2011

DESIGN AND EVALUATION OF HARMONIC SPEECH ENHANCEMENT AND BANDWIDTH EXTENSION

Arvind Venkatasubramanian

DESIGN AND EVALUATION OF HARMONIC SPEECH ENHANCEMENT AND BANDWIDTH EXTENSION

(Spine title: Speech Enhancement and Bandwidth extension)

(Thesis format: Monograph)

By

Arvind Venkatasubramanian

Graduate Program
in
Engineering Science
Department of Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor

Dr. Vijay Parsa

Examiners

Dr. Lyndon Brown

Dr. Raveendra K. Rao

Dr. Ewan Macpherson

The thesis by

Arvind Venkatasubramanian

Entitled:

DESIGN AND EVALUATION OF HARMONIC SPEECH ENHANCMENT AND BANDWIDTH EXTENSION

is accepted in partial fulfillment of the
requirements for the degree of
Master of Engineering Science

Date_____

Dr. Hanif Ladak, Chair of the Thesis Examination Board

Abstract

Improving the quality and intelligibility of speech signals continues to be an important topic in mobile communications and hearing aid applications. This thesis explored the possibilities of improving the quality of corrupted speech by cascading a log Minimum Mean Square Error (logMMSE) noise reduction system with a Harmonic Speech Enhancement (HSE) system. In HSE, an adaptive comb filter is deployed to harmonically filter the useful speech signal and suppress the noisy components to noise floor. A Bandwidth Extension (BWE) algorithm was applied to the enhanced speech for further improvements in speech quality. Performance of this algorithm combination was evaluated using objective speech quality metrics across a variety of noisy and reverberant environments. Results showed that the logMMSE and HSE combination enhanced the speech quality in any reverberant environment and in the presence of multi-talker babble. The objective improvements associated with the BWE were found to be minimal.

Keywords: speech enhancement; signal processing; harmonic; noise reduction; bandwidth extension, wide band; narrow band; spectral band replication, linear prediction coding; measurements

Acknowledgements

Thank you Western! Thanks to Dr.Vijay Parsa for providing the necessary resources, support and guidance in order to complete this thesis. Without his guidance, all this work would not have been possible. Dr. Parsa, you had been a great teacher. Thanks to the thesis panel: Dr. Lyndon Brown, Dr. Raveendra K. Rao, and Dr. Ewan Macpherson. Thanks to my parents, friends and family. Thanks to Pushpabalaji. Thanks Bala Chitappa. Thanks to Steve and Jon for answering questions and emails. Thanks to Alex Radisavljevic and Troy Giles. Your support at Zounds was very helpful. Thanks Shriram, Boon, Bala, Vijai, Bilal and all my Indian friends at western. Thanks to my class and research mates Ben, Jon, Alex, Nazanin and Julie for the chats and support. Thanks western TSA friends.

Thank you Canada for being friendly to International students like me):

Table of Contents

List of Tables

List of Figures

Nomenclature

ACF	Adaptive comb filter
ACR	Absolute categorization rating
AIR	Acoustic Impulse Response
BRIR	Binaural room Impulse Response
BWE	Bandwidth Extension
CDT	C/C++ Development tool
dBFS	Decibels below full scale
EMSR	Ephraim-Malah suppression rule
FDNFP	Frequency domain noise floor parameters
HF	High frequency
HSE	Harmonic Speech Enhancement
IDE	Integrated development environment
IPP	Intel Integrated performance primitives
IS	Itakura-Saito
LPC	Linear prediction coding
LF	Low frequency
MB	Multi band Spectral subtraction
MDCT	Modified Discrete Cosine Transform
MMSE	minimum mean square estimation
MOS	Mean opinion score
NB	Narrow band

NCA	National Center for Audiology
NLD	Non linear distortion
PESQ	Perceptual evaluation of speech quality
PSD	Power spectral density
RIR	Room Impulse Response
RT	Reverberation time
SBR	Spectral band replication
SNR	Signal to Noise ratio
SRMR	Speech to Reverberation modulation energy ratio
STSA	Short-time spectral attenuation
VAD	Voice activity detector
WB	Wide band

List of Symbols

α	Phase of clean speech
β	Weighting factor set to 0.98 in logMMSE
δ	Delta constant set to 0.025 in spectral subtraction (ACF)
ε	Undesired components of a corrupted speech (distortion and residual noise)
γ	*a posteriori SNR*
λ	Variance
μ	mean
π	Constant usually denoting Nyquist frequency
χ	Adaptive comb filter frequency gain vector
κ	Frequency domain noise floor parameters
σ	Adaptive comb filter width
τ	delay
ς	0.15 for logMMSE statistical VAD decision threshold
ν	Noisy phase
ω	Angular frequency in radians
ξ	*a priori* SNR
ψ	Factor used in Spectral flatness measure calculation in harmonic test
∞	Infinity
Φ	moment generating function
Γ	Gamma function
p	Analysis frame number
n	Time index
N	FFT size
k	Spectral bin index
ks	Spectral band index
x	Clean speech
X	Vector FFT of clean speech
$\hat{X}$	Desired signal spectrum
y	Corrupted speech
Y	Vector FFT of corrupted speech
d	Noise background
D	Vector FFT of noise
A	Spectral magnitude of clean speech
F	Fourier transform
R	Spectral magnitude of corrupted speech
G	Short time spectral amplitude estimate gain made by combination of *a priori* SNR and *a posteriori SNR*
h	System Impulse response
H	System Frequency response
S	Power spectrum
Q	Gain vector of combination of FDNFP κ and Adaptive comb filter χ
dist	Distorted measure

Chapter 1: Introduction

1.1 Need for Speech Enhancement

In all electronic speech communication applications that require at least one microphone, the signal of interest is usually contaminated by background noise and reverberation. The contaminated speech signal has a detrimental effect on its quality and understandability [14]. It is therefore desirable that the corrupted microphone signal has to be cleaned through digital signal processing tools before it is played out, transmitted, or stored. The objective of speech enhancement may be to improve the overall quality, to enhance the intelligibility, to reduce the listener fatigue, etc.

Engineers and researchers in various disciplines have shown considerable recent interest in speech enhancement. For example, considerable amount of research continues to be expended on improving speech communication over landline or wireless telecommunication channels in challenging environments through signal processing. Similarly, there is a considerable research effort from hearing aid manufacturers on speech enhancement algorithms, as understanding speech in a noisy environment is especially challenging for hearing impaired listeners [6]. In the following sections, this need for speech enhancement is further described for two types of environments, and a brief introduction to speech enhancement strategies are provided.

1.1.1 Speech Corrupted by Noise

Noise is omnipresent in this world. For example, environments such as offices, streets, restaurants, trains, exhibitions, airports and motor vehicles are replete with interfering background noise, and this interfering noise degrades the intelligibility and quality of speech. Two categories of algorithms can be recognized when it comes to enhancing speech corrupted by background noise. The first category uses multiple microphones during the signal acquisition stage (i.e. adaptive microphone arrays) while the second uses only a single microphone. The multiple microphone processing exploits any spatial separation between the desired speech and the interfering noise sources [48]. In the second category, single microphone speech enhancement attempts to extract speech from a single acoustic mixture of speech and the background noise and is therefore a much more challenging problem.

During the past few decades, several single microphone digital signal processing strategies have been put forward to cleanse the noisy speech samples. Loizou [14] provides a comprehensive review of these algorithms and broadly groups them into (a) spectral-subtractive algorithms, (b) Wiener filtering algorithms, (c) statistical model-based algorithms, and (d) subspace algorithms. Briefly, spectral subtractive algorithms attempt to estimate the background noise spectrum and remove it from the noisy speech spectrum. Wiener filtering algorithms design and apply an optimum filter that minimizes the mean-squared error between output and desired signal. Statistical model-based algorithms employ statistical methods in estimating and enhancing the speech spectrum. Finally, subspace algorithms attempt to decompose the noisy signal into signal and noise subspaces and subsequently nullify the noise subspace. Hu and Loizou [12] compared

the performance of these different classes of algorithms and concluded that the statistical model-based algorithms performed the best.

In addition to these four classes, another class of speech enhancement algorithms exists which exploits the harmonic nature of speech components [2]. In particular, these algorithms identify and enhance the harmonic portion of the noisy speech spectrum, while suppressing non-harmonic portions to noise floor. This thesis investigates the performance of a statistical model-based algorithm and the harmonic speech enhancement algorithm, both in isolation and as a combination across a number of noisy environments.

1.1.2 Speech Corrupted by Reverberation

Reverberation is one of the primary factors that degrade the quality of speech when collected by a distant microphone. Reverberation is caused by the fact that the microphone not only picks up the direct transmission of the signal, but also its reflections [41]. When speech signals are obtained in an enclosed space by one or more microphones positioned at a distance from the talker, the observed signal consists of a superposition of many delayed and attenuated copies of the speech signal due to multiple reflections from the surrounding walls and other objects [45]. The perceptual effects of reverberation can be a box effect in which the reverberated speech signal can be viewed as the same source signal coming from several different sources positioned at different locations in the room and thus arriving at different times and with different intensities. This adds spaciousness to the sound and makes the talker sound as if positioned

inside a box or as the distant talker effect where the perceived spaciousness explained in the previous point makes the talker sound far away from the microphone [45].

In cases of excessive reverberation, intelligibility of speech is degraded. Reverberation alters the characteristics of the speech signal, and changes the shape of time domain signal which is problematic for signal processing applications including speech recognition, source localization speaker identification, and channel communication. As the distance between the talker and the microphones is increased, the reverberation effects become harmful to target signal processing schemes which do not take room effects into consideration.

Similar to noise abatement discussed in previous section, several signal processing strategies have been put forth to combat reverberation [45]. These include spectral subtraction algorithms, Wiener filtering algorithms, and dereverberation algorithms based on harmonicity and speech production models. This thesis investigates the performance of a statistical model-based algorithm and the harmonic speech enhancement algorithm in diminishing the deleterious effects of reverberation on speech quality.

1.2 Bandwidth extension

Bandwidth extension (BWE) refers to methods that increase the frequency spectrum, or bandwidth of electronic signals. Such frequency extension is desirable if at some point the frequency content of the signal has been reduced, as can happen, for example, during recording, transmission (including storage), or reproduction, mostly because of economical constraints.

Examples of situations in which bandwidth reduction occurs are telephony, perceptual audio coding (at low bit rates), and sound reproduction with non-ideal transducers [9]. As a further example, traditional landline and cellular communications utilize a bandwidth of 300 – 3400 Hz for transmitting speech [9]. This implies that if the speech input has significant spectral content beyond 3400 Hz, it will be filtered out by the telecommunication network. Although adequate for speech communication, this narrow bandwidth of 300 – 3400 Hz has an impact on both the quality and intelligibility of transmitted speech. For specific components of speech the effect of bandwidth is even more dramatic. Fricatives such as /s/, /sh/, /f/, whose spectral energy extends beyond 3400 Hz, are more affected by the bandwidth. Stelmachowicz et al. [8] showed that normal hearing adults achieved only 33% accuracy in identifying the /s/ phoneme spoken by a female talker when the bandwidth was 5 kHz and this improved to 80% with a bandwidth increase to 6 kHz. In addition to the increase in intelligibility, wider bandwidth is also associated with increased "brightness", "naturalness", and overall quality of speech [16]. For example, Moore and Tan [69] demonstrated a decrease in sound quality for speech signal when the bandwidth was less than 10.8 kHz.

Several algorithms exist for extending the bandwidth of the speech signal, and these can be classified into (a) blind algorithms which reconstruct the high frequency portion of the speech spectrum from lower frequency content, and (b) algorithms which exploit some *a priori* knowledge of the high frequency spectral information. In this thesis, a blind bandwidth extension algorithm is employed in conjunction with noise reduction or dereverberation algorithm.

1.3 Speech Quality Measurements

As mentioned in the previous sections, several choices exist for noise reduction, dereverberation, and bandwidth extension. This necessitates a method for benchmarking the effect of various candidate algorithms on speech quality and intelligibility. This thesis concentrates on the speech quality attributes and a brief discussion of the measurement of speech quality is given below.

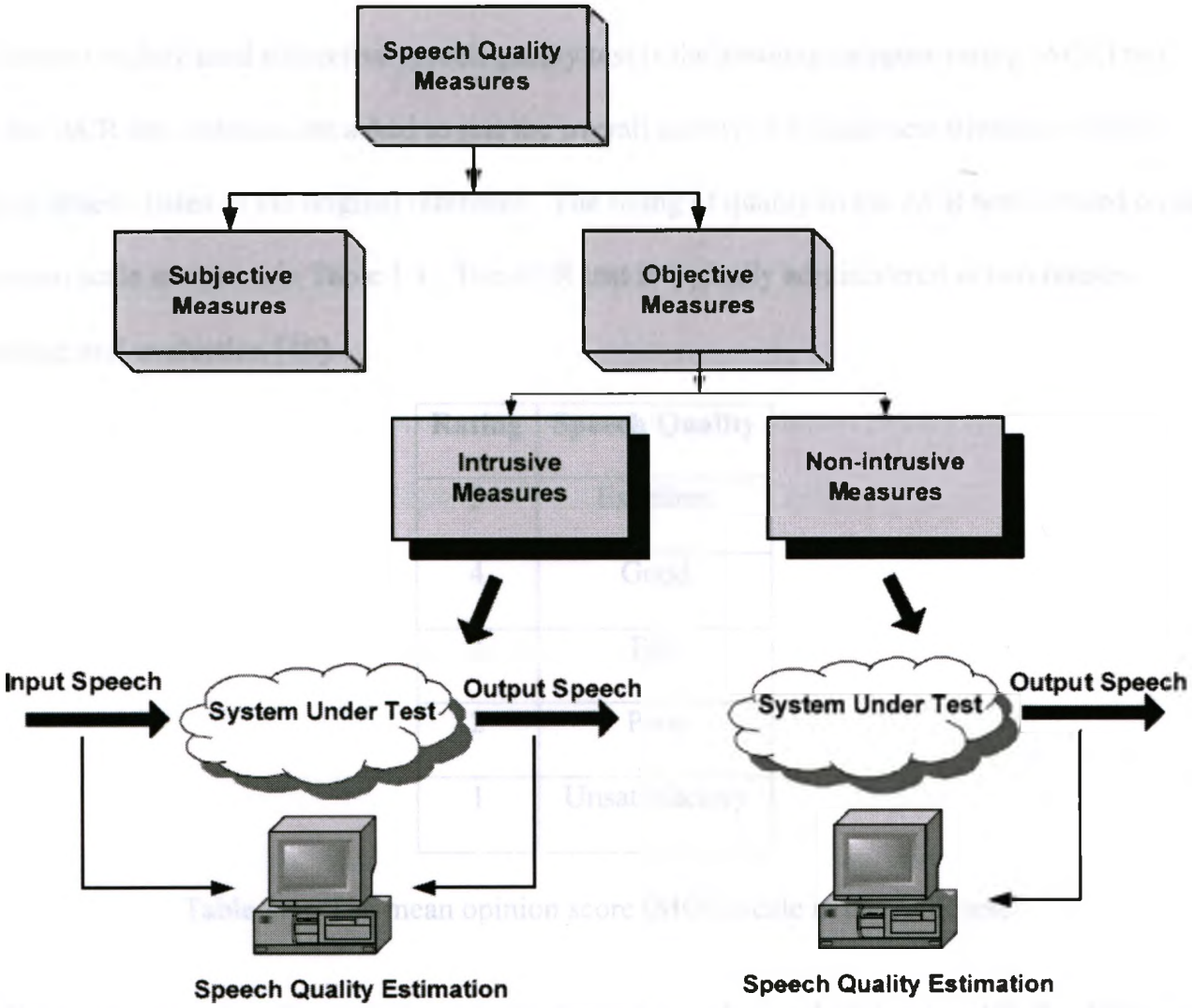

Figure 1.1: The classification of speech quality measurement [40]

In general, speech quality can be measured through subjective means or through instrumental objective measurements, as shown in Figure 1.1. Subjective measures assess speech quality by having people listen and respond the quality perception. On the other hand, objective measures assess speech quality through the use of physical characteristics of the speech signal and appropriate computational models. Normally, the performance of an objective measure is usually evaluated by calculating the degree of correlation between the objective measure and the subjective quality scores [40].

The most widely used subjective speech quality test is the absolute category rating (ACR) test. In the ACR test, subjects are asked to rate the overall quality of a single test stimulus without being able to listen to the original reference. The rating of quality in the ACR test is based on an opinion scale as shown in Table 1.1. The ACR test is typically administered in two phases: training and evaluation [40].

Rating	Speech Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Unsatisfactory

Table 1.1: The mean opinion score (MOS) scale in the ACR test

In the training phase, subjects hear a set of reference speech signals that exemplify the different quality categories. This process is meant to give all listeners the same subjective range and

origin in their quality ratings. In the evaluation phase, subjects listen to the speech under test and rate each sample in terms of the quality categories. The average of opinion scores of the subjects gives the Mean Opinion Score (MOS) [40].

Objective measures are generally divided into intrusive and non-intrusive measures. The intrusive measures assess speech quality by comparing known, controlled test signals and the corresponding outputs of the system under test. In the intrusive measures, both the original and processed signals are available for the computational models [40]. The original input signal is assumed to be of perfect or near-to perfect quality, and the computed differences between the original and processed signal are mapped into a speech quality score. In order for this difference to result in a meaningful metric, it is crucial to synchronize the input and output signals of the system under test. Otherwise, the calculation of the difference between these two signals does not reliably convey the necessary information. Another type of objective quality evaluation, that is, non-intrusive measures, circumvents this need for synchronization as they estimate speech quality based only on the output signal of the system under test [40].

While subjective tests are "gold standard" and are attractive for high face validity, they are also time and cost consuming. Objective measures, on the other hand, are time and resource efficient, and are preferable for rapid benchmarking of different algorithms. Objective metrics that have high degree of correlation with subjective scores are attractive, as they can be used as substitute for lengthy subjective tests. In this thesis, validated intrusive and non-intrusive measures are utilized to evaluate the performance of noise reduction, dereverberation, and bandwidth extension algorithms.

1.4 Thesis Objectives: Speech Enhancement and Bandwidth Extension

Speech corrupted by background noise and/or reverberation has severe impact on perceived quality and intelligibility. In addition, the reduced bandwidth associated with narrowband telephony and non-ideal transducers may further compromise speech quality. To address these issues, this thesis proposes and evaluates the performance of a cascaded set of algorithms that aim to reduce noise, mitigate reverberation, and extend the bandwidth of corrupted input speech. Figure 1.2 shows the proposed system architecture in which a statistical model-based speech enhancement algorithm is cascaded with the harmonic speech enhancement algorithm, the output of which is given to a bandwidth extension algorithm. The system designed for this project will be eventually ported to a Linux based hand held device that would be used by hearing impaired listeners for evaluating hearing aid digital signal processing (DSP) algorithms.

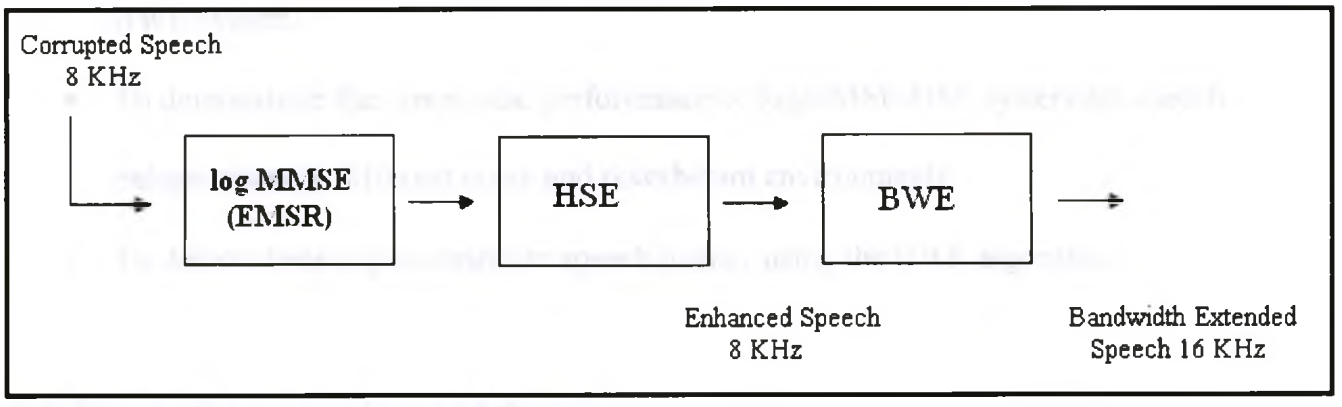

Figure 1.2: System block diagram

The input to the proposed system is corrupted speech samples acquired at a sampling rate of 8 kHz. The Input samples to the system in Figure 1.2 can be either corrupted by noise or by reverberation but not both of them simultaneously. The corrupted speech is cleaned using the

log MMSE (a statistical model-based algorithm) and the Harmonic speech enhancement (HSE) system. The output of this sub-system also operates at a sample rate of 8 kHz. The enhanced speech is served as an input to the BWE system. The BWE system replicates the low frequency spectrum to the high frequency region and makes spectral adjustments to make the sound natural. The output of the entire system now operates at a sample rate of 16 kHz, which implies that the highest frequency of the signal has changed from 4 kHz in the input part to 8 kHz in the output portion. If there are artifacts in the HSE output, the BWE output may have replication of those artifacts. If the output of the HSE has low SNR, the same effects will be seen in the BWE output. The task of the project is to create a HSE output with less perceivable artifacts so that BWE is able to give an output that has considerably fewer perceived artifacts and a feasible signal to noise ratio (SNR). In summary, here is a list of main objectives of this thesis:

- To design a real-time capable logMMSE plus HSE system for speech enhancement and BWE system.
- To demonstrate the synergistic performance of logMMSE-HSE system for speech enhancement in different noisy and reverberant environments.
- To demonstrate improvement in speech quality using the BWE algorithm.

1.5 Thesis Organization and Scope

Chapter 1 provided a brief overview of need for speech enhancement and bandwidth extension in enhancing speech, a description of different speech enhancement algorithms, and methods to benchmark their performance. Chapter 2 reviews the literature on bandwidth extension using

several methods like linear prediction coding (LPC), and spectral band replication (SBR). It also covers speech enhancement techniques such as the Ephraim-Malah noise suppression, Harmonic speech enhancement (HSE.) and a review of objective speech quality measures. Chapter 3 covers implementation details of the speech enhancement algorithms covered in this project. HSE and BWE implementation details are provided and algorithm's performance against the EMSR algorithms is discussed. In Chapter 4, objective results are provided for HSE in comparison with the EMSR algorithms for speech corrupted by noise and reverberation. Objective results for BWE are also analyzed in this Chapter. Finally, a summary of the main contribution of this thesis, along with a discussion of future research directions are presented in Chapter 5.

Chapter 2: Literature Review

2.1 Speech Enhancement

The target of speech enhancement is to improve speech quality by using various algorithms. The benefit of speech enhancement is possible improvement in intelligibility and/or overall perceptual quality of a degraded speech signal using audio digital signal processing techniques. Enhancing of speech degraded by noise, or reverberation is of interest to engineers and used for many applications such as mobile phones, speech recognition, and hearing aids.

Single-channel speech enhancement degraded by additive noise has been studied with much interest in the past. Some techniques are prescribed in [19]-[30] to use the harmonics of voiced speech for enhancing the speech quality. In [19] and [20], speech is modeled as harmonic components plus noise-like components, and enhancement is performed by modeling the harmonic components thereby reducing the additive noise components. A hidden Markov model [21] minimum mean square error (MMSE) estimator is extended to enhance the harmonics for voiced speech. The sinusoidal model is adopted in the speech enhancement context in the algorithms of [22]–[26]. Adaptive comb filtering techniques used in [32] and [33] are used to improve the quality of voiced speech by post-enhancing the harmonic structures. In [27] fundamental frequency is used to narrow down the *a priori* probability distribution (PD) of the DFT amplitude to improve the estimation of the DFT spectrum and enhance the harmonics of speech.

The spectral subtraction algorithm is used widely in speech enhancement [31]. A noise corrupted speech signal $y(n)$ is composed of clean speech signal $x(n)$ and noise $d(n)$. The noise and the clean speech are assumed to be independent and uncorrelated. The spectrum of the noise signal $D(\omega)$ obtained by Fourier transform is subtracted from corrupted speech spectrum to attain clean speech spectrum. The clean speech spectrum is reconstructed to a voltage signal back in the time domain signal using the inverse Fourier transform.

$$y(n) = x(n) + d(n) \tag{2.1}$$

$$X(\omega) = Y(\omega) - D(\omega) \tag{2.2}$$

The method of spectral subtraction presented by Boll in [36] uses a short-time Fourier transform to compute magnitude, subtract bias from the noise estimate, and do a half wave rectification to avoid negative magnitude spectrum owing to errors in estimating noise spectrum in order to reduce noise residual. It also employs a voice activity detector (VAD) that attenuates the noisy signal during non-presence of speech. The non-linear processing of negative values during half-wave rectification creates small isolated peaks in the spectrum. When converted to a time domain signal, these peaks sound similar to tones with frequencies that change randomly from frame to frame; that is, tones that are turned on and off at the analysis frame rate. This type of noise introduced by the half-wave rectification process has a warbling sound along with a tone like quality, and is commonly referred to in literature as '*musical noise*'. Musical noises can be more annoying than the actual background noises like babble noise or street noise. At low SNR's, the noisy phase, after the synthesis can lead to roughness in perceived speech signal. This problem is insignificant when SNR's are high [14]. In Figure 2.1, the top picture shows a noisy spectrogram and the bottom pictures shows a processed clean spectrogram with isolated spectral peaks in the spectrum that contribute to the musical noise phenomenon.

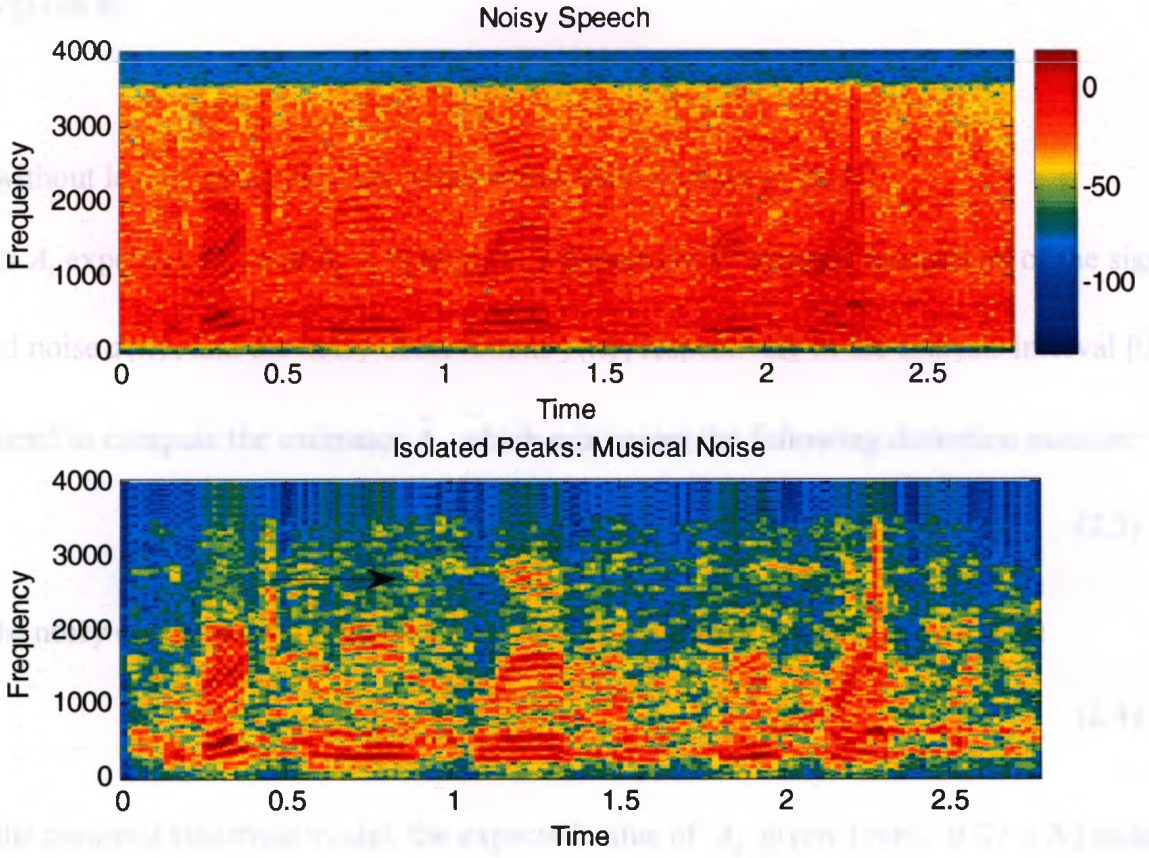

Figure 2.1: Arrows point to isolated spectral peaks that cause musical noise

Ephraim introduced an MMSE estimator to enhance speech in [34], and a log MMSE estimator in [35] to enhance speech to overcome the musical noise phenomenon. Indeed, in a comparative study of several different speech enhancement algorithms, Hu and Loizou [12] showed that the logMMSE algorithm provided the most consistent performance across a number of noisy environments. As such, a detailed description of the logMMSE technique is presented below.

2.1.1 Principles: Ephraim-Malah Suppression Rule (EMSR) log MMSE

Let *x(n)* and *d(n)* denote the speech and the noise processes, respectively. The observed signal $y(n)$ is given by

$$y(n) = x(n) + d(n), \qquad 0 \le n \le N$$

where without loss of generality, the observation interval is set to [0, N].

Let $X_k \cong A_k \exp(j\alpha_k), D_k$, and $Y_k \cong R_k \exp(j\nu_k)$ denote the k^{th} spectral component of the signal $x(n)$ and noise $d(n)$, and the noisy observations $y(n)$, respectively in the analysis interval [0, N].

It is desired to compute the estimator $\hat{A}_k$ which minimizes the following distortion measure:

$$E\{(\log A_k - \log \hat{A}_k)^2\} \tag{2.3}$$

given the noisy observations $\{y(n),\ 0 \le n \le N\}$. This estimator is easily shown to be

$$\hat{A}_k = \exp\{E[\ln A_k \mid y(n)]\}, \qquad 0 \le n \le N\} \tag{2.4}$$

Under the assumed statistical model, the expected value of A_k given $\{y(n),\ 0 \le t \le N\}$ equals to the expected value of A_k, given Y_k only. Since this statement remains true when A_k is replaced by $\ln A_k$, the estimator Eq. (2.4) equals

$$\hat{A}_k = \exp\{E[\ln A_k \mid Y_k]\} \tag{2.5}$$

Let $Z_k = \ln A_k$; Then the moment generating function $\Phi_{Z_k|Y_k}(\mu)$ of Z_k given Y_k

$$\Phi_{Z_k|Y_k}(\mu) = E\{\exp(\mu Z_k) \mid Y_k\} = E\{A^{\mu}{}_k \mid Y_k\} \tag{2.6}$$

$E\{\ln A_k \mid Y_k\}$ is obtained from $\Phi_{Z_k|Y_k}(\mu)$ by

$$E\{\ln A_k \mid Y_k\} = \frac{d}{d\mu}\Phi_{Z_k|Y_k}(\mu)\,|_{\mu=0} \tag{2.7}$$

From (2.6) $\Phi_{Z_k|Y_k}(\mu)$ is given by

$$\Phi_{Z_k|Y_k}(\mu) = E\{A_k^{\mu} \mid Y_k\} = \frac{\int_0^{\infty}\int_0^{2\pi} a_k^{\mu} p(Y_k \mid a_k,\alpha_k) p(a_k,\alpha_k)\, d\alpha_k da_k}{\int_0^{\infty}\int_0^{2\pi} p(Y_k \mid a_k,\alpha_k) p(a_k,\alpha_k)\, d\alpha_k da_k} \tag{2.8}$$

On the basis of the Gaussian model $p(Y_k \mid a_k, \alpha_k)$ and $p(a_k, \alpha_k)$ are given by

$$p(Y_k \mid a_k, \alpha_k) = \frac{1}{\pi\lambda_d(k)} \exp\{-\frac{1}{\lambda_d(k)} \mid Y_k - a_k^{ejak} \mid^2\} \tag{2.9}$$

$$p(a_k, \alpha_k) = \frac{a_k}{\pi\lambda_x(k)} \exp\{-\frac{a_k^2}{\lambda_x(k)}\} \tag{2.10}$$

where $\lambda_x(k) \cong E\{\mid X_k^2 \mid\}$ & $\lambda_d(k) \cong E\{\mid D_k^2 \mid\}$ are the variances of the k^{th} spectral component of the speech and noise respectively. On substituting (2.9) and (2.10) into (2.8), and using integral representation of the modified Bessel function of zero order $I_0(.)$ we obtain

$$\Phi_{Z_k|Y_k}(\mu) = \frac{\int_0^{\infty} a_k^{\mu+1} \exp(-a_k^2/\lambda_k)\, I_0(2a_k\sqrt{v_k/\lambda_k})\, da_k}{\int_0^{\infty} a_k \exp(-a_k^2/\lambda_k)\, I_0(2a_k\sqrt{v_k/\lambda_k})\, da_k} \tag{2.11}$$

where λ_k satisfies the following relation

$$\frac{1}{\lambda_k} = \frac{1}{\lambda_x(k)} + \frac{1}{\lambda_d(k)} \tag{2.12}$$

and v_k is defined by

$$v_k \cong \frac{\xi_k}{1+\xi_k}\,\gamma_k; \quad \xi_k \cong \frac{\lambda_x(k)}{\lambda_d(k)}; \quad \gamma_k \cong \frac{R_k^2}{\lambda_d(k)} \tag{2.13}$$

where the terms ξ_k and γ_k are referred to as the *a priori* and *a posteriori* SNRs. The integrals in (2.11) are evaluated to get

$$\Phi_{Z_k|Y_k}(\mu) = \mu_k^{\mu/2}\ \Gamma(\frac{\mu}{2}+1)\ M(-\mu/2;\ 1;\ -v_k) \tag{2.14}$$

where $\Gamma(.)$ is the gamma function and M(a; c; x) is the confluent hypergeometric function. Note that $\Phi_{Z_k|Y_k}(\mu)$ is the formula of the μ^{th} moment of a Rician random variable. The derivative of $\Phi_{Z_k|Y_k}(\mu)$ with respect to μ is obtained as follows. First, we note that $M(a;c;x)$ is defined by

$$M(a;c;x) = \sum_{r=0}^{\infty} \frac{(a)_r}{(c)_r}\frac{x^y}{r!} \tag{2.15}$$

where $(a)_r \triangleq 1.a.(a+1),\ldots(a+r-1)$, *and* $(a)_0 \triangleq 1$. $M(-\mu/2;\ 1;\ -v_k)$ can be differentiated term by term for $|\mu|<2$ since the series of its derivatives converges uniformly on that interval. The derivative of $M(-\mu/2;\ 1;\ -v_k)$ at $\mu=0$ is obtained by the above way and it equals

$$\frac{\partial}{\partial\mu}M(-\mu/2;\ 1;\ -v_k)|_{\mu=0} = -\frac{1}{2}\sum_{r=1}^{\infty}\frac{(-v)^r}{r!}\frac{1}{r}. \tag{2.16}$$

The derivative of $\Gamma(\mu/2+1)$ is conveniently obtained through the derivative of $\ln\Gamma(\mu/2+1)$ by using

$$\frac{\partial}{\partial\mu}\Gamma(\frac{\mu}{2}+1) = \Gamma(\frac{\mu}{2}+1)\ \frac{\partial}{\partial\mu}\ln\ \Gamma(\frac{\mu}{2}+1) \tag{2.17}$$

The derivative of $\ln\Gamma(\mu/2+1)$ is obtained by utilizing its series expansion given by

$$\Gamma(\frac{\mu}{2}+1) = -c\frac{\mu}{2} + \sum_{r=2}^{\infty}\frac{(-\mu)^r}{2_r^r}\alpha_r \quad |\mu|<2 \tag{2.18}$$

where

$$\alpha_r \triangleq \sum_{n=1}^{\infty} \frac{1}{n^r}$$

and c = 0.5772156649 is the Euler constant. Differentiating (2.18) term by term, and using (2.17) gives

$$\frac{d}{d\mu}\Gamma(\frac{\mu}{2}+1)|_{\mu=0} = -c/2 \tag{2.19}$$

Now, by using (2.16) and (2.19) we obtain from (2.14)

$$\frac{d}{d\mu}\Phi_{Z_k|\gamma_k}(\mu)|_{\mu=0} = \frac{1}{2}\ln\lambda_k - \frac{1}{2}(c + \sum_{r=1}^{\infty}\frac{(-v_k)^r}{r!}\frac{1}{r}) \tag{2.20}$$

$$= \frac{1}{2}\ln\lambda_k + \frac{1}{2}(\ln v_k + \int_{v_k}^{\infty}\frac{e^{-t}}{t}dt)$$

The integral in (2.20) is known as the exponential integral of v_k, and can be efficiently calculated. On substituting (2.20) into (2.7) and using (2.13) and (2.5) we get the desired amplitude estimator

$$\hat{A}_k = \frac{\xi_k}{1+\xi_k}\exp\{\frac{1}{2}\int_{v_k}^{\infty}\frac{e^{-t}}{t}dt\}\,R_k. \tag{2.21}$$

It is useful to consider $\hat{A}_k$ as being obtained from R_k, by a multiplicative nonlinear gain function which depends only on the a priori and the a posteriori SNR ξ_k and γ_k, respectively. This gain function is defined by

$$G(\xi_k,\gamma_k) \triangleq \frac{\hat{A}_k}{R_k} \tag{2.22}$$

In Chapter 3, implementation details surrounding the computation of this multiplicative gain function are provided.

2.1.2 Principles: Harmonic Speech Enhancement

In the given problem, we have a single-channel speech degraded by additive background noise. The noisy observation can be written as

$$\mathbf{y} = \mathbf{x} + \mathbf{d} \tag{2.23}$$

where ***y***, ***x***, and ***d*** are N x 1 vectors representing the noisy speech, clean speech, and additive noise, respectively. In each analysis time frame, N is the total number of samples (assuming frame length=FFT size). According to the central limit theorem, the additive noise is statistically uncorrelated to the clean speech [2]. Denote by F* the N x N Fourier transform matrix, where (.)* indicates matrix Hermitian. The N-point short time Fourier transform (STFT) of the noisy speech is then given by

$$Y \triangleq F^* y = F^* x + F^* d \triangleq X + D \tag{2.24}$$

where Y, X, and D the Fourier transforms of the noisy speech, clean speech, and noise, respectively. The speech enhancement task aims to find a spectral domain linear estimator O such that $\hat{X} = OY$ produces a close approximation to the clean speech spectrum [2]. Ideally, the enhanced signal spectrum $\hat{X}$ should be identical to the clean speech spectrum X. To minimize the error norm between the estimated and clean speech spectra is to be calculated and several approaches to doing this have been studied with interest in the past. Such schemes were covered earlier in this chapter. However, in practical applications residual noise components always exist in the enhanced speech. It is impossible to remove all that we consider as background noise that

is meaningfully irrelevant. It is unnecessary to remove the background noise completely because in some scenarios, noise may be meaningful and relevant. Moreover, maintaining a comfort level of residual noise in the enhanced speech will actually improve the perceived quality in many situations [2]. For example, a train noise in the background in a telephone conversation may help the talker understand the location of the conversation to be in a train. Therefore, considering all these points, the linear estimator is set in such a way that the enhanced speech spectrum $\hat{X}$ tends to

$$\hat{X} = X + \Lambda D \tag{2.25}$$

where Λ is a N x N diagonal matrix with real-valued diagonal elements κ_k, and k = 0,…, N-1 is the frequency index. The parameters κ_k admit certain level of noise to appear at each frequency band k in the enhanced speech ($0 \le \kappa_k \le 1$). κ_k is frequency variable and also controls the residual noise level at each frequency band k. Therefore, κ_k is described as the frequency-dependent noise-flooring parameter (FDNFP) in [2]. With Eq. (2.25) the estimation error is

$$\begin{aligned} \varepsilon &= OY - (X + \Lambda D) \\ &= (O - I)X + (O - \Lambda)D \\ &= \varepsilon_x + \varepsilon_D \end{aligned} \tag{2.26}$$

where $\varepsilon_x \triangleq (O - I)X$ and $\varepsilon_D \triangleq (O - \Lambda)D$ represent the speech distortion and residual noise, respectively. Let

$$\bar{\varepsilon}_x^2 = trE\{\varepsilon_x \varepsilon_x^*\} \tag{2.27}$$

be the energy of speech distortion, where E is the expectation and tr is matrix trace. Also, let

$$\bar{\varepsilon}_{D,k}^2 = E\{I_k \varepsilon_D \varepsilon_D^* I_k^*\} \qquad k = 0,....., N-1 \tag{2.28}$$

is the residual noise energy in the k^{th} frequency band. I_k is the k^{th} spectral component selector defined as

$$I_k = [\underbrace{0....0....1}_{k}.....0.....0]$$

The speech enhancement task is done by the following optimization problem mentioned in [67]

$$\min_{O} \ \bar{\varepsilon}_x^2 \tag{2.29}$$

subject to

$$\varepsilon_{D,k}^2 \le T_k \qquad k = 0,....., N-1 \tag{2.30}$$

where T_k is the threshold used to suppress noise at the *kth* spectral component. The solution to this constrained optimization problem is given by [2]:

$$Q_k = \min\left(\sqrt{\frac{T_k}{S_d(k)}} + \kappa_k, 1\right) \tag{2.31}$$

where Q_k are the diagonal elements of O. Equation (2.31) is simplified by setting the threshold T_k to be a proportion of the noise power spectrum $S_d(k)$. Let $T_k = \chi_k S_d(k)$, where χ_k is the proportionality factor and specifies the amount of attenuation of noise power [2]. Then (2.31) can be rewritten as

$$Q_k = \min\left(\chi_k^{1/2} + \kappa_k, 1\right) \tag{2.32}$$

A small value of κ_k will be needed to maintain a low-level of residual noise in the enhanced speech. The parameter $\chi_k^{1/2}$ dominates the value of suppression gain. If we let $\kappa_k = 0$ for all,

then $Q_k = \min(\chi_k^{1/2}, 1)$ and the second term "ΛN" on the right-hand side of (2.25) becomes zero implying enhanced speech spectrum $\hat{X}$ will approach the clean speech spectrum X [2]. If we allow

$$\chi_k = \left(\frac{S_x(k)}{S_x(k) + S_d(k)} \right)^2$$

and $\kappa_k = 0$ then (2.32) reduces to the classical Wiener filter. Therefore, (2.32) is a combination of a gain factor and a small positive noise-flooring parameter. The value κ_k controls the level of admissible residual noise in the speech enhancement output. The quantity $\chi_k^{1/2} + \kappa_k$ determines the final suppression level of the noise. It is a balanced combination of admission and suppression of noise floor using these two parameters that the speech enhancement system needs. The implementation in Chapter 3 involves using frequency-dependent parameters χ_k and κ_k to enhance harmonics of voiced speech [2].

2.2 Reverberation

Reverberation is the collection of reflected sounds from the surfaces in an enclosure. Figure 2.2 shows the sound received by a single listener B as a function of time as a result of a sharp sound pulse from source A. The direct sound received first and is followed by distinct reflected sounds called early reflections and then a collection of many reflected sounds (diffused reflections) which blend and overlaps the direct path signal to give a composite effect called reverberation.

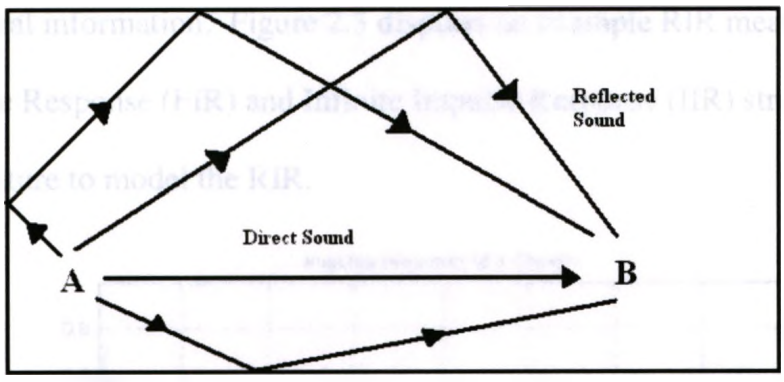

Figure 2.2: Reverberation- collection of reflected sound

The delay between the direct sound and the first early reflection is a significant characteristic for an enclosure, though not as important as the overall reverberation time. Rooms with reverberation are often characterized by Reverberation Time (RT_{60}) which is defined as the time required, in seconds, for the average sound in a room to decrease by 60 dB after a source stops generating sound. A space with a long reverberation time as in a big church or concert hall is referred to as a live environment. Table 2.1 displays the impact of reverberation on perceived quality and intelligibility for speech and music samples.

	Reverberation Time (seconds)			
	0.8 - 1.3	1.4 - 2.0	2.1 - 3.0	Optimum**
Speech	Good	Fair - Poor	Unacceptable*	0.8 - 1.1
Contemporary music	Fair - Good	Fair	Poor	1.2 - 1.4
Choral music	Poor - Fair	Fair - Good	Good - Fair	1.8 - 2.0+
* With an adequately designed and installed sound system, speech intelligibility concerns can be mitigated. ** Optimum reverberation time can be somewhat subjective and can shift based on numerous variables.				

Table 2.1: Reverberation time table [66]

The Room Impulse Response (RIR) is another important characteristic of a reverberant environment. The RIR mathematically models an enclosure's acoustics and carries important

temporal and spectral information. Figure 2.3 displays an example RIR measured in a church. Both Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) structures have been utilized in the literature to model the RIR.

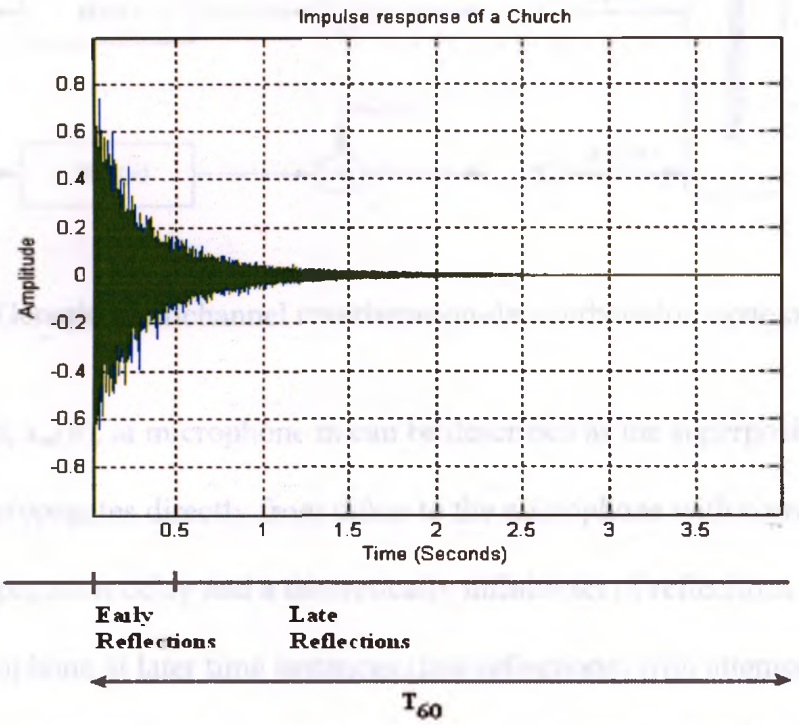

Figure 2.3: MATLAB plot of an impulse response

2.2.1 Speech Enhancement by Dereverberation: General System Description

A generic system diagram for multichannel dereverberation is shown in Figure 2.4. The speech signal, $s(n)$, from the talker propagates through acoustic channels, $H_m(z)$ for m = 1 to M. The output of each channel is observed using M microphones to give signals $x_m(n)$. All noise in the system is assumed additive and is represented by $v_m(n)$. [45]

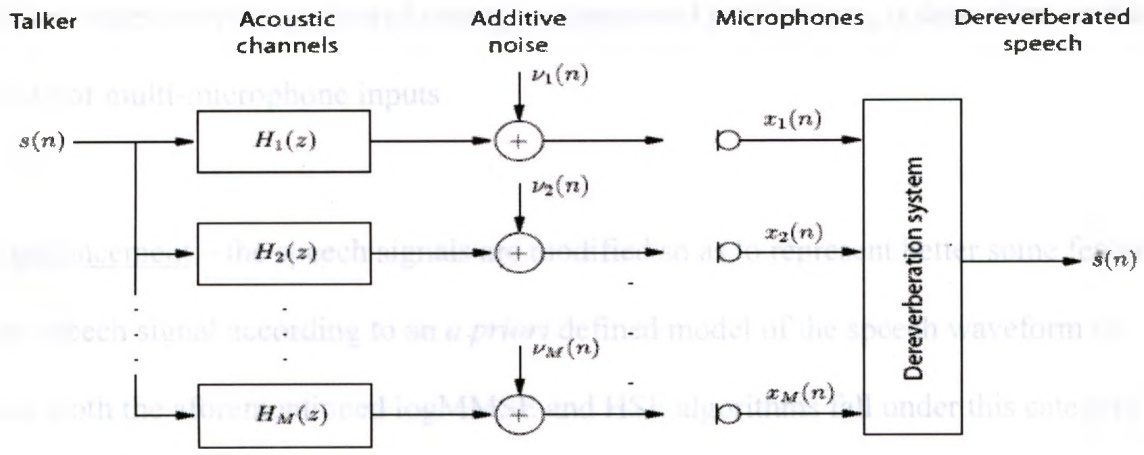

Figure 2.4: Generic multichannel reverberation-dereverberation system model [45]

The observed signal, $x_m(n)$, at microphone m can be described as the superposition of the direct-path signal, which propagates directly from talker to the microphone with corresponding attenuation and propagation delay and a theoretically infinite set of reflections of the talker signal arriving at the microphone at later time instances (late reflections) with attenuation dependent on the properties of the reflecting surfaces and their absorption coefficients [45]. This can be expressed as:

$$x_m(n) = \sum_{i=0}^{\infty} h_{m,i}(n)s(n-i), \tag{2.33}$$

where the acoustic channel impulse responses $h_{m,i}(n)$ represent the attenuation and the propagation delay corresponding to the direct signal and all the reflected components. The aim of speech dereverberation is to find a system with input $x_m(n)$, $m = 1, \ldots, M$ and output $\hat{s}(n)$, which is a good estimate of $s(n)$. It may be desired to estimate $s(n)$ with minimum Mean Square Error (MSE) [45]. Dereverberation algorithms broadly fall into three main categories:

Beamforming – the signals received at the different microphones are filtered and weighted so as to form a beam of enhanced sensitivity in the direction of the desired source and to attenuate

sounds from other directions. Beamforming or directional null forming is dependent on the availability of multi-microphone inputs.

Speech enhancement – the speech signals are modified so as to represent better some features of the clean speech signal according to an *a priori* defined model of the speech waveform or spectrum. Both the aforementioned logMMSE and HSE algorithms fall under this category and are further explored later in the thesis.

Blind deconvolution – the acoustic impulse responses are identified blindly, using only the observed microphone signals, and then used to design an inverse filter that compensates for the effect of the acoustic channels [45].

2.2.1 Literature Review on Dereverberation Techniques

A temporal and spectral reverberant speech enhancement technique is mentioned in [41]. In this technique, the temporal processing involves ways of identifying and enhancing high signal-to-reverberation ratio (SRR) regions in the temporal domain. Spectral processing involves ways of removing diffused reverberant tail components in the spectral domain.

In [63] harmonicity based dereverberation (HERB) is proposed, which uses properties of speech, harmonics, and estimates an inverse filter for an unknown impulse response. Even in severely reverberant environments, if a large amount of acoustically stable training data is available this algorithm models an accurate inverse filter.

Oppenheim and Schafer [46], [47] proposed a technique for speech enhancement by dereverberation. Simple echoes were observed as distinct peaks in the cepstrum of the speech signal and using a peak picking algorithm peaks were identified and were attenuated with a comb filter. Some researchers posit that the Linear Prediction (LP) residual signal contained the effects of reverberation, comprising peaks corresponding to resonances in voiced speech together with additional peaks due to the reverberant channel [48], [49]. These techniques aim to preserve the original characteristics of the residual and also to suppress the effects of reverberation such that dereverberated speech can be synthesized using the processed residual and the all-pole filter resulting from prediction analysis of the reverberant speech. It is assumed in these methods that the effect of reverberation on the Autoregressive (AR) coefficients is insignificant [48]. It was shown in [50] that multichannel observations can be used to estimate the AR coefficients precisely.

A wavelet extrema clustering was used to reconstruct an enhanced prediction residual by Griebel and Brandstein et al. [52], [53]. Coarse RIR estimates are employed to apply a matched filter type operation to obtain weighting functions for the reverberant residuals in [54]. A multichannel time-aligned Hilbert envelope was used in [55] to represent the strength of the peaks in the prediction residuals by Yegnanarayana et al. The Hilbert envelopes are then summed and the result used as a weight vector, which is applied to the prediction residual of one of the microphones. A weighting function was derived based on the signal-to-reverberant ratio in different regions of the prediction residual [49].

Gillespie et al. [56] demonstrate the kurtosis of the residual to be a useful reverberation metric, which they then maximize using an adaptive filter. This method was extended by Wu and DeLiang [57], who added a spectral subtraction stage to further suppress the remaining reverberation. These methods attenuate the resonances due to reverberation in the prediction residual but they also reduce naturalness in the dereverberated speech. A method closely related to the before mentioned scheme was proposed by Nakatani et al. [58] which assumes a sinusoidal speech model. In this scheme, first the fundamental frequency of the speech signal was identified from the reverberant observations and then the remaining sinusoidal components are identified. Using the identified magnitude and phases of these sinusoids, an enhanced speech signal is synthesized. Then, the reverberant and the dereverberated speech signals were used to derive an equivalent equalization filter. The processing is performed in short time frames and the inverse filter was updated in each time frame. It was shown that this inverse filter tends to the RIR equalization filter but this method is computationally expensive [58].

Spectral subtraction was applied to dereverberation by Lebart et al. and extended to the multichannel case by Habets [61], [62]. The scheme assumes a statistical model of the RIR comprising Gaussian noise modulated by a decaying exponential function. The decay rate of this exponential function is controlled by the RT_{60}. It was shown that the PSD of the impulse response could be identified and removed by spectral subtraction, if a blind estimation of RT_{60} can be done [61].

Unlike speech enhancement for noisy background, for which Hu and Loizou [12] systematically compared a number of speech enhancement algorithms, no comprehensive subjective evaluation

of dereverberation algorithms has been reported. In a recent paper, Jeub et al. [7] compared the performance of four different dereverberation algorithms using an objective speech quality measure, and showed that a dereverberation algorithm that combines spectral subtraction and Wiener filtering provides the best performance. In this thesis, a combination of the logMMSE and HSE is used to reduce the reverberant portions, and the performance of this combination will be compared to the results reported in Jeub et al. [7].

2.3 Bandwidth Extension of Speech

Different methods have been mentioned in research for HF bandwidth extension of speech [11]. In general there is a blind method wherein no useful information about HF envelope is sent through the channel for reconstruction of full band spectrum. In the other, some *a priori* knowledge of HF spectrum is sent.

Blind approach: When the BWE algorithm is blind, no information about the missing high-frequency components is passed to the decoder that reconstructs the HF part of the sound. Usually spectral envelope information of the HF region would be needed to adjust the HF band gains, if high quality sound is desired. Therefore, in blind approaches assumptions on the statistics of audio signals can be used to design such systems. The main advantages of using this approach are that such a BWE system can be applied to a wide class of signals like music and speech and that there are no requirements on the signal format, because the only required information is the actual signal waveform. The computation is not expensive and can be deployed in real-time bandwidth extension schemes. The drawback is that the quality of the

bandwidth-extended output signal is significantly lower than that of the original full-bandwidth signal, even though it is higher than the bandwidth-limited signal. This is due to the lack of information about the missing high frequencies [9]. The BWE implementation in Chapter 3 was designed to work at an audio IO latency of approximately 1-3 ms to fit in real-time projects in the future. One such blind scheme called a non-linear distortion BWE was used to achieve desired project results.

***A Priori* knowledge approach:** The BWE algorithm does have a priori knowledge regarding the missing high-frequency components. Usually spectral envelope information of HF region would be sent through the channel to adjust the HF band gains at the decoder, if high quality sound is desired. More exact reconstruction of the original full-bandwidth signal is possible in this case which is not possible with the blind approach. Therefore, the quality of the bandwidth-extended signal can be indistinguishable from the original full bandwidth signal. The main advantage of this approach is high perceptual sound quality of the output signal. The drawback is the computation cost, as usually it is hard to make an encoder-decoder scheme work in real-time project applications. The techniques of spectral band replication (SBR [37]), linear prediction coding (LPC), and modified discrete cosine transform (MDCT) belong to this second category. In [9] some examples that follow this kind of BWE are mentioned. The line spectral frequencies (LSF) are an alternative representation of LPC coefficients. The low band LSF of the synthesis signal are obtained from the input speech signal and the high band LSF are estimated from the low band ones using statistical models. There is also a method of bandwidth extension based on codebook mapping mentioned in [9] in page 217. The implementation of BWE in this project eventually will be used in a real-time hand application. It may not be

practical to do a real-time optimization of a coding-decoding mechanism for BWE (like SBR) along with speech enhancement algorithms as it's expensive.

2.4 Objective Speech Quality Measures

As mentioned in Chapter 1, objective speech quality measures offer an attractive way to benchmark the performance of a speech processing algorithm. In this thesis, both intrusive and non-intrusive speech quality measures are utilized. In particular, the Perceptual Evaluation of Speech Quality (PESQ) metric as standardized by the International Telecommunication Union (ITU) [40], the Itakura-Saito (IS) metric, and the Speech To Reverberation Masking Ratio (SRMR) metrics are utilized. A brief description of these objective metrics is give in the following sections.

2.4.1 Perceptual Evaluation of Speech Quality (PESQ)

The structure of the PESQ measure is shown in Figure 2.5. The clean and degraded signals are first level-equalized to a standard listening level, and filtered by a filter with response similar to a standard telephone handset. The signals are then synchronized in time to compensate for any time delays, and then processed through an auditory transform to obtain the loudness spectra. The auditory transform in PESQ uses a psychoacoustic model which translates the reference and degraded signals into a representation of perceived loudness in time and frequency [14].

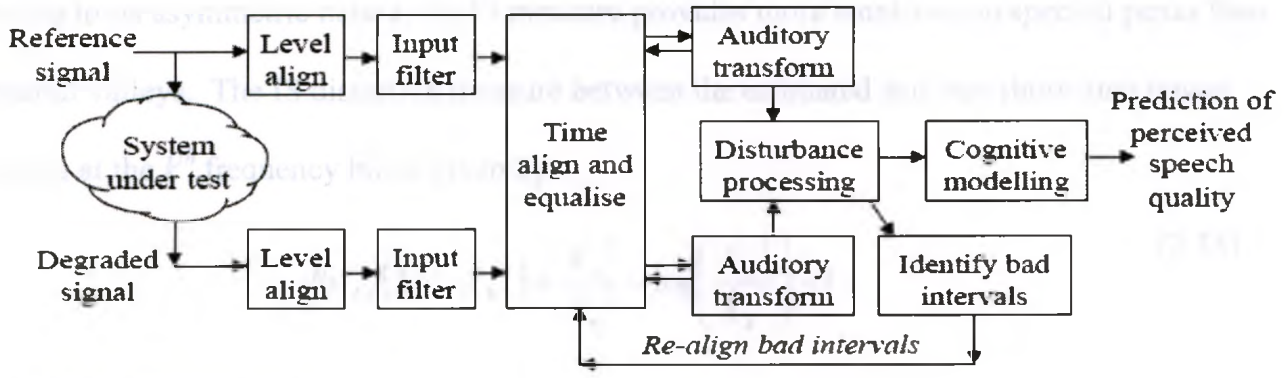

Figure 2.5: Structure of perceptual evaluation of speech quality (PESQ) model [14].

This is accomplished by first calculating the instantaneous power spectrum in each frame and grouping the spectrum into bins equally spaced on a modified Bark scale [14]. The Bark spectra are then equalized for linear filtering and gain variation, and converted to a loudness scale using a frequency-dependent threshold and exponent. These so-called "internal representations" are then compared, and the absolute differences between them are weighted appropriately based on masking, deletion, and asymmetry [14]. Finally the frequency – specific and frame-by-frame differences are aggregated and mapped to a predicted Mean Opinion Score (MOS) using an optimized and validated mapping function.

2.4.2 Itakura-Saito (IS)

The Itakura–Saito distance is a measure of the perceptual difference between a reference power spectrum $S(\omega)$ and a test spectrum $X(\omega)$. It was proposed by Fumitada Itakura and Shuzo Saito in the 1970s while they were with Nippon Telegraph and Telephone.

$$d_{IS}(X(\omega),S(\omega)) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\frac{S(\omega)}{X(\omega)} - \log\left(\frac{S(\omega)}{X(\omega)}\right) - 1\right] d\omega \tag{2.34}$$

Owing to its asymmetric nature, the IS measure provides more emphasis on spectral peaks than spectral valleys. The IS distortion measure between the estimated and true short-time power spectra at the k^{th} frequency bin is given by:

$$dist_{IS}\left(X_k^{\,2}, \hat{X}_k^{\,2}\right) = \frac{X_k^{\,2}}{\hat{X}_k^{\,2}} - \log\left(\frac{X_k^{\,2}}{\hat{X}_k^{\,2}}\right) - 1 \tag{2.35}$$

2.4.3 PESQ-IS Overall

In a study comparing the performance of different objective speech quality metrics in predicting the subjective speech quality ratings of speech enhancement algorithms, Hu and Loizou [13] discovered that a combination of PESQ and IS resulted in a correlation coefficient greater than 0.9 between predicted and actual quality scores. This combination is given by:

$$BF1 = \max(0, PESQ - 1.696); \tag{2.36}$$
$$BF2 = \max(0, IS - 11.708); \tag{2.37}$$
$$BF3 = \max(0, IS - 3.559); \tag{2.38}$$
$$BF4 = \max(0, PESQ - 2.431); \tag{2.39}$$
$$BF5 = \max(0, PESQ - 2.564); \tag{2.40}$$
$$Y\_TOTAL = 1.757 + 1.740 \times BF1 + 0.047 \times BF2 - 0.049 \times BF3 - 2.593 \times BF4 + 11.549 \times BF5; \tag{2.41}$$

This thesis utilizes this combination of PESQ and IS to evaluate the performance of speech enhancement algorithms in noisy backgrounds.

2.4.4 Speech to Reverberation modulation energy ratio (SRMR):

SRMR is a non-intrusive quality measure metric. The processing performed by the cochlea is simulated by filtering $\hat{x}(n)$ by a 23-channel gamma-tone filter bank. The filter center

frequencies range from 125 Hz to nearly 4 kHz (half the sampling rate) and the filter bandwidths are characterized by the equivalent rectangular bandwidth. For simplicity $\hat{x}(n)$ will be used to denote the (de)reverberant speech signal [42].

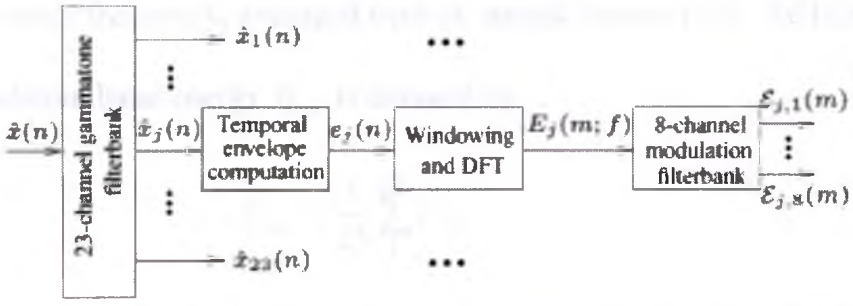

Figure 2.6: Signal processing involved in the computation of modulation spectra [42]

For each of the 23 spectral bands, the time domain temporal envelope $e_j(n)$ of the j^{th} filter output signal $\hat{x}_j(n)$ is then computed using the Hilbert transform H{.} as

$$e_j(n) = \sqrt{x_j(n)^2 + \mathrm{H}\{x_j(n)\}^2}. \tag{2.42}$$

This results in 23 time domain signal vectors. Each of these 23 temporal envelopes $e_j(n)$ are multiplied by a 256-ms Hamming window with 32-ms shifts and the windowed envelope for each frame is represented as $e_j(m,n)$. Modulation spectral energy for each critical band is then computed as the squared magnitude of the discrete Fourier transform F{.}of temporal envelope $e_j(m,n)$

$$E_j(m;f) = |F(e_j(m;f)|^2 \tag{2.43}$$

where f indexes the modulation frequency bins. Again, an auditory-inspired modulation filterbank is simulated by grouping modulation frequency bins into eight bands [42].

The notation $\overline{\eta}_{j,ks}$ is used to denote the average modulation energy over all frames of the j^{th} critical-band signal grouped by the modulation filter, with j =1, ..., 23, ks = 1,..., 8. The modulation spectrogram shows modulation energy distribution as a function of modulation frequency and acoustic frequency, averaged over all speech frames [42]. Additionally, the average per-modulation band energy $\overline{\eta}_{j,k}$ is denoted by

$$\overline{\eta}_{j,ks} = \frac{1}{23}\sum_{1}^{23}\overline{\varepsilon}_{j,ks} \tag{2.44}$$

It was shown in [42] that LF modulation energy is reduced for reverberant and dereverberated speech signals. Such effects are relatively independent of reverberation time and are likely due to early reflections. But reverberation time dependency was observed for higher frequency modulation channels [42]. It was also shown [42] that the modulation energy increases almost linearly with reverberation time. Moreover, the delay and sum beamformer is shown to reduce high-frequency modulation energy by approximately 1 dB relative to reverberant speech. An approximate 6.5 dB difference remains between anechoic and dereverberated speech for a reverberation time of 533 ms [42]. Using this insight, an adaptive measure termed speech to reverberation modulation energy ratio (SRMR) is proposed for non-intrusive quality measurement of reverberant and dereverberated speech. The measure is given by

$$SRMR = \frac{\sum_{k=1}^{4}\overline{\eta}_{ks}}{\sum_{k=5}^{K^*}\overline{\eta}_{ks}} \tag{2.45}$$

and is adaptive as the upper summation bound K^* in the denominator is dependent on the speech signal under test [42]. In this thesis, the SRMR measure is used to benchmark the performance of speech enhancement algorithms in reverberant environments.

Chapter 3 : Algorithm Implementation and Results

3.1 System Description

The entire software was written in C with floating point precision using the Intel IPP DSP library in an Eclipse CDT IDE on Linux operating system. Gnuplot [15], Octave and MATLAB were used for plotting and analysis. This chapter is divided into three main headings: Ephraim-Malah log MMSE, HSE and BWE. All these algorithms are cascaded in series, in the same order. The speech enhancement is followed by BWE, cascaded in series. Outputs of the speech enhancement system comprising the logMMSE and HSE were sampled at a rate of 8 kHz. The analysis frame rate was set to 32 ms (256 audio samples). The hop-size was 8 ms at an analysis overlap of 75%. The FFT size was set to 256. A Hamming window was used for analysis of corrupted inputs. The outputs of the BWE were sampled at 16 kHz per frame rate.

3.2 EMSR logMMSE Implementation

As shown in Chapter 2, the desired amplitude estimator is given by:

$$\hat{A}_k = \frac{\xi_k}{1+\xi_k} \exp\{\frac{1}{2} \int_{v_k}^{\infty} \frac{e^{-t}}{t} dt\} R_k. \tag{3.1}$$

It is useful to consider $\hat{A}_k$ as being obtained from R_k, by a multiplicative nonlinear gain function which depends only on the a priori and the a posteriori SNR ξ_k and γ_k, respectively. This gain function is defined by

$$G(\xi_k, \gamma_k) \triangleq \frac{\hat{A}_k}{R_k} \tag{3.2}$$

The logMMSE estimator algorithm can therefore be implemented using the following four steps: For each windowed speech frame:

1. Compute the DFT of the noisy speech signal: $Y(\omega_k) = Y_k \exp(j\theta_y(k))$.

2. Estimate the *a posteriori* SNR as $\gamma_k = Y_k^2 / \lambda_d(k)$ where $\lambda_d(k)$ is the power spectrum of the noise signal computed during non-speech activity (e.g., during initial silence periods or during speech pauses). Then estimate ξ_k using any one of the two equations.

$$\xi_k(p) = \max(\bar{\gamma}_k(p) - 1, 0) \tag{3.3}$$

$$\xi_k(p) = \beta \frac{\hat{X}_k^2(p-1)}{\lambda_d(k, p-1)} + (1-\beta)\max[\bar{\gamma}_k(p) - 1, 0] \tag{3.4}$$

where $0 < \alpha < 1$ is the weighting factor and p is time frame number. This equation needs initial conditions for the first frame. The following initial conditions were recommended:

$$\xi_k(0) = \beta + (1-\beta)\max[\bar{\gamma}_k(0) - 1, 0] \tag{3.5}$$

Good results were obtained with $\beta = 0.98$.

3. Estimate the enhanced signal magnitude.

4. Construct the enhanced signal magnitude $\hat{X}_k(\omega) = \hat{X}_k \exp(j\theta_y(k))$. Compute the inverse DFT of $\hat{X}_k(\omega_k)$ to get the enhanced time-domain signal $\hat{x}(n)$ corresponding to a given input speech frame.

3.2.1 *A Priori* Estimator: Decision Directed Approach

The MMSE amplitude estimator was derived under the assumption that the *a priori* SNR and the noise variance are known. In practice, however, we only have access to the noisy speech signal.

The noise variance can be estimated easily assuming noise stationarity, and can in principle be computed during non speech activity. Ephraim-Malah found that the MMSE estimator was relatively insensitive to small perturbations of the ξ_k value. More interestingly, the MMSE was more sensitive to underestimates rather than overestimates of the *a priori* SNR ξ_k. There are several proposed methods in the literature for computing the *a priori* SNR estimates but in this implementation, a decision-directed approach is followed.

Let $\xi_k(p)$, $A_k(p)$, $\lambda_d(k,p)$ and $\gamma_k(p)$ denote the *a priori* SNR, the amplitude, the noise variance, and the *a posteriori* SNR, respectively, of the corresponding k^{th} spectral component in the p^{th} analysis frame. The derivation of the *a priori* SNR estimator is based here on the definition of $\xi_k(p)$, and its relation to the *a posteriori* SNR $\gamma_k(p)$, as given below:

$$\xi_k(p) \cong \frac{E\{A_k^2(p)\}}{\lambda_d(k,p)} \tag{3.6}$$

$$\xi_k(p) = E\{\gamma_k(p) - 1\} \tag{3.7}$$

Using the above equations, we get

$$\xi_k(p) = E\left\{\frac{1}{2}\frac{A_k^2(p)}{\lambda_d(k,p)} + \frac{1}{2}[\gamma_k(p) - 1]\right\} \tag{3.8}$$

$$\xi_k(p) = \beta \frac{\hat{A}_k^2(p-1)}{\lambda_d(k, p-1)} + (1-\beta)P[\gamma_k(p) - 1], 0 \le \beta \le 1 \tag{3.9}$$

where $\hat{A}_k(p-1)$ is the amplitude estimator of the k^{th} signal spectral component in the $(p-1)^{th}$ analysis frame, and $P[.]$ is an operator which is defined by

$$P[x] = \begin{cases} x & if\ x \geq 0 \\ 0 & Otherwise \end{cases} \tag{3.10}$$

3.2.2 Elimination of Musical Noise

Cappe [39] noted that the effectiveness of the *a priori* SNR estimator ξ_k is closely coupled to the suppression rule. The Ephraim-Malah suppression rule (EMSR) is greatly affected by both *a priori* (ξ_k) and *a posteriori* (γ_k) parameters. Of these two parameters, the *a priori* ξ_k is the dominant one and exerts the most influence on suppression.

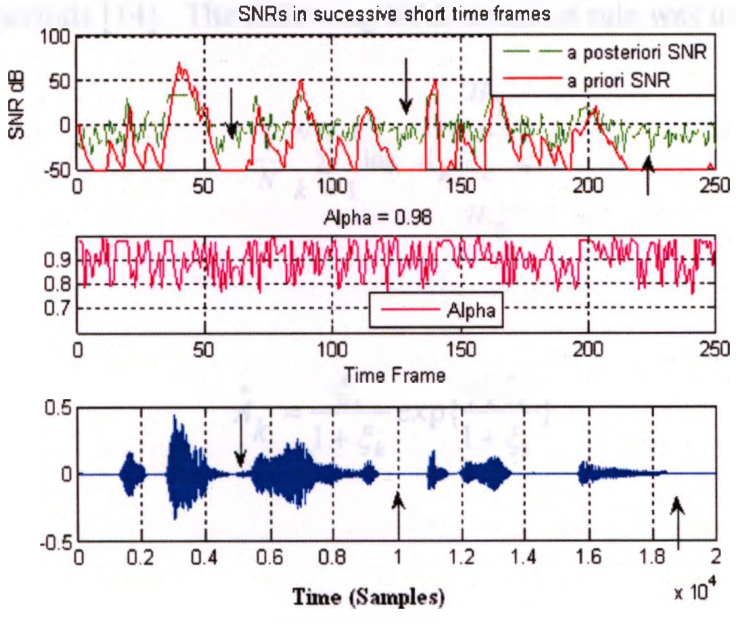

Figure 3.1: The *a posteriori* and *a priori* adaptation for an adaptive α smoothing factor

As shown in Figure 3.1, during speech presence, the *a posteriori* clearly follows *a priori*. It must be noted that there is no big fluctuation in values of these two SNRs. When γ_k stays below or is in proximity to 0 dB, the ξ_k estimate corresponds to a smoothed version of γ_k over successive

short-time frames. As a consequence, the variance of ξ_k is much smaller than the value of γ_k. When γ_k is considerably larger than 0 dB, the ξ_k estimates follow the γ_k estimates very closely with a simple delay of one short-time frame. In both cases, the decision-directed estimator of ξ_k produces smoothed estimates of the true *a priori* SNR. In contrast, the spectral subtraction algorithm depends on the estimation of the *a posteriori* SNR, which can change radically from frame to frame [14]. Therefore, the decision-directed estimator is highly preferred.

3.2.3 Statistical Model VAD

A statistical-model based voice activity detector (VAD) was used to update the noise spectrum during speech-absent periods [14]. The following VAD decision rule was used:

$$\frac{1}{N}\sum_{k=1}^{N-1}\log A_k \underset{H_0}{\overset{H_1}{\gtrless}} \varsigma$$

where

$$\hat{A}_k = \frac{\xi_k}{1+\xi_k}\exp\{\frac{\gamma_k \xi_k}{1+\xi_k}\}$$

where γ_k and ξ_k are the *a posteriori* and *a priori* SNR's and ξ_k is computed using the decision directed approach with $\beta = 0.98$, N is the size of FFT, H_1 denotes the hypothesis of speech presence, H_0 denotes the hypothesis of speech absence, and ς is a fixed threshold, which was set

to $\varsigma = 0.15$. When speech absence was detected, the noise power spectrum was updated according to:

$$D_k(p) = (1-\beta).Y_k^{\,2}(p) + \beta D_k(p-1)$$

where $\beta = 0.98$, $D_k(p)$ is the noise power spectrum in frame p (for frequency bin k) and $Y_k^{\,2}(p)$ is the noisy speech power spectrum [14].

3.2.4 The influence of factor β (adaptive)

The alpha parameter was also adaptively calculated as mentioned in [68]. In the decision-directed approach, the parameter β is used to control the speed of the forgetfulness of the estimator. A low value of β will be suitable for rapidly changing speech regions, while a high value of α will be suitable for near stationary speech frames. A fixed α value was normally chosen to be in the range of 0.95 to 0.99. In this implementation, a fixed value of 0.98 also works very well. However, it is possible to deduce whether the speech frames are changing rapidly or not by computing the frame energy. The common assumption is that noise is stationary and the noise energy does not change significantly from frame to frame. Hence, significant changes in frame energy from frame to frame must be due to the underlying speech changes. A possible formulation using only the previous frame energy is given below:

$$\beta = \sqrt{1 - \frac{FE_p - FE_{p-1}}{\max(FE_p, FE_{p-1})}} \tag{3.11}$$

where

$$FE = \sum_k Y(k)^2 \tag{3.12}$$

3.3 HSE Implementation Details

The signal can be corrupted by background noises like multi-talker babble, train, automobile, restaurant noises, and street noises or it can be corrupted by early and diffused reflections in any sort of reverberant chamber ranging from a closed space to a concert hall. The HSE algorithm enhances the corrupted signal by harmonic emphasis using an adaptive comb filtering scheme. An adaptive resonant comb frequency response was used to enhance the harmonic sinusoidal peaks in the spectrum by submerging the noisy components in the valleys between resonant comb structures in the spectral domain. The Inverse FFT was applied to the clean spectrum to reconstruct back the signal in time domain. Two design parameters were employed in the suppression gain, namely, the Frequency domain noise floor parameters (FDNFP, κ_k) and gain factor (χ_k). The FDNFP controls the level of allowable residual noise in the enhanced speech. Enhanced harmonic structures were incorporated into the FDNFP by time-domain processing of the linear prediction residuals of voiced speech. An adaptive was deployed for enhancing the harmonics further. This algorithm was designed with an intention to integrate it into hearing aid or cell phone speech enhancement applications.

3.3.1 Frequency Domain Noise Floor Parameter

The fundamental frequency was found using the autocorrelation method described later in this chapter. Because voiced speech is periodic in nature, its magnitude spectrum exhibits peaks and valleys separated by harmonics of the fundamental frequency. The harmonic structure of clean voiced speech is often corrupted by the additive noise spectrum. Besides suppressing the noise

to a comfortable low-level, the FDNFP can be used to enforce a harmonic-shaping on the residual noise spectrum in the enhanced speech.

The motivation for time domain processing is to preserve the correlation between both spectral amplitudes and phases when restoring the harmonics. Because the phase coherence in voiced speech is a significant source of correlation and corresponds to energy localization in the time domain, the harmonic information from noisy speech is retrieved by enhancing the excitation peaks in the linear prediction residuals [2].

The correct linear prediction (LP) residual peaks separated in a given time frame due to the periodicity of speech are found by using the algorithm depicted in Figure 3.2. The LP residual array is sorted in descending order and stored in array X. The first element is assumed to be a right LP peak and marked as a reference peak. An artificial LP peak array Y is formed using this reference peak and fundamental time period information such that the $LP\ peak\ array \in (0, frame\ length - 1)$ in samples .i.e. the LP peak array is limited to values in between 0 and frame length-1.

The elements in Y are traced for nearest neighbors in array X and real LP spikes are stored in array Z. If all the elements in Z are closer to elements in artificial array Y within a small threshold limit (5 samples), the array Z is declared to have the correct LP spikes for an analysis time frame. If not, the next element in array X is chosen to be the reference LP peak and the whole sequence of processing is repeated until the correct LP peaks are obtained. Kaiser

windows are applied around these LP peaks as shown in Figure 3.3 to get smooth FDNFP in the time domain.

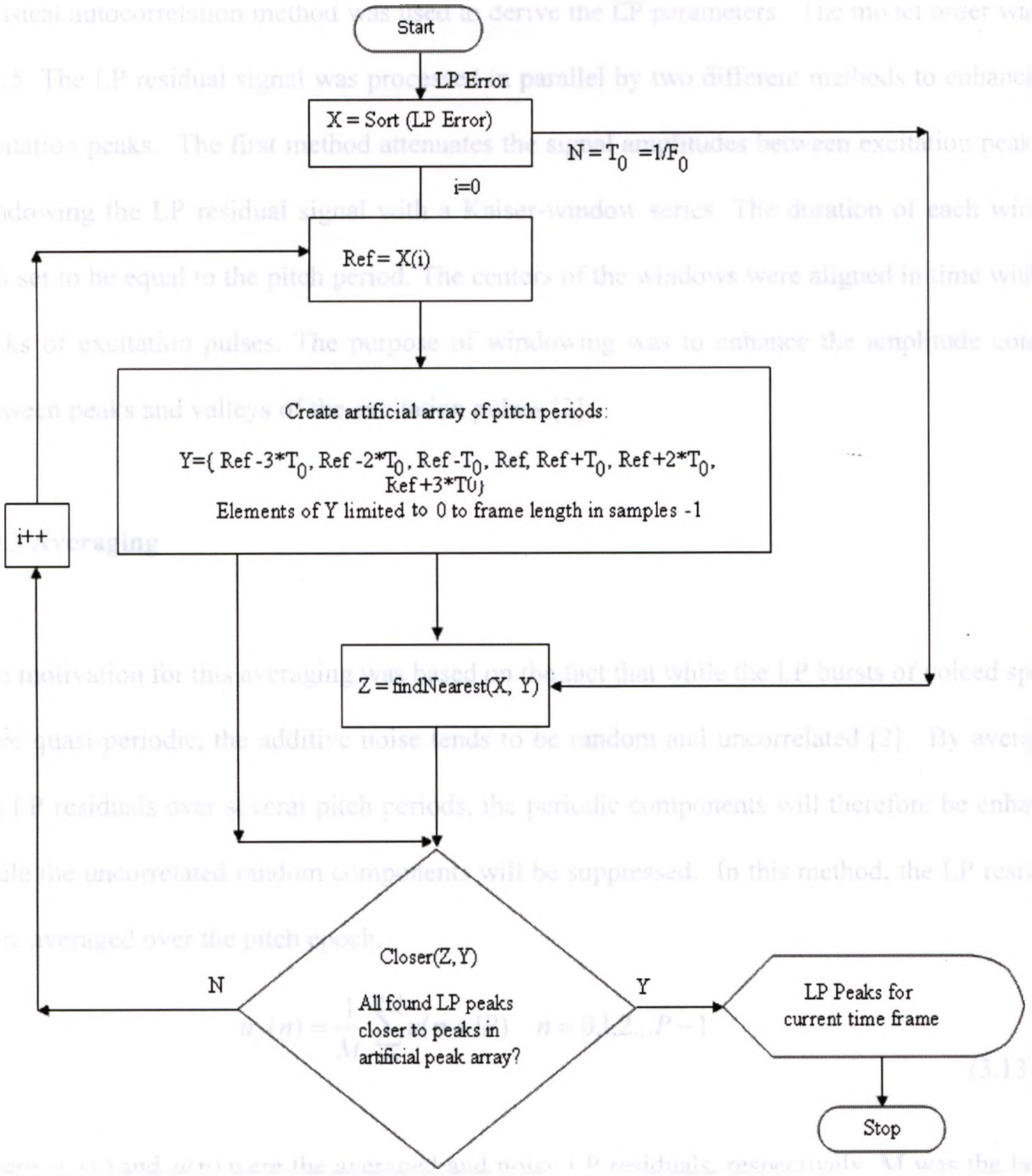

Figure 3.2: System for finding the LP residual peaks

3.3.2 Windowing

For voiced speech, a linear prediction (LP) analysis was performed on the noisy speech. A classical autocorrelation method was used to derive the LP parameters. The model order was set to 15. The LP residual signal was processed in parallel by two different methods to enhance the excitation peaks. The first method attenuates the signal amplitudes between excitation peaks by windowing the LP residual signal with a Kaiser-window series. The duration of each window was set to be equal to the pitch period. The centers of the windows were aligned in time with the peaks of excitation pulses. The purpose of windowing was to enhance the amplitude contrast between peaks and valleys of the excitation pulses [2].

3.3.3 Averaging

The motivation for this averaging was based on the fact that while the LP bursts of voiced speech were quasi-periodic, the additive noise tends to be random and uncorrelated [2]. By averaging the LP residuals over several pitch periods, the periodic components will therefore be enhanced while the uncorrelated random components will be suppressed. In this method, the LP residuals were averaged over the pitch epoch.

$$u_a(n) = \frac{1}{M}\sum_{i=0}^{M-1} u(n+iP) \quad n = 0,1,2...P-1 \tag{3.13}$$

where $u_a(n)$ and $u(n)$ were the averaged and noisy LP residuals, respectively. M was the largest integer number of pitch periods in the current analysis frame. P was the number of samples in one pitch period. n was the time sample index, and i was the pitch epoch index. It should be

noted that the duration of $u_a(n)$, the averaged LP residual, was exactly one pitch period [2]. Subsequently, $u_a(n)$ was repeated during the whole analysis frame. In order to provide the necessary pitch information for the aforementioned windowing and averaging process, a pitch detection algorithm was run in parallel to determine the pitch period of the current frame. The SIFT (Simple Inverse Filter Tracking) method was used for pitch determination [3].

$$u_h(n) = qu_w(n) + (1-q)u_c(n) \quad n = 0,....L-1 \tag{3.14}$$

where q was a weighting factor, and $u_w(n)$ was the window-enhanced LP residuals. $u_c(n)$ was obtained by periodically extending $u_a(n)$ in over the entire duration of the analysis frame. $u_h(n)$ was the final LP residual with enhanced periodicity. Because the averaging-enhanced residuals may not be as accurate as windowing-enhanced residuals, due to shimmer for example, the parameter was set to 0.8. $u_h(n)$ was then transformed to the frequency domain, and its magnitude spectrum was normalized to 0 dB by its maximum magnitude. Finally, the FDNFP was down by 5 dB for strongly voiced speech [2] to allow for some headroom.

The β parameter for Kaiser Window was set to 4 by default. As the value of parameter β goes higher, the windows become narrow and hence the smoother the LP residual. The β parameter was adjusted based on performance of the system for different inputs.

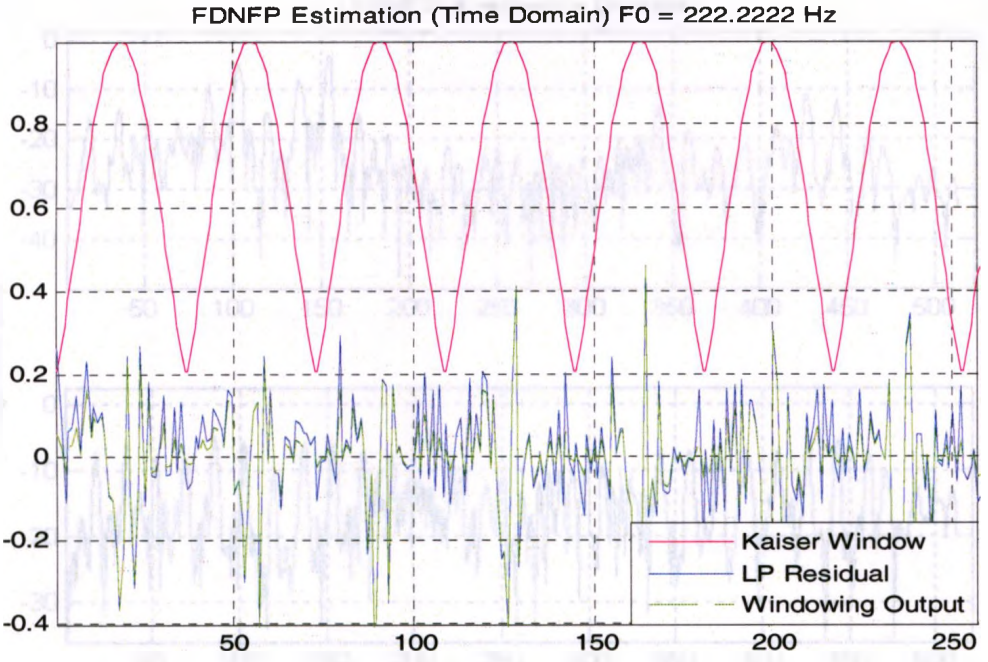

Figure 3.3 : FDNFP Computation: Kaiser Windowing.

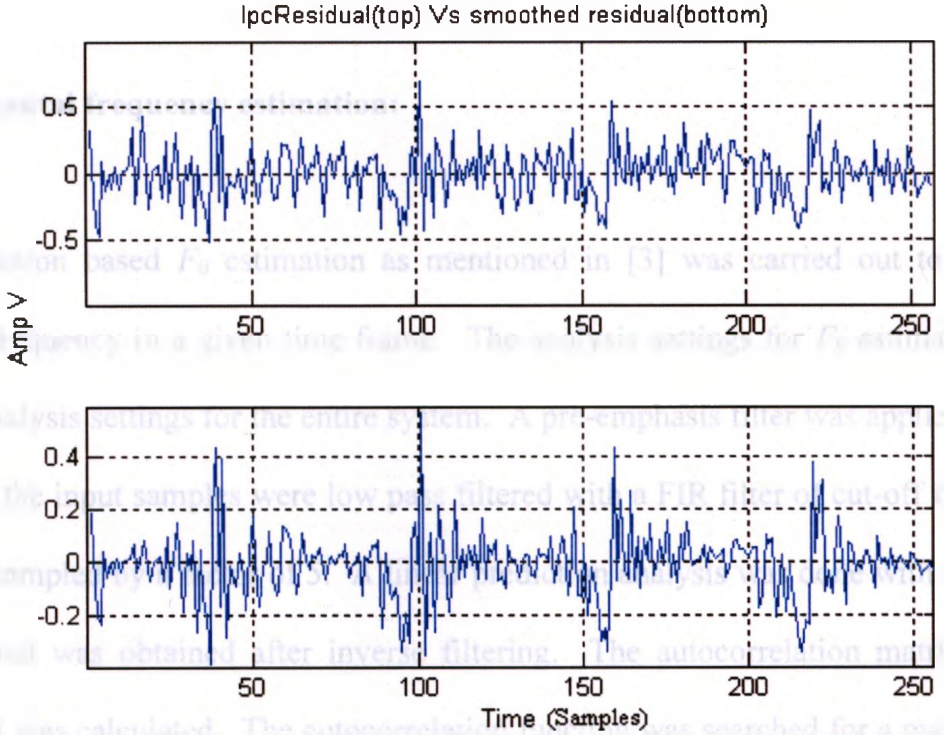

Figure 3.4: FDNFP Computation: LP Residual (top); smooth LP residual (bottom)

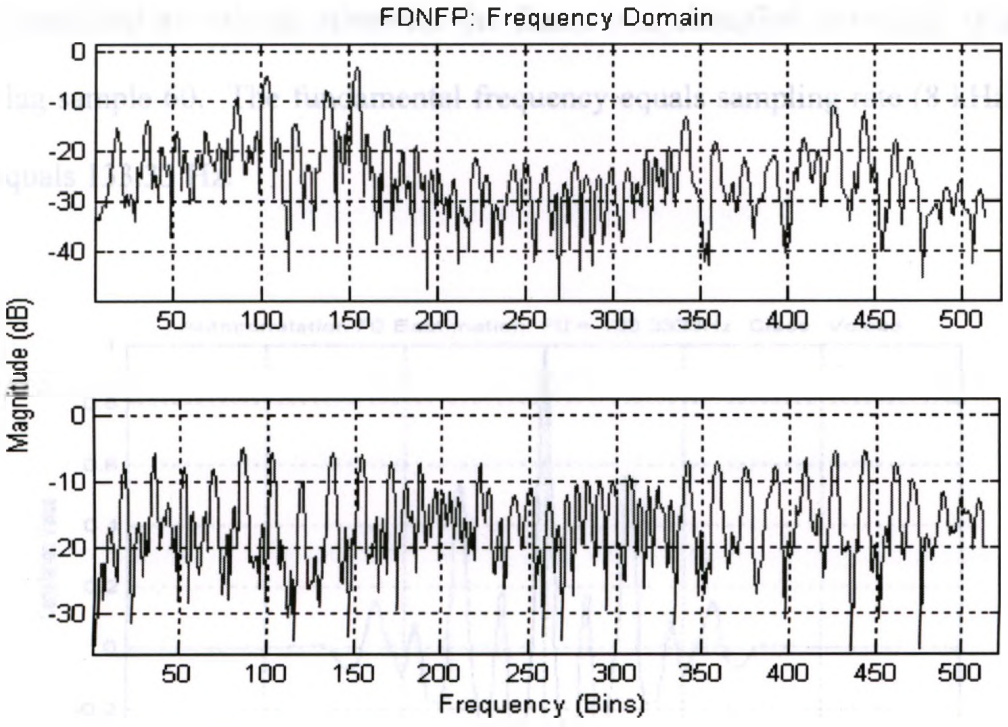

Figure 3.5: FDNFP: LP residual Frequency Spectrum (top); Smooth spectrum (bottom)

3.3.4 Fundamental frequency estimation:

An autocorrelation based F_0 estimation as mentioned in [3] was carried out to estimate the fundamental frequency in a given time frame. The analysis settings for F_0 estimation were the same as the analysis settings for the entire system. A pre-emphasis filter was applied to the input signal. Then, the input samples were low pass filtered with a FIR filter of cut-off frequency 800 Hz and downsampled by a factor of 5. A linear prediction analysis was done with an order of 4. The LP residual was obtained after inverse filtering. The autocorrelation matrix for the LP residual signal was calculated. The autocorrelation function was searched for a maximum (other than the value at zero lag) within a range of allowed values which corresponded to F_0 from 40 to 500 Hz, and which also exceeded a certain threshold value (0.4). If this search succeeded, the

frame was classified as voiced; otherwise the frame was classified unvoiced. Figure 3.6 has maximum lag sample 60. The fundamental frequency equals sampling rate (8 kHz) divided by 60 which equals 133.33 Hz.

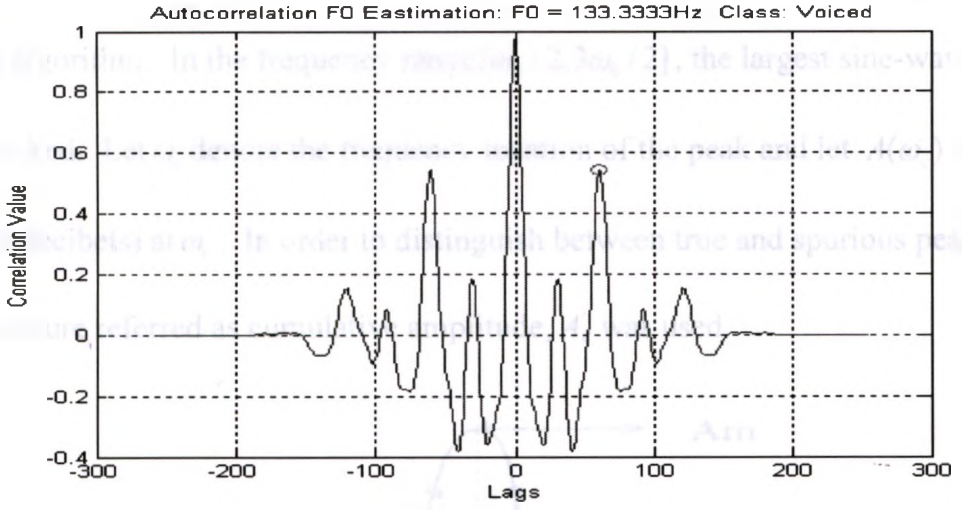

Figure 3.6: Autocorrelation based fundamental frequency estimation.

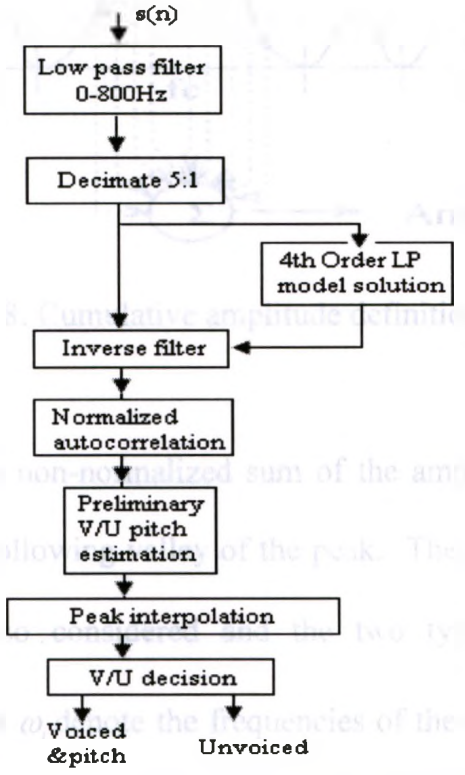

Figure 3.7: Block diagram for fundamental frequency detection [3].

3.3.5 Peak picking algorithm & Harmonic peak test:

The peak picking algorithm picks harmonic peaks from the smooth FDNFP spectrum. For voiced frames, the estimation of the maximum voiced frequency was based on the following peak picking algorithm. In the frequency range $[\omega_0/2, 3\omega_0/2]$, the largest sine-wave amplitude (peak) was picked. Let ω_c denote the frequency location of the peak and let $A(\omega_c)$ denote the amplitude (in decibels) at ω_c. In order to distinguish between true and spurious peaks, a second amplitude measure referred as cumulative amplitude, A_c was used.

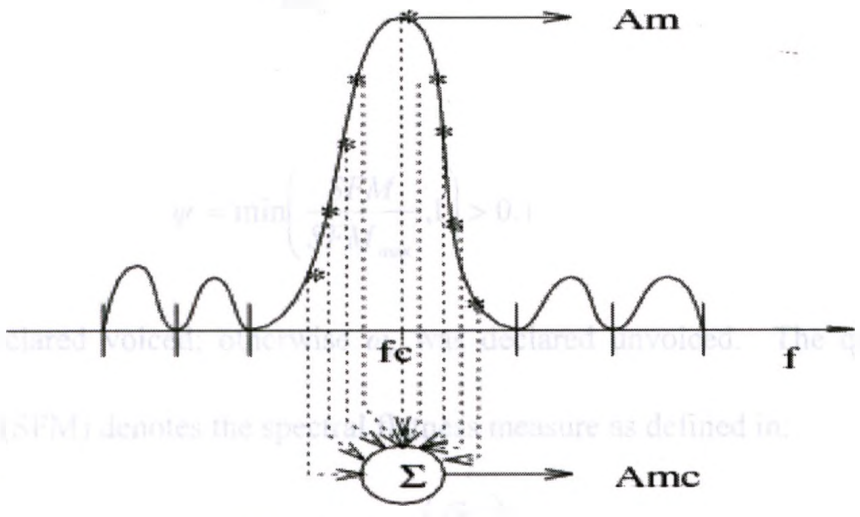

Figure 3.8: Cumulative amplitude definition [4]

This amplitude was defined as a non-normalized sum of the amplitudes of all of the samples from the previous valley to the following valley of the peak. The peaks in the frequency range $[\omega_c - \omega_0/2, \omega_c + \omega_0/2]$ were also considered and the two types of the amplitudes were calculated for each peak [4]. Let ω_i denote the frequencies of these peaks and let $AM(\omega_i)$ and $\overline{AM}_c(\omega_i)$ be the amplitude and cumulative amplitude, respectively, at ω_i. Denote by

$AM_c(\omega_l)$ the mean value of these cumulative amplitudes, and by l the number of the nearest harmonic to ω_c, the following "harmonic test" was applied to the peak at ω_c if

$$\frac{AM_c(\omega_c)}{\overline{AM}_c(\omega_l)} > 1.3 \tag{3.15}$$

or

$$AM_c(\omega_c) - \max\{AM(\omega_l)\} > 8 \tag{3.16}$$

then if

$$\frac{\omega_c - l\omega_0}{l\omega_0} < 11\% \tag{3.17}$$

and

$$\psi = \min\left(\frac{SFM}{SFM_{\max}}, 1\right) > 0.1 \tag{3.18}$$

frequency was declared voiced; otherwise ω_c was declared unvoiced. The quantity Spectral Flatness Measure (SFM) denotes the spectral flatness measure as defined in:

$$SFM = 10\log 10\left(\frac{G_m}{A_m}\right) \tag{3.19}$$

where G_m and A_m are the geometric mean and arithmetic mean of the power spectrum in the range $[\omega_c - \omega_0/2, \omega_c + \omega_0/2]$ [2]. In this software SFMmax was set to -50 dB, which indicates that the signal is entirely tone like. Having a classified frequency as voiced or as unvoiced, then the interval $[\omega_c - \omega_0/2, \omega_c + 3\ \omega_0/2]$ was searched for its largest peak and the same "harmonic test" was applied. The process was continued throughout the speech band. The ψ parameter was used as a tonality test. The advantage of the tonality test was to effectively remove spurious

peaks caused by white noise [2]. MVF is the maximum voiced frequency or peak frequency in a spectral chunk.

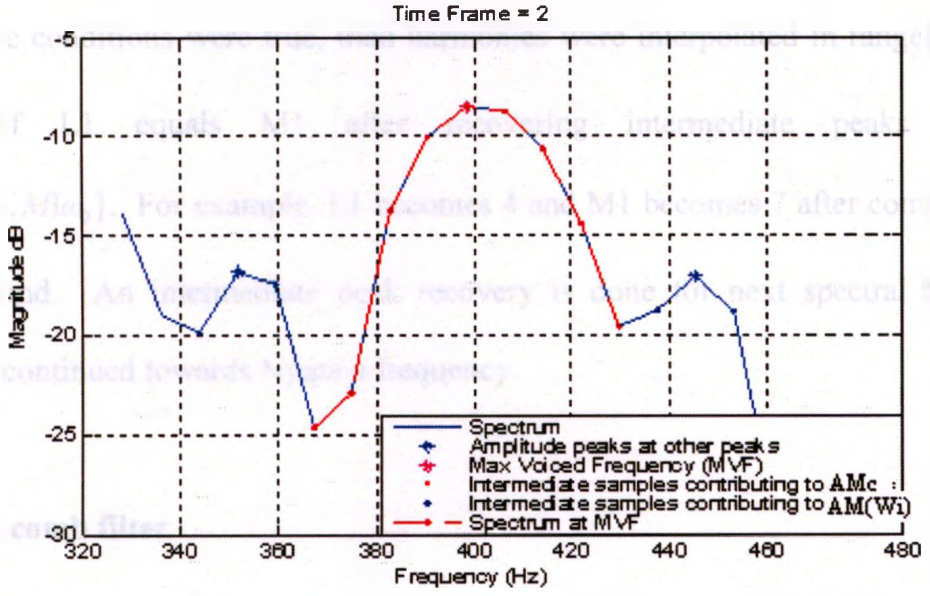

Figure 3.9: Part of the spectrum within a time frame subjected to harmonic peak test.

3.3.6 Post-processing

The post-processing is done in three stages:

1. Interpolation of a single harmonic peak: A local peak was declared a harmonic peak if its frequency was within 15% of, the nearest harmonic frequency and there were at least three peaks before and two peaks after it [2].
2. Rejection of isolated peaks: A harmonic peak was rejected if its distance to the nearest neighboring peaks was either less than $0.85\omega_0$ or greater than $1.15\omega_0$.
3. Recovery of multiple submerged intermediate peaks: Assume $L1$ and $M1$ be some positive integers $L1+3 \leq M1$. If $L1$=1, then M needs to be minimum value of 4. Multiple harmonic peaks were interpolated if

- There were no peaks picked in the frequency range $[L1\omega_0, M1\omega_0]$.

- There were at least three good harmonic peaks in the range $[0, L1\omega_0]$ and at least another three harmonics in $[M1\omega_0, \pi]$ [2]

If both of above conditions were true, then harmonics were interpolated in range $[L1\omega_0, M1\omega_0]$. The value of L1 equals M1 after recovering intermediate peaks within the bandwidth $[L1\omega_0, M1\omega_0]$. For example, L1 becomes 4 and M1 becomes 7 after computing for the first spectral band. An intermediate peak recovery is done for next spectral band and the computation is continued towards Nyquist frequency.

3.3.7 Adaptive comb filter

After finding as many additional frequency locations of harmonic peaks as possible, an adaptive comb filter was designed. In the first step, an initial comb filter was implemented in the frequency domain as:

$$H_1(\omega_k) = \begin{cases} B_c e^{\frac{-2(\omega_k - \omega_c)^2}{\sigma}} & , \omega_k \in [\omega_c - \omega_0/2, \omega_c + \omega_0/2] \\ B_k, & Otherwise \end{cases} \tag{3.20}$$

where ω_c was the peak frequency as determined by the modified peak-picking method and post-processing. $H_1(\omega_k)$ was the frequency response of the initial comb filter at frequency ω_k. σ controls the width of the comb filter and was set to 0.002 in our implementation.

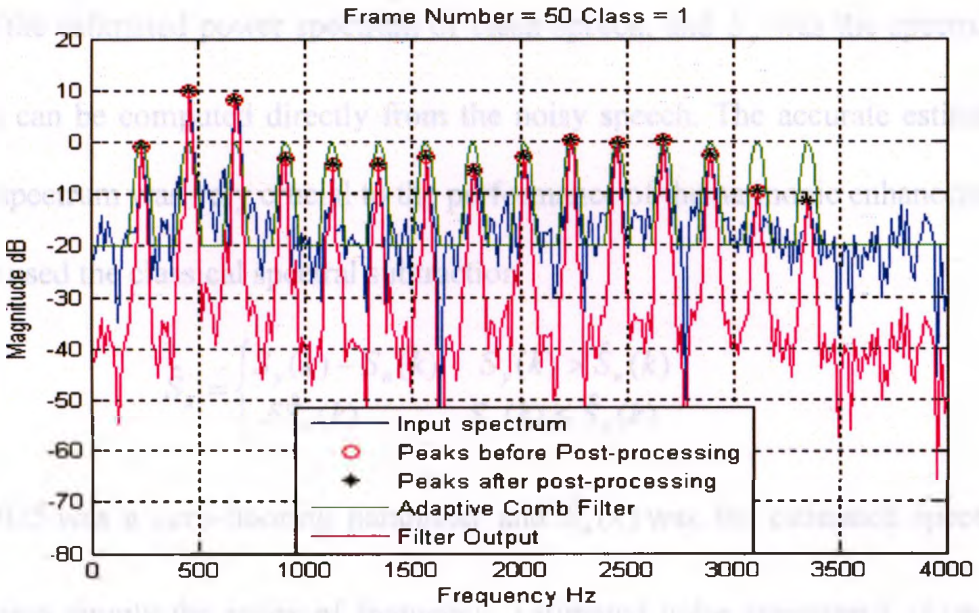

Figure 3.10: Adaptive comb filter (min ACF gain -20dBFS)

The quantity B_c specifies the filter gain at peak frequency ω_c. The comb structures were only implemented within the vicinity of one fundamental frequency range centered at the peak frequency. The value of B_k determines the filter response outside the frequency range. Since there were many design choices for the gain factor, designs of B_c and B_k were also flexible [2].

3.3.8 Spectral Subtraction

The B_c and B_k gains were implemented as Wiener-type gains

$$B_c = \left[\frac{\hat{S}_x(\omega_c)}{S_y(\omega_c)}\right]^2 \tag{3.21}$$

$$B_k = \left[\frac{\hat{S}_x(\omega_k)}{S_y(\omega_k)}\right]^3 \tag{3.22}$$

where $\hat{S}_x$ was the estimated power spectrum of clean speech, and $\hat{S}_y$ was the spectrum of noisy speech which can be computed directly from the noisy speech. The accurate estimation of the clean speech spectrum was very crucial to the performance of the harmonic enhancement method [2]. We have used the classical spectral subtraction

$$\hat{S}_x = \begin{cases} S_y(k) - S_n(k), & S_y(k) > \hat{S}_n(k) \\ \delta \hat{S}_n(k), & S_y(k) \le \hat{S}_n(k) \end{cases} \tag{3.23}$$

where $\delta = 0.025$ was a zero-flooring parameter and $\hat{S}_n(k)$ was the estimated spectrum of the noise. $\hat{S}_n(k)$ was simply the index of frequency. Estimated noise spectrum $\hat{S}_n(k)$ was obtained from the initial noise only frames. Eventually, the gain factor was obtained by

$$\chi_k = \max(H1(\omega_k), -20\,dB) \tag{3.24}$$

The minimum adaptive comb filter response was mentioned to be -20 dBFS in [2] according to equation (3.24). For silent frames, it was mentioned as -30 dBFS. However, in the HSE speech enhancement or logMMSE-HSE system, -36 dB was set as the minimum response for the dereverberation system and -6 dB or -36 dB for the noise reduction system as mentioned in the section on software settings. These values give best possible sound quality and objective scores.

3.3.9 Software settings: Noise reduction & Dereverberation

In order to achieve best possible objective results, the software parameter settings in Table 3.1 need to be used. 'min ACF Gain' is the minimum adaptive comb filter response gain.

Algorithm	Speech Enhancement Focus	min ACF Gain (dBFS)	Noise Estimate Updates	Log Frequency Gain	Input Stimuli
logMMSE	Noise reduction	N/A	Yes	optional	Noisy speech
HSE	Noise reduction	-36	No	ON	Noisy speech
logMMSE+ HSE	Noise reduction	-6	No	ON	Noisy speech
MB + HSE	Noise reduction	-6	No	ON	Noisy speech
logMMSE	Dereverberation	N/A	Yes	optional	Speech degraded by Reverberation
HSE	Dereverberation	-36	No	ON	Speech degraded by Reverberation
logMMSE+ HSE	Dereverberation	-36	No	ON	Speech degraded by Reverberation
MB + HSE	Dereverberation	-36	No	ON	Speech degraded by Reverberation

Table 3.1: Software parameter settings (Speech Enhancement)

It turned out that using the VAD for HSE and logMMSE plus HSE algorithms, had a negative effect on getting high objective MOS scores. In noise-only frames, no useful harmonic peaks were picked by the harmonic peak test subroutine. In such cases, the response of the comb filter is flat. Therefore, the adaptive comb filter suppresses the noise heavily in noise only frames. Hence, there was no need to deploy a VAD. The adaptive comb filter indirectly does the job of a VAD. Since VAD was not used, noise estimate updates were also not involved in the getting the objective scores reported in chapter 4.

The log-frequency spectrum of the FDNFP can be optionally multiplied with a gain factor of 1.5 (as shown in Figure 3.11) after filtering the log-magnitude spectrum with ACF coefficients and just before reconstructing the time domain signal using IFFT. This operation boosts the peaks further and suppresses noise contents further. Care must be taken to multiply this gain factor

only in the log frequency domain spectrum. Doing this, significantly improves objective MOS scores in many cases.

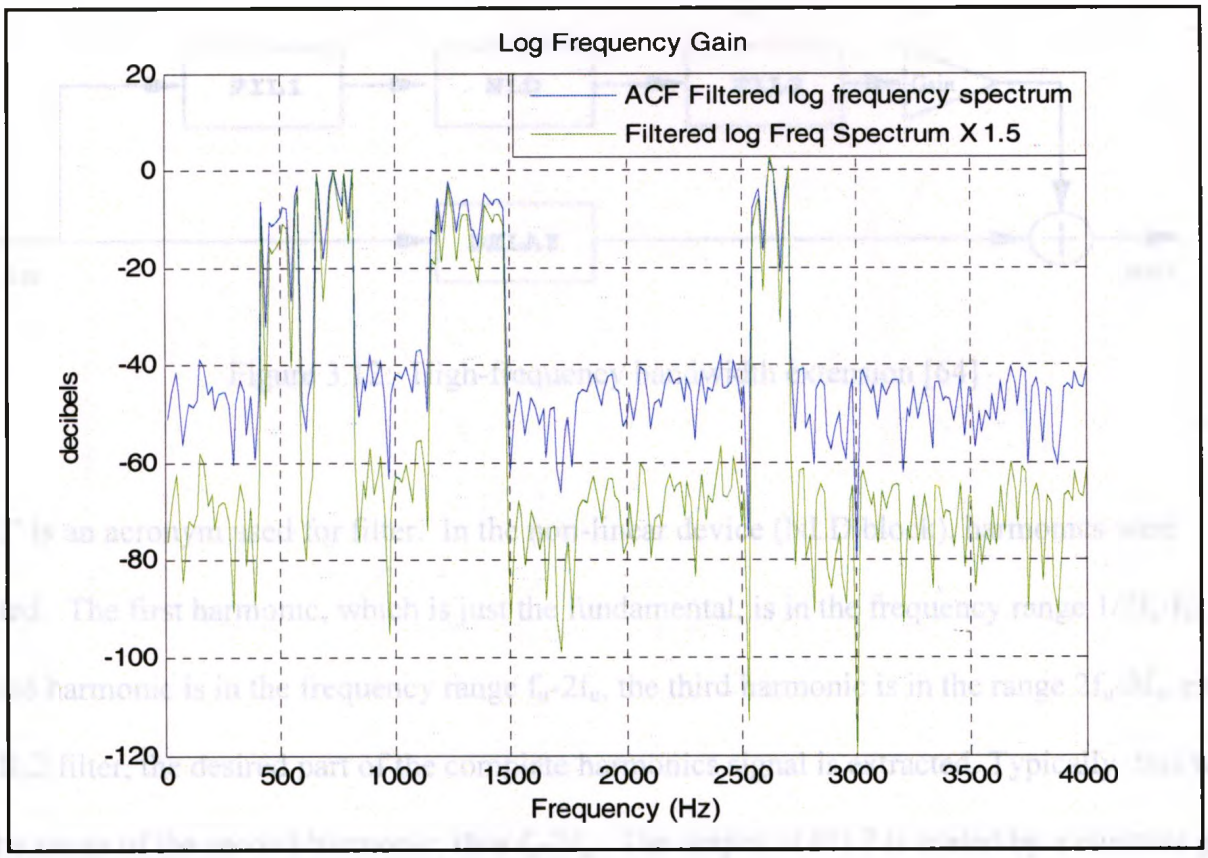

Figure 3.11: Multiplying log-Frequency spectra by gain 1.5 just before signal reconstruction

3.4 Bandwidth Extension (BWE) Implementation

This algorithm reconstructs HF harmonics but complies with requirements of low computational complexity, low memory requirements, independence of signal format (PCM), applicability to music and speech and no requirement of *a priori* knowledge about the missing high frequencies. Figure 3.12 displays the proposed processing scheme. There are two signal branches, the lower of which passes the input signal unprocessed. The spectrum extension and all signal processing

takes place in the upper branch. Using FIL1 filter, the highest octave present in the signal is extracted *viz* $1/2f_u$-f_u, where f_u is the upper frequency limit of the input signal.

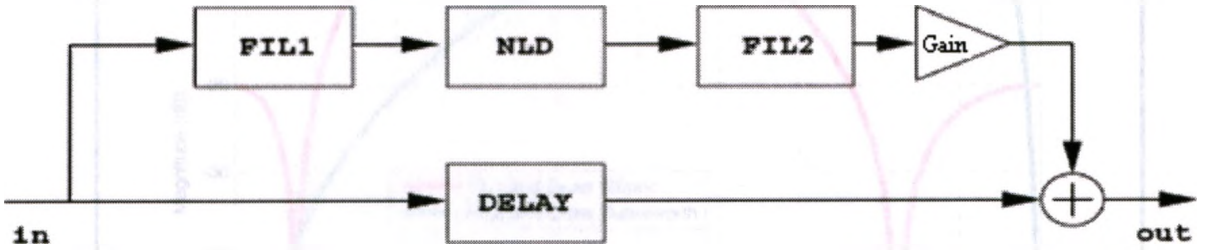

Figure 3.12: High-frequency bandwidth extension [64]

"FIL" is an acronym used for filter. In the non-linear device (NLD block), harmonics were created. The first harmonic, which is just the fundamental, is in the frequency range $1/2f_u$-f_u; the second harmonic is in the frequency range f_u-$2f_u$, the third harmonic is in the range $2f_u$-$3f_u$, etc. In FIL2 filter, the desired part of the complete harmonics signal is extracted. Typically, this will be the range of the second harmonic, thus f_u-$2f_u$. The output of FIL2 is scaled by a constant gain factor Gain = 0.5. The processed delayed signal is added to the direct unprocessed signal.

The HF limit of the output signal now equals $2f_u$, double that of the input signal. Depending on the application, the filters FIL1 and FIL2 may be fixed or signal dependent. The non-linear device NLD is the element that creates the additional high frequencies in the output spectrum [64]. The target is to add only the next highest octave to the input spectrum. Therefore, a nonlinear device that generates mainly the second harmonic is needed. Independent of signal level, the system should add the same amount of harmonics to the signal. Therefore, amplitude linearity is desirable.

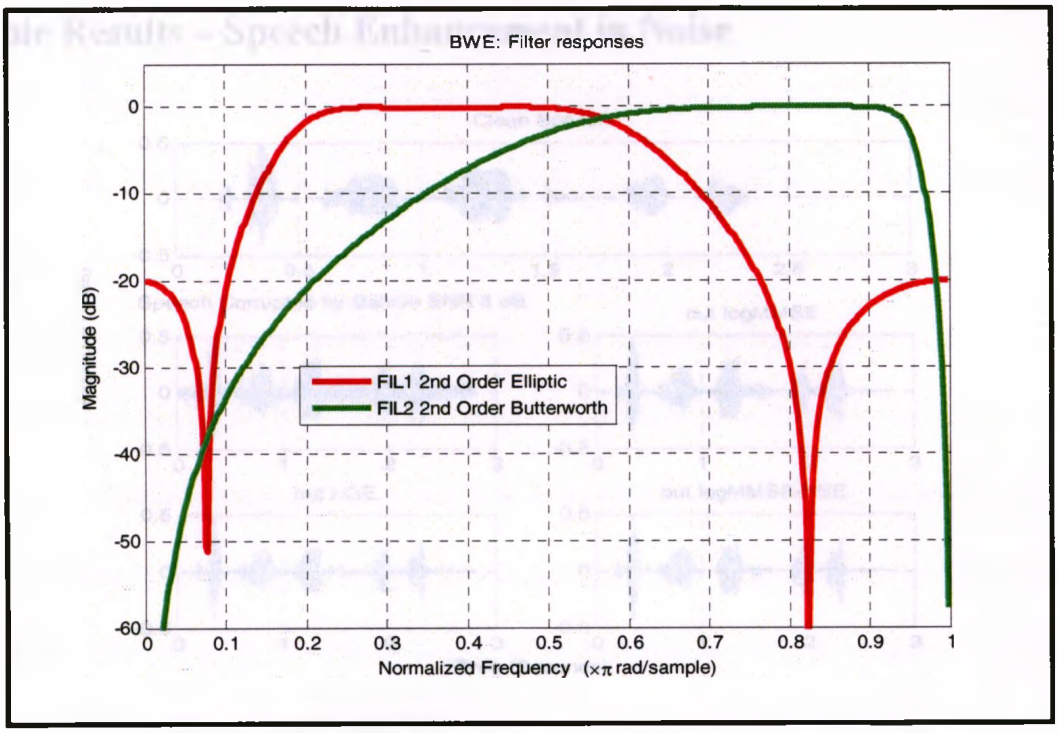

Figure 3.13: Squared filter magnitudes, FIL1 and FIL2 [64]

A full-wave rectifier has both these characteristics and is therefore highly suitable for use as non-linear device in the scheme of Figure 3.12. On a negative note, the non-linear processing besides generating harmonic frequencies also introduces inter-modulation distortion. In some situations this can give rise to audible artifacts [64]. If an appropriate delay is also used in the lower branch, the two signal branches will add exactly in phase [64]. This has the advantage that transients in the input signal will remain compact in the output (because of the filter's constant group delay), which is beneficial for perceptual quality. Therefore, filters FIL1 and FIL2 should be either FIR filters, or linear phase IIR filters (using time-forward and time-reversed filtering filtfilt function in MATLAB), which may be more efficient [64]. In the thesis code, an elliptic filter of the order 2 was chosen as a choice for FIL1 and a second order Butterworth filter was chosen for FIL2.

3.5 Sample Results – Speech Enhancement in Noise

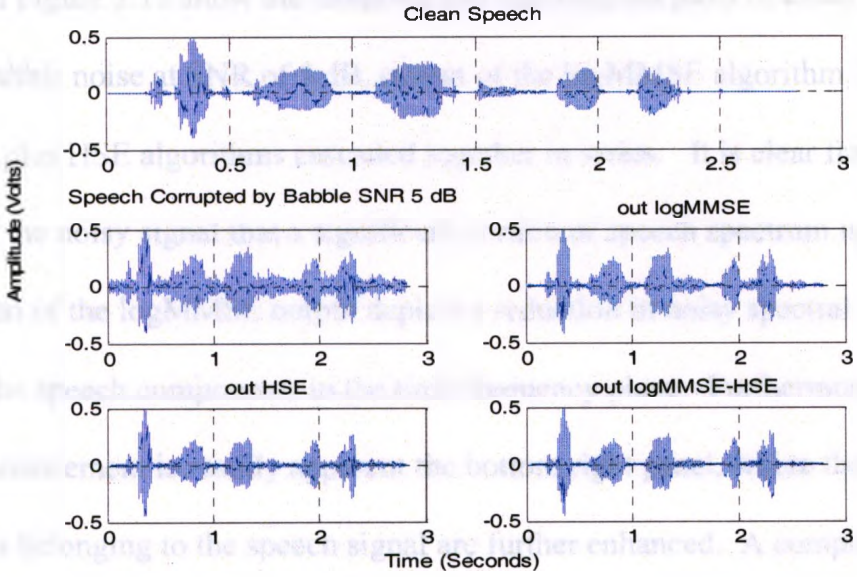

Figure 3.14: Time Domain output of different algorithms

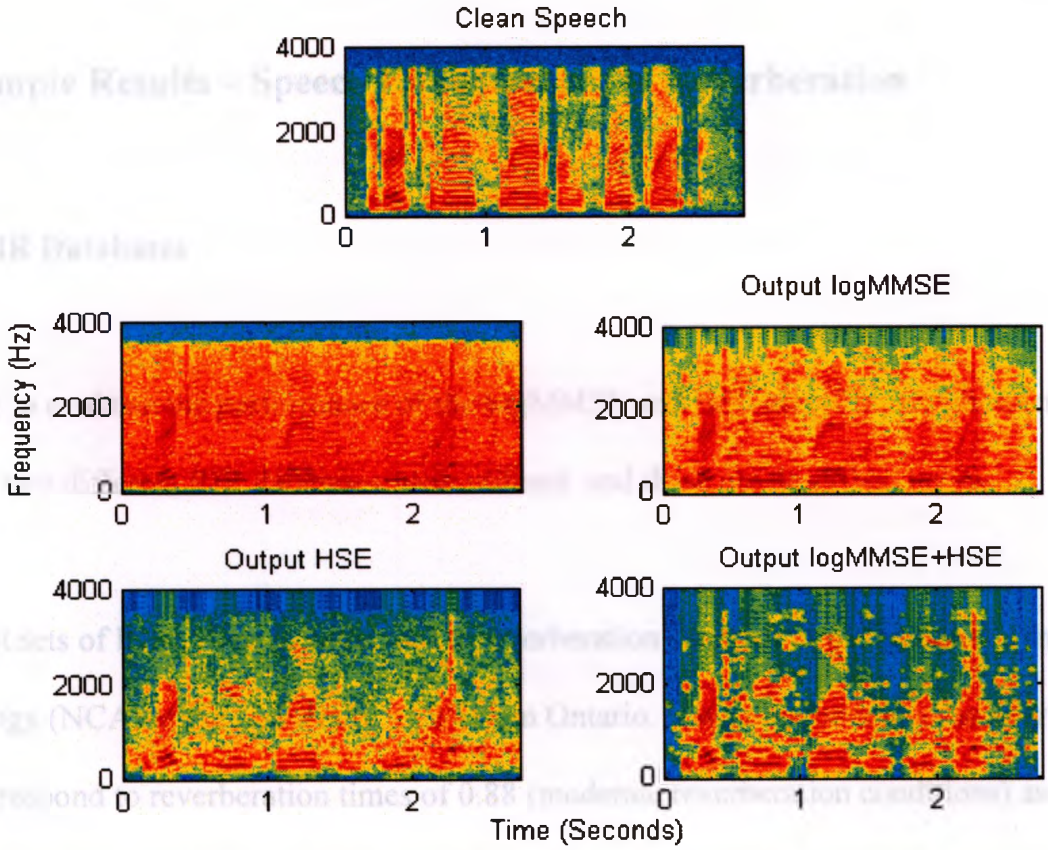

Figure 3.15: Sonogram of output of different algorithms

Figure 3.14 and Figure 3.15 show the temporal and spectrogram plots of clean speech, speech corrupted by babble noise at SNR of 5 dB, output of the logMMSE algorithm, HSE algorithm, and logMMSE plus HSE algorithms cascaded together in series. It is clear from the spectrogram of the noisy signal that a significant portion of speech spectrum is masked by noise. The spectrogram of the logMMSE output depicts a reduction in noisy spectral components and a betterment of the speech components in the time-frequency plane. Furthermore, the application of harmonic enhancement is readily apparent the bottom right panel, where the dominant harmonic peaks belonging to the speech signal are further enhanced. A complete objective characterization of the logMMSE and HSE algorithms across a number of noisy environments is given in Chapter 4.

3.6 Sample Results – Speech Enhancement in Reverberation

3.6.1 RIR Databases

In order to evaluate the performance of the logMMSE and HSE algorithms with reverberated speech, two different RIR databases were utilized, and these are described below.

The first sets of RIRs were recorded at the reverberation chamber at the National Centre for Audiology (NCA) at the University of Western Ontario. Two reverberation chamber settings that correspond to reverberation times of 0.88 (moderate reverberation conditions) and 1.39 seconds (severe reverberation conditions) were set up and impulse responses were collected.

The impulse responses were sampled at 44.1 kHz and later downsampled to 8 kHz or 16 kHz. These impulse responses were convolved with the clean speech signals sampled at 8 kHz to create the necessary reverberant speech samples.

The Aachen Impulse Response (AIR) [65] database is a set of impulse responses that were measured in a wide variety of rooms. The first version of this database was published in 2009 and included binaural room impulse responses (BRIR) measured with a dummy head in different locations with different acoustical properties, such as reverberation time and room volume. In a first update, the database was extended to BRIRs with various azimuth angles between head and desired source.

All impulse responses of the AIR database are stored as double-precision binary floating-point MAT-files which can be directly imported into MATLAB. Additionally, a load function (load_air.m) as well as an example script (load_air_example.m) was provided in [65] to allow for a rapid integration into existing evaluation frameworks.

Room #	Type	Speaker-Microphone Distances
1	Booth	{0.5m, 1m, 1.5m}
2	Office	{1m, 2m, 3m}
3	Meeting	{1.45m, 1.7m, 1.9m, 2.25m, 2.8m}
4	Lecture	{2.25m, 4m, 5.56m, 7.1m, 8.68m, 10.2m}
5	Stairway	{1m, 2m, 3m}

Table 3.2: AIR database settings and options

A MATLAB script was used to choose room options, speaker-microphone distances and AIR azimuth of 90 degrees was set for front (range [0 180] in 15 degree increments). The impulse

responses were first loaded and sampled at 44.1 kHz and later downsampled to 16 kHz. Once the appropriate binaural impulse response was loaded for a set of parameters, clean stereo speech files were convolved with the impulse responses and the output speech corrupted by reverberation was written to hard disk and finally downsampled again by a factor of 2 to get reverberation corrupted speech sampled at 8 kHz.

These corrupted files were passed as Inputs to the logMMSE-HSE dereverberation algorithm, and the outputs of the algorithm were stored for objective analysis using SRMR. The objective SRMR analysis results are discussed in Chapter 4, and detailed tables are published in the appendix.

3.6.2 Sample Results

Figure 3.16 displays the results of applying the combination of the logMMSE and HSE algorithms to reverberant speech generated using the NCA RIR database. The impact of reverberation on speech components can be seen in the subplots in the second column of this plot, where the well-defined harmonic content and the smooth formant transitions are obscured by reverberation, more so at higher reverberation time. Application of the enhancement algorithm appears to retrieve some of the harmonic content.

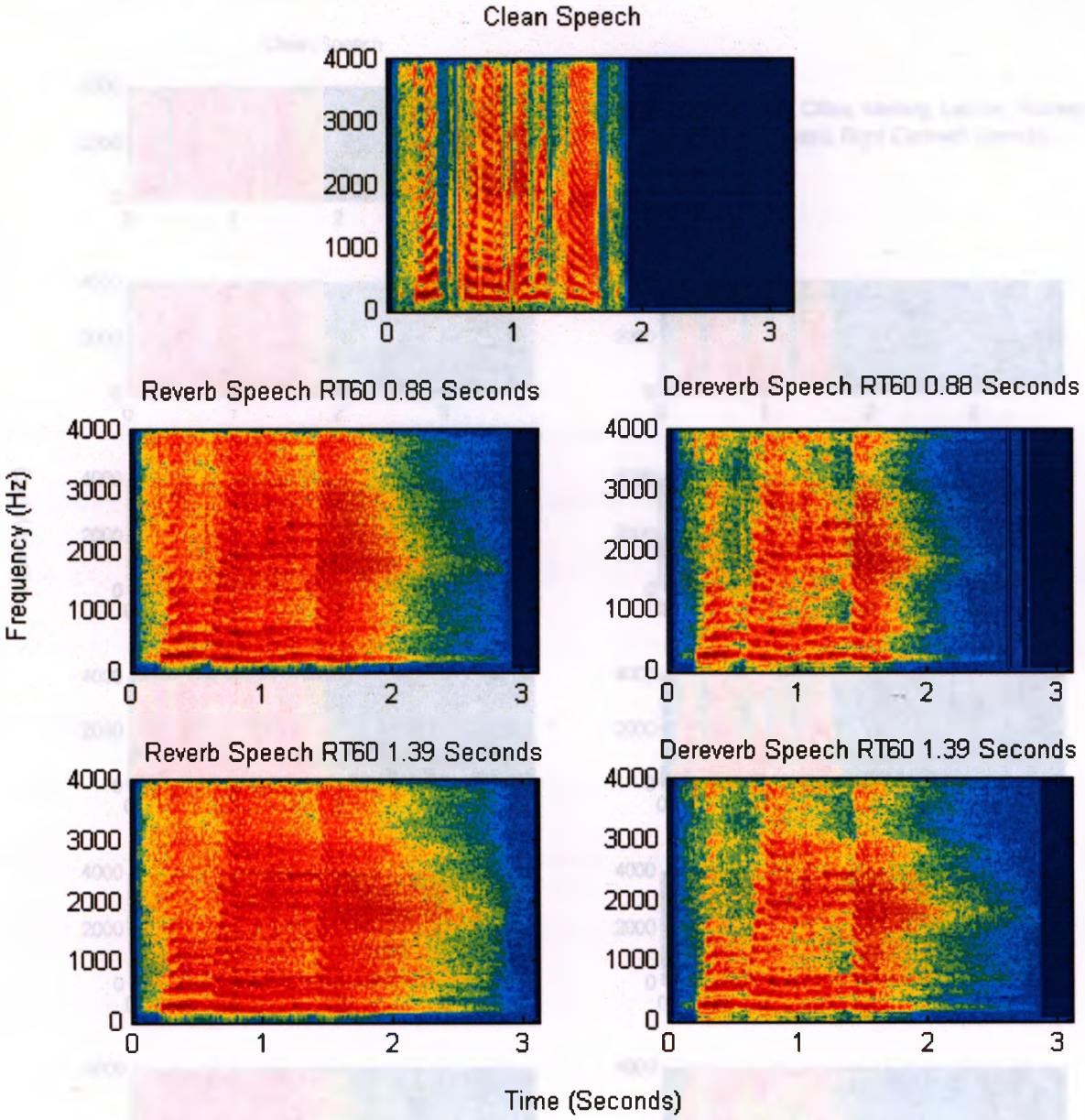

Figure 3.16: Dereverberation using logMMSE-HSE (NCA Impulse Response)

Figure 3.17 depicts a similar set of results with the AIR database. From the left column of this Figure, it can be observed that a progressive increase in the reverberation time results increased smudging of the speech components in the time-frequency plane. The speech enhancement algorithms appear to restore some of the structural elements of the underlying clean speech signal.

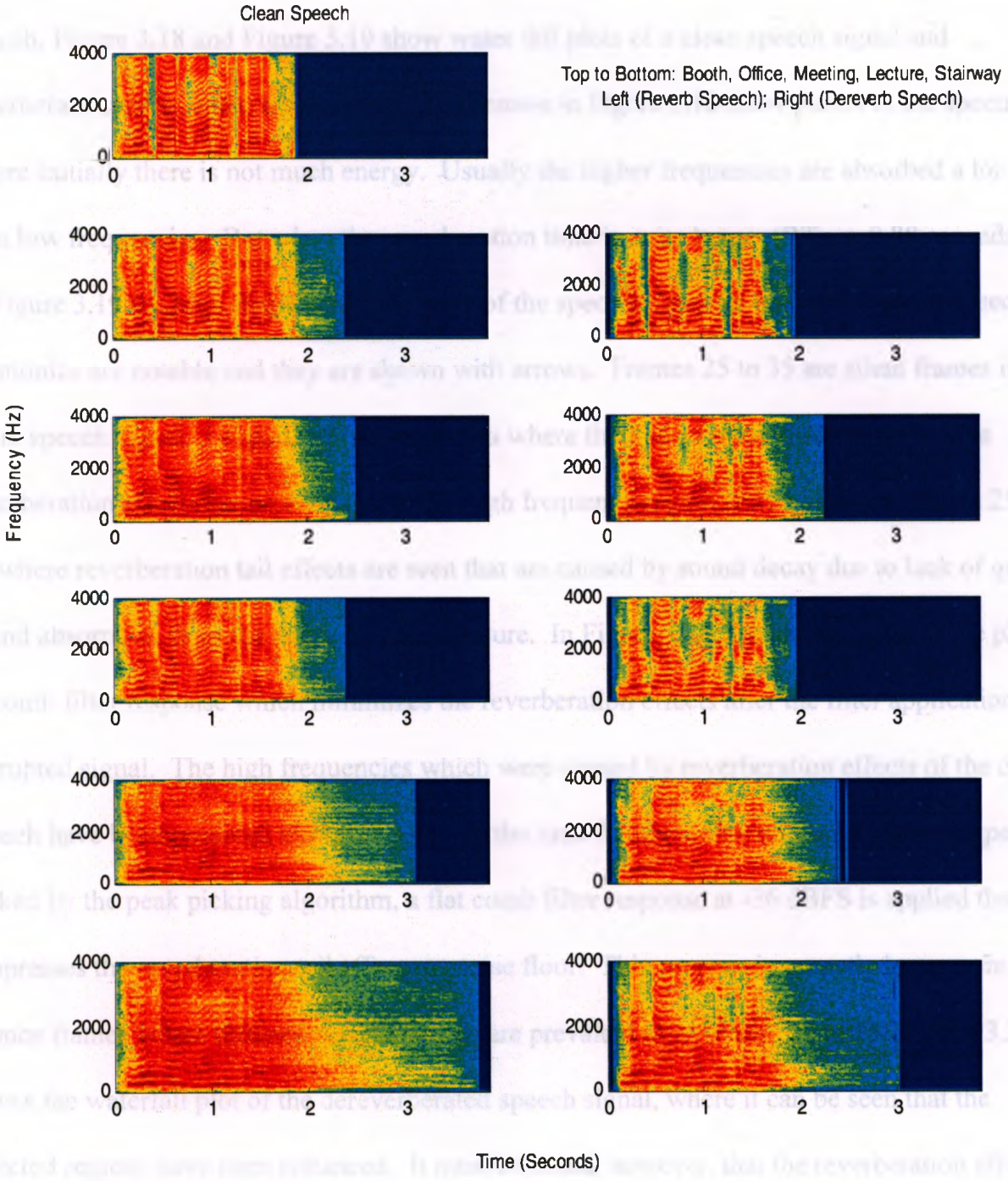

Figure 3.17: Dereverberation: AIR database Spectrograms

Speaker-Microphone distances used: Booth (1.5m), Office (3m), Meeting (1.9m), Lecture (5.56m) and Stairway (3m).

For a closer look at the performance of the speech enhancement algorithms with reverberant speech, Figure 3.18 and Figure 3.19 show water fall plots of a clean speech signal and reverberant speech signal respectively. The arrows in Figure 3.18 show places in the spectrum where initially there is not much energy. Usually the higher frequencies are absorbed a lot faster than low frequencies. But when the reverberation time is quite longer (RT_{60} = 0.88 seconds) as in Figure 3.19 the high frequencies and parts of the spectrum that do not contribute to speech harmonics are notable and they are shown with arrows. Frames 25 to 35 are silent frames in the clean speech signal, .i.e. parts of the wave data where there is no meaningful speech. The reverberation effects are pronounced in the high frequency region and in between frames 25 to 35 where reverberation tail effects are seen that are caused by sound decay due to lack of quick sound absorption from the surrounding enclosure. In Figure 3.20, the arrows point to the places in comb filter response which minimizes the reverberation effects after the filter application to corrupted signal. The high frequencies which were caused by reverberation effects of the clean speech have been removed in Figure 3.21. In the time frames where no useful harmonic peak is picked by the peak picking algorithm, a flat comb filter response at -36 dBFS is applied that suppresses the reverberation tail effects to noise floor. This suppression mostly happens in silence frames where diffused reverberations are prevalent as in frames 25 to 35. Figure 3.21 shows the waterfall plot of the dereverberated speech signal, where it can be seen that the affected regions have been enhanced. It must be noted, however, that the reverberation effect is not completely removed. The time frames of the speech signal used in this waterfall plot example, do not start at the actual 0^{th} frame or the beginning of the sound. It was arbitrarily focused or zoomed on to the end of a speech syllable and start of a consecutive speech syllable to show the reverb tail effects in the silent/noise only frame inbetween speech syllables.

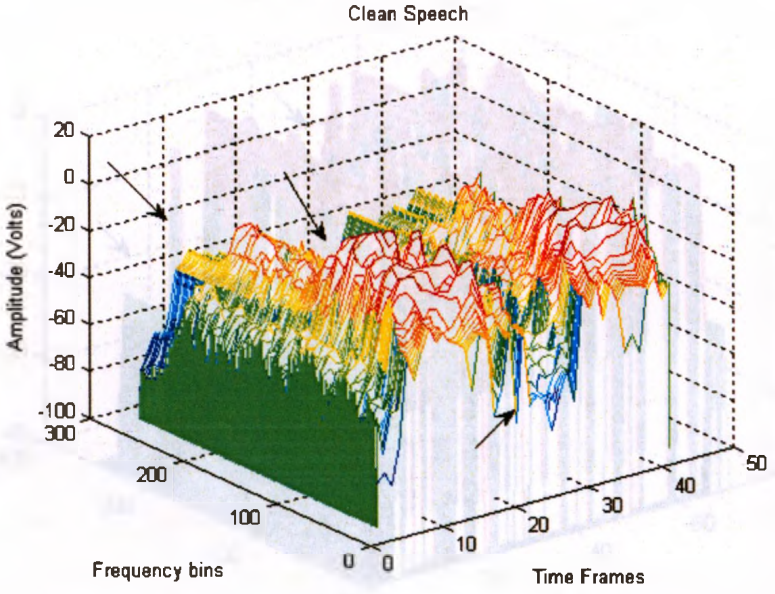

Figure 3.18: Waterfall plot of clean signal

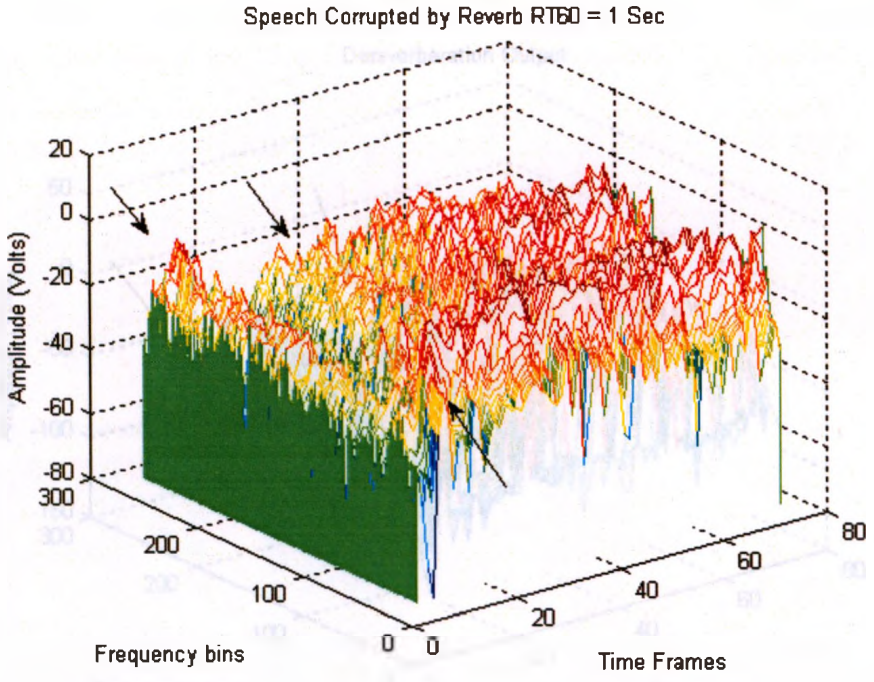

Figure 3.19: Waterfall plot of speech corrupted by Reverb (RT60 = 0.88 sec)

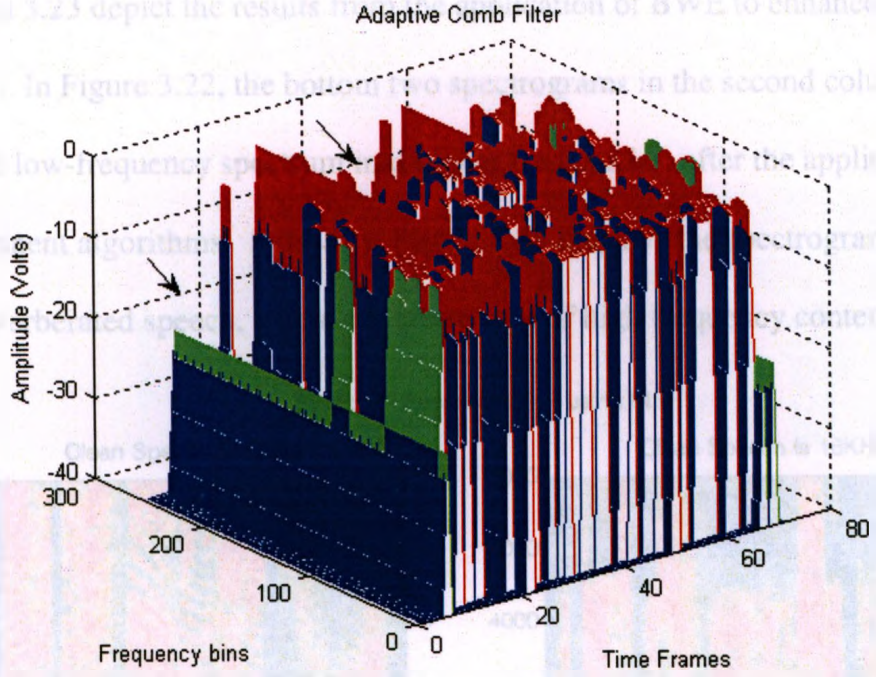

Figure 3.20: Waterfall plot of ACF with flat response at -20dBFS where there is a decay of reflections

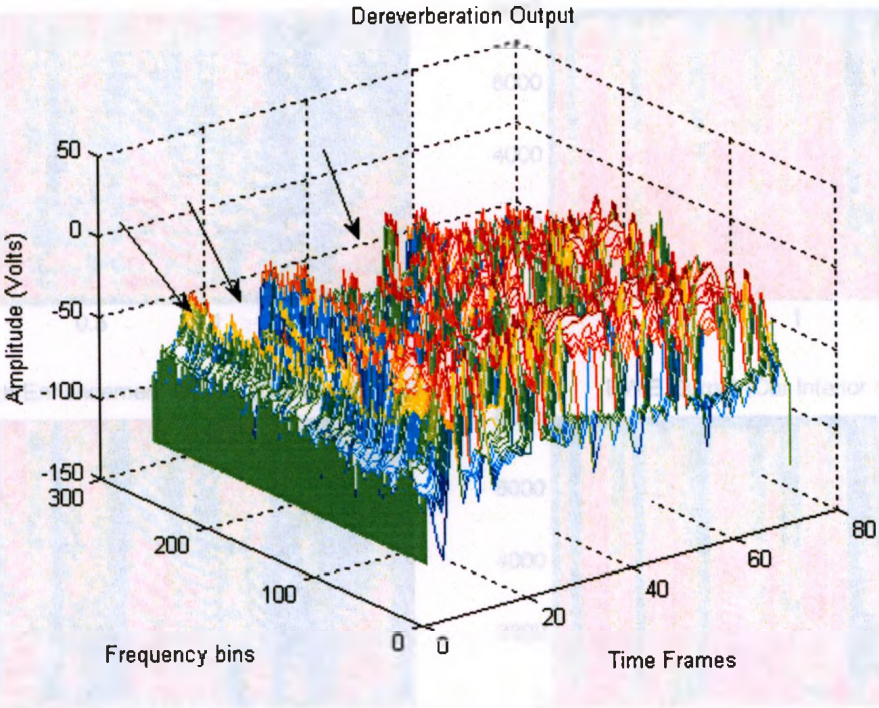

Figure 3.21: Waterfall plot of Dereverberation output effects of reverb removed

Figures 3.22 and 3.23 depict the results from the application of BWE to enhanced narrowband speech samples. In Figure 3.22, the bottom two spectrograms in the second column show the extension of the low-frequency spectrum into higher frequencies, after the application of the speech enhancement algorithms. Similarly, Figure 3.23 displays the spectrograms of BWE applied to dereverberated speech, where the generation of high frequency content is notable.

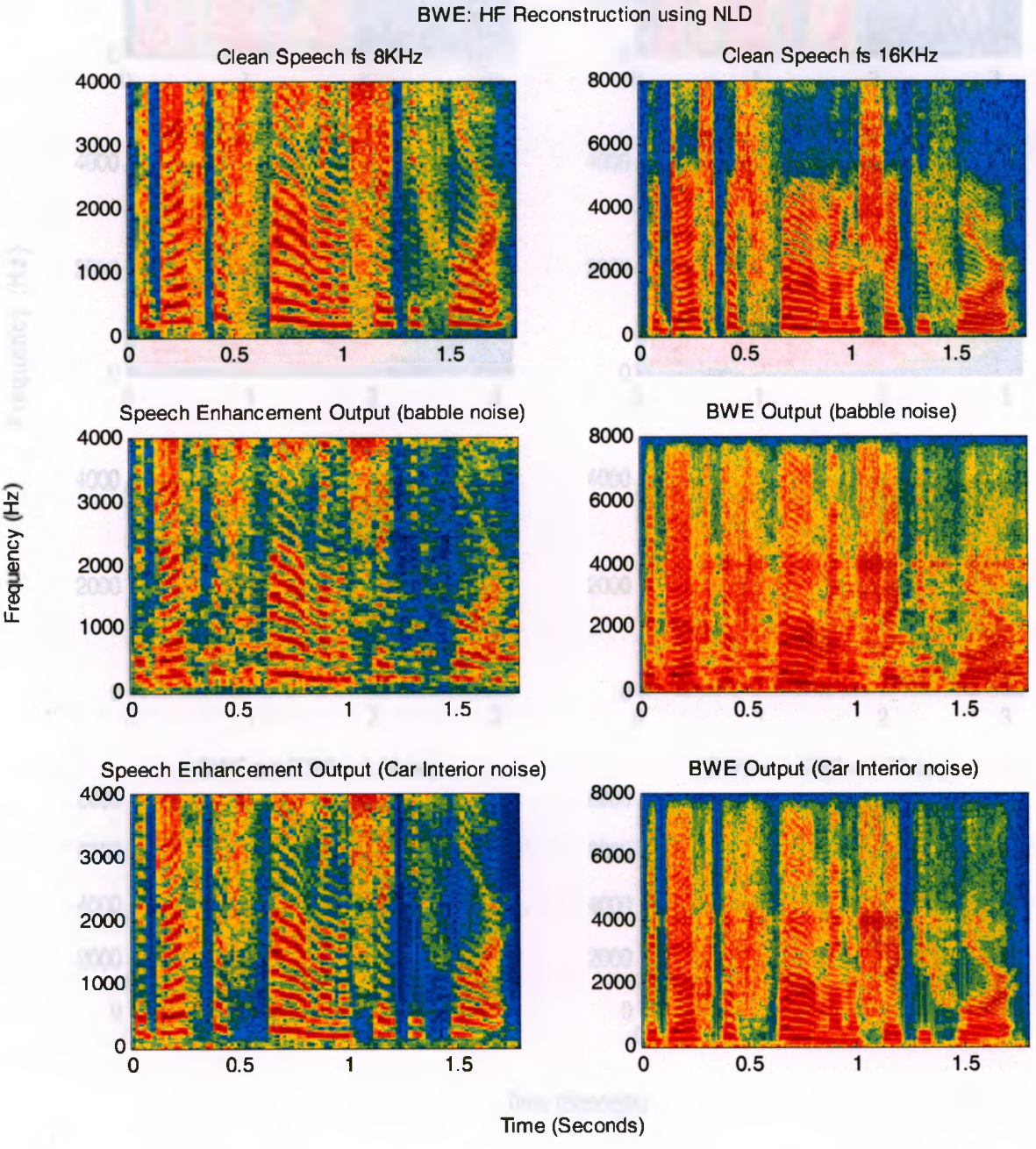

Figure 3.22: Noise reduction-BWE output Spectrograms (for SNR 5dB)

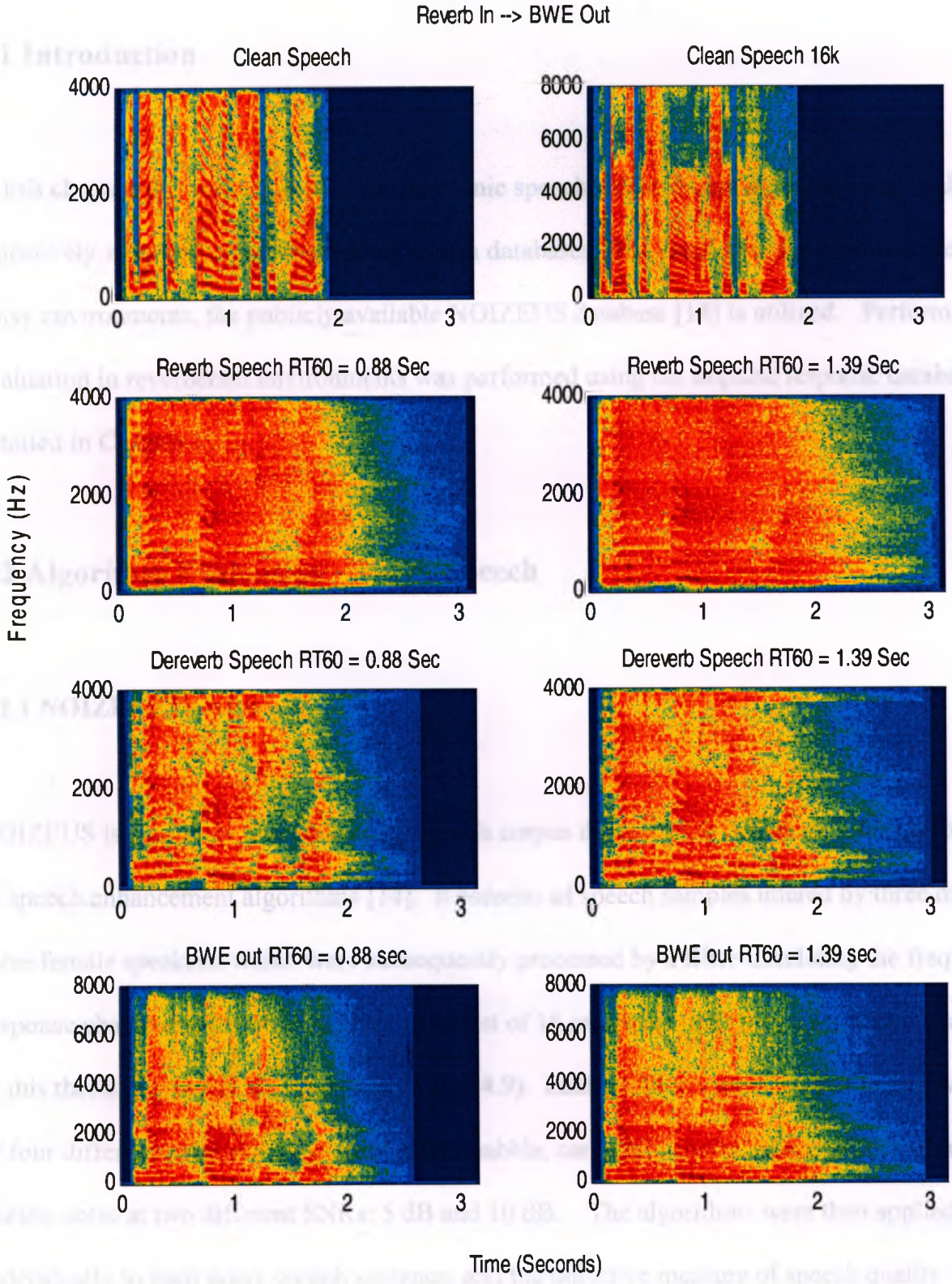

Figure 3.23: Reverberant speech, dereverberated speech, and BWE (NLD) speech.

Chapter 4 : Algorithm Evaluation

4.1 Introduction

In this chapter, the performance of the harmonic speech enhancement algorithm was evaluated objectively with noisy and reverberant speech databases. For evaluating the performance in noisy environments, the publicly available NOIZEUS database [14] is utilized. Performance evaluation in reverberant environments was performed using the impulse response databases detailed in Chapter 3.

4.2 Algorithm Evaluation – Noisy Speech

4.2.1 NOIZEUS Database

NOIZEUS is a publicly available noisy speech corpus developed for evaluating the performance of speech enhancement algorithms [14]. It consists of speech samples uttered by three male and three female speakers, which were subsequently processed by a filter simulating the frequency response characteristics of telephones. The list of 16 sentences used for performance evaluation in this thesis is given in the Appendix (Table 4.9). Each of these sentences was mixed with one of four different noise samples: multi-talker babble, car interior noise, street noise, and train station noise at two different SNRs: 5 dB and 10 dB. The algorithms were then applied individually to each noisy speech sentence, and the objective measure of speech quality was estimated for the enhanced speech sample using the procedure described below.

4.2.2 Comparisons to Other Algorithms

As discussed in Chapter 2, Hu and Loizou [12], [13] evaluated a number of speech enhancement algorithms both objectively and subjectively. These algorithms are tabulated in Table 4.1, and although these algorithms have not been thoroughly studied in this thesis, their objective speech quality scores are included in the subsequent results for comparative purposes, and to position the results from HSE in context.

KLT	Karhunen-Loeve Transform
pKLT	Perceptual Karhunen-Loeve Transform
MMSE SPU	Minimum Mean Square Estimation Speech Presence Uncertainty
logMMSE	Log Minimum Mean Square Estimation
logMMSE SPU	Log Minimum Mean Square Estimation Speech Presence Uncertainty
pMMSE	Speech Enhancement based on perceptually motivated Bayesian Estimators of the Magnitude Spectrum
RDC	Spectral Subtraction using Reduced Delay Convolution and Adaptive Averaging.
RDC-ne	RDC Algorithm That Included Noise Estimation
MB	Multi-Band Spectral Subtraction
WT	Speech Enhancement based on Wavelet Thresholding the Multitaper Spectrum
Wiener-as	Speech Enhancement based on A Priori Signal To Noise Estimation
AudSup	Speech Enhancement based on Audible Noise Suppression

Table 4.1: List of speech enhancement algorithms included for comparative purposes.

4.2.3 Results

Figures 4.1 and 4.2 depict the predicted speech quality scores averaged over the 16 speech sentences for the multi-talker babble condition at 5 dB and 10 dB SNRs respectively. Table 6 in [12] presents results obtained from comparative statistical analysis of overall quality (OVRL)

scores. The table mentions list of algorithms that perform equally well despite variations in the subjective MOS scores. The subjective scores of such speech enhancement algorithms are plotted in blue and magenta colors in the bar plots of Figure 4.1 to Figure 4.8. Some algorithms mentioned in that table were said to perform poorly.

With logMMSE score as reference, Weiner-as was found to be the closest algorithm with MOS scores that indicated poor performance. The algorithms that had scores in between the scores of logMMSE and Weiner-as were said to be performing same as the logMMSE algorithm. There was a MOS score decrement of 0.25 between logMMSE and Weiner-as algorithms. Therefore, a change of 0.25 in MOS scores was found to be needed in order to sense a significant perceptual degradation of sound quality. Thereby, a confidence interval of 0.25 was introduced around logMMSE subjective MOS in the plots of the following figures and objective MOS of HSE, logMMSE-HSE and MB-HSE were plotted against it. Since the correlation between the subjective and objective MOS of logMMSE and Noisy were found to be very good, it seemed justifiable to have subjective MOS of different algorithm being compared against the objective scores of HSE and logMMSE plus HSE. A confidence interval of 0.25 and 0.20 were used for 5 dB and 10 dB SNRs respectively across all cases of background noises after analyzing the table 6 in [12]. Figure 4.2 has all the subjective and objective MOS differences labeled properly for clarification. It is seen that all cases of 5 dB SNRs irrespective of background noises, the logMMSE-HSE lies within the confidence interval but for most of the 10 dB SNR cases, the logMMSE-HSE objective MOS exceeds the 0.25 confidence interval which equates to significant perceptual improvement. In almost all cases of logMMSE-HSE, MOS improvements are seen as compared against logMMSE.

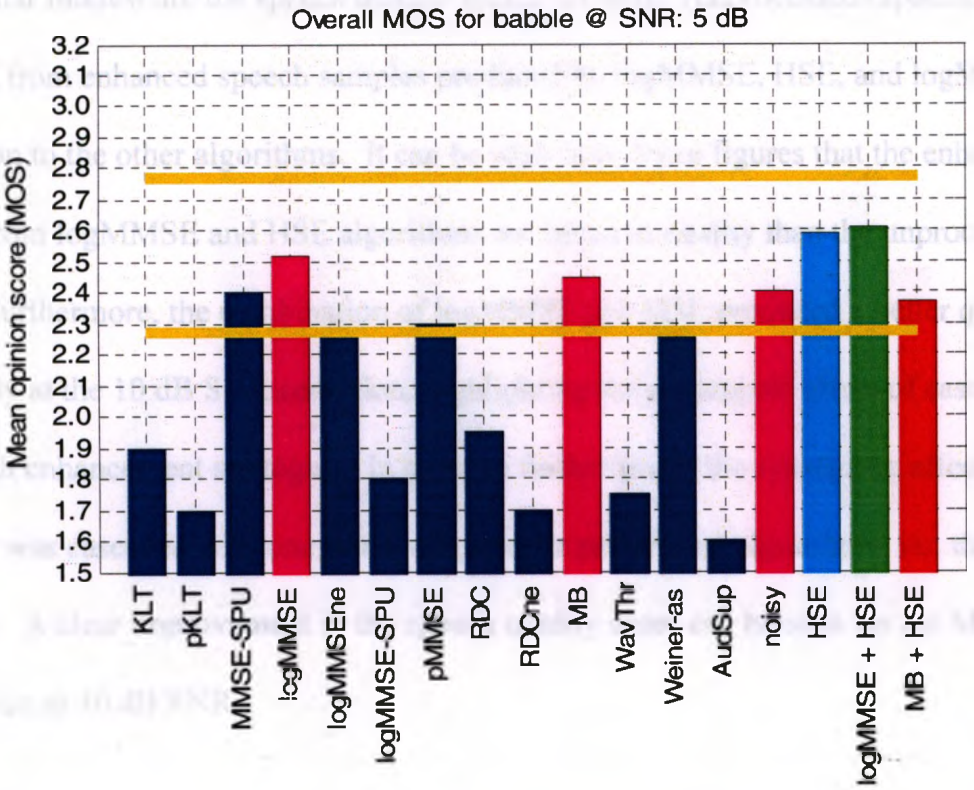

Figure 4.1: Predicted speech quality for different algorithms: multi-talker babble at 5 dB SNR.

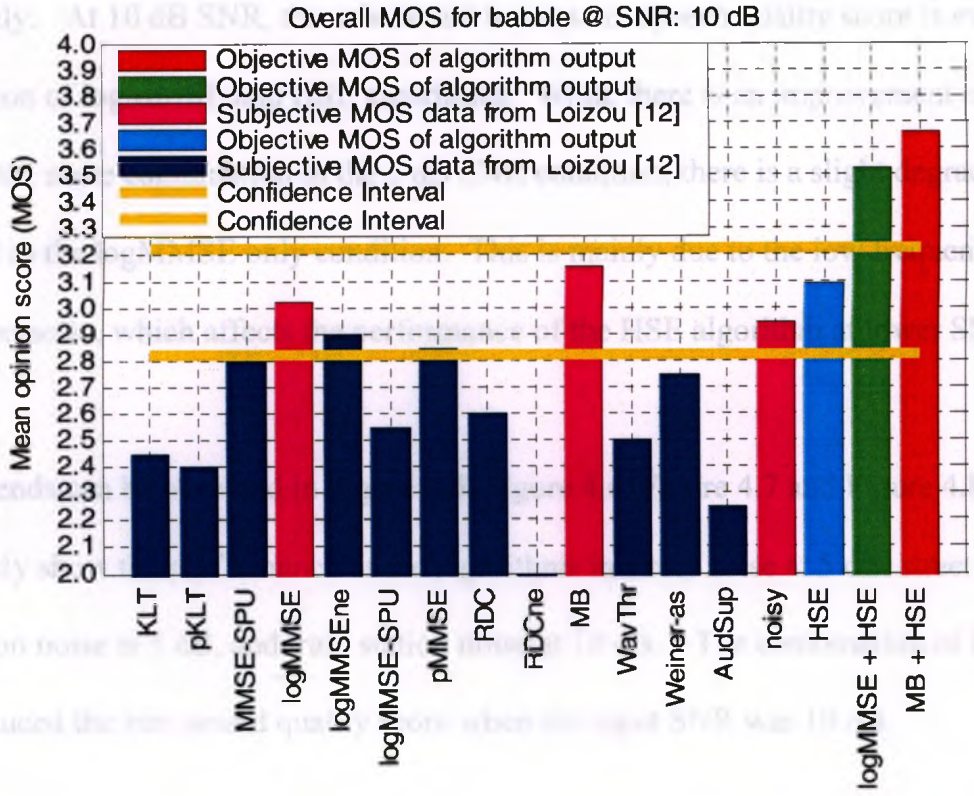

Figure 4.2: Predicted speech quality for different algorithms: babble at 10 dB SNR.

Of particular interest are the speech quality scores for noisy (unprocessed) speech, and the scores computed from enhanced speech samples produced by logMMSE, HSE, and logMMSE-HSE in comparison to the other algorithms. It can be seen from these figures that the enhanced speech samples from logMMSE and HSE algorithms are better in quality than the unprocessed noisy speech. Furthermore, the combination of logMMSE and HSE produced a better quality score, particularly at the 10 dB SNR condition, highlighting the synergistic effect of cascading these two speech enhancement strategies. In order to further probe the synergistic effect, the HSE algorithm was cascaded with one of the other better performing algorithms, *viz.* the MB algorithm. A clear improvement in the speech quality score can be seen for the MB + HSE combination at 10 dB SNR.

Figure 4.3 and Figure 4.4 display the results with car interior noise at 5 dB and 10 dB SNRs respectively. At 10 dB SNR, the substantial increase in speech quality score is evident with the combination of logMMSE and HSE algorithms. While there is an improvement in the quality score for the same combination in the 5 dB SNR condition, there is a slight degradation when compared to the logMMSE only condition. This is mainly due to the low frequency nature of the car interior noise, which affects the performance of the HSE algorithm at lower SNRs.

Similar trends can be observed in Figure 4.5, Figure 4.6, Figure 4.7 and Figure 4.8 which respectively show the performance of the algorithms in street noise at 5 dB, street noise at 10 dB, train station noise at 5 dB, and train station noise at 10 dB. The combination of logMMSE and HSE produced the best sound quality score when the input SNR was 10 dB.

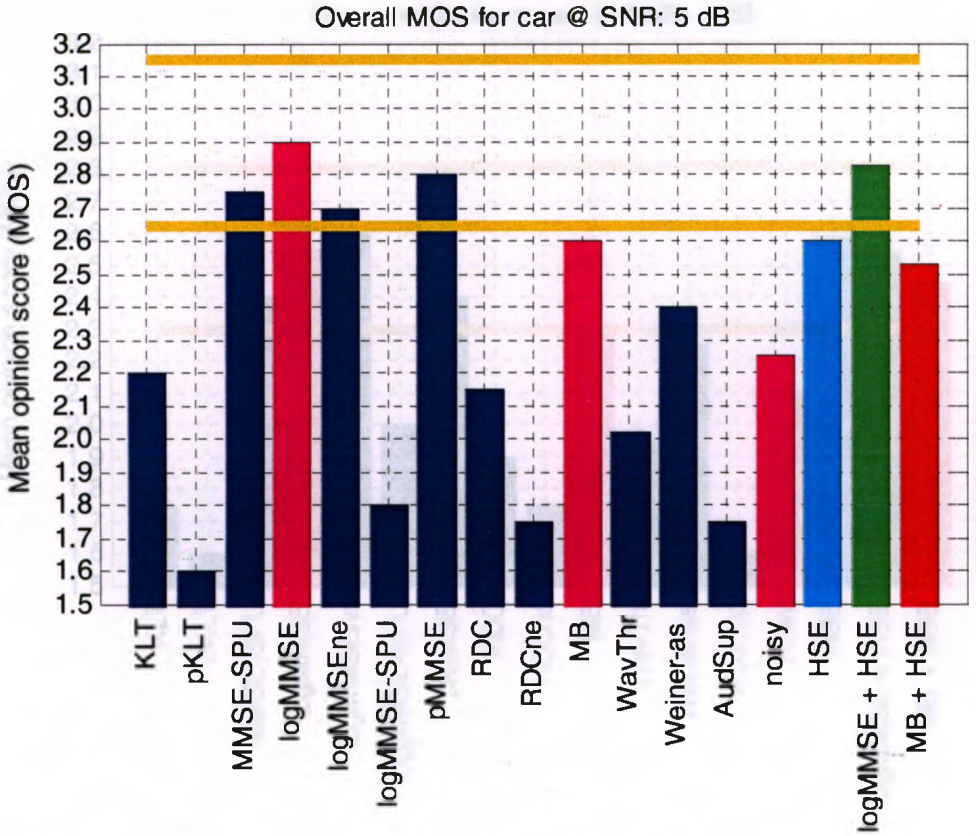

Figure 4.3: Predicted speech quality score for different algorithms: car noise at 5 dB SNR.

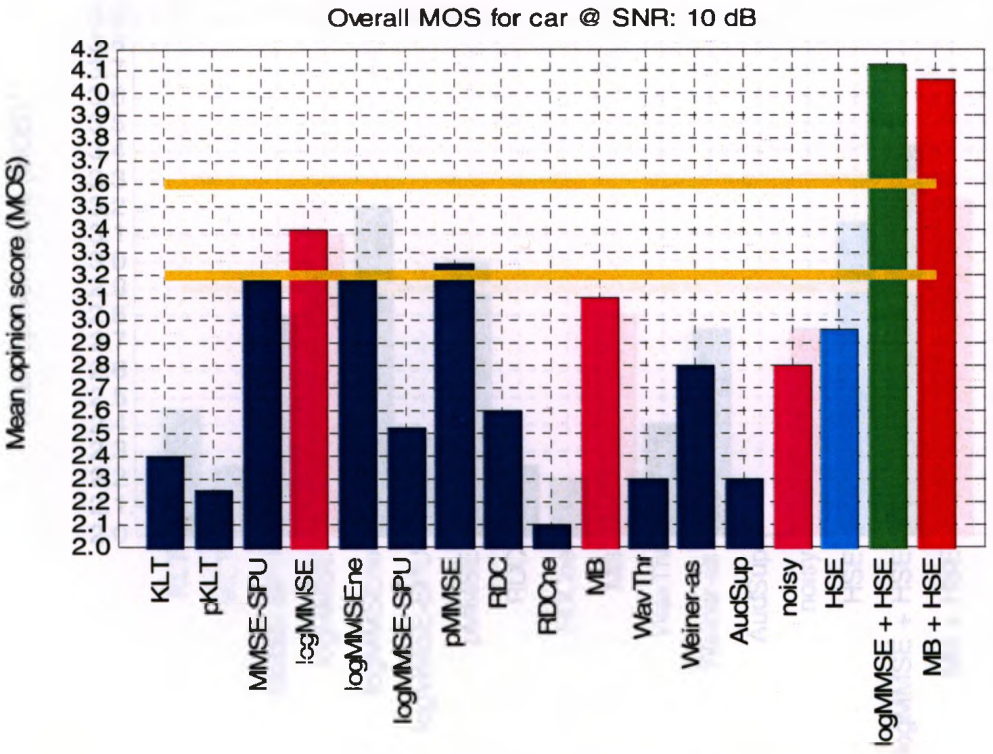

Figure 4.4: Predicted speech quality score for different algorithms: car noise at 10 dB SNR.

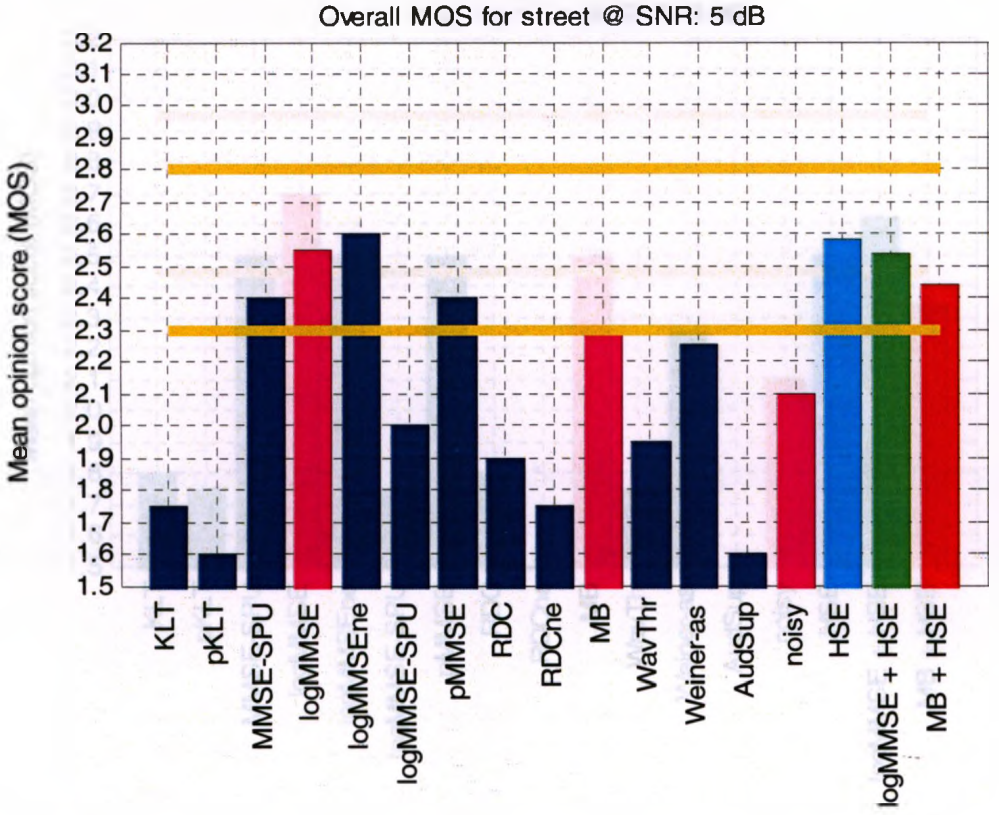

Figure 4.5: Predicted speech quality score for different algorithms: street noise at 5 dB SNR.

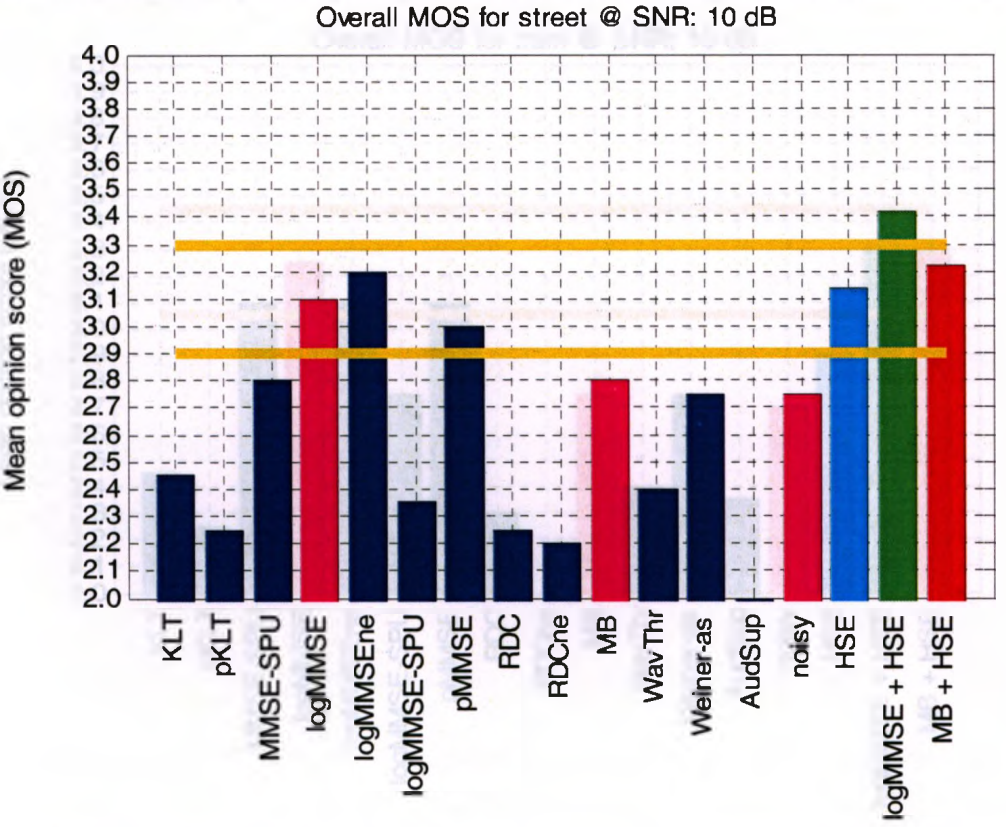

Figure 4.6: Predicted speech quality score for different algorithms: street noise at 10 dB SNR.

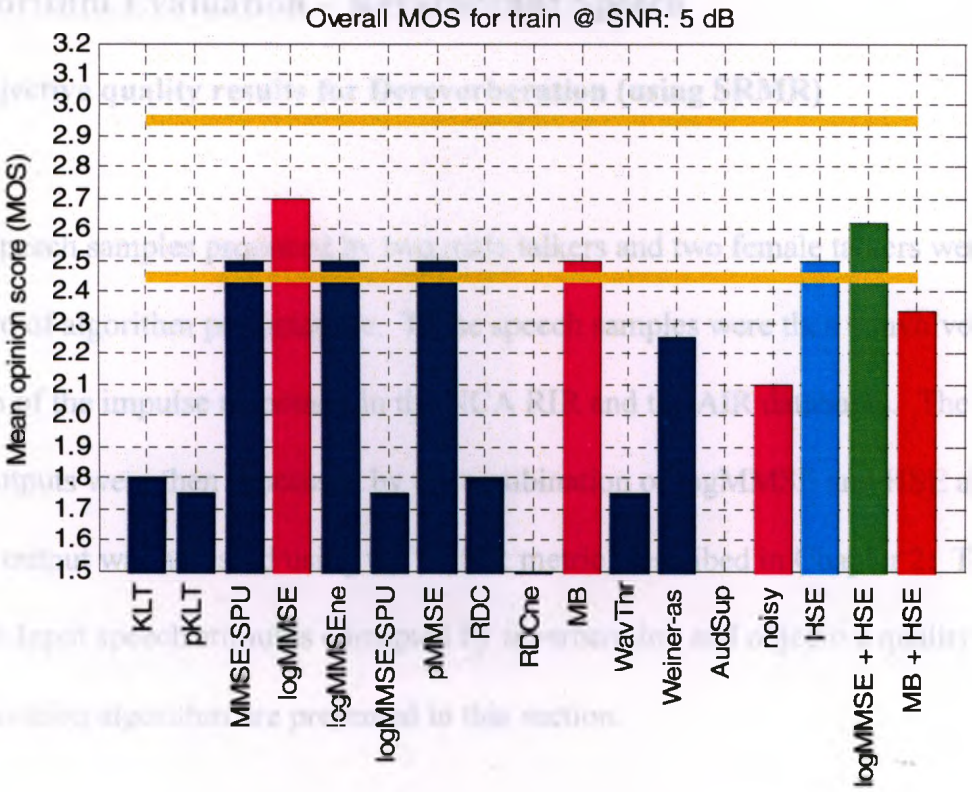

Figure 4.7: Predicted speech quality score for different algorithms: train noise at 5 dB SNR.

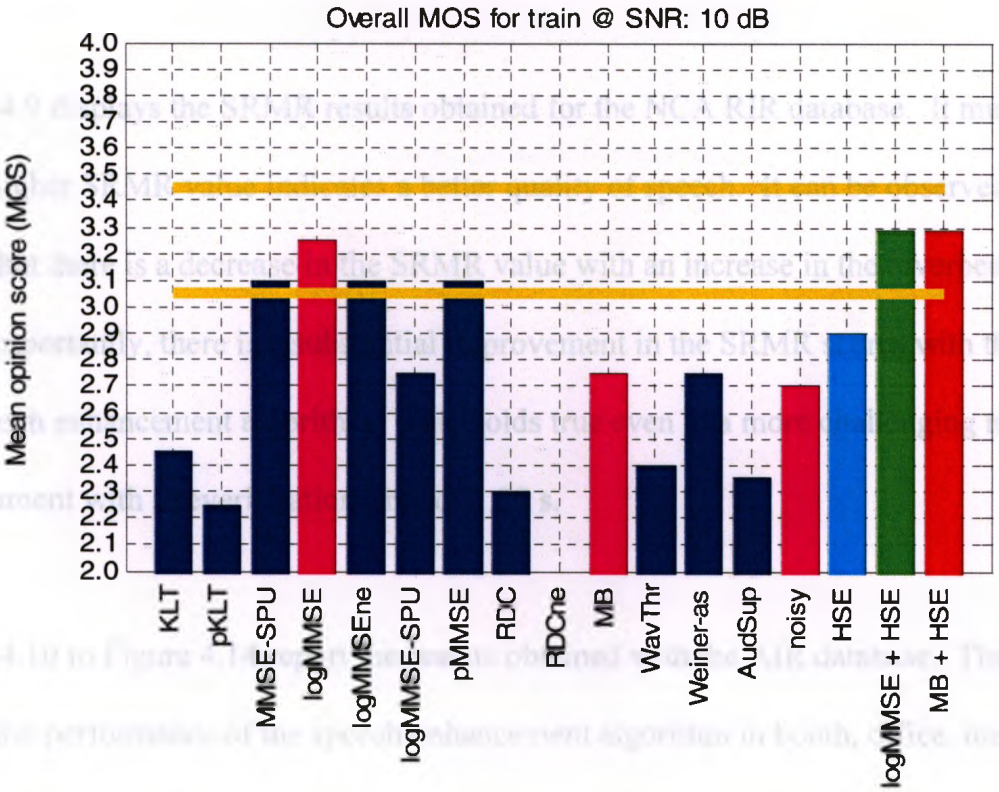

Figure 4.8: Predicted speech quality score for different algorithms: train noise at 10 dB SNR.

4.3 Algorithm Evaluation – Reverberant Speech

4.3.1 Objective quality results for Dereverberation (using SRMR)

Sixteen speech samples produced by two male talkers and two female talkers were used for the evaluation of algorithm performance. These speech samples were then convolved individually with each of the impulse responses in the NCA RIR and the AIR databases. The reverberant speech outputs were then processed by the combination of logMMSE and HSE algorithms. The resulting output was assessed using the SRMR metric, described in Chapter 2. The procedures to create the Input speech stimulus corrupted by reverberation and objective quality scores of the dereverberation algorithm are presented in this section.

4.3.2 Results

Figure 4.9 displays the SRMR results obtained for the NCA RIR database. It must be noted here that a higher SRMR value indicates a better quality of speech. It can be observed from this graph that there is a decrease in the SRMR value with an increase in the reverberation time. More importantly, there is a substantial improvement in the SRMR scores with the application of the speech enhancement algorithm. This holds true even in a more challenging reverberant environment with a reverberation time of 1.39 s.

Figure 4.10 to Figure 4.14 report the results obtained with the AIR database. These graphs depict the performance of the speech enhancement algorithm in booth, office, meeting room, lecture hall,

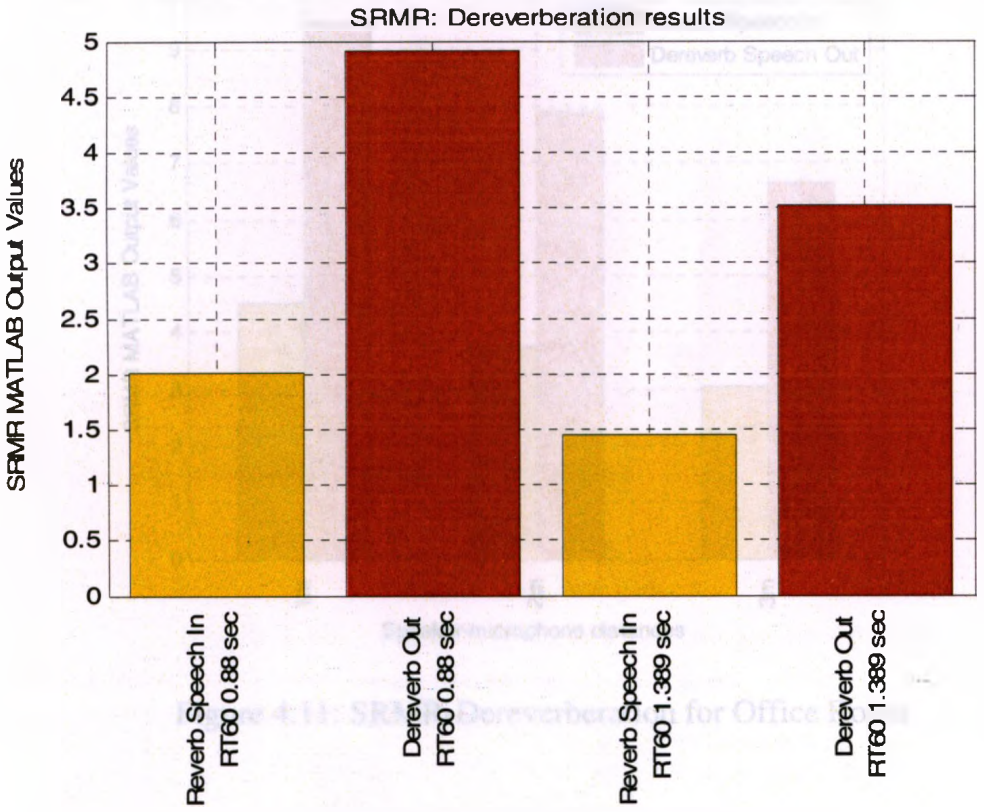

Figure 4.9: SRMR Dereverberation Results

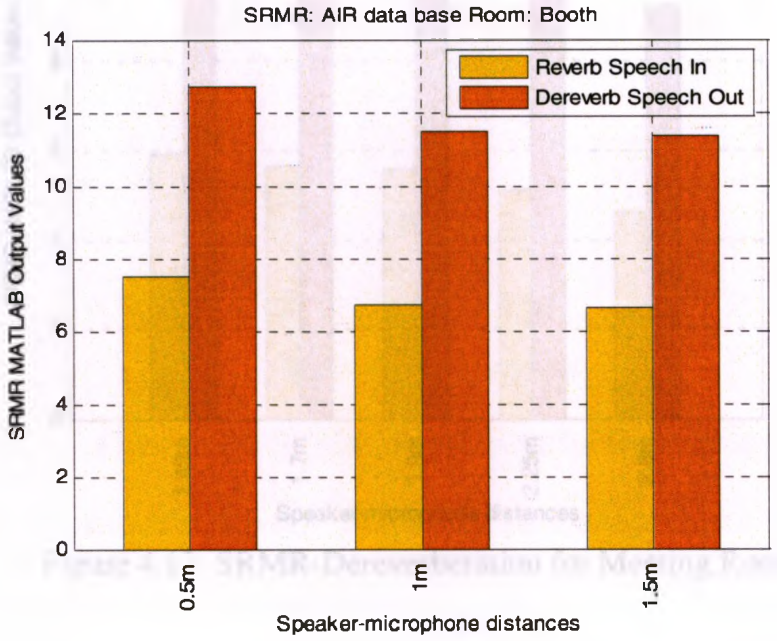

Figure 4.10: SRMR-Dereverberation for Booth Room

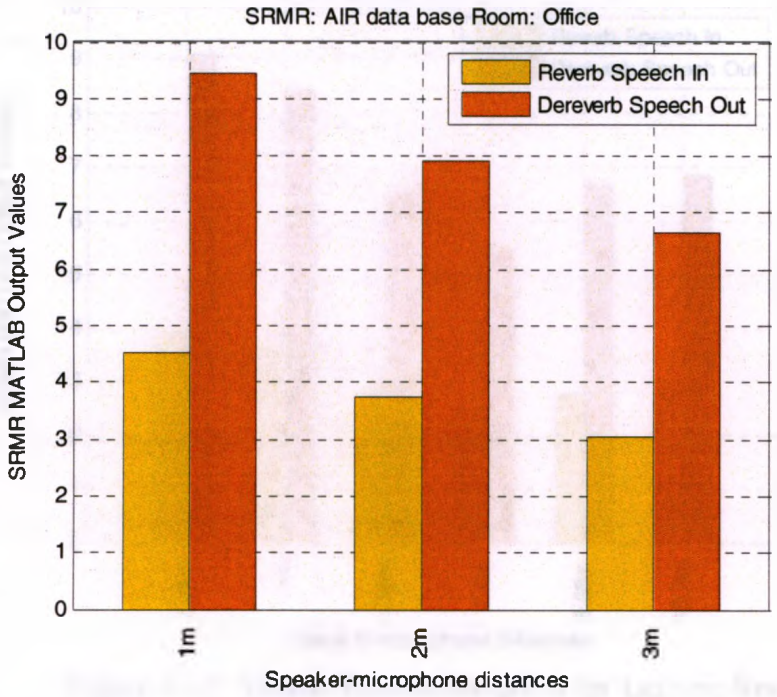

Figure 4.11: SRMR-Dereverberation for Office Room

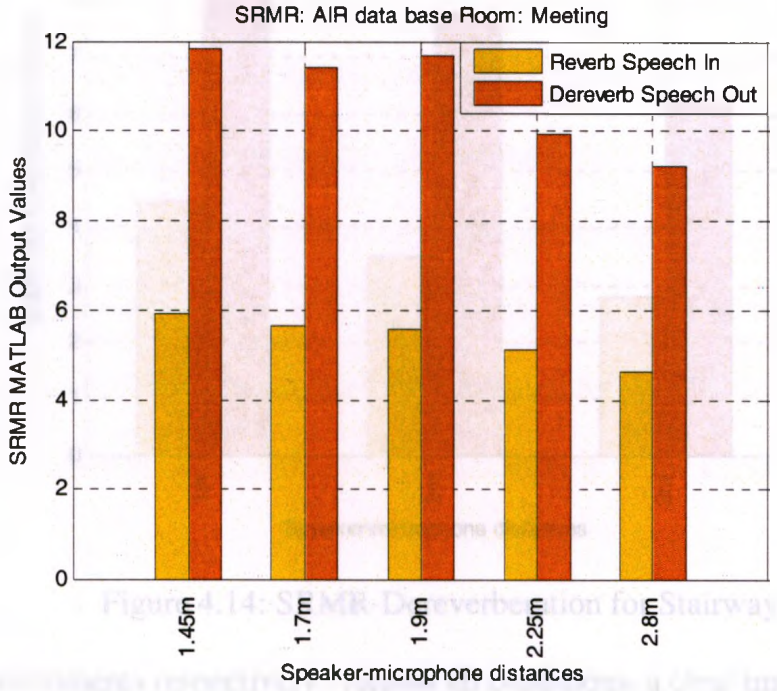

Figure 4.12: SRMR-Dereverberation for Meeting Room

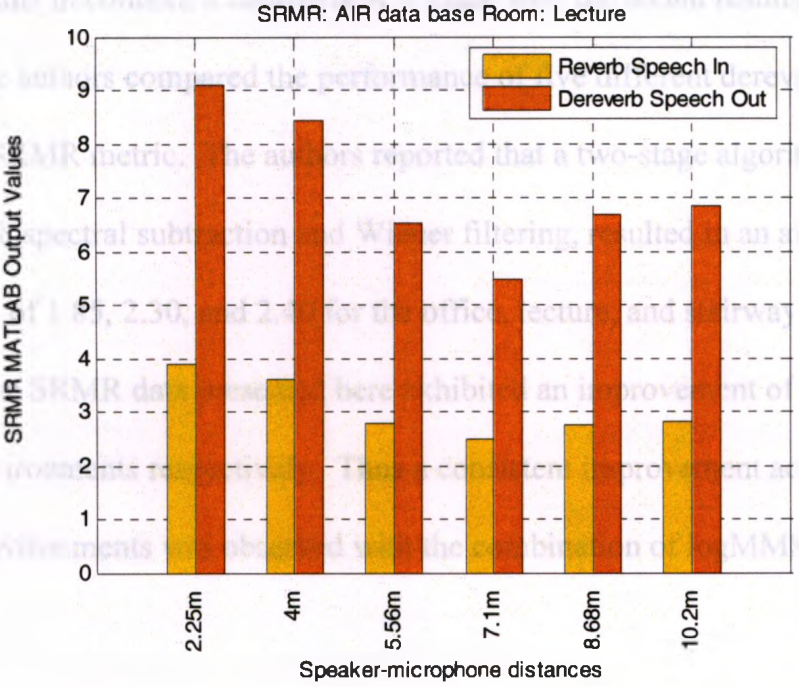

Figure 4.13: SRMR-Dereverberation for Lecture Room

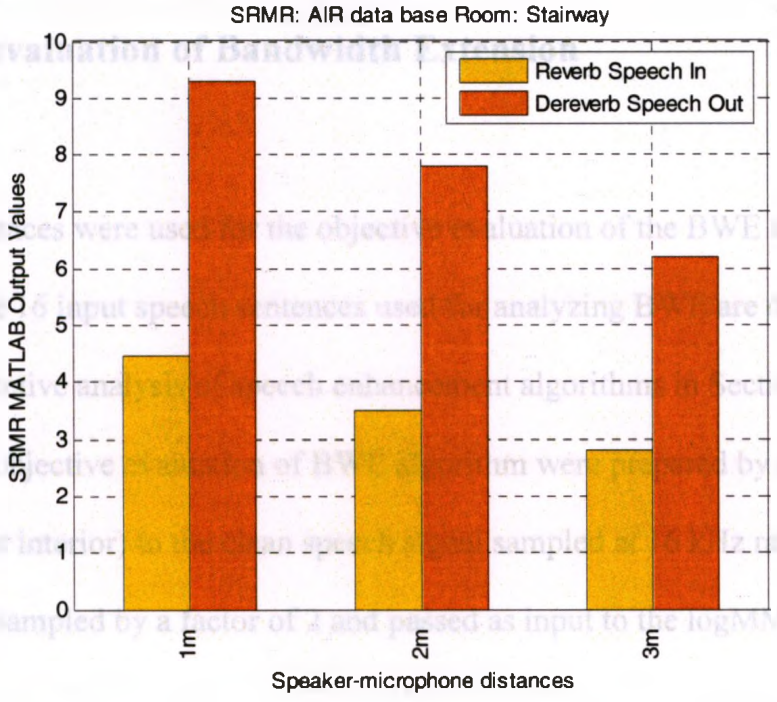

Figure 4.14: SRMR-Dereverberation for Stairway

and stairway environments respectively. Across all conditions, a clear improvement in the objective score is apparent in these Figures.

To put these results in context, a comparison is made with the recent results reported by Jeub et al. [7], where the authors compared the performance of five different dereverberation algorithms using the same SRMR metric. The authors reported that a two-stage algorithm, one that incorporates both spectral subtraction and Wiener filtering, resulted in an average improvement in SRMR scores of 1.85, 2.30, and 2.40 for the office, lecture, and stairway environments respectively. The SRMR data presented here exhibited an improvement of 4.23, 4.12, and 4.17 for the same environments respectively. Thus a consistent improvement across a range of reverberation environments was observed with the combination of logMMSE and HSE algorithms.

4.4 Objective evaluation of Bandwidth Extension

A total of 16 sentences were used for the objective evaluation of the BWE algorithm. It must be noted here that the 16 input speech sentences used for analyzing BWE are different from the ones used for objective analysis of speech enhancement algorithms in Section 4.2. The noisy samples used for objective evaluation of BWE algorithm were prepared by adding environmental noises (babble, car interior) to the clean speech signal sampled at 16 kHz rate. This corrupted speech was downsampled by a factor of 2 and passed as input to the logMMSE and HSE systems.

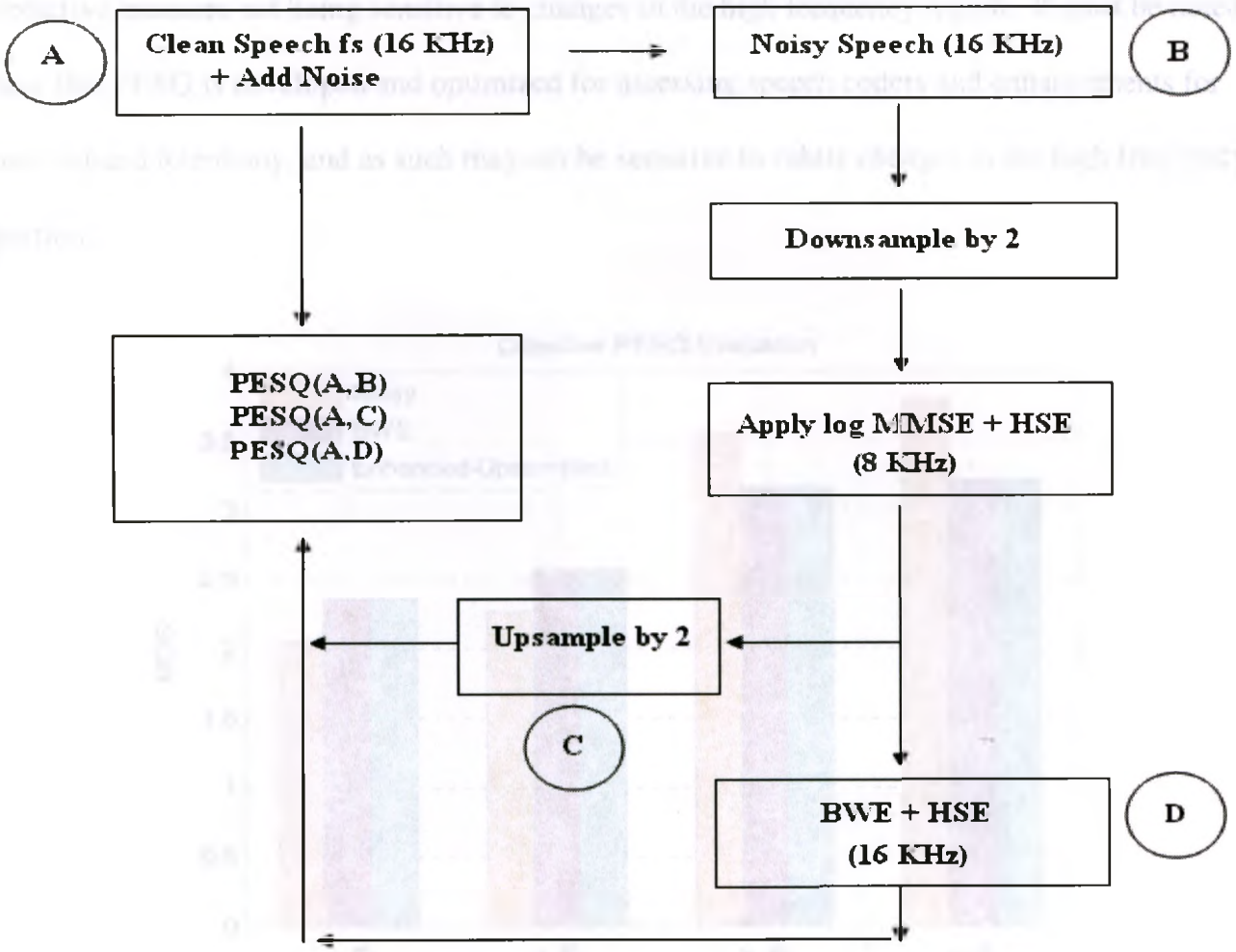

Figure 4.15: Objective evaluation of BWE algorithm

The sample rate of these systems (logMMSE-HSE) is 8 kHz. The output of these systems is upsampled by a factor of 2 by passing it into the BWE system. PESQ analysis is done by comparing clean speech to noisy speech and clean speech to BWE speech. Figure 4.16 and Figure 4.17 show the PESQ results for noise reduction plus BWE speech and SRMR results for dereverberation-BWE respectively. From Figure 4.16, it can be noticed that there is an improvement in the speech quality score for the enhanced speech samples in the multi-talker babble condition. However, there was no statistically significant difference between the BWE output and a simple upsampling of the narrowband enhanced speech. This is possibly due to the

objective measure not being sensitive to changes in the high frequency region. It must be noted here that PESQ is developed and optimized for assessing speech coders and enhancements for narrowband telephony, and as such may not be sensitive to subtle changes in the high frequency portion.

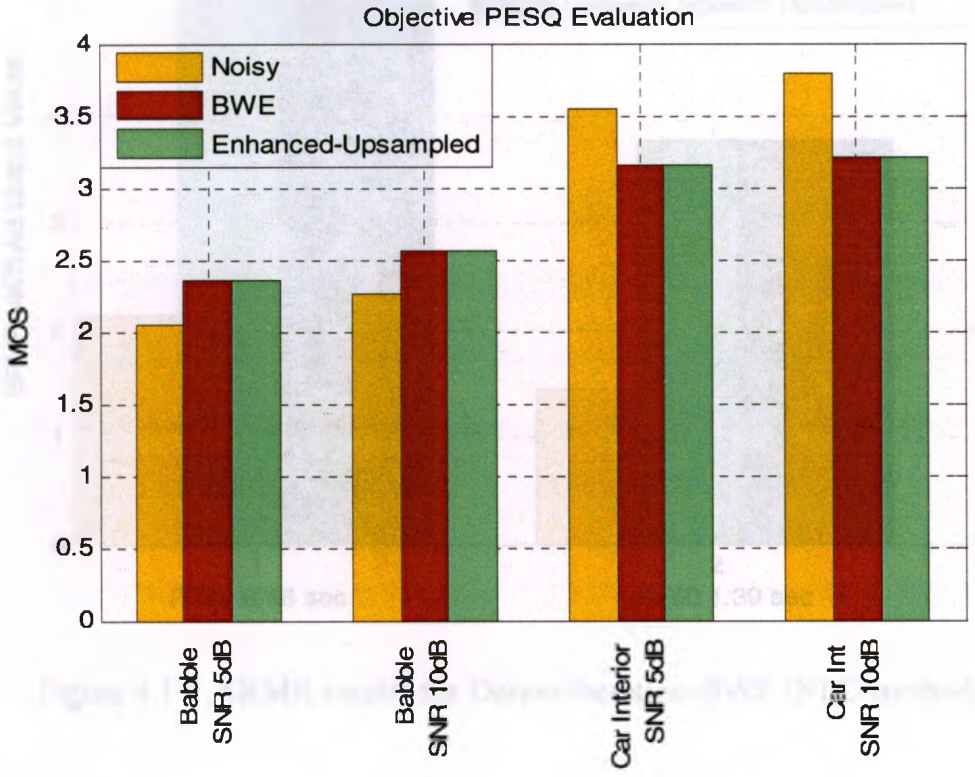

Figure 4.16: PESQ results for Noise reduction-BWE (NLD method)

A validated wideband objective metric is desirable in order to properly benchmark the BWE Output. The MOS scores for bandwidth extended speech in car interior noise was lower than noisy case because the car interior noise was colored and most of its energy was below 500 Hz. Figure 4.17 displays the results from applying bandwidth extension to dereverberated speech. A slight improvement in the SRMR scores is noted with BWE speech. A subjective evaluation

study is necessary in order to determine whether this small improvement is statistically significant.

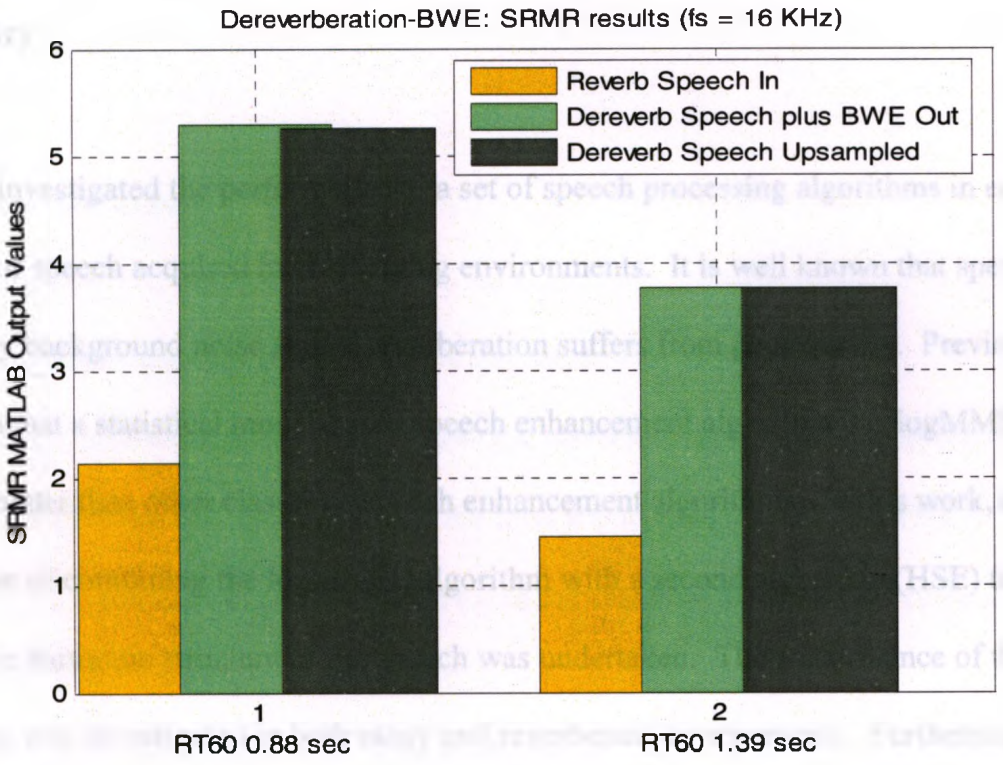

Figure 4.17: SRMR results for Dereverberation-BWE (NLD method)

Chapter 5: Conclusion

5.1 Summary

This thesis investigated the performance of a set of speech processing algorithms in enhancing the quality of speech acquired in challenging environments. It is well known that speech corrupted by background noise and/or reverberation suffers from poor quality. Previous studies have shown that a statistical model-based speech enhancement algorithm *viz.* logMMSE performed better than other classes of speech enhancement algorithms. In this work, an investigation of combining the logMMSE algorithm with a second algorithm (HSE) that enhances the harmonic structure of the speech was undertaken. The performance of this combination was investigated in both noisy and reverberant environments. Furthermore, the effect of extending the bandwidth of enhanced speech using a simple technique was also investigated.

For the speech-in-noise conditions, the logMMSE plus HSE resulted in speech quality improvements, more notably at 10 dB SNR than at 5 dB SNR, as evidently seen from the confidence interval plots shown in chapter 4. Furthermore, the combination was the best for the multi-talker babble condition. In poorer SNRs (5 dB) and predominantly lower frequency noise sources, the HSE algorithm was found not to perform as effectively. For the reverberation conditions, there was a significant improvement in the objective quality scores with the application of the logMMSE and HSE. This was consistent across multiple impulse response databases, and multiple reverberant environments.

In case of BWE using the HF harmonic reconstruction using the non-linear distortion method, there was significant improvement in MOS when the input to the Speech Enhancement system was corrupted by babble background noise.

5.2 Major Contributions

The following list details the contributions of this thesis:

- Realtime-capable versions of the Harmonic Speech Enhancement (HSE) and logMMSE algorithms were implemented using the Intel IPP signal processing library.
- logMMSE and HSE algorithms were cascaded and their synergistic performance was demonstrated for certain noise condition.
- The same combination was evaluated with reverberant speech samples and shown to perform better than recently reported dereverberation algorithms.
- A simple, realtime capable bandwidth extension algorithm was applied to enhanced speech in both noisy and reverberant environments.

5.3 Future Extensions

This system is very close to becoming a real-time speech processing system. The speech enhancement C-code written for this thesis project has an audio IO latency ranging from 1 to 3 milliseconds for 256 audio samples (32 millisecond analysis frame rate at 8 kHz) on a Red Hat Linux (Fedora) platform. Eventually, the code can be ported to a hand held device (Linux

environment) and optimized for real-time performance following [1] using tools like Jack audio connection kit.

The highly successful outcome with processing reverberant speech samples should be validated with subjective evaluation experiments. In particular, the performance of the logMMSE plus HSE should be evaluated with hearing impaired listeners. For speech at 10 dB SNR, this thesis has demonstrated significant improvement in speech quality. Since hearing aids amplify the processed speech, impaired listeners may perceive improvement in sound quality from the output of this algorithm more effectively than normal listeners. The background noise and reverberation effects were reduced by the speech enhancement algorithms and hence do not get amplified much. Furthermore, the speech harmonics were either preserved or emphasized by the HSE algorithm.

In most of the real world cases, speech is corrupted by both noise and reverberation simultaneously. Therefore, both objective and subjective evaluations need to be carried out with both noise and reverberation. In a similar vein, the BWE stimuli should be evaluated by both normal hearing and hearing impaired listeners. BWE of speech in hearing aids is a fairly new area of research.

References

[1] Grimm, G., Herzke., T, Hohmann, V., "Application of Linux Audio in Hearing aid research", LAC, 2009

[2] Wen, J., Xin, L., Scordilis, M.S, Lu, H., "Speech Enhancement Using Harmonic Emphasis and Adaptive Comb Filtering," IEEE Trans. Audio, Speech, Lang. Process, vol.18, no.2, pp.356-368, Feb. 2010

[3] Veprek, P., Scordilis, M. S., "Analysis, enhancement and evaluation of five pitch determination techniques," Speech Commun., vol. 37, pp. 249–270, Jul. 2002.

[4] Stylianou, Y., "Applying the harmonic plus noise model in concatenative speech synthesis", IEEE Trans. Speech Audio Process., vol. 9, no. 1, pp. 21–29, Jan. 2001.

[5] McAulay, R., Quatieri, T., "Speech analysis/Synthesis based on a sinusoidal representation," Acoustics, Speech and Signal Processing, IEEE Transactions on , vol.34, no.4, pp. 744- 754, Aug 1986

[6] Dillon, H., "Hearing Aids", Boomerang Press, 2000

[7] Jeub, M., Schäfer, M., Esch, T., Vary, P.,"Model-Based Dereverberation Preserving Binaural Cues", IEEE Trans. Audio, Speech, Lang. Process, vol 18, no.7, 2010

[8] Stelmachowicz, P.G., Pittman, A.L., Hoover, B.M. & Lewis, D.E., "Effect of stimulus bandwidth on the perception of /s/ in normal and hearing-impaired children and adults. J Acoust Soc Am, 110(4), 2183-2190, 2001

[9] Larsen, E., Aarts, R.M., "Audio Bandwidth Extension", John Wiley & Sons, Ltd, 2004

[10] http://www.utdallas.edu/~thib/rehabinfo/tohl.htm

[11] Laaksonen, A., "Bandwidth extension in high-quality audio coding", Master's Thesis, Helsinki University of technology, 2005

[12] Hu, Y., Loizou, P.C., "Subjective comparison and evaluation of speech enhancement algorithms", presented at Speech Communication, pp.588-601, 2007

[13] Hu, Y., Loizou, P.C., "Evaluation of Objective Quality Measures for Speech Enhancement", IEEE Trans. Audio, Speech, Lang. Process, vol.16, no.1, pp.229-238, Jan. 2008

[14] Loizou, P.C, "Speech Enhancement: Theory and Practice", CRC Press, Boca Raton, FL, 2007

[15] Janert, P. K., "Gnuplot in action: understanding data with graphs", Manning Publications, 2009

[16] Johnson, E., Ricketts, T., Hornsby, B., "The effect of extending high-frequency bandwidth on the Acceptable Noise Level (ANL) of hearing-impaired listeners", Int J Aud.; 48:353-362, 2009

[17] Sunohara,M., Terada, K., Okuno,T., Iwakura, T., "Speech Enhancement Algorithm for Digital Hearing Aids on the basis of Auditory Scene Analysis", RION CO., LTD, 2010

[18] Thomas, I. B., Niederjohn, R. J., "The intelligibility of filtered-clipped speech in noise", J. Audio Eng. Soc., vol. 18, pp.299 - 303, 1970

[19] Hardwick, J., Yoo, C. D., Lim, J. S., "Speech enhancement using the dual excitation model," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 367–370, 1993

[20] Dubost, S., Cappe, O., "Enhancement of speech based on non-parametric estimation of a time varying harmonic representation," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 1859–1862, 2000

[21] Deisher, M. E., Spanias, A. S., "HMM-based speech enhancement using harmonic modeling", in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 1175–1178, 1997
[22] Deisher, M. E., Spanias, A. S., "Speech enhancement using state-based estimation and sinusoidal modeling," J. Acoust. Soc. Amer., vol. 102, no. 2, pp. 1141–1148, Aug. 1997.
[23] Jensen, J., Hansen, J. H. L., "Speech enhancement using a constrained iterative sinusoidal model," IEEE Trans. Speech Audio Process., vol. 9, no. 7, pp. 731–740, Oct. 2001
[24] Anderson, D. V., Clements, M. A., "Audio signal noise reduction using harmonic modeling," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 805–808, 1999
[25] Morgan, D., George, B., Lee, L., Kay, S. M., "Co-channel speaker separation by harmonic enhancement and suppression," IEEE Trans. Speech Audio Process., vol. 5, no. 5, pp. 407–424, Sep. 1997.
[26] Quatieri, T. F., Danisewicz, R. G., "Co-channel speaker separation by harmonic enhancement and suppression," IEEE Trans. Acoust., Speech, Signal Process., vol. 38, no. 1, pp. 56–69, Jan. 1990
[27] Erell, A., Weintraub, M., "Estimation of noise-corrupted speech DFT spectrum using the pitch period", IEEE Trans. Speech Audio Process., vol. 2, pp. 1–8, Jan. 1994.
[28] Yu, A.T., Wang H.C., "New speech harmonic structure measure and it application to post speech enhancement," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 729–732, 2004
[29] Li, C., Anderson, S. V., "Inter-frequency dependency in MMSE speech enhancement," in Proc. 6th Nordic Signal Process. Symp., pp. 200–203, 2004
[30] Plapous, C., Marro, C., Scalart P., "Speech enhancement using harmonic regeneration," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 157–160, 2005
[31] Boll, S. F., "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.
[32] Grancharov, V., Plasberg, J. H., Samuelsson, J. , Kleijn, W. B., "Generalized post-filter for speech quality enhancement," IEEE Trans. Audio, Speech, Lang. Process., vol. 16, no. 1, pp. 57–64, Jan. 2008.
[33] Chen, J. H., Gersho, A., "Adaptive postfiltering for quality enhancement of coded speech," IEEE Trans. Speech Audio Process., vol. 3, no. 1, pp. 59–71, Jan. 1995.
[34] Ephraim, Y., Malah, D., "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 1109-1121, Dec. 1984.
[35] Ephraim, Y., Malah, D., "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," IEEE Trans. Acoust., Speech, Signal Process., vol. 33, no. 2, pp. 443–445, Apr. 1985.
[36] S. F. Boll, "Noise in Speech Using Spectral Subtraction", Trans. Acoust. Speech Signal Process. vASSP-27 i2. 113-120, 1979
[37] M. Dietz, L. Liljeryd, K. Kjörling, O. Kunz, "Spectral band replication, a novel approach in audio coding", Proc. 112th AES convention, Munich, Germany, May 2002
[38] I. B. Thomas and R. J. Niederjohn, "Enhancement of Speech Intelligibility at High Noise Levels by Filtering and Clipping," J. Audio Eng. Soc. 16, 412, 1968
[39] O. Cappé, 1994. "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor", IEEE Trans. Speech, and Audio Processing, vol. 2, no. 2, 345-349

[40] G.Chen, "Statistical Model-Based Objective Measures of Speech Quality", Doctoral thesis, The University of Western Ontario, 2007
[41] Krishnamoorthy, P., Prasanna, S.R.M., "Reverberant speech enhancement by temporal and spectral processing", IEEE Trans. Audio, Speech, Lang. Process, vol.17, no.2, pp. 253-266, Feb. 2009
[42] Falk, T. H., Chan, W.Y., "A non-intrusive quality measure of dereverberated speech," in Proc. Int.Workshop Acoust. Echo Noise Control, 1766 – 1774, Sep. 2008
[43] M. Jeub , M. Schfer and P. Vary "A binaural room impulse response database for the evaluation of dereverberation algorithms", Proc. 16th Int. Conf. Digital Signal Process. (DSP), pp 1-4, 2009.
[44] https://ccrma.stanford.edu/realsimple/imp_meas/
[45] Naylor, P.A., Gaubitch, N.D, "Speech Dereverberation", Springer-Verlag London Limited, 2010
[46] Oppenheim, A.V., Schafer, R.W., "Digital signal processing", 1 edn. Prentice Hall, 1975
[47] Oppenheim, A.V., Schafer, R.W., Stockham, T.G., "Nonlinear filtering of multiplied and convolved signals", IEEE Trans. Audio Electroacoust. AU-16(3), 437–466, 1968
[48] Brandstein, M.S., Griebel, S.M., "Nonlinear, model-based microphone array speech enhancement", In: S.L. Gay, J. Benesty (eds.) Acoustic Signal Processing for Telecommunication, pp.261–279. Kluwer Academic Publishers, 2000
[49] Yegnanarayana, B., Satyanarayana, P., "Enhancement of reverberant speech using LP residual signal", IEEE Trans. Acoust., Speech, Signal Process. 8(3), 267–281, 2000
[50] Gaubitch, N.D., Ward, D.B., Naylor, P.A., "Statistical analysis of the autoregressive modeling of reverberant speech", J. Acoust. Soc. Am. 120(6), 4031–4039, 2006
[51] Allen, J.B., "Synthesis of pure speech from a reverberant signal", U.S. Patent No. 3786188, 1974
[52] Griebel, S.M., "A microphone array system for speech source localization, denoising and dereverberation", Ph.D. thesis, Harvard University, Cambridge, Massachusetts, 2002
[53] Griebel, S.M., Brandstein, M.S., "Wavelet transform extrema clustering for multi-channel speech dereverberation", In: Proc. Int. Workshop Acoust. Echo Noise Control (IWAENC). Pocono Manor, Pennsylvania, 1999
[54] Griebel, S.M., Brandstein, M.S, "Microphone array speech dereverberation using coarse channel estimation", In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. 201–204, 2001
[55] Yegnanarayana, B., Prasanna, S.R.M., Rao, K.S., "Speech enhancement using excitation source information", In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. 541–544, 2002
[56] Gillespie, B.W., Malvar, H.S., Florencio, D.A.F., "Speech dereverberation via maximum kurtosis subband adaptive filtering", In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 6, pp. 3701–3704, 2001
[57] Wu, M., Wang, D., "A two-stage algorithm for one-microphone reverberant speech enhancement", IEEE Trans. Audio, Speech, Lang. Process. 14(3), 774–784, 2006
[58] Nakatani, T.,Miyoshi,M., Kinoshita, K., "Single-microphone blind dereverberation", In: J. Benesty, S. Makino, J. Chen (eds.) Speech Enhancement, 1 edn. Springer Verlag, 2005
[59] Benesty, J., Makino, S., Chen, J. (eds.), "Speech enhancement", Springer, 2005
[60] Davis, G.M., "Noise reduction in speech applications", CRC Press, 2002

[61] Habets, E. A. P., "Multi-channel speech dereverberation based on a statistical model of late reverberation", Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., p.173, 2005
[62] Habets, E.A.P., "Single- and multi-microphone speech dereverberation using spectral enhancement", Ph.D. Thesis, Technische Universiteit Eindhoven, 2007
[63] Kinoshita, K., Nakatani, T., and Miyoshi, M., "Fast estimation of a precise dereverberation filter based on the harmonic structure of speech," Acoustical Science and Technology, vol.28, no.2, pp.105-114, 2007
[64] Aarts, R.M, Larsen, E., Schobben, D., ""Improving perceived bass and reconstruction of high frequencies for band limited signals", in Proc IEEE Benelux Workshop on Model based Processing and Coding of Audio MPCA2002, 2002
[65] http://www.ind.rwth-aachen.de/en/research/speech-and-audio-processing/aachen-impulse-response-database/
[66] http://www.reverberationtime.com/
[67] Ephraim, Y., Van Trees H. L., "A signal subspace approach for speech enhancement", IEEE Trans. Speech Audio Process., vol. 3, pp.251–266, Jul. 1995
[68] Soon, I.Y., Koh, S.N., "Low Distortion Speech Enhancement", IEE Proc., Vis. Image Signal Process, Vol. 147, No. 3, pp. 247-253, June 2000
[69] Moore, B.C.J. & Tan, C.T., "Perceived naturalness of spectrally distorted speech and music", J Acoust Soc Am, 114, 408 419, 2003

Appendix A

A.1 Real-time Implementation:

The algorithm implementation was done in floating point C using the Intel IPP audio DSP library with an intention to port it into a hand held device with Linux operating system that a hearing impaired subject could carry to various reverberation and noisy environments to validate the algorithms in real-time. Therefore, this appendix provides some information on the operating system, IPP DSP library and IDE's used in developing these test algorithms.

A.1.1 Intel IPP Audio DSP library:

Intel® Integrated Performance Primitives (Intel® IPP) is an extensive library of multicore-ready, highly optimized software functions for multimedia, data processing, and communications applications. Intel IPP offers thousands of optimized functions covering frequently used fundamental algorithms. It is a multi-threaded library of functions for multimedia and data processing applications, produced by Intel. The library supports Intel and compatible processors and is available for Windows, Linux, and Mac OS X operating systems. It is available separately or as a part of Intel Parallel Studio.

This document describes the structure, operation and functions of the Intel® Integrated Performance Primitives (Intel® IPP) for Intel® architecture that operate on one-dimensional signals. This is the first volume of the Intel IPP Reference Manual, which also comprises

descriptions of Intel IPP for image and video processing (volume 2), operations on small matrices, 3D data processing and rendering (volume 3), and cryptography functions (volume 4). The Intel IPP software package supports many functions whose performance can be significantly enhanced on Intel architecture, particularly using the MMX™ technology, Streaming SIMD Extensions (SSE), Streaming SIMD Extensions 2 (SSE2), Streaming SIMD Extensions 3 (SSE3), as well as Intel® Itanium® architecture.

The Intel IPP for signal processing software is a collection of low-overhead, high-performance operations performed on one-dimensional (1D) data arrays.

The Intel IPP for Intel architecture software enables taking advantage of the parallelism of the single-instruction, multiple-data (SIMD) instructions that make up the core of the MMX technology and Streaming SIMD Extensions. These technologies improve the performance of computation-intensive signal, image, and video processing applications. Use of Intel IPP primitive functions can help to drastically reduce development costs and accelerate time-to-market by eliminating the need of writing processor-specific code for computation intensive routines.

A.1.2 Hardware and Software Requirements:

The Intel IPP for Intel architecture software runs on personal computers that are based on processors using IA-32, Intel® 64 or IA-64 architecture and running Microsoft Windows* OS, Linux* OS, or Apple Mac OS* X. Intel IPP can be integrated into the customer's application or library written in C or C++.

A.1.3 Platforms Supported:

Intel IPP for Intel architecture software runs on Windows* OS, Linux* OS, and Mac OS* X platforms. The code and syntax used in this manual for function and variable declarations are written in the ANSI C style. However, versions of Intel IPP for different processors or operating systems may, of necessity, vary slightly.

A.1.4 Intel IPP usage in C software:

Type	Usual C type	Intel IPP type
8u	Unsigned char	Ipp8u
8s	Signed char	Ipp 8s
16u	Unsigned short	Ipp16u
16s	Signed short	Ipp16s
16sc	Complex short	Ipp16sc
32u	Unsigned int	Ipp32u
32s	Signed int	Ipp32s
32f	Float	Ipp32f
32fc	Complex float	Ipp32fc
64s	__int64 (Windows*) or long long (linux)	Ipp64s
64f	Double	Ipp64f
64fc	Complex double	Ipp64fc

Table 4.2: Data Types Supported by Intel IPP for Signal Processing

covers all the Intel IPP data types. In this software implementation, Ipp32f an Intel floating point data type was used prevalently. Here is an implementation example for linear prediction and FFT method

```
// Initialization

IppStatus st;
int lenSrc = 256;
int lpOrder = 12;
```

```
// Memory Allocation

Ipp32f* lpCoeff = ippsMalloc_32f(lpOrder);  // pointer to LP-Coeff vector
Ipp32f* pPreEmpOut = ippsMalloc_32f(lenSrc);// pointer to pre-emphasis output


// Calculate Linear prediction coefficients: Function call

st = ippsLinearPrediction_Auto_32f(pPreEmpOut, lenSrc, lpCoeff, lpOrder);


// FFT Function definition

IppStatus myFFT(Ipp32f* pSrc, Ipp32f* pSrcSpec, struct paramSet* pz)
{

  IppStatus st;

  // FFT settings:
  IppsFFTSpec_R_32f *ppFFTSpec;
  int flagFFT = IPP_FFT_DIV_FWD_BY_N;
  IppHintAlgorithm hint = ippAlgHintAccurate;
  int fftOrder = (pz)->fftOrder;
  int fftSize = (pz)->fftSize;


  // Memory allocation
  Ipp32f* pSrcFFT = ippsMalloc_32f(fftSize);


  // pointer to complex FDNFP Vector
  Ipp32fc* pSrcCplx = ippsMalloc_32fc(fftSize);


  // FFT memory allocation
  st=ippsFFTInitAlloc_R_32f(&ppFFTSpec, fftOrder, flagFFT, hint);

  // Fast-Fourier transform
  ippsFFTFwd_RToCCS_32f(pSrc, pSrcFFT, ppFFTSpec, NULL);


  // Converts the data in CCS format to complex data format
  ippsConjCcs_32fc(pSrcFFT, pSrcCplx, fftSize);


  // Calculate Magnitude spectrum
  ippsAbs_32fc_A24(pSrcCplx, pSrcSpec, fftSize);

  // Closes FFT specification structure
  ippsFFTFree_R_32f(ppFFTSpec);
  return st;

}
```

A.2 Linux (Fedora):

Fedora is a Linux-based operating system, a collection of software that makes your computer run. You can use Fedora in addition to, or instead of, other operating systems such as Microsoft Windows™ or Mac OS X™. The Fedora operating system is completely free of cost for you to enjoy and share.

The Fedora Project is the name of a worldwide community of people who love, use, and build free software from around the globe. We want to lead in the creation and spread of free code and content by working together as a community. Fedora is sponsored by Red Hat, the world's most trusted provider of open source technology. Red Hat invests in Fedora to encourage collaboration and incubate innovative new free software technologies.

A.3 Eclipse CDT IDE:

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ruby (including Ruby on Rails framework), Scala, and Scheme. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java, and Eclipse PDT for PHP.

The CDT Project provides a fully functional C and C++ Integrated Development Environment based on the Eclipse platform. Features include: support for project creation and managed build for various tool-chains, standard make build, source navigation, various source knowledge tools, such as type hierarchy, call graph, include browser, macro definition browser, code editor with syntax highlighting, folding and hyperlink navigation, source code refactoring and code generation, visual debugging tools, including memory, registers, and disassembly viewers.

A.4 MATLAB, Gnuplot & Audacity:

MATLAB (for matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, and Fortran.

In our project, MATLAB was used in simulating the Harmonic speech enhancement model and Ephraim-Malah noise reduction prior to porting it into a C program. Besides this, MATLAB was used extensively for data plotting and analysis.

Gnuplot is a portable command-line driven graphing utility for linux, OS/2, MS Windows, OSX, VMS, and many other platforms. The source code is copyrighted but freely distributed (i.e., you don't have to pay for it). It was originally created to allow scientists and students to visualize mathematical functions and data interactively, but has grown to support many non-interactive

uses such as web scripting. It is also used as a plotting engine by third-party applications like Octave. Gnuplot has been supported and under active development since 1986.

Gnuplot supports many types of plots in either 2D or 3D. It can draw using lines, points, boxes, contours, vector fields, surfaces, and various associated text. It also supports various specialized plot types. Gnuplot supports many different types of output: interactive screen terminals (with mouse and hotkey input), direct output to pen plotters or modern printers, and output to many file formats (eps, fig, jpeg, LaTeX, metafont, pbm, pdf, png, postscript, svg, ...). Gnuplot is easily extensible to include new output modes. Recent additions include an interactive terminal based on wxWidgets and the creation of mousable graphs for web display using the HTML5 canvas element.

Gnuplots are used in C code by using pipes. Here is a sample C code used in the HSE-BWE algorithm:

```
// Given pointers to X and Y data, the function plots and returns control to main
function or subroutines

void plot2D(Ipp32f* xData, Ipp32f* y1Data, Ipp32f* y2Data, Ipp32f* y3Data, int
dataSize, char* xlabel, char* ylabel, char* title)
{
	FILE* pipe, * tempDataFile;
	char* tempDataFileName;

	Ipp32f x,y1,y2,y3;
	int i;
	int length = 200;
	tempDataFileName = "tempData.dat";

	char* xlbl = (char*) malloc (sizeof(char)*length);
	char* ylbl = (char*) malloc (sizeof(char)*length);
	char* ttl = (char*) malloc (sizeof(char)*length);

	strcpy(xlbl,"set xlabel '");
	strcat(xlbl, xlabel);
	strcat(xlbl,"'\n");

	strcpy(ylbl,"set ylabel '");
	strcat(ylbl, ylabel);
	strcat(ylbl,"'\n");
```

```
        strcpy(ttl,"set title '");
        strcat(ttl, title);
        strcat(ttl,"'\n");

        pipe = popen("gnuplot -persist","w");
        fprintf(pipe, ttl);
        fprintf(pipe, xlbl);
        fprintf(pipe, ylbl);
        fprintf(pipe, "set grid\n");
        fprintf(pipe, "set autoscale\n");

        if (pipe)
        {
                fprintf(pipe,"set key top right\n");
                fprintf(pipe,"set key box\n");
                fprintf(pipe,"plot \"%s\" using 1:2 title 'Src' with lines, "
                             " \"%s\" using 1:3 title 'Est' with lines, "
                             " \"%s\" using 1:4 title 'Err' with lines\n",
                             tempDataFileName,tempDataFileName,tempDataFileName);
                fflush(pipe);

                tempDataFile = fopen(tempDataFileName,"w");
                for (i=0; i <= dataSize; i++)
                {
                        x = xData[i];
                        y1 = y1Data[i];
                        y2 = y2Data[i];
                        y3 = y3Data[i];
                        fprintf(tempDataFile,"%1.10f %1.10f %1.10f %1.10f\n",x,y1,y2,y3);
                }
                fclose(tempDataFile);
                printf("press enter to continue...");
                getchar();
                remove(tempDataFileName);
                fprintf(pipe,"exit \n");
        }
        else
        {
                printf("gnuplot not found...");
        }
}
```

Audacity® is free, open source software for recording and editing sounds. It is available for Mac OS X, Microsoft Windows, GNU/Linux, and other operating systems. Audacity can be downloaded for free from: http://audacity.sourceforge.net/

Appendix B

Appendix B provides all the data tables that were used in creating the plots presented in chapter 4. The nomenclature of some of the sound files used in the objective evaluation is also provided.

B.1 Result Tables:

Wave file name	HSE			logMMSE + HSE			MB + HSE		
_babble_sn5	PESQ	IS	Y_TOTAL	PESQ	IS	Y_TOTAL	PESQ	IS	Y_TOTAL
sp01	2.21	1.84	2.66	2.14	1.86	2.53	2.29	9.31	2.5
sp02	2.35	2.4	2.89	2.27	3.62	2.76	2.35	23.7	2.48
sp03	2.2	2.76	2.64	2.22	3.7	2.66	2.24	29.2	2.28
sp04	2.39	2.21	2.96	2.29	2.27	2.79	2.29	12.3	2.39
sp06	2.44	2.95	3.03	2.38	2.37	2.95	2.48	7.85	2.78
sp07	2.12	2.61	2.49	2.09	2.99	2.45	2.27	19.1	2.34
sp08	2.32	3.28	2.85	2.23	2.38	2.69	2.24	14.6	2.29
sp09	2	2.26	2.29	2.08	2.11	2.43	2.18	6.73	2.44
sp11	2.04	1.98	2.36	2.1	1.57	2.47	2.19	9.09	2.35
sp12	1.99	2.25	2.27	2.16	3.41	2.57	2.13	27.2	2.07
sp13	2.17	2.77	2.58	2.29	4.75	2.74	2.18	18.8	2.18
sp14	2.14	5.12	2.45	2.07	3.47	2.4	2.28	35.2	2.32
sp16	2.06	2.52	2.39	2.05	2.36	2.37	2.17	3.13	2.59
sp17	2.2	1.6	2.63	2.39	1.76	2.96	2.32	11.1	2.48
sp18	2.01	3.9	2.29	2.12	7.09	2.32	2.11	28.9	2.04
sp19	2.3	5.44	2.71	2.37	7.46	2.74	2.38	42.3	2.49
_babble_sn10	PESQ	IS	Y_TOTAL	PESQ	IS	Y_TOTAL	PESQ	IS	Y_TOTAL
sp01	2.37	2.44	2.93	2.54	3.62	2.94	2.5	37.6	2.53
sp02	2.61	1.93	3.36	2.6	1.67	3.26	2.74	10.4	4.49
sp03	2.47	2.04	3	2.71	4.47	4.41	2.84	33.1	5.46
sp04	2.49	1.9	2.99	2.58	1.42	3.14	2.69	3.74	4.3
sp06	2.7	3.61	4.35	2.77	1.93	5.12	2.83	13.6	5.42
sp07	2.58	2.46	3.2	2.77	1.64	5.14	2.91	9	6.41
sp08	2.59	3.23	3.23	2.48	1.17	3	2.55	2.99	2.93

sp09	2.18	2.57	2.6	2.27	3.5	2.75	2.34	15.1	2.47
sp11	2.41	2.1	3.01	2.38	0.996	2.94	2.54	1.5	2.95
sp12	2.44	2.24	3.03	2.6	3.58	3.36	2.66	21.7	3.49
sp13	2.48	2.65	2.99	2.54	2.89	2.94	2.39	20.3	2.55
sp14	2.51	5.5	2.88	2.58	4.34	3.08	2.72	37.8	4.14
sp16	2.41	1.98	3	2.7	2.85	4.41	2.62	14.5	3.14
sp17	2.38	1.23	2.94	2.29	1.16	2.78	2.44	1.14	3.03
sp18	2.44	2.21	3.03	2.64	2.12	3.76	2.57	25.8	2.61
sp19	2.42	3.21	3.02	2.6	2.61	3.29	2.58	25.5	2.65

Table 4.3: Predicted speech quality for different algorithms: babble noise SNR 5&10 dB

Wave file name	**HSE**			**logMMSE + HSE**			**MB + HSE**		
_car_sn5	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**
sp01	2.2	2.15	2.63	2.27	3.62	2.75	2.3	16.8	2.4
sp02	2.32	2.06	2.84	2.44	2.15	3.03	2.35	16.6	2.48
sp03	2.24	3.16	2.71	2.3	5.24	2.73	2.43	32	2.59
sp04	2.28	2.52	2.78	2.4	2.49	2.98	2.32	7.64	2.64
sp06	2.24	3.83	2.69	2.22	4.27	2.63	2.23	25.9	2.26
sp07	2.12	2.89	2.49	2.45	3.67	3.01	2.43	15.1	2.63
sp08	2.28	4.97	2.7	2.39	4.18	2.94	2.35	39	2.45
sp09	1.94	2.65	2.18	2.33	3.05	2.86	2.34	8.47	2.65
sp11	2.12	3.29	2.49	2.46	3.46	3.01	2.34	27.4	2.44
sp12	2.09	2.78	2.43	2.44	2.79	3.03	2.47	18.3	2.59
sp13	2.09	2.02	2.45	2.09	2.02	2.45	2.16	4.86	2.49
sp14	2.25	4.01	2.7	2.47	5.91	2.89	2.43	39	2.58
sp16	2.36	2.5	2.92	2.3	2.71	2.82	2.4	7.46	2.78
sp17	2.02	2.02	2.32	2.21	1.73	2.65	2.29	5.51	2.7
sp18	2.12	2.28	2.5	2.15	2.3	2.54	2.21	11.8	2.25
sp19	2.34	4.28	2.83	2.36	3.12	2.92	2.39	26.5	2.54
_car_sn10	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**
sp01	2.35	2.74	2.9	2.31	3.68	2.83	2.43	42.6	2.57
sp02	2.53	2.74	2.95	2.68	3.3	4.21	2.62	25.8	3.09
sp03	2.57	4.23	2.91	2.64	3.66	3.77	2.83	40.9	5.26
sp04	2.43	2.34	3.03	2.52	3.09	2.96	2.43	9.97	2.71
sp06	2.66	8.27	3.68	2.76	11.6	4.6	2.9	68.2	6.01
sp07	2.43	3.43	3.03	2.84	2.45	5.85	2.89	17.9	6.04
sp08	2.55	9.85	2.62	2.55	11.5	2.55	2.71	59	3.95

sp09	2.2	2.72	2.64	2.58	3.39	3.05	2.58	11.8	2.7
sp11	2.39	6.41	2.82	2.61	10.7	3.02	2.67	65.5	3.53
sp12	2.35	2.64	2.9	2.81	2.83	5.51	2.83	24.4	5.35
sp13	2.56	2.68	2.92	2.8	2.71	5.5	2.69	15.1	3.81
sp14	2.52	7.72	2.76	2.75	10.2	4.59	2.69	53.8	3.73
sp16	2.63	2.22	3.63	2.64	2.14	3.78	2.62	7.67	3.27
sp17	2.56	1.8	2.93	2.78	1.5	5.26	2.76	4.16	5.02
sp18	2.48	2.73	3	2.68	2.66	4.18	2.68	19.2	3.78
sp19	2.54	8.62	2.69	2.71	4.54	4.41	2.73	33.6	4.22

Table 4.4: Predicted speech quality score for different algorithms: car noise SNR 5&10 dB

Wave file name	**HSE**			**logMMSE + HSE**			**MB + HSE**		
_street_sn5	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**
sp01	2.1	2.43	2.46	2.05	2.27	2.37	2.19	2.89	2.62
sp02	2.18	2.76	2.6	2.16	3.11	2.56	2.14	14.4	2.13
sp03	2.15	1.98	2.55	2.01	2.39	2.3	2.37	13.1	2.52
sp04	2.28	2.57	2.77	2.13	2.28	2.51	2.14	4.7	2.47
sp06	2.35	3.67	2.9	2.38	2.83	2.96	2.46	24.7	2.59
sp07	2.11	2.17	2.48	2.07	2.03	2.42	2.37	3.36	2.92
sp08	2.16	5.38	2.48	2.1	4.21	2.43	2.08	25.9	2
sp09	1.8	2.75	1.94	1.83	2.68	1.98	2.01	3.89	2.3
sp11	2.1	5.83	2.35	2.17	15.8	2.18	2.01	72.6	1.77
sp12	2.06	2.21	2.4	2.19	1.81	2.62	2.33	4.28	2.82
sp13	2.25	2.03	2.72	2.33	2.83	2.86	2.25	6.84	2.57
sp14	2.24	5.67	2.59	2.37	8.97	2.66	2.31	52.8	2.35
sp16	2.39	2.63	2.96	2.53	2.49	2.95	2.51	9.32	2.68
sp17	2.32	2.52	2.84	2.33	3.53	2.86	2.23	35.2	2.25
sp18	2.08	2.21	2.42	2.04	2.05	2.36	2.13	2.96	2.51
sp19	2.45	8.46	2.78	2.41	13.7	2.59	2.41	44.5	2.54
_street_sn10	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**
sp01	2.57	4.98	2.94	2.55	7.78	2.73	2.64	36	3.28
sp02	2.67	2.84	4.03	2.67	1.83	4.07	2.7	5.88	4.27
sp03	2.54	3.87	2.93	2.56	3.63	2.92	2.57	48.2	2.54
sp04	2.39	2.28	2.97	2.42	1.57	3.02	2.52	2.43	2.96
sp06	2.58	7.17	2.95	2.68	5.94	4.03	2.6	36.7	2.81
sp07	2.52	3.12	2.96	2.62	1.96	3.48	2.65	3	3.82
sp08	2.4	5.99	2.86	2.46	1.98	3.01	2.58	7.75	2.9

sp09	2.19	2.31	2.62	2.41	1.92	3	2.64	8.1	3.53
sp11	2.37	11.3	2.55	2.47	11.7	2.6	2.59	69.5	2.63
sp12	2.3	2.47	2.8	2.32	2.03	2.85	2.47	8.77	2.75
sp13	2.73	3.42	4.69	2.76	2.87	5.01	2.62	10.9	3.16
sp14	2.66	16.7	3.52	2.69	14.9	3.87	2.62	70.7	3.05
sp16	2.67	2.85	4.02	2.73	5.12	4.61	2.56	37.4	2.47
sp17	2.56	2.22	2.92	2.65	1.79	3.86	2.72	7.41	4.43
sp18	2.45	2.71	3.02	2.53	1.64	2.95	2.66	3.64	3.94
sp19	2.57	16.9	2.55	2.58	15.7	2.66	2.62	71.9	2.97

Table 4.5: Predicted speech quality score for different algorithms: street noise SNR 5&10 dB

Wave file name	**HSE**			**logMMSE + HSE**			**MB + HSE**		
_train_sn5	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**
sp01	1.97	2.22	2.24	1.97	2.37	2.23	2.22	7.02	2.51
sp02	2.25	2.1	2.71	2.25	2.46	2.72	2.37	11.2	2.55
sp03	2.27	3.4	2.76	2.23	9.52	2.39	2.26	65.3	2.22
sp04	2.08	2.86	2.43	2.22	5.05	2.6	2.18	27.8	2.17
sp06	2.33	3.57	2.85	2.31	3.17	2.83	2.42	12.6	2.62
sp07	2.23	3.32	2.68	2.19	10.8	2.27	2.43	56.4	2.55
sp08	2.26	6.95	2.58	2.32	12.1	2.43	2.15	49.9	2.07
sp09	1.77	3.37	1.88	2.05	4.08	2.35	1.98	12.6	1.86
sp11	1.95	3.01	2.2	2.25	4.36	2.68	2.19	23.8	2.2
sp12	1.87	2.22	2.06	2.14	2.29	2.53	2.15	12.8	2.14
sp13	2.22	2.12	2.67	2.28	2.02	2.78	2.25	3.73	2.71
sp14	2.1	3.46	2.46	2.27	5.56	2.66	2.17	50.1	2.1
sp16	2.28	2.7	2.77	2.33	3.07	2.85	2.18	6.71	2.44
sp17	2.09	2.79	2.45	2.38	4.08	2.93	2.21	25	2.23
sp18	2.14	2.46	2.54	2.4	4.19	2.94	2.3	12.7	2.41
sp19	2.22	3.01	2.67	2.26	2.78	2.73	2.37	10.3	2.6
_train_sn10	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**	**PESQ**	**IS**	**Y_TOTAL**
sp01	2.41	2.67	2.99	2.51	6.18	2.84	2.52	38.3	2.51
sp02	2.51	2.2	2.97	2.55	2.16	2.94	2.68	10.8	3.79
sp03	2.32	2.48	2.84	2.41	2.95	2.99	2.5	19.1	2.56
sp04	2.34	2.97	2.88	2.43	3.77	3.02	2.62	19	3.06
sp06	2.49	3.97	2.96	2.45	1.63	3.02	2.53	8.42	2.71
sp07	2.34	2.86	2.87	2.37	2.6	2.93	2.61	16.3	2.99
sp08	2.48	4.43	2.95	2.54	1.59	2.94	2.75	8	4.7

sp09	2.07	2.38	2.41	2.27	1.86	2.75	2.56	4.66	2.87
sp11	2.23	2.33	2.68	2.5	3.51	2.98	2.54	22.4	2.52
sp12	2.42	3.15	3.01	2.65	4.18	3.77	2.67	26.4	3.66
sp13	2.45	2.45	3.02	2.61	5.48	3.35	2.57	9.97	2.67
sp14	2.51	5.2	2.89	2.57	4.16	3.01	2.71	38.6	4.06
sp16	2.58	2.77	3.09	2.66	3.24	3.95	2.49	14.3	2.58
sp17	2.48	3.04	3	2.78	3.33	5.23	2.74	24.2	4.36
sp18	2.42	3.12	3.03	2.6	3.62	3.28	2.58	23.1	2.69
sp19	2.56	5.68	2.82	2.63	3.42	3.68	2.78	23.9	4.84

Table 4.6: Predicted speech quality score for different algorithms: train noise SNR 5&10 dB

In the following tables, the names of wave files are to be deciphered as in the example below:

In "R21S01A90.wav", R implies Room, 2 implies room number (NCA database 1-low RT60, 2-moderate RT60 of 0.88 seconds, 3-another moderate RT60, 4- severe RT60 of 1.39 seconds and 5-anechoic chamber; AIR database 1-booth, 2-office, 3-meeting, 4-lecture, 5-stairway), 1 implies speaker-microphone distances if any, S01 implies name of the clean speech file that was convolved with the impulse response and A90 implies azimuth 90.

	SRMR Values				
Filename (Reverb Corrupt Speech)	**Input RT_{60}=1 sec**	**Output RT60= 1sec (logMMSE+HSE)**	**Filename (Reverb Corrupt Speech)**	**Input RT_{60}= 1.5 sec**	**Output RT60= 1.5 sec (logMMSE+HSE)**
R21S01A90.wav	1.591	3.884	R41S01A90R.wav	1.142	2.391
R21S02A90.wav	2.04	5.385	R41S02A90R.wav	1.574	4.3
R21S03A90.wav	1.579	4.05	R41S03A90R.wav	1.175	3.074
R21S04A90.wav	1.47	2.726	R41S04A90R.wav	1.066	1.959
R21S05A90.wav	1.625	4.135	R41S05A90R.wav	1.178	2.938
R21S06A90.wav	1.596	3.73	R41S06A90R.wav	1.153	2.484
R21S07A90.wav	1.424	2.683	R41S07A90R.wav	1.113	2.143
R21S08A90.wav	2.302	7.394	R41S08A90R.wav	1.527	3.752
R21S09A90.wav	3.069	9.071	R41S09A90R.wav	2.151	6.153
R21S10A90.wav	2.678	6.857	R41S10A90R.wav	1.952	4.925

R21S11A90.wav	2.986	7.147	R41S11A90R.wav	1.92	4.493
R21S12A90.wav	1.949	4.145	R41S12A90R.wav	1.439	3.243
R21S13A90.wav	2.475	5.548	R41S13A90R.wav	1.871	4.859
R21S14A90.wav	2.201	5.087	R41S14A90R.wav	1.622	4.126
R21S15A90.wav	1.741	4.007	R41S15A90R.wav	1.23	3.143
R21S16A90.wav	1.473	2.637	R41S16A90R.wav	1.141	2.3

Table 4.7: SRMR modulation values for 2 Reverberation time sets (Speech Enhancement)

	SRMR Values				
Filename (Reverb Corrupt Speech)	**Input RT_{60}=1 sec**	**Output RT60= 1sec (logMMSE+HSE)**	**Filename (Reverb Corrupt Speech)**	**Input RT_{60}= 1.5 sec**	**Output RT60= 1.5 sec (logMMSE+HSE)**
R21S01A90.wav	1.5923	3.9160	R41S01A90R.wav	1.1420	2.3381
R21S02A90.wav	2.2395	5.9164	R41S02A90R.wav	1.5739	4.8711
R21S03A90.wav	1.6530	4.4495	R41S03A90R.wav	1.1755	3.2794
R21S04A90.wav	1.5345	2.7658	R41S04A90R.wav	1.0664	2.0504
R21S05A90.wav	1.8040	4.5343	R41S05A90R.wav	1.1783	3.1573
R21S06A90.wav	1.9791	4.8392	R41S06A90R.wav	1.1534	3.0583
R21S07A90.wav	1.5116	2.9130	R41S07A90R.wav	1.1129	2.2588
R21S08A90.wav	2.6647	8.2069	R41S08A90R.wav	1.5274	4.2609
R21S09A90.wav	3.1759	10.2854	R41S09A90R.wav	2.1511	6.8170
R21S10A90.wav	2.7581	6.9826	R41S10A90R.wav	1.9523	4.6559
R21S11A90.wav	3.3099	7.9329	R41S11A90R.wav	1.9201	5.3477
R21S12A90.wav	1.9629	4.2918	R41S12A90R.wav	1.4389	3.5659
R21S13A90.wav	2.3921	5.3436	R41S13A90R.wav	1.8711	4.8371
R21S14A90.wav	2.4174	5.8504	R41S14A90R.wav	1.6216	4.7059
R21S15A90.wav	1.6800	3.8388	R41S15A90R.wav	1.2304	2.8291
R21S16A90.wav	1.5022	2.6244	R41S16A90R.wav	1.1407	2.2862

Table 4.8: SRMR modulation values for 2 Reverberation time sets (BWE)

NOIZEUS is a noisy speech corpus for evaluation of speech enhancement algorithms. The details of speech material, noise sources and algorithms that are used to compare against can be obtained from [12], [14].

Table 1: List of sentences used in NOIZEUS			
Filename	**Speaker**	**Gender**	**Sentence**
sp01.wav	CH	M	The birch canoe slid on the smooth
sp02.wav	CH	M	He knew the skill of the great young actress
sp03.wav	CH	M	Her purse was full of useless trash
sp04.wav	CH	M	Read verse out loud for pleasure
sp05.wav	CH	M	Wipe the grease off his dirty face
sp06.wav	DE	M	Men strive but seldom get rich
sp07.wav	DE	M	We find joy in the simplest things
sp08.wav	DE	M	Hedge apples may stain your hands green
sp09.wav	DE	M	Hurdle the pit with the aid of a long pole
sp10.wav	DE	M	The sky that morning was clear and bright blue
sp11.wav	JE	F	He wrote down a long list of items
sp12.wav	JE	F	The drip of the rain made a pleasant sound
sp13.wav	JE	F	Smoke poured out of every crack
sp14.wav	JE	F	Hats are worn to tea and not to dinner
sp15.wav	JE	F	The clothes dried on a thin wooden rack
Table 2: List of sentences used in NOIZEUS			
sp16.wav	KI	F	The stray cat gave birth to kittens
sp17.wav	KI	F	The lazy cow lay in the cool grass
sp18.wav	KI	F	The friendly gang left the drug store
sp19.wav	KI	F	We talked of the sideshow in the circus
sp20.wav	KI	F	The set of china hit the floor with a crash
sp21.wav	SI	M	Clams are small, round, soft and tasty
sp22.wav	SI	M	The line where the edges join was clean
sp23.wav	SI	M	Stop whistling and watch the boys march
sp24.wav	SI	M	A cruise in warm waters in a sleek yacht is fun
sp25.wav	SI	M	A good book informs of what we ought to know
sp26.wav	TI	F	She has a smart way of wearing clothes
sp27.wav	TI	F	Bring your best compass to the third class
sp28.wav	TI	F	The club rented the rink for the fifth night
sp29.wav	TI	F	The flint sputtered and lit a pine torch
sp30.wav	TI	F	Let us all join as we sing the last chorus

Table 4.9: The sentences used in the subjective evaluation are underlined. Courtesy [12]

	Reverb Input	Dereverb Output		Reverb Input	Dereverb Output		Reverb Input	Dereverb Output
Distance	0.5m	0.5m		1m	1m		1.5m	1.5m
R11S01A90R.wav	4.3	13	R12S01A90R.wav	3.7	7.1	R13S01A90R.wav	3.7	6.3
R11S02A90R.wav	5.9	11	R12S02A90R.wav	5.1	8.8	R13S02A90R.wav	4.9	7.6
R11S03A90R.wav	3.8	7	R12S03A90R.wav	3.9	7.3	R13S03A90R.wav	4.3	6.6
R11S04A90R.wav	3.8	5.4	R12S04A90R.wav	3.3	4.5	R13S04A90R.wav	3.8	5.7

R11S05A90R.wav	2.6	4.7	R12S05A90R.wav	2.7	5.1	R13S05A90R.wav	4	9.5
R11S06A90R.wav	3	5.1	R12S06A90R.wav	3.3	5.5	R13S06A90R.wav	3	4.5
R11S07A90R.wav	4.7	9.4	R12S07A90R.wav	4.8	9.9	R13S07A90R.wav	5.9	12
R11S08A90R.wav	6.1	14	R12S08A90R.wav	6.1	17	R13S08A90R.wav	6.1	10
R11S09A90R.wav	8.4	14	R12S09A90R.wav	6.6	12	R13S09A90R.wav	8.1	20
R11S10A90R.wav	12	20	R12S10A90R.wav	10	17	R13S10A90R.wav	8.9	16
R11S11A90R.wav	9.5	13	R12S11A90R.wav	8	11	R13S11A90R.wav	8.1	11
R11S12A90R.wav	11	15	R12S12A90R.wav	8.4	12	R13S12A90R.wav	8.2	11
R11S13A90R.wav	12	17	R12S13A90R.wav	12	17	R13S13A90R.wav	11	17
R11S14A90R.wav	14	28	R12S14A90R.wav	13	24	R13S14A90R.wav	10	18
R11S15A90R.wav	8.6	14	R12S15A90R.wav	7.9	14	R13S15A90R.wav	8.6	17
R11S16A90R.wav	10	13	R12S16A90R.wav	8.9	12	R13S16A90R.wav	7.6	10
	7.48	12.72		6.73	11.51		6.63	11.38

Table 4.10: SRMR Dereverberation results for Booth room

	Reverb Input	**Dereverb Out**		**Reverb Input**	**Dereverb Output**		**Reverb Input**	**Dereverb Output**
Distance	1m			2m			3m	
R21S01A90R.wav	2.5	5.3	R22S01A90R.wav	2.5	4.9	R23S01A90R.wav	2.1	4.1
R21S02A90R.wav	4.7	14	R22S02A90R.wav	3.1	7.3	R23S02A90R.wav	2.1	3.9
R21S03A90R.wav	2.9	6.6	R22S03A90R.wav	3.1	7.3	R23S03A90R.wav	2.4	5.3
R21S04A90R.wav	2.8	4.8	R22S04A90R.wav	2.3	4.3	R23S04A90R.wav	2.3	4.6
R21S05A90R.wav	2.7	7.6	R22S05A90R.wav	2.2	5	R23S05A90R.wav	2	4
R21S06A90R.wav	2.7	6	R22S06A90R.wav	2.9	6.2	R23S06A90R.wav	1.9	3.9
R21S07A90R.wav	3.5	9.6	R22S07A90R.wav	3.1	7.4	R23S07A90R.wav	2.3	4.5
R21S08A90R.wav	3.6	7.9	R22S08A90R.wav	3	7.2	R23S08A90R.wav	2.3	4.7
R21S09A90R.wav	6.5	17	R22S09A90R.wav	4	7.7	R23S09A90R.wav	4.6	12
R21S10A90R.wav	4.8	7	R22S10A90R.wav	4.9	8.6	R23S10A90R.wav	4.7	12
R21S11A90R.wav	5.9	11	R22S11A90R.wav	5.3	9.1	R23S11A90R.wav	4.4	11
R21S12A90R.wav	6.6	13	R22S12A90R.wav	4.2	8.1	R23S12A90R.wav	3.2	6.4
R21S13A90R.wav	6.6	11	R22S13A90R.wav	5.8	13	R23S13A90R.wav	3.2	5.2
R21S14A90R.wav	6.7	14	R22S14A90R.wav	5.3	14	R23S14A90R.wav	4.1	7.9
R21S15A90R.wav	4.6	8.5	R22S15A90R.wav	3.7	7.9	R23S15A90R.wav	4.2	11
R21S16A90R.wav	5.1	8	R22S16A90R.wav	4.2	8.4	R23S16A90R.wav	3.2	5.8
Mean	4.51	9.45		3.72	7.9		3.06	6.64

Table 4.11: SRMR Dereverberation results for Office room

	Reverb Input	**Dereverb Output**		**Reverb Input**	**Dereverb Output**		**Reverb Input**	**Dereverb Output**
Distance	1.45m			1.7m			1.9m	
R31S01A90R.wav	4.3	12	R32S01A90R.wav	4.5	13	R33S01A90R.wav	4.3	14
R31S02A90R.wav	5.9	16	R32S02A90R.wav	3.6	5.9	R33S02A90R.wav	4.5	11
R31S03A90R.wav	4.1	9.1	R32S03A90R.wav	4.4	11	R33S03A90R.wav	4.9	15
R31S04A90R.wav	3.5	5.7	R32S04A90R.wav	3.3	5.9	R33S04A90R.wav	3.2	5.2
R31S05A90R.wav	2.8	5.2	R32S05A90R.wav	3.2	8.4	R33S05A90R.wav	2.9	6.5
R31S06A90R.wav	3.9	7.6	R32S06A90R.wav	3.3	7.7	R33S06A90R.wav	3.1	7

R31S07A90R.wav	4	12	R32S07A90R.wav	4.1	13	R33S07A90R.wav	3.9	12
R31S08A90R.wav	8.5	19	R32S08A90R.wav	4.4	12	R33S08A90R.wav	4.4	12
R31S09A90R.wav	6.2	11	R32S09A90R.wav	8.6	19	R33S09A90R.wav	8.3	19
R31S10A90R.wav	7.5	15	R32S10A90R.wav	7.3	13	R33S10A90R.wav	7	13
R31S11A90R.wav	7.5	9.7	R32S11A90R.wav	8.3	12	R33S11A90R.wav	7.4	11
R31S12A90R.wav	6	11	R32S12A90R.wav	6.2	11	R33S12A90R.wav	5.7	10
R31S13A90R.wav	9.2	16	R32S13A90R.wav	8.7	14	R33S13A90R.wav	10	15
R31S14A90R.wav	7.2	13	R32S14A90R.wav	8	16	R33S14A90R.wav	8	16
R31S15A90R.wav	6.8	15	R32S15A90R.wav	6.1	11	R33S15A90R.wav	5.8	12
R31S16A90R.wav	7.5	12	R32S16A90R.wav	6.6	9.9	R33S16A90R.wav	5.9	8
	5.93	11.83		5.66	11.42		5.58	11.66
	Reverb Input	**Dereverb Output**		**Reverb Input**	**Dereverb Output**			
	2.25m			2.8m				
R34S01A90R.wav	3.5	8.3	R35S01A90R.wav	3	5.9			
R34S02A90R.wav	4	7.8	R35S02A90R.wav	3.6	6.5			
R34S03A90R.wav	3.7	7.9	R35S03A90R.wav	3.3	6.3			
R34S04A90R.wav	2.8	5.2	R35S04A90R.wav	2.5	3.6			
R34S05A90R.wav	2.6	6.1	R35S05A90R.wav	3	7.2			
R34S06A90R.wav	2.8	5.2	R35S06A90R.wav	2.7	4.9			
R34S07A90R.wav	3.4	7.6	R35S07A90R.wav	3.7	7.4			
R34S08A90R.wav	4.2	12	R35S08A90R.wav	3.9	11			
R34S09A90R.wav	6.7	14	R35S09A90R.wav	6	15			
R34S10A90R.wav	7.5	15	R35S10A90R.wav	5.5	8.9			
R34S11A90R.wav	8.3	14	R35S11A90R.wav	8.3	18			
R34S12A90R.wav	5.9	10	R35S12A90R.wav	4.5	6.5			
R34S13A90R.wav	8.7	15	R35S13A90R.wav	6.8	13			
R34S14A90R.wav	7.1	13	R35S14A90R.wav	7.5	17			
R34S15A90R.wav	5.7	10	R35S15A90R.wav	5.1	9.1			
R34S16A90R.wav	5.3	7.5	R35S16A90R.wav	4.5	7			
	5.13	9.91		4.61	9.20			

Table 4.12: SRMR Dereverberation results for Meeting room

	Reverb Input	**Dereverb Output**		**Reverb Input**	**Dereverb Output**		**Reverb Input**	**Dereverb Output**
Distance	2.25m	2.25m		4m	4m		5.56m	5.56m
R41S01A90R.wav	2.3	4.7	R42S01A90R.wav	2.6	5.4	R43S01A90R.wav	1.8	3.4
R41S02A90R.wav	3.9	9.7	R42S02A90R.wav	2.9	6.2	R43S02A90R.wav	2.3	5
R41S03A90R.wav	2.9	7	R42S03A90R.wav	3	8	R43S03A90R.wav	2	4.4
R41S04A90R.wav	2.4	4.2	R42S04A90R.wav	2.2	3.6	R43S04A90R.wav	2	3.7
R41S05A90R.wav	2.3	6	R42S05A90R.wav	1.8	3.7	R43S05A90R.wav	2	4.7

R41S06A90R.wav	2.8	7.3	R42S06A90R.wav	2	3.5	R43S06A90R.wav	1.9	3.6
R41S07A90R.wav	3.2	12	R42S07A90R.wav	2.8	7.7	R43S07A90R.wav	2.2	4.5
R41S08A90R.wav	4.4	13	R42S08A90R.wav	3.8	15	R43S08A90R.wav	3.2	14
R41S09A90R.wav	4	8.1	R42S09A90R.wav	4.3	10	R43S09A90R.wav	3.3	8.5
R41S10A90R.wav	5.7	13	R42S10A90R.wav	7.3	18	R43S10A90R.wav	4.6	11
R41S11A90R.wav	5.4	11	R42S11A90R.wav	4.6	8	R43S11A90R.wav	3.3	5.4
R41S12A90R.wav	4.5	9.2	R42S12A90R.wav	3.3	5.9	R43S12A90R.wav	2.9	6.5
R41S13A90R.wav	5.5	15	R42S13A90R.wav	4.3	9.2	R43S13A90R.wav	2.8	5.7
R41S14A90R.wav	5	10	R42S14A90R.wav	5	11	R43S14A90R.wav	3.6	9.3
R41S15A90R.wav	4.2	8.9	R42S15A90R.wav	4.1	11	R43S15A90R.wav	2.8	6.8
R41S16A90R.wav	3.7	6.5	R42S16A90R.wav	3.8	8.8	R43S16A90R.wav	3.6	7.7
Mean	3.88	9.1		3.6	8.4		2.76	6.51
	Reverb Input	**Dereverb Output**		**Reverb Input**	**Dereverb Output**		**Reverb Input**	**Dereverb Output**
	7.1m			8.68m			10.2m	
R44S01A90R.wav	1.9	3.4	R45S01A90R.wav	2.1	4.2	R46S01A90R.wav	2.5	5.5
R44S02A90R.wav	2.3	6	R45S02A90R.wav	2.4	5.8	R46S02A90R.wav	2.8	6.3
R44S03A90R.wav	1.9	3.8	R45S03A90R.wav	1.8	3.5	R46S03A90R.wav	2	4
R44S04A90R.wav	1.8	3.6	R45S04A90R.wav	1.5	2.7	R46S04A90R.wav	2.2	4.3
R44S05A90R.wav	1.6	3.1	R45S05A90R.wav	1.6	3.5	R46S05A90R.wav	1.9	3.9
R44S06A90R.wav	2.1	4.8	R45S06A90R.wav	2.1	4.8	R46S06A90R.wav	2.1	4
R44S07A90R.wav	2.3	4.9	R45S07A90R.wav	1.9	4.1	R46S07A90R.wav	2	3.8
R44S08A90R.wav	2.1	6	R45S08A90R.wav	3.1	13	R46S08A90R.wav	2.8	9
R44S09A90R.wav	3.3	8.6	R45S09A90R.wav	2.6	5.3	R46S09A90R.wav	3.5	8.6
R44S10A90R.wav	3.5	6.9	R45S10A90R.wav	5.6	16	R46S10A90R.wav	5	16
R44S11A90R.wav	3.7	7	R45S11A90R.wav	4.7	9.4	R46S11A90R.wav	3.6	6.8
R44S12A90R.wav	3	6.4	R45S12A90R.wav	3	7.1	R46S12A90R.wav	2.6	6.1
R44S13A90R.wav	2.9	6.2	R45S13A90R.wav	2.9	6.3	R46S13A90R.wav	3.9	14
R44S14A90R.wav	2.9	6.9	R45S14A90R.wav	3.1	6.6	R46S14A90R.wav	3	7

R44S15A90R.wav	2.4	5.5	R45S15A90R.wav	2.4	6.2	R46S15A90R.wav	2.1	4.9
R44S16A90R.wav	2.4	4.7	R45S16A90R.wav	3.3	8.4	R46S16A90R.wav	2.8	5.4
Mean	2.50	5.4		2.7	6.68		2.8	6.85

Table 4.13: SRMR Dereverberation results for Lecture room

	Reverb Input	**Dereverb Output**		**Reverb Input**	**Dereverb Output**		**Reverb Input**	**Dereverb Output**
Distance	1m	1m		2m	2m		3m	3m
R51S01A90R.wav	2.5	5.4	R52S01A90R.wav	2.5	6.9	R53S01A90R.wav	1.9	3.7
R51S02A90R.wav	3.6	7.5	R52S02A90R.wav	2.5	4.9	R53S02A90R.wav	2.5	6.8
R51S03A90R.wav	2.5	5.4	R52S03A90R.wav	3.2	8.3	R53S03A90R.wav	2.3	4.8
R51S04A90R.wav	2.5	4.4	R52S04A90R.wav	2.1	4	R53S04A90R.wav	2	3.6
R51S05A90R.wav	2	4.1	R52S05A90R.wav	2	4.3	R53S05A90R.wav	2.2	5.2
R51S06A90R.wav	2.3	4.3	R52S06A90R.wav	2.4	4.9	R53S06A90R.wav	1.5	2.6
R51S07A90R.wav	3.1	7.7	R52S07A90R.wav	3	9.7	R53S07A90R.wav	2.6	6.4
R51S08A90R.wav	4.2	14	R52S08A90R.wav	3.2	8.6	R53S08A90R.wav	2.4	5.5
R51S09A90R.wav	4.5	9.1	R52S09A90R.wav	4.2	9.4	R53S09A90R.wav	3.7	8.9
R51S10A90R.wav	6.9	14	R52S10A90R.wav	3.3	5.9	R53S10A90R.wav	3.6	7.8
R51S11A90R.wav	5.7	9.7	R52S11A90R.wav	5	9.7	R53S11A90R.wav	3.5	7.7
R51S12A90R.wav	5.4	9.9	R52S12A90R.wav	4.6	8.7	R53S12A90R.wav	3.2	6.3
R51S13A90R.wav	7.7	15	R52S13A90R.wav	5.3	13	R53S13A90R.wav	3.7	7.8
R51S14A90R.wav	7	16	R52S14A90R.wav	5.8	13	R53S14A90R.wav	4	10
R51S15A90R.wav	5.3	11	R52S15A90R.wav	3.4	7.8	R53S15A90R.wav	3.1	7.3
R51S16A90R.wav	6.3	11	R52S16A90R.wav	3.4	5.5	R53S16A90R.wav	2.6	4.8
	4.46	9.28		3.49	7.78		2.8	6.2

Table 4.14: SRMR Dereverberation results for Stairway room

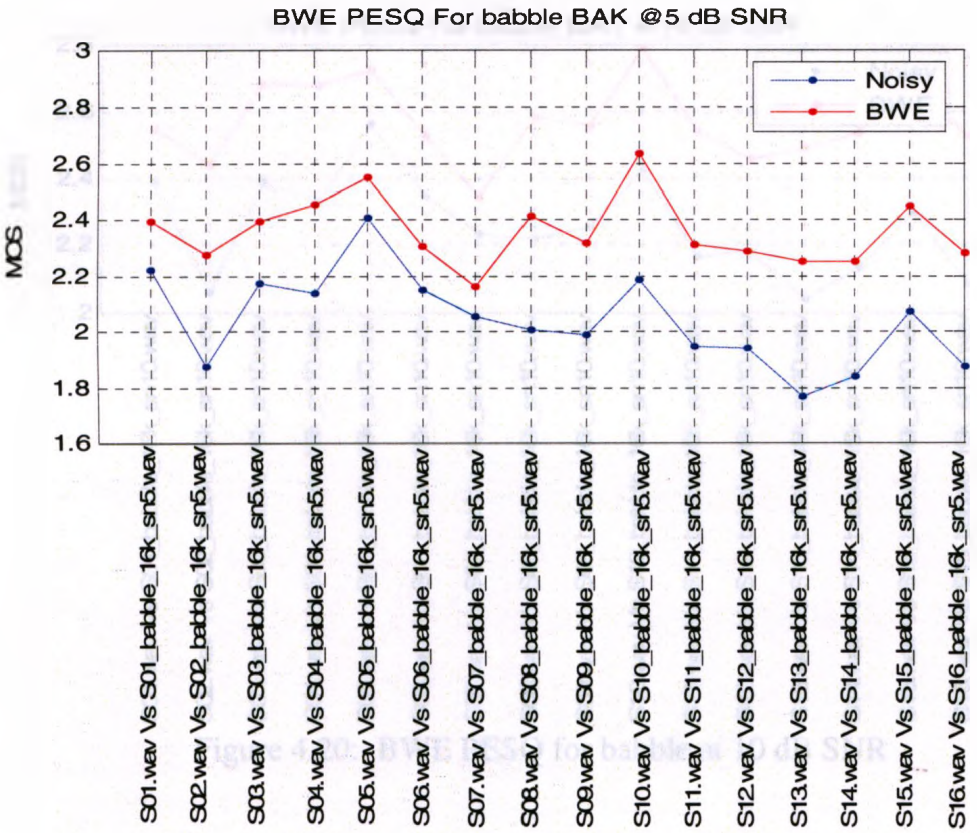

Figure 4.18: BWE PESQ for babble at 5 dB SNR

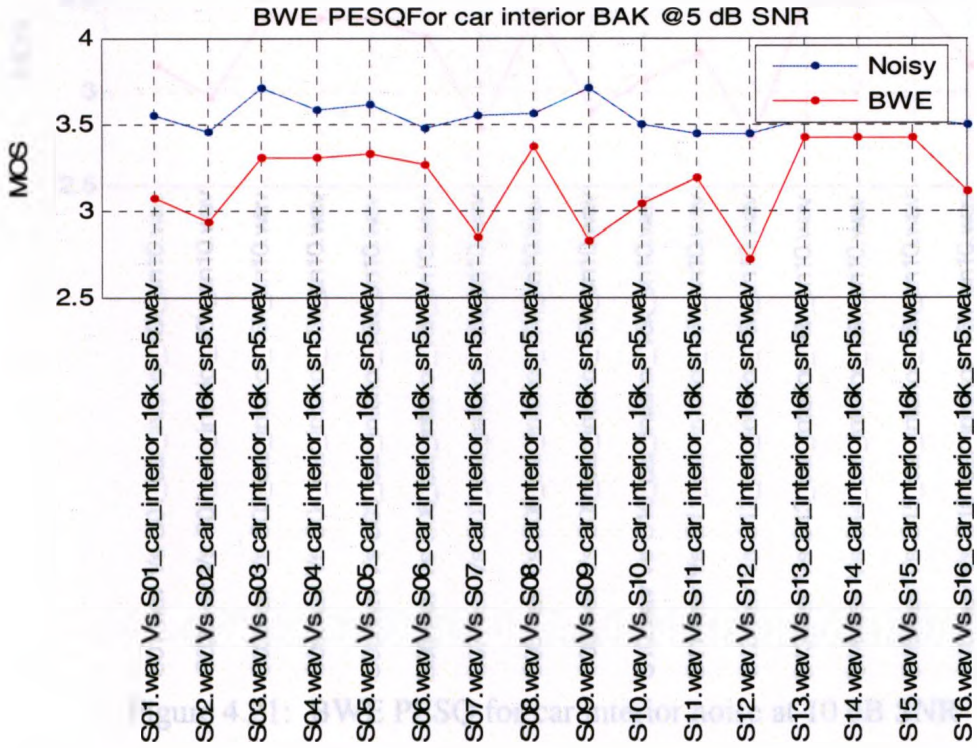

Figure 4.19: BWE PESQ for car Interior noise at 10 dB SNR

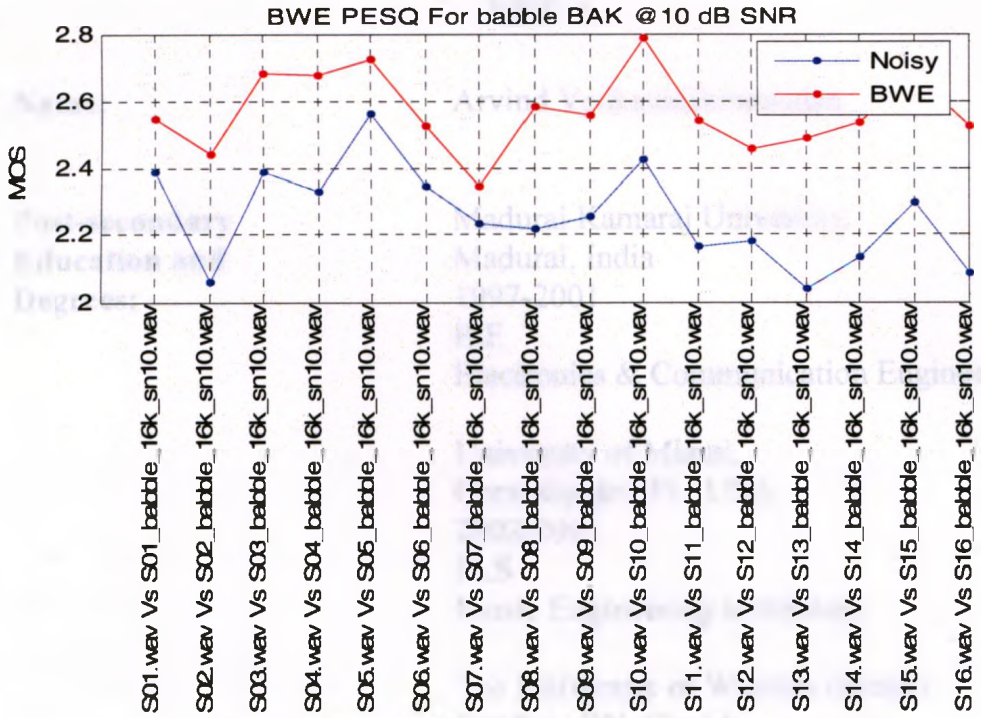

Figure 4.20: BWE PESQ for babble at 10 dB SNR

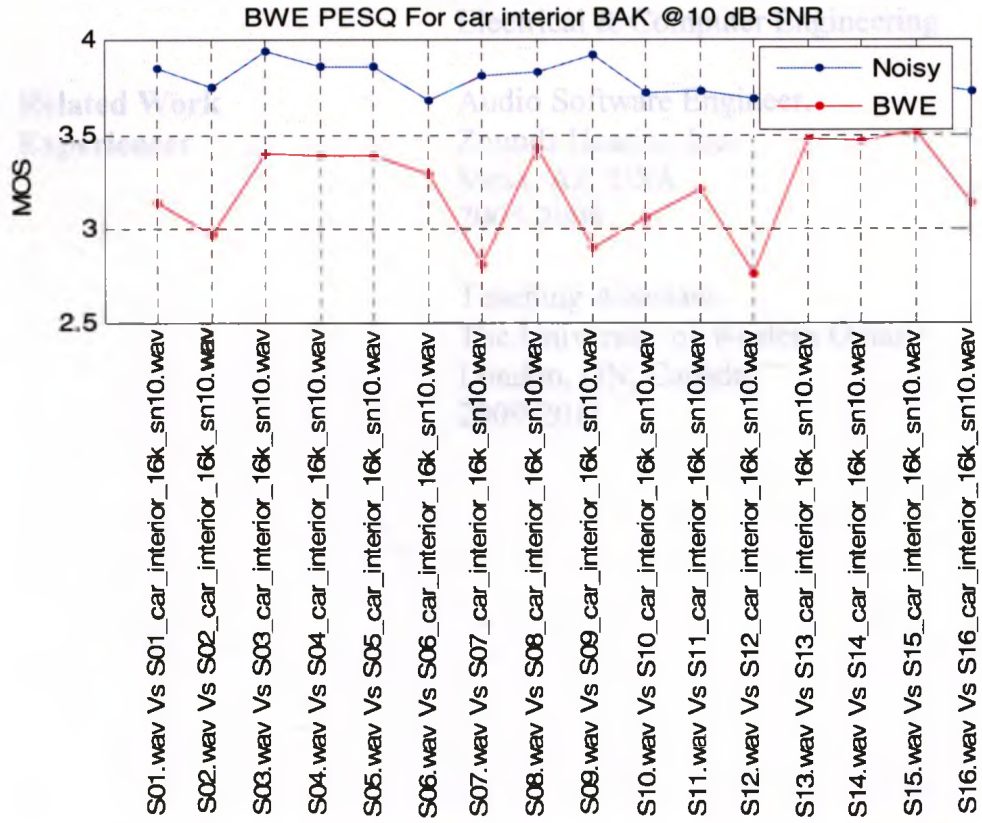

Figure 4.21: BWE PESQ for car interior noise at 10 dB SNR