DEVELOPMENT OF NEW STRATEGIES FOR REAL TIME EMBEDDED SPEECH ENHANCEMENT AND DETECTION FOR WIRELESS COMMUNICATION

A thesis submitted for the award of the
Degree of

DOCTOR OF PHILOSOPHY
in
Electrical Engineering

By
Jigar H. Shah

सत्यं शिवं सुन्दरम्

Electrical Engineering Department
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara – 390 001
Gujarat, India

November – 2011

Dedicated to

MY TEACHERS

for instilling within me the thirst for knowledge and the quest for excellence.

DEVELOPMENT OF NEW STRATEGIES FOR REAL TIME EMBEDDED SPEECH ENHANCEMENT AND DETECTION FOR WIRELESS COMMUNICATION

A thesis submitted for the award of the Degree of

DOCTOR OF PHILOSOPHY
in
Electrical Engineering

By

Jigar H. Shah

सत्यं शिवं सुन्दरम्

Electrical Engineering Department

Faculty of Technology and Engineering

The Maharaja Sayajirao University of Baroda

Vadodara – 390 001

Gujarat, India

November – 2011

Certificate

This is to certify that the thesis entitled, ***‘Development of New Strategies for Real Time Embedded Speech Enhancement and Detection for Wireless Communication***’ submitted by ***Jigar H. Shah*** in fulfillment of the degree of ***Doctor of Philosophy in Electrical Engineering***, is a bona fide record of investigations carried out by him at the Electrical Engineering Department, Faculty of Technology and Engineering, The M. S. University of Baroda, Vadodara under my guidance and supervision. In my opinion the standards fulfilling the requirements of the Ph.D. Degree as the prescribed regulations of the University has been attained.

Dr. A. N. Mishra
Dean,
Faculty of Technology and Engineering,
The M. S. University of Baroda,
Vadodara – 390 001.

Prof. Satish K. Shah
Guide and Head,
Electrical Engineering Department,
Faculty of Technology and Engineering,
The M. S. University of Baroda,
Vadodara – 390 001.

November 2011

Declaration

I, **Jigar H. Shah** hereby declare that the work reported in this thesis entitled '**Development of New Strategies for Real Time Embedded Speech Enhancement and Detection for Wireless Communication**' to be submitted by me for the award of the degree of **Doctor of Philosophy in Electrical Engineering** is original and has been carried out at the Electrical Engineering Department, Faculty of Technology & Engineering, M. S. University of Baroda, Vadodara. I further declare that this thesis is not substantially the same as one, which has already been submitted in part or in full for the award of any degree or academic qualification of this University or any other Institution or examining body in India or abroad.

November 2011 **Jigar H. Shah**

ABSTRACT

Most of the speech communication applications viz. telephony, hands-free communication, voice recording, automatic speech recognition, interactive voice response system, human-machine interfaces, etc. that require at least one microphone, desired speech signal is usually contaminated by background noise and reverberation. As a result, the speech signal has to be “cleaned” with digital signal processing tools before it is played out, transmitted, or stored. An attempt has been made here to explain the requirements and scope of improvements in the field of speech enhancement and its real time embedded implementation.

Several single channel speech enhancement and detection strategies are suggested in the past which are surveyed here; out of which the popular ones have been simulated and compared. But still they are offering some hindrances i.e. underperformance at low SNRs (≤5dB) and in real world noisy and reverberant environments. Hence, the objective is to modify or combine single channel speech enhancement and detection algorithms having appreciable noise suppression characteristic in the low SNR range for various real world additive noises such as airport, car, restaurant, train, station etc. as well as in reverberant environments.

Considering the improvisations required in overcoming the flaw of traditional methods; efforts have been made to develop a new hybrid algorithm, which outperforms in adverse conditions. The hybrid algorithm interactively combines MMSESTSA approach and RASTA approach. It has been simulated and evaluated on the objective and subjective scale. The evaluation has been carried out as per IEEE recommendation and ITU guidelines. The outcomes of the method are well appreciable as compared to the other methods.

Finally, the hybrid algorithm is tried for the real time and embedded implementations. The real time implementation is done on PC using SIMULINK from Mathworks and embedded implementation is done on TMS320C6713 DSP from Texas Instruments using DSP Starter kit-DSK 6713 from Spectrum Digital Incorporation. The SIMULINK, Real Time Workshop and Target Support Package TC6 toolboxes from Mathworks and Code Composer Studio version 3.3 from Texas Instruments are used as development tools. The profile report of both the implementations are generated and compared. Final comment is made after comparison of profile results.

ACKNOWLEDGEMENTS

No significant work can be successfully completed without any support. This is a platform to acknowledge all who had given me direct or indirect help. It gives me immense pleasure to express my sincere gratitude towards my guide Prof. S.K.Shah whose constant support and endless and unconditional guidance gave a fruitful result to my endeavor in reaching the depth of the topic. It has been an honor to be his Ph.D. student. He has taught me, both consciously and unconsciously, how productive and stimulating experimental research work can be done. I express my profound thanks to all the faculty members and supporting staff of EED, FTE, Vadodara for their heartily support and invaluable guidance.

Also, I take this opportunity to thank Prin. R.C.Joshi, Campus director, SVMIT, Bharuch, for giving me the opportunity for joining the doctorate studies. He has taken a keen and active interest in timely completion of my Ph.D. work and helped me a lot in the preparation of research proposal, synopsis and thesis writing.

I am also thankful to Dr. S.S.Patel of SVMIT, Bharuch for helping me in understanding of some mathematical and statistical concepts. I am very much thankful to all faculty members and supporting staff of ETC Dept., SVMIT, Bharuch for their great support and for playing the role of listening crew and performing the hectic task of listening various speech outputs and their valuable evaluations.

With immense affection, I wish to express my sincere feelings to my grandmother and mother whose blessings motivated me throughout this endeavor. Also, a special thanks to my wife and son for being supportive all these years and particularly comprehensive with my lack of time.

Last but not the least; I am thankful to all who helped me directly or indirectly in progressing towards my goal.

Contents

Chapter-3

Chapter-10

List of Figures

List of Tables

Nomenclatures

Symbol	Meaning
$y(n)$	Degraded Speech signal
$x(n)$	Clean speech signal
$d(n)$	Additive noise
α	Over subtraction factor
β	Spectral floor parameter
p	Spectral power
η	Smoothing constant
K	Discrete frequency bin
δ	Tweaking factor
μ, v	Parameters of Wiener filter
$\xi(K)$	*A priori* SNR at frequency bin K $= \frac{\lvert\hat{X}(K)\rvert^2}{\lvert\hat{D}(K)\rvert^2}$
$\gamma(K)$	*A posteriori* SNR at frequency bin K $= \frac{\lvert Y(K)\rvert^2}{\lvert\hat{D}(K)\rvert^2}$
i	Frequency band
$\phi_y(K)$	Phase of signal y(n) at frequency bin K
F_s	Sampling frequency
f_i	Upper frequency in the i^{th} frequency band
λ	Wavelet co-efficient threshold
PR	Power ratio

Acronyms

Acronym	Meaning
AIR	Aachen impulse response
BSS	Berouti Spectral Subtraction
CCS	Code Composer Studio
DSK	DSP Starter Kit
ITU	International Telecommunication Union
LLR	Log Likelihood Ratio Distance Measure
LSA	Log Spectral Amplitude
MBSS	Multiband Spectral Subtraction
MMSE	Minimum Mean Square Error
MOS	Mean Opinion Score
PSEQ	Perceptual Evaluation of Speech Quality
RASTA	Relative Spectral Analysis
RTW	Real Time Workshop
SS	Spectral Subtraction
SSNR	Segmental SNR Measure
STDFT	Short time Discrete Fourier Transform
STSA	Short Time Spectral Amplitude
VAD	Voice Activity Detector
WSS	Weighted Spectral Slope Distance Measure

Chapter 1

Introduction

Speech is the primary means of communication among human beings and is a result of complex interaction among vocal folds vibration at the larynx and voluntary articulators' movements (i.e. mouth, tongue, jaw, etc.). People use speech to communicate messages amongst themselves. When a speaker and a listener are nearer to each other in a quiet environment, communication is by and large easy and accurate. However, when people are separated by distance or if there is a noisy environment, they find it rather difficult to understand, that is to say, their ability to grasp receives a setback. Historically speaking the task of sophisticated speech signal enhancement in the field of communication engineering is said to have commenced just after the invention of telephone by Alexander Graham Bell in the year 1850. However, in the initial stages, the speech signal transmission, processing and reception was analog in nature and used only wired communication with a restricted number of users only. The meaningful work was started in this field after establishment of Bell Telephone Laboratories at New Jersey, USA. Since then the evolving discrete time signal processing techniques along with the development in digital hardware and software technologies have helped the rapid growth of purely digital speech signal processing applications like speech coding, speech synthesis, speech recognition, speaker verification and identification and speech enhancement [1]. At present the wireless communications industry is heavily dependent upon advanced speech coding techniques, while the integration of computers and voice technology (speech recognition, synthesis etc.) are poised for growth. Both the speech coding and recognition require some speech enhancement strategy to be embedded into them. An attempt has been made here to explain the requirements and scope in the field of speech enhancement and its real time implementation.

1.1 Requirements of Speech Enhancement

In electronic communication systems speech signal is transmitted electrically; the conversion media (microphone, loudspeaker, headphones, earphones), as well as the transmission media (wired or wireless), typically introduce distortions, yielding a noisy and distorted speech signal. Such degradation can lower the intelligibility (the likelihood of being correctly understood) and/or quality (naturalness and freedom from distortion, as well as ease for listening) of speech. The speech enhancement techniques are required to improve the speech signal quality and intelligibility in many applications either for the human listener or to improve the speech signal so that it may be better exploited by other speech processing algorithms. For human listeners speech enhancement technique should aim at high quality as well as intelligibility, while quality is largely irrelevant if the enhanced speech serves as input to a recognizer [2-4]. For coders or recognizers, speech could actually be “enhanced” in such a way as to sound worse, as long as the analysis process eventually yields a high-quality output i.e., if the “enhanced” input allows more efficient parameter estimation in a coder or higher accuracy in a recognizer, it serves its overall purpose. For example, pre-emphasizing the speech (to balance relative amplitudes across frequency) in anticipation of broadband channel noise (which may distort many frequencies) does not enhance the speech as such, but allows easier noise removal later (via de-emphasize). The present day speech enhancement techniques improve speech quality without increasing intelligibility; in fact in some cases it reduces intelligibility. Aspects of quality are worthwhile general objective. However, when there are distortions in speech, it is usually considered more important to make it intelligible rather than merely make it more pleasing.

In recent years with the increasing use of wireless communication in cellular and mobile phones with or without ‘hands free’ system, voice over internet protocol (VOIP) phones, voice messaging service (voice mail), call service centers, cord less hearing aids etc. require efficient real time speech enhancement strategies to combat with additive noise and convolutive distortion (e.g., reverberation and echo) that generally occurs in any communication system [6]. The other areas of application areas include aircraft and military communication, aids for hearing impaired persons, communication inside vehicles and telephone booths, enhancing emergency calls and black box recordings etc. Besides, speech enhancement is also required as a pre-processing block in other speech processing systems like speech recognition, speaker recognition, speaker identification and speech coding [4]. The speech codecs used in 3G cellular mobile phones require speech enhancement in a post-processing stage [41]. Most speech enhancement algorithms are needed to detect intervals in a noisy signal where speech is absent in order to estimate aspects of the noise alone. This is done by using voice activity detector (VAD) and hence the voice activity detection is an integral part of most speech enhancement techniques. The performance of most speech enhancement algorithms is highly dependent on VAD [4]. Hence speech enhancement and detection must be treated simultaneously. The VAD also finds application in mobile phones to detect speech/silence to reduce power consumption during non speech periods.

1.2 Scope of Research in Speech Enhancement and Detection

Speech enhancement algorithms have been made applicable to problems as diverse as background noise removal, cancellation of reverberation and multi-speech separation (speaker separation) in modern speech communication systems. This is outlined in figure 1.1. This figure depicts a speech signal being degraded by an additive background noise. In pursuance of this, a speech enhancement algorithm is used to restore the quality of the speech, before finally being presented to the listener. The listener here is taken to be either a human or a machine [7-8]. Besides, other sources of degradation also exist for speech signals, such as distortion from the microphone or reverberation from the surrounding environment. The approach to speech enhancement varies considerably depending upon the type of degradation.

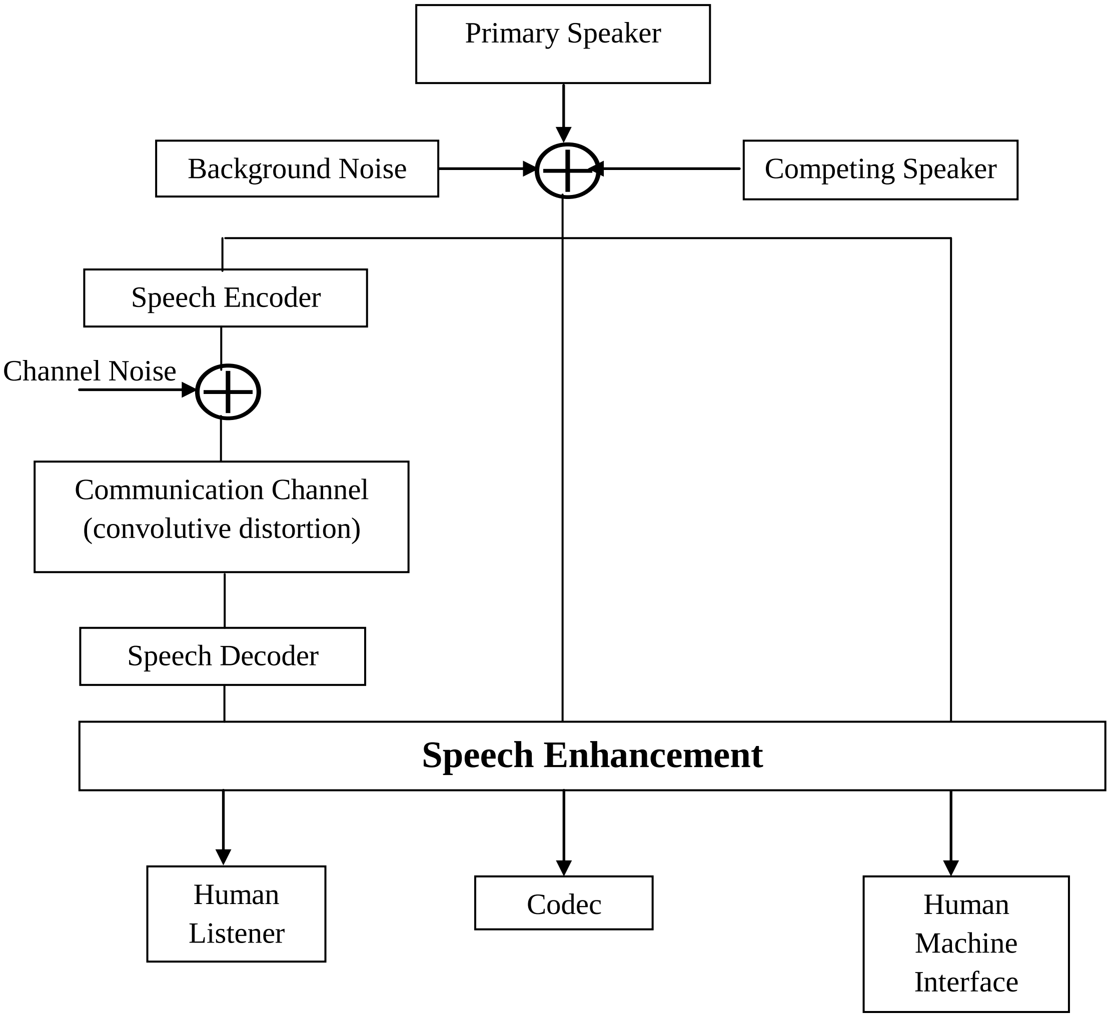


Fig. 1.1 Requirements of speech enhancement methods

The speech enhancement techniques can be classified into two basic categories: (i) Single channel and (ii) Multiple channels (array processing) based on speech acquired from single microphone or multiple microphone sources respectively [3]. However, single channel (one microphone) signal is available for measurement or pick up in real environments and hence the focus is here on single channel speech enhancement methods. That apart, the methods must also have other characteristics like real-time implementation, reasonable computational complexity while processing, low level of speech distortion, operation with low level SNR, separation as cleaned speech signal, adaptation to background noise, controlled level of noise suppression in speech, possibility of using a graphic equalizer for removing the stationary hindrances and easy integration with target applications etc. etc.

The pioneer work in the field has been done by Lim and Oppenheim [9] in 1979. Since then several methods have been evolved in the literature for single channel speech enhancement during last thirty years. The major contributors in this area are Boll and Berouti, (1979), Ephraim and Malah (1984), Sclarat (1986), Virag (1999), Kamath (2002) and so on [10-18]. The approach to speech enhancement varies considerably depending upon type of degradation. Various domains of speech enhancement are discussed throughout the thesis. Most of the methods assume the noise to be stationary and VAD estimates the noise characteristics during speech pauses or silent period [19-20]. However, some researchers have proposed the method to handle non-stationary noise [21].The limitations of these methods still pose a considerable challenge to researchers in this area. The objectives of speech enhancement vary widely, namely; reduction in noise level, increased intelligibility, reduction of auditory fatigue, etc. For communication systems, two general objectives depend on the nature of the noise, and often on the signal-to-noise ratio (SNR) of the distorted speech. With medium-to-high SNR (e.g., > 5dB), reducing the noise level can produce a subjectively natural speech signal at a receiver (e.g., over a telephone line) or can obtain reliable transmission (e.g., in a tandem vocoder application). For low SNR (e.g., ≤5dB), the objective could be to decrease the noise level, while retaining or increasing the intelligibility and reducing the fatigue caused by heavy noise (e.g., train or street noise). [1]

[1] A paper entitled “Requirements and Scope of Speech Enhancement Techniques in Present Speech Communication Systems” is presented in National Technical Paper Contest-2010 (NTPC-2010) for seniors at IETE Vadodara centre, Vadodara in March 2010 and won 3rd prize.

In the present work, the goal is to design a single channel speech enhancement algorithm having good noise suppression characteristic in the low SNR range (0-5dB) for various noise characteristics.

1.3 Objectives of the Research Work

The research topic is motivated by the fact that the speech is the most important signal transmitted using communication system and it is always subjected to background and surrounding noise and distortion. Accordingly the performance of speech communication system is greatly improved if speech enhancement is embedded into this system and if it works in real time. Several strategies have been suggested in the past for that and still however some of the challenges have remained unsolved. Hence it is essential to develop new strategies for real time embedded speech enhancement and detection considering the communication application [37-39]. The use of technical computing development support tools such as MATLAB, SIMULINK and related Toolboxes [42-48] make simulation study as well design of graphical user interface more simple and refined.

The work described in the thesis includes; amongst others the following:-

- Literature survey for existing techniques and modifications suggested by various researchers in present application scenario.
- Simulation of transform domain techniques using MATLAB/SIMULINK.
- Objective and subjective evaluation of simulated techniques.
- Limitations of existing techniques considering communication applications and suggesting new strategies to overcome it.
- Simulation, performance evaluation and comparison of suggested strategy using MATLAB/SIMULINK.
- Real time and hardware implementation of existing and modified techniques using SIMULINK on PC and using SIMULINK/ RTW/ Embedded Target for TI C6000 toolboxes of MATLAB and CCS V3.3 on DSK 6713 from Spectrum Digital Corporation.
- Hardware profiling of techniques considering it as embedded real time application.

1.4 Organization of the Thesis

The thesis is organized in the form of ten chapters as follows:

Chapter: 1 **Introduction:** This chapter provides an overview and the context for the remainder of the thesis.

Chapter: 2 **Speech Enhancement Techniques: State of Art:** This chapter describes the survey of different speech enhancement techniques and existing algorithms which is the main part of the literature survey. The exhaustive search is done to find out the basic techniques and modifications suggested by various researchers. The classification is also presented in this chapter considering the type of degradation, processing domain and tools used. A case study of simulation and implementation work using Normalized Least Mean Square (NLMS) algorithm for noise and echo cancellation is described. MATLAB and SIMULINK are used for simulation and Real Time Workshop (RTW), Embedded Target for TI C6000 toolboxes from MATLAB, Code Composer Studio version 3.3 (CCS V3.3) and DSK 6713 hardware platform are used for implementation.

Chapter: 3 **Speech Enhancement and Detection Techniques: Transform Domain:** This chapter describes techniques for additive noise removal which are transform domain methods and based mostly on short time Fourier transform (STFT). The discrete Fourier transform is used as transformation tool in these techniques. These methods are based on the analysis-modify-synthesis approach. They use fixed analysis window length (usually 20-25ms) and frame based processing. They are based on the fact that human speech perception is not sensitive to spectral phase but the clean spectral amplitude must be properly extracted from the noisy speech to have acceptable quality speech at output and hence they are referred to as short time spectral amplitude or attenuation (STSA) based methods. The phase of noisy speech is preserved in the enhanced speech. The synthesis is mostly done using overlap-add method. They have been one of the well-known and well investigated techniques for additive noise reduction. Also they have less computation complexity and easy implementations. The detailed mathematical expression for the transfer gain function for each method is described along with the terms used in the function. The relative pros and cons of all available methods as well as applications are mentioned. However, they require the use of voice activity detector (VAD) and the performance depends on the accuracy of VAD. The magnitude spectral slope distance VAD is the simplest and reasonably accurate and its operation is described in this chapter. The other transformation tool used in speech enhancement is discrete wavelet transform (DWT) and the techniques based on DWT are also described in brief here. The performance evaluation of any algorithm is very important for comparisons. There are several objective measures are available to evaluate the speech enhancement algorithms. They are described in brief in this chapter.

Chapter: 4 **MATLAB Implementation and Performance Evaluation of Transform Domain Methods:** The simulation carried out to describe the functionality and behavior of STSA methods under various additive noise conditions is described in this chapter. The simulation work is concreted and converged by preparing a MATLAB GUI (Graphical User Interface). This GUI can be used to simulate any transform domain algorithm for different noise conditions. The IEEE standard database NOIZEUS (noisy corpus) is used to test algorithms. The database contains clean speech sample files as well as real world noisy speech files at different SNRs and noise conditions like airport, car, restaurant, train, station etc. The performance evaluation using different objective measures is also carried out and explained in this chapter. The GUI also includes evaluation of algorithms using objective measures. The limitations and present implementations of these methods are also mentioned.

Chapter: 5 **Relative Spectral Analysis-RASTA:** This chapter describes the Relative Spectral Analysis (RASTA) processing of speech which is originally proposed for automatic speech recognizers to work in reverberant environments. The original algorithm is modified later on for direct speech enhancement. In the present work this algorithm for speech enhancement is simulated and evaluated under different noise conditions. The original filter is redesigned to have better performance. The algorithm throws the challenge for real time implementation as it is non linear and non causal. However, it does not require the use of VAD and can be used to combat with additive and convolutive distortions.

Chapter: 6 **Hybrid Algorithm for Performance Improvement:** It is suggested here that for better performance the transform domain algorithm can be combined in some way with RASTA approach. The best performing transform domain algorithm is MMSE STSA85 (LSA) and it is combined with modified RASTA approach which is the modified and suggested approach for speech enhancement. This algorithm is also simulated and tested under different additive noise conditions using the NOIZEUS database and compared with the original algorithms. The results of performance evaluation using objective measures are described in this chapter. The comparison using alone objective measures is not sufficient as it will not ensure the quality of speech signal for human listeners and hence the subjective evaluation is also required to perform. The IEEE recommended and ITU-R BS.562-3 standard mean opinion score (MOS) listening test is carried out. The chapter describes the various guidelines followed to perform this test. The original and modified algorithms are compared based on this test and conclusion is made regarding quality of output of different algorithms. The algorithm is also tested under different reverberation condition using the Aachen impulse response (AIR) database developed by RWTH Aachen University, institute of communication systems and data processing (India). It is a set of impulse responses that were measured in a wide variety of rooms. This database allows realistic studies of signal processing algorithms in reverberant environments. The comments are made about performance of algorithms in the simulated reverberant conditions.

Chapter: 7 **Hardware Implementation Tools:** This chapter describes the suitable hardware implementation tools available for speech processing system. The SIMULINK, RTW, Target Support Package TC6 available with MATLAB package can be used for implementing algorithms on various DSP processors and microcontrollers. The 32 bit floating point TMS320C6713 DSP from Texas Instruments is suitable for embedding speech processing algorithms for real time applications. For rapid prototyping the DSP starter kit DSK 6713 is available from Spectrum Digital Incorporation. The code can be loaded on DSP using the compiler Code Composer Studio and DSK6713. All these tools together can be used to implement speech processing algorithm in real time. They are described in brief in this chapter.

Chapter:8 **Real Time and Embedded Implementation of Hybrid Algorithm:** The hybrid algorithm is first tried for real time implementation on PC. For dedicated hardware implementation the DSP implementation platform using TMS320C6713 processor from Texas Instruments is selected. Various profiling results are also obtained and compared in this chapter.

Chapter:9 **Conclusions and Future Scopes:** Final conclusions and future extension of the work and future scope in this field are elaborated in this chapter.

Chapter: 10 **References:** It contains Bibliography which includes the list of references used in each chapter.

Chapter 2

Speech Enhancement Techniques: State of Art

The speech signal degradations may be attributed to various factors; viz. disorders in production organs, different sensors (microphones) and their placement (hands free), acoustic non-speech and speech background, channel and reverberation effect and disorders in perception organs. Considerable research recently has examined ways to enhance speech, mostly related to speech distorted by background noise (occurring at the source or in transmission)-both wideband (and usually stationary) noise and (less often) narrowband noise, clicks, and other non-stationary interferences [1-7]. Most cases assume noise whose pertinent features change slowly (i.e., locally stationary over analysis frames of interest), so that it can be characterized in terms of mean and variance (i.e., second-order statistics), either during non-speech intervals (pauses) of the input signals or via a second microphone (called reference microphone) receiving little speech input [1].

In ideal scenario there should be no degradation in quality and/or intelligibility of original speech and/or human subjects have normal speech production and perception systems. In practical scenario there is degradation in quality and/or intelligibility and/or human subjects have impaired speech production and perception systems. So the goal of speech enhancement is to enhance quality and intelligibility. Except when inputs from multiple microphones are available (in some specially arranged cases), it has been very difficult for speech enhancement systems to improve intelligibility. Thus most speech enhancement methods raise quality, while minimizing any loss in intelligibility. As observed, certain aspects of speech are more perceptually important than others. The auditory system is more sensitive to the presence than absence of energy, and tends to ignore many aspects of phase. Thus speech enhancement algorithms often focus on accurate modeling of peaks in the speech amplitude spectrum, rather than on phase relationships or on energy at weaker frequencies. Voiced speech, with its high amplitude and concentration of energy at low frequency, is more perceptually important than unvoiced speech for preserving quality. Hence, speech enhancement usually emphasizes improving the periodic portions of speech. Good representation of spectral amplitudes at harmonic frequencies and especially in the first three formant regions is paramount for high speech quality. All enhancement algorithms introduce their own distortion and care to be taken to minimize distortion

Weaker, unvoiced energy is important for intelligibility, but obstruent are often the first to be lost in noise and the most difficult to recover. Some perceptual studies claim that such sounds are less important than strong voiced sounds (e.g., replacing the former by noise of corresponding levels causes little decrease in intelligibility). In general, however, for good intelligibility, sections of speech (both voiced and unvoiced) undergoing spectral transitions (which correspond to vocal tract movements) are very important. Speech enhancement often attempts to take advantage of knowledge beyond simple estimates of SNR in different frequency bands. Some systems combine speech enhancement and automatic speech recognition (ASR), and adapt the speech enhancement methods to the estimated phonetic segments produced by the ASR component. Since ASR of noisy speech is often less reliable, simpler ASR of broad phonetic classes is more robust, yet allows improved speech enhancement [2].

2.1 Interferences and Suppression Techniques

Different types of interference may need different suppression techniques. Noise may be continuous, impulsive, or periodic, and its amplitude may vary across frequency (occupying broad or narrow spectral ranges); e.g., background or transmission noise is often continuous and broadband (sometimes modeled as “white noise”- uncorrelated time samples, with a flat spectrum). Other distortions may be abrupt and strong, but of very brief duration (e.g., radio, static, fading). Hum noise from machinery or from AC power lines may be continuous, but present only at a few frequencies. These noises are generally additive in nature. Most speech enhancement techniques are devised to handle the additive background noise. Noise which is not additive (e.g., multiplicative or convolutional) can be handled by applying a logarithmic transformation to the noisy signal, either in the time domain (for multiplicative noise) or in the frequency domain (for convolution noise), which converts the distortion to an additive one (allowing basic speech enhancement methods to be applied). Varieties of techniques are devised to handle convolutive distortion and reverberation.

Interfering speakers present a different problem for speech enhancement. When people hear several sound sources, they can often direct their attention to one specific source and perceptually exclude others. This “cocktail party effect” is facilitated by the stereo reception via a listener’s two ears [3]. In binaural sound reception, the waves arriving at each ear are slightly different (e.g., in time delays and amplitudes); one can often localize the position of the source and attend to that source, suppressing perception of other sounds. How the brain suppresses such interference, however, is poorly understood. Monaural listening (e.g., via a telephone handset) has no directional cues, and the listener must rely on the desired sound source being stronger (or having major energy at different frequencies) than competing sources. When a desired source can be monitored by several microphones, techniques can exploit the distance between microphones [3]. However, most practical speech enhancement applications involve monaural listening, with input from one microphone. Directional and head-mounted noise-cancelling microphones can often minimize the effects of echo and background noise. The speech of interfering speakers occupies the same overall frequency range as that of a desired speaker, but such voiced speech usually has fundamental (pitch) frequency F0 and harmonics at different frequencies. Thus some speech enhancement methods attempt to identify the strong frequencies either of the desired speaker or of the unwanted source, and to separate their spectral components to the extent that the components do not overlap. Interfering music has properties similar to speech, allowing the possibility of its suppression via similar methods (except that some musical chords have more than one F0, thus spreading energy to more frequencies than speech does). The multi speech separation (speaker separation) requires multiple microphone solution. The single microphone techniques are not sufficient for this type of interference. Very little literature is available and still this problem is not exactly solved for any general case.

2.2 Recent Trends - Speech Enhancement Techniques

The approach to speech enhancement varies considerably depending upon type of degradation. The speech enhancement techniques can be divided into two basic categories: (i) Single channel and (ii) Multiple channels (array processing) based on speech acquired from single microphone or multiple microphone sources respectively [3]. However, single channel (one microphone) signal is available for measurement or pick up in real environments and hence focus is here on single channel speech enhancement methods. Figure 2.1 shows the chart of the latest single channel speech enhancement methods for three different kinds of problems.

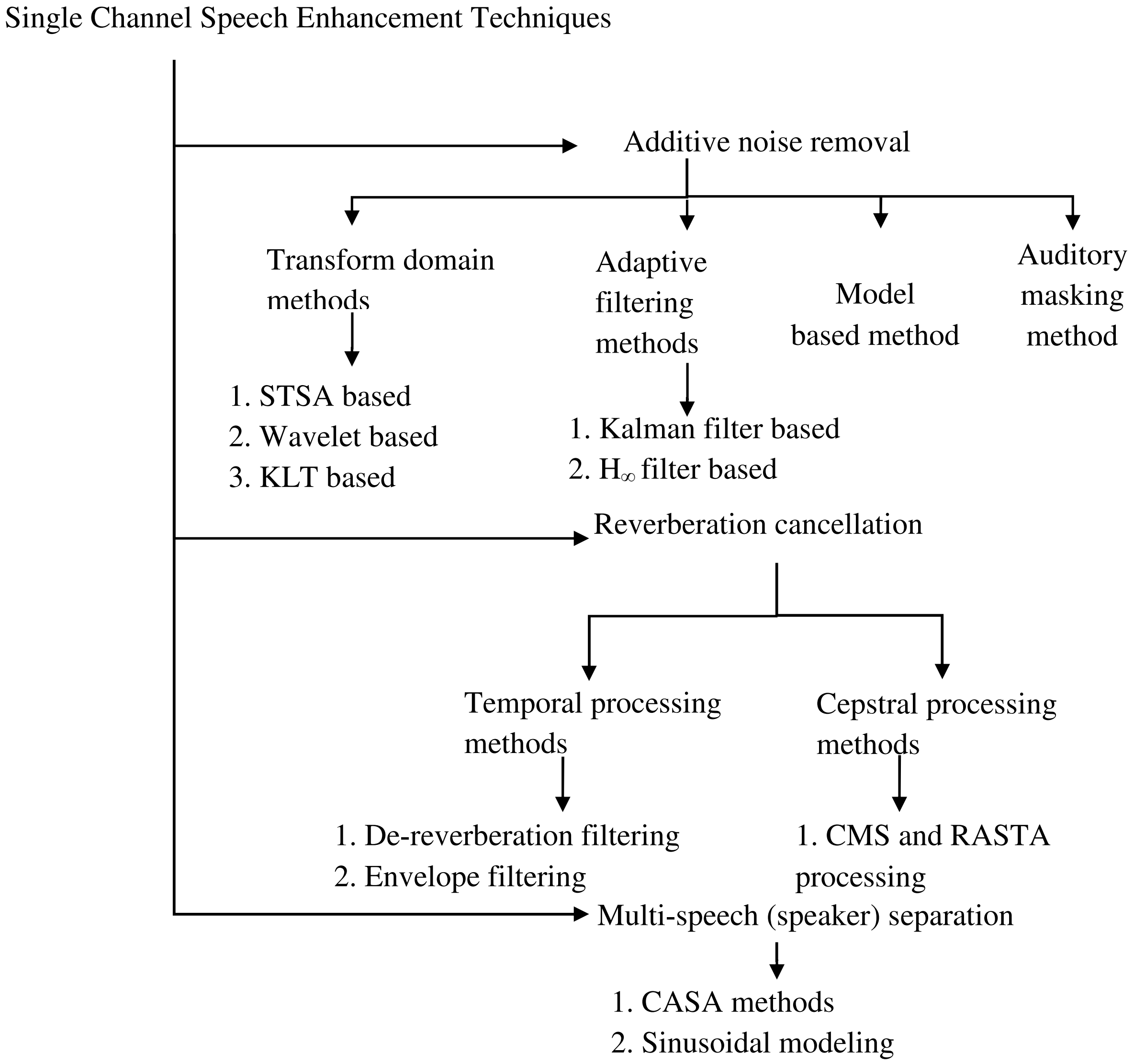

Single Channel Speech Enhancement Techniques
Additive noise removal
Transform domain methods
Adaptive filtering methods
Model based method
Auditory masking method
1. STSA based
2. Wavelet based
3. KLT based
1. Kalman filter based
2. H∞ filter based
Reverberation cancellation
Temporal processing methods
Cepstral processing methods
1. De-reverberation filtering
2. Envelope filtering
1. CMS and RASTA processing
Multi-speech (speaker) separation
1. CASA methods
2. Sinusoidal modeling

Fig. 2.1 A chart showing summary of existing speech enhancement methods

2.2.1 Additive Noise Removal

In most cases the background random noise is added with the desired speech signal and forms an additive mixture which is picked up by microphone. It can be stationary or non-stationary, white or colored and having no correlation with desired speech signal. Variety of methods suggested in literature so far to overcome this problem. The majority of them belong to following four categories.

2.2.1.1 Transform Domain Methods

The most commonly used methods are transform domain methods. They are most conventional methods. They transfer the time domain signal into other domain using different transforms and involve some kind of filtering to suppress noise and then inverse transform filtered signal into time domain. They follow the analysis-modify-synthesis approach. The transformation used is DFT, WT or KLT.

- **DFT based (STSA methods):** They are most popular as they have less computational complexity and easy implementation. They use short time DFT (STDFT) and have been intensively investigated; also known as spectral processing methods. They are based on the fact that human speech perception is not sensitive to spectral phase but the clean spectral amplitude must be properly extracted from the noisy speech to have acceptable quality speech at output and hence they are called short time spectral amplitude (STSA) based methods [5,7]. In practice power density of signal is used instead of amplitude. Methods of this category remove an estimate of noise from noisy signal using spectral subtraction (SS). The noise power spectrum estimation is obtained by averaging over multiple frames of a known noise segment; which can be detected using voice-activity detector (VAD) [4]. However the basic SS method suppresses noise but it has limitation in terms of an artefact called *musicality* [2]. This gives rise to distortion in enhanced speech. Several modifications in basic method are suggested by Boll and Berouti *et al.* [4] to reduce the musical noise. However this requires very careful parameter selections. The other modification in basic SS is using McAuly's maximum likelihood (ML) estimation [4] of output speech; which assumes noise with complex Gaussian distribution. In general all SS methods estimate *a posteriori* SNR. Also SS methods are suitable for stationary white noise only. The solutions to this are suggested using smoothing time varying filter called Wiener filter [4]. The combination of SS and Wiener filter is used in most real applications.

  The optimal Wiener filter for the noisy speech can be designed in frequency domain via the estimated ratio of the power spectrum of clean speech; called object power spectrum to that of noisy speech (*a priori* SNR). This spectrally varying attenuation accommodates coloured noise, and can be updated at any desired frame rate to handle non-stationary noise. A major problem with this approach is estimating background noise spectrum at every frame which is limited by the performance of VAD. This requires noise adaptation [4] in VAD for every frame. However estimation of object power spectrum considering current frame only is non-realistic as well as time varying and non-stationary process. A solution to this is suggested by Ephrahim and Malah [4] known as decision direct (DD) method which estimates *a priori* SNR of current frame using *a posteriori* SNR of current frame, estimated noise for current frame and estimated clean speech in previous frame. So in practice a Wiener filter is combined with DD approach to give realistic system. The Wiener filter shows substantial reduction in musical noise artefacts compared to SS methods.

The realistic and optimal object power spectrum estimation without artefacts requires model based statistical methods. The stochastic estimation methods such as minimum mean square error (MMSE) and its variant MMSE log spectral amplitude (LSA) suggested by Ephrahim and Malah [4] are commonly used estimation methods. They are based on modelling spectral components of speech and noise processes as independent Gaussian variables. Almost all literature mentions that the performance of Wiener filter and MMSE LSA is outstanding in terms of both subjective and objective evaluations. The stochastic estimation method called MAP (maximum *a posteriori*) is very close in performance with MMSE LSA with simpler computations. All of these methods assume speech presence in the frequency bin under consideration; but it is not always true. These methods can be extended by incorporating a two state speech presence/absence model which leads to a soft decision based spectral estimation and further improves performance at the cost of computational complexity. Further improvements were observed by using Laplacian model for speech spectral coefficients rather than Gaussian model. The various kinds of noise adaptation strategies used like hard/soft/mixed decision also affect the performance. The soft decision based noise adaptation found satisfactory in removing musical artefact but at the cost of increased processing requirements.

A background noise suppression system developed by Motorola is included as a feature in IS-127, the TIA/EIA standard for the Enhanced Variable Rate Codec (EVRC) to be used in CDMA based telephone systems [8]. EVRC was modified to EVRC-B and later on replaced by Selectable Mode Vocoder (SMV) which retained the speech quality at the same time improved network capacity. Recently, however, SMV itself has been replaced by the new CDMA2000 4GV codecs. 4GV is the next generation 3GPP2 standards-based EVRC-B codec [9]. The EVRC based codec uses combination of STSA based approaches: multiband spectral subtraction (MBSS) and minimum mean square error (MMSE) gain function estimator for background noise suppression as a pre-processor. The voice activity detector (VAD) used to decide speech/silence frame is embedded within the algorithm. Its quality has been proven good through commercial products. Nevertheless, the quality may not be sufficiently good for a wide range of SNRs, which were not given much attention when it was standardized. Another algorithm suggested by A.Sugiyama, M.Kato and M. Serizawa [3] uses modified MMSE-STSA approach based on weighted noise estimation. The subjective tests on this algorithm claim to give maximum difference in mean opinion score (MOS) of 0.35 to 0.40 compared to EVRC and hence its later version is equipped within 3G handsets. The modified STSA-MMSE algorithm based on weighted noise estimation is employed in millions of 3G handsets as the one and only commercially available 3GPP-endorsed noise suppressor [3]. But still there are open questions like how the parameters of statistical models can be estimated in a robust fashion and what can be meaningful optimization criteria for speech enhancement; which will require further research.

- **Wavelet based:** The DFT based methods use short time spectral measurements and hence are suffered by time-frequency resolution trade-offs. Wavelet based methods are developed which provides more flexibility in time-frequency representation of speech. The Wavelet de-noising algorithm is most commonly used and based on soft thresholding [7, 10] of the Wavelet coefficients. However uniform thresholding results in suppression of noise as well as unvoiced components of desired speech. So, Wavelet transform combined with smoothing filter like Wiener filter in Wavelet domain is suggested. Presently, a method is suggested in which the soft thresholding decision is taken based on statistical models. Unfortunately Wavelet based techniques are failed to achieve the great success and popularity in speech enhancement. The STFT and Wavelet based techniques are described in next chapter and simulation is presented in chapter 4.
- **KLT based:** The frequency domain methods are nowhere close to offering fully satisfactory solutions to their inherent problems: the musical noise artefact and the inevitable trade-off between signal distortion and the level of residual noise. The signal subspace approach (SSA) for speech enhancement has been originally introduced by Dendrinos *et al.* operates in eigen domain [11]. It uses the singular value decomposition (SVD) of a data matrix to remove the noise subspace and then reconstruct the desired speech signal from the remaining subspace.

This approach is modified by Ephrahim and Van Trees and proposes the use of Eigen value decomposition (EVD) of covariance matrix of input signal vector. This method consists in estimating a transform, namely the *Karhunen-Loeve transform (KLT)* [12], which will project the input signal vector into a subspace called the signal subspace hence readily eliminating the components in the orthogonal noise only subspace. The enhanced signal is reconstructed in time domain using inverse KLT. The SSA was found to outperform frequency domain methods but yet not received much attention and its use in practice is still scarce due to high computational load. However, with the sharp computation hardware available today this method can become a serious candidate to compete with the currently employed noise reduction methods.

2.2.1.2 Adaptive Filtering Methods

The adaptive filters which are mostly used in adaptive control applications can also be useful for speech enhancement. Mostly LMS and its variants are useful in multi microphone additive noise and echo cancellation problems. But for single channel speech enhancement Kalman and H_∞ adaptive filters are found suitable. They can also address the problem of colored noise removal as the noise is not always white is real environments. The transform domain methods degrade in such situations.

- **Kalman filter based**: In Wiener filtering approach the analysis has shown that the amount of noise attenuation is in general proportional to the amount of speech degradation. Kalman filtering [13] provides optimal time domain estimations and can be used instead of Wiener filtering at the cost of computational complexity and complicated implementation hardware. Literature suggests a large number of variants of basic Kalman filtering algorithm used in speech enhancement. It can be integrated with autoregressive (AR) speech models; but still the robust estimation of model parameters requires further research.
- **Robust-H_∞ filter based**: Recently H_∞ filtering [14] has been shown to overcome unrealistic assumptions of Wiener and Kalman filtering methods. Furthermore, both Wiener and Kalman estimators may not be sufficiently robust to the signal model errors. The estimation criterion in the H_∞ filter design is to minimize the worst possible effects of the modeling errors and additive noise on the signal estimation errors. Since the noise added to speech is not Gaussian in general, this filtering approach appears highly robust and more appropriate in practical speech enhancement. Furthermore, the H_∞ filtering algorithm is straightforward to

implement. Still this algorithm has not got enough attention in implementation for speech enhancement.

As a preliminary work the simulation and implementation of adaptive noise and echo cancellation is described in section 2.4.

2.2.1.3 Model Based Methods

The third method adopts a specific speech production model (e.g., from low-rate coding), and reconstructs a clean speech signal based on the model, using parameter estimates from the noisy speech [1, 7]. This method improves speech signals by parametric estimation and speech re-synthesis. Speech synthesizers generate noise-free speech from parametric representations of either a vocal tract model or previously analyzed speech. Most synthesizers employ separate representations for vocal tract shape and excitation information, coding the former with about 10 spectral parameters (modeling the equivalent of formant frequencies and bandwidths) and coding the latter with estimates of intensity and periodicity (e.g., F0). Standards methods (e.g., LPC) do not replicate the spectral envelope precisely, but usually preserve enough information to yield good output speech. Such synthesis suffers from the same mechanical quality as found in low-rate speech coding and from degraded parameter estimation (due to the noise), but can be free of direct noise interference, if the parameters model the original speech accurately. In general, re-synthesis is the least common of the speech enhancement techniques, due to the difficulty of estimating model parameters from distorted speech and due to the inherent flaws in most speech models. It nonetheless has application in certain cases like improving the speech of some handicapped speakers.

2.2.1.4 Auditory Masking Methods

Several perceptual based approaches are also investigated, where unwanted component of signal is masked by the presence of another component and taking advantage of simultaneous masking property of human auditory system. Instead of removing all noise from signal these methods attempt to attenuate the noise below the audible threshold. Virag [15] proposed the noise reduction algorithm based on this principle and shown that the auditory masking algorithm outperforms other noise suppression algorithms with respect to human perception; the algorithm was judged to reduce musical artifacts and give acceptable speech distortion. However, the disadvantage is the large computational load due to sub-band decomposition and additional DFT analyzer required for psychoacoustic modeling. The RelAtive SpecTral Amplitude processing

(RASTA) algorithm proposed by Hermnsky and Morgan [19] to enhance speech for automatic speech recognition in reverberant environment. This algorithm was later modified for additive noise removal. This algorithm is required further investigations and can be tested for real-time implementation. The RASTA algorithm and its simulation are described in chapter 5.

2.2.2 Reverberation Cancellation

The reverberation is a convolutive distortion that occurs to the speech while it is picked up by microphone. The speech signal is convolved with ambient or channel impulse response. The objective here is to recover the original speech without *a priori* information of channel or environment through which speech is collected or recorded. The acoustic echo can also be considered as one kind of reverberation effect. The *blind deconvolution* is the obvious remedy to the reverberation and acoustic echo which involves some kind of inverse filtering and equalization operation. They basically classified in two categories; however multistage algorithms [16, 17] which use combination of these methods are also proposed in literature.

2.2.2.1 Temporal Processing Methods

The temporal processing methods obtain the enhancement by processing the reverberant speech in time domain.

- **De-reverberation filtering**: Here the signal is passed through a filter having impulse response that is inverse of reverberation process. A blind estimation of filter is always difficult. Douglas *et al.* and Yagnanarayan *et al.* proposed inverse filter estimation based on LP residual and Gillespe *et al.* proposed same based on correlation shaping [7]. These methods were partially successful but failed in environment with long reverberation time because of assumption made about LP residue of speech signal that it is independent and identically distributed. A more robust filter can be obtained from the harmonic structure of reverberant speech signal called *harmonicity based de-reverberation filter (HERB*) [3]. It estimates the inverse or de-reverberation filter as the time average of a filter that transforms observed reverberant signals into the output of an adaptive harmonic filter. This achieves high quality de-reverberation, provided a sufficient number of observed signals (training data) are available. Several modifications still require making it useful in real practice like reduction in training data size, enhanced approximation to speech harmonicity etc. Further research in this direction is required.

- **Envelope filtering**: This method does not require obtaining impulse response of an environment. It is based on *modulation transfer function (MTF)* of speech [2]. It assumes that the temporal envelope of surrounding environment impulse response decays exponentially with time and the carrier signals of the impulse response and a speech signal can be modelled as mutually independent white noise functions. However, these assumptions are not accurate with regard to real speech and reverberation. So, this approach yet not achieved high quality de-reverberation.

2.2.2.2 Cepstral Processing Methods

The cepstral processing methods process the speech signal in cepstral domain. The homomorphic signal processing and cepstral mean subtraction (CMS) method proposed by Oppenheim et al.[2] achieved de-reverberation by removing cepstral components corresponding to the impulse response by applying low time lifter in cepstral domain. Also as an alternate the cepstral filtering can be done using a comb filter. It is successful for cancellation of simple echoes but have a limited performance for real environments. A generalization of CMS is RelAtive SpecTral Amplitude processing (RASTA) algorithm [19]. It uses a cepstral lifter to remove high and low modulation frequencies and not simply the DC component, as does CMS. It is also motivated by certain auditory principle that auditory system is particularly sensitive to signal change. Still there is a scope of research in proper implementation of this algorithm for speech enhancement. It can also be used to remove additive noise. The RASTA algorithm is described in detail in chapter 5.

2.2.3 Multi-speech (speaker) Separation

Here a low-level speaker may be sought in presence of a loud interfering speaker and the signal picked by single microphone containing additive mixture of both signals. Here speech of other speakers is degradation and speech of desired speaker to be enhanced. The problem of multi-speaker separation is the most difficult to handle and still the research done is limited in the context of problem solution [7, 18]. There are certain problems faced here like difficulty due to spectral similarity, pitch of different speakers may cross or overlap, number of talkers is not known, talker amplitude varies in an utterance etc. Very few approaches are proposed in literature for single microphone solution to this problem.

2.2.3.1 CASA Method

One approach called computational auditory scene analysis (CASA) which replicates the perceptual processes by which human listener segregate simultaneous sounds. It involves segregating speech of desired speaker in the presence of degradation, treat speech of desired speaker as stream of segments, approach to localize and select these streams and stitch them together in sequence to obtain speech of desired speaker. Most works had been carried out recently and it suffers from two deficiencies: First, it is not able to separate unvoiced segments and second, the vocal-tract related filter characteristics are not given importance compared to excitation signal. Also, evaluation is an important issue for CASA that requires further thought. However, it is still under research and adherence to the general principles of auditory processing is likely to give rise to CASA systems that make fewer assumptions and it will turn into superior performance in real acoustic environments [3, 18].

2.2.3.2 Sinusoidal Modeling

Another approach to the problem is to use sinusoidal modeling [2] of speech. Here the speech signal generated by two different simultaneous talkers can be represented by a sum of two sets of sine waves, each with time-varying amplitudes, frequencies and phases. The algorithm separates amplitudes, frequencies and phases for each speaker and re-synthesizes the signal for each speaker. Separation of the spectra of each speaker is done with the help of her/his pitch estimation. The performance depends on how best pitch of each speaker can be estimated and joint pitch estimation is the most difficult task in multi-speaker case.

The single channel techniques are not having enough power to solve this problem. However, the multi-microphone techniques like beam forming and blind source separation are far more superior and suitable for this problem. They exploit spatial information and additional reference for processing. This problem is ruled out here in the context of single channel solution.[1]

[1] A paper entitled “A Review on Single Channel Speech Enhancement Techniques for Wireless Communication Systems” is presented in National conference on Information Sciences (NCIS-2010) organized by MCIS, Manipal University, Manipal in April 2010.

2.3 A Case Study of Speech Enhancement Technique Using Adaptive Filtering Algorithms:

The initial preliminary research work carried out has focused on single channel speech enhancement techniques where no reference signal for noise is available. However, as a preliminary starting work the two microphone enhancement technique using adaptive algorithm called adaptive noise cancellation (ANC) is taken as a case study. It is simulated and implemented for additive noise reduction and echo cancellation purposes. When more than one microphone is available to furnish pertinent signals, speech degraded by many types of noise can be handled. Processed version of a second “reference” signal $u(n)$ (containing mostly or exclusively interference noise) is directly subtracted in time from the primary noisy speech signal $y(n)$. The block diagram is shown in figure 2.2.

While other speech enhancement filtering methods get good results with a dynamic filter that adapts over time to estimated changes in the distortion, such adaptation is essential in ANC. Since there will be a delay between the times the interference reaches different microphones and since the microphones may pick up different versions of the noise (e.g., the noise at the primary microphone may be subject to echoes and/or spectrally variable attenuation), a secondary signal must be filtered so that it closely resembles the noise present in the primary signal. In most adaptive system, the digital filter used is FIR because of simplicity and guaranteed stability. There are several ways to obtain the filter coefficients, of which the most attractive is the least-mean-squares (LMS) method via steepest descent [20], due to its simplicity and accuracy. More computationally expensive exact least-squares (LS) methods typically yield only marginal gains over the faster stochastic-gradient LMS method; the latter is also useful for enhancement of one-microphone speech degraded by additive noise [1].

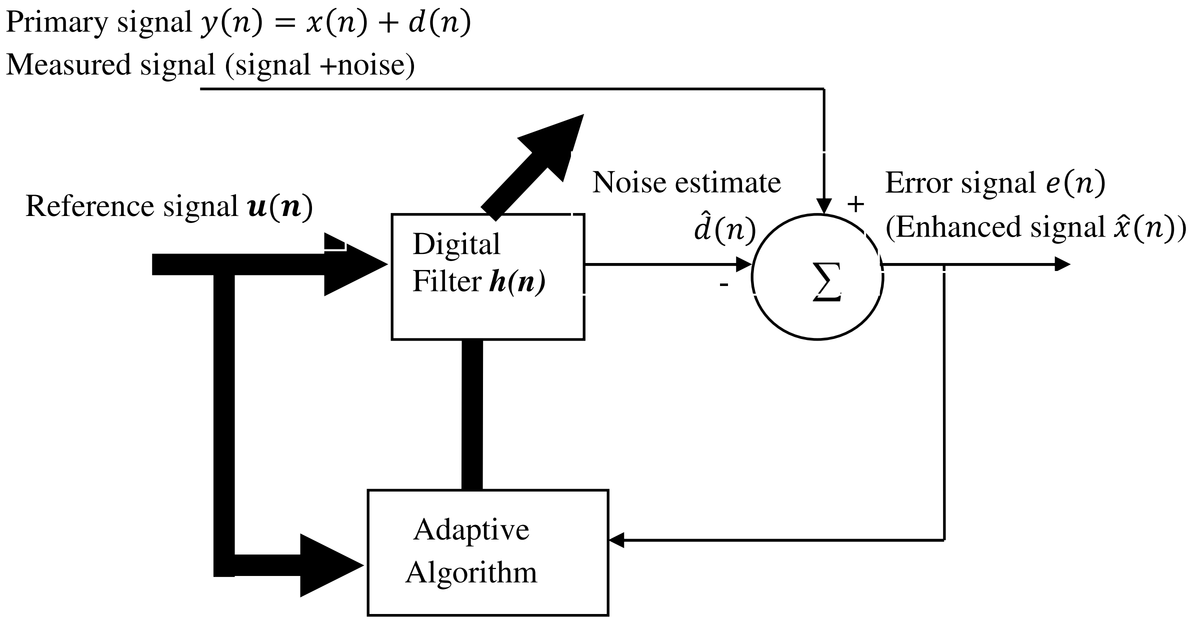

Primary signal $y(n) = x(n) + d(n)$
Measured signal (signal +noise)
Reference signal $\boldsymbol{u(n)}$
Digital Filter $\boldsymbol{h(n)}$
Noise estimate
$\hat{d}(n)$
+
-
Σ
Error signal $e(n)$
(Enhanced signal $\hat{x}(n)$)
Adaptive Algorithm

Fig. 2.2 Block diagram of adaptive noise cancellation (ANC) system

2.3.1 ANC Using NLMS Algorithm

Filter coefficients are chosen so that the energy in the difference or residual error signal $e(n)$ (i.e., the primary signal $y(n)$ minus a filtered version $\hat{d}(n)$ of the reference$u(n)$) is minimized. Thus one can select FIR filter weight co-efficients (or impulse response) $h(k)$ so that the energy in

$$e(n) = y(n) - \hat{d}(n) = y(n) - \sum_{k=1}^{L} h(k)u(n-k) \tag{2.1}$$

is minimized. Here $y(n)$ is a signal with noise to be processed and $\hat{d}(n)$ is a filtered version of reference signal $u(n)$. As long as the two microphone signals ($u(n)$ *and* $y(n)$) are uncorrelated, minimizing $e^2(n)$ (a "least mean squares" approach) over time should yield a filter that models the transformed reference, which can thus be subtracted from $y(n)$ to provide enhanced speech, which is actually the minimized residual $e(n)$. This provides the signal estimate or enhanced signal $\hat{x}(n)$. Correlation between $u(n)$*and* $y(n)$ is undesirable because then the $h(k)$ values are affected by speech and $\hat{d}(n)$ will partly contain speech rather than only transformed noise, and part of the desired speech will be suppressed. Solving Equation (2.1) can exploit LS or LPC methods, or simpler LMS techniques which do not require calculating correlation matrices or inverting them. The LMS approach uses steepest-gradient iteration [20] to get

$$h_i(n+1) = h_i(n) + \mu e(n)u(n-i) \tag{2.2}$$

Where μ is scalar parameter (0< μ < ${}^{1}/_{MS_{max}}$). Here M is the tap size of the filter and S_{max} is power spectral density of reference input $u(n)$. It is called adaptation step size. A large value for μ speeds up convergence, but may lead to stability problems. A modified version with better stability is often used, called normalized LMS (NLMS) [20]:

$$h_i(n+1) = h_i(n) + \frac{\tilde{\mu}e(n)u(n-i)}{(\sum_{k=0}^{N-1} u^2(n-k))} = h_i(n) + \mu(n)e(n)u(n-i) \tag{2.3}$$

with control factor (step size) 0 <$\tilde{\mu}$ < $2\frac{E[|u(n)|^2]D(n)}{E[|e(n)|^2]}$, where $E[|e(n)|^2]$ = error signal power, $E[|u(n)|^2]$ = input signal power and $D(n)$ = mean square deviation of filter weight co-efficients. It can be briefly described as follows:

- Initialization: If prior knowledge of the tap weight vector $\boldsymbol{h(n)}$ is available, use it to select an appropriate value for $\boldsymbol{h(0)}$. Otherwise, set $\boldsymbol{h(n) = 0}$.
- Data:
  - Given $\boldsymbol{u(n)}$ = M by 1 tap input vector at time *n* = $[u(n), u(n-1), \dots \dots, u(n-M-1)]^T$, $y(n)$ = noisy speech signal at time *n*.
  - To be computed: $\boldsymbol{h(n)}$ =estimate of tap-weight vector at time *n*
- Computation: $\hat{d}(n) = \boldsymbol{h(n)}^T\, \boldsymbol{u(n)},$

  $e(n) = y(n) - \hat{d}(n)$

  $\boldsymbol{h(n+1)} = \boldsymbol{h(n)} + \tilde{\mu}. e(n). \boldsymbol{u(n)}/||\boldsymbol{u(n)}||^2.$

Figure 2.3 shows the flow chart to implement the algorithm.

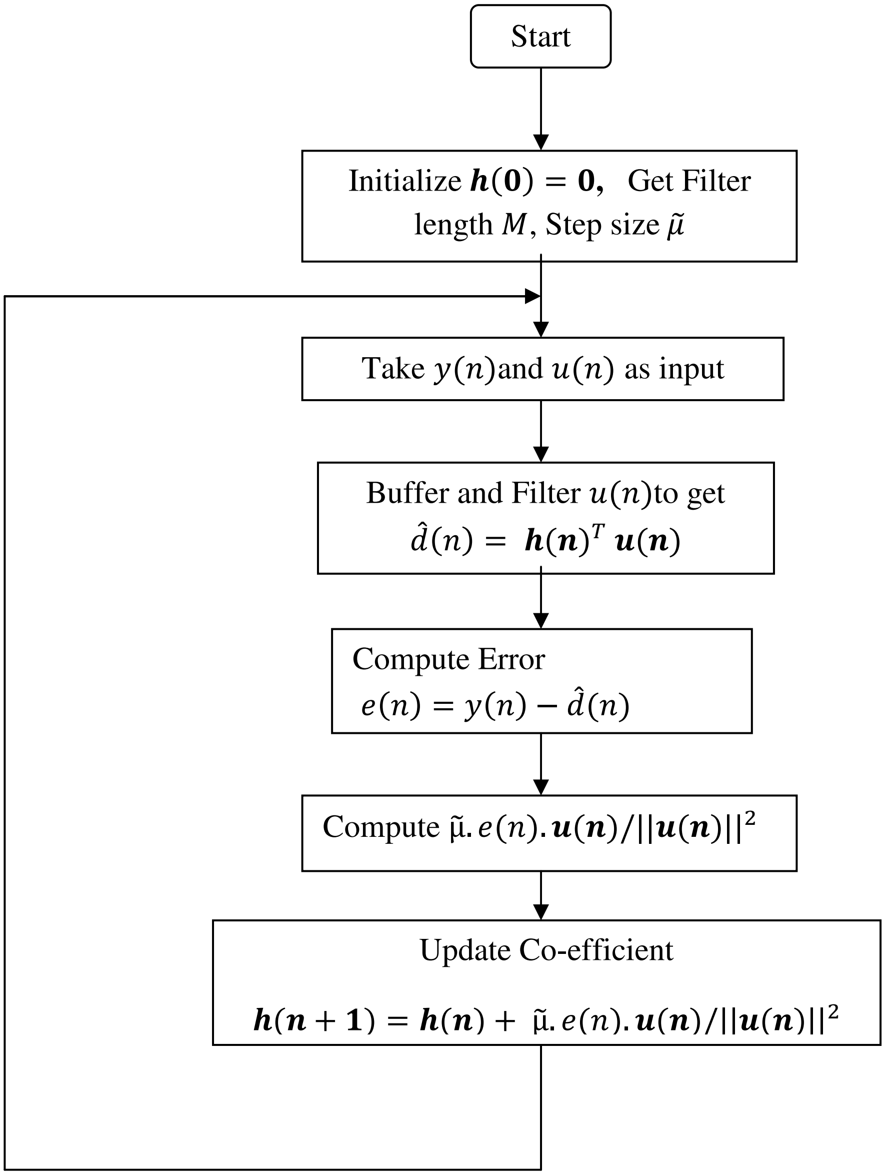
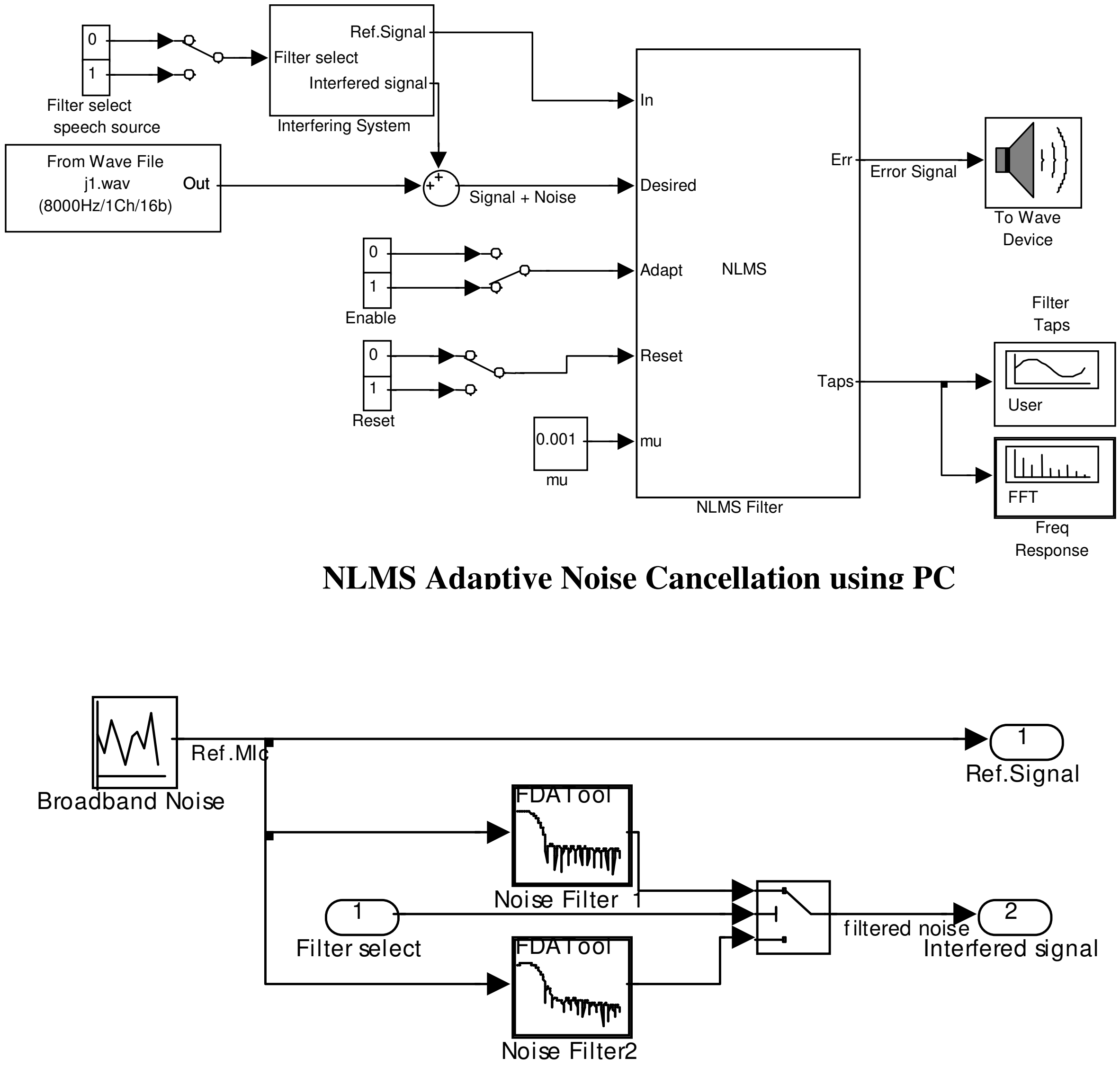

Start
Initialize $\boldsymbol{h}(\boldsymbol{0}) = \boldsymbol{0}$, Get Filter length M, Step size $\tilde{\mu}$
Take $y(n)$and $u(n)$ as input
Buffer and Filter $u(n)$to get $\hat{d}(n) = \boldsymbol{h}(\boldsymbol{n})^T \boldsymbol{u}(\boldsymbol{n})$
Compute Error $e(n) = y(n) - \hat{d}(n)$
Compute $\tilde{\mu}. e(n). \boldsymbol{u}(\boldsymbol{n})/||\boldsymbol{u}(\boldsymbol{n})||^2$
Update Co-efficient
$\boldsymbol{h}(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{h}(\boldsymbol{n}) + \tilde{\mu}. e(n). \boldsymbol{u}(\boldsymbol{n})/||\boldsymbol{u}(\boldsymbol{n})||^2$

Fig. 2.3 Flow chart for implementation of ANC using NLMS algorithm

2.3.2 Practical Implementation of ANC with NLMS Algorithm

The NLMS algorithm is first implemented in SIMULINK. The SIMULINK model is prepared using the techniques like masking, subsystems, conditional subsystems, and in-built S functions etc. Here the *.wav file is used as a signal source. It is added with filtered random white noise. The parameters of filter can be selected to any suitable value using FDA Toolbox. Also the noise characteristics can be varied by selecting appropriate parameters for the Noise block in the model. The noise can be either filtered by low pass filter or by band pass filter. The selection switch is provided in the model. The NLMS block accepts one input (on “In port”) directly from reference source as white noise. The other input (on “Desired port”) from added mixture of signal and filtered noise. The step size parameter (µ) can be set to any value from input port labeled (mu). Also, it has two control inputs, one to enable adaption and other to reset the filter weights to zero at any time and then allowing them to readapt. The output signal can be obtained from output port labeled “Error”, which is actually clean output signal. This port is connected to speaker or headphone through PC sound card using the block “To Wave Device”. To record the clean signal replace this block by “To Wave File” block and give the name of file in the parameter dialog box of that block. Also, the output port labeled “weights” can be used to see the updating of filter coefficients and variable frequency response by connecting suitable blocks at that port.



NLMS Adaptive Noise Cancellation using PC

Interfering System

Fig. 2.4 SIMULINK implementation of ANC using NLMS algorithm

Also, the same NLMS algorithm is implemented for real time application using the Real Time Workshop and Embedded Target for TI C600 Toolboxes. The hardware setup is shown in figure 2.5. Here the SIMULINK model is developed for C6713 DSP. Here the control signals like adaption enable, reset and noise filter selection is done by using switches on the DSK. The speech source signal can be applied to “Line In” or “Mic In” source of DSK depending on selected option in the block of ADC in model file. The noise source is again simulated here from the SIMULINK block using the same techniques as described above. Connecting speaker or headphone at “Line Out” or “HP Out” port of DSK can obtain the output signal. To operate this model from beginning, it is required to follow the following procedure.

1. Connect a speech source to the 'line in' or ‘mic in’ jack of the DSK board.
2. Set the required parameters by choosing Simulation -> Configuration Parameters.
3. To generate code choose Tools -> Real-Time Workshop ->Build Model (or Ctrl-B).
4. After generating code, Real-Time Workshop connects to Code Composer Studio (CCS) and creates a new project. After compiling and linking the code, Real-Time Workshop downloads the COFF (Common Object File Format) file to the DSK and begins execution. At this time, if speakers (or headphone) are connected to the audio output jack of the DSK, one could hear the noisy signal.
6. Now, the system is ready to begin the adaptation algorithm. By Pressing down the user DIP switch (SW0) on the DSK, initiate the algorithm. One could hear the noise component of the signal slowly decrease in volume as the filter adapts.
7. To control the adaptive filter during execution, move the User DIP Switches as follows:

   Switch 0:

   ‘Off’ — pause adaptation process, ‘On’ — start/resume NLMS adaptation process.

   Switch 1:

   ‘Off’ — disable reset, ‘On’ — reset LMS adaptation process.

   Switch 2:

   ‘Off’ — apply band pass noise model, ‘On’ — apply low-pass noise model.

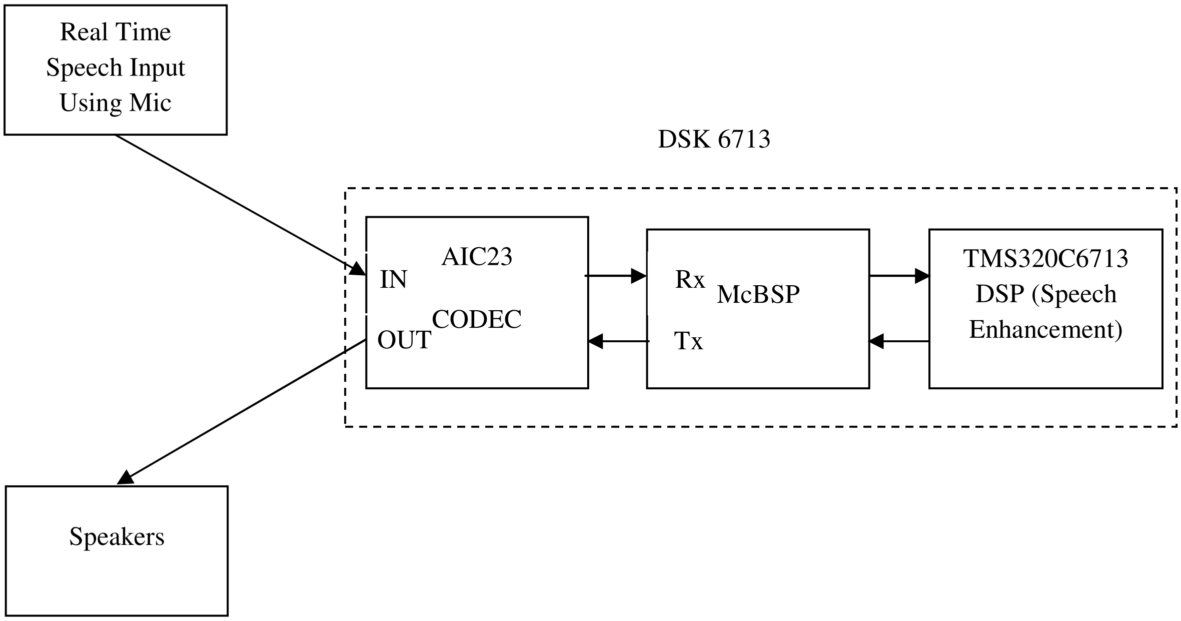
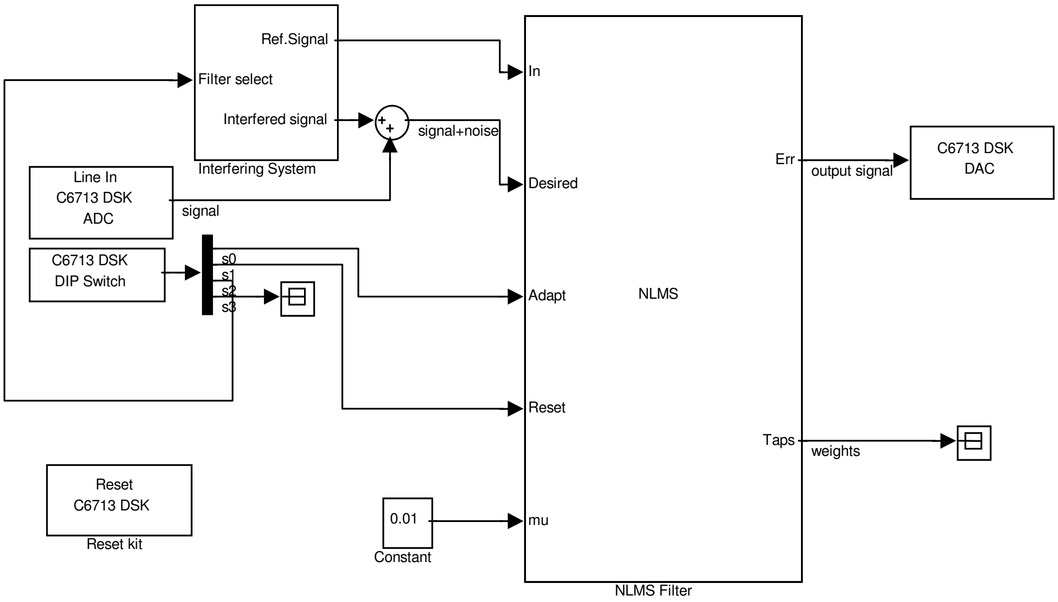


Fig. 2.5 Hardware setup for implementation on DSK 6713



Adaptive Noise Cancellation using NLMS & DSK6713

Fig. 2.6 DSK 6713 implementation of ANC using NLMS algorithm

2.3.3 Performance of NLMS algorithm for ANC

In order to design and implement any adaptive filter for a given application, it is required to determine the values of parameters such as the step size $\tilde{\mu}$, the filter length M, and the initial coefficient weight vector $\boldsymbol{h}(\mathbf{0})$. To properly select these parameters, it is required to understand important properties of adaptive algorithms [21] as summarize here.

1. Stability conditions

As seen in above sections, the adaptive filter uses FIR filter, which is inherently stable. However, the whole adaptive filter is not always stable. The stability depends on the algorithm that adjusts its coefficients. Different analysis and criteria shows that the step size $\tilde{\mu}$ must be within some range to satisfy the stability condition. In most practical cases for NLMS algorithm it should be between 0 and 1 according to optimization criterion. According to stability criterion, it should be between 0 and 2/M. The stability improves with the lower value of step size $\tilde{\mu}$, but it requires larger filter length M.

2. Convergence rate

In applications with slowly changing signal statistics, the performance function drifts in time. Adaptation is the process of tracking the signals and environments. Thus, speed of convergence is the most important considerations. Algorithm convergence is attained when the Mean Square Error (MSE) is reduced to minimum value. The analysis has shown that the average time needed for the algorithm to converge is inversely proportional to the step size $\tilde{\mu}$. But it is not recommended to use arbitrary large step sizes to speed up convergence because of the stability constraint.

3. Steady state performance

With a true gradient and under noise-free conditions, the adaptive algorithm converges to the minimum MSE and remains there because the gradient is zero at the optimum solution. But actually the NLMS algorithm not uses the true gradient but approximate estimate of it. This causes the coefficients to be updated randomly around the optimum values. This generates extra noise at the output in steady state. This is measured by a parameter called excess MSE and it is proportional to step size $\tilde{\mu}$, and filter length M. Thus, using a longer filter length not only requires higher cost, but also introduces more noise. To obtain a better steady state performance, a smaller value of $\tilde{\mu}$ is required, but results in slower convergence.

4. Finite precision effects

For adaptive filters, the dynamic range of the filter output is determined by the time-varying filter co-efficients, which are unknown at the design stage. Also, the feedback of $e(n)$ makes signal scaling (to avoid overflow) more complicated. The leaky NLMS algorithm can be used to reduce numerical errors accumulated in filter coefficients. This prevents overflow in a finite-precision implementation by providing a compromise between minimizing the MSE and constraining the values of the adaptive filter coefficients. In implementation with C6713 DSP double precision floating point arithmetic is used, which provides sufficient accuracy and hence there is no need to implement leaky NLMS here.

5. Computational complexity and filter order M

The NLMS algorithm requires 2^*M+1 additions and 2^*M+1 multiplications at any iteration n, where M is the tap length or filter order. So, the computation complexity depends on the order of filter and it must be carefully chosen. The order M of the filter is usually a function of the separation of the two sound sources as well as of any offset delay in synchronization between the two (or, equivalently, a function of the echo delay in telephony). In many cases, delays of 10-60 ms lead to fewer than 500 taps (at 8000 samples/sec), and NLMS algorithm is feasible on a single chip [4]. Unless the delay is directly estimated, M must be large enough to account for the maximum is possible delay, which may lead to as many as 1500 taps when the two microphones are separated by a few meters (or even exceeding 4000 taps in cases of acoustic echo cancellation in rooms). Such long filter responses can lead to convergence problems as well as to reverberation in the output speech [4]. The noise (echo) can be minimized by optimizing the step size ($\tilde{\mu}$ in Equation (2.3), which changes the filter coefficients each iteration), at the cost of increased settling time for the filter. For large delays, versions of ANC operating in the frequency domain may be more efficient [1], e.g., sub-band systems [20].

To test the NLMS algorithm for different values of parameters like step size and filter length and its effect on stability and convergence, the SIMULINK model as shown in figure 2.7 is used. Here the input reference is noise source and desired signal is only filtered noise. So in the steady state conditions the error signal must be zero and weights are adjusted so that it exactly adapts to the same filter that has filtered noise. Here the mean square error (MSE) signal and deviation of weight vector is measured.

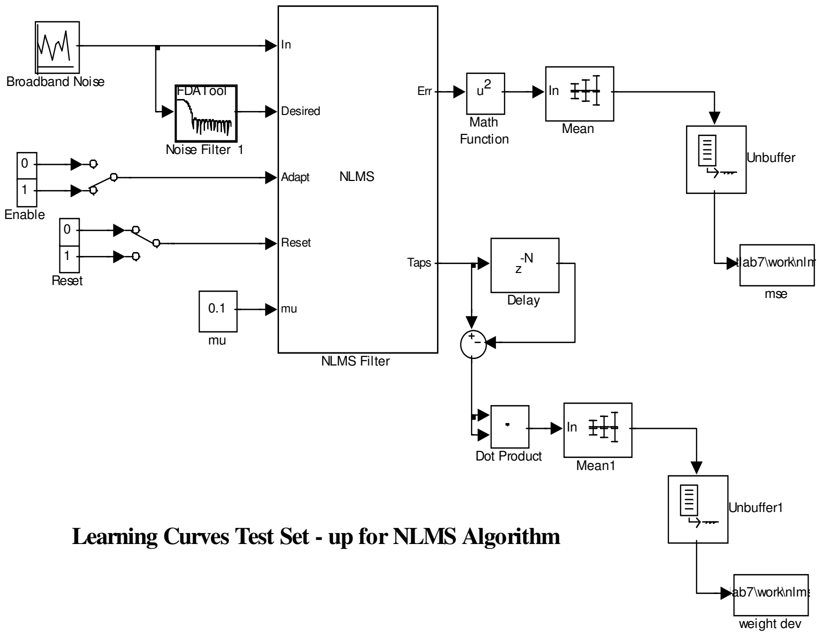


Fig. 2.7 SIMULINK model to obtain learning curves of ANC using NLMS algorithm

The results of the testing are given in figure 2.8 with different values of step size ($\tilde{\mu}$) and filter length (M). In this test, reference signal is Gaussian noise with zero mean and unity variance. The desired signal is given as filtering version of this signal, with 4000Hz Bandwidth. So, the actual speech signal applied is zero signal. In ideal situation the error signal output must be zero at all times, but due to stability and convergence properties of algorithm, it will not achieve ideal performance. The simulation time is set to 1 second and results are recorded. The graph of iterations (time) →mean square error (MSE) and frames (time)→ mean square deviation of 2nd norm of weight vector are obtained by test procedure and plotted in figure 2.8. They are termed as the "Learning Curves". Table 2.1 indicates the numerical values of parameters used for testing the algorithm.

Step Size ($\tilde{\mu}$))	0.001	0.01	0.1	1.0	1.5
Filter Length (M)	16	32	64	128	

Table 2.1 Parameter values for testing NLMS algorithm performance

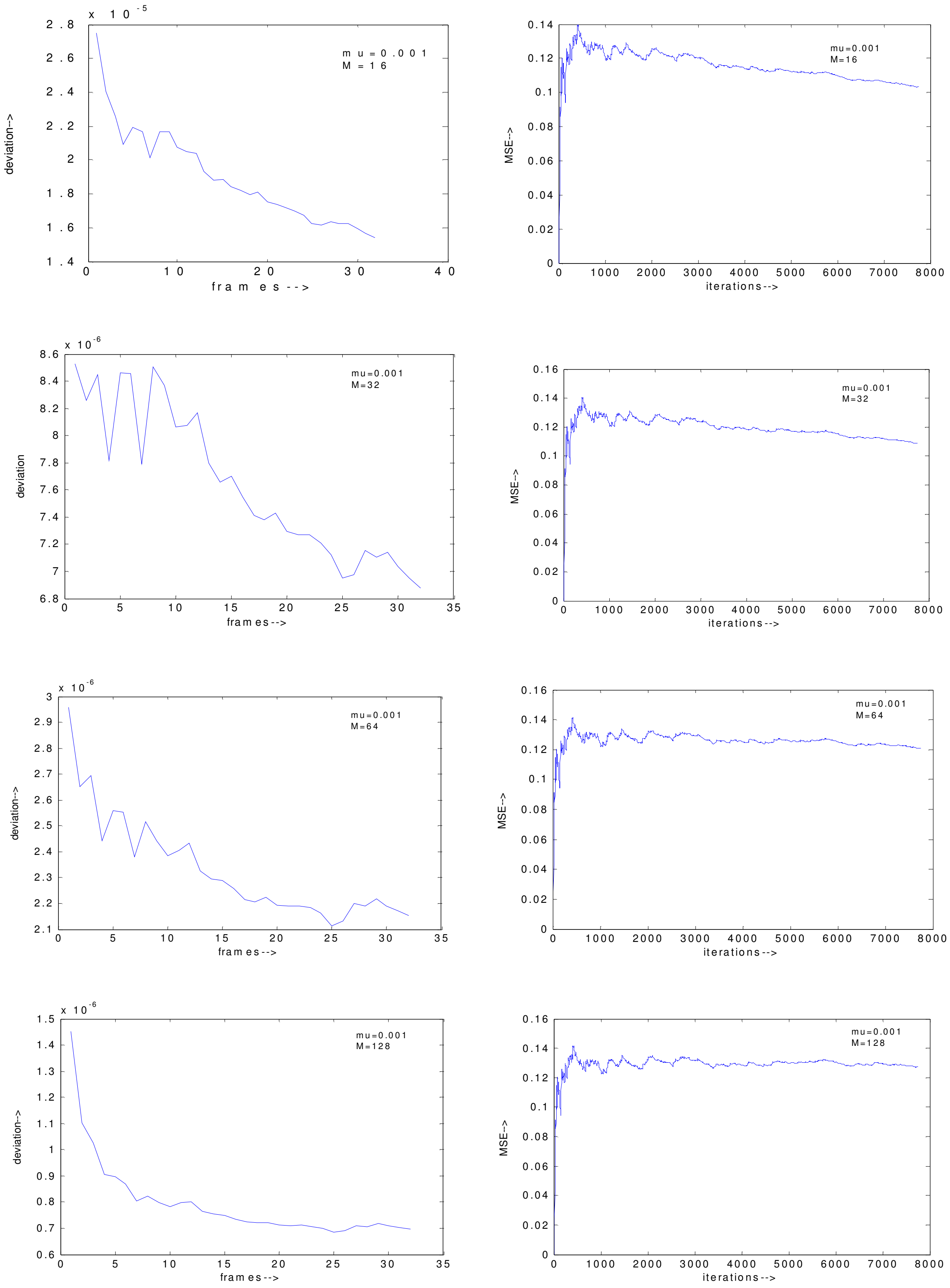


Fig. 2.8(a) Learning curves for NLMS algorithm with $\tilde{\mu} = 0.001$

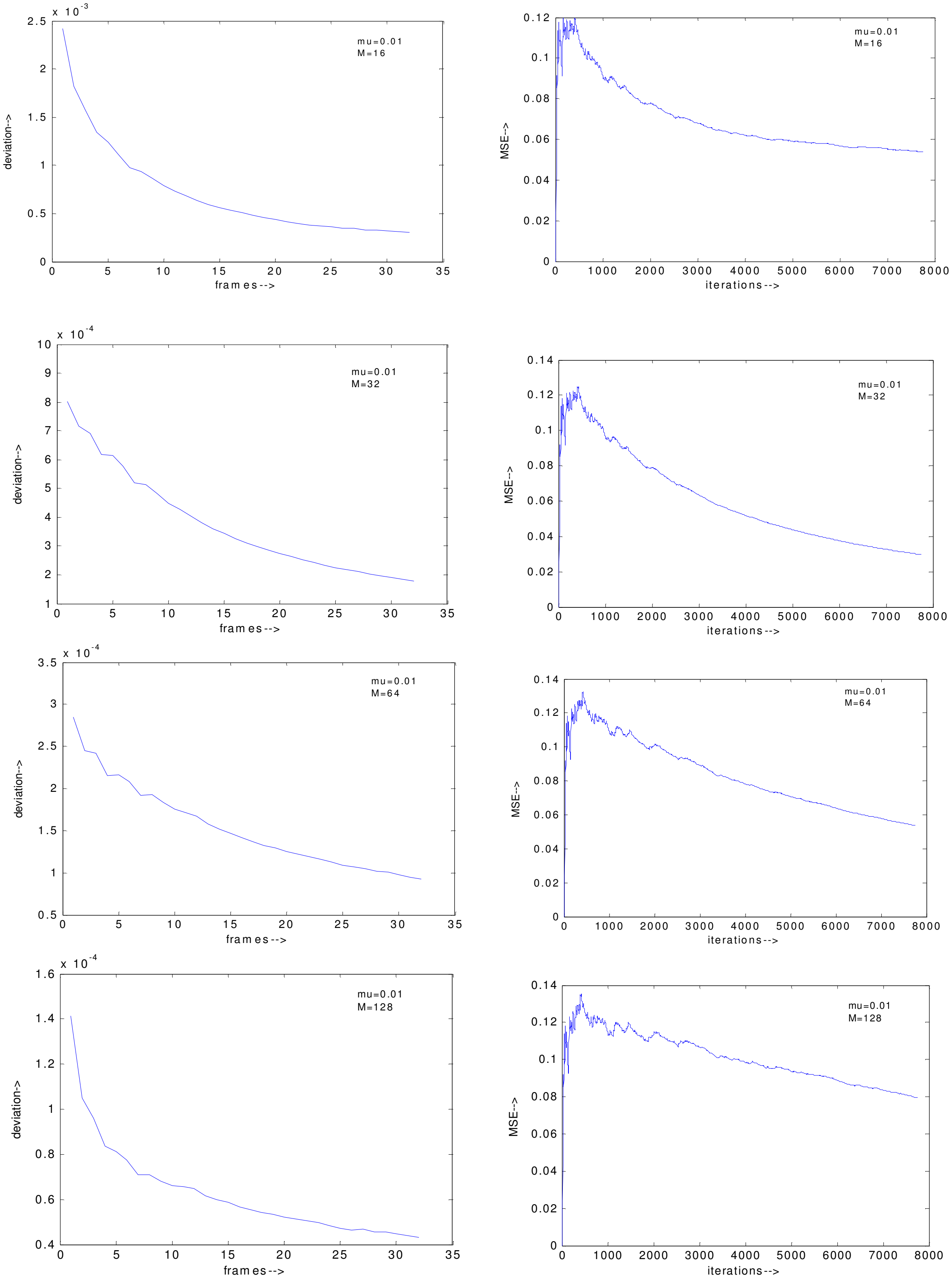


Fig. 2.8(b) Learning curves for NLMS algorithm with $\tilde{\mu}$ = 0.01

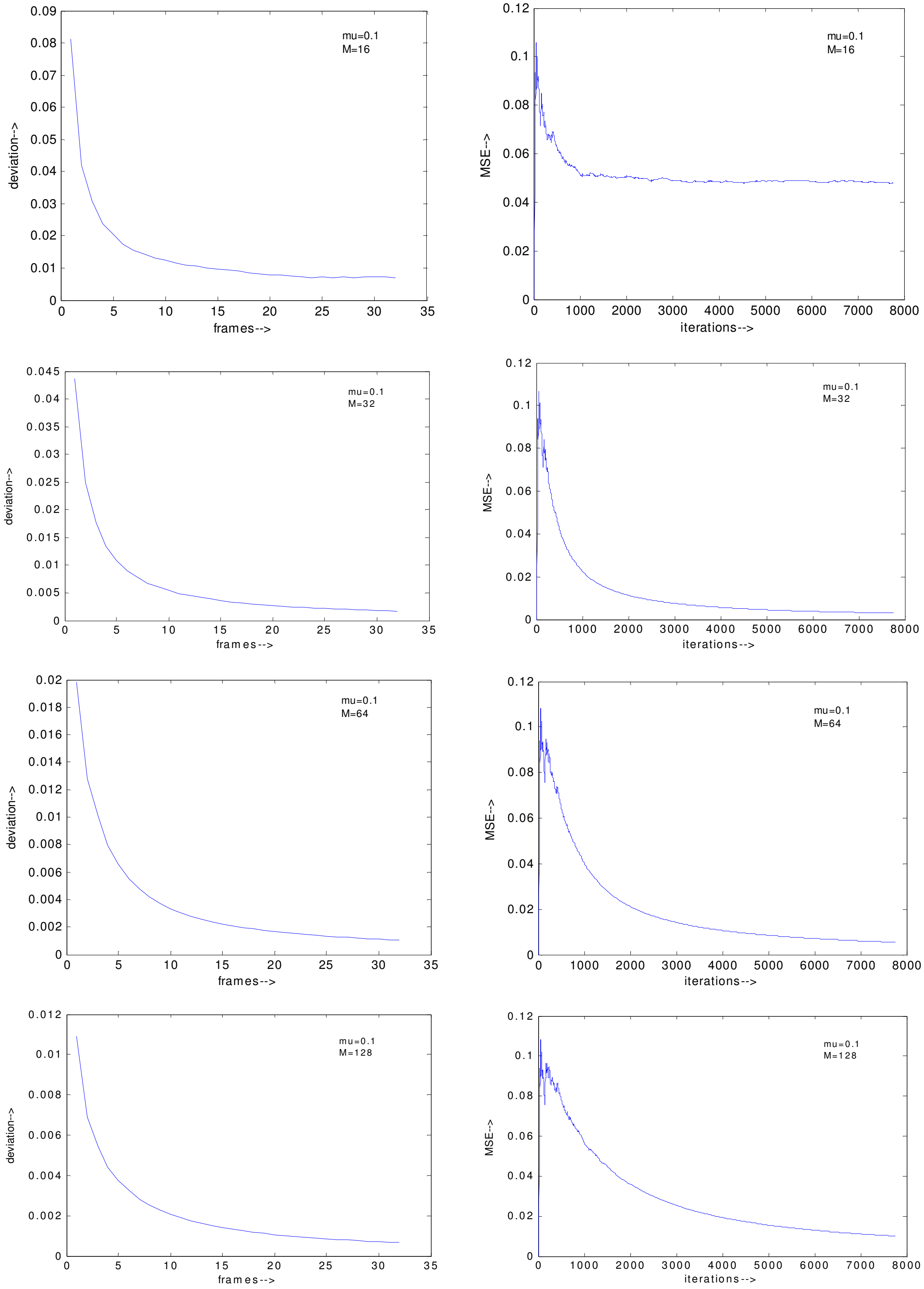


Fig. 2.8(c) Learning curves for NLMS algorithm with $\tilde{\mu} = 0.1$

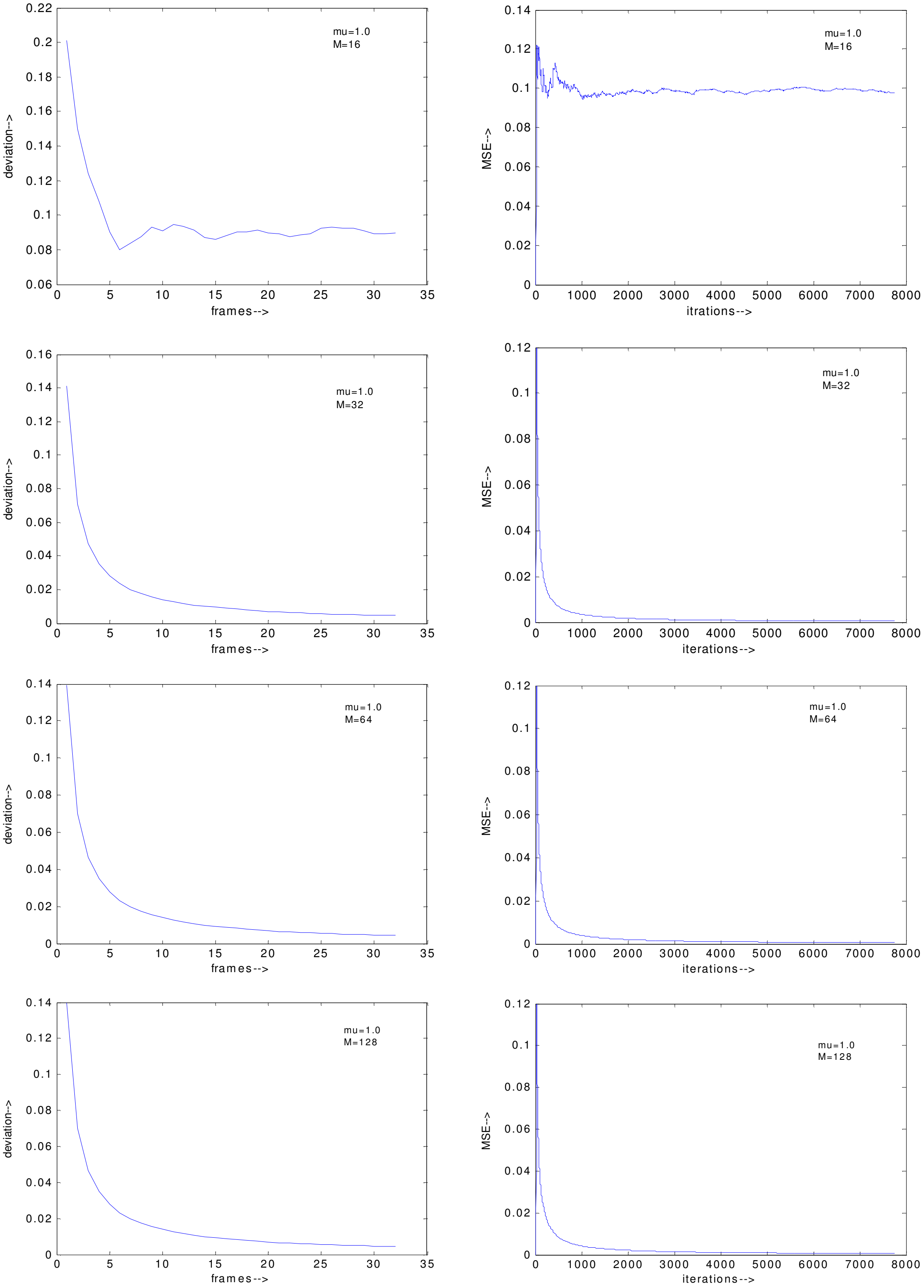


Fig. 2.8(d) Learning curves for NLMS algorithm with $\tilde{\mu} = 1.0$

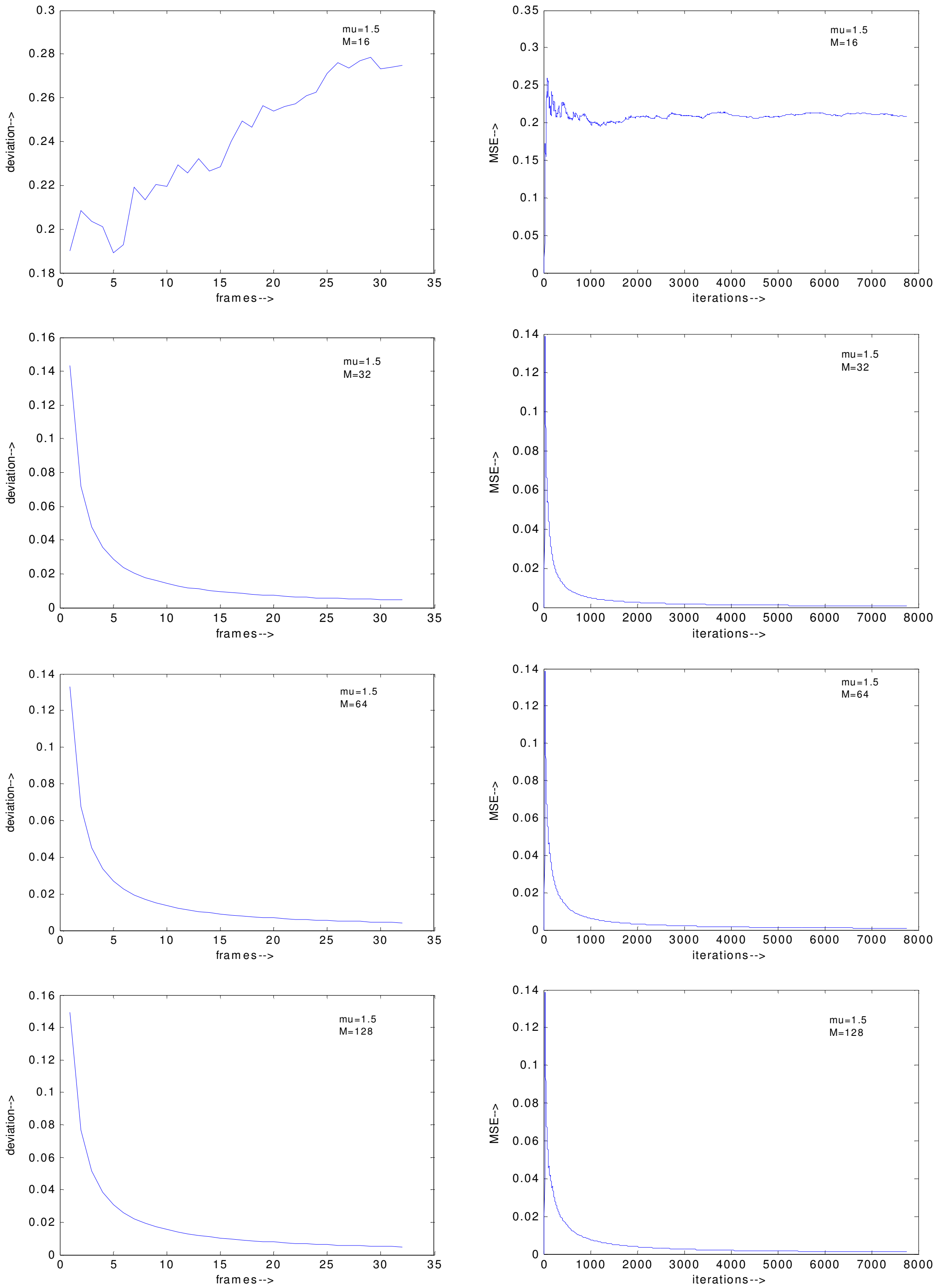


Fig. 2.8(e) Learning curves for NLMS algorithm with $\tilde{\mu} = 1.5$

From these curves it can be concluded that the convergence is faster if larger step size is selected and it is slower if step size is small. But it is not recommended to use very large step size, to account for stability. Also, for larger step size the deviation in weight coefficient vector is large, which introduces its own noise in output (excess MSE). Also, the upper bound on step size is inversely proportional to filter length. So, unnecessary lager filter length must be avoided. The increase in filter length can improve stability but it degrades the steady state performance by introducing excess MSE and more deviation of weight coefficients. Hence for given noise source, the step size of 0.01 to 0.1 and filter length from 32 to 64 is the optimized value. These values are used in all the programs implemented using this algorithm.

The ANC method relies on the microphones being sufficiently apart or on having an acoustic barrier between them. The ANC method is less successful when the secondary signal contains speech components from the primary source, or when there are several or distributed sources; its performance depends on locations of sound sources and microphones, reverberation, and filter length and updating. ANC does best when the microphones are separated enough so that no speech appears in secondary signal, but close enough so that the noise affecting the main signal is also strong in the secondary signal.

2.3.4 Echo Cancellation Using NLMS Algorithm

Echo in a telecommunication system is the delayed and distorted sound which is reflected back to the source. There are two types of echo encountered in telecommunications: acoustic echo, which results from the reflection of sound waves and acoustic coupling between the microphone and loudspeaker, and electrical (line) echo, generated at the two-to-four wire line conversion hybrid transformer due to imperfect impedance matching. Here the model is developed which is equally applicable to both the cases. Here $y(n)$ is a signal with echo or containing both desired speech $x(n)$ from the near end, plus undesired echo $d(n)$ from the far end and $u(n)$ is the near end receive input and $e(n)$ is the output [1, 4].

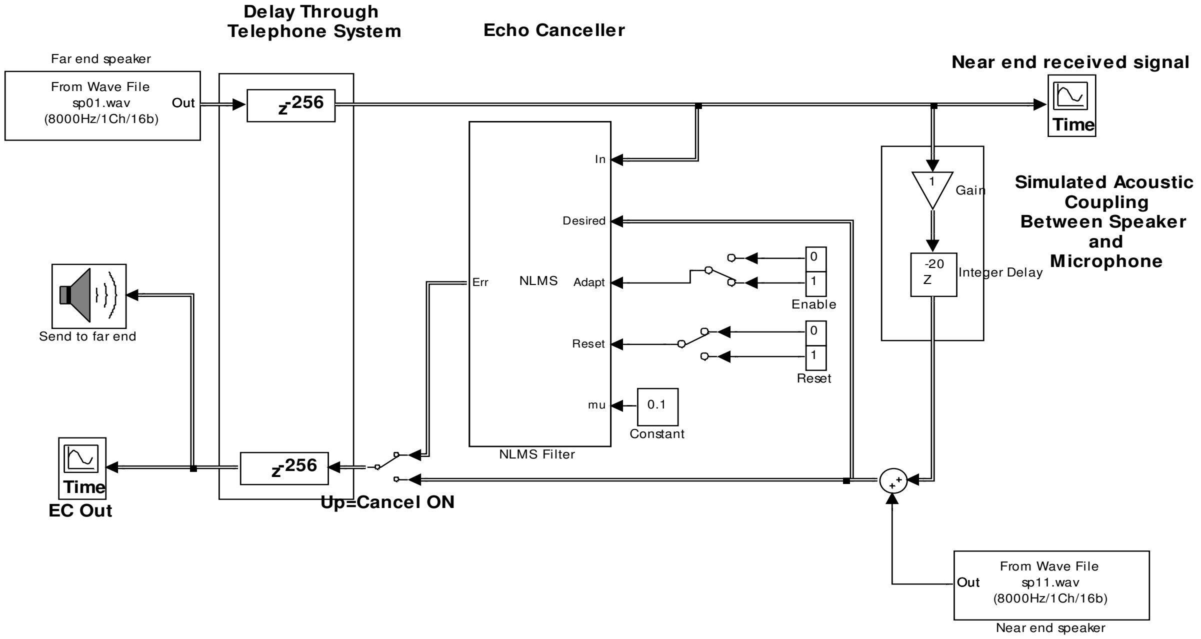


Fig. 2.9 Echo cancellation using NLMS algorithm - SIMULINK model

The SIMULINK implementation of echo cancellation using NLMS algorithm is shown in figure 2.9. The operation and arrangement of various blocks are very much similar to ANC. The same arrangement can be used for room reverberation cancellation. In practice, echo cancellers are applied on both ends to cancel the echoes in each direction.

2.4 Summary

In this chapter an exhaustive survey of various speech enhancement techniques useful for wireless communication systems has been described. Various techniques for all three kinds of major speech enhancement problems that arise in wireless communication are addressed. For noise removal problem it was stated that the DFT based approach is most common but most powerful. It estimates spectral amplitude of clean speech but no attempt is made to estimate the phase of the desired signal; rather the phase of the noisy signal is preserved. Further explanation is given in next chapter. For reverberation cancellation problem the single algorithm is not sufficient for all environments. Multistage algorithms must be used in some combination. Further scope for improvement is seen in RASTA processing. This can be used to handle both noise and reverberation cancellation. The details of RASTA processing are given in chapter 5. The proper investigation in this direction is suggested here. The problem of speaker separation is the most difficult to handle and still the research done is limited in the context of problem solution. The adaptive algorithms like LMS and NLMS which are popular in adaptive control systems can also be used for speech enhancement. Their applications for additive noise removal and echo cancellation are described. The real time SIMULINK and DSK6713 implementation is also mentioned as a case study. However, the problem with this approach is the requirement of reference signal which can be obtained by placing the second microphone to pick up the background noise reference. This is not possible in every situation and hence the single channel solution is the prime requirement in communication systems.

Chapter 3

Speech Enhancement and Detection Techniques: Transform Domain

This chapter describes techniques for additive noise removal which are transform domain methods and based mostly on short time Fourier transform (STFT). The discrete short time Fourier transform is used as transformation tool in most techniques used at present [1-2, 4]. These methods are based on the analysis-modify-synthesis approach. They use fixed analysis window length (usually 20-25ms) and frame based processing. They are based on the fact that human speech perception is not sensitive to spectral phase but the clean spectral amplitude must be properly extracted from the noisy speech to have acceptable quality speech at output and hence they are called short time spectral amplitude or attenuation (STSA) based methods [3]. The phase of noisy speech is preserved in the enhanced speech. The synthesis is mostly done using overlap-add method. They have been one of the well-known and well investigated techniques for additive noise reduction. Also they have less computation complexity and easy implementations. The detailed mathematical expression for the transfer gain function for each method is described along with the terms used in the function. The relative pros and cons of all available methods as well as applications are mentioned. The chapter starts with the brief of analysis and synthesis procedures used in the methods. The other transformation used is discrete wavelet transform (DWT) and the techniques based on DWT are also described in brief here.

The performance evaluation of any algorithm is very important for comparisons. There are several objective and subjective measures are available to evaluate the speech enhancement algorithms. The objective measures are described in brief in this chapter.

3.1 Signal Processing Framework

This section discusses backbone signal processing theories utilized by STSA algorithms.

3.1.1. Short Time Fourier Transform (STFT) Analysis

The short time Fourier transform (STFT) is a time varying Fourier representation that reflects the time varying properties of the speech waveform. The short – time Fourier transform (STFT) is given by:

$$X(n,\omega)=\sum_{m=-\infty}^{\infty}x(m)w(n-m)e^{-j\omega n} \tag{3.1}$$

where $x(m)$ is the input signal, and $w(m)$ is the analysis window, which is time – reversed and shifted by n samples as shown in figure 3.1. The STFT is a function of two variables: the discrete – time index, n, and the (continuous) frequency variables ω. To obtain $X(n+1,\omega)$, slide the window by one sample, multiply it with $x(m)$, and compute the Fourier transform of the window signal. Continuing this will generate a set of STFTs for various values of n until the end of the signal$x(m)$ is reached.

A discrete version of the STFT is obtained by sampling the frequency variable ω at N uniformly spaced frequencies, i.e., at $\omega_k = \frac{2\pi k}{N}, k = 0,1, \dots\dots, \mathrm{N}-1$. The resulting discrete STFT is defined as:

$$X(n, \omega_k) \triangleq X(n, k) = \sum_{m=-\infty}^{\infty} x(m)\, w(n-m) e^{\frac{-j2\pi km}{N}} \tag{3.2}$$

The STFT *X(n,ω)* can be interpreted in two distinct ways, depending on how one treat the time (n) and frequency (ω) variables. If n is fixed, but ω varies, $X(n,\omega)$ can be viewed as the discrete time Fourier transform of the windowed sequence $x(n-m)w(m)$. As such, $X(n,\omega)$ have the same properties as the DTFT. If ω is fixed and the time index n varies, a filtering interpretation emerges.

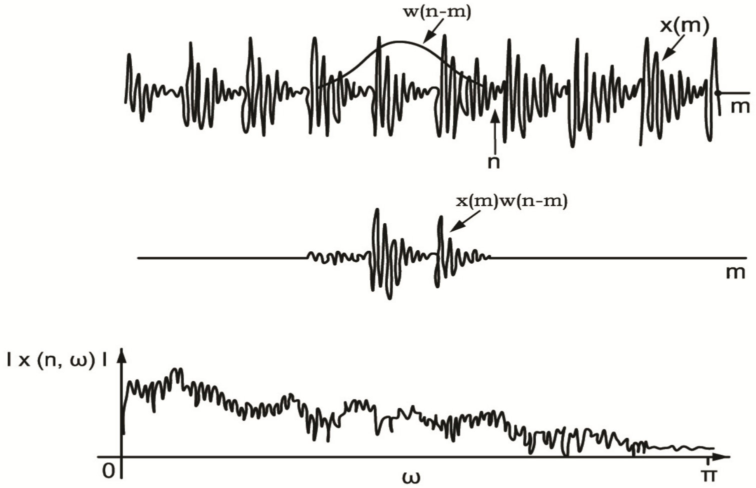


Fig. 3.1 STFT of speech signal

The STFT $X(n,\omega)$ is a two dimensional function of time n and frequency ω. In principle, *X(n, ω)* can be evaluated for each value of n; however, in practice *X(n, ω)* is decimated in time due partly to the heavy computational load involved and partly to the redundancy of information contained in consecutive values of $X(n,\omega)$ (e.g., between $X(n,\omega)$ and $X(n+1,\omega)$). Hence, in most practical applications $X(n,\omega)$ is not evaluated for every sample but for every R sample, where R corresponds to the decimation factor, often express as a fraction of the window length. The sampling, in both time and frequency, has to be done in such a way that $x(n)$ can be recovered from $X(n,\omega)$ without aliasing.

Considering the sampling of $X(n,\omega)$ in time, from equation 3.2 it can be shown that bandwidth of the sequence $X(n,\omega_k)$ (along n, for a fixed frequency ω_k) is less than or equal to the bandwidth of the analysis window $w(n)$. This suggests that $X(n,\omega_k)$ has to be sampled at twice the bandwidth of the window $w(n)$ to satisfy the Nyquist sampling criterion. For the L – point Hamming window, which has an effective bandwidth of:

$$B = \frac{2F_s}{L}\ Hz \tag{3.3}$$

where F_s is the sampling frequency. For this window,$X(n,\omega_k)$ has to be sampled in time at a minimum rate of 2B sample/sec $=\frac{4F_s}{L}$ sample /sec to avoid time aliasing. The corresponding sampling period is $\frac{L}{4F_s}$ sec or L/4 samples. This means that for an L –point Hamming window $X(n,\omega_k)$ needs to evaluate at most every L/4 samples, corresponding to a minimum overlap of 75% between adjacent windows. This strict requirement on the minimum amount of overlap between adjacent windows can be relaxed if zeros are allowed in the window transform [5]. In speech enhancement application, it is quite common to use a 50% rather than 75% overlap between adjacent windows. This implies that $X(n,\omega_k)$ is evaluated every L/2 samples; that is, it is decimated by a factor of L/2, where L is the window length. As STFT $X(n,\omega_k)$(for fixed n) is the DTFT of the window sequence $w(m)x(n-m)$. Hence, to recover the windowed sequence $w(m)x(n-m)$ with no aliasing, it is required that the frequency variable ω be sampled at $N\ (N \geq L)$ uniformly spaced frequencies, i.e., at $\omega_k = 2\pi k/N\ , k = 0,1, \dots \dots . . N-1$.

3.1.2. Overlap Add Synthesis

The method for reconstructing $x(n)$ from its STFT is overlap add method, which is widely used in speech enhancement. Assuming the STFT $X(n,\omega)$ sampling in time every R samples as $X(rR,\omega)$, the overlap add method is given by following equation [5]:

$$y(n) = \sum_{r=-\infty}^{\infty} [\frac{1}{N}\sum_{k=0}^{N-1} X(rR,\omega_k)e^{j\omega_k n}] \tag{3.4}$$

The term in brackets is an IDFT yielding for each value of *r* the sequence:

$$y_r(n) = x(n)w(rR-n) \tag{3.5}$$

Equation 3.4 can be expressed as:

$$y(n) = \sum_{r=-\infty}^{\infty} y_r(n) = x(n) \sum_{r=-\infty}^{\infty} w(rR - n) \tag{3.6}$$

From Equation 3.6 it can be seen that the signal $y(n)$ at time n is obtained by summing all the sequences $y_r(n)$ that overlap at time n. Provided that the summation term in Equation 3.6 is constant for all n, we can recover $x(n)$ exactly (within a constant) as:

$$y(n) = C.x(n) \tag{3.7}$$

where C is a constant. It can be shown that if $X(n, \omega)$ is sampled properly in time, i.e., R is small enough to avoid time aliasing, and then C is equal to:

$$C = \sum_{r=-\infty}^{\infty} w(rR - n) = \frac{W(0)}{R} \tag{3.8}$$

independent of time n [5]. Equation 3.7 and Equation 3.8 indicate that $x(n)$ can be reconstructed exactly (within a constant) by adding overlapping sections of the windowed sequences $y_r(n)$. The constraint imposed on the window is that it satisfies equation 3.8; that is, the sum of all analysis windows shifted by increments of R samples adds up to a constant. Furthermore, R needs to be small enough to avoid time aliasing. With $R = {}^{L}/_{2}$ (i.e., 50% window overlap), which is most commonly used in speech enhancement, the signal $y(n)$ consists of two terms:

$$y(n) = x(n)w(R - n) + x(n)w(2R - n);\ 0 \leq n \leq R - 1 \tag{3.9}$$

Figure 3.2 shows how the overlap addition is implemented for an L-point Hamming window with 50% overlap ($R = L/2$). In the context of speech enhancement, the enhanced output signal in frame t consists of the sum of the windowed signal [with $w(R - n)$] enhanced in the previous frame $(t - 1)$ and the windowed signal [with $w(2R - n)$] enhanced in the present frame (t).

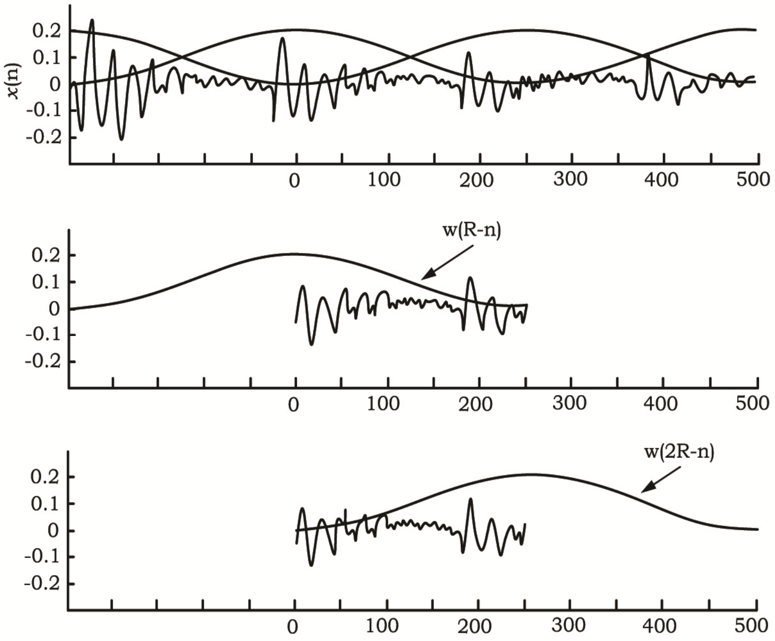


Fig. 3.2 Overlap add synthesis with 50% overlap (L=500, R=L/2)

Figure 3.3 shows the flow diagram of the analysis-modify-synthesis method, which can be used in any frequency domain speech enhancement algorithm. The L-point signal sequence needs to be padded with sufficient zeroes to avoid time aliasing. In the context of speech enhancement, the input signal $x(n)$ in figure 3.3 corresponds to the noisy signal and the output signal $y(n)$ to the enhanced signal.

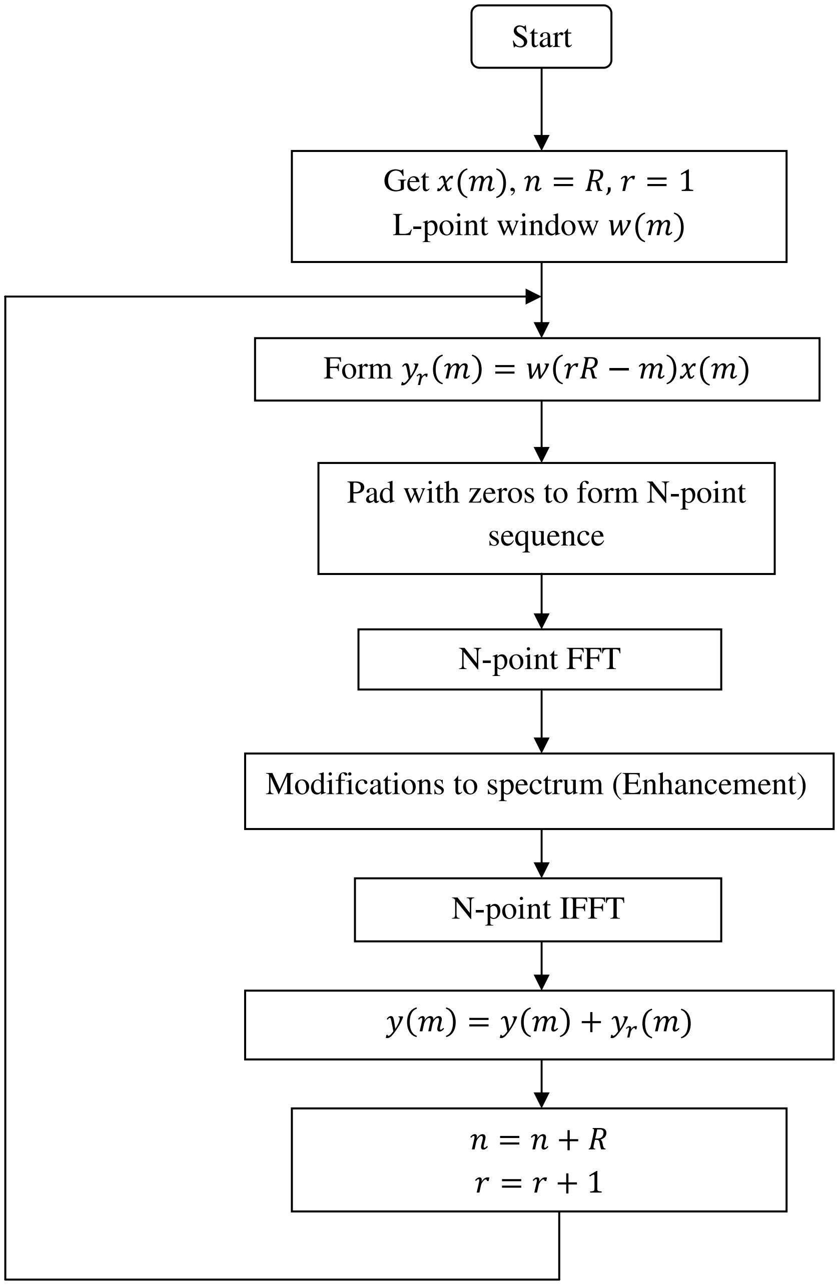


Fig. 3.3 Flow chart of analysis-modify-synthesis method

3.1.3. Spectrographic Analysis of Speech Signals

The two dimensional function $|X(n, \omega)|^2$ provides the spectrogram of the speech signal – a two dimensional graphical display of the power spectrum of speech as a function of time.

This is a widely used tool employed for studying the time varying spectral and temporal characteristic of speech. It is given by:

$$S(n,\omega) = |X(n,\omega)|^2 \tag{3.10}$$

The spectrogram describes the speech signal's relative energy concentration in frequency as a function of time and, as such, it reflects the time varying properties of the speech waveform. Frequency is plotted vertically on the spectrogram with time plotted horizontally. Amplitude, or loudness, is depicted by gray scale or color intensity. Color spectrograms represent the maximum intensity as red gradually decreasing through orange, yellow, green and blue (illustrated in figure 3.5).

Two kinds of spectrograms, narrow-band and wide-band, can be produced, depending on the window length used in the computation of $S(n,\omega)$. A long duration window (at least two pitch periods long) is typically used in the computation of the narrow-band spectrogram and a short window in the computation of the wide band spectrogram. The narrow-band spectrogram gives good frequency resolution but poor time resolution. The fine frequency resolution allows the individual harmonics of speech to be resolved. These harmonics appear as horizontal striations in the spectrogram (figure 3.5, top panel). The main drawback of using long windows is the possibility of temporally smearing short-duration segments of speech, such as the stop consonants. The wideband spectrogram uses short-duration windows (less than a pitch period) and gives good temporal resolution but poor frequency resolution. The main consequence of the poor frequency resolution is the smearing (in frequency) of individual harmonics in the speech spectrum, yielding only the spectral envelope of the spectrum (figure 3.5, bottom panel). The fundamental frequency (reciprocal of pitch period) range is about 60-150Hz for male speakers and 200-400Hz for females and children [5]. So the pitch period varies approximately 2-20ms. Therefore, in practice a compromise is made by setting a suitable practical value for window duration of 20-30ms. This way it is possible to accommodate a broad range of general speakers. These values are used throughout the research work. This also represents the harmonic structure of speech fairly correctly.

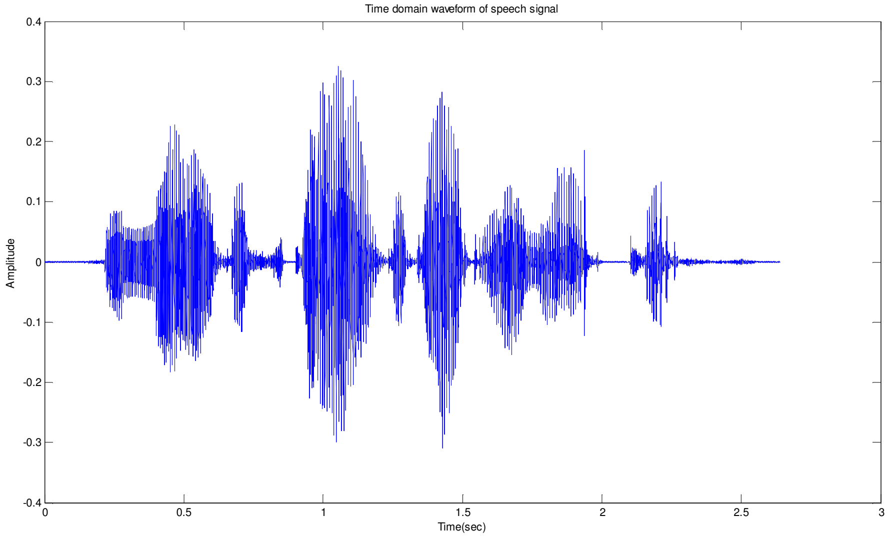


Fig. 3.4 Time domain waveform of speech signal containing sentence 'He knew the skill of the great young actress'

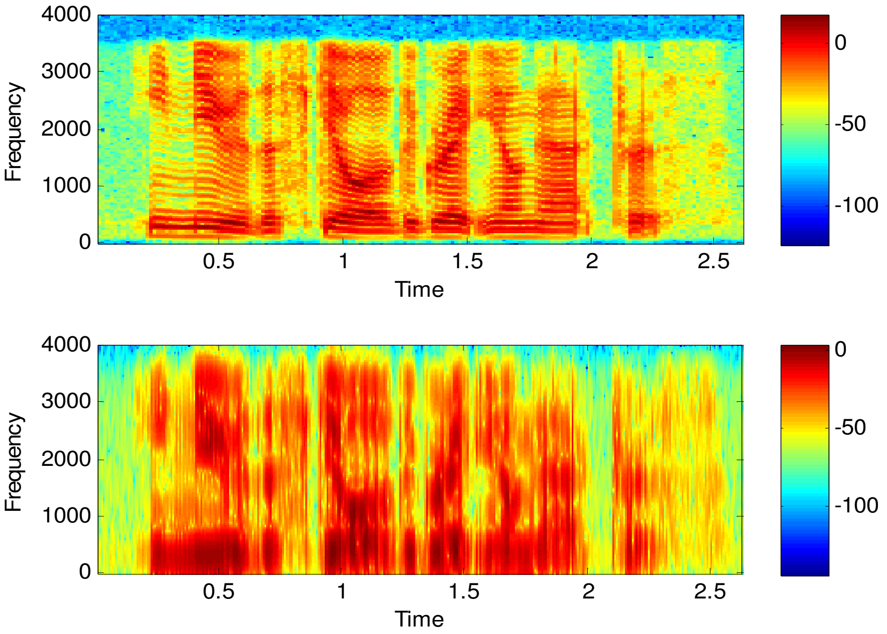


Fig. 3.5 Narrowband (top panel) and wideband (bottom panel) spectrogram of the speech signal in figure 3.4

3.2 Short Time Spectral Amplitude (STSA) Algorithms

Figure 3.6 specifies the various STSA algorithms along with their original proposer. STSA based approaches assume that noise is additive white noise and stationary for a frame and changes slowly in comparison with the speech. Most real environmental noise sources such as vehicles, street noise, babble noise etc. are non-stationary and coloured in nature. Therefore complete noise cancellation is more complex as it is not possible to completely track such noises. However, using this assumption it is possible to achieve significant reduction in the background noise levels using simple techniques. The noise statistics are typically characterized during voice-inactivity regions between speech pauses using a voice activity detector (VAD). The VAD always becomes an integral part of any STSA based algorithm [3-4]. The operation and types of VAD are described in next section. Table 3.1 describes the list of symbols used in STSA methods description.

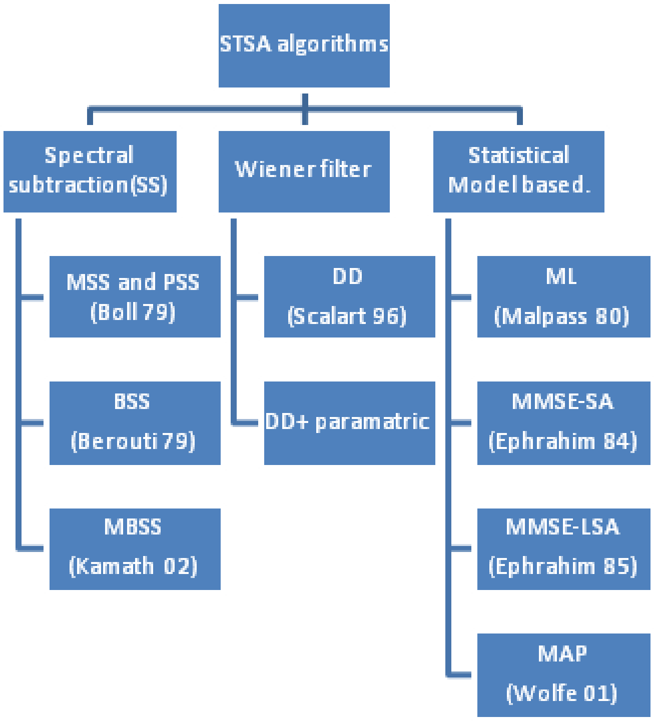


Fig. 3.6 A chart showing various STSA algorithms

Symbol	Meaning
$y(n)$	Degraded Speech signal
$x(n)$	Clean speech signal
$d(n)$	Additive noise
α	Over subtraction factor
β	Spectral floor parameter
p	Spectral power
η	Smoothing constant
K	Discrete frequency bin
δ	Tweaking factor
μ, v	Parameters of Wiener filter
$\xi(K)$	*A priori* SNR at frequency bin K $= \frac{\lvert\hat{X}(K)\rvert^2}{\lvert\hat{D}(K)\rvert^2}$
$\gamma(K)$	*A posteriori* SNR at frequency bin K $= \frac{\lvert Y(K)\rvert^2}{\lvert\hat{D}(K)\rvert^2}$
i	Frequency band
$\phi_y(K)$	Phase of signal y(n) at frequency bin K
F_s	Sampling frequency
f_i	Upper frequency in the i^{th} frequency band

Table 3.1 List of symbols used in STSA algorithms

3.3 Spectral Subtraction (SS) Methods

Spectral subtraction method was first proposed by S.F.Boll [7]. The basic principle of spectral subtraction is to subtract an estimate of the average noise spectrum from noisy speech magnitude spectrum. Degraded speech signal is modelled as

$$y(n) = x(n) + d(n) \tag{3.11}$$

Taking DFT of (1) gives

$$Y(K) = X(K) + D(K) \tag{3.12}$$

The estimate of $D(K)$ is obtained by using VAD and updated during non-speech or silence periods. For good initial estimate it requires initial silence period of around 0.2 seconds.

3.3.1 Magnitude and Power Spectral Subtraction (MSS and PSS)

From equation 3.12 taking only magnitude of spectrum we can write

$$\left|\hat{X}(K)\right| = \begin{cases} |Y(K)| - \left|\hat{D}(K)\right| \; if \; |Y(K)| > \left|\hat{D}(K)\right| \\ 0 \;\; else \end{cases} \tag{3.13}$$

The half wave rectification process is only one of many ways of ensuring non-negative$\left|\hat{X}(K)\right|$. The original speech estimate is given by preserving the noisy speech phase $\emptyset_y(K)$. This is partly motivated by the fact that phase that does not affect speech intelligibility [19], may affect speech quality to some degree.

$$\hat{X}(K) = [|Y(K)| - |\hat{D}(K)|]e^{j\emptyset_y(K)} \qquad (3.14)$$

The preceding discussion of magnitude spectrum subtraction can be extended to power spectrum domain as

$$|\hat{X}(K)|^2 = \begin{cases} |Y(K)|^2 - |\hat{D}(K)|^2 & if \;\; |Y(K)|^2 > |\hat{D}(K)|^2 \\ 0 \;\; else \end{cases} \qquad (3.15)$$

The spectral power subtraction can be generalized [11] with an arbitrary spectral order p, called generalized spectral subtraction (GSS) and defined as

$$|\hat{X}(K)|^p = \begin{cases} |Y(K)|^p - |\hat{D}(K)|^p \; if \;\; |Y(K)|^p > |\hat{D}(K)|^p \\ 0 \;\; else \end{cases} \qquad (3.16)$$

The general block diagram of spectral subtraction method is shown in figure 3.7.

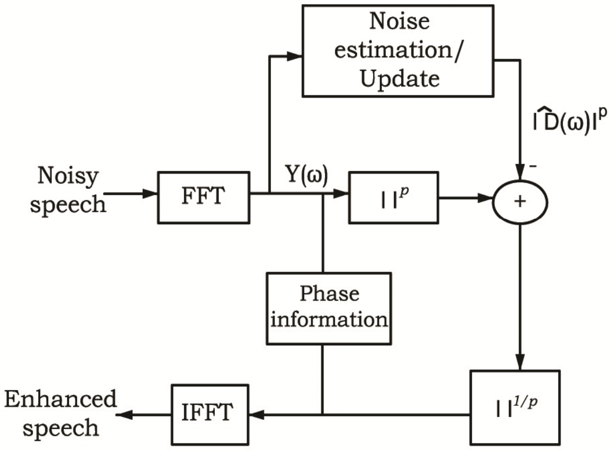


Fig. 3.7 Block representation of general spectral subtraction method

3.3.2 Berouti Spectral Subtraction (BSS)

The major problem of the basic spectral subtraction is that, the algorithm may itself introduce a synthetic noise, called musical noise. The half wave rectification is non-linear process and it creates small, isolated peaks in the spectrum occurring at random frequency locations in each frame. In time domain these peaks result in tones with randomly changing frequency from frame to frame. This musical noise is more disturbing to the listener than the original noise. Most researchers suggest that it is difficult to minimize musical noise without affecting the speech signal. So there is always a trade-off between the amount of noise reduction and speech distortion.

Berouti *et al.* [8] proposed an important variation of the original method, which improves the noise reduction compare to the basic spectral subtraction. It introduces an over subtraction factor ($\alpha \geq 1$) and spectral floor parameter ($0 < \beta < 1$) ; and it is defined as

$$\left|\hat{X}(K)\right|^2 = \begin{cases} |Y(K)|^2 - \alpha\left|\hat{D}(K)\right|^2 \; if \;\; |Y(K)|^2 > (\alpha+\beta)\left|\hat{D}(K)\right|^2 \\ \beta\left|\hat{D}(K)\right|^2 \quad else \end{cases} \tag{3.17}$$

The parameter β controls the amount of remaining residual noise and the amount of perceived musical noise. Large β produces audible residual noise but small musical noise and vice versa. The parameter α affects the amount of speech spectral distortion caused by the subtraction in equation 3.17. Large values of α produce high speech distortion and vice versa [9]. The value of α should vary linearly with SNR in dB on per frame basis as

$$\alpha = \alpha_0 - s \times (SNR) \tag{3.18}$$

where α_0 is the value of α at 0 dB SNR, s is slope and SNR is estimated *a posteriori* frame SNR in dB. The optimized value of $\alpha_0$is between 3 to 6 and that of β is in the range of 0.02 to 0.06 for SNR$\leq 0dB$ and in the range of 0.005 to 0.02 for SNR$> 0dB$. Though usage of over subtraction of the noise spectrum and the introduction of a spectral floor serve to minimize residual noise and musical noise, musical noise is not completely avoided.

Equation 3.17 can be extended for general p^{th} power as

$$|\hat{X}(K)|^p = \begin{cases} |Y(K)|^p - \alpha\left|\hat{D}(K)\right|^p \qquad if \;\; |Y(K)|^p > (\alpha+\beta)\left|\hat{D}(K)\right|^p \\ \beta\left|\hat{D}(K)\right|^p \;\; else \end{cases} \tag{3.19}$$

From this

$$H(K) = \frac{\left|\hat{X}(K)\right|}{|Y(K)|} = \begin{cases} \sqrt[p]{1 - \alpha\frac{|\hat{D}(K)|^p}{|Y(K)|^p}} = \frac{(\sqrt{\gamma(K)}^p - \alpha)^{\frac{1}{p}}}{\sqrt{\gamma(K)}} \quad if \;\; |\gamma(K)|^{p/2} > (\alpha+\beta) \\ \beta^{\frac{1}{p}} \; \frac{1}{\sqrt{\gamma(K)}} \quad else \end{cases} \tag{3.20}$$

In the context of linear system theory, $H(K)$ is known as the system's transfer function. In speech enhancement, $H(K)$ is referred to as the gain function, or suppression function. $H(K)$ in equation 3.18 is real and, in principle, is always positive, taking values in the range of $0 \leq H(K) \leq 1$. Negative values are sometimes obtained owing to inaccurate estimates of the noise spectrum.$H(K)$ is called the suppression function because it provides the amount of suppression (or attenuation, as $0 \leq H(K) \leq 1$) applied to noisy power spectrum $|Y(K)|^2$ at a given frequency to obtain the enhanced power spectrum $|\hat{X}(K)|^2$. The shape of the suppression function is unique to a particular speech enhancement algorithm. For this reason, different algorithms are compared by comparing their corresponding suppressions functions.

3.3.3 Multiband Spectral Subtraction (MBSS)

This method proposed by S.D.Kamath [10] performs spectral subtraction with different over subtraction factor in different non-overlapped frequency bands. It is based on the fact that, in general, noise will not affect the speech signal uniformly over the whole spectrum. Some frequencies will be affected more adversely than others depending on the spectral characteristics of the noise. This can address the problem of colored noise reduction. The spectral subtraction rule in i^{th} frequency band is given by

$$\left|\widehat{X}_i(K)\right|^2 = \begin{cases} |\bar{Y}_i(K)|^2 - \alpha_i \delta_i \left|\widehat{D}_i(K)\right|^2 \; if \;\; |Y_i(K)|^2 > \alpha_i \delta_i \left|\widehat{D}_i(K)\right|^2 \\ \beta |\bar{Y}_i(K)|^2 \;\; else \qquad\qquad\qquad for \; b_i \leq K \leq e_i \end{cases} \tag{3.21}$$

where the spectral floor parameter β is set to 0.002. The over subtraction parameter in i^{th} band is specified as

$$\alpha_i = \begin{cases} 4.75 & SNR_i < -5\,dB \\ 4 - \dfrac{3}{20}\,(SNR_i) & -5\,dB \leq SNR_i \leq 20\,dB \\ 1 & SNR_i > 20\,dB \end{cases} \tag{3.22}$$

where the band SNR_i is given by:

$$SNR_i(dB) = 10log_{10}\left(\frac{\sum_{K=b_i}^{e_i} |\bar{Y}_i(K)|^2}{\sum_{K=b_i}^{e_i} |\widehat{D}_i K)|^2}\right) \tag{3.23}$$

The additional over subtraction factorδ_i; called tweaking factor provides additional degree of control in each frequency band. The values of this factor are empirically determined and set according to following equation. Usually 4-8 linearly spaced frequency bands are used.

$$\delta_i = \begin{cases} 1 & f_i < 1\,kHz \\ 2.5 & 1\,kHz \leq f_i \leq \dfrac{F_s}{2} - 2\,kHz \\ 1.5 & f_i > \dfrac{F_s}{2} - 2\,kHz \end{cases} \tag{3.24}$$

In the preceding equations $\bar{Y}_i(K)$ is the smoothed noisy spectrum of the i^{th} frequency band estimated in the preprocessing stage. A weighted spectral average is taken over preceding and succeeding frames of speech as follows:

$$\left|\bar{Y}_j(K)\right| = \sum_{i=-M}^{M} W_i \left|Y_{j-i}(K)\right| \tag{3.25}$$

The number of frames M is limited to 2 to prevent spectral smearing and weights $W_i = [0.09,0.25,0.32,0.25,0.09]$ set empirically. To further mask any remaining musical noise, a small amount of the noisy spectrum is introduced back to the enhanced spectrum as follows:

$$|\bar{\bar{X}}_i(K)|^2 = |\hat{X}_i(K)|^2 + 0.05\,|\bar{Y}_i(K)|^2 \tag{3.26}$$

where $|\bar{\bar{X}}_i(K)|^2$ is the newly enhanced power spectrum.

The block diagram of the multiband method proposed in [10] is shown in figure 3.8. The signal is first windowed and the magnitude spectrum is estimated using FFT. The noisy speech spectrum is then preprocessed to the noise and speech spectra are divided into N contiguous frequency bands and the over subtraction factors for each band are calculated. The individual frequency bands of the estimated noise spectrum are subtracted from the corresponding bands of the noisy speech spectrum. Lastly, the modified frequency bands are recombined and the enhanced signal is obtained by taking the IFFT of the enhanced spectrum using the noisy speech phase. The motivation behind the preprocessing stage is to reduce the variance of the spectral estimate and consequently reduce the residual noise. The preprocessing serves to precondition the input data to surmount the distortion caused by errors in the subtraction process. Hence, instead of directly using the power spectrum of the signal, a smoothed version of the power spectrum is used. Smoothing of the magnitude spectrum as per [7] was found to reduce the variance of the speech spectrum and contribute to speech quality improvement. However it is not reducing the residual noise [10].

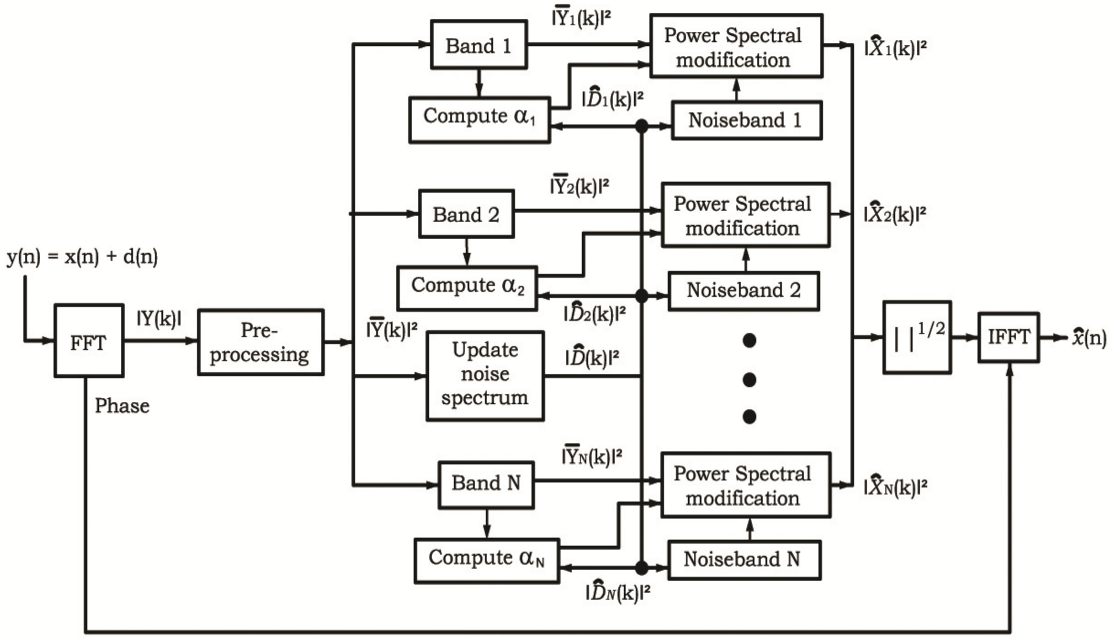


Fig. 3.8 Block diagram of MBSS method

3.4 Wiener Filtering Methods

The traditional Wiener filter used in most adaptive filtering and control applications can also be applied to speech enhancement. The Wiener filter is an optimal filter that minimizes the mean square error of a desired signal in time domain and assumes that the speech and noise are uncorrelated. In terms of our speech enhancement problem the Wiener filter is given by

$$\left|\hat{X}(K)\right| = \frac{\xi(K)}{1+\xi(K)}|Y(K)| \qquad (3.27)$$

This filter is a function of *a priori* SNR.

3.4.1 Decision Direct (DD) Approach

The Wiener filter is non-causal and cannot be implemented in real time as it requires a prior knowledge of clean speech signal spectrum $\left|\hat{X}(K)\right|$. As a solution, Ephraim and Malah [13] proposed the decision directed rule to estimate this ratio and it is used by Scalart *et al.* [15] with Wiener filter. The decision direct rule for frame *t* is given by

$$\xi^{(t)}(K) = \eta\frac{\left|\hat{X}^{(t-1)}(K)\right|^2}{\left|\hat{D}^{(t)}(K)\right|^2} + (1-\eta)max(\gamma^{(t)}(K)-1,0) \qquad (3.28)$$

Where $0 \leq \eta \leq 1$ is smoothing constant and normally it is set to 0.98. In Wiener filter $0 \leq H(K) \leq 1$, and $H(K) \approx 0$ when $\xi(K) \rightarrow 0$ (i.e., at extremely low-SNR regions) and $H(K) \approx 1$ when $\xi(K) \rightarrow \infty$ (i.e., at extremely high-SNR regions). So, according to equation 3.26, the Wiener filter emphasizes portions of the spectrum where the SNR is high and attenuates portions of the spectrum where the SNR is low. This recursive relationship provides smoothness in the estimate of $\xi(K)$, and consequently can eliminate the musical noise [18]. Good performance was reported in [17] with the algorithm. Speech enhanced by the preceding algorithm had little speech distortion but had notable residual noise.

3.4.2 DD Approach with Parametric Wiener Filter

A more general Wiener filter gain function estimation was obtained by Lim and Oppenheim [6] and it is called parametric Wiener filter and it is given by

$$\left|\hat{X}(K)\right| = \left(\frac{\xi(K)}{\mu + \xi(K)}\right)^{v} |Y(K)| \tag{3.29}$$

By varying parameters μ and v we can obtain different Wiener filters with different attenuation characteristics.

3.5 Statistical Model Based Methods

The Wiener filter is a linear estimator of the complex spectrum of the signal; an alternate approach is to use non-linear estimators of the magnitude spectrum only using various statistical model and optimization criteria. These estimators consider the probability density function (pdf) of speech and noise DFT coefficients explicitly into account and use Gaussian distribution. Various techniques of estimation theory can be applied to speech enhancement problem and mainly they fall in following categories.

3.5.1 Maximum Likelihood (ML) Approach

The ML approach is first applied to speech enhancement by McAulay and Malpass [12]. The magnitude and phase of clean signal are assumed to be unknown but deterministic. The pdf of noise Fourier transform coefficients is assumed to be zero-mean complex Gaussian. Based on this the ML estimation is given by

$$\left|\hat{X}(K)\right| = \frac{1}{2}\left(|Y(K)| + \sqrt{|Y(K)|^2 - \left|\hat{D}(K)\right|^2}\right) \tag{3.30}$$

Analysis reports that it provides smaller attenuation at lower SNRs compared to SS and Wiener filter methods and hence this method is not preferred as speech enhancement method.

3.5.2 Minimum Mean Square Error (MMSE) Approach

This method takes MMSE estimate of spectral amplitude rather than complex spectrum as in Wiener filter. The MMSE-SA optimization suggested by Ephrahim and Malah [13] is given by the equation:

$$\left|\hat{X}(K)\right| = \frac{\sqrt{\pi}}{2}\frac{\sqrt{v(K)}}{\gamma(K)}e^{-\frac{v(K)}{2}}\left[\left(1+v(K)\right)I_0\left(\frac{v(K)}{2}\right)+v(K)I_1\left(\frac{v(K)}{2}\right)\right]|Y(K)|; \tag{3.31}$$

$$v(K) = \frac{\xi(K)}{1+\xi(K)}\gamma(K)$$

Here $I_0(.)$ And $I_1(.)$ Denote the modified Bessel functions of zero and first order. This estimation assumes that the speech and noise signal spectrum are statistically independent zero mean complex Gaussian random variables. The decision direct rule is used to estimate *a p*riori SNR. Research shows that with the speech corrupted by an additive white noise; enhanced speech with this approach has colorless residual noise; that is, the residual noise produced by this method is not musical as in SS, Wiener filter and ML method. The speech distortion is also less compared to Wiener filter. The smoothing parameter η controls the trade-off between speech distortion and residual noise. In summary, it is the smoothing behavior of the decision-directed approach in conjunction with the suppression rule that is responsible for reducing the musical noise effect in the MMSE algorithm. Using the method of Lagrange multipliers, the optimal solution for phase estimation can be shown to be

$$\exp\left(j\hat{\phi}_x\right) = \exp\left(j\phi_y\right) \tag{3.32}$$

That is, the noisy noise phase (ϕ_y) is the optimal in the MMSE sense.

3.5.3 MMSE Log Spectral Amplitude (LSA) Approach

As a variant, Ephrahim and Malah [14] proposed MMSE log spectral amplitude (MMSE-LSA) estimator based on the fact that a distortion measure with the log spectral amplitudes is more suitable for speech processing. It minimizes the mean square error of the log amplitude spectra and the estimate of the clean speech is given by the equation:

$$\left|\hat{X}(K)\right| = \frac{\xi(K)}{1+\xi(K)}\exp\left(-\frac{1}{2}\int_{v(K)}^{\infty}\frac{e^{-t}}{t}dt\right)|Y(K)| \tag{3.33}$$

The integral in preceding equation is exponential integral and can be evaluated numerically. The exponential integral,$Ei(x)$, can be approximated as follows [11]:

$$Ei(x) = \int_{x}^{\infty} \frac{e^{-x}}{x} dx \approx \frac{e^{x}}{x} \sum_{k} \frac{k!}{x^{k}} \tag{3.34}$$

This method reduces the residual noise considerably without introducing much speech distortion.

3.5.4 Maximum a Posteriori (MAP) Approach

This method estimates clean speech spectral amplitude based on maximization of the *a posteriori* (MAP) pdf [16]. The MAP estimator is given by the equation

$$\left|\hat{X}(K)\right| = \frac{\xi(K) + \sqrt{\xi(K)^{2} + 2(1+\xi(K))\frac{\xi(K)}{\gamma(K)}}}{2(1+\xi(K))} |Y(K)| \tag{3.35}$$

The MAP and MMSE estimates are nearly same for high *a prior*i and *a posteriori* SNRs. The MAP phase estimate gives the noisy phase, which also happens to the MMSE phase estimate. Also, the MAP estimator gives simple computation compared to MMSE.

Table 3.2 summarizes the gain function (suppression function) of various STSA methods. In all the spectral subtraction methods the spectral floor can be set as per equation 3.18 with different values of parameters α, β and p. A noise pre-processor based on STSA has been developed by Motorola for enhanced variable rate codec (EVRC) being used in CDMA based telephone systems. In this pre-processor the input speech spectrum is divided into 16 non-uniform, non-overlapping bands similar to MBSS where input speech spectrum is divided into 3 bands. The speech is enhanced by using a gain function similar to MMSE based methods to each band. The VAD used to decide speech/silence frame and noise estimation is embedded within the algorithm. The sub-modules of EVRC noise pre-processor are optimized and highly inter-dependent.

Sr. No.	Class	Method	Gain (suppression or attenuation) function H(K)	Remarks
1	Spectral subtraction	1.MSS	$\frac{\sqrt{\gamma(K)}-1}{\sqrt{\gamma(K)}}$	Simple High Residual noise
		2.PSS	$\sqrt{\frac{\gamma(K)-1}{\gamma(K)}}$	Simple Musical noise artifact
		3.GSS	$\frac{(\sqrt{\gamma(K)}^{p}-1)^{1/p}}{\sqrt{\gamma(K)}}$	Flexible Musical and residual noise trade-off
		4.BSS	$\sqrt{\frac{\gamma(K)-\alpha}{\gamma(K)}}$	Simple Less musical noise High Residual noise
2	Wiener	1.Scalart	$\frac{\xi(K)}{1+\xi(K)}$	Non-causal
		2.Para-metric	$\left(\frac{\xi(K)}{\mu+\xi(K)}\right)^{v}$	Non-causal but flexible
3	Statistical modeling	1.ML	$0.5+0.5\sqrt{\frac{\gamma(K)-1}{\gamma(K)}}$	Less attenuation Not preferred High musical noise
		2.MMSE -SA	$\frac{\sqrt{\pi}}{2}\frac{\sqrt{v(K)}}{\gamma(K)}e^{-\frac{v(K)}{2}}\left[(1+v(K))I_0\left(\frac{v(K)}{2}\right)+v(K)I_1\left(\frac{v(K)}{2}\right)\right]$	Complicated Less musical and residual noise but Speech distortion
		3.MMSE -LSA	$\frac{\xi(K)}{1+\xi(K)}\exp\left(-\frac{1}{2}\int_{v(K)}^{\infty}\frac{e^{-t}}{t}dt\right)$	Complicated Less musical and residual noise with less speech distortion
		4.MAP	$\frac{\xi(K)+\sqrt{\xi(K)^2+2(1+\xi(K))\frac{\xi(K)}{\gamma(K)}}}{2(1+\xi(K))}$	Simple Alternate to MMSE

Table 3.2 A summary of STSA methods

3.6 Voice Activity Detection (VAD) and Noise Estimation

In speech communications, speech can be characterized as a discontinuous medium because of the pauses which are a unique feature compared to other multimedia signals, such as video, audio and data. The regions where voice information exists are classified as voice-active and the pauses between talk spurts are called voice-inactive or silence regions. An example illustrating active and inactive voice regions for a speech signal is shown in figure 3.9. A voice activity detector (VAD) is an algorithm employed to detect the active and inactive regions of speech.

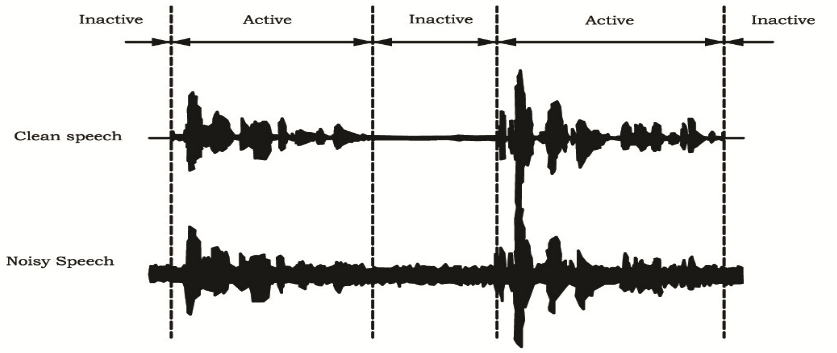


Fig. 3.9 Voice active and inactive regions

A practical speech enhancement system consists of two major components, the estimation of noise power spectrum, and the estimation of clean speech. The first part is performed along with voice activity detection (VAD) and second part uses output from first part and apply algorithm for clean speech estimation. Therefore, a critical component of any frequency domain enhancement algorithm is the estimation of the noise power spectrum [19]. The basic VAD and noise estimation operation is described in figure 3.10.

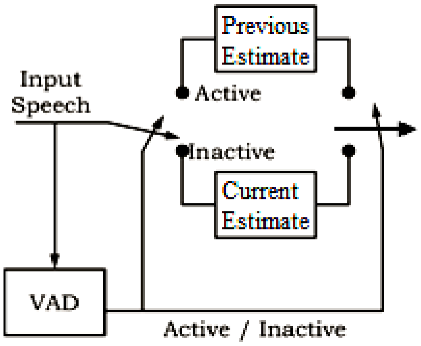


Fig. 3.10 Block diagram of VAD and noise estimation

The speech/silence detection finds out the frames of the noisy speech that contain only noise. Speech pauses or noise only, frames are essential to estimate noise. If the speech/silence detection is not accurate then speech echoes and residual noise tend to be present in the enhanced speech. Several methods are used for VAD, such as voiced/unvoiced classification used in ITU G.723.1, zero crossing method used in G.729, and spectral comparison used in both G.729 and

GSM vocoders in addition to different power thresholds variations. However they are suitable for clean speech only. For speech enhancement it is required to operate with noisy speech and hence the magnitude spectral distance VAD which is generic, simple and easy to integrate with speech enhancement algorithm is most common in applications. In [20] it is reported that this VAD is most suitable for real time implementation.

Let $|Y(K)|$ is the current frames magnitude spectrum, which is to be labeled as noise or speech, N is noise magnitude spectrum template (estimation), NC is noise counter which reflects the number of immediate previous noise frames, NM is noise margin and it is spectral distance threshold. Hangover counter is the number of noise segments after which the "Speech flag" resets (goes to zero). "Noise flag" is set to one if the segment is labeled as noise. Spectral distance is calculated by using following formula and based on this the decision is taken.

$$\begin{gathered} Spectral\ distance\ = log_{10}(|Y(K)|) - log_{10}(N) \\ If\ Spectral\ distance < NM\!: Noise\ flag = 1, NC = NC + 1 \\ Else\!:\ Noise\ flag = 0, NC = 0 \\ If\ NC > Hangover\ counter\!: Speech\ flag = 0 \\ Else\!: Speech\ flag = 1 \end{gathered} \tag{3.36}$$

3.7 Speech Enhancement Using Wavelet Transform

The STFT allows representing the signal in frequency domain through time windowing function. The window length determines a constant time and frequency resolution. Thus, a shorter time windowing is used in order to capture the transient behavior of a signal at the cost of frequency resolution. The nature of the speech signals is quasi-stationary; such signals cannot easily be analyzed by conventional transforms. So, an alternative mathematical tool- wavelet transform should be selected to extract the relevant time amplitude information from a signal. In this thesis, only some key equations and concepts of wavelet transform are stated, more rigorous mathematical treatment of this subject can be found in [21]. A continuous time wavelet transform (CWT) of signal $x(t)$ is defined as:

$$X_w(\tau, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) {h_{\tau,a}}^*(t) dt\,; \quad h_{\tau,a}(t) = \frac{1}{\sqrt{a}} h\left(\frac{t-\tau}{a}\right) \tag{3.37}$$

Here $a\ (Scaling\ factor), \tau (Time\ shift) \in R, a \neq 0$ and they are dilating and translating coefficients, respectively. This multiplication of $\frac{1}{\sqrt{a}}$ is for energy normalization purposes so that the transformed signal will have the same energy at every scale. The analysis function $h(t)$, the so-called mother wavelet (basic or prototype wavelet), is scaled by a, so a wavelet analysis is often called a time-scale analysis rather than a time-frequency analysis. The wavelet transform decomposes the signal into different scales with different levels of resolution by dilating a single prototype function, the mother wavelet. Furthermore, a mother wavelet has to satisfy that it has a zero net area, which suggest that the transformation kernel of the wavelet transform is a compactly support function (localized in time), thereby offering the potential to capture the transients [21].

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. It turns out, rather remarkably, that if scales and positions are based on powers of two; so-called *dyadic* scales and positions; then the analysis will be much more efficient and just as accurate. Such an analysis forms the discrete wavelet transform (DWT) of discrete time signal $x(n)$.

$$a = 2^m \quad and \quad \tau = n2^m;\ m, n \in N \tag{3.38}$$

$$X_w(n, m) = \sum_{p=-\infty}^{\infty} x(p) h_{n,m}{}^{*}(p) \tag{3.39}$$

The family of dilated mother wavelets of selected $a\ and\ \tau$ constitute an orthonormal basis of $L^2(R)$. In addition, sampling of $X_w(\tau, a)$ in dyadic grid also called dyadic orthonormal wavelet transform. Due to the orthonormal properties, there is no information redundancy in the DWT. In addition, with this choice of $a\ and\ \tau$, there exists the multi-resolution analysis (MRA) algorithm, which decomposes a signal into scales with different time and frequency resolution. MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. The discrete time dyadic wavelet transform can be efficiently implemented by using filter banks. The filtering implementation of the forward transform is given by an iterative cascade of identical stages, each stage consisting of low pass and high pass decomposition of the signal followed by the 2 to 1 down-sampling. A similar iterative structure can be used for inverting the wavelet transform from the wavelet coefficients. Further details can be obtained from [21].

The differences between different mother wavelet functions (e.g., Haar, Daubechies, Coiflets, Symlet, Biorthogonal and etc.) consist in how scaling signals and the wavelets are defined. The choice of wavelet determines the final waveform shape; likewise, for Fourier transform, the decomposed waveforms are always sinusoid. To have a unique reconstructed signal from wavelet transform, it is needed to select the orthogonal wavelets to perform the transforms.

3.7.1 Thresholding of Wavelet Co-efficients for Speech Enhancement

One of the first wavelet-based methods de-noising was developed by Donoho and Johnstone [22-23]. It reduces noise by thresholding the wavelet coefficients so that only the coefficient with values above the threshold are retained. Signal energy is concentrated on a small number of wavelet coefficients in many signals; while wavelet coefficients of noise are spread over a wide number of coefficients. Appropriate thresholding of wavelet coefficients can lead to high noise reduction with low signal distortion. The general wavelet de-nosing procedure is as follows:

- Apply DWT to the noisy signal to produce the noisy wavelet coefficients to the level.
- Select appropriate threshold limit at each level and threshold method to best remove the noises.
- Inverse DWT of the thresholded wavelet coefficients to obtain a de-noised signal.

Performing the DWT of equation 3.11

$$Y_{j,k} = X_{j,k} + D_{j,k} \tag{3.40}$$

where $Y_{j,k}$ is the k^{th} wavelet coefficient in the scale j. There are two common ways to threshold (λ) the resulting wavelet coefficients. The first is referred to as hard thresholding which sets the coefficients to zero whose absolute value is below the threshold.

$$\hat{Y}_{j,k} = \begin{cases} Y_{j,k} \ if \ |Y_{j,k}| > \lambda \\ 0 \ else. \end{cases} \tag{3.41}$$

Soft thresholding goes one step further and decreases the magnitude of the remaining coefficients by the threshold value

$$\hat{Y}_{j,k}^{soft} = sign\,(Y_{j,k})\,max(|Y_{j,k}| - \lambda, 0) \tag{3.42}$$

Hard thresholding maintains the scale of the signal but introduces ringing and artifacts after reconstruction due to a discontinuity in the wavelet coefficients. Soft thresholding eliminates this discontinuity resulting in smoother signals slightly decreases the magnitude of the reconstructed signal.

Many methods for setting the threshold have been proposed. The most time-consuming way is to set the threshold limit on a case-by-case basis. The limit is selected such that satisfactory noise removal is achieved. For a Gaussian noise if orthogonal wavelet transform is applied to the noise signal, the transformed signal will preserve the Gaussian nature of the noise, which the histogram of the noise will be a symmetrical bell-shaped curve about its mean value. To obtain the threshold value for a signal of length $d,$ the approach in [22] seeks to minimize the maximum error over all possible samples. This method assumes that $d(t)$ having some known standard deviation σ. The universal threshold is given by

$$\lambda_{uni} = \sigma\sqrt{2\log{(d)}} \tag{3.43}$$

is shown to be asymptotically optimal in the minimax sense when employed as a hard threshold with $\sigma = MAD/0.6745,$ where MAD represents the absolute median estimated on the first scale Y_{HH_1}. Donoho and Johnstone [23] also proposed a more advanced strategy based on Stein's unbiased risk estimate (SURE). Here, soft thresholding is used because it is more mathematically tractable (i.e., continuous) and the clean signal is estimated as

$$\lambda_{SURE} = argmin_{0\le\lambda\le\sqrt{2log\;(d)}}SURE(\lambda, Y_j) \tag{3.44}$$

$$where;\; SURE(\lambda, Y_j) \;= \sigma^2 + \frac{1}{c}\sum_{k=1}^{c}[min\;(|Y_{j,k}|, \lambda]^2 - \frac{2\sigma^2}{c}\sum_{k=1}^{c}I(|Y_{j,k}| < \lambda)$$

Johnstone and Silverman [24] studied the correlated noise situation and proposed a "level-dependent" threshold

$$\lambda_j = \sigma_j\sqrt{2\log{(d_j)}} \tag{3.45}$$

with $\sigma = MAD/0.6745,$ and d_j is the number of samples in scale $j.$

During the past decade, the wavelet transforms have been applied to various research areas. Their applications include signal and image de-noising, compression, detection, and pattern recognition. To the best of knowledge, de-noising methods based on the wavelet thresholding have not been successfully applied to speech enhancement. The difficulties are simultaneously associated to the speech signal complexity and to the nature of the noise.

However, to improve the wavelet thresholding enhancement, following suggestions are proposed [25-27]:

- The use of the wavelet packet transform (WPT) instead of the wavelet transform,
- To extend the concept of the level-dependent threshold (Equation 3.45) to the WPT,
- The use of time-adapted threshold based on the speech waveform energy.

As a result the wavelet based techniques are ruled out here for further refinements. It is considered in next chapter only for comparison with the STSA based techniques.

3.8 Objective Quality Measures for Speech Enhancement Methods

Quality is one of many attributes of the speech signal. Quality is highly subjective in nature and it is difficult to evaluate reliably. This is partly because individual listeners have different internal standards of what constitutes "good" or "poor" quality, resulting in large variability in rating scores among listeners. Quality measures assess 'how' a speaker produces an utterance, and includes attributes such as "natural", "raspy", "hoarse", "scratchy", and so on. Quality possesses many dimensions, too many to enumerate. For practical purposes it is restricted to only a few dimensions of speech quality, depending on the application.

Intelligibility measures assess "what" the speaker said, i.e., the meaning or the content of the spoken words. Unlike quality, intelligibility is not subjective and can be easily measured by presenting to a group of listeners speech material (sentences, words, etc.) and asking them to identify the words spoken. Intelligibility is quantified by counting the number of words or phonemes identified correctly. The relationship between speech intelligibility and speech quality is not fully understood, and this is in part because no one has yet identified the acoustic correlates of quality and intelligibility [28]. A good speech enhancement algorithm needs to preserve or enhance not only speech intelligibility but also speech quality. This is based on the observation that it is possible for speech to be both highly intelligible and of poor quality. Also, although two different algorithms may produce equal word intelligibility scores, listeners may perceive the speech of one of the two algorithms as being more natural, pleasant, and acceptable. There is, therefore, the need to measure other attributes of the speech signal besides intelligibility. Reliable evaluation of speech quality is considered to be a much more challenging task than the task of evaluating speech intelligibility

Quality assessment of speech enhancement algorithms can be done using subjective listening tests or objective quality measures. Subjective listening tests uses mean opinion score

(MOS) to evaluate the performance of speech enhancement algorithms [17]. But they are time consuming, expensive, involve human subjects, not easily repeatable and rating is based on their overall perception (possess inherent variability in interpretation). A consistent listening environment is required and perceived distortion can vary with factors such as the playback volume and type of listening instrument used. For provisional investigations objective quality measures can be used. Objective evaluation involves a mathematical comparison of the original and processed speech signals. Objective measures quantify quality by measuring the numerical "distance" between the original and processed signals. Clearly, for the objective measures to be valid, it needs to correlate well with subjective listening tests, and for that reason much research has been focused on developing objective measures that model various aspects of the auditory system [29].

Objective measures of speech quality are implemented by first segmenting the speech signal into 10-30 ms frames and then computing a distortion measure between the original and processes signals. A single, global measure of speech distortion is computed by averaging the distortion measures of each speech frame. A large number of objective measures have been evaluated, particularly for speech coding applications. Reviews of objective measures can be found in [30]. The focus here is on a subset of those measures that have been found to useful for evaluation of speech enhancement algorithms [29].The STSA and wavelet based algorithms are compared using several objective measures and results are shown in chapter 4. In addition, the MOS subjective measure is also used to compare modified and proposed method with existing algorithms and it is described in chapter 6. A final comment on the quality of the enhanced speech can be made only after referring to both the objective measures and subjective test. Figure 3.11 illustrates the typical system setup.

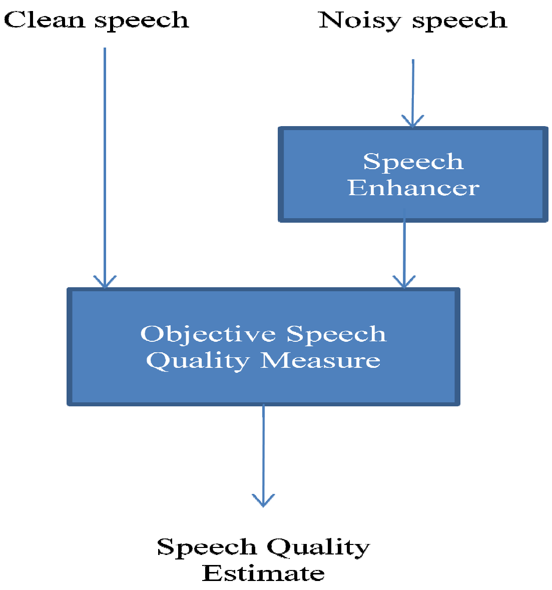


Fig. 3.11 Objective speech quality measuring system

Table 3.3 presents a brief summary of important objective measures used for speech quality assessments.

Sr. No.	Objective measure	Mathematical relation	Terminology and significance
1	Segmental SNR (SSNR) [31]	$\frac{10}{M}\sum_{m=0}^{M-1} log_{10}($ $\frac{\sum_{n=Nm}^{Nm+N-1} x^2(n)}{\sum_{n=Nm}^{Nm+N-1}(x(n)-\hat{x}(n))^2})$	$x(n)$ is the original (clean) signal, $\hat{x}(n)$ is the enhanced signal, N is the frame length (typically chosen to be 15-20 ms), and M is the number of frames in the signal. It is based on the geometric mean of the SNRs across all frames of the speech signal.
2	Log Likelihood Ratio Distance (LLR) [4]	$d(a_x, \bar{a}_{\hat{x}}) = log\frac{\bar{a}_{\hat{x}}^T\, R_x\, \bar{a}_{\hat{x}}}{a_x^T R_x a_x}$	$a_x^T =$ $[1, -\alpha_x(1), -\alpha_x(2), \dots, -\alpha_x(p)]$ are the LPC coefficient of the clean signal, $\bar{a}_{\hat{x}}^T =$ $[1, -\alpha_{\hat{x}}(1), -\alpha_{\hat{x}}(2), \dots, -\alpha_{\hat{x}}(p)]$ are the coefficients of the enhanced signals, and R_x is the $(p+1) \times (p+1)$ autocorrelation matrix (Toeplitz) of the clean signal. It is based on the dissimilarity between all-pole models of the clean and enhanced signals.
3	Weighted Spectral Slope Distance (WSS) [32-34]	$d_{WSS}(C_x, \bar{C}_x) = \sum_{k=1}^{L} W(k)(S_x(k)$ $- \bar{S}_{\hat{x}}(k))^2$ $S_x(k) = C_x(k+1) - C_x(k)$ $\bar{S}_{\hat{x}}(k) = \bar{C}_{\hat{x}}(k+1) - \bar{C}_{\hat{x}}(k)$	$C_x(k)$ is clean and $C_{\hat{x}}(k)$ is enhanced critical-band spectra expressed in dB, $W(k)$ is weight for band k, L is the number of critical bands. It is based on phonetic distance. Thirty six overlapping filter of progressively larger bandwidths to estimate the smoothed short time speech spectrum every 12 ms are used. The filter bandwidths approximate auditory critical bands so as to give equal perceptual weight to each band.
4	Perceptual Evaluation of Speech Quality (PSEQ) [35]	The process is described by block diagram in figure 3.12.	It closely resembles to the subjective MOS measure. The range of the PESQ score is 0.5 to 4.5.

Table 3.3 Objective measures used for speech quality assessments

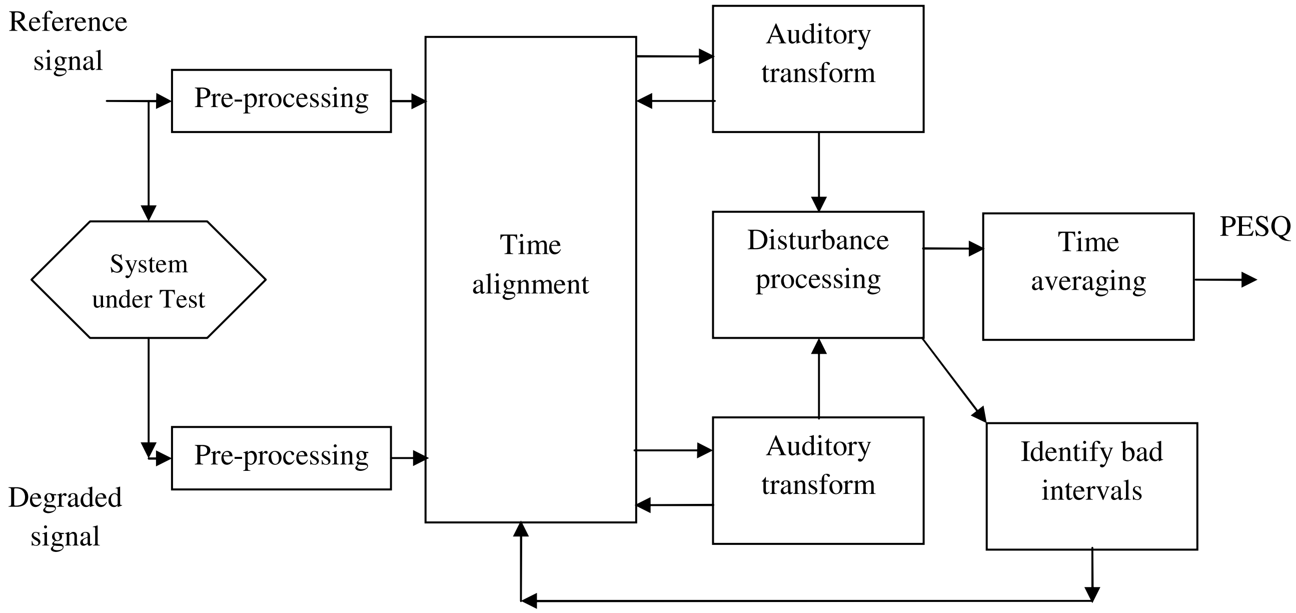


Fig. 3.12 Block diagram of PESQ measure computation

3.9 Summary

The transform domain techniques particularly STSA techniques are frequent in speech enhancement and they are discussed in detail. They are characterized by their gain function. The gain function requires computation of a *posteriori* and/or *a priori* SNR. The frame by frame processing using decision direct rule allows the computation of both SNRs. The gain function depicts the complexity of computation. The MMSE-STSA85 (LSA) method has complex gain function but provides good resistance against musical noise. The amount of speech distortion perceived is also reduced. So it is preferred in practical applications. The wavelet based transform domain techniques are also touched here. The de-noising is done by using thresholding of wavelet co-efficients. There is no optimized way for thresholding and hence they are still inferior in comparison to STSA techniques. The objective quality measures SSNR, LLR, WSS and PESQ are used to assess the effectiveness of speech enhancement algorithms. In next chapter the simulation and objective evaluation results of these techniques are presented.

Chapter 4

MATLAB Implementation and Performance Evaluation of Transform Domain Methods

The simulation work is carried out to understand the functionality and behavior of all transform domain methods explained in chapter 3. An automatic VAD based on magnitude spectral distance proposed in [1] is integrated with STSA methods. The MATLAB simulation work is concreted and converged by preparing a MATLAB GUI (Graphical User Interface). This GUI can be used to simulate any transform domain algorithm for different noise conditions. The performance comparisons of various methods based on spectrographic analysis and objective tests are reported in this chapter. Various objective measures are available to evaluate speech enhancement techniques and they are described in brief here. The IEEE standard database NOIZEUS (noisy corpus) is used to test algorithms [18]. The database contains clean speech sample files as well as real world noisy speech files at different SNRs and noise conditions like airport, car, restaurant, train, station etc. The GUI also includes evaluation of algorithms using objective measures. The basic wavelet de-noising methods are also implemented in MATLAB and objective measures are obtained and compared with the STSA methods. The limitations and present implementations of these methods are also mentioned.

4.1 MATLAB Implementation -STSA Techniques

Eight important STSA algorithms viz. magnitude spectral subtraction (MSS) proposed by Boll [2], power spectral subtraction (PSS) proposed by Boll [2], Berouti spectral subtraction (BSS) proposed by Berouti [3], multi-band spectral subtraction (MBSS) proposed by Kamath [4], Wiener Scalart (WS) proposed by Scalart [5], maximum likelihood (ML) proposed by McAulay and Malpass [6], minimum mean square error with spectral amplitude (MMSE-SA) and minimum mean square error with log spectral amplitude (MMSE-LSA) proposed by Ephrahim and Malah [7,8] are simulated in the MATLAB environment. The sampling rate of the speech signal used in all the experiments carried out here is 8 KHz. The Hamming window of 25ms (200 samples) with 40% (10ms) overlap is selected. The FFT and IFFT are calculated using 256 points radix-2 algorithms. The general flow chart is shown in figure 4.1.

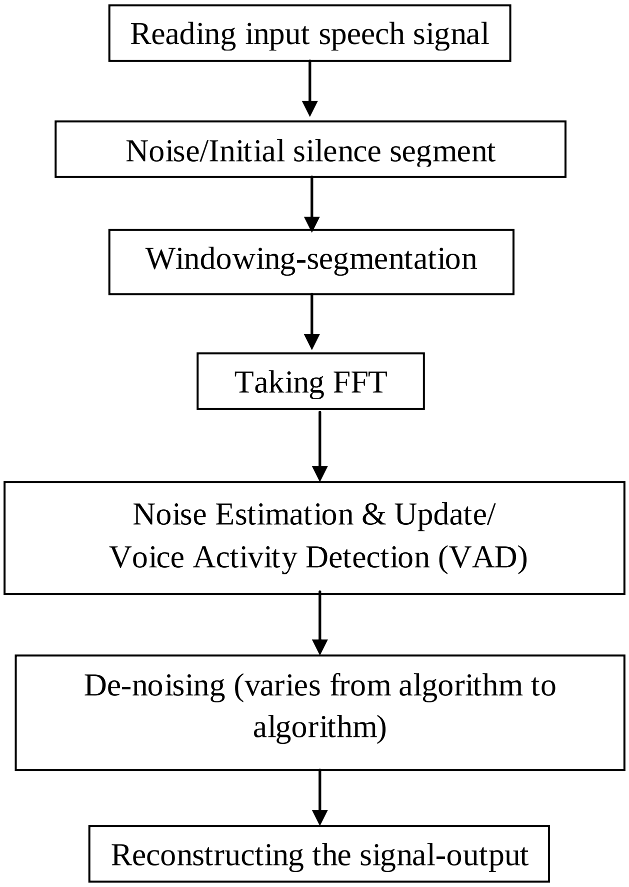

Reading input speech signal
Noise/Initial silence segment
Windowing-segmentation
Taking FFT
Noise Estimation & Update/
Voice Activity Detection (VAD)
De-noising (varies from algorithm to algorithm)
Reconstructing the signal-output

Fig. 4.1 A flow chart showing general implementation of STSA algorithms

The MSS and PSS methods described by equations 3.13, 3.14 and 3.15 and block diagram in fig. 3.7 are implemented with some additional features suggested in literature to improve the performance. The spectral magnitude is averaged over three successive frames (one past, current and one next) before applying the spectral subtraction. This will smooth the spectrum and helps to reduce the musical noise in the enhanced speech [1]. Before applying the half wave rectification to the speech frames; the residual noise reduction is applied by considering minimum spectral component from the minimum of three: the current clean estimate of spectral component, past frame noisy smoothed spectral component and next frame noisy smoothed spectral component [1]. For non-speech (silence) frame the spectral floor is applied to maintain the floor noise in the enhanced speech which will reduce the listener fatigue. The flow chart is shown in figure 4.2.

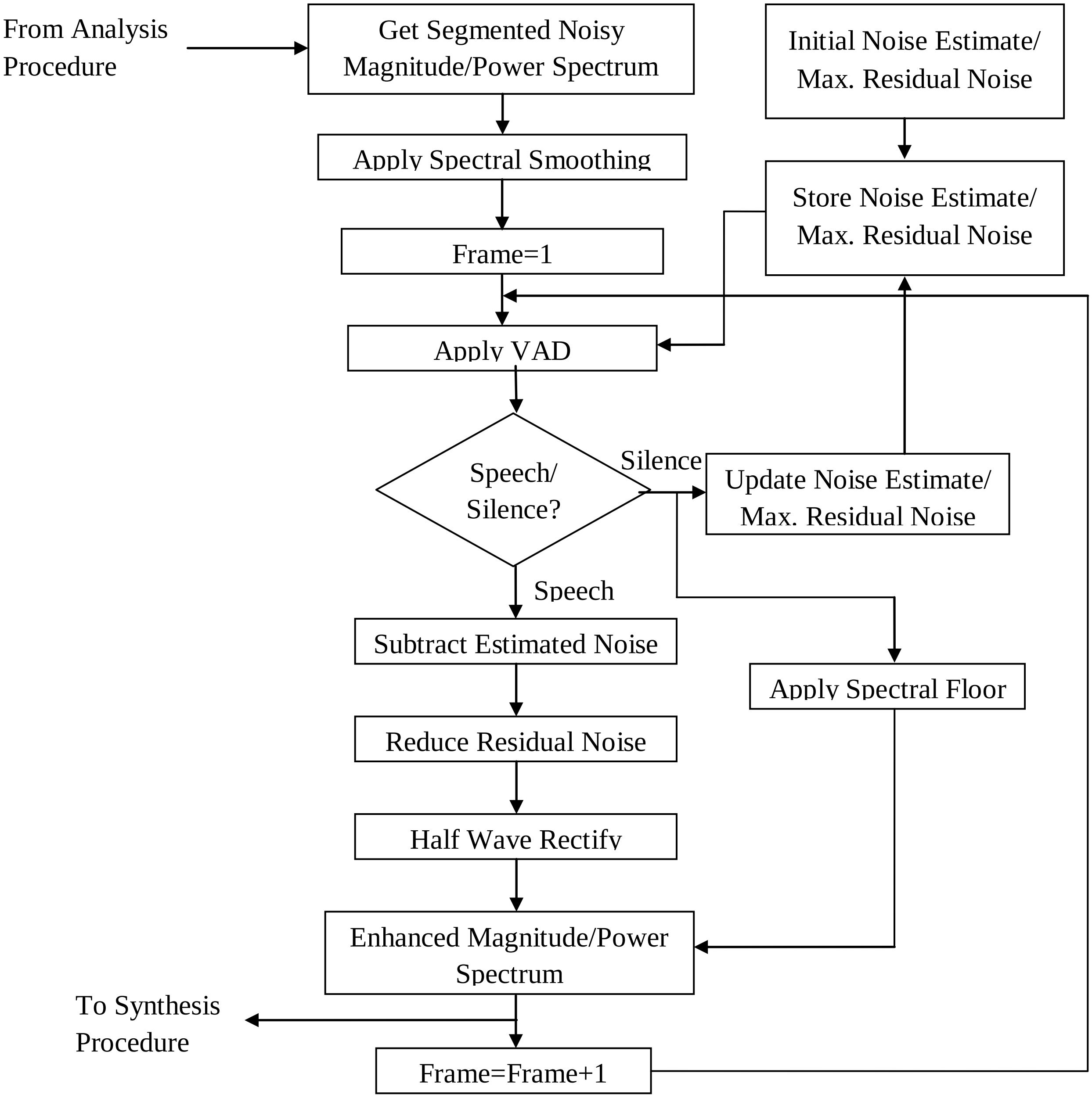


Fig. 4.2 Flow chart for MSS and PSS implementation

The BSS represents general spectral subtraction and its implementation is described by the block diagram shown in figure 4.3. The spectral floor parameter β is taken 0.03. The value of over subtraction factor α is adapted according to SNR values as per equation 3.18. The value of parameters α_0 and *s* is set considering the SNR varies between -5 to 20dB and the value of α lies between 1 and 3. These parameter settings are subjectively found optimal values for wide range of SNR values, except for very low SNR values below 0dB [3].

The MBSS method is described by figure 3.8. For simplification in implementation and comparison with other spectral subtraction algorithms the parameters α and β are set to same values as described earlier. The parameter δ is set to the originally prescribed values [4] for different frequency bands.

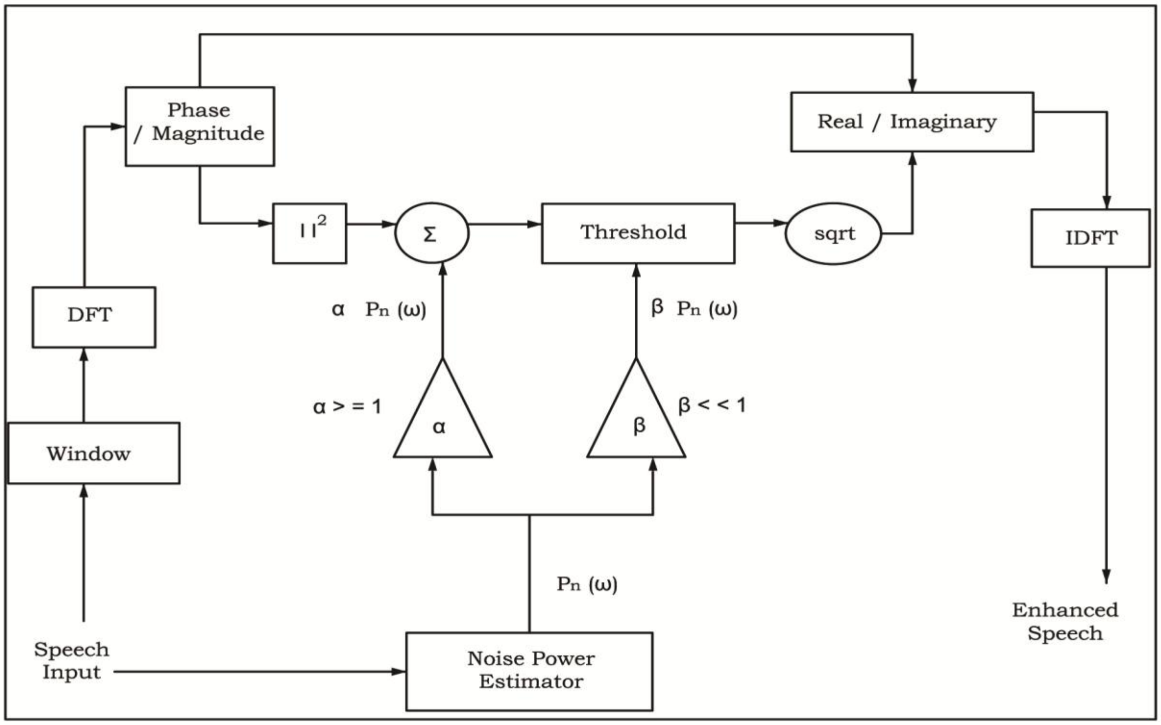


Fig. 4.3 A block diagram for BSS implementation

The Wiener filter implementation described by equation 3.27 is non-causal as it requires evaluating the *a priori* SNR. The *a priori* SNR is estimated using decision direct rule described by equation 3.28 which estimate it from *a posteriori* SNR of previous and current frames. The optimum value of smoothing constant η is taken 0.99 [6]. For first frame the *a posteriori* SNR is assumed unity and which is obvious.

The implementation of ML method is similar to that of MSS and PSS except the spectral subtraction equation is replaced by equation 3.30. For implementation of MMSE SA and LSA methods equation 3.31 and 3.32 are used along with decision directed rule to estimate *a priori* SNR.

In the implementation of all the above algorithms initial silence period of around 0.25

second (10 frames) in recorded speech is assumed. From this the initial noise estimate is derived by computing the mean (average) value of spectral components $|\widehat{D}(K)|$ and the variance of the spectral components $\lambda_D(K)$ during the initial silence period. These two are updated in every non-speech frame detected by VAD. One more parameter called noise smoothing factor σ is used during updating noise estimate and update. It is initialized to 9 for optimum smoothing [1]. The noise estimate update for frame *t* is described by following equations.

$$|\widehat{D^t}(K)| = \frac{\sigma|\widehat{D^{(t-1)}}(K)| + |Y^t(K)|}{\sigma + 1} \tag{4.1}$$

$$\lambda_D{}^t(K) = \frac{\sigma\lambda_D{}^{(t-1)}(K) + |Y^t(K)|^2}{\sigma + 1} \tag{4.2}$$

The VAD used is magnitude spectral distance type. It operates on a framed data. The terms involved here are explained as follows.

"Signal" is the current frame's magnitude spectrum and it is input to VAD, which is to be labeled as noise or speech. "Noise" is noise magnitude spectrum template (estimation), "noise counter" is the number of immediate previous noise frames, "noise margin" (default 3) is the spectral distance threshold. The noise margin is fixed to 3, which is the threshold value for comparison with the SNR of the current frame. "Hangover" (default 8) is the number of noise segments after which the "Speech flag" is reset (goes to zero). "Noise flag" is set to one if the segment is labeled as noise. "Dist" is the mean spectral distance. Spectral distance is calculated by using the SNR formula

$$Spectral\ distance\ = log_{10}(signal) - log_{10}(noise) \tag{4.3}$$

Mean of this spectral distance value is the "Dist" value. This "Dist" value is the real SNR value of the current frame. This value is compared with the noise margin and if this value is lesser than noise margin then the "Noise flag" is said to one and "Noise counter" is incremented by one.

If the "Dist" value is greater than the "Noise Margin" then the "Noise flag" is set to zero and the noise counter is reset (i.e., zero). If the "Noise counter" value is greater than the "Hangover period" then the speech flag is reset to zero and if vice versa then the "Speech Flag" is set (i.e., one). Its implementation is shown in figure 4.4 by means of a flow chart.

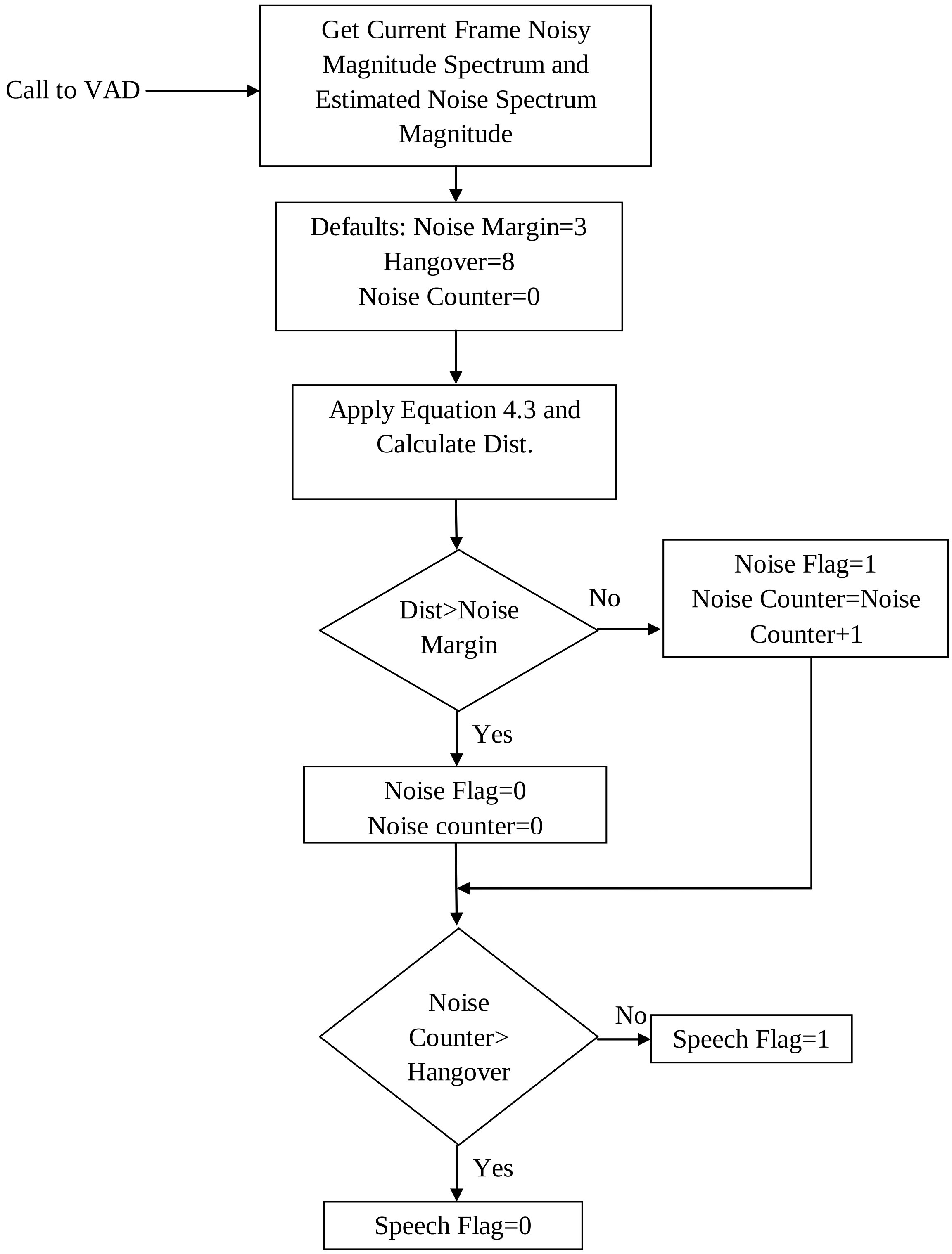


Fig. 4.4 Flow chart for magnitude spectral distance VAD implementation

4.2 Spectrographic Results of Simulation

The spectrograms of enhanced speech from the enhancement methods under comparison are plotted in figure 4.5 [9, 10], in order to compare the noise suppression capabilities based on presence of residual and musical noise in the enhanced speech. The spectrogram used for comparison is narrowband spectrogram obtained using Hamming window of 32ms (256 points) with 50% overlap and 256 point DFT. Figure 4.5(top panel) and (bottom panel) shows the spectrograms of clean and noisy speech sentence corrupted by 0dB white noise respectively. Figure 4.6 shows the spectrograms of enhanced speech by various algorithms as indicated.

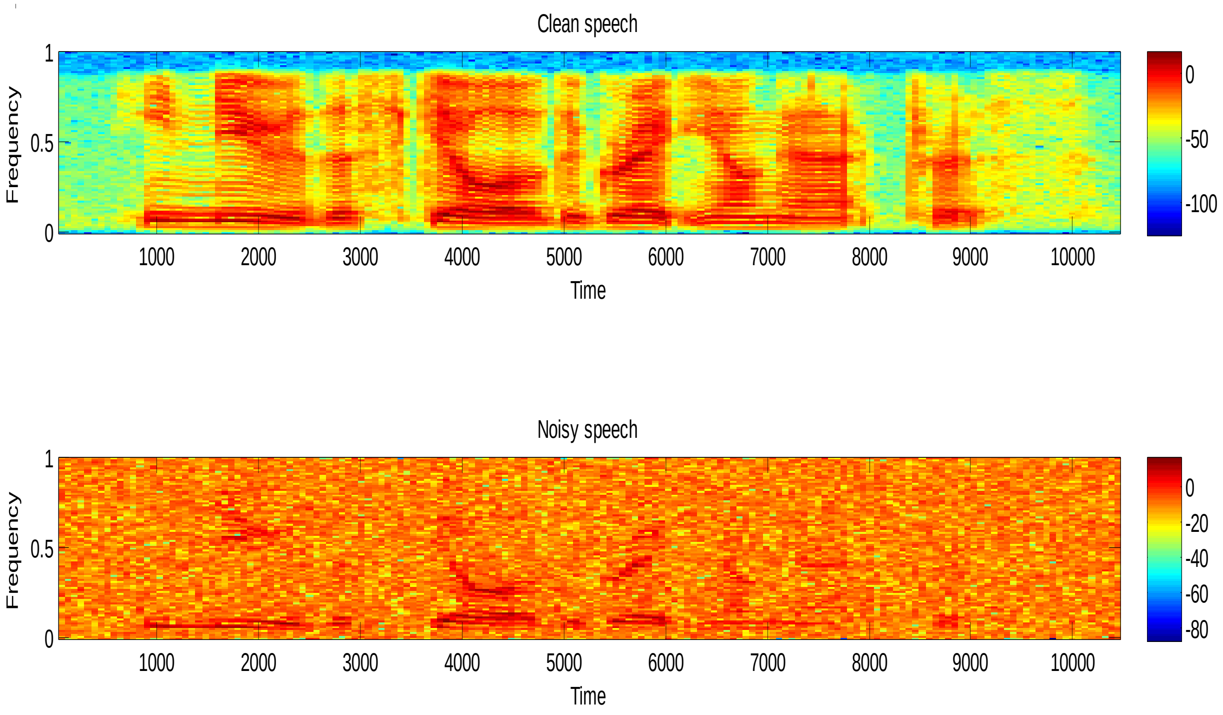
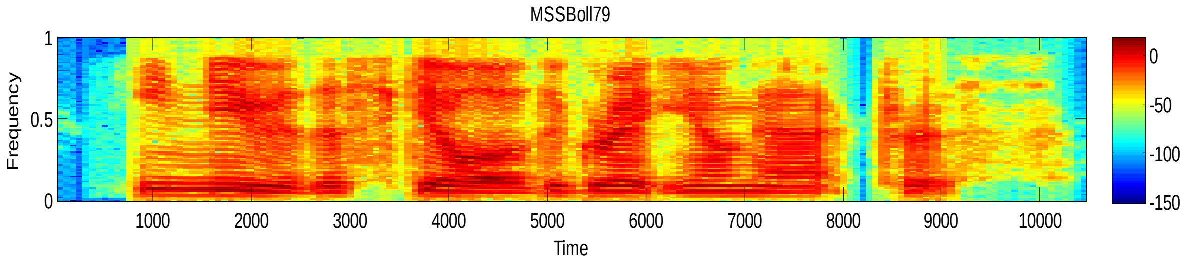
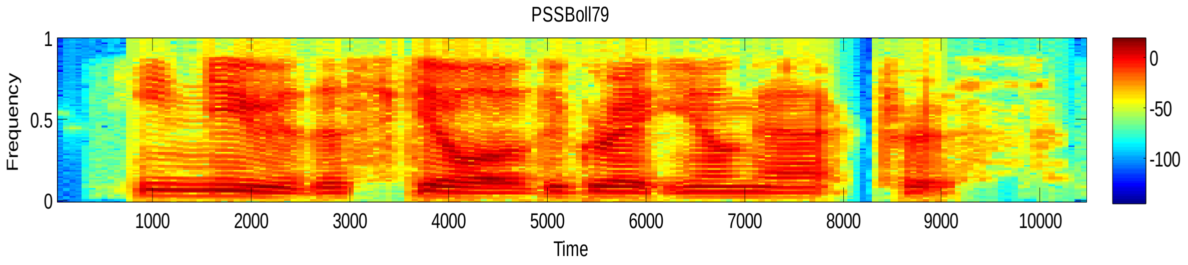
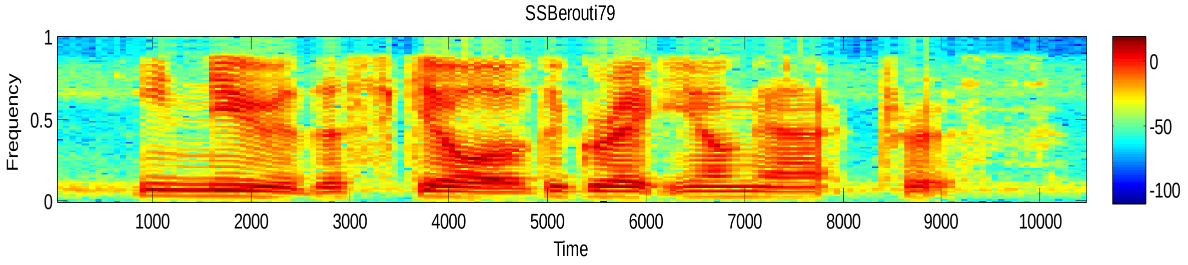
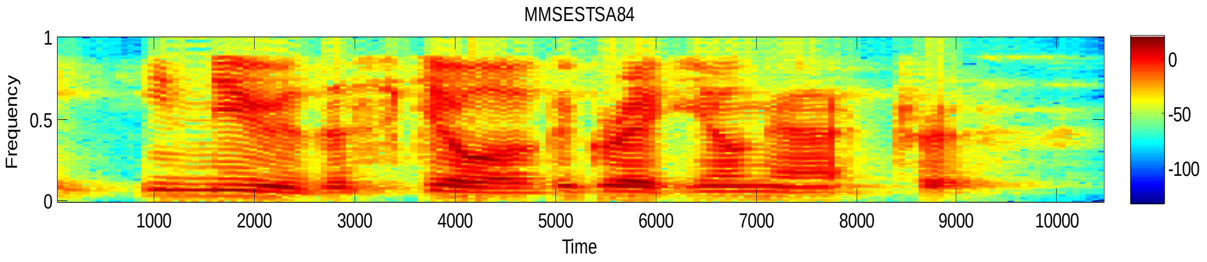


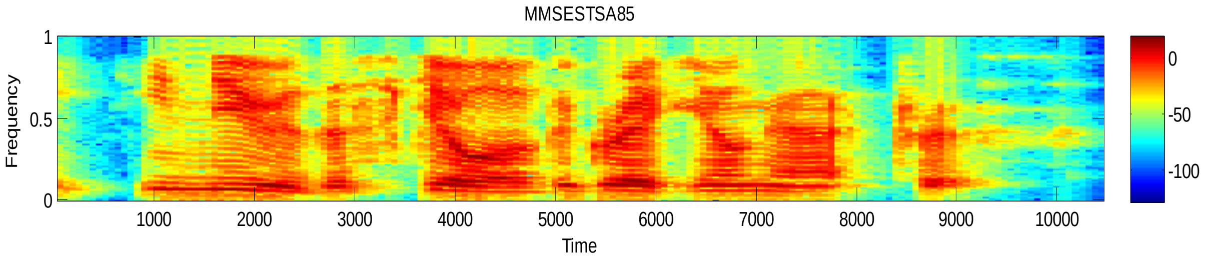
Fig. 4.5 Spectrogram of clean speech signal containing sentence ‘He knew the skill of the great young actress’ (top panel) and spectrogram of the signal subjected to 0dB white noise (bottom panel)


MSSBoll79
Frequency
1
0.5
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Time
0
-50
-100
-150


PSSBoll79
Frequency
1
0.5
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Time
0
-50
-100


SSBerouti79
Frequency
1
0.5
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Time
0
-50
-100


MMSESTSA84
Frequency
1
0.5
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Time
0
-50
-100


MMSESTSA85
Frequency
1
0.5
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Time
0
-50
-100

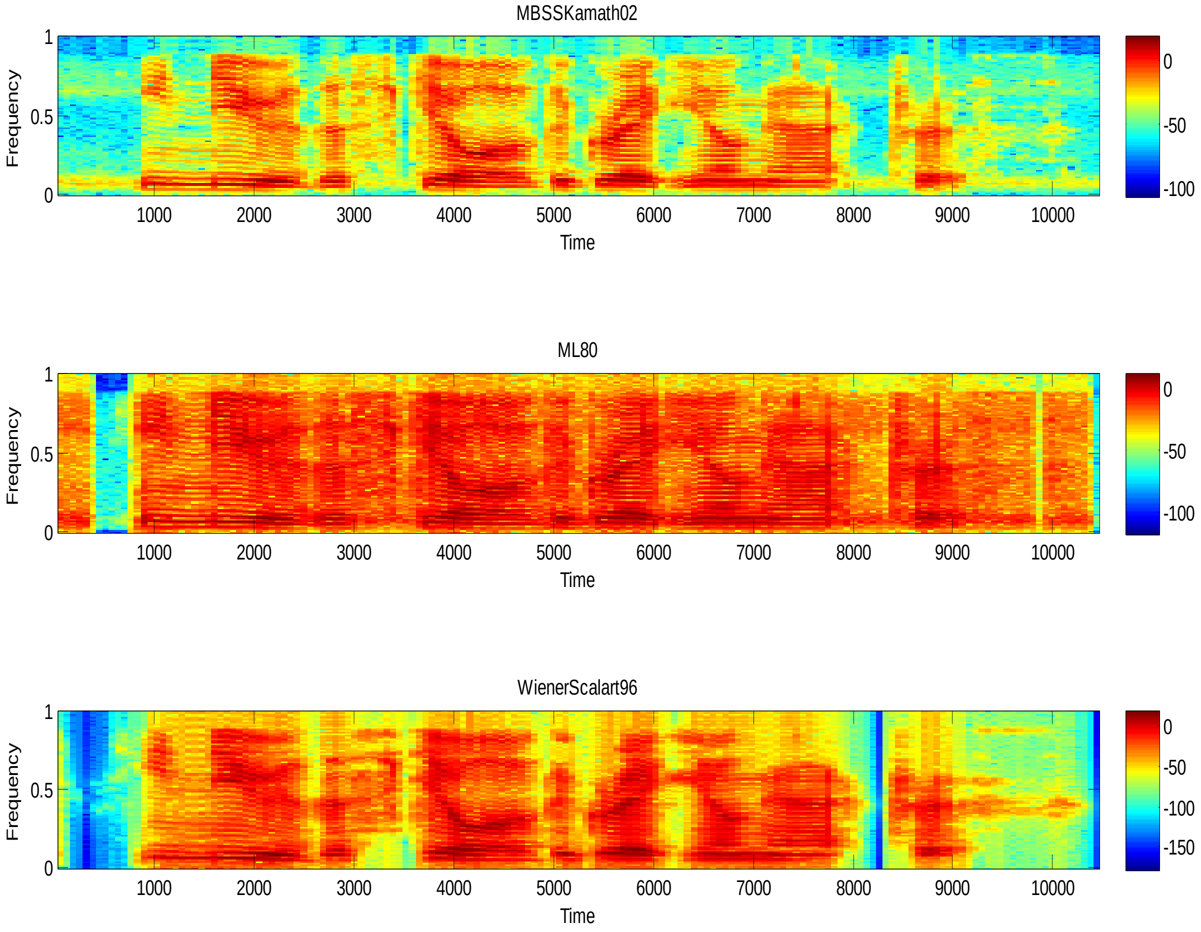


Fig. 4.6 Spectrogram of enhanced speech signal using various algorithms indicated on the head of each panel

The spectrographic analysis shows that the speech enhanced by MSS, PSS and BSS have random dots in the spectrogram compared to MBSS method. The random dots in the spectrogram represent sharp spectral peaks in the enhanced speech and contribute to musical noise. Also if we compare the results with original spectrogram the MBSS is more nearer to original. Hence in spectral subtraction category the MBSS is performing best. In statistical modeling method the ML method is worst while the MMSE-STSA85 (MMSE-LSA) gives best result. The Wiener filter method also gives less random dots but slightly more distortion in spectrogram (results in more residual noise or speech distortion) compared to MBSS and MMSESTSA85 methods. The formal listening also backs the results obtained. The MMSE-LSA method and MBSS methods give optimized performance compared to other methods in terms of residual and musical noise trade off. The MMSE-LSA is found the best from these two from listening point of view. A more useful judgement is obtained using objective measures described in the following section.

4.3 The NOIZEUS Database for Performance Evaluation

NOIZEUS is a noisy speech corpus recorded in Center for Robust Speech Systems, Department of Electrical Engineering, University of Texas at Dallas, Richardson, Texas to facilitate comparison of speech enhancement algorithms among research groups. The noisy database contains 30 IEEE sentences [14] produced by three male and three female speakers (five sentences /speaker), and was corrupted by eight different real-world noises at different SNRs. It is available at [18] and researchers can download it free of cost.

Thirty sentences from the IEEE sentence database were recorded in a sound proof booth using Tucker Davis Technologies (TDT) recording equipment. The IEEE database was selected because it contains phonetically balanced sentences with relatively low word-context predictability. The 30 sentences were selected from the IEEE database so as to include all phonemes in the American English language. The sentences were originally sampled at 25 KHz and down sampled to 8 KHz. A subset of the sentences recorded is given in Table 4.1. To simulate the receiving frequency characteristics of telephone handsets, the speech and noise signals were filtered by the modified Intermediate Reference System (IRS) filters used in ITU-T P.862 [16] for evaluation of the PESQ measure. Noise was artificially added to the speech signal as follows. The IRS filter was independently applied to the clean and noise signals. The active speech level of the filtered clean speech signal was first determined using method B of ITU-T P.56 [17]. A noise segment of the same length as the speech signal was randomly cut out of the noise recordings, appropriately scaled to reach the desired SNR level, and finally added to the filtered clean speech signal.

Filename	Speaker	Gender	Sentence Text
sp01.wav	CH	M	The birch canoe slid on the smooth planks.
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
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	TI	M	Clams are small, round, soft and tasty
sp22.wav	TI	M	The line where the edges join was clean
sp23.wav	TI	M	Stop whistling and watch the boys march
sp24.wav	TI	M	A cruise in warm waters in a sleek yacht is fun
sp25.wav	TI	M	A good book informs of what we ought to know
sp26.wav	SI	F	She has a smart way of wearing clothes
sp27.wav	SI	F	Bring your best compass to the third class
sp28.wav	SI	F	The club rented the rink for the fifth night
sp29.wav	SI	F	The flint sputtered and lit a pine torch
sp30.wav	SI	F	Let us all join as we sing the last chorus

Table 4.1 Sentences from the NOIZEUS speech corpus used in quality evaluation

Noise signals were taken from the AURORA database [15] and included the following recordings from different places: babble (crowd of people), car, exhibition hall, restaurant, street, airport, train station, and train. The noise signals were added to the speech signals at SNRs of 0, 5, 10, and 15dB. The NOIZEUS speech corpus is used in the objective quality evaluation of STSA based speech enhancement algorithms and it is described in the next section.

4.4 Objective Evaluation of STSA Algorithms

Eight STSA algorithms are evaluated using objective measures SSNR, LLR, WSS and

PESQ. The evaluation is done using NOIZEUS database. The MATLAB function that implements and returns the values of the SSNR, LLR, WSS and PESQ is available at [19] and it is widely accepted by researchers for quality evaluation of their speech enhancement algorithms [12,13]. The reason for using the above mentioned code for evaluation is to maintain authenticity, consistency and compatibility with results obtained by other researchers. The measures have been observed over 0-10dB range of SNRs with all eight types of colored noises included in NOIZEUS database. Each algorithm is evaluated here as well as in all future cases on all 30 phonetically balanced speech sentences from NOIZEUS data base corrupted by 3 different SNR values (0, 5 and 10dB) in all 8 colored noise environments. So for one algorithm the number of test runs are 30speech sentences * 3SNRs * 8Noise types = 720. In addition to speech sentences corrupted by colored noise included in the database; a synthesized white noise added to clean speech sentences of NOIZEUS database in 0-10dB SNR range is also used to test the algorithms. This adds another 90 test runs on one algorithm. Hence each algorithm has been tested for total of 810 different conditions. This is sufficient to reflect the real life scenario in which almost all speech communication systems have to work. The results are tabulated in tables 4.2 to 4.10.

AIRPORT NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-1.8033	98.1232	0.944	1.7485	0.1593	80.9384	0.7754	2.1597	2.1429	64.6463	0.6044	2.5111
PSSBoll79	-3.1179	82.7364	0.8976	1.663	-1.7308	69.4178	0.7467	2.1239	-0.0603	57.764	0.5867	2.4847
SSBerouti79	-3.6373	82.9816	0.8967	1.8111	-2.0258	66.2091	0.7119	2.1437	-0.1931	51.4494	0.5261	2.4931
ML80	-3.9171	75.6016	1.0743	1.2804	-3.5451	64.3518	1.0479	1.549	-3.1767	53.5394	1.0235	1.7805
MMSESTSA84	-3.0485	86.4321	0.9334	1.8364	-1.3994	68.1463	0.7522	2.239	0.341	52.4128	0.5691	2.5556
MMSESTSA85	-2.4197	97.1345	1.0146	1.8019	-0.7883	78.9599	0.8287	2.2261	0.99	61.751	0.6462	2.5643
WienerScalart96	-1.4812	123.103	1.2835	1.5979	0.0496	101.768	1.0489	2.0812	1.8138	78.9998	0.8351	2.4489
SSMultibandKamath02	-3.3835	80.3375	0.8987	1.787	-1.7897	64.7142	0.7196	2.1499	0.0325	50.8774	0.5464	2.4947

Table 4.2 Objective quality evaluation with airport noise

CAR NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	0.8033	80.1232	0.944	1.7485	1.2048	67.5299	0.8189	2.1252	3.19	56.7015	0.6279	2.5632
PSSBoll79	-2.1179	72.6456	0.8976	1.663	-1.4238	61.8076	0.8297	2.0597	0.254	52.6771	0.6421	2.4307
SSBerouti79	-2.8973	72.9816	0.8967	1.8111	-1.5631	62.0198	0.7977	2.0808	0.1389	48.8516	0.5826	2.4489
ML80	-2.9171	65.6016	1.0743	1.2804	-2.8386	61.5944	1.0412	1.5508	-2.6228	54.196	1.049	1.7026
MMSESTSA84	-1.0485	66.4321	0.8231	2.0808	-0.647	54.1602	0.7559	2.2373	1.0084	42.6779	0.5838	2.6155
MMSESTSA85	-0.4197	67.5299	0.9052	2.0597	-0.0036	63.0102	0.8231	2.2368	1.7412	50.5242	0.6545	2.653
WienerScalart96	0.1389	103.103	1.2835	1.5979	1.2449	93.088	1.1127	1.9947	2.9627	74.2115	0.9052	2.5361
SSMultibandKamath02	-2.3835	70.3375	0.8987	1.787	-1.3157	57.5471	0.7686	2.0931	0.4256	45.6784	0.579	2.4836

Table 4.3 Objective quality evaluation with car noise

STREET NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-1.351	84.6089	1.0555	1.663	0.2108	68.916	0.8763	2.0632	2.1157	56.1325	0.6477	2.4959
PSSBoll79	-3.027	71.8156	1.0212	1.6219	-1.5639	60.713	0.8514	2.0325	-0.0251	51.0499	0.6398	2.4324
SSBerouti79	-3.3585	75.1916	1.0318	1.7536	-1.7424	60.468	0.8265	2.0736	-0.0168	48.2113	0.599	2.4432
ML80	-3.8069	66.8665	1.1807	1.3986	-3.3169	58.655	1.1216	1.5591	-3.0933	51.687	1.0913	1.7089
MMSESTSA84	-2.6324	72.1776	1.0089	1.8595	-1.1443	58.033	0.8167	2.1747	0.4857	46.73	0.6183	2.5199
MMSESTSA85	-1.9643	84.2573	1.079	1.8572	-0.5892	68.0733	0.8875	2.1911	1.0071	55.3345	0.6774	2.5461
WienerScalart96	-0.972	118.2388	1.3972	1.6009	0.2629	95.8322	1.1828	1.9547	1.7746	74.9996	0.8795	2.4198
SSMultibandKamath02	-3.0104	70.6299	0.9921	1.7235	-1.4438	58.6753	0.7972	2.0858	0.2581	47.3427	0.5989	2.4732

Table 4.4 Objective quality evaluation with street noise

TRAIN NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-1.995	79.6182	1.2669	1.5185	0.2759	65.3372	1.0021	2.024	2.0976	52.8069	0.7394	2.4006
PSSBoll79	-3.5103	71.2996	1.2267	1.6337	-1.7645	59.2825	1.0313	1.985	-0.3135	48.9557	0.7648	2.3338
SSBerouti79	-3.7293	72.7156	1.1909	1.6997	-1.8556	58.9354	0.9815	2.0019	-0.2832	46.9488	0.7073	2.3307
ML80	-4.3353	65.4476	1.3645	1.3839	-3.5039	56.0807	1.2954	1.5387	-3.3991	50.1037	1.1885	1.6934
MMSESTSA84	-2.9603	65.7305	1.172	1.7879	-1.0325	51.5249	0.9158	2.1436	0.4313	41.7044	0.6774	2.4606
MMSESTSA85	-2.2732	75.85	1.2313	1.764	-0.3563	60.3413	0.9752	2.1757	1.1985	48.9939	0.7452	2.5194
WienerScalart96	-1.2984	109.7372	1.5581	1.4948	0.5225	87.0401	1.2741	1.9865	2.1049	69.9746	1.0154	2.4287
SSMultibandKamath02	-3.4043	71.5309	1.152	1.668	-1.5685	58.1421	0.9543	2.0134	-0.0054	46.7996	0.6989	2.3709

Table 4.5 Objective quality evaluation with train noise

BABBLE NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-2.1658	99.8102	1.0416	1.6979	-0.0676	84.5867	0.8295	2.1014	2.0667	63.6597	0.6222	2.4865
PSSBoll79	-3.178	80.2496	0.9505	1.7806	-1.7223	70.5785	0.7779	2.109	-0.2454	58.0147	0.6136	2.4399
SSBerouti79	-3.8347	84.3648	0.981	1.7724	-2.0109	68.0814	0.7689	2.1053	-0.3475	52.8799	0.5664	2.4539
ML80	-3.9253	74.3201	1.0864	1.3516	-3.4184	64.8609	1.0413	1.5392	-3.1517	54.6316	1.0314	1.7431
MMSESTSA84	-3.2938	84.7417	1.0154	1.8368	-1.4556	69.2224	0.8002	2.1987	0.2845	50.5584	0.5969	2.5374
MMSESTSA85	-2.6609	98.5622	1.1069	1.8157	-0.8334	81.7387	0.8885	2.1765	0.8869	60.5833	0.6794	2.5458
WienerScalart96	-1.8441	127.917	1.3919	1.6144	-0.0682	106.0967	1.1065	2.0202	1.7351	80.4268	0.8561	2.4259
SSMultibandKamath02	-3.5064	80.0677	0.9553	1.7772	-1.8025	66.3579	0.7644	2.1148	-0.1396	52.2343	0.5767	2.4656

Table 4.6 Objective quality evaluation with babble noise

STATION NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-1.355	87.7056	1.0241	1.6836	0.747	71.2343	0.7916	2.1252	2.8388	59.5597	0.6034	2.553
PSSBoll79	-3.1837	79.5555	0.98	1.6856	-1.5158	62.519	0.7625	2.1065	0.1521	54.6333	0.6167	2.4464
SSBerouti79	-3.5612	80.1046	0.9741	1.7731	-1.7323	62.9343	0.741	2.1472	0.0173	49.5866	0.5592	2.4684
ML80	-3.8867	72.506	1.11	1.3706	-3.0294	61.1479	1.001	1.5762	-2.766	54.0446	1.0038	1.707
MMSESTSA84	-2.6902	73.681	0.9704	1.8226	-1.0064	59.0474	0.7407	2.2417	0.6919	45.6896	0.5755	2.6019
MMSESTSA85	-2.1629	85.1952	1.041	1.8299	-0.4963	68.7516	0.8068	2.2265	1.4095	53.9997	0.651	2.619
WienerScalart96	-1.0319	120.2635	1.35	1.5471	0.5771	97.8194	1.0681	1.9713	2.5065	75.8354	0.8545	2.4995
SSMultibandKamath02	-3.2386	75.9053	0.9541	1.7445	-1.5342	59.8868	0.7329	2.1477	0.2864	47.1549	0.5675	2.4862

Table 4.7 Objective quality evaluation with station noise

EXHIBITION NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-1.5704	89.9787	1.2204	1.604	0.5896	81.0013	0.9169	2.049	2.4415	66.3214	0.7154	2.4528
PSSBoll79	-3.0198	71.932	1.2158	1.6095	-1.5056	64.2678	0.9535	2.0437	0.0304	57.8017	0.7363	2.3891
SSBerouti79	-3.5836	81.1314	1.2256	1.6753	-1.8016	67.1195	0.9347	2.034	-0.0964	53.8661	0.684	2.3953
ML80	-3.4806	66.6221	1.3068	1.3764	-2.9098	59.2169	1.2174	1.5708	-2.8403	51.8378	1.1681	1.6954
MMSESTSA84	-2.8118	76.5844	1.1343	1.7032	-1.0386	64.3108	0.8679	2.1156	0.6413	51.2957	0.6976	2.4996
MMSESTSA85	-2.1916	88.072	1.1912	1.6525	-0.3228	74.2186	0.9222	2.114	1.3673	59.7444	0.7604	2.5295
WienerScalart96	-1.112	124.0819	1.4734	1.3924	0.6991	100.6981	1.1933	1.9494	2.325	81.3709	1.0114	2.4211
SSMultibandKamath02	-3.1278	73.9685	1.1429	1.6363	-1.4479	62.3126	0.8799	2.0504	0.2523	50.913	0.6583	2.4288

Table 4.8 Objective quality evaluation with exhibition noise

RESTAURANT NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-2.1791	97.2494	0.9992	1.6782	-0.2867	79.0998	0.7826	2.113	1.8766	62.717	0.6157	2.4885
PSSBoll79	-3.1735	80.1378	0.9106	1.7692	-2.0154	68.8552	0.7432	2.0829	-0.3076	57.4583	0.5764	2.4577
SSBerouti79	-3.792	82.0692	0.9269	1.7892	-2.2321	66.2487	0.7272	2.0831	-0.3805	51.5288	0.5373	2.4705
ML80	-3.8993	72.75	1.0709	1.3825	-3.9688	62.7957	1.1046	1.5372	-3.4398	52.6755	1.0534	1.7371
MMSESTSA84	-3.3229	84.9662	0.9705	1.7367	-1.7239	67.2118	0.7577	2.1413	0.1966	50.8995	0.5764	2.5284
MMSESTSA85	-2.7603	97.998	1.0605	1.6748	-1.1707	78.324	0.8522	2.1295	0.7521	60.1569	0.6649	2.5248
WienerScalart96	-2.0164	122.7877	1.356	1.4891	-0.4357	98.5382	1.0725	1.9893	1.4381	76.1611	0.8663	2.4148
SSMultibandKamath02	-3.5731	80.4985	0.9269	1.7802	-2.0399	66.1375	0.7322	2.0886	-0.2064	51.9146	0.5575	2.4774

Table 4.9 Objective quality evaluation with restaurant noise

WHITE NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
MSSBoll79	-1.1132	83.3377	1.7319	1.5769	0.5497	73.6445	1.4188	1.8581	2.9451	64.4099	1.0996	2.3998
PSSBoll79	-2.94	81.7164	1.7683	1.6055	-1.6546	70.093	1.4921	1.9465	-0.1395	60.4332	1.2165	2.3313
SSBerouti79	-3.0658	90.7983	1.7465	1.6889	-1.4241	78.0174	1.4524	2.0059	0.0974	63.7341	1.1594	2.3402
ML80	-3.0383	80.8097	1.781	1.5117	-2.2353	71.5198	1.5827	1.7821	-1.8503	65.438	1.4659	1.9233
MMSESTSA84	-2.1017	77.7371	1.5934	1.7619	-0.6197	65.2738	1.2853	2.1548	1.2022	51.0057	0.9809	2.5624
MMSESTSA85	-1.7984	88.4065	1.6884	1.7397	-0.3158	75.3228	1.3865	2.1378	1.6449	59.4232	1.077	2.5553
WienerScalart96	-0.4152	131.7448	1.9454	1.4284	0.9242	110.4736	1.6645	1.7615	2.8012	85.6904	1.3557	2.3741
SSMultibandKamath02	-2.6404	82.0069	1.6006	1.5931	-1.2496	70.8528	1.3271	1.9842	0.232	58.3821	1.0682	2.366

Table 4.10 Objective quality evaluation with white noise

For comparison purpose the results of SSNR, WSS, LLR and PESQ for all conditions are shown in the form of bar chart in figures 4.7 to 4.11 respectively. The SSNR value for MSS and Wiener filter is higher under all test condition as compared to other methods. The WSS score is lower for MMSE STSA methods for most of the cases which reveals that the speech enhanced by these methods has lesser spectral distortion. In some cases the ML and MBSS methods give lower WSS but they have low SSNR in comparison with MMSE STSA methods. From LLR comparison the MMSE STSA algorithms have value less than one for most cases. Ideally LLR should be zero. The PESQ score above 2.5 is desirable from the noise perception and speech quality point of view. In this regards the MMSE STSA algorithms work satisfactorily. Hence it is concluded here that from all eight STSA algorithms the MMSE STSA algorithms are performing better compared to any other algorithm. Now from MMSE STSA 84 and MMSE STSA 85 algorithms the use of MMSE STSA85 algorithm is recommended for any future enhancement as it follows the LSA (Log Spectral Attenuation) characteristics of human ear.

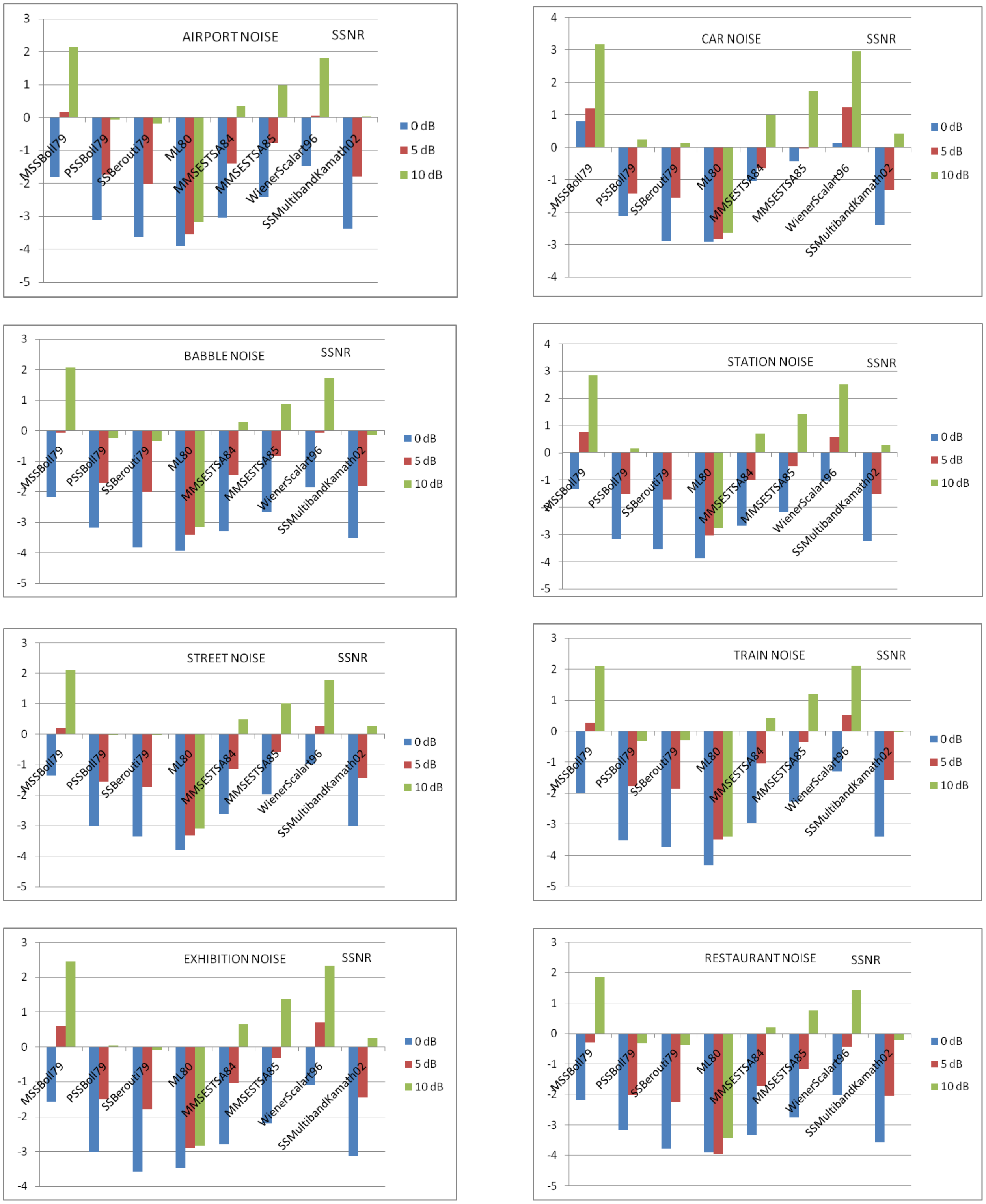


Fig. 4.7 SSNR comparison of STSA algorithms over NOIZEUS database

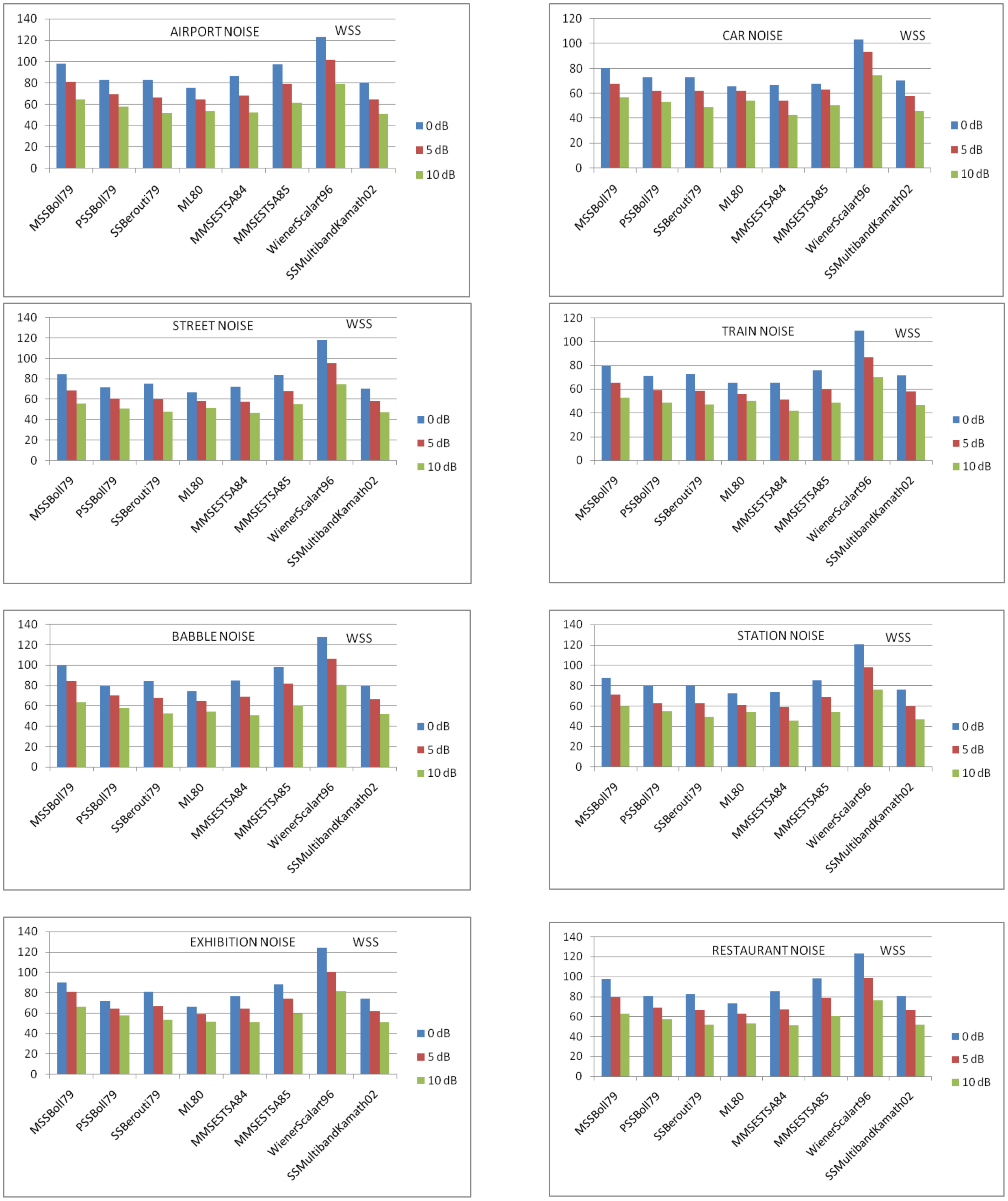


Fig. 4.8 WSS comparison of STSA algorithms over NOIZEUS database

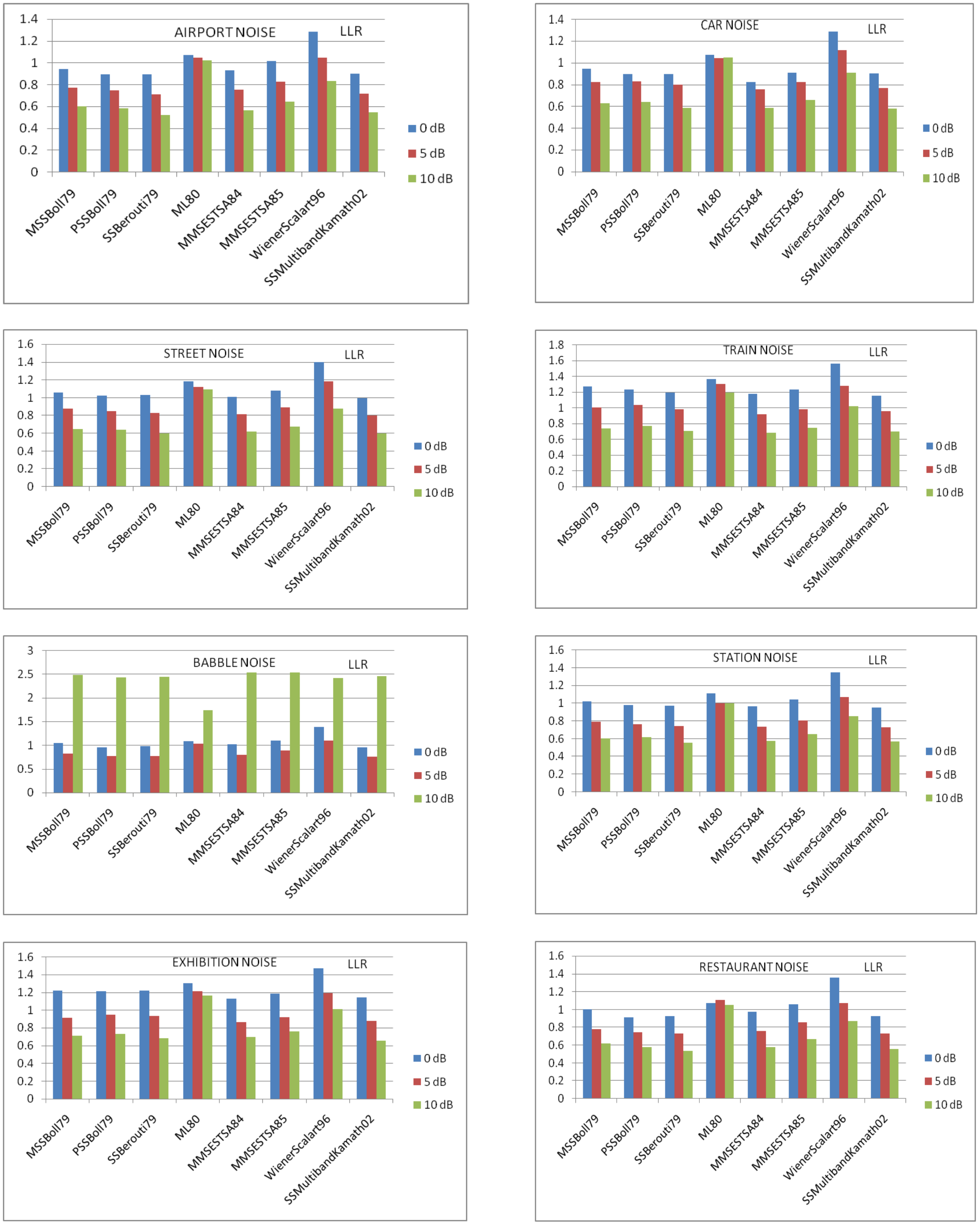


Fig. 4.9 LLR comparison of STSA algorithms over NOIZEUS database

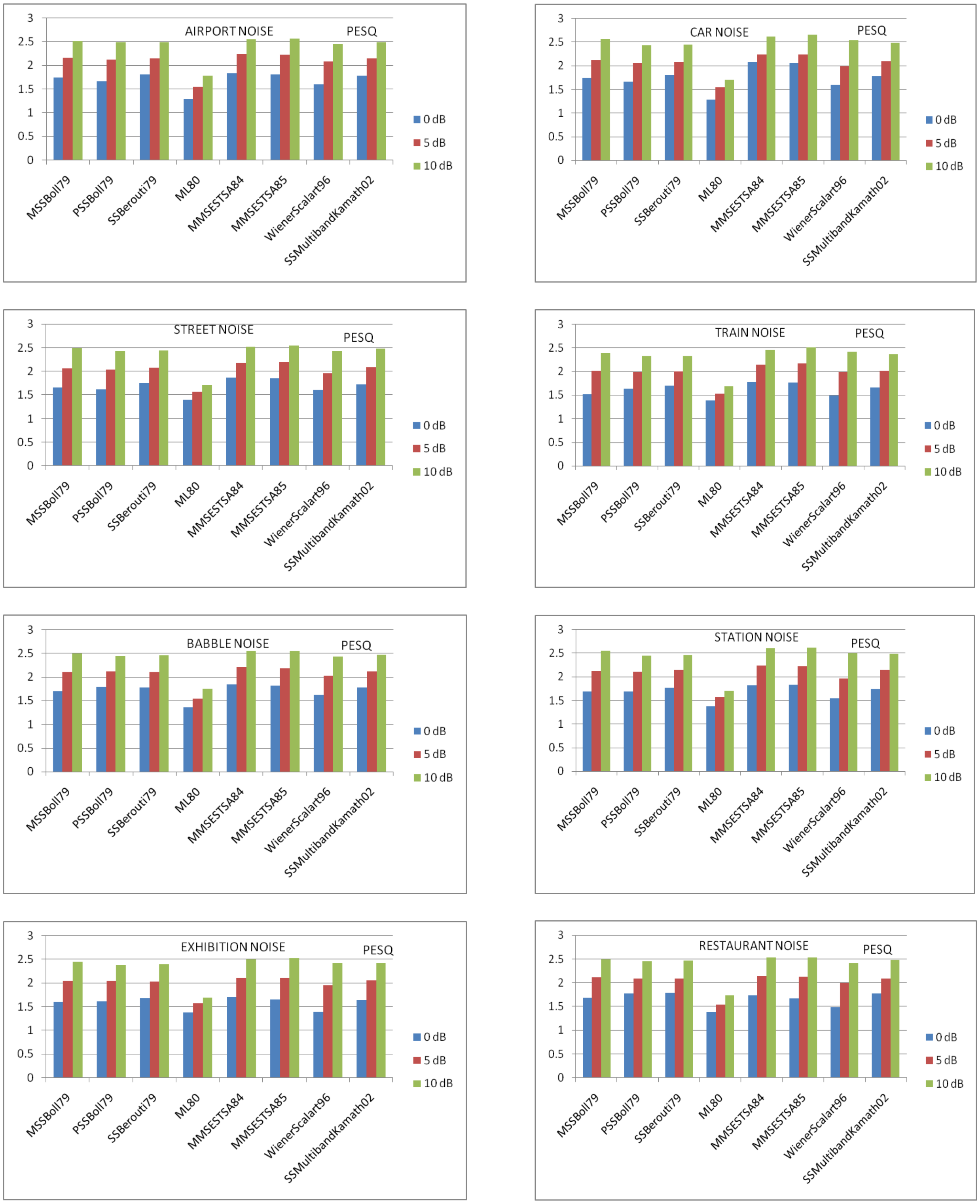


Fig. 4.10 PESQ comparison of STSA algorithms over NOIZEUS database

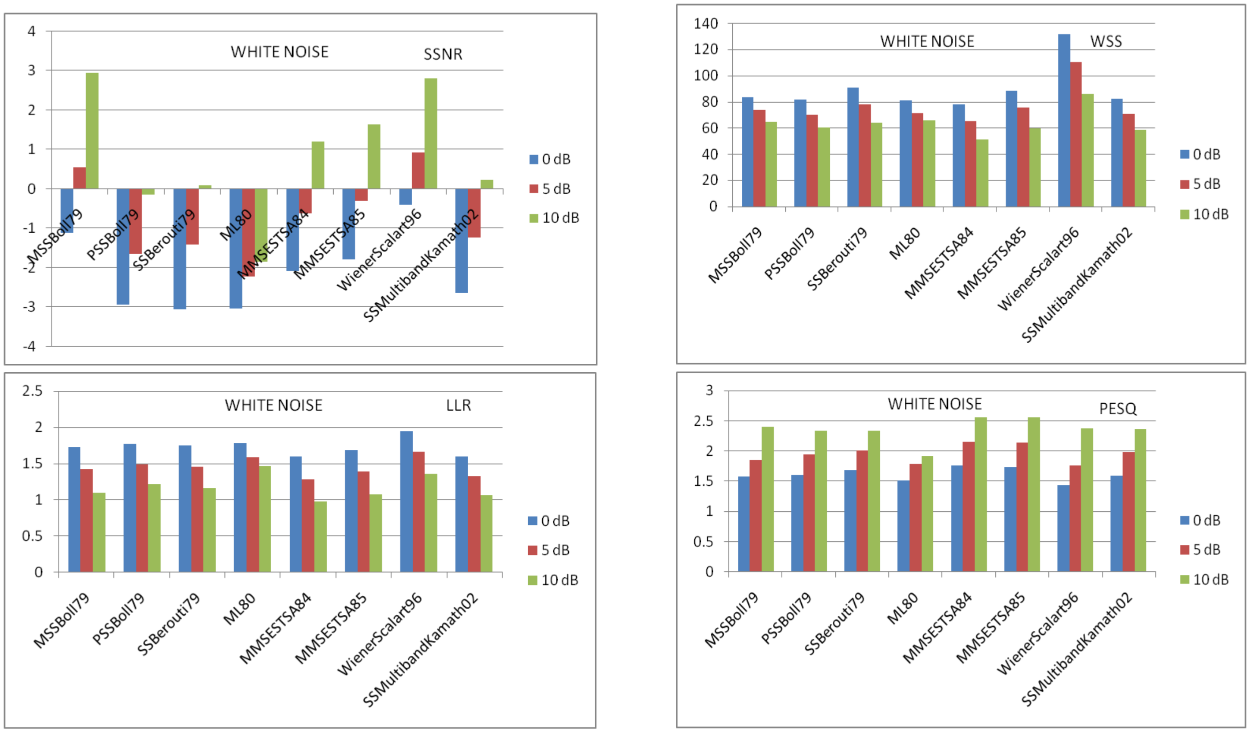
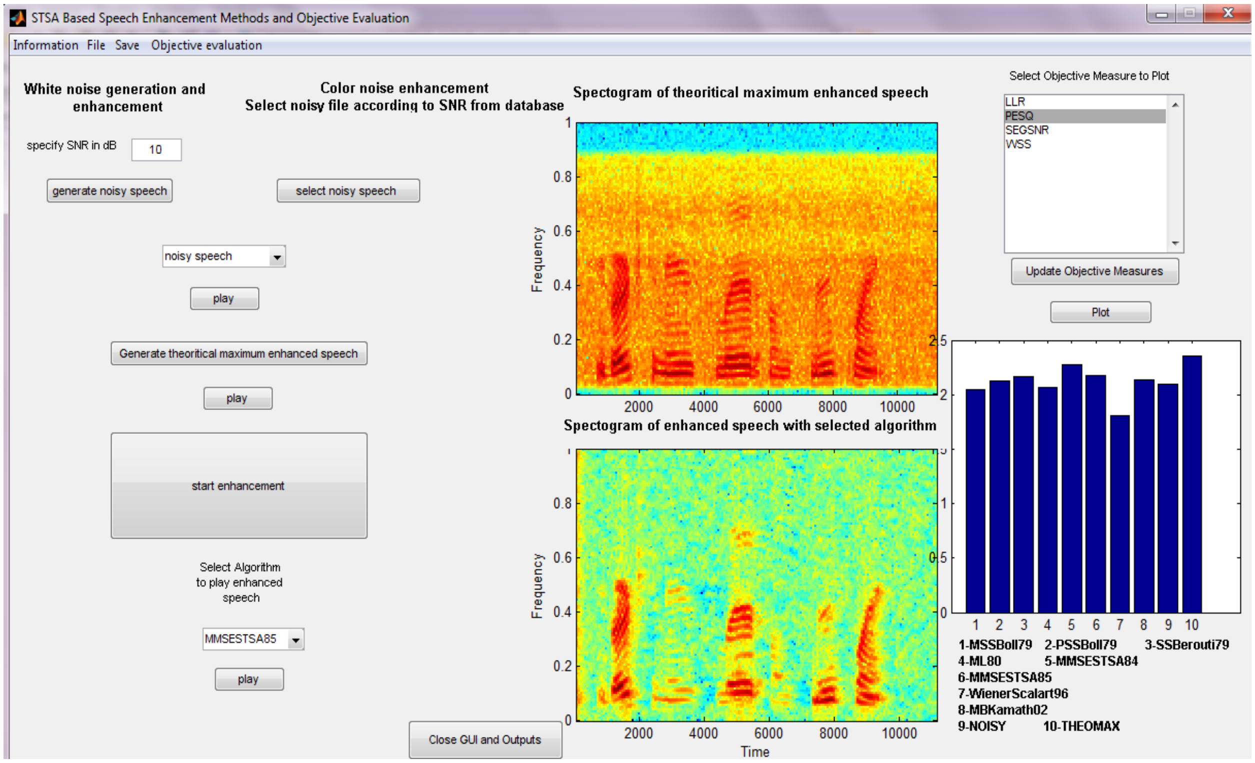


Fig. 4.11 Objective evaluation of STSA algorithms under white noise

4.5 Graphical User Interface (GUI)

To consolidate the simulation of STSA algorithms with white and colored noise a MATLAB GUI [11] is designed and it is depicted in figure 4.12.



specify SNR in dB 10

noisy speech

Select Algorithm to play enhanced speech: MMSESTSA85

Select Objective Measure to Plot: PESQ

STSA Based Speech Enhancement Methods and Objective Evaluation
Information File Save Objective evaluation
White noise generation and enhancement
Color noise enhancement
Select noisy file according to SNR from database
generate noisy speech
select noisy speech
play
Generate theoritical maximum enhanced speech
play
start enhancement
play
Close GUI and Outputs
Spectogram of theoritical maximum enhanced speech
Spectogram of enhanced speech with selected algorithm
Frequency
Time
0 0.2 0.4 0.6 0.8 1
2000 4000 6000 8000 10000
LLR
PESQ
SEGSNR
WSS
Update Objective Measures
Plot
0 0.5 1 1.5 2 2.5
1 2 3 4 5 6 7 8 9 10
1-MSSBoll79 2-PSSBoll79 3-SSBerouti79
4-ML80 5-MMSESTSA84
6-MMSESTSA85
7-WienerScalart96
8-MBKamath02
9-NOISY 10-THEOMAX

Fig. 4.12 MATLAB GUI for STSA algorithms

The important features of GUI are as follows.

1. It allows selecting a clean speech file from NOIZEUS database or any other .wav file at 8 KHz sampling frequency and 16bits/sample resolution.
2. The user can specify the SNR in dB for white noise which is added to clean speech file to generate the noisy file or the noisy file with particular SNR and type from NOIZEUS database.
3. The spectrograms of clean and noisy files are displayed and the file can be played to have listening experience. The eight different STSA algorithms can be applied to noisy file and the spectrogram of enhanced speech signal is displayed in GUI as well as in separate window for storage purpose.
4. The enhanced speech signal can be played as well as it can be saved in .wav file by specifying the name of output file.
5. The GUI also supports the objective evaluation using SSNR, WSS, LLR and PESQ scores.

6. The results can be displayed in tabular and bar graph forms. For reference the spectrograms and objective measures are also displayed for theoretical maximum limit (obtained by combining clean magnitude and noisy phase) as well as for noisy speech. The snapshot of the developed GUI is shown in figure 4.12. [1]

4.6 Implementation of Wavelet De-noising Methods

The wavelet de-noising using hard and soft thresholding with universal and SURE level dependent thresholds described in section 3.6 is implemented in MATLAB with different mother wavelets (Daubechies 20, Coiflets 4 and Symlet 20 with level 3). The objective evaluation results with white noise over SNRs 0 dB to 10 dB are summarized in table 4.11.

WHITE NOISE	0 dB				5 dB				10 dB			
	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ	SSNR	WSS	LLR	PESQ
UNI_H_DB20	-2.8905	160.0671	5.775	0.6488	-1.0113	140.841	5.4235	0.9801	1.0555	118.8326	4.9883	1.3446
UNI_H_SYM20	-2.8715	159.5993	5.7625	0.6519	-0.9866	140.3427	5.3847	0.9738	1.1052	118.2422	4.9159	1.3666
UNI_H_COIF4	-2.9599	254.0805	3.7652	0.7467	-1.1484	208.4518	3.5026	1.0651	0.8767	168.5184	3.1939	1.4492
UNI_S_DB20	-3.0632	160.3541	5.8567	1.3663	-1.481	142.1192	5.5544	1.4404	0.1617	120.6072	5.1556	1.6331
UNI_S_SYM20	-3.049	160.0692	5.8474	1.342	-1.4585	141.4369	5.5194	1.4182	0.1893	119.821	5.0981	1.627
UNI_S_COIF4	-3.1417	258.2193	3.8408	1.3886	-1.6097	214.5133	3.626	1.4529	-0.021	175.2163	3.3394	1.6603
SURE_H_DB20	-3.121	97.7551	2.7404	1.6331	-0.8038	73.1521	1.5973	1.9518	1.9458	58.0661	1.1924	2.2562
SURE_H_SYM20	-3.1384	96.0477	2.5555	1.6397	-0.8195	72.9212	1.5699	1.952	1.9662	57.802	1.2075	2.2572
SURE_H_COIF4	-3.1715	86.0584	1.9675	1.6295	-0.9351	69.9765	1.3421	1.9366	1.8849	56.7398	1.1165	2.2503
SURE_S_DB20	-2.2239	100.9366	3.0619	1.9025	0.1356	77.5271	1.7359	2.2349	2.7272	61.6227	1.2222	2.5328
SURE_S_SYM20	-2.2133	99.7078	2.8725	1.904	0.1685	76.8788	1.6873	2.2413	2.7609	61.2674	1.2223	2.5414
SURE_S_COIF4	-2.2821	90.8677	2.1646	1.8536	0.0487	74.0614	1.3256	2.1825	2.6355	60.0887	1.047	2.4916

Table 4.11 Objective quality evaluation of wavelet de-noising methods

For comparison purpose same results are shown in bar chart form in figure 4.13. The results are very poor compared to STSA algorithms especially at low SNRs. The results with SURE soft thresholding are somewhat comparable to STSA methods. However, the results explain the reason for non popularity of wavelet de-noising for speech enhancement. The poor performance also encountered in colored noise conditions.

[1] A paper entitled “Performance Evaluation of STSA based Speech Enhancement Techniques for Speech Communication Systems” is presented in National conference on Wireless Communication and VLSI design (NCWCVD-2010) Organized by GEC, Gwalior and IEEE MP Subsection in March 2010.

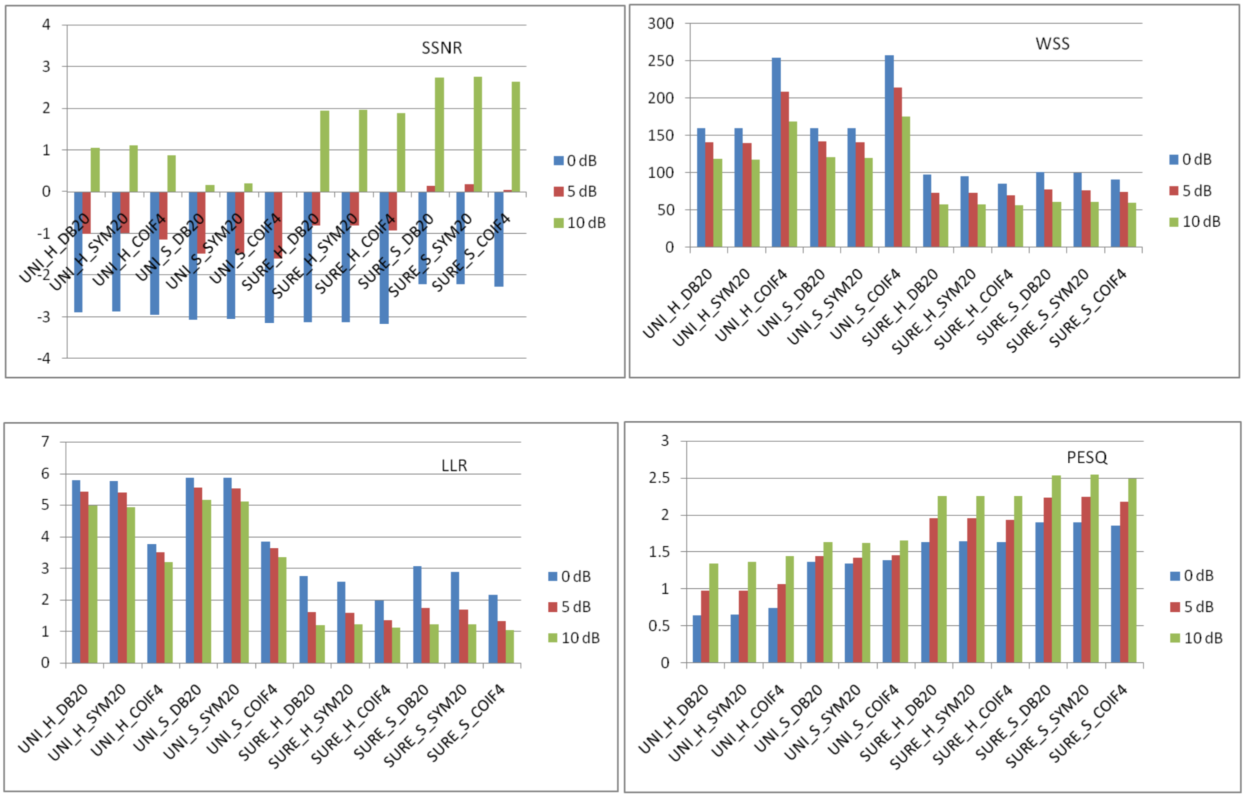


Fig. 4.13 Objective evaluation of wavelet algorithms under white noise

4.7 Summary

The MATLAB simulation of STSA and wavelet de-noising techniques along with their objective evaluations are described in this chapter. The details for implementation are shown using flow charts and block diagrams. The objective evaluation is performed by finding SSNR, LLR, WSS and PESQ scores for all algorithms under different noise conditions. The NOIZEUS database is utilized for evaluation. The MATLAB GUI is prepared which can simulate the STSA algorithms and also evaluates them. The discussion of objective evaluation results has concluded that the MMSE STSA85 algorithm is superior compared to all other STSA algorithms. The performance is found consistent in both white and colored noise environments. However, the performance is not satisfactory at low SNR conditions. The wavelet de-noising is not found much successful and feasible for real time speech enhancement systems. It is recommended here to shift the focus to other domains. The relative spectral analysis (RASTA) is novel approach for speech enhancement and it is described in the next chapter and performance is compared with the STSA algorithms.

Chapter 5

Relative Spectral Analysis-RASTA

The need for noise reduction and suppression technology is more important than ever. Mobile phones, portable communication devices and other phones are widely used in noisy environments. As a result, phone calls contain, in addition to the speaker's voice, unwanted signals like other people talking around, vehicle engine noise and horns, wind noise, keyboard-strokes etc. A background noise suppression system developed by Motorola is included as a feature in IS-127, the TIA/EIA standard for the Enhanced Variable Rate Codec (EVRC) to be used in CDMA based telephone systems [1]. EVRC was modified to EVRC-B and later on replaced by Selectable Mode Vocoder (SMV) which retained the speech quality at the same time improved network capacity. Recently, however, SMV itself has been replaced by the new CDMA2000 4GV codecs. 4GV is the next generation 3GPP2 standards-based EVRC-B codec [2]. The EVRC based codec uses combination of STSA based approaches: multiband spectral subtraction (MBSS) and minimum mean square error (MMSE) gain function estimator for background noise suppression as a preprocessor. The voice activity detector (VAD) used to decide speech/silence frame is embedded within the algorithm. Its quality has been proven good through commercial products. Nevertheless, the quality may not be sufficiently good for a wide range of SNRs, which were not given much attention when it was standardized. Another algorithm suggested by A.Sugiyama, M.Kato and M. Serizawa [3] uses modified MMSE-STSA approach based on weighted noise estimation. The subjective tests on this algorithm claim to give maximum difference in mean opinion score (MOS) of 0.35 to 0.40 compared to EVRC and hence its later version is equipped within 3G handsets [3].

The STSA based algorithms are able to suppress the noise effectively subject to accurate estimation of noise during silence interval detected by VAD. Its performance depends on VAD. Also, the STSA based approaches have their common problems of musical noise and speech distortion. Hence it is needed to shift the enhancement domain itself. This leads to investigate the use of RelAtive SpecTral Amplitude (RASTA) processing of speech originally proposed by Hermansky and Morgan [4] and designed to alleviate effects of convolutional and additive noise in automatic speech recognition (ASR). Recently, RASTA was also applied to direct enhancement of noisy speech in communication systems [4, 6]. A noise suppression system for cellular communications based on RASTA has been proposed in [5].

This chapter describes the RASTA (Relative Spectral Analysis) processing of speech. It involves temporal processing and motivated by some auditory masking features. This algorithm for speech enhancement is simulated in MATLAB and evaluated under different noise conditions using NOIZEUS database. The results are compared with STSA based algorithms. The original filter is redesigned to have better performance. The algorithm throws the challenge for real time implementation as it is non linear and non causal. However, it does not require the use of VAD and can be used to combat with additive and convolutive distortions.

5.1 Auditory Masking Features

In the phenomenon of auditory masking, one sound component is concealed by the presence of another sound component. The RASTA algorithm use auditory masking principle in reducing the perception of noise. There are two different psychoacoustic phenomena referred to as frequency and temporal masking. Research in psychoacoustic has also shown that human ear can have difficulty in hearing weak signals that fall in frequency or time vicinity of stronger signals (as well as those superimposed in time or frequency on the masking signal, as in the above two cases). A small spectral component may be masked by a stronger nearby spectral component. A similar masking can occur in time for two closely-spaced sounds. In speech enhancement, this principle of masking is exploited for noise-reduction in frequency domain. While temporal masking by adjacent sounds has proven useful, particularly in wideband audio coding [12], it has been less widely used in speech processing because it is more difficult to quantify. The frequency domain masking is based on the concept of a critical band. Using this paradigm, it is possible to determine the masking threshold for complex signals such as speech. The speech masking threshold is the spectral level (determined from the speech spectrum) below which non-speech components are masked by speech components in frequency.

5.1.1 Frequency-Domain Masking Principles

As explained in [13] the basilar membrane, located at the front-end of the human auditory system, can be modeled as a bank of about 10,000 overlapping band-pass filters, each turned to a specific frequency (the characteristic frequency) and with bandwidths that increase roughly logarithmically with increasing characteristic frequency. These psychologically based filters thus perform a spectral analysis of sound pressure level appearing at the ear-drum. In contrast, there also exist psycho-acoustically based filters that relate to human’s ability to perceptually resolve sound with respect to frequency. The bandwidths of these filters are known as the critical bands of hearing and are similar in nature to the physiologically based filters.

Frequency analysis by a human has been studied by using perceptual masking. A tone at some intensity that human ear trying to perceive is called the maskee. A second tone, adjacent in frequency, attempts to drown out the presence of the maskee called the masker. If one can determine the intensity level of the maskee (relative to the absolute level of hearing) at which it is not audible in the presence of the masker. This intensity level is called the masking threshold of the maskee. The general shape [12] of the masking curve for a masking tone at frequency Ω_0 with a particular sound-pressure level (SPL) in decibels is shown in figure 5.1. Adjacent tones that have an SPL below the solid lines are not audible in the presence of the tone atΩ_0 . As it is shown that there is a range of frequencies about the masker whose audibility is affected. Tones with intensity below the masking threshold curve are masked by the masking tone. The curve has asymmetric nature around Ω_0 .

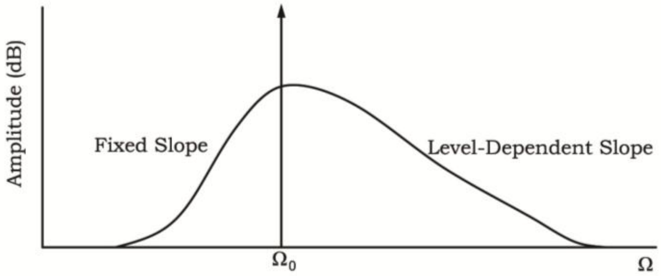


Fig. 5.1 General shape of the masking threshold curve for a masking tone at frequency Ω_0

Another important property of masking curves is that the bandwidth of these curves increases roughly logarithmically as the frequency of the masker increases. Experiments conducted have given the roughly logarithmically increasing width of the critical band filters and suggested about 24 critical band filters cover our maximum frequency range of 15000Hz for human perception. A means of mapping linear frequency to this perceptual representation is through the bark scale. In this mapping, one bark covers one critical band with the functional relation of frequency f to bark z given by [15].

$$z = 13\tan^{-1}(0.76f) + 3.5\tan^{-1}\left(\frac{f}{7500}\right) \tag{5.1}$$

In the low end of the bark scale (<1000 Hz), the bandwidths of the critical band filters are found to be about 100Hz and in higher frequencies the bandwidths reach up to about 3000Hz [15]. A similar mapping uses the mel scale. The mel scale is approximately linear up to 1000Hz and logarithmic thereafter [15].

$$m = 2595 log_{10}\left(1 + \frac{f}{700}\right) \tag{5.2}$$

Although equation 5.2 provides a continuous mapping from linear to bark scale, most perceptually motivated speech processing algorithms use quantized bark numbers of 1,2,3…24 that correspond approximately to the upper band edges of the 24 critical bands that cover range of hearing of human ear. This allows exploiting the perceptual masking properties with feasible computation in speech signal processing.

5.1.2 Masking Threshold Computation

For speech signal, the effects of individual masking components are additive; the overall masking at a frequency component due to all the other frequency components is given by the sum of the masking due to the individual frequency components, giving a single masking threshold [16]. For a background noise disturbance (the maskee) in the presence of speech (the masker) it is required to determine the masking threshold curve, as determined from the speech spectrum, below which background noise would be inaudible. For the speech threshold calculation, the masking ability of tonal and noise components of speech (in masking background noise) is different.

5.1.3 Exploiting Frequency Masking in Noise Reduction

In exploiting frequency masking, the basic approach is to attempt to make inaudible spectral components of annoying background residual (from an enhancement process) by forcing them to fall below a masking threshold curve as derived from a measured speech spectrum. The interest is in masking this annoying (often musical) residual while maximizing noise reduction and minimizing speech distortion. There are a variety of psycho-acoustically motivated speech enhancement algorithms that seek to achieve this goal by using suppression filters similar to those from spectral subtraction and Weiner filtering [16-18]. Each algorithm establishes a different optimal perceptual tradeoff between the noise reduction, background residual (musical) artifacts, and speech distortion.

There are two particular suppression algorithms that exploit masking in different ways. The first approach by Virag [19] applies less attenuation when noise is heavily masked so as to limit speech distortion. In this approach a masking threshold curve is used to modify parameters of a Berouti spectral subtraction scheme. The parameters α and β are adapted to the masking threshold curve on each frame. Virag found that the proposed spectral subtraction scheme that adapts to auditory masking outperformed the more classical spectral subtraction approaches, according to the objective measures. Finally Virag used the subjective Mean Opinion Score (MOS) test to show that the auditory based algorithm also outperforms other subtractive type noise suppression algorithms with respect to human perception; the algorithm was judged to reduce musical artifacts and give acceptable speech distortion. These results motivate the research work to include perceptual features in speech enhancement algorithms.

The second approach by Gustafsson, Jax, and Vary [20] seeks residual noise that is perceptually equivalent to an attenuated version of the input noise without explicit consideration of speech distortion. In this approach, rather than using the masking threshold curve to modify a standard suppression filter, the masking threshold is used to derive a new suppression filter that results in perceived noise which is an attenuated version of the original background noise.

An extension of the suppression algorithm by Gustaffson, Jax, and Vary that reduces speech distortion has been introduced by Govindasamy [18]. This method uses frequency-domain masking to explicitly seek to hide speech distortion simultaneously with the noise distortion.

However, all these approaches which exploit the auditory masking features; but perform speech enhancement by applying various forms of spectral subtraction and Wiener filtering on short time speech segments, holding the time variable fixed in the STFT. Throughout the review process it has been observed that if the same direction of thinking continues; the inherent problems of musical noise and speech distortion will not be solved together. So if a different approach is taken in which the frequency variable is kept fixed and the filtering is applied along time trajectories of STFT (temporal domain filtering) the problems may get resolved. It is also required to embed the perceptual features described here in the temporal domain filter. This leads to the development of RASTA processing of speech. Frame-by-frame analysis of speech dates from early days of speech analysis-synthesis. RASTA processing represents a departure from this paradigm. It is a step in the direction of modeling some temporal properties of human auditory processing. It has a potential for further improvements as more knowledge about the modeling of human auditory perception will be available.

5.2 RASTA Processing System

In ASR the task is to decode the linguistic message in speech. This linguistic message is coded into the movements of the vocal tract. The speech signal reflects these movements. The rate of change of non linguistic components in speech often lies outside the typical rate of change of vocal tract shape. The relative spectral processing (RASTA) uses this fact. It is motivated by some auditory features which are, in part, similar to that for adaptivity in the Weiner filter of Section 5.2.3. It suppresses the spectral components that changes more slowly or quickly than the typical range of change of speech. This rate of change of the short-time spectral envelope can be described by the modulation spectrum (temporal feature), i.e. the spectrum of the time trajectories described by the short-time spectral envelope [7]. For a wide range of frequency bands, the modulation spectrum of speech exhibits a maximum at about 4Hz, the average syllabic rate. RASTA exploits this modulation frequency preference. With slowly varying (rather than fixed) channel degradation, and given human ear insensitivity to low modulation frequencies, in RASTA a filter that notches out frequency components at and near DC is applied to each channel. In addition, the RASTA filter suppresses high modulation frequencies to account for the human's preference for signal change at a 4Hz rate. Disturbances such as additive noise may have different modulation spectrum properties than speech and often have modulation frequency components outside the speech range, and could in principle be attenuated without significantly affecting the range with relevant linguistic information. The RASTA processing suppresses the spectral components outside the typical modulation spectrum of speech. The maximum modulation frequency of the modulation spectrum is half of the sampling frequency of RASTA filter. The sampling frequency of RASTA filter is decided by frame rate. The frame rate is obtained by taking ratio of sampling frequency of speech signal to the number of shift points in a frame.

RASTA based speech enhancement suggested in [6] involves linear filtering of the trajectory of the short-term power spectrum of noisy speech signal as shown in figure 5.2.

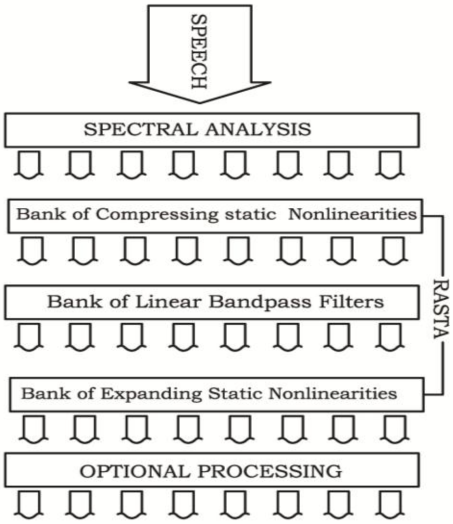

SPEECH
SPECTRAL ANALYSIS
Bank of Compressing static Nonlinearities
Bank of Linear Bandpass Filters
Bank of Expanding Static Nonlinearities
RASTA
OPTIONAL PROCESSING

Fig. 5.2 Block diagram: RASTA processing system

RASTA algorithm processing steps for each analysis frame are…

- Compute the short time power spectrum of windowed signal.
- Transform spectral amplitude through a compressing static nonlinear transformation.
- Filter the time trajectory of each transformed spectral component.
- Transform the filtered speech representation through expanding static nonlinear transformation.
- Perform the overlap add synthesis and reconstruct the signal.

5.3 RASTA Method

It is a generalization of cepstral mean subtraction (CMS) that was introduced in section 2.2.2.2. The original algorithm addresses the problem of a slowly time-varying linear channel (i.e., convolutional distortion) in contrast to the time invariant channel removed by CMS. The essence of RASTA is a cepstral lifter that removes low and high modulation frequencies and not simply the DC component, as does CMS. The filter suggested in [4] is the fixed IIR band pass filter for all time trajectories given by transfer function

$$P(z) = 0.1z^4 * \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.94z^{-1}} \tag{5.2}$$

For the sampling frequency of 100Hz i.e., the frame interval corresponds to 10ms, the filter is designed with lower cut-off frequency of 0.26Hz. The filter slope decline 6dB/octave from 12.8Hz with sharp zeros at 28.9Hz and at 50Hz. The frequency response of the filter is shown in figure 5.3.

The low cut-off frequency of the filter determines the fastest spectral change of the non-linearly transformed spectrum, which is ignored in the output, whereas the high cut-off frequency determines the fastest spectral change that is preserved in the output parameters. The high-pass portion of the equivalent band pass filter is expected to alleviate the effect of convolutional noise introduced in the channel. The low-pass filtering helps to smooth some of the fast frame-to-frame spectral changes present in the short-term spectral estimate due to analysis artifacts.

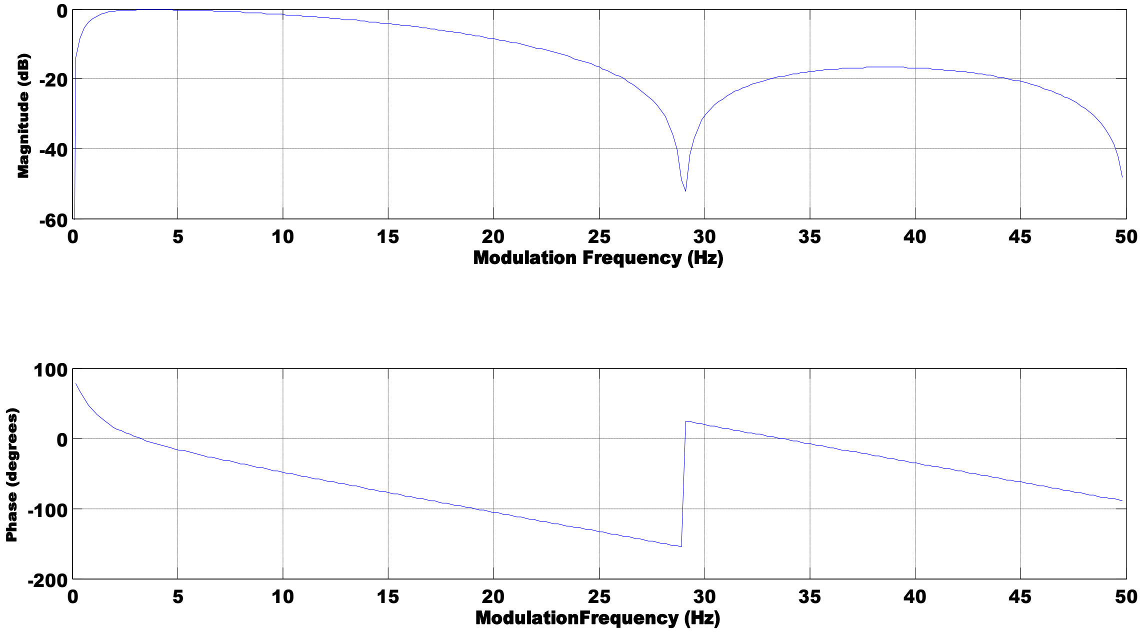


Fig. 5.3 Frequency response of fixed IIR RASTA filter

In RASTA for convolutive distortion reduction, the compressing static nonlinear operator becomes the magnitude followed by the logarithm operator. The expanding static nonlinear operator is inverse logarithm (exponential). The RASTA enhancement for convolutive distortion reduction is given by

$$\left|\hat{X}(n.\omega)\right| = exp\left\{\sum_{m=-\infty}^{\infty} p(n-m)log|Y(m,\omega|\right\} \tag{5.3}$$

The RASTA filter is seen to peak at about 4Hz. As does CMS, RASTA reduces slowly varying signal components, but in addition, suppresses variations above about 16Hz. The complete RASTA temporal processing for blind de-convolution is illustrated in figure 5.4. In this figure, a slowly varying distortion$log|\mathrm{G}(\mathrm{n},\omega)\ |$, due to a convolutional distortion g(n) to be removed by the RASTA filter $p(n)$, is added to the rapidly varying speech contribution $log|X(n,\omega)\ |$.

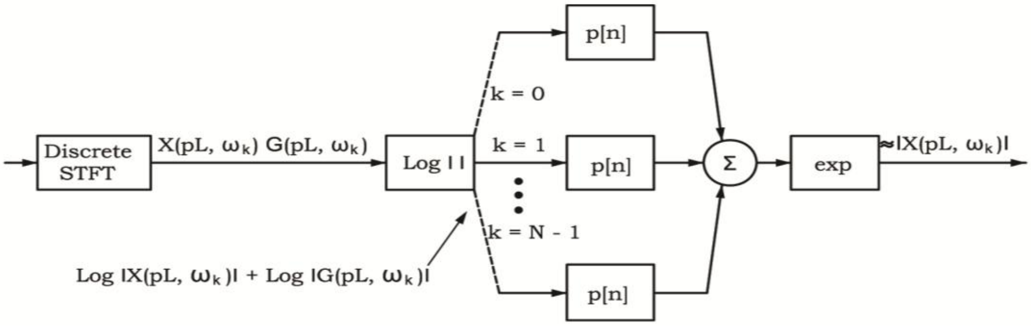


Fig. 5.4 Flow diagram of RASTA processing for de-convolution

In addition to reducing convolutional distortion, RASTA can also be used to reduce additive noise. The temporal processing is applied to the STFT magnitude and the original (noisy) phase along each temporal trajectory is kept intact. In performing noise reduction along STFT temporal trajectories, it is assumed that the noise background changes slowly relative to the rate of change of speech, which is concentrated in the 1-16Hz range. A nonlinear operator used in equation 5.3 such as the logarithm, however does not preserve the additive noise property and thus linear trajectory filtering is not strictly appropriate. Nevertheless, a cubic-root for compression and cubic power for expansion of the power spectrum (2/3rd power for compression and 3/2th power for expansion of magnitude spectrum) results in a noise reduction [6-8]. However, applying rather aggressive fixed ARMA RASTA filters (designed for suppression of convolutional distortions in ASR) yields results similar to spectral subtraction, i.e., enhanced speech often contains musical noise and the technique typically degrades clean speech. Also, in [4] it is stated that for the RASTA enhancement processing described above, neither formal perceptual experiments were run nor any significance objective evaluation using corpus of noisy data was performed. The parameters and filter described are influenced by audible results only.

A noise suppression system for cellular communication based on RASTA processing [5] has been proposed. In this method the fixed IIR band pass filter is replaced by multiband non-causal FIR Wiener like filters with 21 tapes to achieve more reliable noise reduction. The impulse response of the filer bank is shown in figure 5.5. Here the 256 point window with 192 points of overlap for 8 KHz signal sampling frequency is used. This gives sampling frequency of RASTA filter as 125Hz. The filter in the band 0-100Hz is having almost flat frequency response and has all pass characteristics. In the bands 150-250Hz and 2700-4000Hz the filters low gain low pass filters with at least 10dB attenuation for modulation frequencies above 5Hz. For region 300-2300Hz the filter have a band pass characteristics, emphasizing modulation frequencies around 6-8Hz. Compared to original fixed IIR filter, the low frequency band stop is much milder, being only at most 10dB down from the maximum. Here each filter is designed optimally to map a time window of noisy speech spectrum of specific frequency to a single estimate of short time magnitude spectrum of clean speech as determined from a training speech database.

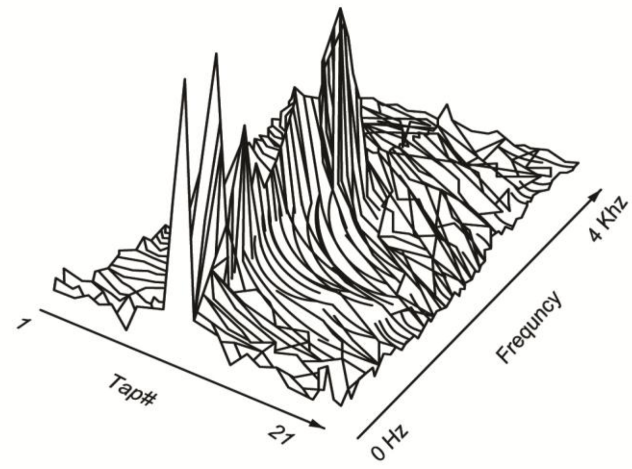


Fig. 5.5 Impulse response of multiband 21 taps FIR filters for additive noise removal in RASTA processing

In general the power law modification of the magnitude trajectories for additive noise removal is given by following equation.

$$\left|\hat{X}(n.\omega)\right| = \left\{ \sum_{m=-\infty}^{\infty} P(n-m,\omega)|Y(m,\omega|^{1/\gamma} \right\}^{\gamma} \tag{5.4}$$

With the filter design technique described above, the value of γ=3/2 for the power-law nonlinearity was found in informal listening to give preferred perceptual quality. The general block diagram of RASTA processing for additive noise removal is shown in figure 5.6.

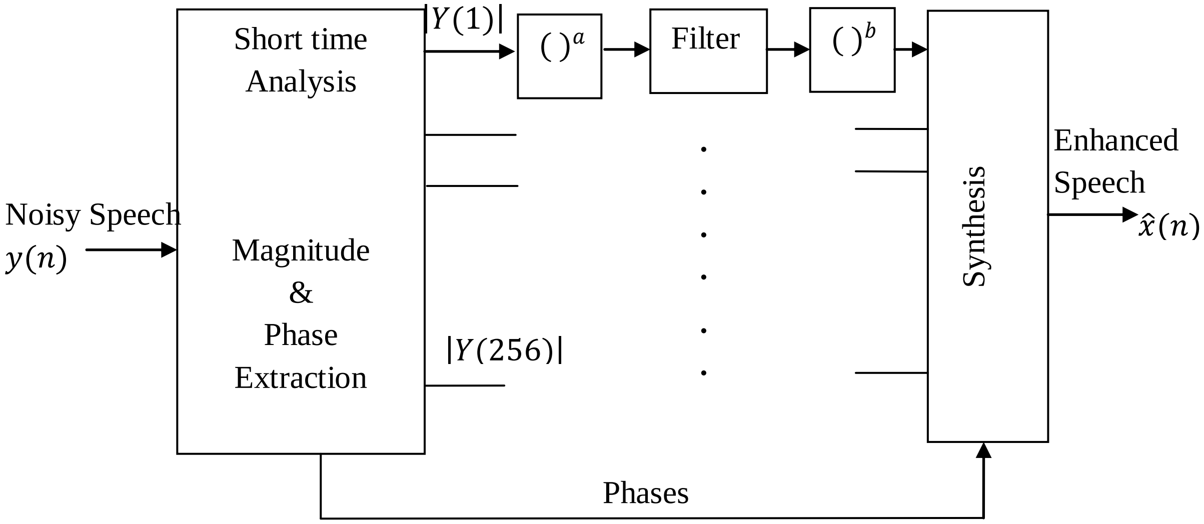
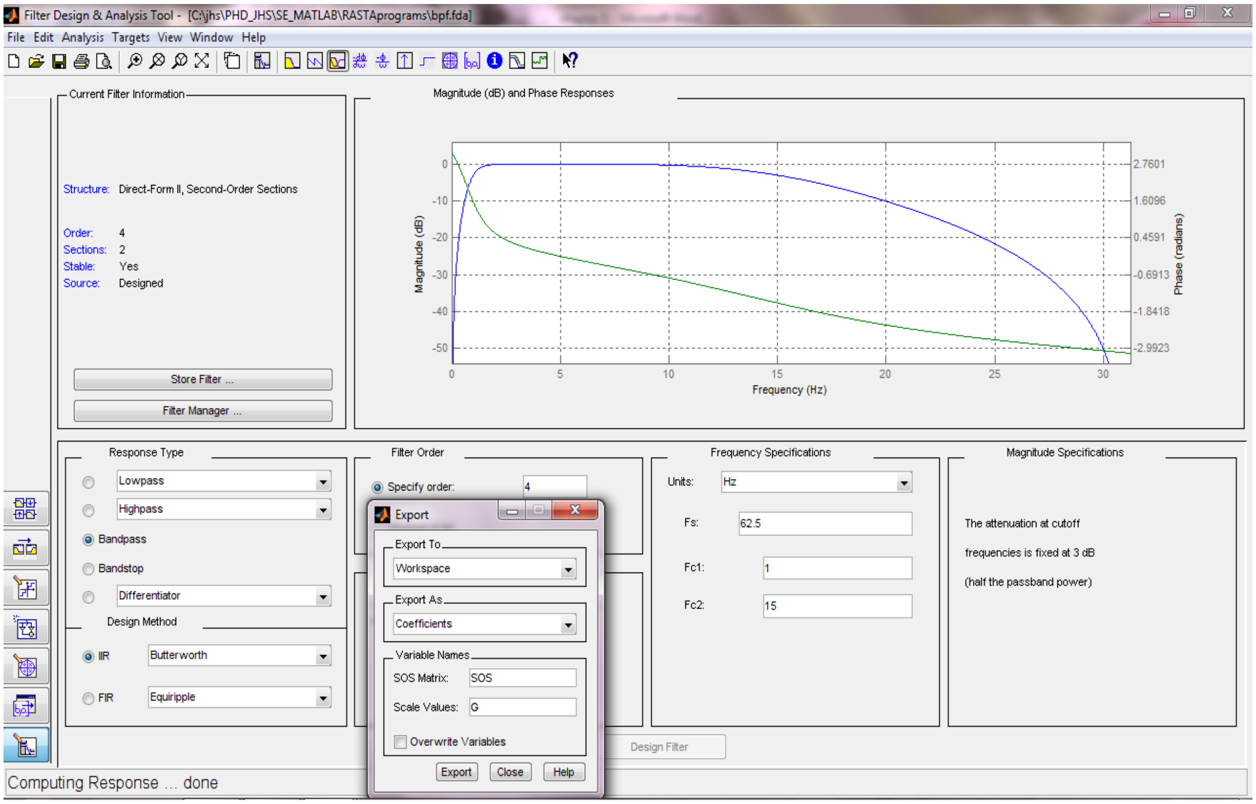


Fig. 5.6 Flow diagram of RASTA processing for additive noise removal (*b=1/a=γ*)

5.4 RASTA Algorithm Implementation and Modifications

In simulation of original RASTA filter; the filter for each time trajectory is implemented by fixed IIR band pass filter with transfer function given by equation 5.2. To perform the RASTA filtering a Hamming window of 200 samples length with an overlap of 120 samples is used. With 8 KHz signal sampling frequency this gives 25ms window duration and 15ms overlap. The frame rate and hence the sampling frequency of RASTA filter is 100Hz. The maximum modulation frequency is 50Hz. The filter is designed with lower cut-off frequency of 0.26Hz. The filter slope decline 6dB/octave from 12.8Hz with sharp zeros at 28.9Hz and at 50 Hz. The frequency response of the filter is shown in figure 5.3. The algorithm used for obtaining FFT generates 256 points complex FFT, which gives magnitude and phase for first 129 points. Each spectral value is filtered using the filter described in equation 5.2. These filtered spectral values are combined with the phase of noisy spectrum, 256 point IFFT is applied and overlap-add operation is performed to reconstruct the enhanced speech. The value of parameter a is set to 2/3 and hence b is set to 3/2 which are proposed in [4-6].

As per auditory principles; the nonlinear compression and expansion is critical in RASTA approach. In simulation experiment the parameters a and b are tested for different values. The parameters suggested in [4] are based on audible experience only. Hence the simulation is carried out with originally fixed values of parameters a and b as well as with the modified values. From the listening experience the parameter values *a*=3/4 and *b*=4/3 are found more satisfactory. To confirm it the evaluation is carried out here. The objective evaluation explained in next section as well as subjective listening test explained in chapter 6 also backs the results. Also the original fixed RASTA filter is modified and it is replaced by multiband filters as suggested in [5]. However, from better implementation point of view the non-causal FIR Weiner like filters are approximated by fourth order Butterworth filters. For implementation 256 point Hamming window with 50% overlap is used which gives the sampling frequency of RASTA filter as 62.5Hz. For very low frequency band 0-100Hz no filtering is performed. The filters for the band 300-2300Hz are approximated by band-pass filter with lower cut-off frequency of 1Hz and higher cutoff frequency of 15Hz. The filters for the bands 100-300 Hz and 2300-4000Hz are approximated by low pass filters with cut-off frequency of 10Hz. The design of the filters using FDATOOL in MATLAB [9, 10] is illustrated in figure 5.7 and 5.8. The frequency responses of these filters are shown in figures 5.9 and 5.10.



Structure: Direct-Form II, Second-Order Sections

Order: 4
Sections: 2
Stable: Yes
Source: Designed

○ Lowpass
○ Highpass
● Bandpass
○ Bandstop
○ Differentiator

● IIR Butterworth
○ FIR Equiripple

● Specify order: 4

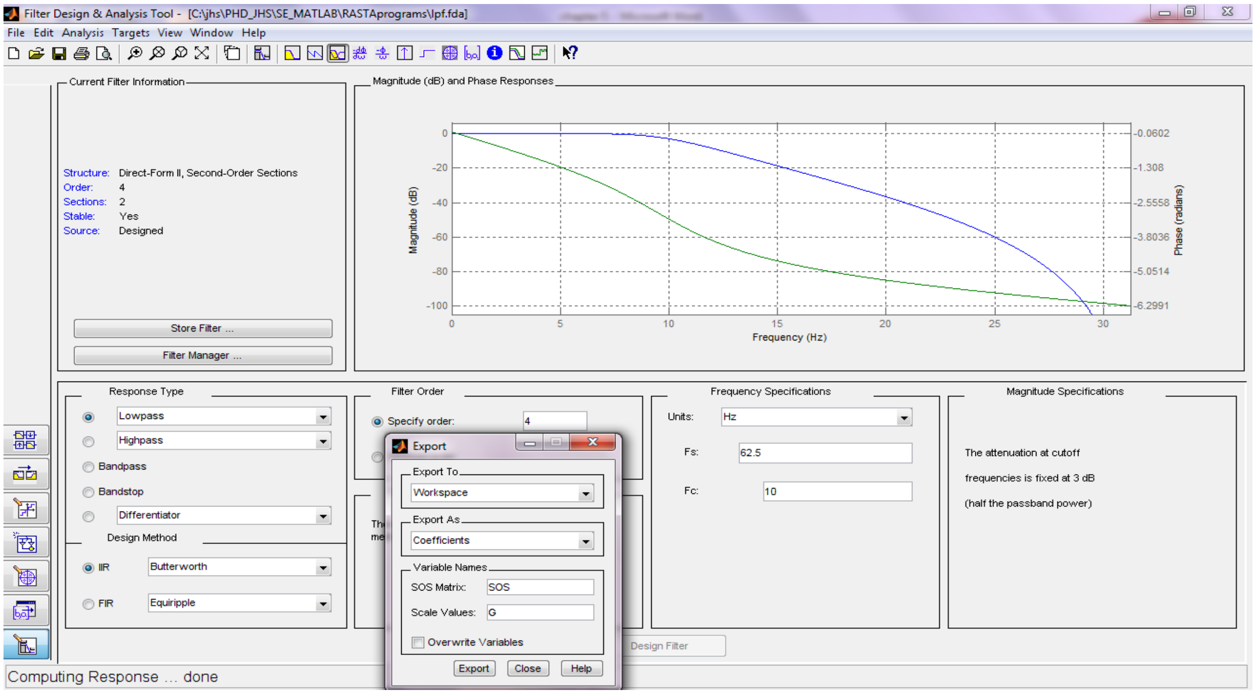
Units: Hz
Fs: 62.5
Fc1: 1
Fc2: 15

The attenuation at cutoff frequencies is fixed at 3 dB (half the passband power)

Export To: Workspace
Export As: Coefficients
Variable Names
SOS Matrix: SOS
Scale Values: G
□ Overwrite Variables

Computing Response ... done

Fig. 5.7 Design of multiband RASTA filter in 300-2300Hz band



Filter Design & Analysis Tool - [C:\jhs\PHD_JHS\SE_MATLAB\RASTAprograms\lpf.fda]

Current Filter Information

Structure: Direct-Form II, Second-Order Sections
Order: 4
Sections: 2
Stable: Yes
Source: Designed

Response Type: ● Lowpass

Design Method: ● IIR Butterworth

Filter Order: ● Specify order: 4

Frequency Specifications
Units: Hz
Fs: 62.5
Fc: 10

Magnitude Specifications

The attenuation at cutoff frequencies is fixed at 3 dB (half the passband power)

Export
Export To: Workspace
Export As: Coefficients
Variable Names
SOS Matrix: SOS
Scale Values: G
☐ Overwrite Variables

Computing Response ... done

Fig. 5.8 Design of multiband RASTA filter in 100-300Hz and 2300-4000Hz band

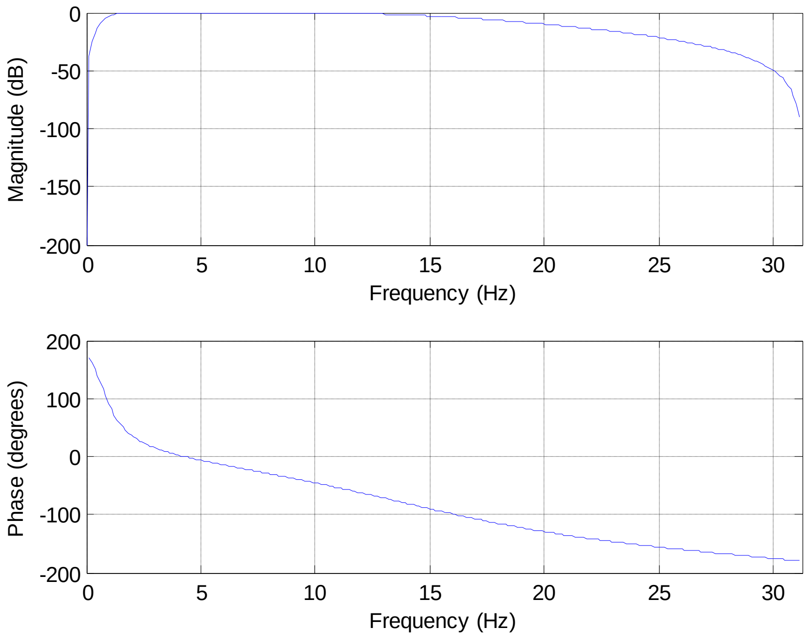


Fig. 5.9 Frequency response of multiband RASTA filter in 300-2300Hz band

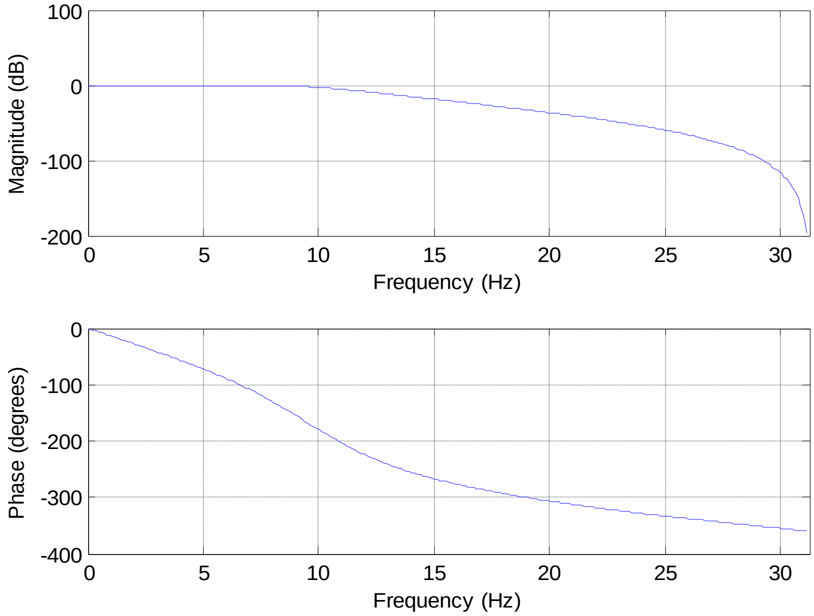


Fig. 5.10 Frequency response of multiband RASTA filter in 100-300Hz and 2300-4000Hz band

The band pass filter is designed using FDATOOL in MATLAB [9, 10]. The implementation details are given below.

```
%
% Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
%
% Generated on: 12-May-2011 18:20:50
%

% Coefficient Format: Decimal

% Discrete-Time IIR Filter (real)
% -------------------------------
% Filter Structure    : Direct-Form II, Second-Order Sections
% Number of Sections  : 2
% Stable              : Yes
% Linear Phase        : No


SOS matrix:
1  2  1  1  -0.80995029893913273  0.51158976469417017
1  2  1  1  -0.60203320225744494  0.12355934398734861

Scale Values:
0.17540986643875933
0.13038153543247591
```

Following are the implementation details of the low pass filter designed using FDATOOL in MATLAB [9, 10].

```
%
% Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
%
% Generated on: 12-May-2011 18:25:12
%

% Coefficient Format: Decimal

% Discrete-Time IIR Filter (real)
% -------------------------------
% Filter Structure    : Direct-Form II, Second-Order Sections
% Number of Sections  : 2
% Stable              : Yes
% Linear Phase        : No

SOS matrix:
1  0  -1  1  -1.8586755234480816   0.8690982221474296
1  0  -1  1  -0.13519171958723636  0.20489026841573793

Scale Values:
0.49660644391298536
0.49660644391298536
```

5.5 Objective Evaluation and Results

To test and evaluate performance of RASTA algorithms; the objective measures are obtained by using test files from NOIZEUS database [11]. Speech enhancement for white noise, and eight different colored noises at 0dB, 5dB and 10dB SNR level is carried out using two STSA algorithms: MBSS and MMSE-LSA and using the RASTA filtering algorithm with two different cases of parameters *a* and *b*.

Figures 5.11 to 5.15 illustrate SSNR, WSS, LLR and PESQ score bar chart comparison of five algorithms (two good performing STSA algorithms and three versions of RASTA algorithms explained in previous section) with eight different colored noises at 0, 5 and 10dB SNRs. Also the same comparison is given under white noise. It can be observed that at relatively high SNR two STSA algorithms are performing well compared to RASTA algorithms. But at the lower SNRs the RASTA algorithms have performance comparable to STSA algorithms. Also the performance of RASTA algorithms is consistent in white and colored noise environments. The modified RASTA algorithm performs well in most noise conditions compared to its original version. Also the RASTA algorithm has unique advantage that it does not require voice activity detector (VAD). So it can be concluded that neither STSA nor RASTA method alone is not self sufficient for noise reduction. Though RASTA method alone is not able to perform satisfactorily, its capability of suppressing the slowly varying spectral components from the noisy speech can be used for achieving better speech enhancement along with STSA based method. So it is required to combine the RASTA approach in some way with the STSA approaches to have better results. This is explained in next chapter.[1]

[1] A paper entitled "Evaluation of RASTA Approach with Modified Parameters for Speech Enhancement in Communication Systems" is presented in IEEE symposium on Computers and Informatics (ISCI 2011) Organized by IEEE Malaysia section at Kuala Lumpur, Malaysia in March 2011. ISBN: 978-1-61284-690-3. Listed and Indexed in IEEE Xplore, DOI:10.1109/ISCI.2011.5958902, pp.159-162, INSPEC 12123318.

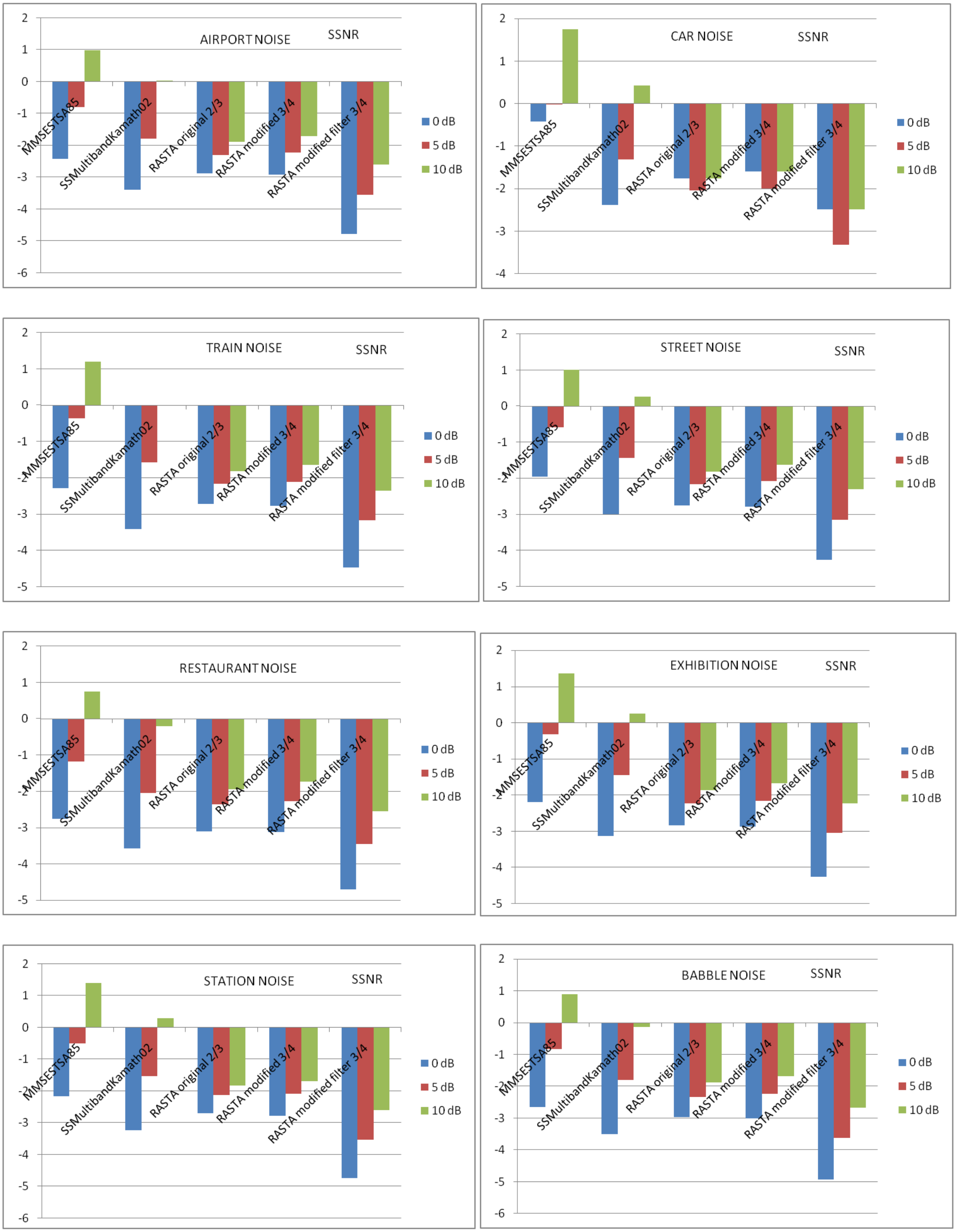


Fig. 5.11 SSNR comparison of RASTA algorithms over NOIZEUS database

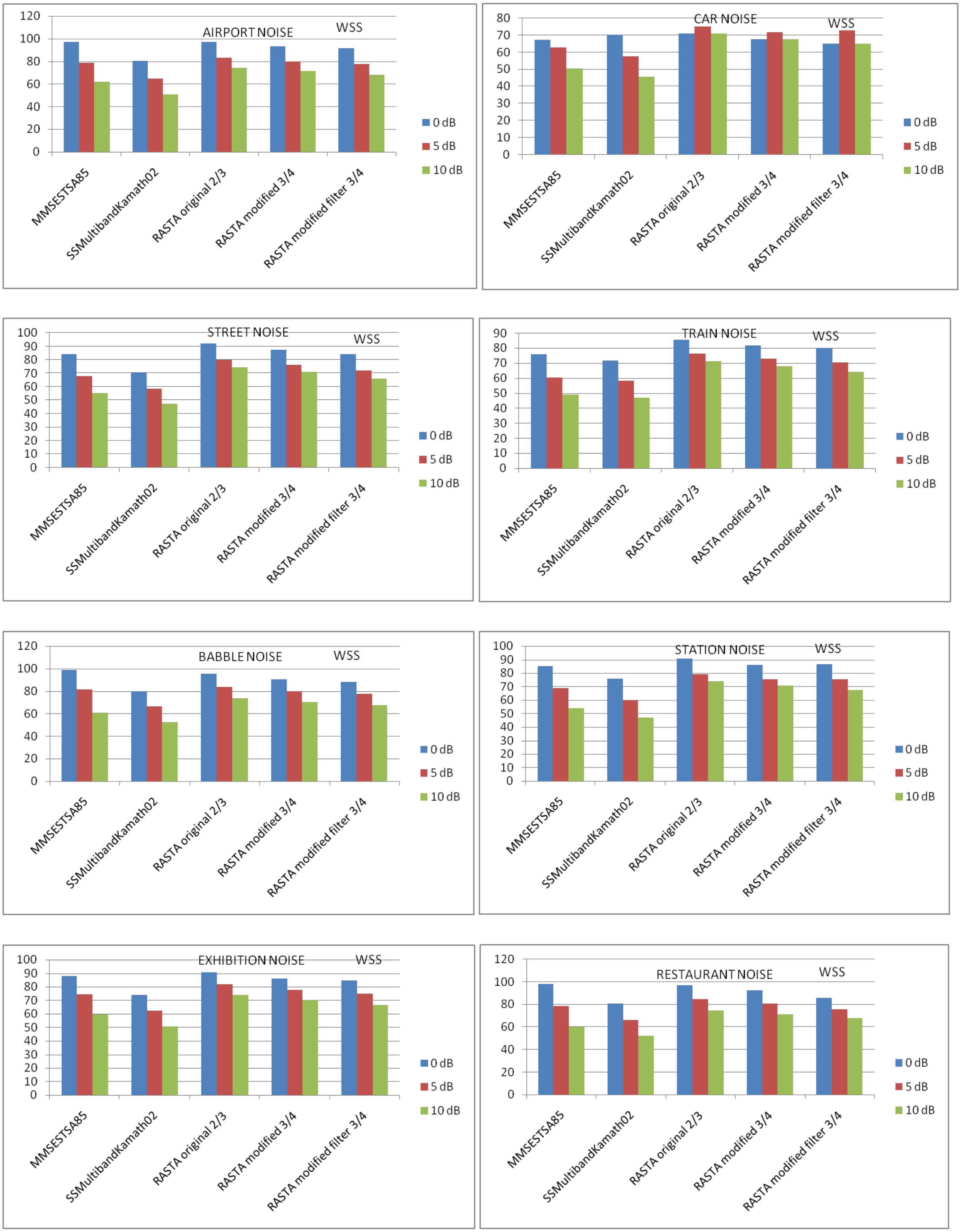


Fig. 5.12 WSS comparison of RASTA algorithms over NOIZEUS database

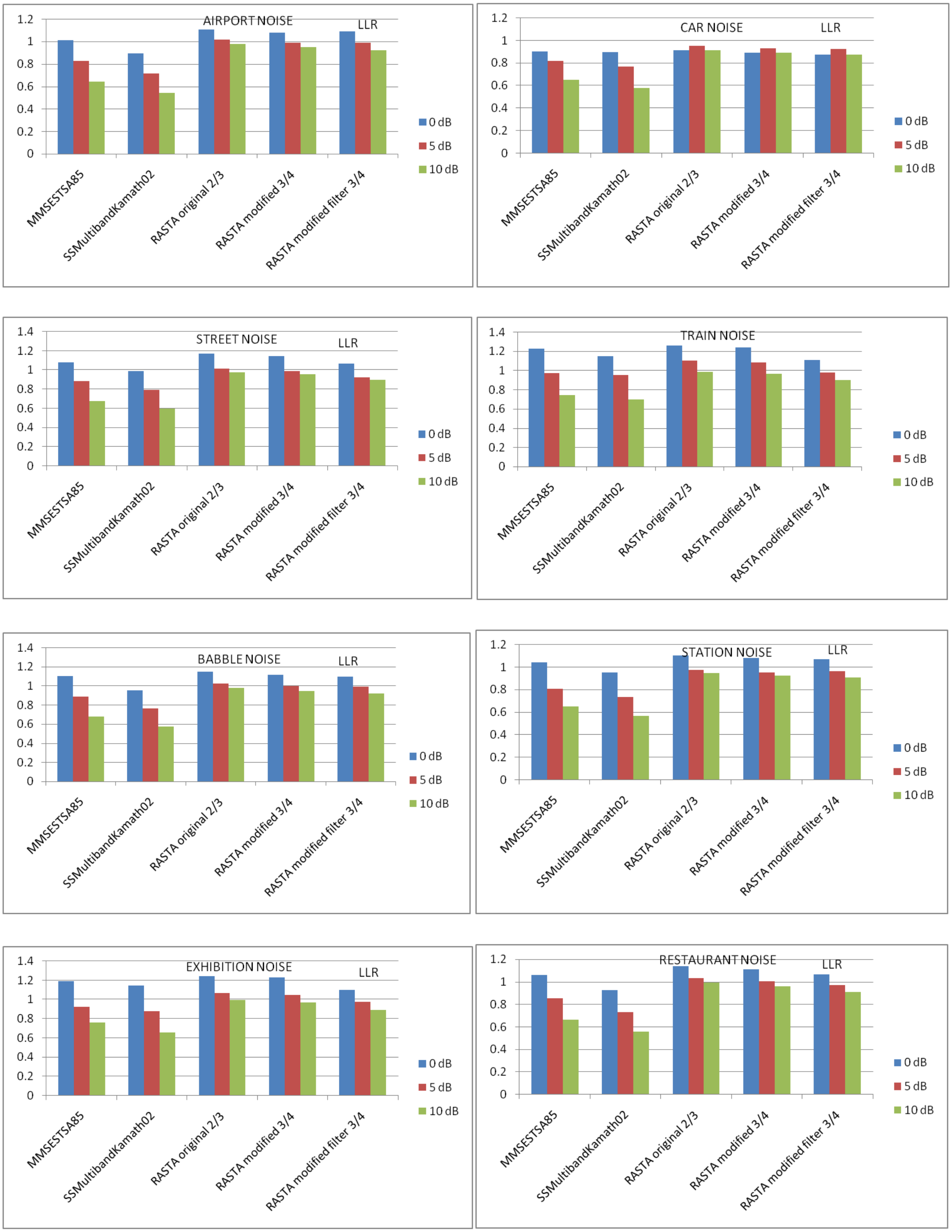


Fig. 5.13 LLR comparison of RASTA algorithms over NOIZEUS database

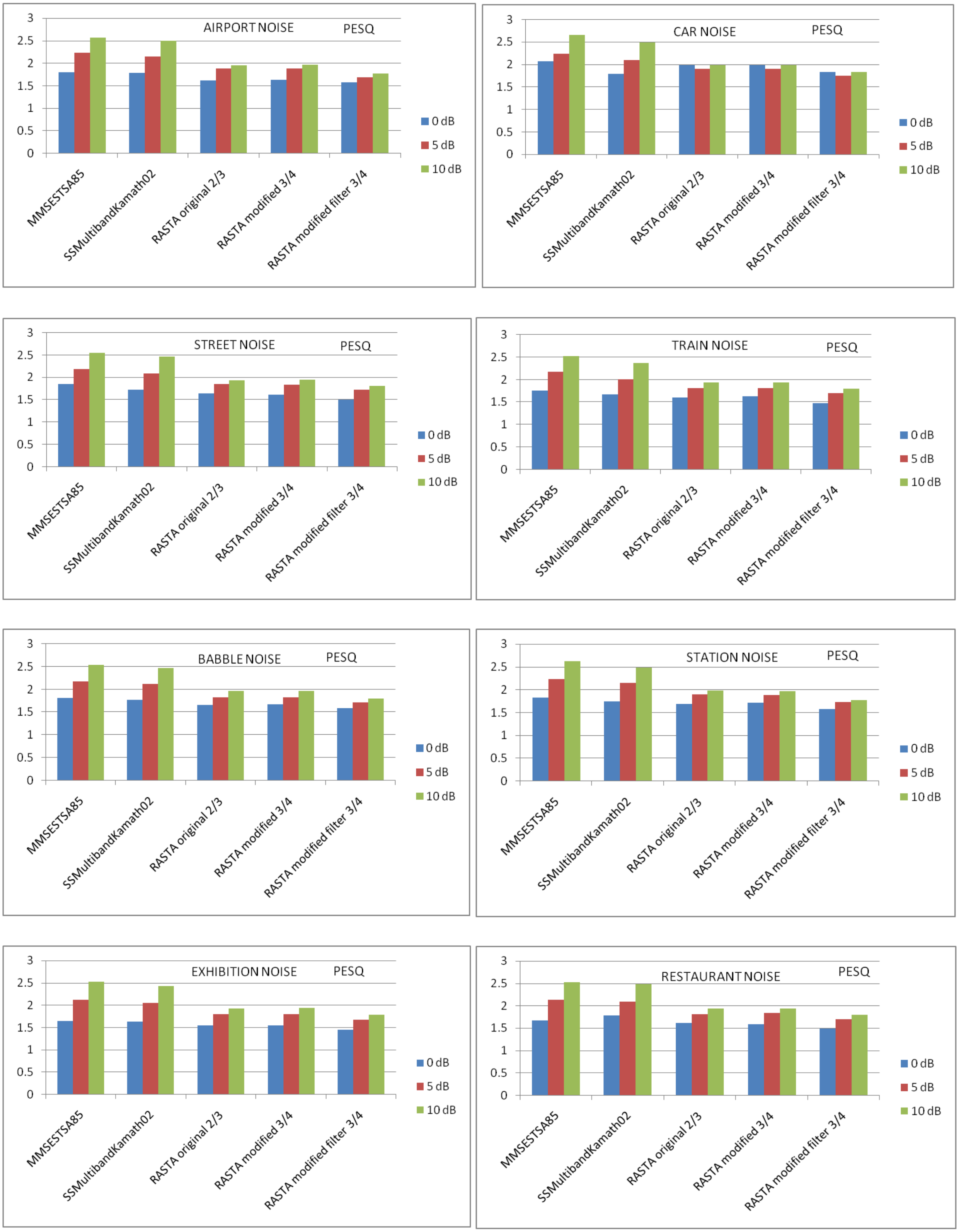


Fig. 5.14 PESQ comparison of RASTA algorithms over NOIZEUS database

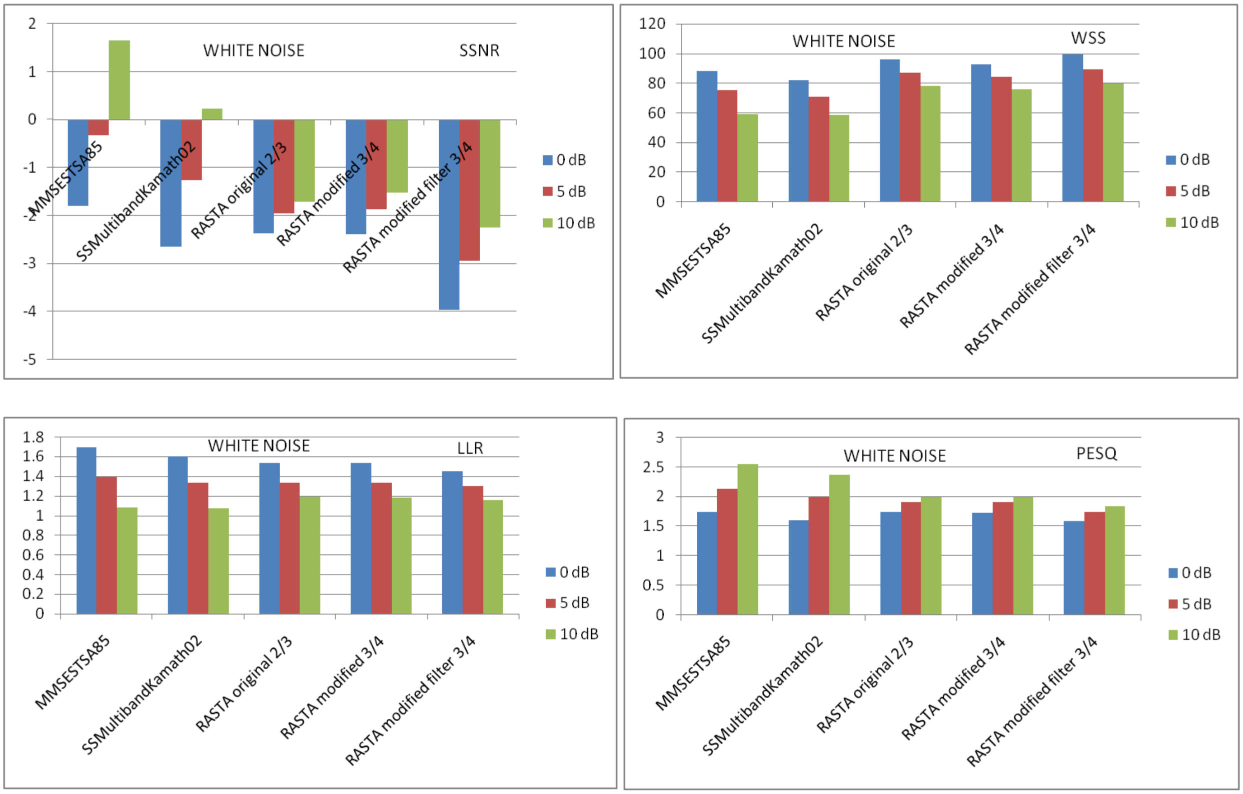


Fig. 5.15 Objective evaluation of RASTA algorithms under white noise conditions

Figure 5.16 also shows the spectrogram of the enhanced speech form 0dB white noise using various RASTA algorithms. The clean speech signal spectrogram and noisy signal spectrogram are shown in figure 4.5. The comparison is self explanatory that modified RASTA with modified filter algorithm is better speech enhancer of the three. Also it can be compared with spectrogram of enhanced speech by various STSA algorithms as shown in figure 4.6. The performance of modified RASTA with modified filter algorithm is comparable with the MMSE LSA (STSA85) and MBSS – the two outstanding STSA algorithms. Also it can be seen that the speech enhanced by RASTA methods have very high residual noise during the initial portion of the enhanced speech (Initial 4 to 10 frames). Generally this initial period is assumed to be silence period and not much more to bother about it. After this period the filters get initialized and residual noise starts reducing. Like STSA methods RASTA processed speech also generates musical noise and speech distortion as seen from the spectrograms. So a better performance can be expected by combining these two different approaches.

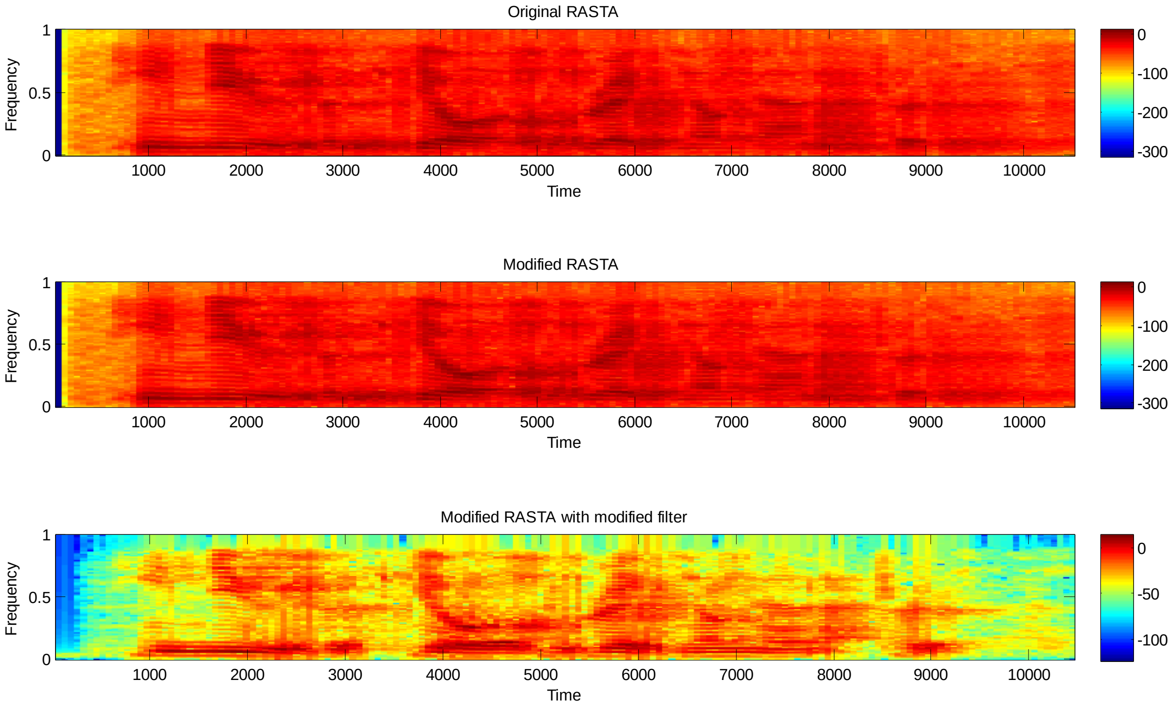


Fig. 5.16 Spectrogram of enhanced speech signal using various RASTA algorithms containing sentence 'He knew the skill of the great young actress'

5.6 Summary

The RASTA algorithm which utilizes temporal processing and auditory features is discussed and simulated here for performance evaluation. The original RASTA algorithm is used for de-convolution and later on it is modified for additive noise removal. However, the objective measures indicate the poor performance of RASTA algorithm compared to MMSE STSA85 algorithm. The original fixed RASTA filter is modified to multiband filter. The static nonlinear compression and expansion factor is also moderately changed to alleviate the additive noise. Even with the modification the RASTA algorithm gives distortion in output speech compared to STSA counterpart. But the positive outcome is the improvement at lower SNRs. So it is suggested here to use the hybrid algorithm which combines STSA and RASTA approach by some means. This hybrid approach is discussed in next chapter.

Chapter 6

Hybrid Algorithm for Performance Improvement

It was suggested in last chapter that for better performance the STSA algorithm can be combined in some way with RASTA approach. The best performing STSA algorithm is MMSESTSA85 (MMSE-LSA) as discussed in chapter 4. It is combined with modified RASTA multiband filter approach which is evaluated in chapter 5. The hybrid algorithm is proposed here and also it is simulated and tested under different additive noise conditions using the NOIZEUS database and compared with the original algorithms. The results of performance evaluation using objective measures are described in this chapter. The comparison using alone objective measures is not sufficient as it will not ensure the quality of speech signal for human listeners and hence the subjective evaluation is also required to perform. The IEEE recommended and ITU-R BS.562-3 standard mean opinion score (MOS) listening test is carried out. The chapter describes the various guidelines followed to perform this test. The original and modified algorithms are compared based on this test and conclusion is made regarding quality of output of different algorithms.

Reverberation is one type of convolutive distortion that occurs commonly in communication systems. The speech enhancement algorithm must be able to tackle it. The proposed algorithm is also tested under different reverberation condition using the Aachen impulse response (AIR) database developed by RWTH Aachen University, institute of communication systems and data processing (India). It is a set of impulse responses that were measured in a wide variety of rooms. This database allows realistic studies of signal processing algorithms in reverberant environments. The comments are made about performance of algorithms in the simulated reverberant conditions.

6.1 Proposed New Approach

The proposed modified approach for speech enhancement uses combination of MMSE STSA85 algorithm and multiband RASTA filter. The connection is not simple cascade but the blocks are interacting as shown in figure 6.1. The noisy speech is presented simultaneously to both multiband RASTA and MMSE STSA85 algorithms. The VAD is required to estimate speech/silence segment for MMSE STSA85 algorithm. This block is responsible for malfunctioning of algorithm if the detection is false. The MMSE STSA85 algorithm is highly dependent of VAD false rate. So VAD is not directly getting the noisy speech for estimation but the output of multiband RASTA filter is given to VAD for estimation. The RASTA approach does not require VAD and reduce the noise moderately as discussed in chapter 5. Some speech distortion and musical and residual noise remain in enhanced speech by RASTA algorithm. However, the VAD can now better detect the speech/silence segment compared to direct detection from noisy speech. But the white noise after RASTA filtering gets converted into colored noise with sharp spectral peaks. Hence, the accuracy in noise estimation reduces; this causes the rise in musical noise. So the noise power is estimated for RASTA filtered as well as original noisy speech spectrum. The ratio of original noise power to the filtered noise power (PR) is calculated and it is used to calculate a priori SNR. A mild linear compression is required to avoid over suppression. The modified decision direct rule taking this factor into consideration is given by following equation for frame *t*.

$$\xi^{(t)}(K) = \eta \frac{\left|\hat{X}^{(t-1)}(K)\right|^2}{\left|\hat{D}^{(t)}(K)\right|^2/PR} + (1-\eta)max(\bar{\gamma}^{(t)}(K) - 1,0) \tag{6.1}$$

$$where;\ \bar{\gamma}(K) = \frac{|\bar{Y}(K)|^2}{\left|\bar{\hat{D}}(K)\right|^2/\mathrm{PR}}$$

The enhanced speech obtained after this modification has almost no musical noise.

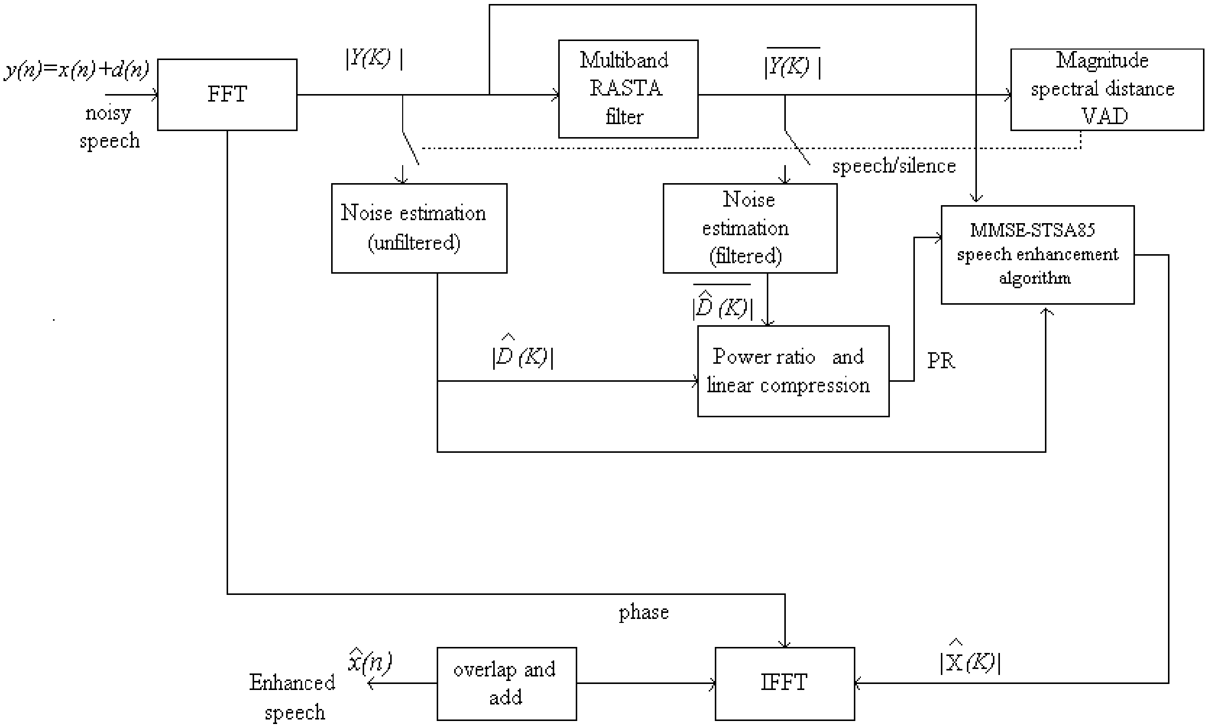

y(n)=x(n)+d(n)
noisy speech
FFT
|Y(K) |
Multiband RASTA filter
$|\overline{Y(K)}|$
Magnitude spectral distance VAD
speech/silence
Noise estimation (unfiltered)
Noise estimation (filtered)
MMSE-STSA85 speech enhancement algorithm
$|\overline{\hat{D}(K)}|$
$|\hat{D}(K)|$
Power ratio and linear compression
PR
phase
$\hat{x}(n)$
Enhanced speech
overlap and add
IFFT
$|\hat{X}(K)|$

Fig. 6.1 Block diagram of proposed speech enhancement method

)

6.2 MATLAB Implementation of Proposed Algorithm

The input speech sampled at 8 KHz is applied to 32ms hamming window with 50% overlap and 256 point FFT is applied. From complex FFT the magnitude and phase are separated. Due to symmetry property 128 point spectral values are filtered using multiband RASTA with nonlinear compression parameter a=3/4 and expansion parameter b=4/3. The filter is initialized with zero values. The filtered input speech spectrum is used by magnitude spectral distance VAD to identify the current frame as speech/silence. If the current frame is silence frame, the filtered as well as unfiltered noise estimate is updated by using noise estimation rule described in section 4.2. The power ratio is calculated and linear compression is applied to avoid over suppression. The linear compression is implemented using straight line equation and it ensures the ratio to be between 1 and 2. The actual speech enhancement is performed by MMSE STSA85 method. The enhanced spectral values are combined with the phase of the noisy spectrum. 256 point IFFT is applied and overlap add synthesis is performed to reconstruct the speech signal as final output.

6.3 Spectrographic and Objective Evaluation of Proposed Algorithm

The spectrogram of the enhanced speech for the clean speech with spectrogram shown in figure 4.5 and subjected to 0dB white noise is enhanced by the proposed approach. The spectrogram of the proposed approach is shown in figure 6.2. Comparison of this with the spectrograms of speech enhanced by MMSE STSA85 (figure 4.6) and with the modified RASTA filter (figure 5.16) indicates that the speech enhanced by using proposed approach more closely resembles to the clean speech signal. Still there are some randomly distributed spots present in the enhanced speech spectrogram which results in small level of musical noise. The residual noise is very less compared to two original algorithms.

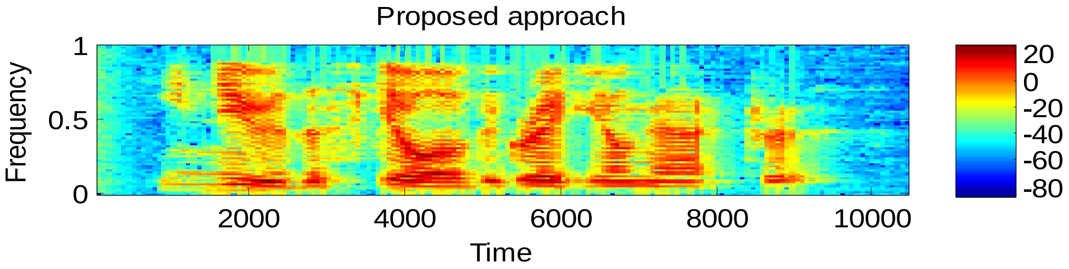


Fig. 6.2 Spectrogram of enhanced speech signal using proposed approach

The objective quality measures SSNR, WSS, LLR and PESQ observed over 0dB, 5dB and 10 dB SNRs using NOIZEUS database [1] are given in figures 6.3 to 6.7 in the form of bar chart. As mentioned in section 4.5 the number of test runs on each algorithm is 810. The comparison of proposed approach is done with MMSE STSA85 and Modified RASTA filter algorithms. The quality measures for noisy speech and maximum theoretical limits (obtained by using clean magnitude and noisy phase) are also included for comparison.

The WSS measure indicates the spectral distortion in the speech and the comparison shows that in all types of noise conditions at 0 and 5dB SNRs the proposed algorithm gives good improvement. Except in white noise and airport noise condition at 10 dB SNR the WSS for proposed algorithm is improved in all other noises at 10 dB SNR. The LLR measure is better for proposed approach in all noises at all SNRs compared to MMSE STSA85 algorithm. The PESQ score at 0dB SNR is comparable with MMSE STSA85 algorithm in most of the cases and in few cases it shows improvement. For restaurant noise it is noticeably improved. For 5 dB SNR this measure slightly degrades compared to original MMSE STSA85 algorithm in all cases but it is marginal. For 10 dB SNR this measure shows some degradation in all cases. Putting these results altogether; it is noticed that at low SNR levels like 0 to 5dB the proposed approach gives better performance while at higher SNR levels (≥10 dB) the original MMSE STSA85 algorithm performs better. Also the proposed algorithm outperforms the original algorithm in car noise, restaurant noise and train noise conditions. In these kinds of noisy environments the person using communication equipment has to combat with surroundings from the confined area only and the SNRs in such situation are always weak. As the primary goal of this research work is to design an algorithm for low SNR conditions the proposed approach is recommended to use in such circumstances.[1] However, the comments made here are still based on objective measures only; but this needs to correlate well with subjective listening tests which involves the human beings. For that it is required to do the subjective evaluation of algorithms [2]. The procedure and the experiment conducted for this purpose is explained in next section.

[1] A paper entitled "Objective Evaluation of STSA Based Speech Enhancement Techniques for Speech Communication Systems with Proposed" is presented in IEEE International conference on Communication, Network and Computing (CNC 2010) Organized by ACEEE at Calicut in October 2010. IEEE CS- CPS ISBN: 978-0-7695-4209-6. Listed in IEEE Xplore by IEEE Computer Society, DOI:10.1109/CNC.2010.13, pp.19-23. Archived in ACM digital library.

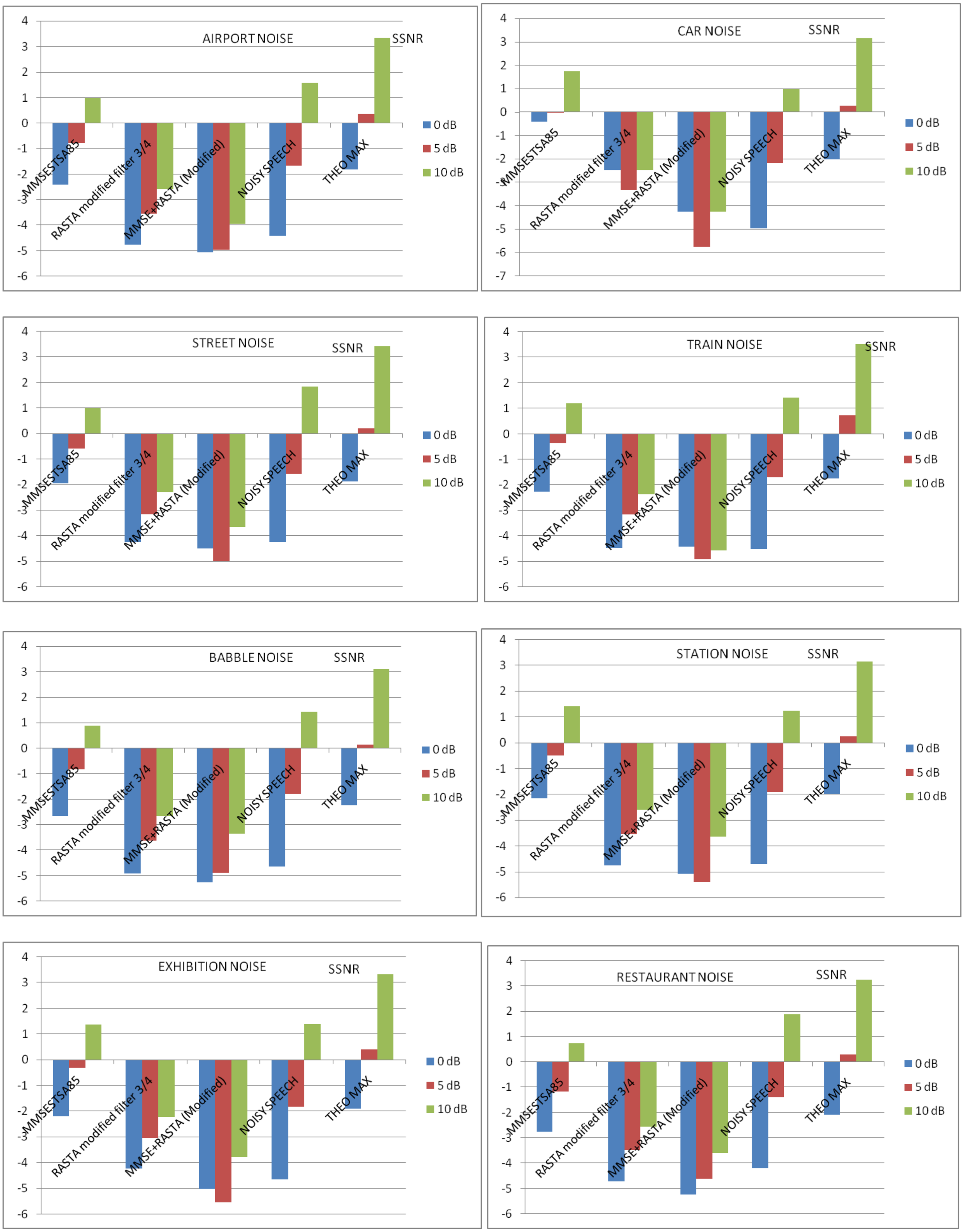


Fig. 6.3 SSNR comparison of proposed algorithm over NOIZEUS database

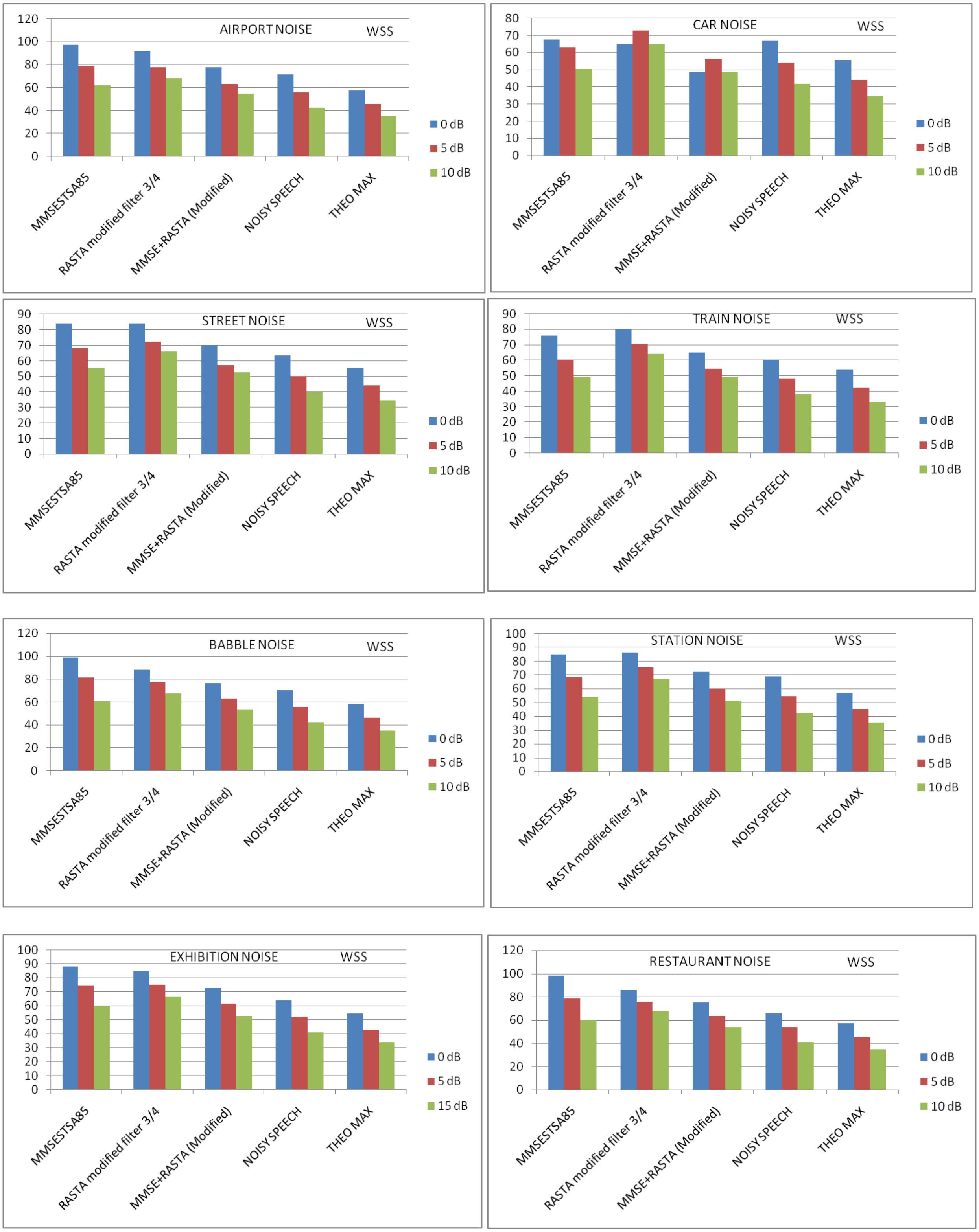


Fig. 6.4 WSS comparison of proposed algorithm over NOIZEUS database

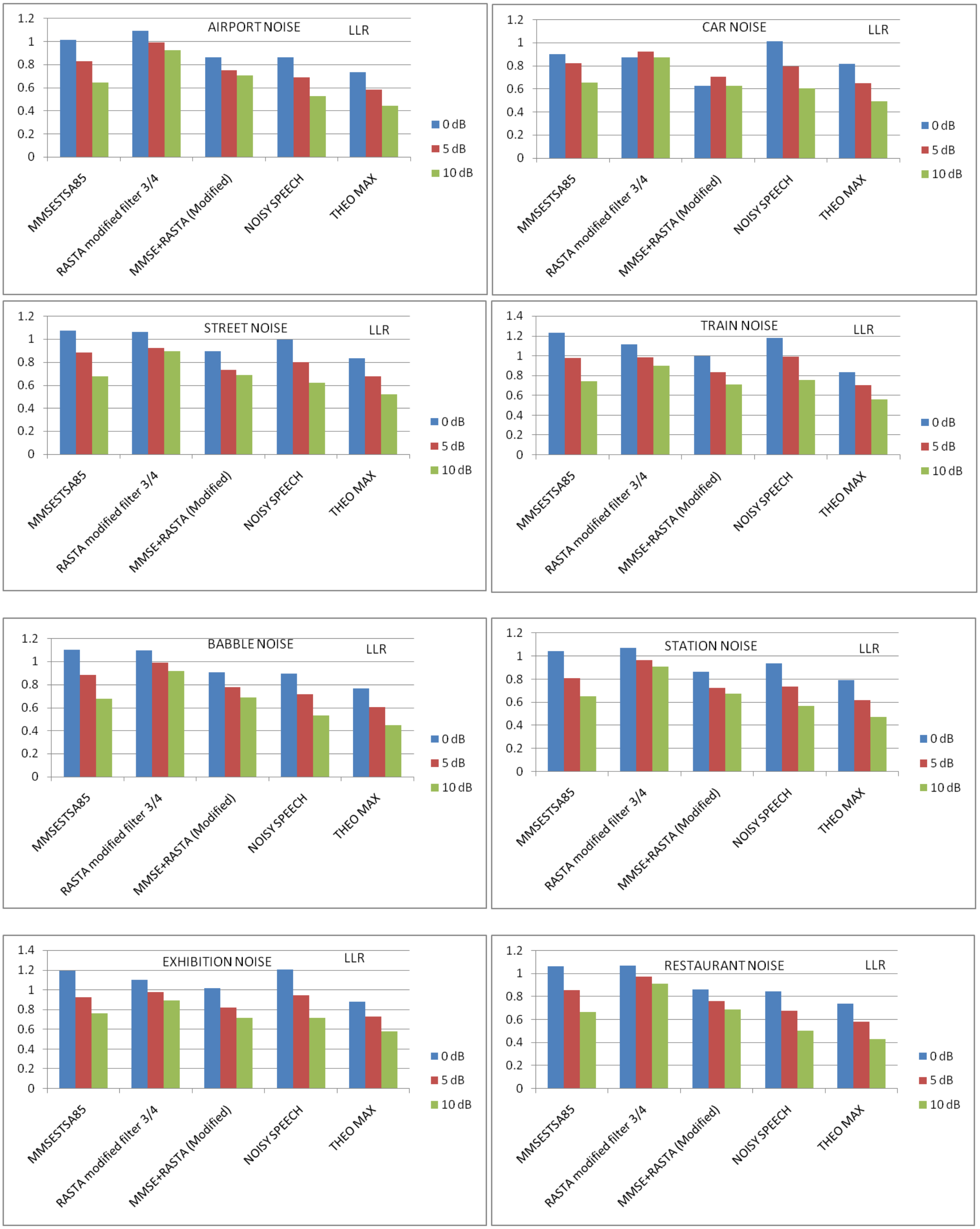


Fig. 6.5 LLR comparison of proposed algorithm over NOIZEUS database

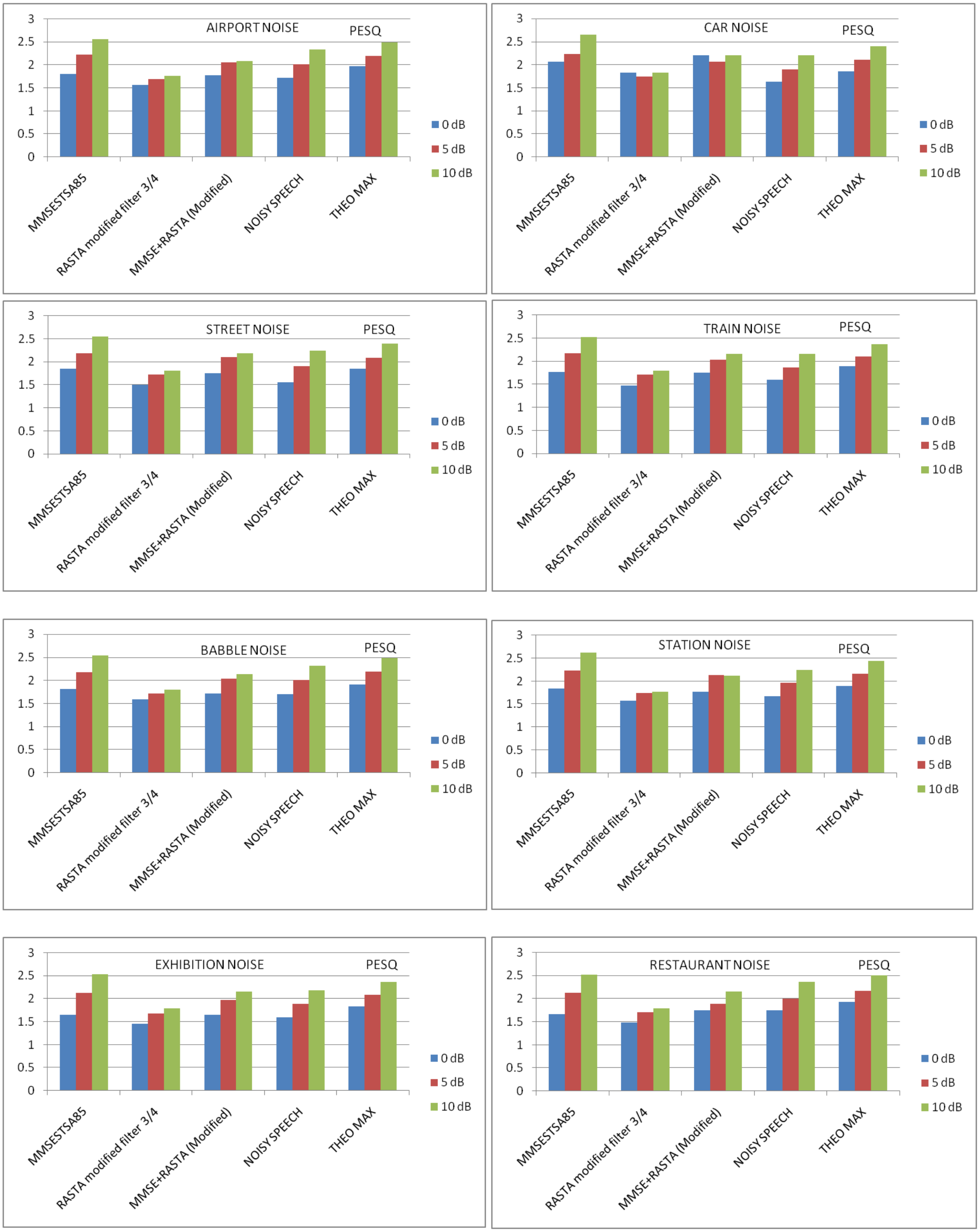


Fig. 6.6 PESQ comparison of proposed algorithm over NOIZEUS database

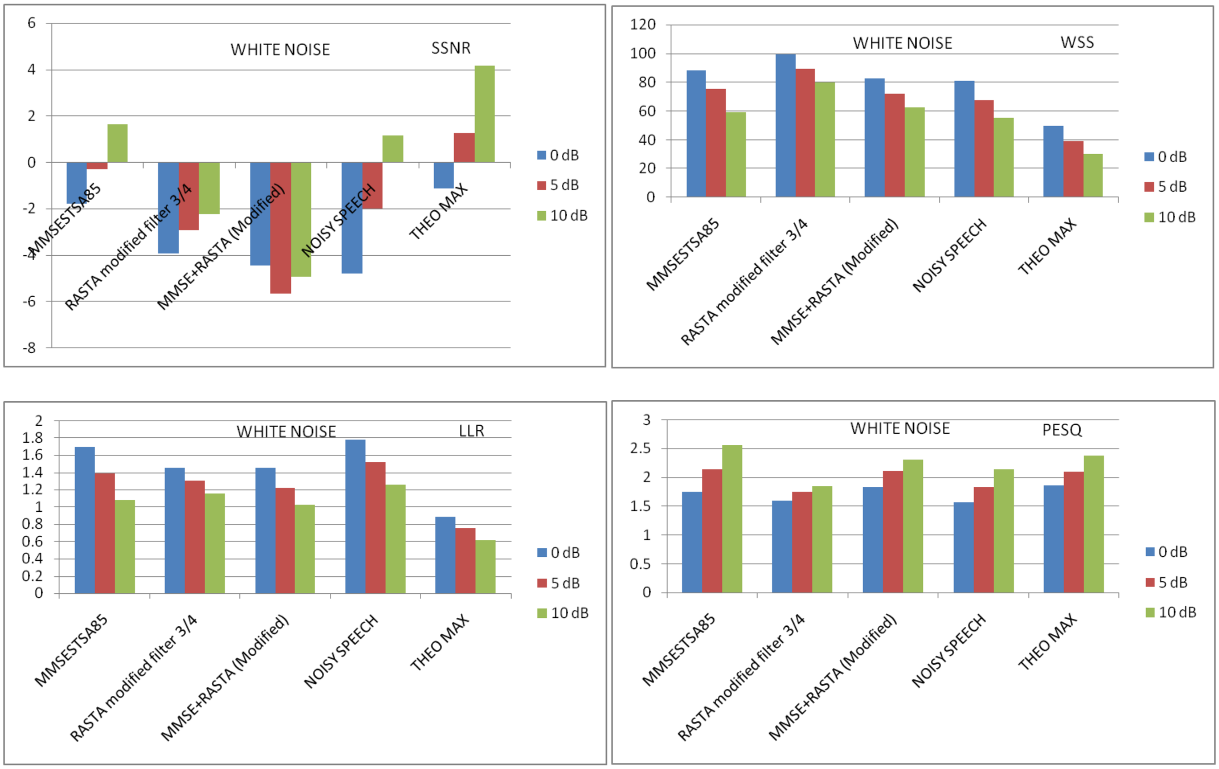


Fig. 6.7 Objective evaluation of proposed algorithm under white noise

6.4 Subjective Evaluation

Subjective evaluation involves comparisons of original and processed speech signals by a group of listeners who are asked to rate the quality of speech along a predetermined scale. The most widely used direct method of subjective quality evaluation is the category judgment method in which listeners rate the quality of the test signal using a five-point numerical scale as shown in table 6.1, with 5 indicating “excellent” quality and 1 indicating “unsatisfactory” or “bad” quality. This method is one of the methods recommended by IEEE subcommittee on Subjective Methods [3] as well as by ITU [5, 6]. The measured quality of the test signal is obtained by averaging the scores obtained from all listeners. This average score is commonly referred to as the Mean Opinion Score (MOS).

The MOS test is administered in two phases: training and evaluation. In the training phase, listeners hear a set of reference signals that exemplify the high (excellent), the low (bad), and the middle judgment categories. This phase, also known as the “anchoring phase,” is very important as it is needed to equalize the subjective range of quality rating of all listeners- that is, the training phase should in principle equalize the “goodness” scales of all listeners to ensure, to the extent possible, that what is perceived as "good" by one listener is also perceived as "good" by the other listeners. A standard set of reference signals need to be used and described when reporting the MOS scores [3]. In the evaluation phase, subjects listen to the test signal and rate the quality of the signal in terms of the five quality categories (1-5) shown in table 6.1. Reference signals can be used to better facilitate comparison between MOS tests conducted at different times, different laboratories, and different languages [4].

Rating	Speech Quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying, but not objectionable
1	Bad	Very annoying and objectionable

Table 6.1 MOS rating scale

Detailed guidelines and recommendations for administering the MOS test can be found in the ITU-R BS.1116-1 standard [5] and include:

1. *Selection of listening crew*: Different number of listeners is recommended, depending on whether the listeners have had extensive experience in assessing sound quality. Minimum number of non expert listeners should be 20, and minimum number of expert listeners should be 10.
2. *Test procedure and duration*: Speech material (original and degraded) should be presented in random order to subjects, and the test session should not last more than 20 minutes without interruption. This step is necessary to reduce listening fatigue.
3. *Choice of reproduction device*: Headphones are recommended over loudspeakers, as headphone reproduction is independent of the geometric and acoustic properties of the test room. If loudspeakers are used, the dimensions and reverberation time of the room need to be reported.

6.5 Setup for Subjective Evaluation

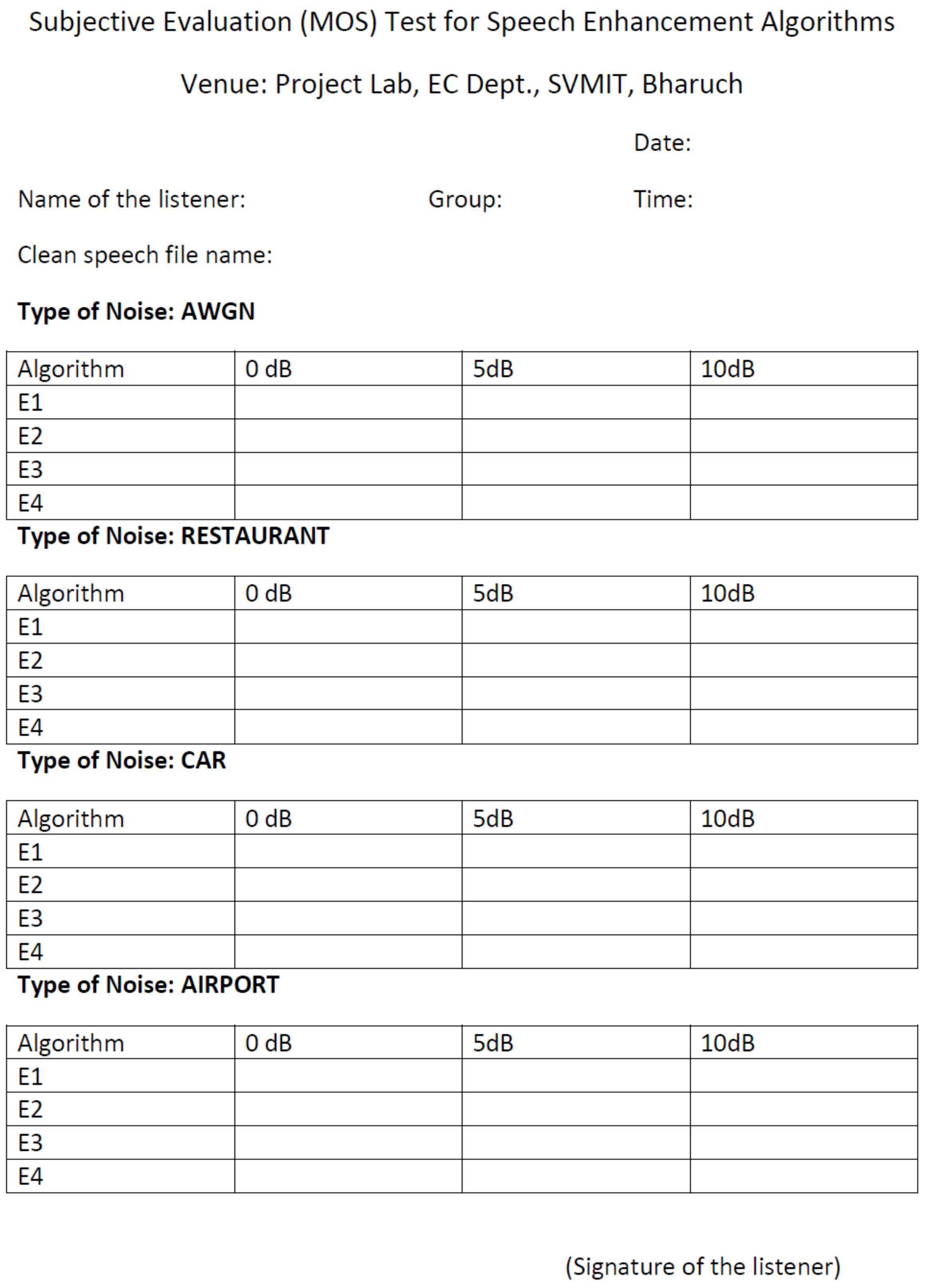
For subjective evaluation four algorithms namely MMSE STSA85, wavelet de-noising, modified RASTA filter and proposed algorithm (combination of MMSE STSA85 and modified RASTA filter) are selected. The speech sentences from NOIZEUS database are selected contained in files sp02.wav (male speaker) and sp11.wav (female speaker) mentioned in table

4.1. The speech sentences corrupted by white noise, restaurant noise, car noise and airport noise at 0, 5 and 10dB SNRs are selected for enhancement. As mentioned in section 6.4 the performance of proposed algorithm is very good in car noise and restaurant noises but inferior in white noise and airport noise conditions compared to original MMSE STSA85 algorithm. So, all these four types of noises are selected for subjective evaluation. However, it can be extended for all other types of noises but to complete the test as per guidelines mentioned in section 6.5 within stipulated time the restrictions are applied.

For conducting the MOS test following procedure is obeyed:

1. *Selection of listening crew*: Total 20 listeners are selected having age in between 19 years to 38 years. It includes 9 undergraduate final year Electronics and communication engineering students, 9 faculty members of Electronics and communication engineering department and 2 laboratory assistants from S.V.M. Institute of Technology, Bharuch. The crew includes 13 male and 7 female listeners.
2. *Test procedure and duration*: The listeners are presented with clean speech file, noisy speech file and enhanced speech file by each algorithm. The care is taken when the enhanced speech files are named so that the identity of the algorithm remains undisclosed. The file names are not reflecting the type and name of algorithm by any means. The listeners are having freedom to play the clean, noisy and enhanced speech files at any time during the test. This is done to eliminate the overlay effect of the previously listened speech.
3. *Choice of reproduction device*: Good quality headphones are provided to each listener. The test is conducted in project laboratory of electronics and telecommunication engineering department of S.V.M. Institute of Technology, Bharuch in quiet environment.

The pro forma for filling up the MOS test score for different algorithms is shown in figure 6.8.



Subjective Evaluation (MOS) Test for Speech Enhancement Algorithms

Venue: Project Lab, EC Dept., SVMIT, Bharuch

Date:

Name of the listener: Group: Time:

Clean speech file name:

Type of Noise: AWGN

Algorithm	0 dB	5dB	10dB
E1			
E2			
E3			
E4			

Type of Noise: RESTAURANT

Algorithm	0 dB	5dB	10dB
E1			
E2			
E3			
E4			

Type of Noise: CAR

Algorithm	0 dB	5dB	10dB
E1			
E2			
E3			
E4			

Type of Noise: AIRPORT

Algorithm	0 dB	5dB	10dB
E1			
E2			
E3			
E4			

(Signature of the listener)

Fig. 6.8 Pro forma for filling up the MOS

6.6 Subjective Evaluation of Proposed Algorithm

Figure 6.9 shows the MOS test results obtained for various algorithms. The comparison shows that wavelet de-noising is the worst algorithm in all four algorithms. The proposed algorithm give high MOS scores for 0 and 5dB SNRs in all noise conditions. For 10dB SNR the performance of proposed algorithm is comparable with the original MMSESTSA85 algorithm. Hence the proposed algorithm performs well in low SNR conditions compared to original algorithm. This validates the results obtained from objective measures.

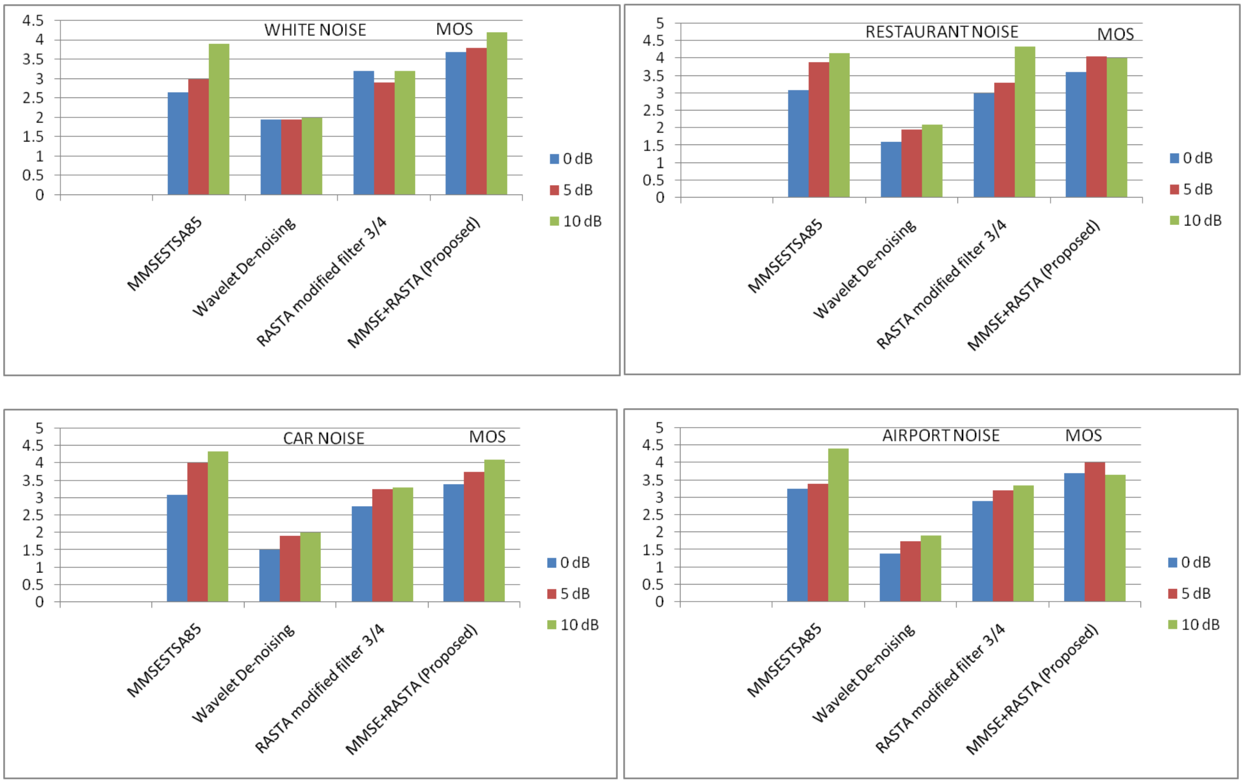


Fig. 6.9 Results of MOS test

6.7 Evaluation of Proposed Algorithm in Reverberant Environments

So far the objective and subjective evaluation is carried out using NOIZEUS database which contains the speech sentences corrupted with additive noise. In real circumstances the additive noise is not only the corrupting factor but some reverberation is also present. For wireless mobile communication systems the reverberant environment will change as the user moves from place to place. Hence it is required to test the proposed algorithm under different reverberant conditions. To test the algorithm in simulated reverberation environment a database called the Aachen Impulse Response (AIR) database is used [7].

It is a set of impulse responses that were measured in a wide variety of rooms. The initial aim of the AIR database was to allow for realistic studies of signal processing algorithms in reverberant environments with a special focus on hearing aids applications. It offers binaural room impulse responses (BRIR) measured with a dummy head in different locations with different acoustical properties, such as reverberation time and room volume. Besides the evaluation of de-reverberation algorithms and perceptual investigations of reverberant speech, this part of the database allows for the investigation of head shadowing influence since all recordings where made with and without the dummy head. Since de-reverberation can also be applied to telephone speech, it also includes (dual channel) impulse responses between the artificial mouth of a dummy head and a mock-up phone. The measurements were carried out in compliance with the ITU standards for both the hand held and the hands free position.

A MATLAB reference implementation is available at [7]. All impulse responses of the AIR database are stored as double precision binary floating point MAT-files which can be directly imported into MATLAB.

Table 6.2 shows the parameters to be specified to obtain a particular room impulse response. The clean speech signal can be convolved with this impulse response to generate the reverberant speech in particular environment. Table 6.3 specifies the combination of parameters used in the evaluation of proposed algorithm.

Parameter	Structure of parameter
Type of impulse response	rir_type '1': binaural (with/without dummy head) acoustical path: loudspeaker -> microphones next to the pinna '2': dual-channel (with mock-up phone) acoustical path: artificial mouth of dummy head-> dual-microphone mock-up at hand held or hands free position
Room type	room 1,2,..,10: 'booth', 'office', 'meeting', 'lecture', 'stairway','stairway1','stairway2', 'corridor','bathroom','lecture1' Available rooms for (1) binaural: 1,2,3,4,5 (2) phone: 2,3,4,6,7,8,9,10
Select channel	channel '0': right; '1': left
Select RIR with or without dummy head (for 'rir_type=1' only)	head '0': no dummy head; '1': with dummy head
Position of mock-up phone (for 'rir_type=2' only)	phone_pos '1': HHP (Hand-held), '2': HFRP (Hands-free)

RIR number (increasing distance, for 'rir_type=1' only)	rir_no
	Booth: {0.5m, 1m, 1.5m}
	Office: {1m, 2m, 3m}
	Meeting: {1.45m, 1.7m, 1.9m, 2.25m, 2.8m}
	Lecture: {2.25m, 4m, 5.56m, 7.1m, 8.68m, 10.2m}
	Stairway: {1m, 2m, 3m}

Table 6.2 Specification of parameters for generation of impulse response

Reverberant Environment	rir_type	Room	phone_pos	rir_no
Reverb1	dual-channel	Office	Hands-free	3m
Reverb2	binaural	Booth	Hand-held	1m
Reverb3	binaural	Meeting	Hands-free	2.25m
Reverb4	dual-channel	Bathroom	Hands-free	------
Reverb5	dual-channel	lecture1	Hands-free	2.25m

Table 6.3 Set of parameters for testing proposed algorithm in reverberant environments

Table 6.4 shows the comparison of MMSE STSA85 and proposed algorithm under the simulated reverberation environments. The table clearly indicates that the WSS and LLR score for the proposed algorithm is very less compared to MMSE STSA85 which means very less distortion present in the enhanced speech. The PESQ score also shows improvement for proposed algorithm in all different reverberation conditions. However, the reverberation is not much bother to human listener as intelligibility is preserved in the reverberant speech; but it is much more significant for automatic speech recognizers. Hence the proposed algorithm can be used in speech communication systems as well as preferred as a preprocessing stage in ASR.

Reverberant Environment	**Algorithm**	**SSNR**	**WSS**	**LLR**	**PESQ**
Reverb1	MMSE STSA85	-0.9959	55.4408	0.9366	2.4614
	Proposed	-8.9914	34.6862	0.5034	2.6324
Reverb2	MMSE STSA85	-9.8105	47.9322	0.7945	3.1278
	Proposed	-8.4320	31.7140	0.2780	3.3966
Reverb3	MMSE STSA85	-9.8257	53.4960	0.8777	2.6979
	Proposed	-8.8671	37.8436	0.5962	2.8037
Reverb4	MMSE STSA85	-0.8166	62.3047	0.9325	2.7068
	Proposed	-8.8137	39.5675	0.5355	2.7890
Reverb5	MMSE STSA85	-0.4522	46.1798	0.8741	2.7540
	Proposed	-9.2058	24.8601	0.4645	2.9386

Table 6.4 Objective evaluation of proposed algorithm in reverberant environments

6.8 Summary

The combination of STSA and RASTA approach is termed here as hybrid approach which is proposed algorithm to improve the performance at lower SNRs (0-5dB). The performance evaluation using objective measures shows the improvement at lower SNRs compared to original STSA algorithm. The subjective listening tests also back the result. The proposed algorithm also found more superior compared to original algorithm under reverberant environments. Hence it is recommended to use hybrid approach in low SNR conditions and reverberant environments. However, the RASTA algorithm is non linear and non causal which throws the challenge for real time and hardware implementation. This is dealt in next chapter.

Chapter 7

Hardware Implementation Tools

The testing and embedding speech processing algorithm on general purpose PC and dedicated DSP platform require specific hardware implementation tools. Real time digital signal processing made considerable advancements after the introduction of specialized DSP processors. Suitable starter kits with a specific DSP processor and related software tools such as compilers, assemblers, simulators, debuggers, and so on, are provided in order to make system design and application development easier. The 32-bit floating point processor TMS320C6713 from Texas Instruments is very powerful for real time speech and audio processing algorithm implementations. This DSP processor is based on the VLIW (Very Large Instruction Word) technology, which allows fast parallel computing jointly using its optimized "C" compiler. For a rapid evaluation of the TMS320C6713 processor a Developer Starter Kit 6713 (DSK 6713) is available from Spectrum Digital Incorporation; comprises a board and the software tools. The board must be connected to a standard PC running under its integrated development environment- Code Composer Studio (CCS IDE). For rapid prototyping, testing and debugging of developed algorithm the Real Time Workshop (RTW) toolbox and MATLAB link for embedded target called Target Support Package TC6 is used.

7.1 The Digital Signal Processor: TMS320C6713

The TMS320C6000 platform of digital signal processors (DSPs) is part of the TMS320 family of DSPs. The TMS320C67x (’C67x) devices are floating-point DSPs in the TMS320C6000 platform. The TMS320C67x DSPs (including the TMS320C6713 device) compose the floating-point DSP generation in the TMS320C6000 DSP platform [1]. The C6713 device is based on the high-performance, advanced very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making this DSP an excellent choice for multichannel and multifunction applications. Operating at 225 MHz, the C6713 delivers up to 1350 million floating-point operations per second (MFLOPS), 1800 million instructions per second (MIPS), and with dual fixed-/floating-point multipliers up to 450 million multiply-accumulate operations per second (MMACS). Operating at 300 MHz, the C6713 delivers up to 1800 million floating-point operations per second (MFLOPS), 2400 million instructions per second (MIPS), and with dual fixed-/floating-point multipliers up to 600 million multiply-accumulate operations per second (MMACS).

The C6713 uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 4K-byte direct-mapped cache and the

Level 1 data cache (L1D) is a 4K-byte 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 256K-byte memory space that is shared between program and data space. 64K bytes of the 256K bytes in L2 memory can be configured as mapped memory, cache, or combinations of the two. The remaining 192K bytes in L2 serve as mapped SRAM.

The C6713 has a rich peripheral set that includes two Multichannel Audio Serial Ports (McASPs), two Multichannel Buffered Serial Ports (McBSPs), two Inter-Integrated Circuit (I^2C) buses, one dedicated General-Purpose Input/Output (GPIO) module, two general-purpose timers, a host-port interface (HPI), and a glue less external memory interface (EMIF) capable of interfacing to SDRAM, SBSRAM, and asynchronous peripherals. The two McASP interface modules each support one transmit and one receive clock zone. Each of the McASP has eight serial data pins, which can be individually allocated, to any of the two zones. The serial port supports time-division multiplexing on each pin from 2 to 32 time slots. The C6713B has sufficient bandwidth to support all 16 serial data pins transmitting a 192 kHz stereo signal. Serial data in each zone may be transmitted and received on multiple serial data pins simultaneously and formatted in a multitude of variations on the Philips Inter-IC Sound (I^2S) format. In addition, the McASP transmitter may be programmed to output multiple S/PDIF, IEC60958, AES-3, CP-430 encoded data channels simultaneously, with a single RAM containing the full implementation of user data and channel status fields. The McASP also provides extensive error checking and recovery features, such as the bad clock detection circuit for each high-frequency master clock, which verifies that the master clock is within a programmed frequency range. The two I^2C ports on the TMS320C6713 allow the DSP to easily control peripheral devices and communicate with a host processor. In addition, the standard multichannel-buffered serial port (McBSP) may be used to communicate with serial peripheral interface (SPI) mode peripheral devices.

The TMS320C6713 device has two boot modes: from the HPI or from external asynchronous ROM. The TMS320C67x DSP generation is supported by the TI eXpressDSP - set of industry benchmark development tools, including a highly optimizing C/C++ Compiler, the Code Composer Studio - Integrated Development Environment (IDE), JTAG-based emulation and real-time debugging, and the DSP/BIOS kernel.

7.1.1 DSP 6713 Features

- Highest-Performance Floating-Point Digital Signal Processor (DSP):

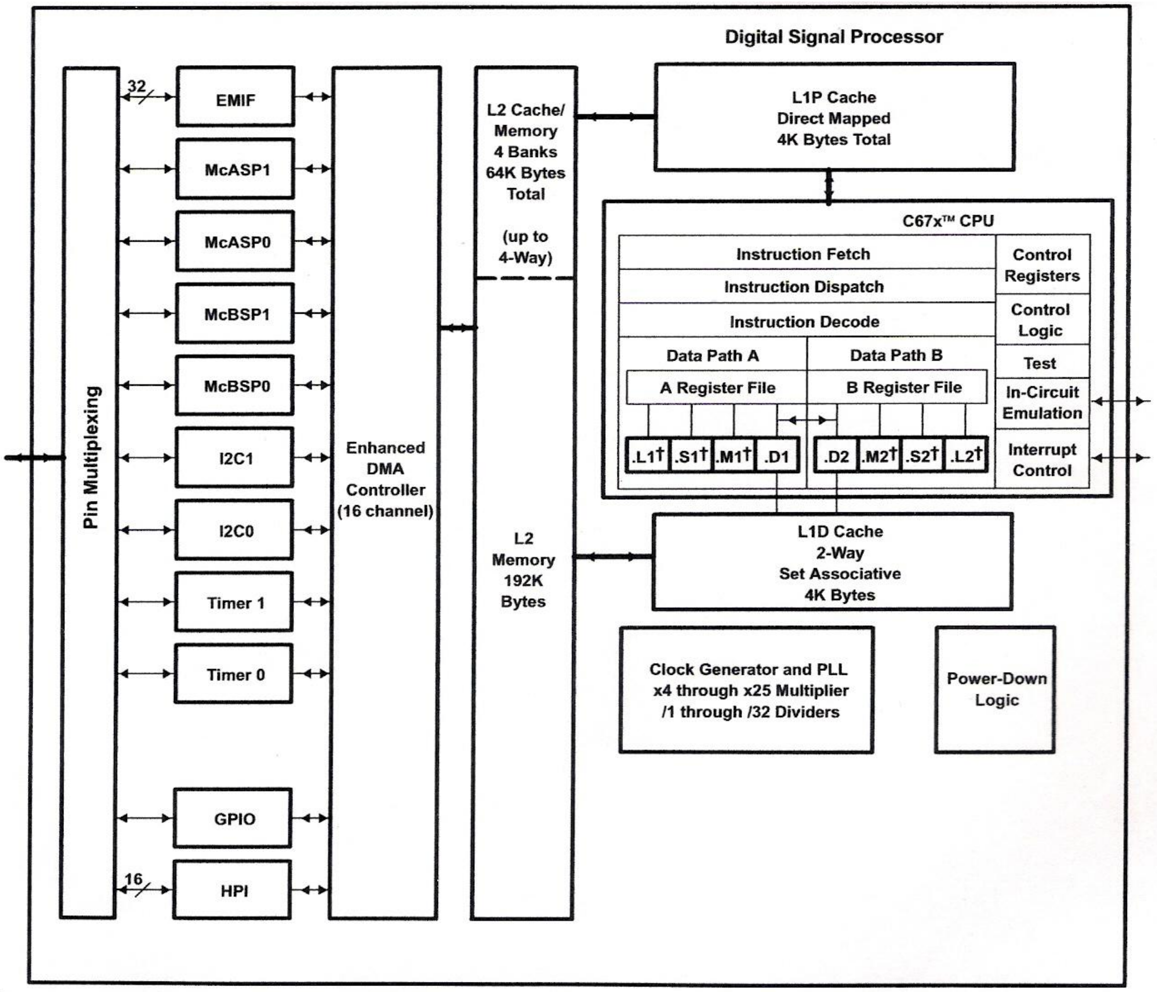
- Eight 32-Bit Instructions/Cycle
- 32/64-Bit Data Word
- 300-, 225-, 200-MHz (GDP and ZDP), and 225-, 200-, 167-MHz (PYP) Clock Rates
- 3.3-, 4.4-, 5-, 6-Instruction Cycle Times
- 2400/1800, 1800/1350, 1600/1200, and 1336/1000 MIPS/MFLOPS
- Rich Peripheral Set, Optimized for Audio
- Highly Optimized C/C++ Compiler

- Advanced Very Long Instruction Word (VLIW) TMS320C67x DSP Core
    - Eight Independent Functional Units:
        - 2 ALUs (Fixed-Point)
        - 4 ALUs (Floating-/Fixed-Point)
        - 2 Multipliers (Floating-/Fixed-Point)
    - Load-Store Architecture with 32 32-Bit General-Purpose Registers
    - Instruction Packing Reduces Code Size
    - All Instructions Conditional
- Instruction Set Features
    - Native Instructions for IEEE 754: Single and Double precision
    - Byte-Addressable (8-, 16-, 32-Bit Data)
    - 8-Bit Overflow Protection
    - Saturation; Bit-Field Extract, Set, Clear; Bit-Counting; Normalization
- L1/L2 Memory Architecture
    - 4K-Byte L1P Program Cache (Direct-Mapped)
    - 4K-Byte L1D Data Cache (2-Way)
    - 256K-Byte L2 Memory Total: 64K-Byte L2 Unified Cache/Mapped RAM, and 192K-Byte Additional L2 Mapped RAM
- Device Configuration
    - Boot Mode: HPI, 8-, 16-, 32-Bit ROM Boot
    - Endianness: Little Endian, Big Endian
- 32-Bit External Memory Interface (EMIF)
    - Glue less Interface to SRAM, EPROM, Flash, SBSRAM, and SDRAM

    - 512M-Byte Total Addressable External Memory Space
- Enhanced Direct-Memory-Access (EDMA) Controller (16 Independent Channels)
- 16-Bit Host-Port Interface (HPI)
- Two McASPs
    - Two Independent Clock Zones Each (1 TX and 1 RX)
    - Eight Serial Data Pins per Port: Individually Assignable to any of the Clock Zones
    - Each Clock Zone Includes:
        - Programmable Clock Generator
        - Programmable Frame Sync Generator
        - TDM Streams From 2-32 Time Slots
        - Support for Slot Size: 8, 12, 16, 20, 24, 28, 32 Bits
        - Data Formatter for Bit Manipulation
    - Wide Variety of I2S and Similar Bit Stream Formats
    - Integrated Digital Audio Interface Transmitter (DIT) Supports:
        - S/PDIF, IEC60958-1, AES-3, CP-430 Formats
        - Up to 16 transmit pins
        - Enhanced Channel Status/User Data
    - Extensive Error Checking and Recovery
- Two Inter-Integrated Circuit Bus (I^2C Bus) Multi-Master and Slave Interfaces
- Two Multichannel Buffered Serial Ports:
    - Serial-Peripheral-Interface (SPI)
    - High-Speed TDM Interface
    - AC97 Interface
- Two 32-Bit General-Purpose Timers
- Dedicated GPIO Module with 16 pins (External Interrupt Capable)
- Flexible Phase-Locked-Loop (PLL) Based Clock Generator Module
- IEEE-1149.1 (JTAG) Boundary-Scan-Compatible
- 208-Pin Power PAD PQFP (PYP)
- 272-BGA Packages (GDP and ZDP)
- 0.13-µm/6-Level Copper Metal Process

- ➢ CMOS Technology
- 3.3-V I/Os, 1.2 -V Internal (GDP/ZDP/ PYP)
- 3.3-V I/Os, 1.4-V Internal (GDP/ZDP) [300 MHz]

The functional block diagram and CPU core diagram is shown in figure 7.1.



† In addition to fixed-point instructions, these functional units execute floating-point instructions.

EMIF interfaces to:
−SDRAM
−SBSRAM
−SRAM,
−ROM/Flash, and
−I/O devices

McBSPs interface to:
−SPI Control Port
−High-Speed TDM Codecs
−AC97 Codecs
−Serial EEPROM

McASPs interface to:
−I2S Multichannel ADC, DAC, Codec, DIR
−DIT: Multiple Outputs

Fig. 7.1 Functional block and CPU (DSP core) diagram of C6713

7.2 DSK 6713

The DSK6713 is a low cost standalone development platform that enables users to evaluate and develop applications for the TI 67XX DSP family. The block diagram describing the board is shown in figure 7.2. Key features include:

- A TI TMS320C6713 DSP operating at 225 MHz.
- An AIC 23 stereo codec.
- 4 user LEDs and 4 DIP switches.
- 16 MB SDRAM and 512 KB non-volatile Flash memory.
- Software board configuration through registers implemented in CPLD.
- JTAG (Joint Test Action Group) emulation through on-board JTAG emulator with USB host interface or external emulator.
- Single voltage power supply (+5V).

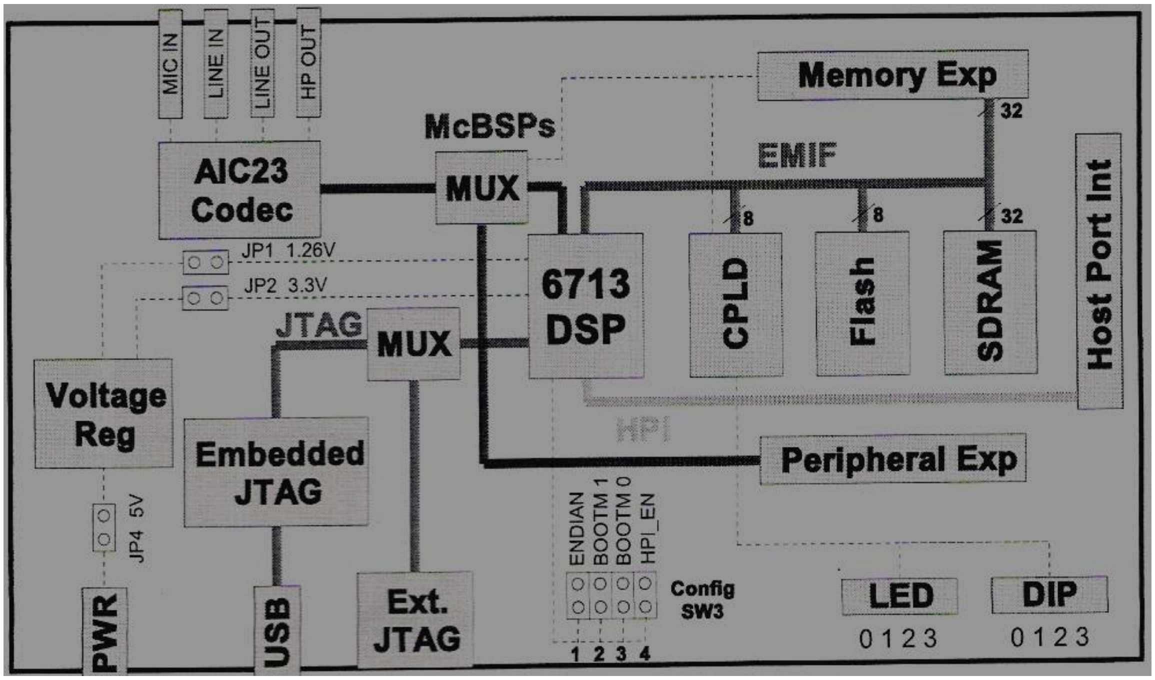


Fig. 7.2 DSK 6713 block diagram

7.2.1 Functional Overview of DSK 6713

The DSP on the 6713 DSK interfaces to on-board peripherals through a 32-bit wide EMIF (External Memory Interface). The SDRAM, Flash and CPLD are all connected to the bus. All addresses are 32 bits wide. Portions of the internal memory can be reconfigured in software as L2 cache rather than fixed RAM. The DSP interfaces to analog audio signals through on-board TLV320AIC23 codec and 3.5mm audio jacks (microphone input, line input, line output and headphone output). The codec can select the microphone input (monaural input) or the line input (stereo input) as active input. The analog output is driven to both the line out (fixed gain) and headphone/speaker out (adjustable gain) connectors. The codec communicates using two serial channels, one to control the codec's internal configuration registers and one to send and receive digital audio samples. McBSP0 is used to send commands to the codec control interface while McBSP1 is used for bi-directional digital audio data. The codec has a 12 MHz system clock. The internal sample rate generate subdivides the 12 MHz clock to generate common frequencies such as 48 KHz, 44.1 KHz and 8 KHz. The sample rate is set by the codec's SAMPLERATE register. Figure 7.3 shows the codec interface on the C6713 DSK.

A programmable logic device called a CPLD is used to implement glue logic that ties the board components together. The CPLD has a register based user interface that lets the user configure the board by reading and writing to its registers. The DSK includes 4 LEDs (D7-D10) and 4 DIP switches (SW1) as a simple way to provide the user input/output. Both are accessed by reading and writing to the CPLD registers.

The PC's USB port cannot be directly connected to DSP C6713. An XDS (eXtended Development System) JTAG emulator is connected to the PC's USB port and DSP is communicated through the JTAG emulator on the DSK. CCS uses USB port to control DSP via JTAG port.

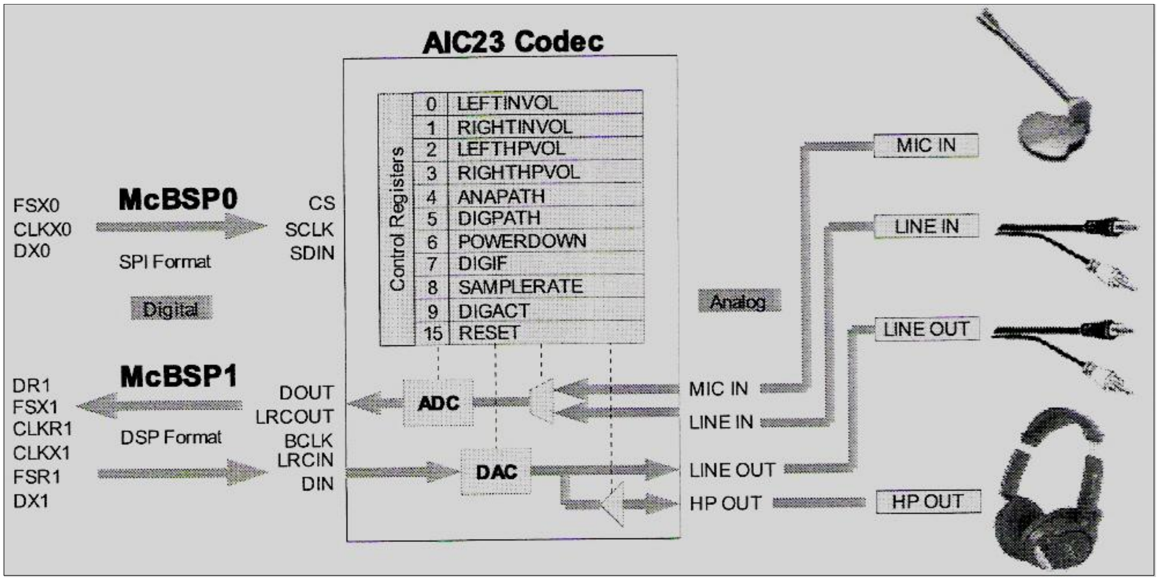


Fig. 7.3 AIC- DSP interface

7.3 Code Composer Studio Integrated Development Environment (CCS IDE)

The Code Composer Studio (CCS) application provides an integrated environment with the following capabilities [2]:

- Integrated development environment (IDE) with an editor, debugger, project manager, profiler, etc.

- ‘C/C++’ compiler, assembly optimizer and linker (code generation tools).
- Simulator.
- Real-time operating system (DSP/BIOS).
- Real-Time Data Exchange (RTDX) between the Host and Target.
- Real-time analysis and data visualization.

The CCS Project Manager organizes files into folders for source files; include files, libraries and DSP/BIOS configuration files. Once the files are added to the project any changes in any of source files will be reflected automatically in the project files. This allows multi user system development. CCS also provides the ability to debug mixed, multi-processor designs simultaneously. It also includes new emulation capabilities with Real Time Data Exchange (RTDX), plus advanced DSP code profiling capabilities. An improved Watch Window monitors the values of local and global variables and C/C++ expressions. Users can quickly view and track variables on the target hardware. It has ability to share C and C++ source and libraries in a multi-user project. The CCS IDE V3.3 is used for implementation here.

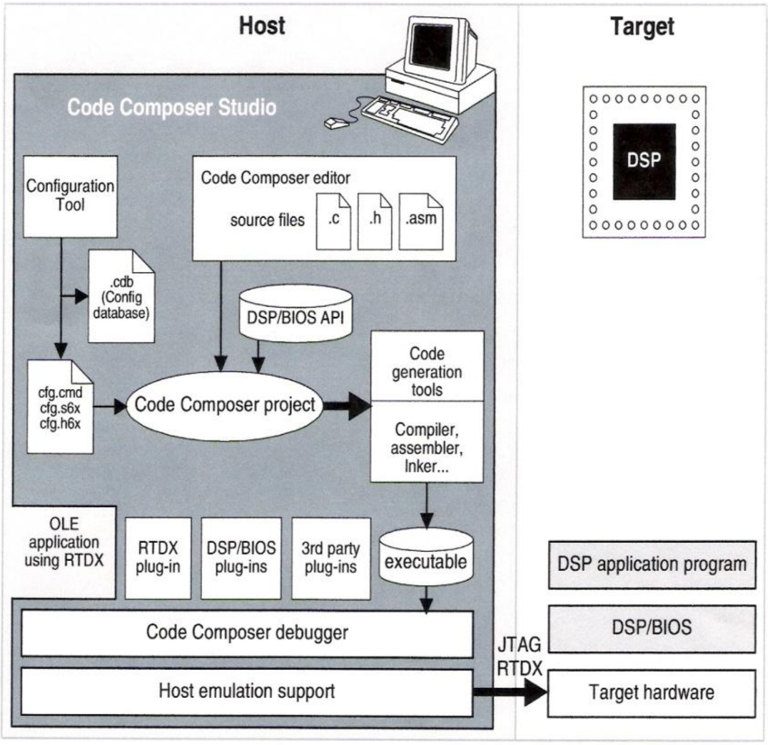


Fig. 7.4 Working of code composer studio

7.4 MATLAB/SIMULINK in Real Time Applications

Rapid prototyping is a new approach in digital signal processing systems development. With the advent of MATLAB's Real Time Workshop (RTW) toolbox it is now possible to compile, load, and execute graphically designed SIMULINK models on an actual DSP platform, without spending many workdays coding in typical DSP-oriented languages (assembly languages), or C/C++ compilers. RTW supports the powerful Texas Instruments 'C6000 series, including the TMS320C6713 DSP. The basic steps of the complete project development include designing an algorithm for the given task, implementing a suitable algorithm in MATLAB and SIMULINK and finally, translating it into target DSP code by means of a rapid prototyping approach. The original code was developed in MATLAB and so the MATLAB's Real Time Workshop (RTW) platform is used for rapid prototyping. Real-Time Workshop builds applications from SIMULINK diagrams for prototyping, testing, and deploying real-time systems on a variety of target computing platforms, including Texas Instruments C6000 class DSP processors (Target Support Package TC6).

7.4.1 Real Time Workshop Toolbox

Real Time Workshop is an extension of capabilities of SIMULINK and MATLAB that automatically generates packages and compiles source code from SIMULINK models to create real-time software applications on a variety of systems [3]. By providing a code generation environment for rapid prototyping and deployment, Real-Time Workshop is the foundation for production code generation capabilities. Along with other tools and components from MATLAB, Real-Time Workshop provides automatic code generation tailored for a variety of target platforms, a rapid and direct path from system design to implementation, seamless integration with MATLAB and SIMULINK, a simple graphical user interface, an open architecture and extensible make process. The principal components and features of Real-Time Workshop [4] are:

- SIMULINK Code Generator: - Automatically generates C code from the SIMULINK model.
- Make Process: - The Real-Time Workshop user-extensible make process lets us customize compilation and linking of generated code for our own production or rapid prototyping target.

- SIMULINK External Mode: - External mode enables communication between SIMULINK and a model executing on a real-time test environment, or in another process on the same machine. External mode lets us to perform real-time parameter tuning, data logging, and viewing using SIMULINK as a front end.
- Targeting Support: - Using the targets bundled with Real-Time Workshop, we can build systems for real-time and prototyping environments. The generic real-time and other bundled targets provide a framework for developing customized rapid prototyping or production target environments.
- Rapid Simulations: - Using SIMULINK Accelerator, the S-Function Target, or the Rapid Simulation Target, we can accelerate our simulations by 5 to 20 times on average. Executables built with these targets bypass normal SIMULINK interpretive simulation mode. Code generated by SIMULINK Accelerator, S-Function Target, and Rapid Simulation Target is highly optimized to execute only the algorithms used in our specific model. In addition, the code generator applies many optimizations, such as eliminating ones and zeros in computations for filter blocks.
- Large-Scale Modeling: - Support for multilevel modeling (termed "model referencing"), which lets us to generate code incrementally for a hierarchy of independent component models, as they evolve.

The Target Language Compiler (TLC) tool is an integral part of the Real-Time Workshop. It enables customizing the C code generated from any SIMULINK model and generates optimal, inline code for SIMULINK blocks. Figure 7.5 illustrates how Real-Time Workshop, helps us in real time system development process and figure 7.6 explains its working.

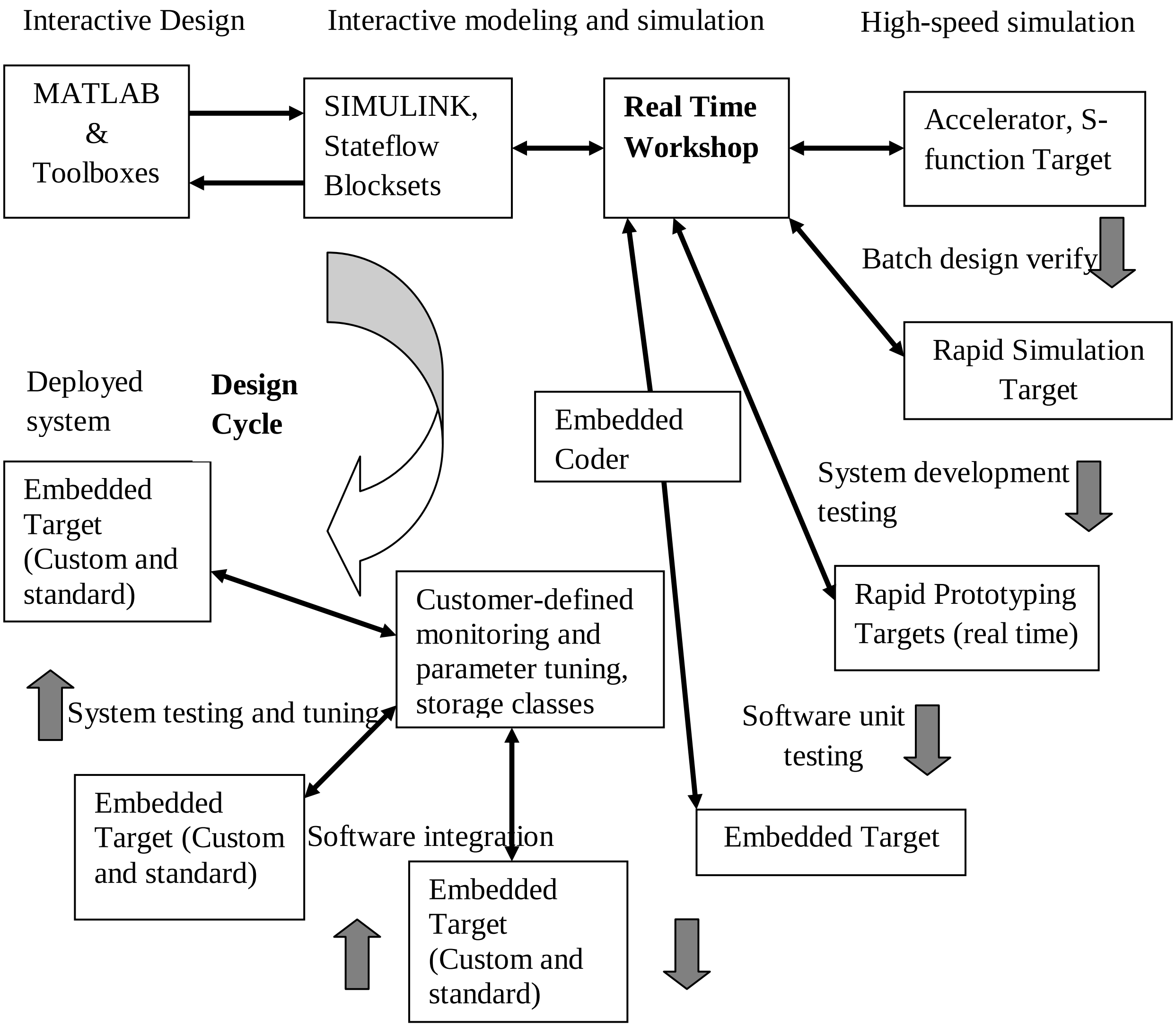

Interactive Design
Interactive modeling and simulation
High-speed simulation
MATLAB & Toolboxes
SIMULINK, Stateflow Blocksets
Real Time Workshop
Accelerator, S-function Target
Batch design verify
Rapid Simulation Target
Deployed system
Design Cycle
Embedded Coder
System development testing
Embedded Target (Custom and standard)
Customer-defined monitoring and parameter tuning, storage classes
Rapid Prototyping Targets (real time)
System testing and tuning
Software unit testing
Embedded Target (Custom and standard)
Software integration
Embedded Target
Embedded Target (Custom and standard)

Fig. 7.5 Role of real time workshop

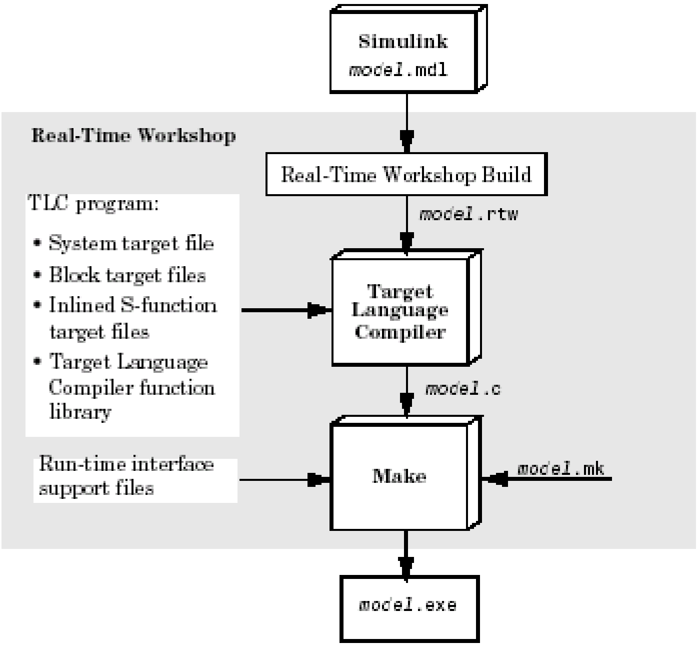


Fig. 7.6 Working of real time workshop

7.4.2 Target Support Package TC6

This platform integrates SIMULINK and MATLAB with Texas Instruments eXpressDSP tools. The software collection allows developing and validating digital signal processing designs from concept through code. It consists of the TI C6000 target that automates rapid prototyping on C6000 hardware targets [5]. The target uses C code generated by RTW and CCS to build an executable file (.out) for the targeted processor. The RTW build process loads the targeted machine code to target board and runs the executable file on the digital signal processor. All the features provided by CCS, such as tools for editing, building, debugging, code profiling, and project management help in developing the applications using MATLAB, SIMULINK, RTW, and the supported hardware (DSK 6713). Executing code generated from RTW on a particular target in real time requires that RTW generate target code that is tailored to the specific hardware target. Target-specific code includes I/O device drivers and an interrupt service routine (ISR). Since these device drivers and ISRs are specific to particular hardware targets, it must be ensured that the target-specific components are compatible with the target hardware. To build an executable, TC6 uses the MATLAB links to invoke the code building process from within CCS. Once executable file is downloaded to the target and run, the code runs wholly on the target; one can access the running process only from the CCS debugging tools or across a link for CCS [6] or Real Time Data Exchange (RTDX). Otherwise the running process is not accessible.

7.5 Summary

The hardware implementation tools viz. DSK 6713, CCS IDE, SIMULINK, MATLAB RTW and Target Support Package TC6 together can be used to implement any complex speech processing algorithm on TMS320C6713 DSP platform. The ADC and DAC needed for such applications are provided on DSK 6713. Also SIMULINK can be used for real time implementation of speech processing algorithm on PC[1]. The sound card on PC contains necessary ADC, DAC and audio power amplifiers. The hybrid algorithm developed here is tested for real time implementation on PC as well as on DSP. The implementation details are described in the next chapter.

[1] A paper entitled “Simulation and Real Time Implementation of Spectral Subtraction and Wavelet De-Noising Embedded Algorithms for Speech Enhancement” is published in International Journal of Recent Trends in Engineering and Technology, (IJRTET), Vol. 4, No. 4, Nov. 2010, pp. 146-149, ACEEE, USA. ISSN (Online):2158-5563, ISSN (Print): 2158-5555. Archived in SEARCH digital library.

Chapter 8

Real Time and Embedded Implementation of Hybrid Algorithm

Since the complete implementation of the hybrid algorithm proposed here has a great computational complexity, it is necessary to test the possibility of implementing it in a real time and embedded environment. As the developed algorithm is at a primary research level, it is needed to perform tests on a flexible platform that allows the implementation of the non-optimized algorithms with a reasonable effort. The algorithm is first tried for real time implementation on PC using SIMULINK. For dedicated hardware implementation the DSP platform using 32-bit floating point processor TMS320C6713 from Texas Instruments is selected. DSK 6713 from Spectrum Digital Incorporation is used for implementing algorithm on the TMS320C6713 DSP. The Code Composer Studio Integrated Development Environment version 3.3 (CCS IDE V3.3) from Texas Instruments is used as compiler and debugger. This tool is invoked from MATLAB using RTW and Target Support Package TC6 toolboxes. Various profiling results are obtained and compared in this chapter.

8.1 Typical Setup for Developing Models

Figure 8.1 presents a block diagram of the typical setup for developing models, along with the input and output connected to the C6713 DSK [1].

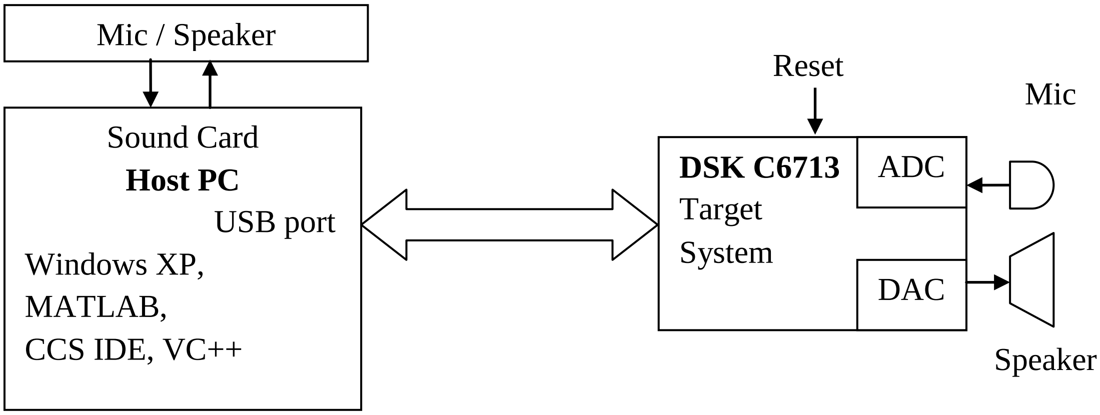


Fig. 8.1 Typical hardware and software setup for developing models

8.2 Real Time Implementation of Hybrid Approach on PC

Figure 8.2 presents a block diagram of real time PC implementation of hybrid algorithm. The data buffering and windowing, hybrid algorithm and overlap-add blocks represent sub-systems used to implement the overall speech enhancement system. In the set up the audio device (microphone) catch up the noisy speech signal from real environment. It digitizes the monophonic speech signal with 8 KHz sampling rate and 16 bits/sample resolution. The buffering and windowing block frames the incoming data into a frame of 256 samples with 50% overlap and windowed using Hamming window. The RASTA algorithm is non-causal and requires future frames for filtering; which throws the challenge for real time implementation. So a sub-frame concept is used to overcome it. Here the framed data is divided into four sub-frames each consists of 64 samples. The matrix concatenate block is used to make a 64 x 4 data block from four 64 x 1 sub-frame. It is shown in figure 8.3. After proper framing the 256 point FFT is taken and from complex spectrum the magnitude is taken for further processing and phase is given for reconstruction with enhanced magnitude. The hybrid algorithm sub-system performs the speech enhancement operation using the combine RASTA and STSA approach described in chapter 6. Figure 8.4 shows the internal blocks of the sub-system. The entire hybrid algorithm is incorporated as an embedded MATLAB function. Finally from noisy phase and enhanced magnitude the enhanced complex spectrum is obtained for a frame. After 256 point IFFT operation, the overlap-add synthesis is performed to reconstruct the signal in time domain. Figure 8.5 shows the internal blocks to obtain overlap-add synthesis. The enhanced speech can be heard on speaker or headphone. The amplitude of output speech can be controlled by setting the gain value in the block before the wave device block (speaker/headphone).

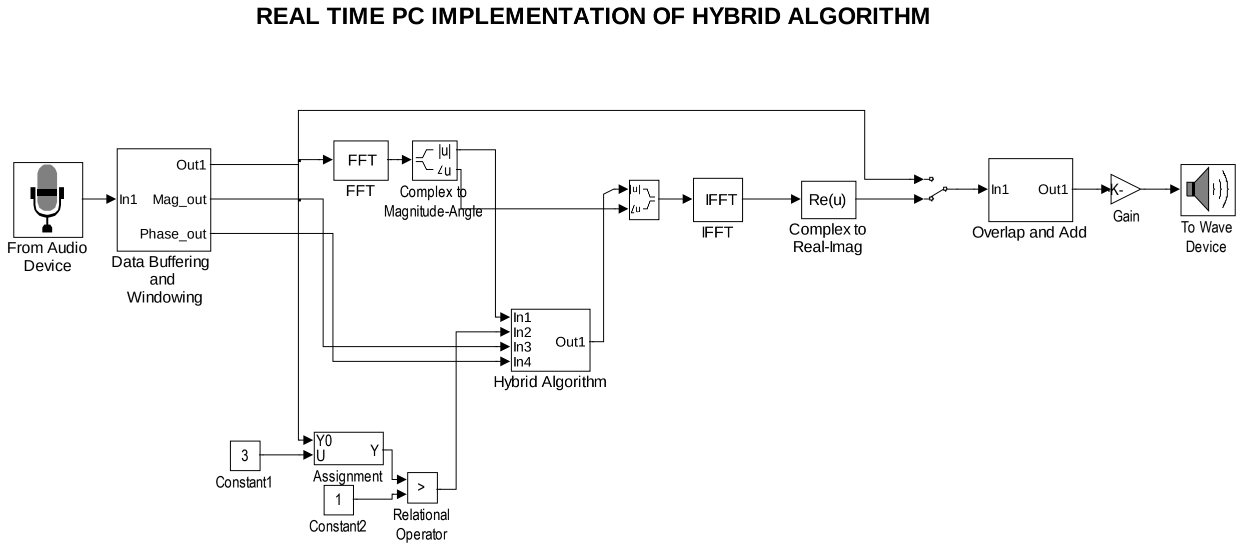


Fig. 8.2 SIMULINK block for real time implementation of hybrid algorithm on PC

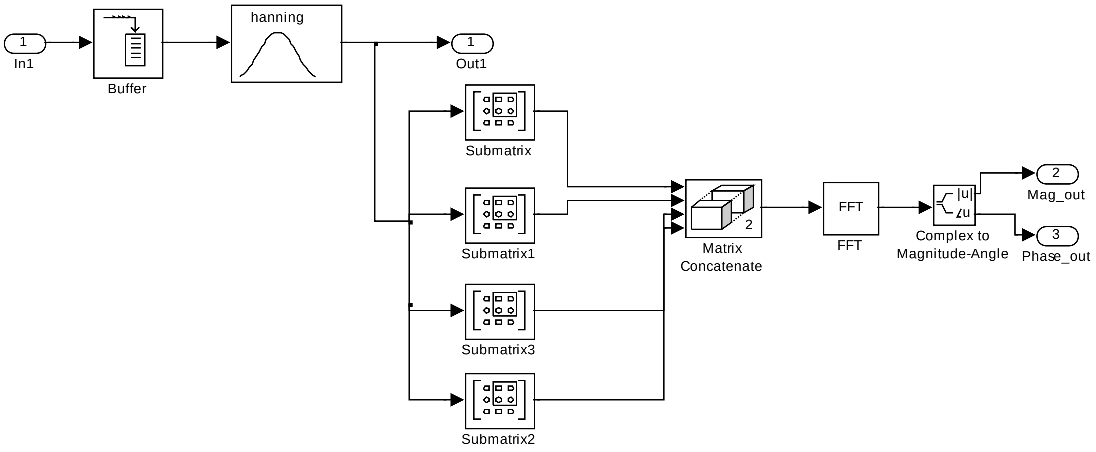


Fig. 8.3 Internals of sub-system data buffering and windowing

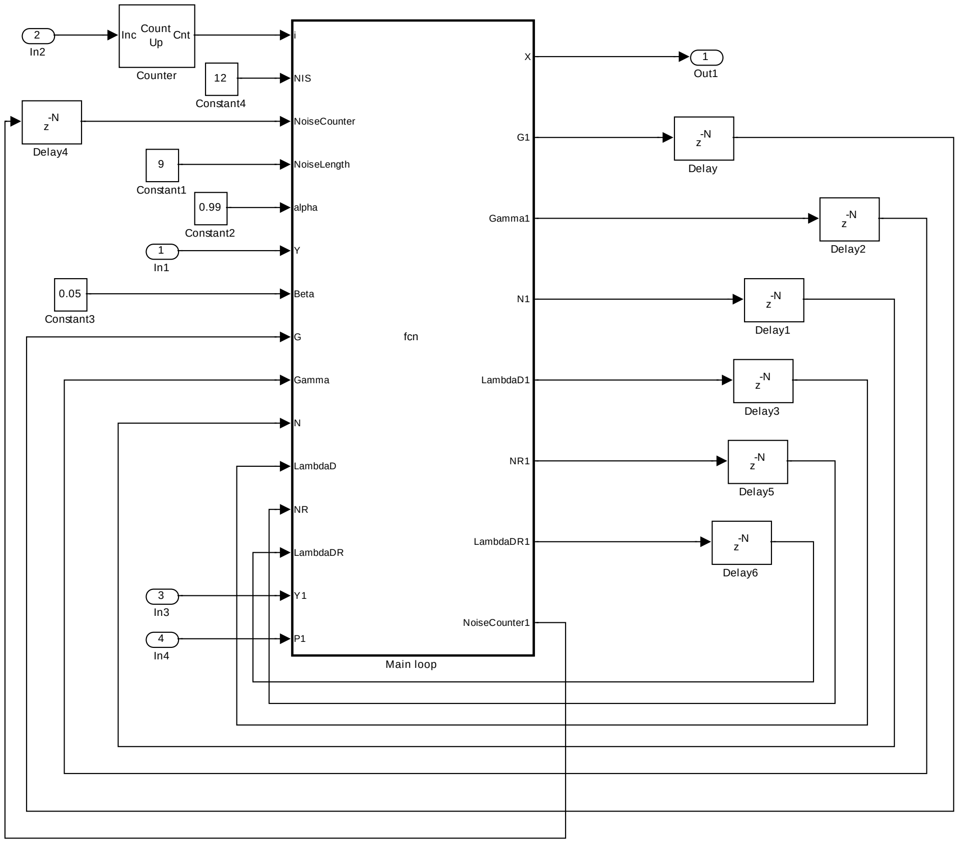


Fig. 8.4 Internals of sub-system hybrid algorithm

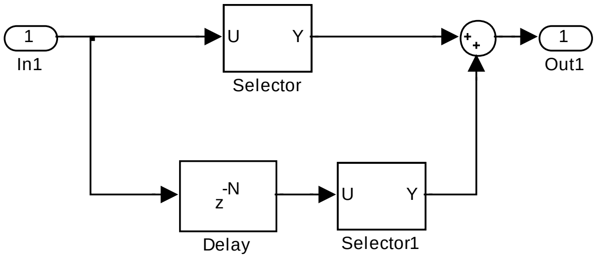
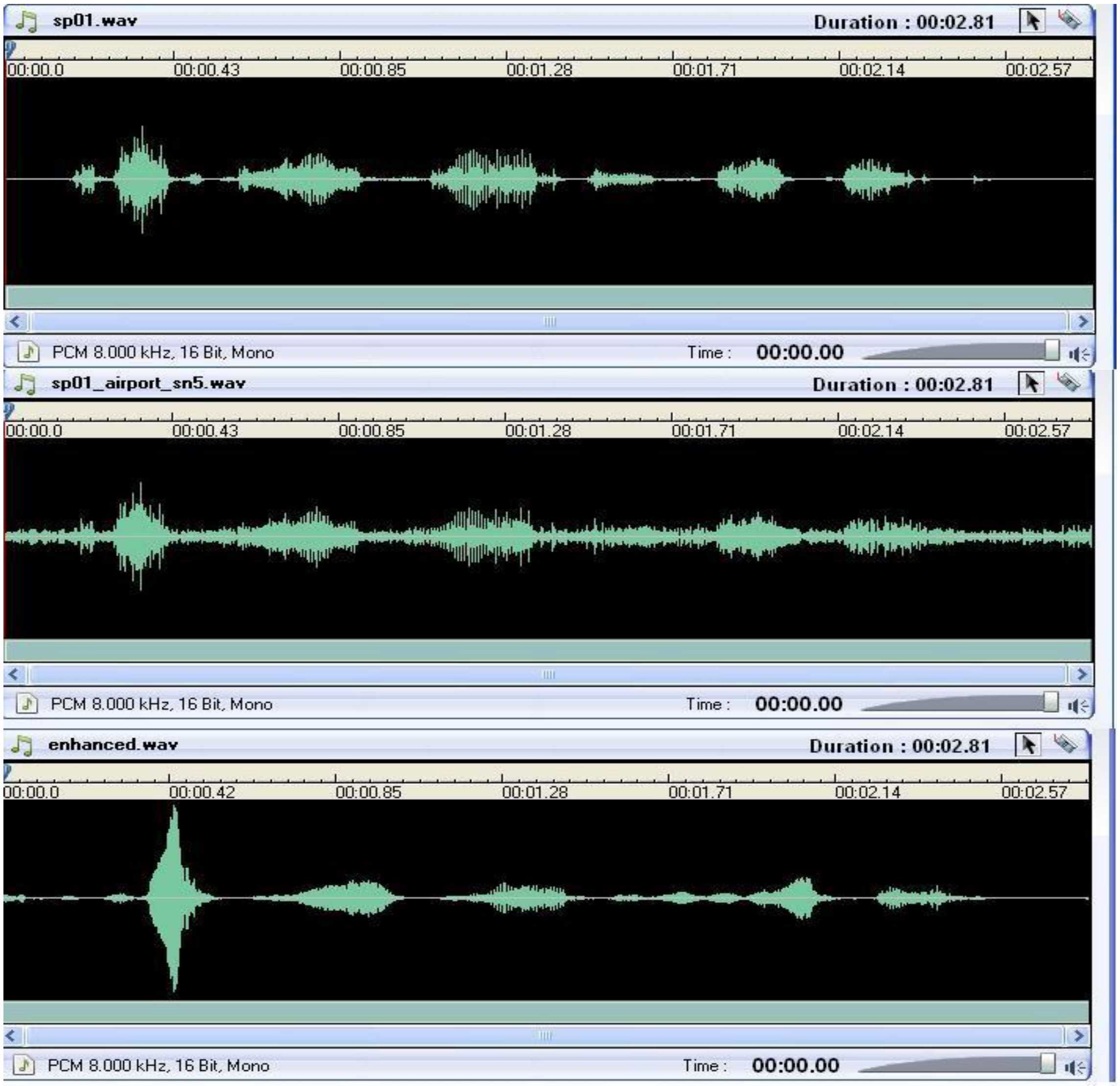


Fig. 8.5 Internals of sub-system overlap-add synthesis

Figure 8.6 shows the time domain waveform of clean speech, noisy speech corrupted by airport noise of 5dB and enhanced speech using the real time hybrid algorithm. It is self explanatory from this figure that the background noise is completely eliminated from the speech.


sp01.wav
Duration : 00:02.81

PCM 8.000 kHz, 16 Bit, Mono
Time : 00:00.00

sp01_airport_sn5.wav
Duration : 00:02.81

PCM 8.000 kHz, 16 Bit, Mono
Time : 00:00.00

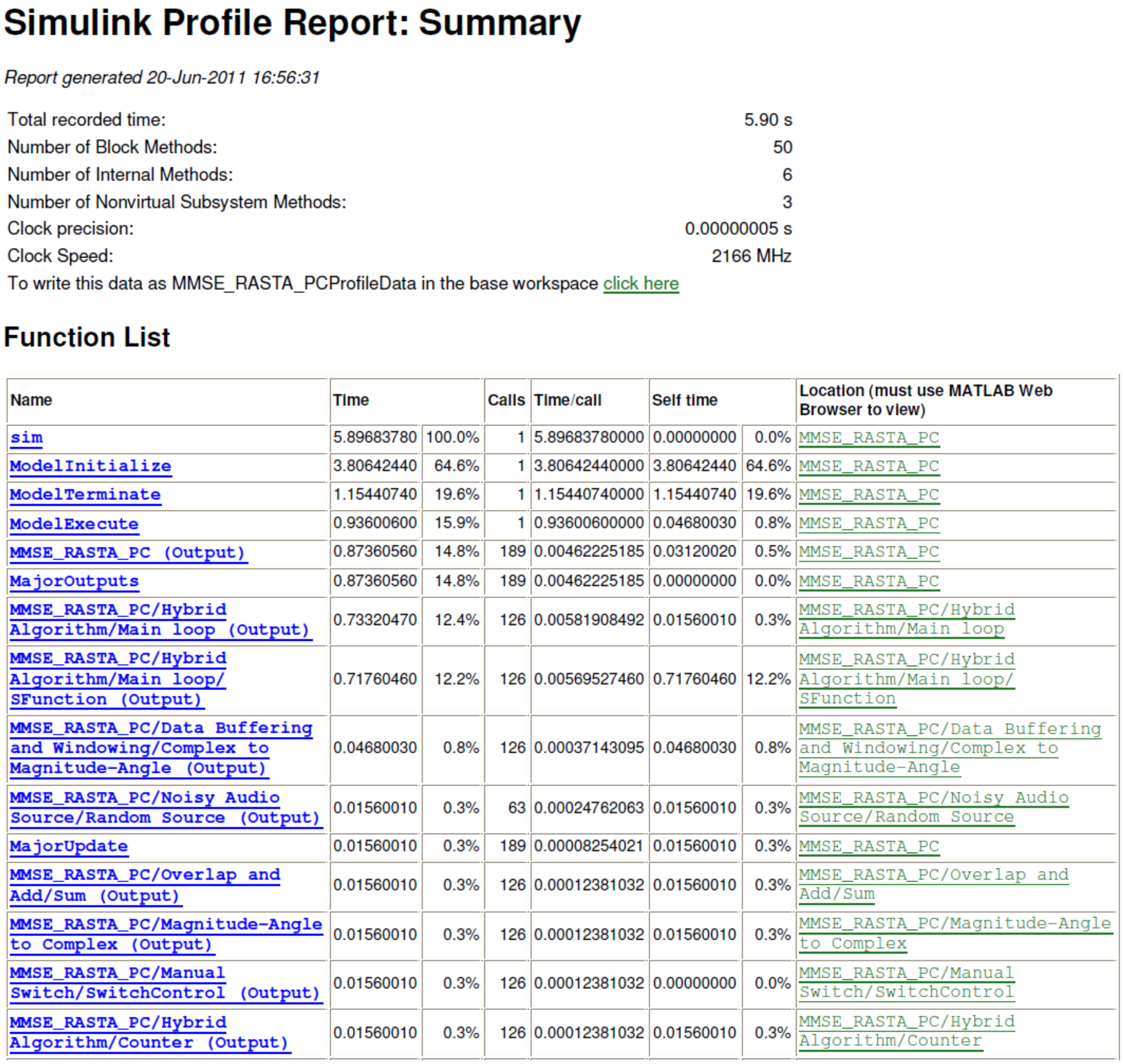
enhanced.wav
Duration : 00:02.81

PCM 8.000 kHz, 16 Bit, Mono
Time : 00:00.00

Fig. 8.6 Waveforms of clean, noisy and enhanced speech using real time hybrid algorithm

8.3 SIMULINK Profile Results

The Profiler allows running a program and then looking at how long each block took to execute. The profiler captures data while the model runs and identifies the parts of model requiring the most time to simulate. With this information one can concentrate on optimizing the sections of code that take up the most time. Figure 8.7 shows the SIMULINK profile results for the hybrid algorithm.



Simulink Profile Report: Summary

Report generated 20-Jun-2011 16:56:31

Total recorded time:	5.90 s
Number of Block Methods:	50
Number of Internal Methods:	6
Number of Nonvirtual Subsystem Methods:	3
Clock precision:	0.00000005 s
Clock Speed:	2166 MHz

To write this data as MMSE_RASTA_PCProfileData in the base workspace click here

Function List

Name	Time		Calls	Time/call	Self time		Location (must use MATLAB Web Browser to view)
sim	5.89683780	100.0%	1	5.89683780000	0.00000000	0.0%	MMSE_RASTA_PC
ModelInitialize	3.80642440	64.6%	1	3.80642440000	3.80642440	64.6%	MMSE_RASTA_PC
ModelTerminate	1.15440740	19.6%	1	1.15440740000	1.15440740	19.6%	MMSE_RASTA_PC
ModelExecute	0.93600600	15.9%	1	0.93600600000	0.04680030	0.8%	MMSE_RASTA_PC
MMSE_RASTA_PC (Output)	0.87360560	14.8%	189	0.00462225185	0.03120020	0.5%	MMSE_RASTA_PC
MajorOutputs	0.87360560	14.8%	189	0.00462225185	0.00000000	0.0%	MMSE_RASTA_PC
MMSE_RASTA_PC/Hybrid Algorithm/Main loop (Output)	0.73320470	12.4%	126	0.00581908492	0.01560010	0.3%	MMSE_RASTA_PC/Hybrid Algorithm/Main loop
MMSE_RASTA_PC/Hybrid Algorithm/Main loop/ SFunction (Output)	0.71760460	12.2%	126	0.00569527460	0.71760460	12.2%	MMSE_RASTA_PC/Hybrid Algorithm/Main loop/ SFunction
MMSE_RASTA_PC/Data Buffering and Windowing/Complex to Magnitude-Angle (Output)	0.04680030	0.8%	126	0.00037143095	0.04680030	0.8%	MMSE_RASTA_PC/Data Buffering and Windowing/Complex to Magnitude-Angle
MMSE_RASTA_PC/Noisy Audio Source/Random Source (Output)	0.01560010	0.3%	63	0.00024762063	0.01560010	0.3%	MMSE_RASTA_PC/Noisy Audio Source/Random Source
MajorUpdate	0.01560010	0.3%	189	0.00008254021	0.01560010	0.3%	MMSE_RASTA_PC
MMSE_RASTA_PC/Overlap and Add/Sum (Output)	0.01560010	0.3%	126	0.00012381032	0.01560010	0.3%	MMSE_RASTA_PC/Overlap and Add/Sum
MMSE_RASTA_PC/Magnitude-Angle to Complex (Output)	0.01560010	0.3%	126	0.00012381032	0.01560010	0.3%	MMSE_RASTA_PC/Magnitude-Angle to Complex
MMSE_RASTA_PC/Manual Switch/SwitchControl (Output)	0.01560010	0.3%	126	0.00012381032	0.00000000	0.0%	MMSE_RASTA_PC/Manual Switch/SwitchControl
MMSE_RASTA_PC/Hybrid Algorithm/Counter (Output)	0.01560010	0.3%	126	0.00012381032	0.01560010	0.3%	MMSE_RASTA_PC/Hybrid Algorithm/Counter

Fig. 8.7 SIMULINK profile results of hybrid algorithm

The model initialize, terminate and execute times are not a major concerned for real time implementations. The Main loop of the hybrid algorithm occupies the majority of execution time as expected. Form figure 8.7 it is 12.4%. The other blocks require comparatively less time. Hence the main loop function of the hybrid algorithm is the major concern for embedded implementations. It can be concluded here that with the given complexity of the hybrid algorithm it is suitable for real time implementation on PC. It is interesting to see the same profiling results when the algorithm is downloaded on dedicated hardware.

8.4 Real Time Implementation of Hybrid Approach on DSK6713

Figure 8.8 presents a diagram for real time implementation of hybrid algorithm on DSK6713. It differs from the previous block only by I/O which is C6713 DSK ADC and DAC here. The link and procedure for downloading this model on the kit has been already described in chapter 7.

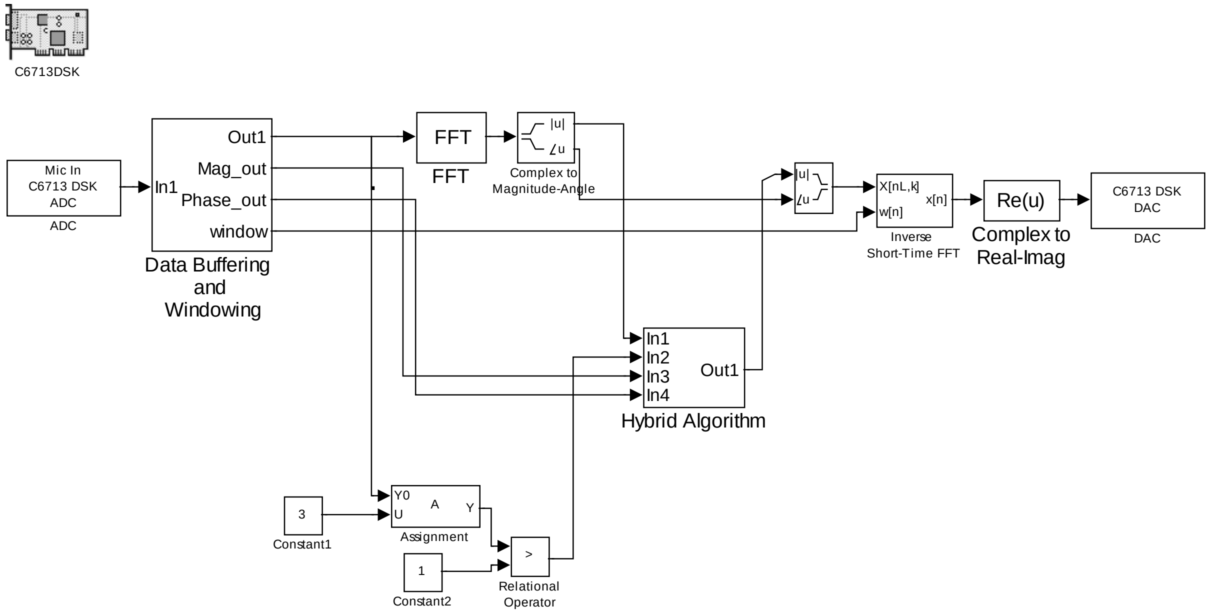


Fig. 8.8 SIMULINK block for real time implementation of hybrid algorithm on DSK6713

8.5 Profiling Results for DSK 6713 Implementation

Target Support Package TC6 software [2] supports DSP/BIOS features as options when code is generated for target and some ways it is possible to use the real-time operating system (RTOS) features of DSP/BIOS in the application. As a part of the Texas Instruments eXpressDSP™ technology, TI designed DSP/BIOS to include three components:

- DSP/BIOS Real-Time Analysis Tools — these tools and windows within Code Composer Studio IDE are used to view program as it executes on the target in real-time.
- DSP/BIOS Configuration Tool — enables to add and configure any and all DSP/BIOS objects that used to instrument the application. This tool is used to configure interrupt schedules and handlers, set thread priorities, and configure the memory layout on DSP.
- DSP/BIOS Application Program Interface (API) — lets to use C or assembly language functions to access and configure DSP/BIOS functions by calling any of over 150 API functions. Target Support Package TC6 software uses the API to access DSP/BIOS.

These components can be linked into application, directly or indirectly referencing only functions that need for the application to run efficiently and optimally. Only functions that specifically reference become part of the code base. Others are not included to avoid adding unused code to the project. In addition, after adding one or more functions from DSP/BIOS, the configuration tool helps to disable feature that do not need later, letting to optimize the program for speed and size.

While generating code that includes the DSP/BIOS options DSP/BIOS objects become part of the generated code. With these in place the profiling option in Target Support Package TC6 software can be used to check the performance of application running on target, gauge performance and find bottlenecks. To generate code that includes DSP/BIOS options, the Target Preferences block must select DSP/BIOS from the Operating system list on the Board Info pane. By selecting profile real-time task execution in the RTW software options, it inserts statistics (STS) object instrumentation at the beginning and end of the code for each atomic subsystem in the model. After the code has been running for a few seconds on target, the profiling results from target can be retrieved and it displays the information in a custom HTML report. Code profiling works only on atomic subsystems in the model. By designating subsystems of the model as atomic, each subsystem is forced to execute only when all of its inputs are available. Waiting for all the subsystem inputs to be available before running the subsystem allows the subsystem code to be profiled as a contiguous segment. Nested subsystems are profiled as part of their parent systems—the execution time reported for the parent subsystem includes the time spent in any profiled child subsystems. When the model is configured to use single-tasking mode, all atomic subsystems in the model are profiled and appear in the report. However, all systems and subsystems do run once before the program terminates. This allows obtaining profiling results for all systems. The following tasks compose the process of profiling the code generated.

1. Enable DSP/BIOS for the code.
2. Enable profiling in the Real-Time Workshop software.
3. Create atomic subsystems to profile in the model.
4. Build, download, and run the model.
5. Use profile to view the MATLAB profile report.

The report shows the amount of time spent computing each subsystem, including outputs and updates of code segments, and provides links that open the corresponding subsystem in the SIMULINK model. Following are the definitions of report entries.

- **System name**

  Provides the name of the profiled model.
- **Number of iterations counted**

  The number of interrupts that occurred between the start of model execution and the moment the statistics was obtained.
- **CPU clock speed**

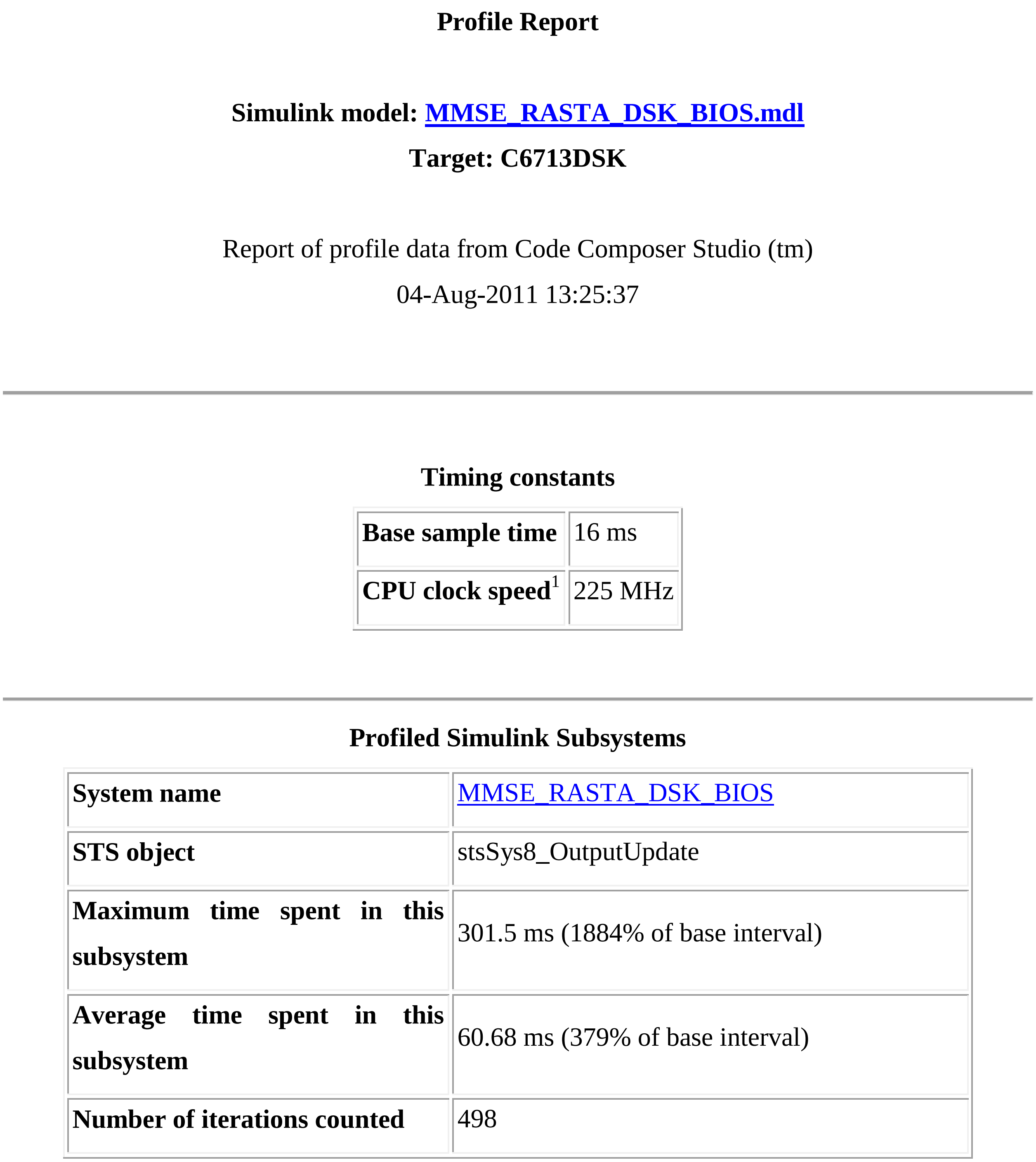
  The instruction cycle speed of the digital signal processor.
- **Maximum time spent in this subsystem per interrupt**

  The amount of time spent in the code segment corresponding to the indicated subsystem in the worst case. Over all the iterations measured, the maximum time that occurs is reported here. Since the profiler only supports single-tasking solver mode, no calculation can be preempted by a new interrupt. All calculations for all subsystems must complete within one interrupt cycle, even for subsystems that execute less often than the fastest rate.
- **Maximum percent of base interval**

  The worst-case execution time of the indicated subsystem, reported as a percentage of the time between interrupts.
- **STS objects**

  Profiling uses STS objects to measure the execution time of each atomic subsystem. One STS object can be used to profile exactly one segment of code. Depending on how RTW generates code for each subsystem, there may be one or two segments of code for the subsystem; the computation of outputs and the updating of states can be combined or separate.

Using the above mentioned settings the report obtained for DSK6713 implementation of the model is shown in figure 8.9.



Profile Report

Simulink model: MMSE_RASTA_DSK_BIOS.mdl

Target: C6713DSK

Report of profile data from Code Composer Studio (tm)

04-Aug-2011 13:25:37

Timing constants

Base sample time	16 ms
CPU clock speed[1]	225 MHz

Profiled Simulink Subsystems

System name	MMSE_RASTA_DSK_BIOS
STS object	stsSys8_OutputUpdate
Maximum time spent in this subsystem	301.5 ms (1884% of base interval)
Average time spent in this subsystem	60.68 ms (379% of base interval)
Number of iterations counted	498

Fig. 8.9 Profile report of real time implementation of hybrid algorithm on DSK6713

System name	MMSE_RASTA_DSK_BIOS/Hybrid Algorithm
STS object	stsSys5_OutputUpdate
Maximum time spent in this subsystem	286.7 ms (1791% of base interval)
Average time spent in this subsystem	45.95 ms (287% of base interval)
Number of iterations counted	498

System name	MMSE_RASTA_DSK_BIOS/Hybrid Algorithm/Main loop
STS object	stsSys4_OutputUpdate
Maximum time spent in this subsystem	286.2 ms (1788% of base interval)
Average time spent in this subsystem	45.48 ms (284% of base interval)
Number of iterations counted	498

System name	MMSE_RASTA_DSK_BIOS/ADC
STS object	stsSys0_OutputUpdate
Maximum time spent in this subsystem	15.9 ms (99% of base interval)
Average time spent in this subsystem	124.4 µs (0.777% of base interval)
Number of iterations counted	250

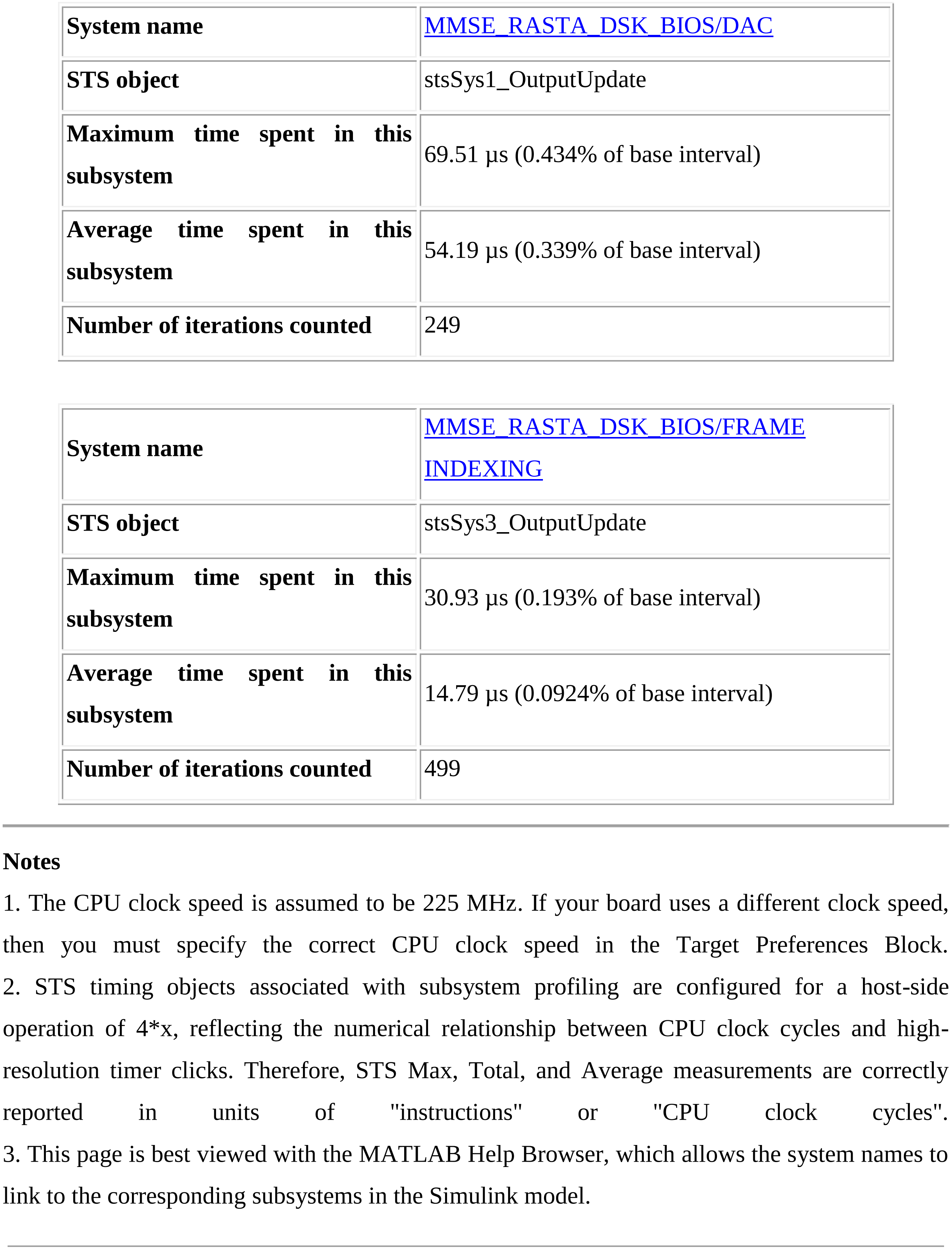
Fig. 8.9 Profile report of real time implementation of hybrid algorithm on DSK6713 (cont.)

System name	MMSE_RASTA_DSK_BIOS/Data Buffering and Windowing
STS objects	stsSys2_Output, stsSys2_Update
Maximum time spent in this subsystem	11.14 ms (69% of base interval)
Average time spent in this subsystem	10.83 ms (67% of base interval)
Number of iterations counted	499

System name	MMSE_RASTA_DSK_BIOS/STFT
STS object	stsSys7_OutputUpdate
Maximum time spent in this subsystem	2.774 ms (17.3% of base interval)
Average time spent in this subsystem	2.663 ms (16.6% of base interval)
Number of iterations counted	499

System name	MMSE_RASTA_DSK_BIOS/ISTFT
STS objects	stsSys6_Output, stsSys6_Update
Maximum time spent in this subsystem	1.2 ms (7.5% of base interval)
Average time spent in this subsystem	1.126 ms (7.04% of base interval)
Number of iterations counted	498

Fig. 8.9 Profile report of real time implementation of hybrid algorithm on DSK6713 (cont.)



System name	MMSE_RASTA_DSK_BIOS/DAC
STS object	stsSys1_OutputUpdate
Maximum time spent in this subsystem	69.51 µs (0.434% of base interval)
Average time spent in this subsystem	54.19 µs (0.339% of base interval)
Number of iterations counted	249

System name	MMSE_RASTA_DSK_BIOS/FRAME INDEXING
STS object	stsSys3_OutputUpdate
Maximum time spent in this subsystem	30.93 µs (0.193% of base interval)
Average time spent in this subsystem	14.79 µs (0.0924% of base interval)
Number of iterations counted	499

Notes

1. The CPU clock speed is assumed to be 225 MHz. If your board uses a different clock speed, then you must specify the correct CPU clock speed in the Target Preferences Block.
2. STS timing objects associated with subsystem profiling are configured for a host-side operation of 4*x, reflecting the numerical relationship between CPU clock cycles and high-resolution timer clicks. Therefore, STS Max, Total, and Average measurements are correctly reported in units of "instructions" or "CPU clock cycles".
3. This page is best viewed with the MATLAB Help Browser, which allows the system names to link to the corresponding subsystems in the Simulink model.

Fig. 8.9 Profile report of real time implementation of hybrid algorithm on DSK6713 (cont.)

Looking at the report the hybrid algorithm block occupies 284% average time of the base sample time. That is the constraint for the DSK6713 implementation of the same model which has no problem at all when runs on PC. The output speech obtained is obviously no longer as per the requirements. The algorithm needs some optimizations before its implementation on DSK6713. The comparison of both these implementations is shown in table 8.1.

Function/Block	**PC Implementation**	**DSP Implementation**
CPU clock speed	2166MHz	225MHz
	Average Execution Time	
Input	0.3%	0.78%
Data buffering/windowing	0.8%	6.7%
Hybrid algorithm (Main loop)	12.4%	284%
Overlap-add	0.3%	7.04%
Output	0.3%	0.34%

Table 8.1 Profile results comparison

8.6 CCS Profiling Results for DSK 6713 Implementation

To create an efficient application, it is needed to focus on performance, power, code size, or cost depending upon goals. Application code analysis is the process of gathering and interpreting data about factors that influence an application's efficiency. CCS IDE provides profile tool to help in analyzing the code [3]. These profiling are incorporated for use with a simulator (C6713 device cycle accurate simulator with little endian is used in the application), and will not function properly with a DSK hardware configuration. This activity measures the total cycles consumed by entire application and calculates the total code size of application. The settings for the same are described in [3]. Using this summary of the profiling of the program loaded in simulator is obtained and described in table 8.2. To optimize performance it is required to decrease stall cycles and increase hit ratio of various cache memories. However it requires a complex tuning process.

Event	Count	Percentage
Total Cycles	423782	
NOP cycles	26392	49.71
Stall Cycles	370688	87.47
L1P Stall Cycles	201339	47.51
L1D Stall Cycles	218253	51.50
Instructions decoded	45214	
Instructions executed	40015	88.50
Instructions conditioned false	5199	11.50
Execute Packets	32699	
Branches taken	6924	
Total Loads	2697	
Total Stores	6643	
Instruction cache references	19370	
Instruction cache hits	15979	82.49
Instruction cache misses	3391	17.51
Data cache references	9340	
Data cache reads	2697	28.88
Data cache writes	6643	71.12
Data cache hits	552	5.91
Data cache read hits	293	10.86
Data cache write hits	259	3.90
Data cache misses	8788	94.09
Data cache read misses	2404	89.14
Data cache write misses	6384	96.10
L2 cache references	14	
L2 cache data reads	0	0.00
L2 cache data writes	0	0.00
L2 cache instruction reads	14	100.00
L2 cache hits	1	7.14
L2 cache data read hits	0	0.00
L2 cache data write hits	0	0.00
L2 cache instruction hits	1	7.14
L2 cache misses	13	92.86
L2 cache data read misses	0	0.00
L2 cache data write misses	0	0.00
L2 cache instruction misses	13	92.86
L2 SRAM references	4497	
L2 SRAM data reads	14	0.31
L2 SRAM data writes	4376	97.31
L2 SRAM instruction reads	107	2.38

Table 8.2 CCS profile summary of hybrid algorithm

8.7 Summary

This chapter has described the real time implementation of hybrid algorithm on PC as well as DSK6713 through SIMULINK. The profiling results are obtained and described. For PC implementation the algorithm works fine and gives the real time enhanced speech output. But for DSK6713 implementation it is not the case. The enormous resources available on PC are responsible for the better performance. For DSK implementation as already indicated the main loop requires optimization as the execution can’t be completed within base sample time. More powerful platform like media processor DM6437 may provide the desired result. Further optimization of the code can be done through the algorithm tuning process[1].

[1] A paper entitled “Real Time and Embedded Implementation of Hybrid Algorithm for Speech Enhancement” is accepted for presentation in IEEE World Congress on Information and Communication Technologies (WICT 2011) Co-organized by Machine Intelligence Research Labs (MIR Labs) and University of Mumbai, Mumbai.

Chapter 9

Conclusions and Future Scopes

The work described in this thesis is focused on designing single channel real time speech enhancing system for the low SNR (0-5dB) range. By qualitative and quantitative analysis, the explanation in the thesis has shown that a practical single channel real time embedded speech enhancement technique can enhance signal quality. The hybrid approach suggested here can work in 0-5dB SNR range and can handle additive noise and convolutive distortion. Both the objective and subjective tests advocate the improvements compared to original algorithms in low SNR range (0-5dB) with the hybrid approach. The major difficulty in real time implementation is due to non-causal nature of RASTA algorithm. The sub-framing approach is used to solve this problem. This thesis has described the real time implementation of hybrid algorithm on PC as well as on DSK6713 through SIMULINK, Real Time Workshop and Target Support Package TC6. The profiling results are obtained and compared. For PC implementation the algorithm works fine and gives the real time enhanced speech output. But for DSK6713 implementation it is not the case. The enormous resources available on PC are responsible for the better performance. For DSK implementation as already indicated the main loop requires optimization as the execution can't be completed within base sample time. So it is concluded here that the hybrid algorithm is found suitable for real time embedded implementation in communication systems but requires optimization before final real time hardware implementation.

Table 9.1 shows the list of contemporaneous research work done in similar direction by other researchers. It is not possible to make exact comparisons as the results are reported using different database and performance measures. Also no one has reported about embedded or real time implementations. However, an attempt is made here to brief the most common results and can be compared with the hybrid approach. The hybrid approach offers comparatively appreciable results under different noisy and reverberant conditions.

<table>
<tr><th>Ref. no.</th><th>Brief of technique/principle</th><th colspan="4">Results reported</th></tr>
<tr><td rowspan="4">[1]</td><td rowspan="4">Multidimensional MMSE STSA estimators based on correlation between spectral components, an optimization parameter γ between 0 to 1 places lower and upper bounds.</td><td colspan="4">Advantageous at high SNR, PESQ under white noise:</td></tr>
<tr><td>γ</td><td>0</td><td>0.5</td><td>1</td></tr>
<tr><td>5 dB</td><td>1.3</td><td>1.24</td><td>1.21</td></tr>
<tr><td>10 dB</td><td>1.57</td><td>1.52</td><td>1.46</td></tr>
<tr><td>[2]</td><td>MMSE STSA85 for speech enhancement and independent component analysis (ICA) for noise estimation</td><td colspan="4">Real railway station is used to test the algorithm,
MOS: 4.2</td></tr>
</table>

[3]	β power MMSE STSA85 (β SA), β is power of the optimization cost function	β=-1 gives good compromise between noise reduction and speech distortion, MOS at 0 dB white noise: 2.7, PESQ under white noise:	
		0 dB	1.47
		5 dB	1.72
		10 dB	1.96
[4]	Maximum likelihood phase equivalence of speech and noise, Generalized Gamma distribution function for speech and noise spectral amplitudes	SSNR improvement compared to MMSE STSA85 under white noise:	
		0 dB	8.6%
		5 dB	7.4%
		10 dB	6.3%

Table 9.1 Comparison of results from the research papers published contemporaneous

Significant progress has been made in the development of single channel speech enhancement algorithms. Robust human-human communication with only two sensors and channels even in adverse conditions still haunts researchers in the field. Future work will explore some of the research directions pointed out in the thesis so far.

- The hybrid approach suggested here can be optimized by merging MMSE and RASTA algorithm together. Also the RASTA filters can be redesigned with better specifications.
- The algorithm is still unable to handle highly non-stationary noise. So such a scheme can be incorporated into hybrid algorithm.
- This thesis highlighted the importance of magnitude spectrum information for estimating the true clean speech magnitude in all algorithms. However, an attempt can be made at estimating phase and the complex spectral subtraction instead of magnitude spectral subtraction can be used.
- Instead of single channel approach a multichannel approach can also be investigated for viable real time implementation. In the present scenario the size and cost of microphone array limits the multichannel approach to challenge the single channel approach for speech enhancement. Multichannel speech enhancement algorithms are more robust for different noise conditions compared to single channel speech enhancement techniques [5]. With the advent of nano-technology and MEMS, the miniaturisation of

microphones is emerging quickly. This will overcome the said disadvantage of the multi channel techniques. In near future, many devices like mobile phones, laptops, PCs etc. can have the microphone array embedded into them. In fact, the research work in this direction has already begun [6]. The commercial noise canceller in mobile phones will be soon in market as reported in [7].

- The multi speaker separation problem can also be tackle by using array processing.
- Also the real time implementation suggested here is done using embedded target toolbox of MATLAB. So the developed assembly code may not be highly optimized. Further optimization of the assembly code can be done.
- With a faster DSP or media processor or even an FPGA implementation, a number of improvements can be made without the need for higher power algorithms.
- Also due to high complexity the algorithm can be implemented using soft computing techniques like fuzzy logic, neural network and genetic algorithms.

The investigation of these implications is a valuable topic for future research and might yield substantial improvements.

Chapter 10

References

Chapter 1

[1] Douglas O'Shaughnessy, Speech Communications, 2nd Ed., University press (India) Ltd., Hydrabad, 2001.

[2] Thomas F. Quatieri, Discrete-time Speech Signal Processing, 1st Indian reprint, Pearson education signal processing series, Delhi, 2004.

[3] J.Benesty, S.Makino, J.Cheng, Speech Enhancement, Springer series of signals and communication technology, Heidelberg, 2005.

[4] A.M.Kondoz, Digital Speech, 2nd Ed., Wiley India Pvt. Ltd., New Delhi, 2007.

[5] L.R.Rabiner, R.W.Schafer, Digital Processing of Speech Signals, 1st Ed., Pearson Education, Delhi, 2004.

[6] N.Magotra, Y.Yang, R.Whitman, P.Kasthuri, "Real time speech enhancement for wireless communication systems," Thirty first Asilomar Conf. on Signals, Systems and Computers, Vol. 1, pp. 159-63, November 1997.

[7] M.P.Cooke, "Making sense of everyday speech: a glimpsing account," Speech Separation by Humans and Machines, Edited by P.Divenyi, New York, 2004.

[8] T.V.Ramabadran, J.P.Ashley, M.J.McLauglin, "Background noise suppression for speech enhancement and coding," IEEE Workshop on Speech Coding for Telecommunications Proceedings, 1997.

[9] J.S.Lim, A.V.Oppenheim, "Enhancement and bandwidth compression of noisy speech," in Proc. IEEE, Vol. 67, pp.-1586-1604, December 1979.

[10] S.F.Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-27, pp.-113-120, April 1979.

[11] M.Berouti, R.Schwartz, J.Makhoul, "Enhancement of speech corrupted by acoustic noise," ICASSP'79, Vol.4, pp. 208-211, April 1979.

[12] S.Kamath, P.Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2002.

[13] B.L.Sim, Y.C.Tong, J.S.Chang, C.T.Tan, "A parametric formulation of the generalized spectral subtraction method," IEEE Trans. on Speech and Audio Processing, Vol. 6,no. 4, pp. 328-337, July 1998.

[14] R.J.McAulay, M.L.Malpass, "Speech enhancement using a soft decision noise suppression filter," IEEE Trans. on Acoustics, Speech and Signal Processin*g*, Vol. 28, pp. 137-145, 1980.

[15] Y.Ephrahim, D.Malah, “Speech enhancement using a minimum mean square error short time spectral amplitude estimator,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.ASSP-32,no. 6, pp. 1109-1121, December 1984.

[16] Y.Ephrahim, D.Malah, “Speech enhancement using a minimum mean square error log spectral amplitude estimator,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.ASSP-33, no. 2, pp. 443-445, April 1985.

[17] P.Scalart, J.V.Filho, “Speech enhancement based on a priori signal to noise ratio estimation,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 96, pp. 629-632, May 1996.

[18] P.Wolfe, S.Godsill “Simple alternatives to the Ephrahim and Malah suppression rule for speech enhancement,” in Proc. 11th IEEE Workshop Statistical Signal Processing, 2001, pp. 496-499, 2001.

[19] P.Vary, “Noise suppression by spectral magnitude estimation-mechanism and theoretical limits,” Signal Processing, Vol. 8, pp. 387-400, 1985.

[20] Md. Rashidul Islam, Hasibul Haque, M.Q. Apu, Md. Kamrul Hasan, “On the estimation of noise from pause regions for speech enhancement using spectral subtraction,” in Proc. 3rd International Conference on Electrical and computer Engineering ICECE 2004, Dhaka, Bangladesh, pp. 402-405, December 2004.

[21] Kotta Manohar, Preeti Rao, “Speech enhancement in non-stationary noise environments using noise properties,” Speech Communication, Vol. 48, pp. 96-109, 2006.

[22] A.Rezayee, S.Gazor, “An adaptive KLT approach for speech enhancement,” IEEE Trans. Speech and Audio processing, Vol. 9, pp. 87-95, February 2001.

[23] M.Gabrea, “Robust adaptive Kalman filtering based speech enhancement algorithm,” in Proc. IEEE ICASSP 2004, vol. 1, pp-I301-304, May 2004.

[24] J.H.Chang, S.Gazor, N.S.Kim and S.K.Mitra, “Multiple statistical models for soft decision in noisy speech enhancement,” Pattern Recognition, Vol. 40, pp. 11123-34, March 2007.

[25] S.Manikandan, “Speech enhancement based on wavelet de-noising,” ACAD journal, Vol.17, part 1/P7, 2006.

[26] X.Shen, L.Deng, “Discrete H_∞ filter design with application to speech enhancement,” in Proc. IEEE ICASSP’95, pp.1504-1507, 1995.

[27] Mingyang Wu, DeLiang Wang, “A two stage algorithm for enhancement of reverberant speech,” in Proc. IEEE ICASSP 2005, pp. 1085-88, 2005.

[28] Zhaozhang Jin, DeLiang Wang, “Learning to maximize signal-to-noise ratio for reverberant speech segregation,” in Proc. IEEE ICASSP 2009, pp. 4689-92, 2009.

[29] Serajul Haque, Roberto Togneri, Anthony Zaknich, "Auditory Features for Speech Recognition and Enhancement," VDM Verlag Dr, Müller Aktiengesellschaft & Co., Germany, 2009.

[30] N.Virag, "Single channel speech enhancement based on masking properties of the human auditory system," IEEE Trans. Speech and Audio Processing, Vol. 7, pp. 126-37, March 1999.

[31] Hynek Hermansky, Nelson Morgan "RASTA processing of speech," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.2, pp. 578-589, October1994.

[32] H.Hermanskey, E.A.Wan, C.Avendano, "Noise suppression in cellular communications," 2nd IEEE workshop on Interactive Voice Technology for Telecommunications Applications IVTT 94, Kyoto, Japan, September 1994.

[33] Hynek Hermansky, Nelson Morgan, Hans-Gunter Hirsch, "Recognition of speech in additive and convolutive noise based on RASTA spectral processing," IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-93, 1993.

[34] H.Hermansky, E.A.Wan, C.Avendano, "Speech enhancement based on temporal processing," International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95, 1995.

[35] Carlos Avendano, Hynek Hermansky, "On the properties of temporal processing for speech in adverse environments," in Proc. WASPA'97, Mohonk, New York, 1997.

[36] A.Hu, P.Loizou, "Subjective comparisons of speech enhancement algorithms," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, May 2006.

[37] R.Singaram, P.Guru Raghavendran, S.Shivaramakrishnan, R.Srinivasan, "Real time speech enhancement using Blackfin processor BF533," Journal of Instrumentation Society of India, Vol. 2, pp.67-79, November 2004.

[38] Chung-Hsien Yang, Jia-Ching Wang, Jhing Fa Wang, Chung Hsien Wu, Kai Hsing Chang, "Design and implementation of subspace based speech enhancement under in-car noisy environments," IEEE Trans. Vehicular Technology, Vol. 57, pp. 1466-79, May 2008.

[39] Creighton Doraiswami, "Real time implementation of an adaptive filter for speech enhancement," in Canadian Conference on Electrical and Computer Engineering, Vol. 4, pp. 2201-2204, May 2004.

[40] The NOIZEUS database.
Available: http://www.utdallas.edu/~loizou/speech/noize. Accessed on 30-10-2009.

[41] 3GPP2 Specifications.
Available: http://www.3gpp2.org/Public_html/specs/index.cfm. Accessed on 01-08-2009.

[42] MATLAB R2009a: Documentation CD - The Mathworks Inc.

[43] User Guide: Using MATLAB7.8 (R2009a) - The Mathworks Inc.

[44] User Guide: Using SIMULINK - The Mathworks Inc.

[45] User Guide: Creating Graphical User Interfaces - The Mathworks Inc.

[46] User Guide: Real Time Workshop ToolBox (use with MATLAB) - The Mathworks Inc.

[47] User Guide: Target Support Package TC6 ToolBox (use with MATLAB) - The Mathworks Inc.

[48] User Guide: Embedded IDE Link CC ToolBox (use with MATLAB) - The Mathworks Inc.

Chapter 2

[1] Douglas O'Shaughnessy, Speech Communications, 2nd Ed., University press (India) Ltd., Hydrabad, 2001.

[2] Thomas F. Quatieri, Discrete-time Speech Signal Processing, 1st Indian reprint, Pearson education signal processing series, Delhi, 2004.

[3] J.Benesty, S.Makino, J.Cheng, Speech Enhancement, Springer series of signals and communication technology, Heidelberg, 2005.

[4] A.M.Kondoz, Digital Speech, 2nd Ed., Wiley India Pvt. Ltd., New Delhi, 2007.

[5] L.R.Rabiner, R.W.Schafer, Digital Processing of Speech Signals, 1st Ed., Pearson Education, Delhi, 2004.

[6] P.Krishnamoorthy, S.R.Mahadeva Prasanna, "Processing noisy speech for enhancement," IETE Journal of Technical Review, Vol. 24, no. 5, pp. 351-57, September-October 2007.

[7] P.Krishnamoorthy, S.R.Mahadeva Prasanna, "Temporal and spectral processing methods for processing of degraded speech: a review," IETE Journal of Technical Review, Vol. 26, Issue 2, pp. 137-48, March-April 2009.

[8] T.V.Ramabadran, J.P.Ashley, M.J.McLauglin, "Background noise suppression for speech enhancement and coding," IEEE Workshop on Speech Coding for Telecommunications Proceedings, 1997.

[9] 3GPP2 Specifications.
Available: http://www.3gpp2.org/Public_html/specs/index.cfm. Accessed on 01-08-2009.

[10] J.H.Chang, S.Gazor, N.S.Kim, S.K.Mitra, "Multiple statistical models for soft decision in noisy speech enhancement," Pattern Recognition, Vol. 40, pp. 11123-34, March 2007.

[11] Chung-Hsien Yang, Jia-Ching Wang, Jhing Fa Wang, Chung Hsien Wu, Kai Hsing Chang, "Design and implementation of subspace based speech enhancement under in-car noisy environments," IEEE Trans. Vehicular Technology, Vol. 57, pp. 1466-79, May 2008.

[12] A.Rezayee, S.Gazor, "An adaptive KLT approach for speech enhancement," IEEE Trans. Speech and Audio processing, Vol. 9, pp. 87-95, February 2001.

[13] M.Gabrea, "Robust adaptive Kalman filtering based speech enhancement algorithm," in Proc. IEEE ICASSP 2004, vol. 1, pp.301-304, May 2004.

[14] X.Shen, L.Deng, “Discrete H_∞ filter design with application to speech enhancement,” in Proc. IEEE ICASSP’95, pp.1504-1507, 1995.

[15] N.Virag, “Single channel speech enhancement based on masking properties of the human auditory system,” IEEE Trans. on Speech and Audio processing, Vol. 7, pp. 126-37, March 1999.

[16] Mingyang Wu, DeLiang Wang, “A two stage algorithm for enhancement of reverberant speech,” in Proc. IEEE ICASSP 2005, pp. 1085-88, 2005.

[17] Zhaozhang Jin, DeLiang Wang, “Learning to maximize signal-to-noise ratio for reverberant speech segregation,” in Proc. IEEE ICASSP 2009, pp. 4689-92, 2009.

[18] M.P.Cooke, “Making sense of everyday speech: a glimpsing account,” in Speech Separation by Humans and Machines,” Edited by P.Divenyi, New York, 2004.

[19] Hynek Hermansky, Nelson Morgan “RASTA processing of speech,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.2, pp. 578-589, October 1994.

[20] Simon Haykin, Thomas Kailath, Adaptive Filter Theory, 4th Ed., Pearson Education, Delhi, 2005.

[21] S.M.Kuo, Woon Seng Gun, Digital Signal Processors – Architectures, Implementation and Applications, 1st Ed., Pearson Education, Delhi,2005.

Chapter 3

[1] Douglas O’Shaughnessy, Speech Communications, 2nd Ed., University press (India) Ltd., Hydrabad, 2001.

[2] Thomas F. Quatieri, Discrete-time Speech Signal Processing, 1st Indian reprint, Pearson education signal processing series, Delhi, 2004.

[3] J.Benesty, S.Makino, J.Cheng, Speech Enhancement, Springer series of signals and communication technology, Heidelberg, 2005.

[4] A.M.Kondoz, Digital Speech, 2nd Ed., Wiley India Pvt. Ltd., New Delhi, 2007.

[5] L.R.Rabiner, R.W.Schafer, Digital Processing of Speech Signals, 1st Ed., Pearson Education, Delhi, 2004.

[6] J.S.Lim, A.V.Oppenheim, “Enhancement and bandwidth compression of noisy speech,” in Proc. IEEE, Vol. 67, pp.-1586-1604, December 1979.

[7] S.F.Boll, “Suppression of acoustic noise in speech using spectral subtraction,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-27, pp.-113-120, April 1979.

[8] M.Berouti, R.Schwartz, J.Makhoul, “Enhancement of speech corrupted by acoustic noise,” ICASSP’79, Vol.4, pp. 208-211, April 1979.

[9] P.Vary, “Noise suppression by spectral magnitude estimation-mechanism and theoretical limits,” Signal Processing, Vol. 8, pp. 387-400, 1985.

[10] S.Kamath, P.Loizou, “A multi-band spectral subtraction method for enhancing speech corrupted by colored noise,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2002.

[11] B.L.Sim, Y.C.Tong, J.S.Chang, C.T.Tan, “A parametric formulation of the generalized spectral subtraction method,” IEEE Trans. on Speech and Audio Processing, Vol. 6,no. 4, pp. 328-337, July 1998.

[12] R.J.McAulay, M.L.Malpass, “Speech enhancement using a soft decision noise suppression filter,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 28, pp. 137-145, 1980.

[13] Y.Ephrahim, D.Malah, “Speech enhancement using a minimum mean square error short time spectral amplitude estimator,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.ASSP-32, no. 6, pp. 1109-1121, December 1984.

[14] Y.Ephrahim, D.Malah, “Speech enhancement using a minimum mean square error log spectral amplitude estimator,” IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.ASSP-33, no. 2, pp. 443-445, April 1985.

[15] P.Scalart, J.V.Filho, “Speech enhancement based on a priori signal to noise ratio estimation,” in Proc. IEEE International conference on Acoustics, Speech and Signal Processing ICASSP 96, pp. 629-632, May 1996.

[16] P.Wolfe, S.Godsill “Simple alternatives to the Ephrahim and Malah suppression rule for speech enhancement,” in Proc. 11th IEEE workshop Statistical signal processing, 2001, pp. 496-499, 2001.

[17] A.Hu, P.Loizou, “Subjective comparisons of speech enhancement algorithms,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 153-156, May 2006.

[18] O.Cappe, “Elimination of the musical noise phenomenon with the Ephrahim and Malah noise suppressor,” IEEE Trans. on Speech and Audio Processing, Vol. 2, No.2, pp. 346-349, 1994.

[19] Md. Rashidul Islam, Hasibul Haque, M.Q. Apu, Md. Kamrul Hasan, “On the estimation of noise from pause regions for speech enhancement using spectral subtraction,” in Proc. 3rd International Conference on Electrical and Computer Engineering ICECE 2004, Dhaka, Bangladesh, pp. 402-405, December 2004.

[20] R.Singaram, P.Guru Raghavendran, S.Shivaramakrishnan, R.Srinivasan, “Real time speech enhancement using Blackfin processor BF533,” Journal of Instrumentation Society of India, Vol. 37, No.2, pp. 67-79, November 2004.

[21] B.Jawerth, W.Sweldens, “An overview of wavelet based multi resolution analysis,” SIAM Review, Vol. 36, no. 3, pp. 377–412, 1994.

[22] D.L.Donoho, “De-noising by soft-thresholding,” IEEE Trans. on Information Theory, Vol. 41, no. 3, pp. 613–627, May 1995.

[23] D.L.Donoho, I.M.Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” Biometrika, Vol. 81, no. 3, pp. 425–455, 1994.

[24] I.M.Johnstone, B.W.Silverman, “Wavelet threshold estimators for data with correlated noise,” Journal of Royal Statistics Society, Vol. 59, pp. 319-351, 1997.

[25] M.Bahoura and J.Rouat, “Wavelet speech enhancement based on the Teager energy operator,” IEEE Signal Processing Letters, Vol.8, no.1, pp. 10-12, 2001.

[26] V.Balakrishnan, Nash Borges, Luke Parchment, “Wavelet de-noising and speech enhancement,” Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore.

[27] S.Manikandan, “Speech enhancement based on wavelet de-noising,” Academic Open Internet Journal on www.acadjournal.com Vol. 17, 2006.

[28] W.Voiers, “Interdependencies among measures of speech intelligibility and speech quality,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 703-705, 1980.

[29] Y.Hu, P.Loizou, “Evaluation of objective quality measures for speech enhancement,” IEEE Trans. on Audio, Speech, and Language Process., Vol. 16, no. 1, pp. 229–238, January 2008.

[30] S.Dimolitsas, “Objective speech distortion measures and their relevance to speech quality assessments,” in Proc. IEEE International Conference on Vision, Image and Signal Processing, pp. 317-324, 1989.

[31] J.H.L.Hansen, B.Pellom, “An effective quality evaluation protocol for speech enhancement algorithms,” in Proc. International Conference on Spoken Language Process, pp. 2819–2822, December 1998.

[32] S.Wang, A.Sekey, A.Gersho, “An objective measure for predicting subjective quality of speech coders,” IEEE Journal of Selected Areas of Communication, Vol. 10, no. 5, pp. 819-829, 1992.

[33] D.Klatt, “Prediction of perceived phonetic distance from critical band spectra,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 7, pp. 1278-1281, 1982.

[34] M.Karjalainen, “Sound quality measurements of audio systems based on models of auditory perception,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 9, pp. 132-135, 1984.

[35] J.G.Beerends, A.P.Hekstra, A.W.Rix, M.P.Hollier, “Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment part II -psychoacoustic model,” Journal of Audio Engineering Society, Vol. 50, no. 10, pp. 765–778, October 2002.

Chapter 4

[1] R.Singaram, P.Guru Raghavendran, S.Shivaramakrishnan, R.Srinivasan, "Real time speech enhancement using Blackfin processor BF533," Journal of Instrumentation Society of India, Vol. 37, No.2, pp. 67-79, November 2004.

[2] S.F.Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-27, pp.-113-120, April 1979.

[3] M.Berouti, R.Schwartz, J.Makhoul, "Enhancement of speech corrupted by acoustic noise," ICASSP'79, Vol.4, pp. 208-211, April 1979.

[4] S.Kamath, P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2002.

[5] P.Scalart, J.V.Filho, "Speech enhancement based on a priori signal to noise ratio estimation," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 96, pp. 629-632, May 1996.

[6] R.J.McAulay, M.L.Malpass, "Speech enhancement using a soft decision noise suppression filter," IEEE Trans. on Acoustics, Speech and Signal Processin*g*, Vol. 28, pp. 137-145, 1980.

[7] Y.Ephrahim, D.Malah, "Speech enhancement using a minimum mean square error short time spectral amplitude estimator," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.ASSP-32,no. 6, pp. 1109-1121, December 1984.

[8] Y.Ephrahim, D.Malah, "Speech enhancement using a minimum mean square error log spectral amplitude estimator," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.ASSP-33, no. 2, pp. 443-445, April 1985.

[9] MATLAB R2009a: Documentation CD - The Mathworks Inc.

[10] User Guide: Using MATLAB7.8 (R2009a) - The Mathworks Inc.

[11] User Guide: Creating Graphical User Interfaces - The Mathworks Inc.

[12] B.Grundlehner, J.Lecocq, R.Balan, J.Rosca, "Performance assessment method for speech enhancement systems," in Proc. 1st Annual IEEE BENELUX/DSP Valley Signal Processing Symposium, 2005.

[13] Y.Hu, P.C.Loizou, "Subjective comparison and evaluation of speech enhancement algorithms," Speech Communication, Vol. 49, pp. 588–601, 2007.

[14] IEEE Subcommittee, IEEE recommended practice for speech quality measurements, IEEE Trans. on Audio Electroacoustics, Vol. 17, Issue 3, pp. 225-246, September 1969.

[15] H.Hirsch, D.Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in Proc. ISCA ITRW ASR 2000, pp. 181-188, 2000.

[16] ITU, PESQ and objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs, ITU-T recommendation P.862, 2000.

[17] ITU-T, Objective measurement of active speech level, ITU-T recommendation P.56, 1993.

[18] The NOIZEUS database.
Available: http://www.utdallas.edu/~loizou/speech/noize. Accessed on 30-10-2009.

[19] The composite objective measures software.
Available: http://www.utdallas.edu/~loizou/speech/software.html. Accessed on 17-12-2009.

Chapter 5

[1] T.V.Ramabadran, J.P.Ashley, M.J.McLauglin, "Background noise suppression for speech enhancement and coding," IEEE Workshop on Speech Coding for Telecommunications Proceedings, 1997.

[2] 3GPP2 Specifications (2007).
Available: http://www.3gpp2.org/Public_html/specs/index.cfm

[3] J.Benesty, S.Makino, J.Cheng, Speech Enhancement, Springer series of signals and communication technology, 2005.

[4] Hynek Hermansky, Nelson Morgan "RASTA processing of speech," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.2, pp. 578-589, Oct.1994.

[5] H.Hermanskey, E.A.Wan, C.Avendano, "Noise suppression in cellular communications," 2nd IEEE workshop on Interactive Voice Technology for Telecommunications Applications IVTT 94, Kyoto, Japan, Sept. 1994.

[6] Hynek Hermansky, Nelson Morgan, Hans-Gunter Hirsch, "Recognition of speech in additive and convolutive noise based on RASTA spectral processing", IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-93, 1993.

[7] H.Hermansky, E.A.Wan, C.Avendano, "Speech enhancement based on temporal processing", International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95, 1995.

[8] Carlos Avendano, Hynek Hermansky, "On the Properties of Temporal Processing for Speech in Adverse Environments", in Proc. WASPA'97, Mohonk, New York, 1997.

[9] MATLAB R2009a: Documentation CD - The Mathworks Inc.

[10] User Guide: Using MATLAB7.8 (R2009a) - The Mathworks Inc.

[11] The NOIZEUS database.
Available: http://www.utdallas.edu/~loizou/speech/noize. Accessed on 30-10-2009.

[12] N. Jayant, J. Johnston, R.Safranek, “Signal compression based on models of human perception,” in Proc. IEEE, Vol. 81, no. 10, pp. 1385-1422, October 1993.

[13] Thomas F. Quatieri, Discrete-time Speech Signal Processing, 1st Indian reprint, Pearson education signal processing series, Delhi, 2004.

[14] D.Donahue, I.Johnson, “Ideal de-noising in an orthonormal basis chosen from a library of bases,” C.R. Academy of Science, Paris, France, Vol.1, no. 319, pp. 1317-1322, 1994.

[15] Douglas O’Shaughnessy, Speech Communications, 2nd Ed., University press (India) Ltd., Hydrabad, 2001.

[16] D.Sen, D.H.Irving, W.H.Holmes, “Use of an auditory model to improve speech coders,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 2, pp. 411-414, April 1993.

[17] A.Czyzewski, R.Krolikowski, “Noise reduction in audio signals based on the perceptual coding approach,” in Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New York, October 1999.

[18] S.Govidasamy, “A psychoacoustically motivated speech enhancement system,” M.E. Thesis, MIT, Dept. of Electrical Engineering and Computer Science, January 2000.

[19] N.Virag, “Single channel speech enhancement based on masking properties of the human auditory system,” IEEE Trans. on Speech and Audio Processing, Vol. 7, pp. 126-37, March 1999.

[20] S.Gstafsson, P.Jax, P.Vary, “A novel psychoacoustically motivated audio enhancement algorithm preserving background noise characteristics,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 1, pp. 397-400, May 1998.

Chapter 6

[1] The NOIZEUS database.
Available: http://www.utdallas.edu/~loizou/speech/noize. Accessed on 30-10-2009.

[2] Y.Hu, P.C.Loizou, “Subjective comparison and evaluation of speech enhancement algorithms,” Speech Communication, Vol. 49, pp. 588–601, 2007.

[3] IEEE Subcommittee, IEEE recommended practice for speech quality measurements, IEEE Trans. on Audio Electroacoustics, Vol. 17, Issue 3, pp. 225-246, September 1969.

[4] D.Goodman, R. Nash, “Subjective quality of the speech transmission conditions in seven different countries,” IEEE Trans. on Communication, Vol. 30, no. 4, pp. 642-654, 1982.

[5] ITU, Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems, ITU-R recommendation BS.1116-1, 1997.

[6] ITU, Subjective performance assessment of telephone band and wideband digital codecs, ITU-T recommendation P.830, 1998

[7] The AIR database with MATLAB code.
Available: http://www.ind.rwth-aachen.de/en/research/speech-and-audio-processing/aachen-impulse-response-database/. Accessed on 10-01-2011.

Chapter 7

[1] TMS320C6713 datasheet.
Available: www.ti.com. Accessed on 24-04-2010.

[2] User Guide: CCS IDE V3.3 – Texas Instruments.

[3] User Guide: Using SIMULINK - The Mathworks Inc.

[4] User Guide: Real Time Workshop ToolBox (use with MATLAB) - The Mathworks Inc.

[5] User Guide: Target Support Package TC6 ToolBox (use with MATLAB) - The Mathworks Inc.

[6] User Guide: Embedded IDE Link CC ToolBox (use with MATLAB) - The Mathworks Inc.

Chapter 8

[1] S.M.Kuo, B.H.Lee, W.Tian, Real Time Digital Signal Processing: Implementations and Applications, 2nd Ed., John Wiley & Sons Ltd., West Susex, England, 2006.

[2] User Guide: Target Support Package TC6 ToolBox (use with MATLAB) - The Mathworks Inc.

[3] User Guide: CCS IDE V3.3 – Texas Instruments.

Chapter 9

[1] E.Plourde, B.Champagne, “Multidimensional STSA estimators for speech enhancement with correlated spectral components,” IEEE Trans. on Signal Processing, Vol. 59, no. 7, pp. 3013-3024, July 2011.

[2] R.Okamoto, Y.Takahashi, H.Saruwatari, K.Shikano, “MMSE STSA estimator with non-stationary noise estimation based on ICA for high-quality speech enhancement,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP- 2010, pp. 4778-4781, 2010.

[3] E.Plourde, B.Champagne, “Further analysis of the β-order MMSE STSA estimator for speech enhancement,” in Proc. Canadian Conference on Electrical and Computer Engineering, CCECE 2007, pp. 1594-1597, 2007.

[4] B.J.Borgstrom, A.Alwan, “A unified framework for designing optimal STSA estimators assuming maximum likelihood phase equivalence of speech and noise,” IEEE Trans. on Audio, Speech and Language Processing, Vol. 19, no. 8, pp. 2579-2590, 2011.

[5] Prof. Dr. Walter Kellermann, Acoustic source localization based on independent component analysis.
Available: http://www.lms.lnt.de/research/activity/audio/topics/local. Accessed on 06-06-2011.

[6] Hakon Strande, program manager, Microsoft Corporation, Microphone array support in windows longhorn.
Available: http://download.microsoft.com/download/9/8/f/98f3fe47-dfc3-4e74-92a3-088782200fe7/twen05009_winhec05.ppt. Accessed on 06-06-2011.

[7] Stacey Moser, Texas Instruments, Cancel noise in your mobile phones and headphones, 02-06-2011.
Available: http://www.mobiledevdesign.com/tutorials/cancel-noise-mobile-phones-headphones-0611. Accessed on 19-08-2011.

Appendix-A

Publications/ Presentations Based on Research Work

Table A.1 lists the papers presented/published in various national/international conferences/symposiums/journals and indexed in databases; based on the work described in the thesis.

Sr. No.	Conference/Symposium/Journal	Year	Title
1.	National Technical Paper Contest-2010 (NTPC-2010) for seniors at IETE Vadodara centre, Vadodara. (Won 3rd prize)	Mar. 2010	Requirements and Scope of Speech Enhancement Techniques in Present Speech Communication Systems
2.	National conference on Wireless Communication and VLSI design (NCWCVD-2010) Organized by GEC, Gwalior and IEEE MP Subsection.	Mar. 2010	Performance Evaluation of STSA based Speech Enhancement Techniques for Speech Communication Systems
3.	National conference on Information Sciences (NCIS-2010) Organized by MCIS, Manipal University, Manipal.	Apr. 2010	A Review on Single Channel Speech Enhancement Techniques for Wireless Communication Systems
4.	IEEE International conference on Communication, Network and Computing (CNC 2010) Organized by ACEEE at Calicut. IEEE CS- CPS ISBN: 978-0-7695-4209-6. Listed in IEEE Xplore by IEEE Computer society. DOI: 10.1109/CNC.2010.13, pp.19-23, Archived in ACM digital library.	Oct. 2010	Objective Evaluation of STSA based Speech Enhancement Techniques for Speech Communication Systems with Proposed Modifications
5.	IEEE symposium on Computers and Informatics (ISCI 2011) Organized by IEEE Malaysia section at Kuala Lumpur, Malaysia. ISBN: 978-1-61284-690-3. Listed and indexed in IEEE Xplore DOI:10.1109/ISCI.2011. 5958902, pp.159-162, INSPEC 12123318.	Mar. 2011	Evaluation of RASTA Approach with Modified Parameters for Speech Enhancement in Communication Systems
6.	International Journal of Recent Trends in Engineering and Technology, (IJRTET), Vol. 4, No 4, Nov. 2010, pp. 146-149, ACEEE, USA. ISSN (Online):2158-5563 ISSN (Print): 2158-5555 Archived in SEARCH digital library.	Nov. 2010	Simulation and Real Time Implementation of Spectral Subtraction and Wavelet De-Noising Embedded Algorithms for Speech Enhancement

7.	IEEE World Congress on Information and Communication Technologies (WICT 2011) co-organized by Machine Intelligence Research Labs (MIR Labs) and University of Mumbai, Mumbai.	Dec. 2011	Real Time and Embedded Implementation of Hybrid Algorithm for Speech Enhancement

Table A.1 List of papers published/presented

Appendix-B

Training Programs Attended in the Area During Research Work

Appendix-B Training Programs Attended in the Area During Research Work

Table B.1 lists the short term training programs attended during the research work.

Sr. No.	Subject	Place	Year	Duration
1.	Winter School on Speech and Audio Processing (WISSAP 2010) (Audio Content Analysis and Retrieval)	I.I.T, Bombay.	Jan. 2010	04 Days
2.	Winter School on Speech and Audio Processing (WISSAP 2011) (Speech Enhancement)	I.I.T., Guwahati	Jan. 2011	04 Days

Table B.1 List of short term training programs attended