OFFLINE AND REAL TIME NOISE REDUCTION IN SPEECH SIGNALS USING THE DISCRETE WAVELET PACKET DECOMPOSITION

A Thesis

Submitted to the
Graduate Faculty of Environment and Technology
University of the West of England
in partial fulfilment of the requirements for the degree of
Master of Philosophy

Mehmet Alper Oktar

October 2017

ABSTRACT

In many speech processing applications, speech has to be handled in the existence of undesirable background noise like white Gaussian noise, coloured noise, car interior noise and many other real life noises. It is vital to enhance the original speech signal using noise reduction techniques in order to get rid of any undesirable background noise. Speech enhancement aims to improve the speech quality for human listener or improve the speech signal for other speech processing algorithms. Different methods have been implemented for the suppression of background noise without decreasing the speech quality. These approaches can be classified into two major categories: single-microphone and multi-microphone methods. Although multi-microphone methods show good performance in many situations, there are possible circumstances where we are restricted to the use of a single microphone. In this context, the thesis presents a novel speech enhancement method based on discrete wavelet packet decomposition using a single microphone.

First of all, a wavelet based speech enhancement algorithm is implemented in software. A different algorithm, called wavelet packet decomposition, is implemented rather than the most common method wavelet pyramid decomposition. The next part concentrates on implementing wavelet packet decomposition method for speech enhancement on real platforms. The algorithm, implemented initially in software (Matlab), is improved by using C for less memory usage for processors with a small size memory. The final part outlines the implementation of a real time wavelet de-noising algorithm using wavelet packet decomposition method. The Xad-ML100, an audio acquisition platform, which can acquire 16 analogue audio signals from custom build audio equipment and store and play them in custom build computer is used as a real time platform*. The results are compared using output SNR values and Mean Opinion Score (MOS). It shows that the wavelet packet decomposition is not giving good results for fixed frequency noise mixed with white noise. Therefore, a new method which consists of combining a notch filter with wavelet packet decomposition, is implemented. The results of the proposed algorithm are very promising.

* The platform has been developed by Xad Communications Ltd., the company who funded this research.

Dedicated to

My beloved wife, Sema and my son, Demir Ali and
my wonderful parents, Mukaddes and Ali..

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Mokhtar Nibouche for the continuous support of my MPhil study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank my directors Yusuf Baltaci and M. Fahim Tariq for their encouragement, insightful comments and support.

I would like to thank my friends Sedat Gormus, Esra Tunc Gormus and Gokhan Koc for making so many ordinary moments, extraordinary.

I would like to thank my parents, Mukaddes Oktar, Ali Oktar and my sister, Bilge Celikoglu for their support and guidance, which gave me an opportunity to pursue higher studies. To my family Nazir Dumanli, Gonul Dumanli, Hilal Dumanli, Esma Kadizade, Ayca Dumanli Ozcan, Cihangir Celikoglu, Erdal Ozcan, Unsal Kadizade for cheering me up all the time.

Finally, to my caring, loving, and supportive wife, Sema Dumanli Oktar: my deepest gratitude. Your encouragement when the times got rough are much appreciated and duly noted. My heartfelt thanks.

Table of Contents

List of Figures

List of Tables

List of Acronyms

List of Publications

Conference Papers

- “Denoising Speech By Notch Filter And Wavelet Thresholding In Real Time”, ***Mehmet Alper Oktar**; Mokhtar Nibouche; Yusuf Baltaci, 24th IEEE Conference on Signal Processing and Communications Applications, 16-19 May 2016 Zonguldak, Turkey.

  ** This conference paper is the extract of my research project work which has solely been done (academic investigation and technical development) by Mehmet Alper Oktar. The other authors' contributions are limited to academic supervision and resources provision.*

- “Speech Denoising Using Discrete Wavelet Packet Decomposition Technique”, ***Mehmet Alper Oktar**; Mokhtar Nibouche; Yusuf Baltaci, 24th IEEE Conference on Signal Processing and Communications Applications, 16-19 May 2016 Zonguldak, Turkey.

  ** This conference paper is the extract of my research project work which has solely been done (academic investigation and technical development) by Mehmet Alper Oktar. The other authors' contributions are limited to academic supervision and resources provision.*

Journal Paper in Preparation

- Mehmet Alper Oktar, Mokhtar Nibouche and Yusuf Baltaci, "Wavelet Domain Based Real Time Speech Enhancement", IET Signal Processing.

1 INTRODUCTION

One of the vital activity of human beings is communication through speech. People acquire different ways to get information from the outside world or to communicate with other people. Speech, image and written text are the three most valuable origins of information. Speech can be highlighted as the most effective and comfortable one in many ways. Speech does not only transmit linguistic contents, but also delivers other useful information like the feelings of the speaker.

Speech processing offers practical and theoretical understandings of how human speech can be processed using both signal processing methods and knowledge from hearing sciences, phonetics, linguistics, and psychology. The rapid progress in digital signal processing makes the speed of representing, storing, retrieving and processing speech data easier and faster. Also, the development of speech processing methods started new and different application areas such as speech enhancement, speech synthesis, speech coding and speech recognition.

The main factors that affect the accuracy of the results in speech processing are based on noise and distortion. This is why modelling and removing noises and distortions are among the most important theoretical and practical considerations in speech processing. The dominant analytical tool for frequency domain analysis is the Fourier Transform (FT) based analysis. Although the Fourier Transform is a useful tool for analysing stationary signal components, where there is no change in the properties of a signal, it cannot give any information on the spectrum changes with respect to time. To overcome the problem, a modified version of the Fourier Transform, called the Short Time Fourier Transform (STFT), is used. It can represent the signal in both time and frequency using a time windowing function by determining a constant time and frequency resolution. Thus, capturing the transient behaviour of a signal, for example, is achieved by using a shorter time window which sacrifices the frequency resolution. As the real speech signal is nonperiodic and transient, conventional transforms cannot easily analyse such signals. Recently, however, wavelet transformation based methods have become very popular. Signals corrupted by noise can be filtered by powerful wavelets[1]. Wavelet basis naturally preserves both time and frequency(scale) information of the signal, providing thus a more elegant way of analysing non stationary signal.

1.1 Problem Statement

- *How to enhance noisy speech data without decreasing the speech quality in real time speech recording?*

  Background noise creates signal distortions and causes total unintelligibility of speech signals. This degradation substantially decreases the performance of speech coding and automatic speech recognition systems. Therefore, efficient noise reduction algorithms are required[2]. In order to achieve this mission, we propose to develop a speech processing method using both wavelets and thresholding techniques. This choice is driven by the non-stationary nature of speech and the ability of wavelets in dealing with this type of signals.
- *Which denosing method is suitable for developing reliable and robust real time speech enhancement system?*

  Over the years, different approaches to the problem have been developed. The approaches can be classified into two major categories: single-microphone and multi-microphone methods. Although multi-microphone methods show good performance in many situations, there are still some circumstances where we are restricted to the use of a single microphone. Spectral subtraction is the most applied single microphone algorithm for speech enhancement. While it has a great potential of removing background noise, it inserts additional artefacts known as the musical noise. Other approaches using both basic and iterative Wiener filtering and comb filtering have been proposed in recent years as another mean of enhancing corrupted speech. Donoho proposed wavelet thresholding as an effective method in speech enhancement especially for additive white noise.
- *How to implement a real time speech denosing algorithm with a minimum delay?*

Although the application of wavelet thresholding for speech enhancement has been applied in many works [3] [4], there are several issues yet to be resolved for a successful application of the method to real time speech signals degraded by various noise types. The speech information often needs to be processed while the audio stream becomes available, making the complete task even more challenging[5].

- *How to implement the denoising algorithm that can be ported to a low memory battery operated mobile device?*

1.2 Research Objectives

The main objective of this research is to devise a speech enhancement method based on wavelet theory which cleans noisy signal in effective manner, without decreasing the speech quality in real time using a single microphone in the following ways:

1. Implement an offline speech enhancement algorithm using discrete wavelet packet transform instead of the traditional wavelet tree decomposition. To validate this initial work, different speech and noise types are denoised using different wavelet families, wavelet types and wavelet decomposition levels.
2. Evaluate the results using Signal to Noise Ratio (SNR) and the Mean Opinion Score (MOS), respectively.
3. Port the algorithm to a suitable hardware platform for less memory usage for low memory processors. Develop an efficient implementation that uses limited memory resources for this purpose. This will make the algorithm more suitable for hardware platforms with limited resources.
4. Devise a custom build audio equipment which real time noise enhancement algorithm can be ported.
5. Implement real time speech enhancement algorithm using a custom build equipment which the latency results is in the tolerable limits for real time processing.
6. Design a Graphical User Interface(GUI) for user friendly speech enhancement which gives flexible settings to the end user to obtain the enhanced speech output. Different types of noises are used in order to assess the effectiveness of the method.
7. Implement a new method to overcome the problem of denoising fixed frequency and white noise combination.

1.3 Scope

This thesis describes the development of a wavelet based speech enhancement system to process speech corrupted with various amounts of white Gaussian noise (signal to noise ratios vary between 0db to 15db). The system has been tested with real life noises such as: pink noise, car engine noise and F16 noise. Six different utterances from a noisy speech corpus (NOIZEUS)[6] library and three different speeches with 6 different noise types from SpEAR Database[7] were used to evaluate the wavelet packet decomposition thresholding algorithm. Ultimately, the algorithm is ported to a hardware platform for real time applications and the problem of white noise and fixed frequency noise combination is addressed using a modified wavelet algorithm.

1.4 Equipment and Materials

The following tools were used in this research:

- Codeblocks IDE is used for coding and testing purposes[8].
- Glade Interface Designer is used for GUI design[9].
- Matlab is used for simulation purposes[10].
- MathType is used for developing graphs and bar-charts[11].
- NOIZEUS[6] and SpEAR[7] speech databases are used for speech enhancement algorithm experiments.

1.5 Organisation

In chapter two, speech and speech production are described in details. Also, different auditory models are explained as well as noise in audio communications. In chapter three, past and current research in the area of speech enhancement is described. Single channel speech enhancement methods are analysed and compared using a comparison table. Most common methods, such as Comb filter, adaptive filters, spectral subtraction and Wiener filtering, are analysed in this chapter. In chapter four, the wavelet based approach to speech enhancement is discussed. The basic concepts of the classic wavelet transform and its relationship to the Fourier transform are also introduced. Moreover, the implementation of the discrete wavelet transform is explained. In chapter five, the wavelet thresholding method is applied to the actual noisy speech data using Matlab and C. The results are then analysed using objective measures such as signal to noise ratio(SNR) and subjective measures such as informal listening tests. In chapter six, the real time wavelet denoising algorithm using a custom build audio equipment called Xad-ML100 is implemented. The wavelet algorithm is applied to fixed frquency noise and white Gaussian noise and the results are improved by combining notch filtering with the wavelet denoising algorithm. The results are analysed using objective and subjective measures. Finally, in chapter seven, the thesis summary and conclusions are presented.

1.6 Contributions of the Thesis

The main contributions of this thesis are:

- Literature review of important background knowledge for developing a noise reduction algorithm in speech signals is introduced.
- A new discrete wavelet packet algorithm is implemented for the speech enhancement instead of the traditional wavelet decomposition tree method. Although discrete wavelet packet algorithm is commonly used for image processing, it is rarely used for speech processing. This approach, though time consuming, provides finer wavelet coefficients at higher levels of decomposition and consequently provides a more efficient way of calculating the threshold value.
- Different wavelet families, thresholding methods and levels are experimented to evaluate the proposed wavelet packet decomposition algorithm using six different utterances from a noisy speech corpus. The results have been evaluated using objective and subjective measures, which are in this case the SNR and the MOS, respectively. The discrete wavelet packet algorithm performed well and produced good quality speech signals.
- A new and efficient implementation that uses limited memory resources has been developed. This makes the algorithm more suitable for mobile hardware platforms with limited resources such as memory, speed and energy.
- A new GUI has been designed to allow users to implement speech enhancement through graphical icons and visualise the improvement before and after the enhancement algorithm.

- The results shows that the proposed wavelet thresholding algorithm performs far better in white Gaussian noise and pink noise when dealing with different types of noise signals, such as, white Gaussian noise, pink noise, f16 noise, burst noise, factory noise and Volvo noise.
- The implementation of the real time wavelet enhancement algorithm using the wavelet packet decomposition method is outlined using a custom build computer, Xad-ML100. After the experiments, our real time wavelet enhancement algorithm gives less than 10 milliseconds latency for different combinations of noise types, wavelet type, wavelet level of decomposition and thresholding method.
- A new combination of noise reduction methods is introduced to overcome the fixed frequency noise mixed with white Gaussian noise. Notch filter is applied prior to applying the wavelet enhancement algorithm to remove the fixed frequency noise. The fixed frequency noise actually misleads the thresholding value calculation and confuses thus the wavelet enhancement algorithm. Experimental results show that the combination of Notch filtering with the wavelet enhancement algorithm is successful in dealing with fixed frequency noise when combined with white Gaussian noise in real time.

2 BACKGROUND

2.1 Speech

Speech is the vocalised form of human communication. According to information theory, speech can be represented in terms of either its message content, or information. An alternative way of characterising speech is in terms of the signal carrying the message information, that is the acoustic waveform. It can be represented phonetically by a finite set of symbols called the *phonemes* of the language. The number of phonemes is between 32 and 64 for most languages and they have a unique appearance in the speech form. These differences result from the distinctively different ways that these sounds are produced.

Figure 2.1 shows the entire process of producing and perceiving speech from formulating a message in the brain of a talker, to the creation of the speech signal, and finally to the understanding of the message by a listener. The process starts in the upper left as a message represented somehow in the brain of the speaker. The second step consists of generating a language code and converting text symbols to phonetic symbols along with continuous information and stress. In the third step, the neuromuscular system moves the velum, jaw, teeth, lips and tongue in a manner that is consistent with the sounds of the desired spoken message, including the desired degree of emphasis. The final step in the process involves the vocal tract system. This system creates necessary sound sources and appropriate vocal tract shapes over time to generate an acoustic waveform.

The result of moving from speech production to speech perception is the encoding of the message that can be effectively transmitted by acoustic wave propagation and robustly decoded by the hearing mechanism of a listener.

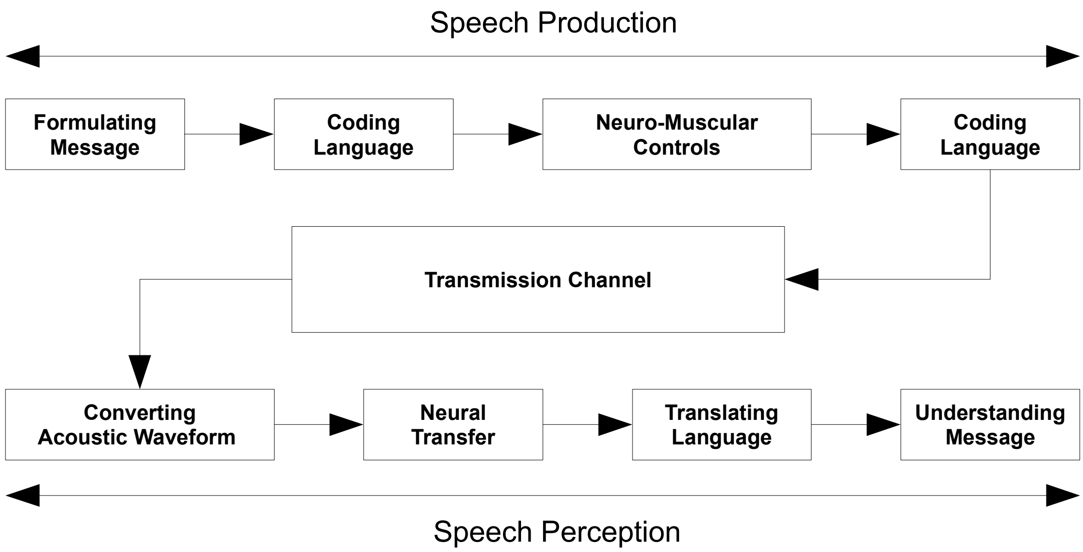

Figure 2.1 : The Speech Production and Perception Process

The speech perception includes many steps such as the transmission of the speech signal to the ear and the understanding of the speech signal content[12]. Initially, the acoustic waveform is converted to a spectral representation. Then, these spectral features are converted into sound features that the brain can decode and process. Sound features are then converted into a set of sentences, phonemes and words. Finally, the meaning of the message is understood and the speech perception process is complete.

2.1.1 Auditory Models

In which way the brain handles the extracted patterns is largely unknown. Many studies have presented how humans perceive tones and bands of sound [13], [14]. Many auditory models that simulate the functionality of the human ear have been generated based on that knowledge [13-16]. Several computational models of the IHC have been proposed.

2.1.1.1 Meddis IHC Model

One of these models is the Meddis inner hair cell model. The Meddis model is a widely acknowledged in-depth computer model of the human inner hair cell [17]. IHC are held in the organ of Corti which is located on top of the basilar membrane, inside the cochlea and can be thought of as the body's microphone. The mechanical stimulus in the cochlea is transferred by the hair cells to fire the auditory nerve cells. Transmission is completed between three reservoirs in a re-uptake and re-synthesis process loop as seen in Figure 2.2.

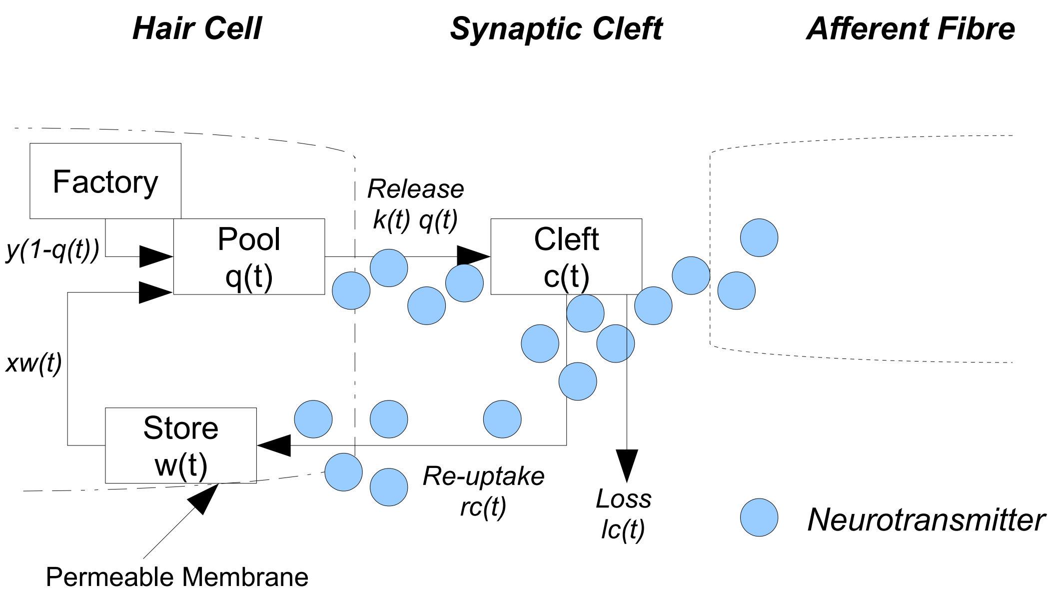

Figure 2.2: The Meddis Inner Hair Cell Model

The transmitter factory is the first reservoir which discharges neurotransmitters at the hair cell boundary and transfers them to the free transmitter pool. The changes in the permeability of the cell membrane handles the quantity of neurotransmitters released from the pool into the cleft. This oscillates as a part of the intra-cellular voltage which is linked to the instantaneous mechanical stimulus amplitude. Some neurotransmitters are lost since there is a diffusion in the cleft and some are taken back up into the cell. The post synaptic afferent fibre of an auditory nerve cell is stimulated by the remaining transmitter in the cleft. The transmitters removed from the cell are initially processed again and then stored in a third reservoir in preparation for delivery to the free transmitter pool.

2.1.1.2 Lyon's Model

A model of an analogue electronic cochlea is established by Richard F. Lyon. The model is based on the knowledge of how the cochlea works [18]. His suggestion was to design the fluid-dynamic wave medium of the cochlea by a cascade of filters depending on the detected properties of the medium as shown in Figure 2.3[19].

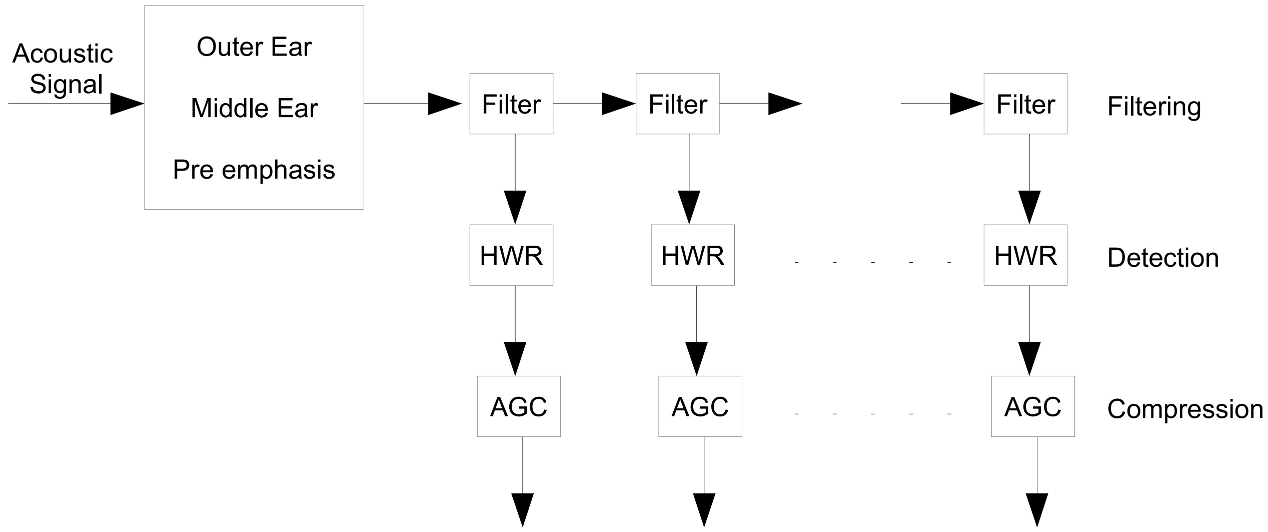

Figure 2.3: Lyon's Model Cascade Filter, Copyright © 2016, IEEE

2.1.1.3 Auditory Image Model

The auditory image model(AIM) is a computational model that simulates human auditory processing[20]. It contains various alternative modules containing the phases of processing by the peripheral auditory system as shown in Figure 2.4. The first phase is called middle ear filtering. The middle ear filtering is a simple linear filter that enhances middle frequencies. The second phase is spectral analysis. The spectral analysis can be accomplished with a functional (gammatone) or a physiological auditory (nonlinear transmission) filter. Gammatone filter bank is a constant bandwidth filter bank. The gammatone auditory filter can be explained by its impulse response as shown in equation 2.1.

$$y_{tone}(t) = at^{n-1}e^{2\pi bt}\cos(2\pi f_c t + \phi) \quad \text{for } t>0 \tag{2.1}$$

where the parameters a and b determines the duration of the impulse response and n the filter's order. The product of this stage is the estimated Basilar Membrane Motion (BMM) received in the presence of an input signal. The third stage is the neural encoding stage, which changes the BMM into neural activity pattern (NAP). Two modules are used for producing the NAP: a bank of meddis Inner Hair Cells(IHS) and a bank of two dimensional adaptive threshold units, which rectify and compress the BMM. To complete the stage, we apply adaptation in time and suppression across frequency are applied. The last stage, time-interval

stabilisation, outlines the momentary activity at the output of the NAP stage depending on the idea that
periodic sounds give rise to static perceptions by human listeners.

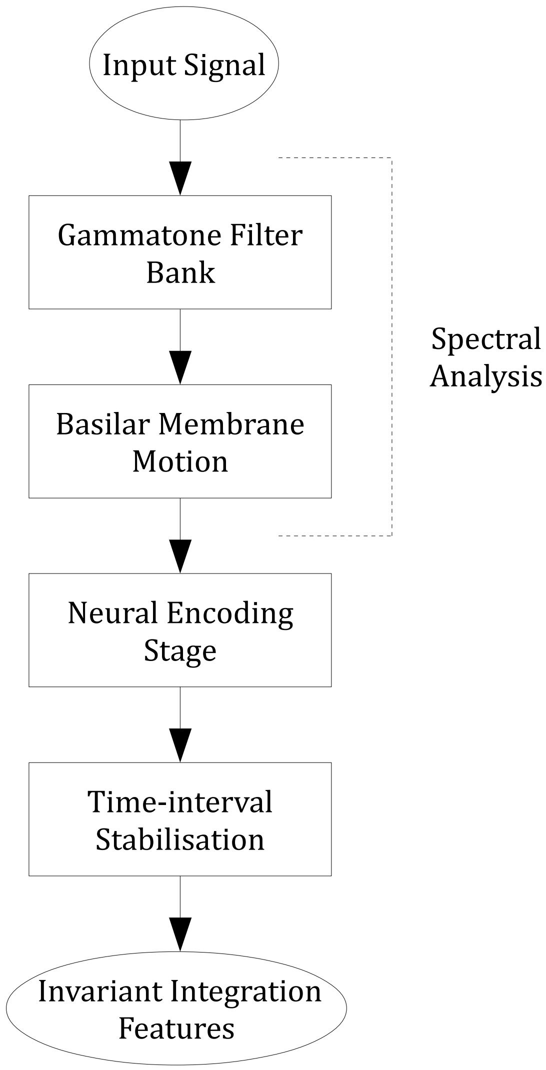

Figure 2.4: Phases of Auditory Image Model

2.2 Noise In Audio Communication

Recording, playback, analysis, synthesis or transmission of speech signals are performed by electronic systems to achieve audio communication. Noise influence must be carefully considered while designing a system for any of these purposes. There are different types of noises and distortions which will be explained later in the thesis.

Number of signal processing concepts exist that can assist in diminishing the effect of noise and as such enhance the quality or intelligibility of the speech signal. Digital signal processing offers a number of powerful tools to specific types of noise corruptions. Different noise reduction methods will be discussed in the next chapter.

2.2.1 Noise Chain

The quality of a speech signal may be deteriorated by a wide range of influential factors during transmission, acquisition and generation of speech. External interferences such as background noise in the recording, echoic effects, non-linear distortions introduced by analogue electro-acoustic devices or amplifiers can be included to the noise chain[21]. Potential sources of noise and distortion in a speech communication systems can be highlighted and categorised as shown in Figure 2.5.

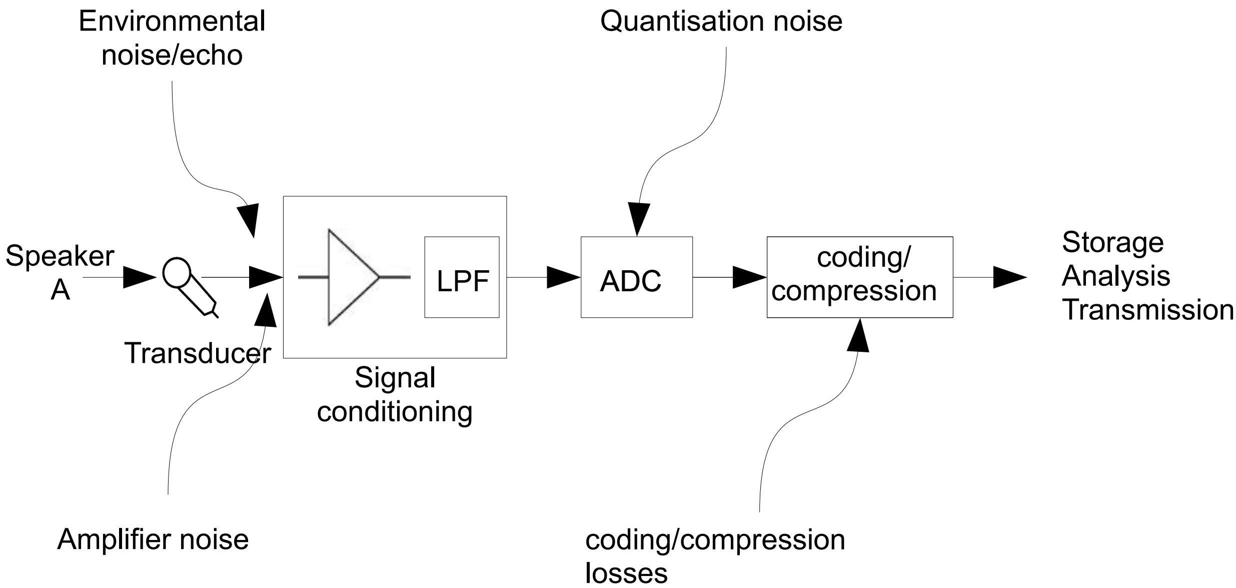

Figure 2.5: Speech signal acquisition stages

Speaker A is connected to listener B through a universal speech communication system as shown in Figures 2.5 and 2.6 respectively. While Figure 2.6 shows a mechanism of speech play back stage, Figure 2.5 represents a universal speech acquisition system.

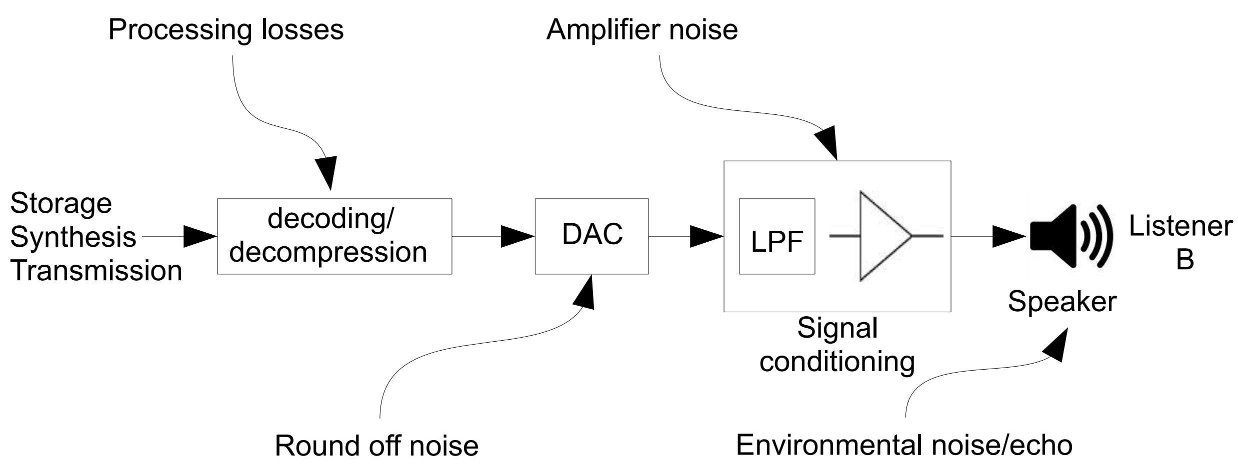

Figure 2.6: Speech play back stages

The distortion of a speech signal is caused by environmental noise, echo feedback, amplifier noise, quantization noise and coding-compression noise during the acquisition stage[22]. Signal quality degrades quickly if the distortion becomes too dominant [23]. In this thesis, we mainly concentrate on distortions in the signal acquisition stages. In particular, the distortion introduced by environmental noise, echo feedback, amplifier noise and quantisation noise.

2.3 Summary

In this chapter, speech production is described using the physical features of human anatomy. Also, the production of a sampled speech signal is modelled by a discrete-time system model called the *source-filter* model of speech production. Then, the process of sound perception is explained from the outer ear to the cochlea, where the mechanical movement is converted into stimuli perceived by the brain. Some auditory models that simulate the functionality of the human ear are introduced to explain how humans perceive tones and bands of sound. Finally, different type of noise signals in audio communication are explained. These noise signals occur during recording, playback, analysis, synthesis or transmission of speech signals. Digital signal processing offers a number of powerful tools to specific types of noise corruptions. Different noise reduction methods will be discussed in chapter 3.

3 SPEECH ENHANCEMENT

3.1 Introduction

Speech signal degradation can originate from different sensors (microphones) and their placement, acoustic non-speech and speech background, as well as from channel and reverberation effects. It exists different techniques for enhancing speech signal to cater for distortions caused by both wideband and narrow band background noise, clicks, and other non-stationary interferences. These are explained in this chapter. It is assumed that the features of noise change slowly, therefore noise can be identified in terms of mean and variance.

The goal of speech enhancement is to enhance quality and intelligibility. Weaker and lower energy is important for intelligibility. The obstruents which are the speech sounds formed by obstructing airflow such as [k], [dʒ], or [f] have low energy and are easily masked in noise, more than vowels. Unfortunately, they are the first to be lost and the most difficult to recover in noise. However, sections of speech spectral transitions are very crucial for good intelligibility. Speech enhancement methods often try to improve speech intelligibility beyond easy improvement in SNR.

Speech enhancement approaches vary remarkably depending upon the type of degradation. Speech enhancement techniques can be divided into two basic categories based on speech acquired either using a single microphone or multiple microphone sources. Our focus is on single channel speech enhancement methods since single channel signal(one microphone) is available for measurement or pick up in real environments.

3.2 Single Channel Speech Enhancement Approaches

Speech enhancement algorithms try to improve the performance of communication systems when a signal is corrupted by noise[24]. The main objective of speech enhancement is to reduce the additive noise while improving the speech quality and intelligibility[25]. Different types of noises may need different enhancement approaches. Noise may be periodic, impulsive or continuous and its amplitude may change across frequency; (e.g., hum noise from AC power lines or machinery). Consequently, to deal with different problems, there are different types of speech enhancement methods, each with its own advantages and limitations.

Noise signals that are transiently short can generally be treated as impulsive while noise signals that are localised in frequency can generally be treated as harmonic. It is possible to identify and clean impulsive and harmonic noises, as the majority of time-frequency representations of a speech signal are not damaged. A car engine noise, a motor noise created by a motorised camera and a hum noise in an electrocardiograph are some examples of such noise signals. In what follows, different techniques for speech enhancement, including their advantages and limitations, are reviewed.

3.2.1 Comb Filter

3.2.1.1 Background

A comb filter has notch frequencies that have equally spaced zeros as shown in Figure 3.1. A comb filter can remove noise by adjusting the notch frequencies to that of the harmonic frequencies of the periodic noise signal. Moreover, the desired signal can be preserved by adjusting the notch gain appropriately, which is basically implemented by changing the gain of the comb filter for chosen unwanted frequencies[26].

A comb filter adds a delayed version of a signal to itself creating constructive and destructive interferences. Feedback and feedforward types are the two different types of comb filters and the names show the direction of the delayed signal before they are added to the input.

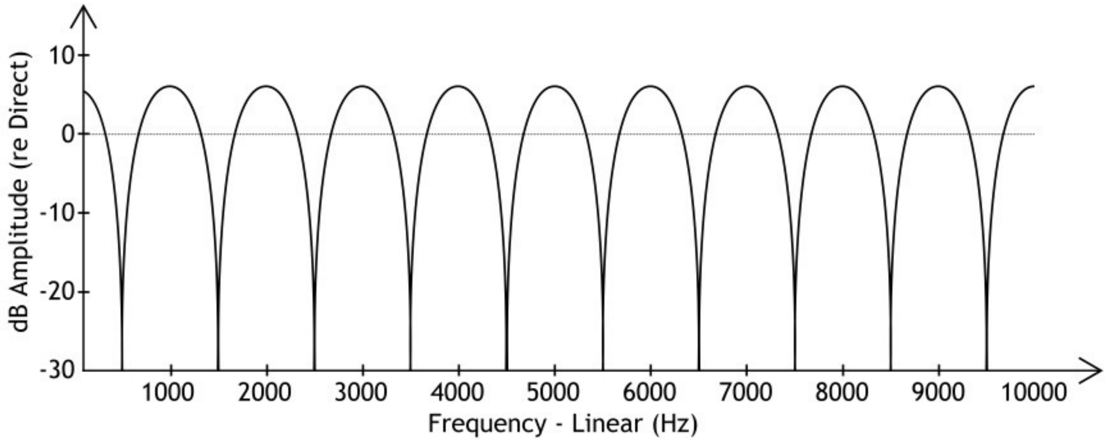

Figure 3.1: Comb Filtering with linear frequency, Copyright © 1979, IEEE

3.2.1.2 Feedback Comb Filter

The main form of feedback comb filter is shown in Figure 3.2 and is described by the difference equation:

$$y[\mathrm{n}] = \mathrm{x}[\mathrm{n}] + gy[\mathrm{n} - \mathrm{M}] \qquad (3.1)$$

where M is the delay length and g is the scaling factor, which is applied to the delayed signal. x[n] and y[n] represent the input and the output signals, respectively.

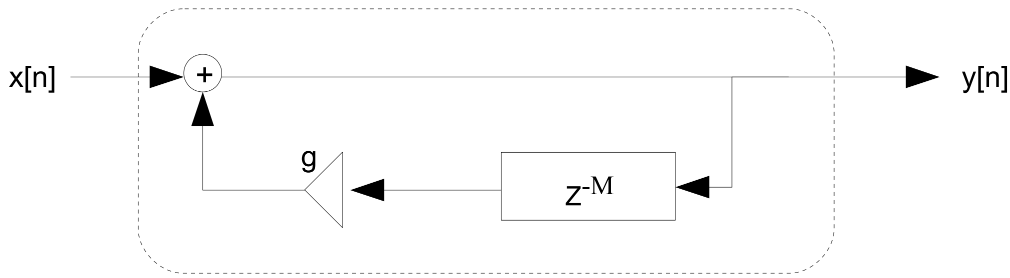

Figure 3.2: Structure of a feedback comb filter

3.2.1.3 Feedforward Comb Filter

The general form of feedforward comb filter is shown in Figure 3.3 and is described by the difference equation:

$$y[n] = x[n] + g\,x[n-M] \qquad (3.2)$$

where M is the delay length and g is the scaling factor which is applied to the delayed signal.

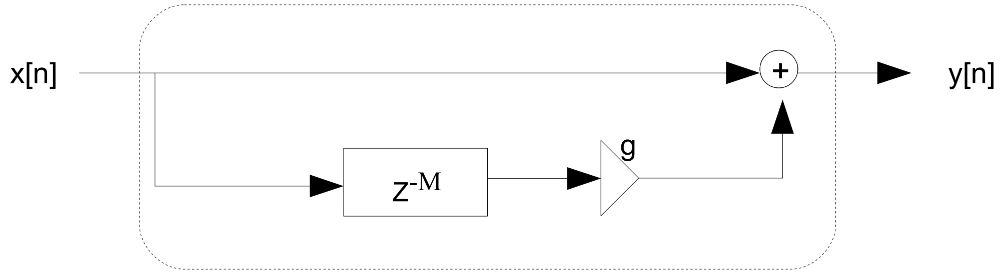

Figure 3.3: Structure of feedforward comb filter

Incrementing the number of poles can achieve larger amounts of stop-band rejection. However, incrementing the number of poles increases the filter complexity and processing time. This is why, a comb filter cannot enhance most of real life speech signals[27].

A Comb filter is a FIR recursive system, therefore it has linear phase and guaranteed stability. Moreover, it is multiplier-free and it only requires two additions which leads to low complexity[28]. A comb filter is capable of producing significant noise reduction for voiced speech, but only at the cost of reduced intelligibility. This is because voiced speech is not purely periodic[29].

3.2.2 Adaptive Noise Enhancement

3.2.2.1 Background

A signal corrupted by additive noise can be enhanced by passing it through a filter that suppresses the noise while keeping the signal relatively unchanged. Direct filtering can be either fixed or adaptive. The fixed filter design needs a priori knowledge of both the speech signal and the noise signal[30]. On the other hand, little or no a priori knowledge of the detailed properties of the signal and the noise are required in the case of adaptive noise filtering. However, a highly correlated noise component of the input signal is required as a reference noise. Since most speech signal applications do not have a reference noise signal, single channel Adaptive Noise Cancellation (ANC) method is proposed for speech signals[31]. This method is applicable in scenarios where the levels of noise rejection are often attainable, which would normally be difficult or impossible to achieve by direct filtering. The process of single channel adaptive noise enhancement is illustrated in Figure 3.4.

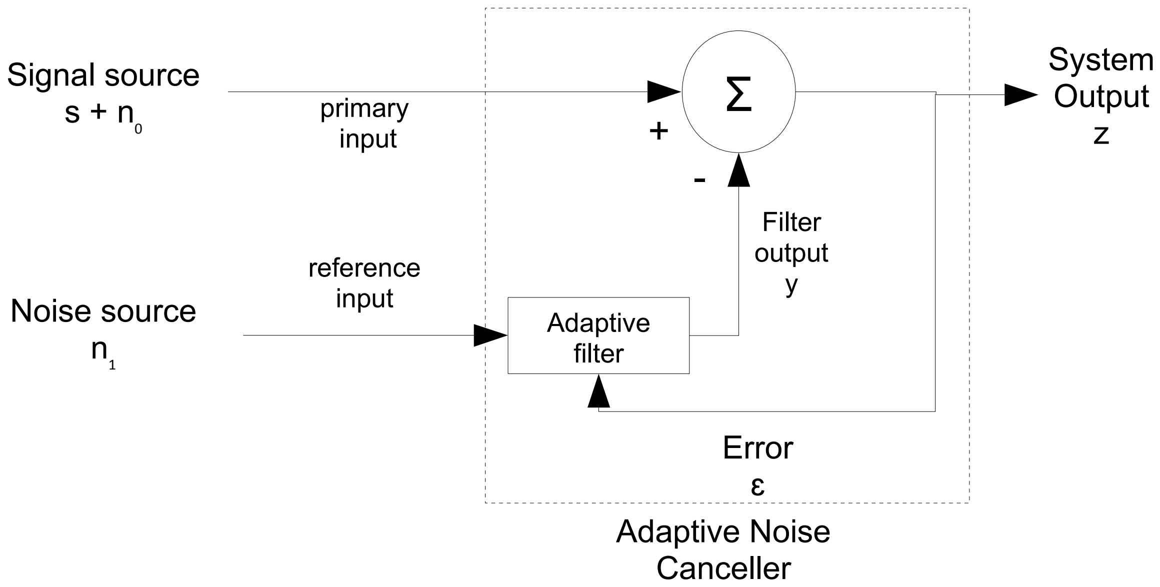

Figure 3.4: Single Channel Adaptive Noise Enhancement, Copyright © 1975, IEEE

A noise n_0 uncorrelated with the signal is transmitted with a signal s to the adaptive noise canceller as shown in Figure 3.4. A noise n_1, uncorrelated with the signal, but correlated with the noise n_0, is sent to the adaptive filter as a reference noise. The system output is then the signal alone, after the filter output is subtracted from the primary input. The nature of the application changes the error signal used in an adaptive process. The main objective in noise cancelling systems is to produce a system output "z = s + n0 – y" and this is achieved by feeding the system output back to the adaptive filter[33].

3.2.2.2 Adaptive Filters

An adaptive filter readjusts itself to changes of input signals according to a given algorithm[34]. The algorithm modifies the coefficients according to an error signal to enhance the performance of the filter. Adaptive filters are being widely used for telephone echo cancellation[35].

3.2.2.3 Adaptive Algorithms

There are many adaptive algorithms used to modify the coefficients of the digital filter in order to achieve the intended response as accurately as possible. Some of the main algorithms are explained in this chapter.

3.2.2.3.1 Least Mean Square Algorithm

The Least Mean Square (LMS) algorithm is an adaptive algorithm introduced by Widrow and Hoff in 1959[36]. It uses a gradient-based method of steepest decent in which the filter is only altered based on the error at the current time. LMS forms an iterative procedure that achieves successive corrections to the weight vector in the direction of the negative gradient vector. From the method of steepest descent, the weight vector $w[n]$ equation is given by:

$$w[n+1] = w[n] + \frac{1}{2}\mu\left[-\nabla\left(E\left\{e^2[n]\right\}\right)\right] \tag{3.3}$$

where n represents the current input sample, μ is the step-size parameter which controls the convergence characteristics of the LMS algorithm, ∇ is the gradient operator, E is the expected value and $e^2[n]$ is the mean square error between the output y[n] and the reference signal d[n]. The mean square error is defined as:

$$e^2[n] = \left[d^*[n] - w^H[n].\,x[n]\right]^2 \tag{3.4}$$

3.2.2.3.2 Recursive Least Squares Algorithm

The main concept of an adaptive Recursive Least Squares(RLS) filter is that the coefficients are regularly altered during the filtering process. The RLS filter is an finite impulse response(FIR) filter of length M with coefficients w_k, where k=0, 1, 2, ..., M-1. The input $x[n]$ is filtered to produce the output $y[n]$. The filter coefficients are updated at each step using an error $e[n]$ as shown in equation 3.5.

$$e[n] = d[n] - y[n] \tag{3.5}$$

where $d[n]$ is the desired response as shown in Figure 3.5.

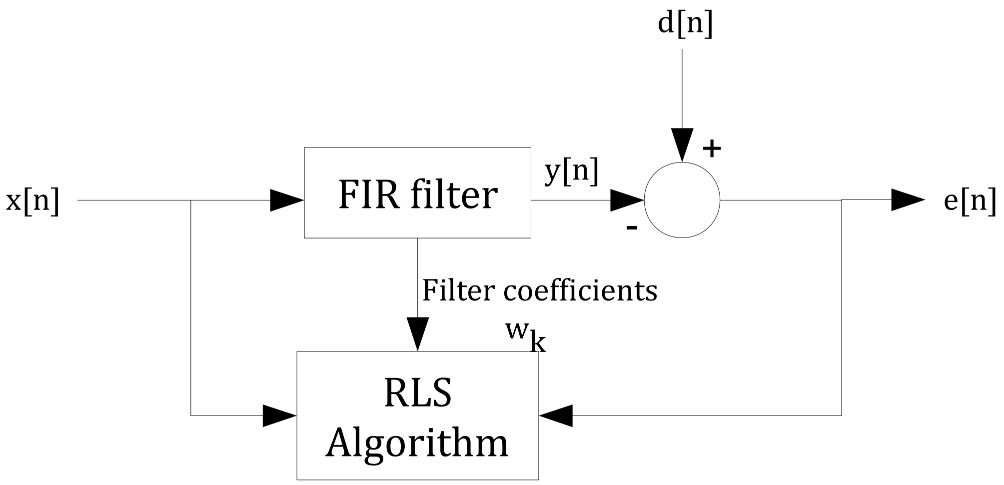

Figure 3.5: RLS Filter Structure

The total squared error $\varepsilon[n]$ at the n[th] iteration is defined as[37]:

$$\varepsilon[n] = \sum_{i=0}^{n} e^2[i] = \sum_{i=0}^{n} \left(d[i] - y[i]\right)^2 \tag{3.6}$$

Both RLS and LMS filters are FIR filters, with M coefficients. The input stream is passed through the filter to generate the output stream. While the input signals are acknowledged deterministic in RLS derivation, they are considered stochastic in LMS derivation. The RLS filter converges faster than LMS filter under most conditions. On the other hand, an RLS filter has more computational complexity than an LMS filter and it is numerically unstable in fixed point arithmetic. The major advantage of these filters is they do not require prior knowledge of the signal or noise characteristics as it is used in some other filters.

3.2.3 Spectral Subtraction

The power or the magnitude spectrum of a signal in additive noise is restored through subtracting the estimate of the average noise spectrum from noisy spectrum using spectral subtraction methods[38]. When the signal is missing and only the noise is present, the noise spectrum is usually predicted and updated from the periods. It is assumed that the noise is stationary and that the noise spectrum is not altered much in between the update periods. An estimate of the magnitude spectrum is merged with the phase of the noisy signal for recovery of time-domain signals and converted to the time domain using an inverse Fourier transform[39].

In some applications, the noise is accessible on a separate channel in addition to the noisy signal and it is possible to restore the signal by subtracting an estimate of the noise from the noisy signal. However, the noisy signal is the only signal available in many applications. It is impossible to cancel the random noise completely in these situations, however, it is possible to diminish the average effects of the noise on the signal spectrum[40]. The magnitude of the mean and the variance of the signal spectrum increases with the effect of an additive noise.

Subtracting an estimate of the noise spectrum mean value from the noisy signal spectrum can remove the increase in the mean of the noise spectrum. However, the random fluctuations of the noise causing the increase in the variance of the signal spectrum cannot be cancelled out.

Spectral subtraction is relatively uncomplicated in terms of computational complexity[41]. However, spectral subtraction can create non-linear processing distortions caused by random variations of the noise spectrum. This distortion creates a musical tone noise due to its narrowband spectrum. The ability of the algorithm to reduce the noise variations and remove the processing distortions can increase the success of spectral subtraction. It has been suggested that the results can be improved by over-subtraction, in other words, subtracting more than the average noise value in spectral subtraction applications[42]. Wiener filters outperform the spectral subtraction due to the fact that spectral subtraction handles too little prior information. Also, the benefits of spectral subtraction lessen as noise levels in the order of 0 dB are approached and exceeded. Results shows that under high noise conditions, the degradations are explicit in the noise estimate, which are broadly believed to have by far the greatest influence on spectral subtraction performance[43].

3.2.4 Wiener Filtering

Wiener filtering is a classical signal estimation technique that has been implemented primarily on one-dimensional continuous signals based upon the continuous Fourier transform[44].
The Wiener filter is a popular technique that has been used in wide range of applications such as echo cancellation, signal restoration, system identification and linear prediction. Wiener theory presumes that the signals are stationary processes.

Wiener solved the least mean square error (the continuous-time estimation) problem in his classic work on interpolation, extrapolation and smoothing of time series[45]. More practical use of Wiener theory for digital signal processors is achieved by extending it from the continuous time form to the discrete time form. Infinite-duration Impulse Response (IIR) and Finite-duration Impulse Response (FIR) filters are two types of Wiener filtering. While the formulation of a FIR Wiener filter leads to a set of linear equations and has a closed-form solution, the formulation of an IIR Wiener filter leads to a set of non-linear equations. FIR Wiener filters are explained in this section because of their computational simplicity, stability and practicality[46].

An FIR Wiener filter is represented by the coefficient vector $\overline{\mathrm{W}}$ and the input vector $\overline{\mathrm{X}}$ as[47]:

$$\bar{W} = [w_0\, w_1\, w_2 \,\ldots\ldots\ldots\ldots\ldots\, w_{N-1}]^T \tag{3.7}$$

where T denotes the transpose of a matrix.

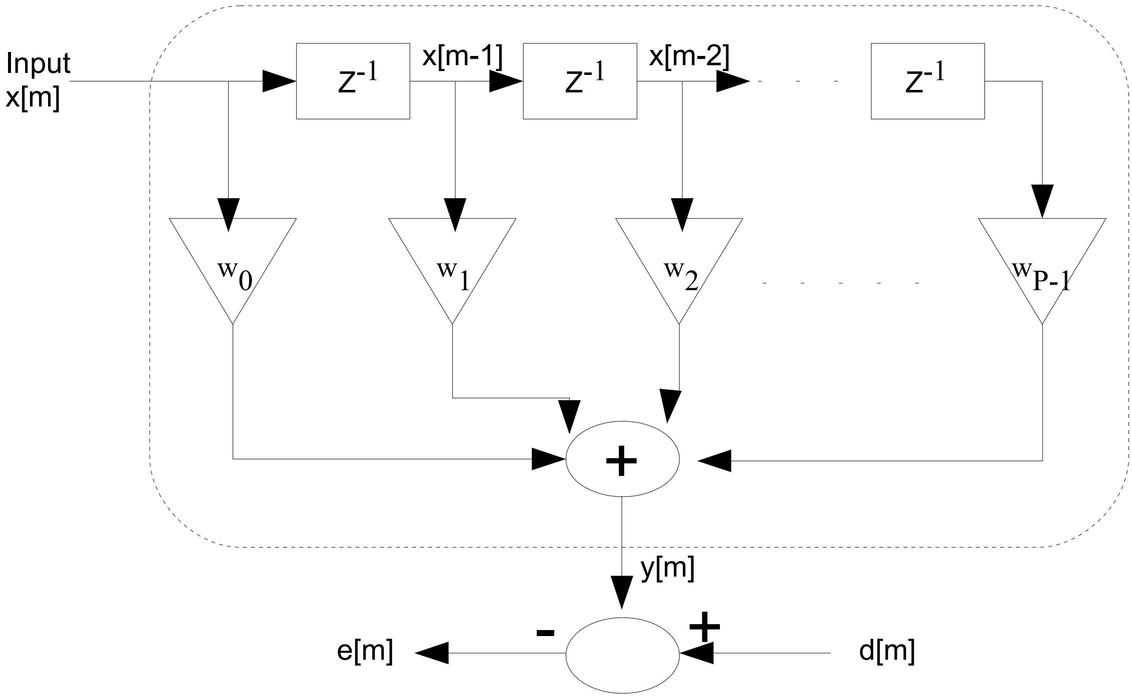

Figure 3.6: FIR Wiener Filter

The input–output relation of the filter is given by:

$$
\begin{aligned}
y[m] &= \sum_{k=0}^{P-1} w_k x[m-k] \\
y[m] &= w^T x
\end{aligned} \tag{3.8}
$$

where *m* is the discrete-time index. The P initial input signal samples [x(−1), . . ., x(−P−1)] are assumed either known or set to zero. The alternative and equivalent forms of a convolutional sum and an inner vector product define the filtering operation in equation 3.8.

The Wiener filter error signal e[m] is defined in Equation 3.9 as the difference between the desired signal d[m] and the filter output signal y[m].

$$
\begin{aligned}
e[m] &= d[m] - y[m] \\
e[m] &= d[m] - w^T x
\end{aligned} \tag{3.9}
$$

The Wiener filter error e[m] depends on the filter coefficient vector w for a desired signal d[m] and a given input signal x[m] as shown in equation 3.9. The filter coefficients are calculated in a way to decrease an average error cost function, such as the average absolute value of error $E[|e[m]|]$, or the mean square error $E[e^2[m]]$. The optimality and the computational complexity of the solution depends on the choice of the error function.

The least mean square error (LSE) between the filter output and the desired signal is the objective criterion in Wiener theory. The LSE criterion is optimal for Gaussian distributed signals and causes a linear and closed form solution for FIR filters as shown in what follows. The Wiener filter coefficients are acquired by minimising an average squared error function $E[e^2[m]]$ with respect to the filter coefficient vector ***w.***

Wiener filtering is one of the most widely used tools in signal processing especially for signal denoising because of its well documented implementations such as LMS, RLS, Kalman filters and easy implementation in real time. On the other hand, Wiener filter has some disadvantages as it gives a fixed frequency response at all frequencies, needs to estimate both the power spectral density of the clean signal and the noise signal prior to filtering and handles processes only with additive noise.

3.3 Summary

Speech enhancement algorithms are used for improving the performance of communication systems when their input or output signals are corrupted by noise. The existence of background noise lets the quality and intelligibility of speech to degrade. Speech enhancement algorithms may be broadly classified into two categories: single and multi-channel enhancement. In this chapter, single channel methods, where the input is only obtained from one microphone, have been discussed. The advantages and limitations of some of the main single channel speech enhancement methods were reviewed and explained in details. Table 3.1 summarises the advantages and disadvantages of these speech enhancement methods. Another speech enhancement technique consists of the wavelet transform. This is very powerful tool and is introduced in detail in the next chapter.

Speech Enhancement Method	Advantages	Disadvantages
Comb Filter	Linear phase and guaranteed stability. Low computational complexity.	Reduces the speech intelligibility.
Adaptive Noise Enhancement	Does not require prior knowledge of the speech signal. Has many well documented implementations(LMS, RLS).	Can only handle processes with additive noise.
Spectral Subtraction	Uncomplicated in terms of computational complexity.	Can create non-linear processing distortions caused by random variations of the noise spectrum. The degradations are explicit under high noise conditions
Wiener Filtering	Easy to implement in real time.	Gives a fixed frequency response at all frequencies.

Table 3.1: Comparison Table of Speech Enhancement Methods

4 WAVELET BASED APPROACH

4.1 Introduction

Wavelets are mathematical functions that separate data into different frequency components and then investigate each component with a resolution matched to its scale. Unlike the traditional Fourier methods, with the wavelet approach, different frequencies are analysed using different resolutions, depending on the signal type instead of using the same window (consequently same resolution) for each spectral component[48]. In the Fourier theory(FT), a signal is considered as a sum of theoretically infinite sines and cosines which makes the FT useful for infinite and periodic signal analysis[49]. The FT dominated the signal processing field for many years because it works well in providing the frequency information of the signal being analysed. Although it provides frequency information, unfortunately, it fails to give any information about the occurrence time, which led the scientists to suggest new methods such as the Short-Time Fourier Transform (STFT). This approach cuts the target signal in several parts and then analyse each part separately, which creates another problem: How to cut the signal? This problem is solved by the wavelet transform, which provides a fully scalable modulated window, requiring thus no signal cutting.

The wavelet transform is described as a mathematical method in which the target signal is analysed in the time domain by using different versions of a dilated and translated basis function called the mother wavelet. The Haar wavelet, the first wavelet transform, was introduced at the beginning of the 20th century by a German scientist, and then named after him. The Haar wavelet basis function has compact support and integer coefficients and was later used in physics to study Brownian motion[50]. From then on, several studies have been carried out either in the development of the wavelet theory or its applications in different fields. In the field of signal processing, the great accomplishments achieved by Mallat, Meyer and Daubechies led to a wide range of successful wavelet-based applications. Meyer[51] developed the first non-trivial wavelets after the work carried out by Mallat[52] on the relationships between the Quadrature Mirror Filters (QMF), pyramid algorithms and orthonormal wavelet basis. Inspired by Mallat's work, another important work was developed by Ingrid Daubechies. Daubechies achieved the cornerstone of many applications by constructing a set of wavelet orthonormal basis functions[53]. The same author in collaboration with others[54] introduced a set of wavelet biorthogonal basis functions which are used in different applications such as image coding, compression and pattern recognition[55].

In this chapter, the basic concepts of the classic wavelet transform and its relationship to the Fourier transform will be introduced. The implementation of discrete wavelet transform will be explained in Section 5 and 6. In Section 7, denoising using wavelet transform will be described. Some relevant information about wavelets is introduced in Section 8 and the chapter is summarised in Section 9.

4.2 Continuous Wavelet Transform

There are different ways to explain the wavelet transform. This goal is in general achieved by introducing the Fourier theory at the beginning. In practice, signals are represented in time-amplitude format in the time domain. However, for most speech processing applications, there is a need for other representations as some important information are hidden in the frequency content of the signal. A Fourier Transform (FT) decomposes the signal into complex exponential functions at different frequencies in order to get the frequency content of the signal as [56]:

$$\mathrm{X}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-2j\omega t} dt \tag{4.1}$$

where t stands for time in sec, ω stands for frequency in rad/sec, x(t) denotes the signal in time domain and X(ω) denotes the signal in the frequency domain, respectively.

The original signal can be recovered , under certain conditions, by the inverse Fourier Transform as follows:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \tag{4.2}$$

The integration covers all time instances from minus infinity to plus infinity. Therefore the frequency component w is equally reflected in the result of the integration, whenever it occurs over time. Simply, the integration won't change whether the frequency component f appears at time t1 or time t2. This makes the Fourier Transform unsuitable for non-stationary signal whose frequency content changes over time. The signal is supposed to have the frequency component w at all times in a way that the Fourier transform will turn to be useful. Therefore the stationary and the non-stationary nature of the signal is of importance to the FT.

It is then natural that a transform with both time and frequency localisation is required for non stationary signals. The short time Fourier transform(STFT) falls into this category of transforms.

There is only a minor difference between the STFT and the FT. In STFT, the signal is divided into small enough segments, where these segments (portions) of the signal can be assumed to be stationary. For this purpose, a window function "w" is required. The width of this window must be equal to the segment of the signal where its stationarity is valid. The STFT is summarised in one line in equation 4.3.

$$\mathrm{STFT}(\theta, \mathrm{w}) = \int_{-\infty}^{\infty} x(t) w(t-\theta) e^{-j\omega t} dt \qquad (4.3)$$

where x (t) represents the signal under consideration and w(t) the window function. As it can be seen from the equation, the STFT of the signal is the FT of the signal multiplied by a window function. The STFT transform of a signal x(t) is thus defined around a time θ through the usage of a sliding window w[57]. As it can be seen from equation 4.3, even if the integral limits are infinite, the analysis is always bounded by the limits [-θ , θ] of the sliding window.

The combination of time domain and frequency-domain analysis yields a more revealing picture of the signal, showing which spectral components are present in a signal at a given time slot[58].

According to Heisenberg's uncertainty principle, it is not possible to know what spectral components exist at a particular time instance[59]. This leads to some limitations of the STFT. A time interval is necessary to find which specific frequencies occur at this specific time. The time information is limited to the time interval, leading to a low resolution. While a Kernel function being infinite in length leads to perfect frequency resolution (with no time information)in the case of the FT, finite window length in STFT assures no perfect frequency resolution. Moreover, the window used in the STFT should be short enough to assure that the signal is stationary. The narrower the better is the time resolution and the poorer is the frequency resolution.

This situation can be improved depending on the application. If the frequency components are well separated, then good time resolution can be targeted while sacrificing the frequency resolution. If not, then a good window function is essential and the wavelet transform is a good candidate to solve this problem by providing good time resolution at high frequencies and good frequency resolution for slowly varying functions.

Grossman and Morlet [60] formulate the Continuous Wavelet Transform(CWT) as seen in equation 4.4 after realising the poor time localisation of the FT and fixed time and frequency localisation limitation of the STFT. Unlike the STFT, with this approach, different frequencies are analysed with different resolutions depending on the signal type instead of using the same window (consequently same resolution) for each spectral component. Low frequency resolution (high time resolution) is reached for low frequencies and vice versa for high frequencies. The Wavelet Transform is useful if the signal to be processed has high frequency components for a short while and low frequency components for longer time; which is typical of speech signals.

Although the CWT is applied in a similar way as the STFT, there are two main differences: Unlike the STFT, the window function is modified as the transform is computed for every single spectral component. The continuous wavelet transform of a function x(t) at a scale s and translational value τ is defined as:

$$CWT_x^{\psi}(\tau, \mathrm{s}) = \frac{1}{\sqrt{|s|}} \int x(\mathrm{t}) \psi^*(\frac{t-\tau}{s}) \, \mathrm{dt} \tag{4.4}$$

where ψ(t) is a continuous function in both time domain and the frequency domain called the mother wavelet and x(t) belongs to the square integrable functions space, $L^2(R)$. In the same way, The inverse CWT can be defined as:

$$x(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{s}} CWT(s,\tau) \psi\left(\frac{t-\tau}{s}\right) \frac{ds\,d\tau}{s^2} \tag{4.5}$$

The C_ψ factor is crucial for reconstruction purposes which is known as the admissibility condition. The time-frequency planes of a STFT and Wavelet Transformation(WT) are illustrated in Figure 4.1. The difference between the STFT and the WT is visually clear.

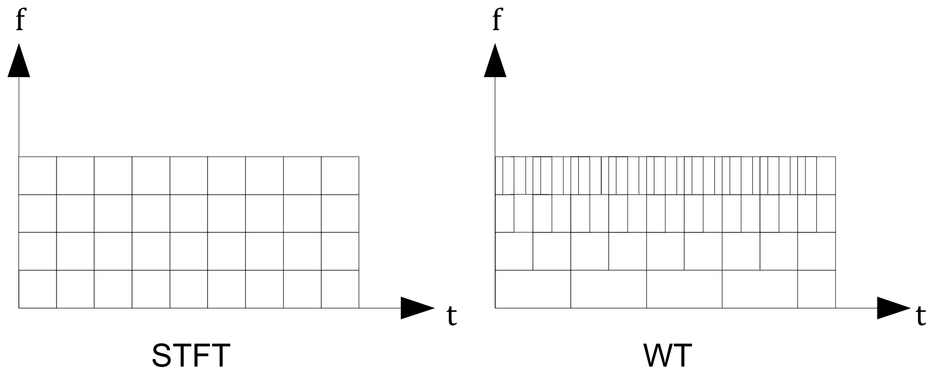

Figure 4.1: Fourier and Wavelet time-frequency plane

After this brief introduction, let's define what wavelet transforms are. Although a wavelet transform is defined as a mathematical tool or technique, there is no agreed definition on the wavelet transform within the scientific community. According to Sweldens, three properties have to be fulfilled to call a particular function a wavelet system[61]:

- Most of the energy of a wavelet is limited in a finite interval and the transform contains frequencies only from a certain frequency band which is called space-frequency localisation.
- Wavelets are building blocks for general functions. Namely, a function is represented in the wavelet space by mean of infinite series of wavelets.
- Wavelets support fast and efficient transform algorithms which is important when implementing the transform.

4.3 Multiresolution Analysis

The Multiresolution Analysis(MRA) concept was initiated by Mallat[52], which provides a natural framework for the understanding of wavelet bases. MRA was formulated based on the study of orthonormal and compactly supported wavelet bases. The attraction of multiresolution is its ability to represent a function at multiple levels of details. Also, it allows the description of a signal in terms of time-frequency or time-scale analysis.

4.3.1 Subspaces

Multiresolution analysis requires the existence of a set of approximation subspaces of $L^2(R)$ (square integrable function space) with different resolutions as:

$$V_{-\infty} \ldots \subset V_{-1} \subset V_0 \subset V_1 \ldots V_\infty \tag{4.6}$$

If $x(t) \in V_j$ then $x(t) \in V_{j+1}$ which means that the subspace containing high resolution will automatically contains those of lower resolution. In a more common case, if $x(t) \in V_0$ then $x(2^k t) \in V_k$ which is known as the scale invariance property.

Three intermediate subspaces are schematically represented in Figure 4.2.

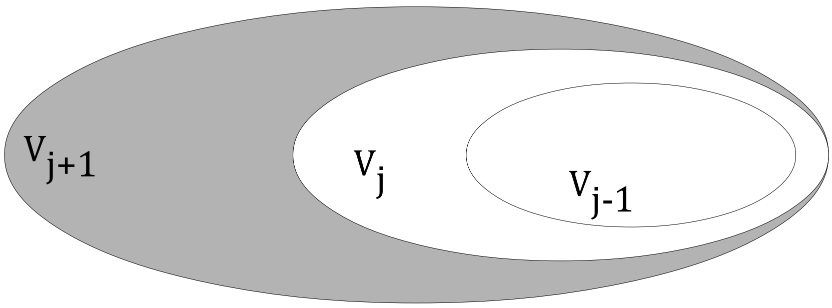

Figure 4.2: Nested subspaces

Copyright:Creative Commons Attribution 3.0 Unported

https://www.intechopen.com/books/advances-in-wavelet-theory-and-their-applications-in-engineering-physics-and-technology/the-wavelet-transform-for-image-processing-applications

4.3.2 Scaling Functions

In order to benefit from multiresolution analysis, the presence of a so-called scaling function $\phi(t)$ is crucial. First of all, we can define the scaling function and then define the wavelet function through it[62]. Let the scaling function be defined by the following equation:

$$\phi_k(t) = \phi_k(t-k) \qquad k \in Z \qquad \phi \in L^2(R) \tag{4.7}$$

which constructs with its translates an orthonormal basis (the orthogonality is not necessary since non orthogonal basis can always be orthogonalised [63]) of the space V_0:

$$V_0 = \underset{k}{\mathrm{span}}\{\phi_k(t)\} \tag{4.8}$$

It means that any function belonging to this space ($x(t) \in V_0$) can be expressed using the scaling function and its consecutive translates(since $\phi_k(t)$ are the basis functions) as a linear combination of a set of expansion coefficients:

$$x(t) = \sum_k a_k \phi_k(t) = \sum_k a_k \phi(t-k) \tag{4.9}$$

where a_k or $a(k)$, the expansion coefficients, are calculated using the inner product:

$$a_k = \langle x(t), \phi_k(t) \rangle \tag{4.10}$$

A two-dimensional scaling function can be produced from the original scaling function by simply scaling and translating using equation 4.7:

$$\phi_{j,k}(t) = \frac{1}{\sqrt{s}} \phi\left(\frac{t-\tau}{s}\right) \tag{4.11}$$

where s is the scaling factor and τ is the shifting factor as defined in equation 4.4. To make the implementation easier, the translation and the scaling factors have been adopted to be a factor of two[50]:

$$s = 2^{-j}, \tau = 2^{-j}.k \tag{4.12}$$

Equation 4.11 can be rewritten by adopting these values for the remaining of the chapter:

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \tag{4.13}$$

Similarly, the two-dimensional scaling function forms an orthonormal space over k with its translates:

$$V_j = \underset{k}{\mathrm{span}}\{\phi_{j,k}(t)\} \qquad k \in Z \quad \text{and} \quad j \in Z \tag{4.14}$$

and x(t), as any function, of this space can be described as:

$$x(t) = \sum_k a(k)\phi(2^j t + k) \tag{4.15}$$

As a result, if $\phi(t) \in V_0$, then since $V_0 \subset V_1$, $\phi(t)$ can be expressed as a linear combination of the scaling function $\phi(2t)$ spanning the space V_1:

$$\phi(t) = \sum_k h(k)\sqrt{2}\phi(2t - k) \tag{4.16}$$

where the coefficients h(k) are the scaling function coefficients. The value $\sqrt{2}$ guarantees that the norm of the scaling function is always equal to the unity. Equation 4.16 is called the multiresolution analysis equation and it is essential for the multiresolution theory as introduced by Mallat.

4.4 Wavelet Function

The same methodology can be applied to the wavelet function as it has been applied to the scaling function, its translates and the corresponding spanned spaces. V_1 can be represented as a combination of V_0 and W_0 when it is assumed that W_0 is an orthogonal complement of the subspace $V_0 \subset V_1$.

$$V_1 = V_0 \oplus W_0 \tag{4.17}$$

where the complementary space W_0 is spanned also by an orthonormal basis:

$$\psi_k(t) = \psi(t-k) \qquad k \in Z \quad \text{and} \quad \psi \in L^2(R) \tag{4.18}$$

where the function $\psi(t)$ is known as the mother wavelet, the wavelet prototype or the wavelet function. A function $x(t) \in W_0$ can be expressed in the same way as it has been done for the scaling function.

$$x(t) = \sum_k d_k \psi_k(t) = \sum_k d(k)\psi(t-k) \tag{4.19}$$

where the expansion coefficients d_k or $d(k)$ are calculated using the inner product:

$$d_k = \langle x(t), \psi_k(t) \rangle \tag{4.20}$$

$\psi(t)$ can be expressed in terms of the scaling function $\phi(2t)$ of the higher space V_1 since $V_0 \subset V_1$.

$$\psi(t) = \sum_k g(k)\sqrt{2}\phi(2t-k) \tag{4.21}$$

where g(k) are the wavelet coefficients. This leads to a dyadic decomposition as represented by the grid of Figure 4.4. The equation 4.17 can be generalised to an arbitrary number of subspaces, such as, V_2 is represented in terms of V_1 and W_1, V_3 in terms of V_2 and W_2, and so on as shown in Figure 4.3.

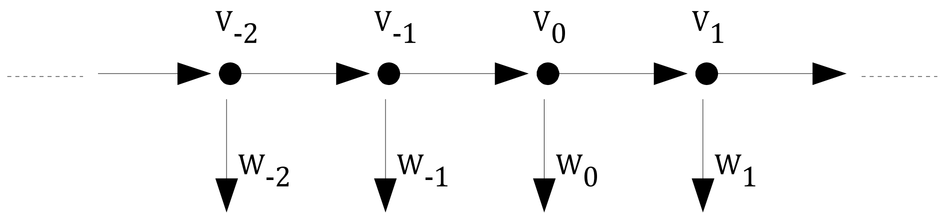

Figure 4.3: Space decomposition[49]

Copyright:Creative Commons Attribution 3.0 Unported

https://www.intechopen.com/books/advances-in-wavelet-theory-and-their-applications-in-engineering-physics-and-technology/the-wavelet-transform-for-image-processing-applications

A subspace V_j is spanned by W_{j-1} and V_{j-1} in general. Thus, the $L^2(R)$ space can be decomposed as follows:

$$L^2(R) = V_j \oplus W_j \oplus W_{j+1} \oplus W_{j+2} \dots \oplus W_0 \oplus W_1 \dots \tag{4.22}$$

The index j shows the depth or the level of decomposition, which is arbitrary in this case. A two dimensional scaled and translated wavelet function is defined by:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{4.23}$$

In such a way:

$$W_j = \operatorname*{span}_k \left\{\psi_{j,k}(t)\right\} \tag{4.24}$$

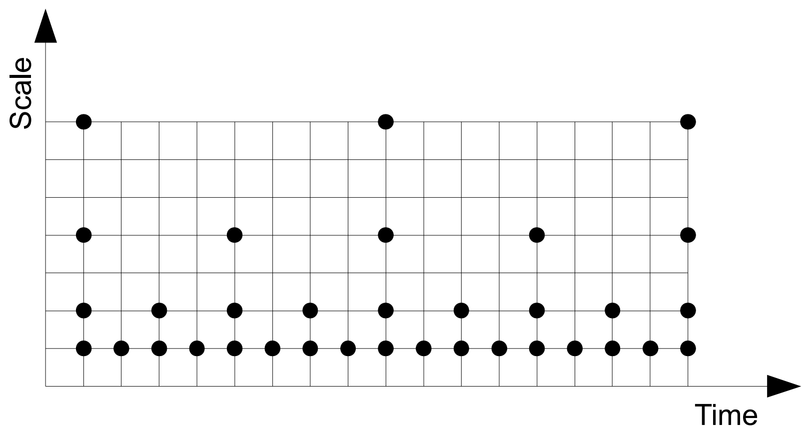

Figure 4.4: Dyadic wavelet transform space representation[49]

Copyright:Creative Commons Attribution 3.0 Unported

https://www.intechopen.com/books/advances-in-wavelet-theory-and-their-applications-in-engineering-physics-and-technology/the-wavelet-transform-for-image-processing-applications

4.5 Discrete Wavelet Transforms

While the continuous wavelet transform uses contraction and dilatation of the wavelet functions, the discrete wavelet transform (DWT) benefits from filter banks to construct the multi-resolution time-frequency plane[64]. Multi resolution filter banks and special wavelet filters are used for the analysis and reconstruction of signals[65].

The DWT is a discrete-time framework or a fast algorithm for calculating the wavelet transform. The analysis can be described as a single continuous to discrete conversion procedure followed by an iterative discrete-time processing instead of implementing the analysis as a sequence of continuous filter and sample operations[66]. The synthesis can also be similarly formulated.

4.6 Filter Bank Implementation of the Discrete Wavelet Transform

In general, wavelet transform-based applications include discrete coefficients instead of scaling and/or wavelet functions. Discrete time filter banks are necessary for practical and computational reasons. The signal is analysed at different resolutions by decomposing the signal into approximation and detail information[67]. The approximation coefficients a_n and detail coefficients d_n of a given signal x(t) can be obtained via a filtering and a downsampling procedure as as shown in equation 4.25:

$$\begin{aligned} a_n &= \sum_n \mathrm{x[n]\,h[2\,k-n]} \\ d_n &= \sum_n \mathrm{x[n]\,g[2\,k-n]} \end{aligned} \tag{4.25}$$

The signal is decomposed into different frequency bands by consecutive lowpass and highpass filtering of the time domain signal as seen in Figure 4.5.

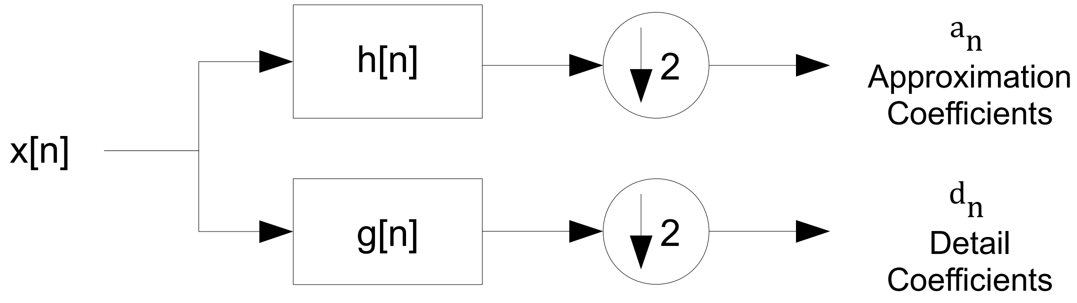

Figure 4.5: A single stage Discrete Wavelet Transform Analysis

A low pass filter h[n] and a high pass filter g[n] are applied to the original signal x[n]. The filtering process is followed by down sampling by a factor of 2. Consequently, half of the sample are discarded for each filter.

The synthesis algorithm is structured in a similar manner as shown in Figure 4.6. The signals are upsampled by a factor two at every level and passed through the synthesis low pass filter h'[n], the synthesis high pass filter g'[n] and then merged. Since halfband filters form orthonormal bases, synthesis is achieved by following the same procedure in reverse order.

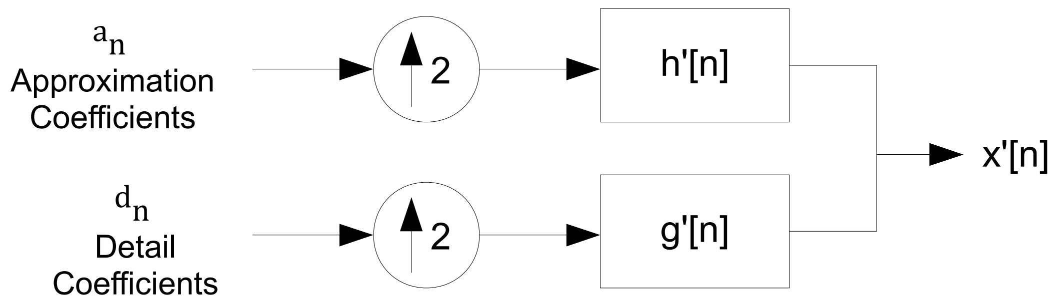

Figure 4.6: A single stage Discrete Wavelet Transform Synthesis

The analysis and synthesis filters are identical with each other, therefore the synthesis formula becomes:

$$x[\mathrm{n}] = \sum_{k=-\infty}^{\infty} (\mathrm{a}_n\, \mathrm{h}[-\mathrm{n}+2\,\mathrm{k}] + d_n\, \mathrm{g}[-\mathrm{n}+2\,\mathrm{k}]) \qquad (4.26)$$

The relationship between the impulse responses of the low pass filter and the high pass filter is one of the most important property of discrete wavelet transform. The high pass and the low pass filter are related to each other as shown in equation 4.27[68]:

$$\mathrm{g'}[\mathrm{n}] = \mathrm{g}[\mathrm{L}-1-\mathrm{n}] = (-1)^n h[\mathrm{n}] \qquad (4.27)$$

where g[n] is the highpass filter, h[n] is the low pass filter and L is the filter length. Filters satisfying this condition are called Quadrature Mirror Filters(QMF), which is valid for orthogonal discrete wavelets and commonly used in signal processing.

The analysis divides the time resolution into two since the signal is characterised by only half the number of samples. On the other hand, the frequency resolution is doubled by this process, since the frequency band of the signal spreads half of the previous frequency band. This procedure, called subband coding, can be repeated for further analysis. The filtering and subsampling will halve the number of samples and the frequency band spanned. A signal can thus be divided into many lower resolution components by an analysis process called Subband Coding structure, as shown in Figure 4.7. At each stage, the spectrum frequency of the analysed signal is halved by a factor of two[62]. The decomposition process can be continued indefinitely in theory as it is iterative. However, the decomposition can proceed only until the individual details consist of a single sample.

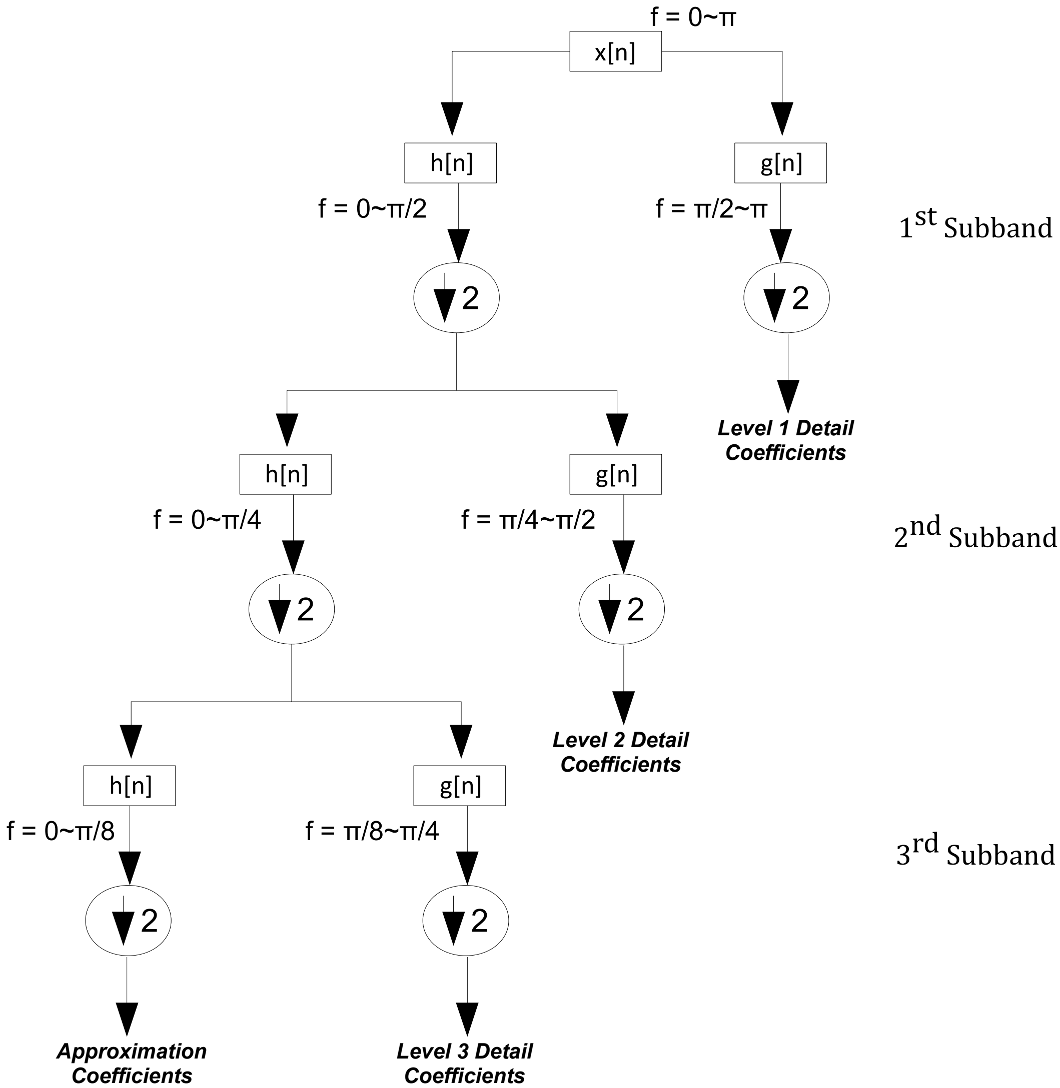

Figure 4.7: Three-Stage analysis Subband Coding

x[n] is the original signal to be analysed, h[n] and g[n] are the lowpass and the highpass filters, respectively. The signal bandwidth is symbolised as "f" at every level.

The original signal x[n] has n sample points divided from 0 to Π in the frequency band. The signal is passed through a high pass and a low pass filters and followed by downsampling by 2 at the first analysis level. The product of the high pass filter which constitutes the first level of the DWT coefficients has n/2 points and it is only divided from Π/2 to Π in the frequency band.

The output of the lowpass filter also has n/2 samples which are spanned in the other half of the frequency band. This signal is then passed through the same lowpass and high pass filters for further analysis. The DWT of the original signal is then acquired by collecting all coefficients starting from the last level of analysis.

The most dominant frequencies in the original signal will appear as high amplitudes in that DWT signal region, which has those particular frequencies. The time localisation of these frequencies will not disappear as it is the case for the FT. If the high frequencies have the main information of the signal, as it is the case, the time localisation of these frequencies will be more accurate since more number of samples are representing them. If the low frequencies have the main information of the signal, the time localisation of these frequencies will not be very accurate since few samples are representing them. This process provides a good frequency resolution at low frequencies and a good time resolution at high frequencies, which is the case for most practical signals.

More detailed information can be offered by a DWT method called the wavelet packet transform. A signal is split into an approximation and a detail in the wavelet analysis. Then a second-level approximation and detail are obtained from the approximation and the process is repeated[69]. However, the details as well as the approximations are divided into parts in a wavelet packet transform.

A 3-layer wavelet packet decomposition tree or Mallat's pyramid is shown in Figure 4.8 where x[n] represents the original signal. The low-frequency components(details) of the original signal are represented with cA and the high-frequency components(approximations) are represented with cD. The index number defines the layer number of the wavelet packet decomposition[70].

Since the one-dimensional decomposition and reconstruction schemes have been already introduced before, we will explain the two-dimensional scheme. The two-dimensional decomposition approach is based on the property of separation of the functions into arbitrary x and y directions. The first step is identical to the one-dimensional approach, however, instead of keeping the low-level resolution and processing the high level resolution, both are processed using two identical filter bank after a transposition of the incoming data. Thus, the signal is scanned in both horizontal and vertical directions in this technique.

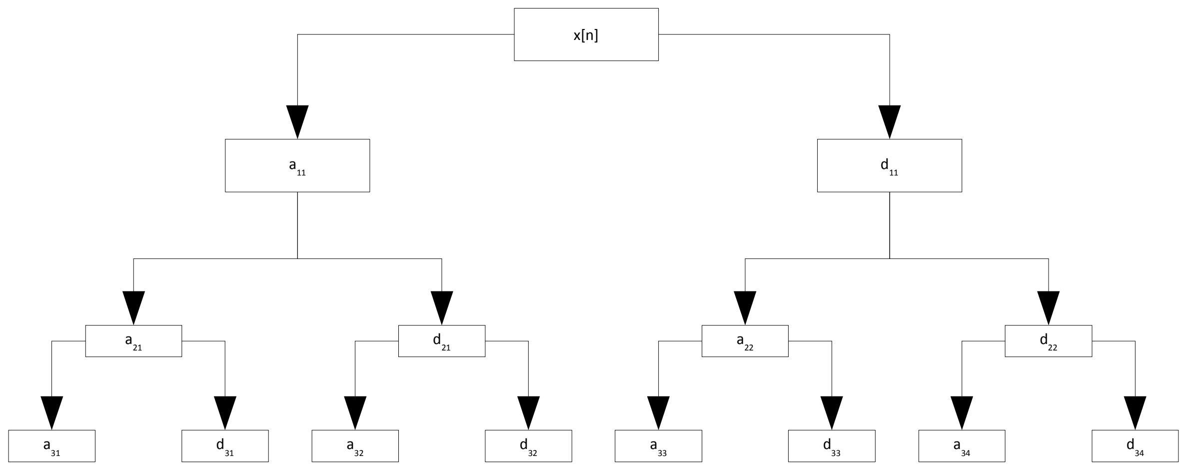

Figure 4.8: The 3-layer wavelet packet decomposition tree

4.7 Signal Denoising Using Wavelet Transform

In 1995, Donoho introduced wavelet thresholding (shrinking) as a powerful tool in denoising signals degraded by additive white noise. Wavelets have been successfully applied to denoising tasks and there has been recent interest in using the Wavelet Transform in speech denoising because of its robust features[71]. A modified wavelet speech enhancement is proposed by M. Bahoura and J. Rouat which is based on the teager energy operator[72]. Although an explicit estimation of the noise level or a priori knowledge of the SNR is usually needed in many popular speech enhancement methods, this technique does not require them. A wavelet transform approach to the blind adaptive filtering(BAF) of speech from unknown noises is introduced by Veselinovic and Graupe[73]. The paper describes the BAF system which performs Discrete Wavelet Transform of noisy speech signals and identifies and separates noise from speech. Lu and Wang introduced a closed-form solution of gain factor for wavelet coefficients in a subband by constraining the residual noise to be lower than the noise masking threshold[74]. The paper shows that this approach can efficiently remove the corrupting noise without any empirical factor. A wavelet-based speech enhancement is proposed by S. Liby using an improved discrete wavelet packet decomposition method called wavelet-packet frequency algorithm[75]. The system can be more robust to noise and spectrum distortions by including more time and frequency information to the denoising wavelet algorithm.

A pre-processing phase based on wavelet denoising for extracting MFCC features in the existence of white Gaussian noise is proposed by Farooq and Datta[77]. They found that MFCC features extracted after wavelet denoising were improved recognition by 2 to 28 % for SNRs in the range 20 to 0 dB.

Bionic Wavelet Transform (BWT) for speech signal processing is used by Yao and Zhang in cochlear implants[77]. The BWT is an alteration of a wavelet transform that includes the active cochlear mechanism into the transform. After speech material processed with the BWT and the WT is compared, they concluded that application of the BWT in cochlear implants has some advantages such as enhanced recognition rates for both vowels and consonants, decline in the number of channels in the cochlear implant and the average stimulation duration for words, better noise tolerance and higher speech intelligibility rates.

Another wavelet speech enhancement scheme which is based on the Teager energy operator is proposed by Bahoura and Rouat[78]. The Teager energy operator is a nonlinear operator which can extract signal energy based on mechanical and physical analysis[79]. They used a wavelet thresholding method where the discriminative threshold in various scales was time adapted to the speech waveform. Their method gives higher SNR improvements after they compared their results with the results Ephraim et al are obtained[80]. Unlike the speech enhancement method of Ephraim et al, the method of Bahoura et al did not need explicit evaluation of the noise level or a priori knowledge of the SNR.

Another wavelet based speech enhancement research has been conducted by Saeed Ayat, M. T. Manzuri and R. Dianat using a new thresholding algorithm[81]. The noisy signal is transferred to wavelet domain in this research. The wavelet coefficients are improved with applying a thresholding algorithm. A new method of thresholding is introduced at this point for speech enhancement. At last the enhanced signal is obtained using the inverse transform. The speech signals corrupted by additive white Gaussian noise with various global signals to noise ratios (SNR) from OdB to 12dB were used for performance evaluation of all thresolding algorithms. The best SNR improvement is achieved as 3.86dB which is used for comparison with our results. The shortcoming of this research is only one type of noise, white Gaussian noise, is used to test the new method. Also, the method is not evaluated for memory usage and operational speed which are important parameters of a new method.

Bing-yin XIA and Chang-chun BAO presented a new method for speech enhancement using a wavelet fusion method[82]. In the proposed method, the noisy speech is first decomposed into several sub-bands by wavelet packet analysis, and then enhanced by the statistical model based method and wavelet thresholding method, respectively. The output of each sub-band is obtained under the fusion rule based on the cross-correlation and the a priori SNRs of the two enhanced coefficient sets. Finally, the enhanced coefficients are transformed back to time domain to get the enhanced speech. The performance results are compared with some other reference algorithms such as Weighted Euclidean Distortion Measure (WEDM) and wavelet thresholding. In their experiments, the clean speech sequences are chosen from the NTT speech database of Chinese language and Gaussian White noise is generated by software, and coloured noise signals are taken from the ITU-T and NoiseX-92 noise database. Both noise and speech samples have been down-sampled to 8 kHz for their tests. Although the result of noise reduction test using proposed method gives better SNR results compared to WEDM, it does not provide better SNR results compared to wavelet thresholding when Gaussian white noise is used as a reference noise. Similar results are obtained using coloured noise in their experiments. The test is carried out under four kinds of noise conditions, including babble, factory, street and white noise for objective result comparison. The SNR conditions of 6dB, 12dB and 18dB are used. The quality of enhanced speech produced by the proposed algorithm is slightly improved in comparison with the WEDM method and wavelet thresholding. The best score is obtained as 3.25 out of 5 with the proposed algorithm under 18dB factory noise while the noisy speech is rated as 3 out of 5.

In conclusion, wavelet transform is a powerful tool for modelling and analysing non-stationary signals, such as speech signals. These signals show slow temporal variations in low frequencies and sudden changes in high frequencies. Therefore, the wavelet transform can provide an appropriate model of speech signal for denoising applications and that is why it is used in our research work.

4.7.1 Wavelet Thresholding

The wavelet thresholding approach can be used to reduce the background noise, especially white noise, in speech signals. White noise can be transformed without losing its features using the orthogonality of the the DWT. Therefore, if x_{jk} are the wavelet coefficients of x_i, where j is the decomposition level and k is the index of the coefficient in this level, the wavelet decomposition is:

$$x_{jk} = a_{jk} + d_{jk} \tag{4.28}$$

where a_{jk} are the clean wavelet approximation coefficients and d_{jk} are the detail coefficients. The wavelet coefficients of the observed signal have the effect of noise on the original signal and most coefficients of the noiseless signal in a wavelet transform are effectively zero. For that reason, recovering the coefficients of the observed signal which are relatively bigger than Gaussian white noise can enhance the speech signal. In other words, coefficients with small magnitude can be evaluated as noise and can be set to zero. This method, which matches each coefficient with a threshold value to agree whether it is a beneficiary part of the original signal, is called wavelet thresholding.

The thresholding is only applied to the detail coefficients instead of the approximation coefficients since the latter ones contain low-frequency elements which represent the main components of the signal. The most important coefficients are extracted by setting the coefficients which are below a determined threshold value denoted λ.

The result of the thresholding are obtained using either hard or soft thresholding methods. Hard thresholding can be described as the process of setting the elements whose absolute values are lower than the threshold to zero as:

$$\delta_\lambda(d_{jk}) = \begin{cases} 0, & \text{if } |\mathrm{d}_{jk}| \le \lambda \\ \mathrm{d}_{jk}, & \text{if } |\mathrm{d}_{jk}| \rangle \lambda \end{cases} \tag{4.29}$$

Soft thresholding is an extension of hard thresholding which shrinks the non zero coefficients towards 0 and then sets to zero the elements whose absolute values are lower than the threshold as:

$$\delta_\lambda(d_{jk}) = \begin{cases} 0, & \text{if } |\mathrm{d}_{jk}| \leq \lambda \\ \mathrm{d}_{jk} - \lambda, & \text{if } |\mathrm{d}_{jk}| \rangle \lambda \\ \mathrm{d}_{jk} - \lambda, & \text{if } |\mathrm{d}_{jk}| \langle -\lambda \end{cases} \tag{4.30}$$

where δ_{jk} is the coefficient of the denoised noisy speech signal, d_{jk} is the coefficient of the noisy speech signal and λ is the threshold value which is generally a function of j and k.

The thresholding value λ is estimated by different algorithms using thresholding method. The first action is to calculate the noise level σ in these algorithms. If the noisy signal is not fairly flat, it cannot be assumed that σ is equivalent to the standard deviation of the coefficients. Donoho and Johnstone propose a well-known estimate for the noise level σ[83]. According to the median absolute deviation, σ can be calculated based on the last level of the detail coefficient as:

$$\sigma = \frac{median(\{|\mathrm{d}_{jk}|\})}{0.6745} \tag{4.31}$$

where the number in the denominator is the scale factor. It is equal to 0.6745 for a normally distributed data and depends on the distribution of d_{jk}.

For a series of length n, the universal threshold value is:

$$\lambda = \sigma\sqrt{2\log n} \tag{4.32}$$

4.8 Other Aspects of the Wavelet Families

In most practical problems, both the orthonormal basis [84] and the biorthogonal basis [54] can be used. There are some similarities and differences between these two bases. The following briefly explains the main characteristics of orthonormal and biorthogonal bases together. The wavelet families that are used in our work are also explained in details in this section.

4.8.1 Orthonormal Basis

Mallat's and Daubechies's work [54] [58] initiated the orthonormal basis. The orthonormality property is seen as the discrete version of the orthogonality property to a limited extend[85]. However, the basis functions are further normalised. These terms have been explained when the multiresolution feature and the scaling function have been introduced. The admissibility and the orthogonality conditions guarantee the existence and the orthogonality feature of the scaling function as shown in equation 4.27. This is achieved if:

$$\sum_{n} h[n] = \sqrt{2} \tag{4.33}$$

and

$$\sum_{n} h[n]\, h[n+2k] = \delta[k] \tag{4.34}$$

The orthogonality of the scaling function and the wavelet function can be derived using the two equations above alongside with equation 4.27. This can be achieved only if the following equality is confirmed.

$$g[n] = (-1)^{n}\, h[1-n] \tag{4.35}$$

The orthogonality between the wavelet coefficients and the scaling coefficients is then:

$$\sum_{n} h[n]\, g[n] = 0 \tag{4.36}$$

The scaling coefficients are called Quadrature Mirror Filters (QMF) if the equation 4.36 is satisfied.

The analysed signal has to be identical to the synthesised one to achieve perfect reconstruction as shown in Figure 4.9. In other words, $a_j(n)=\hat{a}_j(n)$, where $a_j(n)$is the input and $\hat{a}_j(n)$ is the output of a filter bank.

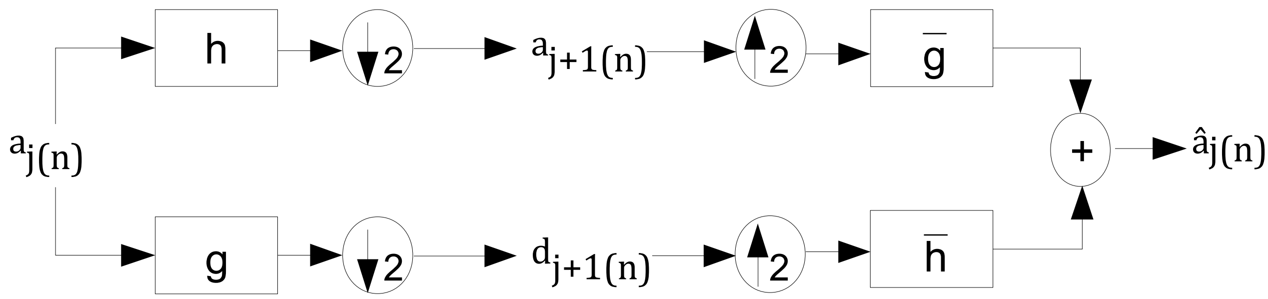

Figure 4.9: Two-band analysis and synthesis filter bank

4.8.2 Biorthogonal Basis

Biorthogonal wavelet basis is a generalisation of the orthogonal wavelet basis where some placed restrictions on the latter have been softened. The scaling and wavelet functions does not need to be of the same length or even numbered unlike the case of orthogonal basis. Therefore, the quadrature mirror property is not applicable and replaced with a dual property. The scaling and the wavelet coefficients have to fulfil the following equations for the perfect reconstruction equation to hold.

$$\begin{aligned} \overline{g[n]} &= (-1)^n h[1-n] \\ g[n] &= (-1)^n \overline{h[1-n]} \end{aligned} \tag{4.36}$$

It is clear that the system becomes orthogonal when the analysis and the synthesis filters are similar. In this case, the orthogonality condition is defined by:

$$\sum_{n} h[\overline{n}]\,h[n+2k] = \delta[k] \tag{4.37}$$

In orthogonal basis, only the analysis scaling coefficients along with their shifted versions are used. However, in the biorthogonal case, the analysing scaling coefficients are kept unchanged, while their shifted versions are changed by the shifted versions of the synthesis dual filter. Namely, the analysis filter is orthogonal to its synthesis dual filter. The biorthogonal denomination comes from this feature.

One of the most important features in the biorthogonal basis is the linear phase property, which makes the filter coefficients being symmetric when a wavelet system is implemented. Moreover, the difference of length between dual filters must be even, leading either to odd or even length of the low pass and the high pass filters [84].

4.8.3 Wavelet Types

Different families of wavelets can be created based on the selection of the wavelet function $\psi(t)$ and the scaling function $\phi(t)$[85]. This section starts with a list of wavelet properties and then presents particular wavelet families classifying them from those properties point of view, not from the type of wavelet analysis. The suitability of a particular wavelet family depends on the properties of its wavelets and also on the type of application[86].

4.8.3.1 Orthogonal Wavelets

Orthogonal wavelets enable orthogonal MRA as well as the possibility of perfect reconstruction. When orthogonal wavelets are used, the energy of the signal is preserved in both time and frequency domains. Moreover, most of orthogonal wavelets have a compact support, which make them the most often used wavelets in practise.

4.8.3.1.1 Daubechies Wavelets

The Daubechies wavelets, named after Ingrid Daubechies, are orthogonal wavelet families with vanishing moments[53]. The coefficients of Daubechies for the scaling and the wavelet filters are unique. This is why they have a high degree of smoothness. They are normally represented as dbn where n represents the size of the resulting filter.

4.8.3.2 Symlet Wavelets

The solution of smoothness is solved by Symlet wavelets using the symmetry feature of Daubechies wavelet family. They are also recognised as the least asymmetric wavelets and the construction of Symlet wavelets is almost identical to the Daubechies wavelets. Symlets are also known as the Daubechies least asymmetric wavelets. Their construction is very similar to the Daubechies wavelets. They are represented as *symn* where n represents the size of the resulting filter.

4.8.3.3 Coiflet Wavelets

Coiflet wavelets are designed to have better symmetry than the Daubechies wavelets. The Coiflet wavelets family is orthogonal and near symmetric. The near symmetry property results in linear phase characteristics of the Coiflet wavelets[87]. They are represented as *coifn,* where the size of the resulting filter is 6*n.

4.8.3.4 Crude wavelets

Some wavelets are expressed by a mathematical definition and are known as continuous and infinite and are called crude wavelets[88]. Such an expression enables a very good localisation in frequency. Crude wavelets can only be used in the CWT method. Although there is always a way to calculate their explicit formula in equi-spaced points in time to obtain a discrete-time approximation, however such an approximation is not orthogonal and cannot be used in the DWT method.

4.9 Summary

The Wavelet transform is an effective tool in speech enhancement for both its time-frequency localisation and its multi resolution characteristic. Unlike the Fourier transform, the wavelet transform is suitable for applications involving non-stationary signals with transitory phenomena, whose frequency response varies in time. The wavelet coefficients represent a measure of similarity in the frequency content between signal and a chosen wavelet function. These coefficients are computed as a convolution of the signal and the scaled wavelet function, which can be defined as a dilated band-pass filter because of its band-pass like spectrum. The scale is inversely proportional to radian frequency. Therefore, low frequencies are identical to high scales and a dilated wavelet function. By wavelet analysis at high scales, we extract global information from a signal called approximations. Whereas at low scales, we extract fine information from a signal called details. Signals are usually band-limited, which is equivalent to having finite energy, and therefore we need to use just a constrained interval of scales. However, the continuous wavelet transform provides us with lots of redundant information. The discrete wavelet transform needs less space using the space-saving coding based on the fact that wavelet families are orthogonal or biorthogonal bases, and thus do not produce redundant analysis.

5 OFFLINE AUDIO WAVELET DENOISING

In this chapter, the implementation and performance assessment of the wavelet transform based speech enhancement algorithm are presented using three different platforms. The wavelet packet decomposition is used for denoising instead of the common wavelet pyramid decomposition. The method gives much better results than the common wavelet pyramid decomposition. The results are compared in conclusion of this chapter. The proposed method gives better results as the high frequency coefficients are not ignored. Moreover, the algorithm is designed in such a way that the memory is allocated just once at the size of the corrupted audio signal and the same memory is used until the end of the process. It gives users the opportunity to denoise any size audio file without increasing the memory usage and makes the algorithm implementation possible for low memory processors. A graphical user interface (GUI) is also implemented. The GUI allows flexible parameter settings to experience different wavelet levels and wavelet families on different audio files. Threshold values are calculated automatically and can be fine-tuned with a sliding bar which gives user an extra option to improve SNR value or speech quality. Performance assessment of the proposed algorithm is very important. It is carried out using signal to noise ratio(SNR), speech quality and speech intelligibility. SNR is a value that indicates the amount of the audio signal compared to the amount of noise present in the signal and is expressed in decibels. However, SNR cannot be used as the only specification to evaluate the speech enhancement algorithm. Speech quality signifies how fine or natural the speech is and speech intelligibility shows how clear the speech can be understood by listeners. Although objective quality evaluation can reflect the speech quality on the basis of mathematical measures, they cannot be totally consistent with human perception. Subjective measures in addition to objective measures are used for speech enhancement evaluation in this thesis.

In section 5.1, the algorithm is implemented in Matlab and the performance and results of the wavelet transform speech enhancement are presented. Extensive experiments are conducted to find the best wavelet for this audio denoising application. Section 5.2 explains the implementation and performance evaluation of the wavelet transform speech enhancement algorithm using C language. The use of C is a necessary step to porting the proposed denoising method to a hardware platform. Different noise types with different SNR values are used which gives a comprehensive result set for the reader to compare the improvement as an SNR value, human perception and operational time. As already stated, to ease the experimental process, a GUI has been designed. It provides flexible parameter settings for evaluating the performance using different wavelet families, levels, types and thresholds.

5.1 Wavelet Thresholding Implementation And Results In Matlab

Six different utterances from a noisy speech corpus (NOIZEUS) library were used to evaluate the wavelet packet decomposition thresholding algorithm. Sentences from the IEEE sentence database were recorded in a sound-proof booth using Tucker Davis Technologies (TDT) recording equipment. The sentences were produced by three male and three female speakers. The IEEE database was used as it contains phonetically-balanced sentences with relatively low word-context predictability. The six sentences were selected from the IEEE database to allow the inclusion of all phonemes of the American English language. The list of sentences used for NOIZEUS are given in Table 5.1. The sentences were originally sampled at 25 kHz and downsampled to 8 kHz.

File Name	Gender	Sentence Text
Sp01.wav	Male	The birch canoe slid on the smooth planks.
Sp07.wav	Male	We find joy in the simplest things.
Sp11.wav	Female	He wrote down a long list of items.

Sp16.wav	Female	The stray cat gave birth to kittens.
Sp21.wav	Male	Clams are small, round, soft and tasty.
Sp27.wav	Female	Bring your best compass to the third class.

Table 5.1: List of sentences used from NOIZEUS

5.1.1 Implementation

The wavelet transform speech enhancement algorithm through wavelet thresholding is based on the fact that the original signal can be reconstructed by a limited number of wavelet coefficients in the lower bands. The proposed denoising algorithm as shown in Figure 5.1 can be summarised as follow:

1. Choose a wavelet basis.
2. Choose a wavelet packet decomposition level.
3. Compute the discrete wavelet packet transform for a noisy signal.
4. Perform thresholding on the wavelet coefficients based on the standard deviation of noise.
5. Compute the inverse discrete wavelet transform.

In order to extract the features of a noisy signal, a proper wavelet function must be chosen. This choice is motivated by the fact that different wavelet functions have different characteristics. In my experiments, different wavelet functions are selected and their performance are compared.

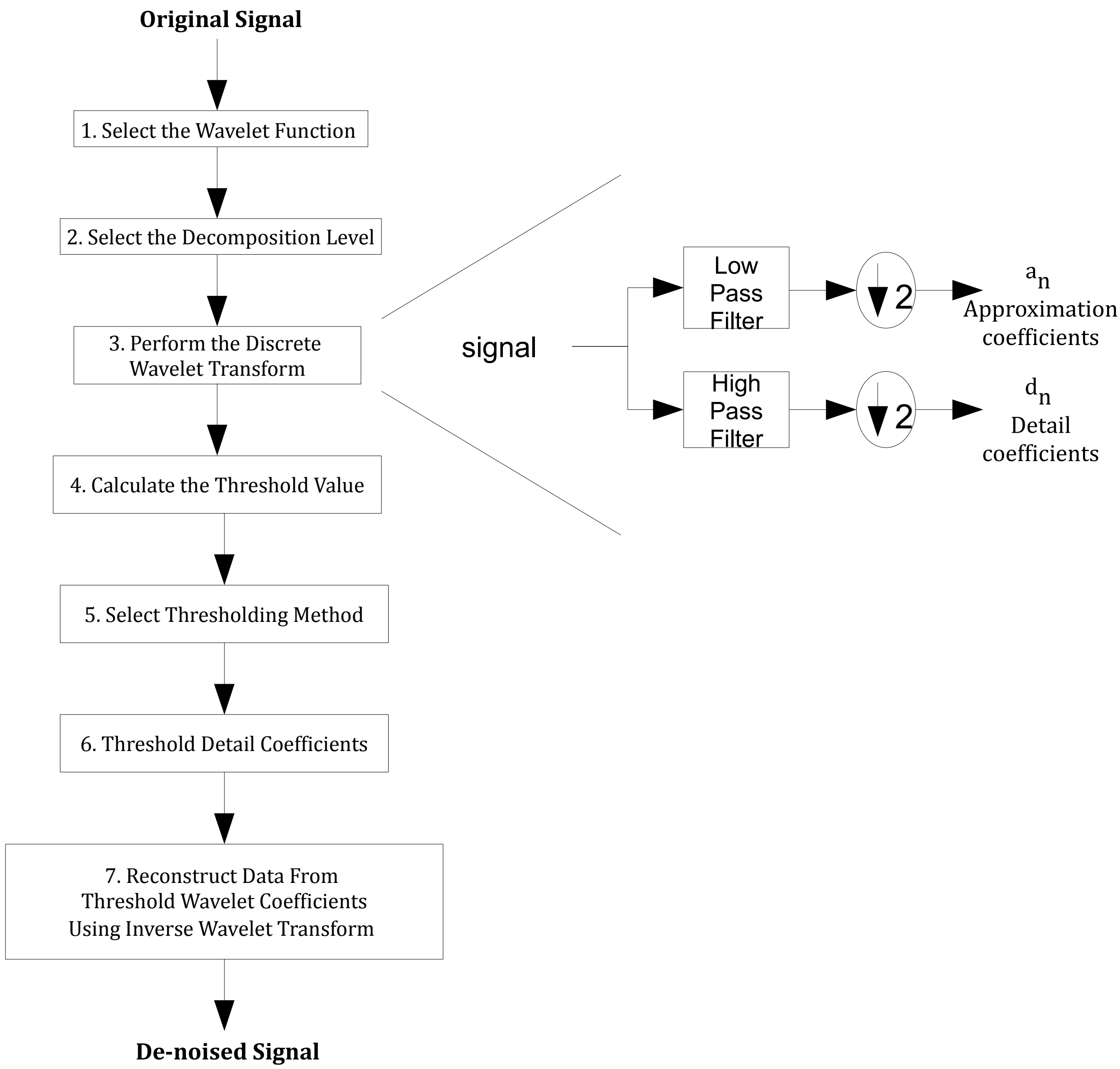

Figure 5.1: Proposed Denoising Algorithm

It is known that the performance of a wavelet based speech denoising method are affected by the level of wavelet decomposition layers[89]. The tinier frequency division is achieved in the

deeper layer. In order to attain the ideal speech denoising result, suitable layers of wavelet packet decomposition should be selected.
The low frequency content is the most important part of the speech signal since the signal's identity is mainly enclosed in it. On the other hand, the high frequency content contains the details of the signal. There are two different methods available for discrete wavelet transform as described in section 4.6. The wavelet packet transform which is a wavelet method that offers more detailed information for signal analysis, is adopted in this research work. A signal is split into an approximation and a detail in the wavelet analysis. Then a second-level approximation and detail are obtained from the approximation and the process is repeated. However, the details as well as the approximations can be split in wavelet packet transform. In theory, the process of analysis can be continued indefinitely because the analysis process is iterative. In practice, the analysis process can continue until there is no sample to analyse.

In order to denoise the signal, thresholding should be performed on the wavelet coefficients. The noise can be eliminated from the noisy signal by thresholding the noisy coefficients to zero. The wavelet threshold denoising algorithm which was proposed by Donoho and described in section 4.7 is adopted in our experiments.

Hard and soft thresholding is applied to the coefficients obtained after wavelet decomposition. Thresholding is applied on the detail's coefficients and the approximate's coefficients are left untouched.

The inverse Wavelet transform can be used to synthesise the signals as described in section 4.6. Synthesis is the process of gathering the main components back into the original signal without, theoretically, any loss of information.

5.1.2 Results

First of all one clean speech sp01.wav is corrupted using white Gaussian noise at the following SNR levels in dB: 0, 5, 10, 15. The results are evaluated using objective and subjective measure approaches.

5.1.2.1 Results Using Objective Measures

The signal-to-noise ratio (SNR) is the most widely used objective measure for speech quality assessment. The global SNR value of an input signal is determined by the following equation:

$$\mathrm{SNRInput}_{db} = 10\log_{10}\left(\frac{\sum_N s^2[N]}{\sum_N n^2[N]}\right) \tag{5.1}$$

where s(N) is a clean speech signal, n(N) is a noise signal and N is the number of samples.

On the other hand, the global SNR value of an output signal is determined by the following equation:

$$\mathrm{SNROutput}_{db} = 10\log_{10}\left(\frac{\sum_N s^2[N]}{\sum_N (s-s')^2[N]}\right) \tag{5.2}$$

As in Equation 5.1, s[N] represents the clean speech signal while s'[N] represents the enhanced speech.

An important condition of using objective measures is that the clean speech and the processed speech must be coordinated during the calculation since there is a time delay after the signal is processed by the enhancement algorithm. Another important objective measure is the computational complexity of the algorithm. Complexity means the number of real multiplications and real additions required by the algorithm[90]. Complexity is evaluated by measuring the time elapsed in each experiment in seconds.

The results of the wavelet based speech enhancement algorithm as applied to the clean speech "sp01.wav" and corrupted by a Gaussian white noise are shown in Table 5.3, Figure 5.2 and Figure 5.3, respectively. Table 5.2 provides the overall results of the simulations for different wavelets. Figure 5.2 and Figure 5.3 provide details about the SNR values and the operation times.

Speech File Name	SNR Input (dB)	SNR output (dB)	Wavelet Family	Thresholding Method	Level	Operation Time (sec)
Sp01.wav	0	5.5	Daub4	Soft	8	0.91
Sp01.wav	0	5.1	Daub4	Hard	8	0.95
Sp01.wav	0	6.1	Daub6	Soft	8	0.99
Sp01.wav	0	5.9	Daub6	Hard	8	1.04
Sp01.wav	**0**	**6.8**	**Daub10**	**Soft**	**8**	**1.2**
Sp01.wav	**0**	**6.1**	**Daub10**	**Soft**	**4**	**0.72**
Sp01.wav	0	6.1	Daub10	Hard	8	1.25
Sp01.wav	0	5.4	Coif1	Soft	8	1.12
Sp01.wav	0	5	Coif1	Hard	8	1.18
Sp01.wav	0	5.9	Coif3	Soft	8	1.22
Sp01.wav	0	5.6	Coif3	Hard	8	1.29
Sp01.wav	0	6.4	Coif5	Soft	8	1.42
Sp01.wav	0	5.9	Coif5	Soft	4	0.96
Sp01.wav	0	5.9	Coif5	Hard	8	1.48
Sp01.wav	0	5.3	Sym6	Soft	8	1.1
Sp01.wav	0	4.9	Sym6	Hard	8	1.18
Sp01.wav	0	5.8	Sym10	Soft	8	1.32
Sp01.wav	0	5.4	Sym10	Hard	8	1.38
Sp01.wav	0	6.2	Sym14	Soft	8	1.54
Sp01.wav	0	5.8	Sym14	Soft	4	1.04
Sp01.wav	0	5.7	Sym14	Hard	8	1.65

Table 5.2: Results of Wavelet based speech enhancement algorithm

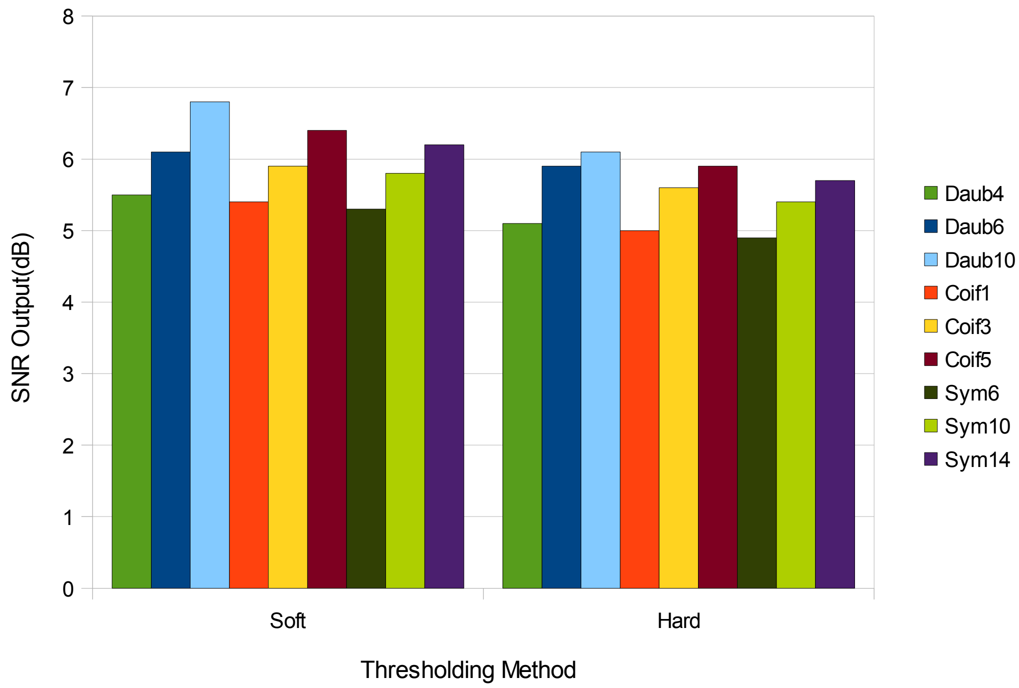

Figure 5.2: SNR output results of the Wavelet based speech enhancement algorithm

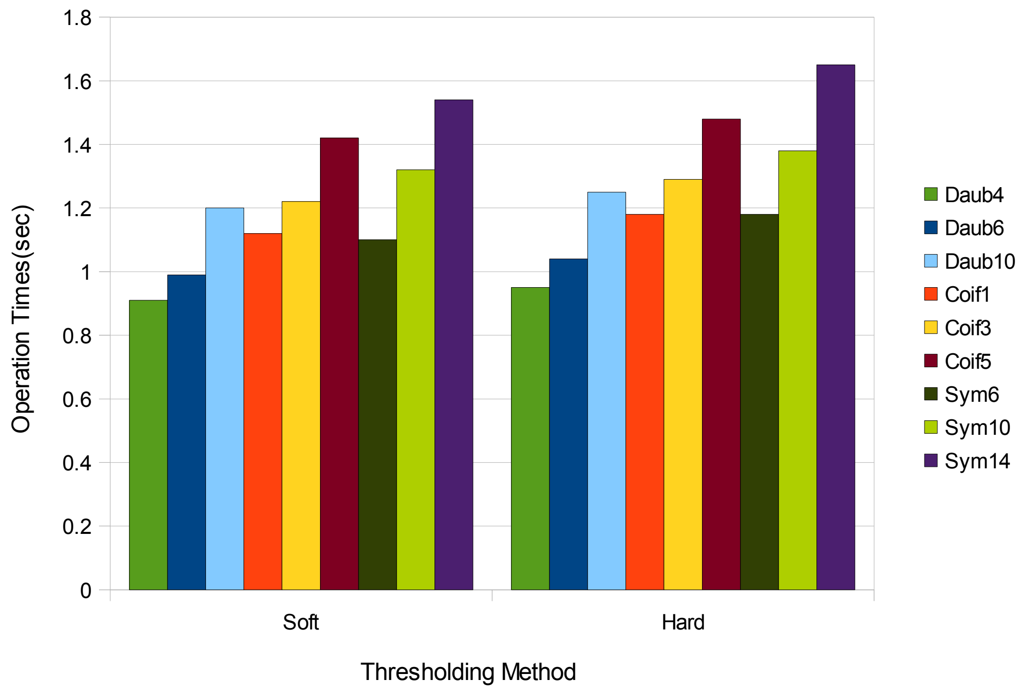

Figure 5.3: Operation times of the Wavelet based speech enhancement algorithm

The results of the wavelet based speech enhancement for a Daub10are shown in Figure 5.4 and Figure 5.5, respectively. Figure 5.4 illustrates the original signal, the corrupted signal and the denoised signal. Figure 5.5 illustrates the spectrum or the frequency domain representation of the signals of Figure 5.4.

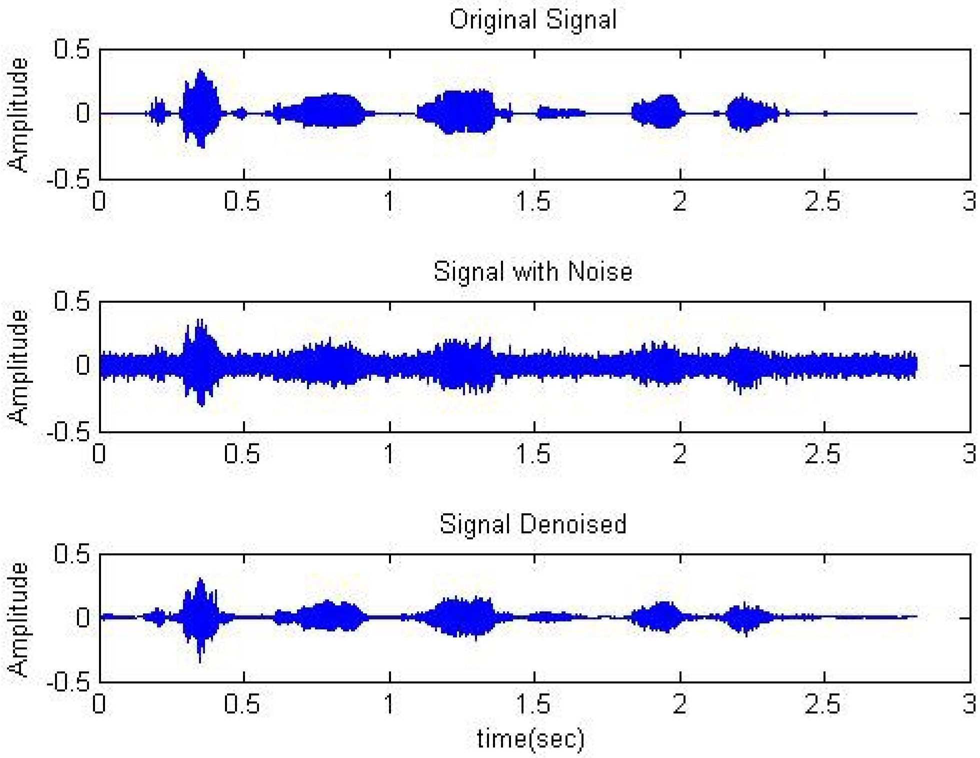

Figure 5.4: Wavelet based speech enhancement using Daub10, level 8 and soft thresholding

The wavelet type was chosen as "Daubechies10", the level was chosen as "eight" and the wavelet thresholding technique was chosen as "soft thresholding" in this example. The input SNR value of the speech was "0dB". The output value of "6.8dB" was achieved as a result. The operational time of the whole process was 1.2 seconds.

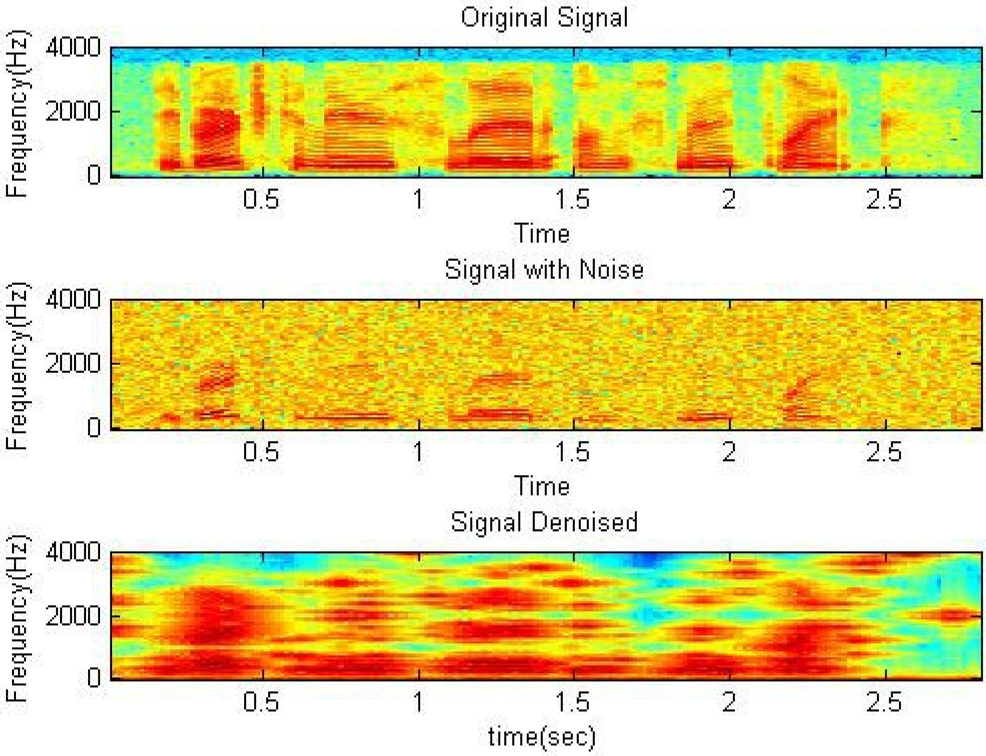

Figure 5.5: Spectrum of original, noisy and denoised signal for Daub10, level 8 and soft thresholding

The best SNR value is achieved using a wavelet "Daub10", with a decomposition of 8 and using the "soft" thresholding method . The operation time increases with the level of decomposition as shown in Table 5.2. Keeping the same settings, the operation time increases from 0.72 seconds at level 4, to 1.2 seconds at level 8 as highlighted in Table 5.2. It is noticed that between level 8 and level 4, there is only a small difference for the output SNR values. The output SNR value after the speech enhancement algorithm is 6.8dB for level 8 and 6.1 dB for level 4. Although the difference in level does not change the SNR value too much, the speech quality decreases drastically as shown in Figure 5.6 and Figure 5.7, respectively.

As shown in Figure 5.7, the speech enhancement algorithm based on daub10 wavelet and using the "soft" thresholding method decreases the speech quality. The spectrum shows the frequency components of the speech before and after the speech enhancement.

While the original speech shown in Figure 5.5 does not have any discontinuities in the spectrum, the denoised signal shown in Figure 5.7 has considerable amount of discontinuities. These discontinuities in the speech frequency map shows that the speech distortion increases while the number of level decreases.

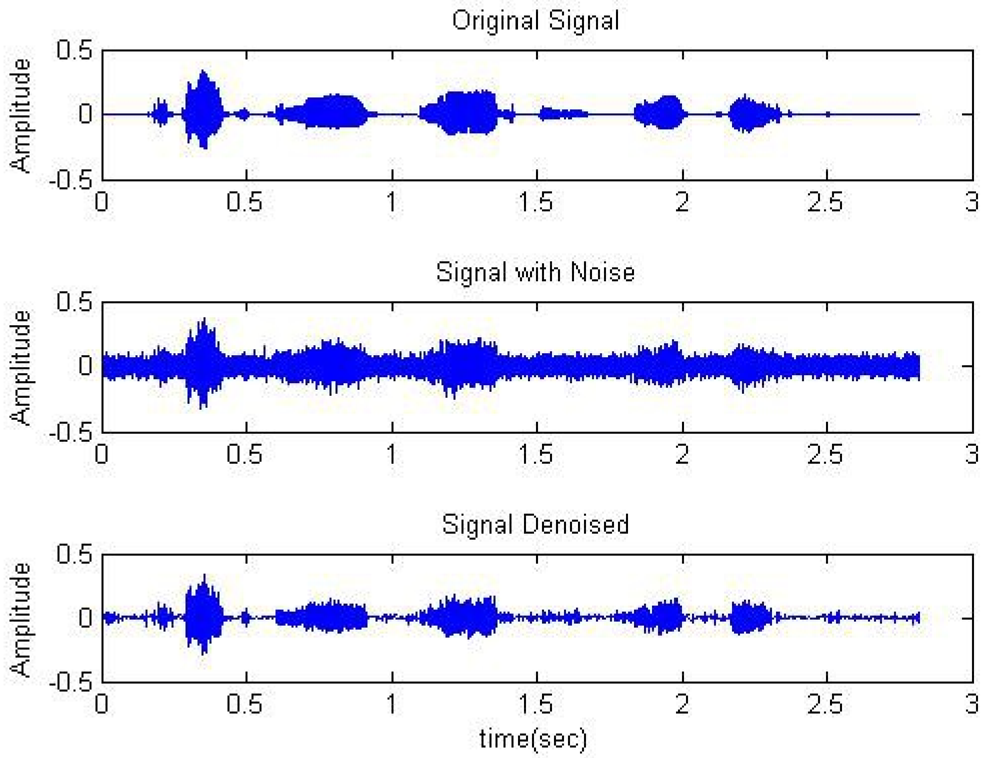

Figure 5.6: Wavelet based speech enhancement using daub10, level 4, soft thresholding

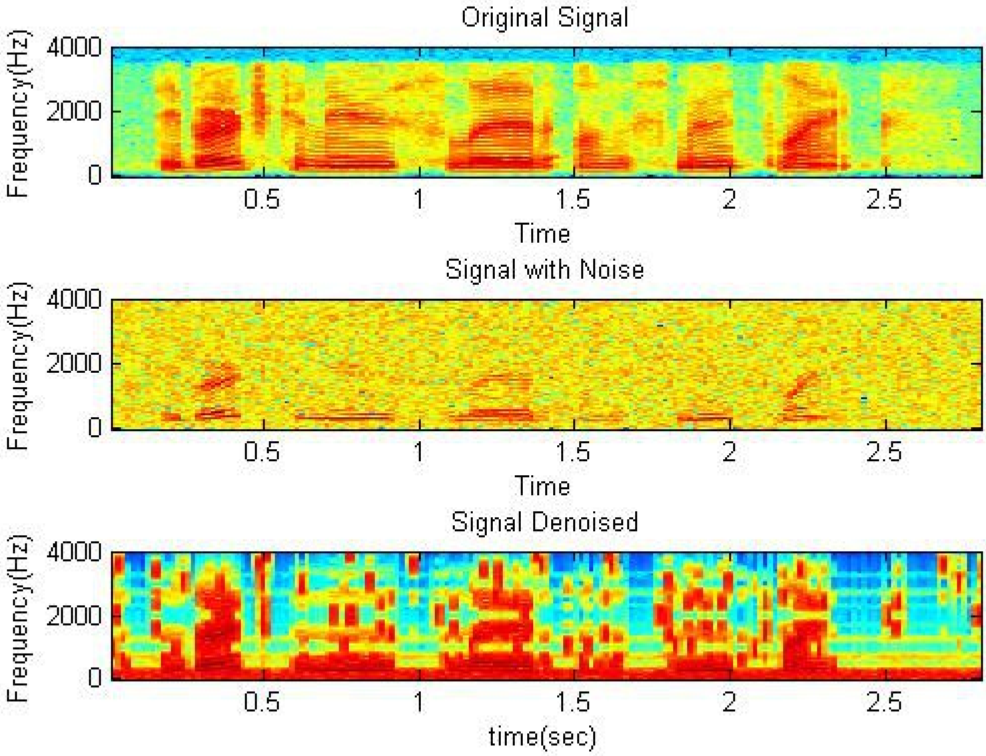

Figure 5.7: Spectrum of original, noisy and denoised signal for Daub10, level 4 and soft thresholding

Three different wavelet families Daubechies 10(Daub10), Coiflet 5(Coif5) and Symlet 14(Sym14) were used to compare the results after each speech enhancement. The wavelet level was 8 and the thresholding method was selected as soft thresholding on all the experiments as shown in both Figure 5.8 and 5.9. The speech samples sp01.wav, sp11.wav and sp07.wav, sp16.wav were corrupted by white Gaussian noise, respectively. The best SNR output results were obtained with Daub10 and the output SNR result is not much different from the results of Coif5 and Sym14. The main difference has appeared on the operation times as seen in Figure 5.10 and 5.11. The operation time of the wavelet based speech enhancement algorithm was the shortest with the Daub10. The experiments shows that the number of coefficients affects the operational time as expected. Since the Daub10 has the least number of coefficients, the algorithm using it completes quicker than the others. Also, the operation time increases in parallel with the wavelet level number as seen in Figure 5.10 and 5.11.

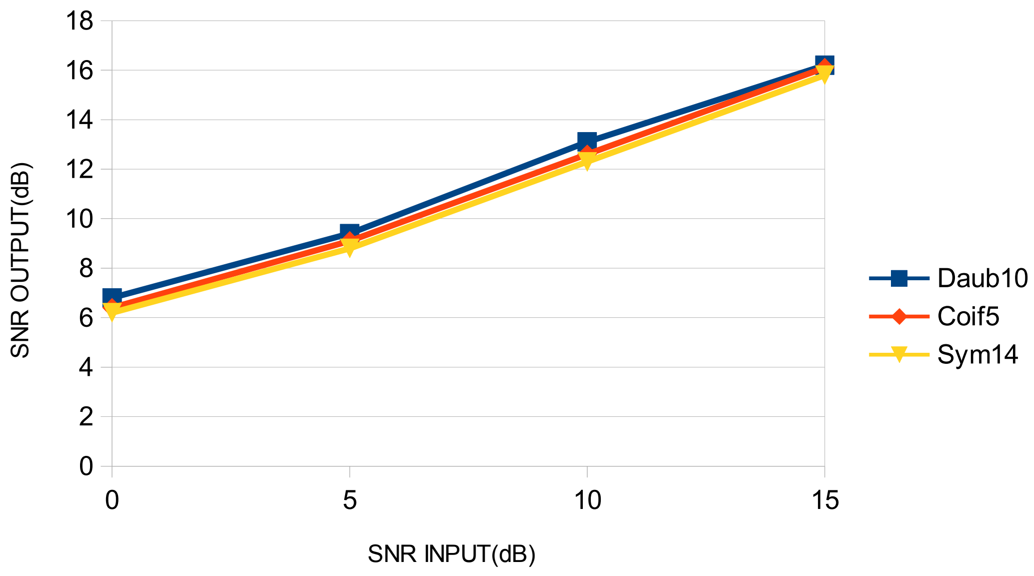

Figure 5.8: SNR results for sp01.wav, corrupted with white Gaussian noise.

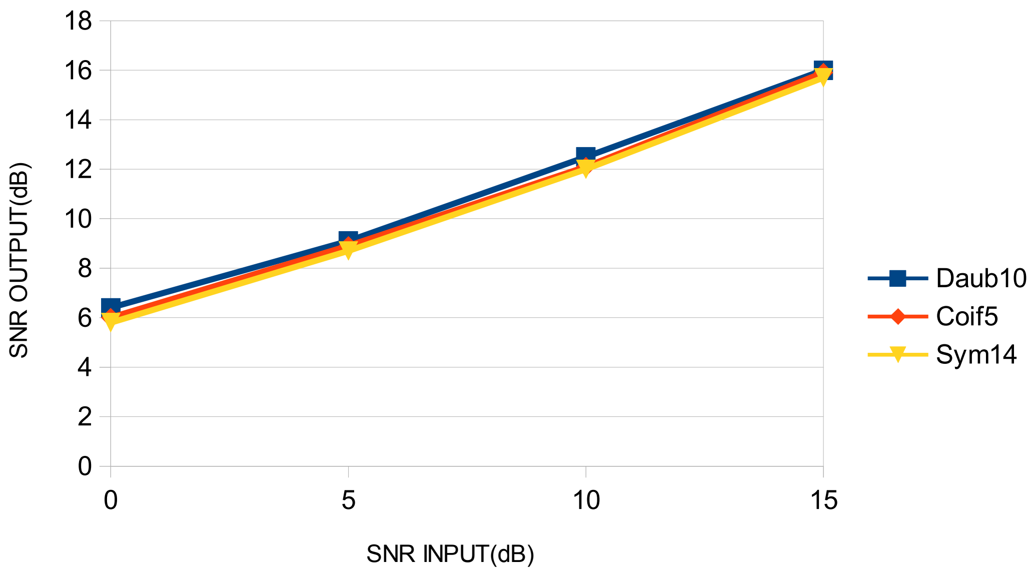

Figure 5.9: SNR results for sp11.wav, corrupted with white Gaussian noise.

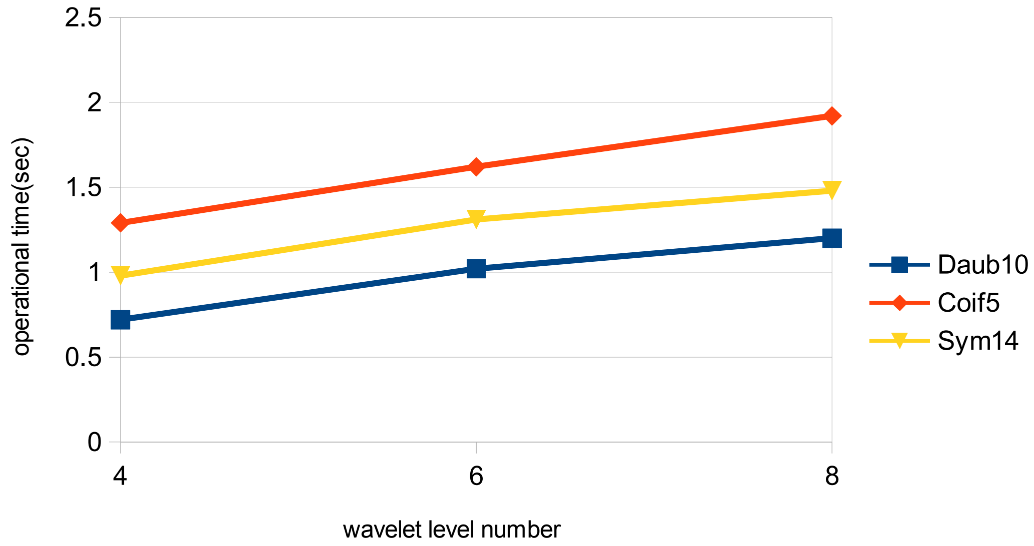

Figure 5.10: Operational time for sp07.wav, corrupted with white Gaussian noise.

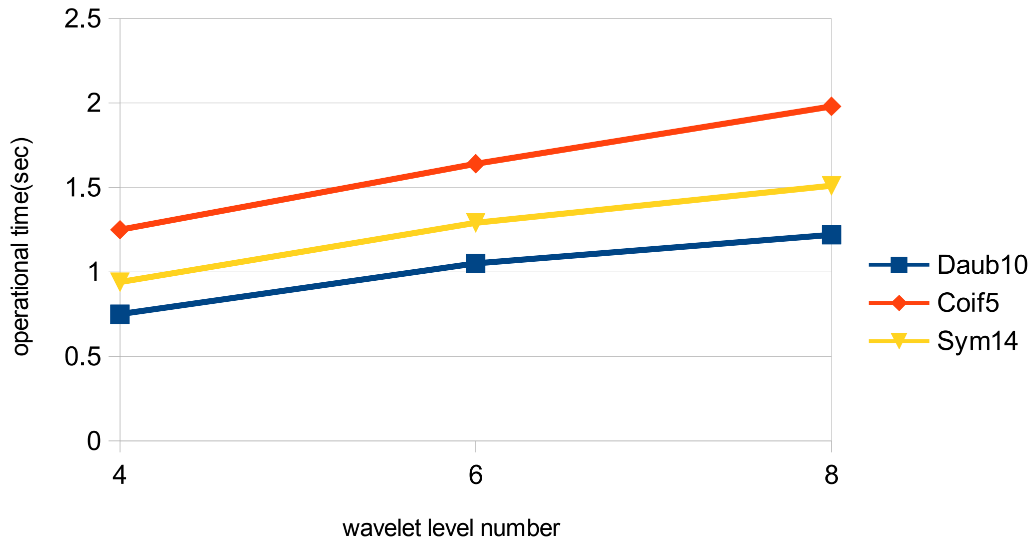

Figure 5.11: Operational time for sp16.wav, corrupted with white Gaussian noise.

5.1.2.2 Results Using Subjective Measures

A subjective judgement by a listener determines the speech quality result after completion of the speech enhancement algorithm. One of the best subjective test method is the Absolute Category Rating (ACR) Test[91].

A group of listeners judge a series of audio files using a five grade scale varying from 1 to 5:

5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

The Mean Opinion Score (MOS) is calculated for each audio file after collecting individual scores. ACRs are averaged to obtain MOS as a quantitative indicator of the system performance. The test should be organised under regulated conditions using quiet environment. High number of listeners increases the stability of the scores.

Table 5.3 shows the MOS scores from an actual ACR test with 8 listeners with no previous familiarity with test materials.

Speech File Name	SNR Input (dB)	SNR output (dB)	Wavelet Family	Thresholding Method	Level	**MOS**
Sp01.wav	0	5.5	Daub4	Soft	8	**4.000**
Sp01.wav	0	5.1	Daub4	Hard	8	**4.000**
Sp01.wav	0	6.1	Daub6	Soft	8	**4.250**
Sp01.wav	0	5.9	Daub6	Hard	8	**4.125**
Sp01.wav	**0**	**6.8**	**Daub10**	**Soft**	**8**	**4.750**

Sp01.wav	0	6.1	Daub10	Hard	8	**4.625**
Sp01.wav	0	5.4	Coif1	Soft	8	**4.000**
Sp01.wav	0	5	Coif1	Hard	8	**3.750**
Sp01.wav	0	5.9	Ciof3	Soft	8	**4.000**
Sp01.wav	0	5.6	Coif3	Hard	8	**3.500**
Sp01.wav	0	6.4	Coif5	Soft	8	**4.500**
Sp01.wav	0	5.9	Coif5	Hard	8	**4.250**
Sp01.wav	0	5.3	Sym6	Soft	8	**4.000**
Sp01.wav	0	4.9	Sym6	Hard	8	**3.750**
Sp01.wav	0	5.8	Sym10	Soft	8	**4.000**
Sp01.wav	0	5.4	Sym10	Hard	8	**3.750**
Sp01.wav	0	6.2	Sym14	Soft	8	**4.250**
Sp01.wav	0	5.7	Sym14	Hard	8	**3.875**

Table 5.3: MOS results of the Wavelet based speech enhancement algorithm

For these tests, the sp01.wav file is corrupted with a 0dB white Gaussian noise. The sp01.wav clean audio file is rated 5 by listeners and 1.25 after 0dB white Gaussian noise corruption.

Different wavelet families, thresholding methods and levels are rated for speech quality measurement. Although the SNR values are similar for different wavelet levels, the speech quality gets better when increasing the wavelet level.

To assess the impact of different SNR values, sp21.wav corrupted using a white Gaussian noise. The clean sp21.wav audio file was rated as 5 by listeners.

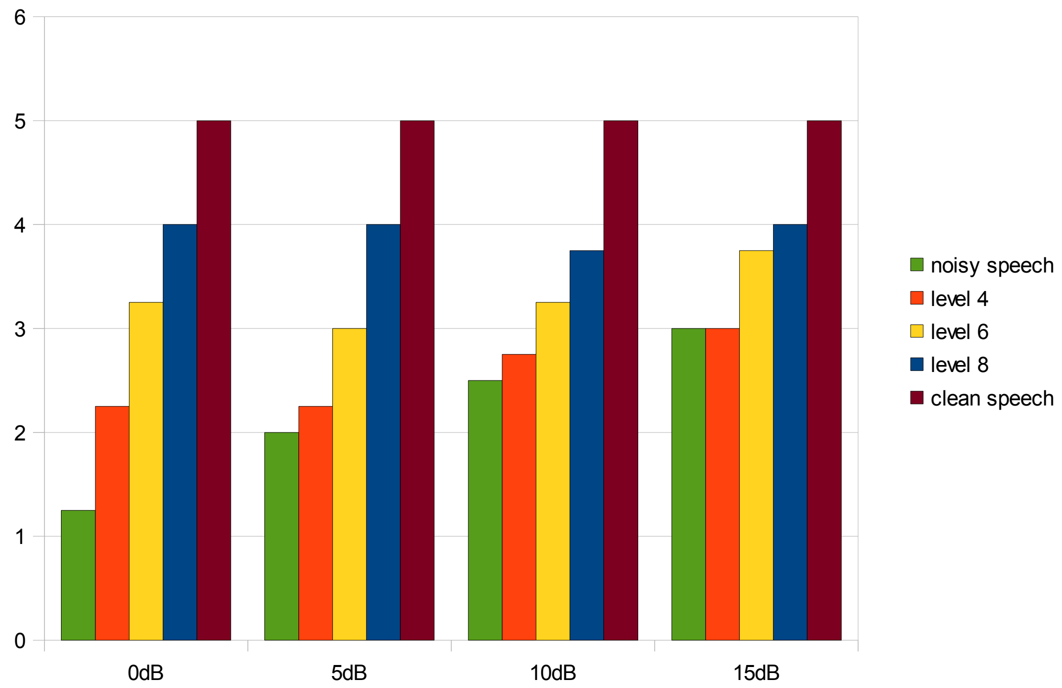

Figure 5.12: MOS results of Wavelet based speech enhancement algorithm

The file corrupted with 0dB, 5dB, 10dB and 15dB white Gaussian noise was rated 1.25, 2, 2.5 and 3, respectively. While the speech quality improvement is considerably higher in the case of the 0dB corrupted audio file, there is a minor speech quality improvement in the 15dB corrupted audio file. Since the MOS values are already high when the speech is corrupted with 15dB noise, the speech quality does not increase a lot.

5.2 Automating the Wavelet Based Speech Enhancement Algorithm

After the implementation of the wavelet based speech enhancement algorithm on Matlab, the same algorithm is automated using C. A GUI has been devised for this purpose. The GUI consists of four different parts as shown in Figure 5.13. While the upper window shows the noisy audio signal, the lower window shows the enhanced audio signal. For both windows, the audio data is represented in volts vs time(sec) and scaled between 0 and 1 (normalised). The lower area, underneath the lower window, allows the user to experiment with different settings, including the type of wavelet, the level of decomposition and the threshold value. It also provides the user with the opportunity to choose either to use an offline file or a real time audio file. The other functionalities include, a threshold calculation button, a filter button to start the offline denoising algorithm, an exit bar for closing the application and a status bar to show the status of the operation.

Six different utterances with white Gaussian noise from a noisy speech corpus (NOIZEUS) library and three different speeches with 6 different noise types from SpEAR Database were used to evaluate the wavelet packet decomposition thresholding algorithm. The same speech files from chapter 5.1 are chosen to make a comparison with Matlab performance of the algorithm.

5.2.1 Implementation

The wavelet speech enhancement algorithm which has been tested successfully in Matlab is implemented in C to allow the algorithm to be ported to a hardware platform easily. The user interface of the program is designed using GTK as shown in Figure 5.13.

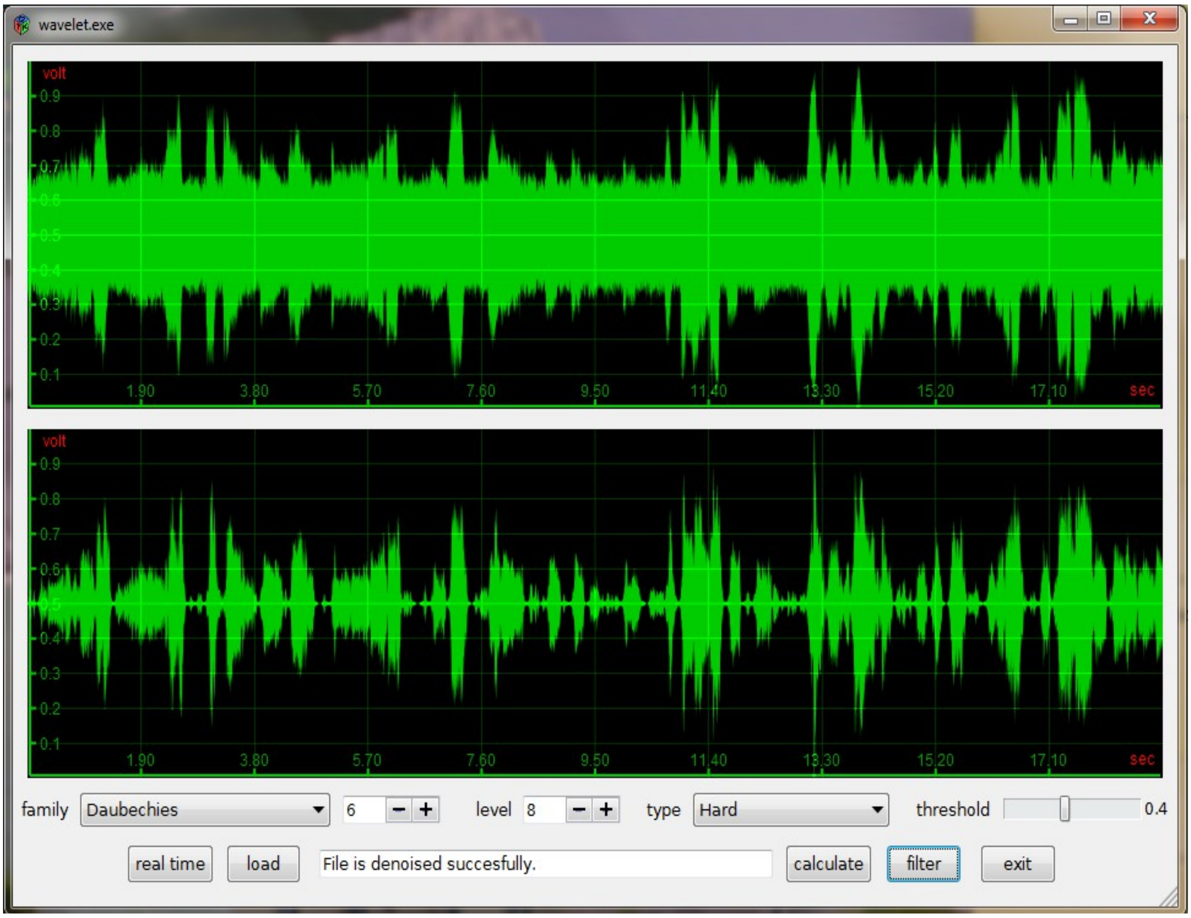

Figure 5.13: The wavelet speech enhancement graphical user interface

The program consists of 8 main blocks as shown in Figure 5.14.

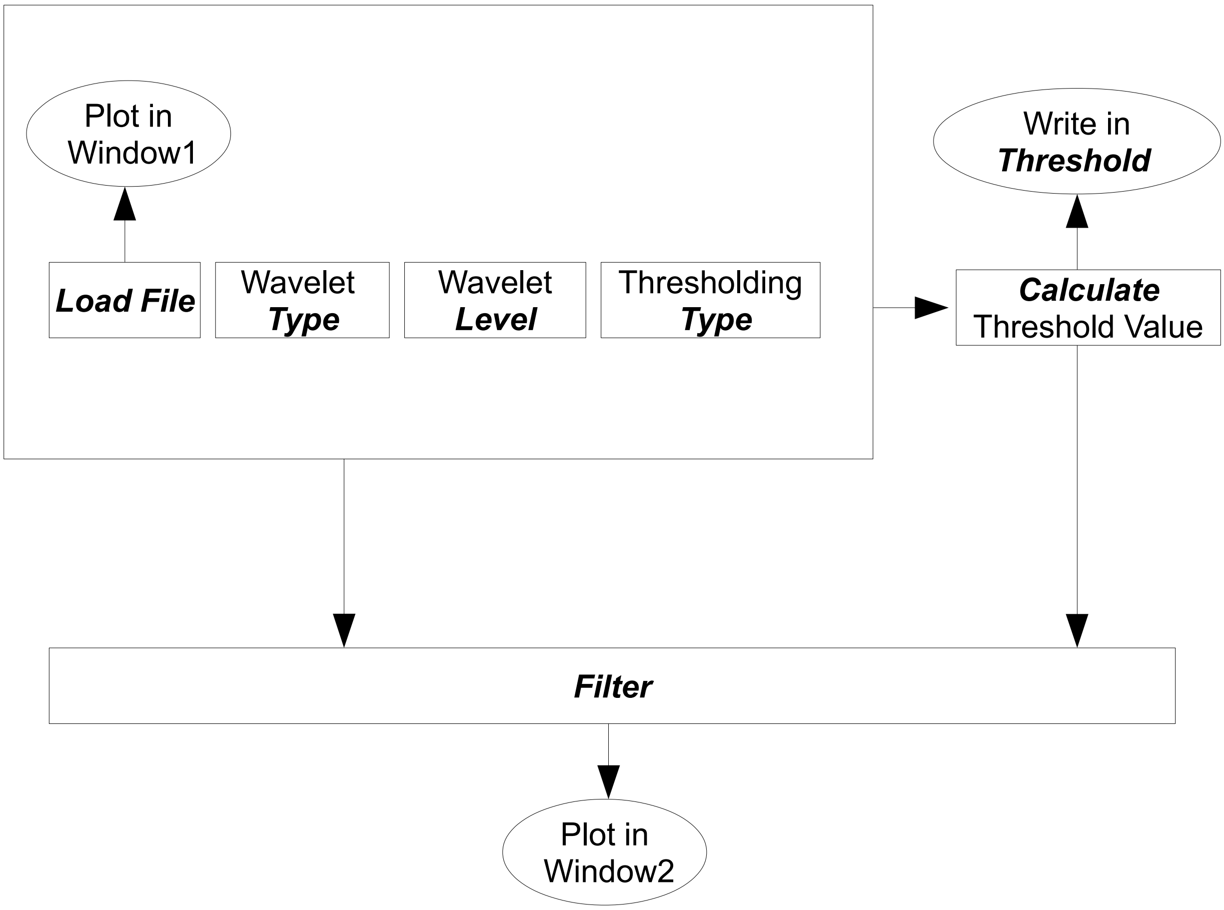

Figure 5.14: The block diagram of C implementation

1. **Load the noisy file**: Any audio file with a wav extension can be loaded using load button as seen in Figure 5.15. The selected noisy file is stored in a buffer in preparation of its enhancement. The selected file is then plotted in window 1. If the file is read and plotted successfully, The message "File is loaded successfully" is written to the message box.

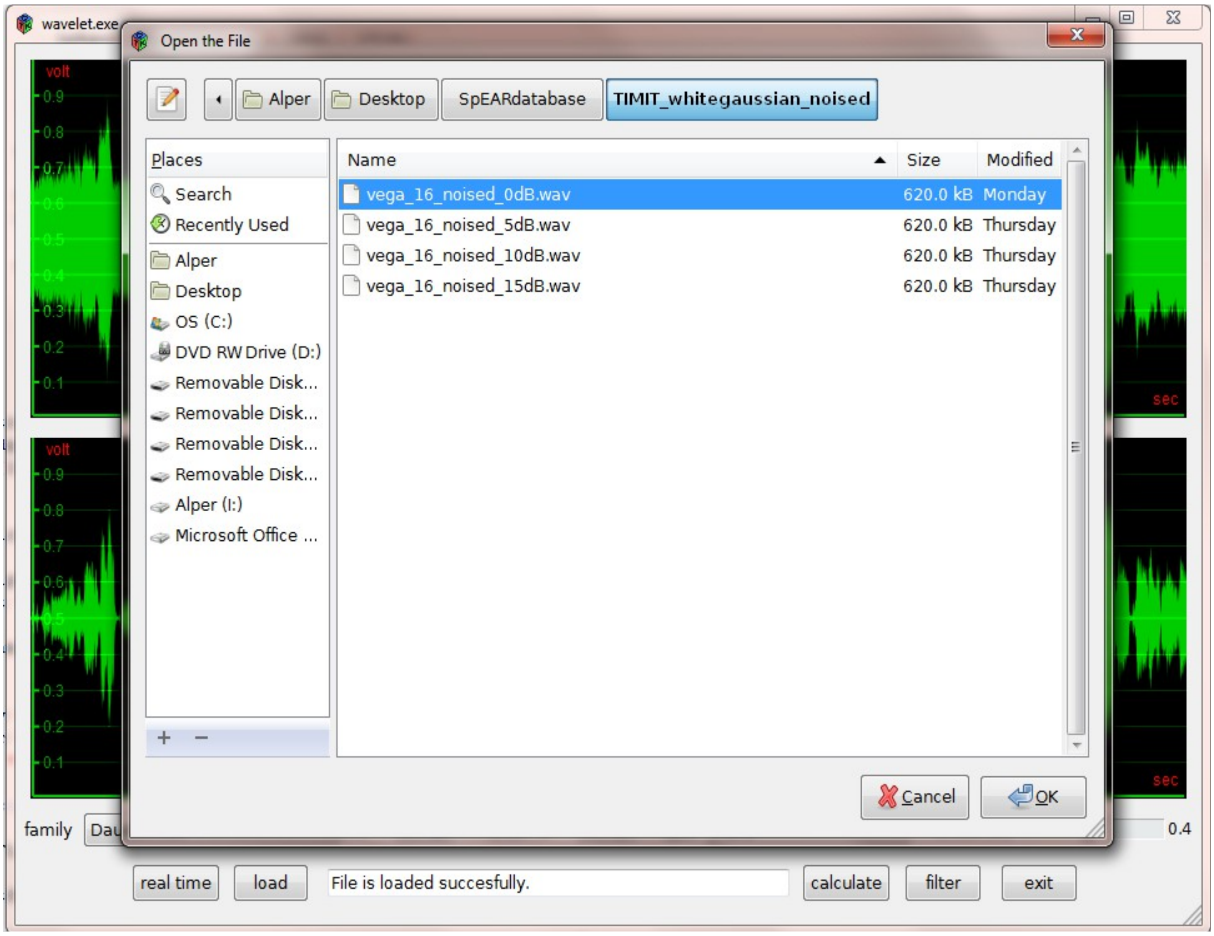

Figure 5.15: Loading a file

2. **Select Wavelet Type:** The wavelet type can be chosen using family dialog box. There are 24 different wavelet families in the dialog box which are Daubechies1-10, Symlets2-10 and Coiflets 1-5. More wavelet families can be added by the user using the GUI. The four different wavelet type filter values are saved in "xwaveletfamilies.h" as arrays. Daub1 coefficients are provided as an example as shown below.

```
static floatanalysis_db1_lp[2] = {0.7071067811865476,0.7071067811865476};
static float analysis_db1_hp[2] = {-0.7071067811865476,0.7071067811865476};
static float synthesis_db1_lp[2] = {0.7071067811865476,0.7071067811865476};
static float synthesis_db1_hp[2] = {0.7071067811865476,-0.7071067811865476};
```

3. **Select the Wavelet Level:** The wavelet level can be chosen using level dialog box. Since the number of samples is divided by 2 at each level, the sample value of the file must be multiples by 2^n where n is the level number. This is why, the sample number is checked against level number when a file is loaded. If the sample number is not a multiple of 2, the speech signal is truncated by ignoring the last values.

4. **Select the Threshold Type:** The thresholding type can be selected using the type dialog box. Two different types of thresholding can be selected: soft or hard thresholding. These two methods are explained in detail in chapter 4.7.1.

5. **Calculate Threshold Value:** The threshold value can be calculated using calculate button. "*XWaveletCalculate(char *faFilePath, int faLevel, float *faAnalysis_lp, float *faAnalysis_hp, int faFilter_size)*" function is called to calculate the threshold value. The selected file is analysed using the wavelet level and wavelet type. The threshold nodes are saved in a temporary buffer as shown in Figure 5.16. Each node stores sample size/2^n value. The default threshold value is calculated using 2^{n-4} th node using the formula described in chapter 4.7.1. The calculated value is written into the threshold section and can then be tuned further using the slide bar for filtering.

1	2	3	4	5	6	- - - - - - - - - - - - - - - - - - - -	2^{n-1}	2^n

Figure 5.16: The temporary buffer after wavelet analysis

6. **Filter the audio file:** The loaded audio file can be filtered using "filter" button. The chosen wavelet type, wavelet level, threshold type and the calculated threshold value is used for filtering the audio file by the fuction *XwaveletFilter (char *faFilePath, int faLevel, float *faAnalysis_lp, float *faAnalysis_hp, float *faSynthesis_lp, float *faSynthesis_hp, int faFilter_size, double faThreshold, XTH_TYPE faType, XFamily faXFamily,int faNumber).*

The filtering process consists of 3 stages as seen in Figure 5.17.

Figure 5.17: Filtering steps

Wavelet analysis can be simply achieved by a tree of digital filter banks. Filter banks have been playing a central role in the area of wavelet analysis. The selection of desired scaling functions and mother wavelets reduces the design process to lowpass and highpass filtering, which represents the two-channel perfect reconstruction (PR) filter bank. The wavelet transform can simply be described as a tree of two-channel PR filter banks as shown in Figure 5.18.

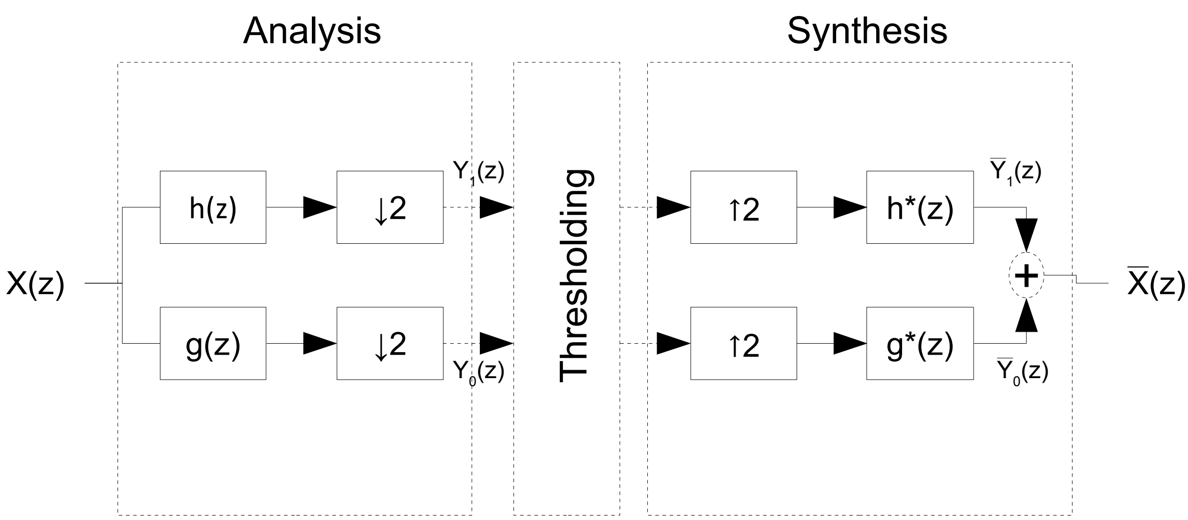

Figure 5.18: One level two channel PR filter banks

where g(z) and g*(z) denote the lowpass filters, while h(z) and h*(z) denote the highpass filters of the PR filter bank, respectively.

As shown in Figure 5.16, the signal X(z) is first analysed by a filter bank consisting of g(z) and h(z), called analysis filters. The outputs of g(z) and h(z) are downsampled by 2 to obtain $Y_0(z)$ and $Y_1(z)$. After thresholding, the modified signals are upsampled and filtered by another filter bank consisting of g*(z) and h*(z) called synthesis filters.

The downsampling operators are decimators and the upsampling operators are expanders. If no processing takes place between the two filter banks, the output signal $\overline{X(z)}$ is identical to the original signal X(z), except for a time delay. Such a system is commonly referred to as a two-channel perfect reconstruction filter bank.

The output $\overline{X}(z)$ is:

$$\bar{X}(z) = \frac{1}{2}\left[g^*(z)g(z) + h^*(z)h(z)\right]^* X(z) + \frac{1}{2}\left[g^*(z)g(-z) + h^*(z)h(-z)\right]^* X(z) \quad (5.3)$$

where one term involves $X(z)$ and the other involves $X(-z)$. For perfect reconstruction, the term with $X(-z)$, traditionally called the *alias term*, must be zero. To achieve this, we need

$$[g^*(z)g(-z) + h^*(z)h(-z)] = 0 \quad (5.4)$$

To fullfil the equality of Equation 5.4, Equation 5.5 must be satisfied.

$$g^*(z) = h(-z) \quad \text{and} \quad h^*(z) = -g(-z) \quad (5.5)$$

There are many different wavelet families and the ones which is used in our experiments fulfil the Equation 5.5. The low pass filter and high pass filter coefficient values are shown in Appendix A.

The low pass filter coefficients are shown as (g0 g1 g2 g3 g4 g5 g6 g7) and the high pass filter coefficients are shown as (h0 h1 h2 h3 h4 h5 h6 h7) in Figure 5.19. They are representing a wavelet type which has 8 coefficients as it would change for the other wavelet families.

Wavelet analysis is achieved using algorithm shown in Figure 5.19, which has especially been selected, instead of matrix calculations, as it leads to low memory usage. g[n] is the low pass analysis filter array and h[n] is the high pass analysis filter array. x[n] represents the input audio samples. a[n] represents the approximation values while d[n] represents the detail values. Downsampling is achieved by shifting the filter values to the left by 2.

$$\begin{pmatrix} g0\ g1\ g2\ g3\ g4\ g5\ g6\ g7 & & \\ h0\ h1\ h2\ h3\ h4\ h5\ h6\ h7 & & \\ & g0\ g1\ g2\ g3\ g4\ g5\ g6\ g7 & \\ & h0\ h1\ h2\ h3\ h4\ h5\ h6\ h7 & \\ & & g0\ g1\ g2\ g3\ g4\ g5\ g6\ g7 \\ & & h0\ h1\ h2\ h3\ h4\ h5\ h6\ h7 \\ & & \dots\dots \\ & & \dots\dots \end{pmatrix} \begin{pmatrix} x0 \\ x1 \\ x2 \\ x3 \\ . \\ . \\ . \\ . \end{pmatrix} = \begin{pmatrix} a0 \\ d0 \\ a1 \\ d1 \\ . \\ . \\ . \\ . \end{pmatrix}$$

Figure 5.19: Wavelet Analysis First Level

The analysis continues in a similar manner until the wavelet level value is achieved. The approximation and detail values are saved in an array in each layer. Since the memory usage increases with the level number, the same array is used in each level in this work as shown in Figure 5.20. While the approximation values are stored in the first half of the lBuffer, the details are stored in the second half of the lBuffer. When the second level is processed, the lbuffer is used as an input to lowpass and highpass filtering. The output of the filtering is also stored in the same lBuffer. At the end of the analysis process, lBuffer is passed to the next stage.

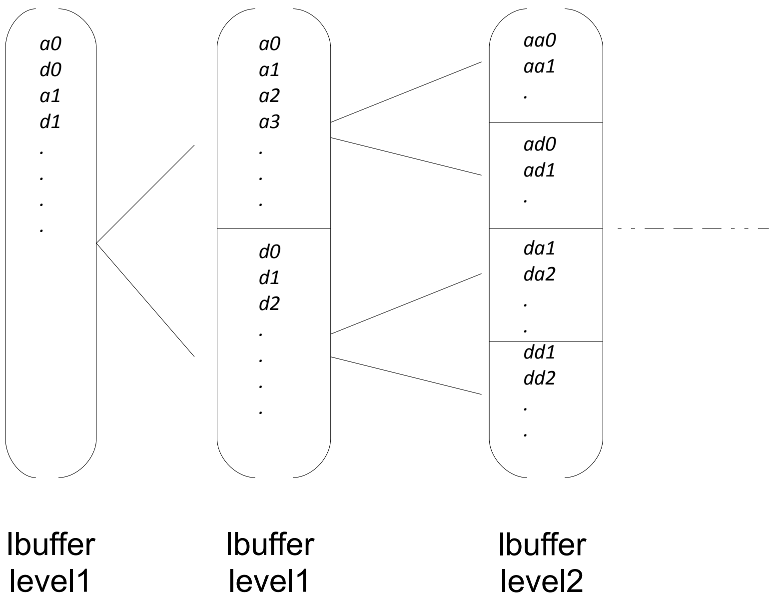

Figure 5.20: Wavelet Analysis First Layer

Wavelet thresholding takes the lBuffer, the thresholding type and thresholding value as an input of the thresholding stage. Soft or hard thresholding methods can be selected by the user and the thresholding is applied as described in chapter 4.7.1.

The synthesis stage is required for perfect reconstruction. The process is shown in Figure 5.21. It works in a similar manner as the analysis stage

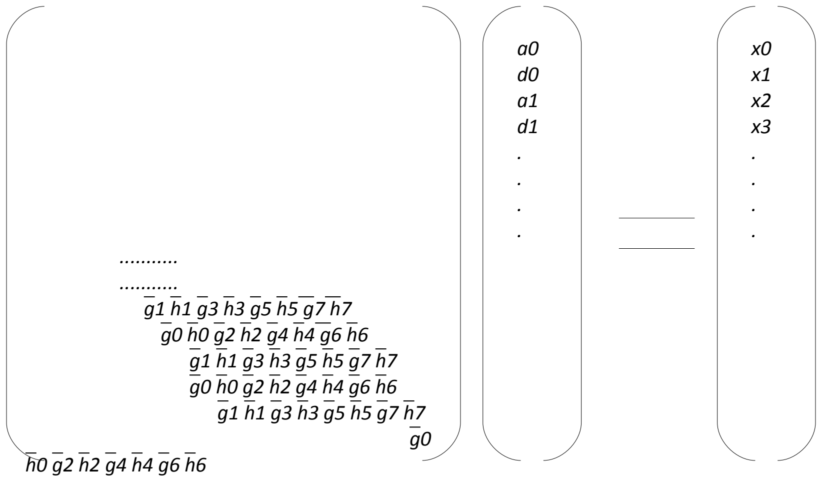

Figure 5.21: First level wavelet synthesis

5.2.2 Results of Automated Wavelet Based Speech Enhancement Algorithm

First of all one clean speech sp01.wav is corrupted using white Gaussian noise at the following SNR levels in dB: 0, 5, 10, 15. The results are evaluated using as objective measures and subjective measures, respectively.

5.2.2.1 Objective Measure

A two minutes long audio file sp01.wav is corrupted by a Gaussian white noise. The results of the wavelet based speech enhancement algorithm are shown in Table 5.4.

Speech File Name	SNR Input (dB)	SNR output (dB)	Wavelet Family	Thresholding Method	Level	Operation Time (msec)
Sp01.wav	0	5.6	Daub4	Soft	8	29
Sp01.wav	0	5.1	Daub4	Hard	8	32
Sp01.wav	0	6.1	Daub6	Soft	8	36
Sp01.wav	0	5.8	Daub6	Hard	8	41
Sp01.wav	**0**	**6.7**	**Daub10**	**Soft**	**8**	**54**
Sp01.wav	0	6.1	Daub10	Hard	8	57
Sp01.wav	0	5.5	Coif1	Hard	8	52
Sp01.wav	0	5	Coif1	Soft	8	54
Sp01.wav	0	5.9	Ciof3	Hard	8	59
Sp01.wav	0	5.7	Coif3	Soft	8	63
Sp01.wav	0	6.4	Coif5	Hard	8	71
Sp01.wav	0	5.9	Coif5	Soft	8	72
Sp01.wav	0	5.4	Sym6	Hard	8	50
Sp01.wav	0	4.9	Sym6	Soft	8	54
Sp01.wav	0	5.7	Sym10	Hard	8	66
Sp01.wav	0	5.4	Sym10	Soft	8	70
Sp01.wav	0	6.2	Sym14	Hard	8	86
Sp01.wav	0	5.8	Sym14	Soft	8	90

Table 5.4: Results of the Wavelet based speech enhancement algorithm in C

As expected the results are similar to the Matlab simulations except the operation times. Operation time results are improved in C as it is clear from Table 5.4.

Sample wavelet thresholding is shown in Figure 5.22. The wavelet type was chosen as "Daubechies10", the level was chosen as "eight" and the wavelet thresholding technique was chosen as "soft thresholding" in this example. The input SNR value of the speech was "0dB". The output value of "6.7dB" was achieved as a result. The operational time of the whole process was 54 milliseconds.

Three different wavelet families Daubechies 10(Daub10), Coiflet 5(Coif5) and Symlet 14(Sym14) were used to compare the results after each speech enhancement. The wavelet level was 8 and the thresholding method was soft thresholding on all the experiments. The speech samples sp01.wav and sp11.wav were corrupted by white Gaussian noise respectively. The SNR results obtained in C implementation are very similar to the ones obtained in Matlab implementation. This is why, the SNR value chart of C implementation is not shown again.

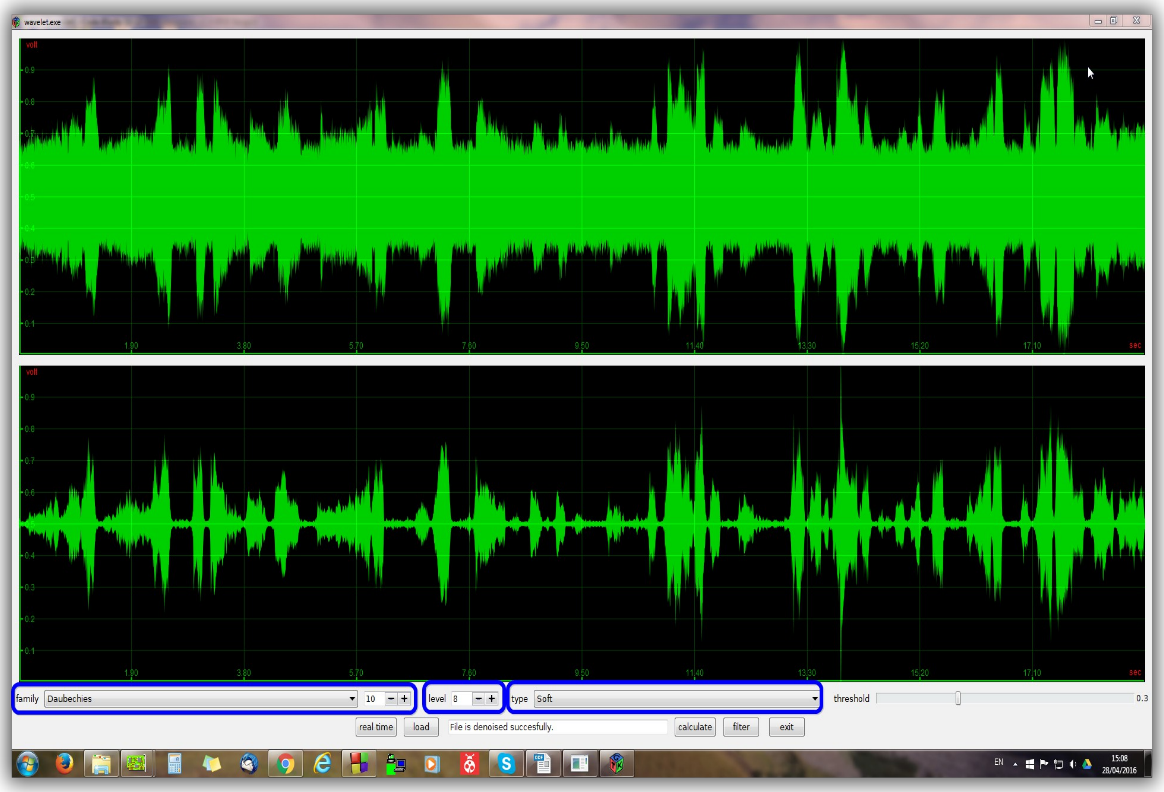

Figure 5.22: Waveform of noisy and denoised signals

The best SNR output results were obtained with Daub10 with not much difference between the noisy signal and the enhanced signal. The main difference has appeared on the operation times as seen in Figure 5.23. The operational time of the wavelet based speech enhancement algorithm was the shortest with the Daub10. The experiments shows that the coefficient number affects the operational time as expected. Since the Daub10 has the least coefficient number, the algorithm using it finishes quicker than the others. Also, the operational time increases in parallel with the wavelet level number as seen in Figure 5.23.

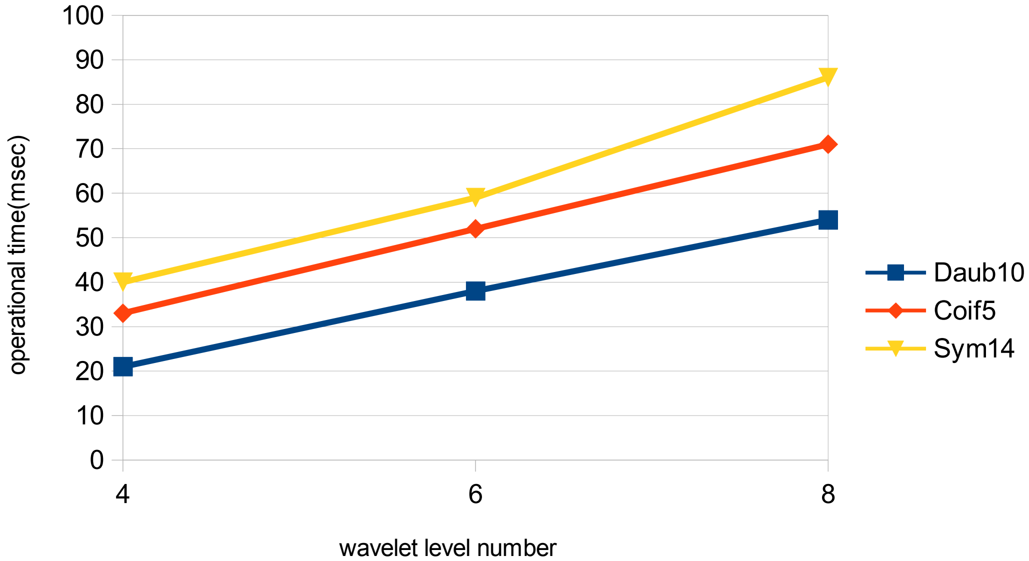

Figure 5.23: Operational time in white Gaussian noise case for sp07.wav

5.2.2.2 Results Using Subjective Measure

The Mean Opinion Score (MOS) is calculated as in section 4.4.1.2 for each audio file after collecting individual scores. The test should be organised under regulated conditions using quiet environment.

Table 5.5 shows the MOS scores from an actual ACR test with 8 listeners with no previous familiarity with test materials.

Speech File Name	SNR Input (dB)	SNR output (dB)	Wavelet Family	Thresholding Method	Level	**MOS**
Sp01.wav	0	5.6	Daub4	Soft	8	**4.000**
Sp01.wav	0	5.1	Daub4	Hard	8	**3.625**
Sp01.wav	0	6.1	Daub6	Soft	8	**4.000**
Sp01.wav	0	5.8	Daub6	Hard	8	**3.750**
Sp01.wav	**0**	**6.7**	**Daub10**	**Soft**	**8**	**4.250**
Sp01.wav	0	6.1	Daub10	Hard	8	**4.000**
Sp01.wav	0	5.5	Coif1	Soft	8	**3.500**
Sp01.wav	0	5	Coif1	Hard	8	**3.250**
Sp01.wav	0	5.9	Ciof3	Soft	8	**4.000**
Sp01.wav	0	5.7	Coif3	Hard	8	**3.750**
Sp01.wav	0	6.4	Coif5	Soft	8	**4.250**
Sp01.wav	0	5.9	Coif5	Hard	8	**4.000**
Sp01.wav	0	5.4	Sym6	Soft	8	**3.750**
Sp01.wav	0	4.9	Sym6	Hard	8	**3.500**
Sp01.wav	0	5.7	Sym10	Soft	8	**4.000**
Sp01.wav	0	5.4	Sym10	Hard	8	**3.500**
Sp01.wav	0	6.2	Sym14	Soft	8	**4.000**
Sp01.wav	0	5.8	Sym14	Hard	8	**3.750**

Table 5.5: MOS results of the Wavelet based speech enhancement algorithm

The sp01.wav file is corrupted with 0dB white Gaussian noise for the tests. The sp01.wav clean audio file is rated 5 by listeners and 1.25 after 0dB white Gaussian noise corruption.

Different wavelet families, thresholding methods and levels are rated for speech quality measurement. Although the SNR values are similar for different wavelet levels, the speech quality gets better with an increase of the wavelet level.

The test was carried out with different SNR values with sp21.wav audio file as seen in Figure 5.24. The clean sp21.wav audio file was rated 5 by all listeners.

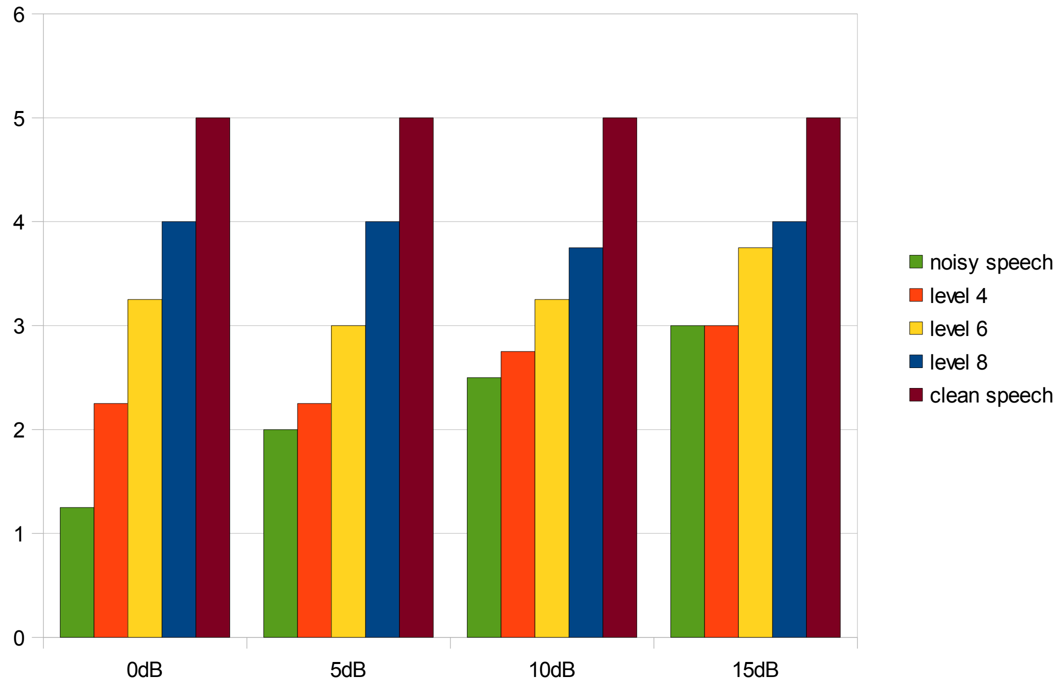

Figure 5.24: MOS results of the Wavelet based speech enhancement algorithm

The file corrupted with 0dB, 5dB, 10dB and 15dB white Gaussian noise was rated 1.25, 2, 2.5 and 3 respectively. While the speech quality improvement is considerably higher in the 0dB corrupted audio file, there is little speech quality improvement in the 15dB corrupted audio file.

5.2.3 Results of Automated Wavelet Based Speech Enhancement Algorithm with different types of noises

Three different speech signals with six different noise types from the SpEAR Database were used to evaluate the wavelet packet decomposition thresholding algorithm to compare the results of different noise types. The SpEAR Database contains carefully selected samples of noise corrupted speech with clean speech references. All speech and noise sources have been acoustically combined and re-recorded. Synchronous clocking is used to provide an exact time-aligned reference to the clean speech signal which is important for SNR calculations.

Playbacks and recordings of the speech signals were done at 48KHz by upsampling from the original sampling rate. All Recordings were subsequently downsampled to 16kHz, and stored as 16 bit wav files. Sampling conversion was performed by a polyphase implementation in MATLAB using the "resample" function. The sentences are shown in Table 5.6.

File Name	Gender	Sentence Text
bigtips.wav	Male	Good service should be rewarded by big tips.
vega.wav	Female	I am sitting in the morning at the diner on the corner. I am waiting at the counter for the man to pour the coffee. And he fills it only halfway and before I even argue. He is looking out the window at somebody coming in.
draw.wav	Female	Draw every outer line first, then fill in the interior

Table 5.6: List of sentences used from SpEAR

Original Speech Files:

Playbacks and recordings of the speech signals were done at 48KHz and all recordings were subsequently downsampled to 16kHz, and stored as 16 bit wav files.

Original Noise Files:

F-16 noise: It is acquired by recording samples from 1/2" B&K condenser microphone onto Digital Audio Tape (DAT). The noise was recorded at the co-pilot's seat in a two-seat F-16, traveling at a speed of 500 knots, and an altitude of 300-600 feet. The sound level during the recording process was 103 dB. It was found that the flight condition had only a minor effect on the noise. The reproduced noise can therefore be considered to be representative. Sampling rate is 19.98 Khz. and Analog to Digital Conversion(ADC) rate is 16 bits[6].

Factory Noise: It is acquired by recording samples from 1/2" B&K condenser microphone onto DAT. This noise was recorded in a car production hall. Sampling rate is 19.98 Khz. and ADC rate is 16 bits[6].

Pink Noise: Pink noise is a representation of coloured noise, which has a predominantly low frequency spectrum[89]**.** It is acquired by sampling a high-quality analogue noise generator. It exhibits equal energy per 1/3 octave. Sampling rate is 19.98 Khz. and ADC rate is 16 bits[6].

Volvo 340 Noise: It is acquired by recording samples from 1/2" B&K condenser microphone onto DAT. This recording was made at 120 km/h, in 4th gear, on an asphalt road, in rainy conditions. Sampling rate is 19.98 Khz. and ADC rate is 16 bits[6].

White Noise: White noise is defined as a random signal process and its frequency spectrum is flat which means that it has equal power in all frequencies[89-96]. It is acquired by sampling a high-quality analogue noise generator and exhibits equal energy in all frequencies. Sampling rate is 19.98 Khz. and ADC rate is 16 bits[6].

Bursting Noise: It is computer generated noise using a white Gaussian random number generator.

The objective and subjective results of the wavelet based speech enhancement algorithm on bigtips.wav, draw.wav and vega.wav with different SNR value noise corruptions are shown in Table 5.7. The Daub10 wavelet type, level 8 and soft thresholding are used in all experiments at this stage. 8 listeners are used to get MOS values.

Speech File Name	Noise Type	SNR Input (dB)	SNR output (dB)	Level	**MOS Before Denoising**	**MOS After Denoising**
bigtips.wav	burst	1	2.3	8	**2.125**	**3.375**
bigtips.wav	f16	1	3.8	8	**2.250**	**3.500**
bigtips.wav	factory	1	3.2	8	**2.125**	**3.250**
bigtips.wav	pink	1	4	8	**2.500**	**4.000**
bigtips.wav	**white**	**1**	**6.6**	**8**	**1.750**	**4.250**
bigtips.wav	volvo	1	5.9	8	**2.500**	**3.750**
draw.wav	burst	1	2.1	8	**2.250**	**3.500**
draw.wav	f16	1	3.3	8	**2.000**	**3.375**
draw.wav	factory	1	3.3	8	**2.000**	**3.500**
draw.wav	pink	1	4.5	8	**2.250**	**4.250**
draw.wav	**white**	**1**	**5.4**	**8**	**1.500**	**4.250**
draw.wav	volvo	1	5.2	8	**2.250**	**3.500**
vega.wav	pink	1	4.6	8	**2.250**	**3.750**
vega.wav	pink	3	6.1	8	**2.750**	**4.000**
vega.wav	pink	5	7.6	8	**3.000**	**4.250**

vega.wav	pink	7	8.7	8	**3.000**	**4.250**
vega.wav	pink	11	11.9	8	**3.250**	**4.500**
vega.wav	pink	15	15.6	8	**3.750**	**4.500**

Table 5.7: The results of the Wavelet based speech enhancement algorithm for different type of noise signals

The results shows that Wavelet based speech enhancement algorithm improves the SNR and MOS values at the same time. The algorithms works especially well for a speech corrupted with white Gaussian noise. While the speech corrupted with "burst" noise is enhanced 1.3dB, the speech corrupted with "white Gaussian" noise is enhanced 5.6dB when the input SNR is 1dB. The reason is that while the white Gaussian noise is spreaded over all frequency bands, the other types of noises are concentrated on specific frequencies. The Wavelet based speech enhancement algorithm is able to decrease the noise in speech for different frequencies without decreasing the speech quality.

5.3 Summary

In this chapter, the Wavelet based speech enhancement algorithm is implemented using Matlab at the beginning. One audio file is corrupted with five different SNR value white Gaussian noise. Different wavelet families, levels and methods are used to denoise the sample audio files. The output SNR values and operation times are recorded using tables and graphs which gives the reader a comprehensive result set about the wavelet packet transform. The best SNR output value is obtained using Daubechies 10 wavelet type, soft thresholding method and level 8. It is found out that the SNR output values are not so different in different wavelet families, thresholding method and level. The main difference is the MOS and the operation time. The increase in the level gives much better results in MOS but worse in operation time. These results lead us to the requirement of a friendly user interface to obtain the best denoising result using flexible wavelet type, thresholding method, level and threshold value selection. The same algorithm is implemented in C with better memory usage and flexible settings. A new memory with the size of the audio file was allocated in Matlab which was increasing the memory usage by the size of level number N. The memory usage is limited to the size of the audio file with the developed algorithm which decreased the memory usage.

Similar results are obtained in C implementation with Matlab as expected. The wavelet packet transform is implemented instead of pyramid wavelet transform. The SNR output results are improved using the wavelet packet transform method when it is compared with the results in Table 5.8[92]. The reason is that the details as well as the approximations can be split in wavelet packet transform which keeps the useful information in high frequencies. Moreover, flexible threshold value adjustment gives the user fine adjustment of the threshold value. By increasing or decreasing the threshold value around the calculated value, the user gets better SNR and MOS values.

SNR input	0	2	4	6	8	10	12
Hard	3.48	2.78	2.65	2.23	1.35	1.38	1.08
Soft	6.53	5.76	5.04	4.31	3.87	3.07	2.36
Firm	6.52	5.76	5.13	4.49	4.02	3.33	2.75
Garrote	6.13	5.3	4.89	4.32	3.63	3.22	2.8
Step-garrote	5.93	5.28	4.93	4.41	3.86	3.41	2.94

Table 5.8: Performance evaluation for different thresholding algorithms in different input SNRs and their SNR improvements

Different wavelet families, thresholding methods and levels are rated for speech quality measurement. Although the SNR values are similar for different wavelet levels, the speech quality gets better with the increasing wavelet level. The best results obtained using Daubecies 10 family, level 8 and hard thresholding is used for denoising different types of noises. The results showed that the audio corrupted with noise spread over all frequencies such as white and pink noise could be denoised more effectively using wavelet transform than with noise accumulated in some frequencies such as burst noise.

In the following chapter, the real time wavelet denoising is implemented using a custom build audio equipment called Xad-ML100. Also wavelet denoising algorithm for real time denoising is impemented with a notch filter for an improved speech enhancement under fixed frequency and white Gaussian noise. At the end, the denoising algorithm results are shown in tables and graphs.

6 REAL TIME AUDIO WAVELET DENOISING

Real time audio denoising allows users to process, edit and listen to a modified audio signal on the fly. Although it requires more CPU power, however, it significantly improves professional digital audio. Real time denoising also helps editing signals where the original content is not modified at the time of editing and can reduce the required disk space for the filtered sections.

Nowadays, real time digital audio denoising on personal computers is becoming more and more common, which makes it easier for any potential end user to experiment digital audio systems without prior knowledge. Audio denoising in real time lets modified audio to be evaluated by listeners while being enhanced[93].

Common denoising methods that are widely used in real time audio operates mainly in the time domain. They consist of a gate that controls the volume of an audio signal, and a filter that enhances the speech signal only in time domain. These systems are very useful for nondemanding applications. In many other applications, an increase of the effectiveness , by transforming a signal to the frequency domain, denoising it and back to the time domain in real time are required. Wavelet denoising algorithm can be computationally efficient and fast enough to achieve this goal.

This chapter outlines the implementation of the proposed wavelet denoising algorithm in real time. The custom Xad-ML100 product is used as a real time platform.

In section 6.1, the custom hardware platform, XAD-ML100, is introduced and explained. Section 6.2 describes the implementation of the real time Wavelet denoising algorithm. In section 6.3, boundary problems of the real time wavelet denoising algorithm are analysed and solutions are suggested. Section 6.4 shows the effectiveness of the method when applied to audio data corrupted with different levels of white noise. In section 6.5, results of audio signals corrupted with white noise and fixed frequency noise using only wavelet based speech enhancement are provided. Section 6.5 describes and demonstrates the benefits of combining notch filtering with discrete wavelet transform to denoise audio signals corrupted with a combination of white noise and fixed frequency noise.

6.1 XAD-ML100 System Overview

The main goal of this unit is to acquire 16 analogue audio signals from custom build audio equipment and store and play them in custom build computers as shown, respectively in Figure 6.1 and 6.2. The audio signals are then sent to a central location over the IP network.

It is mainly used for recording conversations, where the conversations are recorded in a secure environment. The main goal is to use where recorded audio files as evidence when required such as in a court of the law.

The features of XAD-ML100 include:

• 16 embedded audio receivers (2 channels from each receiver board) including low pass filters with a cut-off frequency of 8Khz.

• 48V DC and 220V AC power options.

• Mountable rack for outdoor cabinet installation.

• Based on a scalable architecture to interconnect multiple units.

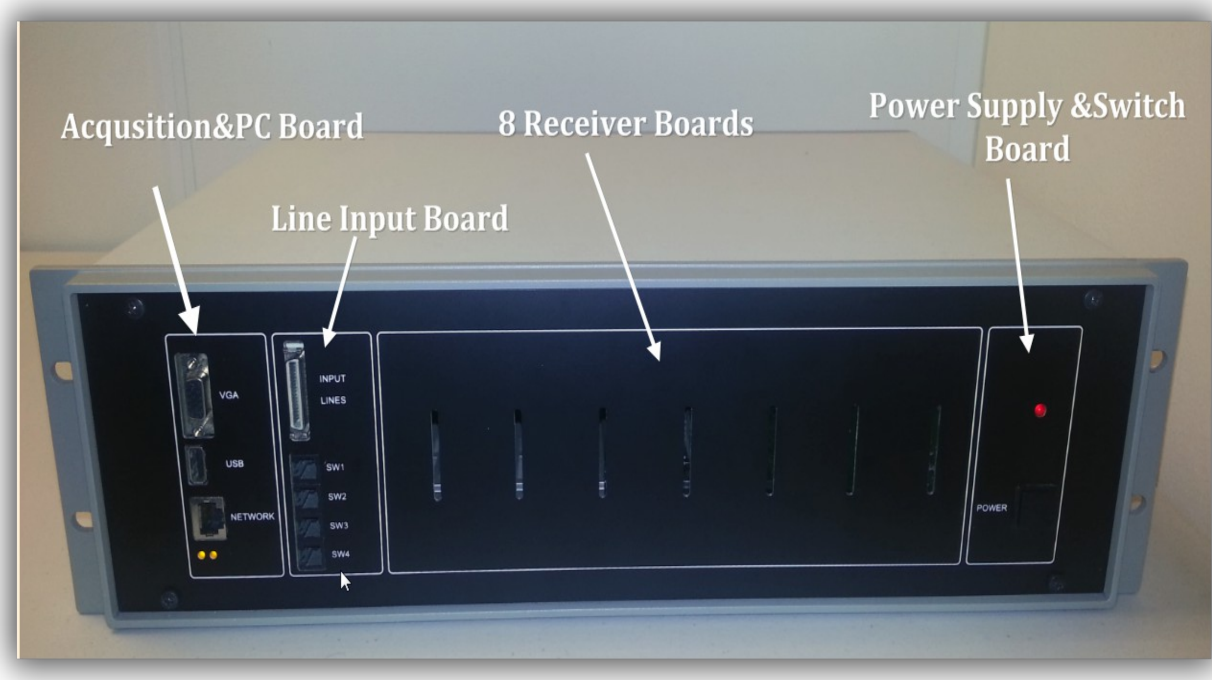

Figure 6.1: Outside View of the Xad-ML100

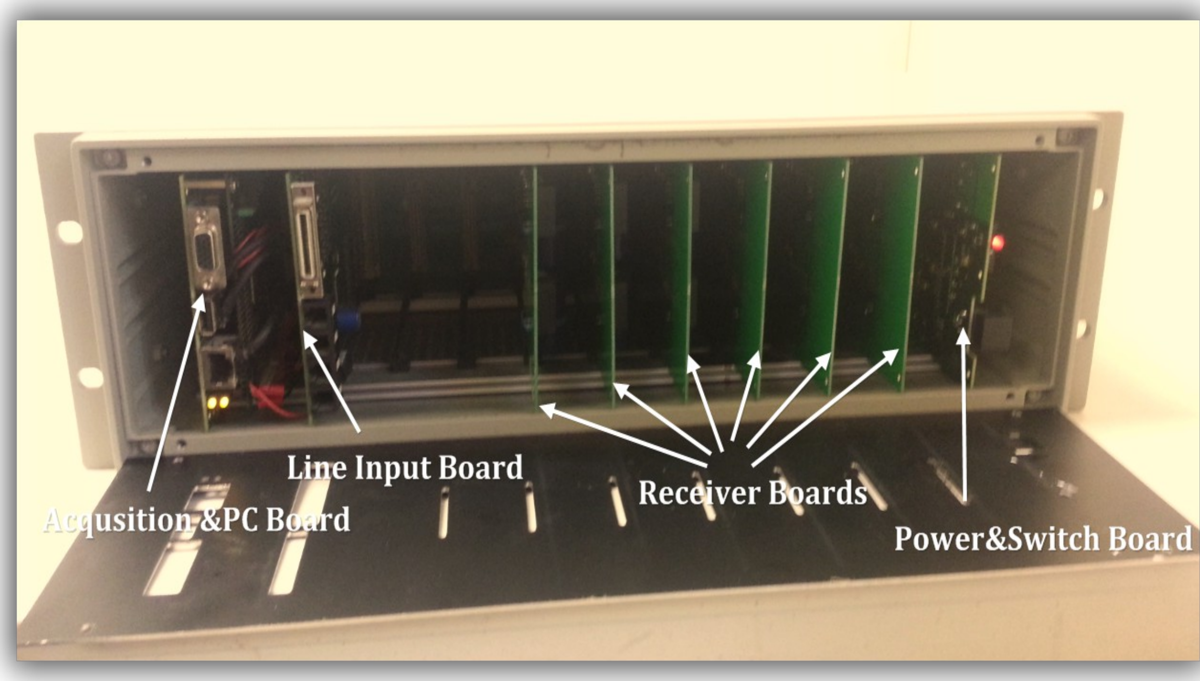

Figure 6.2: Inside View of the Xad-ML100

The block diagram of the Xad-ML100 is shown in Figure 6.3. As it is clearly illustrated, the unit consist of 12 sub units or boards, namely:

1. **Power Supply & Switch Board**:

   Power supply & switch board provides the required input voltages to all other boards. It accepts 90-265 VAC main voltage to produce 48VDC. It also accepts 48VDC as it can be used in telecommunication environment where only 48VDC is available. It produces -12VDC and +12VDC for receiver boards(sub-units), as well as, a 12VDC for the PC board. It also has a microcontroller which communicates with the PC board using RS232 protocol for the switching on and off the whole unit.

2. **Receiver Boards:** There are 8 receiver boards which can process 2 channels. Two analogue lines are connected to the receiver boards. They are pre-amplified to get a better speech quality and amplified with a ratio of 22. The amplified signals are sent to the acqusition&computer board.

3. **Line Input Board:** Line input board is designed to get 16 analogue audio signals with a custom built connector. It connects 16 audio signals to the receiver boards.

4. **Back Plane Board:** All the board are connected to each other using the back plane board. It provides an easy access to all of the boards for maintenance using an easy connection platform.

5. **Acquisition& Computer Board:** The acquisition part of the board gets 16 analogue audio signals and converts them to digital signals before sending them, using a USB protocol, to a mini PC, where all the digital speech data are processed. The processed data is available to users with a UDP connection via a secure protocol. The PC is shown in Figure 6.3 and the specifications of the PC are:

    - IntelR Atom™ E3800 processors (COMe-cBTi6)
    - COM ExpressR modules are very compact, highly integrated computers.
    - A memory unit of up to 2x 8GB
    - 1x USB 3.0 and 8x USB 2.0 is available.

Figure 6.3: XAD-ML100 Mini PC

In summary, the analogue speech signals come to the Line Input Board using custom build 16 pair connectors. The microphones are connected to those lines and can be switched on and off from a remote computer using network connection to the computer inside the ML100. Each receiver board can handle 2 analogue signals. At this stage, the analogue signal is pre-amplified, filtered and then amplified. Finally, the amplified analogue signal is sent to the Acquisition & Computer Board as shown in Figure 6.4.

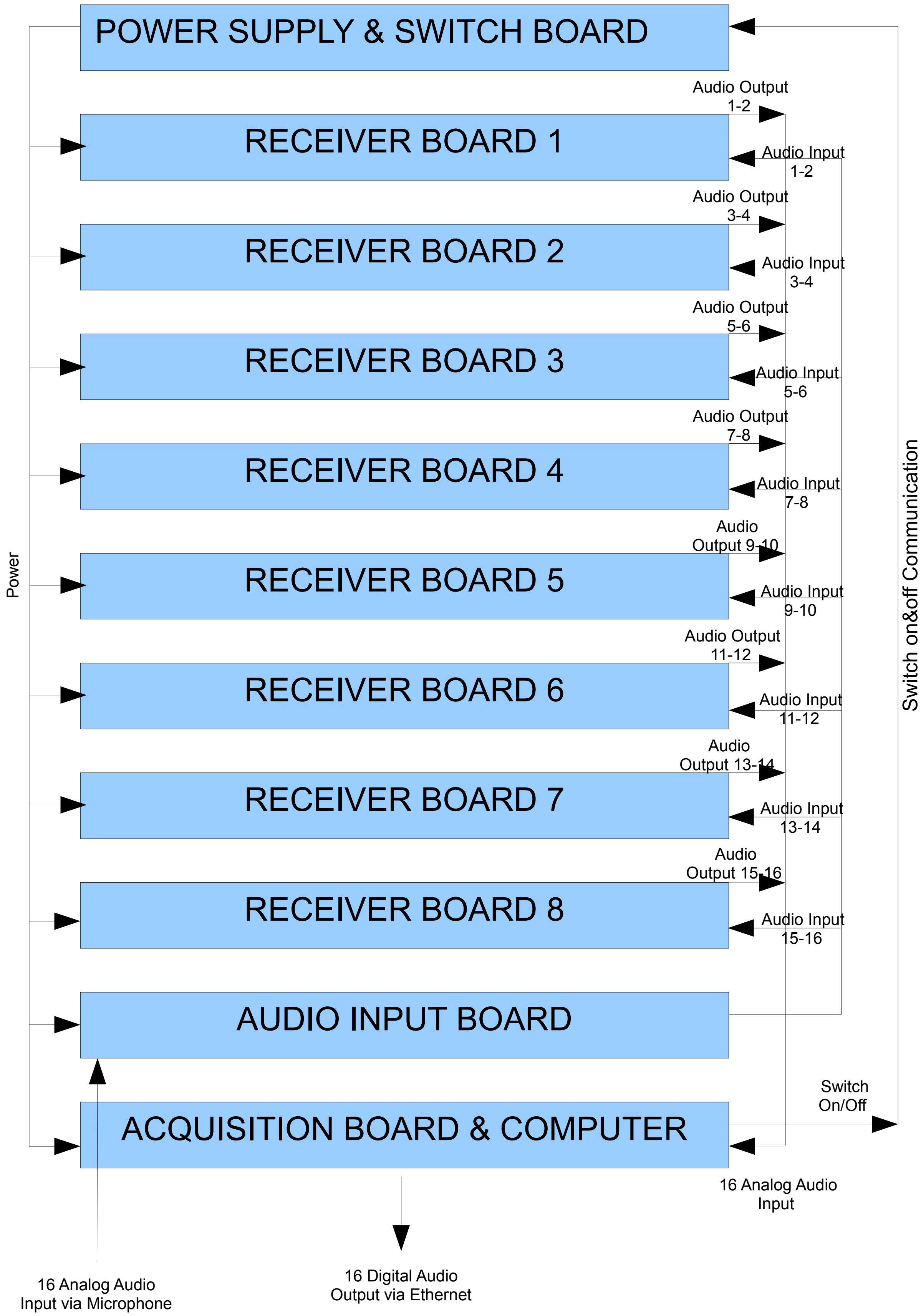

Figure 6.4: XAD-ML100 General Description

The acquisition hardware provides a sampling rate of 16000 samples per second with a 16-bit resolution. A 16 Khz sampling rate, 8-bit resolution is used in the Xad Client Software side. The block diagram of the Acquisition & Computer Board is shown in Figure 6.5.

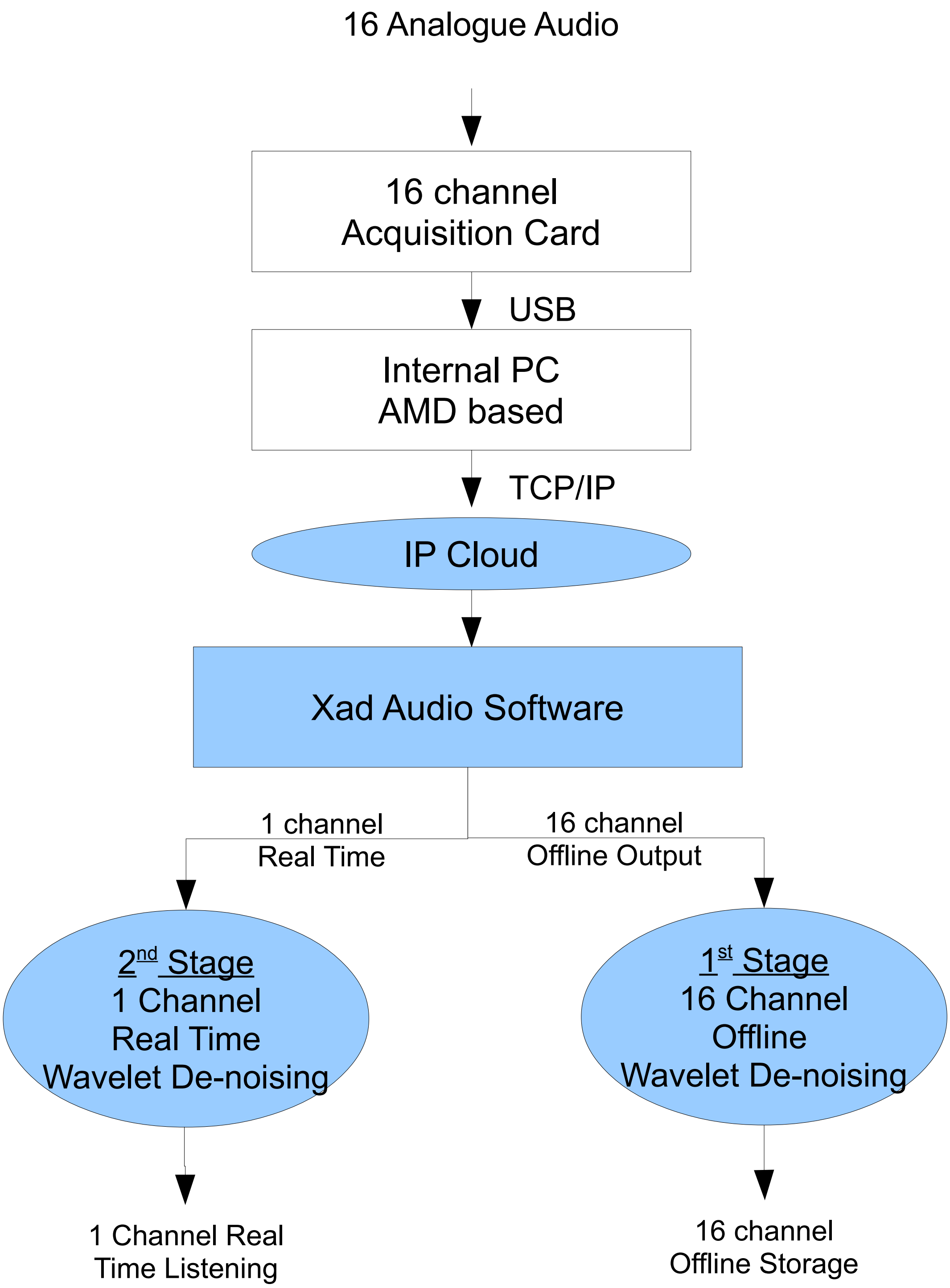

Figure 6.5: Wavelet Denoising using Xad ML100

The 16 analogue audio input signals are converted to digital signals using a 16 channel acquisition card which consists of an ADC and a field programmable gate array (FPGA) which provides a USB interface between the acquisition part and the PC. The converted signals are then sent to an AMD based computer, where all the input speech signals are recorded and can be played.

The audio channel data which is stored in the XAD-ML100 can be acquired by remote users using a TCP/IP protocol. The XAD Audio Software, XaudiNet Client, can be installed on any remote user PC, which can access the 16 recorded offline channels of the XAD-ML100. Also, remote users can access to a single real time audio channel using XaudiNet Client as shown in Figure 6.6.

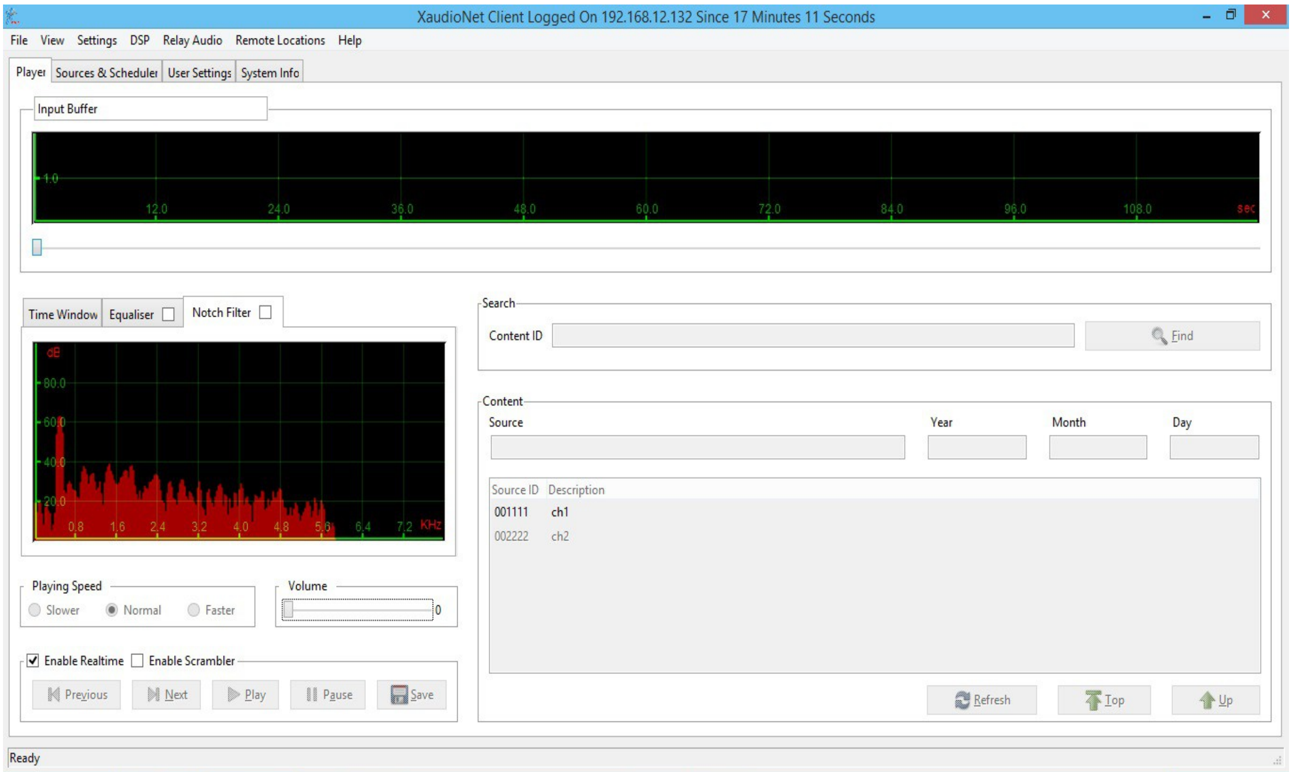

Figure 6.6: Xad Communications Audio Software User Interface

6.2 General Algorithm

Real time audio can be enabled using "Enable Realtime" button in Xaudio Client User Interface as shown in Figure 6.6. The discrete wavelet transform noise reduction algorithm is activated by enabling the "Wavelet Filtering" button as shown in Figure 6.7. The software can display the real time audio signal in the time domain or frequency domain as shown in both Figure 6.6 and Figure 6.7.

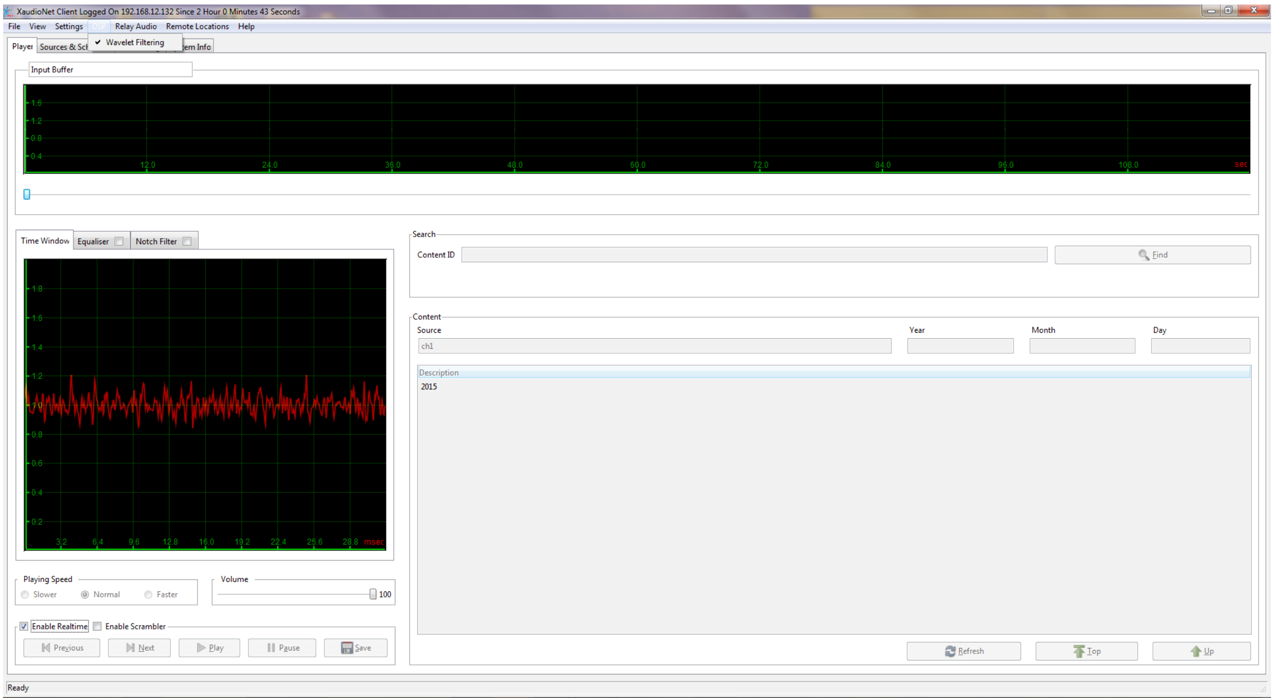

Figure 6.7: Enabling Real Time Wavelet Noise Filtering

The wavelet filtering algorithm is implemented in C as an add-on feature to Xaudionet Client and it can be enabled using Xaudionet Client when the wavelet filtering is requested. UDP based asyncronous, non-blocking socket is opened in between the wavelet filtering and Xaudionet programme as shown in Figure 6.8. In our case, the wavelet filtering program and Xaudionet Client main program are running in the same computer, however, the wavelet filtering program can run on an independent computer with some extra changes. In other words, both programs are running in the same computer in our case but they can run on independent computers as well.

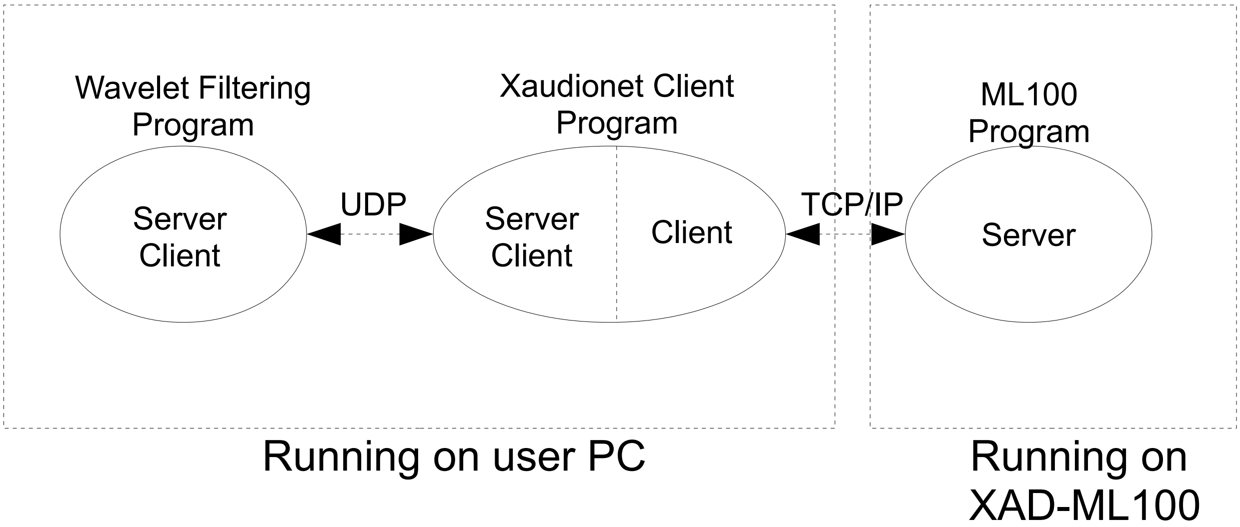

Figure 6.8: Real Time Denoising Program Description

The wavelet filtering program with its interface to the Xaudionet Client program and the XAD-ML100 program is shown in the flowchart of Figure 6.9. First of all, an asyncronous non-blocking socket communication is started between the Xaudionet Client program, running on an user PC, and the XAD-ML100 program, running on an XAD-ML100. All the audio channels are accessible on the user PC through this protocol. The UDP packages are then received from the Xaudionet Client. The audio data is then written into a FIFO and is available to the wavelet denoising program with the required data size. The wavelet denoising program starts when 2048 byte of data is received. The size of the buffer will ensure a minimum real time delay. It is worth noting that a 10 levels of decomposition max for the wavelet denoising algorithm can be achieved with the 2048 FIFO data. The wavelet denoising algorithm is the same as the offline wavelet filtering which has been explained in chapter 5. There are, however, small differences between the offline and the real time wavelet algorithm, which will be explained later in this chapter. The enhanced audio signal is then sent back to Xaudionet Client and displayed on the graphical user interface, both in time and frequency domains, for the user to observe the improvements.

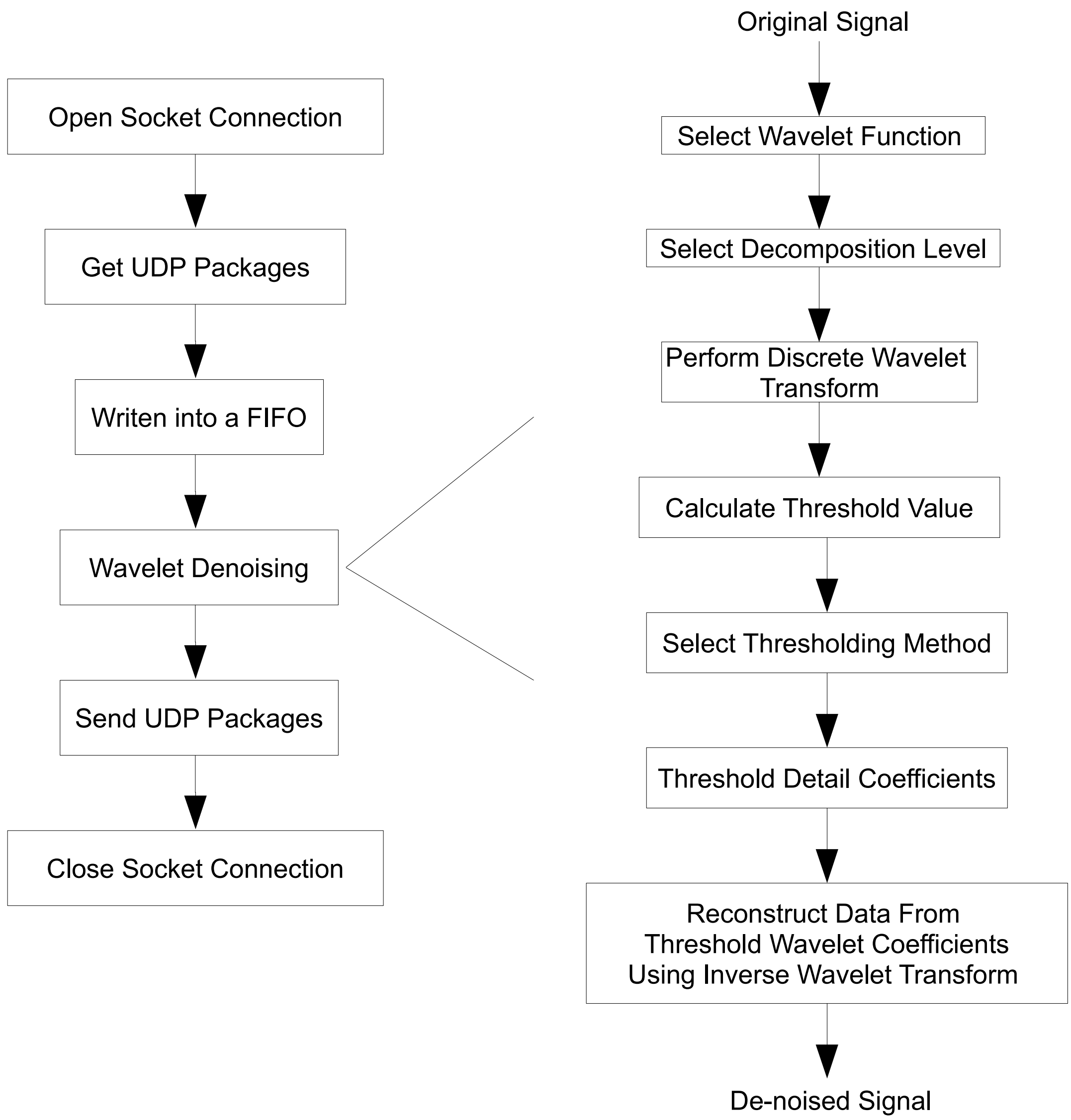

Figure 6.9: Real Time Denoising Algorithm

6.3 Boundary Analysis

In real time wavelet analysis, the convolution needs N samples which is the size of the filter for the multiplication with the input samples. However, N following samples are not available for the first and the last N multiplications since there is not enough input samples. There are many different schemes available to solve this problem by expanding the input data[94]. The most used extension schemes are presented in what follows.

Circular convolution: The input signal is extended with the required size of the first samples. The main disadvantage of this approach is that the time information of the wavelet coefficients is not accurate for the boundaries. Thresholding a coefficient at the edge has an impact on the other edge when it is reconstructed. It is not an acceptable behaviour for real time audio processing.

Zero padding: The input signal is extended with N-2 zeros at end and beginning which causes discontinuities at the borders. There are wavelet coefficients added which makes the 0^{th} level detail size (M+N-1)/2 for an input signal of length M samples. On reconstruction, the padded values are discarded. The main disadvantage of zero-padding is that artificial discontinuities are created at the border.

Symmetric extension: The input signal is extended symmetrically at the borders. The signal is mirrored at the boundaries to achieve this. It does not generate discontinuities at the borders and produces relatively low error. Like with zero padding, there are wavelet coefficients added.

The extension methods only have an impact on the wavelet coefficients at the boundaries not on the inner wavelet coefficients. Also, the extension has an impact on the decomposition process not on reconstruction. This is because the added paddings are eventually discarded during reconstruction.

In the implementation, all the methods described above are compared as a solution to the boundary problem. Experimental results gives better results for audio signals when symmetric extension method is implemented[95]. Whereas, zero padding and circular extension methods creates errors on the boundaries. Also, it is worth mentioning here that symmetric extension method gives better subjective results to listeners.

6.4 Audio Data Corrupted with White Noise

To assess the effectiveness of the implementation, signals are corrupted with white noise. In this context, a clean speech sp01.wav is corrupted with white Gaussian noise at the following SNR levels in dB: 0, 5, 10, 15. The recording is achieved using an external microphone. Secondly, the same audio file is corrupted using a fixed frequency noise at the following SNR levels in dB: 0, 5, 10, 15 as well as a white Gaussian noise. The results are evaluated using objective and subjective measures.

6.4.1 Objective Measure

The objective results of the experiments are shown in Table 6.1. The SNR output values and the operation times are measured to compare the effect of the wavelet type, the thresholding method and the wavelet level on the wavelet algorithm.

Speech File Name	SNR Input (dB)	SNR output (dB)	Wavelet Family	Thresholding Method	Level	Operation Time (msec)
Sp01.wav	0	5.5	Daub4	Soft	8	1.8
Sp01.wav	0	4.9	Daub4	Hard	8	2.1
Sp01.wav	0	5.9	Daub6	Soft	8	2.4
Sp01.wav	0	5.8	Daub6	Hard	8	2.7
Sp01.wav	**0**	**6.5**	**Daub10**	**Soft**	**8**	**3.6**
Sp01.wav	**0**	**6**	**Daub10**	**Soft**	**4**	**1.9**
Sp01.wav	0	6	Daub10	Hard	8	3.7
Sp01.wav	0	5.4	Coif1	Soft	8	3.1
Sp01.wav	0	5	Coif1	Hard	8	3
Sp01.wav	0	5.6	Coif3	Soft	8	3.5

Sp01.wav	0	5.5	Coif3	Hard	8	3.4
Sp01.wav	0	6.3	Coif5	Soft	8	4.1
Sp01.wav	0	6.1	Coif5	Soft	4	2.7
Sp01.wav	0	5.7	Coif5	Hard	8	4
Sp01.wav	0	5.4	Sym6	Soft	8	3.3
Sp01.wav	0	5.3	Sym6	Hard	8	3.4
Sp01.wav	0	5.6	Sym10	Soft	8	4.5
Sp01.wav	0	5.4	Sym10	Hard	8	4.7
Sp01.wav	0	6.1	Sym14	Soft	8	5.8
Sp01.wav	0	5.8	Sym14	Soft	4	3.8
Sp01.wav	0	5.8	Sym14	Hard	8	5.9

Table 6.1: Results of the real time wavelet based speech enhancement algorithm

The SNR output results are consistent with the offline wavelet de-noising results except for the operation times as shown in Figure 6.10. The operation times are much better since the 2048 byte data chunks are used in each denoising step.

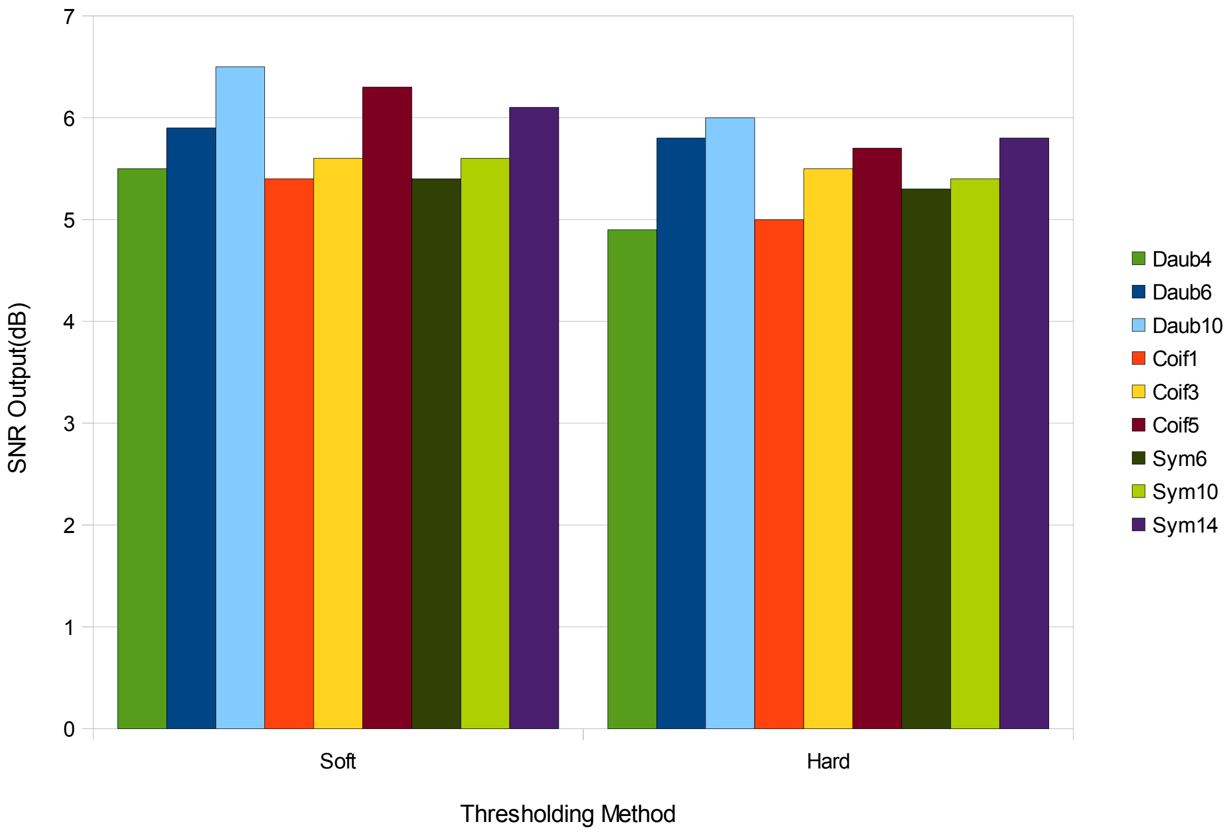

Figure 6.10: SNR results of real time Wavelet speech enhancement algorithm

The SNR output results of the real time wavelet speech enhancement algorithm are shown in Figure 6.10. SNR results do not change too much with different wavelet families and wavelet thresholding methods.

Figure 6.11 shows the operation times of the real time wavelet speech algorithm. While thresholding methods do not change the operation times, wavelet levels and wavelet families affect the operation times. This is because the coefficient numbers increase with the level of decomposition.

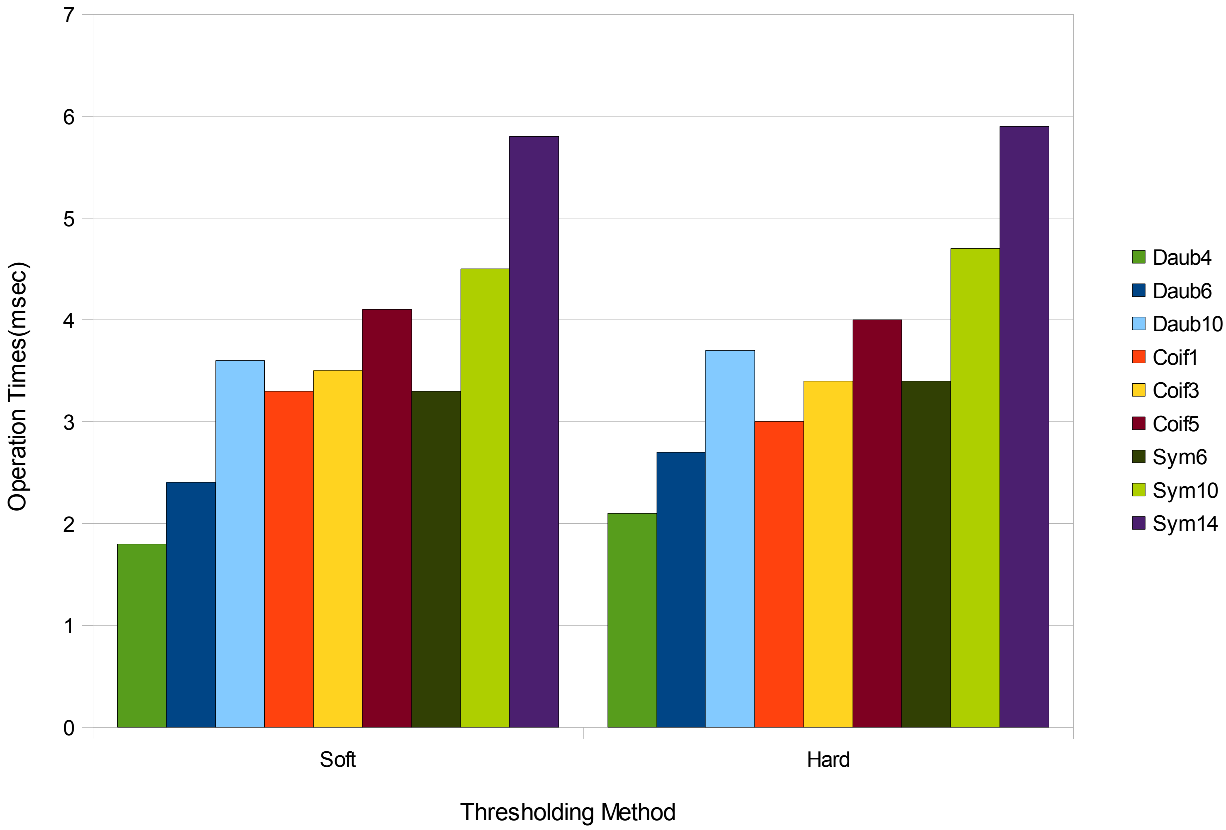

Figure 6.11: Operation times of real time Wavelet speech enhancement algorithm

The best SNR output results were obtained with the Daub10 with no much differences in comparison to other wavelet families. The main difference has appeared on the operation times as seen in Table 6.1. The operational time of the wavelet based speech enhancement algorithm was the shortest with the Daub10. The experiments shows that the coefficient number affects the operational time as expected. Since the Daub10 has the least number of coefficients, the algorithm using it finishes quicker than the others. Also, the operational time increases in parallel with the wavelet level number as seen in Table 6.1.

6.4.2 Subjective Measure

The Mean Opinion Score (MOS) is calculated as defined in section 5.1.2.2 for each audio file after collecting individual scores. The test is organised under regulated conditions using a quiet environment.

Table 6.2 shows the MOS scores from an actual ACR test for 8 listeners with no previous familiarity with test materials.

For this experiment, sp01.wav file is corrupted with 0dB white Gaussian noise for the tests. The sp01.wav clean audio file is rated 5 by listeners and 1.25 after 0dB white Gaussian noise corruption.

Speech File Name	SNR Input (dB)	SNR output (dB)	Wavelet Family	Thresholding Method	Level	**MOS**
Sp01.wav	0	5.5	Daub4	Soft	8	**3.875**
Sp01.wav	0	4.9	Daub4	Hard	8	**3.500**
Sp01.wav	0	5.9	Daub6	Soft	8	**4.000**
Sp01.wav	0	5.8	Daub6	Hard	8	**3.625**
Sp01.wav	**0**	**6.5**	**Daub10**	**Soft**	**8**	**4.125**
Sp01.wav	**0**	**6**	**Daub10**	**Soft**	**4**	**3.375**
Sp01.wav	0	6	Daub10	Hard	8	**4.000**
Sp01.wav	0	5.4	Coif1	Soft	8	**3.250**
Sp01.wav	0	5	Coif1	Hard	8	**3.125**
Sp01.wav	0	5.6	Coif3	Soft	8	**3.625**
Sp01.wav	0	5.5	Coif3	Hard	8	**3.250**
Sp01.wav	0	6.3	Coif5	Soft	8	**3.875**
Sp01.wav	0	6.1	Coif5	Soft	4	**3.000**
Sp01.wav	0	5.7	Coif5	Hard	8	**3.500**
Sp01.wav	0	5.4	Sym6	Soft	8	**3.375**
Sp01.wav	0	5.3	Sym6	Hard	8	**3.000**

Sp01.wav	0	5.6	Sym10	Soft	8	**3.500**
Sp01.wav	0	5.4	Sym10	Hard	8	**3.125**
Sp01.wav	0	6.1	Sym14	Soft	8	**3.875**
Sp01.wav	0	5.8	Sym14	Soft	4	**3.125**
Sp01.wav	0	5.8	Sym14	Hard	8	**3.500**

Table 6.2: MOS results of the real time wavelet based speech enhancement algorithm

Different wavelet families, thresholding methods and levels are rated for speech quality measurement. Although the SNR value enhancements are similar for different wavelet families and thresholding methods, the speech quality gets better when soft thresholding is used. Also, the wavelet type choice is affecting the speech quality. The wavelet types with more coefficients such as Daub10, Coif5 and Sym14 give better speech quality.

6.5 Audio Signal Corrupted with White Noise and Fixed Frequency Noise

The same audio file, sp01.wav, is corrupted using a fixed frequency noise as well as a white Gaussian noise at a 0dB SNR value. The fixed frequency noises are applied to the signals at the same level. Since they are located at single frequencies, they do not change the total SNR values of the signals.

Different frequencies are applied to assess the impact on the real time wavelet denoising algorithm. A Daub10 wavelet type, soft thresholding and an 8 level of decomposition are selected. The selection of the Daub10 wavelet is driven by the results from the previous experiments.

6.5.1 Results Using Objective and Subjective Measure

The SNR output and MOS values are shown in Table 6.3 after combining extra fixed frequency noises with a white Gaussian noise. The fixed frequency noises are obtained using a sine wave siren sound generator. 500, 2000, 4000, 6000 and 8000 Hz are used in our experiments to show the impact on the wavelet denosing algorithm.

Speech File Name	SNR Input (dB)	Frequency (Hz)	SNR output (dB)	MOS before de-noising	MOS after de-noising
Sp01.wav	0	500	6	1.125	3.375
Sp01.wav	0	2000	6	1.250	3.250
Sp01.wav	0	4000	6.2	1.000	3.000
Sp01.wav	0	6000	6.2	1.250	3.000
Sp01.wav	0	8000	6.3	1.125	2.750

Table 6.3: Results of the real time wavelet based speech enhancement algorithm when combining white Gaussian noise with fixed frequency noises

It is clear from Table 6.3 that the SNR for the combined noises, fixed frequency noise and white Gaussian noise, decreases in comparison to the SNR for white Gaussian noise only. This is also true for the MOS values. These values are degraded in the case of combined noises as shown in Table 6.3.

The SNR output values are increasing with the increase in frequency. Since the thresholding value is calculated automatically using filtered coefficients, it is then increasing with the increase of the frequency. The calculated threshold value increase suppresses more coefficients which causes an increase in the SNR output value. However, the MOS outputs after de-noising does not give better results since the thresholding value is miscalculated because of the fixed frequency signal. Moreover, the high frequency content increases the thresholding value, which causes more corruption in the audio signal quality as seen in Table 6.3.

To overcome the problem of fixed frequency noise, a combination of two filtering methods is adopted.

6.6 Notch Filtering with Discrete Wavelet Transform

In the previous section, experimental results have shown that the wavelet transform cannot handle the combination of a white Gaussian noise and a fixed frequency noise. To remedy to this problem, a modified algorithm is suggested. It consists of combining a wavelet filter with a more traditional Notch filter.

6.6.1 Notch Filter

A notch filter decreases the gain of a narrow band frequency spectrum while leaving the rest of the audio signal unaffected. The filter is controlled by three parameters. The first parameter defines the cut-off frequency of the filter, which is in general the centre frequency of the filter. The second parameter of the filter consists of the steepness of the roll-off of the filter. Finally, the third parameter defines the gain, which represent the amount of gain reduction within the frequency band. The shape of the filter and the frequencies are defined by these parameters.

A notch filter can be classified by the length of its impulse response as a finite impulse response(FIR) and infinite impulse response (IIR). While the FIR notch filter is always stable and provides a linear phase response, the IIR notch filter is potentially unstable and do not provide a linear phase response. The advantage of IIR filter structures over FIR filter structures is that they can be designed with a much lower order for fulfilling equivalent magnitude specifications[96]. A digital FIR notch filter uses a large of coefficients to greet the same requirement of the magnitude response. As the number of coefficients increases the delay, which is the main concern in real time audio de-noising, an IIR filter is preferred in real time applications. IIR filters are also called recursive filters because their impulse responses are composed of decaying exponentials. In this thesis, a second order recursive notch filter is used because of its effectiveness and practicality[97].

6.6.1.1 Recursive Notch Filter

Recursive filters are an effective way of obtaining a long impulse response without having to perform a long convolution. Although they execute very quickly, they have less performance and flexibility than other digital filters.

A recursive filter is described by a difference equation of the form:

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + \ldots\ldots + b_1 y[n-1] + b_2 y[n-2] + \ldots\ldots\ldots \quad (6.1)$$

In this equation, $x[\ ]$ is the input signal, $y[\ ]$ is the output signal, and the a's and b's are the coefficients. The transfer function of the filter is then:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ldots}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3} - \ldots.} \quad (6.2)$$

The z-transform can be used for switching between the recursive coefficients and the frequency response, merging cascaded and parallel stages into a single filter and transforming an analogue filter to its digital equivalent using the bilinear transformation, which is used to obtain the recursion coefficients in our algorithm.

A second order analogue notch filter Laplace transfer function:

$$H_a(s) = \frac{s^2 + \Omega_0^{\,2}}{s^2 + Bs + \Omega_0^{\,2}} \quad (6.3)$$

where Ω_0 is the notch frequency, B is 3-dB notch bandwidth and s is the Laplace variable. Using bilinear transform, the imaginary plane axis in the s plane can be mapped onto the unit circle of the z plane using equation 6.3 as:

$$\begin{aligned}
H(z) &= H_a(s)\big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} \\
H(z) &= \frac{(1+\Omega_0{}^2)-2(1-\Omega_0{}^2)z^{-1}+(1+\Omega_0{}^2)z^{-2}}{(1+\Omega_0{}^2+B)-2(1-\Omega_0{}^2)z^{-1}+(1+\Omega_0{}^2-B)z^{-2}} \\
H(z) &= \frac{1+\alpha}{2}\frac{1-2\beta z^{-1}+z^{-2}}{1-2\beta(1+\alpha)z^{-1}+\alpha z^{-2}}
\end{aligned} \tag{6.4}$$

where

$$\begin{aligned}
\alpha &= \frac{1+\Omega_0{}^2-B}{1+\Omega_0{}^2+B} = \frac{1-\tan(B_w/2)}{1+\tan(B_w/2)} \\
\beta &= \frac{1-\Omega_0{}^2}{1+\Omega_0{}^2} = \cos w_0
\end{aligned} \tag{6.5}$$

and where

$$\begin{aligned}
B_w &= 2\Pi(\frac{B}{f_s}) \\
w_0 &= 2\Pi(\frac{f_0}{f_s})
\end{aligned} \tag{6.6}$$

Two parameters must be selected before using equation 6.5: the notch frequency f_0, and the sampling frequency f_s. From these specified values, the intermediate values B_w and w_0 are calculated. Then, the filter coefficients are evaluated using the values α and β as in equation 6.2. The coefficient values are used to calculate the output signal values.

The designed digital notch filter rejects a specific annoying frequency and keep other broadband signals unchanged. In the proposed algorithm, five different frequencies can be attenuated using multiple notch filters. The simplest way consists of cascading single notch filters.

In the proposed algorithm, the notch frequencies can be selected through the graphical user interface in real time. The default notch filter frequency is selected as 6Khz and can be modified by the user. The coefficients are calculated with the selected notch frequency and bandwidth. This approach makes notch filtering much easier and more efficient.

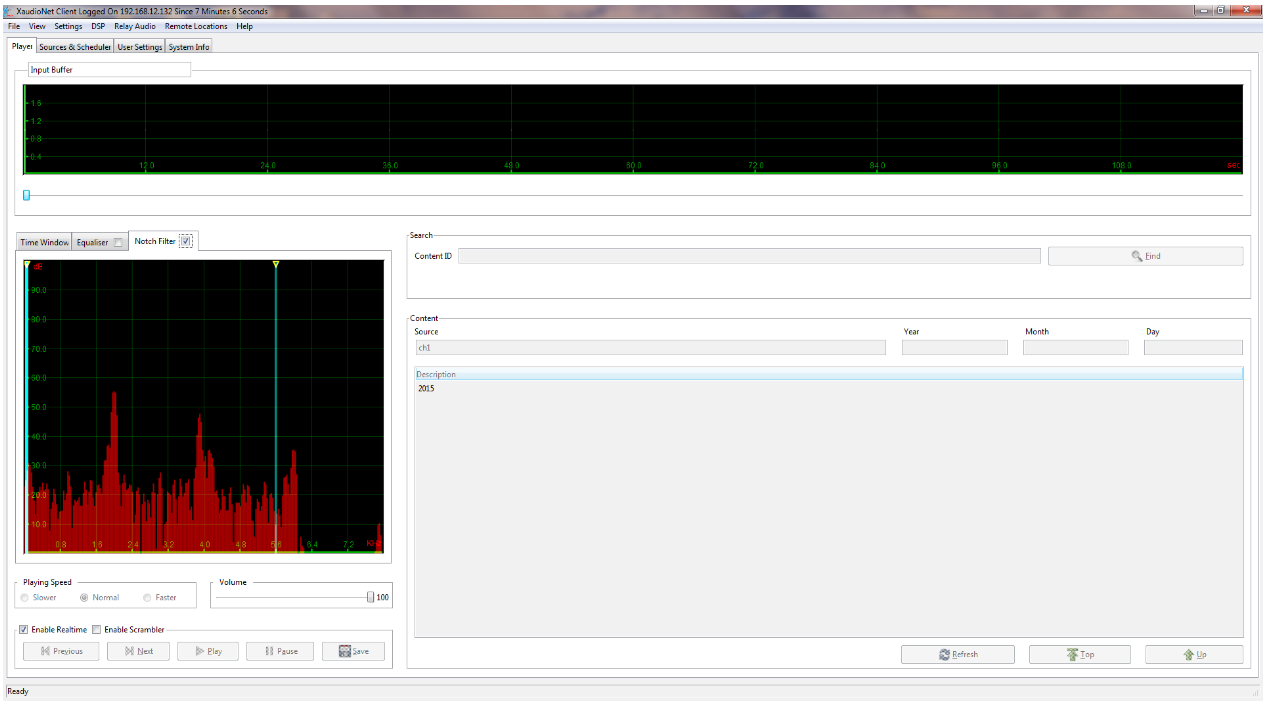

Figure 6.12: Frequency Spectrum Before Notch Filter

A second-order IIR notch filter is applied before applying the discrete wavelet transform as seen in Figure 6.12 and Figure 6.13 using the XaudioNet Client. The 6Khz noise is filtered out using a notch filter as shown in Figure 6.13.

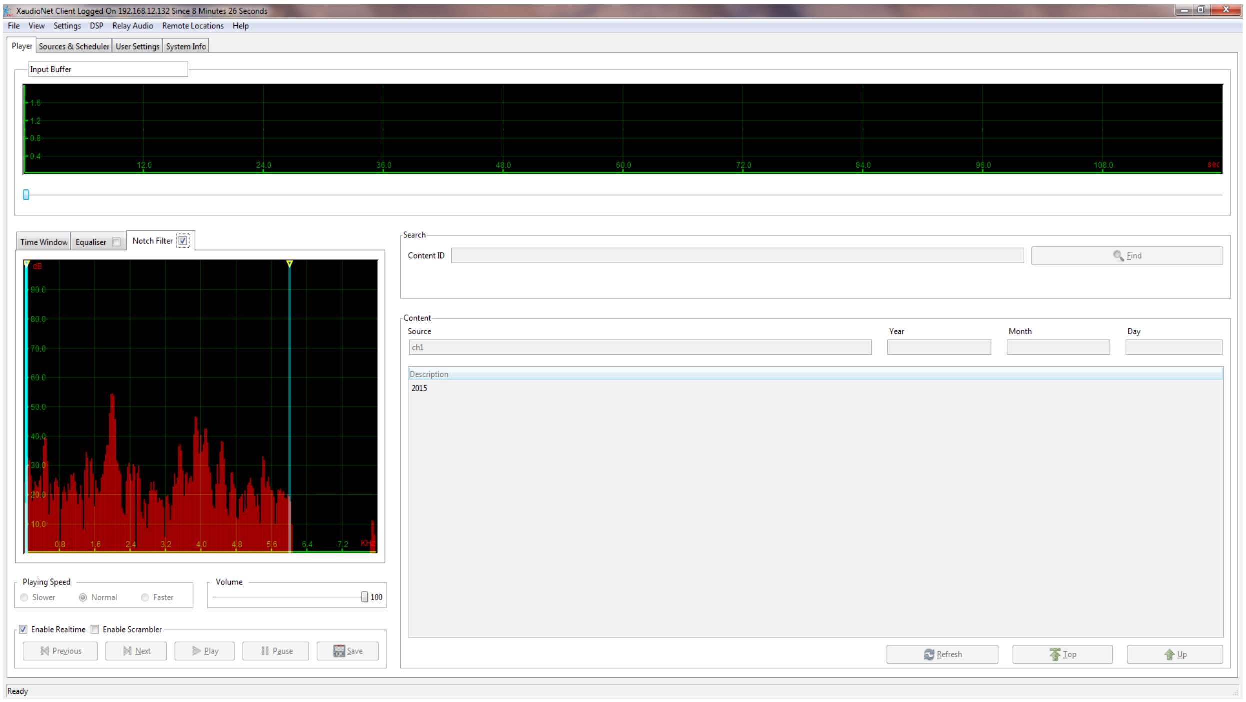

Figure 6.13: Frequency Spectrum After Notch Filtering

Noises with different centre frequencies are filtered using a combination of notch filters and a wavelet filter. The results are shown in Table 6.4.

Speech File Name	SNR Input (dB)	Frequency (Hz)	SNR output (dB)	MOS before de-noising	MOS after de-noising
Sp01.wav	0	500	6.3	1.125	4.125
Sp01.wav	0	2000	6.4	1.250	3.875
Sp01.wav	0	4000	6.3	1.000	3.250
Sp01.wav	0	6000	6.3	1.250	3.500
Sp01.wav	0	8000	6.4	1.125	3.500

Table 6.4: Results of the combination of notch filtering with the wavelet transform speech enhancement algorithm

The SNR output values are improved after combining notch filtering with the wavelet transform speech enhancement algorithm as seen in Table 6.4. The SNR output value was 6.5dB when there was no extra fixed frequency noise insertion. The SNR output values are very close to 6.5dB after the successfull combination of notch filtering with the wavelet transform. The MOS values are also improved after the combination of the two algorithms. They are very close to the MOS values of the signal without the extra fixed frequency noise.

6.7 Summary

In this chapter, the wavelet denoising algorithm is implemented in real time using the custom built Xad-ML100. The main goal of this product is to acquire 16 analogue audio signals from custom built audio equipment and store and play them in custom built computers.

Experimental results show that real time wavelet algorithm is working as effective as offline wavelet algorithm. While SNR output results do not change much with wavelet type, thresholding method and wavelet level, MOS values and operation times change rapidly with wavelet type and wavelet level. The optimisation of wavelet level, wavelet type and thresholding method is crucial for user to get better results and less delay after using real time wavelet denoising algorithm.

After successful results with added white Gaussian noise, the fixed frequency components are added to experiment wavelet denoising algorithm. New method is introduced after getting unsatisfactory experimental results under mixed fixed frequency and white Gaussian noise. Notch filter is applied before wavelet denoising algorithm to remove the fixed frequency noise as fixed frequency noise is misleading the thresholding value calculation and confuses the wavelet denoising algorithm.

Experimental results show that combination of notch filter and wavelet denoising algorithm is successful for mixed fixed frequency and white Gaussian noise in real time.

7 SUMMARY AND CONCLUSIONS

7.1 Introduction

In this chapter, the outcomes of the research work and some of the recommendations for future work in the area of enhancing noisy speech data are summarised. Important points about the research are highlighted and an outline of the achievements is presented as well. This shows how the objectives of the research programme have been met.

7.2 Summary and Conclusion

The first part of the thesis introduces some important concepts in relation with the topic of this research work. In this context, chapter 2 is dedicated to the characteristics of speech signals. It includes the theory behind speech production and perception as well as a description of a variety of auditory models. It also, importantly, highlights the noise chain in an audio system (acquisition stage and play back stage). Chapter 3 critically review some existing techniques used in speech enhancement or denoising. It includes different techniques such as comb filters, adaptive enhancement methods, spectral subtraction, and Wiener filtering. These methods are then compared and their advantages and disadvantages are presented. Chapter 4 consists of a thorough review of the theory on the wavelet transform. The definition of the transform is introduced first, then, concepts such as the continuous wavelet transform and Mallat's multiresolution analysis are presented. The concepts of filters banks and the discrete wavelet transform are introduced in the second part of the chapter. Filters banks are very important when implementing a wavelet based system. Other aspects like wavelet families (orthonormal and biorthogonal) and wavelet types (Daubechies, Coiflet and Symlet) are also presented. To complement chapter 3, some existing work on speech enhancement (denoising) using the wavelet transform is reviewed at the end of this chapter.

The second part of the thesis deals with the objective 1 which is to implement an offline speech enhancement algorithm using discrete wavelet packet transform instead of the traditional wavelet tree decomposition. The implementation uses the discrete wavelet packet algorithm for the decomposition instead of the traditional wavelet decomposition. This approach, though time consuming, will provide finer wavelet coefficients at higher levels of decomposition and consequently provides a more efficient way of calculating the threshold value. Automatic soft thresholding and hard thresholding has been implemented. Since the wavelet transform presents a range of peculiarities not present in any other transform, it was very important to select the best wavelet for the speech enhancement problem. Unlike the Fourier transform, the wavelet transform is not unique. In fact, there are several wavelets having different features and characteristics. These features include the wavelet family (orthonormal or biorthogonal), the wavelet type (Daubechies, Coiflet, etc.) the number of coefficients for both the low pass filter and the high pass filter, the level of the decomposition and the type of threshold if the speech enhancement application is included. In the context of our research work, only some orthonormal wavelets were considered. To validate this initial work, different experiments have been carried out. In these experiments, the proposed wavelet packet decomposition algorithm for speech enhancement has been implemented initially in Matlab. Six different utterances from a noisy speech corpus (NOIZEUS) library were used to evaluate the proposed wavelet packet decomposition algorithm. Clean speech signals have been corrupted with 0dB white Gaussian noise. The experiments were carried out using different wavelet types, levels of decomposition and thresholding methods.

The results have been evaluated using objective and subjective measures as introduced in objective 2, which are in this case the Signal to Noise Ratio (SNR) and the Mean Opinion Score (MOS), respectively. The wavelet thresholding algorithm performed well and produced good quality speech signals. The best SNR value was achieved by the "Daub10" wavelet with a level of decomposition of "8", using the "soft" thresholding method. The operation time increased proportionally with the level of decomposition as it was expected. However, it is worth mentioning that the output SNR values for two different levels were not too much different from each other. For example, the output SNR value after the speech enhancement algorithm was 6.8dB for the level 8 and 6.1 dB for level 4. Although the difference of levels does not improve the SNR value too much, the speech quality is drastically improved in higher levels. While the increase in the number of coefficients for the same wavelet family has not improved the SNR value remarkably, it increased the output speech quality immensely. Unfortunately, the increase of the number of coefficients for the same wavelet family gradually increases the operation time.

The second stage of the experimental work consisted of making the proposed wavelet algorithm suitable for a hardware platform as mentioned in objective 3. For this purpose, an efficient implementation that uses limited memory resources has been developed. This will make the algorithm more suitable for hardware platforms with limited resources. In addition, to ease the experimentation process, a graphical user interface (GUI) has been designed as indicated in objective 6. It allows users to implement speech enhancement through graphical icons and visualise the improvement before and after the enhancement algorithm. Moreover, a sliding bar is added to the GUI to fine-tune the thresholding value, which is automatically calculated as part of the wavelet denoising algorithm. Thresholding value fine adjustment gives user the option to determine the speech quality against SNR value as the increase of the thresholding value leads to a better SNR value; unfortunately, it will make worse the speech quality. Six different utterances with white Gaussian noise from a noisy speech corpus (NOIZEUS) library and three different speeches with six different noise types in different SNR values from SpEAR Database were used to evaluate the wavelet packet decomposition thresholding algorithm in C. The Daub10 wavelet type, level 8, using soft thresholding method is used in all experiments at this stage.

The results are evaluated using objective and subjective measures, which are the signal to noise ratio (SNR) and the Mean Opinion Score (MOS), respectively. The wavelet thresholding algorithm as implemented in C performed well and produced good quality speech signals. It gave similar results to the ones in Matlab. The algorithm is designed such that it does not use much memory. The operation time which is less than 100 milliseconds for a 2 minute long audio file satisfies the operation time requirements for offline speech enhancement algorithms. When dealing with different types of noise signals, such as, white Gaussian noise, pink noise, f16 noise, burst noise, factory noise and Volvo noise, the results have shown that the proposed wavelet thresholding algorithm performs far better in white Gaussian noise, pink noise and Volvo noise. The nature of these different noises is very irregular and non-stationary compared to real life noises, where on the other hand, pink and car interior noises present some low-pass characteristics. Applying a wavelet thresholding method for all six different noises with diverse nature is not a good idea and this is one issue we should further address in any future research work. Moreover, the experimental results showed that the wavelet thresholding algorithm in C performed well and produced good quality speech using a low memory, which makes the algorithm implementation on low memory processors more than plausible as stated in objective 3.

Finally, the implementation of the real time wavelet enhancement algorithm using the wavelet packet decomposition method is outlined. A custom build computer, Xad-ML100, is used as a real time platform as introduced in objective 4. The wavelet packet decomposition method which takes more time compared to wavelet pyramid decomposition is implemented to notice the latency in real time applications as referred in objective 5. After the experiments, our real time wavelet enhancement algorithm gives less than 10 milliseconds latency for different combinations of noise types, wavelet type, wavelet level of decomposition and thresholding method. This latency result is quite satisfactory as tolerable limits to latency for real time processing is estimated to be between 6 and 20 milliseconds[98].

After obtaining very successful results when dealing with added white Gaussian noise, fixed frequency components are added to experiment the wavelet enhancement algorithm. Different wavelet types, wavelet levels and thresholding methods are applied to enhance a clean speech, corrupted by white Gaussian noise in different SNR levels and by a fixed frequency noise. The results are compared using output SNR values and Mean Opinion Score (MOS). The results show that the wavelet packet decomposition is not producing good results for fixed frequency noise mixed with white noise. To cater for this problem, a Notch filter is applied prior to applying the wavelet enhancement algorithm to remove the fixed frequency noise as discussed in objective 7. The fixed frequency noise actually misleads the thresholding value calculation and confuses thus the wavelet enhancement algorithm. Experimental results show that the combination of Notch filtering with the wavelet enhancement algorithm is successful in dealing with fixed frequency noise when combined with white Gaussian noise in real time.

7.3 Suggestions for Future Research

With the continual growing success of the wavelet transform in the field of speech processing, different other aspects related to the transform have to be addressed. Some of these points are summarised in what follows.

7.3.1 Algorithm Experiments

The different wavelet types make different trade-offs between how compactly the basis functions are localised in space and how smooth they are. Since there are different mother wavelets belonging to different wavelet types available, the choice of the wavelet type, mother wavelet and its order greatly affect the accuracy of the analysis. There are different examples of wavelet types such as Haar, Daubechies, Biorthogonal, Symlets, Coiflets, Morlet, Mexican Hat, and Meyer etc. [101]. Daubechies, Coiflet and Symlet are used to experiment suggested algorithms in this thesis. The common feature of these wavelet types is that they are all orthogonal. Biorthogonal wavelet families need to be added for future experiments to get a relation between the features of the wavelets and the characteristics of the input speech signal.

7.3.2 Optimal Wavelet Type Selection

Different wavelet types can be used in speech enhancement. Wavelets vary in the length of support of the mother wavelet, the number of vanishing moments, the symmetry or the regularity, the existence of a corresponding scaling function etc. [100]. Since all the translations and scaling are over the mother wavelet, the selection of the mother wavelet plays a crucial role in obtaining good results in terms of SNR and MOS values in speech enhancement. Moreover, a wavelet type which works well for a particular noise type may not be good for some other noise type. In our work, GUI is designed to give user an easy selection of wavelet types. However, getting the best speech enhancement result takes time in our method since user needs to try all the wavelet types to compare the results. To overcome this problem, the best wavelet type for different types of noises needs to be obtained and suggested to the user automatically in the future.

7.3.3 Embedded Hardware Implementation

Speech enhancement deals with improving some perceptual aspect of speech that has been impacted by background noise. It is very necessary for listeners, especially when the speech is greatly corrupted by the interference on portable devices such as mobile phones. Various speech enhancement and noise suppression techniques has been proposed in the last few decades. However, many of them either have unsatisfied performance, or are too complex to implement, particularly in portable devices [101]. In our work, real time speech enhancement is implemented successfully using Xad-ML100. The suggested wavelet algorithm is implemented on a mini PC which can acquire 16 channels. However, it is not possible to use mini PCs on any portable device since they are not small in size and need large amount of power. However, the evolution of the digital devices aims to small size and low power consumption. Taking all the above into consideration, the suggested speech enhancement system needs to be implemented in an embedded system such as DSP. Our suggested algorithm is improved for low memory usage to make the low memory portable device implementation possible.

APPENDIX A

THE WAVELET FAMILY FILTER COEFFICIENTS

Analysis_..._lp[ ] shows analysis low-pass filter coefficients of the selected wavelet type,

analysis_..._hp[ ] shows analysis high-pass filter coefficients of the selected wavelet type,

synthesis_..._lp[ ] shows synthesis low-pass filter coefficients of the selected wavelet type,

synthesis_..._hp[ ] shows synthesis high-pass filter coefficients of the selected wavelet type.

analysis_Daub4_lp[8] =

{-0.010597401784997278,0.032883011666982945,0.030841381835986965,-0.18703481171888114,0.02798376941698385,0.6308807679295904,0.7148465705525415,0.23037781330885523};

analysis_Daub4_hp[8] =

{-0.23037781330885523,0.7148465705525415,-0.6308807679295904, 0.02798376941698385,0.18703481171888114,0.030841381835986965,-0.032883011666982945,-0.010597401784997278};

synthesis_Daub4_lp[8] =
{0.23037781330885523,0.7148465705525415,0.6308807679295904, 0.02798376941698385,0.18703481171888114,0.030841381835986965,0.032883011666982945,-0.010597401784997278};

synthesis_Daub4_hp[8] =

{-0.010597401784997278, 0.032883011666982945, 0.030841381835986965, 0.18703481171888114,0.02798376941698385, 0.6308807679295904,0.7148465705525415,-0.23037781330885523};

analysis_Daub6_lp[12] =

{-0.0010773010849955, 0.00477725751101065, 0.000553842200993801, 0.031582039318031156,0.0275228655300162,0.09750160558707936, -0.12976686756709563, -0.22626469396516913, 0.3152503517092432, 0.7511339080215775,0.4946238903983854,0.11154074335008017};

analysis_Daub6_hp[12] =

{-0.11154074335008017, 0.4946238903983854, -0.7511339080215775 , 0.3152503517092432, 0.22626469396516913, -0.12976686756709563, -0.09750160558707936, 0.02752286553001629, 0.031582039318031156 , 0.000553842200993801 6, -0.004777257511010651, -0.00107730108499558};

synthesis_Daub6_lp[12] =

{0.11154074335008017, 0.4946238903983854, 0.7511339080215775 , 0.3152503517092432, -0.2262646939651691, -0.12976686756709563 , 0.09750160558707936, 0.02752286553001629, -0.031582039318031156 , 0.0005538422009938016, 0.004777257511010651, -0.00107730108499558};

synthesis_Daub6_hp[12] =

{-0.00107730108499558, -0.004777257511010651, 0.0005538422009938016 , 0.031582039318031156, 0.02752286553001629, -0.09750160558707936 ,-0.12976686756709563, 0.22626469396516913, 0.3152503517092432,-0.7511339080215775, 0.4946238903983854, -0.11154074335008017};

analysis_Daub10_lp[20] =

{-1.326420300235487e-05,9.358867000108985e-05,-0.0001164668549943862,-0.0006858566950046825,0.00199240529499085,0.0013953517469940798,-0.010733175482979604,0.0036065535669883944,0.03321267405893324,-0.02945753682194567,-0.07139414716586077, 0.09305736460380659, 0.12736934033574265, -0.19594627437659665, -0.24984642432648865, 0.2811723436604265, 0.6884590394525921, 0.5272011889309198, 0.18817680007762133, 0.026670057900950818};

analysis_Daub10_hp[20] =

{-0.026670057900950818, 0.18817680007762133, -0.5272011889309198, 0.6884590394525921, -0.2811723436604265, -0.24984642432648865, 0.19594627437659665, 0.12736934033574265, -0.09305736460380659, -0.07139414716586077, 0.02945753682194567, 0.03321267405893324, -0.0036065535669883944, -0.010733175482979604, -0.0013953517469940798, 0.00199240529499085, 0.0006858566950046825, -0.0001164668549943862, -9.358867000108985e-05, -1.326420300235487e-05};

synthesis_Daub10_lp[20] = {0.026670057900950818, 0.18817680007762133, 0.5272011889309198, 0.6884590394525921, 0.2811723436604265, -0.24984642432648865, -0.19594627437659665, 0.12736934033574265, 0.09305736460380659, -0.07139414716586077, -0.02945753682194567, 0.03321267405893324, 0.0036065535669883944, -0.010733175482979604, 0.0013953517469940798, 0.00199240529499085, -0.0006858566950046825, -0.0001164668549943862, 9.358867000108985e-05, -1.326420300235487e-05};

synthesis_Daub10_hp[20] =

{-1.326420300235487e-05, -9.358867000108985e-05, -0.0001164668549943862, 0.0006858566950046825, 0.00199240529499085, -0.0013953517469940798, -0.010733175482979604, -0.0036065535669883944, 0.03321267405893324, 0.02945753682194567, -0.07139414716586077, -0.09305736460380659, 0.12736934033574265, 0.19594627437659665, -0.24984642432648865, -0.2811723436604265, 0.6884590394525921, -0.5272011889309198, 0.18817680007762133, -0.02667005790095081 8};

analysis_Coif1_lp[6] =

{-0.01565572813546454, -0.0727326195128539, 0.38486484686420286, 0.8525720202122554, 0.3378976624578092, -0.0727326195128539};

analysis_Coif1_hp[6] =

{0.0727326195128539, 0.3378976624578092, -0.8525720202122554, 0.38486484686420286, 0.0727326195128539, -0.01565572813546454};

synthesis_Coif1_lp[6] =

{-0.0727326195128539, 0.3378976624578092, 0.8525720202122554, 0.38486484686420286, -0.0727326195128539, -0.01565572813546454};

synthesis_Coif1_hp[6] =

{-0.01565572813546454, 0.0727326195128539, 0.38486484686420286, -0.8525720202122554, 0.3378976624578092, 0.0727326195128539};

analysis_Coif3_lp[18] =

{-3.459977283621256e-05, -7.098330313814125e-05, 0.0004662169601128863, 0.0011175187708906016, -0.0025745176887502236, -0.00900797613666158, 0.015880544863615904, 0.03455502757306163, -0.08230192710688598, -0.07179982161931202, 0.42848347637761874, 0.7937772226256206, 0.4051769024096169, -0.06112339000267287, -0.0657719112818555, 0.023452696141836267, 0.007782596427325418, -0.003793512864491014};

analysis_Coif3_hp[18] =

{0.003793512864491014, 0.007782596427325418, -0.023452696141836267, -0.0657719112818555, 0.06112339000267287, 0.4051769024096169, -0.7937772226256206, 0.42848347637761874, 0.07179982161931202, -0.08230192710688598, -0.03455502757306163, 0.015880544863615904, 0.00900797613666158, -0.0025745176887502236, -0.0011175187708906016, 0.0004662169601128863, 7.098330313814125e-05, -3.459977283621256e-05};

synthesis_Coif3_lp[18] =

{-0.003793512864491014, 0.007782596427325418, 0.023452696141836267, -0.0657719112818555, -0.06112339000267287, 0.4051769024096169, 0.7937772226256206, 0.42848347637761874, -0.07179982161931202, -0.08230192710688598, 0.03455502757306163, 0.015880544863615904, -0.00900797613666158, -0.0025745176887502236, 0.0011175187708906016, 0.0004662169601128863, -7.098330313814125e-05, -3.459977283621256e-05};

synthesis_Coif3_hp[18] =

{-3.459977283621256e-05, 7.098330313814125e-05, 0.0004662169601128863, -0.0011175187708906016, -0.0025745176887502236, 0.00900797613666158, 0.015880544863615904, -0.03455502757306163, -0.08230192710688598, 0.07179982161931202, 0.42848347637761874, -0.7937772226256206, 0.4051769024096169, 0.06112339000267287, -0.0657719112818555, -0.023452696141836267, 0.007782596427325418, 0.0037935128644910141};

analysis_Coif5_lp[30] =

{-9.517657273819165e-08, -1.6744288576823017e-07, 2.0637618513646814e-06, 3.7346551751414047e-06, -2.1315026809955787e-05, -4.134043227251251e-05, 0.00014054114970203437, 0.00030225958181306315, -0.0006381313430451114, -0.0016628637020130838, 0.0024333732126576722, 0.006764185448053083, -0.009164231162481846, -0.01976177894257264, 0.03268357426711183, 0.0412892087501817, -0.10557420870333893, -0.06203596396290357, 0.4379916261718371, 0.7742896036529562, 0.4215662066908515, -0.05204316317624377, -0.09192001055969624, 0.02816802897093635, 0.023408156785839195, -0.010131117519849788, -0.004159358781386048, 0.0021782363581090178, 0.00035858968789573785, -0.00021208083980379827};

analysis_Coif5_hp[30] =

{0.00021208083980379827, 0.00035858968789573785, -0.0021782363581090178, -0.004159358781386048, 0.010131117519849788, 0.023408156785839195, -0.02816802897093635, -0.09192001055969624, 0.05204316317624377, 0.4215662066908515, -0.7742896036529562, 0.4379916261718371, 0.06203596396290357, -0.10557420870333893, -0.0412892087501817, 0.03268357426711183, 0.01976177894257264, -0.009164231162481846, -0.006764185448053083, 0.0024333732126576722, 0.0016628637020130838, -0.0006381313430451114, -0.00030225958181306315, 0.00014054114970203437, 4.134043227251251e-05, -2.1315026809955787e-05, -3.7346551751414047e-06, 2.0637618513646814e-06, 1.6744288576823017e-07, -9.517657273819165e-08};

synthesis_Coif5_lp[30] =

{-0.00021208083980379827, 0.00035858968789573785, 0.0021782363581090178, -0.004159358781386048, -0.010131117519849788, 0.023408156785839195, 0.02816802897093635, -0.09192001055969624, -0.05204316317624377, 0.4215662066908515, 0.7742896036529562, 0.4379916261718371, -0.06203596396290357, -0.10557420870333893, 0.0412892087501817, 0.03268357426711183, -0.01976177894257264, -0.009164231162481846, 0.006764185448053083, 0.0024333732126576722, -0.0016628637020130838, -0.0006381313430451114, 0.00030225958181306315, 0.00014054114970203437, -4.134043227251251e-05, -2.1315026809955787e-05, 3.7346551751414047e-06, 2.0637618513646814e-06, -1.6744288576823017e-07, -9.517657273819165e-08};

synthesis_Coif5_hp[30] =

{-9.517657273819165e-08, 1.6744288576823017e-07, 2.0637618513646814e-06 -3.7346551751414047e-06, -2.1315026809955787e-05, 4.134043227251251e-05, 0.00014054114970203437, -0.00030225958181306315, -0.0006381313430451114, 0.0016628637020130838, 0.0024333732126576722, -0.006764185448053083, -0.009164231162481846, 0.01976177894257264, 0.03268357426711183, -0.0412892087501817, -0.10557420870333893, 0.06203596396290357,

0.4379916261718371, -0.7742896036529562, 0.4215662066908515,
0.05204316317624377, -0.09192001055969624, -0.02816802897093635 ,
0.023408156785839195, 0.010131117519849788, -0.004159358781386048,
-0.0021782363581090178, 0.00035858968789573785, 0.00021208083980379827};

analysis_Sym6_lp[12] =

{0.015404109327027373, 0.0034907120842174702, -0.11799011114819057,
-0.048311742585633, 0.4910559419267466, 0.787641141030194, 0.3379294217276218,
-0.07263752278646252, -0.021060292512300564, 0.04472490177066578,
0.0017677118642428036, -0.007800708325034148};

analysis_Sym6_hp[12] =

{0.007800708325034148, 0.0017677118642428036, -0.04472490177066578,
-0.021060292512300564, 0.07263752278646252, 0.3379294217276218,
-0.787641141030194, 0.4910559419267466, 0.048311742585633,
-0.11799011114819057, -0.0034907120842174702, 0.015404109327027373};

synthesis_Sym6_lp[12] =

{-0.007800708325034148, 0.0017677118642428036, 0.04472490177066578,
-0.021060292512300564, -0.07263752278646252, 0.3379294217276218,
0.787641141030194, 0.4910559419267466, -0.048311742585633,
-0.11799011114819057, 0.0034907120842174702, 0.015404109327027373};

synthesis_Sym6_hp[12] =

{0.015404109327027373, -0.0034907120842174702, -0.11799011114819057
0.048311742585633, 0.4910559419267466, -0.787641141030194,
0.3379294217276218, 0.07263752278646252, -0.021060292512300564,
-0.04472490177066578, 0.0017677118642428036, 0.007800708325034148};

analysis_Sym10_lp[20] =

{0.0007701598091144901, 9.563267072289475e-05, -0.008641299277022422,
-0.0014653825813050513, 0.0459272392310922, 0.011609893903711381,
-0.15949427888491757, -0.07088053578324385, 0.47169066693843925
0.7695100370211071, 0.38382676106708546, -0.03553674047381755,
-0.0319900568824278, 0.04999497207737669, 0.005764912033581909,
-0.02035493981231129, -0.0008043589320165449, 0.004593173585311828,
5.7036083618494284e-05, -0.0004593294210046588};

analysis_Sym10_hp[20] =

{0.0004593294210046588, 5.7036083618494284e-05, -0.004593173585311828
-0.0008043589320165449, 0.02035493981231129, 0.005764912033581909
-0.04999497207737669, -0.0319900568824278, 0.03553674047381755,
0.38382676106708546, -0.7695100370211071, 0.47169066693843925
0.07088053578324385, -0.15949427888491757 -0.011609893903711381
0.0459272392310922, 0.0014653825813050513 -0.008641299277022422
-9.563267072289475e-05, 0.0007701598091144901};

synthesis_Sym10_lp[20] =

{-0.0004593294210046588, 5.7036083618494284e-05, 0.004593173585311828

-0.0008043589320165449, -0.02035493981231129, 0.005764912033581909
0.04999497207737669, -0.0319900568824278, -0.03553674047381755
0.38382676106708546, 0.7695100370211071, 0.47169066693843925
-0.07088053578324385, -0.15949427888491757, 0.011609893903711381 ,
0.0459272392310922, -0.0014653825813050513, -0.008641299277022422
9.563267072289475e-05, 0.0007701598091144901};

synthesis_Sym10_hp[12] =

{0.0007701598091144901, -9.563267072289475e-05, -0.008641299277022422
0.0014653825813050513, 0.0459272392310922, -0.011609893903711381
-0.15949427888491757, 0.07088053578324385, 0.47169066693843925
-0.7695100370211071, 0.38382676106708546, 0.03553674047381755
-0.0319900568824278, -0.04999497207737669, 0.005764912033581909 ,
0.02035493981231129, -0.0008043589320165449, -0.004593173585311828
5.7036083618494284e-05, 0.0004593294210046588};

analysis_Sym14_lp[28] =

{-2.5879090265397886e-05, 1.1210865808890361e-05, 0.00039843567297594335,
-6.286542481477636e-05, -0.002579441725933078, 0.0003664765736601183,
0.01003769371767227, -0.002753774791224071, -0.029196217764038187,
0.004280520499019378, 0.03743308836285345, -0.057634498351326995,
-0.03531811211497973, 0.39320152196208885, 0.7599762419610909,
0.4753357626342066, -0.05811182331771783, -0.15999741114652205,
0.02589858753104667, 0.06982761636180755, -0.002365048836740385,
-0.019439314263626713, 0.0010131419871842082, 0.004532677471945648,
-7.321421356702399e-05, -0.0006057601824664335, 1.9329016965523917e-05,
4.4618977991475265e-05};

analysis_Sym14_hp[28] =

{-4.4618977991475265e-05, 1.9329016965523917e-05, 0.0006057601824664335,
-7.321421356702399e-05, -0.004532677471945648, 0.0010131419871842082,
0.019439314263626713, -0.002365048836740385, -0.06982761636180755,
0.02589858753104667, 0.15999741114652205, -0.05811182331771783,
-0.4753357626342066, 0.7599762419610909, -0.39320152196208885,
-0.03531811211497973, 0.057634498351326995, 0.03743308836285345,
-0.004280520499019378, -0.029196217764038187, 0.002753774791224071,
0.01003769371767227, -0.0003664765736601183, -0.002579441725933078,
6.286542481477636e-05, 0.00039843567297594335, -1.1210865808890361e-05,
-2.5879090265397886e-05};

synthesis_Sym14_lp[28] =

{4.4618977991475265e-05, 1.9329016965523917e-05, -0.0006057601824664335
-7.321421356702399e-05, 0.004532677471945648, 0.0010131419871842082,
-0.019439314263626713, -0.002365048836740385, 0.06982761636180755
0.02589858753104667, -0.15999741114652205, -0.05811182331771783
0.4753357626342066, 0.7599762419610909, 0.39320152196208885,
-0.03531811211497973, -0.057634498351326995, 0.03743308836285345
0.004280520499019378, -0.029196217764038187, -0.002753774791224071 ,

0.01003769371767227, 0.0003664765736601183, -0.002579441725933078,
-6.286542481477636e-05, 0.00039843567297594335, 1.1210865808890361e-05
-2.5879090265397886e-05};

synthesis_Sym14_hp[28] =

{-2.5879090265397886e-05, -1.1210865808890361e-05 ,0.00039843567297594335
6.286542481477636e-05, -0.002579441725933078 ,-0.0003664765736601183
0.01003769371767227, 0.002753774791224071, -0.029196217764038187,
-0.004280520499019378, 0.03743308836285345, 0.057634498351326995,
-0.03531811211497973, -0.39320152196208885, 0.7599762419610909
-0.4753357626342066, -0.05811182331771783, 0.15999741114652205 ,
0.02589858753104667, -0.06982761636180755, -0.002365048836740385
0.019439314263626713, 0.0010131419871842082, -0.004532677471945648
-7.321421356702399e-05, 0.0006057601824664335, 1.9329016965523917e-05
-4.4618977991475265e-05};

REFERENCES

[1] B. Carnero, A. Drygajlo, “Perceptual Speech Coding and Enhancement Using Frame-Synchronized Fast Wavelet Packet Transform Algorithms,” *IEEE Trans. On Signal Processing*, vol. 47, pp. 1622-1635, June 1999.

[2] Simon Doclo and Marc Moonen, "GSVD-Based Optimal Filtering for Single and Multimicrophone Speech Enhancement", *IEEE Trans. On Signal Processing*, vol. 50, no. 9, September 2002.

[3] J.W. Seok and K.S. Bae, “Speech enhancement with reduction of noise components in the wavelet domain,” *in Proceedings of the ICASSP*, pp. II-1323-1326, 1997.

[4] E. Ambikairajah, G. Tattersall and A. Davis, “Wavelet transform-based speech enhancement,” *in Proceedings of ICSLP*, 1998.

[5] R. Rotili, E. Principi, S. Squartini, and B. Schuller, "A Real-Time Speech Enhancement Framework for Multi-party Meetings" ;*In Proceedings of NOLISP,* 2011, 80-87.

[6] Y. Hu and P.C. Loizou, "Subjective comparison and evaluation of speech enhancement algorithms", *In Proceedings of Speech Communication*, 2007, pp.588-601.

[7] Wan, E., Nelson, A., Peterson, R., Speech Enhancement Assessment Resource (SpEAR) Database. <http://cslu.ece.ogi.edu/nsel/data/SpEAR_database.html>. Oregon Graduate Institute of Science and Technology.

[8] Code::Blocks Release 13.12 rev. 9501, The Code::Blocks Team, http://www.codeblocks.org.

[9] Glade Interface Designer Release 3.66 , 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, United States, https://glade.gnome.org

[10] MATLAB and Signal Processing Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States. http://www.mathworks.com.

[11] MathType version 6.9, Design Science Inc., http://www.dessci.com/en/products/mathtype/

[12] Wan, E., Nelson, A., Peterson, R., Speech Enhancement Assessment Resource (SpEAR) Database. <http://cslu.ece.ogi.edu/nsel/data/SpEAR_database.html>. Oregon Graduate Institute of Science and Technology.

[13] B. C. J. Moore, An Introduction to the Psychology of Hearing. London: Academic Press, 1989.

[14] E. Zwicker and H. Fastl, Psycho-acoustics: Facts and Models. New York: Springer Verlag, 1999.

[15] T. Dau, D. Puschel, and A. Kohlrausch, A quantitative model of the effective signal processing in the auditory system. I. Model structure," J. Acoust. Soc. Am., vol. 99, pp. 3615-3622, 1996.

[16] A. Rix, A. Bourret, and M. Hollier, "Models of human perception," BT Technology Journal, vol. 17, pp. 24-34, 1999.

[17] A. McEwan and A. van Schaik, "A Silicon Representation of the Meddis Inner Hair Cell Model", *in Proceedings of the ICSC Symposia on Intelligent Systems&Applications (ISA'2000)*, 2000, paper 1544-078.

[18] Lyon, R. F. and Mead, C. (1988). "An analog electronic cochlea". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1119–1134.

[19] R. K. Dubey and A. Kumar, "Multiple time-scale estimates of Lyon's auditory features for non-intrusive speech quality assessment," *2016 International Conference on Signal Processing and Communication (ICSC)*, Noida, 2016, pp. 328-333.

[20] F. Müller and A. Mertins, "On using the auditory image model and invariant-integration for noise robust automatic speech recognition," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, 2012, pp. 4905-4908.

[22] Lawrence R. Rabiner and Ronald W. Schafer, *Introduction to Digital Speech Processing*, Foundations and Trends in Signal Processing, vol. 1, pp. 1–194, 2007.

[22] Noise Reduction in Speech Applications, Gillian M. Davis , CRC Press, 18 Apr 2002.

[23] *Girod, B., Rabenstein, R., Stenger, A., Signals and Sytems, John Wiley & Sons, Chister, U.K., 2001.*

[24] Girod, B., Rabenstein, R., Stenger, A., *Signals and Sytems*, John Wiley & Sons, Chister, U.K., 2001.

[25] Pawar A.P., Choudhari K.B. and Joshi M.A., Review of Single Channel Speech Enhancement Methods in Spectral Domain, *International Journal of Applied Engineering Research*, ISSN 0973-4562 Vol. 7 No.11 (2012).

[26] Lim, J. S., Oppenheim, A.V. (1979). "Enhancement and bandwidth compression of noisy speech." *Proc.of the IEEE* ,67(12): 1586-1604.

[27] Sugiura, Y., Kawamura, A., Iiguni, Y., A Comb Filter with Adaptive Notch Gain for Periodic Noise Reduction, *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2012 Asia-Pacific, 2012, Page(s): 1-5.

[28] M. G. C. Jiménez, D. E. T. Romero and G. J. Dolecek, "Comb filters: characteristics and applications," in *Encyclopedia of Information Science and Technology,* 3rd Edition, IGI global, pp. 4062-4070, 2015.

[29] J. T. Graf and N. Hubing, "Dynamic time warping comb filter for the enhancement of speech degraded by white Gaussian noise, "*Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, Minneapolis, MN, USA, 1993, pp. 339-342 vol.2.

[30] Hogenauer, Eugene B. (April 1981)."An economical class of digital filters for decimation and interpolation". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2): 155–162.

[31] H.D. Gahng, K.S. "Bae, Single channel adaptive noise cancelation for enhancing noisy speech",*International Symposium on Speech, Image Processing and Neural Networks*, vol.1, pp. 343 – 346, April 1994.

[32] Widrow, B.; Glover, J.R., Jr.; McCool, J.M.; Kaunitz, J.; Williams, C.S.; Hearn, R.H.; Zeidler, J.R.; Eugene Dong, Jr.; Goodlin, R.C., "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol.63, no.12, pp.1692,1716, Dec. 1975.

[33] Sambur, M., "Adaptive noise canceling for speech signals, "Acoustics, Speech and Signal Processing, *IEEE Transactions on*, vol.26, no.5, pp.419,423, Oct 1978.

[34] Speech Signal Enhancement Using Adaptive Noise Cancellation Techniques, Allam Mousa, Marwa Qados, Sherin Bader , Canadian Journal on Electrical and Electronics Engineering Vol. 3, No. 7, September 2012

[35] Breining, C.; Dreiscitel, P.; Hansler, E.; Mader, A; Nitsch, B.; Puder, H.; Schertler, T.; Schmidt, G.; Tilp, J., "Acoustic echo control. An application of very-high-order adaptive filters," *Signal Processing Magazine, IEEE*, vol.16, no.4, pp.42,69, Jul 1999.

[36] B. Widrow, ``Adaptive Sampled-Data Systems -A Statistical Theory of Adaptation,'' *IRE WESCON Convention Record*, 4:74-85, 1959.

[37] MIT Open Course Ware, 2.161 Signal Processing: Continuous and Discrete, Fall 2008.

[38] Boll, S. F., Suppression of Acoustic Noise in Speech Using Spectral Subtraction, *IEEE Tran. on Acoustics, Speech and Signal Processing ASSP-27*, pp. 113-120, 1979.

[39] P.M. Crozier, B.M.G. Cheetham, C. Holt, and E. Munday, "The use of linear prediction and spectral scaling for improving speech enhancement", *in Proc.Eurospeech*, 1993.

[40] Ephraim, Y., "Statistical-model-based speech enhancement systems,"*Proceedings of the IEEE*, vol.80, no.10, pp.1526,1555, Oct 1992.

[41] Ephraim, Y.; Van Trees, H.L., "A signal subspace approach for speech enhancement,"*Speech and Audio Processing, IEEE Transactions on*, vol.3, no.4, pp.251,266, Jul 1995.

[42] McAulay, R.; Malpass, M., "Speech enhancement using a soft-decision noise suppression filter," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol.28, no.2, pp.137,145, Apr 1980.

[43] N. W. D. Evans, J. S. Mason, W. M. Liu and B. Fauve, "On the fundamental limitations of spectral subtraction: An assessment by automatic speech recognition," *Signal Processing Conference, 2005 13th European*, Antalya, 2005, pp. 1-4.

[44] W. B. Davenport, Jr., and W. L. Root, *Random Signals and Noise*. New York: McGraw-Hill, 1968.

[45] Wiener N., *Extrapolation, Interpolation and Smoothing of Stationary Time Series,* MIT Press Cambridge, MA, 1949.

[46] Akaike, H., "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol.19, no.6, pp.716,723, Dec 1974.

[47] Marwan Jaber (2011). The Ultra High Speed LMS Algorithm Implemented on Parallel Architecture Suitable for Multidimensional Adaptive Filtering, Adaptive Filtering, Dr Lino Garcia (Ed.), ISBN: 978-953-307-158-9, InTech, DOI: 10.5772/25070. Available from: http://www.intechopen.com/books/adaptive-filtering/the-ultra-high-speed-lms-algorithm-implemented-on-parallel-architecture-suitable-for-multidimensiona

[48] Pols, L., "Flexible human speech recognition," Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on, vol., no., pp.273,283, 14-17 Dec 1997.

[49] Bouden Toufik and Nibouche Mokhtar (2012). The Wavelet Transform for Image Processing Applications, Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology, Dr. Dumitru Baleanu (Ed.), InTech, DOI: 10.5772/35982. Available from:https://www.intechopen.com/books/advances-in-wavelet-theory-and-their-applications-in-engineering-physics-and-technology/the-wavelet-transform-for-image-processing-applications.

[50] Graps, A. L., An introduction to Wavelets, *IEEE Computational Science and Engineering,* Vol. 2, No. 2, pp. 50-61, ISSN: 1070-9924, 1995.

[51] Meyer, Y., Wavelet With Compact Support. *Zygmund Lectures,* University Chicago 1987.

[52] Mallat, S., A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Recognition and Machine Intelligence,* Vol. 11, No. 7, pp. 674-693, ISSN 0162-8828, July 1989.

[53] Daubechies, I., Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics,* Vol. 41, pp. 909-996, 1988.

[54] Cody, M. A. (1994). The Wavelet Packet Transform, *Dr. Dobb's Journal,* Vol. 19, Apr. 1994.

[55] Gaouda, AM.; El-Saadany, E.F.; Salama, M.M.A; Sood, V.K.; Chikhani, AY., "Monitoring HVDC systems using wavelet multi-resolution analysis," *Power Systems, IEEE Transactions on*, vol.16, no.4, pp.662,670, Nov 2001.

[56] A. V. Oppenheim, R. W. Schafer, *Digital signal processing*, Prentice-Hall, 1975.

[57] Wickerhauser, 1994; Graps, 1995; Burrus et al., 1998; David, 2002 & 40 Oppenheim & Schafer, 2010.

[58] F. Hlawatsch, G. F. Boudreaux-Bartels, "Linear and Quadric Time-Frequency Signal Representations", *IEEE SP Magazine*, pp.21-67, April 1992.

[59] Adhemar Bultheel, "Wavelets with applications in signal and image processing(2003)", available at http://www.win.tue.nl/~morscheh/2DE10/alltxt.pdf(Accessed: 10 August 2011).

[60] Grossman, A. & Morlet, J., Decomposition of hardy functions into square integrable wavelets of constant shape, *SIAM Journal of Mathematical Analysis,* Vol. 15, No. 4, pp. 723-736, ISSN 00361410, (1984).

[61] Sweldens, W., Wavelets: what next?, *Proceedings of the IEEE,* vol. 84, No. 4, pp. 680-685, ISSN 0018-9219, 1996.

[62] Burrus, C. S.; Gopinath, R. A. & Guo, H., *Introduction to Wavelets and Wavelet Transforms: A primer,* Prentice Hall, 1998.

[63] Sweldens, W., The Lifting Scheme: A new philosophy in biorthogonal wavelet constructions, *Wavelet Applications in Signal and Image Processing III,* pp. 68-79, Proc. SPIE 2569, 1995.

[64] Prof. Dr. Ir. M. Steinbuch, Dr. Ir. M.J.G. van de Molengraft, June 7 (2005), Eindhoven University of Technology, Control Systems Technology Group Eindhoven, "Wavelet Theory and Applications", a literature study, R.J.E. Merry, DCT 2005.53.

[65] Gregory W. Wornell, *Synthesis, Analysis, and Processing of Fractal Signals*, Thesis (Ph.D.), Research Laboratory of Electronics Massachusetts Institute of Technology Cambridge, Massachusetts, 1991.

[66] Aggarwal R., Singh J. K., Gupta V. K., Rathore S., Tiwari M., Khare A., Noise Reduction of Speech Signal using Wavelet Transform with Modified Universal Threshold, International Journal of Computer Applications (0975-8887), Volume 20, No.5, April 2011.

[67] Veselinovic, D.; Graupe, D., "A wavelet transform approach to blind adaptive filtering of speech from unknown noises," Circuits and Systems II: Analog and Digital Signal Processing, *IEEE Transactions on*, vol.50, no.3, pp.150,154, Mar 2003.

[68] Mallat, S. G., Multiresolution Representations and Wavelets, PhD thesis, Univ. Pem., Philadelphia, PA, Aug. 1988.

[69] E. Erzin, A.E. Cetin, and Y. Yardimci, "Subband analysis for robust speech recognition in the presence of car noise," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Detroit, MI, vol. 1, pp. 417–420, 1995.

[70] L. Du, R. Xu, F. Xu, D. Wang, H. Chen, "Research on key parameters of speech denoising algorithm based on wavelet packet transform," *Proc. of the IEEE International Conference on Computer Science and Information Tecnology*, Chenghu, vol. 6, pp. 551-556, July 2010.

[71] Z. Wang, S. Li, "Discrete Fourier Transform and Discrete Wavelet Packet Transform in Speech Denoising," *Proc. of the IEEE International Congress on Image and Signal Processing*, China, pp. 1588-1591, Oct. 2012.

[72] M. Bahoura and J. Rouat, "Wavelet speech enhancement based on the teager energy operator," *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 10–12, Jan. 2001.

[73] D. Veselinovic and D. Graupe, "A wavelet transform approach to blind adaptive filtering of speech from unknown noises," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 3, pp. 150–154, Mar. 2003.

[74] C.-T. Lu and H.-C. Wang, "Speech enhancement using perceptually constrained gain factors in critical-band-wavelet-packet transform," *Electron. Lett*., vol. 40, no. 6, pp. 394–396, Mar. 2004.

[75] S. Li, "Speech Denoising Based on Improved Discrete Wavelet Packet Decomposition," *in Proc. International Conference on Network Computing and Information Security*, Guilin, pp. 415-419, May 2011.

[76] Farooq, O. & Datta, S. (2003), "Wavelet-based denoising for robust feature extraction for speech recognition", *Electronics Letters*, vol. 36, no. 1, pp. 163- 165, January 2003.

[77] Yao, J. & Zhang, Y. (2002), "The Application of Bionic Wavelet Transform to Speech Signal Processing in Cochlear Implants using Neural Network Simulations", *IEEE Transactions on Biomedical Engineering*, vol. 49 , no. 11, pp. 1299 – 1309, November. 2002.

[78] Bahoura, M. & Rouat, J. (2001), "Wavelet Speech Enhancement Based on the Teager Energy Operator", IEEE Signal Processing Letters, vol. 8, no. 1, pp. 10- 12, January 2001.

[79] Kaiser, J. F. (1993), "Some useful properties of Teager's energy operators", *Proceedings of IEEE ICASSP '93*, vol. 3, pp. 149–152, Apr. 1993.

[80] Ephraim, Y. & Malah, D., "Speech enhancement using a minimum mean square error short time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, June 1984.

[81] Ayat, S.; Manzuri, M.T.; Dianat, R., "Wavelet based speech enhancement using a new thresholding algorithm," Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004 International Symposium on, vol., no., pp.238,241, 20-22 Oct. 2004.

[82] B. y. Xia and C. c. Bao, "A wavelet fusion method for speech enhancement,"*Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, Beijing, 2012, pp. 473-476.

[83] Cohen R., *Signal Denoising Using Wavelets,* Project Report, Department of Electrical Engineering Technion, Israel Institute of Technology, 2012.

[84] David L. Donoho and Jain M. Johnstone. *Ideal spatial adaptation by wavelet shrinkage*. Biometrika, 81(3):425–455, 1994.

[85] Daubechies, I., Ten lectures on Wavelets, *SIAM,* Philadelphia, 1992.

[86] Masud, S., *VLSI system for Discrete Wavelet Transforms,* PhD thesis, Department of Electrical Engineering, The Queen's University of Belfast, Ireland, 1999.

[87] Soman K.P., Resmi N.G., Ramachandran K.I., *Insight Into Wavelets : from Theory to Practice*, PHI Learning Pvt. Ltd., 2010.

[88] Ruqiang Yan, *Base Wavelet Selection Criteria for Non-stationary Vibration Analysis in Bearing Health Diagnosis*, Thesis (Ph.D.), University of Massachusetts Amherst, 2007.

[89] V.S.R Kumari, Dileep Kumar Devarakonda, "A Wavelet Based Denoising of Speech Signal", International Journal of Engineering Trends and Technology (IJETT) – Volume 5 number 2 - Nov 2013.

[90] L. Ru-Wei, C. Long-Tao and L. Yang, "Pitch Detection Method for Noisy Speech Signals Based on Wavelet Transform and Autocorrelation Function," *Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on*, Beijing, 2013, pp.

153-156.

[91] Rioul, O.; Duhamel, P., "Fast algorithms for discrete and continuous wavelet transforms," *Information Theory, IEEE Transactions on*, vol.38, no.2, pp.569,586, March 1992.

[92] “Subjective video quality assessment methods for multimedia applications,” International Telecommunication Union (ITU) Radiocommunication Assembly, Geneva, Switzerland, 2008, ITU-U P.910.

[93] C. Schremmer, T. Haenselmann, F. Bömers, “Wavelets in RealTime Digital Audio Processing: A Software for Understanding Wavelets in Applied Computer Science” Workshop on Signal Processing Applications (WoSPA), Brisbane, Australia, 2000.

[94] Strang, G.; Nguyen, T.; Wavelets and filter banks, Wellesley-Cambridge Press, 1996.

[95] Florian Bomers, Wavelets in real time digital audio processing: Analysis and sample implementations, PhD Thesis, Department of Computer Science, University of Mannheim, Germany, 2000.

[96] J. Piskorowski, “Digital Q-varying notch IIR filter with transient suppression,” IEEE Trans. on instrumentation and measurement, vol. 59, No.4, Apr., 2010.

[97] C. M. Wang, W. C. Xiao, "Second-Order IIR Notch Filter Design and Implementation of Digital Signal Processing System", Applied Mechanics and Materials, Vols. 347-350, pp. 729-732, Aug. 2013.

[98] S. Kudrle, M. Proulx, P. Carrières and M. Lopez, "Fingerprinting for Solving A/V Synchronization Issues within Broadcast Environments," in *SMPTE Motion Imaging Journal*, vol. 120, no. 5, pp. 36-46, July 2011.

[99] S. Mallat, A Wavelet Tour of Signal Processing. New York: Academic, 1998.

[100] Harjeet Kaur, Rajnessh Talwar, Optimal Selection of Wavelet and Threshold Using Cuckoo Search for Noise Suppression in Speech Signals. , IRECOS vol. 9, September 2014.

[101] Y. Cai, J. Yuan, X. Ma and C. Hou, "Low Power Embedded Speech Enhancement System Based on a Fixed-Point DSP, "*Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on*, Chengdu, 2009, pp. 132-136.