EVALUATION OF SPEECH ENHANCEMENT TECHNIQUES FOR SPEAKER RECOGNITION IN NOISY ENVIRONMENTS

by

Abdel-Aziz El-Solh

A thesis submitted to

The Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science (M.A.Sc.) in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada

August, 2006

Your file Votre référence
ISBN: 978-0-494-18312-0
Our file Notre référence
ISBN: 978-0-494-18312-0

ABSTRACT

In automatic speaker recognition (ASR) applications, the presence of background noise severely degrades recognition performance. There is a strong demand for speech enhancement algorithms capable of removing background noise. In this thesis, a Gaussian mixture model based automatic speaker recognition system is used for evaluating the performance of five different speech enhancement techniques. Previously, it was shown that these techniques improved the SNR of the speech signals corrupted by noise but their effect on the speaker recognition performance was not fully investigated. In this work, we implement these enhancement techniques and evaluate their performance as preprocessing blocks to the ASR engine. We evaluate the performance based on speaker recognition accuracy, average segmental signal-to-noise ratio and perceptual evaluation of speech quality (PESQ) scores. We combine clean speech from the TIMIT database with eight different types of noise from the NOISEX-92 database representing synthetic and natural background noise samples and analyze the overall system performance. Simulation results show that the system is capable of reducing noise with little speech degradation and the overall recognition performance can be improved at a range of different signal-to-noise ratios (SNR) with different noise types. Furthermore, results show that different enhancement techniques have different strengths and weaknesses, depending on their application and the background noise type.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God for blessing me with the ability and endurance to complete my studies.

I would then like to thank Dr. Aysegul Cuhadar and Dr. Rafik Goubran for their direction, guidance, help, resources and commitment throughout the course of my thesis. Their efforts went well beyond the call of duty and were an invaluable asset to this research.

I also thank Joseph Gammal, Geoffrey Green and Zhong Lin for sharing their knowledge and proficiency in the areas of digital signal processing, automatic speaker recognition and signal enhancement. I want to acknowledge Blazenka Power for her administrative assistance and Christine McGregor for her editing skills.

A very special acknowledgement goes out to my parents, Wesam and Rashiqa El-Solh, who were an unlimited source of motivation and support throughout my undergraduate, as well as graduate, years. May God bless them.

I want to thank Communications and Information Technology Ontario (CITO) and Mitel for their financial assistance on this project.

Last, but by no means least, my utmost appreciation and thanks goes out to my colleagues and friends, Kamal Harb, Andrew Soon, John Tran and Vicky Laurens.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

A/D	Analog to Digital
ANC	Adaptive Noise Cancellation
ASR	Automatic Speaker Recognition
ASV	Automatic Speaker Verification
BSS	Blind Source Separation
CTR	Coefficient to Threshold Ratio
EM	Expectation Maximization
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GMM	Gaussian Mixture Model
HPF	High Pass Filter
LBG	Linde, Buzo, Gray
LPC	Linear Prediction Coefficients
LPCC	Linear Prediction Cepstral Coefficients
LPF	Low Pass Filter
MFCC	Mel-Frequency Cepstral Coefficients
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
PDF	Probability Density Function
PESQ	Perceptual Evaluation of Speech Quality
PWAD	Perceptual Wavelet Adaptive Denoising
PWT	Perceptual Wavelet Transform
SE	Speech Enhancement
SNR	Signal to Noise Ratio
SS	Spectral Subtraction
STDFT	Short Time Discrete Fourier Transform
STFT	Short Time Fourier Transform
STSA	Short Time Spectral Amplitude
TDSE	Two-Dimensional Spectral Enhancement
VAD	Voice Activity Detection/Detector

CHAPTER 1
INTRODUCTION

1.1 Speaker Recognition

Speech processing can occur in several areas (including synthesis, analysis, recognition and coding) with relevance to several commercial applications (like biometric security, human-machine interaction and communications). Speaker recognition, in its simplest form, is the use of a machine to detect a speaker from a given database. Throughout this thesis, the terms *speaker* or *claimant* refer to the person using the system for recognition while *user* identifies a model or waveform pre-determined and stored in a recognition database and a *group* represents a collection of valid users or pre-registered speakers that are known to the system. A speaker or claimant provides an *utterance* to the system for analysis. An overview of a typical speaker recognition system is shown in Figure 1.1.

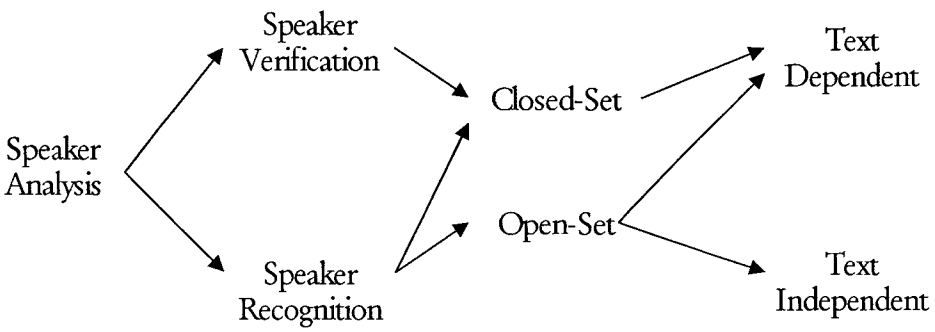

Figure 1.1 Overview of speaker recognition

Automatic Speaker Verification (ASV) attempts to provide a binary answer verifying the identity of the claimant. Therefore, a speaker will claim to be a certain user and the system will accept or reject the claim, based on analysis of his/her utterance. To provide maximum matching confidence, ASV systems are inherently text-dependent in that a speaker will utter a given phrase. A typical implementation of ASV is in applications where access is restricted to a certain group of people. In this situation, a claimant (who has been previously enrolled in the system) will request admission to protected resources by providing his information via an encrypted smart card (hence making a claim). The system will then prompt the claimant to utter the password and verify this with pre-trained models and based on a matching threshold, will grant or deny access to the speaker [1].

Automatic Speaker Recognition (ASR) assumes no a priori knowledge of the speaker and can be either closed-set or open-set. Closed-set speaker recognition attempts to identify the speaker based on a database of pre-trained user models that encompasses members of a group. The system will return the user index with the highest speaker to user matching score. Open-set recognition extends this concept by incorporating an extra user model within the database that is trained using several 'other' users not part of the group. The theory here is that if a speaker is not part of the group, his utterance will be matched to this 'other' entry. Therefore, switching recognition modes between closed- and open-set is as simple as adding the required entry in the database and/or refining thresholds to improve the degree of confidence. Indeed, open-set ASR and ASV

are very similar in functionality, where open-set ASR is sometimes considered to be a more relaxed ASV. ASR systems can be either text-dependent or text-independent, depending on the application. With text-independent systems, where only identification of a user is required (e.g. in teleconferencing), the system will extract speaker-related information regardless of the phrase uttered from the speaker.

1.2 Speech Enhancement

There are several factors that can contribute to reduced speaker recognition performance, including noise and reverberation. One of the most detrimental to recognition is background noise. Several speech enhancement techniques have been developed to remove background noise without significantly altering speech-related features Speech enhancement can be decomposed into two main areas; single-channel and multi-channel approaches.

1.2.1 Single-Channel Techniques

Single microphone enhancement algorithms assume only one input channel containing both noise and speech. For such a system to be robust, it cannot make any direct assumptions regarding the type of noise.

Spectral subtraction (SS) is a popular and computationally efficient means of noise removal involving the suppression of noise in the spectral or frequency domain [2]. The basis of this technique lies in the assumption that an observed signal from the channel is a linear combination of speech and uncorrelated noise as given in (1.1), also

translating to a linear combination of sub-band energies in the spectral domain as given in (1.2). Reproducing the speech becomes a simple matter of subtracting the noise spectrum from the observed signal.

$$s(t) = x(t) + n(t) \tag{1.1}$$

$$|S(f)|^2 = |X(f)|^2 + |N(f)|^2 \tag{1.2}$$

Where $s(t)$ represents the observed or noisy signal, $x(t)$ is the clean speech signal and $n(t)$ is the noise signal. $S(f)$, $X(f)$ and $N(f)$ are their Fourier transform equivalents.

A natural byproduct of SS is musical noise; a consistent acoustic tone or distortion introduced by the de-noising process. This is primarily induced when reconstructing signals that have had a portion of their spectral coefficients "zeroed-out" during the subtraction process. To overcome this, an over-subtraction model is developed that prevents coefficients from being set to zero. This is accomplished by maintaining a fraction of the original coefficients (i.e. if a processed coefficient is equal to zero, then it is replaced by a fraction of its original value) [3].

A crucial aspect of SS is noise estimation, since the degree of noise suppression is directly proportional to how accurately the noise can be estimated. Obviously, this is not a trivial task and is a challenge in its own right. Several techniques have been developed in the literature for estimating noise in different domains (e.g. frequency, wavelet, cepstral, etc.). Recent algorithms take advantage of the short time, bursty nature of

speech to track noise statistics over time. Short-time analysis windows are used to observe change in signal power with time. Quantile based noise estimation algorithms conclude that, within a defined time-frequency window, noise power will remain stationary and hence can be estimated as an arithmetic fraction of total window power. Wavelet based techniques make use of the multi-resolution property of the wavelet transform to provide finer frequency resolution at lower sub-bands containing speech. Other approaches attempt to estimate noise-only frames from noise-plus-speech ones. Effectively, they implement voice activity detection (VAD). Based on this estimate of speech per frame, and consequently the SNR per frame, a noise model is formed from non-speech segments. This technique is also known as minima noise tracking or spectral noise floor estimation [4], [5].

1.2.2 Multi-Channel Techniques

Microphone arrays have been used to beam-form towards a speaker and tune out or attenuate background noise. Given a stationary (i.e. not in motion) source, a speech reference matrix can be used to project a beam at the user while another blocking matrix is used to obtain a noise reference. This can be further extended to apply an adaptive beam-former designed to track a moving speaker.

Multi-microphone setups have also been used to implement blind-source separation to detach valid speech from background noise. Blind source separation (BSS) involves the extraction of unknown sources, based on a mixture of signals [6], [7]. The

assumption here is that, given a room (or scenario) with multiple signal sources, a different mixture of these is observed at each microphone of the array. Furthermore, the system is completely oblivious or blind to the nature of the source signal and information about the mixtures observed. Typical de-noising implementations of BSS imply a strict but reasonably representative assumption of statistical independence between the source signals. A mixing matrix that dictates how these sources were super-imposed is estimated using contrast functions that are derived from entropy, mutual independence, higher-order decorrelations and measures of divergence [6], [8]. Other BSS systems try to estimate the mixing matrix by strictly adhering to higher-order statistics such as kurtosis. Essentially, the kurtosis is a fourth order measure of "how Gaussian" a signal is. A higher kurtosis value indicates that a signal tends towards a super-Gaussian property (speech is considered to follow a largely Laplacian distribution lying within this case), while a lower kurtosis score represents a more Gaussian signal. Hence, based on a measure of kurtosis per incoming channel, one can be chosen with the highest speaker content as reference and the remaining inputs used to adaptively filter out background noise [9].

1.3 Problem Statement and Thesis Objectives

Most ASR systems are trained within controlled conditions while testing settings are generally unspecified, for example, training with a handset locally and testing remotely via a communications channel where the far-end can be a handset or hands-free end point. Performances of ASR engines degrade severely with the presence of background

noise. This can be due to acoustic properties of the recording and/or transmission media, as well as the environmental noise present during testing. As the statistical properties differ between noise types, recognition accuracy will be heavily dependent on how each noise type affects speech properties. Another performance limiting factor is the power of inherent noise. A higher noise power will obviously destroy more speech data and make its recovery more difficult. This thesis will analyze performance of a typical ASR engine with a variety of noise degradation present at a range of noise power levels. This research also implements and analyzes single channel speech enhancement techniques as pre-processing blocks for ASR systems.

1.4 Thesis Contributions

In this thesis, a typical ASR engine has been implemented to evaluate the performance of various noise removal techniques. This common baseline ASR engine allows for direct relation between results captured for each algorithm. Furthermore, it is more beneficial that the experimental results reflect the performance changes of pre-processing speech enhancement only and not any other variable in the system. The following are the contributions of this thesis:

- We compared the performance of two popular speech feature extraction techniques (MFCC and LPCC) in the presence of different types of background noise, for a range of SNRs. Based on recognition accuracy results, MFCC was more noise-robust and chosen for future experimentation with speech enhancement.

- Five adaptive speech enhancement techniques were implemented and four of them modified to allow for recognition accuracy improvement. They were analyzed both individually and compared with one another, given several background noise types at noise levels of 20, 15, 10, 5 and 0 dB input SNR. These algorithms and their modifications (detailed in Chapter 5) are outlined as follows:
  - A two-pass quantile spectral noise estimation technique was modified to use a simple and effective Wiener filter. This noise estimation technique was chosen based on its ability to provide good SNR improvement at lower input SNR levels, because of its q-table training. The Wiener filter was followed by a processing block that allowed a certain fraction of the noise to remain, preventing coefficients from being attenuated to zero.
  - A quantile based noise estimation technique applied in the perceptual wavelet packet domain was implemented. The perceptual wavelet transform allowed for better improvement at higher input SNR levels. Using this noise estimate, an adaptive wavelet threshold was calculated and the wavelet coefficients thresholded using a modified Ephraim and Malah suppression filter.
  - Voice activity detection based noise floor estimation was coupled with a log spectral amplitude based filter. This noise estimation technique was chosen because of its VAD algorithm which allowed for good speech estimation, minimizing the effects of distortion, while the noise floor detector ensured that noise was not being overestimated. The log spectral amplitude filter was

designed such that the residual noise after filtering is white in nature and not modulated in any way, causing distortion or musical noise.

- To benefit from the two-pass spectral's ability to remove noise without distorting speech and the perceptual wavelet's better performance at higher input SNR levels, the first two techniques were combined such that the output of the two-pass algorithm was passed on to the wavelet algorithm.
- Finally, the popular minimum mean square error (MMSE) spectral subtraction process was implemented without any modifications for comparison to the other modified techniques.

- These algorithms were compared using three metrics; recognition accuracy of an ASR system, the popular industry standard Perceptual Evaluation of Speech Quality- Mean Opinion Score (PESQ-MOS) voice quality measurement technique and average segmental SNR improvement. Each algorithm was tested with eight different noise types including white, pink, car, HF channel, factory, babble, F16 and machine gun. The two-pass technique showed best improvement at lower input SNR values. The VAD based noise estimation method trailed behind the two-pass technique, yet exhibited the same general performance trends with the advantage of not requiring any q-table training. The perceptual wavelet packet algorithm showed its ability to work well at higher SNRs, specifically with white and pink noise. This performance was further improved when the perceptual wavelet technique was combined with the two-pass spectral technique. Finally, the MMSE

algorithm showed best average segmental SNR improvements, yet returned negative recognition accuracy metrics.

1.5 Thesis Outline

This thesis is organized into six chapters. Chapter 1 presents a general overview of the goals and types of speaker recognition. It also introduces background noise as a significant deterrent to performance and proceeds to outline different methods in the literature to combat this. Thesis objectives and contributions are also mentioned.

Chapter 2 will give the details of the ASR system used as the recognition engine for experimentation. It follows the steps of how a speech waveform is processed for recognition, where it is first broken down into smaller and more manageable segments. For each segment, speech features are extracted and are compared with other features already present in a pre-trained user database. A description is provided of how the matching between claimants and users in a database is accomplished. The LBG clustering algorithm as well as Gaussian Mixture Models GMMs are presented.

Chapter 3 presents the five main algorithms used in this thesis for speech enhancement. It gives an overview of different stages in the recognition pipeline where the effects of noise can be reduced, as well as some of the reasoning behind quantile processing used in the two-pass spectral technique. A description of the perceptual wavelet packet transform used in the wavelet thresholding based algorithm is also given and its strengths highlighted. This chapter then provides a brief outline of the minimum

mean square error technique. Finally, a voice activity detector used for speech estimation is detailed.

Chapter 4 outlines the experimental setup used for analysis. The database used for training and testing, noise type and level used and choice of system parameters are presented. The database used for noise samples is also presented and briefly outlined. Recognition accuracy, average segmental signal to noise ratio and perceptual evaluation of speech quality are described in more detail as the performance metrics used to analyze and compare the enhancement algorithms.

Chapter 5 presents the modifications made or post-processing appended to the algorithms described in chapter 3 and reports their performance with respect to eight different types of background noise, namely: white, pink, car, HF channel, factory, babble, F16 and machine gun. Noise is added to the system at 20, 15, 10, 5 and 0 dB SNR. Analysis is done based on recognition accuracy, PESQ-MOS and average segmental SNR metrics as described in chapter 4.

Chapter 6 builds upon the results presented in the previous chapter to give a comparison analysis of each algorithm with one another with respect to recognition accuracy. Based on these results, the performance of each algorithm is discussed and trends highlighted.

Chapter 7 draws conclusions based on discussions presented in chapters 5 and 6 and possible future work is outlined. Appendices A & B present the average segmental SNR and PESQ-MOS results of each algorithm side by side for direct comparison. Finally Appendix C contains all recognition accuracy results with and without noise.

CHAPTER 2
SPEAKER RECOGNITION

2.1 Outline of ASR Systems

The problem of speaker recognition boils down to the matching between speech spoken by an unknown user and a set of utterances in a database. First, the raw speech input is processed such that a more efficient representation of the speech can be obtained. This alternative representation is then used to compare the processed input speech with a set of known users in a database. Based on the similarity, a decision is made whether or not the input speech was uttered by a given speaker. An outline of the speaker recognition engine is shown in Figure 2.1.

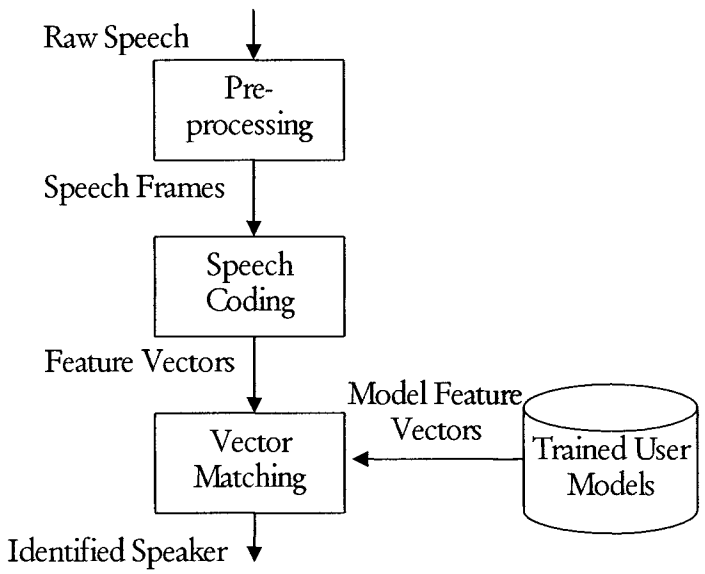

Figure 2.1. A typical speaker recognition engine

2.2 Pre-Processing

In the pre-processing stage, features that can identify a particular speaker are extracted from the raw speech signal.

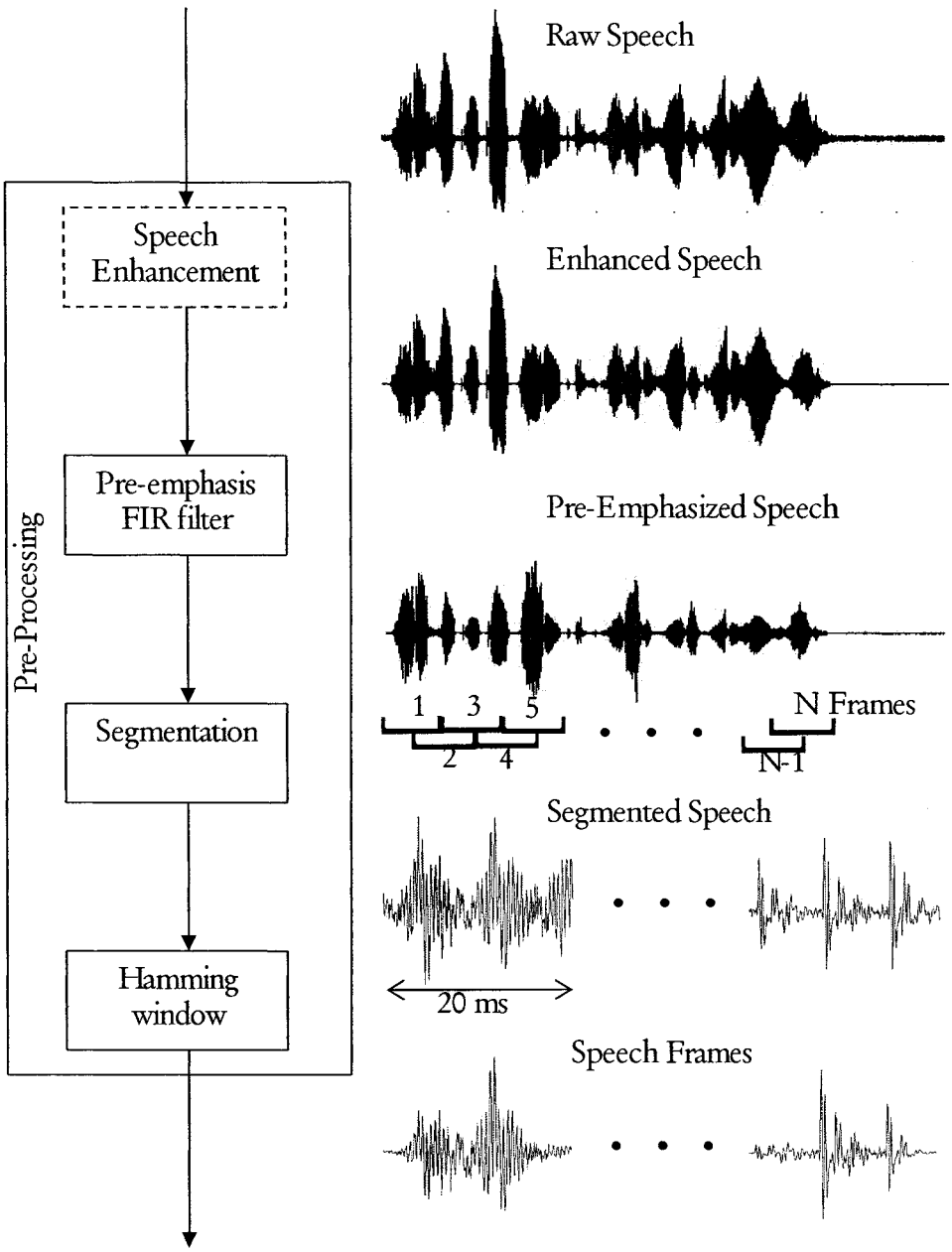

Figure 2.2. Overview of pre-processing block

Depending on the application of the system, the raw speech may first be passed through an enhancement block as given in Figure 2.2. In noisy environments, where speech is distorted, it would be beneficial to employ noise removal algorithms that alter the raw speech such that resulting recognition accuracy is improved. It is in this sub-block that the focus of this thesis lies. A detailed description of the enhancement process is given in Chapter 3.

From an acoustic sense, the energy of speech tends to decrease as its frequency increases. A pre-emphasis filter is designed to eliminate this effect by increasing the relative high-frequency energy of the speech. Therefore, the outcome of the filter allows total speech energy to be uniformly distributed amongst their sub-bands. It is a FIR filter with a transfer function given by [1][2][3]:

$$H(z) = 1 - 0.95z^{-1} \tag{2.1}$$

In order to capture the time variance of the input utterance, speech is broken down into overlapping frames. The percentage of overlap chosen between frames is directly proportional to the time resolution required. A typical and adequate frame size is 20 ms with 10 ms (i.e. 50%) overlap, producing an overall frame rate of 100 analysis frames per second, which has been shown to be sufficient for speech tracking [1] [10] [11] [12]. Finally, a Hamming window is applied to each frame to counteract the edge effects of segmentation/framing.

2.3 Speech Coding/Feature Extraction

Now that we have broken the speech down to segments and adapted for windowing and acoustic phenomena, we can proceed to represent the frames in a format that best captures their speaker related properties. Speech waveforms are random and direct sample by sample processing would prove to be computationally and algorithmically inefficient, hence it is encoded into a set of coefficients commonly known as feature vectors. The goal here is to significantly reduce the number of parameters required to represent the speech waveform without losing embedded speaker properties. This process of feature extraction is also referred to as speech parameterization. Although many techniques have been developed for feature extraction in different domains (e.g. using the wavelet transform), there are two main feature extraction techniques used in ASR systems that offer their own unique advantages.

2.3.1 Linear Prediction Cepstral Coefficients (LPCC)

Linear Prediction Coefficients (LPC) try to model the vocal tract that produced the speech waveform. The assumption here is that a speech signal is produced by a loud speaker placed at the end of a tube of different dimensions as shown in Figure 2.3 [13] [14]. The glottis (space between the vocal cords) represents the loudness of the speaker, described by its intensity and frequency. The vocal tract (throat and mouth) form the tube that is described by its harmonics called formants. LPC analysis encodes a speech signal by removing the effects of the formants from the speech signal, and estimating the intensity and frequency of the remaining waveform.

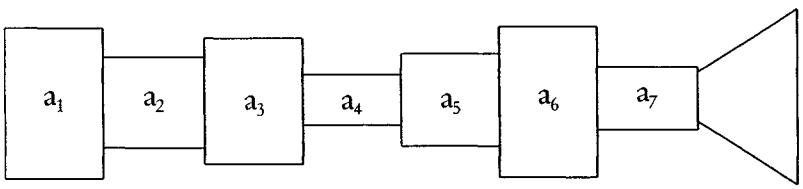

Figure 2.3. LPC coefficients a_1 to a_7 used to model the human vocal tract

LPC will estimate the formants by solving a difference equation that links each sample of the speech signal/waveform to a linear combination of previous samples (hence the term linear prediction). These coefficients are then used to inverse filter the speech signal, relieving any speech harmonics and exposing a pure source where frequency and intensity can be extracted and encoded into the final coefficients. This is accomplished using the autocorrelation method along with the Levinson-Durbin recursion algorithm to optimize the coefficients and minimize mean square error. From these LPC parameters, cepstral coefficients (LPCC) can be further derived recursively [10] [13] as follows (also shown in Figure 2.4)

$$c_1 = a_1 \tag{2.2}$$

$$cn = an + \sum_{m=1}^{n-1} \frac{m}{n} cm\, an - m \qquad 2 \le n \le P \tag{2.3}$$

$$cn = \sum_{m=1}^{n-1} \frac{m}{n} cm\, an - m \qquad n \ge P \tag{2.4}$$

where c_i are the LPCC coefficients and a_i are LPC coefficients. P represents LPC order and n represents the number of LPCC coefficients required, also referred to as the order.

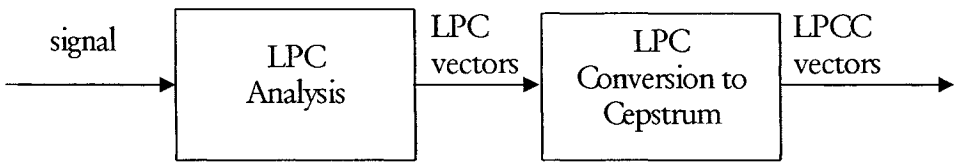

Figure 2.4. Translation process from speech samples to LPCC vectors

2.3.2 Mel-Frequency Cepstral Coefficients

MFCC features are extracted in the cepstral domain. The frequencies of the speech signal are warped to an empirically determined Mel scale that mimics the human auditory response. The objective here is to place more emphasis on the frequency bands of the spectrum that contain highest speech energy, such that the parameters are representative of speech properties. It is this weighting of frequency bands that allows the technique to be more resilient to non-speech noise or noise that exists outside the speech band. The Mel-frequency band centers and widths are designed to be linearly distributed between 0-1000 Hz and logarithmically distributed over 1000 Hz [11][16][17]. Each Mel-band can typically be isolated using a series of triangular filters of specified bandwidths at given center frequencies. From each Mel-band, a single coefficient is determined, therefore the number of filters used is equivalent to the MFCC order required.

The relationship between the frequency and Mel scale is given by:

$$\mathbf{Mel}(f) = 2595\log(1+\frac{f}{700}) \qquad (2.5)$$

When constructing the filter bank, a frequency range to be covered is chosen and defined by f_{min} and f_{max}. These are then warped to the Mel scale using (2.5) to obtain $melf_{min}$ and $melf_{max}$. Therefore, given an analysis frequency range and the required order of MFCC denoted K, a set of overlapping triangular filter banks with band centers corresponding to the Mel-scale can be calculated, as follows [18]:

$$melfcenter_i = 700\left[-1+10^{\frac{(melfmin+\frac{(melfmax-melfmin)*i}{K+1})}{2595}}\right] \qquad i=1,\ldots,K \qquad (2.6)$$

The triangular filter bandwidths are calculated such that for each $melfcenter_i$ the bandwidth stretches from $melfcenter_{i-1}$ to $melfcenter_{i+1}$. The style of overlapping results in each frequency point being mapped to two different Mel-bands.

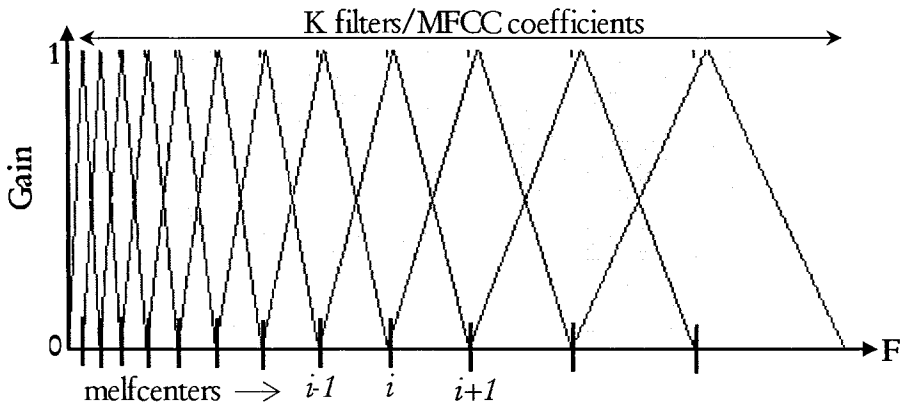

Figure 2.5. Mel-scale filter banks

Based on the filter bank given in Figure 2.5, the MFCC coefficients can be calculated by following these steps:

1. Convert the signal to the FFT domain
2. Find the log of the energy spectrum
3. Apply the filter-bank and gains given in Figure 2.5
4. For each filter (i.e. Mel-band or critical frequency), take the sum of all values in the frame. This results in a single parameter to be calculated per critical band.
5. Apply the discrete cosine transform across all the parameters to obtain the final MFCC feature vectors of order K.

2.4 User Models and Vector Matching

Now that we have an efficient and representative encoding of the speech waveform, the system can create a set of user models to incorporate into its database. Based on these models, any test utterances can be processed in a similar way and compared against these stored values to determine a figure of similarity. This is where the actual recognition of the speaker takes place. For user modeling, a common approach is to use Gaussian mixture modeling [12]. First the feature vectors are quantized to a set of clusters and cluster centers called codebooks. Each codebook is then modeled by a set of Gaussian functions as described in section 2.4.2.

2.4.1 Linde, Buzo, Gray (LBG) Clustering Algorithm

Given a set of feature vectors, it is required that these be represented by a model. As a pre-modeling step, a collection of feature vectors needs to be grouped into clusters. The LBG algorithm is a popular vector quantization scheme amongst ASR engines, due to its non-variational properties involving no differentiation. This allows the algorithm to be trained using discrete data. The discrete training data in this case would be P-dimensional feature vectors where P represents the feature order. Therefore, a set of N vectors of P-dimensions would produce another set of M codebook entries of P-dimensions, depending on the number of clusters being used for the vector quantization process [19] [20].

The LBG algorithm is an iterative technique that starts by classifying all observed feature vectors to a single cluster with a single codebook. The centroids or means of every cluster are used as the codebook entry. For every iteration, it splits each cluster into two smaller ones and recalculates a new optimal codebook. This implies that the cluster count doubles with every split and requires that the final number of clusters or groups be a power of two. After splitting, the feature vectors are re-assigned to the closest cluster, depending on a Euclidean distance metric. A description of the algorithm is given as follows [19].

1. (Initialization) Start by assigning all feature vectors to a single cluster y_0 and calculate its centroid. Hence, number of current clusters and codebook entries L is equal to 1.
2. Split each cluster into two others. Hence, for a given cluster centroid y_i, replace it with two other temporary centroids given by $y_i + \varepsilon$ and $y_i - \varepsilon$, where ε is an arbitrary and small P-dimensional perturbation vector. Replace L with 2L.
3. This is the classification stage, where each cluster has been split and the feature vectors need to be re-assigned to the new temporary centroids. A feature vector $\vec{x}$ will be classified to cluster C_j, if and only if the centroid of that cluster $\vec{z}_j$ is the closest to the vector from a Euclidian distance d, hence:

$$\vec{x} \in C_j \text{ iff } d(\vec{x},\vec{z}_j) \le d(\vec{x},\vec{z}_k) \qquad \forall j \ne k \tag{2.7}$$

$$\begin{aligned} d(\vec{x},\vec{z}) &= (\vec{x}-\vec{z})^T (\vec{x}-\vec{z}) \\ &= \sum_{i=0}^{P-1}\sum_{j=0}^{P-1}(x_i - z_i)(x_j - z_j) \end{aligned} \tag{2.8}$$

4. After classification of all feature vectors to a temporary centroid, new optimized centroids are calculated, based on the grouped vectors. The new centroids $\vec{z}'$ will be the codebook entries and are a simple mean of all the vectors within that cluster $\vec{x}_j$:

$$\vec{z}'_j = \frac{1}{B_j} \sum_{i=0}^{B_j - 1} \vec{x}_j \qquad (2.9)$$

where B_j is the total number of vectors associated with cluster *j*.

5. Is L = M? (i.e. have we reached the desired clustering depth?). If so, then return current clusters and codebook entries (centroids). If not, then repeat steps 2 through 5.

An example set of training vectors and the results of LBG clustering are shown in Figure 2.6 Figure 2.7 for the two-dimensional case (i.e. second order feature vectors).

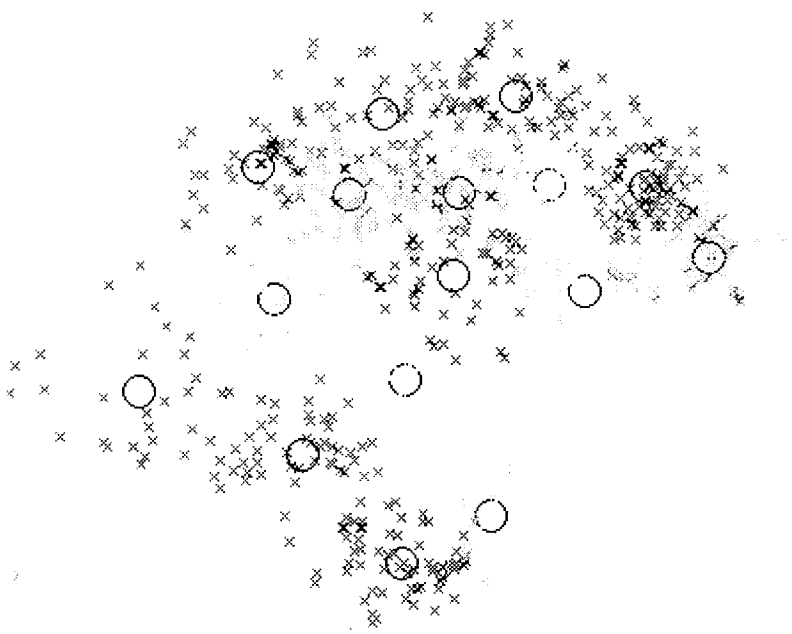

Figure 2.6. LBG clustering using 16 clusters

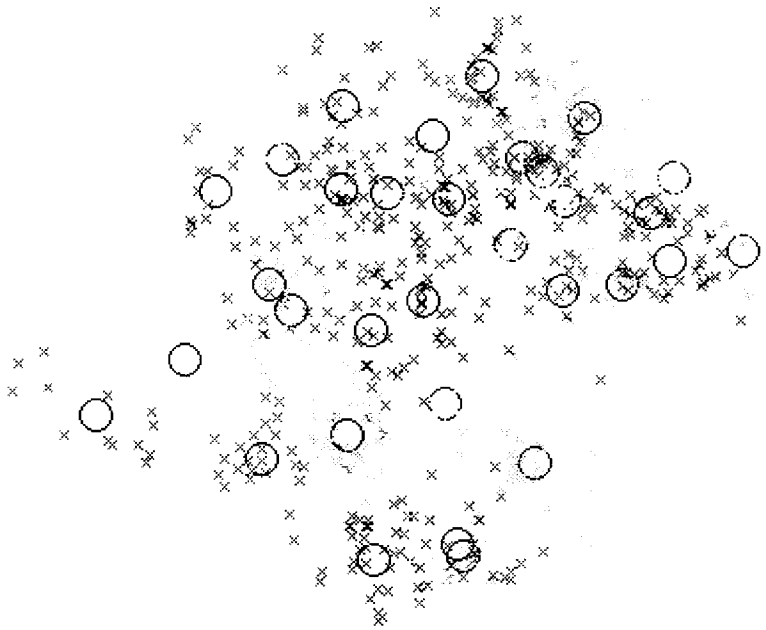

Figure 2.7. LBG clustering using 32 clusters

2.4.2 Gaussian Mixture Models (GMM)

The goal of GMMs is to model the coefficients produced from the feature extraction process using a set of multimodal PDFs. It makes use of the clustering provided by the LBG algorithm described in section 2.4.1. For each cluster of feature vectors, a weighted sum of Gaussian densities is derived to model that specific set. This linear combination of Gaussian densities effectively forms a Gaussian *mixture*. Since each mixture is a collection of Gaussian distribution functions, it can be represented by a mean value (μ), a covariance matrix (Σ) and a mixture weight (p). Hence, each speaker can be modeled by a set of mixtures (M) and is referred to by his/her model λ.

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\} \qquad i = 1, ..., M \tag{2.10}$$

The probability of a single feature vector $\vec{x}$ given a user model λ is calculated by [12]

$$p(\vec{x}\,|\,\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \tag{2.11}$$

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}\,|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)'\Sigma i^{-1}(\vec{x}-\vec{\mu})\right\} \tag{2.12}$$

where D is the dimension of $\vec{x}$ or the order of the feature vectors used.

Equations (2.10), (2.11) and (2.12), coupled with the LBG clustering algorithm, provide us with an initial model of the user, however, the goal of Gaussian mixture modeling is to train a speaker model which best matches the distribution of the training feature vectors. In other words, optimum GMM parameters must be chosen such that the likelihood of the GMM given the training data is maximized. This is where the expectation maximization or EM algorithm is employed to recursively estimate a new model λ_{i+1}, given the current model λ_i, such that EM assures a monotonic increase in the model's likelihood value such that $p(X\,|\,\lambda_{i+1}) \geq p(X\,|\,\lambda_i)$. For each iteration of the algorithm, the new model parameters are calculated using the following equations, where $\bar{p}_i, \bar{\vec{\mu}}_i, \bar{\sigma}_i$ are parameters of the new model λ_{i+1} and $p_i, \vec{\mu}_i, \sigma_i$ are parameters of the current model λ_i [12]

$$\vec{p}_i = \frac{1}{T}\sum_{t=1}^{T} p(i \mid \vec{x}_i, \lambda) \tag{2.13}$$

$$p(i \mid \vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{M} p_k b_k(\vec{x}_t)} \tag{2.14}$$

$$\vec{\bar{\mu}}_i = \frac{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)\vec{x}_t}{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)} \tag{2.15}$$

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)x_t^2}{\sum_{t=1}^{T} p(i \mid \vec{x}_t, \lambda)} - \bar{\mu}_i^2 \tag{2.16}$$

where T represents the total number of feature vectors or frames.

When testing, the input (claimant) speech is being compared or matched to the user models trained above. Therefore, each feature vector of the test speech is compared to the mixture models of a given user U, modeled by λ_U and the likelihood that it belongs (or matches well) with one of those mixtures is calculated. Given S number of users/models in a database, the objective of speaker identification is to find the matching with the highest likelihood [12]

$$\hat{S} = \underset{i=1\ldots S}{\mathbf{arg\,max}} \sum_{t=1}^{T} \mathbf{log}\, p(\vec{x}_t \mid \lambda_k) \tag{2.17}$$

where $p(\vec{x}_t \mid \lambda_k)$ is given by (2.11).

CHAPTER 3
SPEECH ENHANCEMENT

3.1 Enhancement Procedures

ASR systems work very well within controlled environments that are relatively noise free. However, in the presence of background noise, the recognition performance of the system degrades greatly, especially at low signal to noise ratios (SNR). A noise-robust recognition system can be achieved by i) using *speech enhancement* in the speech pre-processing stage to reduce as much noise as possible and form an estimate of clean speech; ii) using *robust feature extraction* in the speech coding stage, where different methods are used to extract information from speech that would be more resilient to the presence of noise; and finally, iii) using *speech modeling* in the user modeling stage where input speech is referenced to speaker models trained in the presence of background noise [1].

In the pre-processing stage, noise is reduced directly at the source where the raw speech is processed and altered such that the resulting recognition accuracy of the system is improved. This does not necessarily require that the SNR of the raw speech be increased, but rather that the claimant to user matching produces a greater number of correct identifications. Since we are dealing directly with the speech waveform, this approach requires that some form of a noise estimate is performed such that its effects

can be suppressed. Enhancement in the speech coding phase aims at minimizing the effects of noise by using a feature extraction scheme that is more robust to noise. This is typically accomplished by using feature coding techniques that closely model human auditory parameters. In the user modeling block, the effects of noise are reduced by modeling the noisy speech. Therefore, the system database is trained using the noisy medium. The primary concern of this research is to observe how different speech enhancement techniques will perform such that an ASR system can be upgraded and made more robust to noise with a simple pre-processing operation. Five techniques were chosen for further analysis, based on their ability to offer good SNR improvement as reported in the literature. They cover a broad range of analysis operations from spectral subtraction to wavelet transform to voice activity detection and are:

1. Adaptive two-pass quantile noise estimation and Wiener filtering
2. Perceptual wavelet based noise estimation and thresholding
3. Minimum mean square error with two-dimensional spectral enhancement
4. Voice activity detector based noise estimation
5. A serial combination of the first two algorithms.

3.2 Adaptive Two-Pass Quantile Noise Estimation

The two-pass quantile noise estimation algorithm uses an adaptive technique to scan through the short-time frequency coefficients of speech and track noise, based on quantile analysis [21]. The result is a noise estimate based on instantaneous SNR

representing speech and non-speech coefficients in both the time and frequency dimensions. At this point, the spectrogram of a signal is the magnitude squared (power) of its Short Time Discrete Fourier Transform (STDFT) given as:

$$STDFT_{l,n}[s(t)] = S[l,n] = \sum_{k=0}^{N-1} s[k+nW]w[k]e^{-j\left(\frac{2\pi}{N}\right)kl} \tag{3.1}$$

$$spectrogram(S[l,n]) = P_S[l,n] = |S[l,n]|^2 \tag{3.2}$$

where l represents the frequency sub-band, n is the frame index, N denotes the size of the analysis frame, W is the frame step (which is 50% in this work) and w is the window function used (square window in this work). Hence, a waveform is broken down into 50% overlapping frames of size N and a $2l$ point FFT is applied to each fame.

3.2.1 Quantile Processing

By definition, the quantile of a dataset $\{x_i, i = 0,...,M\}$ is calculated by first sorting it in ascending order such that:

$$x_0 \leq x_1 \leq ... \leq x_M \tag{3.3}$$

A quantile data point x_q is then determined by:

$$x_q = x_{\mathbf{int}(qM)} \tag{3.4}$$

Where int(...) refers to integer rounding, q being the quantile value or quantile ratio and N is the length of the data set. A q-value of zero returns the minimum of the data set while a q-value of one returns the maximum. Applying this procedure to the

spectrogram of a speech signal allows us to estimate the noise coefficients of a current frame, based on previous frames. Figure 3.1 below illustrates this process.

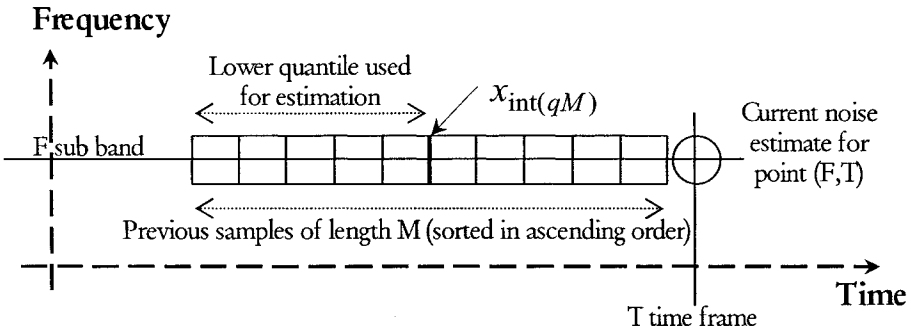

Figure 3.1. The Quantile Estimation process

Since we assume additive and uncorrelated noise as in (1.1), we can conclude that at a specified time-frequency point, the power coefficient is a sum of noise and speech powers. Furthermore, we are assuming that the noise will remain stationary over a specific number of time frames (i.e. within the time analysis window M). Therefore, after sorting a finite time frame of spectral coefficients, the lower quantile is used as an estimate of noise power in which an optimal q-value must be chosen. A typical q-value is the median quantile or q=0.5, based on the assumption that for a certain time window, half of the duration will contain speech [21][22]. However, experimental results given in [24] prove that this may result in over estimation of noise, as it was found empirically that given a 600 ms time window, the probability of having silence more than 20% of the duration is approximately 85%. This indicates that silence will persist for approximately 20% of the time segment. Hence a more conservative q-value of 0.2 is suggested in [21].

3.2.2 Adaptive Two-Pass Quantile Algorithm

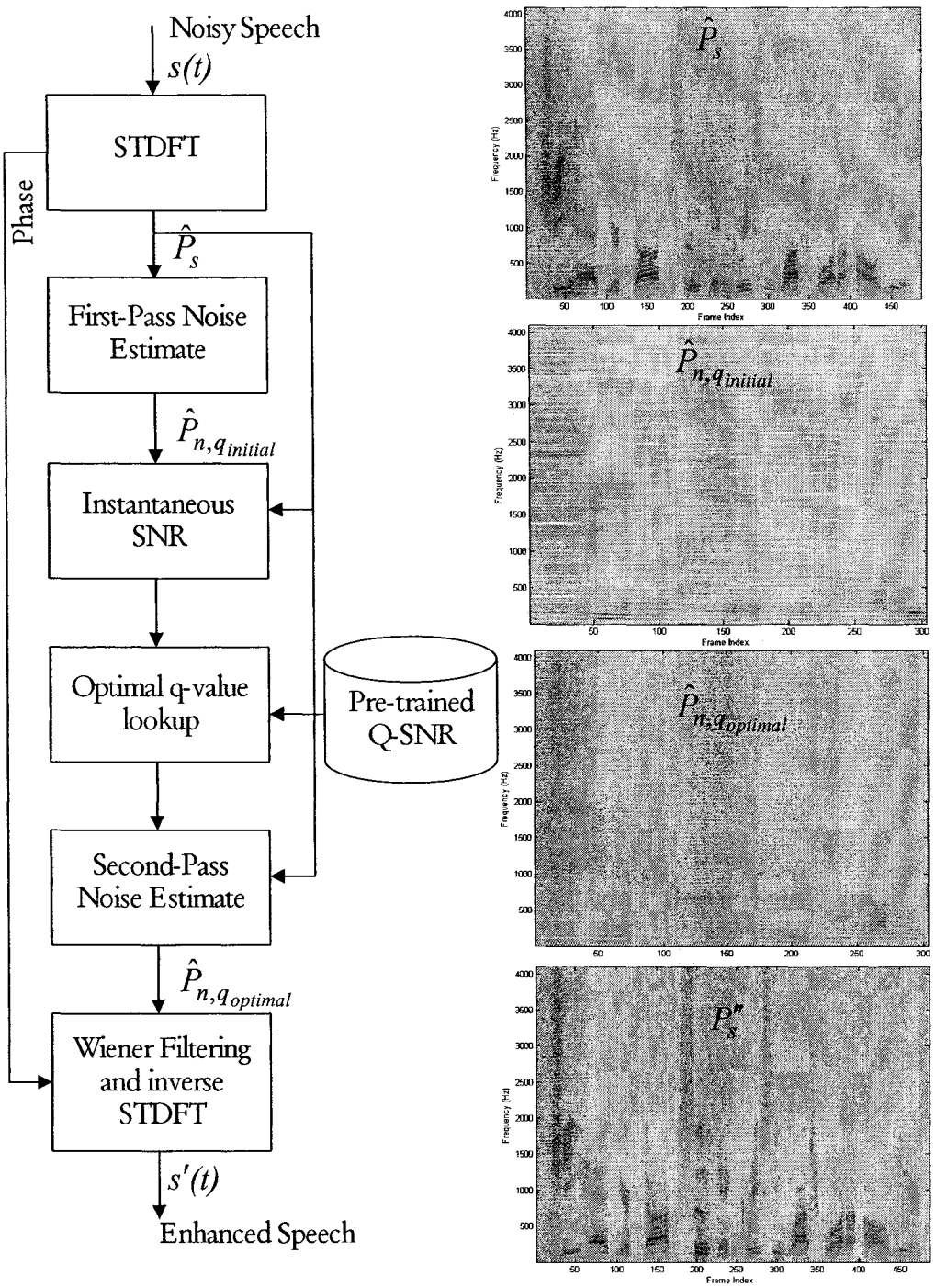

Figure 3.2. Outline of the Two-Pass Quantile Algorithm

The adaptive two-pass algorithm as shown in Figure 3.2 recognizes that the optimal q-value is not a constant, but rather is dependent on the local SNR of a specific time-frequency point (or spectrogram coefficient). This is further derived from the realization that speech activity is not consistent across all analysis windows. Choosing the quantile to associate with noise within an analysis window is therefore dependent on speech activity in this time frame. This dependency is built in to the Q-SNR table training process presented in section 3.2.3. To reduce the effects of outliers, smoothing is achieved on the noise estimate by computing an arithmetic mean of the lower (noise) quantile. The steps are as follows:

1. A normalized spectrogram is computed from the STDFT as per (3.1), (3.2) and the following:

$$\hat{P}_S[l,n] = \frac{\alpha}{L_{win}} P_S[l,n] \qquad (3.5)$$

where $L_{win} = 512$ corresponds to a 32 ms STDFT analysis window and $\alpha = 2.54$ is an empirically determined correction constant used to compensate for noise under-estimation due to the normalization process.

2. Now that we have an estimate of the power spectrum $\hat{P}_S$, we proceed to apply quantile processing (as shown in Figure 3.1) to every time-frequency point to achieve our initial noise estimate $P_{n,q_{initial}}$ as follows.

a. Define a time analysis window $M = 38$ corresponding to 600 ms duration of speech.

b. For each coefficient of $\hat{P}_S[l,n]$, define a set of previous coefficients

$$C[l,n] = \left\{\hat{P}_S[l,n-M+1],\ldots,\hat{P}_S[l,n]\right\} \tag{3.6}$$

that are sorted in ascending order such that

$$\hat{P}_S[l,0] \le \hat{P}_S[l,1] \le \hat{P}_S[l,2] \le \ldots \le \hat{P}_S[l,M-1] \tag{3.7}$$

c. During a 600 ms analysis window, it is expected that noise will be dominant in 20% of the segment, hence use an initial q-value of 0.2 and calculate an initial noise estimate as:

$$\hat{P}_{n,q_{initial}}[l,n] = \frac{1}{\mathbf{int}(qM)+1} \sum_{i=0}^{\mathbf{int}(qM)} C[l,i] \tag{3.8}$$

3. Based on this initial conservative estimation of noise, calculate the instantaneous localized SNR in decibels as:

$$SNR_{inst}[l,n] = 10\log_{10}\left(\frac{\hat{P}_S[l,n]}{\hat{P}_{n,q_{initial}}[l,n]}\right) \tag{3.9}$$

4. This instantaneous SNR will then be used to search through a Q-SNR table and look up a new q-value that is optimal, given this SNR. This table is populated during the training phase of the system. During training, the clean signal will be

known to the system and hence the Q-SNR table can be formed such that the q-value returned gives the greatest match between the clean and noisy signal coefficients. Details of how this table is populated are provided in section 3.2.3.

5. Repeat step 2 using the new optimal q-value to obtain a better noise estimate.

$$\hat{P}_{n,q_{optimal}}[l,n] = \frac{1}{\mathbf{int}(q_{optimal}M)+1} \sum_{i=0}^{\mathbf{int}(q_{optimal}M)} C[l,i] \qquad (3.10)$$

Now that an estimate of the noise is available, it can be removed from the corrupted signal through several options. Direct spectral subtraction can be used, however, it does induce residual musical noise. To avoid this, the authors in [21] used a filtering scheme similar to the Ephraim-Malah filter. However, in this research a noise removal scheme based on a Wiener filter is used and outlined in section 5.2. This results in a recovered spectrogram represented by $P_S''[l,n]$.

6. Finally, we reconstruct the enhanced spectrogram by taking the inverse STDFT, using the same phase as the noisy signal.

3.2.3 The Q-SNR Table

Also known as the Q-SNR map, this is simply a lookup table created during the training phase of the system, where both the clean and noisy speech is known. It defines a set of instantaneous SNRs and the corresponding q-value to achieve the most accurate estimate of noise. The table can be populated in one of two procedures depending on the scenario presented. If the noise model is known during training, a better q-value is chosen such that the new estimate of the noise will be as close as possible to the actual noise given and is calculate using:

$$\left|\hat{P}_{n,q}[l,n] - P_{n_{actual}}[l,n]\right| = 0 \qquad (3.11)$$

$\hat{P}_{n,q}[l,n]$ is determined by (3.8). However, in practice, the noisy medium is treated as a black box where little or no information is known about the noise. A better q-value is then chosen such that it will yield a processed coefficient that is closest to the clean signal given by:

$$\left|P_S''[l,n] - P_{x_{clean}}[l,n]\right| = 0 \qquad (3.12)$$

$P_S''[l,n]$ is the de-noised signal. Effectively, the change in q-value with respect to sub-band SNR is captured while training a Q-SNR map. As most of the speech exists only in a sub-set of frequencies, it would be too general to train one map for the entire spectrogram. Ideally, it would be desirable to train a separate Q-SNR map for every

frequency sub-band, however, depending on the FFT resolution used, this can very quickly lead to computational and storage issues. Instead, a table is created for each critical sub-band, where the optimizations of (3.11) or (3.12) are applied at each time frequency point. Optimal q-values are grouped into corresponding SNR_{inst} frequency bins and the average taken per bin. The Bark scale is a set of frequency bands and centers designed to target psycho-acoustical differences in human hearing. Generally speaking, frequencies within each Bark band are difficult to distinguish, yet tones in different bands are discernible to the human ear. There are 24 Bark bands empirically defined, up to a frequency of 15.5 kHz as shown in Table 3.1 [23]

Critical Band	Bark Centers (Hz)	Bark Frequency Bands (Hz)
1	50	0 - 99
2	150	100 - 199
3	250	200 - 299
4	350	300 - 399
5	450	400 - 509
6	570	510 - 629
7	700	630 - 769
8	840	770 - 919
9	1000	920 - 1079
10	1170	1080 - 1269
11	1370	1270 - 1479
12	1600	1480 - 1719
13	1850	1720 - 1999
14	2150	2000 - 2319
15	2500	2320 - 2699
16	2900	2700 - 3149
17	3400	3150 - 3699
18	4000	3700 - 4399
19	4800	4400 - 5299
20	5800	5300 - 6399
21	7000	6400 - 7699
22	8500	7700 - 9499
23	10500	9500 - 11999
24	13500	12000 – 15500

Table 3.1. Critical frequency bands used by the Bark scale

In this research, speech was sampled at 16 kHz giving an 8 kHz bandwidth, hence only 22 of these 24 critical bands were used when training the Q-SNR table. The final result is a three dimensional matrix consisting of instantaneous SNRs matched with their optimized q-values for every critical sub-band, as shown in Figure 3.3. Note how the maximum q-value alters as frequency sub-band increases. This is due to the fact that at lower sub-bands, speech energy is high and therefore a less aggressive (i.e. lower) q-value is chosen to reduce noise over-estimation minimizing speech distortion. At higher frequencies, noise power is dominant and a larger, more aggressive q-value is used.

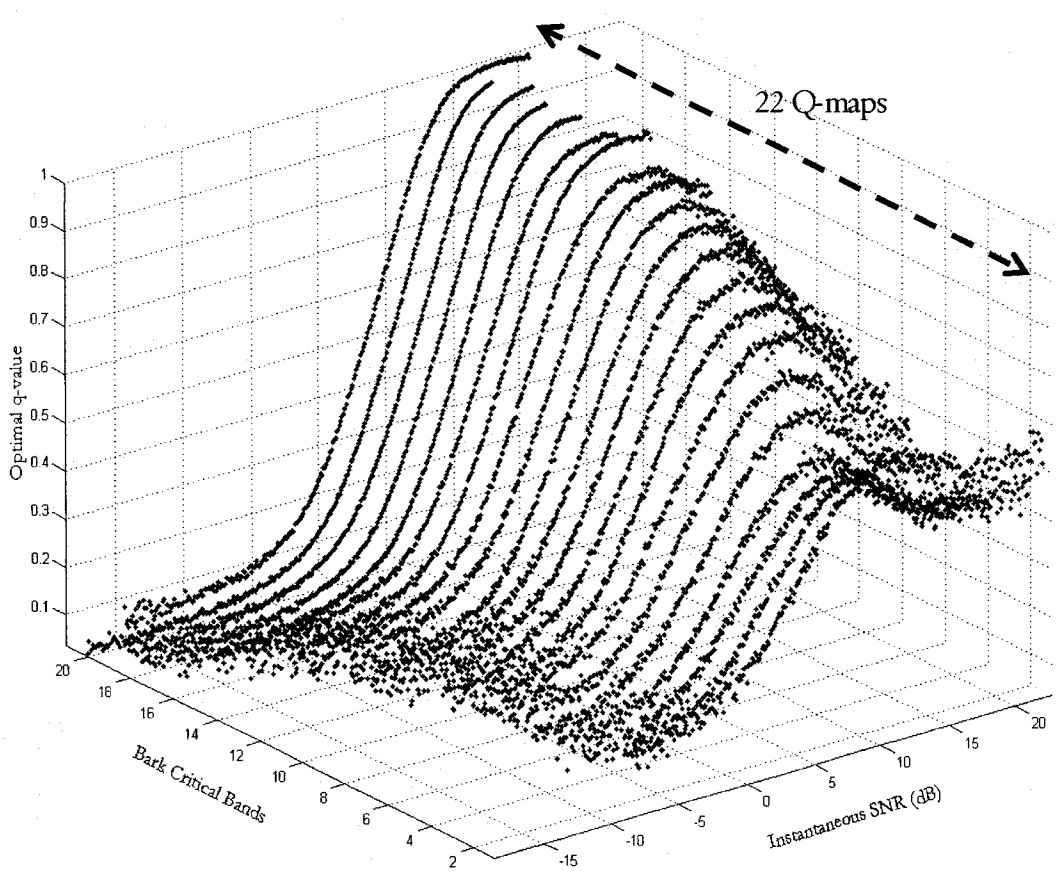

Figure 3.3. Sample Q-SNR tables computed for 22 Bark bands

3.3 Perceptual Wavelet Adaptive De-noising (PWAD)

The motivation behind wavelet-based enhancement lies in the fact that wavelet coefficients of noise only signals are sparse or close to zero [26],[27]. This is beneficial since it allows for noisy coefficients to be minimized with less risk of speech distortion. Also, noise energy in the wavelet domain is spread equally across all coefficients while speech is concentrated in only a few. Noise is assumed to be additive and independent of the speech and given that the wavelet transform is linear, we represent the observed noisy signal as:

$$s(t) = x(t) + n(t) \tag{3.13}$$

$$S(\omega, n) = X(\omega, n) + N(\omega, n) \tag{3.14}$$

where $X(\omega, n)$ is the wavelet transform of $x(t)$. Wavelet de-noising is a non-parametric approach to noise estimation, hence making no assumptions of the type of noise being added. It determines the standard deviation of the additive noise based on a modified quantile technique. Given the estimated noise model, a wavelet threshold is calculated based on a wavelet shrinkage technique given in [26]. PWAD enhancement is a three step process, where the signal is first transformed into the perceptual wavelet domain that is essentially a wavelet packet transform with a transform tree designed to model the human auditory response. The coefficients are then processed to reduce the effects of noise and finally, the signal is restored to the time domain using the inverse perceptual wavelet transform. An outline of the PWAD system is given in Figure 3.4.

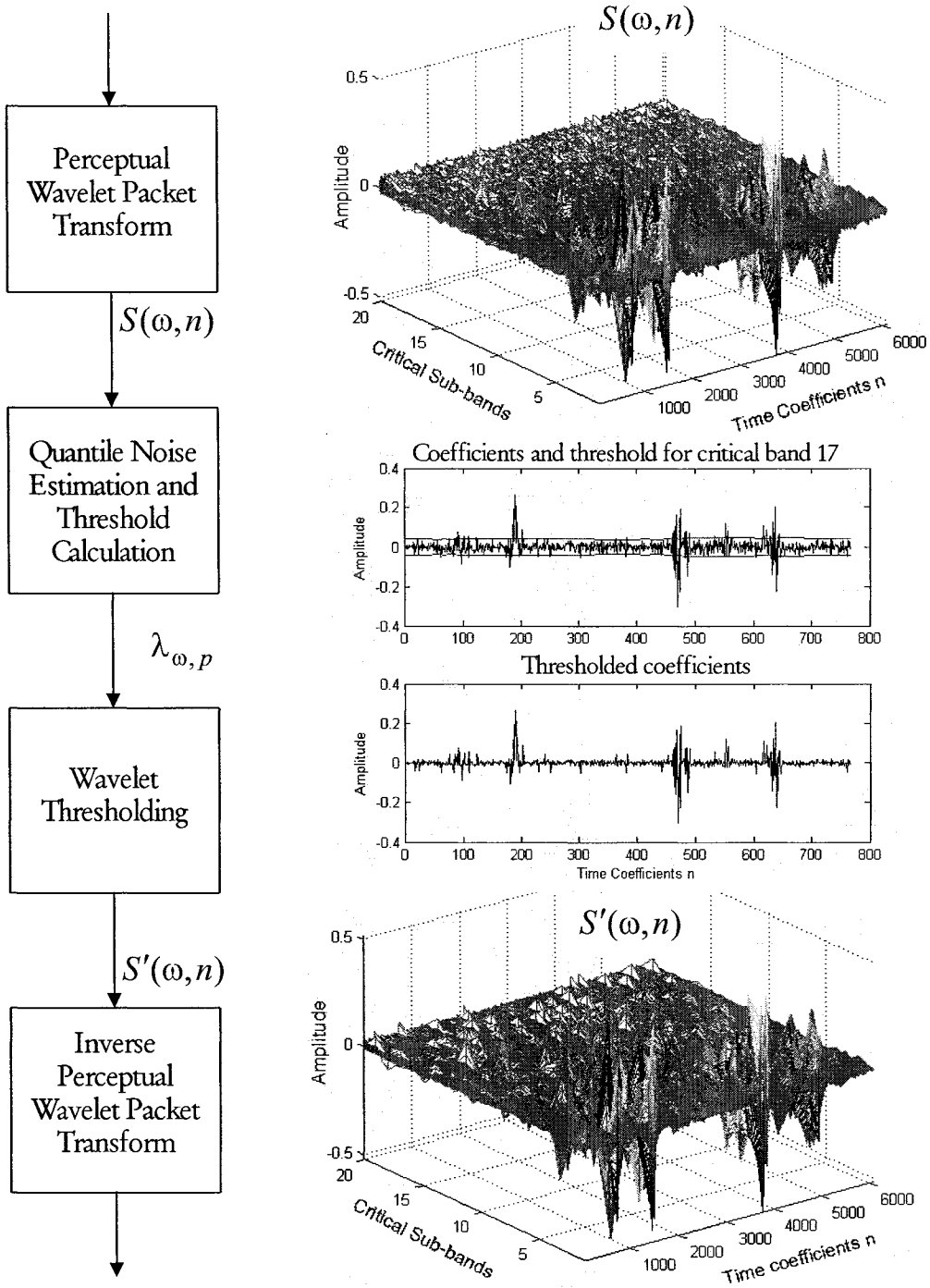

Figure 3.4. Outline of PWAD system implemented

3.3.1 Perceptual Wavelet Transform (PWT)

One of the differences between the two-pass and PWAD algorithms lies in the analysis domain. The STDFT has constant frequency and time resolution depicted by the number of samples considered in the underlying FFT. Hence, the same frequency detail is achieved over all sub-bands of the signal. As speech energy is confined to a specific bandwidth, it would be more beneficial to seek higher resolution within these frequencies to allow for more accurate estimation of noise. This is the goal of the Perceptual Wavelet Transform (PWT). It offers a multi-resolution decomposition of the speech signal into critical frequency sub-bands where higher frequency resolution is placed on speech sub-bands. These critical sub-bands are derived from the standard psychoacoustic model used in MPEG standards, which takes into consideration the masking effects introduced by adjacent frequency components. The theory here is that a frequency coefficient with high power will have an overshadowing effect on its lower energy neighbors, rendering them partially or completely inaudible [26], [28], [29]. Although the main application of this is in audio coding, where redundant data are discarded, it is also used to form the critical sub-bands of the psychoacoustic model. This results in a set of wavelet packets designed to mimic human auditory bands and is given by the wavelet packet tree in Figure 3.5 [30]. The wavelet packets are designed to mimic human auditory critical bands and are derived from 16-tap Daubechies FIR filters. The authors in [30] use an 8-stage tree for their implementation of the PWT. They require that much resolution for their wide-band coding scheme. Since we are concerned with noise removal, as opposed to

waveform coding, a 6-stage tree offers us enough critical band resolution as illustrated in Figure 3.5.

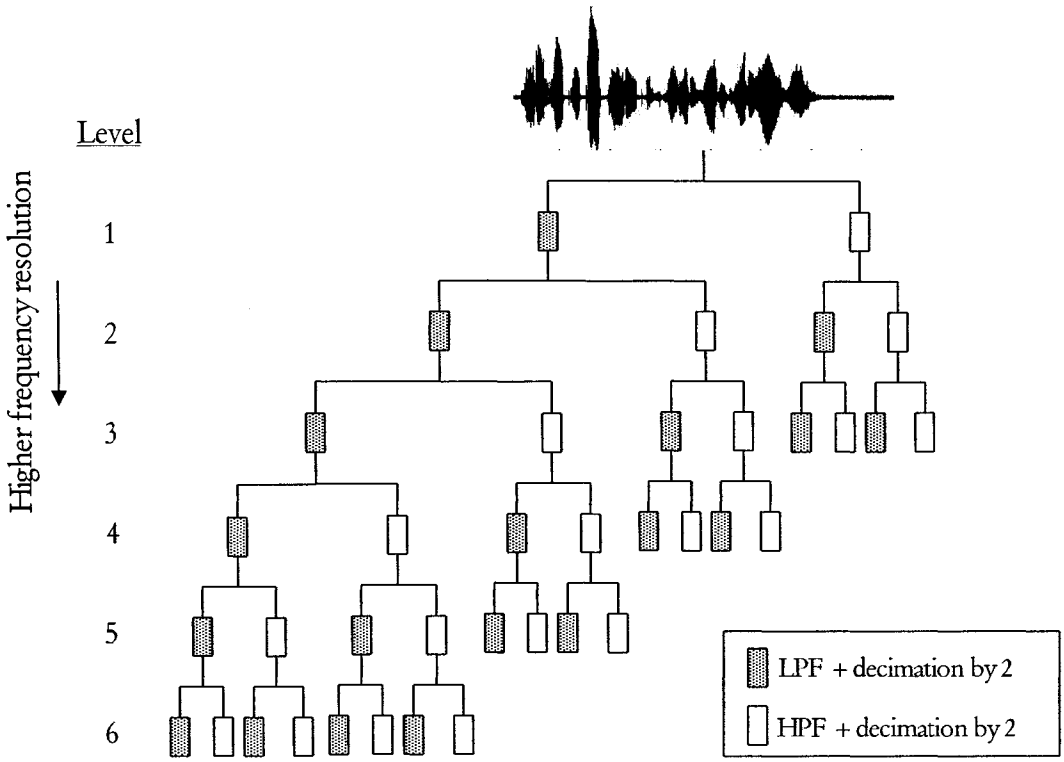

Figure 3.5. Wavelet Packet Tree used for PWT decomposition

3.3.2 Quantile Based Noise Estimation and Threshold Calculation

The purpose here is to exploit the inherent time-frequency multi-rate decomposition properties of wavelet packets, where speech coefficients will be significantly larger in magnitude than noise ones [27]. The quantile noise tracking system is modified to be applicable to the wavelet domain by first segmenting each sub-band into frames and into corresponding segments. For each frame, a standard deviation is

assigned to it by considering the q=0.2 quantile of the corresponding segment. The quantile noise estimation process in the perceptual wavelet domain is given in Figure 3.6.

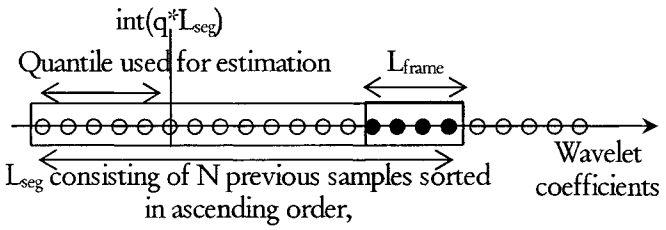

Figure 3.6. Wavelet adapted quantile estimation

Therefore each critical sub-band in $S(\omega, n)$ is segmented into time frames of length L_{frame} and a frame shift of L_{shift}. For each frame p in the sub-band, a noise estimate is associated to it, based on a quantile estimate of the previous samples encompassed in time window L_{seg}. The noise estimate is then given as:

$$\sigma_{\omega,p} = \beta \sum_{j=0}^{\mathrm{int}(qL_{seg}^{\omega})} \frac{\hat{S}(\omega, p)}{L_{seg}^{\omega}} \qquad (3.15)$$

where β and q are empirically defined as 0.38 and 0.3 respectively. The corresponding time lengths L_{seg}, L_{frame} and L_{shift} are defined as 512 ms, 64 ms and 32 ms respectively, based on the same statistical properties of speech presented in section 3.2.1. Note that due to the multi-resolution property of the PWT, each critical band will have a different number of wavelet coefficients. Hence, the number of samples representing each time window differs for each sub-band (symbolized by L_{seg}^{ω}). Based on this estimate, a

wavelet threshold is calculated for each sub-band at the p-th frame, based on the nearly optimal universal wavelet packet de-noising threshold defined by [26] [26] [30] as:

$$\lambda(\omega, p) = \sigma_{\omega,p}\sqrt{2\log(L_{seg}^{\omega}\log_2(L_{seg}^{\omega}))} \qquad (3.16)$$

3.3.3 Wavelet Thresholding and Inverse PWT

Noise removal can be achieved using hard or soft thresholding [26]. Hard thresholding simply sets all coefficients below a defined threshold to zero. This is known to cause speech distortion, since some speech coefficients may briefly fall under the threshold before picking up. Soft thresholding, which is used in wavelet shrinking, attempts to alleviate this by setting all coefficients below the threshold to zero and the others are shrunk by the threshold amount, hence preserving the continuity of the speech coefficients [26] [27]. Although hard and soft thresholding yield obvious visual (and SNR) enhancement to the coefficients, they do fail to provide recognition accuracy improvements, since they are ultimately attenuating on a per coefficient basis. Therefore, this research applies a modified wavelet thresholding technique adapted from the work of Ephraim and Malah to return a coefficient gain $H(\omega,n)$ based on this threshold and given in section 5.3. The enhanced wavelet coefficients are then calculated as:

$$S'(\omega,n) = H(\omega,n)S(\omega,n) \qquad (3.17)$$

The final recovered waveform is then obtained using the inverse perceptual wavelet transform, which is simply a backwards traversal up the perceptual wavelet packet tree, given in Figure 3.5.

3.4 Two-Dimensional Spectrogram Enhancement

The two-dimensional spectrogram enhancement technique classifies and reduces musical noise that is a byproduct of the noise removal process based on spectral subtraction [32]. As mentioned earlier, the main drawback of direct spectral subtraction is musical noise. This is where coefficients that become negative due to the subtraction process are set to zero. Hence this technique is designed to run as a post-processor to a spectral subtraction algorithm. The authors of [32] use a Minimum Mean Square Error Short Time Spectral Amplitude estimator (MMSE-STSA) to enhance noisy speech before applying TDSE. The system is a two stage process where first, coefficients are classified as speech and non-speech, depending on the SNR observed from the reference speech signal extracted during the MMSE-STSA process. Second, the non-speech coefficients are suppressed using time averaged filters. Hence, the enhanced spectrum is:

$$P_{TDSE} = P_{speech} + P_{non-speech} \tag{3.18}$$

During the classification phase, the observed spectrum from MMSE-STSA is first smoothed using a 2D filter given by:

$$P_{smoothed} = \frac{1}{(2K+1)(2N+1)} \sum_{i=-K}^{K} \sum_{j=-N}^{N} P_{MMSE-STSA}(k+i, n+j) \tag{3.19}$$

A classification matrix $S(k,n)$ is formed to label each coefficient of $P_{smoothed}$ as either speech (1) or non-speech (0), where:

$$\begin{aligned} S(k,n) &= 1 \quad if\ P_{smoothed} \geq \lambda \\ S(k,n) &= 0 \quad if\ P_{smoothed} < \lambda \end{aligned} \tag{3.20}$$

This is essentially a voice activity detector where any smoothed coefficients having a value greater than the defined speech threshold given by (3.23) are classified as speech and all others are classified as non-speech. The speech threshold is given by

$$\lambda = e^{(-0.1887 SNR_{est} - 2.4278)} \tag{3.21}$$

where SNR_{est} is the instantaneous SNR given by the MMSE-STSA noise estimate. Therefore, speech and non-speech spectrogram coefficients are defined as:

$$\begin{aligned} P_{speech}(k,n) &= P_{smoothed}(k,n) S(k,n) \\ P_{non-speech}(k,n) &= P_{smoothed}(k,n)[1 - S(k,n)] \end{aligned} \tag{3.22}$$

The non-speech coefficients containing musical noise are then smoothed over the time axis over 2M+1 coefficients:

$$\hat{P}_{non-speech}(k,n) = \frac{\beta}{2M+1} \sum_{m=-M}^{M} P_{non-speech}(k,n+m) \tag{3.23}$$

The suppressing factor β is determined by:

$$\beta = e^{(0.1256 SNR_{est} - 3.3778)} \tag{3.24}$$

3.5 Voice Activity Detector Based Noise Estimation

Under ideal circumstances, the noise model is best estimated using a portion of the observed noisy signal that contains noise only. With this in mind, this algorithm attempts to estimate noise, based on spectral coefficients determined to be noise only. In essence, the algorithm will apply STDFT based voice activity detection [33] [34] [35] [36] and use that to implement a "noise switch". During non-speech segments, the switch is closed and the noise estimate is continuously being updated to reflect the observed power coefficients, while speech segments will open the switch, stopping noise estimation and preserving the previously estimated noise level. A more descriptive outline of the noise estimation algorithm, along with samples, is given later on in this section. One of the advantages of this algorithm is its VAD, which operates in both the time and frequency dimensions allowing for more accurate isolation of speech and noise power. Another advantage is its optimized speech estimation filter, based on a priori and a posteriori SNRs and minimizing the spectral amplitude errors between the observed and enhanced signal. Again, the assumptions of additive and statistically independent noise apply.

The system is broken down into four stages, as follows. First, the observed signal is transformed into the time-frequency domain using the STDFT. A voice activity detector is applied to label individual coefficients as speech or non-speech. This estimate of speech presence/absence allows for extraction of a noise model. The noise model is then used to determine a power spectrum gain function that will attenuate noise coefficients while preserving speech ones. Figure 3.7 gives and outline of the algorithm.

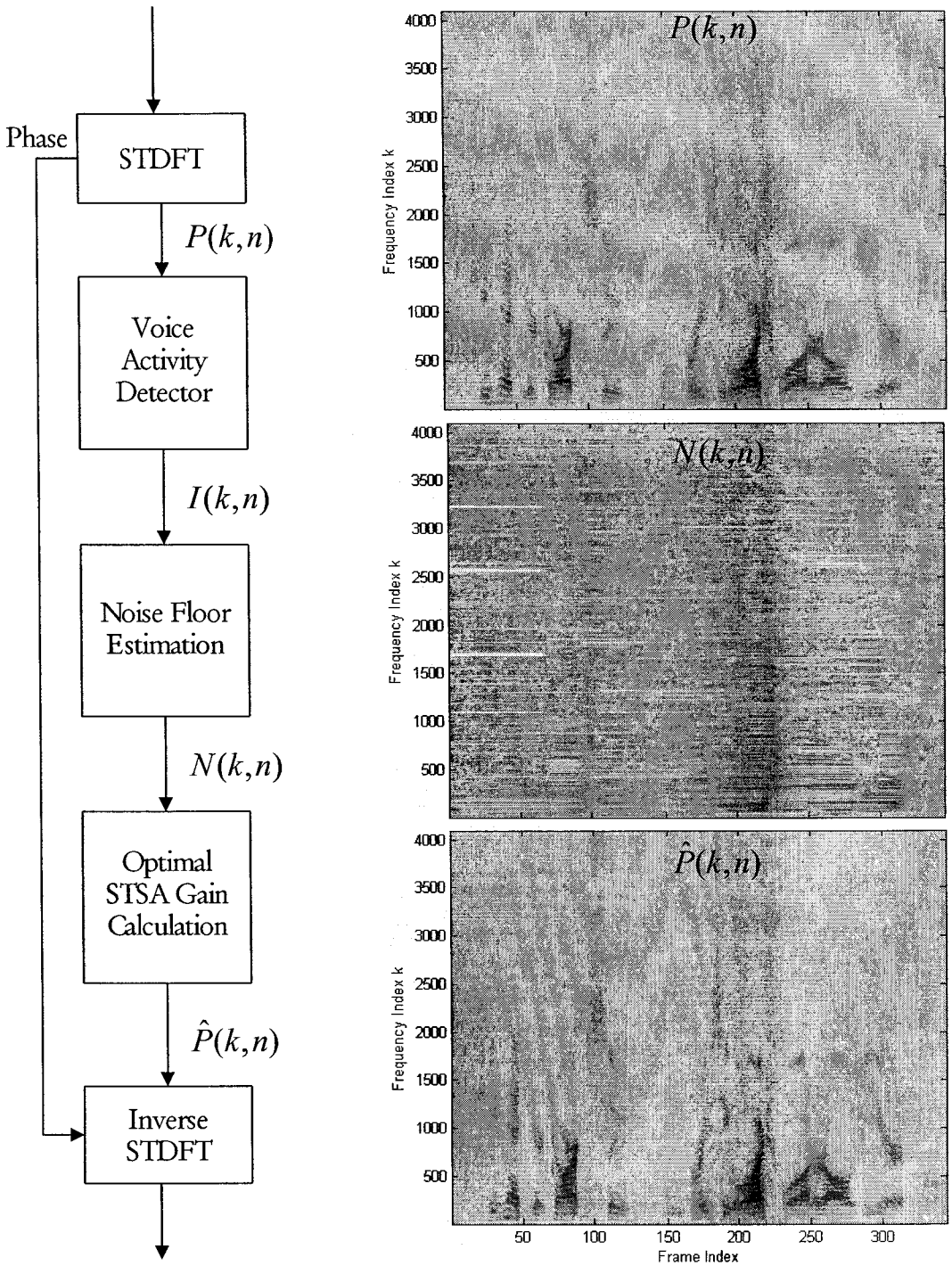

Figure 3.7. Outline of VAD based Minima Noise Estimation

The first and crucial step to noise estimation is the detection of speech and non-speech segments, which is the purpose of this VAD segment. It will average the power spectrum (i.e. magnitude squared of spectrogram) over a specified frequency window first, then over a time window. This process helps to reduce the VAD's sensitivity to outliers within the noise while preserving the relative energy of speech. At this point, spectral noise flooring is applied to obtain an initial estimate of noise, based on the fact that coefficients representing speech and noise will have higher magnitudes than those representing noise only as shown in Figure 3.8 along with the "noise-switch" behavior implemented in this algorithm. Spectral noise flooring is a recursive process where the current noise estimate is chosen as the minima of a set of previous coefficients within each sub-band. This is reminiscent of the quantile noise estimation process defined in section 3.2.1. Indeed, spectral noise flooring is a defined case of quantile processing, where the q-value would be 0, to represent the minimum quantile. As with quantile processing, the set of previous coefficients must stretch long enough to include some silence portions, ensuring the noise floor estimation remains at a minimum. Therefore, the spectral noise floor estimation would be dependent on the length of this set and there is a tradeoff. A longer size would prevent noise over estimation; however, it would hinder it's capability to track non-stationary noise. In our implementation, the selection of this set was based on speech and noise behavior assumptions made in section 3.2.1 and experimentally confirmed. Namely, over a 600 ms window, noise is assumed to be stationary and speech is assumed to be bursty in nature.

The steps of VAD noise estimation are:

1. Calculate the power spectrum of the signal using (3.1) and (3.2). A time window of 32 ms with 50% overlap was used.
2. Average along the frequency axis, with a window size $w = 3$ such that:

$$P_F(k,n) = \frac{1}{2w-1}\sum_{i=-1}^{1} P(k-i,n) \tag{3.25}$$

3. Average along the time axis, with a window size of $B = 2$ such that:

$$P_T(k,n) = \frac{1}{B}\sum_{j=0}^{B-1} P(k,n-j) \tag{3.26}$$

4. Estimate the noise floor (using $C = 38$ or 600 ms):

$$M(k,n) = \min_{c=0...C}\left(P_T(k,n-c)\right) \tag{3.27}$$

5. Establish a speech classification matrix based on localized SNR ($\psi = 20$)

$$I(k,n) = \begin{cases} 1 & \text{if } \dfrac{P_T(k,n)}{M(k,n)} > \psi \\ 0 & \text{otherwise} \end{cases} \tag{3.28}$$

6. The speech presence probability estimator is calculated with control factors $\alpha_P = 0.1$ and $\alpha_\alpha = 0.2$ representing the minimum speech and noise probability respectively. The α_P factor controls how smoothed the noise estimate will be, while α_α controls the level of noise estimation (i.e. a higher value will lead to more noise removal; however, it is subject to noise over-

estimation). We determined both of these values empirically, based on our simulation results.

$$
\begin{aligned}
&\textit{If } I(k,n)=1 \\
&\quad p(k,n)=\alpha_p+(1-\alpha_P)p(k,n-1) \\
&\quad \alpha_N=1 \\
&\textit{else} \\
&\quad p(k,n)=(1-\alpha_P)p(k,n-1) \\
&\quad \alpha_N=\alpha_\alpha+(1-\alpha_\alpha)p(k,n)
\end{aligned}
$$

7. Finally, the noise is estimated using the noise smoothing parameter α_N

$$N(k,n)=\alpha_N N(k,n-1)+(1-\alpha_N)P(k,n) \tag{3.29}$$

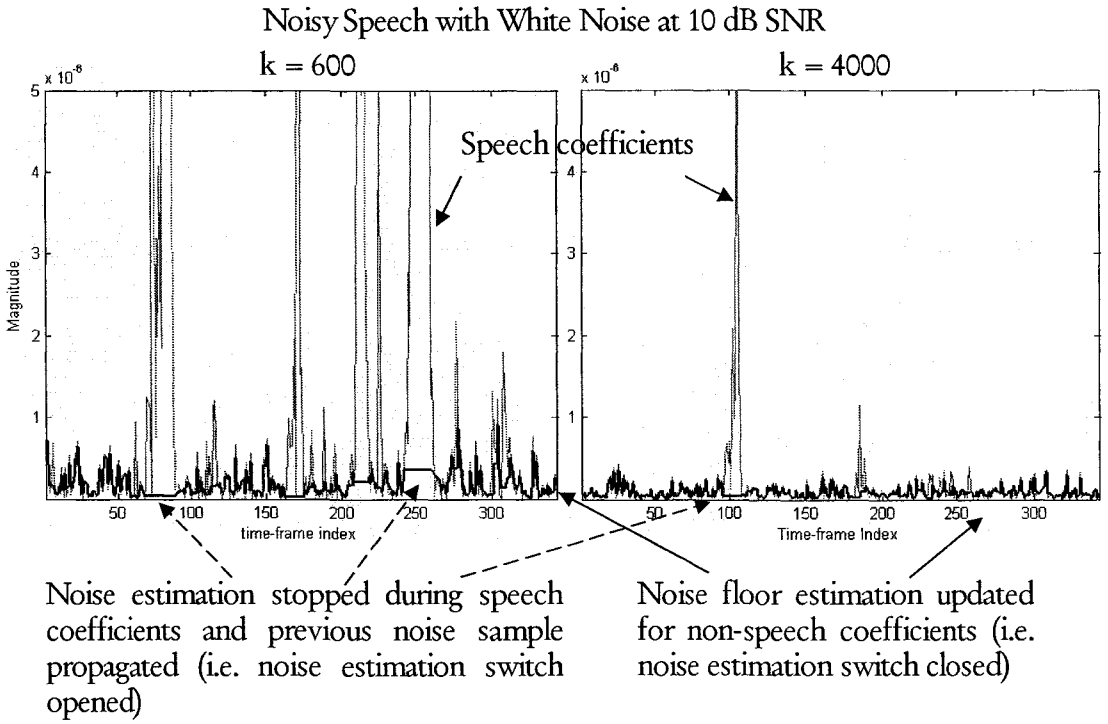

Figure 3.8. Sample power spectra and noise estimation

CHAPTER 4
EXPERIMENTAL SETUP

4.1 Testing and Training

The TIMIT speech corpus was used to train our GMM system. It was originally designed for speech recognition and offers 10 waveforms per user, sampled at 16 kHz. Hence, it is geared towards providing a wide range of phonetic sounds and offers less in total speech length available for training/testing. However, as observed from experimental trials in the results section, the amount of speech provided was enough to allow accurate speaker recognition performance. Specifically, the system was able to achieve a 98% recognition rate with clean speech, suggesting sufficient training speech length. From the corpus, 20 male users, shown in Table 4.1 below, were randomly selected to form our client database.

MESD0	MKCL0	MPAM0	MRKO0
MGJF0	MLIH0	MPAM1	MROA0
MGWT0	MLLL0	MPLB0	MRPC0
MJBR0	MMAB0	MPWM0	MRPP0
MJRF0	MNJM0	MREB0	MSTK0

Table 4.1. List of speakers used from the TIMIT database

Out of the 10 waveforms per user, the three SI type utterances having longer duration with randomly chosen contextual speech were concatenated together to form a

single training speech waveform of approximately 10 seconds. The remaining seven waveforms were used for testing.

For training, the speech waveform was segmented into 20 ms frames with 50% overlap and 24th order feature vectors were extracted from each frame (LPCC or MFCC). The features were then encoded to 16 Gaussian mixtures, to form the GMM for that specific user. While forming the GMMs, the LBG code vector quantization process was run until the total displacement by all 16 centroids was below 1%, indicating that the clusters had reached an optimum spread level or until a maximum iteration number of 10 was reached. It was stated in [12] and [19] that the LBG clustering algorithm would reach a localized minimum in displacement after about 10 iterations.

Testing was performed on the remaining 7 waveforms, where each waveform was passed onto the system as speaker inputs. Therefore, for each waveform, the recognition engine picked one out of 20 male users with the highest matching score. Since male and female frequency characteristics differ, the scores of this research may not be directly applicable to female speech. The test speech was fed into the system as is (i.e. high SNR), as well as with varying levels of background noise from the NOISEX-92 database discussed in section 4.3. Noise was down sampled from 19.98 kHz to 16 kHz to match the bandwidth of the clean speech before addition at SNR levels of 20, 15, 10, 5 and 0 dB, based on total signal to noise power. The test speech signal was then segmented into 20 ms frames with 50% overlap as explained in Chapter 2. Performance metrics were

then calculated. Simulations were run with five pre-processing enhancement setups: Two-pass spectral de-noising, perceptual wavelet de-noising, MMSE-TDSE, VAD based noise estimation and two-pass spectral de-noising followed by wavelet de-noising. The reason for combining the two-pass and wavelet methods is the ability of the wavelet algorithm to perform well at higher SNRs. Therefore, the two-pass algorithm will initially perform some noise removal and the wavelet technique would follow, possibly offering some more improvement.

4.2 Performance Evaluation

To measure and compare the performance of the speech enhancement algorithms used in this research, the average recognition accuracy, average segmental SNR and perceptual evaluation of speech quality (PESQ) were calculated.

4.2.1 Average Recognition Accuracy

Average recognition accuracy is simply an average of all correctly identified test waveforms. For each user, 7 utterances were given to the system as input and the resulting recognition ratio for that user and the system is given by:

$$\text{Ratio}_{\text{user}} = \frac{\text{number of correct identifications}}{\text{total number of test waveforms}}$$

$$\text{Accuracy} = 100 \times \frac{\text{Sum of all Ratio}_{\text{user}}}{\text{total number of users in the system}}$$

4.2.2 Average Segmental SNR

Average segmental SNR is calculated, based on noise extracted from the silence portions of the test speech. To determine such segments, a simple energy based time-domain voice activity detector is applied to the clean speech. Since the clean speech has very little noise due to the high recording quality of the TIMIT database, such a VAD would be accurate. Hence, this calculation requires a clean reference signal, as well as the input signal to analyze, implying that both must be synchronized (i.e. aligned in the time-domain). The reference and input speech are broken down into segments of 10 ms each with no overlap. The frames of the input signal are then classified as speech and non-speech, using an energy threshold applied to the reference frames. A recursive process then cycles through the frames of the input signal chronologically from beginning to end. Frames determined to contain no speech are used to update the noise energy level. For each of the speech frames, a localized signal to noise ratio is calculated, based on this updated noise level. The average segmental SNR is then determined to be the average of these frame-based SNRs. Figure 4.1 illustrates how average segmental SNR can be measured given a noisy speech signal Although this metric is measured in decibels, average segmental SNR and input SNR cannot be directly compared, as they are defined and calculated in different ways. Furthermore, it attempts to reflect the variation of improvements within the speech waveform by averaging them and hence, cannot be compared to SNR improvements as defined by other papers. For this reason, average segmental SNR is reported for both the noisy and enhanced speech in Chapter 5

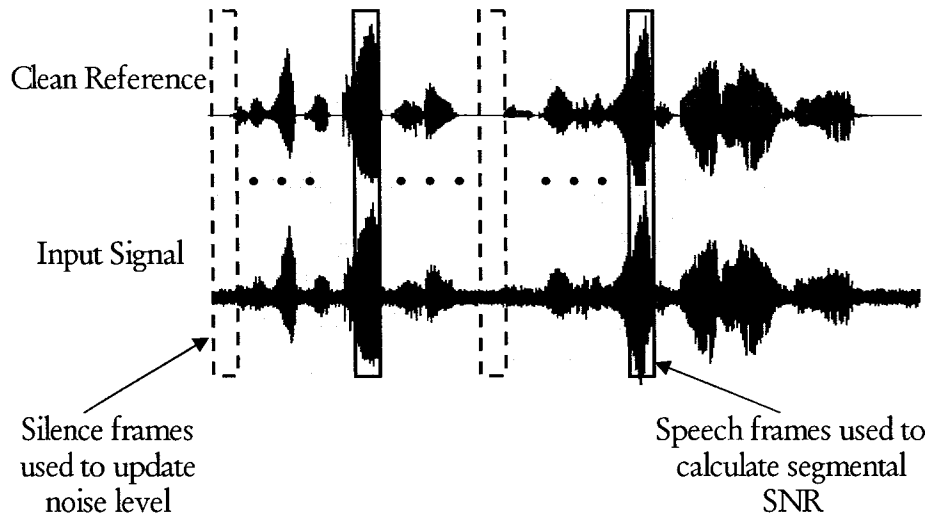

Figure 4.1. Calculation of Average Segmental SNR

4.2.3 Perceptual Evaluation of Speech Quality-Mean Opinion Score (PESQ-MOS)

Designed to monitor end-to-end voice quality of communications systems and popular in the Voice over IP world, PESQ-MOS is an International Telecommunications Union standard (ITU-T P.862). Developed by industry leaders, PESQ-MOS is able to predict subjective quality in a wide range of conditions that may include coding distortions, errors, noise, filtering, delay and variable delay. It returns a score between 0 and 4.5, where a higher value represents closer subjective similarity between the reference (i.e. clean) and input (i.e. noisy or enhanced) signals. The implementation of PESQ-MOS used in this research was provided based on a script resembling the ANSI-C reference code available for use in the P.862 recommendation [37]. It does implement frame based analysis, capturing varying enhancements within the waveform. The stages of the algorithm relevant to analyzing noise enhancement are:

Level Alignment: In order to compare the signals, a gain is applied to the clean reference and input signals to bring their powers down to a consistent range, as given by ITU-T P.830. This makes the system immune to different levels between the reference and input signal.

Auditory Transformation. The clean reference and input signals are each passed through an auditory transform that mimics certain key properties of human hearing. This gives a representation in time and frequency of the perceived loudness of the signal, known as the sensation surface.

Disturbance Processing. The difference between the sensation surfaces for the reference and input shows any audible differences introduced by the enhancement process. The error surface is analyzed by a process that takes account of the effect that small distortions in a signal are inaudible in the presence of loud signals (masking).

4.3 NOISEX-92 Noise Database

Eight noise types were used in the testing phase. These were obtained from the NOISEX-92 database and added to the test waveforms at different SNRs. All noise signals were recorded at a sampling rate of 19.98 kHz with 16 bit A/D conversion. Each noise sample was first down sampled to 16 kHz to match the sampling rate of our speech before adding it to our test waveforms. A list of the noise types and descriptions, as provided by NOISEX, is given below [38].

Noise Type	Description
White	White noise acquired by sampling high-quality analog noise generator (Wandel & Goltermann). Exhibits equal energy per Hz. bandwidth.
Pink	Pink noise acquired by sampling high-quality analog noise generator (Wandel & Goltermann). Exhibits equal energy per 1/3 octave.
HF Channel	Recording of noise in an HF radio channel after demodulation
Volvo	Volvo 340 noise. This recording was made at 120 km/h, in 4th gear, on an asphalt road, in rainy conditions.
Babble	The source of this babble is 100 people speaking in a canteen. The room radius is over two meters; therefore, individual voices are slightly audible.
Factory1	This noise was recorded near plate-cutting and electrical welding equipment.
Machine Gun	The weapon used was a .50 calibre gun fired repeatedly.
F16	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.

Table 4.2. NOISEX-92 noise types used and their characteristics

A time and frequency representation of the above noise types are provided in Figure 4.2 through Figure 4.9. Frequency power was normalized and reported in terms of maximum codec word which in this case is 32768 for signed 16-bit integers. As can be seen from Figure 4.2, white noise has a fairly flat and constant frequency response. Pink noise exhibits higher power at lower frequency sub-bands up to 2000 Hz as shown in Figure 4.3. Figure 4.4 indicates that high frequency channel noise is concentrated in the 100 ~2700Hz sub-band. Car noise has most of its power is located below 300Hz as given in Figure 4.5. Figure 4.6 shows that babble noise exhibits its power within the speech sub-band between 50 to 3000 Hz. Factory noise has greater power at frequencies of 1000 Hz and lower, while showing fairly constant power at frequencies above 1000 Hz as

givenFigure 4.7. Figure 4.8 illustrates that machine gun is a bursty noise type and has its power concentrated between 70~500 Hz. As indicated in Figure 4.9 f16 has high noise power between 2400~2600 Hz and 3950~4150 Hz. These peaks are associated with the characteristics of the aircraft used including the type of jet engine, composite material of the body and so on.

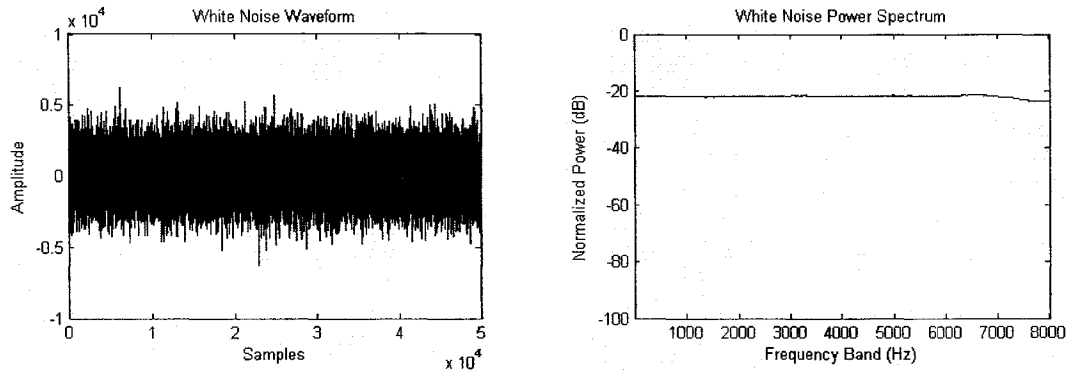

Figure 4.2. Time and frequency plots for white noise

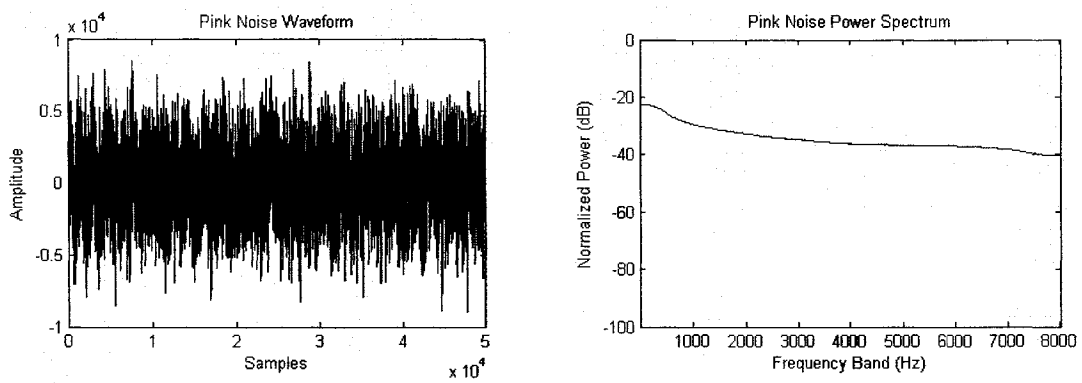

Figure 4.3. Time and frequency plots for pink noise

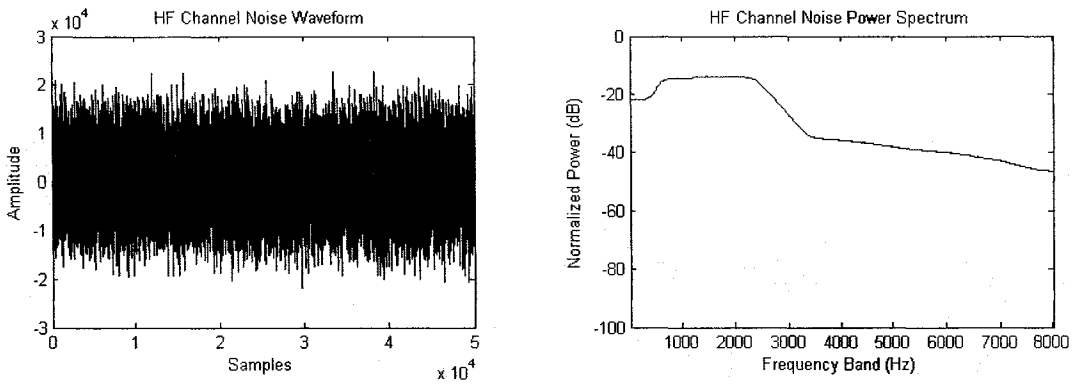

Figure 4.4. Time and frequency plots for HF Channel

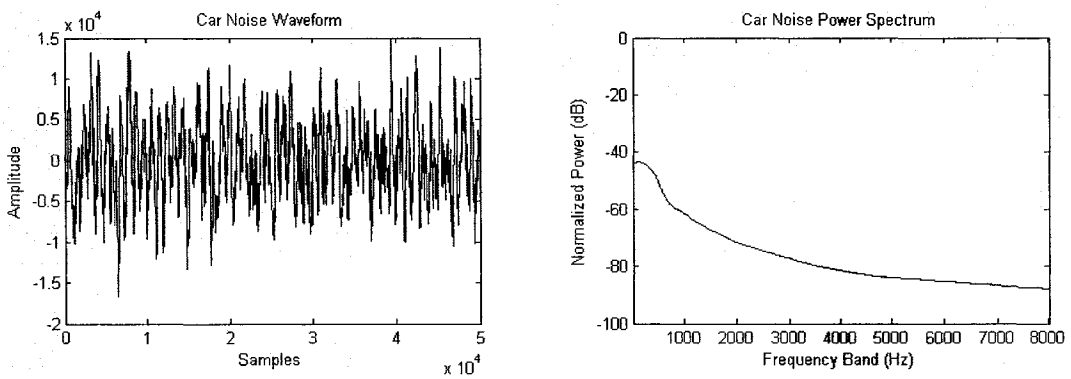

Figure 4.5. Time and frequency plots for car noise

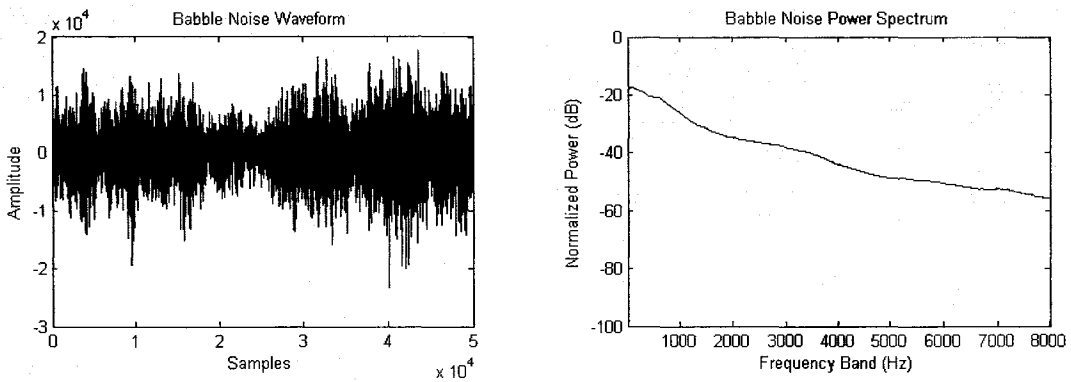

Figure 4.6. Time and frequency plots for babble noise

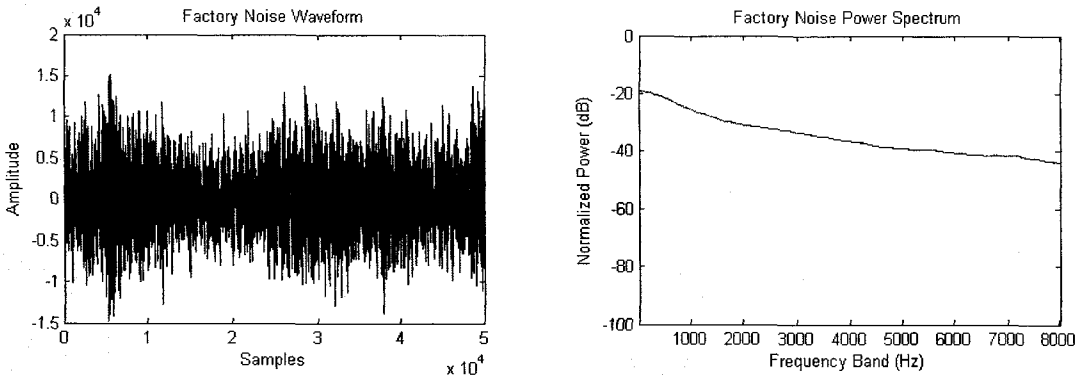

Figure 4.7. Time and frequency plots for factory noise

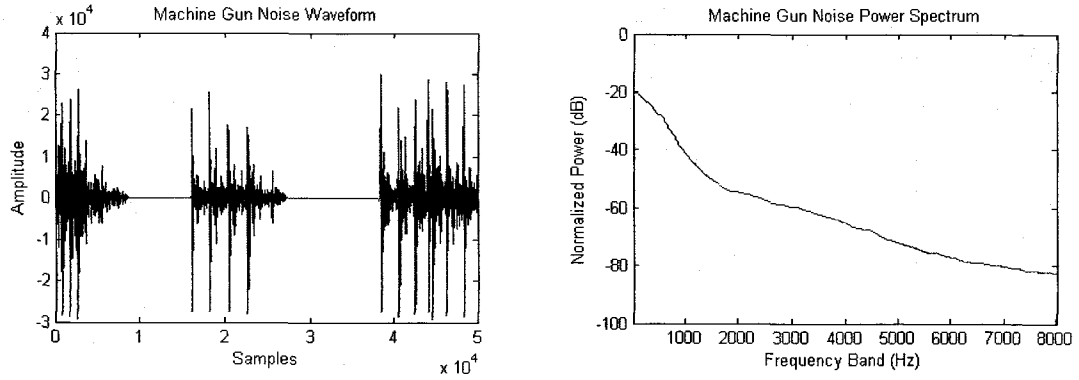

Figure 4.8. Time and frequency plots for machine gun noise

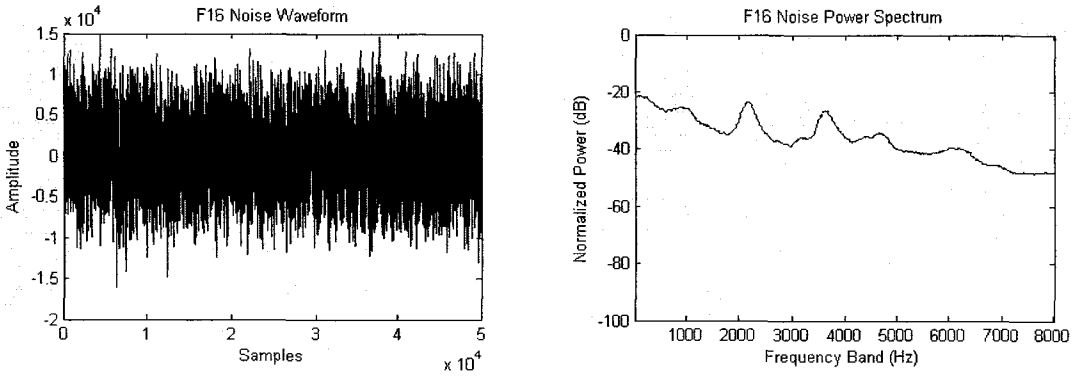

Figure 4.9. Time and frequency plots for F16 noise

CHAPTER 5
ENHANCEMENT ALGORITHM MODIFICATIONS AND ANALYSIS

This chapter describes changes and optimizations made to the algorithms to accommodate recognition accuracy improvement. The goal here is to adapt the various noise removal algorithms chosen in this research within the ASR framework, either by modifying thresholding process or by modifying the filtering techniques. The overall desired effect is to provide recognition accuracy improvement and not necessarily PESQ-MOS or SNR enhancement. Since we are comparing noisy signals with clean reference ones, it would be more beneficial for any enhancement technique geared towards recognition accuracy improvement to place higher priority on speech preservation, rather than noise removal. Hence, the modifications outlined in this chapter are typically less aggressive than their SNR optimized counterparts described by their respective authors. Before we proceed to implement and analyze recognition accuracy performance, simple experimentation was performed to identify a feature extraction scheme more robust to background noise. MFCC and LPCC feature extraction techniques (being most common and computationally efficient) were chosen for comparison and their respective implementation details are given in section 2.3. These trials were performed without any speech enhancement, where only white noise was added to clean input speech at different SNR levels and input into the ASR system. The recognition results were then noted.

5.1 Comparison of MFCC and LPCC

We can see that the MFCC and LPCC provide comparable performance when clean speech is used as shown in Figure 5.1. Essentially, the two techniques compute the speech coefficients in a similar manner. The difference lies in the fact that MFCC uses a triangular filterbank spread logarithmically over the frequency range with varying bandwidth, whereas LPCC can be thought of as implementing a square filterbank spread linearly over the frequency range with equal bandwidth. With clean speech, there is little or no noise in the power spectrogram and the two techniques will exhibit similar behavior. The small differences in recognition accuracies are a result of the fact that MFCC does incorporate the human auditory response in its filterbank design and hence, more of its coefficients are extracted from the speech band. The strength of the MFCC is in its greater resiliency to noise as can be seen from simulation results of Figure 5.1 and Figure 5.2. These are reflective of the fact that most of the MFCC coefficients are extracted from the speech band (i.e. lower frequencies). The MFCC feature vector was significantly corrupted by the white noise induced in the speech band and hence, is more representative of the amount of speech degradation, as opposed to general spectral degradation. Furthermore, we see the recognition accuracy difference between MFCC and LPCC decline to a point where they achieve similar performance at 0 dB SNR. Here, the speech has been severely corrupted by noise and the MFCC is not able to extract reliable coefficients, even from the speech band.

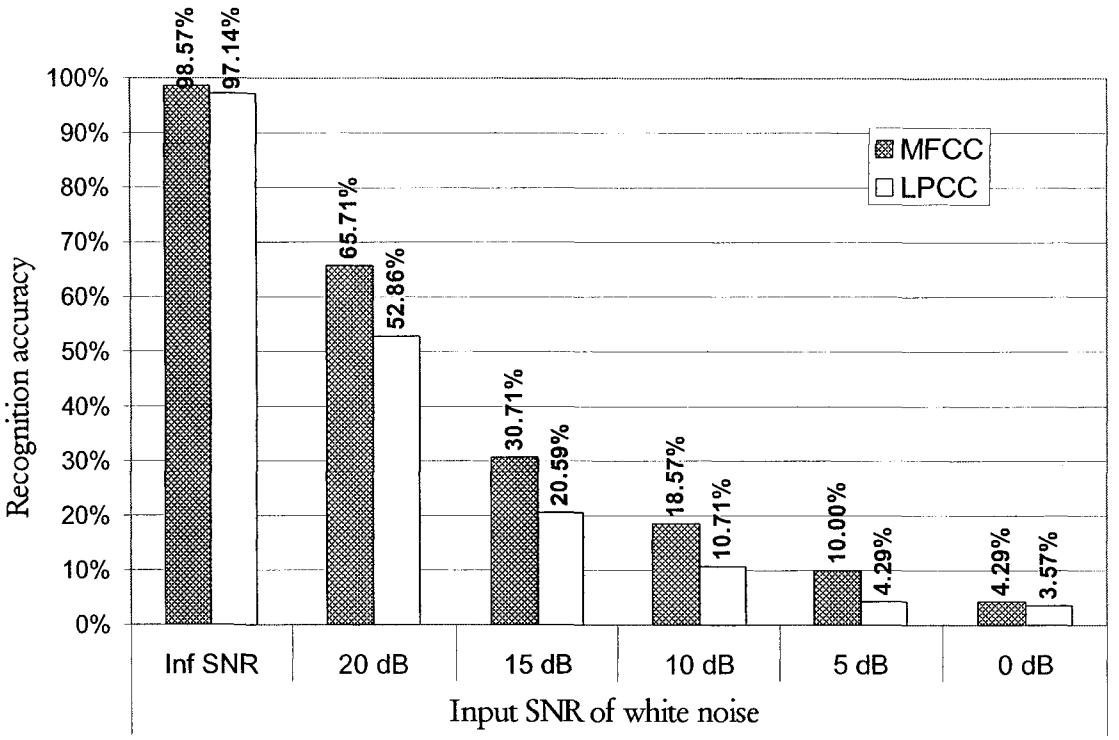

Figure 5.1. MFCC versus LPCC recognition performance with white noise

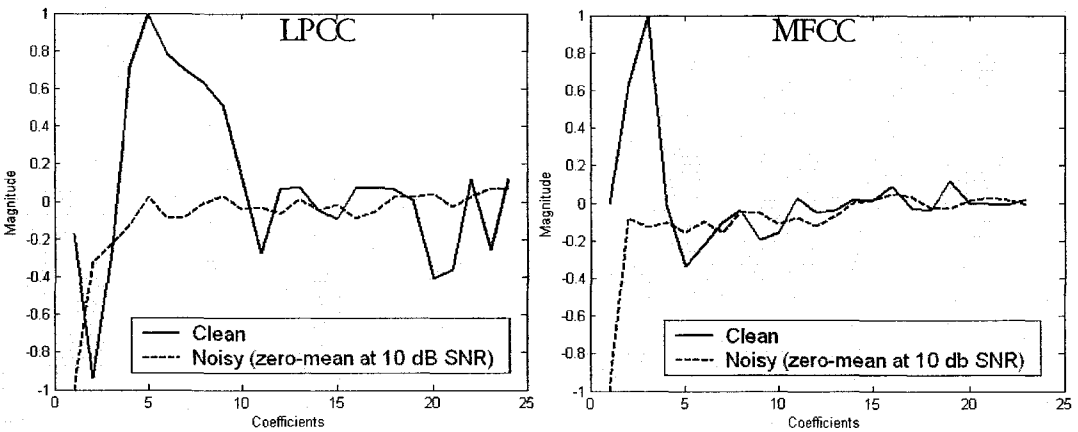

Figure 5.2. MFCC and LPCC coefficients for a specific frame. (Note how the average displacement between the noisy and clean features of MFCC is smaller compared to LPCC)

5.2 Two-Pass Adaptive Noise Estimation

This algorithm was presented in section 3.2 and sweeps through the STDFT twice to achieve an estimate of noise. Given that estimate, simple spectral subtraction can be used; however, due to its subtractive nature some coefficients may end up being reduced to zero. This research implements a simple and effective Wiener filter based noise reduction scheme. This is a two step process where a Wiener filter (H) is designed by:

$$P'_S[l,n] = H[l,n]^2 \, P_S[l,n] \tag{5.1}$$

$$H[l,n] = \frac{\left(P_S[l,n] - \hat{P}_{n,q_{optimal}}[l,n]\right)}{P_S[l,n]} \tag{5.2}$$

Next, to ensure that no coefficients are attenuated to zero, a small fraction of the noise power is allowed to remain, therefore the recovered signal $P''_S[l,n]$ would be given by:

$$P''_S[l,n] = \mathbf{max}\left(P'_S[l,n], \gamma \hat{P}_{n,q_{optimal}}[l,n]\right) \tag{5.3}$$

Figure 5.3 shows the performance of this algorithm with different noise types and input SNRs. It achieves an average improvement of 1.43%, 5.09%, 9.88%, 14.20%, 7.86% over all noise types at 20, 15, 10, 5 and 0 dB input SNR respectively, giving an overall improvement of 7.69%. Recognition accuracy without enhancement (used to calculate the above numbers and for comparison) is given in appendix C. It performs well at low SNRs, showing good improvement at 5 dB input SNR. At higher SNR, there

is less noise for the Q-table to train with, hence less enhancement is achieved; however, an advantage here is the fact that recognition accuracy was not reduced at this higher SNR, indicating that the algorithm does a good job at not distorting speech. This would also make it a good enhancement pre-processing block if it were to be used in conjunction with other de-noising schemes. This is further investigated and tested in section 5.6, where the output of this enhancement algorithm is fed into the wavelet packet based PWAD technique.

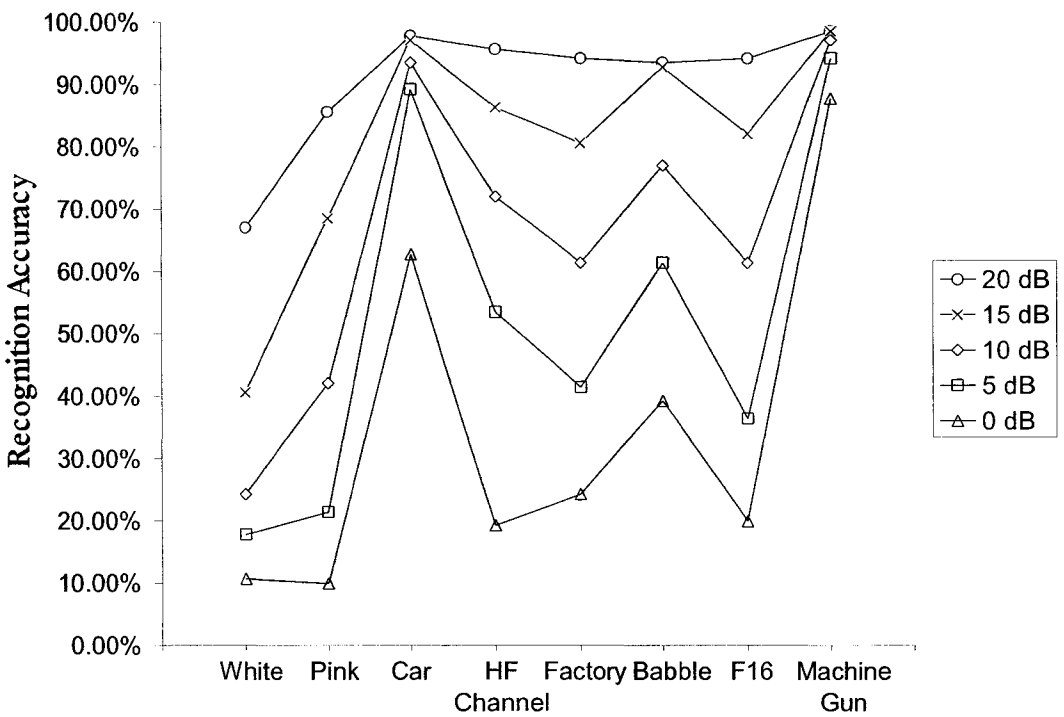

Figure 5.3. Recognition accuracy results of Two-Pass algorithm

PESQ-MOS results are shown in Figure 5.4. We observe an average PESQ-MOS improvement of 0.1554, 0.2088, 0.2282, 0.2208 and 0.1936 for all noise types at 20,

15, 10, 5 and 0 dB input SNR, resulting in an overall PESQ improvement of 0.2013. A minimum suggested PESQ-MOS threshold in the speech evaluation industry is given as about 2.5 [40]. Therefore, this minimum score is maintained at higher input SNRs; however, it drops below the recommended value at 5 and 0 dB input SNR. At this point, noise power is sufficiently high to prevent the two-pass algorithm from bringing PESQ scores above the threshold. However, as can be seen from table 6.1 with white noise, this does not necessarily translate into less recognition accuracy, as the noisy speech can still be enhanced enough to obtain a correct matching with an entry in the user database.

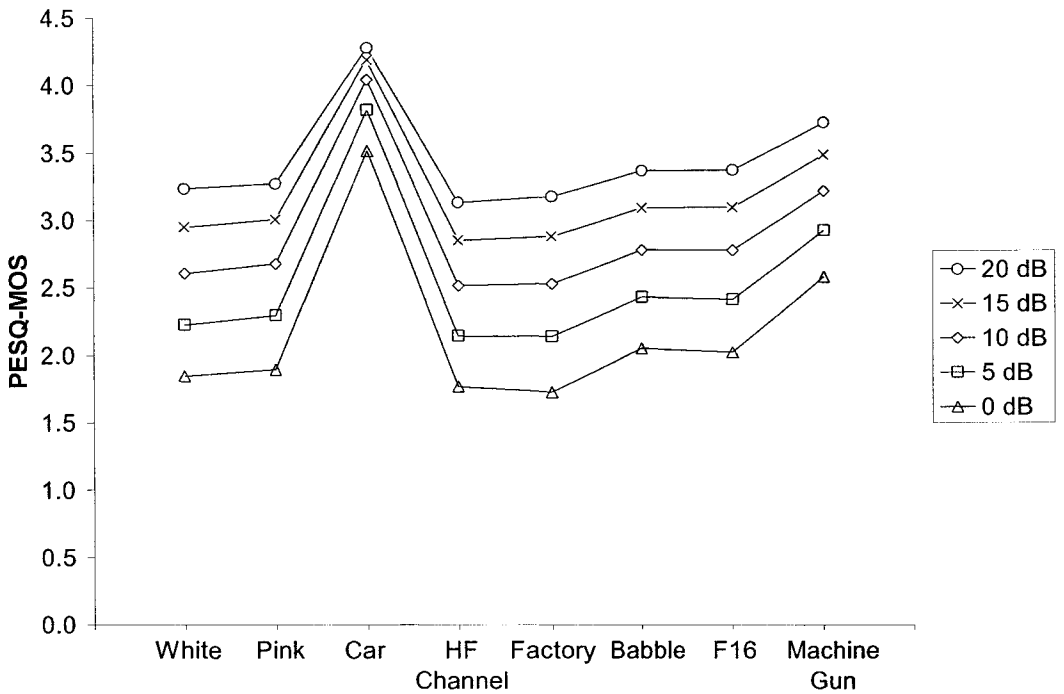

Figure 5.4. PESQ-MOS results of Two-Pass algorithm

In Figure 5.5, average segmental SNR results are plotted for different noise types. We see an average segmental SNR improvement of -0.4168, 3.0096, 5.5350, 7.2351 and

8.3460 dB at 20, 15, 10, 5 and 0 dB input SNR. Recall that average segmental SNR is obtained by comparing the enhanced signal with the clean version in the time domain. Therefore, at higher input SNRs, there is less noise present to remove, causing low average segmental SNR improvement, while lower input SNRs show higher improvement, as comparatively more noise is being removed.

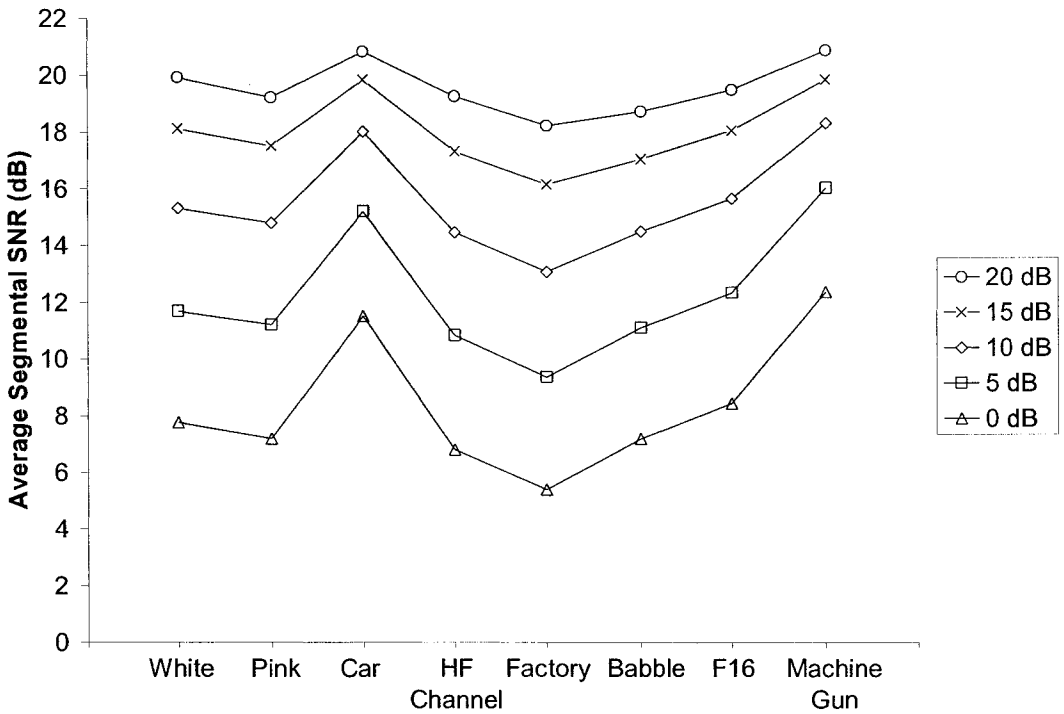

Figure 5.5. Average segmental SNR results of Two-Pass algorithm

5.3 Perceptual Wavelet Adaptive Denoising

To reduce the effects of direct thresholding (i.e. abrupt changes in wavelet coefficients), a modified Ephraim-Malah suppression rule is employed that builds a filter based on a priori and a posteriori knowledge. In other words, it observes the coefficients before and after a decision-directed estimation (i.e. thresholding) and defines a filter that offers high suppression while maintaining a smoothing effect between successive wavelet coefficients. Therefore, for each coefficient, an a posteriori and a priori estimate of the Coefficient to Threshold Ratio (CTR) analogous to an SNR calculation is made as:

$$R_{\omega,p}^{post} = \frac{|S(\omega,n)|}{\lambda_{\omega,p}} \tag{5.4}$$

$$R_{\omega,p}^{priori} = \alpha \frac{|\hat{S}(\omega,n-L_{frame})|}{\lambda(\omega,n-L_{frame})} + (1-\alpha)\,\mathbf{max}\left[0,(R_{\omega,p}^{post}-1)\right] \tag{5.5}$$

where $\alpha = 0.5$ controls the level of suppression required from the filter. $\hat{S}$ represents the outcome of a modified hard-thresholding technique designed to retain some of the noise power and given by:

$$\hat{S}(\omega,n) = \begin{cases} S(\omega,n) & |S(\omega,n| \geq \lambda(\omega,n) \\ \tau S(\omega,n) & |S(\omega,n| < \lambda(\omega,n) \end{cases} \tag{5.6}$$

where λ is the wavelet threshold as calculated in (3.16). An empirically determined value of $\tau = 0.5$ is used. It is important to note here that the reason for use of a modified hard-threshold rather than a soft-threshold lies in the fact that it is not being applied to the coefficients directly, but rather, being fed into the suppression rules above, that will enforce smooth transitions between successive coefficients. Recognition accuracy

experimentation showed that using a soft-threshold in (5.5) above resulted in a suppression rule that was too soft, in that very little thresholding was achieved. With the a priori and a posteriori CTRs defined, the suppression filter is calculated as:

$$H(\omega,n)=\frac{R_{\omega,p}^{priori}}{1+R_{\omega,p}^{priori}}\left[\frac{1}{R_{\omega,p}^{post}}+\frac{R_{\omega,p}^{priori}}{1+R_{\omega,p}^{priori}}\right] \tag{5.7}$$

Figure 5.6 outlines the recognition accuracy performance for the PWAD algorithm over several real and synthesized noise types and at different noise levels.

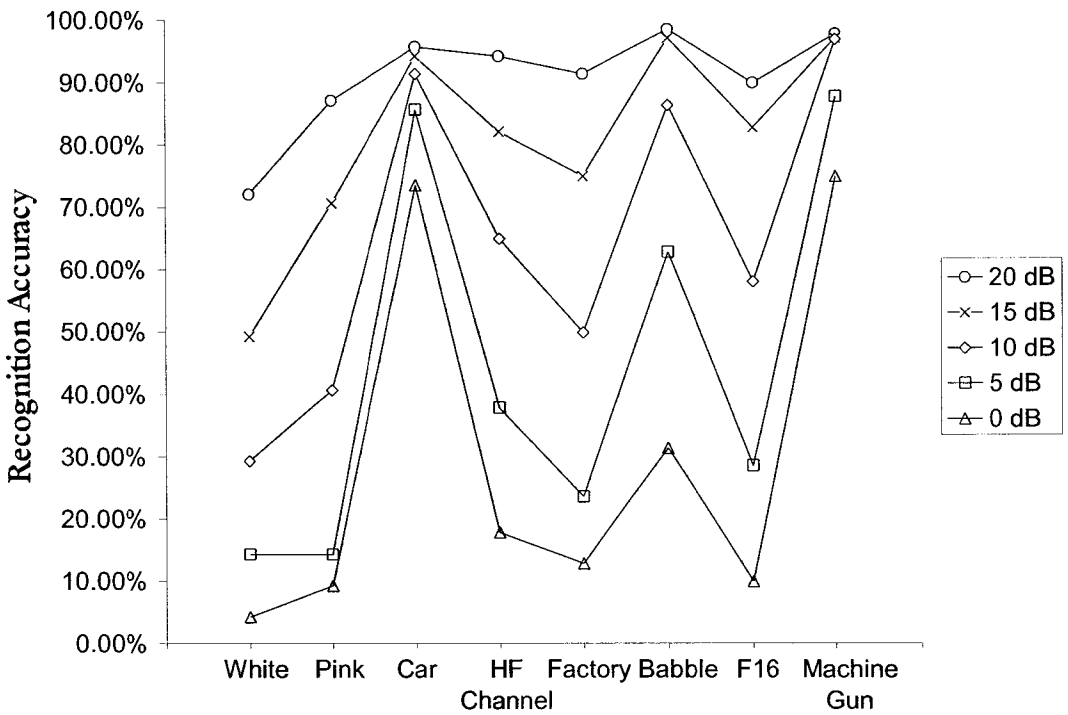

Figure 5.6. Recognition accuracy results of PWAD algorithm

There is an average recognition accuracy improvement of 1.43%, 5.27%, 8.46%, 6.61% and 2.86% over all noise types at input SNRs of 20, 15, 10, 5 and 0 dB

respectively. This gives an average recognition accuracy improvement of 4.93%. At 20 dB input SNR, the technique preserves accuracy performance which demonstrates the technique's ability to provide enhancement while minimizing speech distortion. Recognition accuracy performance metrics with and without enhancement, used to calculate the figures above, are given in appendix C. At 15 and 10 dB input SNR, performance enhancement is high where the algorithm is able to remove more noise from the corrupted signal. Signals corrupted at 5 and 0 dB input SNR also see comparatively lower improvement. This confirms the results given in the literature where wavelet thresholding techniques are generally found to perform well at relatively higher SNRs [27]. The wavelet packet transform produces sparse coefficients where high energy components (i.e. speech) are mapped to higher coefficient values, while lower energy components (i.e. noise) are mapped closer to zero. Hence, when the noise is high, the wavelet noise estimation technique will assume it for speech and will not apply the threshold. An argument here would be to increase the wavelet threshold to account for these elevated noise levels, but this would come at the trade-off of attenuating or distorting speech related features as well, which is detrimental to automatic speaker recognition engines.

We also computed the PESQ-MOS scores and the results are shown in Figure 5.7. Average PESQ-MOS improvement is 0.1027, 0.1261, 0.1424, 0.1503 and 0.1367 for all noise types at 20, 15, 10, 5 and 0 dB input SNR, resulting in an overall enhancement of 0.1316. Again, the PESQ-MOS results show the algorithm's ability to offer good

enhancement at higher noise levels of 15, 10 and 5 dB. From a PESQ-MOS minimum threshold perspective, the scores start to fall below the 2.5 point at 10 dB input SNR only with HF channel and factory noise. All other noise types start falling below the threshold at 5 dB input SNR, with the exception of car noise.

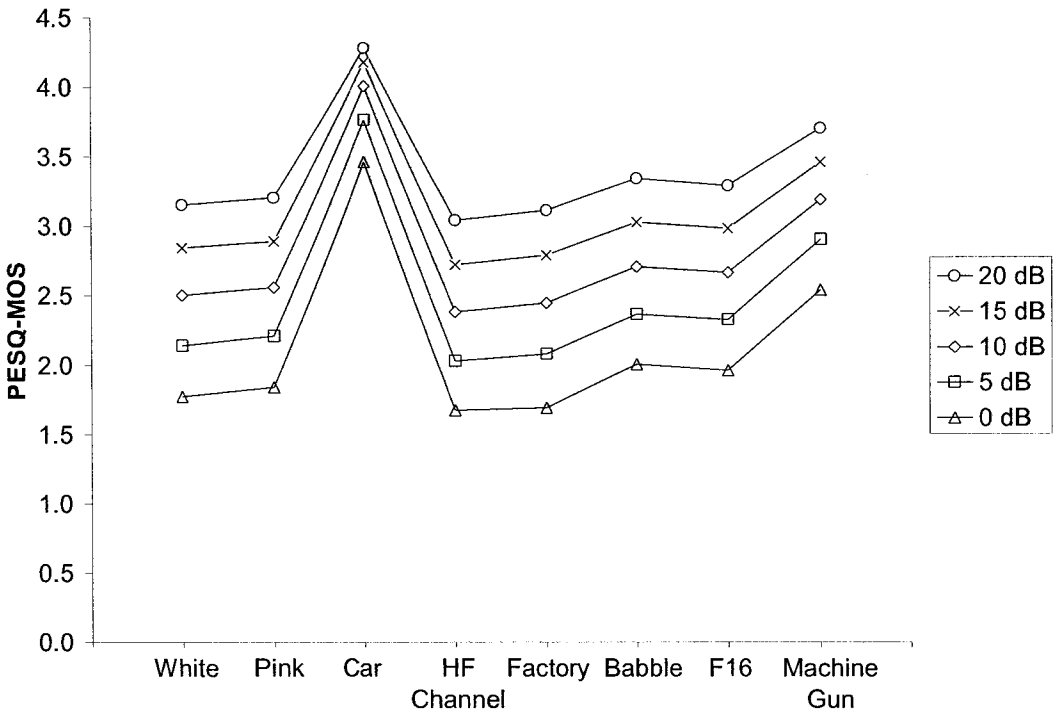

Figure 5.7. PESQ-MOS results of PWAD algorithm

Figure 5.8 illustrates average segmental SNR results for the PWAD algorithm. There is an average segmental SNR improvement of -0.4447, 2.3073, 4.3753, 5.8840 and 7.0435 dB for all noise types at 20, 15, 10, 5 and 0 dB input SNR respectively. The total overall average improvement then becomes 3.8331 dB. These readings show consistent improvement as the input SNR increases to 0 dB, unlike the PESQ-MOS and recognition accuracy results. Average segmental SNR improvement is simply a time domain

measurement of how much noise was removed, based on the silence portion of the waveform. It is understandable that at lower SNRs there is a higher noise level in the speech. Therefore, even if the algorithm removes a constant percentage of noise over all SNRs, the actual noise level reduced will increase at lower SNRs.

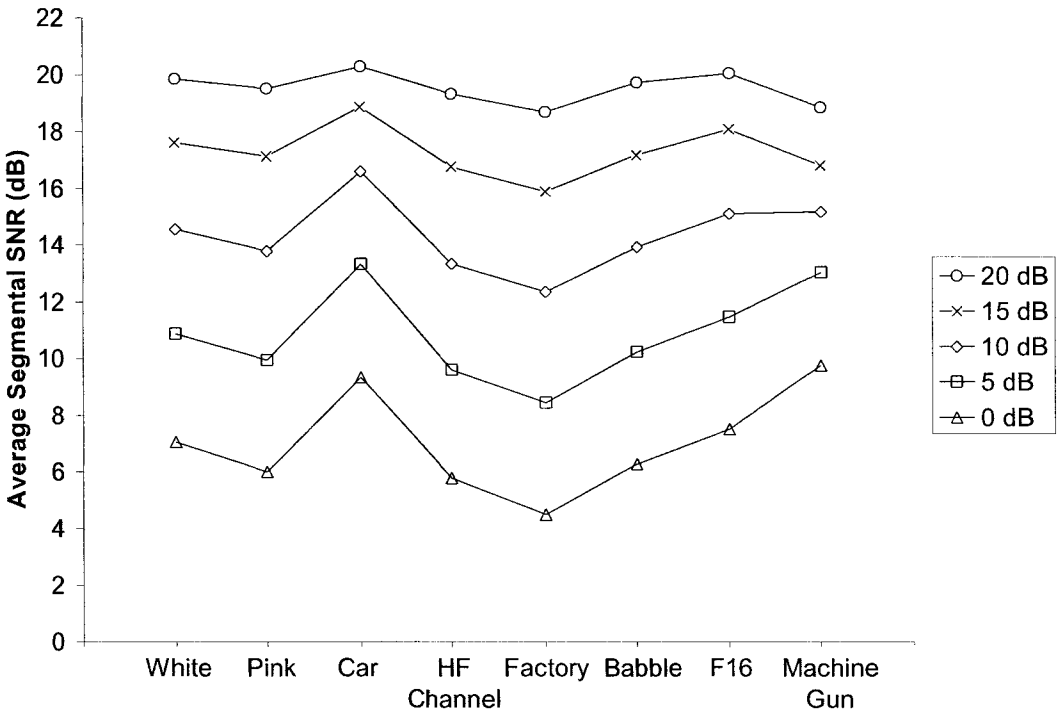

Figure 5.8. Average segmental SNR results of PWAD algorithm

5.4 Two-Dimensional Spectral Enhancement (TDSE)

The TDSE algorithm was not altered in any way from the authors' implementation in [32] and was used mainly for comparison of the other enhancement schemes in this research, with the popular minimum mean square error (MMSE) based method that is optimized for SNR performance. The target of this method is to reduce

the effects of musical noise produced by MMSE through implementation of a 2D smoothing filter as described in section 3.4. The authors prove through SNR results that their addition of the TDSE post-processing block produced better improvement than in its absence [32]. Figure 5.9 below shows recognition accuracy results for this algorithm. This gave an overall average recognition accuracy of -8.56%. Performance is different from the other optimized techniques used in this thesis. At higher SNRs, there was significant reduction in accuracy, giving an average of 20% decline in performance at 20 dB input SNR. As the noise level being added is increased (hence input SNR decreased), the recognition accuracy performance improves, but is never positive.

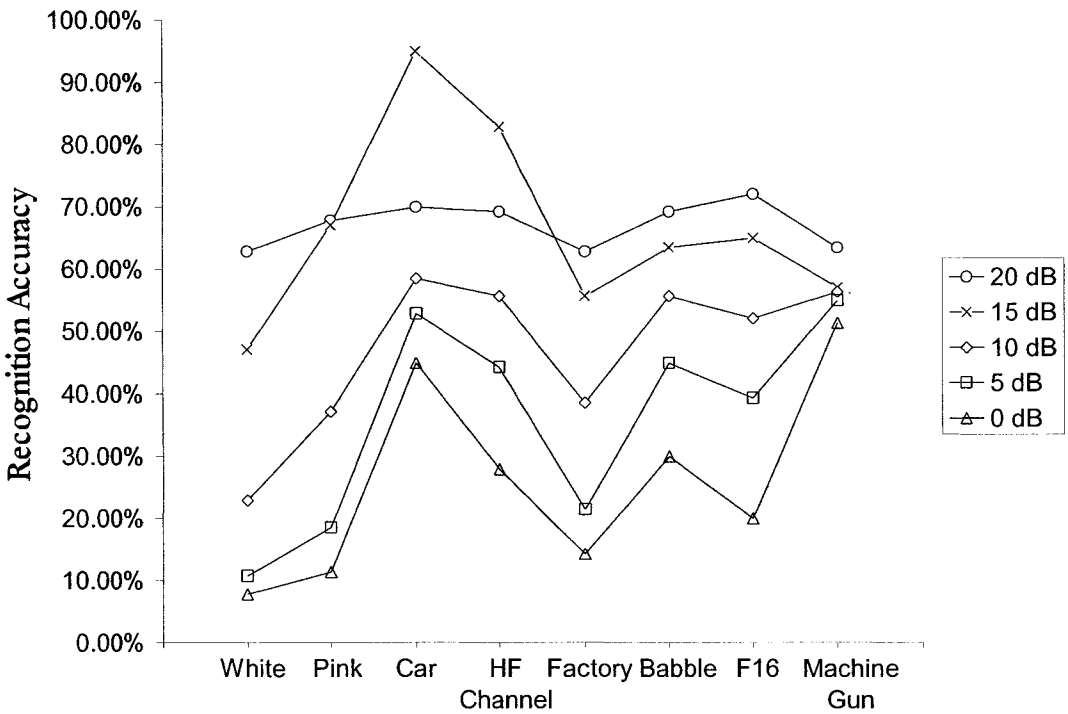

Figure 5.9. Recognition accuracy results of MMSE-TDSE algorithm

Figure 5.10 below presents PESQ-MOS scores for this algorithm. In this case, there is an average PESQ-MOS improvement of -0.0350, 0.1344, 0.2289, 0.2731 and 0.2638 for all noise types at input SNR levels of 20, 15, 10, 5 and 0 dB respectively, producing an overall PESQ-MOS improvement of 0.1730. This greater PESQ-MOS improvement is also reflected in the curves of Figure 5.10, where they are all higher up with less scores falling below the minimum recommended threshold of 2.5. Comparing Figure 5.10 with Figure 5.4,Figure 5.7Figure 5.13Figure 5.17, we see that the MMSE-TDSE algorithm gives comparatively better PESQ-MOS improvement than the other algorithms.

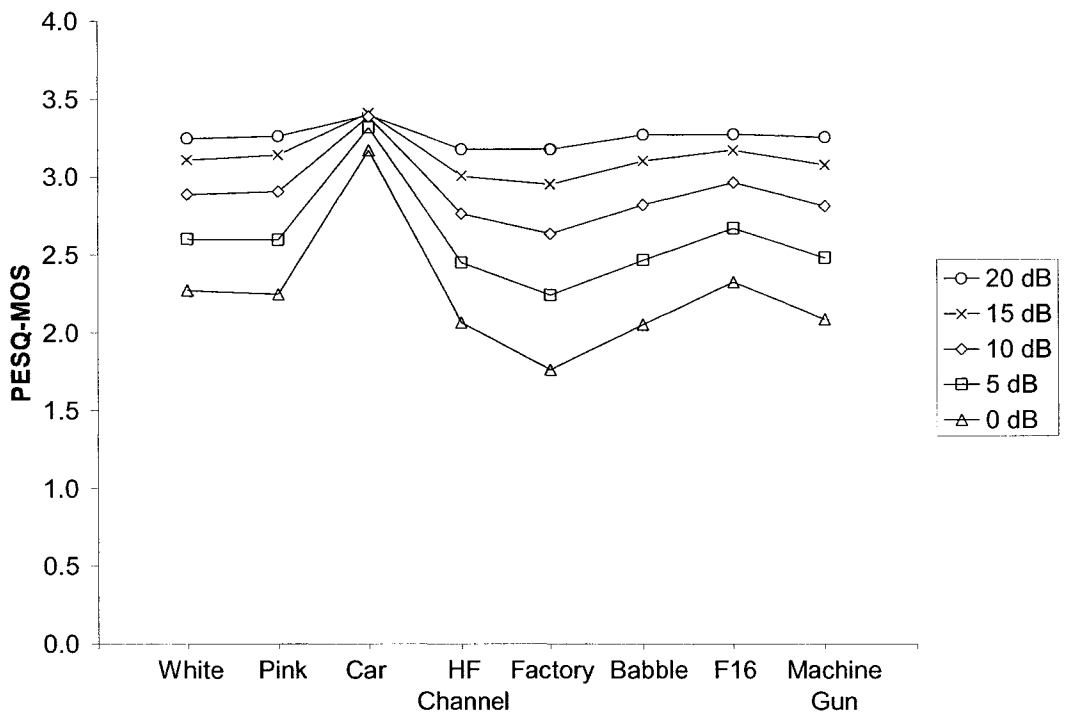

Figure 5.10. PESQ-MOS results of MMSE-TDSE algorithm

Figure 5.11 shows the average segmental SNR results of the MMSE-TDSE technique. We observe an average segmental SNR improvement of -1.1283, 2.2161, 4.8137, 6.8849 and 8.6809 dB for all noise types at 20, 15, 10, 5 and 0 dB input SNR, respectively. Overall average segmental SNR improvement was 4.2935 dB. Based on the three metrics measured for this enhancement method, it is important to note that it provided negative recognition accuracy and PESQ-MOS scores results given lower noise levels (i.e. 20 dB input SNR), which suggests the introduction of speech distortion. Such algorithms designed solely for SNR enhancement do not enforce strict rules on noise over-estimation and will tolerate some speech attenuation. These results further prove that SNR improvement does not necessarily reflect a recognition accuracy improvement.

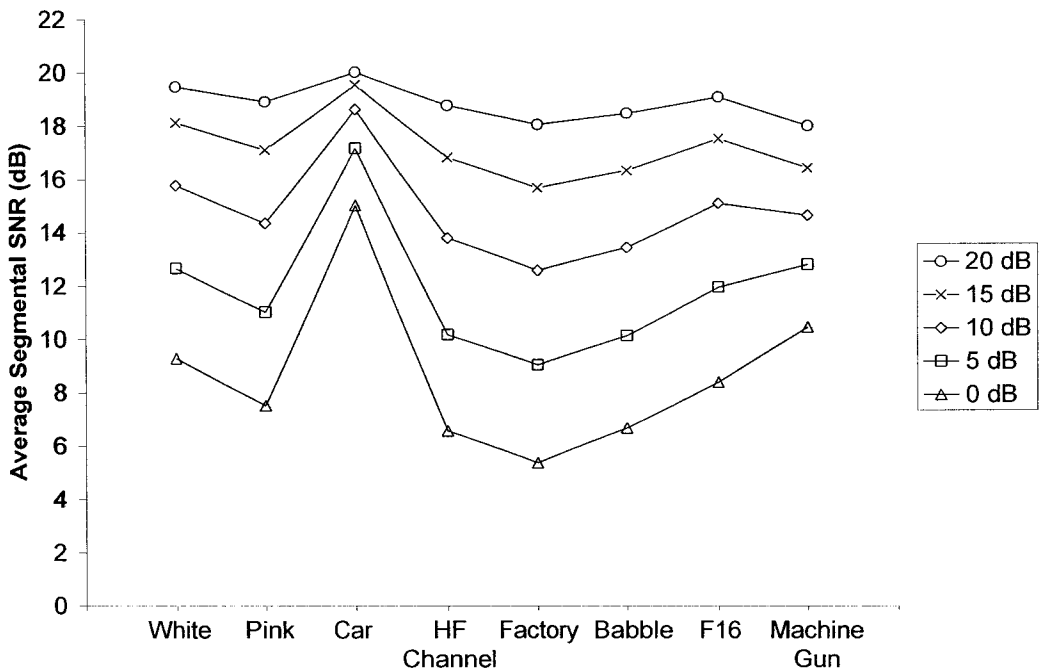

Figure 5.11. Average segmental SNR results of MMSE-TDSE algorithm

5.5 Voice Activity Detection based Noise Floor Estimation

The VAD noise estimation algorithm described in section 3.5 provides us with an estimate of noise. At this point, any filtering scheme can be used, including the one described in section 5.2. However, the authors in [36] propose an optimal short time spectral amplitude (STSA) filter, also commonly known as the Ephraim and Malah filter, that aims at reducing the error in spectral amplitude between the observed and enhanced signal. The reason for optimizing in the spectral amplitude sense stems from the realization that noise cannot be completely removed from a signal without impacting speech properties. Hence, the filter gain is formulated such that after de-noising, the amount of noise left is colorless in nature. It is this property of log spectral amplitude estimation that allows it to achieve good noise removal, while minimizing the effects of residual colored noise. The filter gain function is estimated as:

1. Obtain the a priori SNR ξ based on the variance of the noise and its relation to the a posteriori SNR γ.

$$\xi(k,n) = \alpha G^2(k,n-1)\gamma(k,n) + (1-\alpha)\xi(k,n-1) \tag{5.8}$$

$$\gamma(k,n) = \begin{cases} \dfrac{P(k,n)}{N(k,n)} & \text{if } P(k,n) > N(k,n) \\ 1 & \text{otherwise} \end{cases} \tag{5.9}$$

The averaging parameter $\alpha = 0.5$ was determined by simulation to provide the best recognition accuracy performance, while the gain function G is a result of the Wiener amplitude estimator such that:

$$G(k,n) = \frac{(P(k,n) - N(k,n))^2}{P(k,n)^2} \tag{5.10}$$

2. With those two parameters, the log spectral amplitude estimator can be computed as

$$G_{LSA}(k,n) = \frac{\xi(k,n)}{1+\xi(k,n)} \exp\left(0.5 \int_{t=v(k,n)}^{\infty} \frac{e^{-t}}{t} dt \right) \tag{5.11}$$

where

$$v(k,n) = \frac{\xi(k,n)}{1+\xi(k,n)} \gamma(k,n) \tag{5.12}$$

The recognition accuracy results obtained with this technique are shown in Figure 5.12. Recognition accuracy results show an average improvement of 1.62%, 3.30%, 7.16%, 9.37% and 6.07% for all noise types at 20, 15, 10 and 0 dB input SNR respectively. The overall recognition accuracy improvement for this algorithm is 5.51%. This surpasses the improvement reported by the PWAD method and catches up to the two-pass spectral technique performance. These are interesting results, as this technique does not perform any training, yet manages to give comparatively good results, even at lower SNRs.

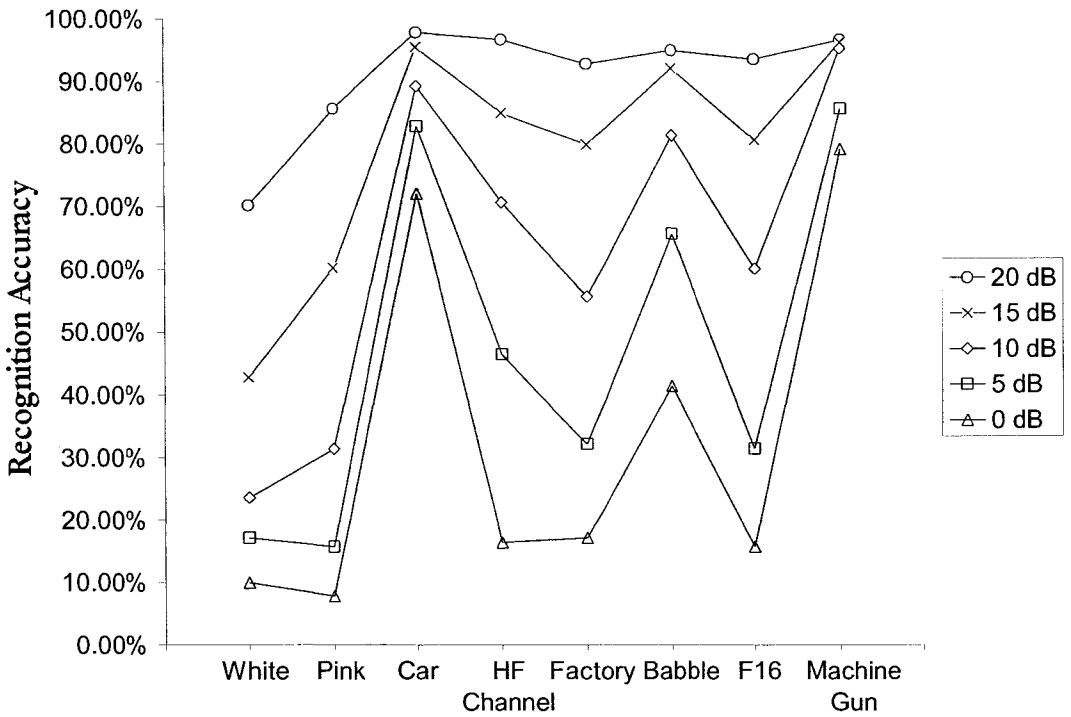

Figure 5.12. Recognition accuracy results of VAD noise estimation

Figure 5.13 presents the PESQ-MOS scores of this algorithm. Average PESQ-MOS improvement is 0.1633, 0.1875, 0.1972, 0.1975 and 0.1774 for all noise types at 20, 15, 10, 5 and 0 dB input SNR. The overall PESQ-MOS improvement is 0.1846. The algorithm manages to keep scores above the minimum 2.5 threshold at input SNRs of 20, 15 and 10 dB, with the exception of car noise. As we have noticed thus far, PESQ-MOS results for car noise are significantly higher, even at low SNRs. Car noise provided by the NOISEX-92 database has its frequency spectrum concentrated between 0-100 Hz (as shown in Figure 5.14), which is a small portion of the speech sub-band. Therefore, this does not reflect as much in PESQ-MOS scores, as they are designed to measure subjective speech quality.

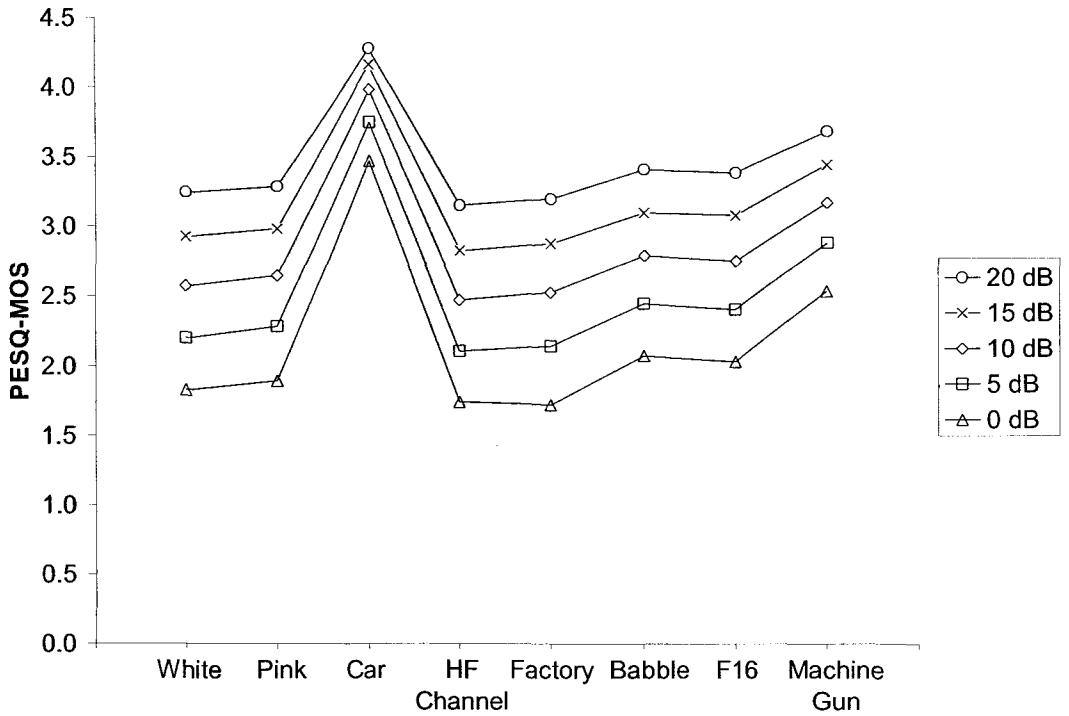

Figure 5.13. PESQ-MOS results of VAD noise estimation

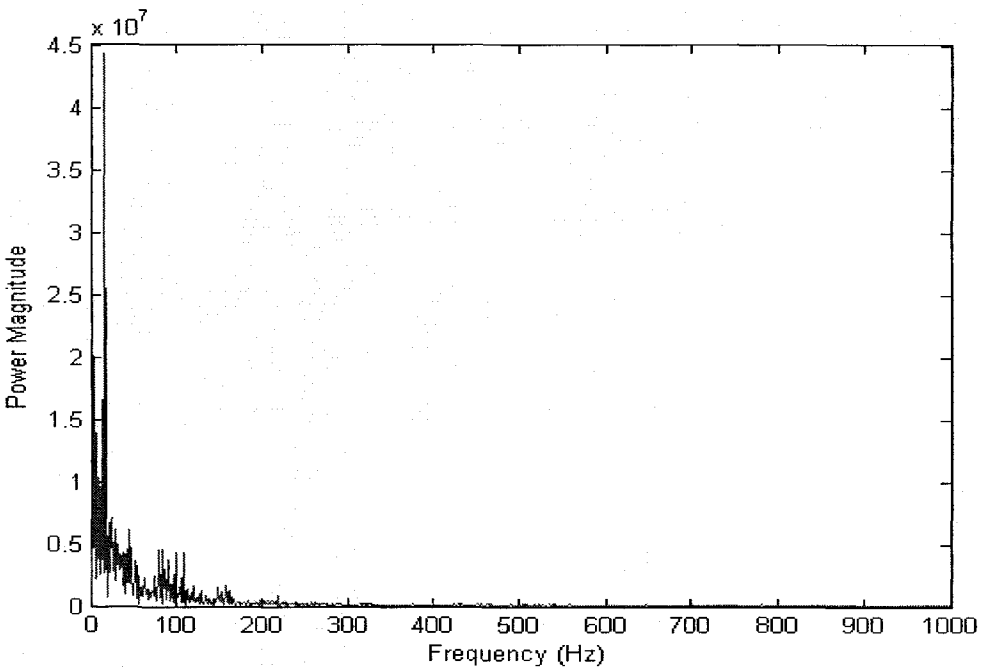

Figure 5.14. Frequency spectrum of NOISEX-92 car noise

Average segmental SNR results are given in Figure 5.15. The algorithm provides an average improvement of 0.0192, 3.1102, 5.4380, 7.0541 and 8.0452 dB for all noise types at 20, 15, 10, 5 and 0 dB input SNR respectively. The overall improvement over different noise types and input SNRs is 4.7333 dB on average. At higher input SNR levels, the improvement is minimal, as the noise floor is lower and there is less noise to remove. Average segmental SNR improvement then gradually increases with the greater noise level, as the algorithm estimates and removes more noise.

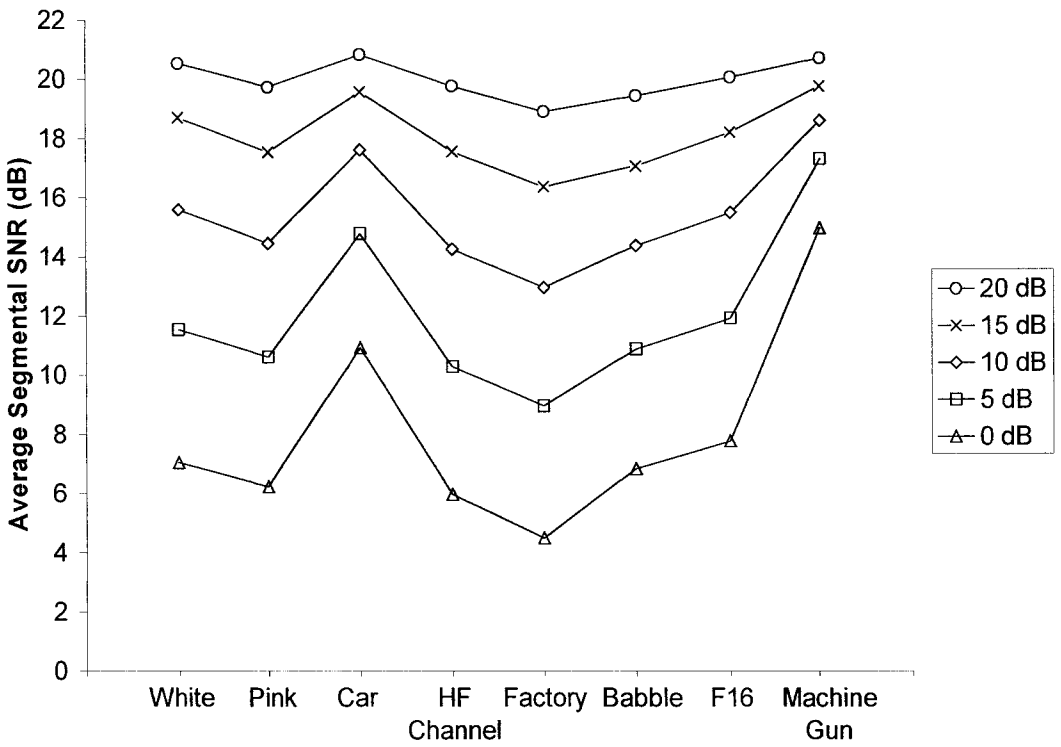

Figure 5.15. Average segmental SNR results of VAD noise estimation

5.6 Combined Two-Pass Spectral and PWAD

The authors in [27] show that when the two-pass and PWAD algorithms are combined, better SNR improvement is achieved. The reason why the two-pass and PWAD algorithms were combined is their compatibility with one another. It is important when using an enhancement algorithm as the pre-processing stage to another that it does not induce any kind of speech distortion when removing noise. This is a strong point of the two-pass technique. The PWAD algorithm has also been proven to work better at higher SNRs and would make a good post-processing stage to the two-pass output. Therefore, the reasoning behind this experiment is to use the two-pass spectral technique to provide initial noise removal (with minimal speech distortion) of the noisy signal such that the PWAD method can offer further improvement. This is implemented by simply running the two-pass algorithm first and then forwarding the enhanced signal to the PWAD block.

For signals that are already at high input SNRs (i.e. have low noise levels), this pairing is expected to produce similar results to the two-pass algorithm run alone, as the PWAD algorithm will see little noise left from the two-pass output and hence apply little or no thresholding. At lower SNRs, performance is expected to increase over the single PWAD implementation, as the output from the two-pass algorithm will still contain enough noise for PWAD to analyze and remove.

Figure 5.16 shows the results of experiments completed with this setup. Average recognition accuracy results over all noise types was -0.80%, 0.09%, 6.66%, 11.25% and 9.20% at 20,15, 10, 5 and 0 dB input SNRs respectively. The overall recognition accuracy was 5.28%, which is slightly greater than the 4.93% given by the PWAD only results. There were better improvements with some specific noise types, where improvements of 5%, 6%, 1% and 3% were observed with car, f16, white and HF channel noise respectively. The other noise types experienced similar performance on average, with the exception of pink noise that saw a 3% performance drop.

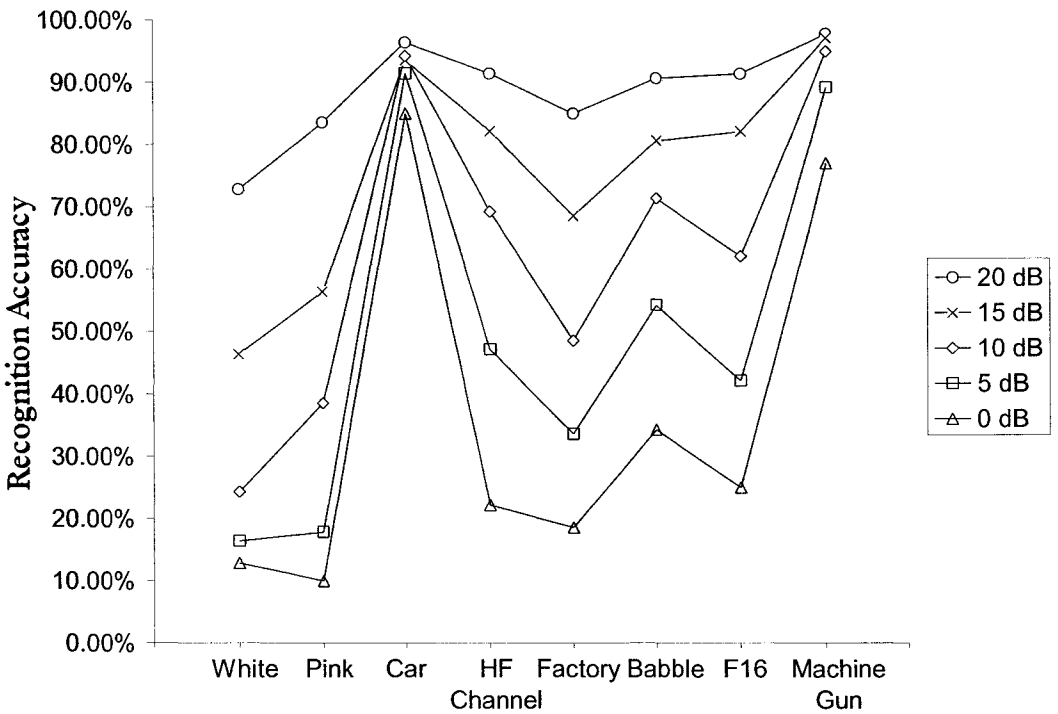

Figure 5.16. Recognition accuracy results of combined algorithm

Figure 5.17 illustrates PESQ-MOS results for this algorithm. Average PESQ-MOS improvement is 0.2024, 0.2644, 0.2918, 0.2919 and 0.2661 for all noise types at 20, 15, 10, 5 and 0 dB input SNR. PESQ-MOS results with and without enhancement, used to calculate the improvement above, are given in appendix C for comparison. The overall PESQ-MOS improvement is 0.2633, which is greater than the 0.1316 improvement given from the PWAD only implementation. This is also apparent from the plots given below, where scores at 10, 5 and 0 dB input SNR are higher than that of Figure 5.7.

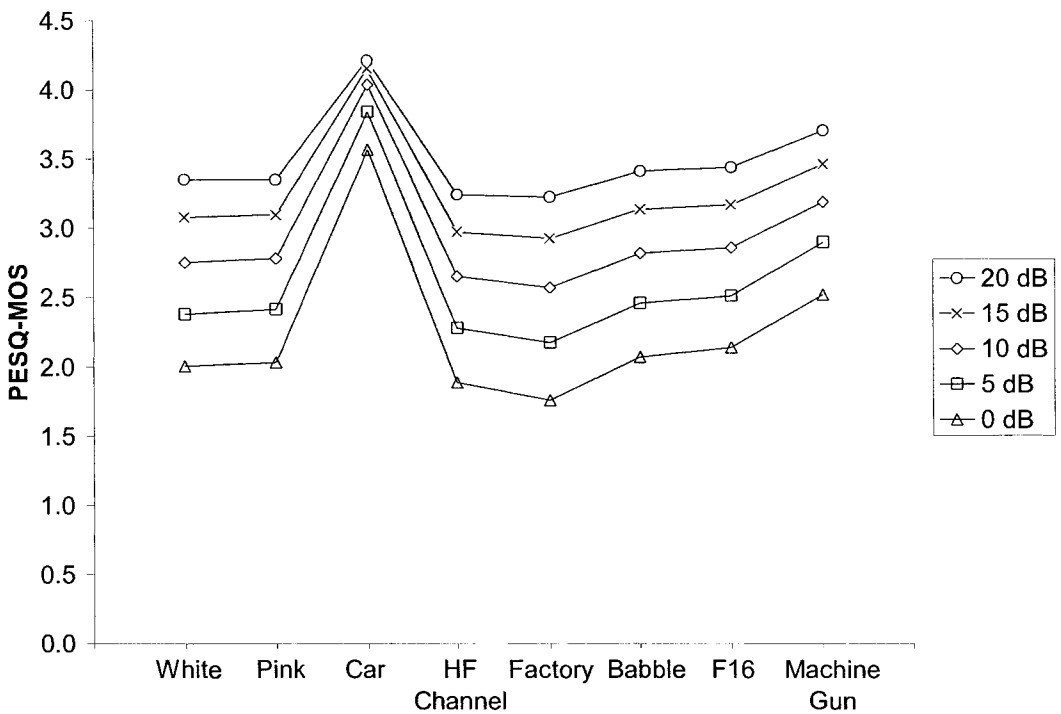

Figure 5.17. PESQ-MOS results of combined algorithm

Figure 5.18 below shows average segmental SNR results. The average improvement across all noise types is -0.9097, 2.5008, 5.0862, 7.1275 and 7.8363 dB at 20, 15, 10, 5 and 0 dB input SNR. Average segmental SNR scores with and without enhancement are given in appendix C. The overall average segmental SNR improvement is 4.5594 dB. Based on performance metrics measured from this algorithm, it provides good recognition accuracy improvement as compared to the no de-noising case and shows similar or better results to the other optimized techniques (i.e. two-pass, PWAD and VAD based noise estimation).

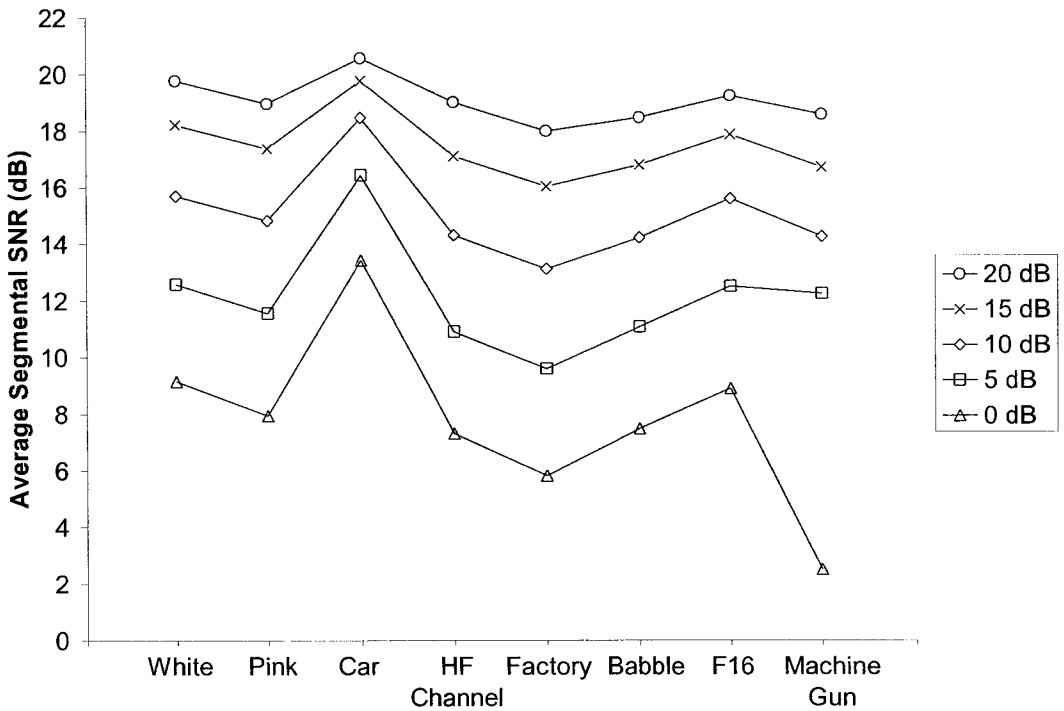

Figure 5.18. Average segmental SNR results of combined algorithm

CHAPTER 6
PERFORMANCE COMPARISON

The main objective of this thesis is to compare the performance of popular noise removal techniques based on ASR recognition accuracy. Enhancement or improvement will refer to the increase in percentage from the no de-noising case, while drop will refer to the opposite, where a decrease in performance is observed. *Benchmark score* signifies recognition accuracy with no background noise added (i.e. high SNR).

With white noise, the perceptual wavelet technique (PWAD) and combined two-pass and wavelet algorithm showed best performance at higher SNRs, giving 72%, 49% and 29% accuracy at 20, 15 and 10 dB SNR respectively, as shown in Figure 6.1 and tabulated in Table 6.1. The perceptual wavelet transform is able to place greater frequency resolution in the lower (i.e. speech) sub-bands. This finer resolution allowed for more accurate thresholding of the wavelet coefficients within the speech sub-bands, maximizing noise removal while minimizing speech distortion. This is similar to the resiliency experienced by the MFCC due to its perceptual analysis nature. However, we see performance of the wavelet technique slip with lower SNRs (starting at 5 dB input SNR) as the noise starts to significantly corrupt speech. At lower SNRs, the two-pass spectral and VAD techniques gave good results with 18% for two-pass and 17% for VAD at 5 dB. This is where spectral noise flooring ensured that the noise level was not being over-estimated, hence, not distorting speech. Two pass spectral and VAD also did

well at 0 dB SNR at 11% and 10% respectively. However, the best performer in terms of recognition accuracy was the combination two-pass spectral and perceptual wavelet system. The advantage of this setup is realized with white noise at 0 dB input SNR, where recognition accuracy was 13% as compared to the 5% given with PWAD only. PESQ-MOS scores showed MMSE-TDSE having the greatest improvement, with an average increase of about 0.7 points over all SNRs. There was no correlation between average segmental SNR results and recognition accuracy, where VAD noise estimation showed best segmental SNR improvement with 20.5447 and 18.7162 dB at 20 and 15 dB input SNR respectively.

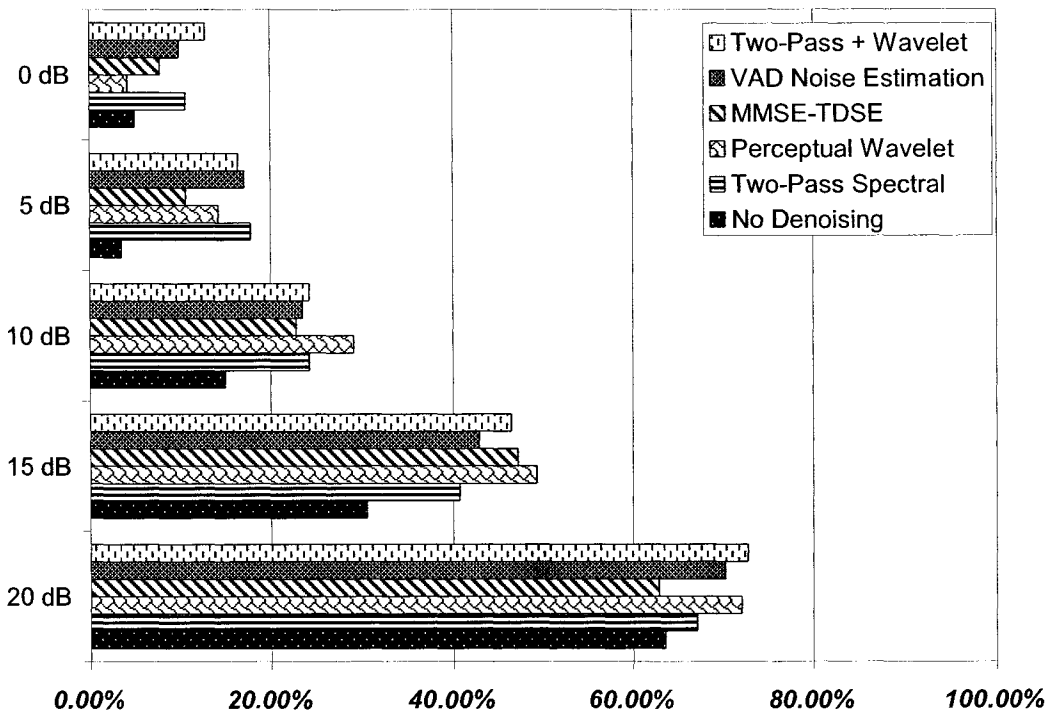

Figure 6.1. Recognition accuracy with white noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
White	Two-Pass Spectral	3.57%	10.00%	9.29%	14.29%	5.71%
	Perceptual Wavelet	8.57%	18.57%	14.29%	10.71%	-0.71%
	MMSE-TDSE	-0.71%	16.43%	7.86%	7.14%	2.86%
	VAD Noise Estimation	6.72%	12.14%	8.57%	13.57%	5.00%
	Two-Pass + Wavelet	9.29%	15.72%	9.29%	12.86%	7.86%

Table 6.1. Recognition accuracy improvement with white noise

With pink noise, the perceptual wavelet (PWAD) technique gave the best recognition performance, as shown in Figure 6.2 and tabulated in Table 6.2. PWAD performed well at higher SNRs with improvement of 5% at 20 dB, 14% at 15 dB and 28% at 10 dB input SNR. At lower SNRs, the two-pass spectral technique showed better recognition accuracy results as the PWAD algorithm reached its SNR limitation. MMSE-TDSE showed negative accuracy improvement at 20 dB input SNR, which is representative of the algorithm's inability to preserve speech properties. At 15 dB input SNR and lower, the noise power being added to the speech is large enough such that even if speech power was attenuated, it did not distort the speech significantly. From a PESQ-MOS and average segmental SNR perspective (as shown in Figure A-2 and Figure B-2), the MMSE-TDSE returned the highest scores, especially at lower SNRs, as it is optimized for SNR improvement. The combination two-pass then PWAD algorithm also returned good SNR and PESQ-MOS improvement very comparable to MMSE-TDSE. This shows that running the two schemes in series is indeed achieving better SNR and PESQ-MOS improvements than those observed from running each algorithm solely.

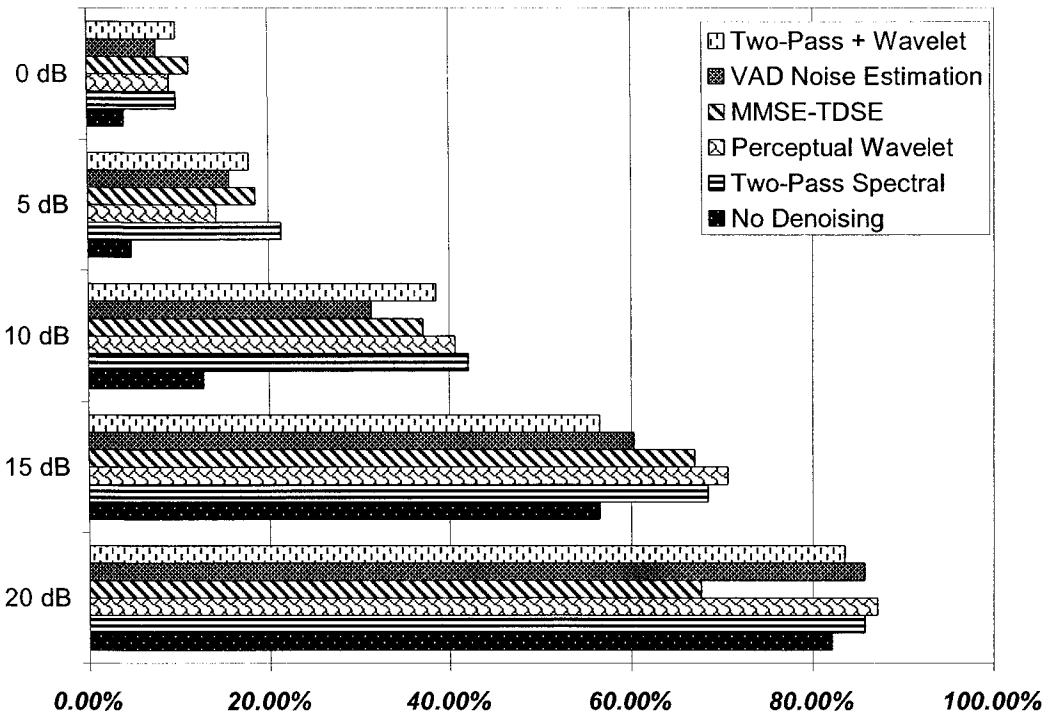

Figure 6.2. Recognition accuracy with pink noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Pink	Two-Pass Spectral	3.57%	12.14%	29.29%	16.43%	5.71%
	Perceptual Wavelet	5.00%	14.29%	27.86%	9.29%	5.00%
	MMSE-TDSE	-14.29%	10.71%	24.29%	13.57%	7.14%
	VAD Noise Estimation	3.57%	3.86%	18.57%	10.71%	3.57%
	Two-Pass + Wavelet	1.43%	0.00%	25.71%	12.86%	5.71%

Table 6.2. Recognition accuracy improvement with pink noise

At high input SNRs with car noise (Figure 6.3 and Table 6.3), VAD noise estimation and two-pass spectral techniques gave best results with recognition accuracies of 98% at 20 dB and 96% at 15 dB input SNR. The PWAD technique was not far behind with 96% at 20 dB and 95% at 15 dB input SNR, indicating that these techniques

are capable of handling stationary noise At 10, 5 and 0 dB input SNR, the two-pass followed by wavelet technique showed best recognition accuracy. In this situation, we again see the effect of using the two-pass algorithm as a pre-processing step to PWAD, where it improved the SNR enough to allow the wavelet algorithm to give even more performance. Observing the difference between PWAD implemented with and without two-pass pre-processing returned an accuracy improvement of 7% on average with pre-processing. This behavior is also observed with F16 cockpit noise, as shown in Figure 6.4, where recognition accuracy with pre-processing improved PWAD performance by 5% at 0 dB input SNR, giving an accuracy increase of 19% over the no de-noising scenario.

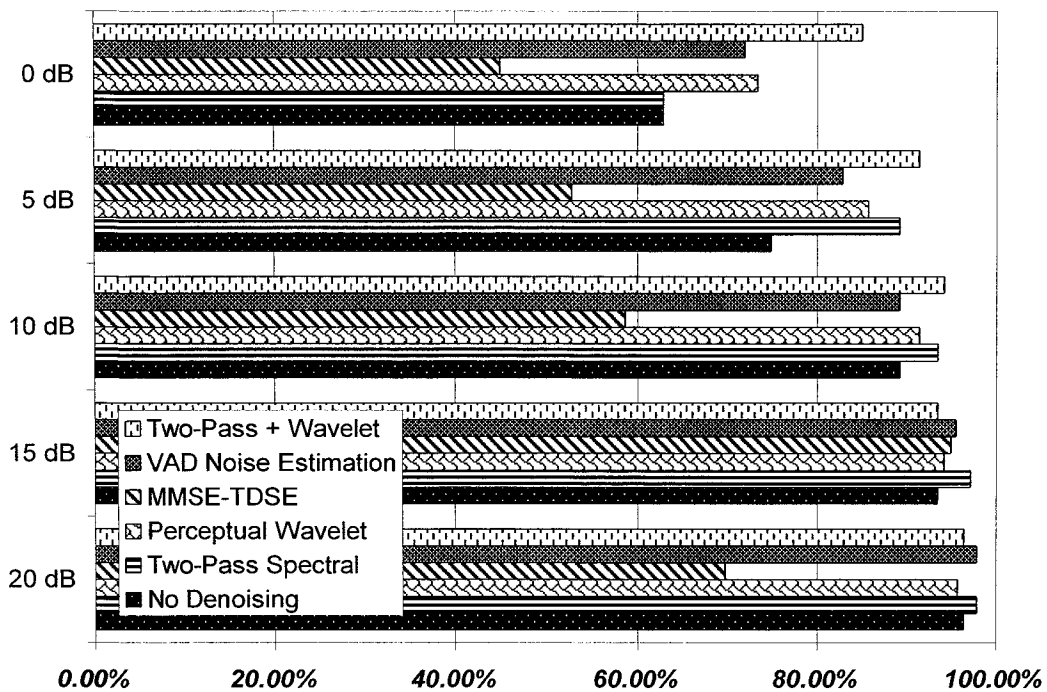

Figure 6.3. Recognition accuracy with car noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Car	Two-Pass Spectral	1.43%	3.57%	4.29%	14.29%	0.00%
	Perceptual Wavelet	-0.71%	0.71%	2.14%	10.71%	10.71%
	MMSE-TDSE	-26.43%	1.43%	-30.72%	-22.14%	-17.86%
	VAD Noise Estimation	1.43%	2.00%	0.00%	7.86%	9.29%
	Two-Pass + Wavelet	0.00%	0.00%	5.00%	16.43%	22.14%

Table 6.3. Recognition accuracy improvement for car noise

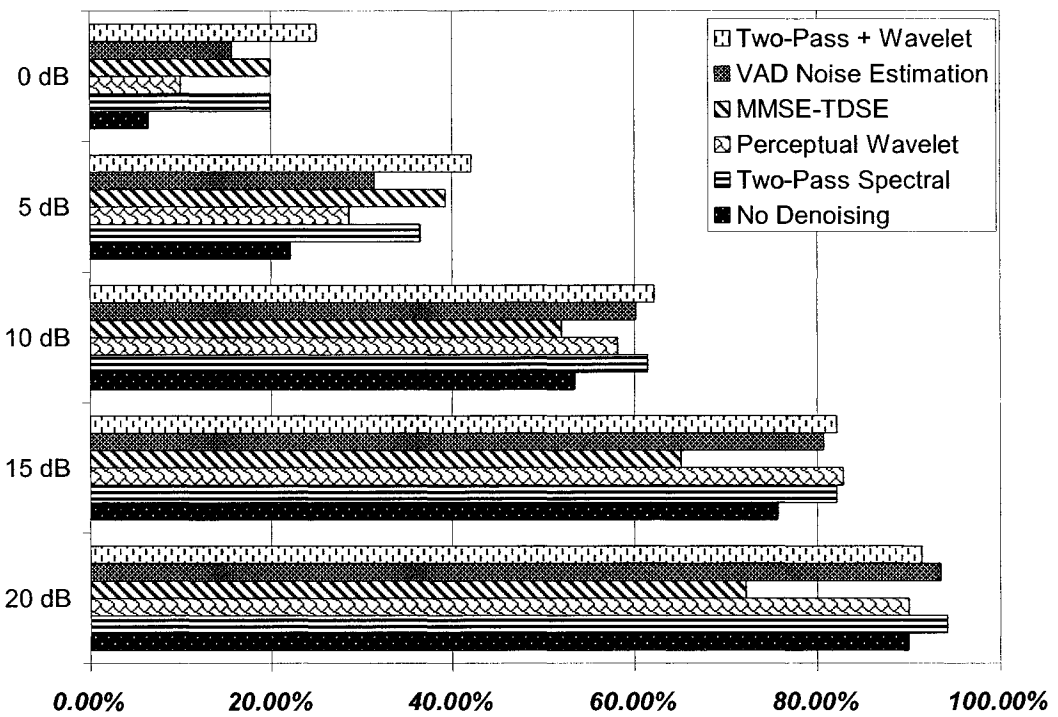

Figure 6.4. Recognition accuracy with F16 noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
F16	Two-Pass Spectral	4.29%	6.43%	7.86%	14.29%	13.57%
	Perceptual Wavelet	0.00%	7.14%	4.57%	6.43%	3.57%
	MMSE-TDSE	-17.86%	-10.71%	-1.43%	17.14%	13.57%
	VAD Noise Estimation	3.57%	5.00%	6.57%	9.29%	9.29%
	Two-Pass + Wavelet	1.43%	6.43%	8.57%	20.00%	18.57%

Table 6.4. Recognition accuracy improvement with F16 noise

With HF channel noise experiments, shown in Figure 6.5 and also tabulated in Table 6.5, all algorithms (with the exception of MMSE-TDSE) gave comparable results with an average recognition accuracy of about 95% at 20 dB input SNR. At 10 dB SNR, VAD noise estimation and two-pass spectral enhancement start to give better performance than the others, showing that the wavelet technique cannot cope with low SNRs. The serially combined two-pass and PWAD scheme gives comparable performance to the PWAD technique at higher SNRs (20 and 15 input SNR) while surpassing PWAD performance at lower SNRs, as the two-pass pre-processing stage removes enough noise for the wavelet technique to provide further enhancement. From a PESQ-MOS and average segmental SNR perspective (Figure A-4 and Figure B-4), the MMSE-TDSE technique gave best results. This is expected, as TDSE was designed to reduce the effects of musical noise resulting from MMSE enhancement and hence, MMSE-TDSE is optimized to achieve maximum segmental SNR and PESQ-MOS improvement.

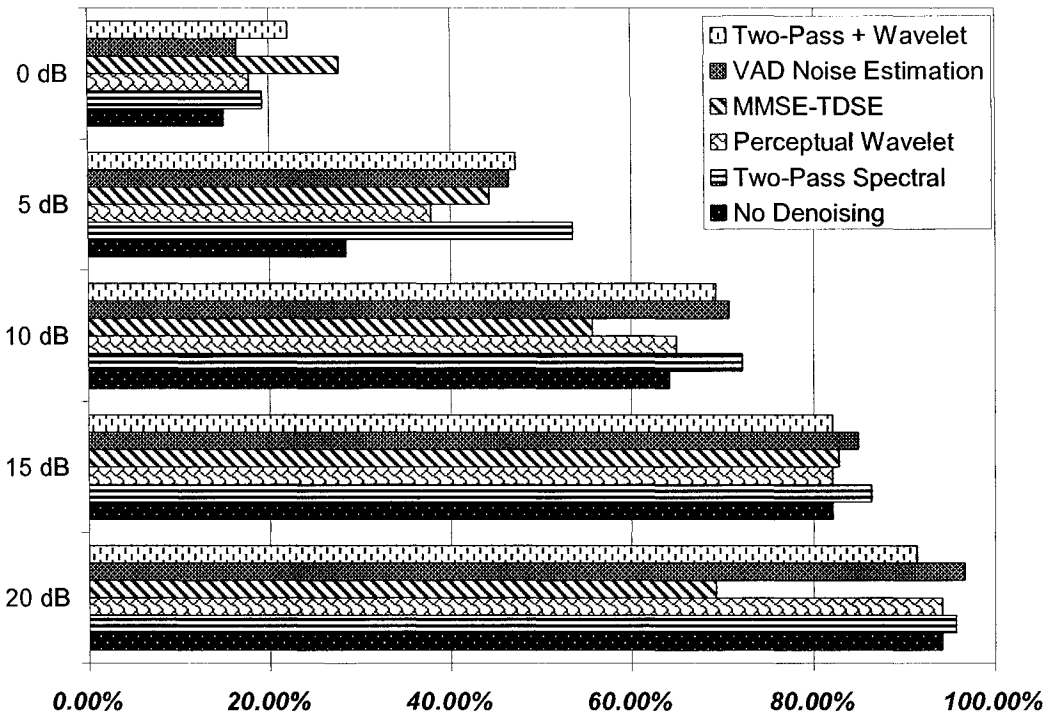

Figure 6.5. Recognition accuracy with HF Channel noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
HF Channel	Two-Pass Spectral	1.43%	4.29%	7.86%	25.00%	4.29%
	Perceptual Wavelet	0.00%	0.00%	0.71%	9.29%	2.86%
	MMSE-TDSE	-25.00%	0.71%	-8.57%	15.72%	12.86%
	VAD Noise Estimation	2.43%	2.86%	6.43%	17.86%	1.43%
	Two-Pass + Wavelet	-2.86%	0.00%	5.00%	18.57%	7.14%

Table 6.5. Recognition accuracy improvement with HF channel noise

Recognition results for speech corrupted with babble noise, given in Figure 6.6 and tabulated in Table 6.6, showed how the perceptual wavelet technique was able to prevent degradation from the no de-noising case. This noise type is difficult for single-channel enhancement techniques to detect and remove since it is harder to distinguish

from the required utterance. PWAD not only maintained accuracy at higher SNRs but also slightly improved performance at lower SNRs, giving 4% accuracy increase at 5 and 0 dB input SNR. This is confirmation that the perceptual wavelet packet transform, coupled with the quantile adapted universal threshold, is preserving speech properties.

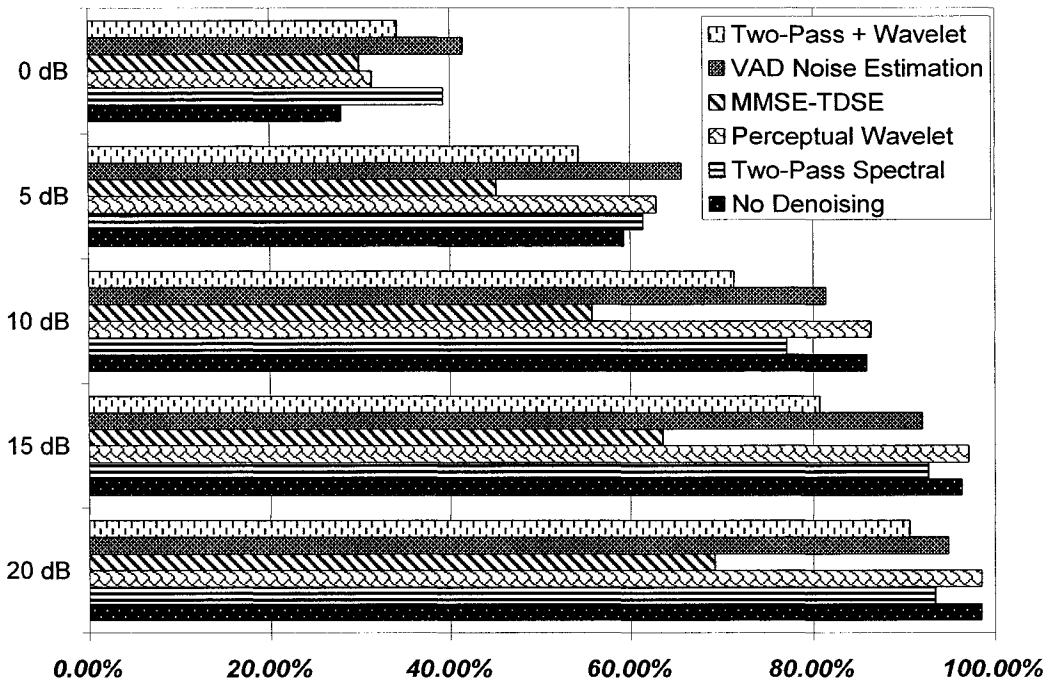

Figure 6.6. Recognition accuracy with babble noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Babble	Two-Pass Spectral	-5.00%	-3.57%	-8.86%	2.14%	11.43%
	Perceptual Wavelet	0.00%	0.71%	0.43%	3.57%	3.57%
	MMSE-TDSE	-29.29%	-32.86%	-30.29%	-14.29%	2.14%
	VAD Noise Estimation	-3.57%	-4.29%	-4.57%	6.43%	13.57%
	Two-Pass + Wavelet	-7.86%	-15.72%	-14.57%	-5.00%	6.43%

Table 6.6. Recognition accuracy improvement with babble noise

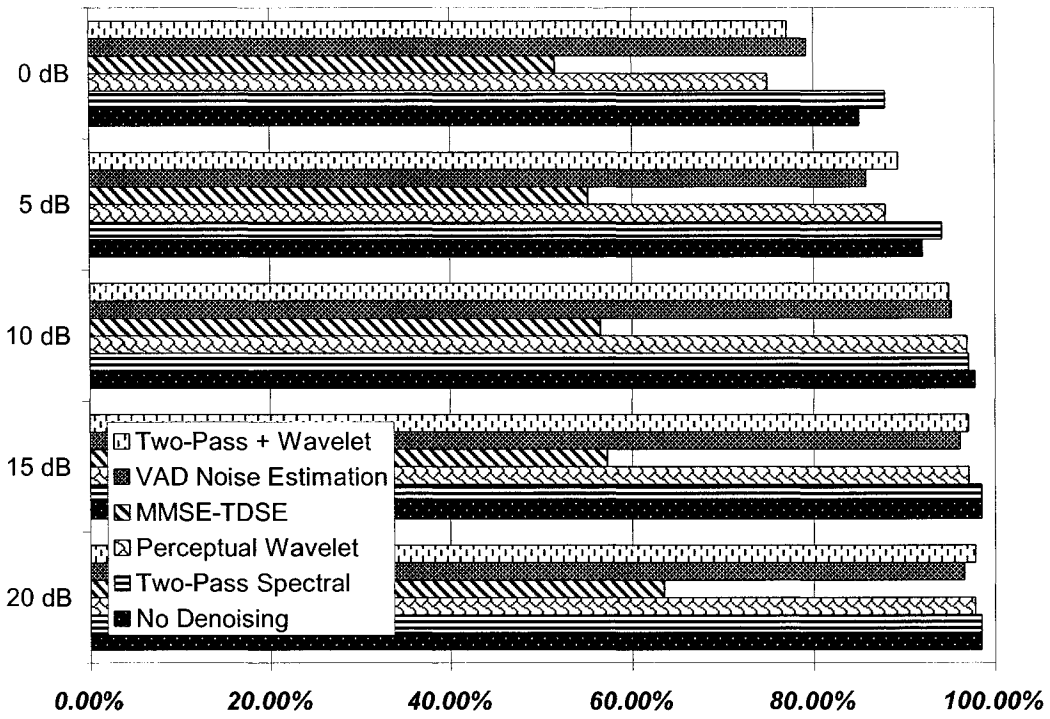

Figure 6.7. Recognition accuracy with machine gun noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Machine Gun	Two-Pass Spectral	0.00%	0.00%	-0.71%	2.14%	2.86%
	Perceptual Wavelet	-0.71%	-1.42%	-0.86%	-4.29%	-10.00%
	MMSE-TDSE	-35.00%	-41.43%	-41.43%	-37.14%	-33.57%
	VAD Noise Estimation	-1.86%	-2.28%	-2.57%	-6.43%	-5.71%
	Two-Pass + Wavelet	-0.71%	-1.43%	-2.86%	-2.86%	-7.86%

Table 6.7. Recognition accuracy improvement with machine gun noise

A rather interesting result is that of performance with machine gun noise in Figure 6.7 and also tabulated in Table 6.7. This type of noise is very difficult to isolate and remove due to its bursty/non-stationary nature and a performance hit was expected. All algorithms exhibited resiliency at 20, 15 and 10 dB SNR, preventing accuracy from

dropping more than 3% (with the exception of MMSE-TDSE). Further inspection showed that machine gun noise has very small corruption energy at higher SNRs (as shown in Figure 6.8) which explains why we still obtained benchmark scores of 98.57% at 20 and 10 dB SNR without de-noising. At lower SNRs, the energy of the noise was large enough to bring accuracy down without enhancement (e.g. 85% at 0 dB SNR). However, the non-stationary property of machine gun noise resulted in over estimation of noise and distortion of speech, where the accuracy dropped up to 7%, except with two-pass spectral, which kept accuracy within 2%, showing the q-tables' ability to track non-stationary noise. PESQ-MOS scores were similar but did favor the two-pass technique with an improvement of about 0.03 across all SNRs, as shown in Figure B-8. Average segmental SNR (Figure A-8) told a completely different story, as its values reduced from the no de-noising case for all algorithms with VAD noise estimation showing the least drop across all input SNRs (e.g. from 18.1424 to 14.9946 at 0 dB SNR).

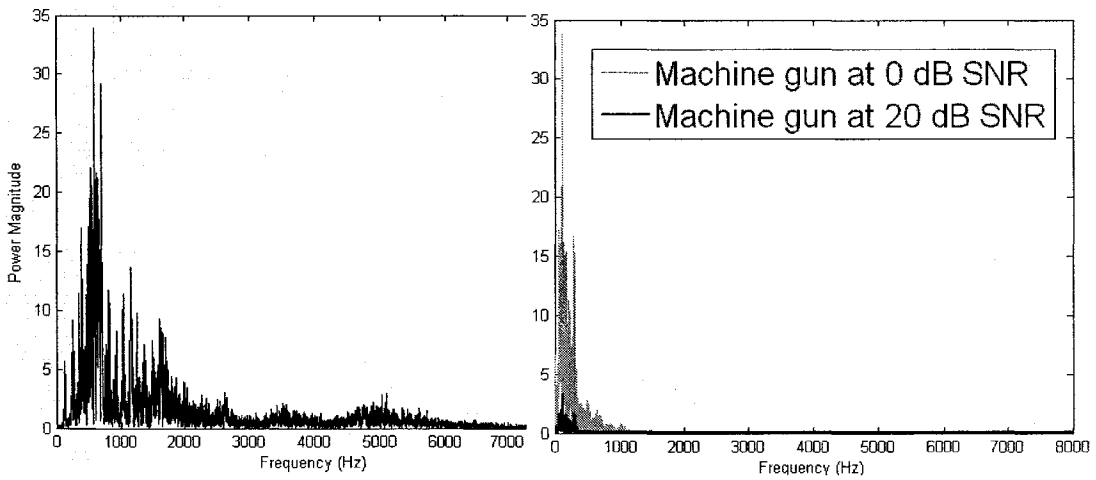

Figure 6.8. Frequency response of clean speech (left) and machine gun noise at 20 and 0 dB input SNR (right)

The two-pass technique provided best results with factory noise, as shown in Figure 6.9 and also tabulated in Table 6.8, resulting in an accuracy improvement of 2% at 20 dB, 8% at 15 dB, 30% at 10 dB, 25% at 5 dB and 20% at 0 dB input SNR. The VAD noise estimation approach followed in performance, giving similar results at 20 and 15 dB input SNR and trailing by an average of 7% recognition accuracy at lower SNRs. Based on the description of factory noise given in section 4.3 and examination of the noise, it is essentially a combination of stationary background noise with bursty equipment noise. The two-pass technique with its training phase was able to adapt and capture this property in its q-table. Hence, it provided the best recognition improvement as the adapted q-table provided for a quantile noise estimate that did not corrupt any speech. The other algorithms were also designed to handle non-stationary behavior and still provided improvement in recognition accuracy from the no de-noising case. However, they were unable to adapt as accurately to this combination of stationary and bursty noise and did not provide as much enhancement as the two-pass technique. From an average segmental SNR (Figure A-5) perspective, all the algorithms performed similarly, with the MMSE-TDSE algorithm showing slightly higher SNR. The PESQ-MOS results (Figure B-5) were also very close, with the two-pass algorithm registering slightly better scores than all the other algorithms. This shows that PESQ-MOS scores may be a better representative of ASR performance as compared to SNR improvements; however, PESQ-MOS and recognition accuracy metrics are still not correlated enough to ensure that a higher PESQ-MOS score would result in better recognition accuracy or vice versa.

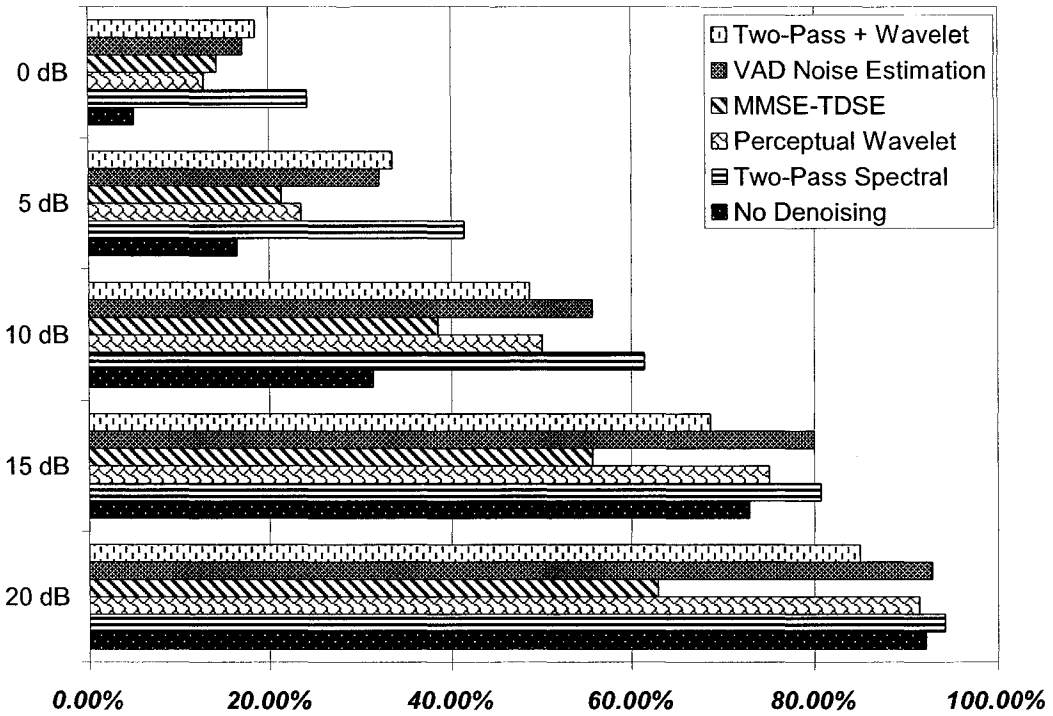

Figure 6.9. Recognition accuracy with factory noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Factory1	Two-Pass Spectral	2.14%	7.86%	30.00%	25.00%	19.29%
	Perceptual Wavelet	-0.71%	2.14%	18.57%	7.14%	7.86%
	MMSE-TDSE	-29.29%	-17.14%	7.14%	5.00%	9.29%
	VAD Noise Estimation	0.71%	7.14%	24.29%	15.71%	12.14%
	Two-Pass + Wavelet	-7.14%	-4.29%	17.14%	17.14%	13.57%

Table 6.8. Recognition accuracy improvement with factory1 noise

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

In the presence of background noise, automatic speaker recognition engines suffer from significant degradation in performance. Several techniques in the literature have tried to address this by using single and multi-channel techniques. In this thesis, we investigated the application of various speech enhancement techniques (two-pass spectral, perceptual wavelet, VAD noise estimation and perceptual wavelet with two-pass pre-processing) with respect to speaker recognition performance. They were chosen based on their ability to provide good SNR improvement, as previously reported in the literature. These algorithms were then modified and applied as pre-processing blocks to an ASR engine and results compared with one another as well as a popular fifth algorithm used without any ASR optimizations (MMSE-TDSE). Each method was tested and analyzed with eight different synthetic and real world noise types (white, pink, car, HF channel, factory, babble, F16 and machine gun) over a range of noise power levels (20, 15, 10, 5 and 0 dB input SNR). Performance was evaluated using three metrics; recognition accuracy, PESQ-MOS and average segmental SNR. Individual results showed SNR and PESQ-MOS improvements for all algorithms, indicating the capability of all five methods to adapt to and remove different types of noise. However,

recognition accuracy improvement was different for each algorithm, reflecting each technique's ability to preserve speech properties during the noise removal process.

The two-pass spectral technique showed good overall performance in terms of recognition accuracy over different noise levels. This is representative of the technique's ability to not only adapt to different input SNRs but also to different noise types, showing good tracking of non-stationary noise as a result of its q-table training. The wavelet technique was able to provide best improvement at higher SNRs, indicative of its ability to preserve speech content in the presence of little or no noise. MMSE-TDSE, which was optimized for SNR and PESQ-MOS improvement, showed that it did distort speech and reduce recognition accuracy for higher SNRs; however, at lower SNRs, it showed some improvement but was not able to compare to the other modified algorithms in terms of recognition accuracy. VAD noise estimation performed well in general and closely followed the two-pass technique in accuracy. The advantage of using this algorithm lies in the fact that it does not require any q-table generation, and hence does not require training. Finally, the two-pass pre-processing stage applied to the perceptual wavelet enhancement system allowed for the shortcomings of the wavelet technique (i.e. higher SNR requirement) to be reduced. The two-pass technique helped the perceptual wavelet algorithm perform better by removing a portion of non-stationary noise, hence, reducing its effects. Therefore, when deciding on a pre-processing enhancement technique to use with ASR systems, average segmental SNR improvement and/or PESQ-MOS scores are not the metrics to use. All techniques returned best

recognition accuracy results with real world noise; however, the two-pass technique gave good results over all input SNRs for most noise types including factory, HF channel and F16 real world noise types. The PWAD technique was best with white and pink noise. VAD noise estimation gave best results with car noise and generally did not fall too far behind in performance from the two-pass technique with other noise types.

Future work in this field would include comparing recognition accuracy enhancement between algorithms that work in the pre-processing stage (as presented in this work) with techniques implemented in the feature extraction and speech modeling stage. We saw MFCC extraction outperform LPCC due to its consideration of the human auditory response. Since the perceptual wavelet transform gave good results at high SNRs, it would be interesting to see how a wavelet domain based feature extraction mechanism would perform with the presence of background noise. An implementation of such a technique is given in [39]. Were it not for timing constraints, it would have been the interest of this research to determine a speech signal performance metric that would be a better representative of recognition results. The purpose here would be to have a method to forecast recognition accuracy behavior before going through the computational complexity and time to produce them. An example of this could be a technique that measures the displacement between clean and noisy MFCC feature vectors for a single waveform and presents an improvement score based on that.

REFERENCES

[1] J.P. Campbell Jr, *"Speaker Recognition: A Tutorial"*, Proceedings of the IEEE, Volume 85, No. 9, Sept. 1997 Page(s):1437 – 1462.

[2] J. Ortega-Garcia, J. Gonzalez-Rodriguez, *"Overview of Speech Enhancement Techniques for Automatic Speaker Recognition"*, Proceedings of ICSLP Fourth International Conference on Spoken Language, Volume 2, 3-6 October 1996, Page(s): 929 – 932.

[3] M. Berouti, R. Schwartz, and J. Makhoul, *"Enhancement of Speech Corrupted by Acoustic Noise"*, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP '79, Volume 4, Apr 1979, Page(s): 208 – 211.

[4] I. Cohen and B. Berdugo, *"Noise estimation by minima controlled recursive averaging for robust speech enhancement"*, IEEE Signal Processing Letters, Volume 9, Number 1, Jan 2002, Page(s): 12-15.

[5] M. Schwab, H-G Kim, Wiryadi and P. Noll, *"Robust noise estimation applied to different speech estimators"*, Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, Volume 2, 9-12 Nov. 2003, Page(s):1904 – 1907.

[6] Jean-François Cardoso, *"Blind Source Separation: Statistical Principles"*, Proceedings of the IEEE, Volume 86, No. 10, Oct. 1998, Page(s): 2009-2025.

[7] T. Gustafsson, B. D. Rao, and M. Trivedi, *"Source Localization in Reverberant Environments: Modeling and Statistical Analysis"*, IEEE Transactions on Speech and Audio Processing, Volume 11, No. 6, November 2003, Page(s): 791 – 803

[8] S. Doclo, and M. Moonen, *"Multimicrophone Noise Reduction Using Recursive GSVD-Based Optimal Filtering with ANC Post-Processing Stage"*, IEEE Transactions on Speech and Audio Processing, Volume 12, No. 1, January 2005, Page(s): 53-69.

[9] S.Y. Low, and S. Nordholm, *"A Hybrid Speech Enhancement System Employing Blind Source Separation and Adaptive Noise Cancellation"*, Proceedings of the 6th Nordic Signal Processing Symposium NORSIG, Espoo, Finland, 9-11 June 2004, Page(s): 204-207.

[10] J.A. Martins, F. Violaro, *"Comparison of parametric representations for hidden Markov models and multilayer perceptron recognizers"*, Telecommunications Symposium, 9-13 August 1998. ITS '98 Proceedings. SBT/IEEE International, Volume 1, 1998, Page(s):141 – 145.

[11] J. Gammal and R.A. Goubran, *"Speaker Recognition in Reverberant Environments"*, Masters Thesis, Carleton University, 2005.

[12] D.A. Reynolds, R.C. Rose, *"Robust text-independent speaker identification using Gaussian mixture speaker models"*, IEEE Transactions on Speech and Audio Processing, Volume 3, No. 1, Jan. 1995, Page(s):72 – 83.

[13] L. Lerato, D.J. Mashao, *"Enhancement of GMM speaker identification performance using complementary feature sets"*, 7th AFRICON Conference in Africa AFRICON 2004, Volume 1, 2004, Page(s):257 – 261.

[14] L.R. Rabiner and R.W. Schafer *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[15] D. Rodriguez-Porcheron, M.Faundez-Zanuy, *"Speaker recognition with a MLP classifier and LPCC codebook"*, IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 2, 15-19 March 1999, Page(s):1005 – 1008.

[16] R. Vergin, D. O'Shaughnessy, A. Farhat, *"Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition"*, IEEE Transactions on Speech and Audio, Volume 7, No. 5, Sept. 1999, Page(s):525 – 532.

[17] S. Davis, P. Mermelstein, *"Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences"*, IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 28, No. 4, Aug 1980, Page(s):357 – 366.

[18] E. Wong and S. Sridharan, *"Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification"*, 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, May 2001, Page(s) 95-98.

[19] Y. Linde, A. Buzo, R.M. Gray, *"An Algorithm for Vector Quantizer Design"*, IEEE Transactions on Communications, Volume 28, No. 1, Jan 1980, Page(s):84 – 95.

[20] X. Wu, K. Zhang, *"A Better Tree-Structured Vector Quantizer"*, Data Compression Conference DCC '91, 8-11 April 1991, Page(s):392 – 401.

[21] E.W. Wan and H. Bai, *"Two-Pass Quantile Based Noise Spectrum Estimation"*, 8th European Conference on Speech Communication and Technology, EuroSpeech 2003, September 1-4, 2003, Geneva Switzerland.

[22] V. Stahl, A. Fischer and R. Bippus, *"Quantile based noise estimation for spectral subtraction and Wiener filtering"*, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'00, Volume 3, Page(s) 1875-1878.

[23] J.O. Smith III and J.S. Abel, *"Bark and ERB bilinear transforms"*, IEEE Transactions on Speech and Audio processing, Volume 7, No. 6, November 1999, Page(s) 697-708.

[24] C. Ris and S. Dupont, *"Assessing Local Noise Level Estimation Methods: Application to Noise Robust ASR"*, Speech Communication, Volume 34, i1-2, April 2001, Page(s) 141-158.

[25] Saeed V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, Second Edition, Chichester, New York, John Wiley, 2000.

[26] D.L. Donoho, *"De-noising by soft-thresholding"*, IEEE Transactions on Information Theory, Volume 41, No. 3, May 1995, Page(s): 613-627.

[27] Q. Fu and E. Wan, *"Perceptual Wavelet Adaptive Denoising of Speech"*, 8th European Conference on Speech Communication and Technology, EuroSpeech 2003, September 1-4, 2003, Geneva Switzerland.

[28] Donald B. Percival and Andrew T. Walden, *Wavelet methods for time series analysis*, Cambridge, England, Cambridge University Press, 2000.

[29] Bruce W. Suter, *Multirate and wavelet signal processing*, San Diego, Academic Press, 1998.

[30] M. Black and M. Zeytinoglu, *"Computationally efficient wavelet packet coding of wide-band stereo audio signals"*, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'95, Volume 5, 9-12 May 1995, Page(s): 3075 – 3078.

[31] M. Bahoura and J. Rouat *"Wavelet speech enhancement based on the Teager energy operator"*, IEEE Signal Processing Letters, Volume 8, No. 1, January 2001, Page(s): 10-12.

[32] Z. Lin and R.A. Goubran, *"Musical noise reduction in speech using two-dimensional spectrogram enhancement"*, Proceedings of the 2nd IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications, 20-21 Sept 2003, Page(s) 61-64.

[33] M. Schwab, H-G Kim, Wiryadi, and P. Noll, *"Robust noise estimation applied to different speech estimators"*, Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers 2003, Volume 2, 9-12 Nov. 2003, Page(s): 1904-1907.

[34] I. Cohen, B. Berdugo, *"Noise estimation by minima controlled recursive averaging for robust speech enhancement"*, IEEE Signal Processing Letters, Volume 9, No. 1, Jan 2002, Page(s): 12-15.

[35] Y. Ephraim and D. Malah, *"Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator"*, IEEE Transactions on Acoustics, Speech and Signal Processing, Volume ASSP-32, No. 6, December 1984, Page(s): 1109-1121

[36] Y. Ephraim and D. Malah, *"Speech enhancement using a minimum mean-square error log-spectral amplitude estimator"*, IEEE Transactions on Acoustics, Speech and Signal Processing, Volume ASSP-33, April 1985, Page(s): 443-445.

[37] International Telecommunications Union, *"Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,"* Telecom Standardization Section, ITU-T, P.862, 2001 [Online]. Available: http://www.itu.int/rec/T-REC-P.862-200102-I/E

[38] Rice University Digital Signal Processing (DSP) group, *"NOISEX noise database"*, Signal Processing Information Base, SPIB, Sept. 1995 [Online], Available: http://spib.rice.edu/spib/select_noise.html

[39] B. Kotnik, Z. Kacic and B. Horvat, *"The usage of wavelet packet transformation in automatic noisy speech recognition systems"* EUROCON , vol 2, 22-24 Sept. 2003, Page(s): 131-134.

[40] VoIP Quality Testing, *"Voice Quality Testing - PESQ"*, Opticom, 2006, Available: http://www.opticom.de/technology/pesq.html

APPENDIX A

This appendix displays average segmental SNR results of each algorithm for comparison.

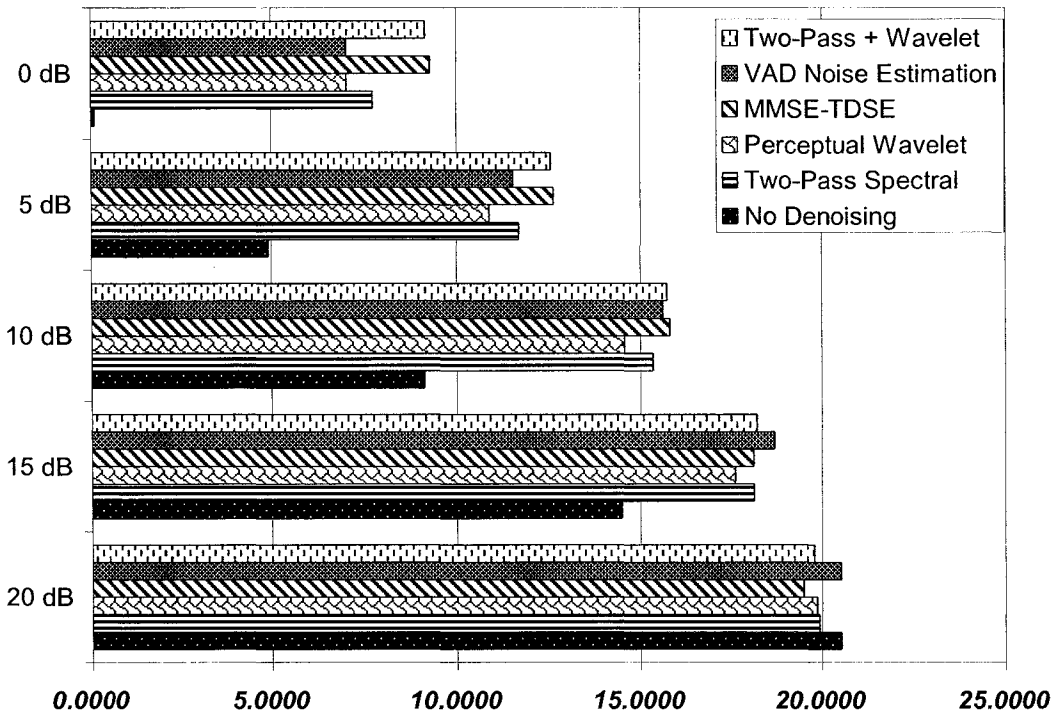

Figure A-1. Average segmental SNR with white noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
White	Two-Pass Spectral	-0.6173	3.6423	6.2046	6.7957	7.6650
	Perceptual Wavelet	-0.6857	3.1246	5.4404	5.9779	6.9624
	MMSE-TDSE	-1.0642	3.6305	6.6659	7.7602	9.1978
	VAD Noise Estimation	-0.0096	4.2066	6.4778	6.6434	6.9483
	Two-Pass + Wavelet	-0.7603	3.7148	6.5875	7.6834	9.0744

Table A-1. Average segmental SNR improvement with white noise

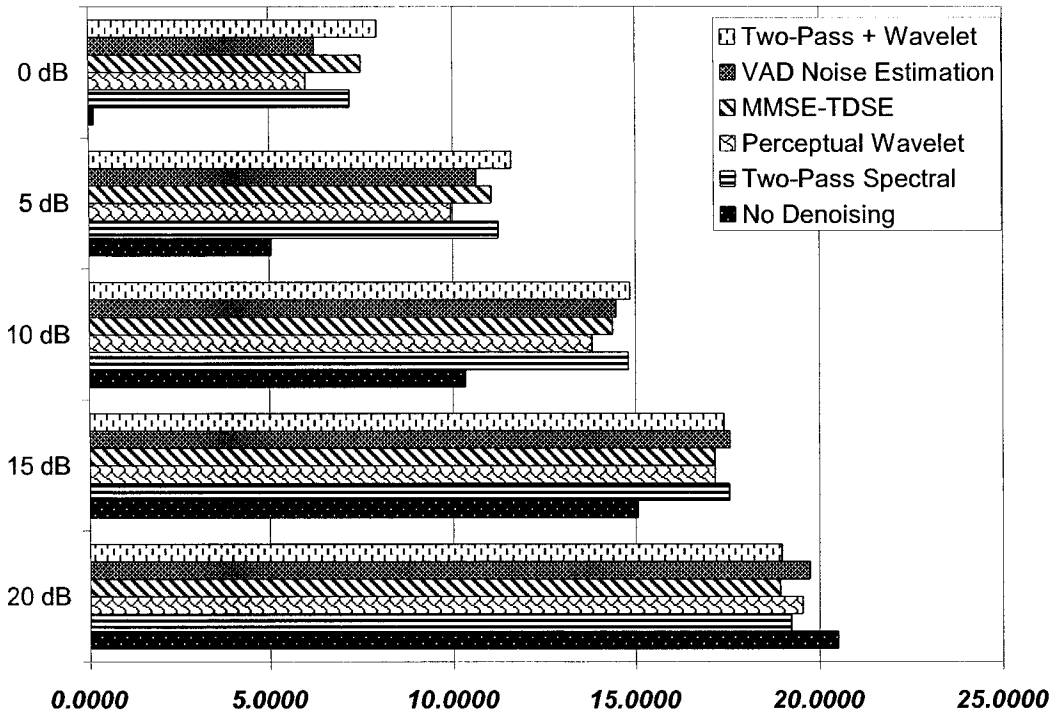

Figure A-2. Average segmental SNR with pink noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Pink	Two-Pass Spectral	-1.2769	2.4711	4.4550	6.1599	7.0729
	Perceptual Wavelet	-0.9823	2.0803	3.4535	4.8984	5.8734
	MMSE-TDSE	-1.5810	2.0742	4.0264	5.9696	7.3908
	VAD Noise Estimation	-0.7658	2.4899	4.1199	5.5677	6.1137
	Two-Pass + Wavelet	-1.5366	2.3200	4.5034	6.4991	7.8227

Table A-2. Average segmental SNR improvement with pink noise

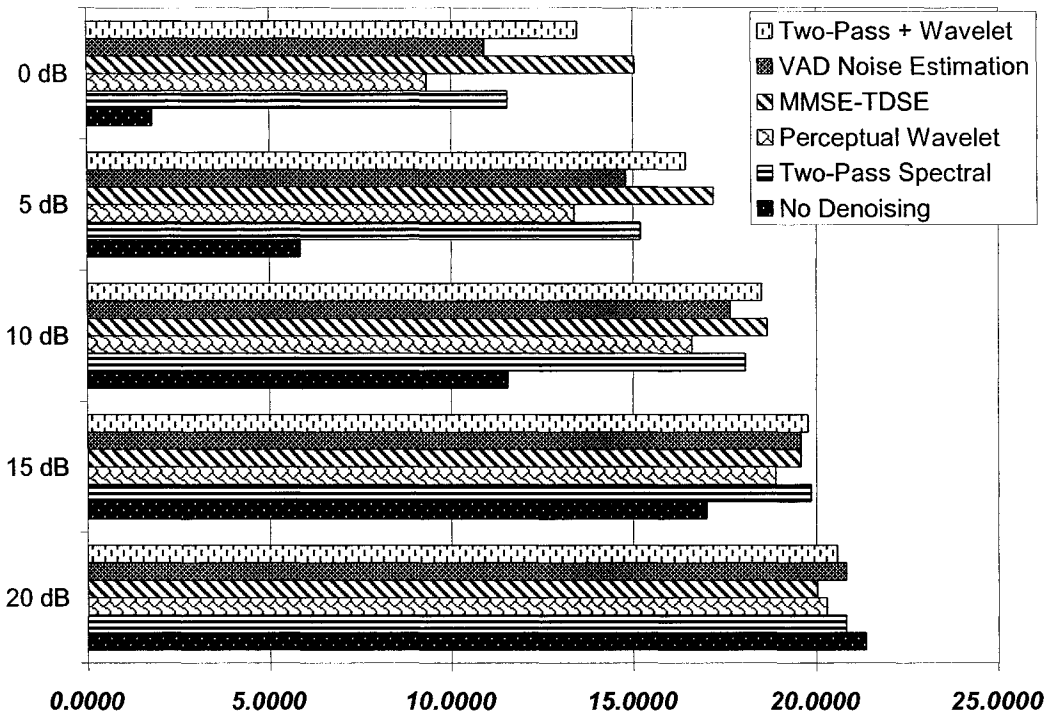

Figure A-3. Average segmental SNR with car noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Car	Two-Pass Spectral	-0.5387	2.8626	6.4917	9.3495	9.7409
	Perceptual Wavelet	-1.0759	1.8773	5.0651	7.4874	7.5552
	MMSE-TDSE	-1.3446	2.5795	7.0970	11.3239	13.2621
	VAD Noise Estimation	-0.5350	2.5862	6.0794	8.9392	9.1408
	Two-Pass + Wavelet	-0.7935	2.7884	6.9369	10.5899	11.6559

Table A-3. Average segmental SNR improvement with car noise

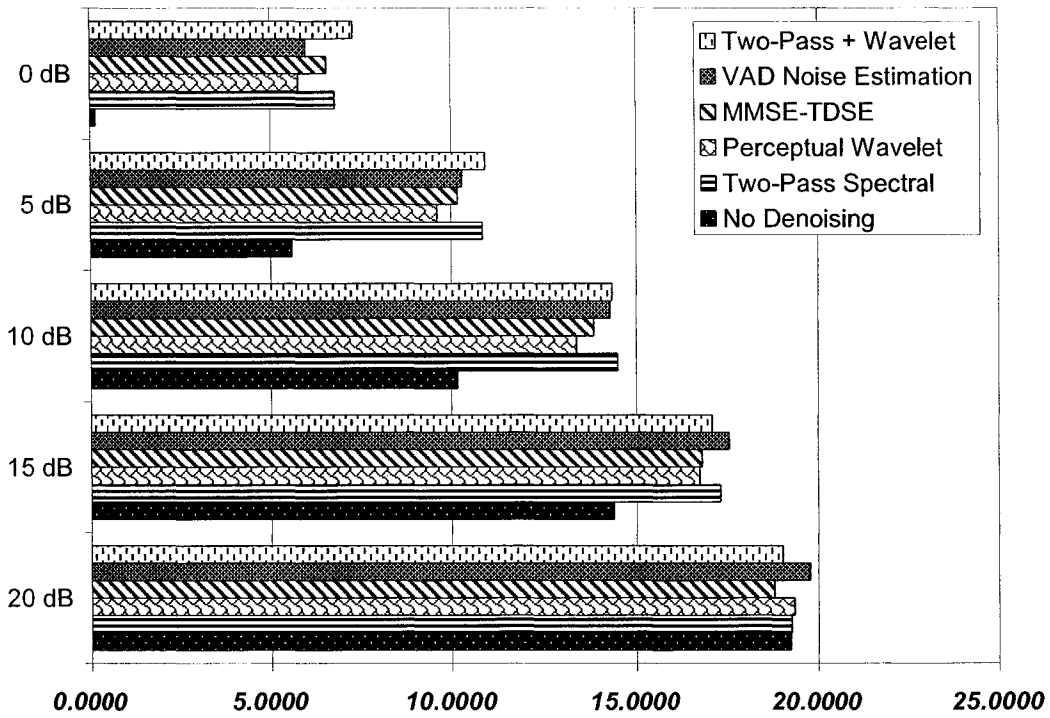

Figure A-4. Average segmental SNR with HF channel noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
HF Channel	Two-Pass Spectral	0.0275	2.9615	4.3071	5.2732	6.6611
	Perceptual Wavelet	0.1059	2.4036	3.1977	4.0427	5.6364
	MMSE-TDSE	-0.4471	2.4681	3.6592	4.6069	6.4311
	VAD Noise Estimation	0.5497	3.2013	4.1056	4.7159	5.8315
	Two-Pass + Wavelet	-0.2163	2.7456	4.1554	5.3405	7.1774

Table A-4. Average segmental SNR improvement with HF channel noise

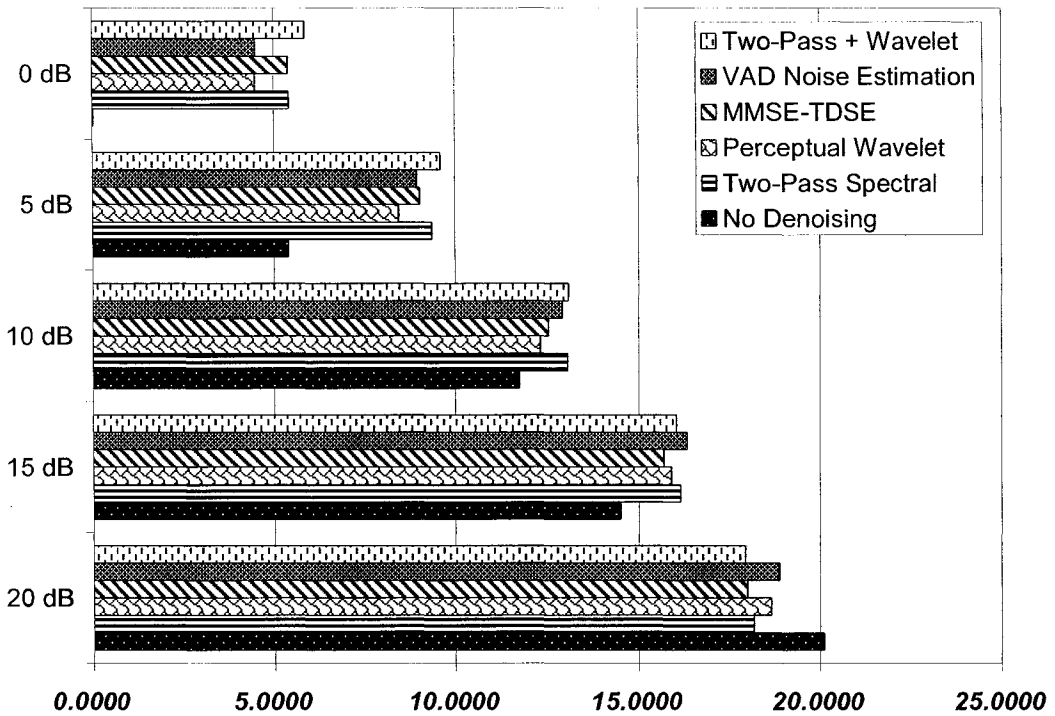

Figure A-5. Average segmental SNR with factory1 noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Factory1	Two-Pass Spectral	-1.8686	1.6703	1.3115	3.9809	5.3893
	Perceptual Wavelet	-1.4029	1.4034	0.5713	3.0635	4.4948
	MMSE-TDSE	-2.0373	1.1947	0.8022	3.6516	5.3749
	VAD Noise Estimation	-1.1855	1.8690	1.1807	3.5796	4.4937
	Two-Pass + Wavelet	-2.1046	1.5556	1.3504	4.2097	5.8205

Table A-5. Average segmental SNR improvement with factory1 noise

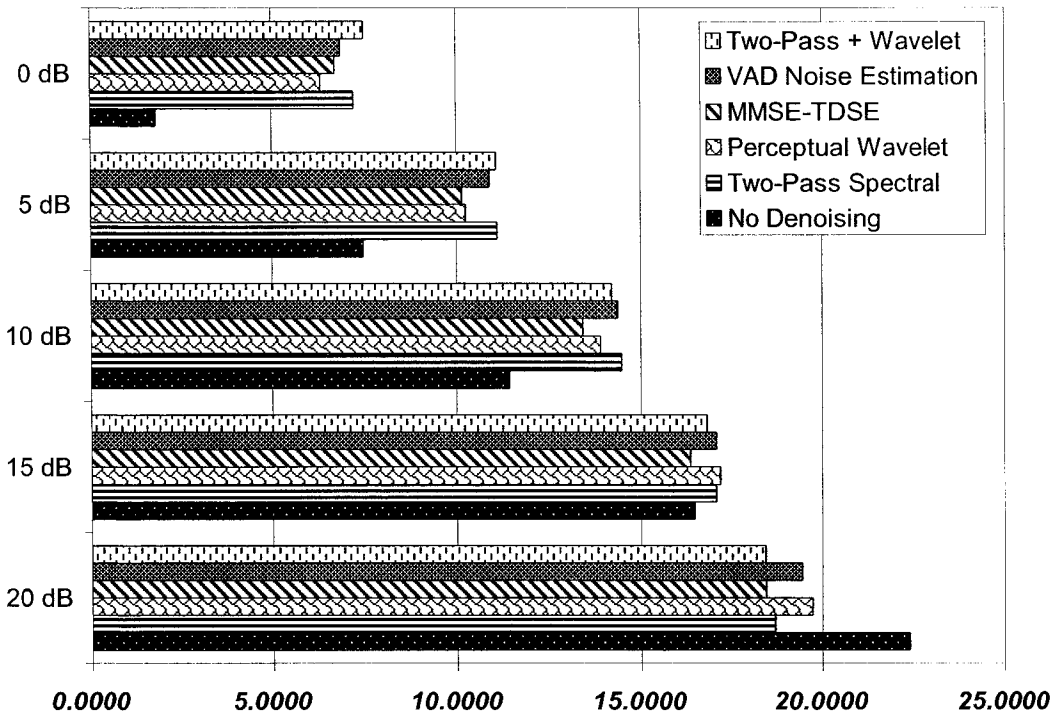

Figure A-6. Average segmental SNR with babble noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Babble	Two-Pass Spectral	-3.6472	0.6196	3.0524	3.6467	5.4214
	Perceptual Wavelet	-2.6323	0.7402	2.4928	2.7935	4.5274
	MMSE-TDSE	-3.8830	-0.1016	2.0130	2.6870	4.9161
	VAD Noise Estimation	-2.9085	0.6305	2.9487	3.4361	5.0727
	Two-Pass + Wavelet	-3.8984	0.3604	2.7986	3.6200	5.7160

Table A-6. Average segmental SNR improvement with babble noise

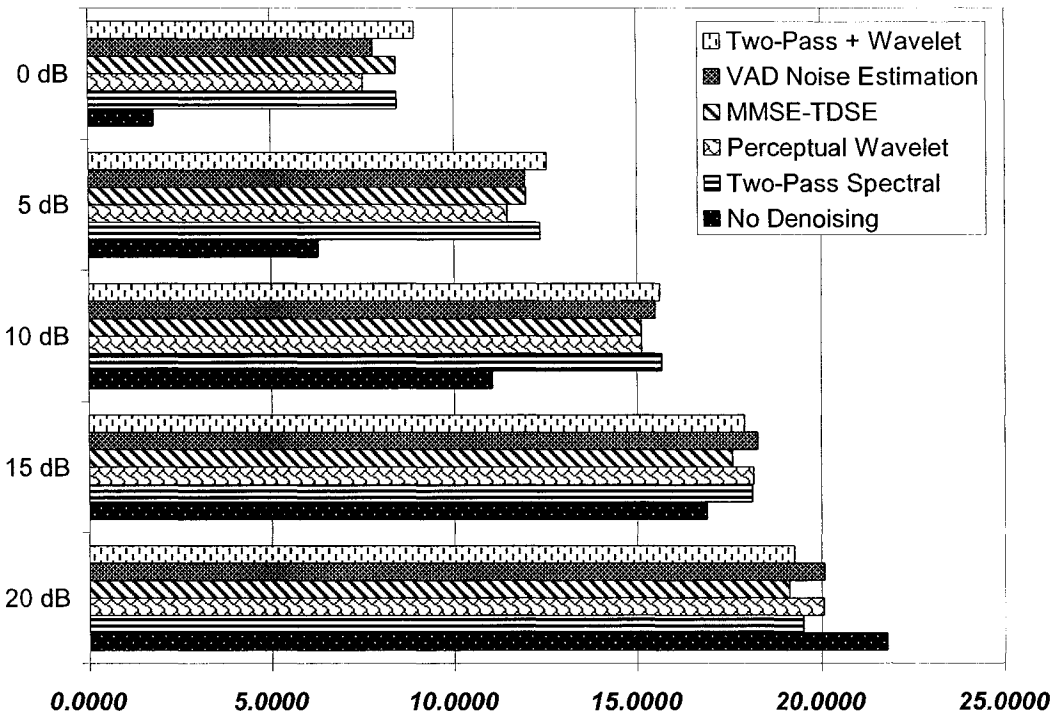

Figure A-7. Average segmental SNR with F16 noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
F16	Two-Pass Spectral	-2.2745	1.2086	4.6248	6.0576	6.6666
	Perceptual Wavelet	-1.7164	1.2372	4.0795	5.1875	5.7547
	MMSE-TDSE	-2.6724	0.6721	4.0746	5.6837	6.6429
	VAD Noise Estimation	-1.6919	1.3604	4.4605	5.6514	6.0117
	Two-Pass + Wavelet	-2.5272	1.0119	4.5675	6.2413	7.1418

Table A-7. Average segmental SNR improvement with F16 noise

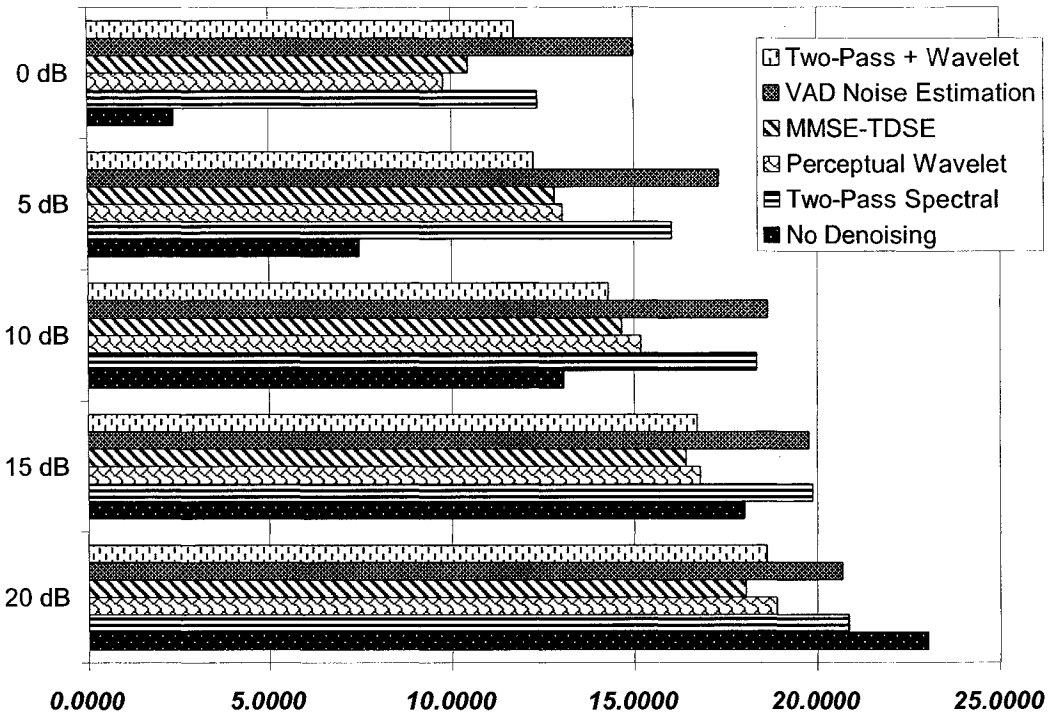

Figure A-8. Average segmental SNR with machine gun noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Machine Gun	Two-Pass Spectral	-2.1537	1.8801	5.2447	8.5409	10.0000
	Perceptual Wavelet	-4.1828	-1.1689	2.1140	5.5445	7.3930
	MMSE-TDSE	-5.0117	-1.5492	1.5829	5.3196	8.0814
	VAD Noise Estimation	-2.3144	1.7770	5.5431	9.8229	12.5986
	Two-Pass + Wavelet	-4.4558	-1.2510	1.2017	4.7598	9.3786

Table A-8. Average segmental SNR improvement with machine gun noise

APPENDIX B

This appendix displays the PESQ-MOS results of each algorithm for comparison.

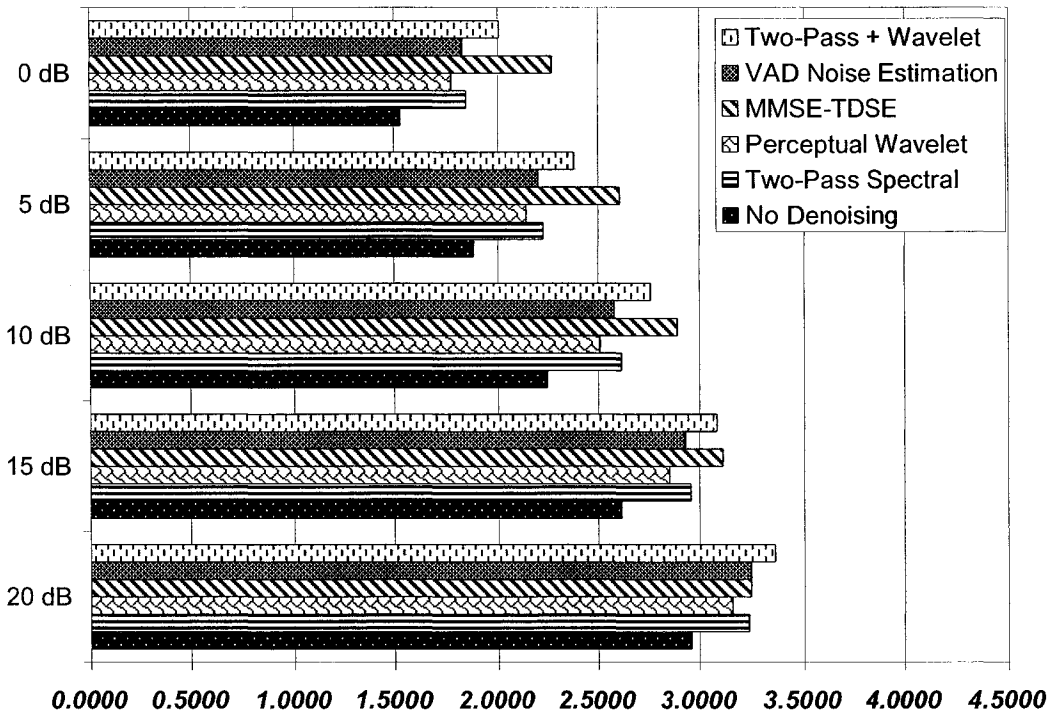

Figure B-1. PESQ-MOS subjective score with white noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
White	Two-Pass Spectral	0.2795	0.3443	0.3641	0.3449	0.3162
	Perceptual Wavelet	0.1986	0.2359	0.2598	0.2605	0.2428
	MMSE-TDSE	0.2881	0.5030	0.6424	0.7211	0.7405
	VAD Noise Estimation	0.2879	0.3183	0.3290	0.3185	0.2960
	Two-Pass + Wavelet	0.3987	0.4729	0.5073	0.4993	0.4749

Table B-1. PESQ-MOS improvement with white noise

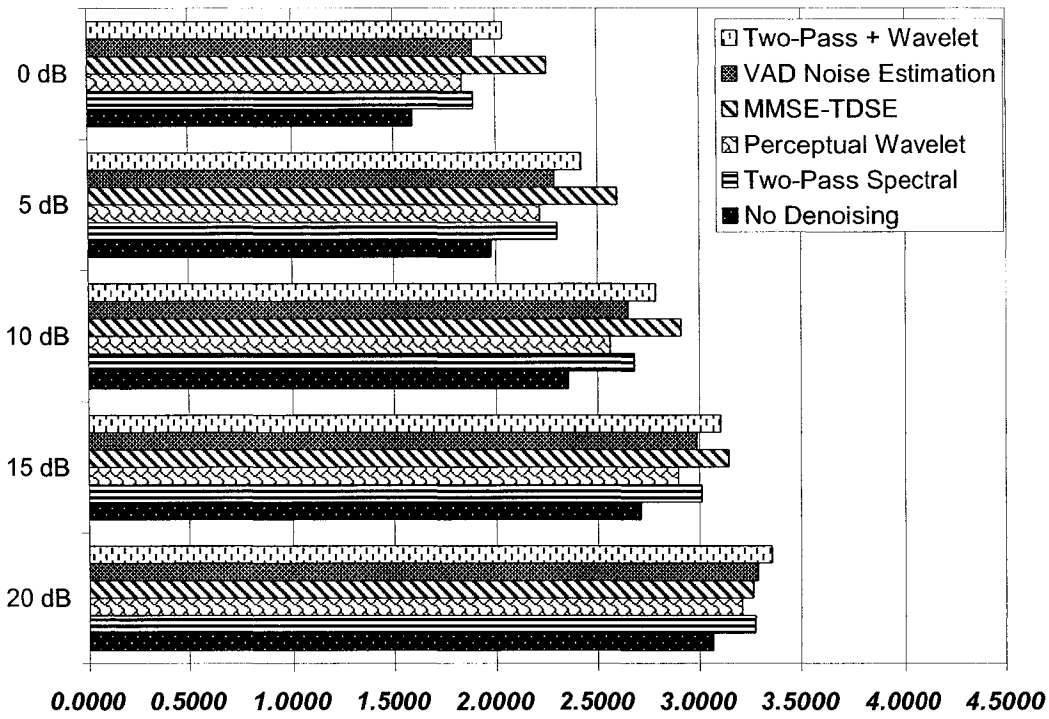

Figure B-2. PESQ-MOS subjective score with pink noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Pink	Two-Pass Spectral	0.2096	0.2953	0.3285	0.3181	0.2946
	Perceptual Wavelet	0.1421	0.1790	0.2094	0.2314	0.2424
	MMSE-TDSE	0.1991	0.4300	0.5558	0.6173	0.6465
	VAD Noise Estimation	0.2228	0.2718	0.2983	0.3053	0.2906
	Two-Pass + Wavelet	0.2894	0.3874	0.4332	0.4379	0.4346

Table B-2. PESQ-MOS improvement with pink noise

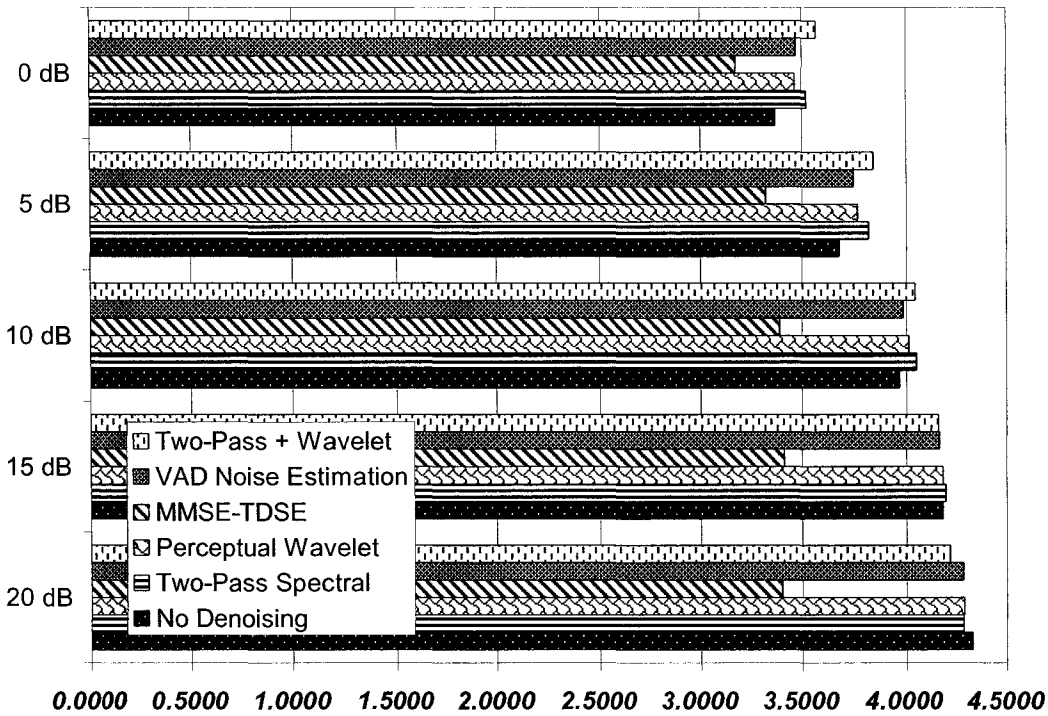

Figure B-3. PESQ-MOS subjective score with car noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Car	Two-Pass Spectral	-0.0457	0.0143	0.0795	0.1383	0.1531
	Perceptual Wavelet	-0.0422	0.0018	0.0441	0.0855	0.1003
	MMSE-TDSE	-0.9282	-0.7692	-0.5779	-0.3637	-0.1950
	VAD Noise Estimation	-0.0473	-0.0154	0.0179	0.0676	0.1040
	Two-Pass + Wavelet	-0.1141	-0.0229	0.0746	0.1623	0.2037

Table B-3. PESQ-MOS improvement with car noise

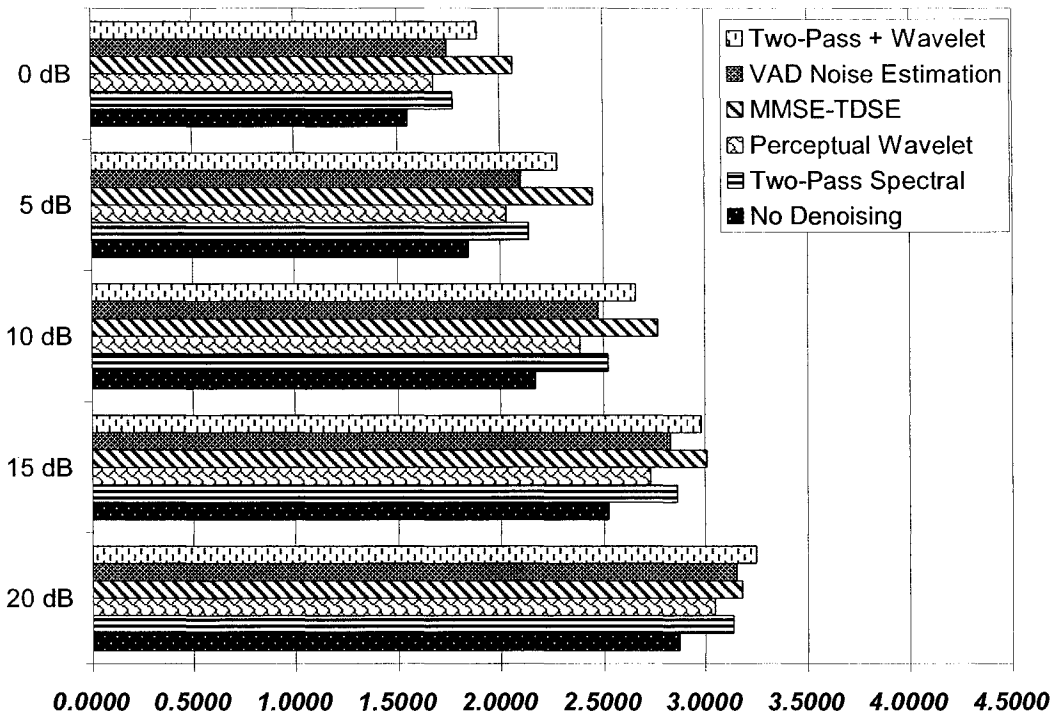

Figure B-4. PESQ-MOS subjective score with HF Channel noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
HF Channel	Two-Pass Spectral	0.2692	0.3391	0.3456	0.2999	0.2203
	Perceptual Wavelet	0.1790	0.2065	0.2110	0.1879	0.1264
	MMSE-TDSE	0.3112	0.4907	0.5889	0.6068	0.5188
	VAD Noise Estimation	0.2842	0.3082	0.2986	0.2624	0.1926
	Two-Pass + Wavelet	0.3789	0.4594	0.4797	0.4373	0.3415

Table B-4. PESQ-MOS improvement with HF Channel noise

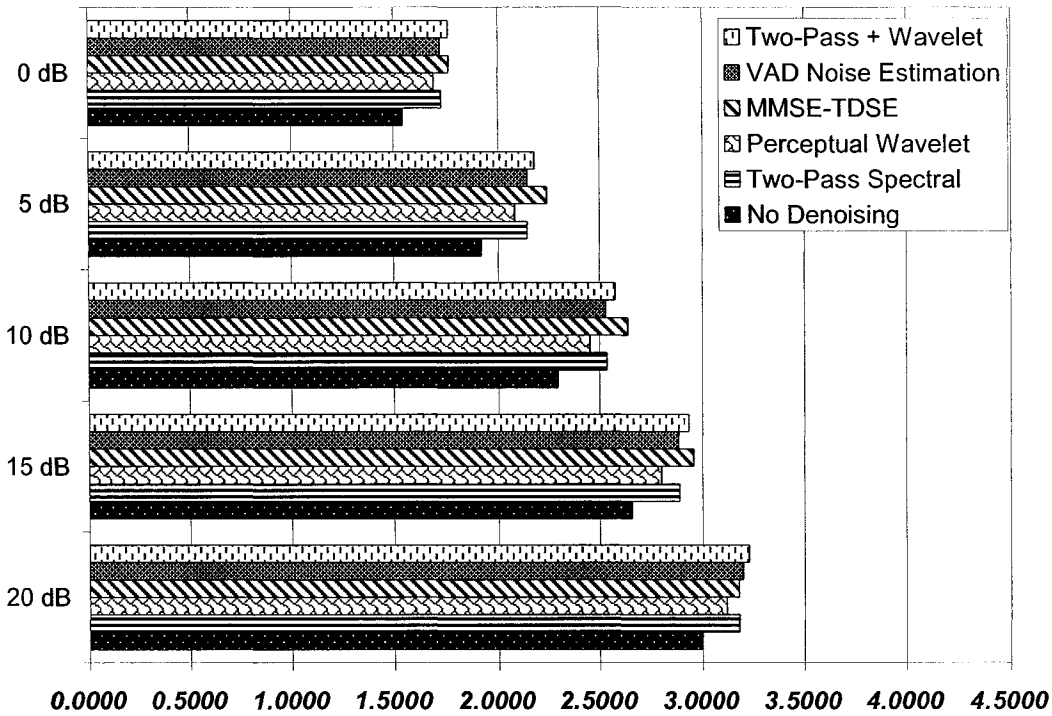

Figure B-5. PESQ-MOS subjective score with factory1 noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Factory1	Two-Pass Spectral	0.1825	0.2304	0.2397	0.2236	0.1806
	Perceptual Wavelet	0.1207	0.1411	0.1564	0.1655	0.1474
	MMSE-TDSE	0.1800	0.3013	0.3418	0.3235	0.2172
	VAD Noise Estimation	0.2006	0.2261	0.2339	0.2247	0.1767
	Two-Pass + Wavelet	0.2299	0.2767	0.2804	0.2608	0.2155

Table B-5. PESQ-MOS improvement with factory1 noise

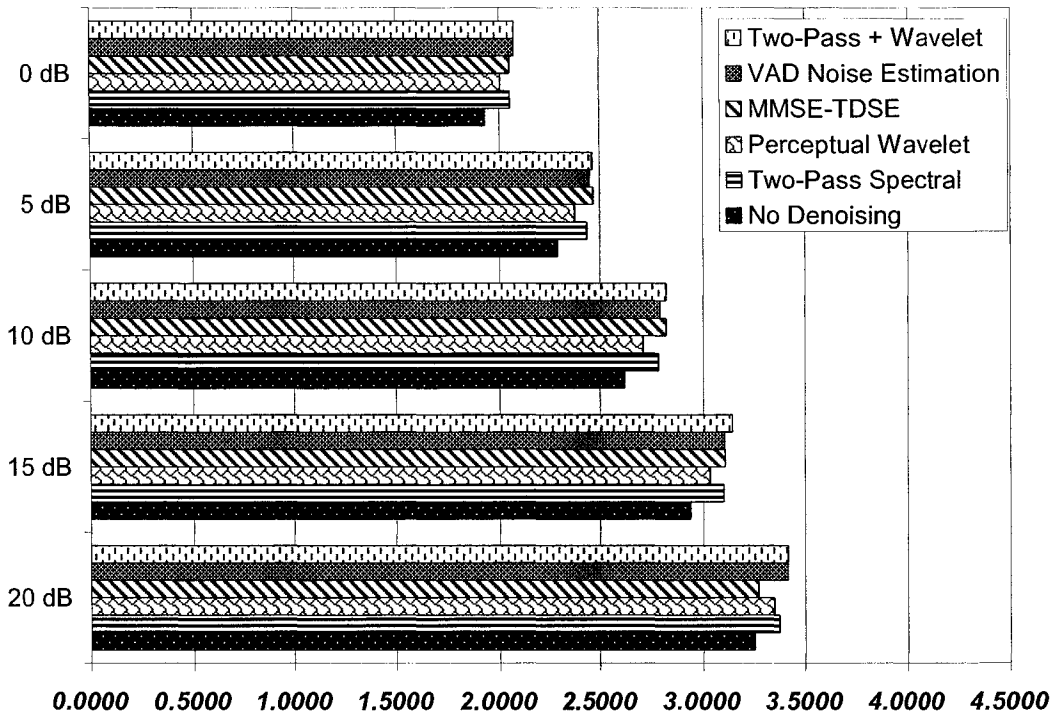

Figure B-6. PESQ-MOS subjective score with babble noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Babble	Two-Pass Spectral	0.1215	0.1596	0.1627	0.1482	0.1208
	Perceptual Wavelet	0.0965	0.0957	0.0907	0.0866	0.0760
	MMSE-TDSE	0.0190	0.1681	0.1998	0.1828	0.1210
	VAD Noise Estimation	0.1618	0.1649	0.1705	0.1641	0.1395
	Two-Pass + Wavelet	0.1640	0.2039	0.2003	0.1765	0.1420

Table B-6. PESQ-MOS improvement with babble noise

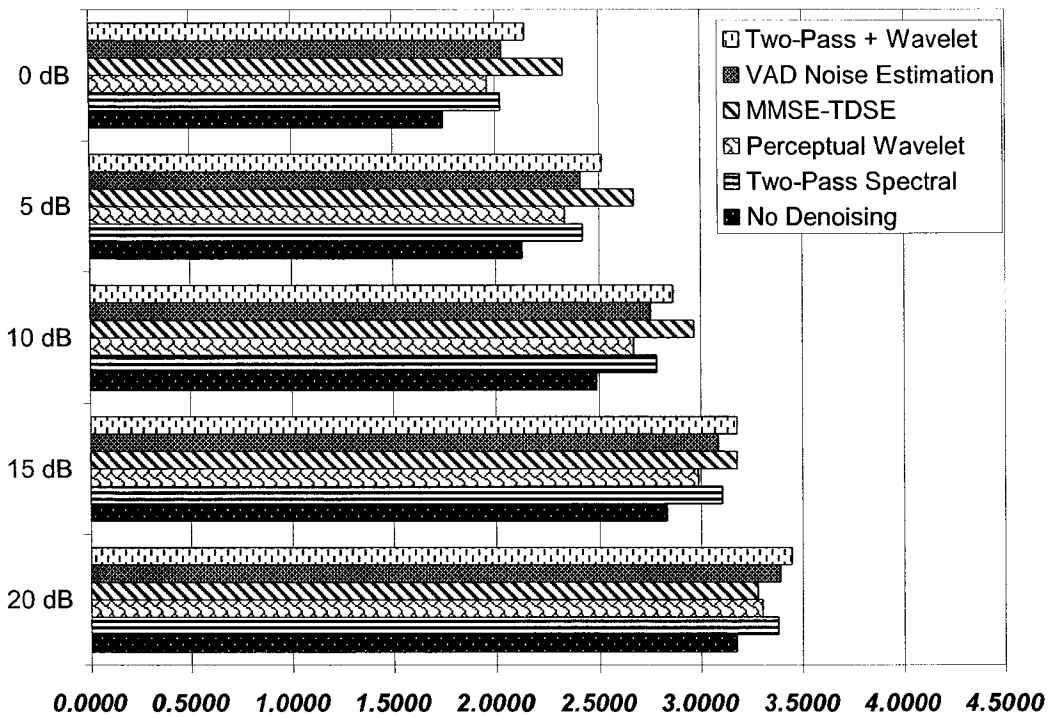

Figure B-7. PESQ-MOS subjective score with F16 noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
F16	Two-Pass Spectral	0.2059	0.2707	0.2964	0.2903	0.2744
	Perceptual Wavelet	0.1257	0.1557	0.1817	0.2023	0.2120
	MMSE-TDSE	0.1036	0.3429	0.4784	0.5440	0.5736
	VAD Noise Estimation	0.2165	0.2512	0.2668	0.2788	0.2802
	Two-Pass + Wavelet	0.2721	0.3436	0.3771	0.3861	0.3895

Table B-7. PESQ-MOS improvement with F16 noise

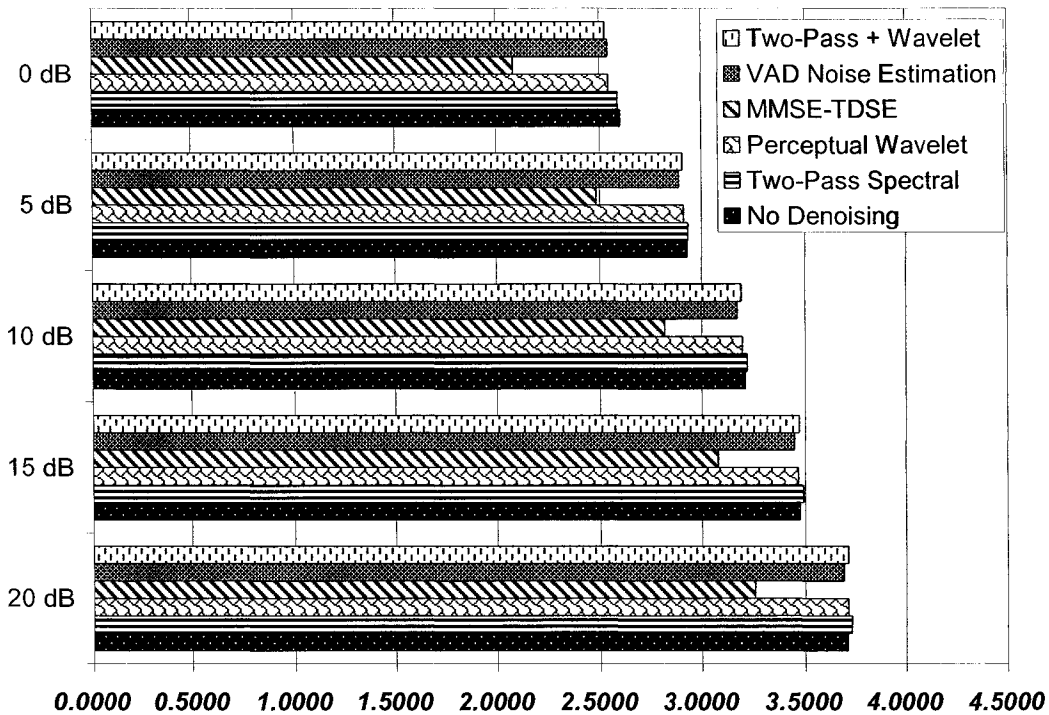

Figure B-8. PESQ-MOS subjective score with machine gun noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Machine Gun	Two-Pass Spectral	0.0203	0.0164	0.0087	0.0034	-0.0115
	Perceptual Wavelet	0.0009	-0.0073	-0.0143	-0.0172	-0.0538
	MMSE-TDSE	-0.4525	-0.3920	-0.3984	-0.4474	-0.5124
	VAD Noise Estimation	-0.0205	-0.0249	-0.0373	-0.0418	-0.0605
	Two-Pass + Wavelet	0.0006	-0.0060	-0.0184	-0.0250	-0.0727

Table B-8. PESQ-MOS improvement with machine gun noise

APPENDIX C

This appendix shows recognition accuracy of the five algorithms for various noise types at different input SNRs with and without enhancement applied.

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
White	No Denoising	63.57%	30.71%	15.00%	3.57%	5.00%
	Two-Pass Spectral	67.14%	40.71%	24.29%	17.86%	10.71%
	Perceptual Wavelet	72.14%	49.29%	29.29%	14.29%	4.29%
	MMSE-TDSE	62.86%	47.14%	22.86%	10.71%	7.86%
	VAD Noise Estimation	70.29%	42.86%	23.57%	17.14%	10.00%
	Two-Pass + Wavelet	72.86%	46.43%	24.29%	16.43%	12.86%
Pink	No Denoising	82.14%	56.43%	12.86%	5.00%	4.29%
	Two-Pass Spectral	85.71%	68.57%	42.14%	21.43%	10.00%
	Perceptual Wavelet	87.14%	70.71%	40.71%	14.29%	9.29%
	MMSE-TDSE	67.86%	67.14%	37.14%	18.57%	11.43%
	VAD Noise Estimation	85.71%	60.29%	31.43%	15.71%	7.86%
	Two-Pass + Wavelet	83.57%	56.43%	38.57%	17.86%	10.00%
Car	No Denoising	96.43%	93.57%	89.29%	75.00%	62.86%
	Two-Pass Spectral	97.86%	97.14%	93.57%	89.29%	62.86%
	Perceptual Wavelet	95.71%	94.29%	91.43%	85.71%	73.57%
	MMSE-TDSE	70.00%	95.00%	58.57%	52.86%	45.00%
	VAD Noise Estimation	97.86%	95.57%	89.29%	82.86%	72.14%
	Two-Pass + Wavelet	96.43%	93.57%	94.29%	91.43%	85.00%
HF Channel	No Denoising	94.29%	82.14%	64.29%	28.57%	15.00%
	Two-Pass Spectral	95.71%	86.43%	72.14%	53.57%	19.29%
	Perceptual Wavelet	94.29%	82.14%	65.00%	37.86%	17.86%
	MMSE-TDSE	69.29%	82.86%	55.71%	44.29%	27.86%
	VAD Noise Estimation	96.71%	85.00%	70.71%	46.43%	16.43%
	Two-Pass + Wavelet	91.43%	82.14%	69.29%	47.14%	22.14%

Table C-1. Recognition accuracy performance with & without enhancement for white, pink, car and HF channel noise.

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
Factory	No Denoising	92.14%	72.86%	31.43%	16.43%	5.00%
	Two-Pass Spectral	94.29%	80.71%	61.43%	41.43%	24.29%
	Perceptual Wavelet	91.43%	75.00%	50.00%	23.57%	12.86%
	MMSE-TDSE	62.86%	55.71%	38.57%	21.43%	14.29%
	VAD Noise Estimation	92.86%	80.00%	55.71%	32.14%	17.14%
	Two-Pass + Wavelet	85.00%	68.57%	48.57%	33.57%	18.57%
Babble	No Denoising	98.57%	96.43%	86.00%	59.29%	27.86%
	Two-Pass Spectral	93.57%	92.86%	77.14%	61.43%	39.29%
	Perceptual Wavelet	98.57%	97.14%	86.43%	62.86%	31.43%
	MMSE-TDSE	69.29%	63.57%	55.71%	45.00%	30.00%
	VAD Noise Estimation	95.00%	92.14%	81.43%	65.71%	41.43%
	Two-Pass + Wavelet	90.71%	80.71%	71.43%	54.29%	34.29%
F16	No Denoising	90.00%	75.71%	53.57%	22.14%	6.43%
	Two-Pass Spectral	94.29%	82.14%	61.43%	36.43%	20.00%
	Perceptual Wavelet	90.00%	82.86%	58.14%	28.57%	10.00%
	MMSE-TDSE	72.14%	65.00%	52.14%	39.29%	20.00%
	VAD Noise Estimation	93.57%	80.71%	60.14%	31.43%	15.71%
	Two-Pass + Wavelet	91.43%	82.14%	62.14%	42.14%	25.00%
Machine Gun	No Denoising	98.57%	98.57%	97.86%	92.14%	85.00%
	Two-Pass Spectral	98.57%	98.57%	97.14%	94.29%	87.86%
	Perceptual Wavelet	97.86%	97.15%	97.00%	87.86%	75.00%
	MMSE-TDSE	63.57%	57.14%	56.43%	55.00%	51.43%
	VAD Noise Estimation	96.71%	96.29%	95.29%	85.71%	79.29%
	Two-Pass + Wavelet	97.86%	97.14%	95.00%	89.29%	77.14%

Table C-2. Recognition accuracy performance with & without enhancement for factory, babble, F16 and machine gun noise.