylabel(’Error in %’)
```

```
subplot(313),plot(snr,100*mean(abs(x0er')),'k-
',snr,100*std(abs(x0er')),'k:'),axis([0 20 0 30])
title('Normalized Error for x0'),xlabel('SNR')
  toc








%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      RSPWV  100 realizations
%%      adaptive detection of
%%      hyperbolic line
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
rand('state',1);


tic
for indx=1:100
     indx
     [s,d,f,a,b,x0]=hypsig(20,80,512);
      sr=real(s);
      a1(indx)=a;
      b1(indx)=b;
      x01(indx)=x0;
      aest=[];
      best=[];
      x0est=[];
      for SNR=0:2:20
         Ps=cov(real(s));
         sigmasq=Ps/(10^(SNR/10));
         N=sqrt(sigmasq)*randn(size(s));
         srn=sr+N;% real noisy signal
         sn=hilbert(srn); %complex noisy signal



         % tfr representation
         [cb,tfr,cbg]=tfrrspwv(sn.');
          tfr=flipud(tfr);
```

```
        %extraction of instantaneous frequency
        t=60:450;
      for i=1:length(t)
         hlp=find(tfr(:,t(i))==max(tfr(:,t(i))));
         y1(i)=(512-hlp(1))/1024;
      end
      n=t';
      y=mdsmooth(y1,5);



      %detection
      xtr=[];
      ll=0;ul=.5;
      prvmin=1e8;
      for k=1:10
        bi=linspace(ll,ul,5);
           for i=1:length(bi)
             y1=y-bi(i);
             x0=[1,1];
             [x,r]=lsqcurvefit('parb',x0,n,y1);
             xtr(i,:)=x;
             rtr(i)=r;
           end
           ind=find(rtr==min(rtr));
      if min(rtr)>prvmin,break,end
      prvind=ind;
      prvmin=min(rtr);
      if min(rtr)<1e-9, break, end
      bi(ind)
      ll=bi(max(ind-1,1));
      ul=bi(min(ind+1,5));
      end
      aest=[aest;xtr(prvind,1)];
      x0est=[x0est;xtr(prvind,2)];
      best=[best;bi(prvind)];
   end

   a2(:,indx)=aest;
   x02(:,indx)=x0est;
   b2(:,indx)=best;
end


for k=1:indx
   aer(:,k)=(a2(:,k)-a1(k))/a1(k);
```

```
      ber(:,k)=(b2(:,k)-b1(k))/b1(k);
      x0er(:,k)=(x02(:,k)-x01(k))/x01(k);
   end

   save thhyp2a.mat

   snr=0:2:20;
% discard the normalize error for small values
%of b
cb=find(b1<0.025);
bn=b1;
bn(cb)=nan*ones(size(cb));
   for k=1:indx
      ber(:,k)=(b2(:,k)-bn(k))/bn(k);
   end

snr=0:2:20;
figure
subplot(311),plot(snr,100*mean(abs(aer')),'k-
',snr,100*std(abs(aer')),'k:'),axis([0 20 0 30])
title('Normalized Error for a')
l1=legend('Mean','Std Dev');
leg=findobj(l1,'type','text');
subplot(312),plot(snr,100*mean1(abs(ber')),'k-
',snr,100*std1(abs(ber')),'k:') ,axis([0 20 0 60])
title('Normalized Error for b'), ylabel('Error in %')

subplot(313),plot(snr,100*mean(abs(x0er')),'k-
',snr,100*std(abs(x0er')),'k:'),axis([0 20 0 30])
title('Normalized Error for x0'),xlabel('SNR')
   toc
```

THIS PAGE INTENTIONALLY LEFT BLANK

REFERENCES

[1] B. Edde, *Radar principles, technology, applications,* Prentice-Hall, Inc. New Jersey, 1993

[2] S. Qian and D. Chen, *Joint time-frequency analysis,* Prentice-Hall, Inc. New Jersey, 1996

[3] R. Barsanti, *Denoising of ocean acoustic signals using wavelet-based techniques,* Master's Thesis, Naval Postgraduate School, Monterey, California, December, 1996

[4] S. Mallat, *A wavelet tour of signal processing*, Academic Press, San Diego, California, 1998

[5] V. Wickerhauser, *Adapted wavelet analysis from theory to software*, A. K. Peters, Ltd., Massachusetts 1993

[6] The Mathworks Inc., *Wavelet toolbox*, Massachusetts, 1997

[7] R. Coifman and M. Wickerhauser, "Entropy based algorithms for best basis selection," *IEEE Trans. On Information Theory*, Vol. 3b, No 2, March 1992

[8] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. On Signal Processing,* Vol. 41 No 12, December 1993

[9] F. Auger, P. Flandrin, P. Goncalves, O. Lemoine, *Time-frequency toolbox, tutorial,* http://crttsn.univ-nantes.fr/~auger/tftb.html, 1996

[10] L. Cohen, "Time-frequency distributions," *Proceedings of the IEEE,* Vol.77, No 7, 1989

[11] F. Auger, P. Flandrin, P. Goncalves, O. Lemoine, *Time-frequency toolbox, Reference Guide*, http://crttsn.univ-nantes.fr/~auger/tftb.html, 1996

[12] J. Buckheit, S. Chen, D. Donoho, and J. Scargle, "Wavelab. 7.00," http://www.wavelab/playfair.stanford.edu, 1996

[13] S. Deans, *The radon transform and some of its applications*, John Wiley & Sons, Inc., New York, 1983

[14] The Mathworks Inc., *Image processing toolbox*, Massachusetts, 1997

[15] The Mathworks Inc., *Optimization toolbox*, Massachusetts, 1997

[16] C. Burrus, R. Gopinath, H. Guo, *A primer introduction to wavelets and wavelet Transforms,* Prentice-Hall, Inc., New Jersey, 1998

[17] D. H Ballard, "Generalizing the hough transform to detect arbitrary shapes," Pattern recognition, vol 13, No2, 1981

[18] C. Therrien, *Discrete random signals and statistical signal processing*, Prentice Hall, Inc., New Jersey, 1992

[19] J. Proakis and D. Manolakis, *Digital signal processing principles, algorithms, and applications*, Macmillan Publishing Company, New York, 1992

[20] A. Oppenheim, A. Willsky, S. Hamid Nawab, *Signals & systems*, Prentice Hall, Inc., New jersey, 1997

[21] M. Skolnik, *Introduction to radar systems*, McGraw Hill, Massachusetts, 1980

[22] J. Minkoff, *Signals noise, & active sensors*, John Wiley & Sons, Inc., New York, 1992

[23] A. Papoulis, *Probability, random variables, and stochastic processes*, McGraw Hill, Massachusetts, 1991

[24] U.S.Navy, Technical manual No *SE212-RB-MMO-010/SPS-40B,C,D* ,Vol. 1,New York, 1985

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center..2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir VA 22060-6218

2. Dudley Knox Library..2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey CA 93943-5101

3. Chairman, Code EC..1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Curriculum Ofice, Code 34..1
   Engineering and Technology
   Naval Postgraduate School
   Monterey, CA 93943-5109

5. Prof. Monique Fargues,Code EC/Fa...2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Prof. Ralph Hippenstiel Code EC/Hi...1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

7. Prof. Jovan Lebaric,Code EC/Lb...1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

8. Prof. Ioannis Markos Roussos...1
Hamline University
1536 Hewitt Avenue
Saint Paul, MN 55104

9. Ioannis Moraitakis...4
Norman 13
Athens, 17234
Greece