Enhancement of Electrolaryngeal Speech by Background Noise Reduction and Spectral Compensation

A dissertation
submitted in partial fulfillment of the

requirements for the degree of

Master of Technology

by

Priyanko Mitra
(Roll No. 04307022)

under the supervision of

Prof. P. C. Pandey

Department of Electrical Engineering
Indian Institute of Technology, Bombay
July 2006

Priyanko Mitra / Prof. P.C. Pandey (Supervisor): “Enhancement of electrolaryngeal speech by background noise reduction and spectral compensation”, *M.Tech. dissertation*, Department of Electrical Engineering, Indian Institute of Technology, Bombay, July 2006.

ABSTRACT

Transcervical electrolarynx is a vibrator held against the neck tissue by a laryngectomy patient to provide excitation to the vocal tract, as a substitute to that provided by glottal vibrations. Major problems with electrolaryngeal speech are lack of voicing and pitch control, deficiency of low frequency content, and background noise from the vibrator and vibrator-tissue interface. Pitch synchronous application of spectral subtraction has been earlier used for reducing the self leakage noise, with averaging based noise estimation (ABNE) on an initial segment with closed lips. As the leakage noise spectrum varies with speech production, and vibrator orientation and pressure, quantile based noise estimation (QBNE) has been used for dynamically estimating the noise spectrum, but had residual noise in the silence regions. A real time implementation has been earlier done without phase reconstruction or spectral compensation. In this project, a real time system has been devised for background noise reduction along with low frequency compensation. Effective noise reduction comparable to that in offline processing is obtained in the real time implementation of ABNE using Analog Devices 16-bit fixed point Blackfin processor ADSP BF 533. A dynamic minimum statistics based noise estimation (MSBNE), with two-pitch period analysis frames and one period overlap is investigated. Minimum value of each spectral sample in a set of past frames is used for dynamically estimating the noise magnitude spectrum. Smoothing of the estimated noise spectrum resulted in better noise reduction. Compared to QBNE, the MS based method gives less residual noise, takes much lower processing time, and requires a lower number of windows for optimum noise updating. Real time implementation of MSBNE showed reasonable noise reduction, comparable to that in offline processing.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my respected guide, Prof. P.C. Pandey, for his invaluable guidance, support, and encouragement throughout the course of this project. I am thankful to Prof. Preeti Rao for her valuable suggestions during the progress of the project.

I express my thanks to G. Gidda Reddy and A. R. Jayan for their assistance and helpful advice in different issues of my project. I would also like to thank K. Pavan Kumar and Ram Singh of Digital Audio Processing Lab, with whom I shared many interesting discussions. I am also thankful to all my friends in the Signal Processing and Instrumentation Lab for their whole-hearted support during the tenure of this project.

Priyanko Mitra

July 2006

CONTENTS

Appendix

LIST OF SYMBOLS

Symbol	Explanation
$h_v(t)$	Impulse response of vocal tract
$h_l(t)$	Impulse response of leakage path
$e(t)$	Excitation pulse
$s(t)$	Speech sound
$l(t)$	Leakage sound
$x(t)$	Noisy speech
$y(t)$	Cleaned speech
b_m	FIR filter coefficients
μ	*C*onvergence parameter
α	Over-subtraction factor
β	Spectral floor factor
γ	Exponent factor
$L(k)$	Average magnitude spectrum of noise
N	Window length
$v_k(n)$	Signal estimate of n'th sample point in k'th iteration
$M_k(\omega)$	Magnitude of Fourier transform of $v_k(n)$
$\theta_k(\omega)$	Phase of Fourier transform of $v_k(n)$
$c(n)$	Cepstral co-efficient
$D(\omega_k)$	Magnitude buffer for holding spectral values of past few frames in minimum statistic technique
$I(\omega_k)$	Index buffer for holding frame number of corresponding value in magnitude buffer

LIST OF ABBREVIATIONS

Abbreviation	Explanation
ABNE	Average based noise estimation
QBNE	Quantile based noise estimation
MSBNE	Minimum statistics based noise estimation
EELS	Enhancement of electrolaryngeal speech
DSP	Digital signal processor
SNR	Signal-to-noise ratio
FIR	Finite impulse response
LMS	Least mean square
FFT	Fast Fourier transform
IFFT	Inverse fast Fourier transform
STFT	Short time Fourier transform
AMT	Auditory masking threshold
SAMT	Supplementary auditory masking threshold
VAD	Voice activity detectors
MS	Minimum statistics
MCRA	Minima controlled recursive averaging
IMCRA	Improved minima controlled recursive averaging
TI	Texas Instruments
GUI	Graphical user interface
ADC	Analog-to-digital converter
DAC	Digital-to-analog converter
DSK	Digital starter kit
DMA	Direct memory access
EDMA	Enhanced direct memory access
EMIF	External memory interface
McBSP	Multi-channel buffered serial port
CPU	Central processing unit
USB	Universal serial bus

BTC	Background telemetry channel
JTAG	Joint test action group
MAC	Multiply-accumulate
RISC	Reduced instruction set computing
PPI	Parallel peripheral interface
SPI	Serial peripheral interface
UART	Universal asynchronous receiver transmitter
RTC	Real-time clock
I/O	Input/output
PGA	Programmable gain amplifier
MUX	Multiplexer
HEX	Hexadecimal
2-D	Two-dimensional
AD	Analog Devices
MSE	Mean squared error

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Problem overview

In natural speech production, the lungs provide the air stream, the vocal chords in the larynx provide the vibration source for production of sound, and the vocal tract provides the spectral shaping of the resulting speech. Periodic sources result in voiced speech, while noisy and aperiodic sources cause unvoiced speech [1]. Laryngeal cancer often necessitates complete surgical removal of the larynx. Having lost the natural voicing source, the person needs an alternative voicing aid to communicate.

The artificial larynx [2], [3] is a device meant to substitute the natural larynx in its absence, and provide vibrations to the vocal tract, which are necessary for speech production. The most popular device of this type is the electronic artificial larynx or electrolarynx. This portable hand-held device consists of an electronic vibration generator which rests against the throat, and transmits pulses through the neck tissue to the vocal tract. The various resonances of the vocal tract shape the harmonic spectrum of the vocal tract vibrations, and this results in speech. The device enables adequate communication, but the resulting speech has an unnatural quality and is significantly less intelligible than normal speech. Voiced segments substitute the unvoiced speech segments. The speech is monotonic due to lack of pitch control. It is deficient in low frequency content due to design of the vibrator and poor coupling efficiency through the throat tissue. Moreover, the major problem with the electrolarynx is that the output speech is corrupted by noise generated from the vibrator and its interface with the neck, which seriously degrades the speech quality and intelligibility [4], [5], [6].

1.2 Project objective

The speech produced with electrolarynx suffers from the presence of background noise leaked from the vibrator housing and its interface with the neck tissue. Use of acoustical shielding of the instrument yielded only a marginal reduction in the directly radiated

sound from vibrator [7]. Different signal processing techniques for suppression of the background noise [8], [9], [10], [11] have been investigated in the past few years. Earlier work in our lab has shown that pitch synchronous spectral subtraction with average (ABNE) and quantile based noise estimation (QBNE) give effective noise suppression [12], [13]. In ABNE, average magnitude spectrum of noise estimated during an initial non-speech region is subtracted from the magnitude spectrum of the noisy speech and combined with the retained noisy phase for resynthesis. But actually the background noise varies due to variations in the place of coupling of the vibrator to the neck tissue, the amount of coupling, and changing impedance offered by the opening and closing of the mouth. QBNE dynamically updates the noise estimate without any speech/non-speech classification [14]. Offline and real time implementations of QBNE [12], [15] showed a certain amount of residual noise in the silence regions. Spectral compensation or phase reconstruction was not done in the above implementation.

The objective of this project is to implement a real time speech enhancement system which incorporates low frequency deficit compensation, dynamic noise estimation, and recombination of the cleaned magnitude spectrum with a clean phase spectrum.

1.3 Dissertation outline

Chapter 2 describes the mechanism of speech production using natural and artificial larynx, characteristics of electrolaryngeal speech, and the background noise generation model. Chapter 3 reviews various signal-processing techniques for the removal of the background noise. Chapter 4 describes the real-time implementation details of this project after reviewing the earlier work done by Budiredla [15]. The first section of Chapter 5 describes the results of MATLAB investigations with different signal processing algorithms, with special emphasis on the minimum statistics based noise estimation technique. Performance of the DSP based real time investigations are discussed in the second half of the chapter. Chapter 6 gives a summary of the report, conclusions which can be drawn from the results, and suggestions for further work.

Chapter 2

ELECTROLARYNGEAL SPEECH

This chapter presents a review of the mechanism of normal speech production and the characteristics of electrolaryngeal speech.

2.1 Normal speech production

Speech signal is a dynamic information-bearing acoustic waveform. These waves are produced due to the sound pressure generated in the mouth of a speaker as a result of some sequence of coordinated movements of a series of structures in the human vocal system. A schematic of the normal speech production system [1] is shown in Fig. 2.1. Speech production can be viewed as a filtering operation in which, a sound source excites a vocal tract filter; the source may be periodic, resulting in voiced speech, or noisy and aperiodic, causing unvoiced speech. The voicing source occurs in the larynx, at the base of the vocal tract, where airflow can be interrupted periodically by vibrating vocal folds. Unvoiced speech is noisy due to random nature of the excitation generated at a narrow constriction either at the vocal folds or at a place in the vocal tract.

2.2 Electrolaryngeal speech production

Laryngeal cancer often necessitates complete surgical removal of the larynx. Hence the patient loses his/her natural voicing source, and needs an alternative voicing source in order to speak. The artificial larynx is a prosthesis meant to provide vibrations, which are necessary for speech production. A number of artificial larynxes have been developed [2], [3], and these can be broadly classified into pneumatic larynges and electronic larynges. The pneumatic artificial larynges make use of the air exhaled out from the lungs to produce the vibrations. Based upon the placement of the artificial larynx, these are sub-classified into two groups as external pneumatic larynges and internal pneumatic larynges. A description of pneumatic larynges can be found in [2], [3].

Electronic artificial larynx or electrolarynx is the most widely used device. In this device, pulses from the vibrating diaphragm, which rests against the throat, are transmitted through the neck tissue to the vocal tract. The various resonances of the vocal

tract shape the harmonic spectrum of the vocal tract vibrations, and this result in speech. A schematic of speech production using this device is shown in Fig. 2.2.

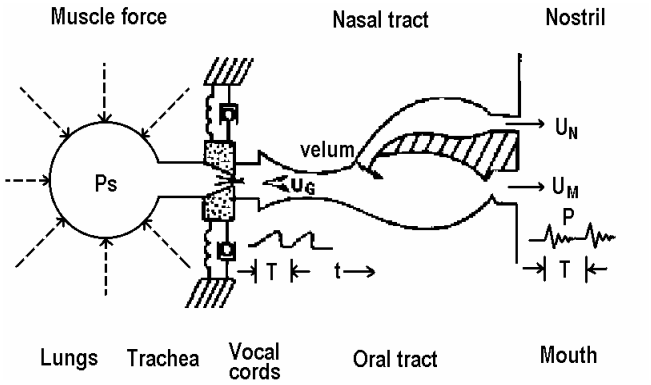


Fig. 2.1 Schematic of normal speech production [1].

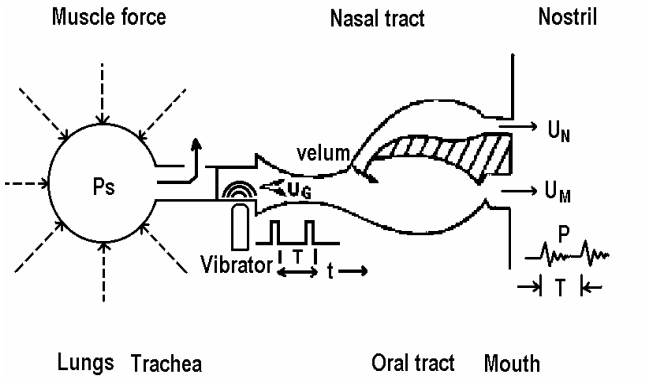


Fig. 2.2 Schematic of speech production with an electronic artificial larynx [13].

2.3 Characteristics of electrolaryngeal speech

Weiss *et al.* [4] reported a detailed study of perceptual and acoustical characteristics of electrolaryngeal speech, using the device Western Electric Model 5. The main problems associated with the external type electronic artificial larynx are :

1. Monotonicity of speech
2. Difficulty in co-ordinating controls
3. Spectral deficit
4. Background or self leakage noise

Earlier models of electrolarynx did not have any pitch control mechanism and the speech was monotonic with an unnatural quality. However, discrete and continuous pitch variation techniques have been incorporated in later models. The electrolarynx is a hand-held device, which has to be coupled to the neck during its operation. During speech production, the speaker has to co-ordinate the manual pitch control with the movement of articulators, which is difficult. It may either lead to a monotonous sound or improper words until the speaker attains the necessary expertise. To solve this problem to a certain extent, a semi-automatic pitch control circuit has been reported [2] and this is briefly described in Appendix A.1.

The artificial larynx when coupled to the neck causes the vibrations to propagate through the neck tissue on to the vocal tract. During propagation of vibrations through the non-uniform mass of neck-tissue, there is a frequency dependent attenuation and non-linear phase shift of the harmonics, because of the mass-spring viscous damping effect. Secondly, since the transmission loss is inversely proportional to the frequency, the low frequency components in the signal are attenuated more. Sometimes they may not propagate through the medium at all, especially when the neck muscles have thickened due to the radiation, generally given after the laryngectomy operation. Hence, the speech produced by an electrolarynx is deficient in low frequency content.

The major problem associated with an electronic artificial larynx is a steady background noise. The front end of the vibrator membrane is coupled to the neck and the back end to the air in the instrument housing. Leakage of the acoustic energy from the housing to the air outside is responsible for the production of the noise which gets added to the speech from the lips and is presented to the listener. Leakage of vibrations from the front end of the vibrator plate due to improper coupling of the vibrator to the neck tissue also contributes to the background noise, which might degrade the electrolaryngeal speech in two ways. First, the noise may result in loss of intelligibility, especially at low SNRs (defined as the ratio of the average level of the vocal peaks in the electrolaryngeal speech to the level of radiated sound measured with the speaker's mouth closed), resulting in confusions between voiced and unvoiced word-initial stop consonants. This is because the presence of a periodic low-frequency signal during the closed portions of voiced stops is an acoustic clue that distinguishes between voiced and voiceless stops. But, due to the continuous operation of the vibrator throughout the utterance, the closure portion of both voiced and voiceless stops may consist of the periodic radiated source noise. Second, the noise may contribute to the unnaturalness and poor quality of electrolaryngeal speech

relative to naturally spoken speech. This is because the electrolaryngeal speech has a stronger concentration of energy between 400 and 1000 Hz and between 2 and 4 kHz [4]. While this may not directly affect intelligibility, the masking effect of noise, especially on the higher formants, can contribute to the unnaturalness and poor quality of electrolaryngeal speech.

Chapter 3

BACKGROUND NOISE REDUCTION IN ELECTROLARYNGEAL SPEECH

Presence of background noise due to self leakage of vibration energy is a main cause of poor quality of electrolaryngeal speech. The self leakage can be reduced by acoustic shielding of the vibrator assembly and by improving the vibrator design. Some improvement was obtained by applying a one inch thick foam as acoustic shield around the electrolarynx [7]. However, the shielding effect of the insulation was counterbalanced by the lack of mechanical damping that is normally provided by hand holding the device. The thick insulation also made it difficult to hold the device. It has also been pointed out that shielding cannot reduce the leakage from the vibrator-tissue interface. Self leakage background noise can be reduced by employing appropriate signal processing techniques. Signal processing can also be employed for enhancing the electrolaryngeal speech by providing spectral compensation for low frequency deficit. Some of these techniques are reviewed here.

3.1 Enhancement of electrolaryngeal speech by adaptive filtering

An adaptive filter for noise removal is based on the assumption that the desired signal is corrupted by an uncorrelated noise, and a reference signal is available that is in some way correlated with the noise, but uncorrelated with the desired signal. Espy-Wilson *et al.* [7] have reported a technique involving simultaneous recording of acoustic signal near the lips and near the electrolarynx, and employing an adaptive filtering, for background noise reduction. Based on minimum mean-square error, the filter coefficients are re-estimated at every sample and adapted dynamically to changes in the input signal. Fig. 3.1 shows the block diagram of such a scheme of adaptive filter. There are two inputs to the filter, one is the noisy speech $x(n) = s(n) + l(n)$, where, $s(n)$ is the speech signal and $l(n)$ is the background interference or the noise. The second signal $r(n)$ is correlated with the noise

$l(n)$. The error $e(n)$ between $x(n)$ and $r(n)$ is used to modify the coefficients of the filter. The coefficients of the filter, b_m's, are adaptively updated based on the minimum mean square error criterion. When the error is minimized, the output of the filter is a good estimate of the noise, and is subtracted from the raw speech signal to produce noise-free speech. The decision to turn on and off the adaptation is based on whether the segment is voiced or unvoiced. For this purpose, a windowed average energy detector was used. Whenever the energy exceeds a threshold, it is classified as a voiced segment and the adaptation is prevented. The filter coefficients are retained to the last value. Whenever the average energy in the window is below the threshold, the interval is marked non-sonorant and the adaptation is allowed to proceed normally from the last value. The FIR filter output is given as

$$y(n) = \sum_{m=0}^{N-1} b_m(n) r(n-m) \tag{3.1}$$

The error is given as

$$e(n) = x(n) - y(n) \tag{3.2}$$

The coefficients of the FIR filter, b_m's, are updated on the basis of the previous coefficients as

$$b_m(n) = b_m(n-1) + \mu e(n) r(n-m), \qquad m = 0 \ldots N\text{-}1 \tag{3.3}$$

where μ is the convergence parameter and N is filter length. When LMS algorithm minimizes the mean square error $e(n)$, the impulse response of the FIR filter gives estimate of the leakage sound $y(n) \approx l(n)$, and the error $e(n)$ is the noise removed signal output.

The adaptation size plays an important role in determining the behavior of the LMS algorithm. Increasing the magnitude of the adaptation constant increases the step size of the iteration thereby increasing the speed with which the algorithm converges. However, it also increases the likelihood of the algorithm responding to spurious events and increases the mean squared error. Increasing the value beyond a certain number results in instability of the algorithm.

Perceptual tests [7] showed that noise reduction using adaptive filtering was more effective during the non-sonorant or the low energy intervals compared to the sonorant periods. Also, the intelligibility of the processed output was, on an average, comparable to that of its unprocessed counterpart. This technique uses two inputs, and the assumption that the signal $r(n)$ is an estimate of the noise present in the signal $x(n)$. However the input

$r(n)$ consists of noise as well as some portion of speech. The presence of the speech in the noise affects the quality of the output processed with the LMS algorithm.

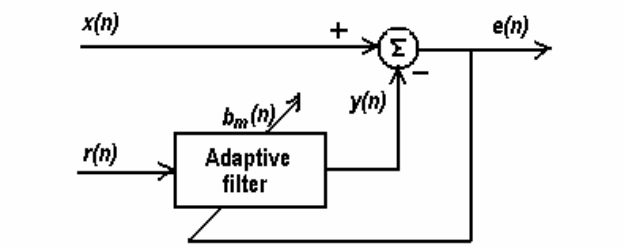


Fig. 3.1 Two input adaptive filter for noise reduction, as used in [41]

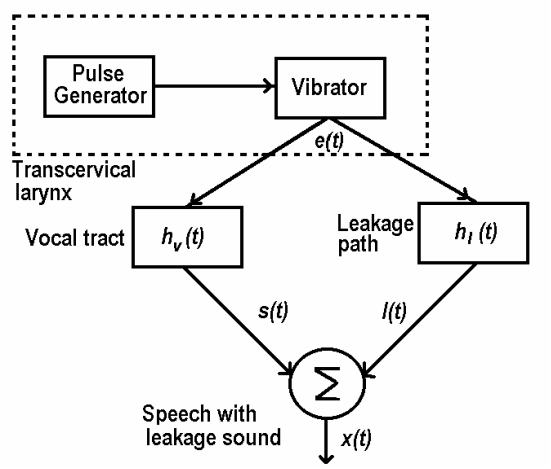


Fig. 3.2 Model of background noise generation in electrolaryngeal speech [13]

3.2 Enhancement of speech by pitch synchronous spectral subtraction

Spectral subtraction is a well-known noise reduction method based on the short time spectral magnitude estimation technique. The basic power spectral subtraction technique, as proposed by Boll [8], is popular due to its simple underlying concept and its effectiveness in enhancing speech degraded by additive noise. A model of the leakage sound generation during the use of electrolarynx is shown in Fig. 3.2. The vibrations generated by the vibrator diaphragm have two paths. The first path is through the neck tissue and the vocal tract. Its impulse response, $h_v(t)$, depends on the length and configuration of the vocal tract, the place of coupling of the vibrator, the amount of coupling, etc. Excitation, $e(t)$, passing through this path results in speech signal, $s(t)$. The second path of the vibrations is through the surroundings, and this leakage component,

$l(t)$, which is assumed to be uncorrelated to the speech signal, gets added to the useful speech, $s(t)$, and deteriorates its intelligibility.

The basic principle of the spectral subtraction method for enhancement of noisy speech is to subtract the power spectrum of noise from that of the noisy speech. An estimate of the noise signal is obtained during silence or non-speech activity in the signal. The principal assumption made in this method is that the clean speech and the noise are uncorrelated, and therefore the power spectrum of the noisy speech signal equals the sum of power spectrum of noise and clean speech [11], [12]. In case of electrolaryngeal speech, the speech signal and interference due to leakage are strongly correlated, and as such spectral subtraction cannot be used. However, it has been shown in [10], [11] that if the spectra are calculated pitch synchronously, the speech and interference become uncorrelated and spectral subtraction can be employed.

With reference to Fig. 3.2, let $x(n)$ be the noisy speech, $h_v(n)$ be the impulse response of the vocal tract, $h_l(n)$ be the impulse response of the leakage path, and $e(n)$ be the excitation signal. The noisy speech signal is given as

$$x(n) = s(n) + l(n) \tag{3.4}$$

where $s(n)$ is the speech signal and $l(n)$ is the background interference or the leakage noise. If $h_v(n)$ and $h_l(n)$ are the impulse responses of the vocal tract path and the leakage path respectively, then

$$s(n) = e(n)*h_v(n) \tag{3.5}$$

$$l(n) = e(n)*h_l(n) \tag{3.6}$$

Taking short-time Fourier transform on either side of (3.1), we get

$$X_n(e^{j\omega}) = E_n(e^{j\omega})[H_{vn}(e^{j\omega}) + H_{ln}(e^{j\omega})] \tag{3.7}$$

Considering the impulse responses of the vocal tract and leakage path to be uncorrelated, we get

$$|X_n(e^{j\omega})|^2 = |E_n(e^{j\omega})|^2[|H_{vn}(e^{j\omega})|^2 + |H_{ln}(e^{j\omega})|^2] \tag{3.8}$$

If the short-time spectra are evaluated using a pitch synchronous window, $|E_n(e^{j\omega})|^2$ can be considered as constant $|E(e^{j\omega})|^2$. During non-speech interval, $e(n)*h_v(n)$ will be negligible and the noise spectrum is given as

$$|X_n(e^{j\omega})|^2_{\text{nospeech}} = |L_n(e^{j\omega})|^2 = |E_n(e^{j\omega})|^2 |H_{ln}(e^{j\omega})|^2 \tag{3.9}$$

By averaging $|L_n(e^{j\omega})|^2$ during the non-speech duration, we can obtain the mean squared spectrum of the noise $|L(e^{j\omega})|^2$. This estimation of the noise power spectra can be used for spectral subtraction during the noisy speech segments.

For implementation of the technique [11], squared magnitudes of the FFT of a number of adjacent windowed segments in non-speech segment are averaged to get the mean squared noise spectrum. This is termed as averaging based noise estimation (ABNE). During speech segment, the noisy speech is windowed by the same window as in earlier mode, and its magnitude and phase spectra are obtained. The phase spectrum is retained for resynthesis. From the squared magnitude spectrum of noisy speech, the mean squared spectrum of noise, determined during the noise estimation mode is subtracted.

$$|Y_n(k)|^2 = |X_n(k)|^2 - |L(k)|^2 \tag{3.10}$$

The resulting magnitude spectrum from the power spectrum is then combined with the earlier phase spectrum,

$$Y_n(k) = |Y_n(k)|\, e^{j\angle Xn(k)} \tag{3.11}$$

Its inverse FFT is taken as the clean speech signal $y(n)$ during the window duration.

$$y_n(m) = IFFT[(Y_n(k)] \tag{3.12}$$

The speech signal is reconstructed by using overlap-add method.

3.3 Shortcomings in spectral subtraction

While the spectral subtraction method is easily implemented and it effectively reduces the noise present in the corrupted signal, there exist some shortcomings, as described below.

3.3.1 Residual noise (musical noise)

It is obvious that the effectiveness of the noise removal process is dependent on obtaining an accurate spectral estimate of the noise signal. The better the noise estimate, the lesser the residual noise content in the modified spectrum. However, since the noise spectrum cannot be directly obtained, we have to use an estimate of the noise. Hence there are some significant variations between the estimated noise spectrum and the actual noise content present in the instantaneous speech spectrum. The subtraction of these quantities results in the presence of isolated residual noise levels of large variance. These residual spectral content manifest themselves in the reconstructed signal as varying tonal sounds resulting in a musical disturbance of an unnatural quality. Hence there is a trade-off between the amount of noise reduction and speech distortion due to the underlying process.

3.3.2 Distortion due to half/full wave rectification

The modified speech spectrum obtained may contain some negative values due to the errors in the estimated noise spectrum. These values are rectified using half-wave rectification (set to zero) or full-wave rectification (set to its absolute value). This can lead to further distortions in the resulting time signal.

3.3.3 Roughening of speech due to the noisy phase

The phase of the noise-corrupted signal is not enhanced before being combined with the modified spectrum to regenerate the enhanced time signal. But, estimating the phase of the clean speech will greatly increase the complexity of the method. Moreover, the distortion due to noisy phase information is not very significant compared to that of the magnitude spectrum. Hence the use of the noisy phase information is considered to be an acceptable practice in the reconstruction of the enhanced speech.

3.4 Modifications to spectral subtraction

Several variants of spectral subtraction method originally developed by Boll [8] have been developed to address the problems of the basic technique, especially the presence of musical noise. This section deals with some of the techniques and enhancements which can be applied to the enhancement of electrolaryngeal speech.

3.4.1 Spectral subtraction using over-subtraction and spectral floor

An important variation of spectral subtraction was proposed by Berouti *et. al*. [9] for reduction of residual noise. This method is also called modified spectral subtraction. The proposed technique could be expressed as

$$|Y_n(k)|^\gamma = |X_n(k)|^\gamma - \alpha|L(k)|^\gamma \tag{3.13}$$

$$\begin{aligned} |Y'_n(k)|^\gamma &= |Y_n(k)|^\gamma \quad \text{if } |Y_n(k)|^\gamma > \beta|L(k)|^\gamma \\ &= \beta|L(k)|^\gamma \quad \text{otherwise} \end{aligned} \tag{3.14}$$

where α is the subtraction factor and β is the spectral floor factor. As in the case of Equation (3.11), the phase spectrum of the noisy speech is coupled with the cleaned magnitude spectrum. Hence a certain degree of distortion is to be accepted.

3.4.2 Spectral subtraction algorithm with full wave rectification

The extended spectral subtraction method was proposed by Berouti *et. al.* [9]. They used full-wave rectification with magnitude subtraction as a solution to the problem of narrow random spikes caused by over subtraction. The absolute value of the difference of the noise magnitude spectrum and noisy speech magnitude spectrum was taken as magnitude spectrum of clean speech and coupled with noisy phase to get clean speech [11]. The quality of enhanced speech was inferior, compared to that with the modified spectral subtraction.

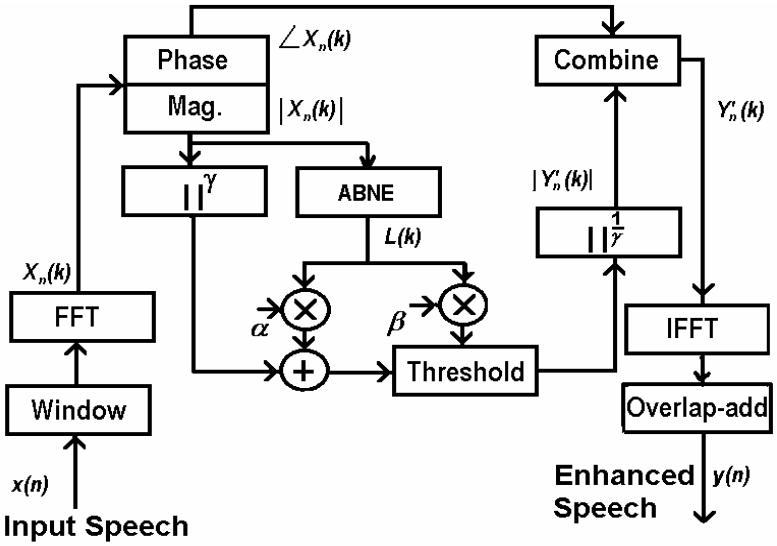


Fig. 3.3 Block diagram of modified spectral subtraction algorithm [9], based on averaging based noise estimation

3.4.3 Extended spectral subtraction algorithm

In the spectral subtraction technique described in Section 3.2, magnitude and phase spectra of noisy speech are computed, and magnitude spectrum of enhanced speech is computed by subtracting the estimated magnitude spectrum of noise. Enhanced speech is resynthesized by associating the phase spectrum of noisy speech with the enhanced magnitude spectrum. Gustafson *et al.* [16] have used a simple method which avoids calculation of the phase spectrum $\angle X_n(k)$, by modifying equation (3.11) to the following equation :

$$\begin{aligned} Y'_n(k) &= |Y'_n(k)|\, e^{j\angle Xn(k)} \\ &= |Y'_n(k)|\, X_n(k) / |X_n(k)| \end{aligned} \tag{3.15}$$

where $Y'_n(k)$ and $X_n(k)$ are Fourier transform of cleaned speech and noisy speech

respectively. By taking inverse Fourier transform of $Y'_n(k)$ we will get clean speech in time domain.

With this change, the algorithm becomes computationally efficient, as there are no explicit phase calculations involved and the errors due to round off of imaginary parts get eliminated. The algorithm requires only subtraction of magnitude spectrum of noise from the magnitude spectrum of noisy speech and multiplying the difference with original speech in frequency domain.

3.5 Enhancement of electrolaryngeal speech by modified spectral subtraction and extended spectral subtraction

Bhandarkar [10], [11] has implemented modified spectral subtraction algorithm for enhancement of the electrolaryngeal speech, with averaging based noise estimation.. A schematic of the modified spectral subtraction algorithm is shown in Fig. 3.3. Investigations were carried out for establishing the window size and optimal values of α, β, and γ using electrolarynx model NP-1. The recordings were done with the microphone positioned at the center between the mouth and the artificial larynx position. The total duration of the recording was 5 s, at a sampling rate of 11.025 kSa/s. During first 2 s, the speaker kept the lips closed, and the recorded speech contained only noise. The best results were obtained for N=244, α=2, β=0.001, γ=1. It was found that background noise was totally removed, but there was a small amount of musical noise present in the output.

Pratapwar [12], [13] further extended the investigations, and studied the effect of window length and position of the window with respect to excitation pulses of the vibrator. For window length of two pitch periods and overlap of 50%, it was found that shifting of the window position had no effect on the quality of enhanced speech output. This leads to an important conclusion that no specific effort is needed to identify the location of excitation impulse for positioning the window. Pratapwar also implemented extended spectral subtraction algorithm, with the schematic as shown in Fig. 3.4. The results were the same as those obtained with modified spectral subtraction.

3.6 Quantile based noise estimation (QBNE)

In the averaging based noise estimation (ABNE) for spectral subtraction [10], [11], the noise is assumed to be stationary. In reality, the background noise varies because of variations in the placement and orientation of the vibrator. This results in variations in the

effectiveness of noise enhancement over an extended period. Hence a continuous up-dating of the estimated noise spectrum is required. However, speech/silence detection in electrolaryngeal speech is rather difficult. Quantile based noise estimation (QBNE) [14], [17], [18], [19] technique does not need speech/non-speech classification and can be used for noise estimation in electrolaryngeal speech. QBNE makes use of the fact that even during speech periods, frequency bins tend not to be permanently occupied by speech i.e. tend not to exhibit high energy levels. Speech/non-speech boundaries are detected implicitly on a per-frequency bin basis, and noise spectrum estimates are updated throughout speech/non-speech periods. Fig. 3.5 shows block diagram of spectral subtraction with QBNE. The degraded signal was analyzed on a frame-by-frame basis, to obtain an array of the magnitude spectral values for each frequency sample, for a certain number of the past frames. Sorting of magnitude values in this array was used for obtaining a particular quantile value, which gave the best match with the ABNE-derived noise estimate.

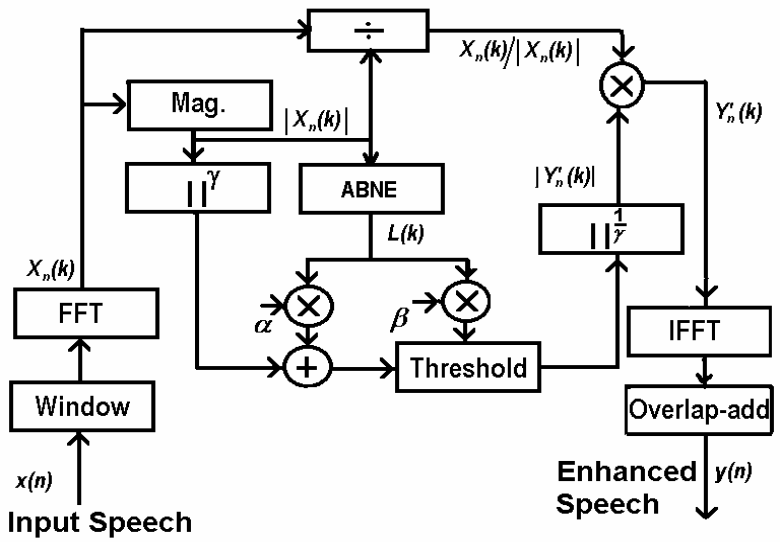


Fig. 3.4 Block diagram of extended spectral subtraction algorithm [12], based on averaging based noise estimation

Pratapwar [12], [13], [20] investigated the use of QBNE for continuous estimation of noise spectrum in electrolaryngeal speech. He carried out investigations involving different quantile estimates for finding an estimate of noise spectrum. The different methods used to make decision on selection of particular quantile value for each frequency sample were single quantile value, two quantile values, and frequency

dependent quantile values. In these methods, quantile values once selected remain constant during entire speech enhancement mode. The leakage noise characteristics change slowly with the application of the vibrator and the configuration of the vocal tract. Hence the spectral subtraction based on fixed quantile values was less effective during weak and non-speech segments. So, a dynamic selection of quantile values based on signal strength and frequency was investigated. It was found that, QBNE with SNR based quantiles showed a more consistent noise reduction and improved speech quality [12], [20].

A wavelet-based speech enhancement technique has been reported [21], [22]. A brief description of this technique is given in Appendix A.2. Its application to enhancement of electrolaryngeal speech requires further investigation.

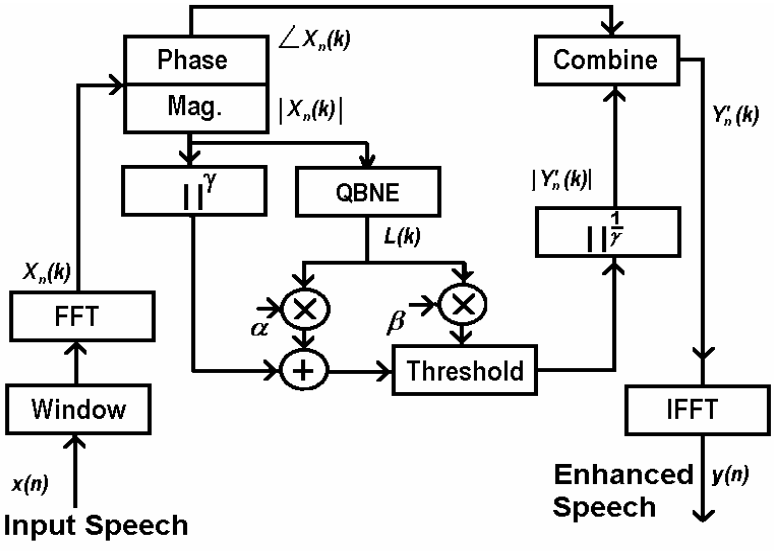


Fig. 3.5 Block diagram of spectral subtraction with quantile based noise estimation [12]

3.7 Phase reconstruction from magnitude spectrum

In the spectral subtraction algorithms implemented, phase spectrum of noisy speech is retained and coupled with the cleaned magnitude spectrum for obtaining each window segment. It is expected that quality can be better if phase spectrum is also noise-free. Towards this, vocal tract as well as leakage path may be modeled as minimum phase system, because of their passive nature. For a minimum phase system, the phase response

can be restored from its magnitude response. The resynthesis of the phase response from the magnitude spectrum can be done using the cepstral analysis [23], [24]. Two algorithms are studied in this regard, one is based on iterative technique, and the other is based on non-iterative technique.

3.7.1 Iterative technique

The iterative algorithm [25] for constructing phase from its magnitude is shown in Fig. 3.6. The function $v_k(n)$ represents the signal estimate on the k^{th} iteration and $\tilde{v}_{k+1}(n)$ is obtained, by imposing causality, by

$$\tilde{v}_{k+1}(n) = \begin{cases} v_k(n), n > 0 \\ v(0), n = 0 \\ 0, n < 0 \end{cases} \tag{3.16}$$

The function $M_v(\omega)$ is the known magnitude and $M_{k+1}(\omega)$ and $\theta_{k+1}(\omega)$ are the Fourier transform magnitude and phase of $\tilde{v}_{k+1}(n)$, respectively.

The algorithm begins with an initial guess $\theta_0(\omega)$ of the desired phase, and the inverse transform of $M_v(\omega)\exp[j\theta_0(\omega)]$ is taken. This step yields $v_0(n)$, the initial estimate of $v(n)$. Next, on the basis of the minimum phase condition, causality and the known value of $v(0)$ are imposed so that $v_0(n)$ is set to zero for $n < 0$ and set to $v(0)$ for $n = 0$, to obtain $\tilde{v}_1(n)$. The magnitude of the Fourier transform of $\tilde{v}_1(n)$ is then used to replace the previous spectrum and the procedure is repeated. The error function E_k has been defined as the mean-square difference between the known magnitude and the estimate $M_k(\omega)$ on each iteration, and it was shown that it is non-increasing [25].

The iterative algorithm has been used as an approach to computing the Hilbert transform, and also as a potential basis for phase unwrapping [25]. The algorithm has been reported to converge sometimes slowly (e.g., after several hundred iterations) and sometimes quickly (e.g., after a few iterations). Consequently, determining rates of convergence in terms of characteristics of the minimum phase signal and initial magnitude or phase estimates, and methods of speeding up convergence, need to be explored. The iterative algorithm has also been said to rely on exact knowledge of the magnitude, phase, and the initial value of the desired signal [25].

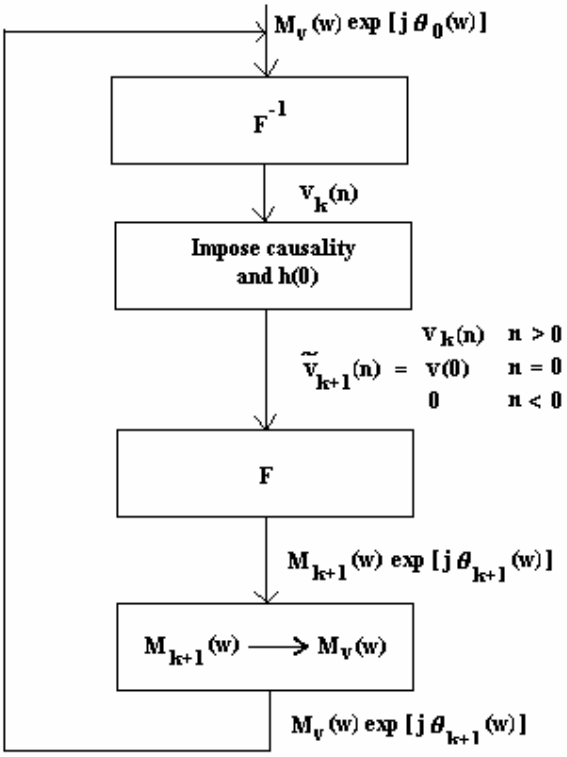


Fig. 3.6 Iterative algorithm to construct phase from magnitude [25].

3.7.2 Non-iterative technique

Non-iterative techniques for the construction of phase from the magnitude have been described by Rabiner and Schafer [1], and also by Yegnanarayana *et al.* [26]. Let $V(\omega)$ be the Fourier transform of a minimum phase sequence $v(n)$ of length N samples. $V(\omega)$ can be written as

$$V(\omega) = |V(\omega)| e^{j\theta_v(\omega)} \tag{3.17}$$

For a minimum phase signal $v(n)$, $\ln V(\omega)$ can be written as

$$\ln V(w) = \frac{c(0)}{2} + \sum_{n=1}^{\infty} c(n) e^{-j\omega n} \tag{3.18}$$

where $c(n)$ is the magnitude cepstrum . Now, since the magnitude spectrum is even and the phase spectrum is odd, using (3.17) and (3.18), we get

$$\ln |V(\omega)| = \frac{c(0)}{2} + \sum_{n=1}^{\infty} c(n) \cos \omega n \tag{3.19}$$

and

$$\theta_v(\omega) = -\sum_{n=1}^{\infty} c(n) \sin \omega n \tag{3.20}$$

Given $|V(\omega)|$, $c(n)$ can be obtained using (3.19), and it can be used for computing $\theta_v(\omega)$ using (3.20).

For DFT realization of the above algorithm, the continuous variable ω is replaced by the discrete variable *k,* to yield the following revised equations:

$$\ln|V(k)| = \frac{c(0)}{2} + \sum_{n=1}^{N-1} c(n) \cos(\frac{2\pi kn}{N}) \quad , k= 0, 1, \ldots, N\text{-}1 \tag{3.21}$$

and

$$\theta_v(k) = -\sum_{n=1}^{N-1} c(n) \sin(\frac{2\pi kn}{N}) \quad , k= 0, 1, \ldots, N\text{-}1 \tag{3.22}$$

Algorithms for signal reconstruction from the short time magnitude spectrum, have been successfully applied to the problems of time scale modification and noise reduction in speech processing [27]. Time-scale modification procedures aim at maintaining the perceptual quality of the original speech while changing the apparent rate of articulation. This is essentially equivalent to preserving the instantaneous frequency locations while changing their rate of change in time. Investigations [27] have shown that the time-scale modified speech, reconstructed from its magnitude spectrum, retained its natural quality and speaker-dependent features and was free from artifacts such as "burbles" and reverberation, though corrupted with a small amount of background "crackle".

In the case of spectral subtraction for noise reduction, a sequential iterative technique for the signal estimation from the cleaned STFT magnitudes has been reported [27]. A 128-point Hamming window with a window spacing of 64 points was used in the investigation. Speech sentences were corrupted by the addition of stationary white noise with a variety of signal-to-noise ratios between 0 dB and 20 dB. It was found that for signal-to-noise ratios above 10 dB, the signal estimates from the modified STFT magnitude had a reduced noise level while retaining their natural speech quality and speaker dependent features. The only processing artifact was the presence of short tone-bursts of varying frequency in the background.

3.8 Spectral subtraction with parameter adaptation based on auditory masking threshold

Recently, an investigation of electrolaryngeal speech enhancement using the frequency domain masking properties of the human auditory system has been reported [28]. The algorithm incorporates an auditory masking threshold (AMT) for parametric adaptation in spectral subtraction. Any noise component below the AMT will not be detectable by the human listener and so perceptually are not important components to suppress. The goal, then, is to minimize only the audible portion of the noise spectrum, and hence to avoid the possibility of musical noise, often associated with over-subtraction. Fig. 3.7 shows the block diagram of the algorithm. In the first stage, a minimum statistic based recursively smoothed noise estimate is used for spectral subtraction. The subtraction parameters are adapted by AMT using the frequency selectivity of the human ear. The AMT values in the low-frequency regions are higher than those in the high-frequency regions. Moreover, the AMT values of electrolaryngeal speech are lower than those of normal speech. If the AMT is low, the subtraction parameters are increased to reduce the noise. The introduced musical noise will be masked by the background noise remaining in the enhanced speech due to the high spectral floor. If the AMT is high, the subtraction parameters are kept to their minimal values because residual noise will stay below the AMT and will be naturally masked and inaudible. A supplementary AMT (SAMT) algorithm is also reported [28], in which a post-processing stage employs cross-correlation spectral subtraction (CCSS) to reduce the correlated noise present in the speech enhanced by the AMT algorithm.

Spectrographic and perceptual evaluation tests showed effective reduction of leakage noise and absence of musical noise, using both AMT and SAMT algorithms. They performed better than conventional power spectral subtraction algorithms especially in case of additive white and highly non-stationary babble noise. Moreover, the SAMT algorithm performed low frequency deficit compensation in addition to residual noise reduction.

3.9 Minimum statistics based noise estimation (MSBNE)

Noise power spectrum estimation is a fundamental component of speech enhancement and speech recognition systems. The robustness of such systems, particularly under low signal-to-noise ratio (SNR) conditions and non-stationary noise environments, is greatly

affected by the capability to reliably track fast variations in the statistics of the noise. Traditional noise estimation methods, which are based on voice activity detectors (VAD's), restrict the update of the estimate to periods of speech absence. Additionally, VAD's are generally difficult to tune and their reliability severely deteriorates for weak speech components and low input SNR [29]-[31]. Alternative techniques, based on histograms in the power spectral domain [32]-[34], are computationally expensive, require large memory resources, and do not perform well in low SNR conditions. Furthermore, the signal segments used for building the histograms are typically of several hundred milliseconds, and thus the update rate of the noise estimate is essentially moderate.

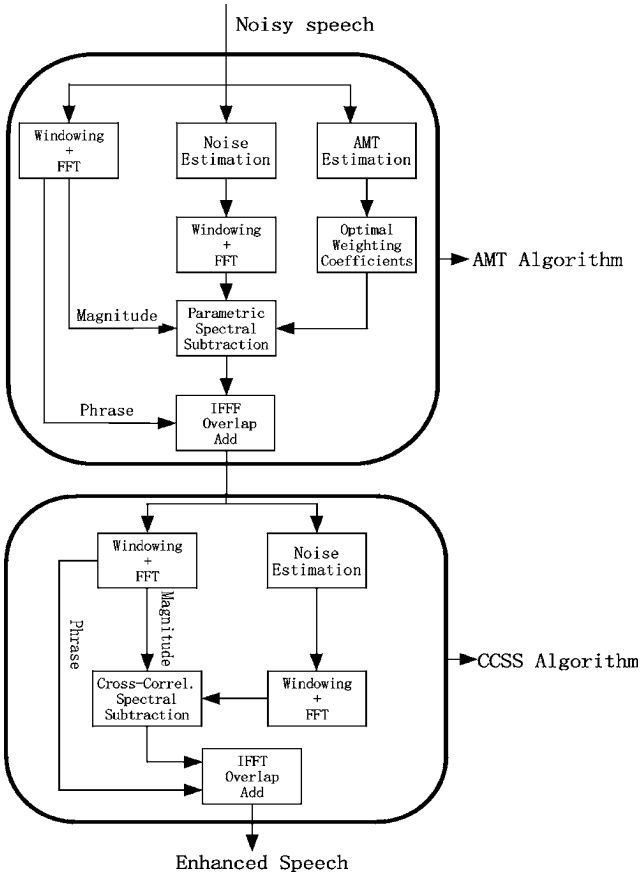


Fig. 3.7 AMT based speech enhancement scheme [28]

A computationally efficient algorithm capable of tracking non-stationary signals without requiring speech activity detection is Minimum Statistics (MS) [35], which tracks the minima values of a smoothed power estimate of the noisy signal. The algorithm is based on the observation that a short time subband power estimate of a noisy speech signal exhibits distinct peaks and valleys as shown in Fig. 3.8. While the peaks

correspond to speech activity, the valleys of the smoothed noise estimate can be used to obtain an estimate of subband noise power. To obtain reliable noise power estimates, the data window for the minimum search must be large enough to bridge any peak of speech activity. Also, since the minimum is biased towards lower values, unbiased estimate is obtained by multiplying with a bias factor. A time-frequency dependent smoothing factor and a bias factor derived from the statistics of the local minimum have been used in the implementations reported [36]. Several variants of MS based algorithms, such as minima controlled recursive averaging (MCRA) and improved MCRA (IMCRA) have been proposed [37]-[40]. They are briefly described in Appendix A.3. All the MS based algorithms have been found to be of high computational efficiency, and very suitable for dynamic noise estimation in adverse environments involving non-stationary noise, weak speech components, and low input signal-to-noise ratio.

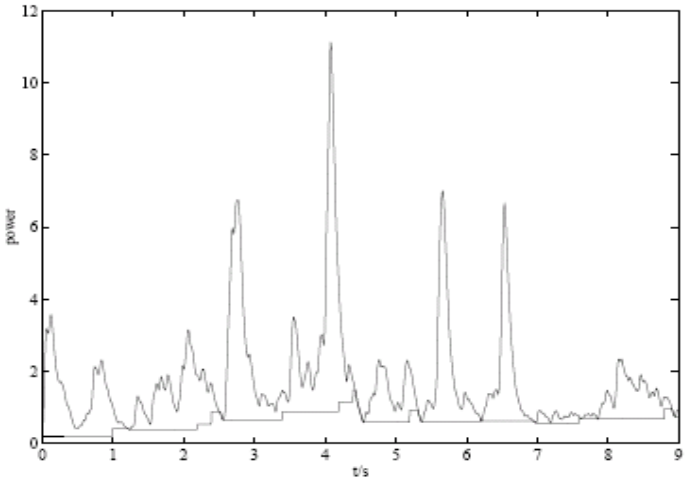


Fig. 3.8 Short time subband power and estimated noise floor of noisy speech signal (f_s = 8kHz, FFT size = 256, subband k=8) [35]

In electrolaryngeal speech enhancement, an averaging based noise estimation can be carried out from the noise alone segments with closed lips. But the characteristics of leakage noise vary with speech production, and the pressure and orientation of holding the vibrator. In this project, a dynamic noise estimation based on minimum statistics is implemented. Investigations and results of background noise reduction of electrolaryngeal speech using smoothed minimum statistic technique are described in Sec. 5.2.5.

Chapter 4

REAL-TIME IMPLEMENTATION ON A DSP BOARD

It was decided to implement the noise reduction of electrolaryngeal speech as a real time embedded system application using a digital signal processor (DSP) board. While choosing a DSP, one has to consider several factors such as performance of the processor, sampling frequency requirements, availability of development tools such as library functions, power consumptions, size, weight and cost requirements etc. [42]. This chapter reviews the earlier work done by Budiredla [15], and then presents the real time implementation details of this project.

4.1 Earlier work

Budiredla [15] has implemented spectral subtraction using QBNE based on SNR, on a DSP starter kit based on TMS320C6211 digital signal processor from Texas Instruments(TI).The software environment used was Code Composer Studio, which is an easy-to-use Graphical User Interface(GUI) for configuring, building, interfacing and debugging purposes.

The DSP starter kit provides AD535 codec (with in-built ADC and DAC) as an interface to analog voice signals. ADC samples the input at a fixed sampling rate of 8 kSa/s, and makes it available to the Multi-channel Buffered Serial Port (McBSP) as 16-bit word. McBSP is used to communicate between the coder-decoder (CODEC) and the Enhanced Direct Memory Access (EDMA). This peripheral helps in performing serial-to-parallel and parallel-to-serial arrangement of the data points. McBSP reads data from ADC, organizes them in 16 bits/frame and then sends them to the memory location specified by EDMA. By setting proper registers, McBSP gives interrupts to EDMA whenever data are available for reading by EDMA, and also whenever it is ready to accept data from EDMA. EDMA controller transfers data between regions in the memory map without intervention by the CPU. It allows movement of data to and from internal memory, internal peripherals, or external devices to occur in the background of CPU operation. In this application, EDMA was used to accept input data from McBSP receiver

register and store it in a pre-determined memory location. As soon as a block of data is transferred by EDMA to the memory location specified by the CPU of the TMS320C6211 DSP processor, an interrupt is activated. The CPU then reads the input data block, processes it for the removal of background noise, and finally places it at the transmitting location. EDMA outputs data from this memory location to the McBSP transmit register, which in turn transfers the data to the CODEC. The DAC takes 16-bit word samples from the McBSP at a fixed sampling rate of 8 kSa/s, reconstructs it into an analog signal and outputs it to the speaker.

Based on earlier work by Pratapwar [12], the following parameters were used for spectral subtraction: spectral subtraction factor α = 2, spectral floor factor β =0.001, exponent factor γ= 1. Signals were acquired at a sampling rate of 11.025 kSa/s with the ADC of PC sound card, and were processed at this rate with MATLAB and C-based implementations. For real time implementation, signals were output by the DAC of the PC sound card and given to the ADC input of the DSP board. Signal acquisition and processing on the DSP board had to be carried out at the fixed rate of 8 kSa/s. The processed output of the DSP board was acquired by PC sound card for playback and analysis. Processing speed limitations of the DSP used restricted the noise estimation to 8 windows, though earlier simulations had revealed that optimal noise reduction was obtained with approximately 55 windows.

All the three implementations – MATLAB, C and DSP – showed noise reduction. However, C implementation was found to be less effective than MATLAB. Actually, in C implementation, the code was written considering the on-chip memory resources, since using external memory limits DSP processor speed. So, in the C code, all variables were declared with lower precision data types. This probably resulted in underflows during subtraction, reducing the effectiveness of the algorithm. Also, one of the most serious drawbacks of the implementation carried out by Budiredla is that there was no phase reconstruction done in the process. Only the magnitudes were passed out after spectral subtraction, without combining them even with the phase of the noisy input speech.

4.2 Platform for real time implementation

ADSP-BF533 Blackfin processor from Analog Devices was chosen for this project. Analog Devices offers this processor in a kit called EZ-Kit Lite. The package contains an evaluation copy of Visual DSP++, which includes C/C++ compiler/debugger. Information

on ADSP processors is given in the processor's data sheet [43]. Features of ADSP-BF533 EZ Kit Lite and the Visual DSP++ development environment are discussed in this section.

The ADSP-BF533 EZ-KIT Lite provides a method for initial evaluation of the ADSP-BF533 Blackfin processor for a wide range of applications including audio and video processing. The EZ-KIT Lite includes an ADSP-BF533 desktop evaluation board and fundamental debugging software to facilitate architecture evaluations via a USB-based PC-hosted tool set. Real-time debugging is made possible via the background telemetry channel (BTC) feature. Through BTC, data can be streamed both to and from the processor over the JTAG connection between host and embedded processor without the overhead involved with halting the target application, getting the desired information, and then restarting the target application. The ADSP-BF533 EZ-KIT Lite provides an evaluation suite of the VisualDSP++ integrated development and debugging environment (IDDE) whose features are briefly discussed in the next sub-section.

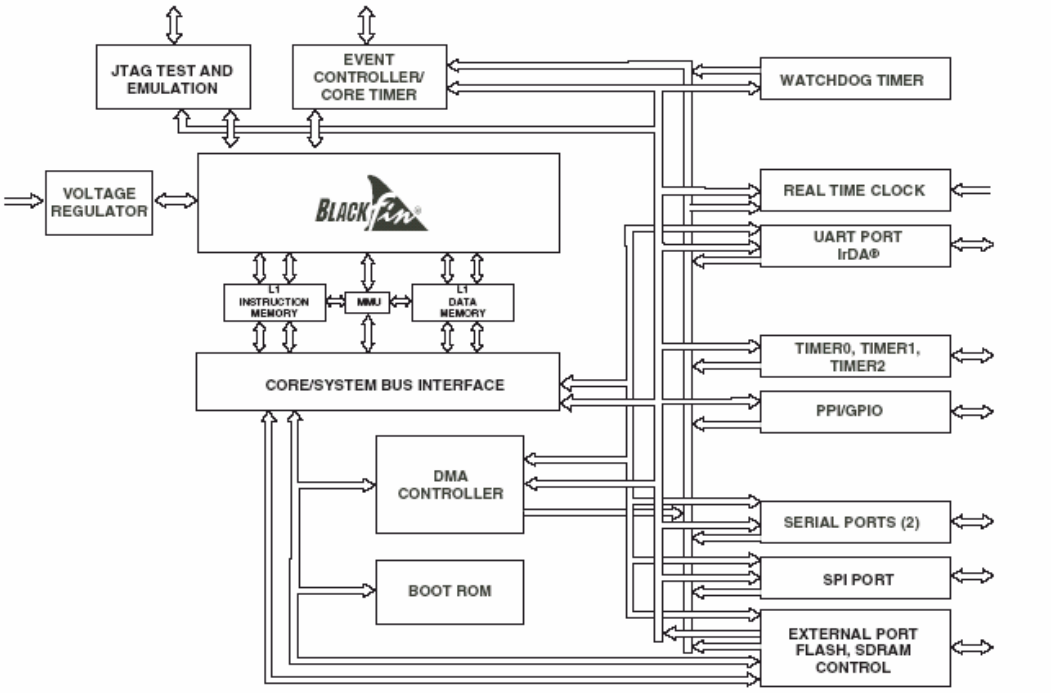


Fig. 4.1 Block diagram of ADSP BF 533 processor [43]

The ADSP-BF533, ADSP-BF532, and ADSP-BF531 processors are enhanced members of the Blackfin processor family that are completely pin compatible, differing only in their performance and on-chip memory. The Blackfin processor core architecture combines a dual MAC signal processing engine, an orthogonal RISC-like microprocessor

instruction set, flexible single instruction multiple data (SIMD) capabilities, and multimedia features into a single instruction set architecture. Blackfin products feature dynamic power management. The ability to vary both the voltage and frequency of operation optimizes the power consumption profile to the specific task The processor system peripherals include parallel peripheral interface (PPI), serial ports (SPORTs), serial peripheral interface (SPI), general-purpose timers, universal asynchronous receiver transmitter (UART), real-time clock (RTC), watchdog timer, and general-purpose I/O (programmable flags). These peripherals are connected to the core via several high bandwidth buses, as shown in Figure 4.1.

The block diagram of AD1836A codec is shown in Figure 4.2. It is a single-chip codec with three stereo DACs and two stereo ADCs using multibit Σ-Δ architecture. A serial peripheral interface (SPI) port is provided to permit adjustment of volume and some other parameters. The AD1836A operates from a 5 V supply, with provision for a separate output supply to interface with low voltage external circuitry. There are four ADC channels in the AD1836A configured as two independent stereo pairs. One stereo pair is the primary ADC with differential inputs. The second pair can be programmed via SPI ADC control register 3 to operate in one of three possible input modes. The ADC section may also operate at a sample rate of 96 kHz with only the two primary channels active. The ADCs include an on-board digital decimation filter with 120 dB stop-band attenuation and linear phase response, operating at an over-sampling ratio of 128 (for 4-channel 48 kHz operation) or 64 (for 2-channel 96 kHz operation). The primary ADC pair should be driven from a differential signal source for best performance. The secondary input pair can operate in one of three modes: (1) Direct differential inputs, (2) PGA mode with differential inputs, where the PGA amplifier can be programmed using the SPI port to give an input gain of 0 dB to 12 dB in steps of 3 dB, and (3) Single-ended MUX/PGA mode, where two single-ended stereo inputs are provided that can be selected using the SPI port and input gain can be programmed same as PGA mode.

The AD1836A has six DAC channels arranged as three independent stereo pairs, with six fully differential analog outputs for improved noise and distortion performance. Each channel has its own independently programmable attenuator, adjustable in 1024 linear steps. ADCs and DACs of AD 1836A can be programmed to set the resolution of 24, 20, or 16 bits. The codec can be programmed using SPI to use it in either I2S or TDM mode. In I2S mode, two stereo inputs and two stereo outputs can be used to process the signals at sampling frequency of 96 kHz or 48 kHz. In TDM mode, simultaneous

processing of two stereo inputs and three stereo outputs can be done at sampling frequency of 48 kHz. ADC and DAC control registers can be programmed to set the sampling rate, resolution, gain of PGA and attenuation.

The AD1836A has six DAC channels arranged as three independent stereo pairs, with six fully differential analog outputs for improved noise and distortion performance. Each channel has its own independently programmable attenuator, adjustable in 1024 linear steps. ADCs and DACs of AD 1836A can be programmed to set the resolution of 24, 20, or 16 bits. The codec can be programmed using SPI to use it in either I2S or TDM mode. In I2S mode, two stereo inputs and two stereo outputs can be used to process the signals at sampling frequency of 96 kHz or 48 kHz. In TDM mode, simultaneous processing of two stereo inputs and three stereo outputs can be done at sampling frequency of 48 kHz. ADC and DAC control registers can be programmed to set the sampling rate, resolution, gain of PGA and attenuation.

VisualDSP++ IDDE enables programmers to move between editing, building and debugging within a single interface. Key features of VisualDSP++ [44] include the C/C++ compiler, plotting tools, statistical profiling, assembler, linker, libraries, simulator, and emulator support. It assesses the inner working of the target processor, and hence offers features for run, step execution and halt of the program, settings for breakpoints and watchpoints, viewing the state of the processor's memory, registers, cycle count and stack, performing trace, cycle-by-cycle pipeline viewing.

4.3 Software development for real time implementation

The programming for most DSPs can be done either in assembly or in C. In this project, the implementation has been done using C language. The software tools for converting these C source files into code executable on DSP include compiler, assembler, and linker. The C compiler accepts C source code and produces ADSP BF-533 assembly language source code. The assembler translates assembly language source files into machine language object files. The linker combines object files into a single executable object module.

4.3.1 Input-output interfacing

In fixed point implementation, all the arithmetic operations should be properly scaled or normalized to avoid overflows. If we add N fixed-point numbers, each of m-bits, the resulting number will be a maximum of $n = m + \log_2 N$ bit number. If the processor has

register length greater than *n,* there will be no overflow during addition itself. However, the result needs to be scaled before subsequent operation. Assuming the processor to be a *k*-bit processor, the addition should be right shifted by (*n-k)* bits. Similarly, when multiplying two *n*-bit numbers the resulting number will be a 2*n* bit number. To make the resulting number *n*-bit, we have to right shift the result by *n* bits. It is to be noted that this implementation, concerned primarily with avoiding overflows, may at times result in significant number of underflows.

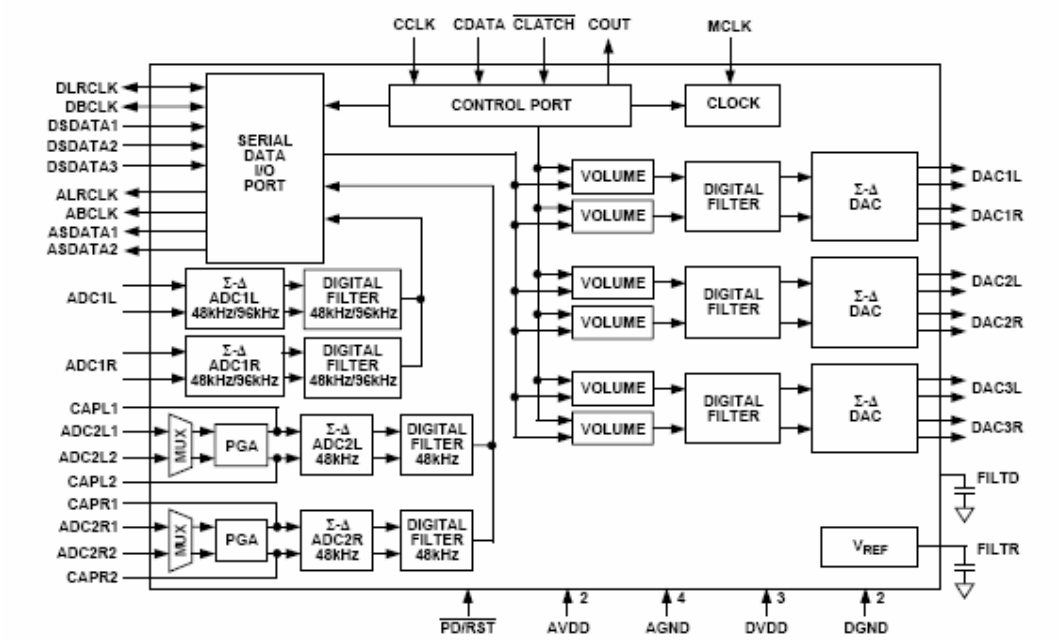


Fig. 4.2 Block diagram of AD1836A [44]

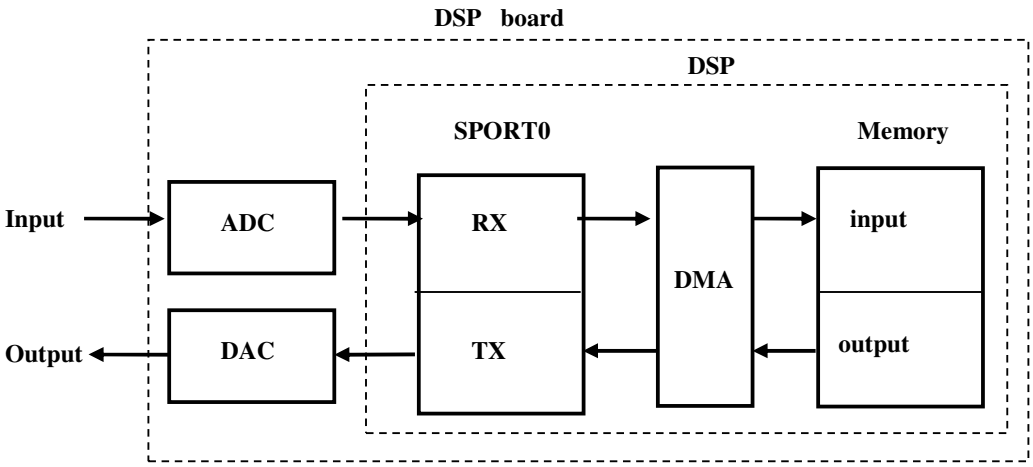


Fig. 4.3 Functional block diagram of I/O signals with SPORT, DMA, and memory

Fig. 4.3 shows the A/D and D/A conversion and transfer of signal data using ADC, DAC, Serial Peripheral Interconnect (SPI), serial port and Direct Memory Access (DMA). First, SPI is configured to initialize the codec to set the parameters like number of input and output channels, sampling rate, resolution of ADC and DAC, PGA gain, and attenuation factor for output. After initialization of the codec, SPI is disabled. The processor uses DMA to transfer data within memory spaces or between a memory space and a peripheral. The processor can specify data transfer operations and return to normal processing while the fully integrated DMA controller carries out the data transfers independent of processor activity. DMA interrupts the processor after completion of transfer of data (a full window of, say, 256 samples) from serial port to memory, while the processor is busy in processing the signal. In the interrupt service routine, the acquired data from the ADC are processed and output sent back to DAC. The example program given along with the Visual DSP++ software from Analog Devices, known as 'talkthrough', was modified for input and output data transfer. Initially the frame synchronization of the data samples was not proper; hence the data got swapped between left and right channels of the two ADC buffers. Numbering the frames in a sequential order from right channel to left solved the problem.

Initially, the following work was done towards real time implementation of the spectral subtraction algorithms. First, pointwise echo program was implemented. The input and output buffer size was 1 sample each. The 24-bit output from the ADC was right-shifted by 8 bits to fit the 16-bit DSP registers. The final processed data were again left-shifted by 8 bits before passing to the 24-bit DAC. The program operation was verified by giving signal input from function generator as well by playing back pre-recorded sound file from PC sound card. The output was observed on analog and digital oscilloscopes for time domain analysis, as well as spectrum analyser for frequency domain investigations.

Next, using the same unity sized buffers, echo program using block processing was attempted. Discontinuity in the output waveform could be observed for block sizes exceeding 128 samples. Also, blockwise FFT and inverse FFT operations were carried out to check whether the input could be recovered at the output.

4.3.2 Implementation of spectral subtraction with ABNE

The Blackfin DSP library has some in-built routines for computation of FFT and inverse FFT, viz, “rfft_fr16” and “ifft_fr16”. The arguments for these functions must be of the

data-type “fract16”, which represents a single 16-bit signed fractional value. The fract16 data representation is shown in Fig. 4.4. To find the value of a fract16 number, all the bit-weights for which the bit is set are to be added. Therefore, to represent 0.25 in fract16, the HEX representation would be 0x2000 (2^{-2}). For (-1), the HEX representation in fract16 would be 0x8000 (-1). Fract16 cannot represent (+1) exactly, but it gets quite close, with 0x7fff.

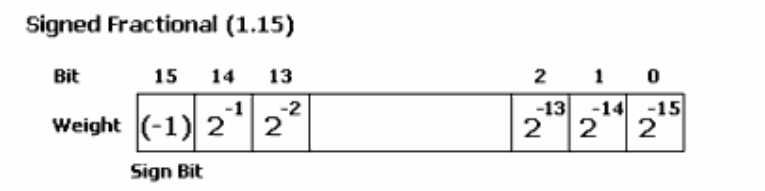


Fig. 4.4 Data representation of fract16 [43]

The “rfft_fr16” function transforms the time domain real input signal sequence to the frequency domain by using the radix-2 FFT. The function takes advantage of the fact that the imaginary part of the input equals zero, which in turn eliminates half of the multiplications in the butterfly. The size of the input array, the output array, and the temporary working buffer is *n*, where *n* represents the number of points in the FFT. Memory bank collisions, which have an adverse effect on run-time performance, may be avoided by allocating all input and working buffers to different memory banks. If the input data can be overwritten, the optimum memory usage can be achieved by also specifying the input array as the output array. To avoid overflow, the function performs static scaling by dividing the input by 1/*n*.

The “iffft_fr16” function transforms the frequency domain complex input signal sequence to the time domain by using the radix-2 Fast Fourier Transform. To avoid overflow, the function scales the output by 1/*n*. So, the output of this function is to be multiplied by *n* before passing it to the DAC. The function “twidfftrad2_fr16” is used to initialize the array of twiddle factors. For all the above functions, the input sequence length must be a power of 2 and at least 16.

In the FFT-IFFT program, distortions were observed for block sizes exceeding 32. This problem arose because Direct Memory Access (DMA) interrupts were generated on receiving every sample. The entire processing was attempted to be done in the time between receiving two successive samples, while in the remaining cycles, the program just remained idle so far as actual processing is concerned.

To overcome the above problem, the facility of "continuous transfers using autobuffering" offered by the BF533 processor [43] was investigated. If a peripheral's DMA data consists of a steady, periodic stream of signal data, DMA autobuffering (FLOW = 1) may be an effective option. Here, DMA is transferred from or to a memory buffer with a circular addressing scheme, using either one- or two-dimensional indexing with zero processor and DMA overhead for looping. Synchronization options include 1D, interrupt-driven, where the software is interrupted at the conclusion of each buffer. The critical design consideration is that the software must deal with the first items in the buffer before the next DMA transfer, which might overwrite or re-read the first buffer location before it is processed by software. This scheme may be workable if the system design guarantees that the data repeat period is longer than the interrupt latency under all circumstances.

The next option, and probably the most viable one, is 2-D, interrupt-driven DMA(double buffering), where the DMA buffer is partitioned into two or more sub-buffers, and interrupts are selected (set DI_SEL = 1 in DMA_CONFIG) to be signaled at the completion of each DMA inner loop. In this way, a traditional double buffer or "ping-pong" scheme could be implemented. For example, two 128-word sub-buffers inside a 1K word buffer could be used to receive 16-bit peripheral data with these settings:

START_ADDR = buffer base address
DMA_CONFIG = 0x10D7 (FLOW = 1, DI_EN = 1, DI_SEL = 1, DMA2D = 1, WDSIZE = 01, WNR = 1, DMA_EN = 1)
X_COUNT = 128
X_MODIFY = 2 for 16-bit data
Y_COUNT = 2 for two sub-buffers
Y_MODIFY = 2, same as X_MODIFY for contiguous sub-buffers

In 2-D, polled synchronization, if interrupt overhead is unacceptable but the loose synchronization of address/count register polling is acceptable, a 2-D multibuffer synchronization scheme may be used. For example, assume receive data needs to be processed in packets of sixteen 32-bit elements. A four-part 2-D DMA buffer can be allocated where each of the four sub-buffers can hold one packet with these settings:

START_ADDR = buffer base address
DMA_CONFIG = 0x101B (FLOW = 1, DI_EN = 0, DMA2D = 1, WDSIZE = 10, WNR = 1, DMA_EN = 1)
X_COUNT = 16
X_MODIFY = 4 for 32-bit data

Y_COUNT = 4 for four sub-buffers
Y_MODIFY = 4, same as X_MODIFY for contiguous sub-buffers

The synchronization core might read Y_COUNT to determine which sub-buffer is currently being transferred, and then allow one full sub-buffer to account for pipelining. For example, if a read of Y_COUNT shows a value of 3, then the software should assume that sub-buffer 3 is being transferred, but some portion of sub-buffer 2 may not yet be received. The software could, however, safely proceed with processing sub-buffers 1 or 0.

Following the second strategy above, a ping-pong buffer scheme was implemented. It was found that the maximum buffer size in the talkthrough program permitted by the linker and guided by memory constraints was 256 samples. An input buffer of size 256 samples was created, and it was divided into two sub-buffers of size 128 samples each. A two-dimensional DMA scheme was implemented. That is, as soon as the first half of the input buffer was filled (when 128 samples were received), an interrupt was generated, and these 128 samples were transferred to the processing buffer. There, it was combined with a given number of samples from the previously processed block (depending on the window overlap desired), and processing was done. In the meantime, input samples continued to be received in the second half of the input buffer. Thus, more time was available for processing. On the output side, overlap-discard method was implemented to properly combine the processed samples into the output stream, i.e. only the last 128 of the 256 processed values were sent to the DAC, while the first 128 samples were discarded. An alternative technique could have been overlap-add method, where the output samples corresponding to the input overlap region are retained for addition, and subsequent division by a factor which is determined by the percentage of overlap (2 in our case).

Programs were written to perform scaling of the input samples, time delay of the output with respect to the input, and change in the processing algorithm from one type to another some time after beginning (which is necessary to switch from noise estimation mode to spectral subtraction mode). All of them gave satisfactory results.

Next, programs were written which performed windowing on the input signal in the time domain. Rectangular and Bartlett windows were used. Also, FIR filtering was done in the frequency domain. Certain discontinuities could be observed in the output on the oscilloscope, though there was no significant degradation in audio quality while hearing. Also, code was written to perform ABNE-based spectral subtraction, but the residual noise after subtraction was above acceptable levels, and other frequencies were

getting distorted. It was inferred that a main cause of this problem was the high sampling rate of the ADC. The ADSP BF 533 EZKIT Lite Board uses AD 1836 codec, which operates at a fixed sampling rate of 48 kHz or 96 kHz. But to do any spectral processing on speech, the processing window should encompass at least twice the pitch period. For this, either the window size should be made larger, or the sampling rate has to be reduced. But the buffer size cannot be increased from 256 to 1024. Hence, the only option was to reduce the sampling frequency from 48 kHz to 12 kHz, which cannot be done by reconfiguring the codec in the present kit. One solution is to go for a programmable codec with our desired low sampling rate. Two such codecs are AD 73322 from Analog Devices and AIC 23 from Texas Instruments, both of whose sampling rates can be programmed in multiples of 8 kHz, starting from 8 khZ upto 48 kHz. The other solution is to modify the software to perform decimation in time. Here, every 4^{th} sample from the input data stream is retained, while the remaining three are discarded. Conversely at the output, a zero order sample and hold is implemented. Every sample is copied three more times before sending to the DAC, i.e. 4 consecutive samples have the same value. The two principal issues regarding real time implementation are kept in mind while doing the above. Firstly, continuity should be maintained in the data stream i.e. there should be no loss of data samples. Secondly, the processing delay between the input and output data should be acceptable for real time processing. Software decimation by a factor of 4 was done in real time, bringing down the effective sampling frequency to 12 kHz.

In the implementation of spectral subtraction, the built-in routines “add_fr1x16” and “sub_fr1x16” were used for 16-bit addition and 16-bit subtraction respectively of the two input fract16 arguments. The built-in function “multr_fr1x16” performs 16-bit fractional multiplication of the two input parameters, and the result is rounded to 16 bits. In case of squaring of spectral magnitudes (when $\gamma = 2$), the above routine gives correct results only if the input magnitude is in the range 0x0090 to 0xff70, outside which, result of squaring was found to be zero. The built-in function “cabs_fr16” computes the absolute value of the input complex argument. “arg_fr16” is the library function that computes the phase associated with a Cartesian number, represented by the complex input argument, and returns the result, scaled by 2π, i.e. in the range -1 to +1.

In this implementation, the processed magnitude is combined with the retained phase using the library routine “polar_fr16”. This function transforms the polar coordinate, specified by the arguments magnitude and phase, into a Cartesian coordinate

and returns the result as a complex number in which the x-axis is represented by the real part, and the y-axis by the imaginary part. The phase argument is interpreted as radians. For the "polar_fr16" function, the phase must be scaled by 2π and must be in the range [0x8000, 0x7ff0]. The value of the phase may be either positive or negative. Also, the domain of the magnitude values is [-1.0,1.0).

In the case of extended spectral subtraction using ABNE, explicit phase calculation using "arg_fr16" function is not done. Instead, firstly, the complex numbers resulting from the FFT of the input sequence are divided by their respective magnitudes using the "cdiv_fr16" function. Then the results of the above operation are multiplied by the real-valued magnitudes obtained from spectral subtraction, using the "cmlt_fr16" function.

In the implementation of ABNE, different numbers of windows were used for obtaining the noise estimate. The schemes used for averaging were different in different cases. When the number of windows was as low as 10, no scaling was necessary. Spectral subtraction using the most recent average was performed from the very first window, and updating of average stopped after the first 10 windows. This scheme failed due to addition related overflows when 100 windows were used for estimating noise i.e. when 100 magnitudes for each frequency bin were added without any scaling and the final sum was divided by 100 in the end. To solve this, while summing the spectral magnitudes of successive windows, each magnitude value was divided by 10, and when summation over 100 windows was complete, the total sum was again divided by 10, to give the average noise estimate. The program outputs the unprocessed input during these first 100 windows, after which it starts spectral subtraction using the final estimate obtained. Hence the first 2 s of sound should not have speech, i.e. the user should keep the lips closed.

4.3.3 Implementation of spectral subtraction with dynamic noise estimation

Dynamic noise estimation has earlier been investigated using QBNE. Because of constraints of memory and processing speeds, dynamic noise estimation using QBNE with 100 or more windows was found to be infeasible on this DSP board. Hence efforts were concentrated on the implementation of MSBNE, which is computationally much faster.

In the real time implementation of dynamic smoothed MSBNE, memory constraints of the DSP limited the number of windows for minima calculation to 10. For a sampling rate of 12 kHz, and window size of 256 samples with 50% overlap, this meant that the minima were calculated over a time duration of 0.1 s. Hence, any speech activity should not continue for more than 0.1 s at a stretch, which is a severely restricting condition.

To overcome the above problem, decimation in time was done for minima calculation, which is a reasonable technique here since the leakage noise characteristics of the electrolarynx change slowly. To allow continuous speech activity over 1 s, a decimation factor of 10 was used. Two schemes were implemented – MSBNE, and MSBNE combined with averaging. In the MSBNE scheme, every 10^{th} frame received was retained for calculating the minima. A circular buffer scheme was used for efficient handling of the values stored for minima calculation. A pointer was maintained which pointed to that location of the circular buffer where the new value was to be stored, over-writing the oldest value. Starting from 0, this pointer was incremented by one every 10^{th} frame, and re-initialized to 0 on reaching a value of 9. In the scheme involving averaging, calculation of minima was preceded by averaging and decimation, i.e. averaging was done over 10 consecutive windows, and minima of such averages of 10 successive windows were calculated. Thus 100 windows contribute to the minima estimation, but the estimate involves minima of local averages. The bias factor or the over-subtraction factor α for optimal spectral subtraction is likely to be different from the value for MSBNE.

In the above process, the minima estimation is done once every 10 frames, resulting in uneven distribution of processing. An alternative technique to uniformly distribute the processing load can be devised by minima calculation as a cascade of two minima estimations. Minima of every 10 successive windows can be calculated and stored in a buffer, and when this buffer gets filled with 10 such minima, the minimum of these local minima can be calculated to obtain an effective minimum over a much larger range. This technique does not discard 9 out of every 10 windows, as in the individual decimated MSBNE, and so may be expected to be more robust. This needs to be implemented and investigated.

The built-in function “vecmin_fr16” was used to calculate the minima, taking the vector and the number of elements in it as inputs. The function assumes that input array

arguments are constant, i.e. their contents will not change during the course of the routine. In particular, this means the input arguments do not overlap with any output argument. In general, better run-time performance is achieved by any of the vector functions if the input vectors and the output vector are in different memory banks. This structure avoids any potential memory bank collisions.

4.3.4 Final implementation

To summarize the real time scheme finally implemented, a 2-D DMA technique was used, with buffers of size 256 samples. Effective sampling rate was 12 kHz, obtained by software decimation. Analysis windows were 256 samples long, with 50% overlap. The number of windows used for obtaining the average noise estimate was varied from 10 to 100, with appropriate scaling technique. Finally we have three programs for "Enhancement of ElectroLaryngeal Speech", namely, **"eels_abne"**, **"eels_msbne"** and **"eels_avmsbne"**. All the programs include spectral compensation, but not phase enhancement.

In **"eels_abne"**, noise estimate calculated during initial 2 s of non-speech is used for spectral subtraction for the remaining duration, There is no dynamic noise estimation involved here. Optimum set of parameters determined from the offline implementation of spectral subtraction with ABNE was used, namely: over-subtraction factor $\alpha = 2$, spectral floor factor $\beta = 0.001$ and exponent factor γ=1.

In **"eels_msbne"**, dynamic noise estimation was used, where minima of the spectral magnitudes of last 10 frames was used as the noise estimate in spectral subtraction. Decimation in time was done for memory constraints. 9 out of 10 frames were dropped, and every 10^{th} frame was retained for minima calculation. Values of the parameters used were: α = 10 (instead of 20, since any higher value of α resulted in severe signal attenuation, probably due to finite precision arithmetic), $\beta = 0.001$ and γ=1.

In **"eels_avmsbne"**, averaging was done over 10 consecutive windows, and minima of such averages of 10 successive windows were calculated. Values of the parameters used were: $\alpha = 6$, $\beta = 0.001$ and γ=1.

Chapter 5

RESULTS AND DISCUSSIONS

Several recorded natural and electrolaryngeal speech utterances were used for the testing of the offline and real time implementations of the various algorithms. Speech recordings for sustained vowels and sentences were made using three models of electrolarynx, viz. Servox, Solatone and NP Voice, with pitch set to 75 Hz, 115 Hz and 94 Hz respectively. The total duration of each recording was approx. 14 s. The first 2 s in each recording corresponded to the non-speech interval. During this interval, the speaker kept his lips closed to prevent any speech from the mouth, and the recorded speech contained only noise. The following 12 seconds contained speech, during which the speaker made the above utterances in the required fashion, with intermittent silence in between the different utterances.

The signals were acquired at a sampling rate of 11.025 kHz with the ADC of PC sound card and were processed at this rate with the MATLAB-based implementation. For real-time implementation the signals were output by the DAC of the PC sound card and given to the ADC input of the DSP board. The signals were acquired by the DSP board at 48 kHz, but down-sampled to 12 kHz for actual processing. The processed analog output of the DSP board was acquired by PC sound card at a sampling rate of 11.025 kHz for playback, and analysis.

5.1 Investigations of low frequency deficit

To investigate the low frequency deficit of the electrolaryngeal speech as reported in different literature, recordings were done on five speakers using three different models of electrolarynx available in our laboratory, viz. Servox, Solatone and NP Voice, with pitch settings 75 Hz, 115 Hz, and 94 Hz respectively. The speakers were asked to utter */a/, /i/, /u/, /aie/,* the Hindi sentence *"aapkaa naam kya hai?",* and *"where were you a year ago?"*. Acquisition was done using Goldwave software, spectral analysis and LPC

analysis were carried out in MATLAB, while spectrographic analysis was done using PRAAT.

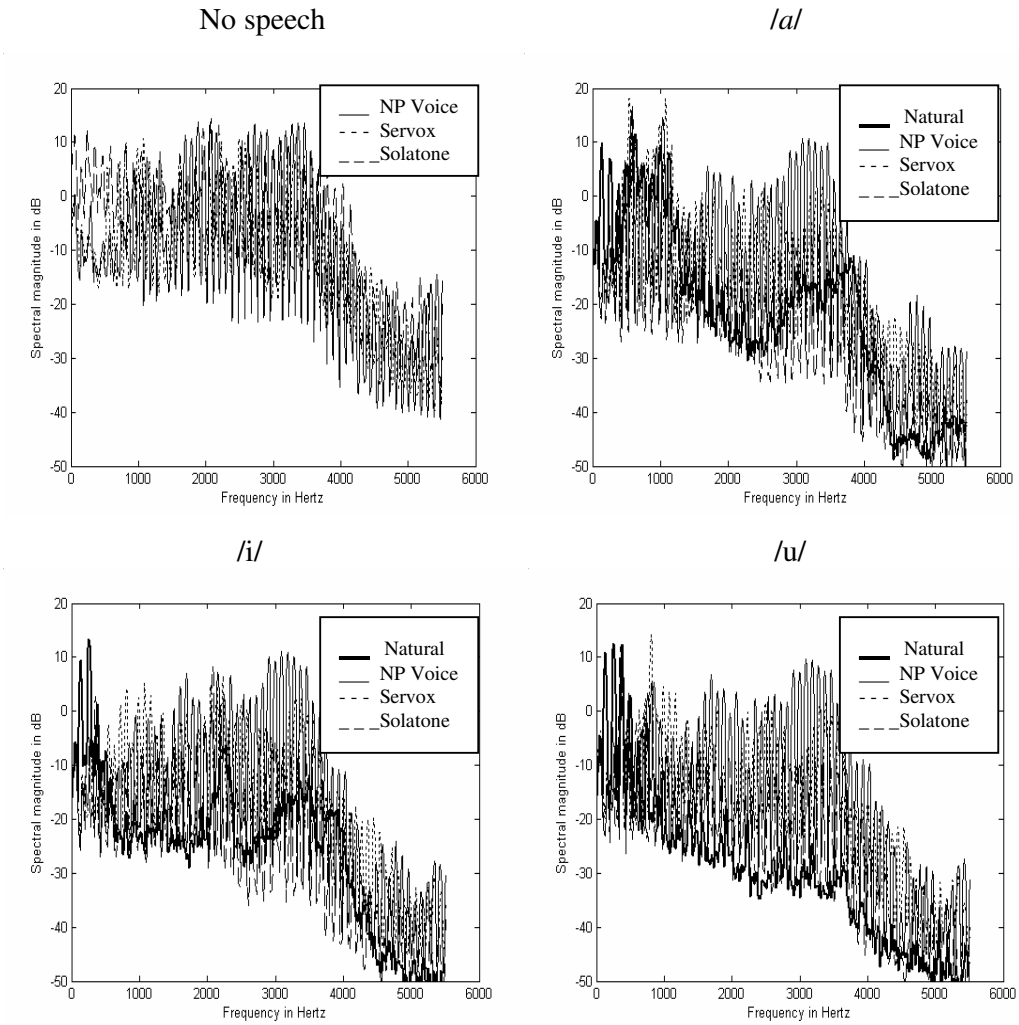
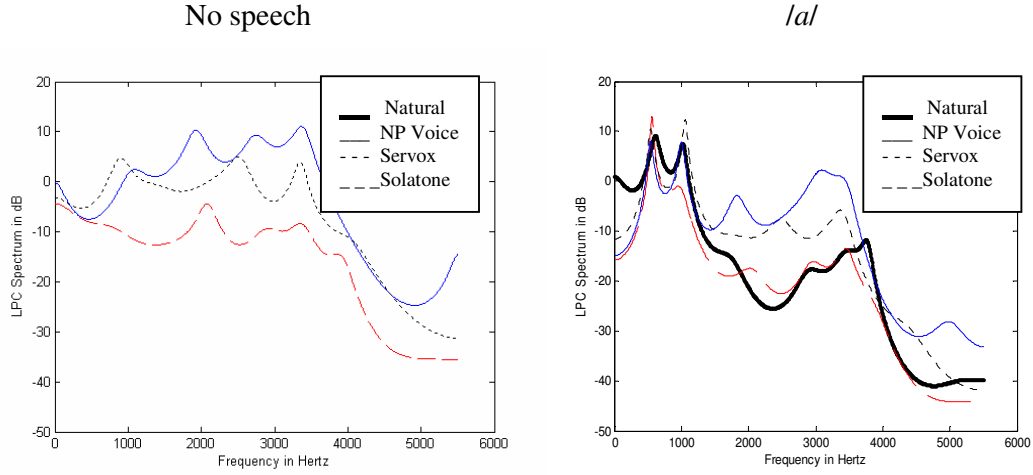


Fig. 5.1 Spectra of "no speech", */a/*, */i/* and */u/*, output from three models of electrolarynx for speaker AJ, averaged over 1024 windows.



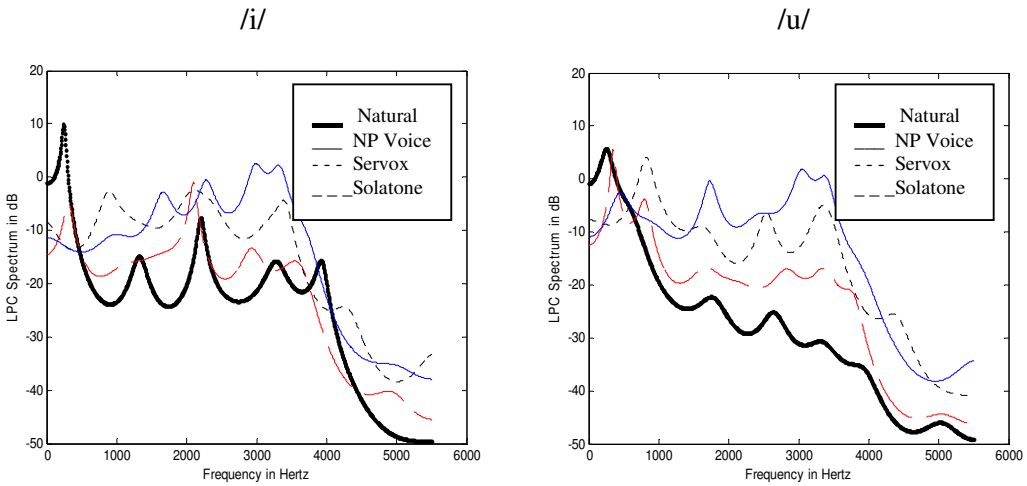


Fig. 5.2 Spectra of "no speech", */a/*, */i/* and */u/*, output from three models of electrolarynx for speaker AJ, averaged over 1024 windows, and smoothed by 16th order LPC

Fig. 5.1 shows the spectra of no-speech segment from the three devices, as used by one normal speaker. These spectra are recorded when the device is held against the neck tissue and the lips are kept closed. The averaged spectrum was obtained from 1024 analysis windows (256-sample windows with 50% overlap). For ease of viewing and better comparison, Fig. 5.2 shows the same spectra smoothened by 16th order LPC. The three devices show different characteristics in their background noise. NP Voice has a peak at about 3.5 kHz, and Servox has a peak at about 1 kHz. The Solatone model does not have a strong peak, and the noise spectrum is relatively wideband. This figure also shows the spectra for three cardinal vowels */a/*, */i/* and */u/*, obtained in the same manner. Spectrum of the naturally uttered vowel is superimposed on the spectral plots of electrolaryngeal speech from the three devices. For all the vowels, the spectra clearly show a low frequency deficit. Further, because of the background noise, the vowel formants appear masked to a certain extent. The most severe effect is visible in case of */u/*.

Figs. 5.3-5.6 show the spectrograms for no-speech segment, */aie/,* the Hindi sentence *"aapkaa naam kya hai?"*, and the English sentence *"where were you a year ago?"*respectively, from the three devices, as used by speaker AJ with normal larynx. It is seen from the spectrograms that the first formant region is generally weak, indicating low frequency deficit. The formant transitions are less visible. Further, the silence regions get

filled by the background noise, and the effect is more pronounced in the high frequency regions.

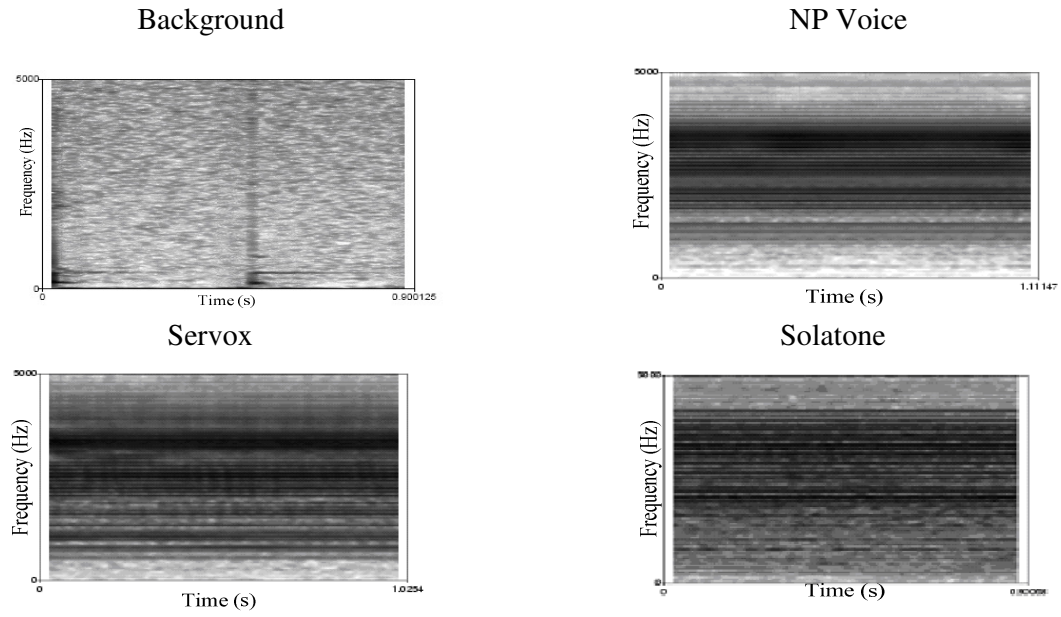


Fig. 5.3 Comparison of spectrograms of noise generated by three electrolarynx devices. */aie/*

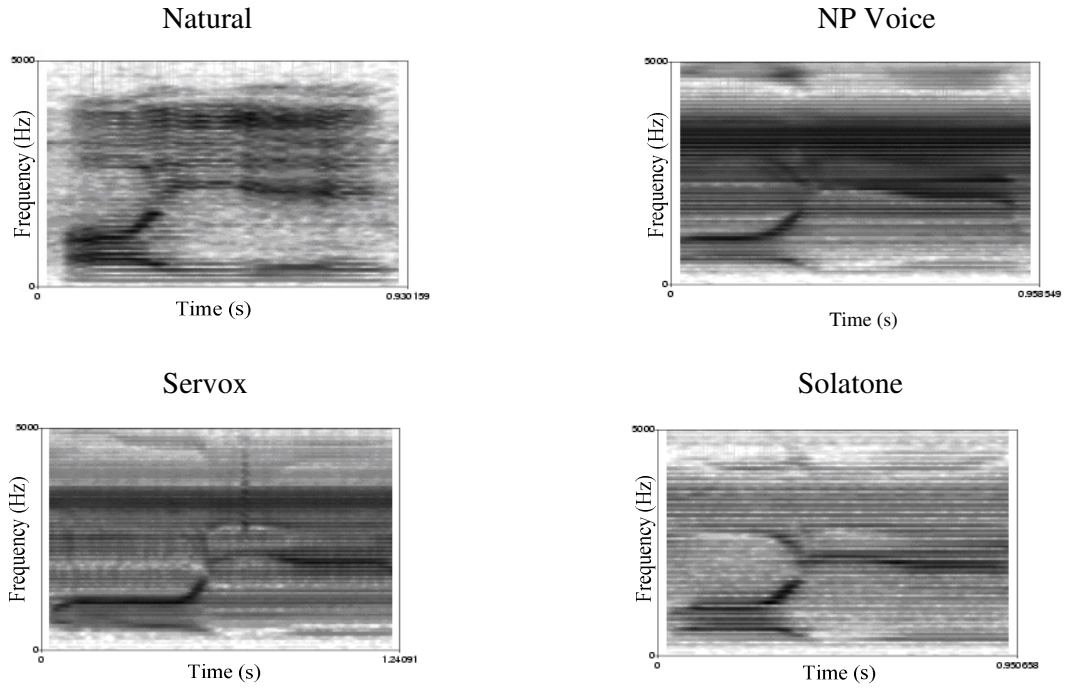


Fig. 5.4 Spectrograms of natural and electrolaryngeal */aie/,* generated by speaker AJ using three models of electrolarynx

"aapkaa naam kya hai"

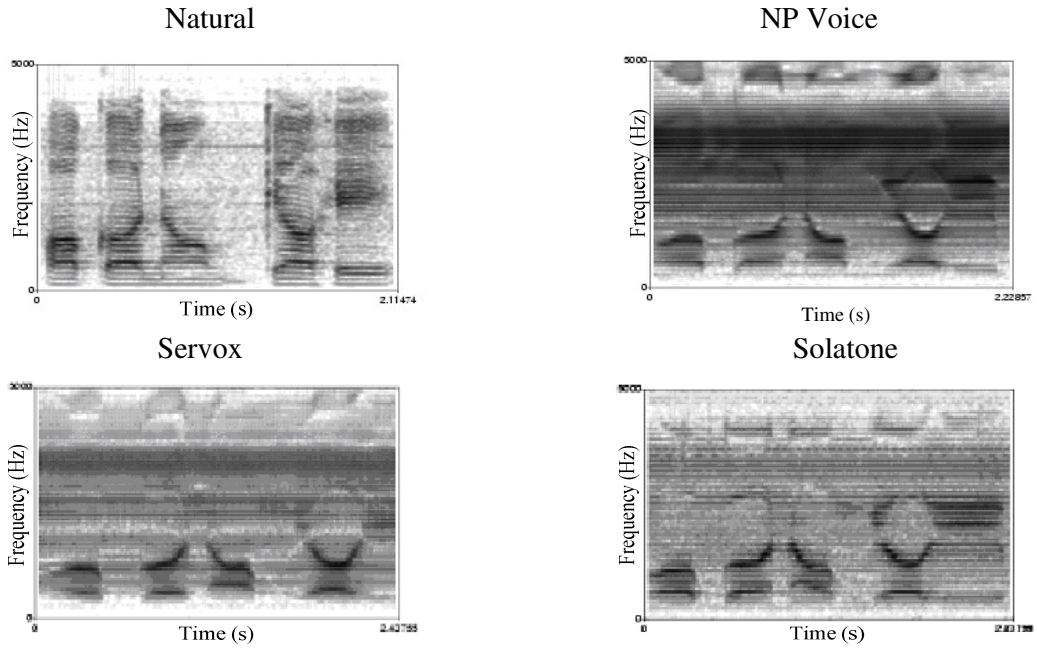


Fig. 5.5 Spectrograms of natural and electrolaryngeal utterance of Hindi sentence "*aapkaa naam kya hai*", generated by speaker AJ using three models of electrolarynx

"where were you a year ago"

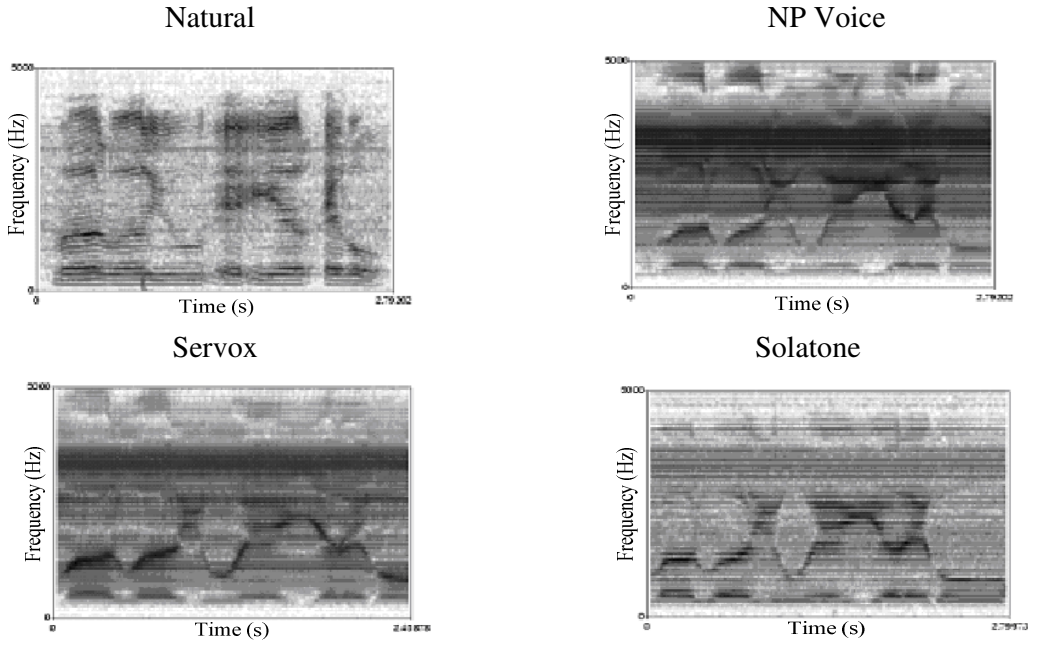


Fig. 5.6 Spectrograms of natural and electrolaryngeal utterance of *"where were you a year ago",* generated by speaker AJ using three models of electrolarynx

It is easily observable from a relative study of the spectra and spectrograms that all the models of electrolarynx tested suffer from poor low frequency content as compared to laryngeal speech. Among the three models, Solatone has the best low frequency performance, while NP Voice has the poorest. In general, the first formants of all the speech segments recorded using the three models of electrolaryx are at a higher frequency than in normal speech.

5.2 MATLAB based implementation

The signals were acquired at a sampling rate of 11.025 kHz with the ADC of PC sound card and were processed at this rate with the MATLAB based implementation. The various implementations in MATLAB are described in the following sub-sections.

5.2.1 Spectral deficit compensation

As earlier stated, the electrolaryngeal speech suffers from a serious low frequency spectral deficit. To validate and compensate for the above, the average spectrum of natural */a/* of a particular speaker was compared with that of electrolaryngeal */a/* uttered by the same speaker with identical vocal tract shape but no glottal excitation. Analysis was done using windows of size 512 samples, with 50% overlap, and smoothing by 16^{th} order LPC parameters. Then, a digital FIR filter of order 256 was designed with a desired frequency response equal to the calculated spectral deficit using the built-in "fir2" command of MATLAB. This command uses the frequency sampling method for designing any arbitrary shape filter B, and windows the impulse response with a Hamming window by default. The filter has linear phase, i.e., symmetric coefficients obeying $B(k) = B(N+2-k)$, $k = 1,2,...,N+1$. The filter response is shown in Fig. 5.7. The electrolaryngeal */a/*, when filtered by this filter, received the desired low frequency spectral boost and acquired a spectrum almost identical to that of natural */a/*. The results are shown in Fig. 5.8. Similar results were obtained for other utterances too.

5.2.2 Modified spectral subtraction based on ABNE

Offline implementation of the modified spectral subtraction algorithm using ABNE was tested in MATLAB to determine the optimum values of the spectral subtraction factor α, spectral floor factor β and exponent factor γ. The different values tested were: $\alpha = 0.5, 1, 2, 4$; $\beta = 0.1, 0.01, 0.001$; $\gamma = 1$ and 2. First, the value of β was fixed, and the optimum

value-pair of (α, γ) was found. These value-pairs for the three values of β were then compared with each other to find the best value of α. By this scheme, the optimum set of values was found to be $\alpha = 2$, $\beta = 0.001$, and $\gamma = 1$. Fig. 5.9(b) shows the results of the modified ABNE implementation in MATLAB, without any spectral compensation.

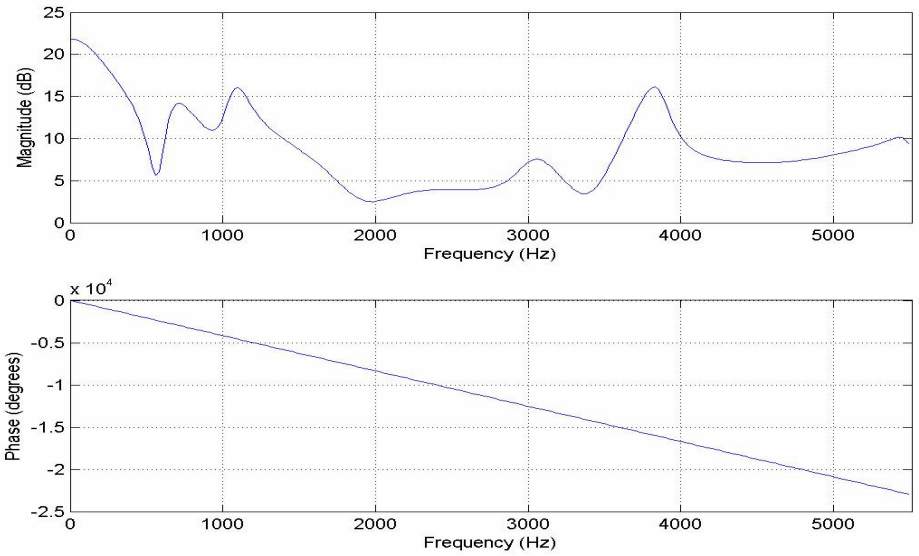


Fig. 5.7 Frequency response of the spectral compensation FIR filter of order 256

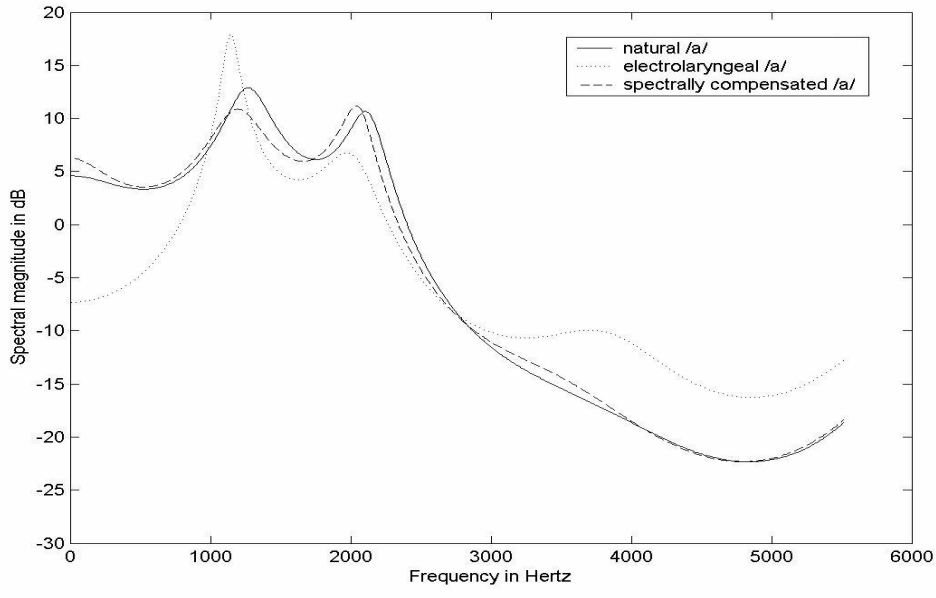
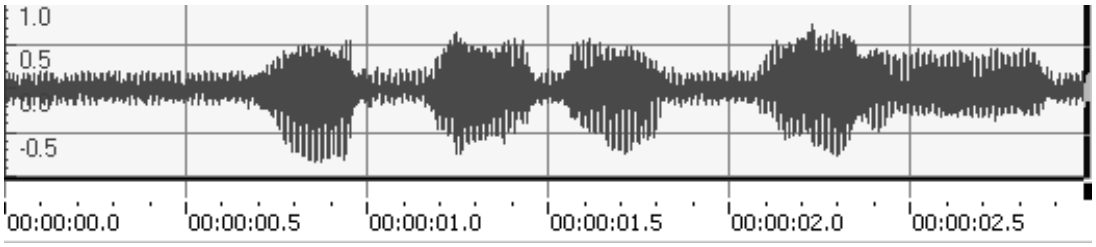
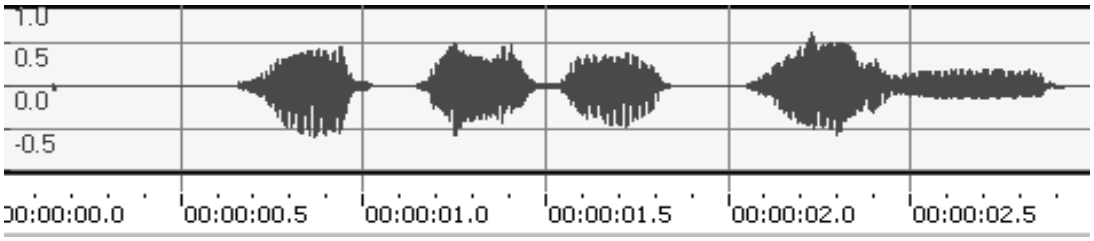


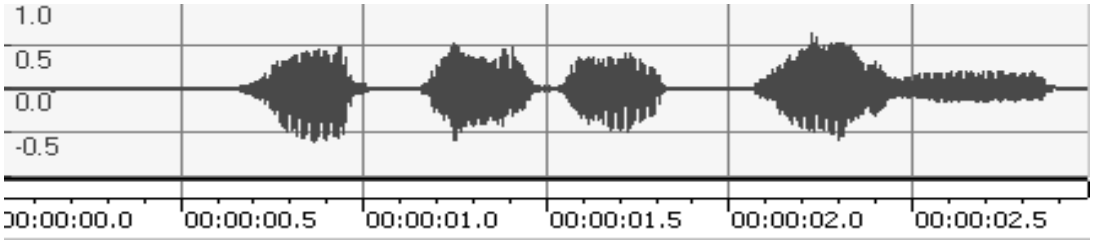
Fig. 5.8 Compensation of low frequency spectral deficit of electrolaryngeal speech



(a) Recorded speech waveform. Speaker AJ, material: "Aapkaa naam kyaa hai", generated using Solatone electrolarynx



(b) Enhanced signal using MATLAB-based modified ABNE



(c) Enhanced signal using MATLAB-based extended ABNE

Fig. 5.9 Recorded and enhanced speech using modified and extended ABNE without spectral compensation. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 1$

5.2.3 Extended spectral subtraction based on ABNE

Reduction in computational complexity by doing away with explicit phase calculation was investigated by offline implementation of extended spectral subtraction using ABNE. As the results in Fig. 5.9(c) show, there is no change compared to modified ABNE as far as noise reduction is concerned, but analysis of computational time showed significant improvement. For example, in the case of the 3 s long speech sample taken, execution time of modified ABNE was 0.83 s, while that of extended ABNE was 0.13 s (found using the "*tic-toc*" command pair in MATLAB).

5.2.4 Phase reconstruction using iterative and non-iterative techniques

Circular aliasing is inherent in the use of DFT for computation of cepstrum. This effect is reduced by padding the sequence with a number of zero valued samples exceeding twice its pitch period. In our implementation, N number of zero-valued samples were padded on either side of the analysis window of length N for proper DFT based cepstrum computation without the effect of circular aliasing. As expected, higher the value of N, better was the reconstructed speech.

The basic problem of using iterative technique of phase reconstruction in our case is that the assumption of $v(0)$ being known is not true here. We do not know any value of the noise-free speech with clean phase. In our implementation, $v(0)$ was taken to be equal to the corresponding value of the noise-free speech with noisy phase, the validity of the assumption being questionable. A difference below 0.1 (for input samples in the range [-1,+1]), of corresponding FFT magnitudes of consecutive iterations, was taken as the criterion for convergence. Number of iterations needed for convergence varied over a wide range, from a few tens to several hundreds. Similar observations have been reported in [24].

In both the iterative and non-iterative techniques, consistent results were obtained for values of N exceeding 1024. There was slight speech distortion, which was removed by increasing the precision to "long" data type in the MATLAB program. But the overall quality of the phase-reconstructed speech was found to be comparable to that of the noise-free speech using noisy phase.

5.2.5 Noise reduction based on minimum statistics

At first, a static noise estimation based on minimum statistics was implemented, in which the minimum magnitude of each spectral component over the entire speech recording was calculated in the beginning. This fixed minimum value, multiplied by an experimentally chosen bias factor (equivalent to the over-subtraction factor of ABNE and QBNE) α, was then taken as the noise estimate and used for spectral subtraction throughout. The optimum value of α was found to be 20. Some residual noise was observed in the processed speech, as shown in Fig. 5.10 (c). As a possible solution, it was decided to use a smoothened version of minima values using L point frequency averaging, by using following equation

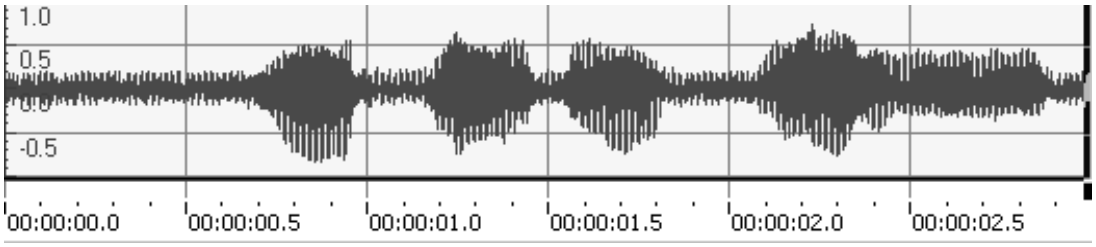
$$M_{av}(k) = (1/L) \sum_{i=-(L-1)/2}^{(L-1)/2} M_n(k-i) \tag{5.1}$$

where $M_n(k)$ are the originally calculated minima values which are averaged to give $M_{av}(k)$. It was found that 9-sample averaging, i.e. spectral smoothing of the minima values over 4 preceding and 4 succeeding frequency components, resulted in removal of the residual noise. The results are presented in Fig. 5.10 (d).

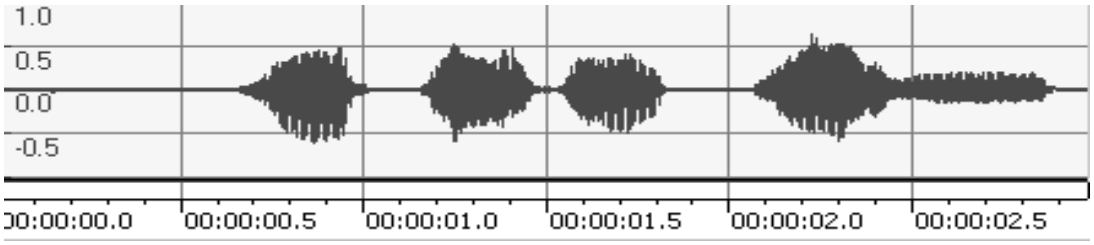
Thereafter, a dynamic noise estimation algorithm based on minimum statistics was implemented. The degraded signal was analyzed on a frame-by-frame basis, to obtain an array of the magnitude spectral values for each frequency sample, for a certain number of the past frames. The minimum magnitude value of each frequency component in this array, multiplied by α, was used as the dynamic noise estimate. The rate at which this algorithm reacts to changes in the noise depends on the number of past frames used. If the number is too small, the estimation will not be accurate. If the number is too large, reaction to changes will be slow. In this approach, the buffer for all the frequency samples has to be reconstructed at each frame. For fast processing, an efficient indexing algorithm [45] was implemented. For each frequency sample ω_k, two buffers were used. Magnitude buffer $D(\omega_k)$ held the spectral magnitude values for M frames. Index buffer $I(\omega_k)$ held the frame number of the corresponding value in $D(\omega_k)$. After computation of magnitude spectrum of sample values in a new frame, the index of oldest value in $D(\omega_k)$ was located from $I(\omega_k)$ and replaced with the new value. Frame numbers in $I(\omega_k)$ were updated. After updating the two buffers, the minimum value in $D(\omega_k)$ was found for each frequency, and these values were used for minimum statistic based noise estimation continuously.

Figs. 5.10 (e) and 5.10 (f) show the results of the implementation of dynamic minimum statistics based noise estimation. A minimum of 20 frames were required in the analysis window for proper noise estimation. But considerable signal attenuation of the speech segments was observed in this case. The performance improved with increase in number of frames in the analysis window, and no significant gain was notable after 50 frames. The result was compared with that obtained by QBNE, using the optimum set of parameters, viz., $\alpha = 20$, $\beta = 0.001$, $\gamma = 1$. Less residual noise was present in the minimum statistic implementation with respect to that in QBNE. Analysis of processing time also revealed the computational efficiency of the minimum statistic technique over the quantile based one. For noise reduction of a 3 s long test utterance, the average time taken by QBNE was 26 s, while the minimum statistic based method took only 1.3 s for the

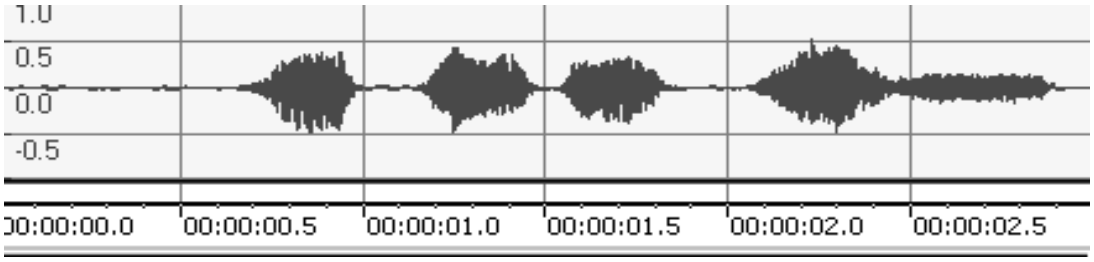
same. From a spectrographic comparison of ABNE and MSBNE (with dynamic estimation over 40 frames) in Fig. 5.11, it is seen that dynamic MSBNE gives better noise reduction than ABNE, especially evident in the non-speech regions. The dynamic nature and computational efficiency of MSBNE make it a suitable technique for real time implementation.



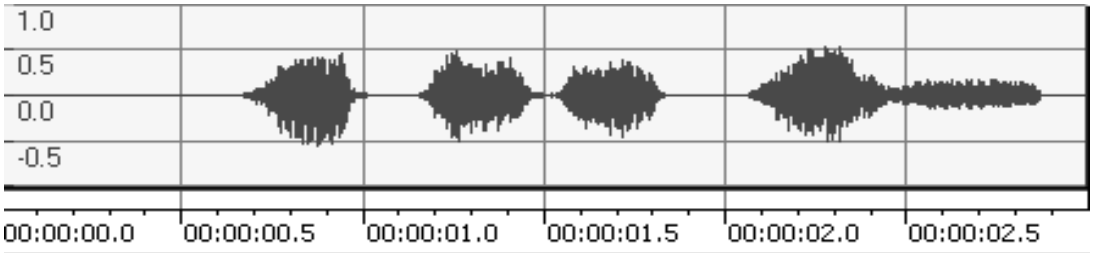
(a) Recorded speech waveform, Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx.



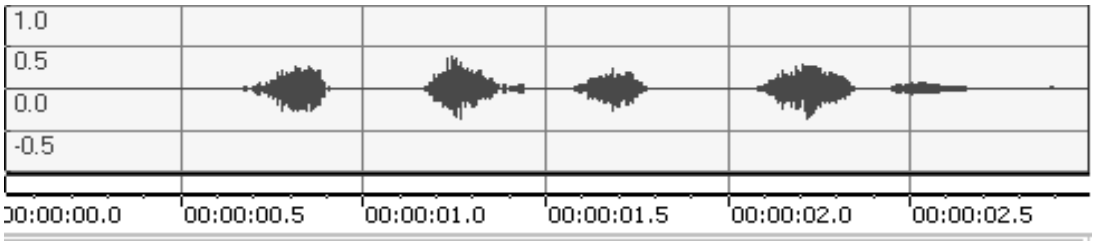
(b) Enhanced signal using MATLAB-based ABNE



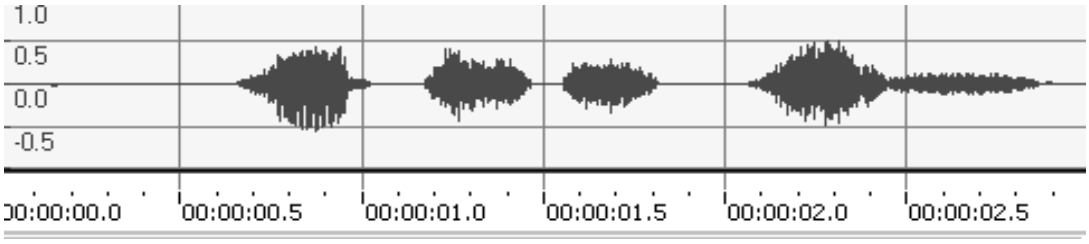
(c) Enhanced speech waveform using static MSBNE



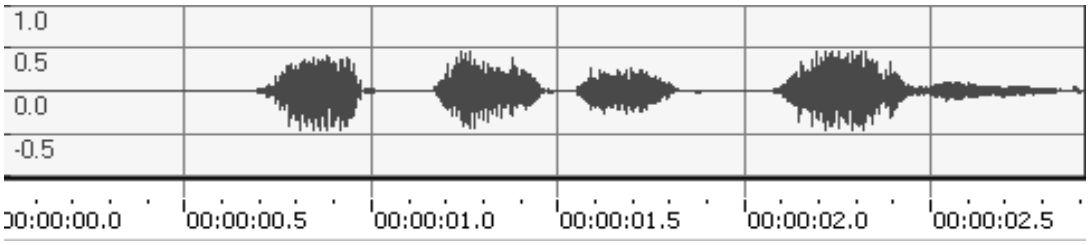
(d) Enhanced speech waveform using static MSBNE with 9-sample frequency domain smoothing



(e) Enhanced speech waveform using dynamic smoothed MSBNE with minima calculation over 20 windows

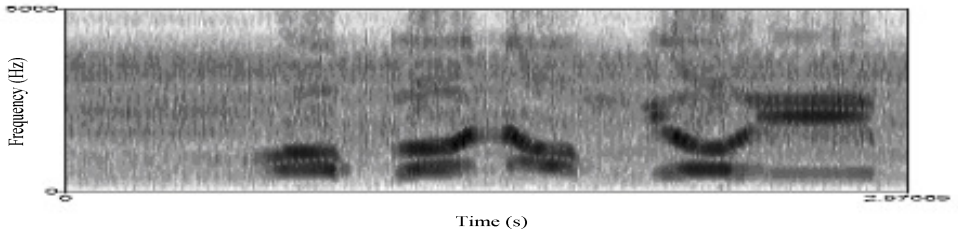


(f) Enhanced speech waveform using dynamic smoothed MSBNE with minima calculation over 50 windows

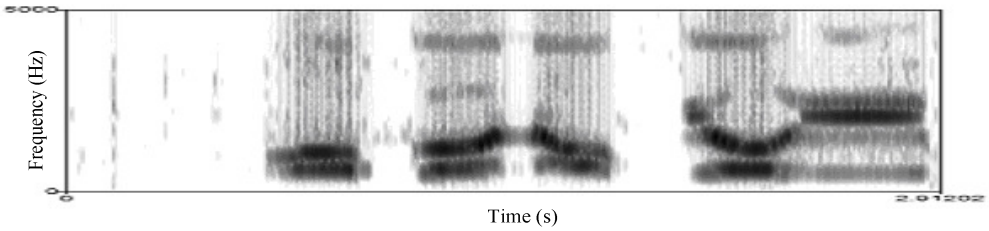


(g) Enhanced speech waveform using dynamic smoothed MSBNE with minima calculation over 50 windows, followed by spectral compensation

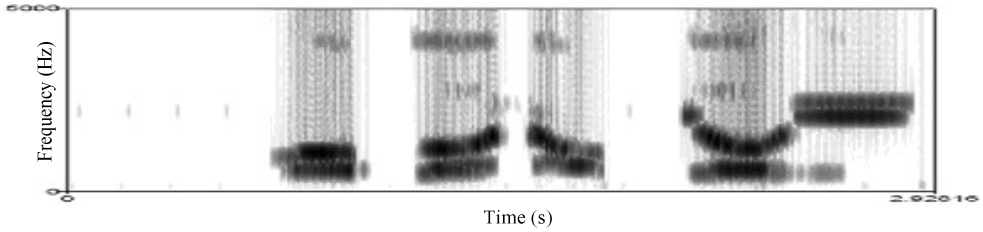
Fig. 5.10 Recorded and enhanced speech using ABNE, static and dynamic MSBNE. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 1$ for ABNE, and $\alpha = 20$, $\beta = 0.001$, $\gamma = 1$ for MSBNE



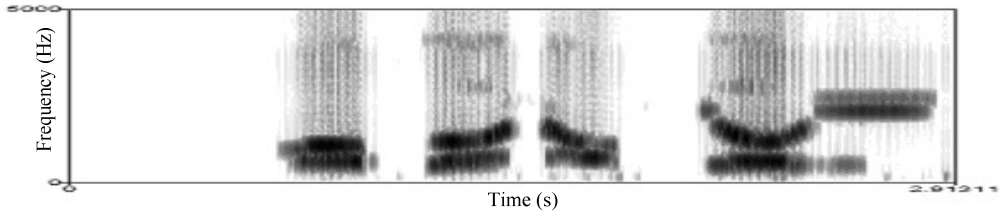
(a) Recorded speech. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx.



(b) Speech enhanced using modified ABNE



(c) Speech enhanced using dynamic smoothed MSBNE



(d) Speech enhanced using dynamic smoothed MSBNE with spectral compensation

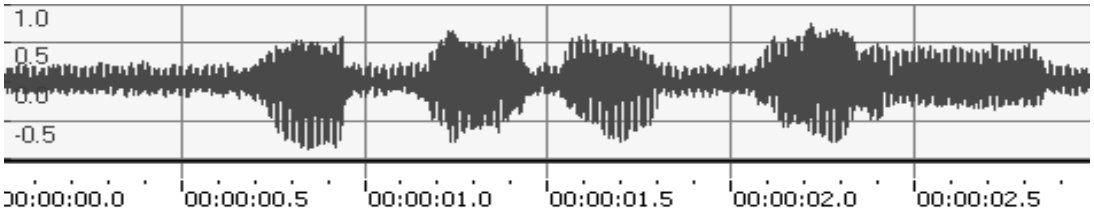
Fig. 5.11 Recorded and enhanced speech using ABNE, and dynamic MSBNE, with and without spectral compensation. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 1$ for ABNE, and $\alpha = 20$, $\beta = 0.001$, $\gamma = 1$ for MSBNE.

5.3 Real time DSP based implementation using ABNE

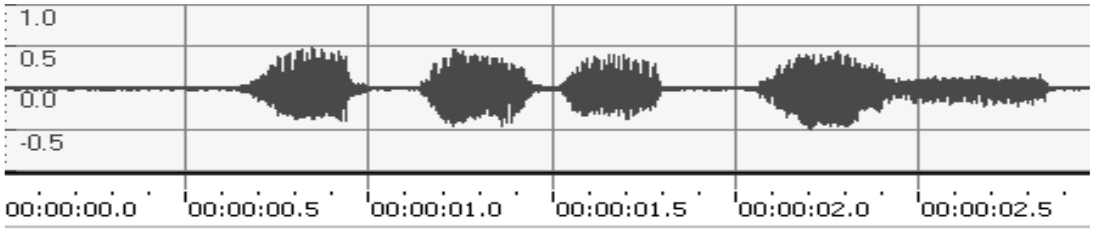
Based on our deduction of the optimum set of parameters from the offline implementation of spectral subtraction, we have used the following parameters: spectral subtraction factor α = 2, spectral floor factor β = 0.001, and exponent factor γ =1. The real-time implementation of ABNE based spectral subtraction has been done pitch synchronously, i.e. taking analysis window length equal to multiple of the pitch period. We have taken

exactly two pitch periods, which are padded with zero value samples to make the total sequence length of 256, for FFT based processing on the DSP board. For an effective sampling rate of 12 kSa/s, and a pitch of 115 Hz (in the Solatone recording), the window length is 208 samples, zero-padded to make it 256 samples. Noise estimation was carried out using modified ABNE, as described in Section 3.4.1. The number of windows used for averaging was varied from 10 to 100, and effects of overflow were compensated by scaling as discussed in Section 4.3.2.

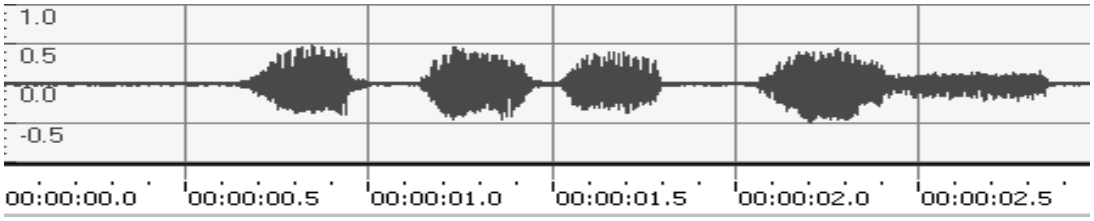
Real time implementation was also carried out using extended spectral subtraction. The quality of noise reduction obtained was similar to that in modified spectral subtraction. Results of real time implementation of modified and extended ABNE are shown in Fig. 5.12 (b) and (c).



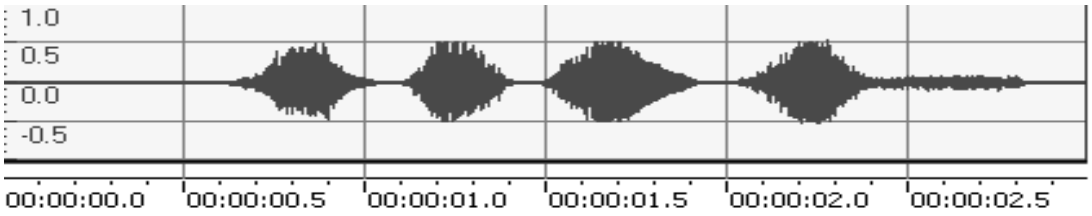
(a) Recorded speech waveform, Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx.



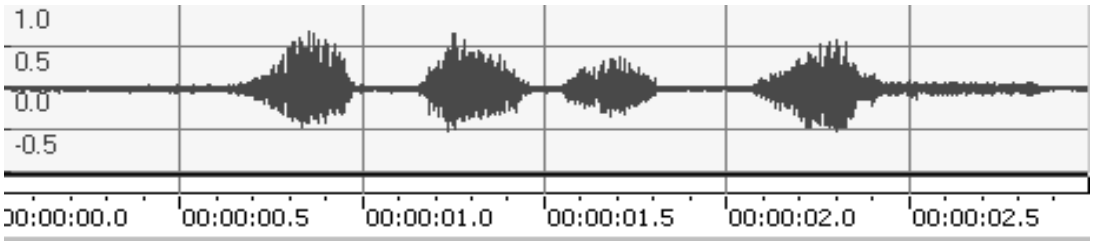
(b) Speech enhanced using real time modified ABNE



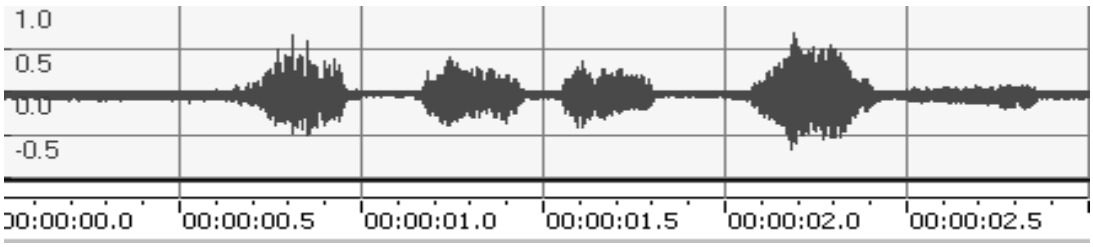
(c) Speech enhanced using real time extended ABNE



(d) Speech enhanced using real time dynamic MSBNE



(e) Speech enhanced using real time dynamic MSBNE with spectral compensation



(f) Speech enhanced using real time dynamic MSBNE combined with averaging

Fig. 5.12 Recorded and enhanced speech using real time ABNE, and dynamic MSBNE, with and without spectral compensation. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 1$ for ABNE, and $\alpha = 10$, $\beta = 0.001$, $\gamma = 1$ for MSBNE.

5.4 Real time DSP based implementation using MSBNE

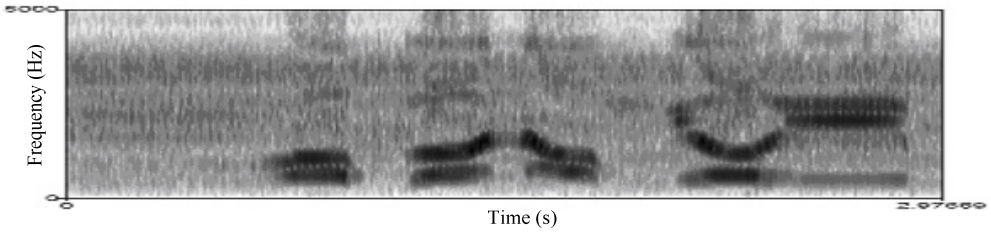
Real time implementation was carried out using minimum statistics based noise estimation, as discussed in Section 3.9. Analysis windows of size 256 samples with 50% overlap were used. The values of the processing parameters were $\alpha = 10$, $\beta = 0.001$, $\gamma = 1$, FFT size = 256 samples. Though the optimum value of α found from offline implementation of MSBNE was 20, use of any value of α exceeding 10 in the real time implementation resulted in severe loss of signal, possibly due to finite precision arithmetic. The dynamic MSBNE scheme was implemented. 10 windows, further decimated in time by a factor of 10, were used for minima calculation, thus giving effective analysis duration of 100 windows.

Spectral compensation was done by shaping the magnitude spectrum using coefficients of a 256 order FIR filter designed in MATLAB. Since the magnitude and phase spectra of a minimum phase system are uniquely related, it is expected that some improvement in performance may be obtained by doing some phase correction along with the magnitude shaping in spectral compensation. This needs to be investigated further. Fig. 5.12 (d) and (e) show the results of real time implementation of dynamic MSBNE without and with spectral compensation respectively.

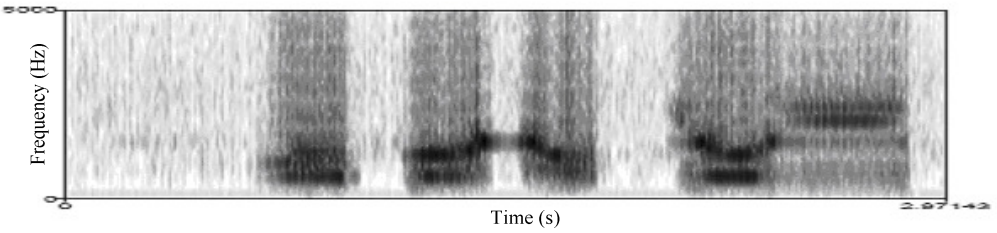
In the above technique, minima were calculated every 10^{th} window, with no processing taking place during these 10 windows. A variation of the above algorithm was also implemented, in which averaging was combined with MSBNE. In this scheme, average spectral magnitudes were calculated during 10 windows, and this average was retained for minima calculation after every 10^{th} window. Results of real time implementation of this combined technique are shown in Fig. 5.12 (f).

From Fig. 5.12, it can be seen that effective noise cancellation is obtained in real time implementations of both ABNE and dynamic smoothed MSBNE. The combined technique presently does not perform as well as the individual ones. The distribution of processing load between averaging and minima calculation may be investigated to improve its performance.

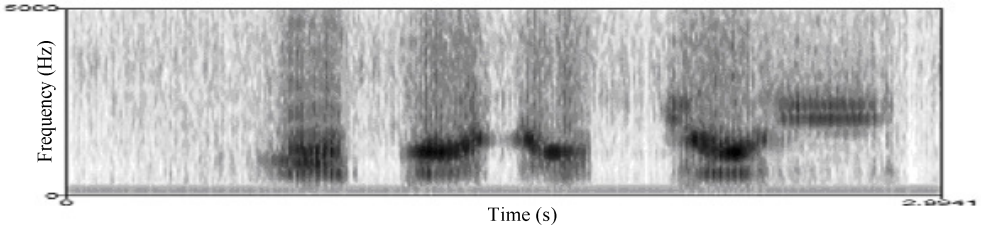
Fig. 5.13 shows a spectrographic comparison of speech enhanced using real time ABNE and MSBNE, with and without spectral compensation. It is seen that in all the implementations, noise is effectively reduced in the non-speech regions, and the sonorant periods are clearly distinct. However, a comparison with Fig. 5.11 shows that noise reduction in real time MSBNE was not as effective as that obtained in offline processing. This is possibly due to the effects of finite precision arithmetic, and also the fact that the value of α could not be made more than 10 in real time processing, though the optimal value was found to be 20 from offline investigations.



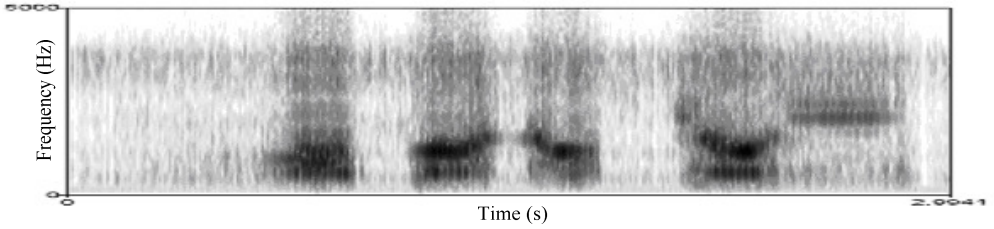
(a) Recorded speech. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx.



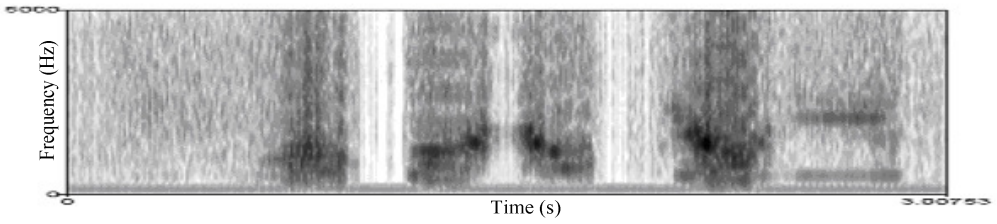
(b) Speech enhanced using real time ABNE



(c) Speech enhanced using real time dynamic smoothed MSBNE



(d) Speech enhanced using real time dynamic smoothed MSBNE with spectral compensation



(e) Speech enhanced using real time dynamic MSBNE combined with averaging

Fig. 5.13 Spectrographic comparison of recorded and enhanced speech using real time ABNE, and MSBNE, with and without spectral compensation. Speaker AJ, material: "*Aapkaa naam kyaa hai*", generated using Solatone electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 1$ for ABNE, $\alpha = 10$, $\beta = 0.001$, $\gamma = 1$ for MSBNE, and $\alpha = 6$, $\beta = 0.001$, $\gamma = 1$ for MSBNE combined with averaging

Chapter 6

SUMMARY AND CONCLUSIONS

6.1 Summary

Some of the major drawbacks of electrolaryngeal speech are its poor intelligibility due to presence of self leakage noise, as well as low frequency spectral deficit. The earlier work in our lab by Bhandarkar [10] and Pratapwar [12] as part of their M.Tech. dissertations have shown that pitch synchronous application of spectral subtraction gives effective noise suppression. In the ABNE technique, noise estimate obtained during an initial non-speech region is subtracted from noisy speech to give an enhanced magnitude spectrum which is then coupled with the phase spectrum of noisy speech for re-synthesizing the clean speech. Characteristics of leakage noise change slowly with speech utterance, and also with the placement and orientation of the vibrator. So, a dynamic noise estimation system is required. Pratapwar [12] has shown that QBNE technique can be effectively used for dynamic estimation of noise, and best results are obtained with quantile values as a function of frequency and SNR. Budiredla [15] had investigated real-time implementation of the QBNE algorithm using TI DSP TMS320C6211 based DSK board.

The objective of this project was to develop a real time electrolaryngeal speech enhancement system, which incorporated:

(a) Low frequency deficit compensation

(b) Use of a clean phase spectrum along with the clean magnitude spectrum during signal resynthesis

(c) Dynamic noise estimation technique for better background noise reduction

Towards this end, spectral and spectrographic investigations were carried out using three models of electrolarynx, namely, Servox, Solatone, and NP Voice, and 5 speakers. A spectral compensation filter was designed using the difference in average spectra of natural and electrolaryngeal speech.

In the absence of a technique for cleaning the noisy phase spectrum, methods of phase reconstruction from the cleaned magnitude spectrum, assuming the cleaned speech

samples to be a minimum phase sequence, were studied. Phase reconstruction was investigated using iterative and non-iterative computations.

Real time implementation of QBNE could not be achieved because of memory constraints of the DSP board used. As an alternative, a dynamic minimum statistics based noise estimation (MSBNE) has been studied, where minimum value of each spectral sample in a set of past frames is used for estimating the magnitude spectrum of noise. This technique is much easier to implement under the memory constraints of a DSP board. Real time implementation of the ABNE and MSBNE algorithms has been carried out using Blackfin processor ADSP BF 533 based EZKIT Lite evaluation board from Analog Devices. In addition to spectral subtraction, the implementation incorporated compensation for low frequency deficit. Phase resynthesis assuming minimum phase model has not been included in this implementation.

6.2 Conclusions

Effective low frequency deficit compensation of electrolaryngeal speech was obtained using a 256 order FIR filter with linear phase. However, since magnitude and phase spectra of a minimum phase system are uniquely related, spectral shaping of the magnitudes alone (as done in real time spectral compensation) may be a source of certain distortions. Phase correction coupled with magnitude shaping may lead to better spectral compensation on the whole.

From the offline implementation of the phase reconstruction techniques, it was found that the non-iterative method performed slightly better than the iterative one. However, since both the techniques did not show any significant improvement in quality over the method using noisy phase, phase reconstruction was not included in the real time implementation.

Offline implementation of dynamic MSBNE showed effective noise reduction. Compared to the other dynamic noise estimation algorithm using QBNE, less residual noise was observed in the silence regions. Spectral smoothing of the estimated noise spectrum resulted in even better noise reduction. Moreover, the optimum number of windows needed for dynamic updating of the noise spectrum is around 40 in MSBNE, less than the 55 required in QBNE. MSBNE was found to be computationally more efficient than QBNE, indicating its suitability for dynamic noise estimation in real time implementation.

Real time spectral subtraction was implemented on 16-bit fixed point DSP Blackfin based EZKIT Lite board, using ABNE and MSBNE. It showed effective noise reduction, though not as good as offline processing, possibly because of the effects of finite precision arithmetic.

6.3 Suggestions for future work

Dynamic noise estimation techniques involving two stages of minima calculation, or minima calculation coupled with some statistical measure like mean or median, need to be investigated. Algorithms for phase reconstruction from the clean magnitude spectrum, and the feasibility of their application in real time implementation require further study. Application of some phase correction, in addition to the magnitude shaping, may lead to better spectral compensation. The real time programs may be written in assembly language to improve the speed and overall performance of processing. Listening tests have to be carried out to quantify intelligibility and quality improvements.

Appendix A

SOME RELATED TOPICS

A.1 Semi-automatic pitch control for an electrolarynx

Speech produced with an electrolarynx is usually monotonous, and even with devices featuring pitch control, controlling continuous pitch movement by hand while speaking is too involved. A semi-automatic pitch control circuit with which discrete pitch movements can be made has been reported [2]. In Dutch intonation, about 70 percent of all pitch contours are variations of the so-called hat pattern, characterized by a fast rise and fall of fundamental frequency on two accented syllables. Also, in many languages including Dutch, there is a slow decline in pitch during phonation, the phenomenon being called "declination".

A pitch control circuit for an electrolarynx incorporates these frequent pitch movements, including an automatic declination function. The lower declination is started when a control button on the electrolarynx is depressed. When the control is slid forward, a resistor in the declination generator is short-circuited, which causes the pitch of the oscillation circuit driving the vibrator to follow the upper declination level. Pitch drops back to the lower declination level when the control is moved backwards again.

The advantage of discrete pitch control over continuously variable pitch control is that the excursion and the duration of the pitch movements need not be controlled by the electrolarynx user. The control task is limited to the correct placements of the pitch movements in time, in order to induce the perception of sentence accents.

A.2 Speech enhancement based on wavelet denoising

We assume the sampled noisy speech signal $y(k)$ is generated from

$$y(k) = s(k) + \sigma(k)n(k), \quad k=0,\ldots,K\text{-}1 \tag{A.1}$$

where $s(k)$ is the clean speech signal, $n(k)$ represents an independent noise source with unit variance, and $\sigma(k)$ is the noise level. Wavelet denoising is a non-parametric

estimation method that has been proposed in recent years for speech enhancement applications [21], [22]. The goal of wavelet denoising is to optimize the mean-squared error between $s(k)$ and its estimate $\hat{s}(k)$, subject to the side condition that, with a high probability, the estimation $\hat{s}(k)$ is at least as smooth as $s(k)$. This constraint provides an optimal trade-off between the bias and variance of the estimate by keeping the two terms of the same order of magnitude. The implementation of wavelet denoising is a three step procedure involving wavelet decomposition, nonlinear thresholding and wavelet reconstructing. Although wavelet denoising provides a theoretical framework to the estimation problem, attributes specific to speech must still be exploited to achieve good performance for the speech enhancement application.

The noisy speech is first preprocessed using a spectral subtraction routine to reduce the noise level while minimizing distortion in speech. Then a wavelet packet (WP) decomposition is designed to mimic the critical bands as widely used in perceptual auditory modeling. This perceptual wavelet (PW) transform is used to decompose the preprocessed signal y'_k into sub-bands.

Wavelet denoising involves thresholding in which coefficients below a specified value (*i.e.* threshold) are set to zero. This is called hard-thresholding. Alternatively, soft-thresholding simply shrinks or scales coefficients below the threshold value. Donoho and Johnstone derived a general optimal universal threshold for the Gaussian white noise under a mean squared error criterion [22]. However, in practice this threshold is not ideal for speech signals due the poor correlation between MSE and subjective quality and the more realistic presence of correlated noise. An adaptive time-frequency dependent threshold estimation method has been reported. This involves first estimating the standard deviation of the noise, σ, for every sub-band and time frame. For this, a quantile-based noise tracking approach has been adapted. Given σ, the threshold, λ, is again calculated for each sub-band and time frame. The last stage simply involves re-synthesizing the enhanced speech using the inverse perceptual wavelet transform.

Test results [22] showed that wavelet denoising itself (*i.e.,* without preprocessing) tends to work remarkably well on signals with moderate levels of noise, while producing greater distortion when the noise level is high. Its application to enhancement of electrolaryngeal speech has not been reported.

A.3 Minima controlled recursive averaging

A noise estimation approach, namely Minima Controlled Recursive Averaging (MCRA) [36], has been proposed, that combines the robustness of the minimum tracking with the simplicity of the recursive averaging. The noise estimate is obtained by averaging past spectral power values, using a smoothing parameter that is adjusted by the speech presence probability in subbands. The speech presence probability is controlled by the minima values of a smoothed periodogram. In contrast to the MS and related methods, the minimum tracking is not crucial, since it only controls the recursive averaging as a secondary procedure. The recursive averaging is carried out without a hard distinction between speech absence and presence, thus continuously updating the noise estimate even during weak speech activity. Additionally, the smoothing of the noisy periodogram is carried out in both time and frequency, which takes into account the strong correlation of speech presence in neighboring frequency bins of consecutive frames. It has been shown that the MCRA noise estimation is computationally efficient, and is characterized by its ability to quickly follow abrupt changes in the noise spectrum.

As an improvement to MCRA, an Improved Minima Controlled Recursive Averaging (IMCRA) approach has been reported [38], for noise estimation in adverse environments involving non-stationary noise, weak speech components, and low input signal-to-noise ratio (SNR). The noise estimate is obtained by averaging past spectral power values, using a time-varying frequency-dependent smoothing parameter that is adjusted by the signal presence probability. The speech presence probability is controlled by the minima values of a smoothed periodogram. The proposed procedure comprises two iterations of smoothing and minimum tracking. The first iteration provides a rough voice activity detection in each frequency band. Then, smoothing in the second iteration excludes relatively strong speech components, which makes the minimum tracking during speech activity robust. Test results show that in non-stationary noise environments and under low SNR conditions, the IMCRA approach is very effective. In particular, compared to a competitive method, it obtains a lower estimation error, and when integrated into a speech enhancement system achieves improved speech quality and lower residual noise.

REFERENCES

[1] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals,* Englewood Cliffs, New Jersey: Prentice Hall, 1978.

[2] Y. Lebrun, "History and development of laryngeal prosthetic devices," *The Artificial Larynx.* Amsterdam: Swets and Zeitlinger, pp. 19-76, 1973.

[3] L. P. Goldstein, "History and development of laryngeal prosthetic devices," *Electrostatic Analysis and Enhancement of Alaryngeal Speech*, Springfield, Ill: Charles C. Thomas, pp. 137-165, 1982.

[4] M. Weiss, G. Y. Komshian, and J. Heinz, "Acoustical and perceptual characteristics of speech produced with an electronic artificial larynx," *J. Acoust. Soc. Am.*, vol. 65, No. 5, pp. 1298-1308, 1979.

[5] H. L. Barney, F. E. Haworth, and H. K. Dunn, "An experimental transistorized artificial larynx," *Bell Systems Tech. J.*, vol. 38, No. 6, pp. 1337-1356, 1959.

[6] Q. Yingyong and B. Weinberg, "Low-frequency energy deficit in electrolaryngeal speech," *J. Speech and Hearing Research*, vol. 34, pp. 1250- 1256, 1991.

[7] C. Y. Espy-Wilson, V. R. Chari, and C. B. Huang, "Enhancement of alaryngeal speech by adaptive filtering," in *Proc. ICSLP*, pp. 764-771, 1996.

[8] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process*, Vol. 27, No. 2, pp.113-120, 1979.

[9] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proc.IEEE ICASSP'79*, pp. 208-211, 1979.

[10] S. M. Bhandarkar / P. C. Pandey (Supervisor), "Reduction of background noise in artificial larynx" *M.Tech. Dissertation,* Dept. of Electrical Engineering, IIT Bombay, January 2002.

[11] P. C. Pandey, S. M. Bhandarkar, G. K. Bachher, and P. K. Lehana, "Enhancement of alaryngeal speech using spectral subtraction" in *Proc. 14th Int. Conf. Digital Signal Processing (DSP 2002),* Santorini, Greece, pp. 591-594, 2002.

[12] S. S. Pratapwar / P. C. Pandey (Supervisor), "Reduction of background noise in artificial larynx", *M.Tech. Dissertation,* Dept. of Electrical Engineering, IIT Bombay, Feb 2004.

[13] S. S. Pratapwar, P. C. Pandey, and P. K. Lehana, "Reduction of background noise in alaryngeal speech using spectral subtraction with quantile based noise estimation", in *Proc. 7th World Conference on Systemics, Cybernetics, and Informatics,*(*SCI 2003*), Orlando, Florida, 2003.

[14] N. W. D. Evans and J. S. Mason, "Time-frequency quantile-based noise estimation", in *Proc. EUSIPCO '02*, 2002.

[15] B. R. Budiredla / P. C. Pandey (Supervisor), "Real-time implementation of spectral subtraction for enhancement of electrolaryngeal speech", *M.Tech. Dissertation,* Dept. of Electrical Engineering Department, IIT Bombay, July 2005.

[16] H. Gustaffson, S. Nordholm, and I. Claesson, "Spectral subtraction using correct convolution and a spectrum dependent exponential averaging method", *University of Karlskrona Research Report*, 1998. Available online at http://www.its.bth.se/ staff/hgu/pub1/index.html, accessed in March, 2005.

[17] V. Stahl, A. Fisher, and R. Bipus, "Quantile based noise estimation for spectral subtraction and wiener filtering," in *Proc. IEEE ICASSP'00*, Vol.3, pp. 1875-1878, 2000.

[18] N. W. D. Evans and J.S. Mason, "Noise estimation without explicit speech, nonspeech detection: a comparison of mean, median and model based approaches," in *Proc. Eurospeech*, 2001.

[19] N. W. D. Evans, J.S. Mason, and B. Fauve, "Effective real-time noise estimation without speech, non-speech detection: An assessment on the aurora corpus" in *Proc. 14th Int. Conf. Digital Signal Processing (DSP 2002),* Santorini, Greece, pp. 985-988, 2002.

[20] P. C. Pandey, S. S. Pratapwar, and P. K. Lehana, "Enhancement of electrolaryngeal speech by reducing leakage noise using spectral subtraction with quantile based dynamic estimation of noise", in *Proc. 18th International Congress on Acoustics,* (*ICA2004*), Kyoto, Japan, 2004.

[21] Q. Fu and E. A. Wan, " A novel speech enhancement system based on wavelet denoising," Center of Spoken Language Understanding, OGI School of Science and Engineering, at OHSU, 2003. Available online at http:// speech.bme.ogi.edu/ publications/ps/fu03.pdf (accessed in June, 2005).

[22] D. L. Donoho and I. M. Johnston, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, Vol. 81, pp. 425-455, September 1994.

[23] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1975.

[24] M. Hayes, J. S. Lim, and A.V. Oppenheim, "Signal reconstruction from phase orand magnitude," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, pp. 672-680, December 1980.

[25] T. F. Quatieri, and A. V. Oppenheim, "Iterative techniques for minimum phase signal reconstruction from phase or magnitude," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, pp. 1187-1193, December 1981.

[26] B. Yegnanarayana and A. Dhayalan, "Noniterative techniques for minimum phase signal reconstruction from phase or magnitude," in *Proc. IEEE ICASSP*, pp. 639-642, 1983.

[27] S. H. Nawab, T. F. Quatieri, and J. S. Lim, "Signal-reconstruction from short time Fourier transform magnitude," *IEEE Trans. Acoust., Speech Signal Process.,* Vol. ASSP-31, No. 4, pp. 986-998, August 1983.

[28] H. Liu, Q. Zhao, M. Wan, and S. Wang, "Application of spectral subtraction method on enhancement of electrolaryngeal speech," *J. Acoust. Soc. Am.*, vol. 120, No. 1, pp. 398-406, July 2006.

[29] B. L. McKinley and G. H. Whiple, "Model based speech pause detection," in *Proc. IEEE ICASSP-97*, pp. 1179-1182, 20-24 April 1997.

[30] J. Meyer, K. U. Simmer, and K. D. Kammeyer, "Comparison of one- and two-channel noise-estimation techniques," in *Proc. 5th International Workshop on Acoustic Echo and Noise Control* (*IWAENC-97*), London, UK, pp. 137-145, 11-12 September 1997.

[31] J. Sohn, N. S. Kim, and W. Sung, "A statistical model based voice activity detector," *IEEE Signal Processing Letters*, Vol. 6, No. 1, pp. 1-3, January 1999.

[32] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *Proc. IEEE ICASSP-95,* pp. 153-156, 8-12 May 1995.

[33] R. J. McAulay and M. L. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Trans. Acoust., Speech Signal Process.,* Vol. ASSP-28, No. 2, pp. 137-145, April 1980.

[34] C. Ris and S. Dupont, "Assessing local noise level estimation methods: application to noise robust ASR," *Speech Communication*, Vol. 34, No.1-2, pp.141-158, April 2001.

[35] R. Martin, "Spectral subtraction based on minimum statistic," in *Proc. 7th European Signal Processing Conf., EUSIPCO - 94*, Edinburgh, Scotland, pp. 1182-1185, 13-16 September 1994.

[36] S. Rangachari, "Noise estimation algorithms for highly non-stationary environments," *M. S. Thesis*, Dept. of Electrical Engineering, University of Texas at Dallas, August 2004. Available online at www.utdallas.edu/~loizou/thesis/sundar_ms_thesis.pdf (accessed in October, 2005)

[37] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Processing Letters,* vol. 9, no. 1, pp. 12-15, January 2002.

[38] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, Vol. 81, No. 11, pp. 2403-2418, November 2001.

[39] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. Speech and Audio Processing*, Vol. 9, No. 5, pp. 504-512, July 2001.

[40] G. Doblinger, "Computationally efficient speech enhancement by spectral minima tracking in sub-bands," in *Proc 4th European Conf. Speech, Communication and Technology, EUROSPEECH'95*, Madrid, Spain, pp.1513-1516, 18-21 September 1995.

[41] B. Widrow, J. R. Glover, J. M. McCool, and J. Kaunitz, "Adaptive noise canceling: principles and applications," *Proc. IEEE*, vol. 63, pp. 1692-1716, December 1975.

[42] P. Lapsley, J. Bier, A. Shoham, E. Lee, *DSP Processor Fundamentals.* New Delhi: S. Chand & Company, 2000.

[43] *ADSP-BF533 Blackfin Processor Hardware Reference Manual,* Analog Devices, 2005.

[44] *VisualDSP++ 3.5 Getting Started Guide for 16-Bit Processors*, Analog Devices, 2005.

[45] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge, UK : Cambridge University Press, 1992.