PROJECT REPORT

ON

SPEECH COMPRESSION USING WAVELETS

DEVELOPED BY

GIRI SHIVRAMAN S.	**GH99316**
NARKHEDE NILESH P.	**GH99340**
PALTERU SWARNALATHA.	**GH99344**
RAMALAKSHMI SUNDARESAN.	**GH99347**
RAJADHYAX AMOL A.	**GH99364**

UNDER THE GUIDANCE OF

Dr. S. C. GADRE

DEPARTMENT OF ELECTRICAL ENGINEERING

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

UNIVERSITY OF MUMBAI

2002-03

Certificate of Approval

This is to certify that the project titled "**Speech Compression using Wavelets**" is a bona fide record of the project work done by

Name	University No.
Palteru Swarnalatha	10178
Ramalakshmi Sundaresan	10180
Giri Shivraman S.	10188
Narkhede Nilesh P.	10203
Rajadhyax Amol A.	10205

during the year 2002-03 under the guidance of Dr. S. C. Gadre, towards the partial fulfillment of Bachelor of Engineering Degree Course in Electrical Engineering of the University of Mumbai.

Dr. B. K. Lande	**Dr. S. C. Gadre**
Head of Department	**Project Guide**
Electrical & Electronics	**Electrical & Electronics**
Engg. Dept.	**Engg. Dept.**
VJTI	**VJTI**

Certificate of Approval by Examiners

This is to certify that the project titled "**Speech Compression using Wavelets**" is a bona fide record of the project work done by

Name	University No.
Palteru Swarnalatha	10178
Ramalakshmi Sundaresan	10180
Giri Shivraman S.	10188
Narkhede Nilesh P.	10203
Rajadhyax Amol A.	10205

during the year 2002-03 and is approved for the degree of Bachelor of Engineering Degree Course in Electrical Engineering of the University of Mumbai.

Examiners

______________________ ______________________

(External) **(Internal)**

Date: **Date:**

PREFACE

Multimedia files in general need plenty of disk space for storage apart from being unwieldy for communication purposes and sound files are no exception. Hence compression of these files has become a necessity and is a ripe subject for research.

The field of signal processing has enjoyed several analysis tools in the past. One of the more recent (and, needless to say, more exciting) developments in this field has been the emergence of a new transform, THE WAVELET TRANSFORM. In fact, its use is not restricted to signal processing alone, but ranges over such diverse fields as image processing, communications, mathematics, computer science to name a few. Wavelet transforms, in their different guises, have come to be accepted as a set of tools useful for various applications. Wavelet transforms are good to have at one's fingertips, along with many other, mostly more traditional, tools.

This thesis considers the application of Wavelet transforms for the compression of human speech signals. It is part of the project on the said topic undertaken by a group of undergrad students. The study was by no means exhaustive, which is beyond the purview of undergrad study.

It can also serve as a guide to understand the working of the software implementation of the project, which is provided in the accompanying CD. Note that you need MATLAB® Version 6 or higher to run the code. Other than this, the software is self-contained in terms of help files and source code, which facilitates alteration of the program to suit your needs.

Chapter 1 is intended to serve as an introduction. Its main purpose is to define the scope of this project.

Chapter 2 presents the drawbacks inherent in the Fourier Methods. It is assumed that the reader is conversant with the Fourier methods. A specific example is taken wherein the shortcomings of Fourier methods become readily evident.

Chapter 3 presents a detailed introduction to the wavelet analysis. It tries to compare Wavelet transform as a signal processing tool in view of its similarities and differences with the Fourier methods. The CONTINUOUS WAVELET TRANSORM (CWT) is discussed next. The chapter concludes by demonstrating how the wavelet transform overcomes the drawbacks of Fourier methods.

Chapter 4 discusses the 'DISCRETE WAVELET TRANSFORM' (DWT), which is a more practical approach than CWT. It also explains an implementation of DWT using filtering schemes.

In chapter 5, the use of Wavelet transform in speech compression is presented. The motivation for using wavelets for speech compression is developed, so is the algorithm used for the same.

The pertinent commands of MATLAB® WAVELET TOOLBOX are explained in brief in chapter 6. Specifically, the commands used for achieving compression are discussed along with their syntax. The software implementation is based on these commands. However, a detailed discussion of software is relegated to the Appendix lest you get inundated with extraneous details of programming.

The conclusion is the subject of chapter 7. Statistical analysis of signals is performed and the results recorded. Based on these, inferences are drawn.

As has been pointed out, this discussion is not comprehensive, to compensate for which plenty of references have been provided at the end. The accompanying CD contains myriad documents that we found on the internet. These should serve as useful guide for anyone interested in pursuing the subject beyond the point where we stopped.

The Appendices are included in the CD

.ACKNOWLEDGEMENTS

A project of this magnitude is unlikely to be completed without the active help of a hidden army of others who make it possible. At the risk of forgetting to mention some names, we would like to thank the following people.

The first and the foremost person who deserves our boundless gratitude is our project guide, Professor Dr. (Ms.) S. C. Gadre. The sheer size of this project was daunting in the beginning and would have deterred us from proceeding had it not been for her support and guidance. We take this opportunity to tender our sincerest thanks to her.

We are also indebted to Mr. Ajay Shah and Mr. Laxman Udgiri who provided us with valuable leads into the subject of wavelet transforms.

Throughout the progress of this project we have enjoyed a close collaboration with a number of people who we feel fortunate to count as seniors and friends, whose views have greatly influenced us. Our deep gratitude to all of them.

Last but not the least, we wish to thank the various internet-communities/e-groups on signal processing/wavelets/MATLAB®/Compression. It was a rich learning experience to have been a part of this wonderful community. Our innumerable thanks to every member of this community.

Table of Contents.

1 INTRODUCTION TO AND SCOPE OF THE PROJECT

1.1 Speech Signals

The human speech in its pristine form is an acoustic signal. For the purpose of communication and storage, it is necessary to convert it into an electrical signal. This is accomplished with the help of certain instruments called ‘transducers’.

This electrical representation of speech has certain properties.

1. It is a one-dimensional signal, with time as its independent variable.
2. It is random in nature.
3. It is non-stationary, i.e. the frequency spectrum is not constant in time.
4. Although human beings have an audible frequency range of 20Hz –20kHz, the human speech has significant frequency components only upto 4kHz,a property that is exploited in the compression of speech.

Digital representation of speech

With the advent of digital computing machines, it was propounded to exploit the powers of the same for processing of speech signals. This required a digital representation of speech. To achieve this, the analog signal is sampled at some frequency and then quantized at discrete levels. Thus, parameters of digital speech are

1. Sampling rate
2. Bits per second
3. Number of channels.

The sound files can be stored and played in digital computers. Various formats have been proposed by different manufacturers for example ‘.wav’ ‘.au’ to name a few.

In this thesis, the ‘.wav’ format is used extensively due to the convenience in recording it with ‘Sound recorder’ software, shipped with WINDOWS OS.

1.2 Compression – An Overview

In the recent years, large scale information transfer by remote computing and the development of massive storage and retrieval systems have witnessed a tremendous growth. To cope up with the growth in the size of databases, additional storage devices need to be installed and the modems and multiplexers have to be continuously upgraded in order to permit large amounts of data transfer between computers and remote terminals. This leads to an increase in the cost as well as equipment. One solution to these problems is-"COMPRESSION" where the database and the transmission sequence can be encoded efficiently.

WHY COMPRESSION?

Compression is a process of converting an input data stream into another data stream that has a smaller size. Compression is possible only because data is normally represented in the computer in a format that is longer than necessary i.e. the input data has some amount of redundancy associated with it. The main objective of compression systems is to eliminate this redundancy.

When compression is used to reduce storage requirements, overall program execution time may be reduced. This is because reduction in storage will result in the reduction of disc access attempts.

With respect to transmission of data, the data rate is reduced at the source by the compressor (coder) ,it is then passed through the communication channel and returned to the original rate by the expander(decoder) at the receiving end. The compression algorithms help to reduce the bandwidth requirements and also provide a level of security for the data being transmitted. A tandem pair of coder and decoder is usually referred to as codec.

APPLICATIONS OF COMPRESSION

1. The use of compression in recording applications is extremely powerful. The playing time of the medium is extended in proportion to the compression factor.
2. In the case of tapes, the access time is improved because the length of the tape needed for a given recording is reduced and so it can be rewound more quickly.
3. In digital audio broadcasting and in digital television transmission, compression is used to reduced the bandwidth needed.
4. The time required for a web page to be displayed and the downloading time in case of files is greatly reduced due to compression.

COMPRESSION TERMINOLOGY

- Compression ratio:- The compression ratio is defined as:-

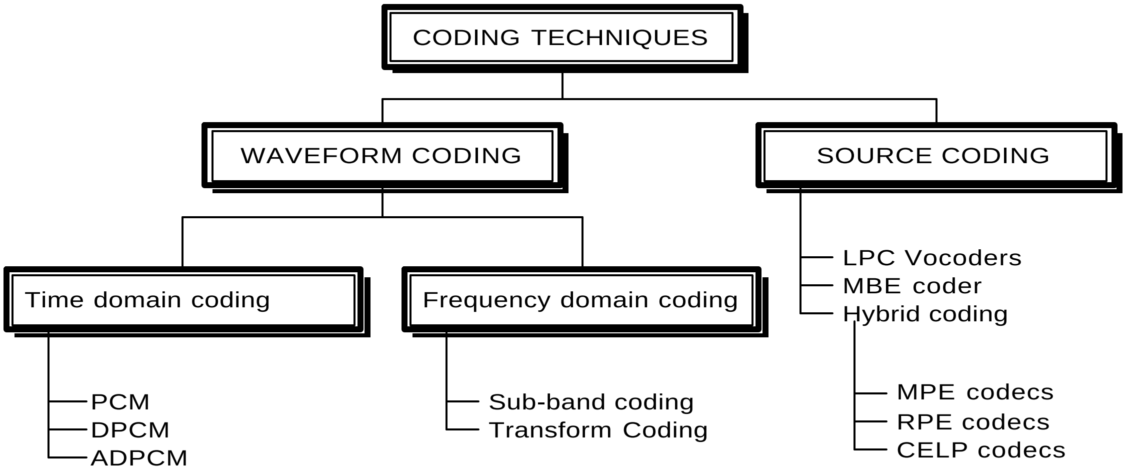
Compression ratio = size of the output stream/size of the input stream

A value of 0.6 means that the data occupies 60% of its original size after compression. Values greater than 1 mean an output stream bigger than the input stream. The compression ratio can also be called bpb(bit per bit),since it equals the no. of bits in the compressed stream needed, on an average, to compress one bit in the input stream.

- Compression factor:- It is the inverse of compression ratio. Values greater than 1 indicate compression and less than 1 indicates expansion.

1.3 Coding Techniques

There are various methods of coding the speech signal



1.4 Aim, Scope And Limitations of This Thesis

The primary objective of this thesis is to present the wavelet based method for the compression of speech. The algorithm presented here was implemented in MATLAB®. The said software is provided in the accompanying CD. Readers may find it useful to verify the result by running the program.

Since this thesis is an application of wavelets, it was natural to study the basics of wavelets in detail. The same procedure was adopted in writing this thesis, as it was felt that without minimal background in wavelets, it would be fruitless, and also inconvenient to explain the algorithm.

However, the wavelet itself is an engrossing field, and a comprehensive study was beyond the scope of our undergraduate level. Hence, attempt is made only to explain the very basics which are indispensable from the compression point of view. This approach led to the elimination of many of the mammoth sized equations and vector analysis inherent in the study of wavelets.

At this stage, it is worthwhile mentioning two quotes by famous scientists
'*So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.*' --Albert Einstein

'*As complexity rises, precise statements lose meaning and meaningful statements lose precision.*' --Lotfi Zadeh [1]

The inclusion of the above quotes is to highlight the fact that simplicity and clarity are often the casualties of precision and accuracy, and vice-versa.

In this thesis, we have compromised on the mathematical precision and accuracy to make matters simple and clear. An amateur in the field of wavelets might find this work useful as it is relieved of most of the intimidating vector analysis and equations, which have been supplanted by simple diagrams. However, for our own understanding, we did found it necessary, interesting and exciting to go through some literature which deal with the intricate details of wavelet analysis, and sufficient references have been provided wherever necessary, for the sake of a fairly advanced reader. Some of the literature that we perused has been included in the CD.

[1] Lotfi Zadeh is considered to be the father of Fuzzy Logic

The analysis that we undertook for wavelets includes only the orthogonal wavelets. This decision was based on the extensive literature we read on the topic, wherein the suitability of these wavelets for speech signals was stated.

Another topic that has been deliberately excluded in this work is the concept of MRA, which bridges the gap between the wavelets and the filter banks and is indispensable for a good understanding of Mallat's Fast Wavelet Transform Algorithm. Instead, we have assumed certain results and provided references for further reading.

Secondly, the sound files that we tested were of limited duration, around 5 seconds. Albeit the programs will run for larger files (of course, the computation time will be longer in this case), a better approach towards such large files is to use frames of finite length. This procedure is more used in real-time compression of sound files, and is not presented here.

Encoding is performed using only the Run Length Encoding. The effect of other encoding schemes on the compression factor have not been studied.

This thesis considers only wavelets analysis, wherein only approximation coefficients are split. There exists another analysis, called wavelet packet analysis, which splits detail coefficients. This is not explored in this thesis.

2 WEAKNESSES OF FOURIER ANALYSIS

Introduction

This chapter develops the need and motivation for studying the wavelet transform. Historically, Fourier Transform has been the most widely used tool for signal processing. As signal processing began spreading its tentacles and encompassing newer signals, Fourier Transform was found to be unable to satisfy the growing need for processing a bulk of signals. Hence, this chapter begins with a review of Fourier Methods. Detailed explanation is avoided to rid the discussion of insignificant details. A simple case is presented, where the shortcomings of Fourier methods is expounded. The next chapter concerns wavelet transforms, and shows how the drawback of FT are eliminated.

2.1 Review of Fourier Methods

For a continuous –time signal x(t) , the Fourier Transform (FT) equations are

$$X(f) = \int_{-\infty}^{\infty} x(t) \bullet e^{-2j\pi ft} dt. \qquad \text{......... ... 2.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \bullet e^{2j\pi ft} df. \qquad \text{............2.2}$$

Equation (2.1) is the analysis equation and equation (2.2) is the synthesis equation.

The synthesis equation suggests that the FT expresses the signal in terms of linear combination of complex exponential signal. For a real signal, it can be shown that the FT synthesis equation expresses the signal in terms of linear combination of sine and cosine terms. A diagrammatic representation of this is as follows:

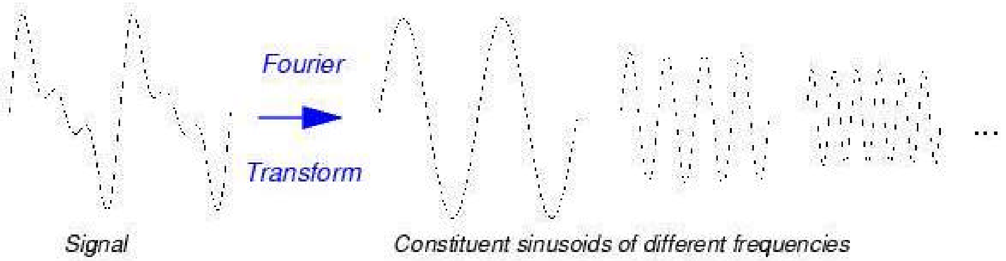


fig 2.1: A signal shown as a linear combination of sinusoids (FT method)

The analysis equation represents the given signal in a different form; as a function of frequency. The original signal is a function of time, whereas the after the transformation, the same signal is represented as a function of frequency. It gives the frequency components in the signal.

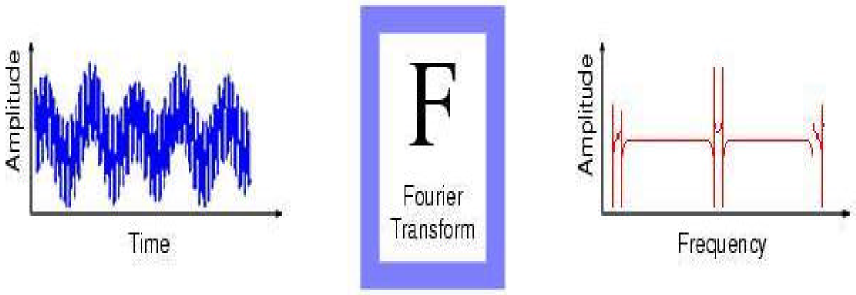


fig 2.2:Transforming a signal from time-domain to frequency-domain, the FOURIER METHOD

Thus the FT is a very useful tool as it gives the frequency content of the input signal. It however suffers from a serious drawback. It is explained through an example in the sequel.

2.2 Shortcomings of FT

EXAMPLE 2.1: Consider the following 2 signals

$x1(t) = \sin(2*\pi*100*t) \qquad 0 <= t < 0.1$ sec

$\quad = \sin(2*\pi*500*t) \qquad 0.1 <= t < 0.2$ sec

$x2(t) = \sin(2*\pi*500*t) \qquad 0 <= t < 0.1$ sec

$\quad = \sin(2*\pi*100*t) \qquad 0.1 <= t < 0.2$ sec

A plot of these signals is shown below.

(Note: A time interval of 0 to 0.2 seconds was divided into 10,000 points. The sine of each point was computed and plotted. Since the signal is of 10,000 points, 16,384 point FFT was computed which represents the frequency domain of the signal. This was done in MATLAB®)

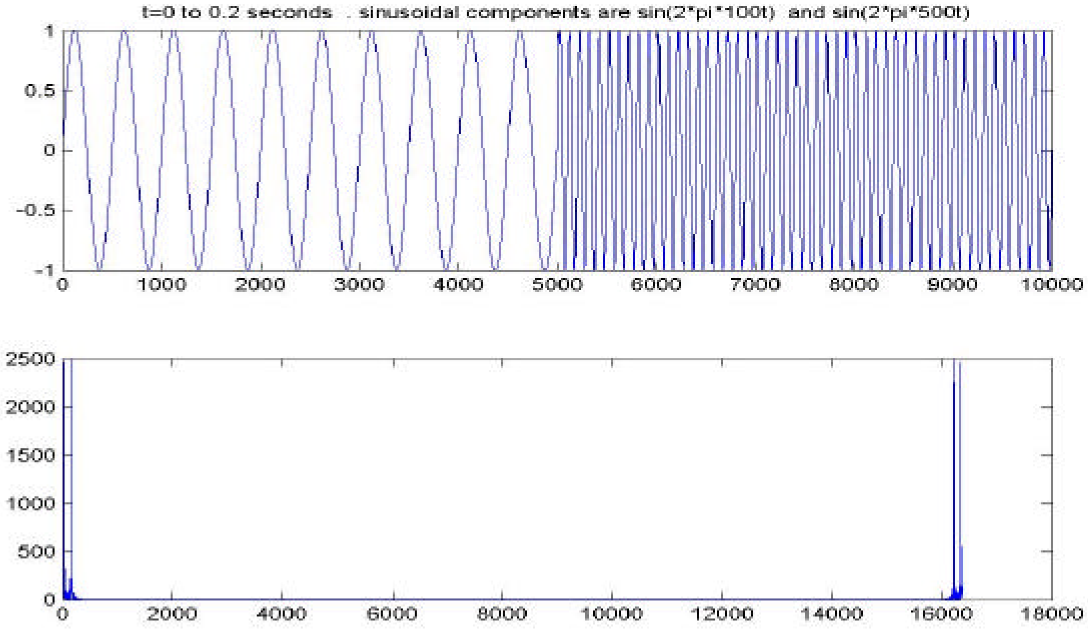


fig 2.3: signal x1(t) and its FFT

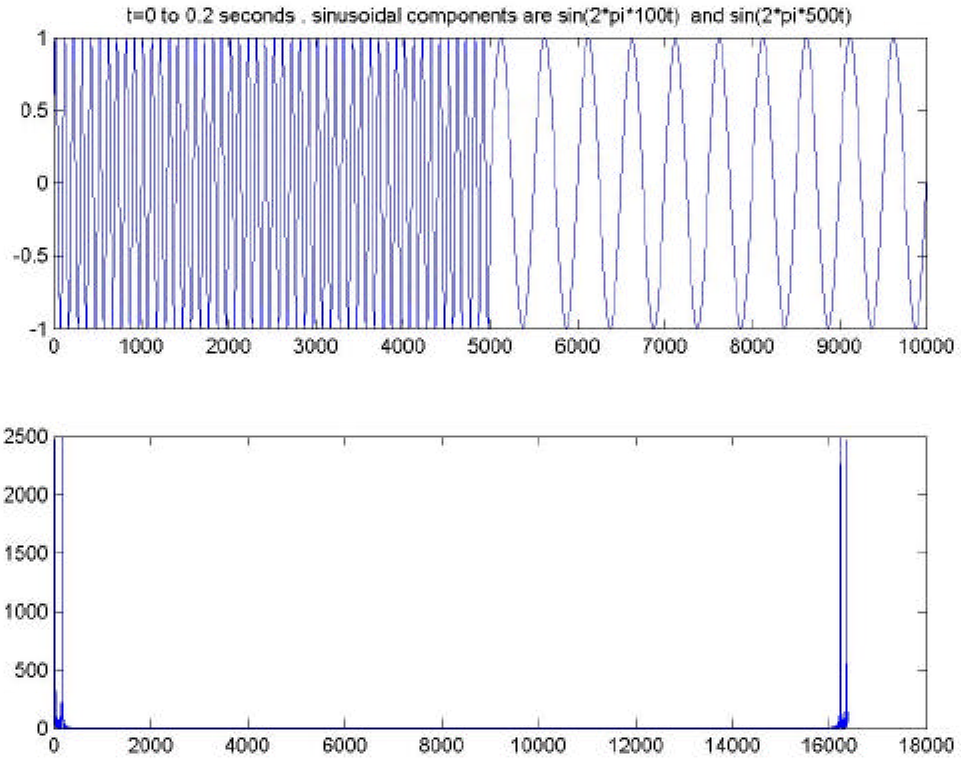


fig 2.4 : Signal x2(t) and its FFT

The above example demonstrates the drawback inherent in the Fourier analysis of signals. It shows that the FT is unable to distinguish between two different signals. The two signals have same frequency components, but at different times**.**

Thus, the FT is incapable of giving time information of signals.

In general, FT is not suitable for the analysis of a class of signals called NON-**STATIONARY SIGNALS**.

This led to the search of new tools for analysis of signals. One such tool that was proposed was the SHORT TIME FOURIER TRANSFORM (STFT). This STFT too suffered from a drawback1 and was supplanted by WAVELET TRANSFORM.

In the sequel, CONTINUOUS WAVELET TRANSFORM is introduced, and the same problem is solved with the help of this transform.

[1] *see the tutorials on 'WAVELET TRANSFORMS' by ROBI POLIKAR for a detailed discussion on this.*

3 INTRODUCTION TO WAVELETS AND THE CONTINUOUS WAVELET TRANSFORM (CWT)

INTRODUCTION:

This chapter provides a motivation towards the study of wavelets as a tool for signal processing. The drawbacks inherent in the Fourier methods are overcome with wavelets. This fact is demonstrated here.

It must be reiterated that the discussion in this chapter is by no means comprehensive and exhaustive. The concepts of time-frequency resolution have been avoided for the sake of simplicity. Instead, the development endeavors to compare the Wavelet methods with the Fourier methods as the reader is expected to be well conversant with the latter.

3.1 Continuous-time Wavelets

Consider a real or complex-valued continuous-time function ψ(t) with the following properties 1

1. The function integrates to zero

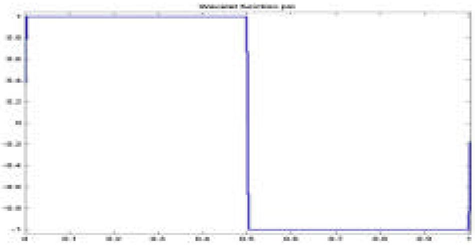
i. $$\int_{-\infty}^{\infty} y\,(t).d(t) = 0 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (3.1)$$

2. It is square integrable or, equivalently, has finite energy:

[1] A third condition, called admissibility condition also exists. For a detailed study of this topic, the reader is referred to the book by Rao (see references, section I, # 10)

$$\int_{-\infty}^{\infty} | y\,(t)\,|^2\ .d\,(t) < \infty \quad \text{………………..……..(3.2)}$$

A function is called mother wavelet if it satisfies these two properties. There is an infinity of functions that satisfy these properties and thus qualify to be mother wavelet. The simplest of them is the 'Haar wavelet'. Some other wavelets are Mexican hat, Morlet. Apart from this, there are various families of wavelets. Some of the families are daubechies family, symlet family, coiflet family etc. In this thesis, the main stress is given on the Daubechies family, which has db1 to db10 wavelets. They are shown in the following figure[1] .



Haar wavelet

[1] db1 is same as haar wavelet

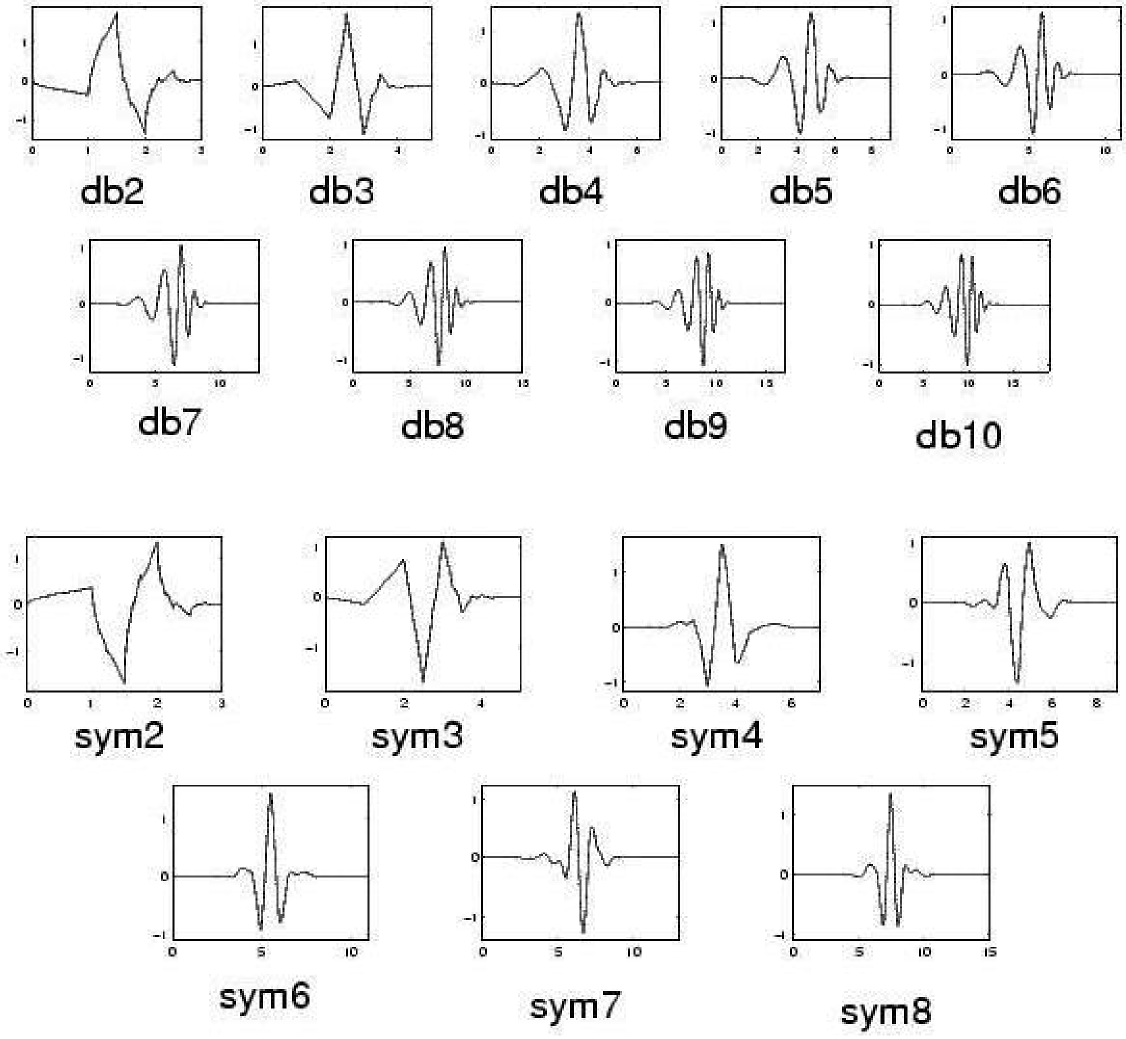


fig 3.1 : Some wavelet functions.

3.2 The Continuous Wavelet Transform (CWT)

Consider the following figure which juxtaposes a sinusoid and a wavelet

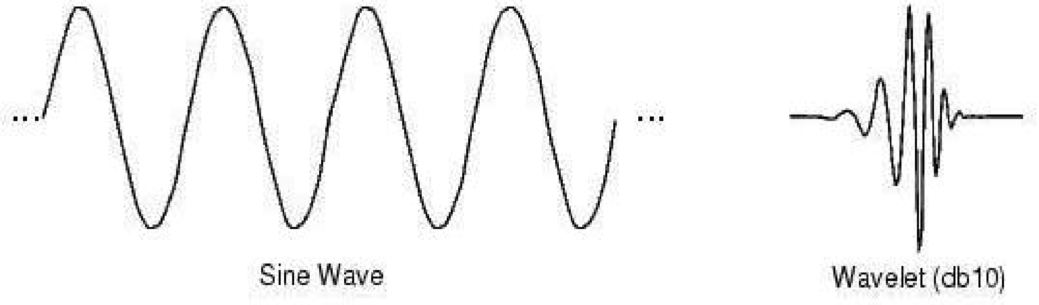


fig 3.2 : comparing sine wave and a wavelet

As has already been pointed out, wavelet is a waveform of effectively limited duration that has an average value of zero.

Compare wavelets with sine waves, which are the basis of Fourier analysis.

Sinusoids do not have limited duration -- they extend from minus to plus infinity. And where sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric.

Fourier analysis consists of breaking up a signal into sine waves of various Frequencies (fig 2.1). Similarly, wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet. Compare the following figure with fig :2.1 .

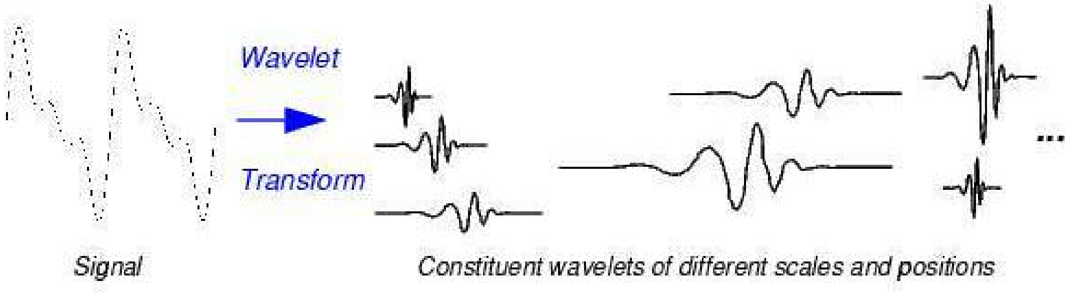


fig 3.3 :figure demonstrating the decomposition of a signal into wavelets

The above diagram suggests the existence of a synthesis equation to represent the original signal as a linear combination of wavelets which are the basis function for wavelet analysis (recollect that in Fourier analysis, the basis functions are sines and cosines). This is indeed the case. The wavelets in the synthesis equation are multiplied by scalars. To obtain these scalars, we need an analysis equation, just as in the Fourier case.

We thus have two equations, the analysis and the synthesis equation. They are stated as follows:

1. Analysis equation or CWT equation:[1]

$$C(a,b)=\int_{-\infty}^{\infty} f(t)\,.\frac{1}{\sqrt{|a|}}\;y^{*}\left(\frac{t-b}{a}\right).d(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.3)$$

2. Synthesis equation or ICWT:

[1] The ‘*’ indicates complex conjugate.

$$f(t)=\frac{1}{K}\int_{a=-\infty}^{\infty}\int_{b=-\infty}^{\infty}\frac{1}{|a|^{2}}C(a,b)\frac{1}{\sqrt{|a|}}y\left(\frac{t-b}{a}\right)d(a).\ d(b) \quad \text{…………(3.4)}$$

K is a constant; it depends on the wavelet

The basis functions in both Fourier and wavelet analysis are localized in frequency making mathematical tools such as power spectra (power in a frequency interval) useful at picking out frequencies and calculating power distributions.

The most important difference between these two kinds of transforms is that individual wavelet functions are localized in space. In contrast Fourier sine and cosine functions are non-local and are active for all time t.

This localization feature, along with wavelets localization of frequency, makes many functions and operators using wavelets “sparse”, when transformed into the wavelet domain. This sparseness, in turn results in a number of useful applications such as data **compression, detecting features in images and de-noising signals.**

Returning to the equations

The quantities ‘a’ and ‘b’ appearing in the above equations represent respectively the scale and shift of mother wavelet.

The wavelet transform of a signal f(t) is the family C(a,b), given by the analysis equation. It depends on two indices a and b. From an intuitive point of view, the wavelet decomposition consists of calculating a "resemblance index" between the signal and the wavelet located at position b and of scale a. If the index is large, the resemblance is strong, otherwise it is slight. The indexes C(a,b) are called coefficients**. The dependence of these coefficients on both ‘a’ and ‘b’ is responsible for the wavelet transform**

preserving time and frequency information. These quantities are explained in the following sections.

3.3 The Scale 'a'

Simply put 'Scaling a wavelet means stretching (or compressing) it ' .To go beyond colloquial descriptions such as "stretching," we introduce the scale factor, often denoted by the letter 'a'. If we're talking about sinusoids, for example, the effect of the scale factor is very easy to see:

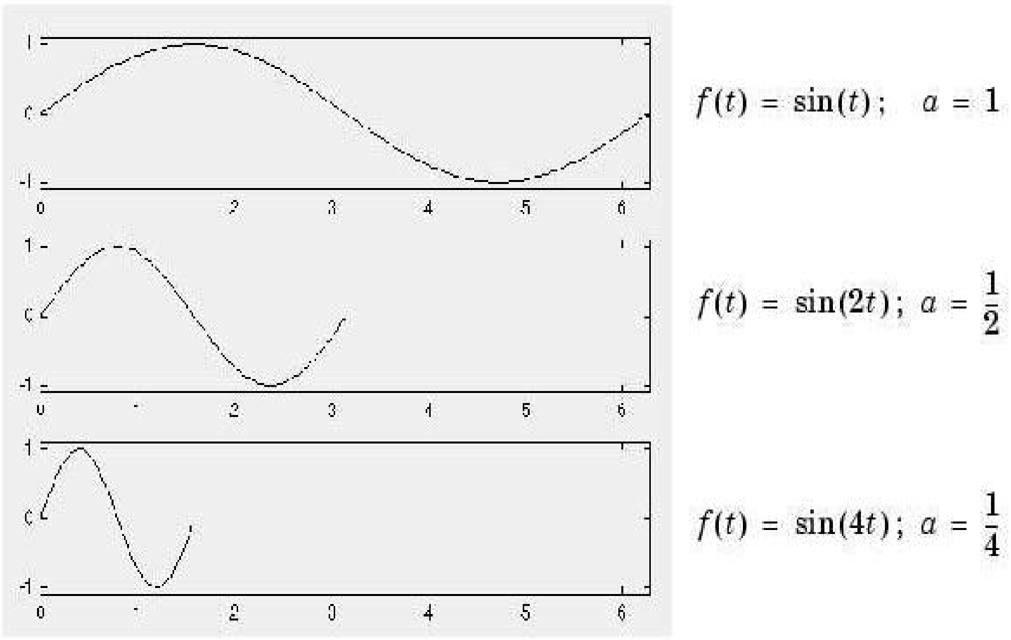


fig 3.4 : Effect of scaling on sine waves

The scale factor works exactly the same with wavelets. The smaller the scale factor, the more "compressed" the wavelet and vice versa.

(see fig 3.5)

It is clear from the diagrams that, for a sinusoid sin(ωt), the scale factor is related (inversely) to the radian frequency ω . Similarly, with wavelet analysis, the scale is related to the frequency of the signal.

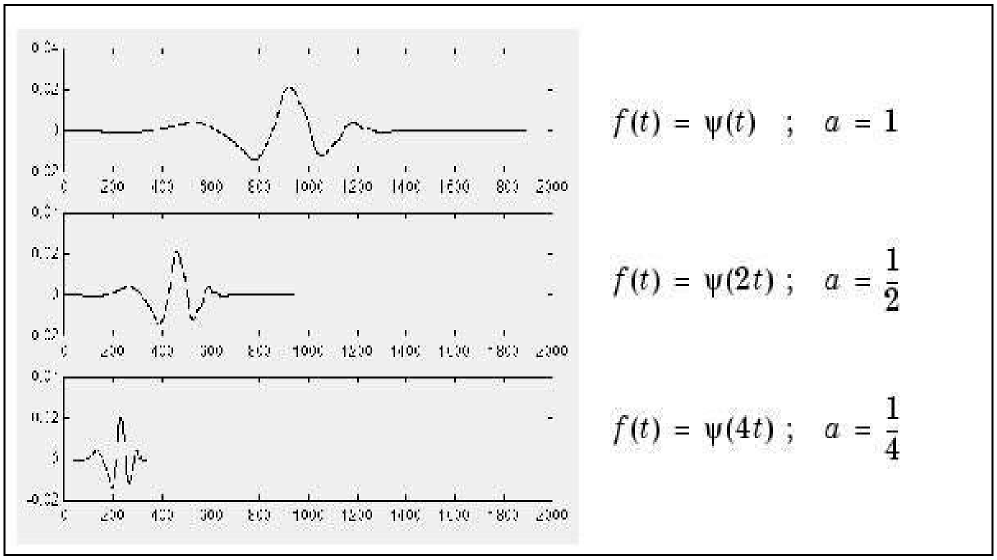


fig 3.5: Effect of scaling on wavelets

Thus the higher scales correspond to the most "stretched" wavelets. The more stretched the wavelet, the longer the portion of the signal with which it is being compared, and thus the coarser the signal features being measured by the wavelet coefficients.

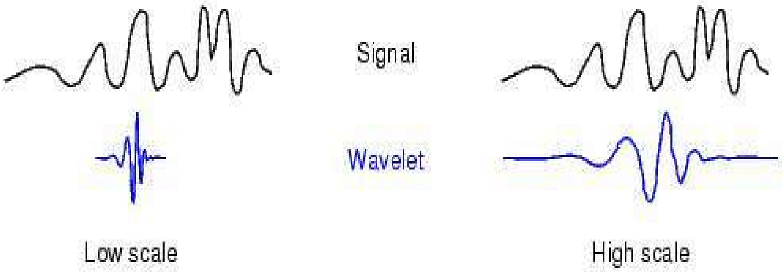


fig 3.6 : Figure demonstrating the effect of stretching the wavelet on the length of the signal being compared

Thus, there is a correspondence between wavelet scales and frequency as revealed by wavelet analysis:

Low scale 'a' =>Compressed wavelet =>Rapidly changing details =>High Frequency (ω)
.

High scale 'a' =>Stretched wavelet =>Slowly changing, coarse features=>low freq (ω)

The exact relation between frequency and scale is given in section 3.5

3.4 Shift 'b'

Shifting a wavelet simply means delaying (or hastening) its onset. Mathematically, delaying a function f(t) by 'b' is represented by f(t-b) :

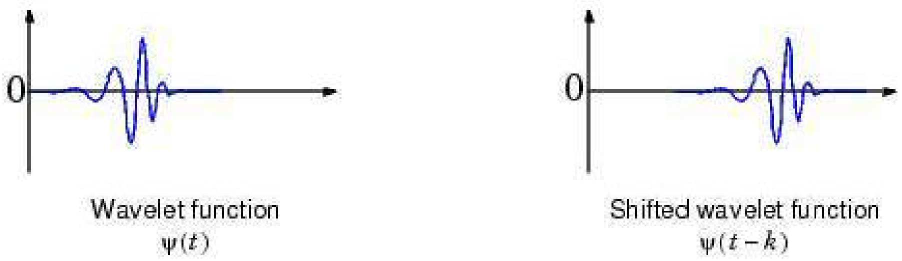


fig 3.7 : Shifting a wavelet

3.5 Five Easy Steps to a Continuous Wavelet Transform

The continuous wavelet transform is the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet. This process produces wavelet coefficients that are a function of scale and position.

It's really a very simple process. In fact, here are the five steps of an easy recipe for creating a CWT:

1. Take a wavelet and compare it to a section at the start of the original signal.

2. Calculate a number, C, that represents how closely correlated the wavelet is with this section of the signal. The higher C is, the more the similarity. More precisely, if the signal energy and the wavelet energy are equal to one, C may be interpreted as a correlation coefficient.

Note that the results will depend on the shape of the wavelet you choose.

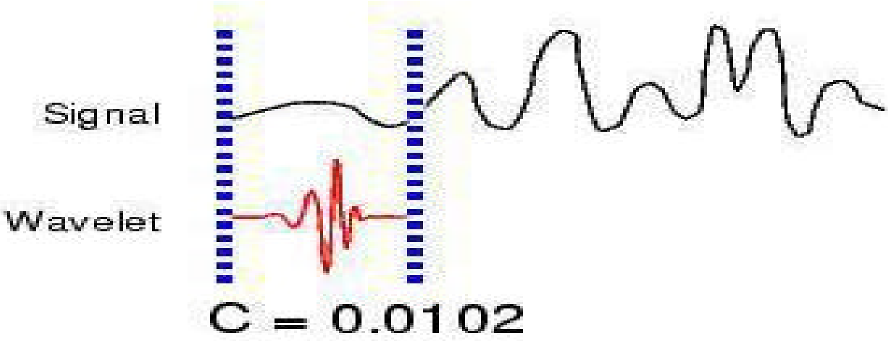


fig 3.8 : Step #2 for calculating CWT

3. Shift the wavelet to the right and repeat steps 1 and 2 until you've covered the whole signal.

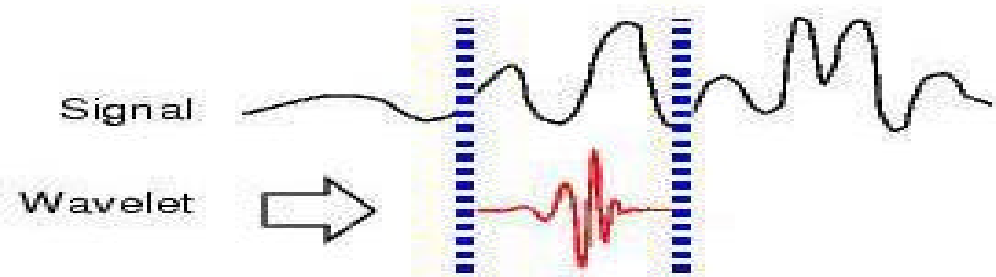


fig 3.9: Step #3 for calculating CWT

4. Scale (stretch) the wavelet and repeat steps 1 through 3.

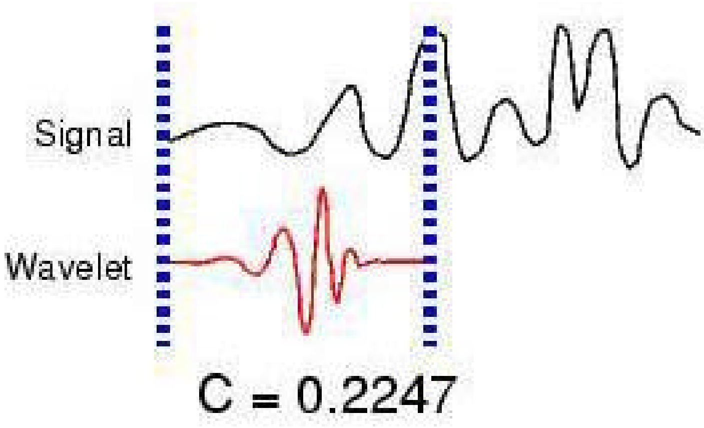


fig 3.10 : Step # 4 for calculating CWT

5. Repeat steps 1 through 4 for all scales.

When you're done, you'll have the coefficients produced at different scales by different sections of the signal. The coefficients constitute the results of a regression of the original signal performed on the wavelets.

How to make sense of all these coefficients? You could make a plot on which the x-axis represents position along the signal (time), the y-axis represents scale, and the color at each x-y point represents the magnitude of the wavelet coefficient C. An example is shown below (black represents low magnitude and white is high magnitude)

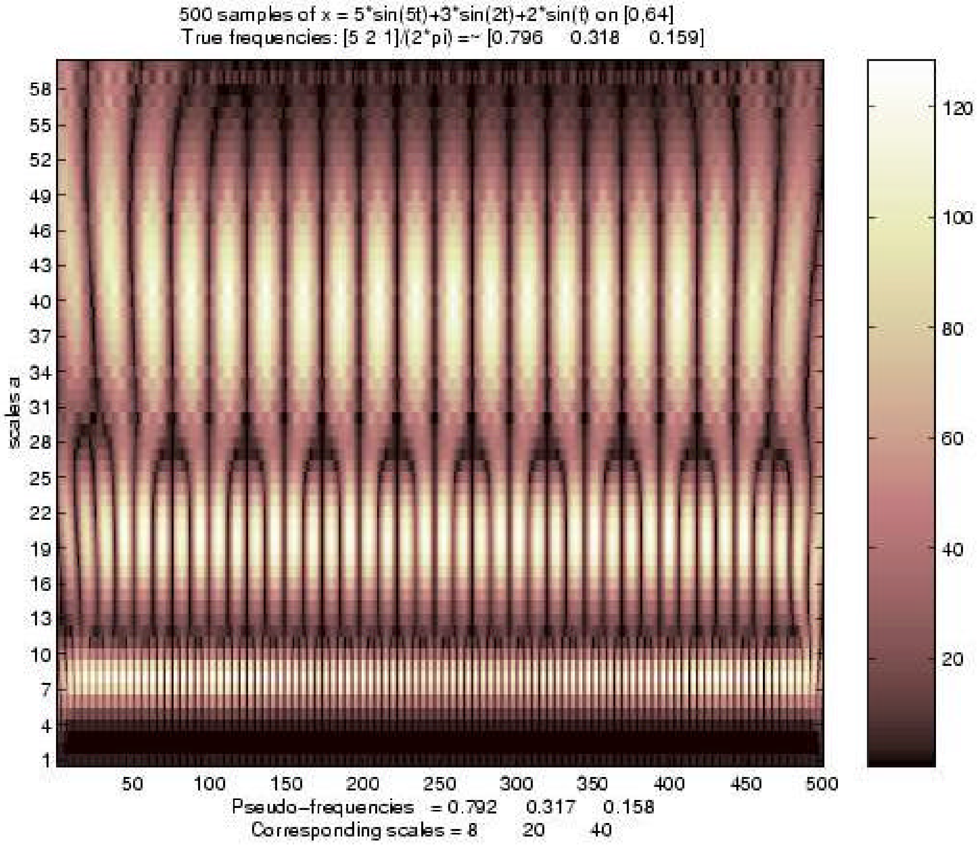


fig 3.11 : A typical scalogram (for demonstration only)

3.6 How to Connect Scale to Frequency?

The answer can only be given in a broad sense, and it's better to speak about the pseudo-frequency corresponding to a scale.

A way to do it is to compute the center frequency Fc of the wavelet and to use the following relationship:

$$F_a = \frac{\Delta \cdot F_c}{a} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.5)$$

where,

a is a scale.

Δ is the sampling period.

Fc is the center frequency of a wavelet in Hz.

Fa is the pseudo-frequency corresponding to the scale a, in Hz.

The idea is to associate with a given wavelet a purely periodic signal of frequency Fc, i.e. to approximate the wavelet by a sinusoid. The frequency maximizing the FFT of the wavelet modulus is Fc.

The following figures display the plot of the wavelet along with the associated approximation based on the center frequency.

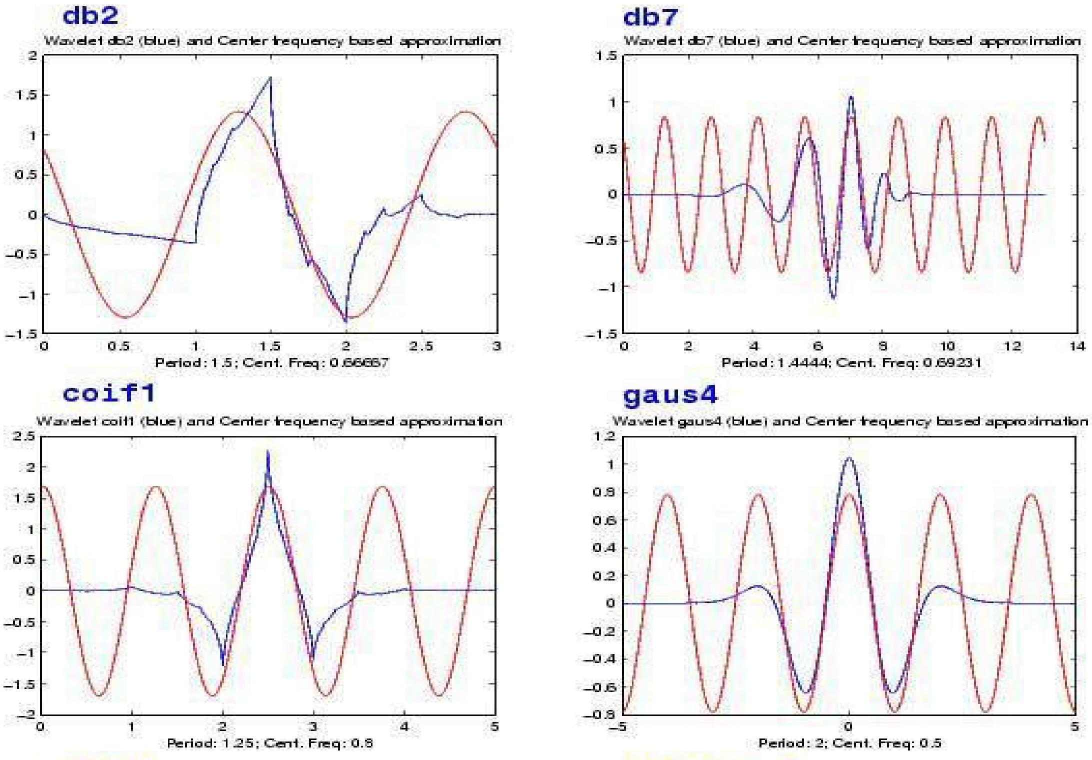


fig 3.12 : Relation between scale and sinusoidal frequency for some wavelet

As you can see, the center frequency-based approximation captures the main wavelet oscillations. So the center frequency is a convenient and simple characterization of the leading dominant frequency of the wavelet.

If we accept to associate the frequency Fc to the wavelet function, then when the wavelet is dilated by a factor a, this center frequency becomes Fc / a. Lastly, if the underlying sampling period is Δ, it is natural to associate to the scale a the frequency:

$$F_a = \frac{\Delta \cdot F_c}{a}$$

3.7 Example 2.1 revisited

In chapter 2 the weakness of FT was demonstrated with an example. We now consider the same example, and show how wavelet analysis distinguishes between the 2 different signals and also gives their frequency content. The 2 signals are repeated here for convenience.

$x1(t) = \sin(2*\pi*100*t) \qquad 0 <= t < 0.1$ sec

$= \sin(2*\pi*500*t) \qquad 0.1 <= t < 0.2$ sec

$x2(t) = \sin(2*\pi*500*t) \qquad 0 <= t < 0.1$ sec

$= \sin(2*\pi*100*t) \qquad 0.1 <= t < 0.2$ sec

The following figures show the signals along with their wavelet scalogram. Note the scalograms of these 2 signals are entirely differently, enabling the wavelet transform to distinguish between the 2 signals.

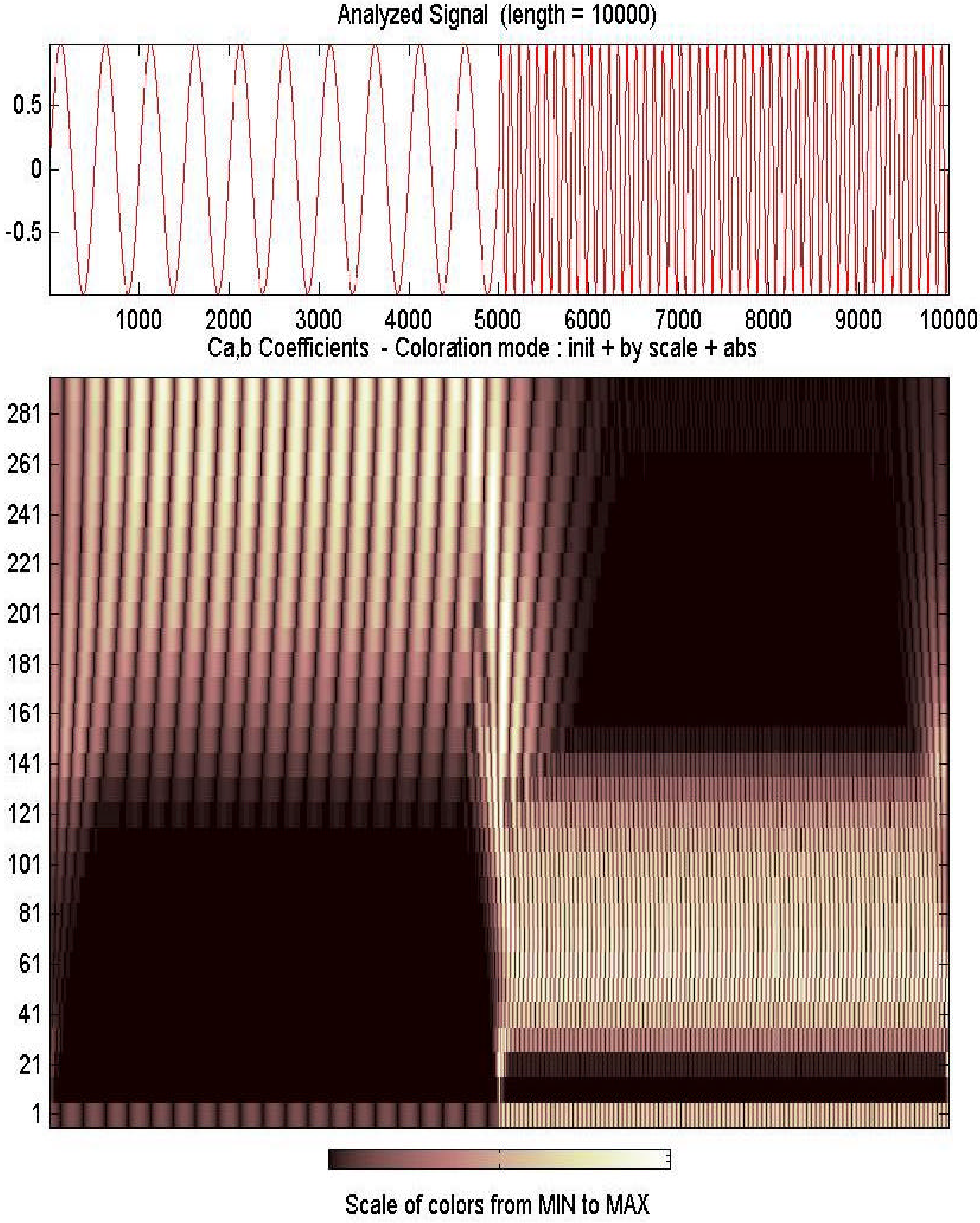


fig 3.13 : x1(t) and its scalogram

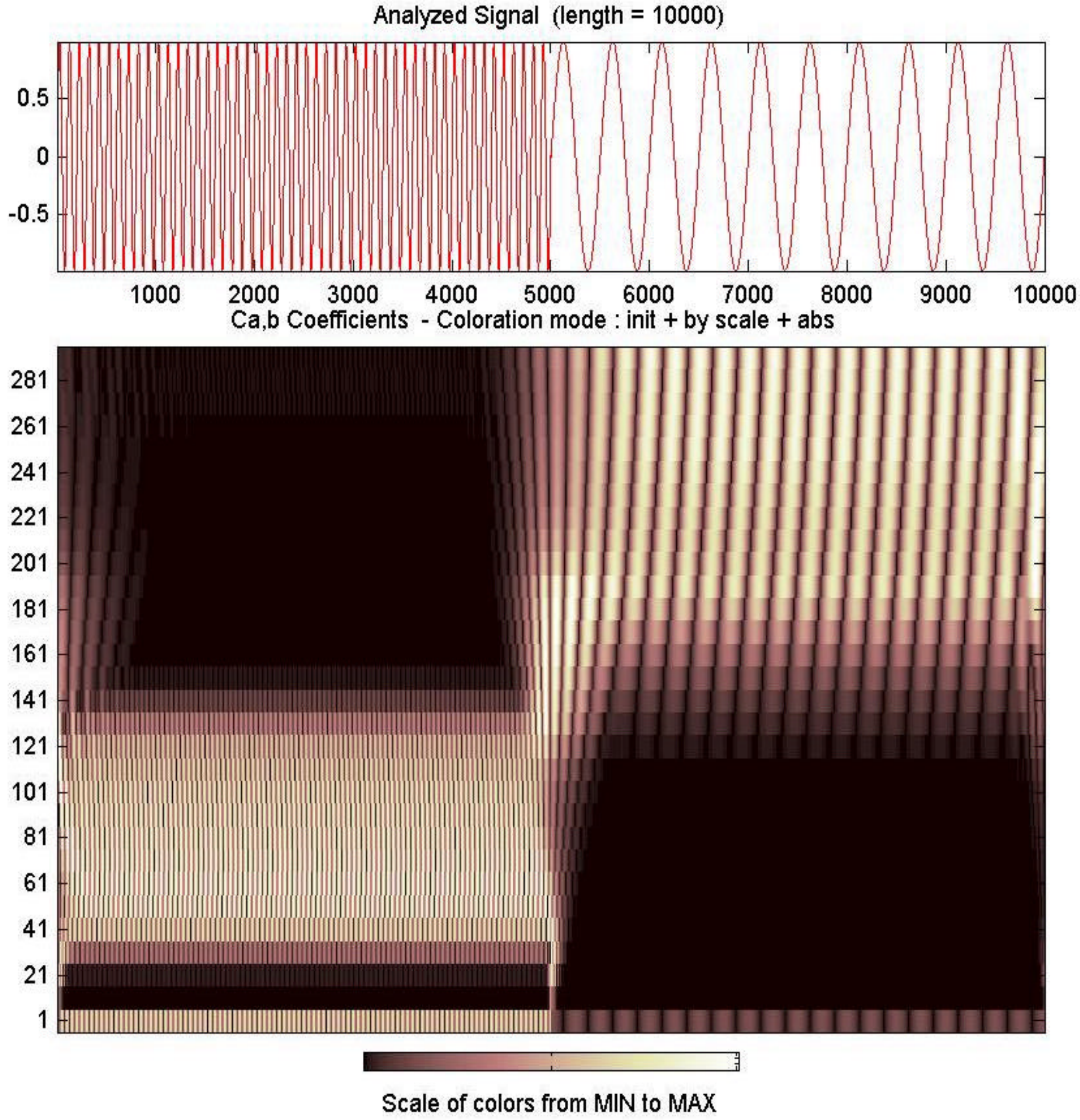


fig 3.14 : x2(t) and its scalogram

The interpretation of the above scalograms to determine the exact frequency components has been relegated to appendix. For now, it suffices to say that WAVELET TRANSFORM IS A SUITABLE TOOL FOR THE ANALYSIS OF NON-STATIONARY SIGNAL, AS IS EVIDENCED BY THE ABOVE TWO DIAGRAMS.

The computation of coefficients C(a,b) for a continuum of values of 'a' and 'b' to get the continuous wavelet transform is impractical. The next chapter deals with a more practical quantity, the DISCRETE WAVELET TRANSFORM (DWT).

Before proceeding, the reader is well advised to have a strong foundation of CWT. A detailed treatment can be found in the following literature.

1. Books: See # 10 in references.
2. Tutorial by Robi Polikar, in particular, tutorial 3(# 2 in reference in the section II)
3. From section III, review the material given in URLs 4 and 5.

4 THE DISCRETE WAVELET TRANSFORM (DWT)

INTRODUCTION

Calculating wavelet coefficients at every possible scale (for continuous WT) is a fair amount of work, and it generates an awful lot of data. What if we choose only a subset of scales and positions at which to make our calculations?

It turns out, rather remarkably, that if we choose scales and positions based on powers of two -- so-called **dyadic scales and positions** -- then our analysis will be much more efficient and just as accurate. We obtain such an analysis from the discrete wavelet transform (DWT).

An efficient way to implement this scheme using filters was developed in 1988 by Mallat. The Mallat algorithm is in fact a classical scheme known in the signal processing community as a two-channel subband coder. This very practical filtering algorithm yields a fast wavelet transform -- a box into which signal passes, and out of which wavelet coefficients quickly emerge.

A discussion of MRA (Multi-resolution analysis or approximation) bridges the gap between wavelets and the filter-bank implementation of DWT explained in this chapter.

Discussion of MRA is beyond the scope of this thesis. Interested readers are referred to

1. The book by Rao (references, section I, #10)
2. Tutorials 3 and 4 by Robi Polikar (references, section II , #2)
3. Papers by S.Mallat[1] (References, section II, # 5 &6)
4. References, section III, #4

[1] We wish to mention here that S.Mallat is one of the brightest stars in the field of wavelets. His papers have revolutionized the computation of DWT

We directly begin our discussion with the formula of DWT and then veer towards the decomposition of signal into approximation and detail coefficients. The filter banks used to achieve this are also discussed. The reverse process, i.e. reconstruction of signal from the coefficients is described later. Examples of haar, and db10 are used to demonstrate the filter coefficients, frequency response of the low and high pass decomposition and reconstruction filters.

This chapter forms the basis for the next chapter, which discusses compression.

4.1 DWT defined mathematically

The Discrete Wavelet Transform (DWT) involves choosing scales and positions based on powers of two- the so called dyadic scales and positions. The mother wavelet is rescaled or "dilated" by powers of two and translated by integers. Specifically, a function *f(t)* $\in L^2(R)$ (defines space of square integrable functions) can be represented as

$$f(t) = \sum_{j=1}^{L} \sum_{k=-\infty}^{\infty} d(j,k)\,.\,\psi(2^{-j}t - k) + \sum_{k=-\infty}^{\infty} a(L,k).\,\phi(2^{-L}t - k)$$

............(4.1)

The function $\psi(t)$ is known as the mother wavelet, while $\phi(t)$ is known as the scaling function. The set of functions

$\{\sqrt{2^{-L}}f(2^{-L}t-k),\sqrt{2^{-j}}y(2^{-j}t-k)..|\ j\leq L;\ j,k,L\in Z\}$ where Z is the set of integers, is an orthonormal basis for L ^{2} (R).

The numbers *a(L, k)* are known as the approximation coefficients at scale L, while *d(j,k)* are known as the detail coefficients at scale j.

These approximation and detail coefficients can be expressed as

$$a(L,k)=\frac{1}{\sqrt{2^{L}}}\int_{-\infty}^{\infty} f(t)\ .f\ (\ 2^{-L}t-k).d\ (t) \qquad \text{………...(4.2)}$$

$$d(\ j,k)=\frac{1}{\sqrt{2^{j}}}\int_{-\infty}^{\infty} f\ (t).\ y\ (2^{-j}t-k).d\ (t) \qquad \text{………...(4.3)}$$

The above 2 equations give a mathematical relationship to compute the approximation and detail coefficients.

This procedure is seldom adopted. A more practical approach is to use Mallat’s Fast Wavelet Transform algorithm. The Mallat algorithm for discrete wavelet transform (DWT) is, in fact, a classical scheme in the signal processing community, known as a ***two channel subband coder*** using conjugate quadrature filters or quadrature mirror filters (QMF). It is developed in the following sections.

4.2 One-Stage Filtering: Approximations and Details

For many signals, the low-frequency content is the most important part. It is what gives the signal its identity. The high-frequency content, on the other hand, imparts flavor or nuance. Consider the human voice. If you remove the high-frequency components, the voice sounds different, but you can still tell what's being said. However, if you remove enough of the low-frequency components, you hear gibberish.

In wavelet analysis, we often speak of approximations and details. The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency components.

The filtering process, at its most basic level, looks like this:

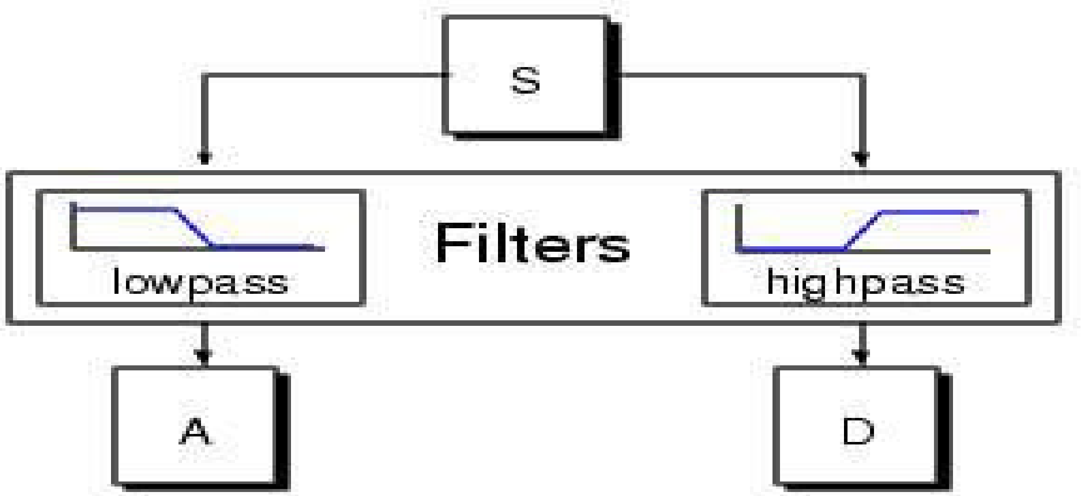


fig 4.1 : One stage filtering scheme producing the approximation and detail components of the signal

The original signal, S, passes through two complementary filters and emerges as two signals.

Unfortunately, if we actually perform this operation on a real digital signal, we wind up with twice as much data as we started with. Suppose, for instance, that the original signal S consists of 1000 samples of data. Then the resulting signals will each have 1000 samples, for a total of 2000.

These signals A and D are interesting, but we get 2000 values instead of the 1000 we had. There exists a more subtle way to perform the decomposition using wavelets. By looking carefully at the computation, we may keep only one point out of two in each of the two 2000-length samples to get the complete information. This is the notion of ***downsampling***. We produce two sequences called cA and cD.

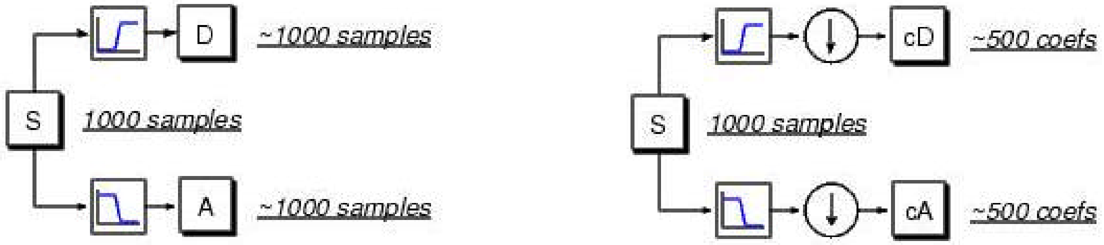


fig 4.2: Producing approximation and detail coefficients at the first level

The process on the right, which includes downsampling, produces DWT coefficients.

To gain a better appreciation of this process, let's perform a one-stage discrete wavelet transform of a signal. Our signal will be a pure sinusoid with high-frequency noise added to it.

Here is our schematic diagram with real signals inserted into it:

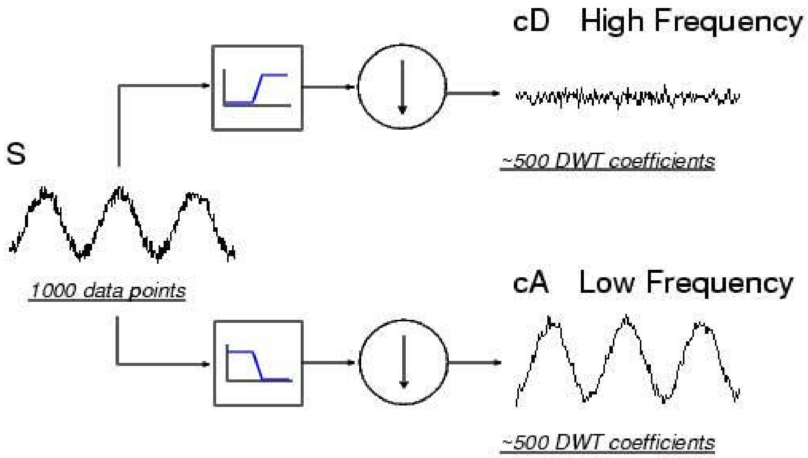


Fig4.3: Demonstration of one-stage filtering scheme for producing approximation and detail coefficient.

Notice that the detail coefficients cD are small and consist mainly of a high-frequency noise, while the approximation coefficients cA contain much less noise than does the original signal.

Note: You may observe that the actual lengths of the detail and approximation coefficient vectors are slightly more than half the length of the original signal. This has to do with the filtering process, which is implemented by convolving the signal with a filter. The convolution "smears" the signal, introducing several extra samples into the result.

In this section, we considered only one-stage decomposition of the signal into cA and cD coefficient. This process can be repeated to get multiple-level decomposition, discussed next.

4.3 Multiple-Level Decomposition:

The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components. This is called the wavelet *decomposition tree*.

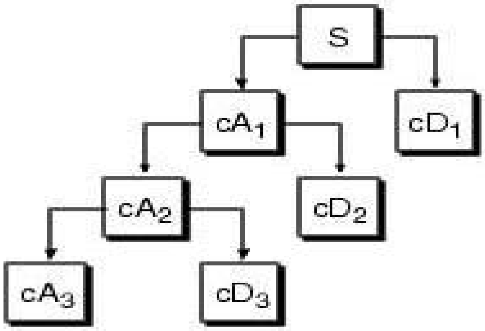


Fig 4.4: Multiple level decomposition tree

Looking at a signal's wavelet decomposition tree can yield valuable information.

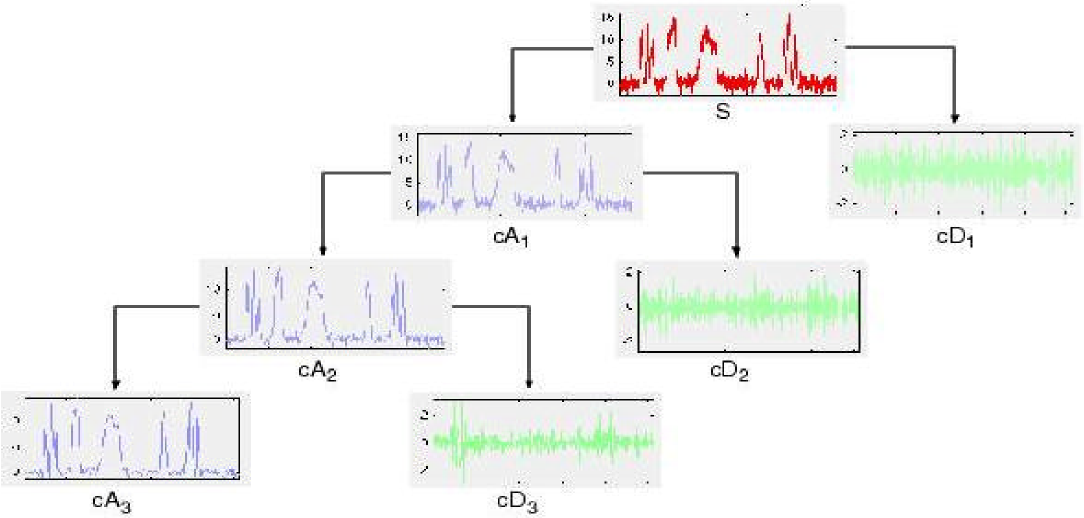


fig 4.5 : Multiple level decomposition of a signal

Number of Levels:

Since the analysis process is iterative, in theory it can be continued indefinitely. In reality, the decomposition can proceed only until the individual details consist of a single sample or pixel. In practice, you'll select a suitable number of levels based on the nature of the signal, or on a suitable criterion such as entropy.

Thus the FAST WT ALGORITHM can be stated as:

Given a signal s of length N, the DWT consists of $\log_2 N$ stages at most. Starting from s, the first step produces two sets of coefficients: approximation coefficients cA_1, and detail coefficients cD_1. These vectors are obtained by convolving s with the low-pass filter Lo_D for approximation, and with the high-pass filter Hi_D for detail, followed by dyadic decimation.

The next step splits the approximation coefficients cA_1 in two parts using the same scheme, replacing s by cA_1 and producing cA_2 and cD_2, and so on.

Now that we have seen the decomposition of a signal into wavelet (approximation and detail) coefficients, it is natural to ask whether the reverse is possible, i.e., is it possible to generate the original signal back from the coefficients, and if yes, how to achieve this.

Forturnately, there does exist a method to do it, and it is very similar to the one used for decomposition. The next few sections demonstrate this.

4.4 Wavelet Reconstruction

We've learned how the discrete wavelet transform can be used to analyze, or decompose, signals and images. This process is called decomposition or analysis. The other half of the story is how those components can be assembled back into the original signal without loss of information. This process is called reconstruction, or synthesis. The mathematical manipulation that affects synthesis is called the inverse discrete wavelet transform (IDWT).

To synthesize a signal, we reconstruct it from the wavelet coefficients:

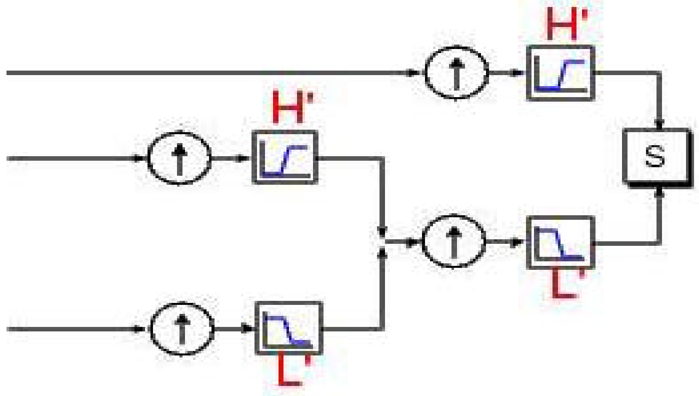


Fig 4.6: Scheme for reconstructing signal from wavelet coefficients

Where wavelet analysis involves filtering and downsampling, the wavelet reconstruction process consists of upsampling and filtering. Upsampling is the process of lengthening a signal component by inserting zeros between samples:

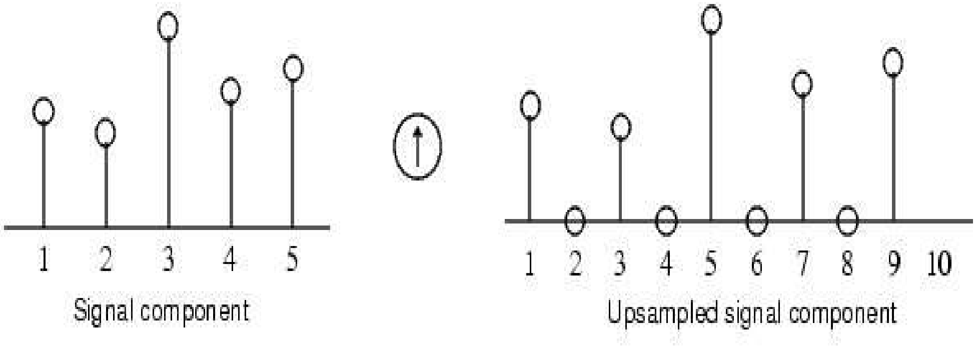


fig 4.7 : The process of Upsampling

4.5 Reconstruction Filters

The filtering part of the reconstruction process also bears some discussion, because it is the choice of filters that is crucial in achieving perfect reconstruction of the original signal.

The downsampling of the signal components performed during the decomposition phase introduces a distortion called aliasing. It turns out that by carefully choosing filters for the decomposition and reconstruction phases that are closely related (but not identical), we can "cancel out" the effects of aliasing.

The low-and high pass decomposition filters (L and H), together with their associated reconstruction filters (L' and H'), form a system of what is called quadrature mirror filters:

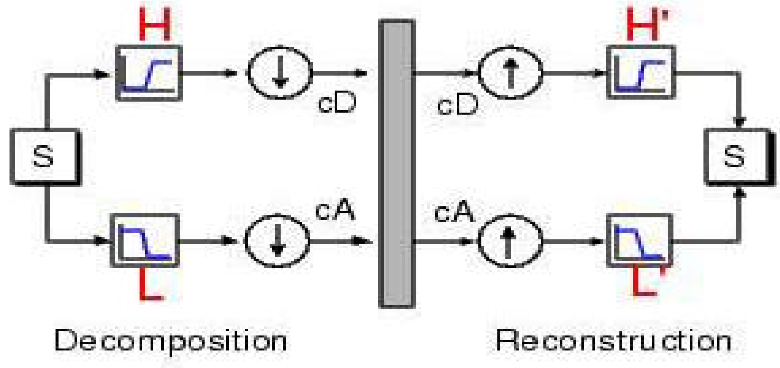


Fig 4.8: Decomposition and reconstruction filters illustrated

4.6 Reconstructing Approximations and Details

We have seen that it is possible to reconstruct our original signal from the coefficients of the approximations and details.

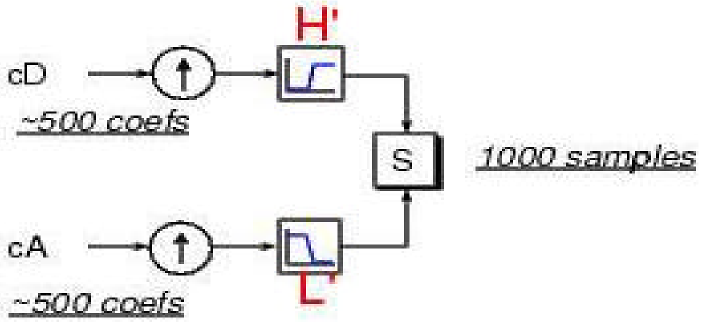


Fig 4.9: Reconstruction of signal from approximation and details.

It is also possible to reconstruct the approximations and details themselves from their coefficient vectors. As an example, let's consider how we would reconstruct the first-level approximation A1 from the coefficient vector cA1.

We pass the coefficient vector cA1 through the same process we used to reconstruct the original signal. However, instead of combining it with the level-one detail cD1, we feed in a vector of zeros in place of the detail coefficients vector:

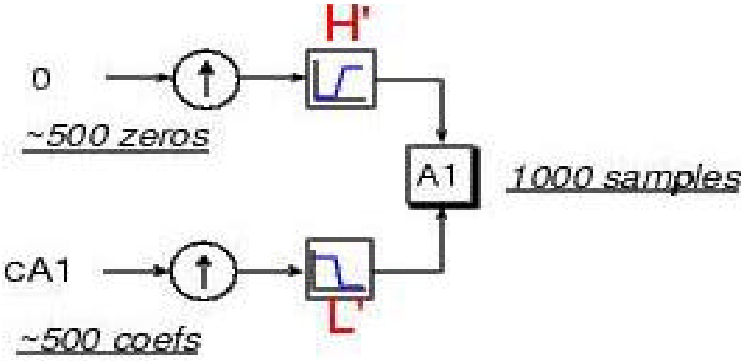


Fig 4.10: Obtaining the first level approximation of the signal.

The process yields a reconstructed approximation A1, which has the same length as the original signal S and which is a real approximation of it.

Similarly, we can reconstruct the first-level detail D1, using the analogous process:

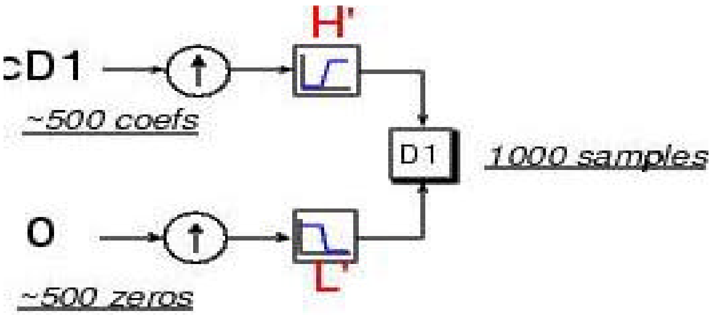


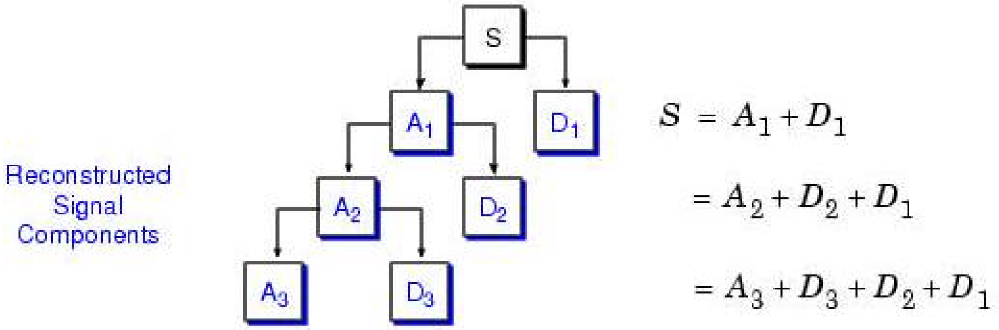
Fig 4.11: Obtaining the first level detail of the signal

The reconstructed details and approximations are true constituents of the original signal. In fact, we find when we combine them that:

$$\mathbf{A_1 + D_1 = S} \qquad \text{……………………}(4.4)$$

Note that the coefficient vectors cA1 and cD1 -- because they were produced by downsampling and are only half the length of the original signal -- cannot directly be combined to reproduce the signal. It is necessary to reconstruct the approximations and details before combining them.

Extending this technique to the components of a multilevel analysis, we find that similar relationships hold for all the reconstructed signal constituents. That is, there are several ways to reassemble the original signal:



$$\begin{aligned} S &= A_1 + D_1 \\ &= A_2 + D_2 + D_1 \\ &= A_3 + D_3 + D_2 + D_1 \end{aligned}$$

fig 4.12 : Relation between the signal and its components

4.7 Multistep Decomposition and Reconstruction

A multistep analysis-synthesis process can be represented as:

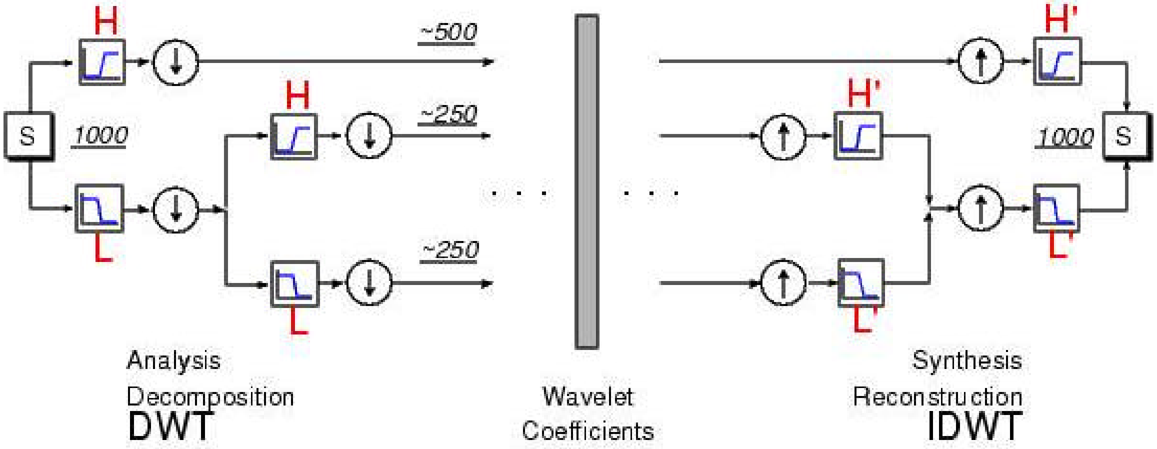


fig 4.13 : Multiple level analysis-synthesis process

This process involves two aspects: breaking up a signal to obtain the wavelet coefficients, and assembling the signal from the coefficients.

We've already discussed decomposition and reconstruction at some length. Of course, there is no point breaking up a signal merely to have the satisfaction of immediately reconstructing it. We may modify the wavelet coefficients before performing the reconstruction step.

We perform wavelet analysis because the coefficients thus obtained have many known uses, de-noising and compression being foremost among them.

But wavelet analysis is still a new and emerging field. No doubt, many uncharted uses of the wavelet coefficients lie in wait.

4.8 Haar and db2 decomposition and reconstruction filters

Having studied the implementation of DWT with the help of filter banks, it is natural to ask: how are these filters implemented ?

These filters are like any other FIR filters, characterized by an impulse response. The design of these filters is beyond the scope of this thesis. In this section we just present the filter coefficients of the 4 filters namely, the low-pass and high-pass decomposition and reconstruction filters for the haar and db2 wavelets. These values are standardized and have been plotted using MATLAB®. Along with this, frequency response of these filters is also provided.

1. **haar :**

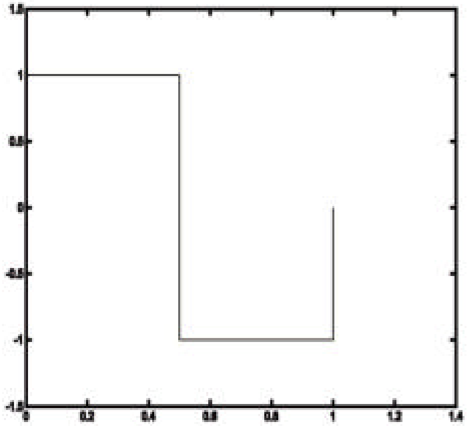
   The filter coefficients are

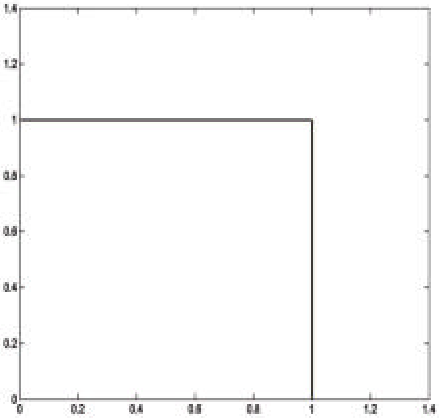
   a) Low pass decomposition filter: h(n)={ 0.7071 , 0.7071}

   b) High pass decomposition filter: h(n)={ -0.7071 , 0.7071}

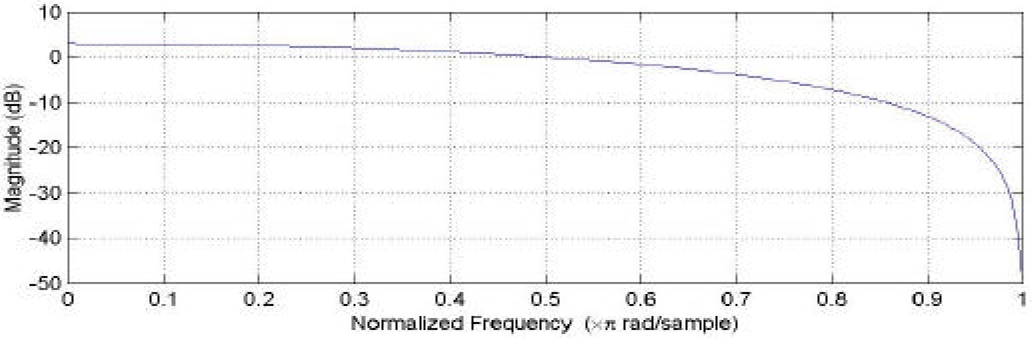
   c) Low pass reconstruction filter: h(n)={ 0.7071 , 0.7071}

   d) High pass reconstruction filter: h(n)={ 0.7071 , -0.7071}

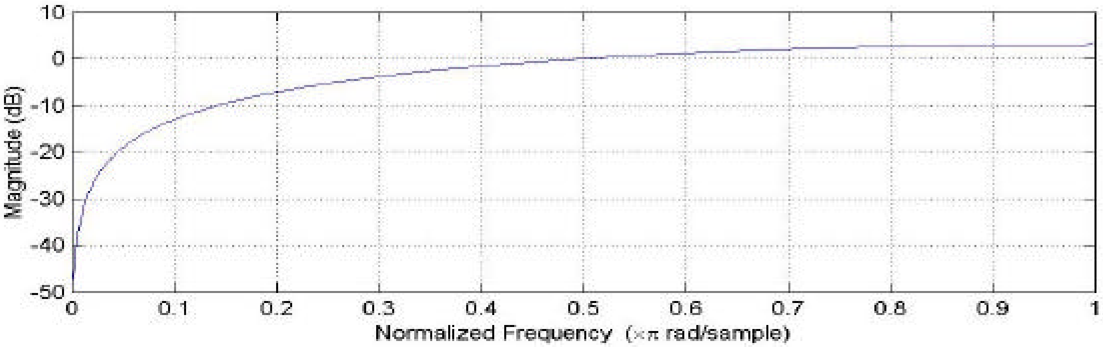




wavelet function **scaling function**



low pass decomposition/reconstruction filter frequency response



High pass decomposition/reconstruction filter frequency (magnitude)

Note that the reconstruction filters are same as decomposition filters for the low-pass filters. For the high-pass filters, the magnitude response will be same, as is evidenced from the diagrams.

2. db2

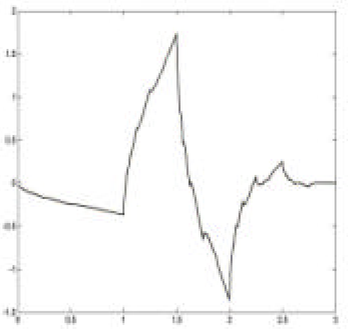
The filter coefficients are

a) Low pass decomposition filter: h(n)={ -0.1294,0.2241,0.8365,0.4830}

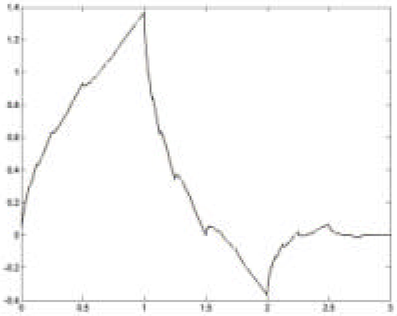
b) High pass decomposition filter: h(n)={-0.4830,0.8365,-0.2241,-0.1294 }

c) Low pass reconstruction filter: h(n)={ 0.4830,0.8365,0.2241,-0.1294}

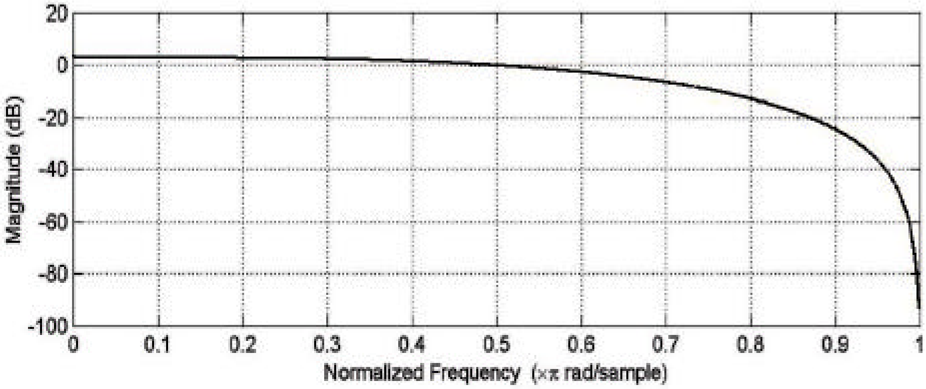
d) High pass reconstruction filter: h(n)={-0.1294,-0.2241,0.8365, -0.4830}



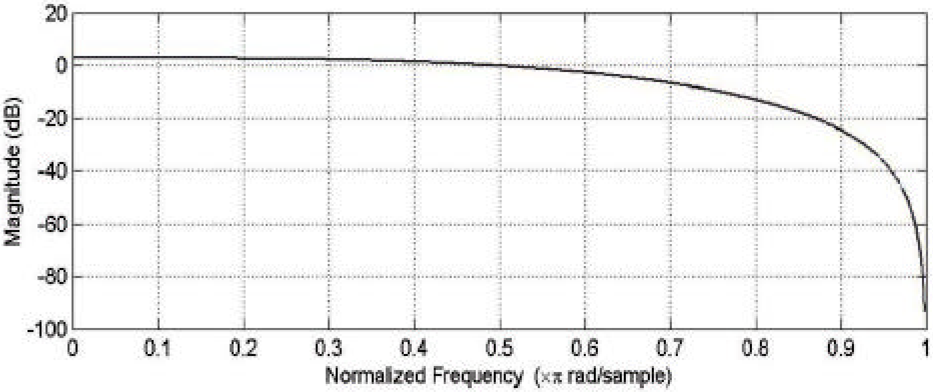
wavelet function



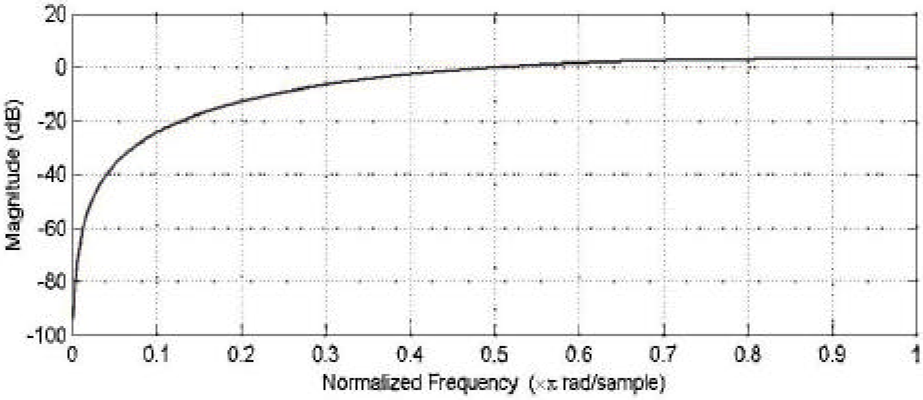
scaling function



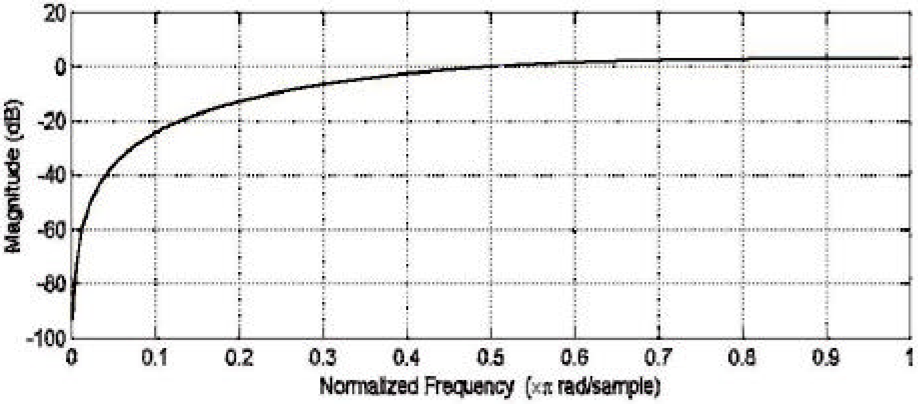
low pass decompos-ition filter



low pass reconst-ruction filter



High pass decompo s-ition filter



high pass reconstruction filter

5 WAVELETS AND SPEECH COMPRESSION

INTRODUCTION

The idea behind signal compression using wavelets is primarily linked to the relative scarceness of the wavelet domain representation for the signal. Wavelets concentrate speech information (energy and perception) into a few neighbouring coefficients. Therefore as a result of taking the wavelet transform of a signal, many coefficients will either be zero or have negligible magnitudes.

.

Another factor that comes into picture is taken from psychoacoustic studies. Since our ears are more sensitive to low frequencies than high frequencies and our hearing threshold is very high in the high frequency regions, we used a method for compression by means of which the detail coefficients (corresponding to high frequency components) of wavelet transforms are thresholded such that the error due to thresholding is inaudible to our ears.

Since some of the high frequency components are discarded, we should expect a smoothened output signal, as is shown in the following figure:

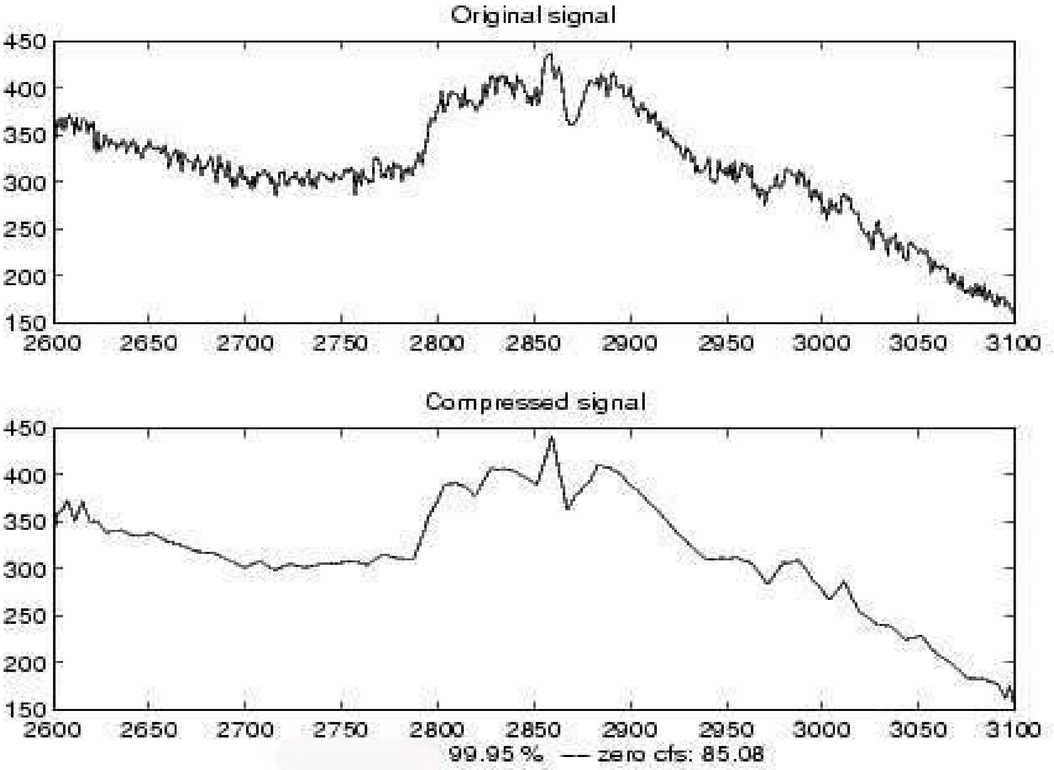


fig 5.1 : Effect of compression (smoothening of the signal)

In summary, the notion behind compression is based on the concept that the regular signal component can be accurately approximated using the following elements: a small number of approximation coefficients (at a suitably chosen level) and some of the detail coefficients

Data compression is then achieved by treating small valued coefficients as insignificant data and thus discarding them. The process of compressing a speech signal using wavelets involves a number of different stages, each of which is discussed below.

5.1 Choice of Wavelet

The choice of the mother-wavelet function used in designing high quality speech coders is of prime importance. Several different criteria can be used in selecting an optimal wavelet function. The objective is to minimize reconstructed error variance and maximize signal to noise ratio (SNR). In general optimum wavelets can be selected based on the energy conservation properties in the approximation part of the wavelet coefficients. A suitable criterion for selecting optimum mother wavelets is related to the amount of energy a wavelet basis function can concentrate into the level 1 approximation coefficients.

In chapter 7, several experiments are conducted and a suitable wavelet is suggested.

5.2 Wavelet Decomposition:

Wavelets work by decomposing a signal into different resolutions or frequency bands, and this task is carried out by choosing the wavelet function and computing the Discrete Wavelet Transform (DWT). Signal compression is based on the concept that selecting a small number of approximation coefficients (at a suitably chosen level) and some of the detail coefficients can accurately represent regular signal components. Choosing a decomposition level for the DWT usually depends on the type of signal being analyzed or some other suitable criterion such as entropy. For the processing of speech signals decomposition up to scale 5 is adequate, with no further advantage gained in processing beyond scale 5. This fact is derived from the experiments described later in chapter 7.

5.3 Truncation of Coefficients:

After calculating the wavelet transform of the speech signal, compression involves truncating wavelet coefficients below a threshold. From the experiments that we conducted, we found that most of the coefficients have small magnitudes. Speaking in general terms, more than 90% of the wavelet coefficients were found to be insignificant, and their truncation to zero made an imperceptible difference to the signal. This means that most of the speech energy is in the high-valued coefficients, which are few. Thus the small valued coefficients can be truncated or zeroed and then be used to reconstruct the signal.

Two different approaches are available for calculating thresholds:

1. **Global threshold:**

It involves taking the wavelet expansion of the signal and keeping the largest absolute value coefficients. In this case you can manually set a global threshold, a compression performance or a relative square norm recovery performance. Thus, only a single parameter needs to be selected. The coefficient values below this value should be set to zero, to achieve compression.

The following figure shows the setting of global threshold for a typical speech signal. In this figure, the X-axis represents the coefficient values. (Since the signal samples are normalized to 1 in MATLAB®, the coefficient values too are normalized and the maximum value is one). The black (dark) vertical line moves to right or left, thereby changing the threshold. The intersection of this line with green line indicates the percentage of zero coefficients below this threshold. Its intersection with the red line indicates the percentage of signal energy retained after truncating these coefficients to zero

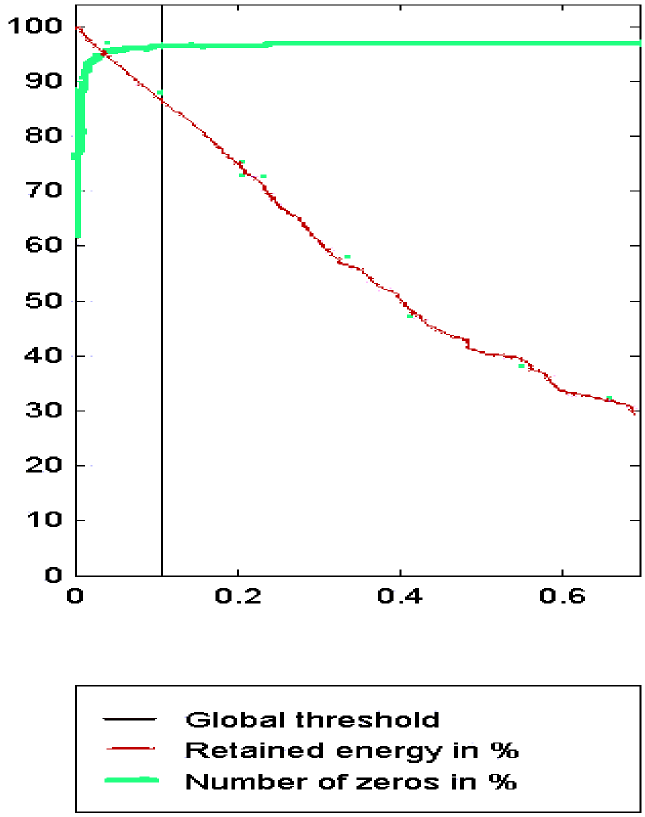


Fig 5.2:. Setting a global threshold

2. **Level dependent thresholding:** This approach consists of applying visually determined level dependent thresholds to each decomposition level in the Wavelet Transform

The following figure shows the level-dependent thresholding. The truncation of insignificant coefficients can be optimized when such a level dependent thresholding is used

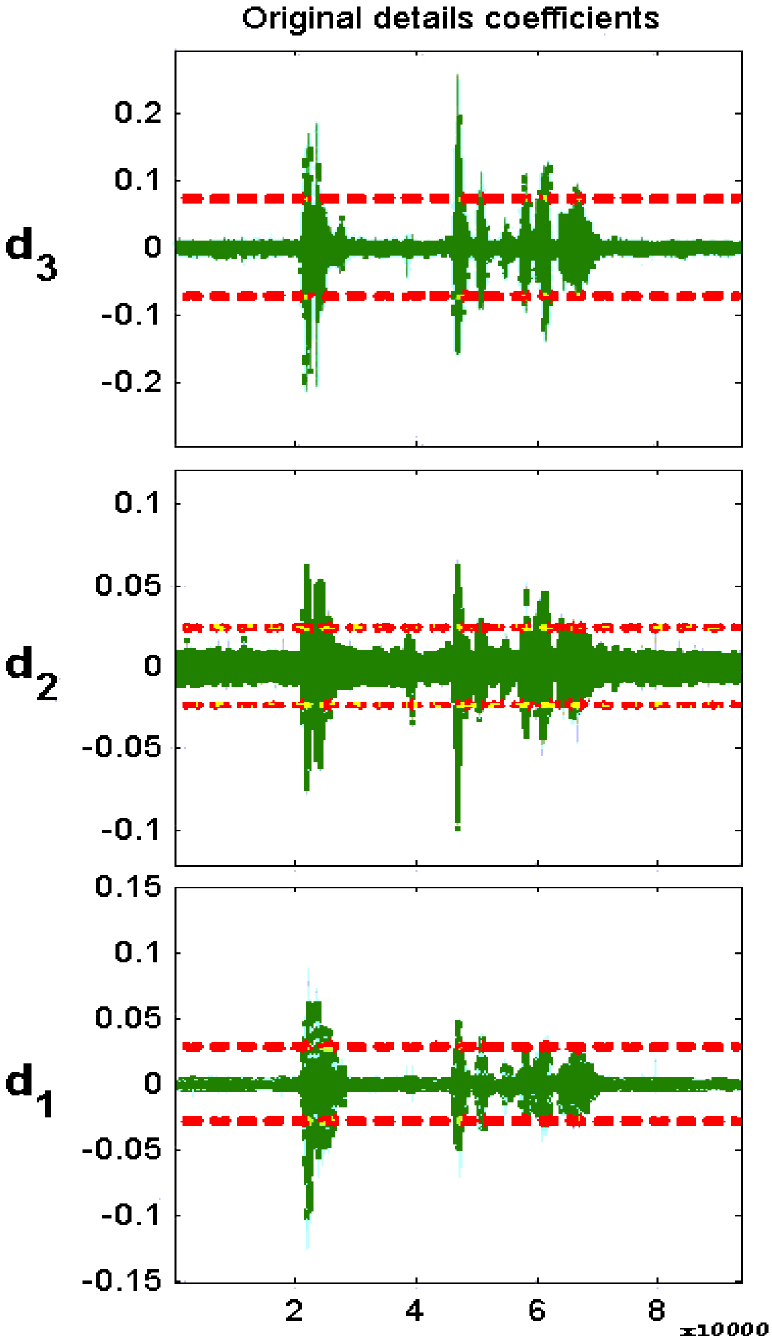


fig 5.3 : level dependent threshold

5.4 Encoding

Signal compression is achieved by first truncating small-valued coefficients and then efficiently encoding them.

One way of representing the high-magnitude coefficients is to store the coefficients along with their respective positions in the wavelet transform vector.

Another approach is the Run Length Encoding (RLE) wherein, the consecutive zero valued coefficients are replaced with two bytes. One byte to indicate a sequence of zeros in the wavelet transforms vector and the second byte representing the number of consecutive zeros. In the thesis, we have used a slightly different approach. The vector of wavelet coefficient, after truncation, is encoded, and is replaced by 2 vectors. One vector contains only the significant coefficients, without any zero values between them. The other vector stores the starting position of a string of zeros and the number of zeros in the string. Thus 2 bytes are needed for every string of zeros.

5.5 Performance Measures

1.<u>Compression factor:</u> It is the ratio of the original signal to the compressed signal. Of course, for the compressed signal we have to take into account all the values that would be needed to completely represent the signal. As has been explained in the previous section, this thesis implements encoding using a modification of RLE wherein 2 vectors are produced, we must take into account the combined length of these 2 vectors.

2.<u>Retained signal energy:</u> This indicates the amount of energy retained in the compressed signal as a percentage of the energy of original signal.

When compressing using orthogonal wavelets, the Retained energy in percentage is defined by:

$$\frac{100*(\text{vector-norm(coeffs of the current decomposition,2)})^2}{(\text{vector-norm(original signal,2)})^2}$$

3. Signal to noise ratio (SNR): This value gives the quality of reconstructed signal. Higher the value, better. It is given by:

$$SNR = 10\log_{10}\left(\frac{s_x^2}{s_e^2}\right)$$

where σ_x and σ_e are respectively the mean square of the speech signal and the mean square difference between the original and reconstructed signals.

4. Percentage of zero coefficient: It is given by the following relation:

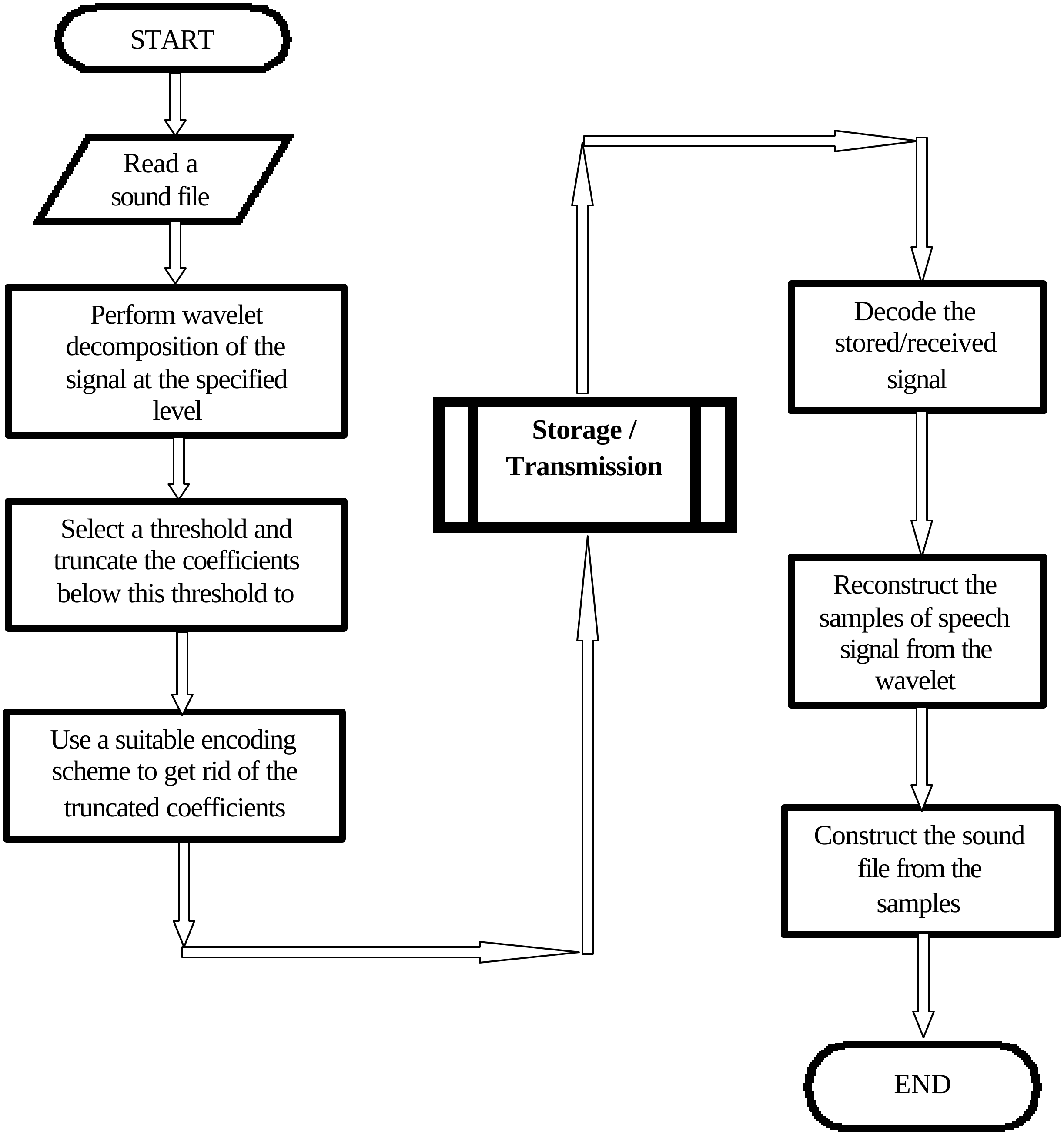
$$\frac{100*(number\ of\ zeros\, of\ the\, current\ decomposition)}{Number\ of\ coefficients}$$

5.Signal energy in the first level approximation: This quantity helps in the selection of appropriate mother wavelet for compression. The higher the amount of energy in the first level approximation, better is the wavelet for compression of that signal.

These parameters discussed in this section are calculated and displayed as output in the software implementation of this work. The experimentation carried out in chapter 7 is based entirely on this.

Before we proceed, we present the flowchart for performing compression-reconstruction using wavelet method:

SPEECH COMPRESSION-RECONSTRUCTION FLOWCHART


START
Read a sound file
Perform wavelet decomposition of the signal at the specified level
Select a threshold and truncate the coefficients below this threshold to
Use a suitable encoding scheme to get rid of the truncated coefficients
Storage / Transmission
Decode the stored/received signal
Reconstruct the samples of speech signal from the wavelet
Construct the sound file from the samples
END

6 IMPLEMENTATION IN MATLAB â

INTRODUCTION

Having studied the steps needed to perform speech compression using the wavelet approach, it now remains to implement it. MATLAB® version 6.1 has numerous functions and graphical tools to achieve this. These have been exploited in our study of wavelet and also in the implementation of software for speech compression.

Recollect that the primary objective was to be able to store a sound file in the .wav format as a compressed file occupying lesser disk space.

Hence this chapter explains each step in the process, from reading a .wav file to the final saving of another file of smaller size, but containing sufficient data to reconstruct the original sound file, with imperceptible degradation.

This chapter is organized as an algorithm, with each section representing a step. The MATLAB® functions used in the various steps are elaborated within the section.

6.1 Reading a sound file

To compress a sound file, we first need to take its samples into a vector. Let 'y' be the vector. The command is

```
[y,fs,bps] = wavread('path of the file') ;
```

This command stores the samples of the sound file in the vector y. The term ‘fs’ stores the sampling frequency of the file and ‘bps’ is the bits per sample. These 2 values are needed to reconstruct the .wav file using ‘wavwrite’ function.

6.2 Performing wavelet decomposition

The vector ‘y’ from the previous step is now decomposed using DWT into approximation and detail coefficients at various levels. The command is

```
[C,L] = wavedec(y,N,'wname');
```

where, **N** = number of decomposition levels.

‘wname’ = name of the wavelet

The output decomposition structure contains the wavelet decomposition vector C and the bookkeeping vector L. The structure is organized as in the level-3 decomposition example shown in fig 6.1:

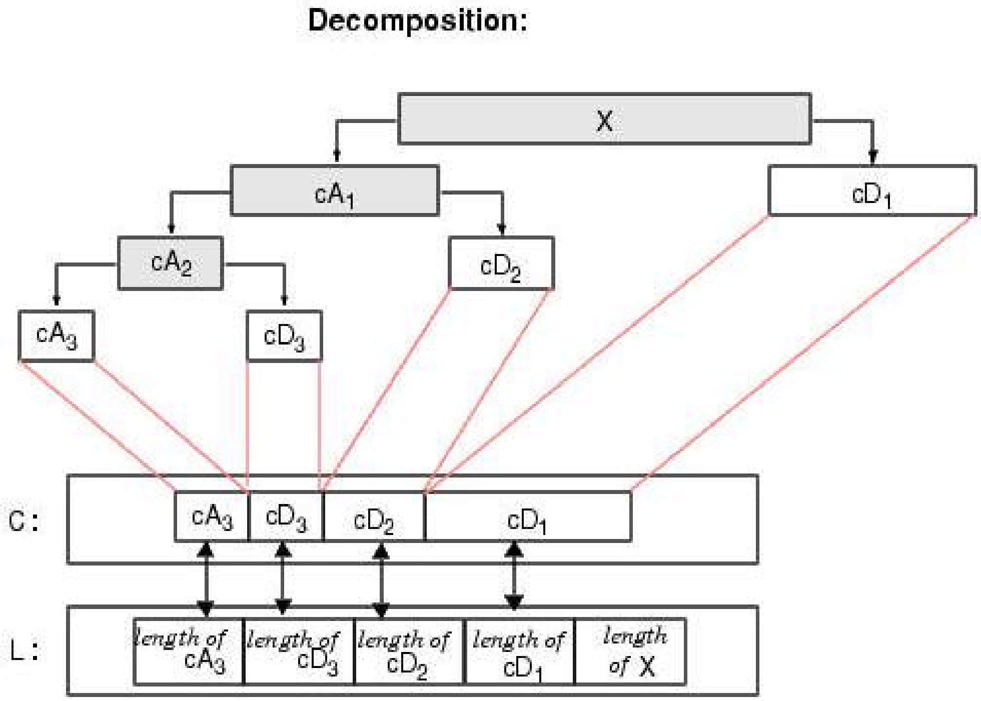


fig 6.1 : Contents of C and L vectors

6.3 Perform Compression

The coefficient vector 'C' computed in the previous step will now be compressed, i.e. the insignificant detail coefficients will be truncated to zero.

Note that only detail coefficients will be truncated. Approximation coefficients will not be affected.

The syntax is:

```
[XC,CXC,LXC,PERF0,PERFL2] = wdencmp('gbl',C,L,wlet,
                                  decomplevel,thr,sorh,keepapp);
```

The various parameters used are described as follows:

1. **Inputs (quantities on the right hand side):**

gbl	Indicates global thresolding. Used ‘lvd’ for level dependent thresholding
C	Wavelet coefficient vector (computed in step 2)
L	Book-keeping vector (computed in step 2)
wlet	Wavelet function used for decomposition in step 2
decomplevel	The number of decomposition levels
thr	The threshold value selected
sorh	It indicates whether soft or hard thresholding is used. We have used hard thresholding throughout.
keepapp	Keep approximation coefficients? Value of ‘1’ indicates that approximation coefficients are left unscathed in the truncation operation.

2. **Outputs (quantities on left hand side):**

XC	Gives the reconstructed signal after compression.
CXC	Truncated coefficients vector
LXC	Book keeping vector
PERF0	% of zero coefficients
PERF2	Retained signal energy (in %)

6.4 Encode the CXC vector

An algorithm was designed and then implemented in MATLAB® for encoding the coefficient vector. The program (M-file) was saved as 'encode1.m'. The logic works as follows:

In the CXC vector, there are long strings of zeros. This CXC is replaced by 2 vectors, 'y' and 'posnum'. The vector 'y' will store, in contiguous positions, only the non-zero members of CXC.

The 'posn`um' vector will store 2 numbers for each string of zeros, the first number storing the index of first zero of the stream of zeros, and the 2nd number storing the number of contiguous-zeros in the stream.

Example: Let CXC = [1,0,0,0,0,0,2,3,0,0,6,0,0,0,7]

Use the command

```
[y,posnum] = encode1(CXC) ;
```

When you give this command,

```
     y =  [1,2,3,6,7]
posnum = [2,5,9,2,12,3]
```

6.5 Decode

It is the reverse of ‘encode1’. The M-file is ‘decode1.m’ and the command is

```
Rx  = decode1(y,posnum,N);
```

where,

Rx = reconstructed coefficient vector. Note this Rx =CXC of section 6.4.

N = Length of vector C (or CXC).

6.6 Reconstructing the signal from wavelet coefficients

The signal can be reconstructed from wavelet coefficient at any level using ‘waverec’ command. The syntax is

```
Y = waverec(C,L,’wavelet’);
```

Note that in this case, perfect reconstruction is being performed, since ‘C’ is used. In actual compression, it is the ‘Rx’ vector that is used in place of ‘C’;

6.7 Speech Comparison GUI

The ‘SPEECH COMPRESSION GRAPHICAL USER INTERFACE (GUI)’ included in the CD has the aforementioned commands at its core. The design of GUI is beyond the scope of thesis and depends to a great extend on the programming skills and familiarity of the user with MATLAB®. The help documentation of MATLAB® is a very good starting point for a beginner.

7 RESULTS

Foreword:

In chapter 5, many key issues regarding the compression were merely stated. Primary among them were the choice of wavelets and the number of decomposition levels. In this chapter, we conducted a series of statistical analysis and arrived at some results which can serve as a guide for selection of these parameters.

We have also attempted to study the dependence of compression on the sampling frequency of the signal. Since human speech has significant components only upto 4khz, according Nyquist rule, minimum sampling frequency required is 8kHz which is also the minimum required representation for the signal. Increasing the sampling frequency introduces redundant information, which will give more compression. Readers are cautioned against being overzealous upon getting very high compression factors when using signals with higher sampling rates. Such results are not indicative of general trend.

We have considered only haar, daubechies and the symlet families in our study. The biorthogonal wavelets have been completely excluded.

We performed a series of trial and errors on a few signals and arrived at an acceptable figure for SNR (signal to noise ratio). We found this value to be 10dB. However, to have safe margin, we took 12dB as the minimum SNR for the reconstructed signal for getting imperceptible degradation.

The threshold values used are global, as explained in section 5.3. We used the MATLAB® function 'ddencmp' to compute the threshold for the signal. However this figure gives very conservative results. Hence we multiply it by some scalar to increase the compression factor. Concomitant of course is the reduction in SNR.

STATISTICAL ANALYSIS:

The rest of this chapter is the tabulation our observations. Some of the data common to all the experiments are given on this page. Three experiments were conducted.

Test Signal 1: "A quick brown fox jumped over the lazy dog."

Test signal 2: "Twinkle twinkle little star, how I wonder what you are."

Sampling frequency (Hz)	Original signal length
22050	66156
8000	23999

Test signal	Type of voice	Default threshold ($f_{s\,=}$ 22kHz)	Default threshold ($f_{s\,=}$ 8kHz)
Test signal 1	Male	0.0027621	0.0049718
	Female	0.0008286	0.0019335
Test signal 2	Male	0.0011049	0.0024859
	Female	0.0022097	0.0044194

(CURRENT THRESHOLD = DEFAULT THR. X THR MULTIPLICATION FACTOR)

EXPERIMENT 1: **-** To choose the optimal wavelet for performing wavelet transform of the speech signal.

PROCEDURE:-

1.Load test signal 1 (male voice) having a sampling frequency of 22kHz.

2.Run the software at a SNR of 12dB for decomposition levels of 4, 5 and 6 for the following wavelets:-Haar,db2, 4, 6, 8, 10 and sym 1,2,4,6,8.

3.Repeat step 2 for test signal 1 (male voice) having a sampling frequency of 8kHz.

4.Repeat step 2 for test signal 1 (female voice) having sampling frequencies of 22kHz and 8kHz.

5. Tabulate the values.

OBSERVATIONS:-

MALE VOICE (SAMPLING RATE = 22kHz):-

Level 4:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	18	93.428	98.993	10.940	93.857
db2	28	94.763	99.656	16.119	93.719
db4	34.4	94.979	99.742	17.727	93.702
db6	34.55	94.927	99.765	17.837	93.967
db8	37	94.880	99.773	17.851	93.742

db10	41.3	94.853	99.776	17.992	93.723
sym1	18.2	93.475	98.993	11.032	93.759
sym2	28	94.763	99.656	16.119	93.719
sym4	34.2	94.965	99.747	17.609	93.704
sym6	34.9	94.939	99.765	17.880	93.871
sym8	38.5	94.916	99.772	18.061	93.728

Level 5:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	17	94.178	98.993	11.684	93.731
db2	25	95.769	99.656	17.779	93.719
db4	29.3	96.123	99.742	20.456	93.713
db6	30.05	96.186	99.765	21.102	93.726
db8	30.75	96.116	99.773	21.238	93.716
db10	31.7	96.181	99.776	21.606	93.693
sym1	17	94.178	98.993	11.684	93.731
sym2	25	95.769	99.656	17.779	93.719
sym4	28.75	96.119	99.747	20.349	93.727
sym6	30.2	96.180	99.765	20.823	93.696
sym8	31.25	96.193	99.772	21.055	93.726

Level 6:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy

Haar	16.75	94.483	98.993	11.697	93.812
db2	23.48	96.294	99.656	18.531	93.738
db4	27	96.734	99.742	21.935	93.696
db6	27.7	96.845	99.765	22.781	93.738
db8	27.9	96.791	99.773	22.859	93.700
db10	28.8	96.861	99.776	23.493	93.729
sym1	16.9	94.483	98.993	11.697	93.812
sym2	23.5	96.299	99.656	18.547	93.722
sym4	26.5	96.707	99.747	21.705	93.714
sym6	27.6	96.817	99.765	22.395	93.714
sym8	28.2	96.860	99.772	22.947	93.726

MALE VOICE (SAMPLING RATE = 8kHz):-

Level 4:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	7	86.438	94.418	4.824	93.925
db2	8.3	89.192	97.416	6.689	93.707
db4	9.1	90.315	98.156	7.805	93.711
db6	9.7	90.392	98.159	8.053	93.734
db8	9.58	90.357	98.206	8.179	93.709
db10	9.82	90.417	98.224	8.368	93.723
sym1	7	86.438	94.418	4.824	93.925
sym2	8.25	89.162	97.416	6.661	93.748
sym4	9.4	90.219	98.145	7.815	93.733

sym6	9.55	90.346	98.205	8.069	93.704
sym8	10	90.548	98.214	8.309	93.719

Level 5:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	6.85	87.380	94.418	4.978	93.692
db2	7.85	90.175	97.416	6.839	93.704
db4	8.6	91.366	98.156	8.124	93.746
db6	9.2	91.469	98.159	8.415	93.729
db8	9.1	91.518	98.206	8.532	93.745
db10	9.23	91.653	98.224	8.866	93.706
sym1	6.85	87.383	94.418	4.978	93.692
sym2	7.8	90.142	97.416	6.812	93.747
sym4	8.75	91.274	98.145	8.094	93.708
sym6	9	91.481	98.205	8.394	93.730
sym8	9.15	91.729	98.214	8.752	93.747

Level 6:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	6.7	87.929	94.418	5.066	93.561
db2	7.6	90.743	97.416	6.946	93.753
db4	8.4	92.096	98.156	8.477	93.728
db6	8.9	92.316	98.159	8.912	93.729

db8	8.85	92.428	98.206	9.108	93.709
db10	8.95	92.576	98.224	9.523	93.706
sym1	6.67	87.858	94.418	5.050	93.626
sym2	7.65	90.772	97.416	6.952	93.718
sym4	8.45	91.984	98.145	8.388	93.701
sym6	8.72	92.324	98.205	8.879	93.692
sym8	8.95	92.623	98.214	9.284	93.729

FEMALE VOICE (SAMPLING RATE = 22kHz):-

Level 4:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	9	88.275	95.914	5.241	94.582
db2	9.185	91.305	97.375	7.664	93.738
db4	9.55	92.101	97.757	8.844	93.713
db6	9.8	92.211	97.822	9.131	93.703
db8	9.64	92.098	97.832	9.375	93.705
db10	9.88	92.188	97.826	9.162	93.698
sym1	9	88.275	95.914	5.241	94.582
sym2	9.185	91.305	97.375	7.664	93.738
sym4	9.9	92.179	97.757	8.995	93.723
sym6	9.8	92.269	97.826	9.014	93.699
sym8	9.92	92.378	97.848	9.379	93.709

Level 5:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	9	88.456	95.914	5.274	94.447
db2	9.1	91.622	97.375	7.622	93.703
db4	9.33	92.457	97.757	8.776	93.703
db6	9.59	92.614	97.822	9.079	93.695
db8	9.43	92.508	97.832	9.299	93.695
db10	9.61	92.598	97.826	9.152	93.698
sym1	9	88.456	95.914	5.274	94.447
sym2	9.1	91.622	97.375	7.622	93.703
sym4	9.7	92.564	97.757	8.955	93.699
sym6	9.56	92.666	97.826	8.962	93.692
sym8	9.68	92.788	97.848	9.361	93.696

Level 6:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	9	88.638	95.914	5.294	94.387
db2	9.02	91.819	97.375	7.616	93.693
db4	9.26	92.667	97.757	8.783	93.692
db6	9.46	92.833	97.822	9.104	93.693
db8	9.33	92.751	97.832	9.316	93.691
db10	9.47	92.858	97.826	9.148	93.692
sym1	9	88.638	95.914	5.294	94.387
sym2	9.02	91.819	97.375	7.616	93.693

sym4	9.56	92.771	97.757	8.940	93.691
sym6	9.47	92.896	97.826	8.997	93.692
sym8	9.52	93.036	97.848	9.376	93.694

FEMALE VOICE (SAMPLING RATE = 8kHz):-

Level 4:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	2.858	82.320	85.482	3.612	92.328
db2	3.381	83.378	89.975	3.946	93.153
db4	3.409	83.793	91.119	3.975	93.699
db6	3.363	83.721	91.324	4.054	93.686
db8	3.429	83.753	91.619	4.166	93.686
db10	3.394	83.513	91.540	4.072	93.689
sym1	2.858	82.320	85.482	3.612	92.328
sym2	3.380	83.378	89.975	3.946	93.153
sym4	3.380	83.718	91.062	4.042	93.679
sym6	3.390	83.700	91.359	4.081	93.682
sym8	3.399	83.915	91.453	4.117	93.668

Level 5:-

Family	Threshold Multiplication	% of zero coefficients	Signal energy in the first level	Compression Factor	Retained signal energy

	Factor		approximation		
Haar	2.857	82.771	85.482	3.646	92.157
db2	3.380	84.054	89.975	3.994	93.005
db4	3.354	84.334	91.119	3.973	93.695
db6	3.356	84.422	91.324	4.109	93.578
db8	3.358	84.423	91.619	4.188	93.693
db10	3.343	84.253	91.540	4.117	93.671
sym1	2.859	82.771	85.482	3.646	92.157
sym2	3.381	84.054	89.975	3.994	93.005
sym4	3.335	84.251	91.062	4.041	93.688
sym6	3.330	84.306	91.359	4.078	93.686
sym8	3.337	84.564	91.453	4.116	93.680

Level 6:-

Family	Threshold Multiplication Factor	% of zero coefficients	Signal energy in the first level approximation	Compression Factor	Retained signal energy
Haar	2.858	83.4475	85.482	3.717	92.069
db2	3.380	83.464	89.975	3.601	94.008
db4	3.330	85.112	91.119	4.083	93.694
db6	3.299	85.152	91.324	4.204	93.681
db8	3.339	85.379	91.619	4.344	93.682
db10	3.315	85.243	91.540	4.251	93.679
sym1	2.857	83.447	85.482	3.717	92.069
sym2	3.385	84.955	89.975	4.143	92.855
sym4	3.306	85.024	91.062	4.146	93.689
sym6	3.301	85.181	91.359	4.199	93.684

sym8	3.303	85.491	91.453	4.286	93.687

CONCLUSIONS:-

1. Within a given family, compression factor, threshold multiplication factor and % of zero coefficients increases with increase in decomposition level at constant SNR.
2. Considering compression factor and signal energy in 1st level approximation at a given SNR:-

   a) db10 and sym8 wavelets are best for male speech signal

   b) db8 and sym8 wavelets are best for female speech signal.

3. Haar, sym1 and db1 wavelets show nearly similar characteristics for constant SNR.

EXPERIMENT 2:- To find the optimum decomposition level.

PROCEDURE:-

1. For test signal 1 at a threshold multiplication factor of 5, run the software for decomposition levels 1 to 10 for db10,sym6 and Haar wavelets at sampling frequencies of 22kHz and 8kHz on male and female voices. Tabulate the values of SNR and compression factor.

OBSERVATIONS:-

Sampling rate = 22kHz:-

db10:-

Level	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
1	27.6566	3.0523	18.5334	3.2546
2	23.4328	5.0838	16.5411	4.1152
3	21.8954	6.7943	15.9036	4.4222
4	21.4316	7.8014	15.6949	4.5325
5	21.2537	8.1153	15.6406	4.554
6	21.1736	8.2468	15.6137	4.5761
7	21.1364	8.3152	15.5679	4.6533
8	21.098	8.4307	15.5443	4.6942
9	21.0807	8.455	15.529	4.7079
10	21.0769	8.4631	15.5239	4.7079

sym6:-

Level	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
1	27.4479	3.1185	18.1079	3.494
2	23.4272	5.1814	16.411	4.2692
3	21.8907	6.7396	15.8744	4.5192
4	21.4299	7.5954	15.699	4.6118
5	21.2468	7.8468	15.6441	4.6263
6	21.1755	7.9476	15.6179	4.6415
7	21.1484	7.9899	15.5701	4.6919
8	21.1092	8.0806	15.5443	4.7264
9	21.0912	8.1064	15.5303	4.7393
10	21.0809	8.1233	15.5243	4.739

Haar :-

Level	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
1	22.5634	3.0498		2.7243
2	20.4174	4.0116	20.0044	2.9973
3	19.7622	4.3818	19.3497	2.9985
4	19.4602	4.5481	18.9258	3.0034
5	19.3808	4.5887	18.8462	3.0023
6	19.3341	4.6028	18.7766	3.004
7	19.3176	4.605	18.7289	3.0059

8	19.2948	4.6147	18.7062	3.0055
9	19.2884	4.6182	18.6923	3.0048
10	19.2815	4.6182	18.6855	3.0041

Sampling rate = 8kHz:-

db10:-

Level	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
1	19.6826	2.7835	12.8405	3.2252
2	16.7846	4.3278	10.7026	4.8004
3	16.1497	5.3421	9.9782	5.4874
4	15.7859	5.8599	9.6135	5.7298
5	15.5983	6.1152	9.3884	6.0489
6	15.4685	6.3869	9.3372	6.5028
7	15.4038	6.5508	9.3025	6.7821
8	15.3712	6.6212	9.2748	6.913
9	15.3566	6.6561	9.2534	6.9611
10	15.3465	6.6784	9.238	6.9652

sym6:-

Level	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
1	19.5694	2.8663	12.8614	3.2875
2	16.7886	4.2868	10.7141	4.7746

3	16.0433	5.2326	9.9952	5.4575
4	15.7033	5.6783	9.6255	5.6877
5	15.4977	5.8901	9.4522	5.9381
6	15.388	6.0657	9.4032	6.333
7	15.3342	6.2278	9.3689	6.5849
8	15.3102	6.2898	9.348	6.712
9	15.3008	6.318	9.3259	6.763
10	15.2954	6.3246	9.3164	6.7554

Haar:-

Level	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
1	17.175	2.6719	12.5276	3.2174
2	15.2609	3.3673	10.2738	4.0144
3	14.7541	3.7046	9.4954	4.3115
4	14.4906	3.8707	9.2642	4.3919
5	14.3624	3.9445	9.0925	4.4948
6	14.2746	3.989	9.0247	4.6633
7	14.2316	4.0225	8.9865	4.7425
8	14.2121	4.0394	8.9597	4.7908
9	14.201	4.0428	8.937	4.81
10	14.196	4.0435	8.9321	4.8139

CONCLUSION:-

For a sampling frequency of 22kHz, no performance advantage is gained above decomposition level 5 in terms of compression factor. The same is observed for 8kHz sampling frequency at level 3.

Also at these levels, there was good clarity in speech for male and female voices.

EXPERIMENT 3:- To observe the difference between male and female speech signals and effect of threshold values on it.

PROCEDURE:-

1.Run the software for db10 wavelet at decomposition level 5 for 22kHz and 8kHz on two sets of male and female speech signals.

2.Tabulate the values of SNR and compression factor.

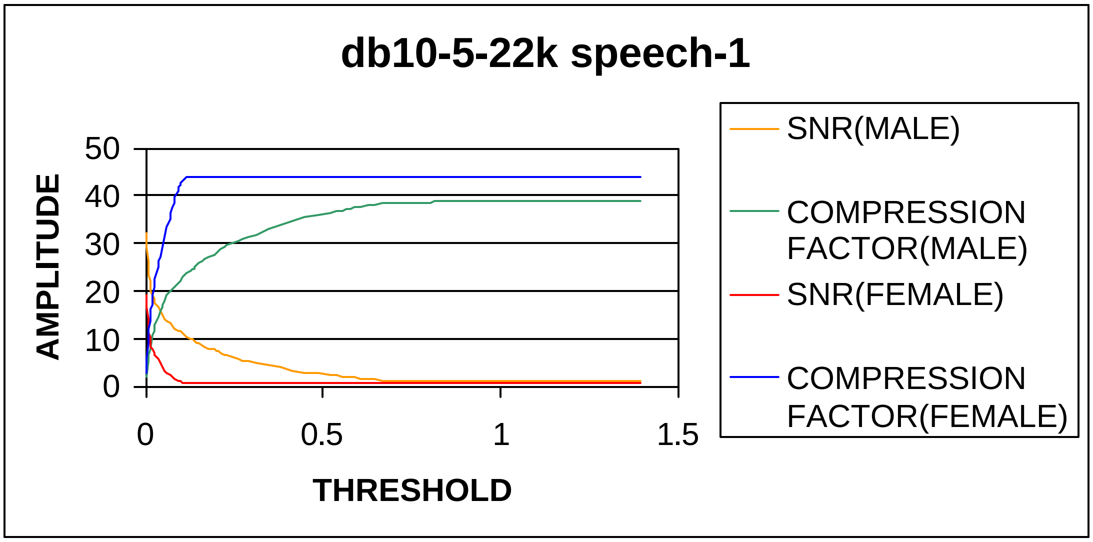
OBSERVATIONS:-

db10, Decomposition level-5, Sampling freq.-22kHz:-

Test signal 1:-

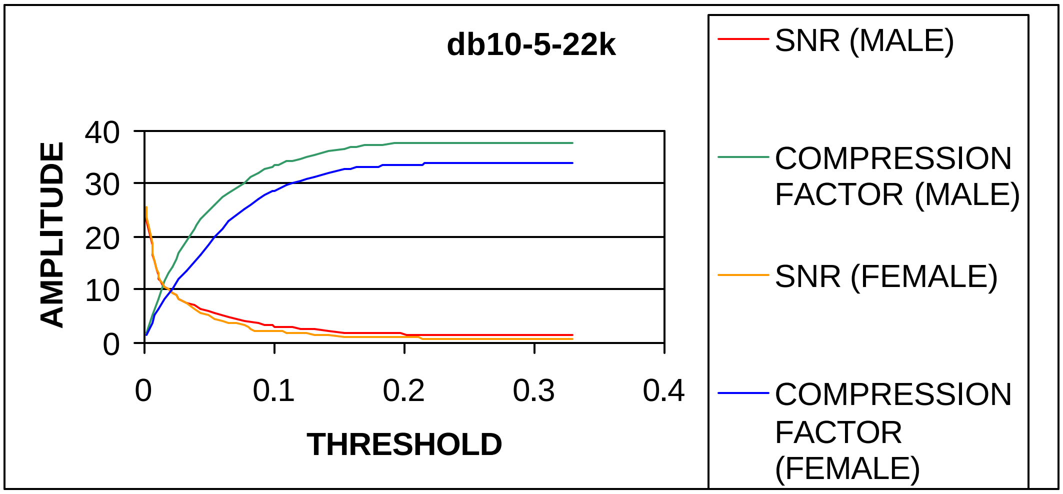
Current Threshold	Male voice		Female voice	
	SNR (dB)	Compression Factor	SNR (dB)	Compression Factor
0.0027	32.30	2.1	19.36	2.75
0.0138	21.25	8.11	9.36	14.86
0.0276	17.69	12.77	6.57	22.44
0.0414	15.70	15.94	4.80	27.29
0.0550	14.31	18.09	3.30	31.60
0.0690	13.16	19.84	2.34	35.33
0.0828	12.29	21.14	1.51	39.61
0.0966	11.52	22.31	0.996	42.51
0.1104	10.90	23.20	0.815	43.52
0.1242	10.10	24.34	0.67	44.13
0.1384	9.58	25.01	0.67	44.13
0.1658	8.49	26.57	0.67	44.13

0.1932	7.63	27.80	0.67	44.13
0.220	6.68	29.31	0.67	44.13
0.276	5.55	31.11	0.67	44.13
0.345	4.40	32.88	0.67	44.13
0.414	3.31	34.96	0.67	44.13
0.483	2.72	36.05	0.67	44.13
0.552	2.21	37.00	0.67	44.13
0.607	1.71	37.73	0.67	44.13
0.690	1.24	38.68	0.67	44.13
0.800	1.20	38.75	0.67	44.13
0.814	1.15	38.82	0.67	44.13
0.828	1.15	38.82	0.67	44.13
1.390	1.15	38.82	0.67	44.13



Test signal 2:-

Current Threshold	Male voice		Female voice	
	SNR (dB)	Compression Factor	SNR (dB)	Compression Factor
0.0022	24.71	1.78	25.54	1.5
0.011	12.9	8.45	13.03	6.25
0.022	9.56	14.48	9.50	10.27
0.033	7.72	19.17	7.38	13.44
0.044	6.40	23.2	5.80	16.76
0.055	5.49	26.2	4.60	19.98
0.066	4.84	28.34	3.81	22.92
0.077	4.22	30.29	3.22	25.04
0.088	3.69	31.91	2.27	27.11
0.099	3.27	33.24	2.35	28.60
0.110	2.99	34.06	2.05	29.69
0.132	2.48	35.58	1.61	31.38
0.154	2.07	36.55	1.25	32.65
0.176	1.75	37.39	1.04	33.31
0.209	1.56	37.80	0.94	33.54
0.220	1.56	37.80	0.827	33.77
0.275	1.56	37.80	0.827	33.77
0.330	1.56	37.80	0.827	33.77

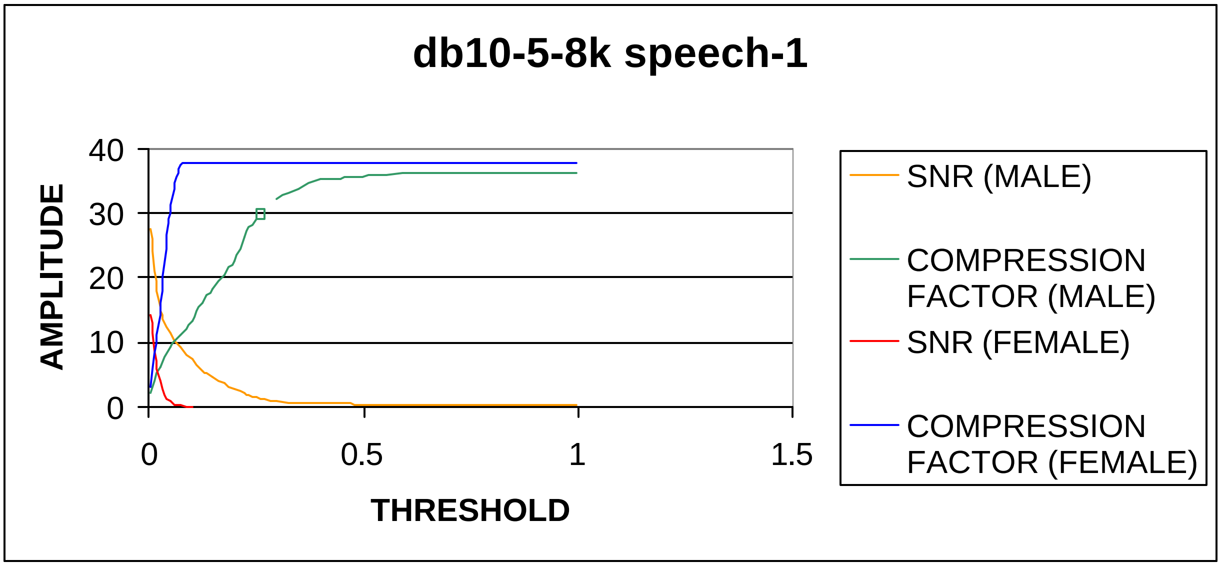


db10, decomposition level – 5, Sampling rate – 8kHz:-

Test signal 1:-

Current Threshold	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor
0.004972	27.449	2.116	14.175	3.178
0.024859	15.599	6.115	3.993	14.219
0.049718	11.565	9.210	0.881	30.04
0.074577	9.231	11.322	0.177	37.386
0.099436	7.351	13.254	0.136	37.739
0.124295	5.683	16.251	0.136	37.739
0.149154	4.659	18.253	0.136	37.739
0.174013	3.629	20.567	0.136	37.739
0.198872	2.909	22.643	0.136	37.739
0.223731	2.055	26.405	0.136	37.739

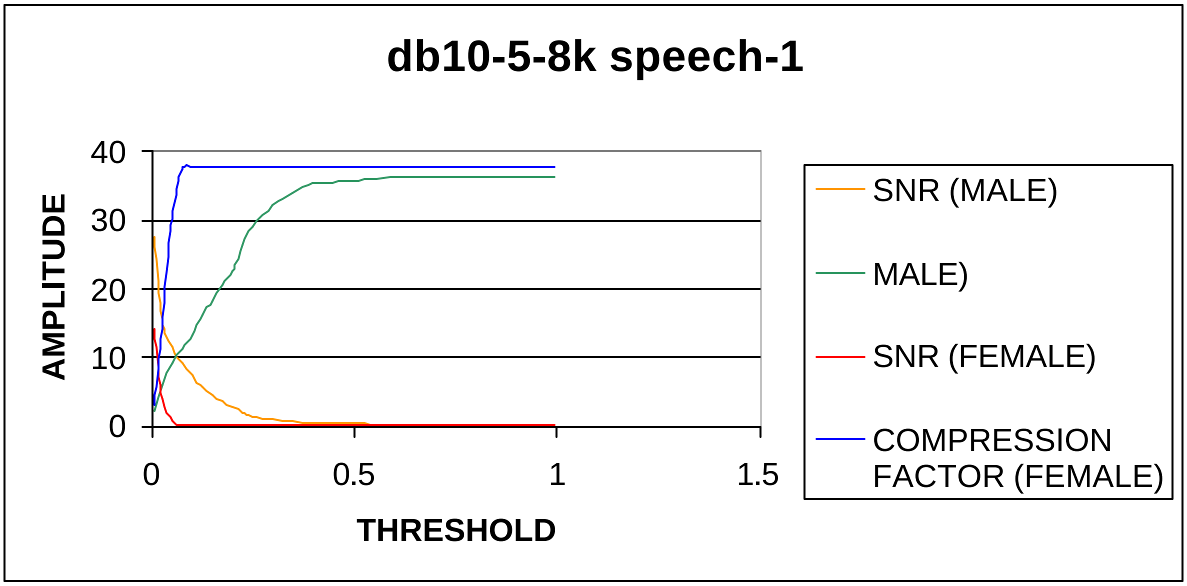
0.248590	1.502	29.023	0.136	37.739
0.298308	1.004	32.131	0.136	37.739
0.348026	0.741	33.853	0.136	37.739
0.397744	0.481	35.245	0.136	37.739
0.447462	0.452	35.401	0.136	37.739
0.497180	0.381	35.717	0.136	37.739
0.621475	0.195	36.367	0.136	37.739
0.745770	0.195	36.367	0.136	37.739
0.870065	0.195	36.367	0.136	37.739
0.994360	0.195	36.367	0.136	37.739



Test signal 2:-

Current Threshold	Male voice		Female voice	
	SNR (dB)	Compression factor	SNR (dB)	Compression factor

0.004419	17.819	2.150	18.662	1.792
0.022097	6.823	8.019	6.814	5.839
0.044194	3.320	16.565	2.657	13.747
0.066291	1.696	23.812	0.985	23.601
0.088388	0.856	29.523	0.411	28.205
0.110485	0.506	31.875	0.171	30.306
0.132582	0.278	33.336	0.094	30.851
0.154679	0.191	33.758	0.094	30.851
0.176776	0.149	33.901	0.094	30.851
0.198873	0.149	33.901	0.094	30.851
0.220970	0.149	33.901	0.094	30.851
0.265164	0.149	33.901	0.094	30.851
0.309358	0.149	33.901	0.094	30.851
0.353552	0.149	33.901	0.094	30.851
0.397746	0.149	33.901	0.094	30.851
0.441940	0.149	33.901	0.094	30.851
0.552425	0.149	33.901	0.094	30.851
0.662910	0.149	33.901	0.094	30.851
0.773395	0.149	33.901	0.094	30.851
0.883880	0.149	33.901	0.094	30.851



CONCLUSIONS:

1. It is observed that after some particular threshold value, the SNR and compression factor doesn’t change. This is because at this point all the detail coefficients are truncated to zero and only approximate coefficients remain.

2. Male voices have relatively more approximate coefficients than female voices.

3.The threshold value required for complete detail truncation depends upon the amplitude of the i/p speech signal.

8 FURTHER STUDY

This thesis was, admittedly, limited in scope. The main objective was to develop an appreciation for wavelet transforms, discuss their application in compression of human speech signals and study the effect of a few parameters on the quality of compression. The parameters studied are: Sampling frequency, type of wavelet, threshold, male/female voice.

There are a few topics that were deliberately excluded due to the limitation of our undergraduate study. Some of them are stated below. Any or all of these topics can be explored further with a view to achieving better performance.

The analysis that we undertook for wavelets includes only the orthogonal and compactly supported wavelets. The reader may find it interesting to study the effect of other wavelets on compression.

Secondly, the sound files that we tested were of limited duration, around 5 seconds. Albeit the programs will run for larger files (of course, the computation time will be longer in this case), a better approach towards such large files is to use frames of finite length. This procedure is more used in real-time compression of sound files, and was not discussed here.

Encoding is performed using only the Run Length Encoding. The effect of other encoding schemes on the compression factor have not been studied. In fact, higher compression ratios are expected with coding techniques like Huffman coding.

This thesis considered only wavelets analysis, wherein only approximation coefficients are split. There exists another analysis, called wavelet packet analysis, which splits detail coefficients too. This was not explored in this thesis.

Last but not the least, the effect of wavelet transform on voiced and unvoiced speech is different[1], and thus compression ratios ought to be different.

[1] See References, section II, #3

BIBLIOGRAPHY AND REFERENCES:-

I : Books:

1. Ainsworth WA. 'Speech recognition by machine'. London: Peter Peregrinus, 1988.

2. Ben Gold and Nelson Morgan. 'Speech and Audio Signal Processing'. John Wiley and Sons, 2000.

3. Gonzalez R.C., Woods R.E.. 'Digital Image Processing'. Massachusetts: Addison Wesley, 1988.

4. Haykin Simon. 'Digital Communication'. New York: John Wiley and Sons, 1988.

5. Haykin Simon 'Communication System'. New York: John Wiley and Sons, 1978.

6. Held Gilbert. 'Data Compression' (2nd Ed) . John Wiley and Sons, 1983.

7. Jain A.K. 'Digital Image Processing'. New Jersey: Prentice Hall Inc., 1989.

8. Owens F.J. 'Signal Processing of Speech' Macmillan Ltd.

9. Proakis John G. 'Digital Communications' .New York: Macmillan Pub. Co, 1988.

10. Rao R M and Bopardikar A S . 'WAVELET TRANSFORMS: Introduction to theory and applications'. Pearson Education (Singapore) Pte. Ltd.

11. Rudra Pratap. ‘MATLAB: A quick reference for Scientists and Engineers’. Oxford University Press-2002

12. Mitra Sanjit K. ‘Digital Signal Processing: A computer based approach’. Tata McGraw-Hill –2001

13. Salomon David. ‘Data Compression: the complete reference(2nd Ed)’. Springer Verlag, New York- 2000.

14. Watkinson John. ‘Compression in Video and Audio’.Butterworth-Heinemann, 1995.

II: Techncial papers/dissertations/tutorials :

1. *'APPLYING WAVELET ANALYSIS IN CODING OF SPEECH AND AUDIO SIGNALS FOR MULTIMEDIA APPLICATIONS'*- P.S Sathidevi and Y.Venkataramani (Department of Electronics Engineering, Regional Engineering College, Calicut, Kerala.
2. *'THE WAVELET TUTORIAL-PARTS 1 TO 4'* : Polikar Robi
3. *'Thesis: Speech Compression Using Wavelets'* : Rao Nikhil
4. *'SPEECH AND IMAGE SIGNAL COMPRESSION WITH WAVELETS'* . W.Kinsner and A.Langi
5. *"A theory for multiresolution signal decomposition: the wavelet representation,"* IEEE Pattern Anal. and Machine Intell., vol. 11, no. 7, pp. 674-693 , by Mallat, S. (1989).
6. *'MULTIRESOLUTION APPROXIMATIONS AND WAVELET ORTHONORMAL BASES OF $L^2(R)$' , by* Mallat, S.
7. *The MATLABâ wavelet toolbox help :* The MathWorks, Inc.

III: SOME USEFUL URLs:-

http://www.speech.cs.cmu.edu/comp.speech

http://www.data-compression.com/speech.html

http://isc.faqs.org/faqs/compression.faq/part1/

http://cas.ensmp.fr/~chaplais/Wavetour_presentation/Wavetour_presentation_US.html

http://www.ecs.syr.edu/faculty/lewalle/wavelets.html