National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Filterbank Implementations of a Window Based Gabor Transform for SAR Processing

by

Capt. Devashish Paul, B. Eng, MBA.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

Joint Time-Frequency Analysis (JTFA) is used in applications such as music, radar, and sonar, where the frequency content of a signal during successive time intervals must be determined. Synthetic Aperture Radar (SAR) is a specific radar application where JTFA is crucial for the generation of images because the frequency content of the radar signal is required during successive windows of time. Denis Gabor presented a JTF technique where a signal could be described as the sum of shifted and modulated gaussian functions that are weighed by complex constants. These constants are the Gabor Transform coefficients and they describe the magnitude and the time during which certain frequencies occur in a signal. It was not possible to calculate the Gabor Transform coefficients using an analytic equation until only recently. The formulation of an analysis equation for the Gabor Transform by Bastiaans has permitted the practical implementation of the Gabor Transform on digital computers.

In this thesis a Window Based Gabor Transform (WBGT) is presented that is suitable for real time digital signal processing. Other Gabor Transform approaches presented in the literature such as the Orthogonal Like Discrete Gabor Transform (OLGT) are not suitable for real time processing. The reduction in analysis time using the WBGT instead of the OLGT is proportional to the ratio of the analysis window to the signal length. Using the WBGT the number of JTF coefficients required to accurately represent the original signal is far less than when using the STFT. Two filterbank implementations for the WBGT which reduce analysis times are developed, one using serial input and the second using a parallel input, each of which computes the frequency content of the signal using N parallel band pass filters. In the case of the parallel input filterbank, the analysis times are similar to those using a Short Time Fourier Transform with FFT. The parallel input filterbank can be implemented using a matrix formulation for the N band pass filters. This approach is suitable for software implementation and results in analysis times that are lower than when using an STFT or the parallel input implementation with the bank of band pass filters. The WBGT, along with the filterbank implementations result in very accurate JTF spectra. Near perfect reconstruction of the original signal from the JTF spectra was obtained for several cases for the Gabor Transform techniques presented in this thesis. Finally, the use of the parallel input filterbank in the context of SAR processing is presented and the hardware requirements are derived and shown to be achievable using commercially available products.

Acknowledgments

I wish to express my most sincere gratitude to Dr. Willem Steenaart, my thesis supervisor for his patience and the invaluable guidance, and support that he provided in my preparation of this thesis. In addition to this I would like to thank the Department of National Defence for a year and a half of post graduate training sponsorship and Dr. M. Vant of the Defence Research Establishment Ottawa for suggesting the pursuit of a thesis in the area of the Gabor Transform.

Finally, the author wishes to thank his wife, Roxanne for all her encouragement and understanding throughout the period of his post graduate studies.

Contents

List of Figures

List of Tables

Acronyms and Definitions

BPF	Band Pass Filter
CF	Canadian Armed Forces
DFT	Discrete Fourier Transform
DND	Department of National Defence
DREO	Defence Research Establishment Ottawa
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IFFT	Inverse Fast Fourier Transform
JTF	Joint Time-Frequency
JTFA	Joint Time-Frequency Analysis
LPF	Low Pass Filter
MDA	MacDonald Dettwiler and Associates
OLGT	Orthogonal Like discrete Gabor Transform
RMS	Root Mean Square Error
SAR	Synthetic Aperture Radar
STFT	Short Time Fourier Transform
TWBGT	Truncated Window Based Gabor Transform
WBGT	Window Based Gabor Transform
XDM	Exploratory Development Model

List of Symbols

b	basis function used in Gabor Transforming
β	azimuth angle (arctan of cross range/range ratio)
B_W	bandwidth of SAR transmitted signal
B_D	bandwidth of cross range doppler frequency offset
c	speed of light
$C_{m,n}$	Gabor Transform Coefficients
d	distance in cross range between successive pulses
D_m	Time shift used in filterbank diagrams
$D(n)$	Frequency Response of Cross Range Matched Filter
Δf	frequency bandwidth over which most of the energy of the Gabor basis's frequency response is concentrated
ΔM	time shift of analysis and synthesis window in Gabor Transforming
ΔN	frequency interval between frequency samples in Gabor Transform
ΔT	shift of gaussian basis and of STFT analysis window
E	Error in calculating the analysis function
ξ	Oversampling Rate
f_{proc}	Processing power (clock rate) of a digital signal processor
$g(k)$	Impulse response of Band Pass Filter used in filterbank
γ	analysis function used for Gabor Transforming
$\Gamma(f)$	Frequency response of analysis function γ
i	time index used for discrete samples
k	position of sample in a given window
κ	chirp constant for linear FM signal
L	number of time samples in sequence
L_a	physical length of radar antenna
L_{ap}	physical length of synthetic aperture
L_W	number of time samples in analysis and basis functions (this is also number of pulses transmitted over the synthetic aperture)
L_{WT}	number of time samples in a truncated window
λ	wavelength of transmitted SAR radio frequency
m	index for time shifts
n	index for frequency samples

M	total number of shifts that cover sequence
N	number of frequency samples spaced equally around unit circle
N_{rg}	number of range gates in a radar swath
f_p	pulse repitition frequency of transmitted SAR RF signal
t_{matrix}	analysis time when using matrix implementation of parallel input filterbank
$t_{parallel}$	analysis time when using parallel input filterbank
t_S	time savings
t_{serial}	analysis time when using serial input filterbank
t_{OLGT}	analysis time when using OLGT
t_{STFT}	analysis time when using STFT
t_{WBGT}	analysis time when using WBGT
t_{TWBGT}	analysis time when using TWBGT
t_y	the time taken for the aircraft to travel a distance y in cross range
T_{add}	time taken for an adder to execute one complex addition
T_{del}	time between arrival of successive samples to filterbank
T_{latch}	time between input and output of each sample at a delay
T_{mult}	time taken for a multiplier to execute one complex multiplication
T_{PW}	pulse width of transmitted SAR RF signal
T_s	sampling rate in cross range dimension
R	range to point scatterer from radar
r	reconstruction of original signal from JTFA data
s	original transmitted radar signal that is analysed
σ	Standard Deviation of the the gaussian basis function
t	time (analog)
θ_b	half power beamwidth of transmitted radiation pattern
$w(k)$	windowing function used for STFT
$X_{m,n}$	Time-Frequency coefficients achieved by Short Time Fourier Transforming
y	offset from perpendicular in cross range coordinate for stripmapping

Chapter 1: Introduction

In a paper published in 1946, Denis Gabor [4] presented a method of representing signals with both time and frequency axes. This initiated the development and application of Joint Time-Frequency Analysis (JTFA) of signals. Gabor's representation, often referred to as the Gabor Transform was rarely used for signal processing applications because time-frequency coefficients could not be calculated by a direct analysis equation but instead had to be determined by an iterative technique. The formulation of an analysis equation for the Gabor Transform has permitted the practical implementation of the Gabor Transform on digital computers. In this thesis a Window Based Gabor Transform (WBGT) is developed that is suitable for real time processing. Filterbank implementations of the WBGT are then developed and are shown to be suitable for use in processing Synthetic Aperture Radar (SAR) images.

1.1 The Motivation for Synthetic Aperture Radar Systems

SAR is a technology that has been used for remote sensing, defence, forestry and geophysics to generate images of objects, terrain or sea from airborne radars. SAR is an active system that transmits electromagnetic radiation at microwave frequencies and produces an image of the object that is being viewed by recording the energy reflected from it and then processing this information.

Unlike technologies that employ optical frequencies, which may only be used in the day, SAR can be used at any time, because it receives part of the energy that it transmits by reflection from objects. In addition to this, microwave frequencies are not attenuated or scattered by bad weather (as are optical frequencies) and can therefore be used to image areas that may be covered by clouds.

The Canadian Forces (CF) have a use for SAR in both maritime patrol and search and rescue. Maritime patrol aircraft fly along Canada's coastlines, monitoring territorial waters. Tasks such as identifying foreign fishing trawlers that may be operating illegally, surface vessel search and tracking or carrying out drug interdiction are conducted. Search and

rescue operations can require aircraft to locate, identify and make contact with ships in distress. Given that both Atlantic and Pacific coasts are often covered by cloud and fog, there is a requirement to view the ocean surface from aircraft even though optical visibility may be close to zero. This is where SAR is extremely useful.

1.2 Need for Joint Time-Frequency Analysis in SAR

In the era prior to the existence of digital computers, the processing of SAR signals posed a severe technical challenge [1]. Early processors were based on optical systems that used lenses and photographic film to generate SAR images. These systems were large and heavy and were unsuitable for aircraft installations and as a result, could not be used for real time processing in airborne environments [2].

Digital signal processing techniques have been widely used in SAR to generate images ever since the size and speed of computer hardware made it possible. The first commercial airborne SAR processor for non-military applications is believed to be the MacDonald Dettwiler and Associates (MDA) system built for the Canadian Centre for Remote Sensing in 1979 [1].

SAR images are generated by determining the energy of the transmitted radar signal reflected from each resolvable pixel of terrain, the position of which is determined by the range (distance from the aircraft, or more specifically, from the radar antenna) and cross range (coordinate perpendicular to the range axis). Range information is determined by the time frame during which a reflected signal is received and cross range information is gained by calculating the frequency content of the signal in a given time frame. The details of SAR processing are presented in chapter 2.

Joint Time-Frequency Analysis (JTFA) techniques are used for applications where there is a requirement to determine the frequency content of a signal during successive finite time intervals of the signal. This is in contrast with traditional frequency analysis techniques where the frequency content is determined for the signal over its entirety. Thus, JTFA techniques are well suited to SAR where for every range gate (time frame), it is required that the cross range (frequency content) be calculated.

The JTFA technique most commonly used in SAR has been the Short-Time Fourier Transform (STFT). The STFT calculates the Discrete Fourier Transform (DFT) for a finite length window that is shifted along the length of a longer sequence. The Fast Fourier Transform (FFT) is the technique that is generally employed to calculate the DFT used in Short Time Fourier Transforming. Using the STFT one can calculate the frequency spectrum of a signal for a given window of time. The STFT has been used for SAR processing in an Exploratory Development Model (XDM) radar employed by the CF [3].

Although several techniques have been developed to improve the speed of the FFT (which the STFT employs), the STFT suffers a few drawbacks. The STFT generates as many frequency domain samples as there are time domain samples for every window of time domain data. This means that the wider the time window, the larger the number of frequency coefficients required to represent the information contained in the original signal. This obviously slows down the speed of processing STFT data. If a small time shift (relative to window length) is desired between successive calculations of the local frequency spectrum, then a very large quantity of JTF data is generated over the length of the signal. This can become a problem especially when this data must be transmitted (such as from an aircraft to a ground replay facility) or when post processing operations are performed on this large amount of data.

By taking rectangular windows (weighing the input signal by unity), the frequency content for the window of data will contain some amount of ripple (because the real frequency content is essentially being convolved with a sinc function) and thus distort the actual frequency content of that window of data. The general practice applied to solve this problem has been to weigh the input signal by Hamming, Kaiser or Gaussian windows. This however, results in signal information that is weighed out (towards the window edges). As a result of this, considerable overlap is required between adjacent windows to ensure that information about the original signal is not lost. This requirement can further slow down processing of the STFT.

The Gabor Transform (GT) is a JTFA technique that was first developed by Denis Gabor [4]. Gabor was dissatisfied with traditional frequency analysis techniques that did not yield any information about the time frame during which particular frequencies occurred. In a paper, titled "Theory of Communication", published in 1946, Gabor showed that a

infinite non-stationary signal could be expressed as an expansion of shifted and frequency modulated gaussian basis functions. The weighting (or Gabor Coefficients $C_{m,n}$) of each of these bases is the frequency content, n of the signal in a given time frame, m. Gabor did not present an analytical technique for determining $C_{m,n}$, but instead used an iterative technique. In essence, Gabor presented a synthesis equation but not an analysis equation.

Even though it was recognized that the Gabor Transform had potential advantages for various signal processing uses, it was not used for industrial applications until an analysis equation was developed. The development of an analysis-synthesis transform pair in the past two decades has led to applications of the Gabor Transform in industry. The CF have an interest in using the Gabor Transform in SAR processing. This is because claims have been made [5] that it can improve the speed of generating images and can improve the resolution of those images when compared to using a traditional STFT approach.

1.3 Problem Statement

The Gabor Transfrom has been identified as a possible process to be used for generating SAR images. However, its structure is not readily suited to handling SAR signals since the calculation of the Gabor Transform, as described in the literature [6,7] , requires the position of a particular data sample in the overall sequence of the digitized returned signal. It is also unsuitable for processing SAR signals because it can only be calculated once the signal is received in its entirety. This makes it unsuitable for real time processing. The simultaneous reception of a signal and JTFA processing is not possible. The goal of this thesis is to study the performance of the digital implementation of the Gabor Transform and implement it in filterbank structures that could be used to generate SAR images in real time in an airborne environment.

1.4 Thesis Presentation

In this thesis, the Gabor Transform is studied and modified into a Window Based Gabor Transform (WBGT) which is suitable for real time implementation. The WBGT is then implemented in filterbank configurations for SAR image processing.

In Chapter 2, a review of the principles of SAR and the motivation for employing JTFA techniques for SAR image processing is presented. The second half of this chapter

reviews two JTFA techniques: the STFT and the Gabor Transform. These two techniques are described since the STFT is used in SAR image processing and the Gabor Transform has been recognized as a possible candidate for implementation in SAR. The JTFA aspects of Gabor's Theory of Communication are introduced and the problems associated with the Gabor Transform are discussed. The relationship between the Gabor Transform and Wavelet Theory is also briefly discussed.

In Chapter 3, the Orthogonal Like Gabor Transform (OLGT), which is a digital implementation of the Gabor Transform is reviewed. A theory developed by Wexler and Raz [6] for determining the analysis window $\gamma(i)$ for finite length input sequences is presented and then Qian and Chen's [7] extension of this theory to infinite length sequences is described.

In Chapter 4, a Window Based Gabor Transform (WBGT) that can be used for analysis and synthesis of non-stationary non-periodic infinite length sequences is developed. It is suitable for real time processing. The WBGT is based on the OLGT presented in chapter 3. Another approach using truncated versions of the windows used in the WBGT and titled the Truncated WBGT (TWBGT) is also proposed. This version uses analysis and synthesis functions of reduced length in an effort to decrease processing speed without losing significant signal information. The STFT, OLGT, the WBGT and TWBGT are simulated in software and analysed to determine their performance in terms of computational speed (for the calculation of Gabor Transform coefficients) and the error for signal reconstruction from Gabor Transform coefficients.

In Chapter 5, two filterbank implementations for the WBGT are proposed. One approach uses serial signal input to a bank of Band Pass Filters (BPF), while the other uses a parallel signal input. The parallel input implementation is achieved by employing a serial to parallel converter for the input signal. A bank of equally spaced Finite Impulse Response (FIR) BPF's is used for the filtering and decimation is used to control the overlap between successive windows. The filterbanks are digitally simulated and compared with the results of the OLGT, WBGT, and the TWBGT. The parallel input implementation is then performed using a matrix formulation that is suitable when software implementation of the WBGT is desired. The use of the parallel input filterbank is then shown in the context of SAR processing and the hardware requirements for the filterbank are determined.

Chapter 6 provides a conclusion with recommendations for further study.

Chapter 2: Background

A review of some of the principles of SAR and an introduction to JTFA is provided in this chapter. The cross range resolution of a conventional radar is determined by its half power beamwidth [2,8]. This can be improved by placing a number of antennae in a linear array [8]. In SAR systems, a linear array is simulated by transmitting pulses at successive intervals and then processing the returns, simulating that they were transmitted simultaneously as in a linear array. The result is an image of far superior quality than if only one pulse was used. Joint Time Frequency (JTF) processing is used to describe a signal by both time and frequency domain information. Using JTF techniques, one is able to determine the time frame during which a given frequency occurred. This is essential to the formulation of SAR images.

Section 2.1 of this chapter contains a review of SAR outlining the range and cross range processing of a SAR signal. Section 2.2 provides an introduction to JTFA. The two JTFA techniques studied are the Short Time Fourier Transform (STFT) and the Gabor Transform (GT).

2.1 A Review of Synthetic Aperture Radar

SAR can be used to generate images of terrain, sea surface or man made objects from an airborne or spaceborne platform. The application of a synthetic aperture permits the airborne radar operator to achieve resolution far superior to that of using the physical radar aperture only. Superior resolution of SAR results in a finer image quality than when using the radar antenna's physical aperture only.

2.1.1 SAR Configuration

The concept of SAR is similar to using a linear array of antennae. The longer the array, the better its resolution, or half power beamwidth [8], in cross range, where cross range is the coordinate along which array elements are placed. The SAR system is configured as shown below:

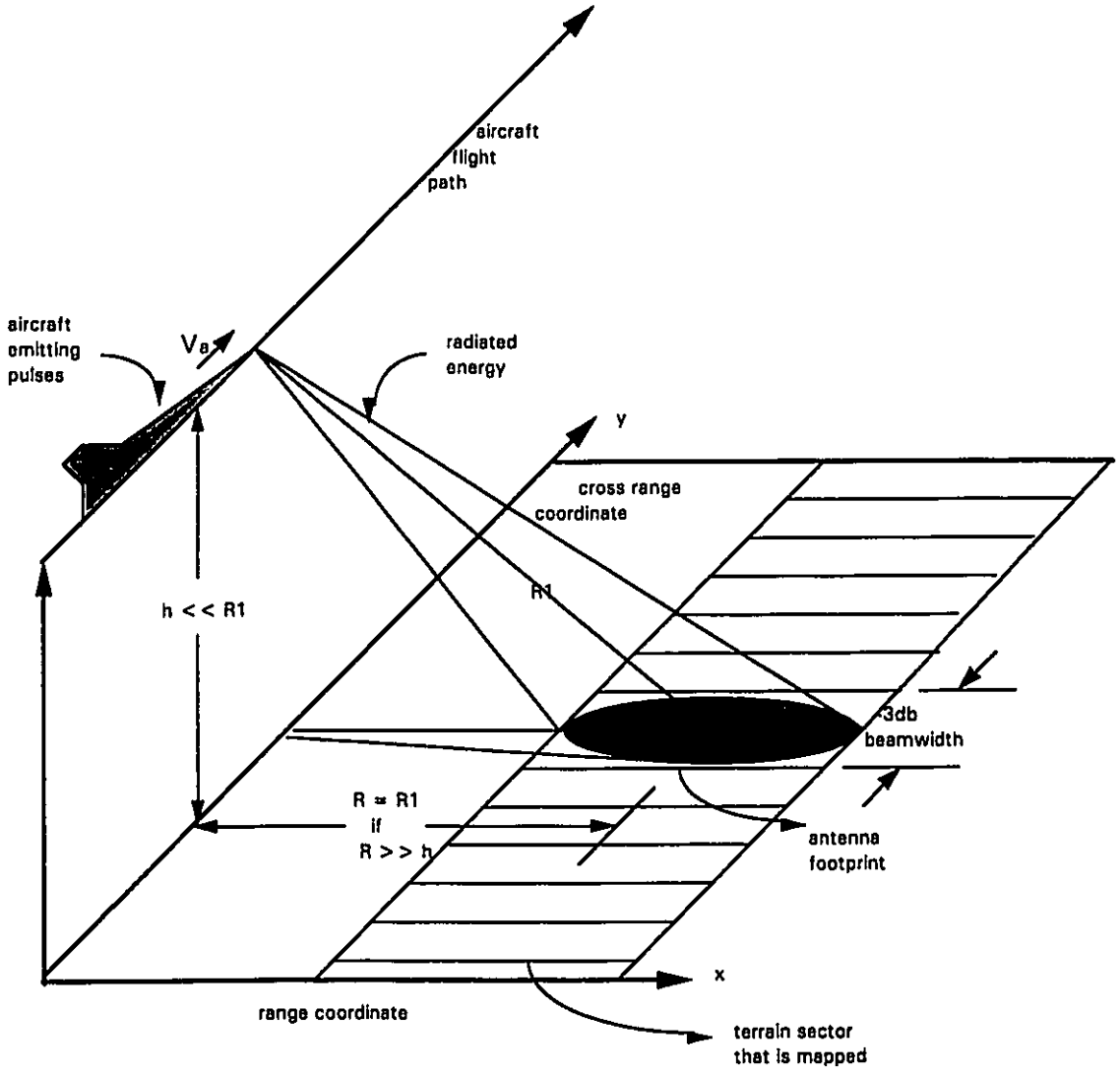

Figure 2-1: SAR Configuration for Stripmapping in Broadside Mode

In the figure 2-1, the coordinate parallel to the flight path (along track coordinate) is cross range, y and the one perpendicular to it (on the surface being imaged) is range, R. The radar emits successive pulses at a given pulse repetition frequency, f_p , in a direction perpendicular to the aircraft flight path (along the range coordinate), illuminating an area of terrain referred to as the antenna footprint. The antenna footprint is that area contained within the -3db beamwidth of the antenna radiation pattern, as shown in figure 2-1. This is the broadside configuration which is used in stripmapping SAR. The broadside configuration results in range and cross range return signals that have no cross correlation [1,9]. This permits independent processing of the range and cross range coordinates.

The distance that the aircraft travels between successive pulses is given by [1]:

$$d = V_a/f_p \qquad 2\text{-}1$$

where V_a is the aircraft velocity. The distance, d, is analogous to the inter-element spacing in a physical array of antennae. At an aircraft velocity V_a = 200 m/s and f_p = 1000 hz, d = 0.2 m. To generate a synthetic aperture of physical length L_{ap}, The radar returns are stored at each transmission and then the number of pulses transmitted during a length L_{ap} are combined and processed coherently (as if they had all been transmitted and received simultaneously). The number of pulses transmitted over the length of the synthetic aperture is L_w and is related to L_{ap} by:

$$L_{ap} = L_w d \qquad 2\text{-}2$$

The limit on the effective length of the synthetic aperture (L_{ap}) is the width of the antenna half power radiation pattern at a range of R (see figure 2-1) which is given by:

$$L_{ap} \leq R\theta_b \qquad 2\text{-}3$$

In (2-3), θ_b is the half power beamwidth of the antenna radiation pattern in radians. At a range R = 20 km and $\theta_b = 2°$, the maximum length of the synthetic aperture is L_{ap} = 698 m. For SAR, L_{ap} will be slightly less than the width of the half power beamwidth unless focused SAR is used [1]. The limitation of (2-3) is imposed by the physical width of the radar beam because pulses can only be considered for use in synthetic aperture processing from a point scatterer during the time frame that the point was illuminated by the radar beam and the range to those points are effectively constant at R [1]. The half power beamwidth can be approximated by:

$$\theta_b = \lambda/L_a \qquad 2\text{-}4$$

where L_a is the physical length of the antenna aperture and λ is the wavelength of the radiated signal energy.

The reflected radar signal following each pulse will consist of the superposition of the reflectivity from every point within the area illuminated by the pulse. The location of each point can be determined by digital or optical processing techniques [2, 11].

2.1.2 SAR Processing

The SAR processor computes the amount of signal energy reflected by each range - cross range pixel (R,y) to generate an image. Provided that the radar transmits in a broadside configuration (as in figure 2-1), digital processing can be decoupled into range and cross range [9] processes. The range and cross-range processes are described below.

2.1.2.1 Range Processing

Conventional radars transmit short pulses (in time duration) and achieve range resolution by separating the reflections returning from different parts of the ground by differing arrival times [12]. The time at which a radar reflected signal is received yields the distance to the reflecting object. All reflections received at the same time originate from points at equal range. These points form an isorange contour. The range resolution process determines the reflectivity, g(R), from an isorange contour of the terrain at a distance R. Mathematically this can be described by:

$$g(R) = \int_{-0.5L_{ap}}^{0.5L_{ap}} g(R,y)dy \qquad 2\text{-}5$$

(2-5) assumes that the isorange contour is effectively a straight line along the cross range coordinate (this assumption is valid for unfocused SAR but requires some phase correction if focused SAR is used [14]. All physical points that fall within a pulse will appear as one point to the radar. As a result, the narrower the pulse (in time) the better the resolution in the range coordinate. An approximation of the system range resolution is given by:

$$\delta_R = cT_{PW}/2 \qquad 2\text{-}6$$

where c is the speed of light and T_{PW} is the signal pulsewidth in time. The shorter the T_{PW}, the better the range resolution. A short pulse however limits the amount of energy

that can be transmitted (the peak power has to be increased to increase the total amount of energy). This results in a pulse that would not have adequate energy in it to produce the sufficient signal to noise ratio for reliable detection [1]. An effectively narrow pulse along with a sufficient amount of transmitted energy (for a suitable signal to noise ratio) can be achieved by using pulse compression. Pulse compression actually uses a physically wide pulse which means that more power can be transmitted than when using a narrow pulse [9, 10]. However, the pulse compression process results in an effectively narrow pulse resulting in superior range resolution.

Pulse compression is achieved by transmitting a signal that is a linear FM chirp pulse [1, 10] and then filtering the reflected signal (after demodulation) with a matched filter. The matched filter's impulse response is the time reversed complex conjugate of the transmitted signal. Generally, the transmitted linear FM pulse [1] is of the form:

$$s(t) = \mathrm{rect}(t/T_{PW})\exp(j2\pi t(f_c + 0.5\kappa t)) \qquad 2\text{-}7$$

where f_c is the carrier frequency, and κ is the chirp constant for the linear FM signal. The frequency of the signal varies in a linear manner with time and the phase factor varies quadratically (with t^2). The longer the pulse the greater the frequency range contained in the transmitted signal. In fact, from the above it is evident that every frequency contained in the pulse can be mapped to a time within the pulse through the linear FM property. The linear FM bandwidth (B_W) of the transmitted signal is given by:

$$B_W = 0.5\kappa T_{PW} \qquad 2\text{-}8$$

The output of the matched filtering (where the reflected signal is filtered by the time reversed complex conjugate of the transmitted signal) is effectively g(R) convolved with a sinc function [1]. The narrower the sinc function the closer the approximation of g(R). As g(R) convolved with a unit impulse gives g(R), and since a sinc function in the limit approaches an impulse, an approximation of g(R) is obtained by the matched filtering process. For linear FM pulses with a large time bandwidth product this sinc function is approximated by $\mathrm{sinc}(2\pi B_W R/c)$, where 2R/c is the round trip travel time for the signal reflecting from a point at a distance R from the radar. The range resolution obtainable using matched filtering is given by [1]:

$$\delta_R = c/2B_W \qquad 2\text{-}9$$

Pulse compression can be achieved in the time domain by convolving the digitized returned radar signal with the matched filter impulse response. A more affective approach used in most SAR systems is to take the Fourier Transform of the returned signal and multiply it with the Fourier Transform of the matched filter and then taking the Inverse Fourier Transform of the product to achieve the pulsed compressed output (this is shown in figure 2-3).

As the aircraft flies along its flight path, pulses are transmitted in the direction of the range coordinate as shown in figure 2-1. The returns from each pulse is digitized and digitally pulse compressed using FFT's to yield a range line. A range line contains the reflectivity of the terrain between the innermost and outermost resolvable ranges within the antenna footprint for a given pulse. Range lines are stored for each successive pulse, forming a two dimensional array as shown below in figure 2-2. The column index of this array is the range line, and the row index contains the resolvable ranges (or range gates) for each range line. This array is then passed to the cross range processor so that the final SAR image can be determined by calculating the intensity of the signal for every range - cross range pixel.

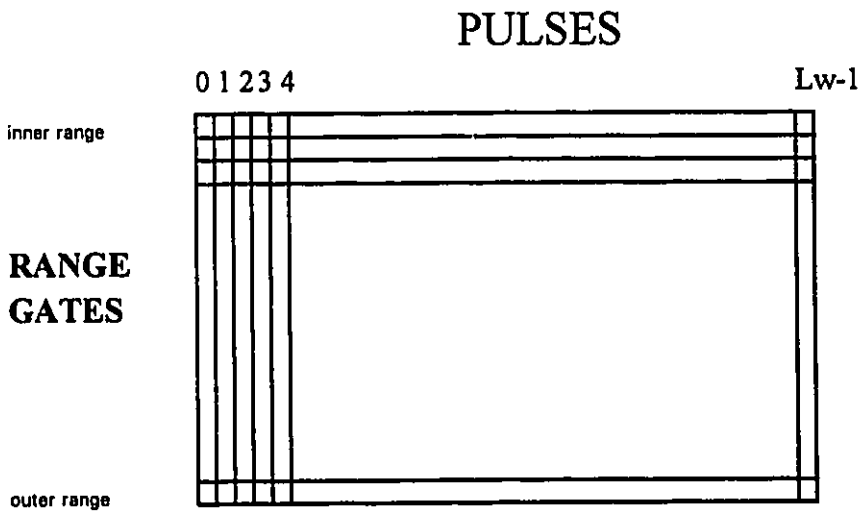

Figure 2-2: Array containing the returns from L_W successive pulses

2.1.2.2 Cross Range Processing

Since the range processing operation computes the signal energy in each range gate for a given range line (or pulse), it is the the cross range processing operation that determines the energy reflected from each point in cross range within each range gate. In SAR, cross range processing is based on doppler filtering [10]. The key observation that ultimately led to SAR and the vastly improved resolution in cross range (compared to using real aperture radar) dates from about 1951 and is attributed to Wiley (1965) [11]. He observed that two point targets, at slightly different angles (off broadside) with respect to the track of the moving radar, have different speeds at any instant relative to the radar. Therefore, the radar pulse when reflected from two targets will have two distinct Doppler frequency shifts.

Point scatterers that are within the half power beamwidth of the radiation pattern but are off the beam centre (see figure 2-1) will form an angle that is slightly off broadside when compared to the aircraft flight path. The angle subtended by the scatterer, the antenna and the flight path is the azimuth angle, β, which is given by:

$$\tan \beta = y/R \qquad 2\text{-}10$$

where y is the offset from beam centre in cross range and x the range. For SAR, the range to the illuminated area is much larger than the offset in cross range R>>y so $\tan(\beta) \approx \beta$ (β measured in radians). Thus

$$\beta \approx y/R \qquad 2\text{-}11$$

The relative motion between the antenna and the ground gives rise to a doppler frequency shift. The doppler frequency shift at all points within the physical radar beam with azimuth angle, β, is given by f_d:

$$f_d = (2V_a \sin \beta)/\lambda \qquad 2\text{-}12$$

From (2-11) $\beta \approx$ y/R and for the angles used in SAR (unfocused [1]) it can be assumed that:

$$\sin \beta \approx y/R \qquad 2\text{-}13$$

Substituting (2-13) in (2-12), the doppler frequency can be described by:

$$f_d = (2V_a\, y)/R\lambda \qquad 2\text{-}14$$

Noting that the time taken to for the aircraft to travel the distance y is given by:

$$t_y = y/V_a \qquad 2\text{-}15$$

and substituting (2-15) in (2-14) it is seen that, the doppler frequency across the radar beamwidth varies in a linear FM fashion [1,3] and is given by:

$$f_d = (2V_a^2\, t_y)/R\lambda \qquad 2\text{-}16$$

By substituting B_D, for f_d and L_{ap} for y in (2-14), the doppler bandwidth, B_D is calculated to be:

$$B_D = (2V_a\, L_{ap})/R\lambda \qquad 2\text{-}17$$

At V_a = 200 m/s, L_{ap} = 698m, R = 20 km and at an X band frequency of 1.3 GHz, this would correspond to a B_D = 60.4 hz.

Given that the SAR reflected signal will have a doppler shift that varies in a linear FM fashion (2-16) [1,10], successive windows of range gate data can be filtered with appropriate matched filters to yield a pulse compressed output [1,3] that yields the energy reflected from each cross range point within each range gate. The matched filter impulse response will be a pulse, frequency modulated by the varying doppler frequency given by (2-16). Since there is a range dependency in (2-17) the cross range matched filter will ideally have a different impulse response for each range gate. However, in practice, five to six matched filters are used to filter several range gate since there isn't a large change in the impulse response of the cross range matched filters between successive range gates [3,22]. The output is of the cross range matched filter is an estimate of the terrain reflectivity in cross range for a given range gate. The achievable resolution in time of the cross range matched filters is the width of the cross range compressed pulse and is given

by $1/B_D$. The resulting cross range resolution is the distance the aircraft travels during the width of the main lobe of the cross range compressed pulse. This is given by:

$$\delta_y = V_a/B_D \qquad 2\text{-}18$$

By substituting (2-17) in (2-18), the theoretical cross range resolution is given by

$$\delta_y = R\lambda/2L_{ap} \qquad 2\text{-}19$$

At L_{ap} = 698m, R = 20 km and at an X band frequency of 1.3 GHz, this would correspond to a δ_y = 3.3m.

For focused SAR [1] where the length of the synthetic aperture is also the width of the radar half power radiation pattern (2-3) and noting that θ_b is given by (2-4) the resulting limit on cross range resolution is given by

$$\delta_y = L_a/2 \qquad 2\text{-}20$$

(2-20) shows that the theoretical limit on cross range resolution is independent of all parameters except the antenna's physical aperture length.

The two - dimensional array containing the range contribution for each pulse is used to determine the SAR image. This array contains the range information for each pulse. By performing cross range matched filtering on successive windows of elements of length L_W (which is equivalent to the number of pulses transmitted during the time in which the aircraft flies a distance equal to the radar half power beamwidth) in each range gate, the intensity of the signal from each range - cross range pixel is determined. A window of length L_W is successively shifted along the row of data in a range gate as range processed data is received. The algorithm for the two dimensional processing operation is shown if figure 2-3.

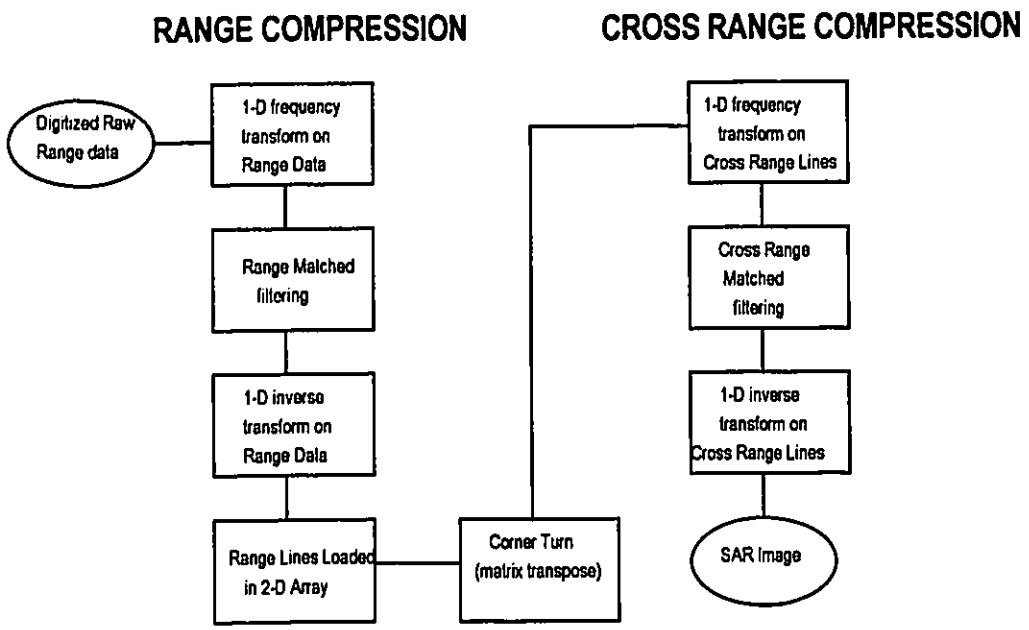

Figure 2-3: Block Diagram of SAR Processor

In figure 2-3, cross range compression is performed after the range processing is complete on a block of L_W range lines. Although the first round of cross range processing occurs after L_W range lines have been range processed, subsequent blocks of L_W range lines are cross range processed after the next ΔM range lines have been range processed, where ΔM is a time shift of range lines that is less than L_W. The amount of time shift, ΔM, between successive cross range computations is determined by the system and the image processing requirements. In systems where processing time is a concern, a large time shift, ΔM between successive windows should be chosen. This is often required where it is desirable to generate SAR signals in real time. The overlap between successive windows for a large shift is small and this results in a coarse image. When image quality is the main objective, a smaller time shift between successive windows is chosen. This leads to a smoother updates of the SAR image. One drawback is that a large amount of JTF data is generated in doing so. The processing time is increased and any image processing that is to be performed at a later time must handle a larger set of data.

From the above discussion it is seen that SAR processing lends itself readily to JTFA, since it is required that the successive windows of data in each range gate data have to be frequency transformed prior to cross-range pulse compression.

2.1.3 Time - Frequency Analysis for SAR

A SAR signal is non-stationary and will change as the terrain reflectivity varies. The statistics and the frequency content of the SAR signal will vary over time. Fourier frequency analysis does not provide a suitable representation of a SAR signal, because it does not provide any temporal information related to the occurrence of a given frequency. The Fourier transform is an integral over the entire time extent of a signal and therefore provides the frequency content of the signal without relating the occurrence of those frequencies to finite time subintervals of the signal.

The problem in SAR signal processing is to determine the signal energy for each cross range cell within a given range gate. This is performed through a two dimensional compression process as outlined in figure 2-3. The cross range compression process requires the transformation of successive windows of data in each range gate into the frequency domain prior to cross range matched filtering. The solution to this is the application of JTFA on the data in each range gate.

2.2 Review of Joint Time Frequency Analysis Techniques

In the 1940's, the British physicist Denis Gabor, published a paper [4], where he noted that the classical process of describing a signal by either temporal means (time domain) or alternately by Fourier analysis left out a lot of information for many classes of signals. Some examples would be speech, music, sonar, and radar. Time domain analysis does not yield any information about the local frequency content of the signal, while frequency analysis renders no information about the point in time where the signal contained a given frequency. Gabor proposed a theory, which, at the time was a novel method for describing signals. Using Gabor's representation, the signal was defined by both time and frequency along orthogonal axes with intensity (the signal's energy at that time-frequency coordinate) as the third dimension as shown below in figure 2.4.

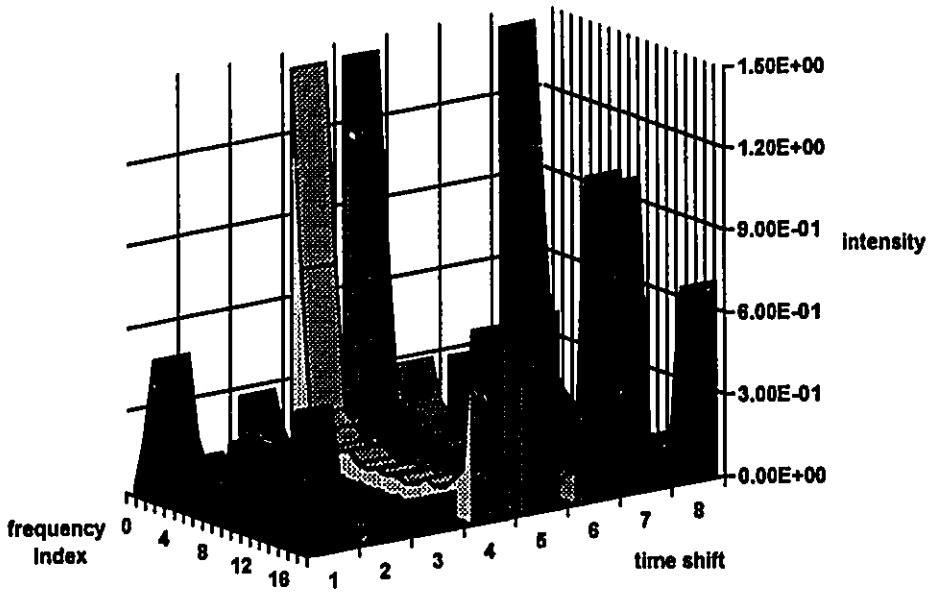

Figure 2-4: The Joint Time Frequency Representation of a Signal

Gabor's theory, which is discussed below, was based on the gaussian function. Due to the complexity in determining the JTF representation using Gabor's Theory, industrial applications were not seen until only recently. Instead, most JTFA had been performed using the Short Time Fourier Transform (STFT). Gabor's theory has itself seen several refinements and modifications in the past few decades and has gone from being a JTFA technique with coefficients that were cumbersome to calculate, to a technique that can be digitally implemented for both analysis and synthesis (reconstruction of original signal from time-frequency data) processes.

2.2.1 The Short Time Fourier Transform

Until recently, the STFT has been predominantly used in most applications where JTFA is employed. The STFT is the Fourier Transform of a windowed section of a longer sequence. The window, w(k) which is limited in time, weighs a finite section of the input signal and suppresses the remainder of the signal so that the spectrum of the windowed samples contains only the frequency content relevant to that window of data (local

frequency content) [13,14]. The window is shifted along the length of the input signal capturing the local frequency information of successive time shifts. The discrete STFT can be defined as:

$$X_{m,n} = DFT\{s(m\Delta T + k)w(k)\} \qquad 2\text{-}21$$

where m is a shift index, ΔT is the time shift increment, n is the discrete frequency which will have a value between 0 and 2π radians on the unit circle spaced apart equally by an interval of $2\pi/L_W$. L_W is the number of discrete time samples in w(k). In SAR processing, L_W is also the number of pulses transmitted over the length of the synthetic aperture. (2-21) can be explicitly written as :

$$X_{m,n} = \sum_{k=0}^{Lw-1} s(m\Delta T + k)w(k)\exp(-j2\pi kn / L_W) \qquad 2\text{-}22$$

where the indices k and n are restricted to the following limits:

$$0 \le n < L_W \quad \text{and} \quad 0 \le k < L_W.$$

In most cases the DFT used in the STFT is implemented using a Fast Fourier Transform (FFT) [15]. The application of the FFT in place of the DFT reduces processing time of each window of data by a ratio of $\{\log_2 L_W\}/L_W$ [15]. Using the STFT, either time domain or frequency domain resolution is optimized. If a long window, w(k) is used, then low frequency transients can be detected but the time domain resolution decreases because of the longer window. Conversely, a short window provides good time domain resolution but poor frequency domain resolution allowing detection of higher frequency content only. For most applications where JTFA is used, the length of the window is pre selected resulting in a fixed time-frequency resolution scale. The STFT is used in SAR processing to transform successive windows of range gate data into the frequency domain prior to the cross range matched filtering process (which is shown in the second column of figure 2-3). For this process, the window length, L_W, is determined by the length of the synthetic aperture as given by (2-2), and from (2-19), the longer the length of the synthetic aperture (or window length), the better the cross range resolution (or frequency resolution).

Chapter 4 contains results from simulations using the STFT and this is compared with Gabor Transform representations introduced in chapters 3 (OLGT) and 4 (WBGT). It should be noted that if identical analysis windows are used and if the number of frequency outputs equals the length of the analysis window, then the STFT and the WBGT are identical. For all other parameters, The STFT is not identical to the WBGT. In general, the STFT uses more coefficients in the JTF spectrum than the Gabor Transform. This is because the STFT generates more frequency outputs from each window than when using the OLGT and the WBGT. Figure 2-5 shows a signal with the corresponding JTF spectrum using the STFT.

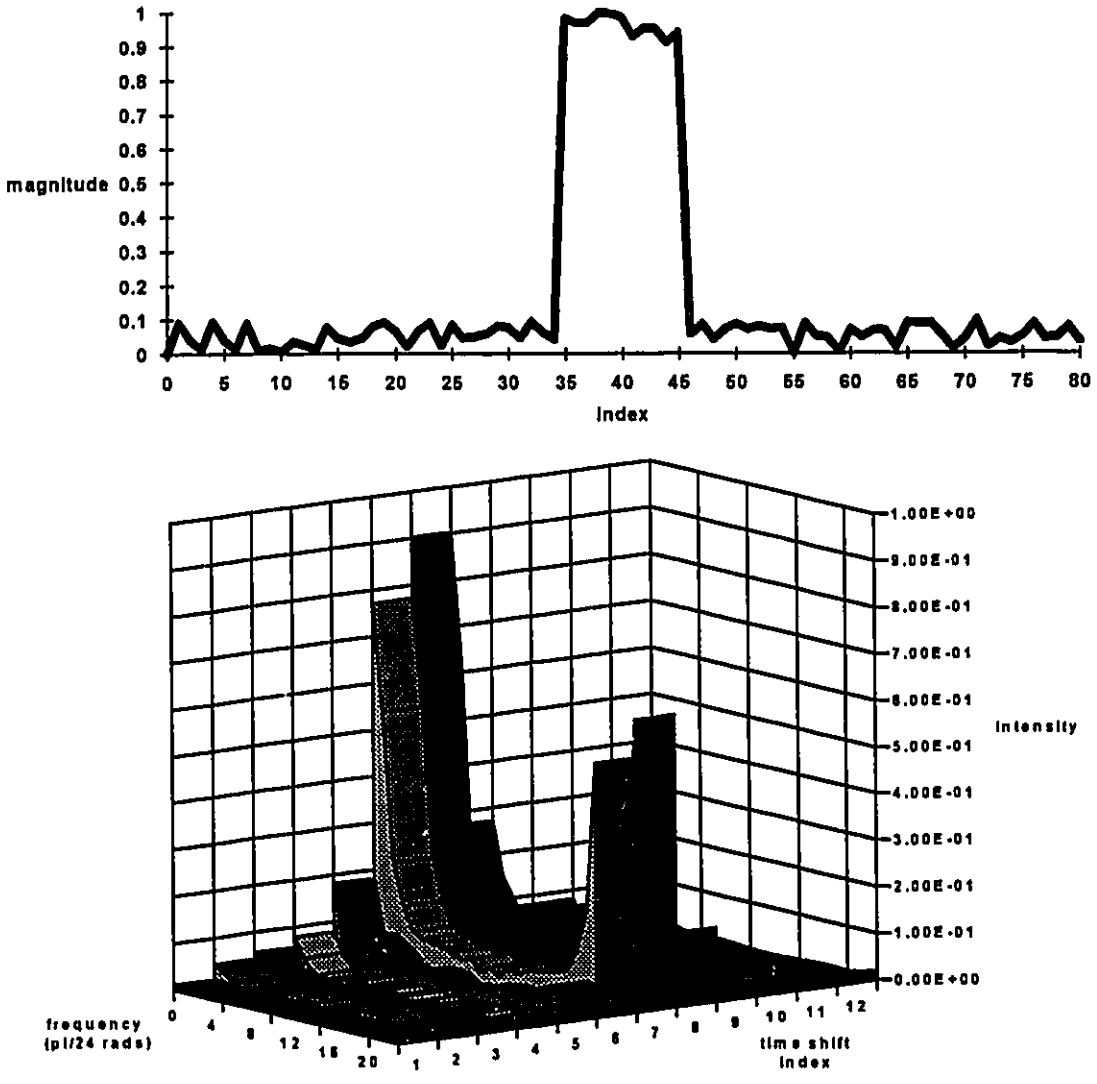

Figure 2-5: A Signal and its Short Time Fourier Transform

2.2.2 Gabor's Theory

The Gabor transform pair can be defined by two equations. The first is the analysis equation and it permits the calculation of the Gabor time-frequency representation from a stationary or non-stationary infinite length signal. The coefficients in the time frequency plane are the Gabor Transform (GT) Coefficients The second is the synthesis equation, and it describes the signal as a sum of weighted, shifted and modulated basis functions. It can be viewed as the reconstruction of the original signal from the GT coefficients. Before the equations are introduced, an overview of Gabor's "Theory of Communications" is presented.

2.2.2.1 The Gabor Transform

In [4], Gabor showed that any signal could be expressed as an expansion of elementary basis functions. These basis functions are gaussian and are weighed by a complex coefficient, shifted in time and modulated by a complex exponential to yield a reconstruction of the original signal s(t) as given by (2-23):

$$s(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{m,n} b(t - m\Delta T) \exp(j2\pi nt / \Delta T) \qquad 2\text{-}23$$

where $C_{m,n}$ are the Gabor coefficients. The $C_{m,n}$ represent the frequency content, n, for the signal s(t) over each shift, m , in time. The function b(t-mΔt) is the shifted gaussian basis and is given by:

$$b(t-m\Delta T) = \exp[-\pi(t-m\Delta T)^2/(2(\Delta T)^2)] \qquad 2\text{-}24$$

where ΔT is the amount of each incremental shift in time and mΔT is the location in time where the shifted basis is centered. The value ΔT is the length of time over which the basis is 'significant', i.e. where most of its energy is concentrated. For the gaussian function, 97% of the energy is concentrated within 2 standard deviations of the median, therefore two to three standard deviations can be considered the 'significant' region [4]. The basis function has the unique property that its Fourier transform is also a gaussian function and this is 'significant' over a length of frequency Δf where Δf = 1/(2ΔT). Each

unit square in the JTF plane is of area $\Delta f \Delta T = 0.5$. According to Gabor, this condition ensured the existence of his representation. Δf is the increment in frequency between samples in the JTF plane. Thus if the basis b(t) is narrow in time then it will be wide in frequency and vice versa. A narrow basis b(t) (for which ΔT is small), results in large increments between adjacent frequency samples in the JTF plane. This means that frequency resolution is somewhat reduced compared to when using a wider basis in time. However a wider basis results in time domain resolution that is reduced compared to using a narrow basis.

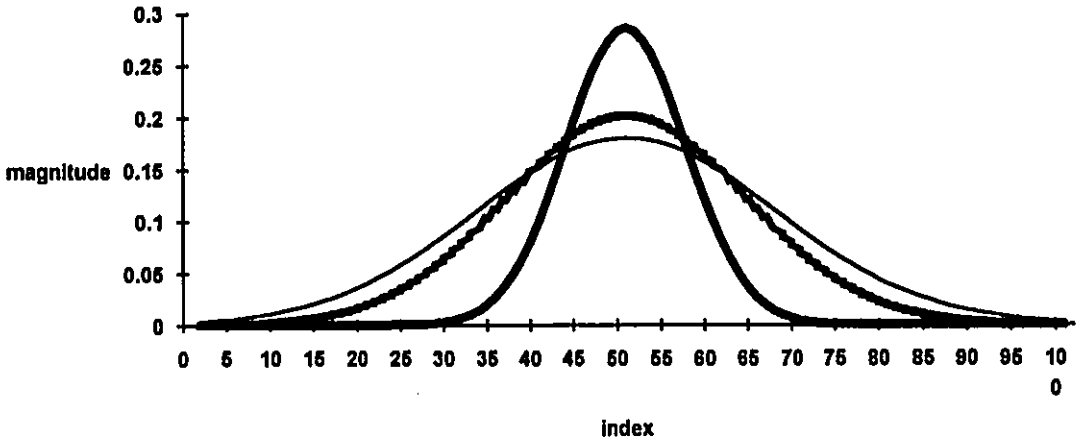

Figure 2-6: Three Different Gaussian Basis Functions for Gabor Transforms

The Gabor representation (2-24) is a 'synthesis equation' only, and does not describe how the coefficients $C_{m,n}$ are calculated. Gabor did not determine an analytical approach to calculating the coefficients $C_{m,n,}$ but instead used an iterative approach based on successive approximations [4]. Since the basis functions are not orthogonal, this makes the direct calculation of $C_{m,n}$ difficult.

2.2.2.2 Equations for Gabor Transform

In the past 15 years, a complete analysis-synthesis approach has been developed for the calculation of the Gabor Transform and the reconstruction of the signal. The breakthrough in formulating an analysis equation was made by Bastiaans [16] in 1980. The Gabor Transform coefficients, $C_{m,n,}$ are given as an integral transform of the signal s(t). The analysis equation as described by Bastiaans and is given by:

$$C_{m,n} = \int_{-\infty}^{\infty} s(t)\gamma^*(t - m\Delta M)\exp(-j2\pi n\Delta N t)dt \qquad 2\text{-}25$$

where the analysis function $\gamma(t)$ is as yet to be defined, but ideally should be concentrated in time to provide a windowing function on the signal s(t). Ideally for JTFA, it should be localized in time to perform a windowing function as was the case for w(k) in the STFT. ΔM is the time shift of the analysis function and $\Delta N = 1/\Delta M$ is the increment between adjacent frequency samples. The synthesis equation is as in (2-21) with a shift of ΔM instead of ΔT and the basis function scaled in width by a factor of π [16]. The new synthesis equation and the corresponding basis function are given by:

$$s(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{m,n} b(t - m\Delta M)\exp(j2\pi n\Delta N t) \qquad 2\text{-}26$$

$$b(t\text{-}m\Delta M) = \exp[-(t\text{-}m\Delta M)^2/(2(\Delta M)^2)] \qquad 2\text{-}27$$

ΔM is the incremental time shift and ΔN is the increment between adjacent frequency samples. The analysis equation is given by the inner product between a shifted analysis function $\gamma(t\text{-}m\Delta M)$ and the original signal. For an analysis-synthesis transform pair to accurately generate the $C_{m,n}$ and reconstruct the signal from them, the analysis function $\gamma(t)$ and the basis b(t) must be such that they satisfy the biorthogonality condition which Bastiaans [16] defined as:

$$\Delta M \Delta N \int_{-\infty}^{\infty} b_{m,n}(t)\gamma^*{}_{k,l}(t)dt = \delta(m-k)\delta(n-l) \qquad 2\text{-}28$$

where

$$b_{m,n}(t) = b(t - m\Delta M)\exp(j2\pi n\Delta N t) \qquad 2\text{-}28a$$

$$\gamma_{k,l}(t) = \gamma^*(t - k\Delta M)\exp(-j2\pi l\Delta N t) \qquad 2\text{-}28b$$

$-\infty \leq k \leq \infty$ and $-\infty \leq l \leq \infty$

Provided that the analysis function $\gamma(t)$ is concentrated in time, then the frequency content calculated for each shift of $\gamma(t)$ will accurately represent the frequency content of the signal that occurs during that shifted section of the signal. This will make the analysis equation useful for JTFA applications [7]. On the other hand, an analysis function $\gamma(t)$ that is spread over a wide time frame results in a calculated frequency spectrum that has no relationship with the actual local frequency content. The requirement of having both $\gamma(t)$ and $b(t)$ concentrated in time for JTFA are conflicting [6,7]. Using the biorthogonality condition (2-28), for a gaussian $b(t)$, $\gamma(t)$ will generally not be localized in time [7] and not gaussian in shape. This would seem to suggest that the analysis equation presented above would be unsuitable for implementation in systems such as SAR, where JTFA is required.

Using the Gabor Transform, time domain and frequency domain resolution is fixed once the length and shape of $\gamma(t)$ are determined using (2-28). For SAR applications this is suitable because a fixed window length is used for cross range processing equal fixed frequency domain resolution is required across the moving radar's doppler bandwidth to develop images with linear scale (such as those obtained at optical frequencies using photographic equipment).

2.2.2.3 Problems in Implementing the Gabor Transform

Although the Gabor Transform has been recognized as being useful for signal processing, its applications have been impeded by difficulties associated with finding the appropriate analysis function $\gamma(t)$ which should be localized in time for JTFA [7]. This is as a result of the biorthogonality condition (2-28) which does not permit both $b(t)$ and $\gamma(t)$ to be localized in time [17]. However, once the $\gamma(t)$ which corresponds to a given $b(t)$ is found, it remains constant throughout the calculation of the $C_{m,n}$, thus the computational burden of determining $\gamma(t)$ is a one time cost.

The Gabor Transform pair is also stated in terms of continuous functions, which requires it to be modified for digital implementation. Recently, the Gabor Transform has been solved for discrete and infinite duration signals, with an appropriately localized analysis function that satisfies the biorthogonality condition [6,7]. This means that the Gabor Transform can be implemented on a computer to analyse input signals that are non-cyclic and have no clearly defined beginning or ending [5]. This would make it suitable for the analysis of SAR signals. The Orthogonal-like discrete Gabor Transform is introduced in chapter 3.

Any Gabor Transform technique that is used for SAR processing must be such that coefficients be generated as soon as successive windows of input data are received to facilitate real time processing. In addition to this the coefficients calculated in a given window of time must independent of the position of the window in the overall input sequence. This is because the SAR process analyses signals for the frequency content in a given time frame, and this analysis must be independent of how that particular time frame fits into the overall input sequence. These issues are addressed in Chapter 4 and a solution to this problem is solved in the form of the Window Based Gabor Transform (WBGT).

2.2.2.4 Relationship betweeen Gabor Transform and Wavelets

There has recently been a large amount of research and application of the Wavelet Transform. The Wavelet Transform is a technique that can be used to process signals where JTFA is required. Wavelet theory states that any time signal can be described as a combination of time-shifted and dilated or compressed basic wavelets [5]. Like the Gabor Transform and the STFT, the Wavelet Transform represents the signal using coefficients along frequency and time axes. However, the scale of the time and frequency axes using the wavelet transform are not linear as is the case for the Gabor Transform. While the Gabor Transform provides constant frequency domain resolution, using an analysis function, $\gamma(t)$ of fixed length and bandwidth, the frequency domain resolution of the Wavelet Transform is varied by changing the length and bandwidth of the basic wavelet [5, 20] by compression or dilation. The wavelet is dilated for analysis of low frequencies and compressed for analysis of high frequencies. However, the basic shape of the wavelet remains the same. Like the analysis function $\gamma(t)$ used in the Gabor Transform, the basic wavelet should be optimally localized in both time and frequency domains [5].

Since the doppler bandwidth of the SAR signal is fixed and is given by (2-17) the fixed frequency domain resolution provided by the Gabor Transfrom or the STFT is adequate for SAR processing. The varying frequency resolution of the Wavelet Transform although beneficial in many applications is unsuitable for SAR processing. This is because the doppler frequency of the reflected radar signal varies in a linear manner across the length of the synthetic aperture as given by (2-16). As a result, fixed frequency domain resolution is required to create a SAR image with linear scales in both the range and cross range dimensions.

2.3 Chapter Summary

An introduction to SAR and JTFA was presented in this chapter. SAR was described as a system that can achieve high resolution in cross range by simulating a long physical antenna through transmission, storing and signal processing of successive pulse returns. Due to the geometry of the SAR system, the processing in range and cross range can be de-coupled. Range processing employs conventional radar techniques of pulse compression. The range resolution was limited by the frequency bandwidth of the linear FM transmitted signal (2-6). Cross range processing is achieved by performing a pulse compression process in the cross range dimension, where the cross range matched filter impulse response is determined by the varying doppler shift across that physical beamwidth. Successive windows of range gate data are filtered by a matched filter whose impulse response is determined by the doppler bandwidth of the returned signal. The crossed range match filtering process requires a frequency transformation of successive windows of range gate data. As a result, JTFA is well suited to SAR processing.

The concept of JTFA was reviewed in this chapter. The STFT was introduced and it was shown to be the Fourier Transform of successive windows of a long signal. Gabor's Theory was also presented in this chapter. His concept was to represent a signal as the sum of weighted shifted and modulated gaussian basis functions. The weights in the Gabor representation were the Gabor Transform coefficients, $C_{m,n}$ and were not obtainable directly through an analysis equation. However Gabor did present an iterative approach to solving for $C_{m,n}$. It was Bastiaans who later solved $C_{m,n}$ using an analysis-synthesis transform pair [16]. The analysis equation developed by Bastiaans is suitable for JTFA provided that a $\gamma(t)$ can be selected that is both concentrated in time and that also satisfies the biorthogonality condition (2-28). The relationship of Wavelet theory to the Gabor Transform was discussed and it was shown that the application of the Wavelet Transform was not ideal for SAR cross range processing because of the requirement for linear frequency scales. It should be noted that the STFT is a subset of the WBGT for certain parameters and this is described in detail in chapter 4.

Chapter 3: The Orthogonal-Like Discrete Gabor Transform

In this chapter the Orthogonal-Like discrete Gabor Transform (OLGT) is introduced. It permits the implementation of the Gabor Transform in practical digital applications. A finite length discrete implementation is first presented. It is used to find a suitable analysis function, $\gamma(i)$ [6]. This is followed by the extension of the finite length implementation to both infinite length sequences and very long sequences [7]. Most of the work covered in this chapter was developed by Wexler and Raz [6] and then subsequently refined by Qian and Chen [7].

In section 3.1 of this chapter, some background information is provided. A discrete implementation of the Gabor Transform for finite length sequences developed by Wexler and Raz is presented in section 3.2. Section 3.3 contains a description of the implementation for infinite length discrete sequences in the form of the Orthogonal-Like discrete Gabor Transform (OLGT) and section 3.4 contains a chapter summary.

3.1 Background

As discussed in chapter 2, some of the problems that have impeded the implementation of the Gabor Transform in industrial applications have been associated with the calculation of the analysis window $\gamma(t)$ that is suitable for JTFA, which corresponds to a given basis function, $b(t)$ as described by the biorthogonality condition (2-28). For the Gabor Transform to be useful in JTFA, it is required that the basis function, $b(t)$, and the analysis function, $\gamma(t)$, are both localized in time. This is a conflicting requirement due to the biorthogonality condition (2-28) and is therefore difficult to achieve [5]. When a gaussian $b(t)$ is used, the corresponding $\gamma(t)$ that satisfies (2-28) is spread out in time and therefore not useful for JTFA.

The Gabor Transform is the transform of a signal $s(t)$ that is windowed by a function $\gamma(t)$ as described by (2-25) [16]. Having $\gamma(t)$ localized in time (energy concentrated in one finite time frame) means that the Gabor Transform coefficients, $C_{m,n}$, for a given time

interval , shifted from the origin, by mΔM, will contain frequency information pertaining to that time interval only. In this context, ΔM is the time frame by which γ(t) is shifted and m is the time shift index.

The problem of determining a discrete analysis window γ(i) which is localized in time was done by Wexler and Raz [6] for oversampled finite length sequences and then refined and expanded for infinite length sequences by Qian and Chen. Their findings are described below. The OLGT is digitally simulated and analysed in chapter 4. It is used as a baseline for comparison with the Window Based Gabor Transform (WBGT) and the Truncated Window Based Gabor Transform (TWBGT), both of which are presented in chapter 4. The OLGT is also used as a baseline in chapter 5 for comparison with the filterbank implementations of the WBGT and TWBGT.

The discrete versions of the continuous time analysis, synthesis and biorthogonality equations (2-25 , 2-26 and 2-28) for the Gabor Transform are given by:

$$C_{m,n} = \sum_{i=0}^{\infty} s(i)\gamma^*(i - m\Delta M)\exp(-j2\pi n\Delta N i) \qquad 3\text{-}1$$

$$s(i) = \sum_{m=0}^{\infty}\sum_{n=0}^{N-1} C_{m,n} b(i - m\Delta M)\exp(j2\pi n\Delta N i) \qquad 3\text{-}2$$

where ΔM is the discrete time shift of the analysis or synthesis function and ΔN = 1/N, is the interval between adjacent frequency bands in the JTF spectrum. s(i) is an input sequence of discrete values, b(i) is the discrete synthesis (or basis function) and γ(i) is the discrete analysis function. The relationship between b(i) and γ(i) is given by the discrete biorthogonality condition whichWexler and Raz [6] defined as :

$$\Delta M\Delta N \sum_{i=-\infty}^{\infty} b_{m,n}(i)\gamma_{k,l}^*(i) = \delta(m-k)\delta(n-l) \qquad 3\text{-}3$$

where :

$$\gamma^*_{k,l}(i) = \gamma^*(i - k\Delta M)\exp(-j2\pi l\Delta N i) \qquad 3\text{-}3a$$

$$b_{m,n}(i) = b(i - m\Delta M)\exp(j2\pi n\Delta N i) \qquad 3\text{-}3b$$

and

$0 < k < \infty,\ 0 \le l \le N\text{-}1,\ 0 < m < \infty,\ 0 \le n \le N\text{-}1..$

(3-1) and (3-2) hold when b(i) and γ(i) are chosen so that (3-3) is satisfied. The process for the selection of a γ(i) that satisfies (3-3) and which is also suitable for JTFA is described in section 3.2. Ideally, the analysis function should resemble a gaussian basis in shape (within a certain scaling factor). This can be achieved only for certain conditions which are described in detail in section 3.2.

Wexlar and Raz recognized that (3-1), (3-2) and (3-3) would be impossible to implement as a result of the infinite bounds on the i summation in the analysis and biorthogonality equations (3-1 and 3-3) and the m summation in the synthesis equation (3-2) [6]. The equations could only be implemented for finite or periodically extended functions in time. The following section presents their approach.

3.2 Finite Length Sequences

Wexler and Raz [6] proposed a digital solution to the problem of finding an appropriate discrete analysis function γ(i) . Their approach was to implement a discrete form of the biorthogonality condition which resulted in an underdetermined set of linear equations. As a result, the system could have an infinite number of possible solutions [6]. The results were based on finite length sequence or periodic sequences only and is described below. The underdetermined system was achieved by oversampling. In this context, oversampling refers to having more sample points in the JTF spectrum than the actual input signal length, L. The resulting γ(i) closely approximated b(i) provided that the time shift, ΔM, the sampling interval in frequency ΔN and the variance of the basis function were all carefully chosen.

3.2.1 Analysis and Synthesis Formulae

Consider a finite length sequence s(i) of duration L (number of samples in sequence). For such a sequence, the Gabor Transform pair are as described by (3-1) and (3-2) with the upper limit on the i summation set to L and the limit on the m summation set to $M = L/\Delta M$. This is given by:

$$C_{m,n} = \sum_{i=0}^{L-1} s(i)\gamma^*(i - m\Delta M)\exp(-j2\pi n\Delta N i / L) \qquad 3\text{-}4$$

$$s(i) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} C_{m,n} b(i - m\Delta M)\exp(j2\pi n\Delta N i / L) \qquad 3\text{-}5$$

M is the number of time shifts of the analysis function $\gamma(i)$ or synthesis function b(i) over the time L, and N is the number of sampling points around the unit circle in the frequency domain. ΔM is the increment of the shift of the analysis function and $2\pi\Delta N/L$ is the interval between sampling points in the frequency domain.

The following constraints hold for the parameters used in (3-4) and (3-5).

$$\mathbf{M\Delta M = L} \qquad 3\text{-}6,$$

$$\mathbf{N\Delta N = L} \qquad 3\text{-}7,$$

$$\mathbf{MN \geq L} \qquad 3\text{-}8.$$

The condition MN = L may lead to unstable calculation of $\gamma(i)$ and generally MN > L is preferred for stable calculation of $\gamma(i)$ [7]. This is the case of oversampling where there are more Gabor Transform coefficients, MN, than the total number of samples, L, in the input sequence. Alternatively, by substituting (3-6) in (3-8), it is shown that oversampling and therefore stable calculation of $\gamma(i)$, is achieved if $N > \Delta M$. In words, more frequency domain points calculated for each time shift than the actual length of the time shift. The oversampling rate is given by:

$$\xi = N/\Delta M \qquad 3\text{-}8a$$

Oversampling leads to a localized $\gamma(i)$ (analysis function concentrated in time) provided that the synthesis function b(i) is also localized in time. Critical sampling, where MN = L, may cause unstable calculation of $\gamma(i)$. Using critical sampling, the resulting analysis function may have its energyspread across the length of the signal, L . This means that the $C_{m,n}$ no longer reflect the local frequency content of the signal. In this discussion, $\gamma(i)$ and b(i) are periodic sequences or finite length sequences of length L and satisfy the discrete biorthogonality given by (3-3). By setting k = l = 0 and truncating the infinite bound to L, (3-3) is simplified to:

$$\Delta M \Delta N \sum_{i=0}^{L-1} b_{-m,-n}(i)\gamma^*(i) = \delta(m)\delta(n) \qquad 3\text{-}9$$

where:

$0 \le m \le M\text{-}1$, and $0 \le n \le N\text{-}1$

3.2.2 Basis Function b(i)

The gaussian function is the basis, b(i) used in the OLGT and it is used throughout this thesis as the basis function for other Gabor Transform techniques. Since the gaussian function has a minimum time - bandwidth product [4], it will be shown to be ideal for filterbank implementations where it is desired that the corresponding analysis function (ideally gaussian in shape) be localized in time (for windowing) and effectively bandlimited in frequency (for generating accurate frequency outputs). More information on these aspects is presented in Chapter 5. Qian and Chen [7] used a gaussian function that was normalized and whose variance was optimized for the parameters of L, ΔM, ΔN. The equation for b(i) is given in (3-10):

$$b(i) = (\pi\sigma^2)^{-0.25}\exp[-(i-0.5(L-1))^2/2\sigma^2] \qquad 3\text{-}10$$

where

$$\sigma^2 = (L\Delta M)/(2\pi\Delta N) \qquad 3\text{-}10a$$

Substituting 3-8a in 3-10 , the expression for the basis function is given by:

$$b(i) = [0.5\xi(\Delta M)^2]^{-0.25}\exp[-\pi(i-0.5(L-1))^2/\xi\Delta M^2] \qquad 3\text{-}10b$$

3.2.3 Implementation of the Discrete Biorthogonality Condition and Calculation of $\gamma(i)$

The discrete biorthogonality equation [6] was given by :

$$\Delta M\Delta N\sum_{i=0}^{L-1} b_{-m,-n}(i)\gamma^*(i) = \delta(m)\delta(n) \qquad 3\text{-}9$$

where:

$0 \le m \le M-1,\ 0 \le n \le N-1$

When MN > L (3-8), it is possible to use (3-9) to achieve a $\gamma(i)$ that closely resembles b(i) (which is gaussian) and is therefore localized in time. It must be noted that the resulting $\gamma(i)$ is only gaussian-like for these paramenters but is nevertheless useful for JTFA. For the above the above conditions, (3-9) is an underdetermined system (more unknowns than equations) and can be written in the following matrix format [6]:

$$\mathbf{H_{pxL}\gamma_{Lx1} = \mu_{px1}} \qquad 3\text{-}11$$

where μ_{px1} is a px1 vector whose elements are [$\Delta M\Delta N/L$, 0, 0, ...], γ_{Lx1} is an unknown vector and $\mathbf{H_{pxL}}$ is a complex matrix with elements given by:

$$\mathbf{b_{k,i} = b(i + mN)e^{-j2\pi nMi/L}} \qquad 3\text{-}12$$

where:

$k = n\Delta N + m$, $0 \le m \le \Delta N-1$, and $0 \le n \le \Delta M - 1$

The problem at this point is to determine γ_{Lx1} thereby yielding the values for $\gamma(i)$ to be used in the analysis equation. Since the system (3-9) is underdetermined when MN > L, there are a number of solutions that are suitable for JTFA. Since it is required that the analysis function $\gamma(i)$, be localized in time, Wexler and Raz [6] suggested choosing $\gamma(i)$ which was most similar to b(i) in the sense of least-squares error. This was given by:

$$\mathbf{E = \min \| \gamma(i)/\|\gamma(i)\| - b(i) \|^2} \qquad 3\text{-}13$$

and yields an analysis function that is gaussian-like. Since $\mathbf{H_{pxL}}$ in (3-11) is not a square matrix a solution can be obtained by taking the pseudo inverse [17] of (3-11) which is given by :

$$\gamma_{Lx1} = (H^T H)^{-1} H^T \mu_{px1} \qquad 3\text{-}14$$

The result of (3-14) is a $\gamma(i)$ that closely resembles b(i) in shape (within a scaling factor that is given by $\Delta M \Delta N/L$). It is therefore localized in time and frequency as it will generally be of gaussian shape. In general the error, E, decreases as the oversampling ratio ξ increases and at the same ξ the E will often be smaller if ΔM is proportional to the time duration of the $\gamma(i)$ and if ΔN is proportional to the bandwidth of $\gamma(i)$ [17]. An example of a gaussian basis function, b(i), with the corresponding analysis functions, $\gamma(i)$, for $\xi = 2$ and $\xi = 3$ are shown in figure 3.1.

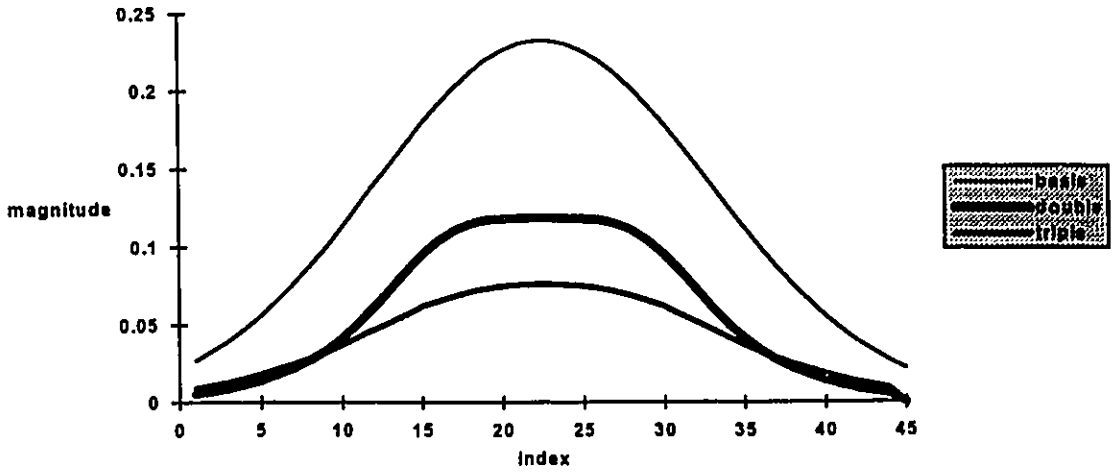

Figure 3-1: Basis function with corresponding analysis functions for double and triple oversampling

3.3 Long and Infinite Length Sequences

The main objective of the implementation of the Gabor Transform for finite length or periodic sequences using the Wexler and Raz approach was to determine an optimum $\gamma(i)$. However as the length of the sequence, L becomes increasingly long, the size of the $\mathbf{H_{pxL}}$ matrix grows proportionally larger. As a result, taking the pseudo inverse of (3-11) to determine γ_{Lx1} becomes computationally burdensome and makes the finite implementation process impractical. Qian and Chen [7] proposed an approach that makes implementation for long or even infinite length sequences practical. This approach is described below.

Consider two finite length or periodic sequences s'(i) and b'(i) that are constructed from two finite length sequences s(i) and b(i) respectively:

$$s'(i) = 0 \quad 0 \le i < L_W$$
$$s'(i) = s(i) \quad L_W \le i < L$$

where s(i) is a non zero sequence of length L and,

$$b'(i) = b(i) \quad 0 \le i < L_W$$
$$b'(i) = 0 \quad L_W \le i < L$$

where b(i) is given by (3-10). If the above sequences are used in (3-4) and (3-5) and noting that the new sequences are zero for several values, we get:

$$C_{m,n} = \sum_{i=0}^{L-1} s'(i)\gamma'^{*}(i - m\Delta M)\exp(-j2\pi n\Delta N i / L) \qquad 3\text{-}15$$

$$s'(i) = \sum_{m=0}^{(L/\Delta M)-1} \sum_{n=0}^{N-1} C_{m,n} b'(i - m\Delta M)\exp(j2\pi n\Delta N i / L) \qquad 3\text{-}16$$

Setting $\Delta N = L/N$ and substituting, in the above, (3-15) and (3-15) simplify to:

$$C_{m,n} = \sum_{i=0}^{L-1} s'(i)\gamma'^{*}(i - m\Delta M)\exp(-j2\pi ni / N) \qquad 3\text{-}17$$

$$s'(i) = \sum_{m=0}^{(L/\Delta M)-1} \sum_{n=0}^{N-1} C_{m,n} b'(i - m\Delta M)\exp(j2\pi ni / N) \qquad 3\text{-}18$$

As described in section 3.1, the condition for stable reconstruction is that MN > L. Substituting ΔN= L/N, the condition for stable reconstruction becomes N > ΔM. This basically means that the number of frequency samples around the unit circle must be greater than the amount time shift used for the analysis, γ'(i), and synthesis functions, b'(i).

Qian and Chen [17] showed for the above, the analysis function is given by:

$$\gamma'(i) = \gamma(i) \quad 0 \le i < L_W$$
$$\gamma'(i) = 0 \quad L_W \le i \le L$$

where γ(i) is computed by the following process:

(i) determine the time shift ΔM desired for analysis;
(ii) using ΔM, calculate M = $L_W/\Delta M$ (this is only used in calculating γ(i));
(iii) choose the oversampling rate, ξ, desired for analysis;
(iv) using ξ and ΔM, calculate the number of frequency points N = ξΔM;
(v) using N, calculate ΔN = L_W/N (this is only used in calculating γ(i));
(vi) calculate γ(i) using $\gamma_{Lx1} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mu_{px1}$ (3-14); and
(vii) set ΔN = L_W/N and M = $L_W/\Delta M$ (these values will be used in the analysis process).

Due to the zero padding of γ'(i), b'(i) and s'(i), the Gabor Coefficients $C_{m,n}$ can be completely determined from (3-16) without γ'(i) shifting over to the next period of L values of the periodic sequence s'(i). This releases the periodic constraint. Then increasing the value of L such that L ----> ∞, but at the same time leaving L_W fixed, Qian and Chen arrived at the Gabor Transform pair for infinite length sequences. The equations are as given below:

$$C_{m,n} = \sum_{i=-(L_w-\Delta M)}^{\infty} s(i)\gamma^*(i-m\Delta M)\exp(-j2\pi ni/N) \qquad 3\text{-}19$$

$$s(i) = \sum_{m=-(L_w/\Delta M)+1}^{\infty} \sum_{n=0}^{N-1} C_{m,n} b(i-m\Delta M)\exp(j2\pi ni/N) \qquad 3\text{-}20$$

For a long sequence, the transform pair is as described in (3-19) and (3-20) with the upper limit on the i summation being L and the upper limit on the m summation being $M = L/\Delta M$. A long sequence is defined as one where $L \gg L_W$. The equations for long sequences are given by:

$$C_{m,n} = \sum_{i=-(L_w-\Delta M)}^{L} s(i)\gamma^*(i-m\Delta M)\exp(-j2\pi ni/N) \qquad 3\text{-}21$$

$$s(i) = \sum_{m=1-L_w/\Delta M}^{M} \sum_{n=0}^{N-1} C_{m,n} b(i-m\Delta M)\exp(j2\pi ni/N) \qquad 3\text{-}22$$

Note that in (3-21) the first few values of m are actually negative. This results in the initial samples of the input signal, s(i), being adequately weighed. This ensures that no information about the original sequence is lost. If the first shift of the analysis function was to have started at 0 instead at $i = -L_W + \Delta M$, then the first few samples would have been essentially weighted out due to the taper towards the limits of the analysis function. The result of this would be an inaccurate JTF representation of the signal and large errors when the synthesis equation (3-22) is applied to reconstruct the original signal.

3.4 Chapter Summary

The OLGT was introduced in this chapter. It was shown to be useful for JTFA of finite and infinite length discrete time signals because the analysis function $\gamma(i)$ is localized in time and therefore the JTF spectrum reflects the local frequency content of the signal s(i) during windows of successive time .

The approach for determining Gabor Transform coefficients for finite length sequences [6] was summarized. The significant result was the calculation of a discrete analysis function $\gamma(i)$ through an oversampled implementation of the discrete biorthogonality condition. The resulting $\gamma(i)$ was chosen to resemble the basis b(i) and was gaussian like in shape for oversampled sequences.

The OLGT, developed by Qian and Chen using the theories presented in [6] was summarized. They were able to extend the Wexler and Raz approach for determining the Gabor Transform for finite length sequences to infinite length sequences while maintaining a $\gamma(i)$ that was non zero for a length of L_W only. This ensured that $\gamma(i)$ was still short compared to s(i) and was therefore suitable for JTFA. Two sets of equations were presented, the first (3-19) and (3-20), was for infinite length input signals and the second (3-21) and (3-22) was for long input signals.

A drawback of the OLGT is that the processing time that is required to generate the Gabor Transform coefficients, $C_{m,n}$, will be slow because of length of ' i ' summation in the analysis equation (3-21) is over the entire length of the input signal for every, $C_{m,n}$. In addition to this the OLGT is not suitable for real time processing, because the entire input signal s(i) must first be received and then stored prior to any JTFA occurring. This makes it unsuitable for an application such as processing a SAR signal in real time. This requirement also inhibits the OLGT from being implemented in a filterbank structure that processes a continuous stream of radar samples in real time. Chapter 4 presents a solution to these problems in the form of the Window Based Gabor Transform (WBGT).

Chapter 4: A Window Based Gabor Transform

A Window Based Gabor Transform (WBGT) is presented in this chapter. The WBGT is an analysis-synthesis transform pair that can be used for processing non-stationary non-periodic infinite length sequences. It is therefore suitable for frequency transforming successive windows of range gate data for cross range matched filtering in SAR processing. The WBGT is formulated to be suitable for filterbank implementation (this will be described in chapter 5). The WBGT is developed in this chapter from the OLGT presented in chapter 3. The effects of truncating the analysis function, $\gamma(i)$, and basis function, b(i), are studied to see how the processing speed can be reduced. Truncating the $\gamma(i)$ for analysis may result in some loss of signal information which would increase the error in reconstructing the original signal from the Gabor Transform Coefficients. This phenomenon is studied in this chapter. The WBGT using truncated windows is referred to as the Truncated WBGT (TWBGT). This chapter includes digital simulations which compare the performance of the STFT, the OLGT, the WBGT and the TWBGT. The algorithms were implemented using Professional Matlab for Windows on a 486 PC with a 66 MHz clock. The results are compared for processing time required to generate the Gabor Transform coefficients, $C_{m,n}$ and for the accuracy of the reconstruction of the original signal s(i) from the $C_{m,n}$. The simulations show that the WBGT reduces processing speed in comparison to the OLGT, but still maintains the same signal information in the JTF spectrum as demonstrated by the low reconstruction error. It is also shown that employing the TWBGT decreases processing speed in proportion to the truncated window length and that in some cases, the error in reconstructing the original signal form JTF data is actually reduced by employing the TWBGT.

The WBGT is based on the OLGT transform pair described by (3-21) and (3-22) but can be used to process an infinite length signal as data samples are received as opposed to the OLGT, which required the entire signal s(i) to be received before any processing could begin. The approach is closer to that used in the STFT (which was presented in chap 2) and therefore makes it more suitable for SAR applications.

In section 4.1 the motivation for the WBGT is described. The analysis and synthesis equations are presented in section 4.2. Section 4.3 contains a description of the TWBGT

and section 4.4 contains an analysis of the performance of the STFT, OLGT, the WBGT and the TWBGT. A summary of the significant results of the chapter is given in section 4.5.

4.1 Motivation for a Window Based Transform

In Chapter 3, the OLGT for both finite and long sequences was presented. The equations for analysis and synthesis of long sequences were given by :

$$C_{m,n} = \sum_{i=-(Lw-\Delta M)}^{L} s(i)\gamma^*(i - m\Delta M)\exp(-j2\pi ni / N) \qquad 3\text{-}21$$

$$s(i) = \sum_{m=1-Lw/\Delta M}^{M} \sum_{n=0}^{N-1} C_{m,n} b(i - m\Delta M)\exp(j2\pi ni / N) \qquad 3\text{-}22$$

Unlike a STFT, whose coefficients, $X_{m,n}$, are computed using the value and index of input signal samples within a finite window of length L_W as described by (2-22), the OLGT coefficients $C_{m,n}$ are computed using the values of the input signal samples, s(i), as well as the index, i, within the entire input signal as described by (3-21). This is because the analysis function, $\gamma(i)$ is not necessarily concentrated within a finite interval of length L_W may be non zero through a range of length L. The computation of each coefficient therefore requires a summation over the entire length of the input signal, s(i). As a result, the corresponding exponential factor, given by $\exp(-j2\pi ni/N)$ contains an i index. It is desirable to calculate $C_{m,n}$ from values of successive samples of the input signal, s(i), that are within the m^{th} window only for a WBGT.

From the above, it is seen that two drawbacks of using the OLGT, are that the calculation of each Gabor Transform Coefficient $C_{m,n}$ requires:

(i) the time index, i, of each sample relative to the origin (i.e. the OLGT cannot be used for most applications unless the origin of the sequence is known and therefore the entire sequence must be received prior to processing); and

(ii) a summation over the length of the entire sequence (all i). This too requires the reception of the entire sequence prior to processing (which precludes real time applications) and increases analysis times especially for very long sequences.

These requirements can make actual implementation impractical in applications such as filterbanks. A filterbank used in SAR systems (especially one implemented in hardware) processes a stream of radar return signal samples as they are received in real time (see chapter 5 for more details). The summation over the entire sequence lengthens the analysis process considerably and makes the processing of an infinite or even a very long sequences of SAR data nearly impossible. Since all samples of s(i) are required for the calculation of a given $C_{m,n}$, the entire signal must be received and stored before the $C_{m,n}$ can be generated. For SAR processing this would preclude processing as pulses are being transmitted and therefore real time image updates would not be achievable using the OLGT.

For general applications $\gamma(i)$ may have any shape and not necessarily be localized in time. However, for JTFA, it is required that $\gamma(i)$ be localized in time (of finite impulse response) [7]. As a result, the summation over the length of the entire input for each $C_{m,n}$ becomes unnecessary since $\gamma(i)$ will be zero outside a particular range (provided that b(i) is also localized in time) [7] as described in chapter 3. Summing over a non-zero range of $\gamma(i)$ (which is L_W or less) only will therefore decrease processing time considerably and permit the real time processing of $C_{m,n}$. The analysis function, $\gamma(i)$, can now be considered as a windowing function that weighs samples in a finite region where it is non zero and suppresses samples outside that region [14].

4.2 Equations for Analysis and Synthesis

The range over which $\gamma(i)$ is non-zero is L_W (which is also the length of b(i)) if (3-10) and (3-14) are used to calculate b(i) and $\gamma(i)$ respectively. This is the maximum range over

which the input signal should be weighed and summed for successive shifts, m, in time as opposed to the general summation over the length of the entire sequence as described by (3-21). The calculation of $C_{m,n}$ based on data in the m^{th} window is inhibited only by the exponential factor in (3-21) that is dependent on the i index. If an exponential such as that used in the STFT, is employed in determining a WBGT, then $C_{m,n}$ will be determined based on information presented in the m^{th} window only. The exponential factor would consist of N equally spaced frequency bands, with the time indices for each calculated from the beginning of the current window, i.e. $\exp(-j2\pi nk/N)$, where k is the position within the window and $0 \leq k \leq L_W$.

The calculation of the coefficients, $C_{m,n}$, obtained using the WBGT is done by first shifting to a point that is m time shifts (of duration ΔM) from the origin (the actual value of m is unimportant). This is given by $m\Delta M$. The next L_W samples are weighed by $\gamma^*(k)$ multiplied by an exponential factor that is determined by the position k of the sample in the window and the frequency, n and then summed to give the Gabor Transform coefficients $C_{m,n}$. The analysis equation for the WBGT is:

$$C_{m,n} = \sum_{k=0}^{Lw-1} s(m\Delta M + k)\gamma^*(k)\exp(-j2\pi nk/N) \qquad 4\text{-}1$$

In (4-1), the value $m\Delta M + k$ is equal to the time index, i, used in (3-21). Instead of shifting the analysis function $\gamma(i)$ across the length of the input signal, s(i), as in (3-21), the input signal is shifted along as $\gamma(k)$ remains fixed with respect to the origin. This facilitates the processing of the input signal in real time.

The synthesis equation is similar to (3-20) except that the exponential factor is set to the conjugate of that in (4-1) (exponential is dependent on the k index only). The basis function, b(k), is of length L_W as is the analysis function, $\gamma(k)$, and the values of ΔM, N, remain the same as for the OLGT. The synthesis equation is given by:

$$s(i = m\Delta M + k) = \sum_{m=1-Lw/\Delta M}^{\infty} \sum_{n=0}^{N-1} C_{m,n} b(k)\exp(j2\pi nk/N) \qquad 4\text{-}2$$

The analysis equation for the WBGT for long sequences is given by (4-1). The synthesis equation for long sequences is given by (4-3), which is the same as (4-2) but with the upper limit on the m summation given by $M = L/\Delta M$.

$$s(i = m\Delta M + k) = \sum_{m = 1 - Lw / \Delta M}^{M} \sum_{n = 0}^{N-1} C_{m,n} b(k) \exp(j2\pi nk / N) \qquad 4\text{-}3$$

In comparing the analysis equations for the OLGT (3-21) and the WBGT (4-1), it is seen that time required to generate the $C_{m,n}$ for the WBGT will be significantly lower than for the OLGT. For the generation of each $C_{m,n}$, the WBGT requires only $2L_W$ multiplications and L_W-1 additions while the OLGT requires 2L multiplications and L-1 additions. Provided that $L_W >> 1$, then the approximate analysis time for the WBGT is given by:

$$\mathbf{t_{WBGT} \approx (L_W/L)(t_{OLGT})} \qquad 4\text{-}4$$

where $\mathbf{t_{WGBT}}$ is the time required to generate the $C_{m,n}$ for the WBGT and $\mathbf{t_{OLGT}}$ is the corresponding time for the OLGT. The time savings in using the WBGT compared to using the OLGT is basically proportional to the ratio of the window length, L_W, divided by the length of the entire input sequence, L.

The constraints for choosing the values for the time shift, ΔM, the length L_W of $\gamma(k)$ and the oversampling rate, ξ are listed below.

(i) The most critical constraint is that $\Delta M \leq L_W$. This ensures that no signal information is lost. If the ΔM is larger than L_W, then some samples in the input sequence will be weighed out during the analysis process, thus resulting in a loss of information.

(ii) The upper limit on the value of N is the length of the window, L_W. When N is greater than L_W (the number of frequency domain points are greater than the number of time samples in a given window), the discrete biorthogonality condition yields a $\gamma(i)$ that contains zeros only. Such a $\gamma(i)$ is useless for JTFA because it

completely weighs out the input signal leaving no information for analysis! As a result, $N_{max} = L_W$.

(iii) The upper limit on the oversampling rate, ξ is $L_W/\Delta M$. In chapter 3, it was shown that when the discrete biorthogonality condition holds, an oversampling rate, ξ, of two or more was required to yield an analysis function $\gamma(i)$ that resembled the basis b(i). Using (3-8a), $\xi_{max} = N_{max}/\Delta M$. Since $N_{max} = L_W$, ξ is limited to $\xi \leq L_W/\Delta M$.

The performance of the WBGT is studied in section 4.4.2. The time savings predicted by (4-4) are confirmed and it is also shown that the reconstruction of the original signal from the coefficients $C_{m,n}$ is identical for both the OLGT and the WBGT, provided that the non zero region of the basis, and analysis functions is of equal length for both.

4.3 Reducing processing time by window truncation

The theory that Wexler and Raz [6] developed for determining the analysis function, $\gamma(i)$. yielded one that was concentrated in time and approximately gaussian in shape. Gabor noted that the gaussian function has most of its energy concentrated about a few standard deviations, σ, from the median value. This is the region in which the gaussian function has 97% of its energy concentrated [4] and is approximately 3σ on each side of the median. In chapter 3, it was shown that the variance of b(i) can be chosen (3-10a) to generate an optimal $\gamma(i)$ for a particular ΔM, N, and L_W. Generally, the shorter the time shift, ΔM, the narrower the basis, b(i), in time as described by (3-10a). Conversely the longer the shift, the wider the b(i) in time. In the WBGT analysis equation (4-1) the summation uses the entire window length of L_W. As a result of the above, it would appeal to one's intuition to reduce the length of the summation used in analysis to an interval over which most of the energy of the gaussian basis, b(k), and corresponding analysis function, $\gamma(k)$, is concentrated (which would result in a corresponding reduction in analysis time) . Using (4-4) it is evident that the shorter the length of the window used, the greater the improvement in processing time. Thus truncating some of the non zero range of the analysis function should result in a reduction in analysis time.

Truncation of the analysis function, will lead to some loss of information contained in the input signal s(i), as some of the weighting provided by the original analysis function $\gamma(k)$ is lost. The amount of truncation will depend on the quantity of information loss that is deemed to be acceptable for a particular application. In some applications reducing processing time is important while in other applications obtaining near perfect reconstruction of the original signal from $C_{m,n}$ may be more important. For such applications, truncation may have to be limited in favor of maintaining the accuracy of signal information in the JTF spectrum.

The truncating is performed from the edges of the analysis function $\gamma(k)$. The tip (initial values) and tail (trailing values) are removed as shown in figure 4.1. This process results in the removal of the weighting of input signal provided by tip and tail of $\gamma(i)$, so there is some loss of signal information.

When truncating it is important to note that the signal can only be truncated such that the truncated window length, L_{WT} is longer than the time shift, ΔM used in analysis. If this is not done, then some signal values will receive zero weighting and a large amount of signal information will be lost during each time shift of the analysis function.

The equations for infinite length sequences are given by (4-1) and (4-2) with the window length L_W replaced by L_{WT} such that $L_{WT} < L_W$.

$$C_{m,n} = \sum_{k=0}^{L_{WT}-1} s(m\Delta M + k)\gamma^*(k)\exp(-j2\pi nk / N) \qquad 4\text{-}5$$

$$s(i = m\Delta M + k) = \sum_{m=1-L_{WT}/\Delta M}^{\infty} \sum_{n=0}^{N-1} C_{m,n} b(k)\exp(j2\pi nk / N) \qquad 4\text{-}6$$

where:

$0 \le k < L_{WT}$ $0 \le n < N$ and $\Delta M \le L_{WT} \le L_W$

For long sequences, the transform pair is as described above, but with the upper limit on the m summation in (4-6) given once again by $M = L/\Delta M$.

$$s(i = m\Delta M + k) = \sum_{m = 1 - Lwr / \Delta M}^{M} \sum_{n = 0}^{N-1} C_{m,n} b(k) \exp(j2\pi nk / N) \qquad 4\text{-}7$$

An example of a basis function and corresponding analysis function is shown in figure 4-1 below and after truncation.

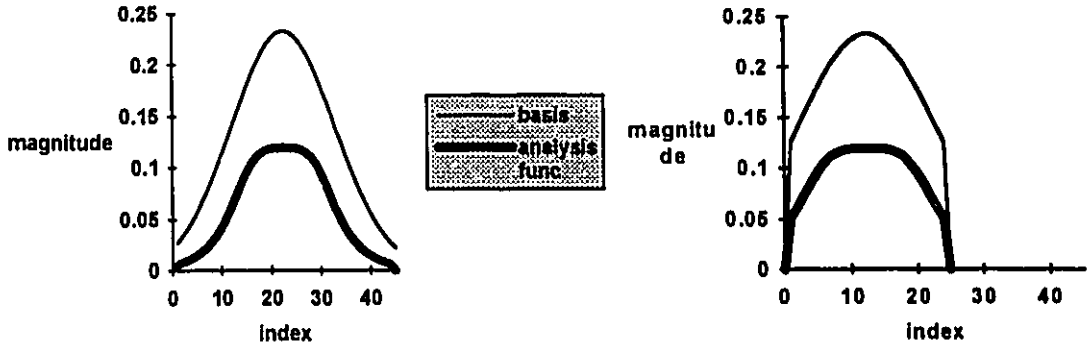

Figure 4-1 : Basis and Analysis Functions (a) is used in WBGT and (b) in TWBGT

Using the same approach as for (4-4), the analysis time for the TWBGT is given by:

$$t_{TWBGT} \approx (L_{WT}/L_W)(t_{WBGT}) \qquad 4\text{-}8$$

or

$$t_{TWBGT} \approx (L_{WT}/L)(t_{OLGT}) \qquad 4\text{-}9$$

The above equations hold only when the time shift, ΔM, and the oversampling rate, ξ, are identical for all three Gabor Transform techniques.

4.4 Performance Analysis of Gabor Transform Techniques

The STFT, OLGT, WBGT and the TWBGT are analysed in this section. A pulse wave form consisting of 512 samples with random noise is the input signal s(i) that is analysed by all three transform processes. The signal is illustrated in figure 4.2. A set of 6 simulations is performed using each Gabor Transform technique. Each simulation uses a different set of parameters to study the effects of varying oversampling rates ξ and time

shifts ΔM. The JTF spectrum is generated and then the original signal is reconstructed, r(i), (implementation of the corresponding synthesis equation) to determine how much of the original information is lost in each of the transform techniques. In appendix A, simulations are performed using a sine waveform with a delta function as well as a random noise waveform. The results are shown to be equivalent to those achieved in this section. The JTF spectrum for each signal is also shown in appendix A.

The time taken for analysis is measured in CPU time using the Matlab **etime** function (this function measures the CPU time that elapses between two time markers in a Matlab program). The analysis time measured in CPU time provides a relative measure of processing speed for comparison between the three different techniques. The analysis time for any given technique is in seconds. The number of multiplications and additions required for analysis is also shown for the OLGT and WBGT as a measure of the relative computation time.

Once r(i) is reconstructed, the error is calculated between s(i) and r(i). The average error, err_{av} is given by the following equation:

$$err_{av} = (1/L)\sum_{i=0}^{L-1} \left| \{s(i) - r(i)\} / s(i) \right| \qquad 4\text{-}10$$

An alternative way of describing the error for the reconstruction of the input signal of s(i) is by describing it in decibels. This is given by :

$$err_{db} = 20\log_{10}[1/L\sum_{i=0}^{L-1} \left| \{s(i) - r(i)\} / s(i) \right|] \qquad 4\text{-}10a$$

The error for reconstruction is not calculated for the STFT, because it has an analysis equation only and there is no synthesis equation (such as 4-3) that can be used to reconstruct the original signal and determine error (the original signal can however be reconstructed using inverse fourier transforms and weighing with the inverse of the analysis window w(k)). However, the accuracy of the JTF spectrum can be assessed qualitatively since its general shape begins to deteriorate once the time shift, ΔM, becomes large in comparison to L_W.

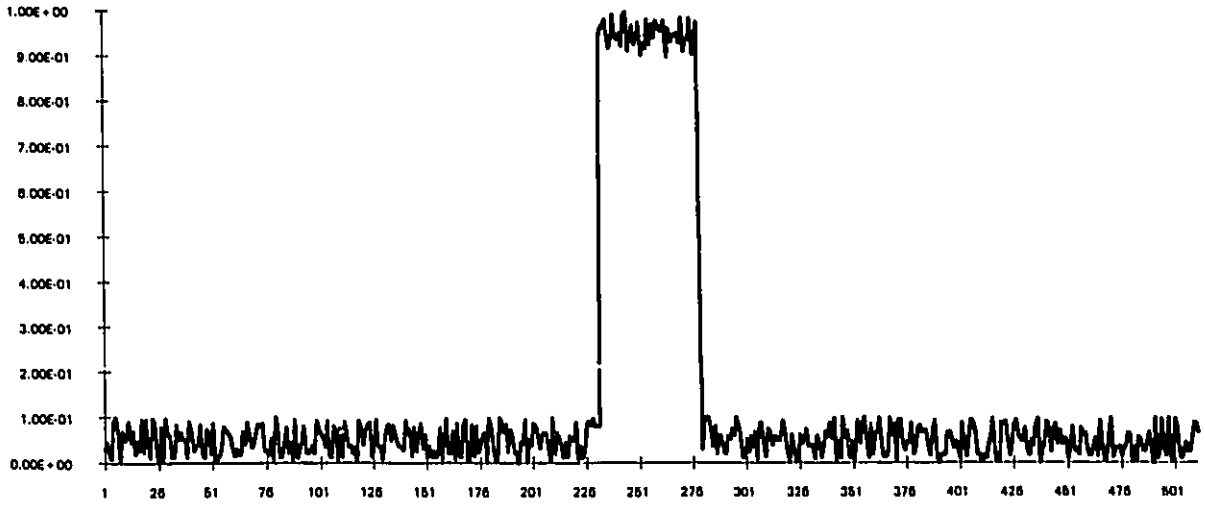

Figure 4-2: Signal s(i) analysed in this chapter. This is a pulse with noise

4.4.1 Performance Analysis of STFT

The Short Time Fourier Transform is analysed in this section, so that the results can be used for comparison with the OLGT, WBGT and the TWBGT in future sections.

The results obtained using the STFT are shown for different time shifts, ΔM, for a window length $L_W = 64$ and an input signal length $L = 512$. Two processes are employed, one using the DFT and the second using FFT's to perform the Fourier Transforming. The results for these two processes are shown in columns 8 and 9 of table 4-1.

L	Lw	ΔM	overlap 1-ΔM/ Lw	M	MxLw	t_{STFT} (DFT)	t_{STFT} (FFT)
512	64	8	.875	71	4544	228.88	6.10
		16	.75	35	2240	113.94	3.20
		24	.625	23	1472	74.92	1.94
		32	.5	17	1088	56.19	1.47
		40	.375	14	896	45.64	1.28
		48	.25	11	704	36.09	1.04
		56	.125	10	640	32.79	0.91
		64	0	8	512	27.24	0.81

Table 4-1: Results of digital simulations obtained using the STFT

From the results of table 4-1, it is seen that the analysis time, t_{STFT}, is directly proportional to the number of JTF coefficients, ML_W, generated by the STFT process. The analysis times obtained using the FFT instead of the DFT were significantly lower because of the processing time of each window of data is reduced by a ratio of $[\log_2 L_W]/L_w$ [15].

The analysis time, t_{STFT}, is related to the time shift, ΔM, that is used. When ΔM is small, the total number of shifts, M increases. For every time shift, the number of frequency domain coefficients is constant and is equal to the window length, L_W. As a result, the total number of JTF coefficients increases as ΔM decreases.

From the above, it might assume that it is most advantageous to use a large ΔM, thereby decreasing the number of JTF coefficients and reducing t_{STFT}. In practice, ΔM should be increased to a point at which the JTF spectrum accurately represents the frequency content of the signal s(i). When $\Delta M \approx L_W$ a large amount of the energy of s(i) is lost as it is weighed out on one time shift and never weighed in the previous or in the next time shift (because of the use of non rectangular windows). From the analysis of table 4-1, it was found that when $2\Delta M > L_W$, the JTF spectrum began to deteriorate and no longer had the general shape of the JTF spectrum for $2\Delta M \leq L_W$.

4.4.2 Performance Analysis of OLGT

The performance of the OLGT is analysed in this section. A set of six simulations, using the OLGT is contained in table 4-2 and serves as the baseline which is used for comparison with the WBGT and the TWBGT later in this chapter. The simulations were performed using (3-21) and (3-22).

Simu-lation	M	N	L_W	Δ M	overlap 1- ΔM/ Lw	ξ	M*N	t_{OLGT}	**mults**	**adds**	err_{av}	err_{db}
#1	71	16	64	8	.875	2	1136	382.76	1163.3k	581.6k	0.1102	-19.15
#2	35	32		16	.75	2	1120	373.33	1146.8k	573.4k	0.1347	-17.41
#3	17	64		32	.5	2	1088	364.16	1114.1k	557.1k	1.09E-14	-279.25
#4	71	24		8	.875	3	1706	578.25	1744.9k	872.5k	0.0259	-31.73
#5	35	48		16	.750	3	1680	559.63	1720.3k	860.5k	0.0534	-25.49
#6	35	64		16	.75	4	2240	756.71	2293.7k	1,146.8k	8.57E-15	-281.34

Table 4-2: Simulations Using the OLGT

From table 4-2, it is seen that the analysis time t_{OLGT}, was much larger than when using the STFT with FFT and even larger than the STFT with DFT (when ΔM was the same for the STFT and the OLGT). This was because of the larger number of multiplications and additions required to complete the analysis process. The calculation of each Gabor Transform coefficient requires 2L multiplications and L-1 additions. It can be seen that there is a linear relationship between the number of multiplications and additions and the t_{OLGT}. It is also seen that for simulations 3 to 6, the reconstruction error of the input signal s(i) from the $C_{m,n}$ is very low.

The analysis time, t_{OLGT}, was lowest for double oversampling, and increased almost linearly for triple and quadruple oversampling due to the larger number of coefficients generated at higher oversampling rates. For a given oversampling rate, ξ, t_{OLGT} was approximately the constant since the total number of JTF coefficients MN was approximately equal for all simulations at a given ξ. For all simulations, about 2.95 coefficients were generated per second. This is because the computation of each

coefficient by (3-21) requires the same number of operations (2L multiplications and L-1 additions).

In general the analysis times using the OLGT were far greater than when using the STFT (with DFT or FFT). However, the OLGT did have the advantage that it generated less JTF coefficients than the STFT as long as $\Delta M < L_W/2$. It should be noted that when $\Delta M > L_W/2$ the accuracy of the JTF spectrum using the STFT deteriorated and it didn't provide the JTF information about the original signal accurately. This is because large amounts of information from the original signal are weighed out on one time shift but not suitably weighed on successive shifts for the STFT. The number of JTF coefficients MN that the OLGT generates when $\Delta M = L_W/2$ is the minimum number of JTF coefficients that can be used to accurately represent the original signal.

The reconstruction errors for triple oversampling (simulations #4 and #5) were between 7 and 12 db less than that achieved when double oversampling was used (simulations #1 and #2). One should note that the analysis and basis functions used in each simulation are a unique pair and are described by (3-10) and (3-14) respectively. The lower errors occurred for triple oversampling because the analysis function, $\gamma(i)$, obtained using (3-14) resembles the basis, b(i) more closely for triple oversampling than for double oversampling and therefore results in a better reconstructed signal from the $C_{m,n}$. A reconstruction error that is effectively zero occurs when $N = L_W$ (in simulations #3 and #6). This is due to the shape of the $\gamma(i)$ which for this condition is almost identical (within a scaling factor) to b(i). While it may be argued that simulation #3 provides a very low error at a low analysis time, t_{OLGT}, and that this may be the ideal choice of parameters for processing, the actual choice may be system dependent. In some SAR processing applications it is desirable that a small shift ΔM be used to permit smoother updates between successive images. For such an application, the parameters of simulation #1 or #2 may be more applicable despite the error being higher than for simulation #3.

Conclusions from the analysis of the OLGT

The conclusions to be drawn from the above simulations are that:

(a) t_{OLGT} will be approximately constant for a given ξ;

(b) t_{OLGT} will increase linearly as ξ increases;

(c) t_{OLGT} is generally larger than t_{STFT};

(d) the reconstruction error generally improves with the oversampling rate except for those cases when N = L_W; and

(e) the choice of parameters is dependent on the application.

4.4.3 Performance Analysis for the WBGT

This section contains simulations that implement the WBGT pair for long sequences given by (4-1) and (4-3). The simulations performed for the OLGT are repeated for comparison in the first section and then the effects of varying the amount of time shift, ΔM is studied for double and triple oversampling.

4.4.3.1 Comparison of WBGT with OLGT

Simu-lation	M	N	L_W	ΔM	ξ	M*N	t_{OLGT}	t_{WBGT}	mults	adds	t_S	err_{av}	err_{db}
#1	71	16	64	8	2	1136	382.76	59.98	145.4k	72.7k	84.4%	0.1102	-19.15
#2	35	32		16	2	1120	373.33	58.22	143.4k	71.7k	84.4%	0.1347	-17.41
#3	17	64		32	2	1088	364.16	58.73	139.3k	69.6k	83.9%	1.09E-14	-279.25
#4	71	24		8	3	1706	578.25	91.79	218.2k	109.1k	84.2%	0.0259	-31.73
#5	35	48		16	3	1680	559.63	86.24	215,0 k	107.5k	84.6%	0.0534	-25.49
#6	35	64		16	4	2240	756.71	119.30	286.7k	143.4k	84.2%	8.57E-15	-281.34

Table 4-3: Comparison of WBGT simulations with OLGT

From table 4-3 it is seen that the WBGT reduces the analysis time compared to the OLGT by approximately 84% for all simulations while maintaining identical error for input signal reconstruction. As in the case of the OLGT, the analysis times for the WBGT, t_{WBGT}, are constant for a given oversampling rate, ξ, and increase as ξ increases. t_{WBGT} was seen to be linearly proportional to the number of multiplications additions used to analyse the input signal s(i). The analysis parameters chosen for each of the simulations of Table 4-3 were identical to those of Table 4-2.

Using (4-4) the theoretical time savings, t_S for L_W = 64 and L = 512, should be 87.5%. The actual time savings, t_S realized in the above simulations is defined by:

$$t_S = (1 - t_{WBGT}/t_{OLGT})100\% \qquad 4\text{-}11$$

In Table 4-3, it is seen that the actual time savings did not match the theoretically calculated savings. The actual time savings was approximately 84% which was 3 to 4 percent lower than the theoretically calculated amount. This was because there is some basic processing overhead that is involved in both the OLGT and the WBGT. The time taken to shift the analysis function as well as Matlab software input and output time required to access values contained in the arrays are examples of overhead that is required to perform the WBGT and the OLGT but that do not appear in the corresponding equations for these Gabor Transform techniques.

From (4-1) it is seen that the calculation of each Gabor Transform coefficient requires a constant $2L_W$ multiplications and L_W-1 additions. As a result, t_{WBGT} increases as the overall number of coefficients, MN increases. For double oversampling t_{WBGT} was in the range of 60 seconds while for triple oversampling this rose to about 90 seconds and then to 120 for quadruple oversampling. t_{WBGT} is dependent only on L_W , and the oversampling rate, ξ for a given signal of length L. In addition to the above, it should be noted that for the same number of JTF coefficients, t_{WBGT} and t_{STFT} were approximately equal when the DFT was used in the STFT.

The reconstruction errors are identical to those achieved using the OLGT and a reconstruction error that is effectively zero occurs when $N = L_W$ (in simulations #3 and #6). This is due to the shape of the $\gamma(i)$ which for this condition is almost identical (within a scaling factor) to b(i). It should also be noted that for these parameters, the analysis equation for the WBGT (4-1) and the STFT are identical provided that identical analysis windows w(k) and $\gamma(k)$ are used! Simulations #3 and #6 in Table 4-2 correspond to simulations b and d in Table 4-1 respectively. Therefore, for these parameters, the STFT (with FFT) should be used rather than the WBGT given that the STFT provides quicker analysis times than the WBGT. For the specific condition of $N = L_W$ and $w(k) \equiv \gamma(k)$, the STFT is identical to the WBGT, but if $N < L_W$ or $w(k) \neq \gamma(k)$ than the STFT and the WBGT are not equivalent. Therefore, the STFT can be considered a subset of the WBGT.

Conclusion from the comparison of the WBGT with the OLGT

The conclusion to be drawn from the above are that:

(a) t_{WBGT} will be approximately constant for a given ξ;
(b) t_{WBGT} will increases linearly as ξ increases;
(c) a large time savings over the OLGT is achieved by using the WBGT. This time savings is approximated by, $t_s \approx (1- L_W/L)x100\%$;
(d) the reconstruction error is identical for the WBGT and the OLGT when the same parameters are employed and the non zero region of the analysis functions used in both is L_W; and
(e) when $N = L_W$ and $w(k) \equiv \gamma(k)$, the STFT is identical to the WBGT.

4.4.3.2 Analysis of WBGT for Varying Amounts of Overlap

In this section the effects of varying the amount of overlap between successive windows is studied for the sequence of length L = 512, L_W = 64 at double and triple oversampling. The amount of overlap is defined as $1-\Delta M/L_W$. The first set of simulations is performed for double oversampling and is presented in table 4-4.

Simu-lation	M	N	L_W	ΔM	$L_W/\Delta M$	overlap $1-\Delta M/L_W$	M*N	t_{WBGT}	err_{av}	err_{db}	E
#a	143	8	64	4	16.00	0.9375	1144	58.39	0.0848	-21.43	0.0728
#b	71	16		8	8.00	0.875	1136	61.13	0.1102	-19.15	0.1044
#c	47	24		12	5.33	0.8125	1128	61.13	0.4620	-6.71	0.7823
#d	35	32		16	4.00	0.75	1120	60.53	0.1347	-17.41	0.0992
#e	28	40		20	3.20	0.6875	1120	61.08	7.26E+5	117.23	9.27E+5
#f	23	48		24	2.67	0.625	1104	58.22	0.7097	-2.97	0.0762
#g	17	64		32	2.00	0.5	1088	59.70	8.57E-15	-281.13	0.0646

Table 4-4: Analysis for WBGT as overlap is decreased for double oversampling

From table 4-4 it is seen that the error for reconstruction is proportional to error, E, in calculating $\gamma(i)$ which is given by (3-13). When the quotient $L_W/\Delta M$, is an integer, the implementation of the discrete biorthogonality condition (3-11) is more precise than when

$L_W/\Delta M$ is not (the quotient is rounded up to the next higher integer in the implementation process). This leads to a corresponding $\gamma(i)$ that has a lower error, E (3-13), and therefore a lower reconstruction error occurs for those simulations. In simulations c,e and f, $L_W/\Delta M$ is not an integer which leads to a large E and a corresponding large reconstruction error compared to the other simulations.

It is also seen that although the time shift, ΔM, increases in length throughout the sequence of simulations, the total analysis time is roughly of the same order of magnitude. This is because the number of frequency domain coefficients for each window is always $N = 2\Delta M$ and the total number of JTF coefficients MN are therefore of roughly the same order of magnitude.

In table 4-5 an overlap analysis is presented using triple oversampling.

Simu-lation	M	N	L_W	ΔM	$L_W/\Delta M$	overlap $1-\Delta M/L_W$	MN	t_{WBGT}	err_{av}	err_{db}	E
#h	143	12	64	4	16.00	.9375	1716	87.83	0.0203	-33.85	0.0152
#i	95	18		6	10.67	.90625	1710	92.44	0.0303	-30.37	0.0300
#j	71	24		8	8.00	.875	1704	88.65	0.0259	-31.01	0.0152
#k	57	30		10	6.40	.84375	1710	89.15	0.2610	-11.67	1.4795
#l	47	36		12	5.33	.8125	1692	92.34	0.4620	-6.71	0.7823
#m	41	42		14	4.57	.7813	1722	90.52	0.2504	-12.02	0.9536
#n	35	48		16	4.00	.75	1680	86.24	0.0534	-25.44	0.0131

Table 4-5: Analysis for WBGT as overlap rate is varied for triple oversampling

When the effects of varying the amount of overlap was analysed for the case of triple oversampling, it was once again found that the reconstruction errors were proportional to E which is effected by the ratio $L_W/\Delta M$. When $L_W/\Delta M$ was not an integer, there was a larger E, (in calculating the $\gamma(i)$) which lead to a large reconstruction error compared to the cases when $L_W/\Delta M$ was an integer. The case of triple oversampling however yielded smaller reconstruction error values than when double oversampling was used. This was because the $\gamma(i)$ resemble the b(i) more accurately at triple oversampling than at double oversampling which is reflected by the lower values for E. In addition to this, the original signal is represented by approximately 50% more coefficients when using triple

oversampling in the JTF spectrum and as a result the reconstruction is more accurate because more information is used for this process.

Conclusions from the analysis of the WBGT with varying amount of time shift

The conclusions to be drawn from the above sets of simulations are given below.

(a) t_{WBGT} is not effected by changing the amount of overlap since the number of JTF coefficients is approximately constant for a given ξ for a sequence of length L .

(b) The reconstruction error (err_{av} or err_{db}) is dependent on the error, E (given by 3-13) which is determined by the ratio $L_W/\Delta M$. When $L_W/\Delta M$ is an integer, err_{av} , err_{db} and E are significantly lower than when it is not. because for a non integer $L_W/\Delta M$ the shape of the analysis function is non gaussian.

(c) In general, the reconstruction error for triple oversampling are lower than for double oversampling for the same $L_W/\Delta M$ because of the closer approximation of the analysis function to a gaussian function at higher oversampling rates.

4.4.4 Performance Analysis for the TWBGT

In this section, the TWBGT is analysed. The transform pair described by (4-5) and (4-7) is implemented. A comparison is first made with the OLGT for windows truncated by 25% and 50%. This section also includes an analysis that measures the time taken for analysis and the corresponding error for reconstruction for the TWBGT where the window length is successively reduced. This analysis is performed for both double and for triple oversampling.

4.4.4.1 Comparison of TWBGT with WBGT and OLGT

Simu-lation	M	N	ξ	M*N	t_{TWBGT}	t_S WBGT	t_S OLGT	t_{WBGT}	t_{OLGT}	err_{av}	err_{db}
#2	34	32	2	1088	43.22	25.8%	88.4%	58.22	373.33	0.1522	-16.35
#5	34	48	3	1632	63.22	26.6%	88.7%	86.24	559.63	0.0023	-52.76
#6	34	64	4	2176	90.03	24.5%	88.1%	119.30	756.71	0.0080	-41.93

Table 4-6: **Analysis for TWBGT for Window Truncated by 25%, $L_W = 64$, $L_{WT} = 32$, $\Delta M=16$**

When the window length is truncated by 25%, it is seen that t_{TWBGT} is reduced by approximately 25% compared to that for the WBGT, confirming (4-8). The actual time savings compared to the WBGT is given by:

$$t_S = (1- t_{TWBGT} / t_{WBGT})x100\% \qquad 4\text{-}12$$

The reconstruction error, which one would expect to increase as the windows are truncated showed peculiar characteristics. In simulation #2 it remained almost constant while in simulation #5 it actually decreased. This behavior of reconstruction error as the amount of truncation increases is studied in greater detail later in this chapter. In simulation #6 the reconstruction error increased compared to the case of the WBGT because the $\gamma(i)$ closely resembled b(i). Any truncation led to errors in reconstruction.

Simu-lation	L	M	N	ξ	M*N	t_{TWBGT}	t_S WBGT	t_S OLGT	t_{WBGT}	t_{OLGT}	err_{av}	err_{db}
#2	512	33	32	2	1120	27.85	52.7%	92.5%	58.22	373.33	0.1522	-16.35
#5		33	48	3	1680	45.81	46.8%	91.8%	86.24	559.63	0.0023	-52.76
#6		33	64	4	2240	57.73	51.1%	92.4%	119.30	756.71	0.0080	-41.93

Table 4-7: **Analysis for TWBGT for Window Truncated by 50%, $L_W = 64$, $L_{WT} = 32$, $\Delta M=16$**

When the window is truncated by 50%, the analysis times are roughly 50% less than for the WBGT for the same set of parameters, confirming the theoretically predicted times using (4-8). The reconstruction error however drops for $\xi = 2$ and increases for $\xi = 3$ and $\xi = 4$. As the oversampling rate, ξ increases for any given set of parameters, the corresponding variance of the b(i) is larger (3-10a). This leads to a $\gamma(i)$ which is wider when (3-14) is implemented. It is therefore expected that a truncation of 50% will lead to greater reconstruction errors for the higher oversampling rates, because more of the energy of the b(i) and $\gamma(i)$ is concentrated near the extremities. This leads to more loss of information during the analysis process as well as a lower amount of energy for each shifted and modulated basis used in the synthesis. No conclusions can be drawn to explain the reasons why the reconstruction error drops for $\xi = 2$ from the analysis in this section, however, this is studied later in the chapter.

Conclusion from the comparison of the TWBGT with the WBGT and the OLGT

From the above analysis the following conclusions are made:

(a) the time saved by truncation is proportional to the amount of truncation and can be approximated by $t_s \approx (1 - L_{WT}/L_W)x100\%$;
(b) the reconstruction error can actually decrease after truncation for lower oversampling rates; and
(c) the higher the oversampling rate, the greater the error for increases in the amount of truncation.

4.4.4.2 Truncation Tradeoff Analysis

Tradeoff Analysis On Simulation #2, $\Delta M = 16$, $L_W = 64$, $\xi = 2$

The tradeoff analysis in this section consists of a study of the effects of truncation on the parameters used in simulation #2 that was used for both the OLGT and the WBGT. The original b(i) and $\gamma(i)$ are shown below before truncation is performed.

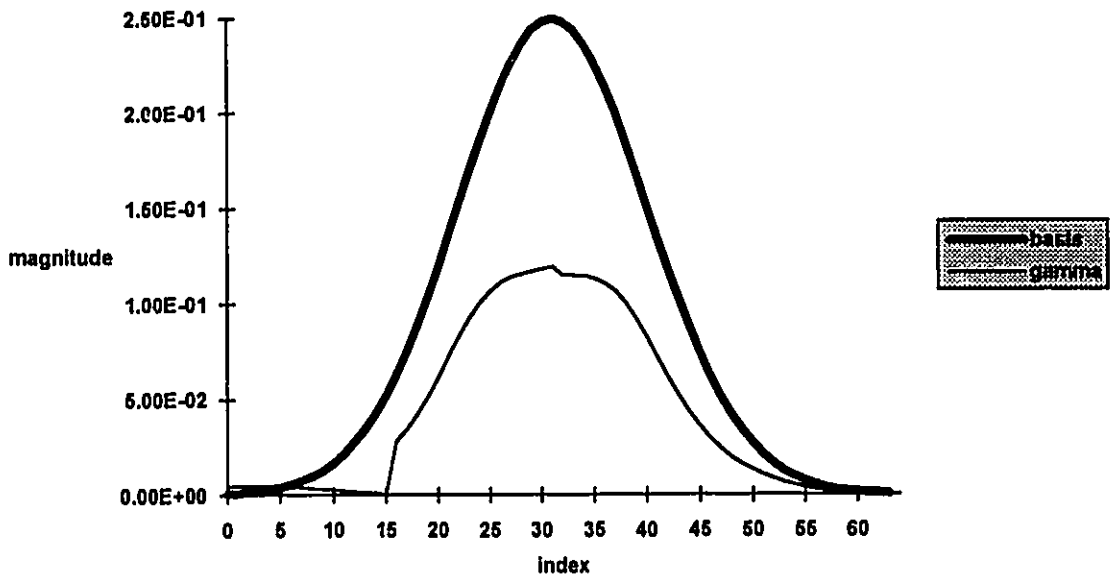

Figure 4-3: **Basis function of variance = 81.48 and analysis function used in TWBGT for $\Delta M = 16$, $\xi = 2$ and $L_W = 64$**

L_{WT}	overlap $1 - \Delta M / L_{WT}$	t_{TWBGT}	err_{av}	err_{db}
64	.75	58.22	0.1347	-17.41
60	.73	53.49	0.1410	-17.01
56	.714	48.77	0.1566	-16.10
52	.692	47.24	0.1587	-15.98
48	.667	43.22	0.1522	-16.35
44	.636	39.10	0.1281	-17.84
40	.600	36.08	0.0895	-20.96
36	.555	31.86	0.0524	-25.61
32	.500	27.85	0.0050	-46.08
28	.438	26.25	0.0199	-34.01

Table 4-8: Tradeoff Analysis On Simulation #2, $\Delta M = 16$, $L_W = 64$, $N = 32$, $\xi = 2$

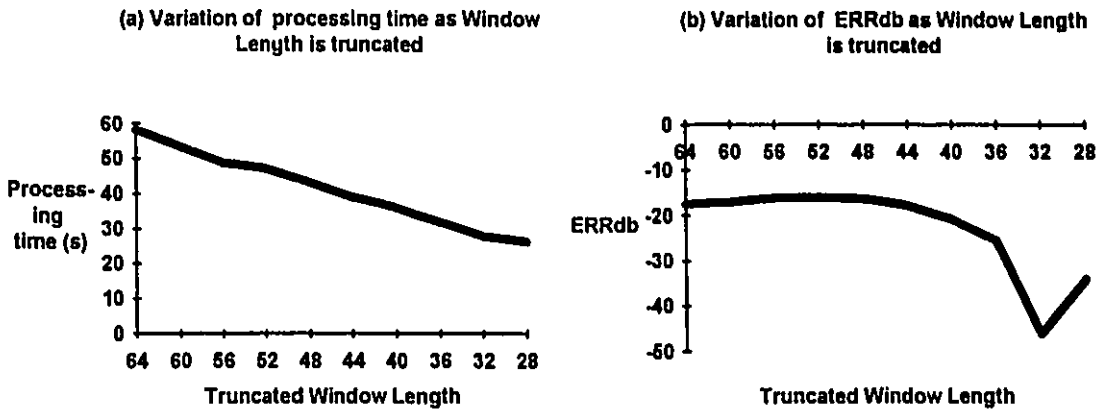

Figure 4-4: Processing Time and err_{db} for truncation analysis for simulation #2

In table 4-8 and figure 4-4, it is seen that the processing time decreases as the L_{WT} is reduced. This result is intuitively expected. However, the reconstruction error first remained approximately constant (until $L_{WT} = 48$) and then decreased with successive truncations (until $L_{WT} = 32$). This is not what one would expect. The logical hypothesis would be that window truncation would result in a loss of information and therefore the reconstruction error would increase. The variance of the basis for this simulation is 81 (standard deviation, $\sigma = 9$) and 97% of the energy of the basis and the corresponding $\gamma(i)$ is concentrated in the middle 36 elements of b(i) and $\gamma(i)$. From figure 4-4, it is seen that the $\gamma(i)$ does not exactly resemble b(i) especially outside the region of two standard deviations from the central sample. The ripple in $\gamma(i)$ towards the extremities, adds to the error and when truncated out actually results in lower reconstruction error. Successive truncation therefore decreases the reconstruction error for this simulation until significant amounts of the energy of $\gamma(i)$ is truncated after $L_{WT} = 32$.

Tradeoff Analysis on Simulation #3, $\Delta M = 32$, $L_W = 64$, $\xi = 2$

The tradeoff analysis in this section consists of a study of the effects of truncation on the parameters used in simulation #3 that was used for both the OLGT and the WBGT. The original b(i) and $\gamma(i)$ are shown below prior to truncation.

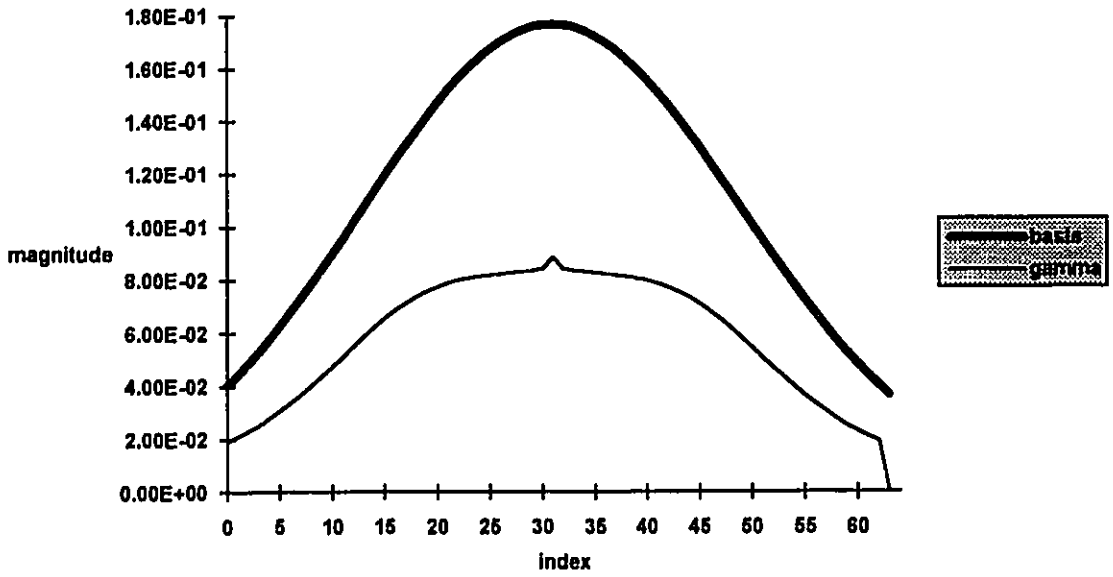

Figure 4-5: **Basis function of variance = 325 with analysis function used in TWBGT for $\Delta M = 32$, $\xi = 2$ and Lw = 64**

L_{WT}	overlap 1-ΔM/LWT	t_{TWBGT}	err_{av}	err_{db}
64	0.5	66.73	1.09E-14	-271.06
60	0.467	58.5	0.0005	-66.02
56	0.428	56.5	0.0141	-37.02
52	0.385	52.6	0.0271	-31.34
48	0.333	47.89	0.0454	-26.85
44	0.273	44.71	0.0708	-22.99
40	0.2	40.26	0.1050	-19.57
36	0.111	37.79	0.1497	-16.49
32	0	32.13	0.2060	-13.72

Table 4-9: **Tradeoff Analysis On Simulation #3, $\Delta M = 32$, $L_W = 64$, N = 64, ξ=2**

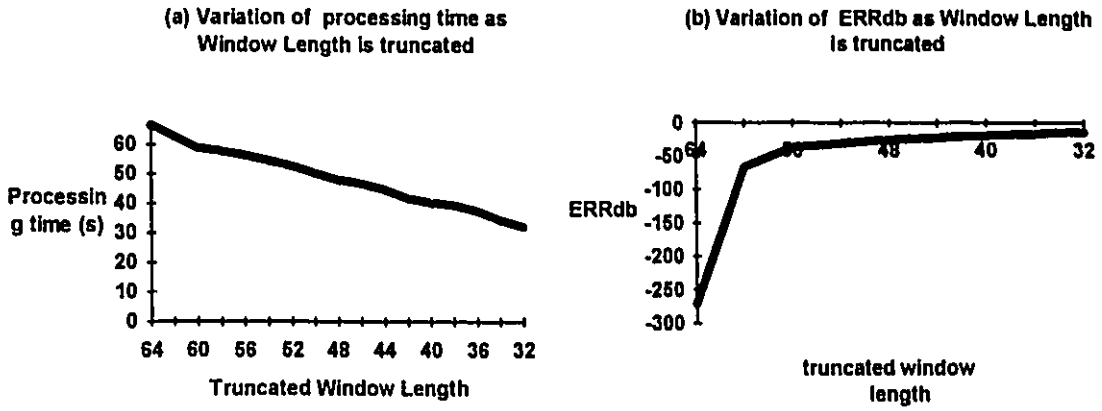

Figure 4-6: Analysis time and err_{db} for truncation analysis on simulation #3

The results in this section confirm the theory presented earlier in this chapter. Figure 4-6 confirms (4-8), with the analysis time dropping linearly as L_{WT} is successively reduced in length.

The reconstruction error increases as L_{WT} is reduced . Since the variance of the basis is 325 (see figure 4-5), the standard deviation, σ = 18. The region within two standard deviations of the central sample extends to 36 samples on each side. Since L_W = 64 any truncation will immediately effect the reconstruction error because a large amount of the energy of $\gamma(i)$ is removed with truncation. Since the analysis function $\gamma(i)$ has a very low E (very closely resembles the basis), each successive truncation adds to the reconstruction error).

Tradeoff Analysis on Simulation #5, $\Delta M = 16$, $L_W = 64$, $\xi = 3$

The tradeoff analysis in this section consists of a study of the effects of truncation on the parameters used in simulation #5 that was used for both the OLGT and the WBGT. The original b(i) and $\gamma(i)$ are shown below prior to truncation.

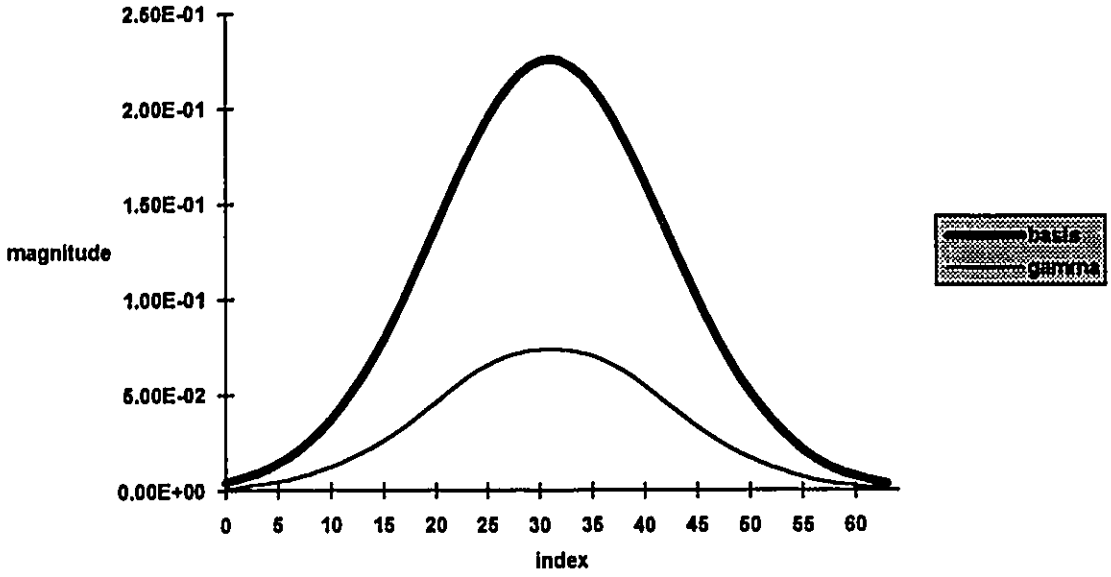

Figure 4-7: **Basis function of variance = 122 and analysis function for $\Delta M = 16$, $\xi = 3$ and Lw = 64**

L_{WT}	overlp $1-\Delta M/L_{WT}$	t_{TWBGT}	err_{av}	err_{db}
64	.75	86.24	0.0534	-25.45
60	.73	77.67	0.0374	-28.54
56	.714	74.26	0.0263	-31.60
52	.692	69.04	0.0153	-36.31
48	.667	63.22	0.0023	-52.76
44	.636	61.96	0.0050	-46.02
40	.6	57.01	0.0107	-39.41
36	.555	52.46	0.0214	-33.39
32	.5	45.81	0.0406	-27.82
28		39.27	0.0729	-22.74

Table 4-10: **Tradeoff Analysis On Simulation #5, $\Delta M = 16$, $L_W = 64$, N = 48, $\xi = 3$**

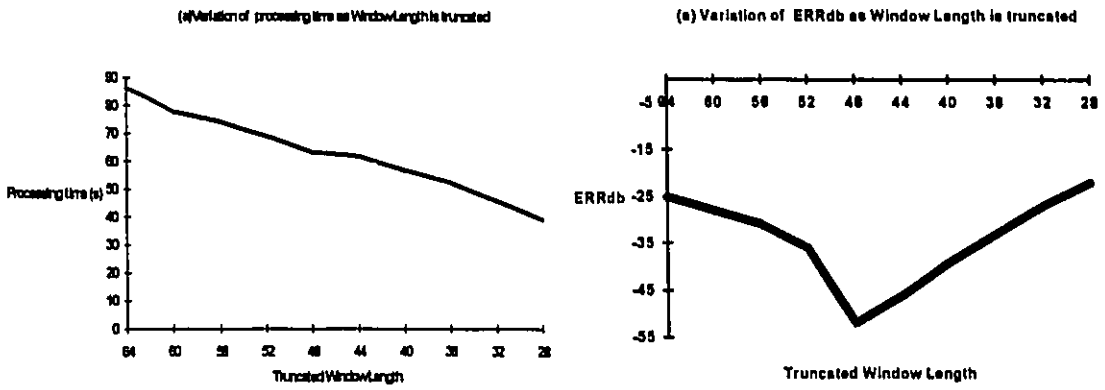

Figure 4-8: Processing Time and Err_{db} for truncation analysis for simulation #5

This series of simulations uses exactly the same set of parameters as used for the truncation analysis on simulation #2 except that $\xi = 3$. The analysis time drops approximately linearly as the window is truncated as in the case of the other simulations. It is seen that although the results exhibit an initial drop in reconstruction error as the window is truncated, the error gradually rises after $L_{WT} = 48$. Since the variance of the basis is 122, the standard deviation, $\sigma = 11.04$. This means that truncation beyond the value of 44 results in the removal of a significant amount of energy in the $\gamma(i)$. This results in inadequate weighting of the input signal s(i) and leads to high error in the reconstruction process.

Conclusion from Truncation Analysis

The following conclusions can be drawn from the truncation analysis:

(a) t_{TWBGT} decreases linearly as L_{WT} is reduced as predicted by (4-8);

(b) The time saved by truncation is proportional to the amount of truncation and can be approximated by $t_s \approx (1 - L_{WT}/L_W)x100\%$;

(c) reconstruction error drops compared to that using the WBGT as truncation increases. This holds for $L_{WT} > 4\sigma$. This is due to the error in $\gamma(i)$ near the its extremities;

(d) reconstruction error begins to increase when $L_{WT} < 4\sigma$ because a significant amount of energy in the $\gamma(i)$ and the b(i) are removed and the input signal s(i) is inadequately weighted; and

(e) when $L_W = N$, reconstruction error increases linearly as L_{WT} decreases.

4.5 Chapter Summary

In this chapter, the WBGT was developed based on the OLGT. It was shown to generated the JTF coefficients in much less time than the OLGT while maintaining identical error for signal reconstruction. The analysis times achieved using the WBGT was approximately L_W/L less than when using the OLGT. The WBGT generates far fewer JTF coefficients than the STFT for a signal of length L and for the same amount of time shift ΔM. However the analysis times for the STFT with FFT were much faster than the WBGT even though a much larger number of JTF coefficients were generally generated. The WBGT representation was shown to be suitable for real time processing, where it may be desirable to begin computing the JTF spectrum while the input signal, s(i), is still being received. It was shown that when the window length equals the number of frequency outputs and if identical windows are used, the STFT and WBGT will be identical.

The TWBGT was developed based on the WBGT and was found to reduce analysis time compared to the WBGT while maintaining low errors for signal reconstruction as long as the truncated window length was greater than or equal to four standard deviations of the gaussian basis, b(i).

From the simulations performed in this chapter it was found that for a sequence of a given length L, analysis function length L_W (or L_{WT} for the TWBGT) and oversampling rate ξ, the analysis time was constant for a given Gabor Transform technique (the OLGT, WBGT or the TWBGT). For a given ξ and L, the analysis time is not affected by changing the amount of time shift, ΔM, for both the OLGT and the WBGT. It was also found that when comparing the different Gabor Transform techniques when using the same analysis parameters, the time savings was directly proportional to the ratio of window lengths used as described by (4-4), (4-8) and (4-9). This was due to the lower number of multiplications and additions required to compute the Gabor Transform coefficients when using shorter windows.

The error for reconstruction of the original signal, s(i), was identical for the WBGT and the OLGT when the same simulation parameters were employed, as shown in tables 4-1 and 4-2. The reconstruction error generally decreased as ξ increased, because more JTFA

data is used to represent the signal in the JTF domain. The exception to this was for $N = L_W$. For this condition the reconstruction error was effectively zero even at double oversampling because of the close resemblance of $\gamma(k)$ to $b(k)$. The reconstruction error fluctuated significantly with changing ΔM when all other simulation parameters were held constant. and was found to be dependent on the ratio $L_W/\Delta M$ and when $L_W/\Delta M$ was an integer, it was significantly lower than when it was not.

For the TWBGT, while $L_{WT} > 4\sigma$, the reconstruction error remained constant or even improved as the analysis and synthesis functions were truncated, demonstrating that accurate signal information in the JTF spectrum was maintained. In those cases where reconstruction error actually improved, it was because truncation eliminated significant error in the $\gamma(k)$ towards the extremities. When all other simulation parameters were held constant, the reconstruction error increased at a greater rate as truncation increased for larger oversampling rates. This was due to the larger σ as given by (3-10a).

For both the WBGT and the TWBGT, the choice of parameters is dependent on the application. Ideally one would choose parameters that reduce the reconstruction error while minimizing the analysis time. The case of $N = L_W$ results in a reconstruction error that is effectively zero, but at the same time, it results in the maximum time shift between successive windows and may therefore make it unsuitable for some applications. The choices of time shift, ΔM, or ξ may be dictated by the system, therefore the engineer is forced to use those parameters that yield the most suitable analysis times and reconstruction errors for the particular application.

There are several advantage in using the WBGT when compared to the STFT. A lower number of coefficients $C_{m,n}$ can be used to represent the original signal than when using the STFT. This could especially be attractive if signal information has to be stored and then transmitted using JTF coefficients. At the receiving end, the original signal can then be constructed from fewer JTF coefficients and the transmission time and bandwidth of JTF coefficients is reduced. For the same values of ΔM, the processing time for the WBGT is lower than the STFT (when using a discrete Fourier Transform). This is due to the larger number of JTF coefficients that are generated for the STFT. The processing time for the STFT is lower than for the WBGT when $2\Delta M > L_W$ but this corresponds to a STFT JTF spectrum that no longer represents the original signal accurately. However, when an FFT was used for the STFT, the processing time was much smaller than for the

WBGT. For the specific condition when $N = L_W$ and $w(k) \equiv \gamma(k)$, the STFT is identical to the WBGT and therefore the STFT should be used because of the lower analysis times.

Chapter 5: A Filter Bank Implementation of the Gabor Transform For SAR

Two filterbank implementations of the WBGT are developed and presented in this chapter. One implementation uses a serial input while the second uses a parallel input. The parallel input version is then implemented using a matrix structure which effectively creates N band pass filters (BPF's) from a Low Pass Filter (LPF) by rotating it to N equally spaced frequency bands. The application of the parallel input filterbank implementation of the WBGT is then studied in the context of SAR range and cross range processing and the hardware requirements for real time processing are determined.

The theory and the results achieved using the WBGT and the TWBGT in chapter 4 provide the basis for the filterbank implementations presented in this chapter. It was shown that although the OLGT (described in Chapter 3) and the WBGT (Chapter 4) both lead to near perfect reconstruction of the original signal s(i) from the Gabor Transform coefficients, $C_{m,n}$, neither could match the speed of the STFT (when an FFT is used to perform the Fourier Transform) for analysis. The only advantage was that the JTF spectrum for both the OLGT and WBGT contained a far lower number of coefficients, $C_{m,n}$, for the same length signal, analysis window and time shift. This is advantageous if the JTF data has to be transmitted from an aircraft to a ground replay facility or if additional signal processing must be performed on the JTFA data. The WBGT and the TWBGT however were still slow, compared to the STFT (with FFT) as seen in chapter 4.

It will be shown that the implementation of the WBGT in filterbank configurations improves processing speed compared to using the analysis equation (4-i) only. This is because the content of each frequency band is generated simultaneously using parallel channels for a given window. It is also shown that the reconstruction error of the original signal, s(i), from the JTF spectrum are the same as those achieved when using the OLGT or the WBGT. The requirements for implementing the WBGT in a filterbank implementation are that:

(a) N frequency outputs be computed once for every ΔM input samples;

(b) each of the N frequency outputs be determined completely from the past L_W inputs, i.e. the impulse response of each filter is of length L_W;

(c) the time shift ΔM should be less than the window length L_W, so that there is overlap between adjacent windows;

(d) the spacing between the centre frequency of adjacent BPFs is $2\pi/N$ radians; and

(e) the number of BPF's, N, divided by the time shift ΔM is the oversampling rate ξ as described in chapter 3.

Section 5.1 and 5.2 describe how frequency translated Finite Impulse Response (FIR) LPF's are used to build a filterbank. In section 5.3 the equations for the implementation of these filters are presented. In section 5.4 a serial and a parallel filterbank implementation of the WBGT (these are also suitable for the TWBGT) are given. In addition to this, a matrix implementation of the parallel input filterbank is presented. In section 5.5, the performance of the filterbank implementations is analysed through digital simulations and compared with the results contained in tables 4-2 and 4-3. Finally in section 5.6 the filterbanks implementations are studied in the context of SAR range and cross range processing and hardware requirements for implementation are determined. Section 5.7 summarizes the results of the chapter.

5.1 Building a Filterbank Using Frequency Shifted Low Pass Filters

In order to implement the WBGT using a filterbank, N equally spaced BPF's are required. If the function g(k) has the frequency response of a LPF, then a time domain phase rotation will result in a frequency domain translation [18], resulting in a BPF, i.e,.

$$\text{if:} \quad \mathcal{F}\{g(k)\} = G(\omega) \qquad 5\text{-}1$$

$$\text{then:} \quad \mathcal{F}\{g(k)e^{j2\pi kn/N}\} = G(\omega - 2\pi n/N) \qquad 5\text{-}2$$

From the above, it is seen that if the impulse response of an LPF (which is symmetric about the zero frequency) is multiplied by $e^{-j2\pi kn/N}$, then the frequency response of that filter will be shifted to a frequency band centered at $2\pi n/N$ radians. By frequency shifting $G(\omega)$ to N different frequencies spaced equally by $2\pi/N$ radians, N equally spaced BPF's are generated. These N BPF's form a filterbank that computes the content of the N frequency bands represented by each filter.

The WBGT is implemented in a filterbank by taking the input signal, s(i), and then filtering each successive segment of the signal by N equally spaced BPF's in parallel. These filters yield N outputs for every sample of the input signal, s(i) [20]. However, for the WBGT, only N samples are required after every ΔM inputs (4-1). By decimating the N outputs following each input so that only one sample for each filter is conserved for every ΔM inputs, it is possible to implement the required time shift of ΔM. Such a filterbank configuration is shown in figure 5-1. In figure 5-1, $G_n(z)$ is the Z transform of the n^{th} BPF. The rate of decimation is ΔM. This is the time shift used in the WBGT and in figure 5-1 this is given by Dm.

Filterbank implementation of WBGT using a serial input technique

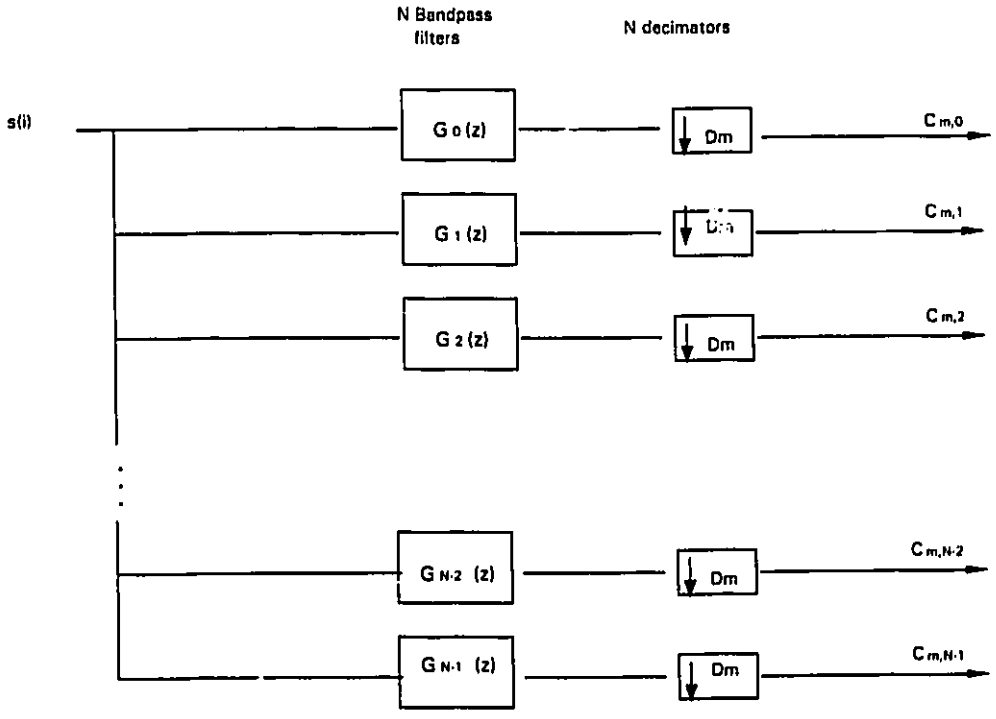

Figure 5-1: Diagram for Implementation of WBGT in a Filterbank

5.2 Design of the Low Pass Filter

Figure 5-1 provides the basic structure for the implementation of the WBGT in a filterbank. However, the properties of each of the BPF's is not described and is therefore addressed below. For the WBGT, it is required that the BPF's window the input signal in time and generate as output the magnitude and phase of each frequency band for that windowed segment. The required filter bandwidth of the BPF's is described later in this section.

For the windowing to reflect the local frequency content of the input signal, the length of the impulse response of the BPF's, L_W should be relatively short in comparison to the length of the signal, L. Good frequency domain resolutions however, requires several BPF's, each with a narrow bandwidth. Filters with a narrow bandwidth generally have a long impulse response (large L_W). The requirements of short L_W for good time domain resolution, and long L_W for good frequency domain resolution are conflicting. The optimum solution to the two conflicting requirements is to use a LPF of gaussian impulse response. The gaussian function has a minimum time-bandwidth product [4] and therefore optimizes the requirement for short impulse response and narrow bandwidth. From the gaussian LPF, BPF's are derived by frequency translation as described by (5-1) and (5-2).

Consider the analysis function, $\gamma(i)$, as the basic LPF to be used in a filterbank implementation of the WBGT. Since it is of finite impulse response, it can perform the windowing function required in the WBGT[14, 20]. When $\gamma(i)$ was computed by the method described in (3-14) it resembled b(i) which was gaussian in shape (provided that an oversampling rate, $\xi \geq 2$, was used). As Gabor noted [4], the frequency response of a gaussian function in the time domain is a gaussian function in the frequency domain. The frequency response of the gaussian function has the characteristics of a LPF since it is effectively bandlimited. Thus the frequency response of $\gamma(i)$, which is denoted by $\Gamma(f)$, will approximate a gaussian LPF and may be suitable for filterbank implementation. Since $\gamma(i)$ is also limited in time to length L_W, it will only weigh the past L_W samples when implemented as an FIR filter [1, 3, 18]. The above properties make $\gamma(i)$ a suitable candidate for the LPF to be used in a filterbank.

The analysis function, $\gamma(i)$, only approximates the frequency response of an ideal LPF because it is limited to a length, L_W (FIR). This means that $\Gamma(f)$ is not strictly bandlimited

(all frequencies will be passed, although those above a certain frequency threshold may be negligible). There will be overlap between the frequency responses of adjacent filters in the filterbank and accurate frequency domain information will be generated, provided that there is only a limited overlap in the frequency responses of the adjacent BPF's. The half power frequency of the BPF's has been used as overlap point between successive filters [19]. $\Gamma(f)$ will be a suitable candidate for use in a filterbank, provided that the half power bandwidth is defined by $|\omega| < \pi/N$ radians. Figure 5-2 shows the frequency responses $\Gamma(f)$ of an analysis function suitable for use in a filterbank. When the half power bandwidth extends into the frequency range $|\omega| > \pi/N$ radians, then the Gabor Transform coefficients, $C_{m,n}$ don't accurately represent the local frequency content for the m^{th} window. This is because the standard deviation of the corresponding impulse response becomes narrow and the $\gamma(i)$ no longer weighs the input signal adequately resulting in a loss of signal information. This results in JTF data that is does not accurately represent the original signal and from which perfect or near perfect reconstruction of the original signal is not achievable.

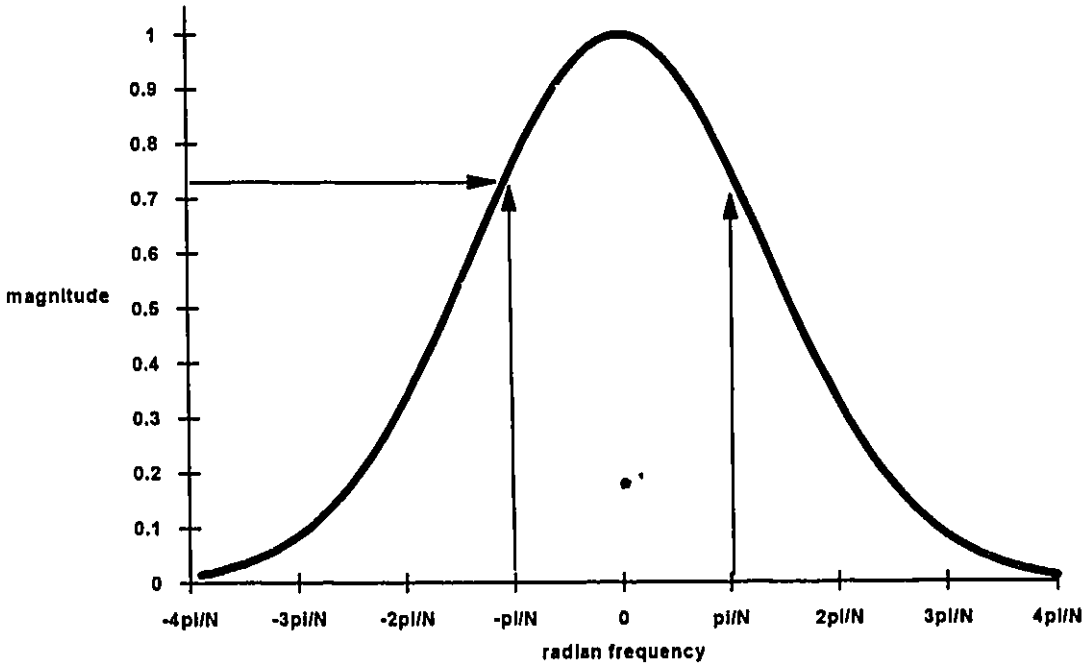

Figure 5-2: Desired frequency response of analysis functions for filterbank. The desired frequency band is between π/N and $-\pi/N$

In chapter 3 it was shown that the analysis function $\gamma(i)$ and the basis b(i) were ideally both gaussian and were nearly identical to one another within a scaling factor when (3-14) was used for $\xi \geq 2$. The variances of b(i) and $\gamma(i)$ will therefore be similar. A large variance for b(i) means a similarly large variance for $\gamma(i)$. A large variance yields a gaussian function that is wider in time, but its frequency response, $\Gamma(f)$ which is also gaussian (as stated earlier in this chapter) will have a small variance and will therefore have most of its energy concentrated near the zero frequency.

The criterion for the selection of the variance for b(i) for the WBGT is given by (3-10a):

$$\sigma^2 = N\Delta M/2\pi \qquad 3\text{-}10a$$

This selection of variance yielded a corresponding $\gamma(i)$ which was suitable for implementation of the OLGT and WBGT. However, this selection will not be suitable for filterbank implementation unless the $\gamma(i)$ has a frequency response that also conforms to:

$$\Gamma(\pi/N) \leq 0.707\Gamma(0) \qquad 5\text{-}3$$

and

$$\Gamma(-\pi/N) \leq 0.707\Gamma(0) \qquad 5\text{-}4$$

Provided that (3-10a) results in b(i) that has a large enough variance so that (5-3) and (5-4) are satisfied, $\Gamma(\omega)$ will be have a bandwidth whose half power frequencies fall within the frequency band of $\pi/N \leq \omega \leq \pi/N$, making it suitable for filterbank implementation

For cases where (5-3) and (5-4) do not hold, there will be either too much or too little overlap between successive BPF's and the errors for signal reconstruction will be high, reflecting a JTF spectrum that does not represent the original signal due to the inappropriate frequency responses of the N BPF's.

5.3 Design Equations For FIR BPF's

In section 5.2 it was shown that it is desirable that each BPF be implemented using a FIR filter. The analysis equation for the WBGT (4-1) can be adapted to that of an FIR filter.

Using FIR filtering, the $C_{m,n}$ are calculated from the past L_W inputs [18] (the length of the FIR filters) for the m^{th} shift. In (4-1), the k index is replaced by -k to reflect that a filterbank structure would have access to values originating during the last L_W inputs. The m index is adjusted to reflect this and now begins at 0 and ends at $(L+ L_W)/\Delta M$ (one time shift greater than in (4-1)). These changes are combined and given in (5-5) :

$$C_{m,n} = \sum_{k=0}^{Lw-1} s(m\Delta M - k)\gamma^*(L_w - k)\exp(-j2\pi n(L_w - k)/N) \qquad 5\text{-}5$$

where:

$0 < m < (L+ L_W)/\Delta M$.

To reconstruct the signal s(i) from the Gabor Transform Coefficients, the synthesis equation (4-3) may be directly employed and is restated here.

$$s(i = m\Delta M + k) = \sum_{m=1-Lw/\Delta M}^{M} \sum_{n=0}^{N-1} C_{m,n} b(k)\exp(j2\pi nk/N) \quad 4\text{-}3$$

In (5-5), $\gamma(i)$ is the impulse response of the LPF that the filterbank is based on. The impulse response of the BPF centered at frequency $2\pi n/N$ is given by:

$$g_n(k) = \gamma^*(L_W - k)exp(-j2\pi n(L_w-k)/N) \qquad 5\text{-}6$$

which is based on the development presented in section 5-1 where it was shown that BPF's could be obtained by using frequency shifted gaussian LPF's. By combining (5-5) and (5-6), the equation for implementation of the WBGT in a filterbank becomes:

$$C_{m,n} = \sum_{k=0}^{Lw-1} s(m\Delta M - k)g_n(k) \qquad 5\text{-}7$$

This is the convolution in the time domain between s(i) shifted to a point mΔM samples from time i = 0, with the function $g_n(k)$. From (5-7), it is seen that the WBGT can be implemented in a filterbank by taking the input signal, s(i), and filtering successive segments of it by N parallel BPF's, each of which is an FIR filter. The following section describes the filterbank configurations for this process.

5.4 Filterbank Implementations of the WBGT

Implementation of (5-7) in two filterbank structures is described below. The first type is as shown in figure 5-1 consisting of a bank of equally spaced BPF's using serial signal input and decimation after the filtering process [20]. The second configuration employs a serial to parallel conversion, followed by decimation prior to filtering by FIR BPF's as shown in figure 5-4. Alternatively, the parallel input configuration of figure 5-4 can be implemented by using a matrix containing phase factors that rotate the basic LPF to N frequencies. The matrix implementation is shown in figure 5-6.

5.4.1 Serial Input Implementation

In the serial input implementation, the s(i) is branched to N parallel lines, each line containing the same sample at any given moment in time. This is shown in figure 5-1. Each one of these lines is an input to an FIR BPF, the impulse response of which is given by (5-6). Each BPF filter calculates the content of the n^{th} frequency band of the input signal, s(i), using the previous L_W samples, as described in (5-7). Each BPF generates one output for every incoming sample of the input signal s(i). The output of each filter is decimated at a rate of one sample for every ΔM input samples (fig 5-1). This process generates the frequency content of the past L_W samples once for every ΔM input samples for each filter. Thus, for every ΔM input samples, there are N output samples generated.

The configuration of each of the BPF's is shown in figure 5-3. The BPF's implement the process described in (5-7) The weighting factor for each tap is provided by the value $g_n(k)$ as given in (5-6). The number of taps is L_W to ensure that the output of each filter is the magnitude and phase of the radian frequency $2\pi n/N$ during the past L_W of the input signal s(i) only.

Implementation of FIR filter
G $_n$(z) using serial input lines

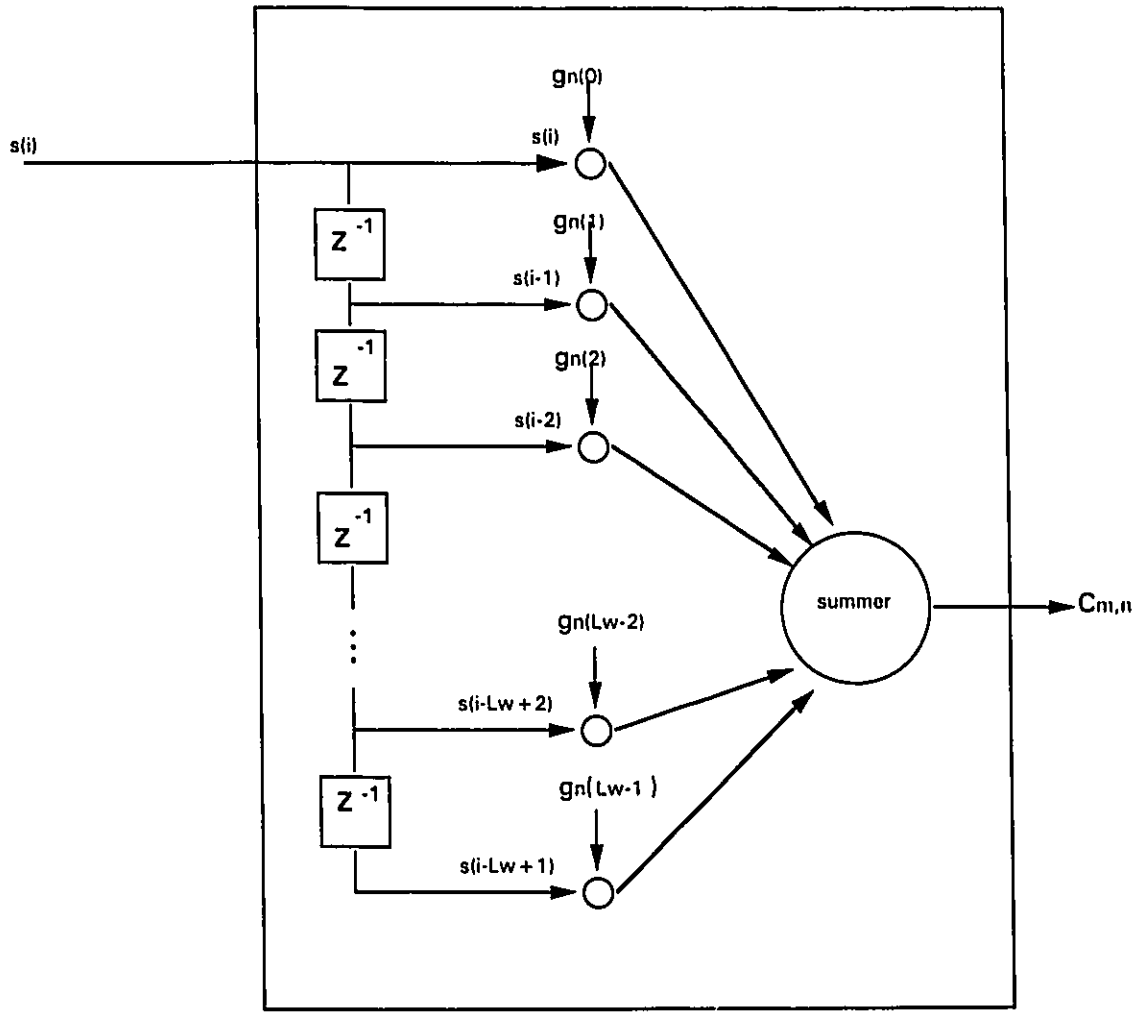

Figure 5-3: FIR Implementation of nth BPF Filter Using A Serial Input

The performance of the serial input filterbank is analysed in section 5.5. Although a large computation time savings is expected compared to the direct implementation of the WBGT (4-1), there is some inherent inefficiency in this filterbank implementation. This is because the N outputs are generated for every sample of s(i) and, only one output sample is saved from each filter for every ΔM input samples of s(i). Thus $(\Delta M-1)N$ outputs are generated and discarded prior to the calculation of the N JTF outputs that are actually

desired for each shifted window. Decimation prior to processing can reduce the total amount of processing that must be performed by the filterbank to generate the $C_{m,n}$. This is performed in the parallel input implementation described below.

5.4.2 Parallel Input Implementation

The parallel input implementation is used to reduce the time required to generate the $C_{m,n}$ compared to the serial input implementation. In the serial input implementation described in section 5.4.1, it was seen that a set of L_W delays is contained in each BPF. Since no processing occurs prior to the occurrence of the L_W delays, these delays can be executed before the bank of BPF's as shown in figure 5-4. The output of each delay is decimated every Dm, where Dm = ΔM . This process is effectively a serial to parallel conversion which results in L_W parallel samples of s(i) for every ΔM input samples of s(i). The L_W parallel lines contain the present sample and the previous L_W-1 samples of s(i). These are the samples required for the implementation of (5-7).

The samples on the L_W parallel lines are then passed as input to each of the N BPF's as shown in figure 5-4. Each BPF takes the form of figure 5-5. This configuration is effectively that of figure 5-3 with the tapped delay process performed outside the filter block. The N outputs are the $C_{m,n}$ given by (5-7).

The structure of the parallel input implementation is more efficient than the serial input implementation. This is because the output of each of the N BPF's is used as a $C_{m,n}$. There is therefore less processing overhead than for the serial input implementation, where (ΔM-1)N outputs had to be generated and discarded for every N desired outputs. This configuration is ideally suited to applications where the desired number of frequency outputs is predetermined and the processing for each filter can be performed on parallel processing boards or one custom boards consisting of parallel circuits to implement each filter. The performance of the parallel input implementation is analysed in section 5-5.

Filterbank implementation of WBGT using a parallel input technique

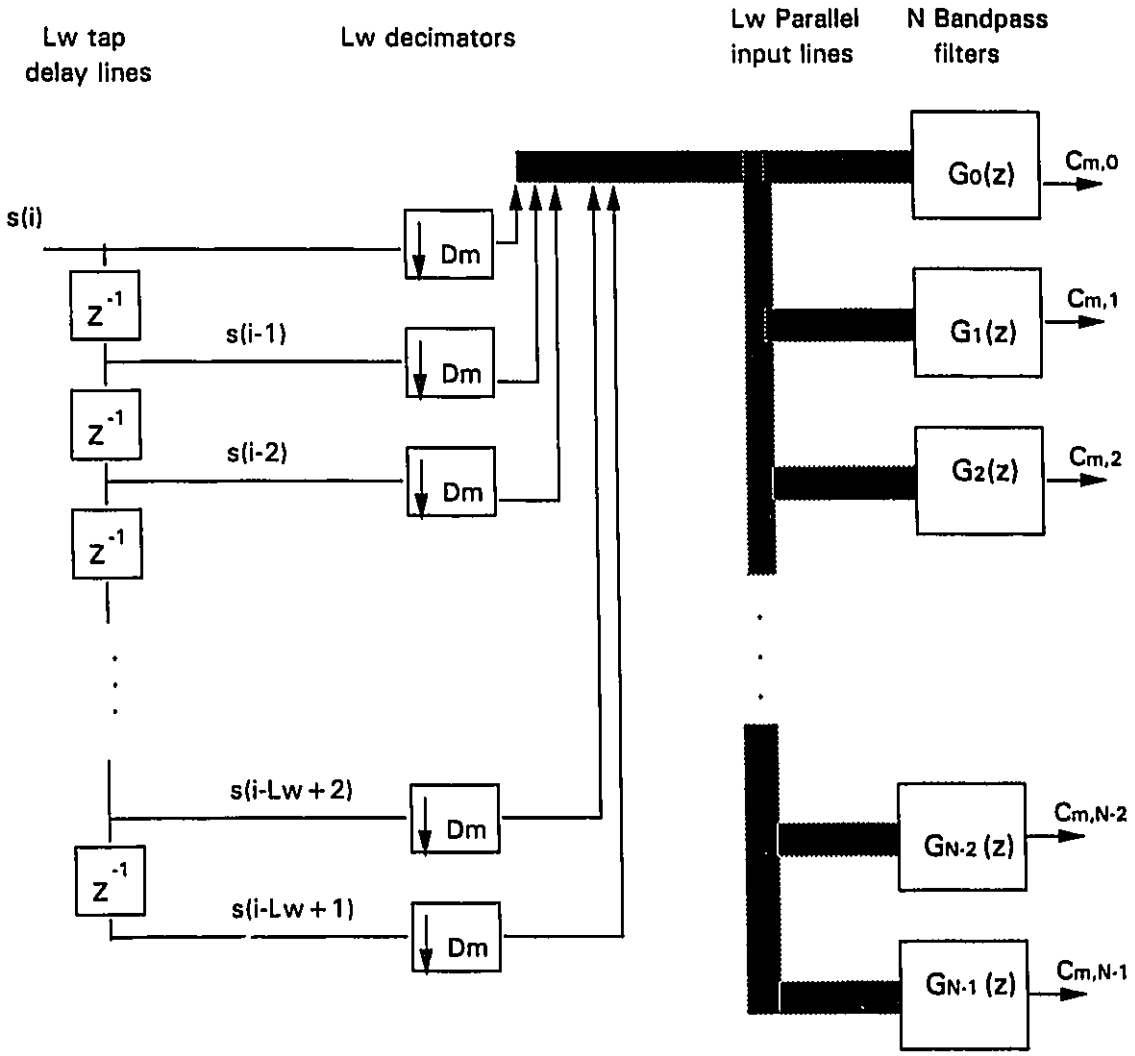

Figure 5-4: Parallel Input Implementation of WBGT in Filterbank

Implementation of FIR filter
$G_n(z)$ using parallel input lines

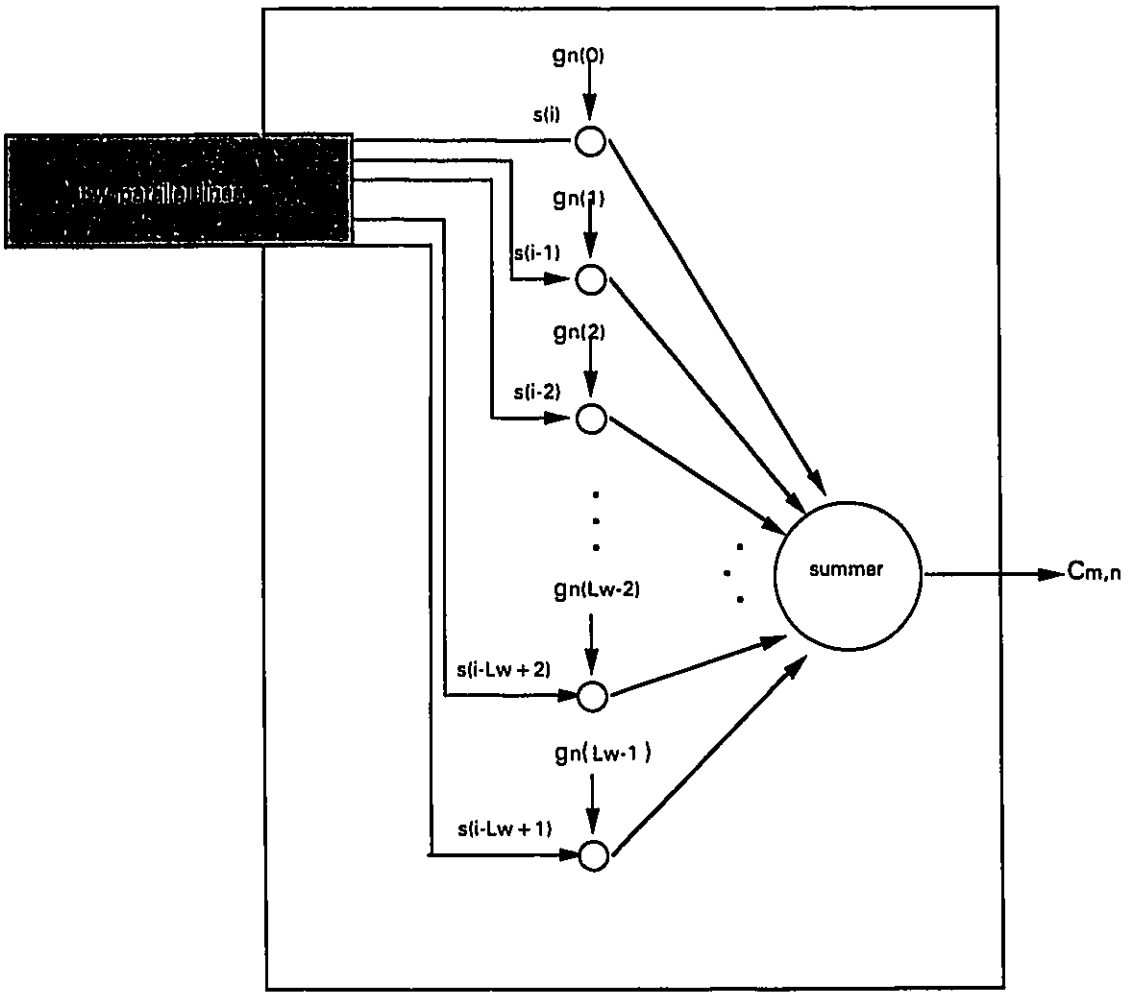

Figure 5.5: FIR Implementation of nth BPF Using Parallel Input lines

5.4.3 Matrix Implementation of Parallel Input Filterbank

An alternative approach is to implement the filterbank section of figure 5-4 by a matrix that effectively rotates the basic LPF to N BPF's and generates the N outputs from those

BPF's. The matrix implementation effectively performs the filtering by N BPF's giving the frequency content of those N frequencies for the signal s(i) during every time shift of ΔM.

As in the case of the parallel input implementation that used BPF's, the matrix implementation employs a serial to parallel conversion. This results in L_W parallel lines containing the present sample and the past $L_W - 1$ samples of s(i) after every ΔM samples. Each sample is then weighed by the complex conjugate of the impulse response of the LPF, $\gamma^*(k)$, that the filterbank is based on. This results in a vector denoted by $\mathbf{x}$ whose dimensions are $L_W x 1$. The $\mathbf{x}$ vector contains a windowed section of the signal, s(i). The $\mathbf{x}$ vector serves as an input to the next block which contains the filter rotation matrix, $\mathbf{W}$. The filter rotation matrix receives the $\mathbf{x}$ vector as input and executes the N band pass filtering operations by the process of pre multiplication. The result of this process is the $\mathbf{c}$ vector. This is stated below in (5-8)

$$\mathbf{c} = \mathbf{W}\mathbf{x} \qquad 5\text{-}8$$

where:

$$\mathbf{x} = [\, \gamma^*(L_W - 1)s(m\Delta M) \;\; \gamma^*(L_W - 2)s(m\Delta M - 1) \;\; \ldots \;\; \gamma^*(0)s(m\Delta M - L_W + 1)]^t \qquad 5\text{-}8a$$

and:

$\mathbf{W}$ is an $N x L_W$ matrix whose elements are given by:

$$W_{n,k} = \exp(-j2\pi n(L_W - k)/N) \qquad 5\text{-}8b$$

$0 \le n \le N-1$, $0 \le k \le L_W - 1$, and $N \le L_W$

The $\mathbf{c}$ vector contains the Gabor Transform coefficients $C_{m,n}$ for the mth shift. This is because (5-8) is effectively a matrix implementation of (5-7). The elements of the $\mathbf{c}$ vector are as follows:

$$\mathbf{c} = [\, C_{m,0} \;\; C_{m,1} \; C_{m,2} \cdots C_{m,N-1}]^t \qquad 5\text{-}8c$$

The implementation of (5-8) is shown below in figure 5-6. Note that the weights $g_o(k) = \gamma^*(k)$ as described in (5-6).

Implementation of Filterbank Using Filter Rotation Matrix

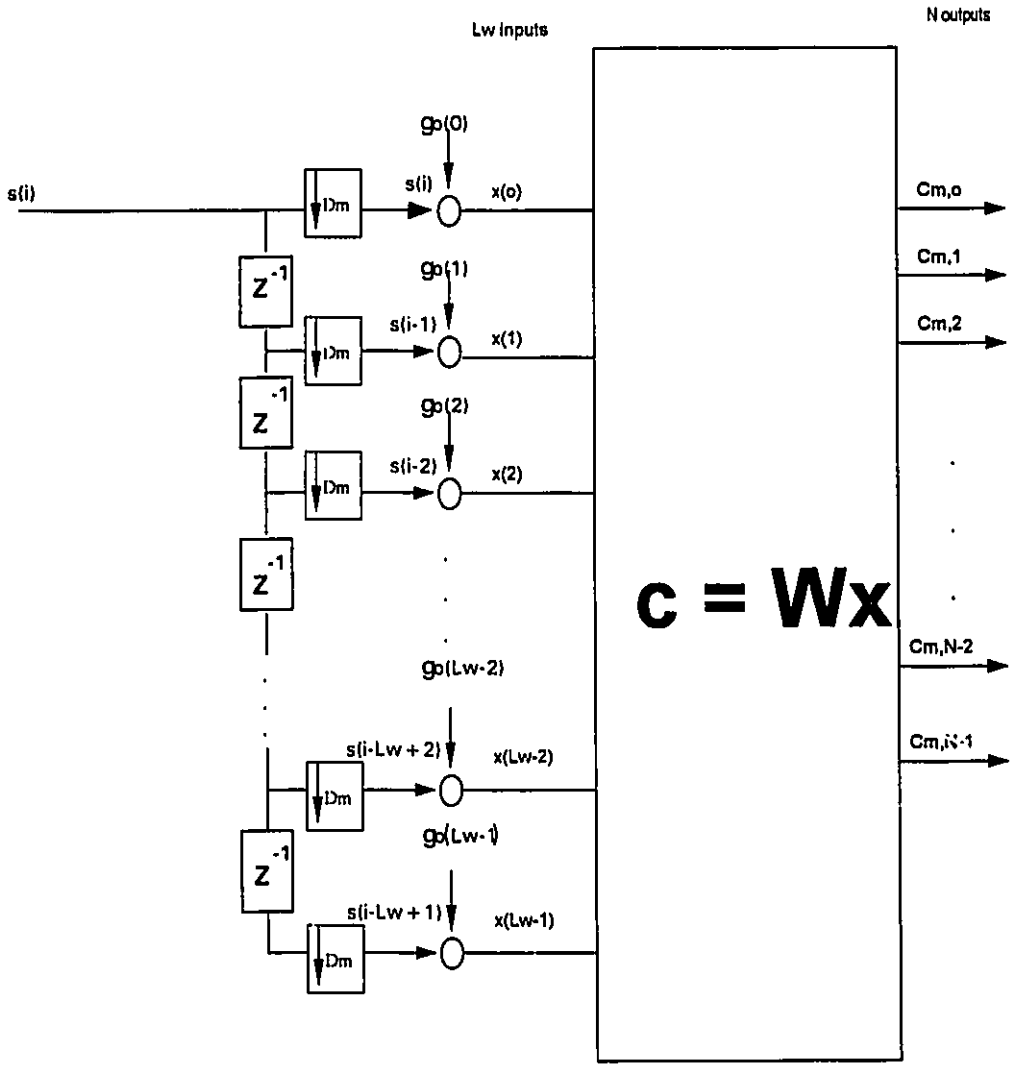

Figure 5-6: Matrix Implementation of the WBGT with Parallel Input

5.5 Performance Analysis of Filterbank Implementations

In this section the input signal, s(i) from chapter 4 is analysed for the 6 simulations performed in chapter 4, using the serial input and the parallel input implementations of the

WBGT. The parallel input implementation is then analysed using the TWBGT. Following this the matrix implementation is also studied. The results are compared to those that were obtained in chapter 4 without using filterbanks. The filterbank structures are simulated in software using Professional Matlab for Windows on a 486 PC with a 66 MHz clock rate. All simulations used the signal s(i) of L = 512 and analysis functions, $\gamma(i)$, of $L_W = 64$

5.5.1 Results for Serial Input Implementation

Table 5-1 contains the simulations performed for the serial input implementation. These simulations use the same analysis parameters as those in tables 4-2 (OLGT) and table 4-3 (WBGT). The filterbank described by figures 5-1 and 5-3 is implemented with the parallel processing of the N frequency bands (as shown in figure 5-3) simulated in software. The analysis time for each simulation is given by t_{serial}. It is assumed that N output samples from the filterbank can be generated prior to the arrival of the next input sample. This results in some constraints that limit the arrival rate of input samples or the speed of the required hardware. This will be addressed below.

The columns framed with a double line pertain to the serial input filterbank implementation and the other columns (t_{OLGT}, t_{WBGT}) are provided for reference and comparison.

Simu-lation	M	N	ΔM	ξ	M*N	t_{OLGT}	t_{WBGT}	t_{serial} filterbank	mults T_{mult},	adds T_{add}	err_{av} filterbank (WBGT)	err_{db} filterbank (WBGT)
#1	71	16	8	2	1136	382.76	59.98	26.3526	512	32k	0.1102	-19.15
#2	35	32	16	2	1120	373.33	58.22	27.0499	512	32k	0.1347	-17.41
#3	17	64	32	2	1088	364.16	66.73	26.2117	512	32k	1.09E-14	-279.25
#4	71	24	8	3	1706	578.25	91.79	26.7251	512	32k	0.0259	-31.73
#5	35	48	16	3	1680	559.63	86.24	27.1697	512	32k	0.0534	-25.49
#6	35	64	16	4	2240	756.71	119.30	25.9088	512	32k	8.57E-15	-281.34

Table 5-1: Comparison of Serial Input Implementation with the WBGT and OLGT, L = 512, $L_W = 64$

The results from table 5-1 show that the serial input filterbank implementation results in a large reduction in analysis time compared to the results of tables 4-2 and 4-3 while maintaining identical reconstruction error.

For the filterbank implementations, samples are shifted by one delay once during every time period of T_{del}. For the serial input implementation, the following operations occur in the N parallel filters (shown in figure 5-3) as each one of the L input samples arrives at the filterbank:

(i) there are L_W-1 delays performed simultaneously the total time for which is the time between the input and the output of one delay and is T_{latch};
(ii) L_W multiplications occur in parallel in each filter the time for which is the time for one multiplication (T_{mult}); and
(iii) L_W-1 additions occur in series resulting in one output for each filter, the time taken for each addition being T_{add}.

T_{mult}, T_{add} and T_{latch} are constants and depend on the hardware chosen to implement the filterbank. As a result, they will be the same for any filterbank implementation provided that the same hardware is used for multipliers and adders.

The time taken to generate N parallel outputs must be less than the rate at which new samples are arriving at the filterbank which is assumed as being the same as T_{del}. Assuming that $L_W >> 1$ and $T_{latch} \leq T_{add}$ [25], then the time taken to generate N parallel outputs is $T_{mult} + L_W T_{add}$. From this it can be said that:

$$T_{del} \geq T_{mult} + L_W T_{add} \qquad 5\text{-}9$$

Assuming that the period between the arrival of successive samples at the filterbank is the same as the period that the signal s(t) is sampled at (T_s), then

$$T_s \geq T_{mult} + L_W T_{add} \qquad 5\text{-}10$$

Note that for SAR signals, the sampling in the cross range dimension is performed by transmitting and receiving successive pulses. The sampling period is therefore the inter

pulse period or $1/f_p$. From Nyquist's sampling theory it is known that the minimum sampling frequency of a signal must be at least twice the signal bandwidth (B_D):

$$1/T_s \geq 2B_D \qquad 5\text{-}11$$

$$f_p \geq 2B_D \qquad 5\text{-}11a$$

For SAR cross range processing the signal bandwidth of interest for JTFA is the doppler bandwidth, B_D. This is because the filterbank processes successive range gates of data as shown in figure 2-3. Combining (5-10) and (5-11), it can be seen that the maximum input signal bandwidth that a serial input filterbank can accommodate is given by:

$$B_D \leq 0.5(T_{mult} + L_W T_{add})^{-1} \qquad 5\text{-}12$$

The computation of the N parallel outputs occurs a total of L times. As described above, the N outputs are saved once only for every ΔM input samples, but the total processing time, t_{serial}, is given by:

$$t_{serial} \approx L(T_{mult} + L_W T_{add}) \qquad 5\text{-}13$$

The result of this is a constant t_{serial} for all six simulations in table 5-1 (approximately 26 seconds) and is reflected by the contant number of multiplications and additions for each simulation. Since the N outputs are decimated after filtering, the decimation rate (which is different in each simulation) does not result in a reduction of t_{serial}. The only variable that could have a real effect on t_{serial} is the filter length, which in the case of all of the above simulations was $L_W = 64$.

The reduction in analysis time compared to the WBGT without filterbank implementation (4-1) was between 56% and 78%, and resulted from the N frequency outputs being generated in parallel. When all other parameters are kept constant, the time reduction compared to the WBGT (without filterbank) increases as ξ increases. This is because the number of frequency bands N increases as ξ increases. Since the N outputs are computed in parallel, the use of the serial input implementation is suitable for applications where high oversampling rates are desired.

It should be noted that he low reconstruction errors for simulations 3, 4, 5 and 6 demonstrate that the choice of parameters for these simulations result in an analysis function whose frequency response $\Gamma(f)$ is suitable for filterbank implementation. This is because the $\Gamma(f)$ meets the requirements of (5-3) and (5-4).

The conclusions that can be drawn from the performance analysis of the serial input filterbank implementation are that:

(i) there is a 56% to 78% reduction in analysis time compared to the OLGT and WBGT because the outputs for the N frequencies are computed in parallel;
(ii) time saved compared to the WBGT (without filterbank) increases as ξ increases;
(iii) identical reconstruction error is achieved compared to the WBGT and OLGT for all simulations; and
(iv) the analysis time, $\mathbf{t_{serial}}$, is approximately constant for all simulations regardless of the parameters chosen for analysis because the same number of operations are performed for each simulations.
(v) the signal doppler bandwidth that the serial input filterbank can accurately process is given by (5-12).

5.5.2 Results for Parallel Input Implementation

5.5.2.1 Filterbank Implementation of WBGT with Parallel Input

In this section the filterbank implementation of the WBGT with parallel input implementation is simulated for the six simulations used in table 5-1 with the signal s(i) of length L and using an analysis window of length $L_W = 64$. The process of figure 5-4 is implemented in this section with the serial to parallel conversion and the parallel processing of the N BPF's simulated in software. The analysis time is given by $\mathbf{t_{parallel}}$. The columns framed with double lines pertain to the parallel input filterbank implementation.

Simu-lation	M	N	ΔM	ξ	M*N	t_{WBGT}	t_{serial} filterbank	$t_{parallel}$ filterbank	mults T_{mult},	adds T_{add}	err_{av} parallel filterbank (WBGT)	err_{db} parallel filterbank (WBGT)
#1	71	16	8	2	1136	59.98	26.3526	3.2945	71	4.4k	0.1102	-19.15
#2	35	32	16	2	1120	58.22	27.0499	1.6906	35	2.2k	0.1347	-17.41
#3	17	64	32	2	1088	66.73	26.2117	0.8193	17	1.1k	1.09E-14	-279.25
#4	71	24	8	3	1706	91.79	26.7251	3.3406	71	4.4k	0.0259	-31.73
#5	35	48	16	3	1680	86.24	27.1697	1.6981	35	2.2k	0.0534	-25.49
#6	35	64	16	4	2240	119.30	25.9088	1.6193	35	1.1k	8.57E-15	-281.34

Table 5-2:Comparison of Parallel Input Implementation with the WBGT

Results from table 5-2 show a significant reduction in analysis time compared to using the serial input filterbank. Once again, identical error is maintained for reconstruction for all simulations compared to the WBGT and OLGT.

For the parallel input implementation, the following operations occur in N parallel filters after the arrival of each successive time shift of ΔM samples as input to the filterbank:

(i) L_W multiplications occur in parallel in each filter the time for which is the time for one multiplication (T_{mult}); and
(ii) L_W-1 additions resulting in one output for each filter, the time taken for each addition being T_{add}.

The total number of these operations is given for each simulation in table 5-2. The time taken to generate N parallel outputs must be less than the time during which ΔM new samples arrive at the fiterbank input and pass through ΔM - 1 delays. As in the case of the serial input implementation the time taken to generate N parallel outputs is $T_{mult}+ L_W T_{add}$ (which takes into account the L_W-1 parallel latches from the serial to parallel conversion). Assuming that T_{mult} and T_{add} are constants, this permits a much smaller arrival period between input samples than for the serial input implementation. This is given by:

$$T_{del} \geq (T_{mult}+ L_W T_{add})/\Delta M \qquad 5\text{-}14$$

Combining (5-11), and (5-14) the permitted doppler bandwidth is given by:

$$B_D \leq 0.5\Delta M(T_{mult} + L_w T_{add})^{-1} \qquad 5\text{-}15$$

Assuming that T_{del} and B_D are determined by the radar system and are independent of the filterbank configuration used, then for the parallel input filterbank, it is possible to use adders and multipliers that are ΔM times slower than for the serial input filterbank.

To compute the analysis time for the entire signal, it is noted that the filtering process occurs in N parallel filters a total of M times and since $L = M\Delta M$, the analysis time is given by:

$$t_{parallel} \approx L(T_{mult} + L_w T_{add})/\Delta M \qquad 5\text{-}16$$

A quick comparison reveals that (5-16) is simply (5-13) divided by ΔM. This is what is intuitively expected by performing the decimation prior to filtering.

In table 5-2, it is seen that the analysis time ,$t_{parallel}$, varies for each simulation and decreases as ΔM is increased to confirm (5-16). As ΔM decreases in length, the number of shifts, M, increases resulting in a greater number times that the filtering process of figure 5-5 must be performed. In simulations #1 and #4, $\Delta M = 8$ and $M = 71$, resulting in $t_{parallel}$ that is about twice as long as for simulations #2 ,#5 and #6 where $\Delta M = 16$ and $M = 35$. These results confirm the predication of (5-16) and are reflected by the corresponding increase in multiplications and additions as M increases. Comparing (5-13) and (5-16), it is seen that $t_{parallel} \approx t_{serial}/\Delta M$.

The oversampling rate, ξ does not have an effect on the $t_{parallel}$. This is because ξ determines the number of filters, N, that are used in the filterbank. As all of these BPF operations are performed in parallel, an increase in ξ has no effect on $t_{parallel}$ even though the number of coefficients $C_{m,n}$, that will be generated for each time shift of length ΔM increases. This makes the parallel input filterbank implementation attractive when high oversampling rates are required.

The parallel input implementation results in a large reduction in the analysis time, $t_{parallel}$, compared to t_{serial}, t_{OLGT}, and t_{WBGT}. The $t_{parallel}$, drops to the levels achieved using the STFT (with FFT) as shown in table 4-1. The results show that the parallel input implementation may be suitable for real time processing as it achieves times similar to the STFT with FFT. The parallel input implementation generates less coefficients than the STFT.

The errors for reconstruction for the parallel input implementation of the filterbank is exactly the same as those calculated when the OLGT and the WBGT were implemented (see tables 4-1 and 4-2). This was because the parallel input filterbank is an implementation of (5-7). Since (5-7) was derived from (4-1) identical reconstruction errors occur on the simulations performed in tables 4-1, 4-2, 5-1 and 5-2.

The conclusions that can be drawn from the performance analysis of the parallel input filterbank are that:

(i) $t_{parallel}$, drops to the levels achieved using the STFT (with FFT);
(ii) for identical simulation parameters $t_{parallel} \approx t_{serial}/\Delta M$
(iii) the smaller the time shift, ΔM, the larger the $t_{parallel}$ for the same oversampling rate due to the larger number of total shifts;
(iv) ξ has no effect on $t_{parallel}$ because the N BPF operations are performed in parallel; and;
(v) error for reconstruction is the same as in tables 4-2 and 4-3 (OLGT and WBGT).
(vi) the doppler bandwidth that can process is given by (5-16) and is ΔM times greater than when using the serial input implementation. Conversely, for identical sample arrival times, slower hardware may be used generate the Gabor Transform coefficients.

It must be remembered that the parallel implementation does result in a lower analysis times unless the processing for the N BPF's can be performed in parallel on dedicated processing boards for each BPF or on a board consisting of parallel chips that implement each BPF function. This may not be possible in some applications where both L_W and N is large and would therefore would require a large number of identical circuits to implement each BPF.

5.5.2.1 Filterbank Implementation of TWBGT with Parallel Input

In this section the filterbank implementation of the TWBGT with parallel input implementation is analysed for the six simulations used in the previous sections with the signal s(i) of length L and using an analysis window L_{WT} = 32. As in the previous section, the serial to parallel conversion and the parallel processing of the N frequency bands are simulated in software. The columns with double frames pertain to this section.

Simu-lation	M	N	ΔM	overlap	ξ	4σ	M*N	$t_{parallel}$ filterbank (WBGT)	err_{db} parallel filterbank (WBGT)	$t_{parallel}$ filterbank (TWBGT)	mult s T_{mult}	adds T_{add}	err_{db} parallel filterbank (WBGT)
#1	67	16	8	0.75	2	18.0	1072	3.6532	-19.15	1.65	71	2.2k	-19.31
#2	33	32	16	0.5	2	36.1	1056	2.0309	-17.41	0.76	35	1.1k	-46.08
#3	16	64	32	0	2	72.2	1024	1.1677	-279.25	0.37	17	0.5k	-13.71
#4	67	24	8	0.75	3	22.1	1608	3.7136	-31.73	1.67	71	2.2k	-29.11
#5	33	48	16	0.5	3	44.2	1584	2.0704	-25.49	0.78	35	1.1k	-27.80
#6	33	64	16	00	4	51.1	2112	1.9581	-281.34	0.85	35	0.5k	-22.37

Table 5-3: Parallel Input Implementation of TWBGT
($L = 512$, $L_W = 64$, $L_{WT} = 32$)

The results for the parallel input implementation of the TWBGT show that in general the trends that were found in the previous section apply, but that in all cases, the analysis time, $t_{parallel}$, was about 50% of those in table 5-2.

The parallel input implementation using the TWBGT reduces the complexity of the filterbank by decreasing the number of delays as well as the number of multipliers and adders required in each BPF. Because the window length is reduced from L_W to L_{WT}, the number of delays is reduced by $L_W - L_{WT}$. In each BPF, the complexity of the filtering operation is reduced as $L_W - L_{WT}$ less multipliers and $L_W - L_{WT} + 1$ adders are required as seen in table 5-3.

When the TWBGT is used, the equation for $t_{parallel}$ is the same as (5-16) except for the window length L_W being replaced by the truncated window length L_{WT}. This results in the following:

$$t_{parallel} \approx L(T_{mult} + L_{WT}T_{add})/\Delta M \qquad 5\text{-}17$$

$t_{parallel}$ decreases by 50% for all simulations between tables 5-2 and 5-3 since L_{WT}= 32 and L_W = 64 and all other parameters in (5-17) remain identical for each simulation. It should also be noted that the computation of each coefficient can be performed in a time that is approximately L_{WT}/L_W faster than when truncated windows are not used which permits a reduced arrival time between successive input samples which is given by (5-14) with L_W replaced by L_{WT}

$$T_{del} \geq (T_{mult} + L_{WT}T_{add})/\Delta M \qquad 5\text{-}18$$

As a result, the permitted signal doppler bandwidth increases to:

$$B_D \leq 0.5\Delta M(T_{mult} + L_{WT}T_{add})^{-1} \qquad 5\text{-}19$$

If a constant sample arrival rate T_{del} is predetermined by the radar system than slower adders may be used than for the parallel input implementation without truncated windows, because less additions are required between the arrival of successive samples.

As with the parallel input line implementation of the WBGT, for each simulation in table 5-3, when ΔM decreased in length, the $t_{parallel}$ increases as a result of a greater number of total time shifts (M) required to analyse the entire input sequence s(i) of length L.

The seventh column in table 5-3 shows the value of four standard deviations, 4σ, of the basis function for each simulation. When $L_{WT} < 4\sigma$, the error for reconstruction is large compared to when $L_{WT} \geq 4\sigma$. This is the result that would have been predicted using the conclusions from chapter 4. From this it is concluded that the allowable reduction in circuit complexity should not exceed $L_W - 4\sigma$; i.e., the number of delays in the serial to parallel conversion reduced to $4\sigma - 1$, the number of multipliers in each BPF is 4σ and the number of adders in each BPF is 4σ-1.

The conclusions that can be drawn from the performance analysis of the filterbank implementation of TWBGT with parallel input are that:

(i) $t_{parallel}$ was reduced by the ratio L_{WT}/ L_W when compared with the filterbank implementation without truncated windows ;
(ii) the number of delays is reduced by $L_W - L_{WT}$.
(iii) in each BPF $L_W - L_{WT}$ less multipliers and $L_W - L_{WT} + 1$ adders are required;
(iv) the allowable reduction in circuit complexity should not exceed $L_W - 4\sigma$; and
(v) the permitted time between the arrival of successive inputs is reduced by approximately L_{WT}/L_W.

5.5.3 Results for Matrix Implementation of the Parallel Input Filterbank

In this section the matrix implementation of the parallel input filterbank is analysed for the six simulations used in the previous sections with the input signal, s(i), of length L and using an analysis window $L_W = 64$. The structure of figure 5-6 was implemented, with the number of delays in the serial to parallel conversion section being $L_W-1 = 63$ to implement the analysis window of length 64. The serial to parallel conversion is the only section in the process that is simulated in software. (5-8) is directly used to calculate the Gabor Transform Coefficients. The analysis time is given by t_{matrix}. Those columns framed with double lines are the results for the matrix implementation of the parallel input filterbank.

Simu-lation	M	N	ΔM	ξ	M*N	t_{OLGT}	t_{WBGT}	t_{serial} filterbank	$t_{parallel}$ filterbank	t_{matrix} filterbank	err_{av} matrix (WBGT)	err_{db} matrix (WBGT)
#1	71	16	8	2	1136	382.76	59.98	26.353	3.295	0.16	0.1102	-19.15
#2	35	32	16	2	1120	373.33	58.22	27.049	1.691	0.32	0.1347	-17.41
#3	17	64	32	2	1088	364.16	66.73	26.212	0.819	0.37	1.09E-14	-279.25
#4	71	24	8	3	1706	578.25	91.79	26.725	3.341	0.23	0.0259	-31.73
#5	35	48	16	3	1680	559.63	86.24	27.169	1.698	0.43	0.0534	-25.49
#6	35	64	16	4	2240	756.71	119.30	25.909	1.619	0.65	8.57E-15	-281.34

Table 5-4: Comparison of Matrix Implementation of WBGT with other techniques

The results from table 5-4 show that the matrix implementation results in further reductions in analysis time compared to the parallel input filterbank, while maintaining identical error for signal reconstruction from the $C_{m,n}$.

The analysis times t_{matrix}, achieved using the matrix implementation with parallel input is dependent on the parameters N, ΔM, L, and L_W. The first step in the process after the serial to parallel conversion is the L_W parallel multiplications required in (5-8) to form the **x** vector. This is followed by a matrix multiplication between the **W** matrix (of dimension NxL_W) and the **x** vector (dimension $L_W x1$). This occurs for every time shift of ΔM. The resulting t_{matrix} is given by :

$$t_{matrix} = M(T_{mult} + (T_{(NxL_W \times L_W \times 1)})) \qquad 5\text{-}20$$

where $T_{(NxL_W \times L_W x1)}$ is the time required to execute the matrix multiplication operation of (5-8) once (this occurs during every time shift). This matrix multiplication operation consists of NxL_W multiplications and $(N-1)xL_W$ additions. From this, it can be seen that as either N or L_W increases, $T_{(NxL_W \times L_W x1)}$ will increase because the resulting **W** matrix is larger. Since L_W was constant for all the simulations, the size of the **W** matrix was only affected by the number of frequency bands, N. When the length of the time shift, ΔM was constant (as in simulations 2, 5 and 6 or for simulations 1 and 4), it was seen that the increase in t_{matrix} was approximately proportional to the increase in N. Using the matrix

implementation the analysis time increases as the total number of JTF coefficients increases because the processing of the N BPF's does not occur in parallel.

From (5-19), it is also evident that when all other parameters are constant, a smaller ΔM will increase the t_{matrix} because a larger number of time shifts. M, will be required to analyse the entire input sequence. This is demonstrated in simulations 1,2 and 3 in table 5-4 where t_{matrix} decreases as ΔM increases. The same holds true for simulations 4 and 5 in table 5-4.

The analysis time, t_{matrix}, for all simulations was found to be less than the parallel input filterbank implementation. This was due to the efficiency of the Matlab multiplication process. In Matlab programming, each complex number is a 1x1 matrix. As a result the total time for the parallel input implementation is that of M matrix multiplications and MxL_W matrix additions. The matrix input implementation required M 1x1 matrix multiplications and M matrix multiplications (5-20). As a result, the matrix input implementation is more efficient in Matlab programming than simulating the other filterbank implementations. This result was significant since the results obtained using the filterbank implementations in tables 5-1, 5-2 and 5-3 were achieved by simulating the processing in N parallel BPF's and the actual time taken for analysis was approximately N times longer than those shown in tables 5-1, 5-2 and 5-3. The low value of t_{matrix} makes it suitable for applications where a large number of parallel processing circuits are not available to perform the N parallel band pass filtering used in the other filterbank implementations.

As in the case of the other filterbank implementations it was found that the error for the reconstruction of the signal s(i) was identical to the results achieved using the OLGT and the WBGT. This result was expected since (5-8) is simply a matrix implementation of (5-7). It should also be noted that (5-7) is an alternative representation of (4-1). The low errors for simulations 3,4,5 and 6 showed that the parameters chosen for these simulations were suitable for filterbank implementation as was the case for tables 5-1, 5-2 .

The conclusions that can be drawn from the above are:

(i) for identical parameters, as ΔM decreases, t_{matrix} increases because the larger number of shifts, N required to analyse the entire signal.

(ii) for identical parameters, as ξincreases t_{matrix} increases because the dimensions of the W matrix increases in (5-20).
(iii) t_{matrix} was lower than the analysis times for all simulations due to some characteristics of Matlab software

5.6 SAR Processing Using Filterbanks

5.6.1 Overview

In chapter 2, the application of JTFA for SAR cross range compression was presented. In figure 2-3, the SAR processing operation was shown to be a two dimensional pulse compression operation. The two pulse compression processes can be performed independently first in range and then in the cross range dimension [1, 3, 9]. The configuration of the parallel input WBGT filterbank is well suited for use in the cross range compression process. The structure for SAR processing using a parallel input filterbank is presented in figure 5-7. This configuration is the same as that presented in figure 2-3 however the cross range matched filter processing is performed by a WBGT filterbank whose outputs are multiplied by individual weights to account for the cross range matched filter frequency response. This is shown in figure 5-8.

The SAR processing operation using filterbanks is described as follows. Digitized pulses are received by the cross range processor where they are range compressed, resulting in N_{rg} range gate samples per pulse. The N_{rg} range gate samples correspond to one range line. An example of the number of range gate samples, N_{rg} used in a SAR can be 1024. This number was used in DND's Compact SAR (CSAR) signal processor [3]. These range samples correspond to all resolvable ranges from the innermost to the outermost range gate of the radar radiation pattern. The most recently processed range line is loaded into the 2-D array from the left and pushes past range lines to the right with the range line that entered L_W pulses ago leaving the 2-D array. The resulting 2-D array is of dimensions N_{rg} x L_W.

After the 2-D array has been filled with range compressed data once (after the first L_W pulses have been transmitted, received, presummed and range compressed) and then once after the arrival of every ΔM range lines thereafter, the matrix is accessed from the range gate dimension (as shown in figure 5-7 by performing a corner turn) prior to the execution

of cross range pulse compression. The corner turn process results in a series to parallel conversion of successive samples within each range gate. Consequently, the requirement for the serial to parallel converter within the subsequent filterbank block is eliminated. For each range gate, the L_W samples are passed to the parallel input filterbank along L_W parallel lines where each successive range gate is cross range compressed.

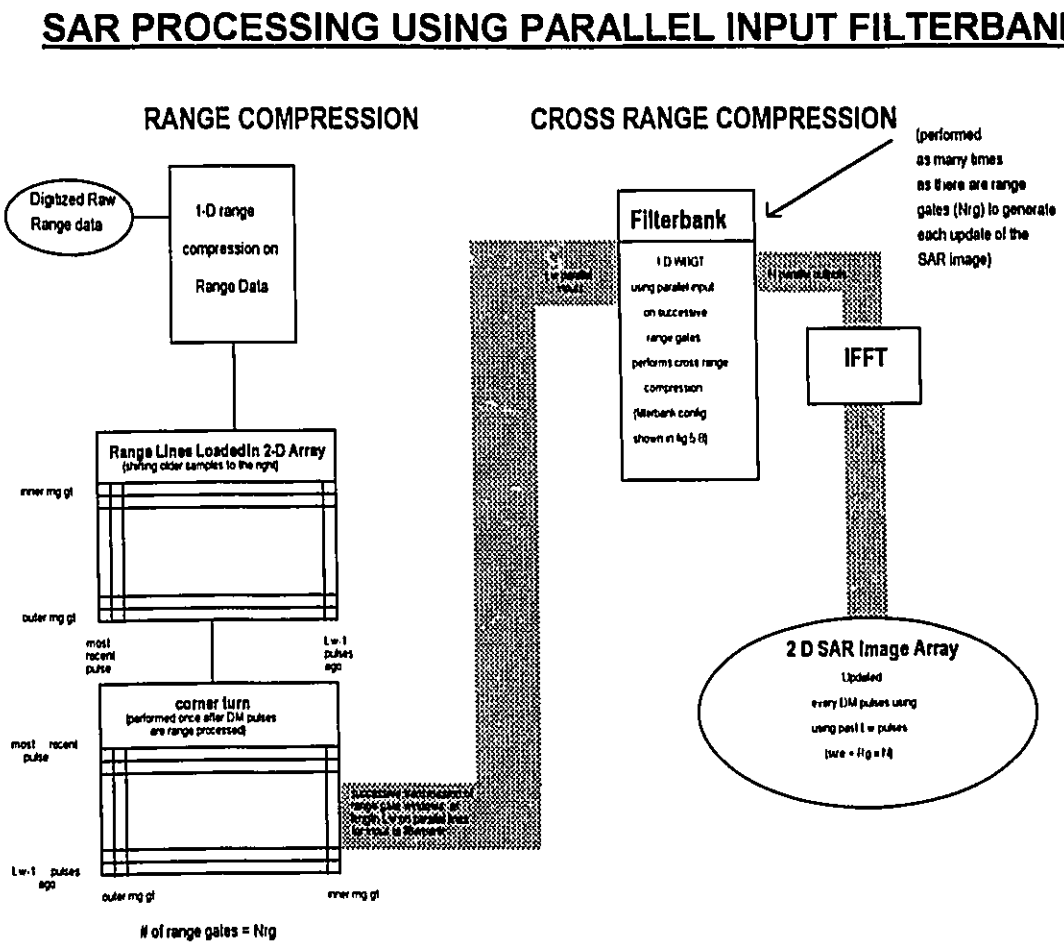

Figure 5-7: SAR Processing Using Parallel Input Filterbank

Cross range compression is performed by matched filtering on the samples in each range gate as described in chapter 2 [1, 10, 22]. A parallel input WBGT filterbank is most suitable for cross range processing since the filterbank block of figure 5-7 receives that input data in parallel format. For each range gate, the L_W points are frequency transformed by the parallel input WBGT filterbank to yield N Gabor Transform coefficients, $C_{m,n}$. These outputs are then multiplied by the frequency response of the cross range matched filter prior to output from the filterbank block (shown in figure 5-8).

The N frequency domain points containing cross range compressed data exit the filterbank. The individual filters contained in figure 5-8 are as described in figure 5-5. An Inverse Fast Fourier Transform (IFFT) on the N outputs yields a cross range compressed line which is then loaded in to the SAR image array. This process is performed N_{rg} times for each time shift, ΔM, to generate the SAR image for that time shift.

Filterbank Implementation of Cross Range Matched Filter Using a Parallel Input WBGT Filterbank

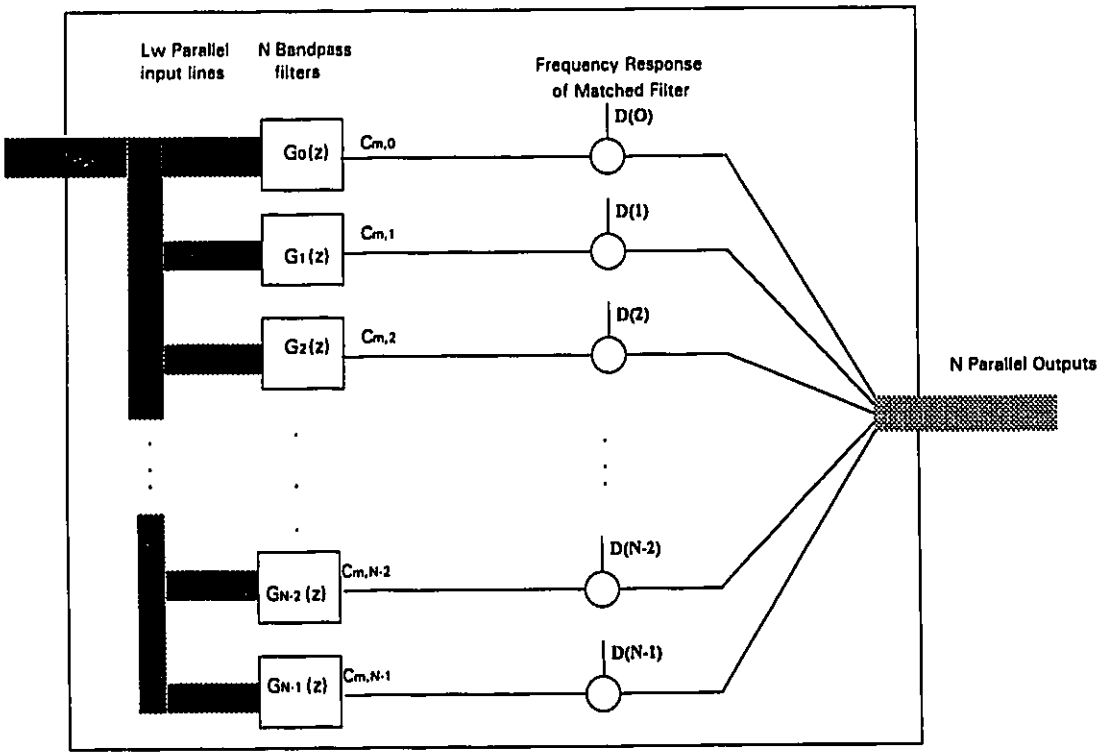

Figure 5-8: Filterbank Implementation of Cross Range Matched Filter

In chapter 2, the maximum length of the synthetic aperture, L_{ap}, was shown to be the physical length of the radar radiation pattern (2-3) in cross range. From (2-3), it is evident that L_{ap}, is dependent on the range from the radar to the terrain being imaged. For R = 40 km and $\theta_b = 2°$, the resulting synthetic aperture length, L_{ap} = 1396 m. Combining (2-1) and (2-2), the number of pulses transmitted over the synthetic aperture is:

$$L_w = f_p L_{ap}/V_a \qquad 5\text{-}21$$

Consider an aircraft velocity of 218 m/s (785 km/hr or 0.66 mach at mean sea level) and a pulse repetition frequency of $f_p = 640$. Using (5-21) and the above parameters, a total of 4096 pulses are transmitted over the length of the synthetic aperture, yielding $L_W = 4096$. The result of this is a maximum window length of $L_W = 4096$. For practical real time SAR processing, this quantity of information may be unmanageable since the 2-D range compressed array would be of dimension 1024x4096. The cross range processing would have to be performed 1024 times on a window length of $L_W = 4096$. Note that if the range R, was increased to 80 km, then there is a corresponding increase to $L_W = 8192$.

A large L_W results in a high sampling rate in the cross range dimension (at a constant range, R) which thereby accommodates a high doppler bandwidth B_D. Given that the aircraft travels a very short distance, d, between transmission of successive pulses (in the order of a few cm), the change in the doppler frequency (2-14) from a point target, is small between a few pulses. This is because its relative offset from the beam centre, y, in the cross range coordinate is very small compared to the range which can be several km. Since it is not required that the samples in cross range be reconstructed from frequency domain data and since large windows in each range gate effectively increase the frequency domain processing requirements, it is possible to reduce the amount of data in each range gate by low pass filtering [1]. This reduces the doppler bandwidth of the cross range signal while at the same time reducing the window size for each range gate. In many cases this low pass filtering reduces the number of samples within a range gate to a number that represents the actual physical doppler bandwidth of the signal. This low pass filtering is performed prior to range compression and is known as presumming [1,3]. The presum ratio or P_{ratio} is the ratio be which the number of samples in each range gate is reduced

compared to the number of samples without presumming. For the example described above, with P_{ratio}= 8, the resulting L_W = 1024 and with P_{ratio} = 16, L_W = 512.

Using the presumming step prior to processing (5-21) becomes:

$$L_w = (f_p L_{ap})/(V_a P_{ratio}) \qquad 5\text{-}22$$

The time allocated for the entire cross range compression process is, t_{cr}, (this is the same as the time between successive SAR image updates) and is given by;

$$t_{cr} = P_{ratio}\Delta M/f_p \qquad 5\text{-}23$$

This is because the time between successive pulses (inter pulse period) for processing is P_{ratio}/f_p and ΔM pulses arrive between successive SAR image updates. The time allocated for the cross range compression of the data in each range gate , Δt_{cr} , is given by:

$$\Delta t_{cr} = t_{cr}/N_{rg} \qquad 5\text{-}24$$

Combining (5-23) and (5-24) the allocated cross range compression time for each range gate is given by;

$$\Delta t_{cr} = (P_{ratio}\Delta M)/(f_p N_{rg}) \qquad 5\text{-}25$$

It shall be assumed that only 50% of this time is actually available for cross range processing of each range gate and the other 50% is used for post processing and other SAR system requirements such as motion compensation and interprocessor communication [1, 22, 23]. It is also assumed that half of the remaining time is available for the filterbank process and the other half for the IFFT process shown in figure 5-7. This leaves $\Delta t_{cr}/4$ as the time available for the filterbank to compute outputs for each range gate. The time required to generate N outputs for the parallel input filterbank after series to parallel conversion is $T_{mult} + L_W T_{add}$. The configuration of figure 5-8 however requires an additional N multiplications in parallel after the Cm,n are generated (to perform the matched filtering in the frequency domain). This results in a total time of $2T_{mult} + L_W T_{add}$ for the execution of the filterbank process for each range gate. Since the time available for the filterbank processing is $\Delta t_{cr}/4$ it can be said that:

$$\Delta t_{cr} \geq 4(2T_{mult} + L_W T_{add}) \qquad 5\text{-}26$$

using a parallel input filterbank and the total time constraint on the cross range compression process is given by:

$$t_{cr} \geq 4N_{rg}(2T_{mult} + L_W T_{add}) \qquad 5\text{-}27$$

5.6.2 SAR Processing Using Parallel Input WBGT Filterbank

In this section, some of the parameters for SAR processing using a parallel input WBGT filterbank are studied. Assuming the following parameters:

θ_b = 2°
R = 40 km
$R\theta_b$ = 1396 m
Lap = 1396 m
Va = 218 m/s (785 km/hr)
Lap/Va = 6.4 s
P_{ratio} = 16
f_p = 640
L_W = 512

then the length of the synthetic aperture, Lap, is traversed in 6.4 seconds, resulting in 4096 pulsed being transmitted and received during this time. Using a presum ratio, P_{ratio} of 16, this yields L_W = 512. Assuming a time shift of ΔM = 128 samples between successive windows for cross range processing, then a SAR image update is required every t_{cr} = 1.6 seconds if real time processing is to be achieved. The 2-D array containing range compressed data would have the dimensions 1024 (L_W) x 1024 (N_{rg}) data points. If double oversampling is used these parameters would yield SAR images following cross range compression that are 256 x 512 and at triple oversampling the image dimension becomes 384 x 512.

To determine the hardware requirements for real time cross range processing a few assumptions are made. Most hardware specification show that the processing times for multipliers are a few orders of magnitude greater than adders. Assuming that [24]

$$T_{mult} \approx 2T_{add} \qquad 5\text{-}28$$

and combining (5-27) with (5-28) the processing requirements for hardware for a given N_{rg} and L_w are given by:

$$T_{add} \leq t_{cr}/N_{rg}(L_w+4) \qquad 5\text{-}29$$

and

$$T_{mult} \leq 2t_{cr}/N_{rg}(L_w+4) \qquad 5\text{-}30$$

For t_{cr} = 1.6 seconds (corresponding to ΔM = 128 pulses), N_{rg} = 1024 range gates of data and L_W = 512 and using (5-29) and (5-30) the hardware requirements for real time processing are :

$$T_{add} \leq 3.028\ \mu s \qquad 5\text{-}31$$

and

$$T_{mult} \leq 6.056\mu s \qquad 5\text{-}32$$

In table 5-5 the effects of varying the pulse repetition frequency, f_p, and the time shift, Δ M, between successive image updates was analysed to determine the requirements for T_{mult} and T_{add} if real time processing is to be realized. The parameters are chosen so that they use 87.5%, 75% and 50% overlap at double and triple oversampling as was the case in simulations #1, #2, #3, #4 and #5 of table 5-1.

	1	2	3	4	5	6
f_p (Hz)	640	640	640	1280	1280	1280
L_w (samples)	512	512	512	1024	1024	1024
ΔM (samples)	128	128	256	128	128	256
ξ	2	3	2	2	3	2
t_{cr} (seconds)	1.6	1.6	3.2	1.6	1.6	3.2
Nrg	1024	1024	1024	1024	1024	1024
Δt_{cr} (ms)	1.563	1.563	3.125	1.563	1.563	3.125
T_{add} (μs)	3.028	3.028	6.056	1.519	1.519	3.039
T_{mult} (μs)	6.056	6.056	12.112	3.039	3.039	6.079

Table 5-5: Effects of Varying f_p and ΔM on SAR processing requirements

From table 5-5, it is seen that as f_p increases, (with all other parameters constant), the requirements for T_{mult} and T_{add} become more restrictive. This is because L_W increases as f_p increases resulting in more additions for each cross range compression operation. The oversampling rate, ξ, has no effect on the processing requirements because the processing for the N filters are performed in parallel BPF's (N increases with ξ as explained in chapter 3). As ΔM increases (and all other parameters are held constant), the upper limit for both T_{mult} and T_{add} increases. This increase is due to the larger time interval between image updates, t_{cr} for the larger ΔM.

For the parameters presented above, a TWBGT filterbank may be employed to reduce the processing requirements with only a minor loss of cross range information. The drawback of a shorter synthetic aperture will be reduced cross range resolution, however, real time processing may be more feasible. Using 50% truncation (as presented in chapter 4) the window length is reduced to $L_{WT} = 0.5L_W$. This results in a reduction in the dimensions of the 2-D array containing range compressed data to $L_{WT} \times N_{rg}$. Due to this the number of points in each range gate that must be cross range compressed is reduced by 50%. The number of multipliers and adders contained in the filterbank of figure 5-7 is reduced and since the interval between SAR image update remains unchanged, the required processing time for each multiplier and adder is reduced. The effects of using the TWBGT filterbank is shown below for the six scenarios presented in table 5-5:

	1	2	3	4	5	6
f_p (Hz)	640	640	640	1280	1280	1280
L_W (samples)	*256*	*256*	*256*	*512*	*512*	*512*
ΔM (samples)	128	128	256	128	128	256
ξ	2	3	2	2	3	2
t_{cr} (seconds)	1.6	1.6	3.2	1.6	1.6	3.2
Nrg	1024	1024	1024	1024	1024	1024
Δt_{cr} (ms)	1.563	1.563	3.125	1.563	1.563	3.125
T_{add} (μs)	*6.009*	*6.009*	*12.019*	*3.028*	*3.028*	*6.056*
T_{mult} (μs)	*12.019*	*12.019*	*24.038*	*6.056*	*6.056*	*12.112*

Table 5-6: SAR processing requirements using TWBGT filterbank

From table 5-6, it is seen that the use of a parallel input TWBGT filterbank results in processing requirements for the multipliers and adders that are about 50% slower than in table 5-5. This is because the window lengths are reduced resulting in a lower number of additions between successive image updates as given by (5-29). The use of the TWBGT filterbank also results in 50% less hardware reducing the cost of building the cross range compression circuit. The processing requirements presented in tables 5-5 and 5-6 are easily achievable using existing hardware that is commercially available. For example, T_{add} for 16 bit addition using the Texas Instruments SN54AS881A and the SN54AS88a Arithmetic Logic Units is 14 ns [25], which is several orders of magnitude lower than the requirements determined in tables 5-4 to 5-6. From (5-28) it is seen that the requirements for multiplication [24] are easily achieved.

A major disadvantage of the parallel input filterbank for cross range processing is that a circuit must be developed in advance for use with certain parameters (L_W, ΔM, ξ etc.). If these parameters are modified, then the number of filters, N and the number of parallel input lines change also. If the resulting L_W and N are greater than in the existing filterbank circuitry, then it cannot be used for real time processing. Furthermore, a dedicated circuit for cross range compression is difficult to practically implement when L_W and N are large (i.e. $L_W = 1024$). A solution to these problems would be to use the matrix implementation of the parallel input filterbank and embedding it as firmware on commercially available digital signal processing circuit boards.

5.6.3 SAR Processing Using Matrix Implementation of the Parallel Input WBGT Filterbank

In section 5.5, it was noted that the matrix implementation of the parallel input filterbank is more suitable than the other filterbank implementations for applications where dedicated hardware is not available for each of the N filters and only a software implementation is possible. This could be attractive for SAR cross range processing provided that the matrix manipulation of (5-8) can be performed N_{rg} times to generate a SAR image in the allocated time of t_{cr} which is required by (5-27) The availability of several commercial digital signal processing circuit boards with high peak processing capacities makes the possibility of use matrix implementation of the parallel input filterbank attractive.

The configuration for application of the matrix implementation for cross range processing is similar to that of figure 5-8. The bank of N BPF's is replaced by multiplication of the parallel inputs by the LPF frequency response given by $g_0(k)$ followed by rotation to N separate frequencies bands by the $\mathbf{W}$ matrix. These outputs are then multiplied by the frequency response of the cross range matched filter as in figure 5-8. The result is N frequency domain outputs that are used for the IFFT stage of the cross range compression as shown in figure 5-7.

Filterbank Implementation of Cross Range Matched Filter Using the Matrix Implementation of the Parallel Input WBGT Filterbank

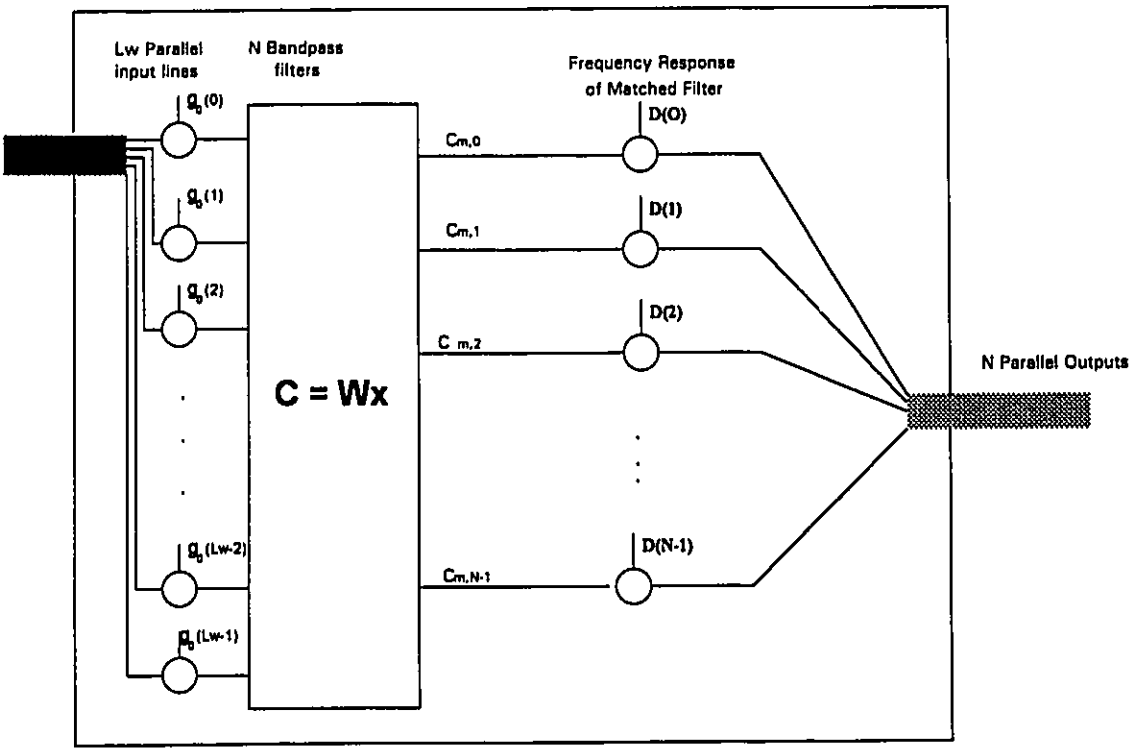

Figure 5-9: Matrix Implementation of Cross Range Processor

From figure 5-9 it is seen that the operations required to generate each set of N outputs for each range gate is sum of multipliers outside the matrix block with all the operations in the matrix block. This is given by:

$$\#\,ops = (L_W + N)mults + (L_W \times N)(mults + adds) \qquad 5\text{-}33$$

Given that 1 complex 16 bit multiplication, and one complex 16 bit addition can be performed in one clock cycle using an Intel i860™ Digital Signal Processor [26] (using the pfam.p command), then the number of clock cycles required to generate each set of N outputs for a window in a given range gate is given by:

$$\#\,cycles = (L_W + N) + (L_W \times N) \qquad 5\text{-}34$$

As described in section 5.6.2 the time available for the filterbank process during cross range compression of each range gate is $\Delta t_{cr}/4$ and as a result:

$$\Delta t_{cr} \leq 4[(L_W + N) + (L_W \times N)]/f_{proc} \qquad 5\text{-}35$$

where f_{proc} is the processor clock rate. Combining (5-24) and (5-35), the constraint for real time processing of SAR images using the matrix implementation is given by:

$$t_{cr} \leq 4N_{rg}[(L_W + N) + (L_W \times N)]/f_{proc} \qquad 5\text{-}36$$

Rearranging (5-36) the processing power required for generating SAR images in real time must be:

$$f_{proc} \geq 4N_{rg}[(L_W + N) + (L_W \times N)]/t_{cr} \qquad 5\text{-}37$$

Given that for an i860™ processor operating at 40 Mhz, the peak processing power is 80 MFlops [26] , then (5-37) becomes:

$$\mathbf{Mflops} \geq 8N_{rg}[(L_W + N) + (L_W \times N)]/t_{cr} \qquad 5\text{-}38$$

Using the parameters from table 5-5 the processing requirements for real time processing are calculated using (5-38) and summarized in table 5-7 below.

	1	2	3	4	5	6
f_p (Hz)	640	640	640	1280	1280	1280
L_W (samples)	512	512	512	1024	1024	1024
ΔM (samples)	128	128	256	128	128	256
ξ	2	3	2	2	3	2
N (samples)	256	378	512	256	378	512
t_{cr} (seconds)	1.6	1.6	3.2	1.6	1.6	3.2
Nrg	1024	1024	1024	1024	1024	1024
MFlops	675	1011	675	1348	2020	1348

Table 5-7: SAR Processing Requirements for Matrix Implementation for R = 40 km

The processing requirements calculated in Table 5-7 show that a large amount of peak processing is required to facilitate the cross range compression process in real time. A market survey reveals that there is an Octal i860™ processor produced by Loral-Rolm (Loral DXP 640) that has a peak processing capacity of 640 MFlops. Using between two and four of these boards would permit real time processing for the scenarios presented in table 5-7. However, at approximately $40,000 per board this solution for achieving real time processing may not be cost effective in many applications.

It may however by feasible to produce images in real time by reducing the range, R, and thereby reducing the length of the synthetic aperture given by (2-4). A reduction in the length of the synthetic aperture results in a corresponding reduction in the window length, L_W, thereby reducing the processing reqirements of (5-37). Using a range, R = 20 km and keeping all other aircraft paramters identical, the corresponding L_W, given by (5-22) is reduced by 50% compared to when R = 40 km was the range used for processing. The resulting processing requirements are summarized in table 5-8.

	1	2	3	4	5	6
f_p (Hz)	640	640	640	1280	1280	1280
L_W (samples)	256	256	256	512	512	512
ΔM (samples)	64	64	128	64	64	128
ξ	2	3	2	2	3	2
N (samples)	128	192	256	128	192	256
t_{cr} (seconds)	0.8	0.8	1.6	0.8	0.8	1.6
Nrg	1024	1024	1024	1024	1024	1024
MFlops	339	508	339	678	1016	678

Table 5-8: SAR Processing Requirements for Matrix Implementation for R = 20 km

From table 5-8, it seen that real time processing may be achieved at a range of 20 km with one to two octal i860™ processors such as the Loral DXP 640. It shoud be noted that since a short synthetic aperture, Lap, is used at a range R = 20 km , the resulting cross range resolution, δ_y, will not be as fine as in the case of R = 40 km (as described by (2-19)). However, this is a tradeoff that is required to implement real time processing at a reasonable cost.

Another method of reducing the hardware requirements is by using smaller window lengths and a reduced number of range gates (resulting from a smaller swath) while maintaining the range, R = 40 km. If truncated windows are used in conjuction with a narrower swath so that N_{rg} = 512 and all other parameters of table 5-7 remain the same, the processing requirements are reduced by approximately a factor of 4. These results are summarized in table 5-9.

From table 5-9 it is seen that real time processing could be achieved using one to two quad i860™ processor such as the Loral DXP 320. This solution is much more cost effective than those required in tables 5-7 and 5-8, however, the cross range resolution is reduced by using a shorter synthetic aperture L_{ap} which results from using truncated windows. In addition to this a smaller area is imaged because of the narrow radar swath.

	1	2	3	4	5	6
f_p (Hz)	640	640	640	1280	1280	1280
L_{WT} (samples)	256	256	256	512	512	512
ΔM (samples)	128	128	256	128	128	256
ξ	2	3	2	2	3	2
N (samples)	256	378	512	256	378	512
t_{cr} (seconds)	1.6	1.6	3.2	1.6	1.6	3.2
Nrg	512	512	512	512	512	512
MFlops	169	253	169	338	506	338

Table 5-9: SAR Processing Requirements for Matrix Implementatiom with $L_{WT} = 0.5\ L_W$ at 40 km.

Using truncated windows, with a narrow swath with the range reduced to R = 20 km and all other parameters identical to table 5-9, the resulting processing requirements are further reduced and are given in table 5-10. From table 5-10, it is seen that the real time processing requirements compared to table 5-9 are reduced by approximately 50% due to the reduction in range. Real time processing could be achieved by using only one quad i860™ processor such as the Loral DXP 320 which has 320 MFlops of peak computing power. The resulting cross range resolution is once again reduced compared to table 5-9 because the length of the synthetic aperture is effectively shortened by using the smaller range.

Further reductions in the processing requirements could be achieved by using larger presum ratios, than the $P_{ratio} = 16$ which was used in the above calculations. This would reduce the number of samples per window, L_W and effectively reduce the doppler bandwidth and reduce the overall processing requirements. Another way to reduce the processing requirements for real time processing is to reduce the aircraft velocity, V_a, and the pulse repitition frequency, f_p . If both V_a and f_p are reduced by 50% and all other parameters remain constant, then the time between successive image updates increases by a factor of two even though processing must be performed on the same quantity of data. This would reduce the processing requirements in tables 5-5 to 5-10 by 50% without sacrificing cross range resolution.

	1	2	3	4	5	6
f_p (Hz)	640	640	640	1280	1280	1280
L_W (samples)	128	128	128	256	256	256
ΔM (samples)	64	64	128	64	64	128
ξ	2	3	2	2	3	2
N (samples)	128	192	256	128	192	256
t_{cr} (seconds)	0.8	0.8	1.6	0.8	0.8	1.6
Nrg	512	512	512	512	512	512
MFlops	85	127	85	170	255	170

Table 5-10: SAR Processing Requirements for Matrix Implementation with $L_{WT} = 0.5\ L_W$ at R = 20 km.

5.7 Chapter Summary

Two filterbank implementations of the WBGT that reduce analysis times were presented in this chapter. One implementation used a serial input while the second used a parallel input. The parallel input version was then implemented using a matrix structure which effectively created N band pass filters (BPF's) from a Low Pass Filter (LPF) by rotating it to N equally spaced frequency bands. The errors for reconstruction of the original signal, s(i), from the JTF spectra computed by the filterbanks were the same as those achieved when using the OLGT or the WBGT synthesis equations. The application of the filterbank implementations of the WBGT was then studied in the context of SAR range and cross range processing and was shown to be feasible for real time SAR image processing.

Using the serial input filterbank implementation, the analysis time, t_{serial}, was approximately constant for all simulations (26 s). This was because approximately the same number of operations were performed for each simulation regardless of parameters as described by (5-13). A reduction in analysis time between 56% and 78% compared to the WBGT using equation (4-1) was realized. The larger reductions in analysis time were achieved at higher oversampling rates, ξ, because the outputs for the N frequencies were computed in parallel channels for a given window of data.

Using the parallel input filterbank implementation, the analysis time, $t_{parallel}$, ranged between 0.8 and 3.3 seconds. The large reduction in analysis time compared to the serial input implementation was achieved by decimation prior to processsing. It was shown that $t_{parallel} \approx t_{serial}/\Delta M$ (5-16) and therefore $t_{parallel}$ decresed as ΔM increased. When truncated windows were used a further reduction in analysis time occured and was shown to be directly proportional to the ratio L_W/L_{WT} as given by (5-17). It was also shown that the window truncation should be limited to $L_{WT} \geq 4\sigma$ if low reconstruction errors are desired. For the TWBGT, $L_W - L_{WT}$ less adders and multipliers were required in each of the N filters compared to the WBGT.

The application of the matrix implementation of the parallel input filterbank resulted in some further reduction in analysis time compared to other filterbank implementations. It was determined that for identical parameters, as ΔM decreased, t_{matrix} increased because a larger number of shifts, M were required to analyse the entire signal and for identical

parameters. It was also noted that as ξ increased t_{matrix} increased because the dimesions of the W matrix increased in (5-20). One significant aspect of the findings was that the matrix implmentation would be suitable where only one DSP board is available to implement the bank of BPF's. In such a case the matrix input implementation could be used to compute the N frequency outputs required for each time shift, ΔM.

The parallel input WBGT filterbank was shown to be suitable for cross range pulse compression in SAR image processing. The filterbank received range compressed data sequentially by range gate and cross range compressed this data by first generating the N Gabor Transform Coefficients and then weighing these outputs by the cross range matched filter impulse response. The output of the filterbank output is then Inverse Fourier Transformed to compute the cross range compressed output for the given range gate. The process was described by figures 5-7 and 5-8. The hardware requirements for multipliers and adders required for SAR image processing with filterbanks were determined and were described by (5-29) and (5-30). They were shown to be affected by the number of range lines, N_{rg}, window length, L_W, time shift, ΔM, pulse repitition frequency, f_p, range, R and half power beamwidth θ_b. For typical SAR parameters, the requirements for T_{mult} and T_{add} were shown to be achieveable with hardware that is available on the market.

The matrix implementation of the parallel input filterbank was also shown to be useful for cross range compression in SAR image processing. It was shown that using truncated windows and a narrow beamwidth in elevation, SAR image updates could be realized in real time using only one quad i860™ processor capable of a peak processing at 320 MFlops. For larger window lengths and wide beamwidth in elevation, two to four octal i860™ processor boards would be required. Due to the cost of some of these processors (about $40, 000 per Loral Octal) this would not be suitable for research purposes although it may be useful for commercial applications. Reducing aircraft velocity and pulse repition frequency proportionally resulted in lower processing requirements than those stated above without sacrificing range cross range resolution. However, this would result in a greater time interval between successive image updates.

Chapter 6: Conclusion

In this thesis, the WBGT was developed based on the OLGT. It permits the computation of Gabor Transform Coefficients as successive window of input data are received as opposed to the OLGT which requires the reception of the entire input sequence prior to processing. As a result, the WBGT was shown to be suitbable for real time processing and can be implemented in filterbank configurations. Two filterbank implementations were developed, one using serial input and the other using parallel input, each generating N frequency outputs after every ΔM inputs using a window of data containing the previous L_W input samples. The parallel input implementation was then implemented using a matrix formulation that rotated an LPF to N BPFs to generate the Gabor Transform coefficients. The hardware required to use the parallel input filterbank (both in the basic implementation that uses N BPF's and using the filter rotation matrix) was shown to be feasible for SAR cross range processing .

The reduction in analysis time when using the WBGT instead of the OLGT was shown to be proportional to the ratio of analysis window length to signal length (L_W/L). The TWBGT was developed based on the WBGT and was found to reduce analysis time by the ratio of the truncated window length to window length (L_{WT}/L_W) when compared to the analysis time for the WBGT. The error for reconstruction of the original signal, s(i), was identical for the WBGT and the OLGT when the same simulation parameters were employed. The reconstruction error generally decreased as the oversampling rate increased, because more JTFA data was used to represent the signal in the JTF domain. For the TWBGT, it was shown that the reconstruction errors remained low as long as the truncated window length was greater than or equal to four standard deviations of the gaussian basis, b(i).

It was shown that when using the STFT with FFT, the resulting analysis times were much lower than when using the WBGT. However, when using the WBGT, one can accurately represent a signal with far fewer JTF coefficients than when using the STFT (especially when a time shift is used that is much shorter than the the window length). This is the a major advantage of the WBGT over the STFT for SAR processing where cross range processing can occur on a smaller set of data. However, when the number of frequency outputs, N equals the window length L_W, the STFT and the WBGT are identical if

identical analysis windows are used ($w(k) \equiv \gamma(k)$). For these cases the STFT should be used due to the lower processing times.

For both the WBGT and the TWBGT, the choice of parameters is dependent on the application. The choices of time shift, ΔM or oversampling rate, ξ may be dictated by the system, therefore the designer is forced to use those parameters that yield the most suitable analysis times and reconstruction errors for the particular applications.

The serial input filterbank resulted in a reduction in analysis time between 56% and 78% compared to the WBGT without filterbank because the outputs for the N frequencies were computed in parallel channels for each window of data. Using the parallel input filterbank implementation, a further reduction in analysis time was achieved by decimation prior to processsing which was shown to be proportional to the rate of decimation, ΔM ($t_{parallel} \approx t_{serial}/\Delta M$). When truncated windows were used a further reduction in analysis time occured and was shown to be directly proportional to the ratio L_W/L_{WT}. The application of the matrix implementation of the parallel input filterbank resulted in reduction in analysis time compared to other filterbank implementations. In addition to this the analysis times were also lower than when using the STFT with FFT. The matrix implementation is suitable for applications where several parallel circuits are not feasible and the parallel input filterbank must be programmed in software on one DSP board. For all filterbank implementations, the reconstruction error of the original signal, s(i), from the JTF spectrum are the same as those achieved when using the OLGT or the WBGT.

The parallel input WBGT filterbank was shown to be suitable for cross range compression in SAR image processing. The timing requirements for multipliers and adders for this process was determined for typical SAR parameters and were shown to be achieveable with hardware that is available on the market. It was shown that by using truncated windows and a narrow beamwidth in elevation, SAR image updates could be realized in real time at ranges of 20 to 40 km using the matrix input implementation of the parallel input filterbank with only one quad i860™ processor capable of a peak processing at 320 MFlops such as the Loral DXP 320.

Although the matrix implementation resulted the possibility of real time SAR processing there is a possibility of inceasing the efficiency of this process. It may be possible to modify the implemention to exploits some of the symmetry in the $\mathbf{W}$ matrix since this matrix contains complex exponentials whose values fall in the range $e^{j0} \leq W_{nk} < e^{j2\pi}$. It is

suggested that a decimation in time, decimation in frequency or butterfly type technique may be used to take the L_W inputs for each window and generate the N frequency outputs per window. In an FFT implementation, there are as many inputs as there are outputs, while in the WBGT there are only N outputs per L_W input samples. As a result, a decimation in time technique would require that a combination process other than that used for the FFT be used. This may be a topic suitable for future research.

The original contribution of this thesis is the development of the WBGT and the TWBGT both of which were based on the OLGT. From this, the WBGT was implemented in filterbank structures with the intention of employing them in SAR cross range processing.

Bibliography

[1] Curlander, J.C. and R. McDonough. Synthetic Aperture Radar: Systems and Signal Processing. New York: Wiley and Sons, 1991.

[2] Goodman, J.W. Introduction to Fourier Optics. New York: McGraw Hill, 1968.

[3] Damini, A. The CSAR Signal Processor: Design and Initial Tests. Ottawa: DREO Report #1200, Dec 1993.

[4] Gabor, D. "Theory of Communication", J. Inst. Elec Eng. vol. 93, pp. 429 - 457, Nov 1946.

[5] Doubrava, C. " A New Algorithm for DSP Applications: Better Joint Time-Frequency Analysis". DSP Applications, Fall 1992.

[6] Wexler, J. and Raz, S."Discrete Gabor Expansions," Signal Processing, Vol 21, No. 3, Nov 1990, pp. 207 - 221.

[7] Qian, S. and D. Chen, "Orthogonal-Like Discrete Gabor Expansion", Proceedings of the 26th Conference on Informations and Sciences and Systems, Princeton Univ. March 18, 1992.

[8] Balanis, C.A. Antenna Theory: Analysis and Design. New York:Wiley, 1982.

[9] Vant, M.R. and Haslam, G.E. A Theory of 'Squinted Synthetic Aperture Radar. Ottawa: DREO publication, 1980.

[10] Skolnik, M.I. Introduction to Radar Systems. New York: McGraw Hill Publishing Company, 1980.

[11] Hovanessian, S.A. Introduction to Synthetic Array and Imaging Radars. Dedham MA.: Artech House, 1980.

[12] Hord, M.R. Remote Sensing: Methods and Applications. New York: John Wiley and Sons, 1986.

[13] Vaidyanathan, P.P. Multirate Systems and Filter Banks. Englewood Cliffs, N.J.: Prentice Hall, 1993.

[14] Allen, J.B. "Short Term Spectral Analysis, Synthesis and Modification by Discrete Fourier Transform", IEEE Transactions on Acoustic, Speech and Signal Processing, vol. ASSP-25, June 1977, pp 235 - 240.

[15] Cooley, J.W. and J.W. Tukey "An Algorithm for the machine calculation of the complex Fourier Series," Math Computation, 19(90), pp.297-301, 1965.

[16] Bastiaans, M.J. "Gabor Expansion of a Signal into Gaussian Elementary Signals," Proceedings IEEE, vol. 68, No. 4, April, 1980, pp. 538 - 539.

[17] Qian, S. and D. Chen, "Optimal Biorthogonal Functions for Finite Discrete-Time Gabor Expansion," Signal Processing, vol. 27, No.2, May 1992, pp.177-185.

[18] Oppenheim, A.V. and R.W. Schafer. Discrete Time Signal Processing. Engelwood Cliffs N.J.: Prentice Hall, 1989.

[19] Stimson, G.W. Introduction to Airborne Radar. El Segundo: Hughes Aircraft Company, 1983.

[20] Hlawtsch, F. and G.F. Boudreaux - Bartels, "Linear and Quadratic Time Frequency Signal Representations, " IEEE Signal Processing Magazine, pp.21-67, April 1992

[21] Auslander, L., I. Gertner and R. Tolimieri, "The discrete Zak Transform application time-frequecy analysis and synthesis of non-stationary signals", IEEE Trans. Signal Process., Vol. 39, No. 4, 1991, pp.825-835.

[22] Cimino et al. The Shuttle Imaging Radar B (SIR-B) Experimental Report. Pasadena CA: NASA Jet Propulsion Laboratory, March 15 1988.

[23] Wehner, Donald, R. High Resolution Radar. Norwood MA.: Arthech House, 1987.

[24] Parhi, K et al. "VLSI for Signal Processing", Electrical Engineering Handbook. Ann Arbor, CRC Press, 1993.

[25] The TTL Data Book. Vol 3. Texas Instruments, 1984.

[26] i860™ XP Microprcessor Data Book. Intel Corporation, Mt. Prospect IL, 1993.

[27] i860™ XP Microprcessor Family Programmer's Reference Manual. Intel Corporation, Mt. Prospect IL, 1993.

[28] Elliot, D.K and K. R. Rao Fast Transforms: Algorithms, Analyses, Applications, Academic Press, New York, 1982.

[29] Friedlander, B. and Porat, B. "Detection of Transient Signals by the Gabor Representation, " IEEE Trans. Acoust, Speech , Sig. Proccessing, vol 37

[30] Harger, R.O. Syntheitc Aperture Radar Systems. New York: Academic Press, 1970.

[31] Haykin, S. Adaptive Filter Theory. Englewood Cliffs N.J.:Prentice Hall, 1991.

[32] Klepko, R. and Tran, K. Testing Various Time Frequency Distribution Techniques. Ottawa: DREO TN 92-17, Nov 1992.

[33] Leon-Garcia, Alberto. Probability and Random Processes for Electrical Engineering. Reading MA.: Addison - Wesley Publishing Company, 1989.

[34] Munsen, D.C. and Visentin R.L. "A Signal Processing View of Stripmapping Synthetic Aperture Radar," IEEE Transactions Acoustic, Speech and Signal Processing, vol 37, pp 2131 -2142, Dec 89.

[35] Orr, R. "Computational assessment of the Gabor representations", Proc. IEEE International Conference an Acoustic Speech and Signal Processing, 91, Toronto, Ontario, May 1991, pp. 2217-2220.

[36] Rioul, O. and Vetterli, M. "Wavelets and Signal Processing," IEEE SP Magazine, Oct, 1991, pp. 14 - 38.

[37] Schafer, R.W. and Rabiner, L.R. "Design and simulation of a speech analysis-synthesis system based on short-time Fourier Analysis, " IEEE Trans. Audio Electroacoust., vol AU-21, pp.165-174, June 1973.

[38] Srinivasan, V., P.Bhatia and S.H. Ong. "A fast implementation of the discrete 2-D Gabor Transform," Signal Processing Vol 31, No. 2, pp. 229-233, March 1993.

[39] Vetterli, M. "Filterbanks Allowing Perfect Reconstruction," Signal Processing, Vol. 37, No 7, pp. 1057-1071, July 1989.

[40] Vetterli, M. and D. Le Gall, "Perfect Reconstruction FIR Filterbanks: Some Properties and Factorizations, IEEE Trans. on Acoust., Speech, Signal Proc., pp1723-1726, Apr. 3-6, 1990.

Appendix A: Results from Simulations using Various Waveforms

This appendix includes analysis of three waveforms using the WBGT. The first is the pulse with noise which was analysed in chapter 4 and is included in this appendix for reference. A sine wave with a delta function is then analysed to show the the WBGT provides good resolution in both the time and frequency domains. Finally a random noise waveform is also analysed.

Analysis of a Pulse with Noise

The analysis of the pulse with noise is presented below. The signal is shown in figure A-1 and its JTF spectrum after applying the WBGT is shown in A-2.

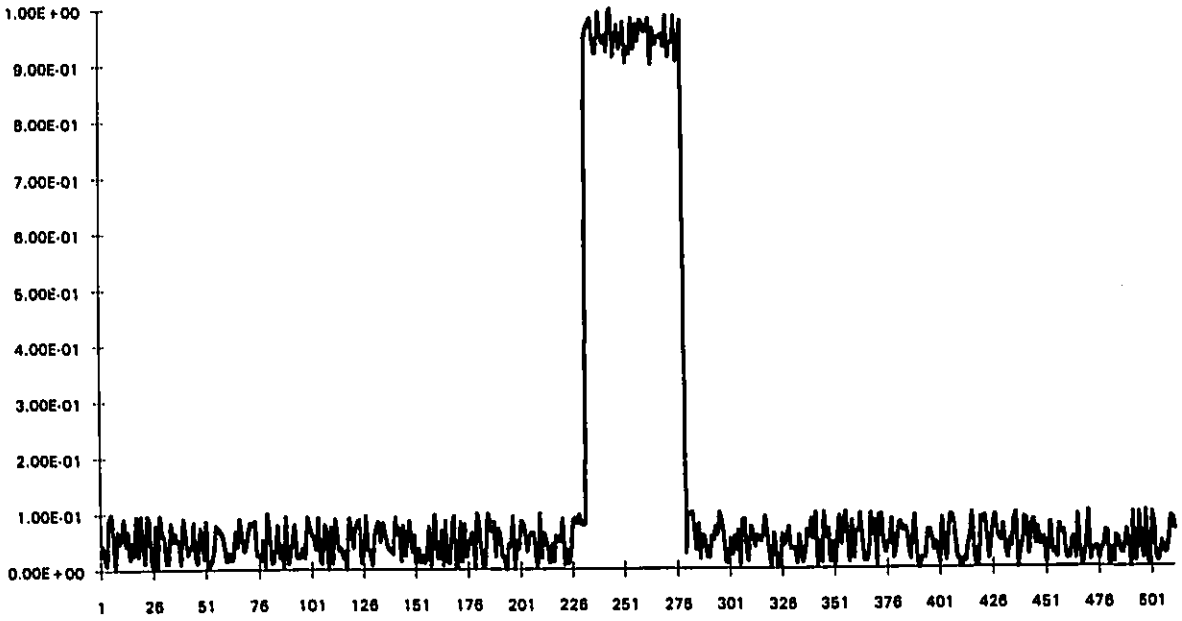

Figure A-1: Pulse waverform with noise

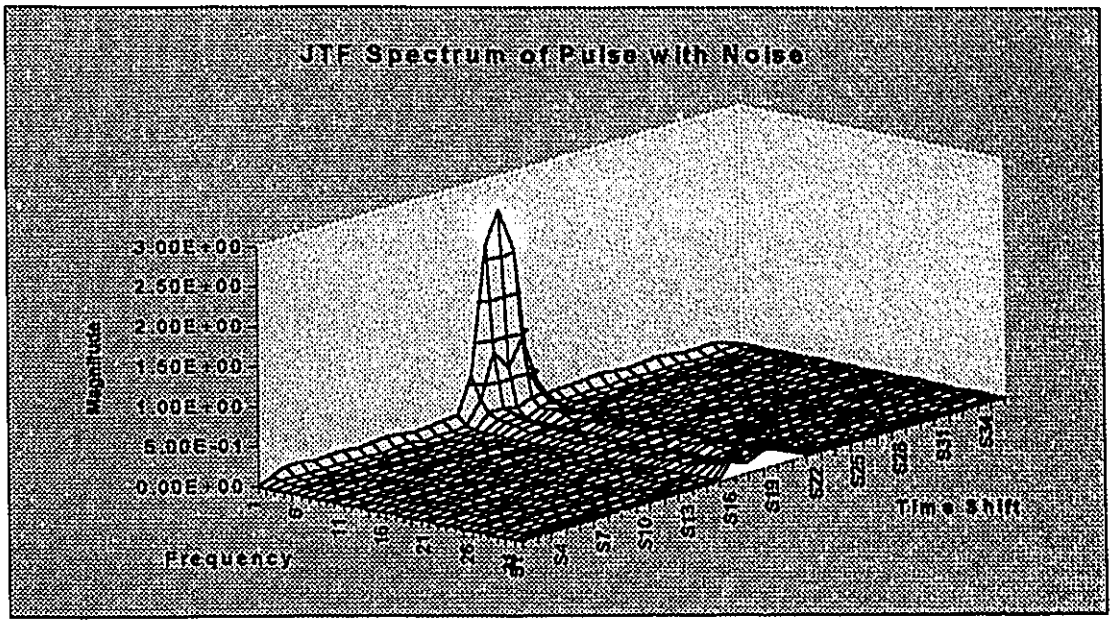

Figure A-2: JTF Spectrum of a Pulse Wavefrom with Noise

Simu-lation	M	N	L_W	ΔM	overlap $1-\frac{\Delta M}{L_W}$	ξ	M*N	err_{av}	err_{db}
#1	71	16	64	8	.875	2	1136	0.1102	-19.15
#2	35	32		16	.75	2	1120	0.1347	-17.41
#3	17	64		32	.5	2	1088	1.09E-14	-279.25
#4	71	24		8	.875	3	1706	0.0259	-31.73
#5	35	48		16	.750	3	1680	0.0534	-25.49
#6	35	64		16	.75	4	2240	8.57E-15	-281.34

Table A-1: Results for WBGT simulations using pulse waveform

The results presented in table A-1 are identical to those of table 4-3 and are provided as reference in this appendix.

Analysis of Sine Wave with Delta Function

The analysis of a sine wave with a delta function is presented below. The signal is shown in figure A-3 and its JTF spectrum after applying the WBGT is shown in A-4.

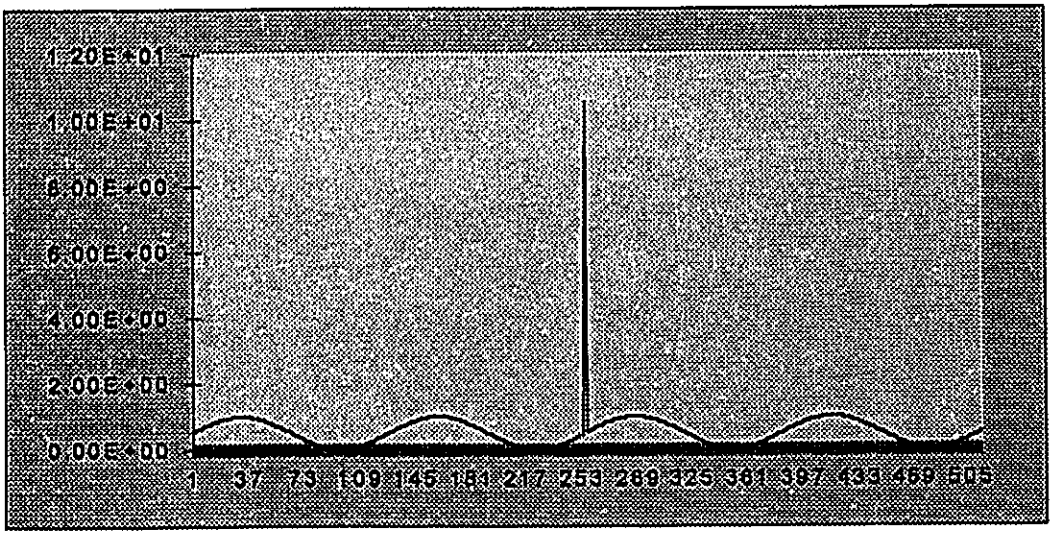

Figure A-3: A sine wave with a delta function

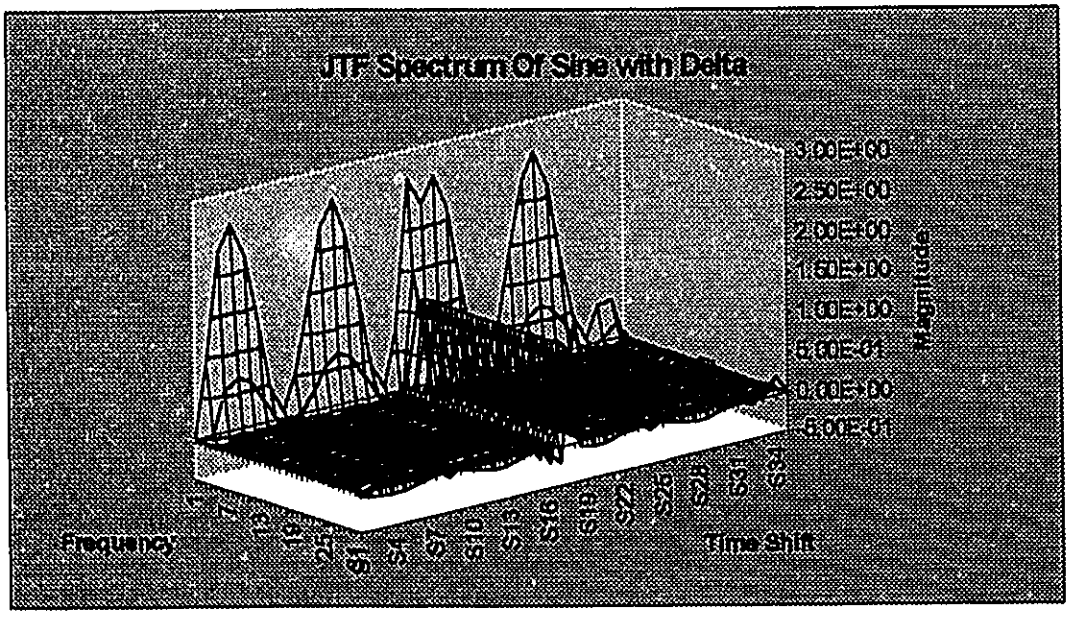

Figure A-4: JTF spectrum of a sine wave with a delta function

Simu-lation	M	N	L_W	ΔM	overlap $1-\frac{\Delta M}{L_W}$	ξ	M*N	err_{av}	err_{db}
#1	71	16	64	8	.875	2	1136	0.1289	-17.79
#2	35	32		16	.75	2	1120	0.4573	-6.79
#3	17	64		32	.5	2	1088	1.19E-14	-278.49
#4	71	24		8	.875	3	1706	0.0463	-26.68
#5	35	48		16	.750	3	1680	0.1111	-19.09
#6	35	64		16	.75	4	2240	9.2E-15	-280.72

Table A-2: Results for WBGT simulations using a sine wave with delta function

The analysis presented above demonstrates that the WBGT provides good time and frequency domain resolution and similar signal reconstruction errors are achieved to the case of the pulse with noise. The frequency domain spikes in figure A-4 reflect the concentrated frequency content of the sine wave, while the spread frequency domain spectrum in the middle of the JTF plane reflects the spectrum of the dleta function that is concetrated in the middle time frame of the signal. For the case of triple oversampling the reconstruction error was once again about 8 to 12 db lower than double oversampling reflecting an analysis function that is smoother and more gaussian in shape than when using double oversampling.

Analysis of Random Noise Waveform

The analysis of a random noise waveform is presented below. The signal is shown in figure A-5 and its JTF spectrum after applying the WBGT is shown in A-6.

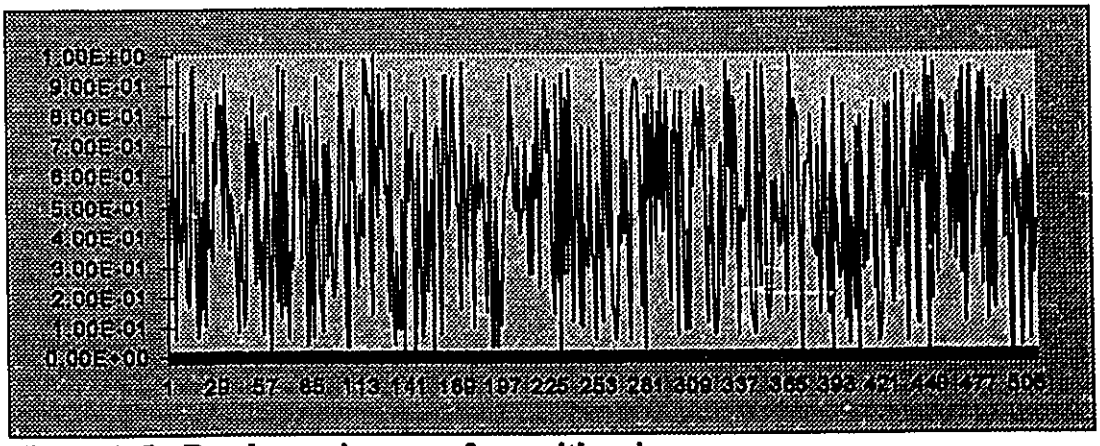

Figure A-5: Random noise waverform with noise

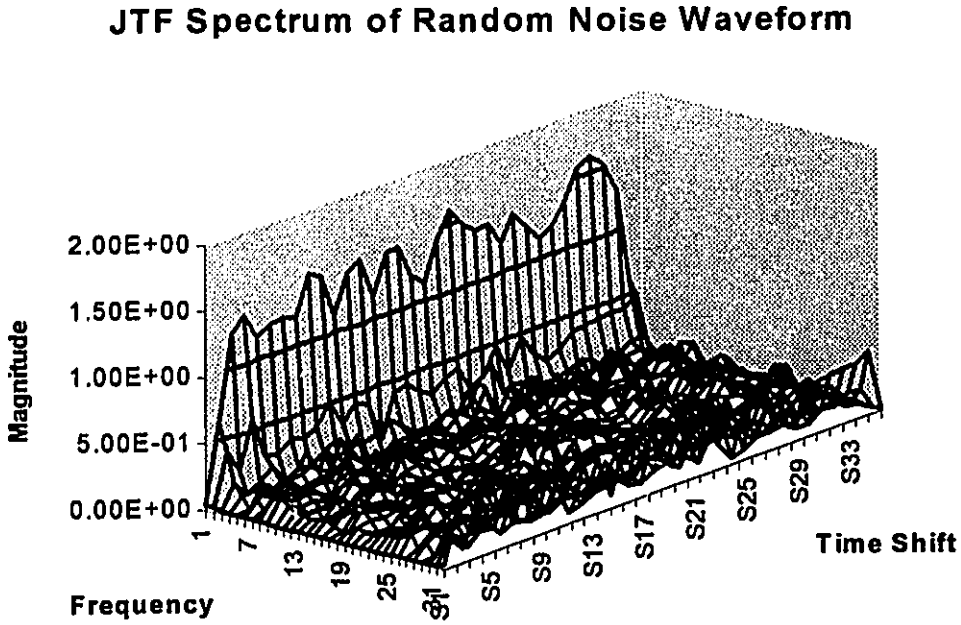

Figure A-6: JTF Spectrum of a Random Noise Waveform

Simu-lation	M	N	L_W	ΔM	overlap $1-\frac{\Delta M}{L_W}$	ξ	M*N	err_{av}	err_{db}
#1	71	16	64	8	.875	2	1136	0.1164	-18.68
#2	35	32		16	.75	2	1120	0.1327	-17.54
#3	17	64		32	.5	2	1088	9.43E-14	-260.51
#4	71	24		8	.875	3	1706	0.0275	-31.21
#5	35	48		16	.750	3	1680	0.0347	-29.19
#6	35	64		16	.75	4	2240	8.43E-15	-281.15

Table 4-3: Results for WBGT simulations using random noise waveform

The results presented above show that the WBGT provides fairly constant reconstruction errors when analysing this waveform compared to the pulse with noise. Once again , triple oversampling results in reconstruction errors that are about 10 db lower than for double oversampling. It is also seen that the JTF spectrum is fairly constant at each time shift, reflecting the nature of the signal.