Real-Time Implementation of Spectral Subtraction for Enhancement of Electrolaryngeal Speech

A dissertation
submitted in partial fulfillment of the
requirements for the degree of
Master of Technology

by

Bhaskara Rao Budiredla
(Roll No. 033307403)

under the supervision of

Prof. P. C. Pandey

Department of Electrical Engineering
Indian Institute of Technology, Bombay
July 2005

Indian Institute of Technology, Bombay

M.Tech. Dissertation Approval

Dissertation entitled "*Real-time Implementation of Spectral Subtraction for Enhancement of Electrolaryngeal Speech*", by *Bhaskara Rao Budiredla (Roll No 03307403)*, is approved for the award of the degree of *Master of Technology* in *Electrical Engineering* with specialization in *Communication Engineering*.

Supervisor : ....…… (Prof. P. C. Pandey)

Internal Examiner : ……..... (Prof. P. Rao)

External Examiner : ……..... (Dr. K. Samudravijaya)

Chairman : ……..... (Prof. A. K. Verma)

Date : 4th July, 2005

Bhaskara Rao Budiredla / Prof. P.C. Pandey (supervisor): “Real-time implementation of spectral subtraction for enhancement of electrolaryngeal speech”, *M.Tech. dissertation*, Department of Electrical Engineering, Indian Institute of Technology, Bombay, July 2005.

Abstract

Laryngeal cancers often necessitates complete surgical removal of the larynx, and thus it results in loss of the natural voicing source for speech production. The transcervical electro larynx is a prosthesis meant to provide vibrations for alaryngeal speech production. The main problem with this device is background noise generated by the device itself. Earlier work on the suppression of the background noise has shown that pitch synchronous application of spectral subtraction with quantile based noise estimation gives best results. The objective of this project is to implement a real time system in which the noisy sound is picked up by a microphone, processed for the removal of the background noise, and output through a speaker. Implementation has been done using a DSP board based on a 32-bit processor TMS320C6211. Offline processing has shown that at least 55 frames should be used in quantile based spectral subtraction for effective suppression of noise. Because of processing speed constraints, number of frames for dynamic estimation of noise is limited to 8, and phase spectrum during re-synthesis has been taken as zero. Real-time implementation provided a significant reduction without much noticeable speech degradation. However, optimization in program code and use of a faster processor may help in enhancing the noise reduction.

Acknowledgement

I would like to express my gratitude to my guide, Prof. P.C. Pandey for his invaluable guidance, support, and encouragement throughout the course of this project. I am also thankful to him for his sparing of invaluable time with me and for giving me a patient hearing. I sincerely thank Prof. V. M. Gadre for use of facilities in TI DSP Elite Lab.

I am thankful to Mr. Donald Pattie, Director of IT and Services, Heriot-Watt University, Scotland, who always replied to my queries on the Gmail DSP e-group. I would also like to thank Omkar Kulkarni, TI DSP Lab System administrator, for his support throughout this course of project. I also thank my colleague Narahari for many interesting and helpful discussions. I am also grateful to my lab-mates for their memorable and enjoyable company.

Bhaskara Rao Budiredla

July 2005

Contents

List of symbols and abbreviations

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
b_m	FIR filter coefficients
μ	Convergence parameter
α	Subtraction factor
β	Spectral floor factor
γ	Exponent factor
$L(k)$	Average magnitude spectrum of noise
N	Window length

Abbreviation	Explanation
SNR	Signal-to-noise ratio
TMS320C6x	TMS320C6000 family DSP processors
DSK	Digital starter kit
EDMA	Enhanced Direct Memory Access
EMIF	External Memory Interface

List of figures

Chapter 1

INTRODUCTION

1.1 Background

Speech production can be viewed as 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 [1]. 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 signal generated at a narrow constriction in the vocal tract. Laryngeal cancer often results in 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 [2], [3] is prosthesis meant to replace the natural larynx when this organ has been surgically removed. This device works similar to that of natural larynx, i.e. it provides vibrations, which are necessary for speech production.

Electronic artificial larynx or electrolarynx is the most widely used device. In this device, a pulse from the vibrating diaphragm, which rests against the throat, is 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 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. In addition to these, the major problem with the electrolarynx is the presence of acoustic energy from the vibrator, its interface with the neck, and the surrounding neck tissue. This background noise degrades the speech quality and intelligibility severely [4], [5], [6].

1.2 Project objective

The speech produced with electrolarynx is deteriorated due to presence of background noise. Source of background is leakage of vibrations from the transducer of vibrator. Effective shielding of the directly radiated sound from vibrator should reduce the

background noise. But, the use of acoustical shielding reportedly yielded only a marginal reduction in the noise [7].

In the past few years, investigations have been carried out for suppression of the background noise [8], [9], [10], [11]. Work in our lab has shown that spectral subtraction with quantile based noise estimation gives effective noise suppression [10], [11], [12], [16] [17]. The objective of this project is to implement a real time system in which the noisy speech is picked up by a microphone for the removal of the background noise, and then output it through a speaker.

1.3 Dissertation outline

Chapter 2 describes the mechanism of speech production using artificial larynx, its characteristics, and background noise generation model. Chapter 3 describes the literature survey and reviews various signal-processing techniques for the removal of the background noise. Chapter 4 describes the selection of a DSP processor and real-time implementation considerations. Chapter 5 describes the results. Summary and future plans are given in the last chapter.

Chapter 2

ELECTROLARYNGEAL SPEECH

In the past few years, research in the field of electrolaryngeal speech enhancement has focused on the suppression of additive noise [7], [10], [11]. The ultimate goal of speech enhancement is to eliminate the additive noise present in speech signal and restore the speech signal to its original form. Several methods have been developed as a result of these research efforts. Most of these methods have been developed with some or the other auditory, perceptual or statistical constraints placed on the speech and noise signals. However, in real world situations, it is very difficult to reliably predict the characteristics of the interfering noise signal or the exact characteristics of the speech waveform. Hence, in effect, the speech enhancement methods are sub-optimal and can only reduce the amount of noise in the signal to some extent.

This chapter presents review on the production of speech in humans and a literature review of the different electrolaryngeal speech enhancement methods used to date.

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 is shown in Fig. 2.1 [12]. Speech production can be viewed as 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 signal generated at a narrow constriction in the vocal tract for such sounds.

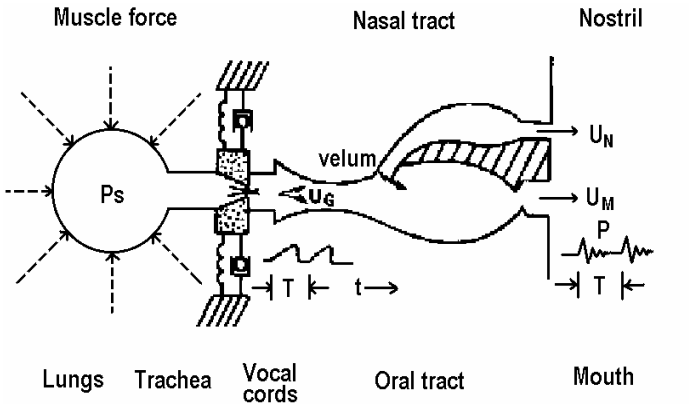


Fig. 2.1 Schematic of normal speech production [12].

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 need an alternative voicing source in order to speak. The artificial larynx [2], [3] 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 larynxes 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 complete description of pneumatic larynges can be found in [2], [3].

Electronic artificial larynx or electrolarynx is the most widely used device. In this device, a pulse from the vibrating diaphragm, which rests against the throat, is 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. 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. In addition to these, the major problem with the electrolarynx is the presence of acoustic energy from the vibrator, its interface with the neck, and the surrounding neck tissue. This background noise degrades the speech quality and intelligibility severely [4], [5].

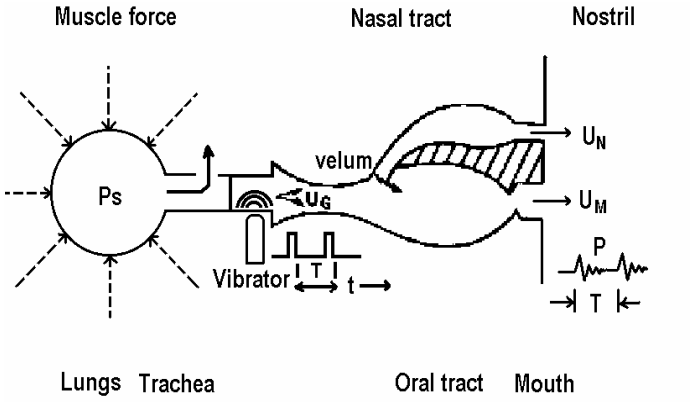


Fig. 2.2 Schematic of speech production with an electronic artificial larynx [12].

2.3 Characteristics of electrolaryngeal speech

Weiss *et al.* [4] reported a detailed study of perceptual, and the acoustical characteristics of electrolaryngeal speech, using the device Western Electric Model 5. The report suggested that the radiated source 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.

The speech produced using an electrolarynx is also deficient in low-frequency spectrum and this can be explained as follows. The mechanical portion of the electrolarynx can be modeled as a single oscillator driven by periodic mechanical impulses [6]. The spectrum of the acoustic signal generated by such an oscillator will

have a resonance frequency that is determined by the ratio between the stiffness (*K)* and the mass (*M)* of the vibrator. Since the ratio K*/M* is typically large for effective mechanical coupling with the input, it is likely that the resonant frequency of such an oscillator will be much higher than the fundamental frequency of the driving force. Hence, the spectral amplitudes of such an oscillator would be expected, at best, to be separable, but more likely to be below those of higher frequency components of the output. Another reason is that acoustic energy is inversely proportional to frequency. Thus, low-frequency components of excitation signals generated by electrolarynx would be expected to be reduced in the transformation process.

2.4 Background noise model for electrolaryngeal speech

The main problem in electrolaryngeal speech is background noise generated by the device itself. The reason for this is one side of transducer of the vibrator is coupled to the neck tissue but the other side gets coupled to the surrounding atmosphere. It results in direct leakage of sound that produces background interference. A model of the leakage sound generation during the use of electrolarynx is shown in Fig. 2.3. 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, $l(t)$, gets added to the useful speech, $s(t)$, and deteriorates its intelligibility.

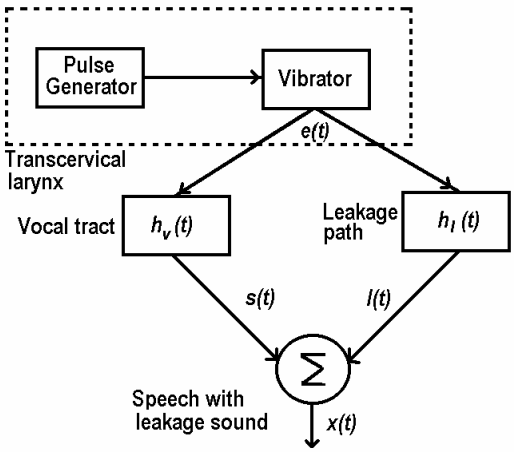


Fig. 2.3 Model of background noise generation in electrolaryngeal speech [10]

Chapter 3

BACKGROUND NOISE CANCELLATION IN ELECTROLARYNGEAL SPEECH

Investigations have been carried out for studying the effect of acoustical shielding to reduce the background noise, and some improvement was obtained by applying a one inch thick foam shield around the larynx [7]. This 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. So the impracticality of acoustic shielding techniques and their limited effectiveness led to consider the use of signal processing techniques.

3.1 Enhancement of electrolaryngeal speech by adaptive filtering

Espy-Wilson *et al.* [7] tried to remove the background noise by simultaneously recording the output at both the lips and the electrolarynx, and then employing the signals in an adaptive filtering algorithm. 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 equations of LMS algorithms are as given below.

The FIR filter output is given as

$$y(n) = \sum_{m=0}^{N-1} b_m(n) r(n-m) \tag{2.1}$$

The error is given as

$$e(n) = x(n) - y(n) \tag{2.2}$$

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

$$b_m(n) = b_m(n-1) + \mu e(n) r(n-m), \quad m = 0 \ldots N\text{-}1 \tag{2.3}$$

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 certain value results in instability of the algorithm.

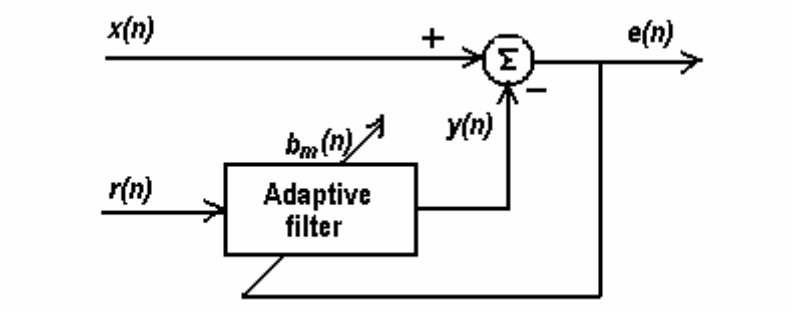


Fig. 3.1 Two input adaptive filter

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 amplitude 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. The basic principle of the spectral subtraction method is to subtract the power spectrum of noise from that of the noisy speech. The noise is assumed to be uncorrelated and additive to the speech signal. An estimate of the noise signal is measured during silence or non-speech activity in the signal.

In the spectral subtraction method for enhancement of noisy speech, basic assumption made 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, as shown in Fig. 2.3 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. 2.3, 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

$$x(n) = s(n) + l(n) \tag{3.1}$$

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

$$s(n) = e(n) * h_v(n) \tag{3.2}$$

$$l(n) = e(n) * h_l(n) \tag{3.3}$$

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

$$X_n(e^{j\omega}) = E_n(e^{j\omega})[H_{vn}(e^{j\omega}) + H_{ln}(e^{j\omega})] \tag{3.4}$$

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

$$|X_n(e^{j\omega})|^2 = |E_n(e^{j\omega})|^2[|H_{vn}(e^{j\omega})|^2 + |H_{ln}(e^{j\omega})|^2] \quad (3.5)$$

If the short-time spectra are evaluated using 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

$$|X_n(e^{j\omega})|^2 = |L_n(e^{j\omega})|^2 = |E_n(e^{j\omega})|^2 |H_{ln}(e^{j\omega})|^2 \quad (3.6)$$

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. 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.

$$|Y_n(k)|^2 = |X_n(k)|^2 - |L(k)|^2 \quad (3.7)$$

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

$$Y_n(k) = |Y_n(k)|\, e^{j\angle Xn(k)} \quad (3.8)$$

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

$$y_n(m) = IFFT[(Y_n(k)] \quad (3.9)$$

3.3 Drawbacks of spectral subtraction

While the spectral subtraction method is easily implemented and effectively reduces the noise present in the corrupted signal, there exist some shortcomings, which are given 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 are forced to use an average 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 time 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 is a difficult and 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 basic technique, especially the presence of musical noise. This section deals with the different techniques and enhancements that have been proposed over the years.

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

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

$$\begin{aligned} |Y'_n(k)|^\gamma &= |Y_n(k)|^\gamma \quad \text{if } |Y_n(k)|^\gamma > \beta|L(k)|^\gamma \\ &= \beta|L(k)|^\gamma \quad \text{otherwise} \end{aligned} \tag{3.11}$$

where α is the subtraction factor and β is the spectral floor factor. As in case of Equation (3.8), 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 [9]. The quality of enhanced speech was inferior, compared to that with the modified spectral subtraction.

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.* [10] have used a simple method which avoids calculation of the phase spectrum $\angle X_n(k)$, by modifying equation (3.8) as follows.

$$\begin{aligned} Y'_n(k) &= |Y'_n(k)|\, e^{j\angle Xn(k)} \\ &= |Y'_n(k)|\, X_n(k) / |X_n(k)| \end{aligned} \tag{3.12}$$

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

$$y_n(m) = \text{IFFT}(Y'_n(k)) \tag{3.13}$$

With this change, the algorithm becomes computationally efficient; as there is no complex calculations involved so 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

Bhandarkar [10], [11] has implemented modified spectral subtraction algorithm for enhancement of the electrolaryngeal speech. A schematic of the modified spectral subtraction algorithm is shown in Fig. 3.2. He carried out investigations for establishing the window size and optimal values of α, β, and γ using electrolarynx model NP-1 manufactured by NP Voice, India. 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, 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. He found that background noise is totally removed, but there is a small amount of musical noise present in the output. As a future work, he suggested the study of window positioning for improvement of processed speech.

3.6 Enhancement of electrolaryngeal speech by extended spectral subtraction

Pratapwar [12] has implemented extended spectral subtraction algorithm for enhancement of the electrolaryngeal speech. He carried out the investigations, as suggested by Bhandarkar, for placing and length of the window. 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. A schematic of the extended spectral subtraction algorithm is shown in Fig. 3.3.

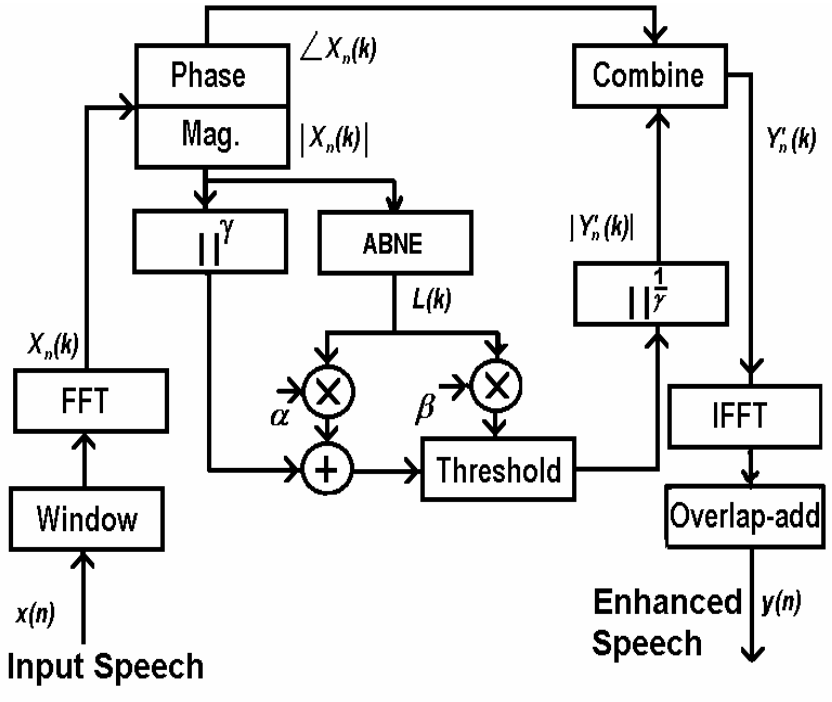


Fig. 3.2 Block diagram of modified spectral subtraction algorithm [11]

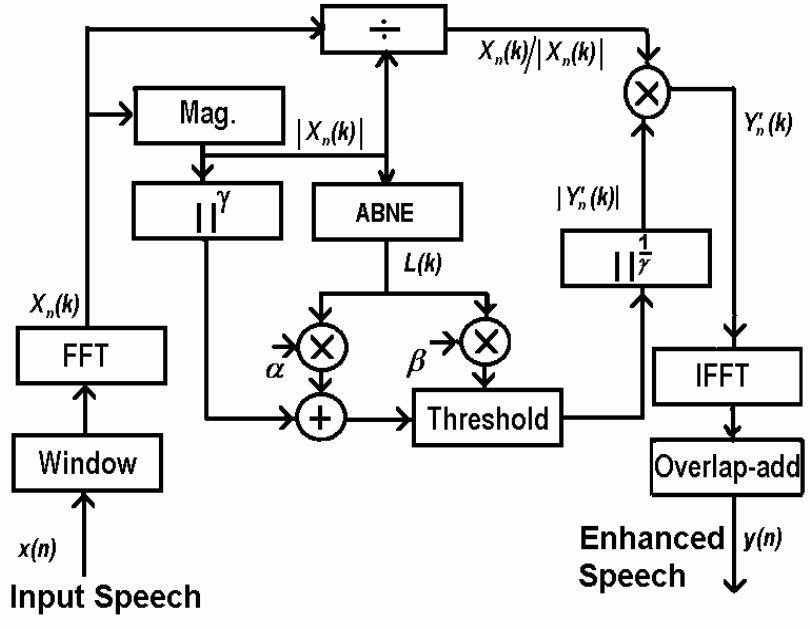


Fig. 3.3 Block diagram of extended spectral subtraction algorithm [12]

With the spectral subtraction algorithm described above, background noise is totally removed, but there is a small amount of musical noise affecting the perceptual quality of the processed speech. The following section describes the investigations carried out by Pratapwar [12] for average based noise estimation for spectral subtraction.

3.7 Enhancement of electrolarngeal speech by quantile based spectral subtraction

In the spectral subtraction algorithm implemented, the noise was assumed to be stationary in the enhancement mode. But actually the background noise varies because of variations in the place of coupling of vibrator to the neck tissue, and the amount of coupling. This results in variations in the effectiveness of noise enhancement over an extended period. Hence a continuous updating of the estimated noise spectrum is required. Recursive averaging of spectra during silence segments may be used for noise spectrum estimation. However, speech/silence detection in electrolaryngeal speech is rather difficult. Quantile-based noise estimation (QBNE) [13], [14], [15], [17] technique does not need speech/non-speech classification and can be used for noise estimation in electrolaryngeal speech [16].

Pratapwar [16], [18] investigated the use of QBNE for continuous estimation of noise spectrum in electrolaryngeal speech [12], [16]. 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, frequency dependent quantile values and SNR based dynamic selection of quantile values. In the first two methods, quantile values once selected remain constant during entire speech enhancement mode. Because of this the leakage noise characteristics change slowly with the application of the vibrator and the configuration of the vocal tract. So 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 was investigated. The dynamic selection of quantile values is based on the estimation of different quantiles for different frequencies. It was found that, QBNE with SNR based quantiles showed better speech quality [18].

3.8 Proposed real-time noise canceller

Fig. 3.4 shows schematic of real time implementation of signal processing algorithm for reduction of self-leakage noise in electrolaryngeal speech. Electro laryngeal speech is recorded by using microphone. Excitation pulse from the pulse generator circuit is given to the signal processor. Recorded speech signal is converted into digital signal and processed using signal processing algorithm. Processed speech is converted into analog signal and given to speakers.

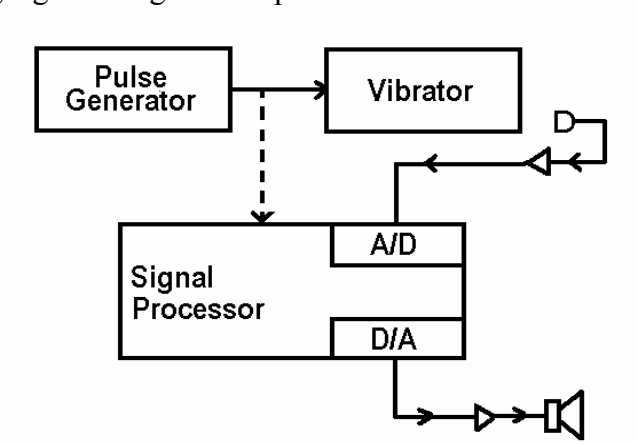
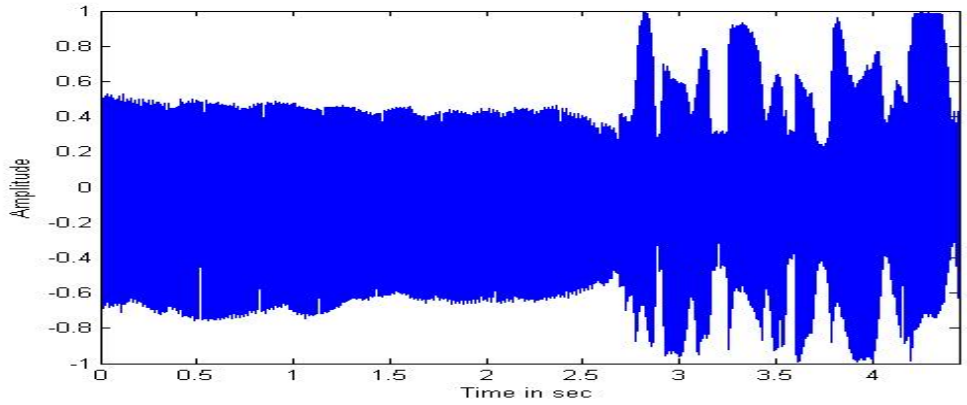


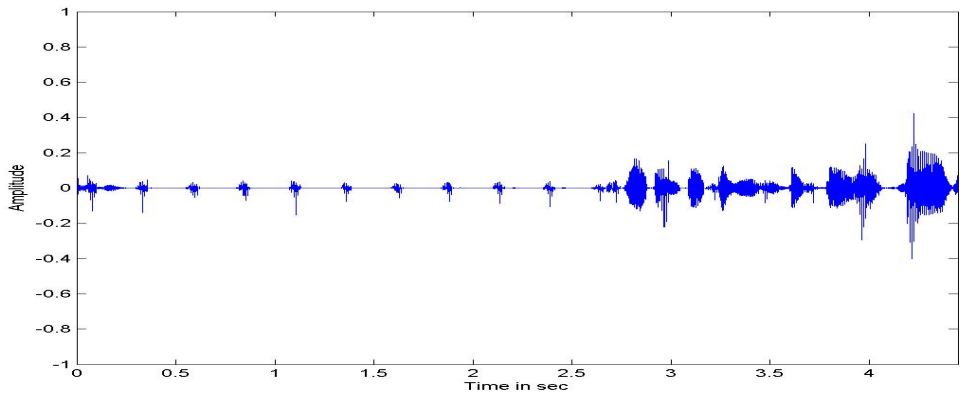
Fig. 3.4 Real-time implementation scheme of noise cancellation with spectral subtraction algorithm [10].

3.9 C code implementation

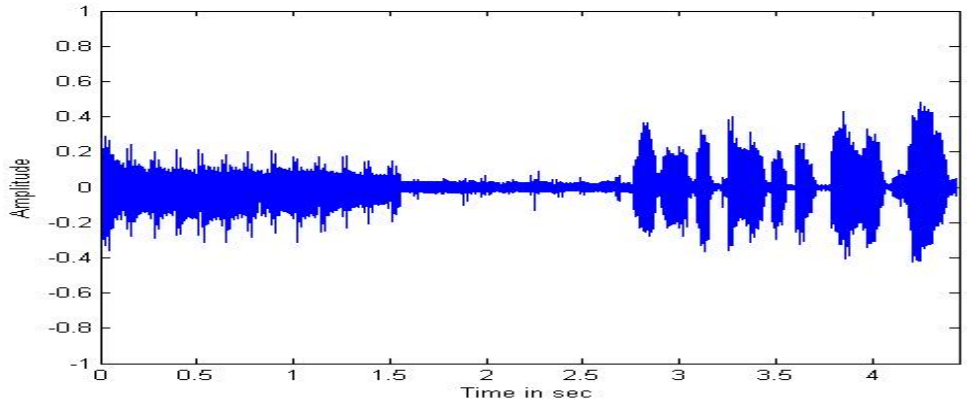
The earlier work in our lab [10], [12] has shown that spectral subtraction with quantile based noise estimation gives effective noise suppression. Real-time implementation of this algorithm can be done using TI DSP, TMS320C6211 DSK. As a first step towards this objective, the earlier MATLAB based ABNE and QBNE programs have been converted to C code for pitch synchronous spectral subtraction. The results have been tested and validated. For evaluation of the implemented C algorithms, a recorded sentence with a sampling frequency of 11.025 ksa/s was used. Recording was done using electrolarynx model “Servox Digital” (manufactured by Servox AG, Germany). Fig. 3.5 and Fig. 3.6 shows unprocessed speech and processed speech signals. The C code has been written considering the memory requirements of the DSP processor. So, while writing the code all the variables were declared as lower precision data types (int and short int). This resulted in degraded performance in case of C code implementation.



(a) Recorded speech waveform

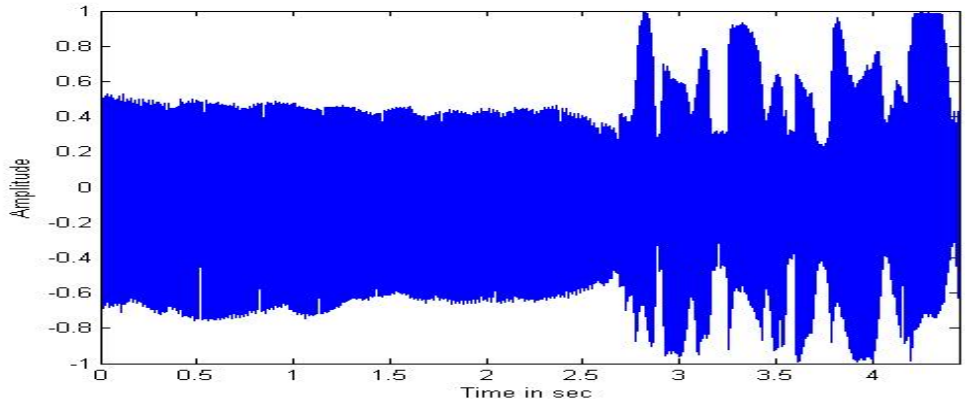


(b) Signal enhanced using Matlab based implementation

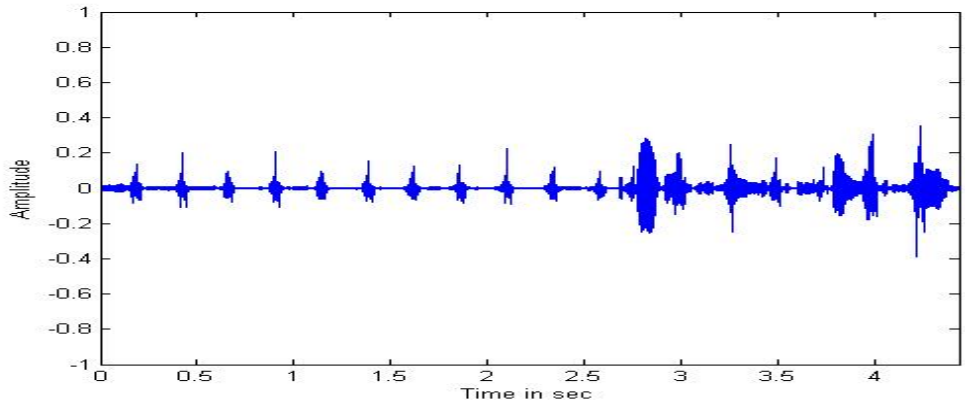


(c) Signal enhanced using C language based implementation

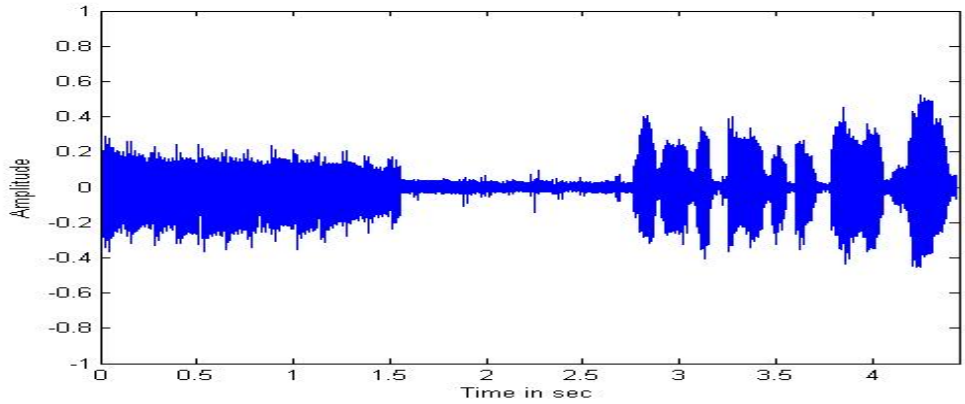
Fig. 3.5 Recorded and enhanced speech using pitch synchronous spectral subtraction algorithm with ABNE. Speaker SP, material: question-answer pair in English, " *What is the time? It is 5'O clock*", generated using Servox electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 2$.



(a) Recorded speech waveform



(b) Signal enhanced using Matlab based implementation



(c) Signal enhanced using C language based implementation

Fig. 3.6 Recorded and enhanced speech using pitch synchronous spectral subtraction algorithm with QBNE. Speaker SP, material: question-answer pair in English, "*What is the time? It is 5'O clock*", generated using Servox electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 2$.

Chapter 4

REAL-TIME IMPLEMENTATION ON A DSP BOARD

While choosing a DSP, for real-time implementation of a signal processing technique, 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, as explained by Lapsley *et al.* [19]. For implementation of QBNE based pitch synchronous spectral subtraction for enhancement of electrolaryngeal speech, it was decided to use a DSP board based on TMS320C6211 digital signal processor from Texas Instruments (TI). The choice was primarily dictated by availability of resources and development tools. This chapter provides a description of the DSP processor, software tools, and the DSP board used for the real-time implantation.

4.1 TMS320C6x DSPs

The TMS320C6x generation of DSPs is a part of TMS320 family of DSPs from TI. The TMS320C62x devices are fixed point DSPs, while the TMS320C67x devices are floating point DSPs in the TMS320C6x generation. The TMS320C62x and TMS320C67x are code compatible and both use VelociTI architecture [20].

4.1.1 VelociTI architecture

The C6x is based on the VLIW (Very Large Instruction Word) architecture. In such architecture, several instructions are fetched and processed simultaneously. The instructions that are fetched simultaneously constitute what is called a fetch packet (FP). In C6x architecture, 8 instructions are fetched simultaneously on a 256-bit wide program data bus. Thus the FP of C6x consists of 8 instructions. The original VLIW architecture has been modified by TI to allow several Execute Packets (EP) to be included within the same FP. An Execute Packet is a group of instructions that are executed simultaneously. An EP of C6x may thus contain a single instruction or from 2 to 8 instructions. This VLIW modification done by TI is called VelociTI. As

compared to VLIW, VelociTI architecture reduces code size and increases performances when instructions reside off-chip.

4.1.2 Hardware description

Fig. 4.1 and Fig. 4.2 give an overview of C6x architecture and Functional block diagrams respectively. A C6x CPU has two register files (A and B) each containing 16 general purposes registers, 8 functional units (.L1, .L2, .S1, .S2, .M1, .M2, .D1, .D2), two load-from-memory paths (LD1, LD2), two store-to-memory paths (ST1, ST2) and two-register file cross paths (1X, 2X).

The general-purpose registers are 32-bit registers (A0 to A15, B0 to B15). These support 32-bit and 40-bit fixed-point data. The 32-bit data can reside in any one register. When 40-bit data are to be stored, these registers need to be used in pairs. These are sixteen valid pairs of registers for 40-bit data viz - A15:A14, A13:A12, A11:A10, A9:A8, A7:A6, A3:A2, A1:A0 and similarly for the B- register file. The 32 LSBs are stored in the even numbered register and 8 MSBs are stored in the 8 LSBs of next upper register (which is always an odd register).

The so called .M unit is used for multiplication operation, .L unit is used for logical and arithmetic operations, .S unit is used for branch, bit manipulation and arithmetic operations and .D is used for loading/storing and arithmetic operations.

The data from memory can be addressed as a byte (8-bits), half-word (16-bits) or a word (32-bits). For a discussion on the data alignment requirements of the C6x, the reader is referred to the book by [19].

The C62x and C67x share an instruction set. All of the instructions valid for C62x are also valid for C67x. However, since the C67x is a floating-point device, there are some instructions that are unique to it and do not execute on fixed-point device. For a detailed description of the instruction set, refer TMS320C62x/C67x CPU and Instruction Set Reference Guide from TI [22].

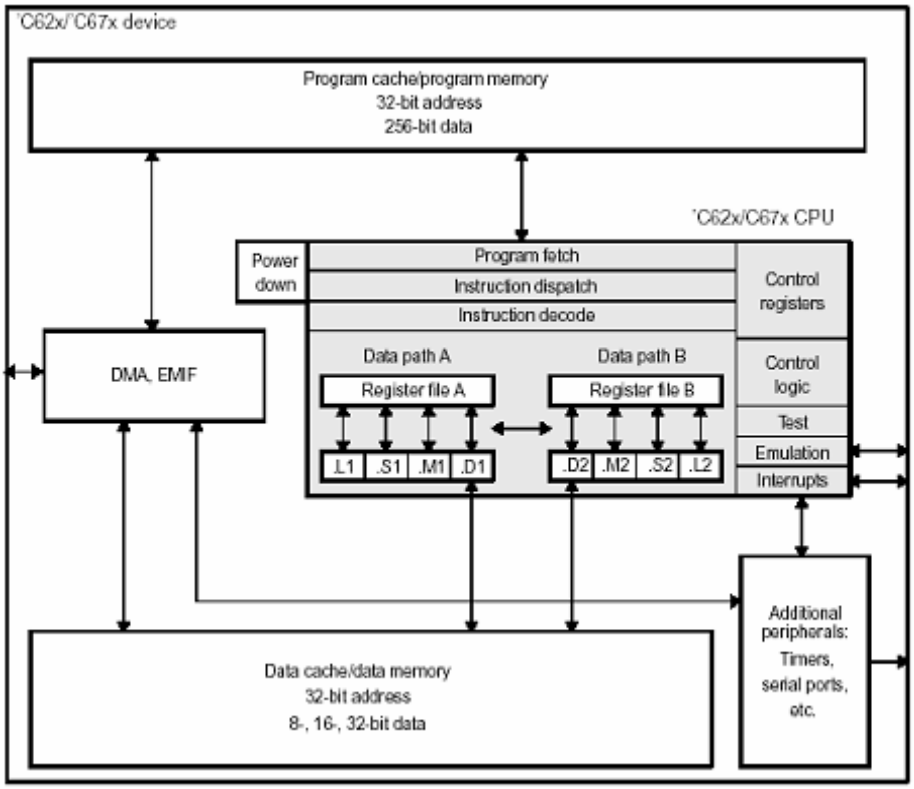


Fig. 4.1 TMS320C6x architectural overview [22]

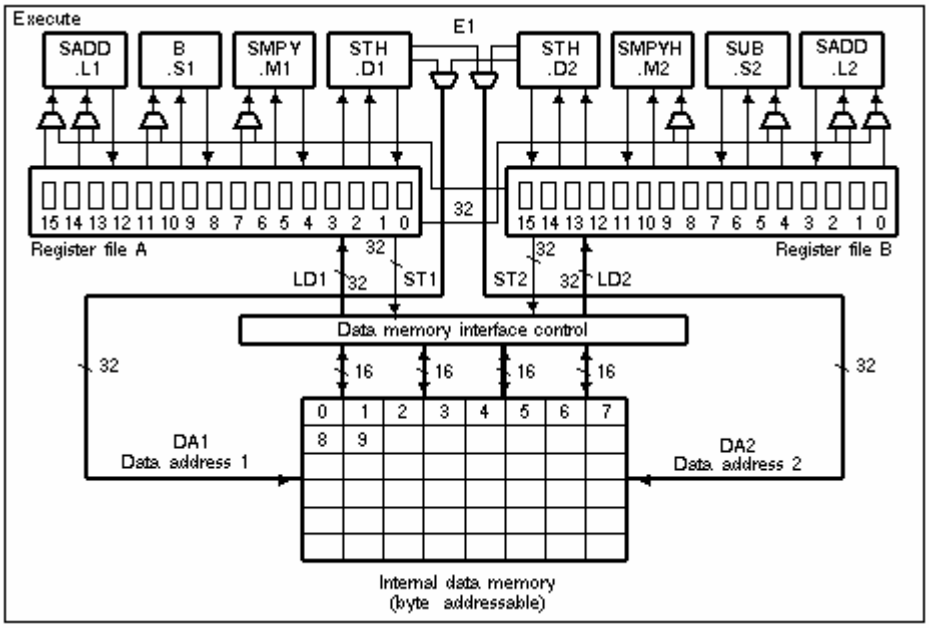


Fig 4.2 Functional block diagram of TMS320C6x [22]

4.1.3 The code composer studio

The programming for the most DSPs can be done either in assembly or C. Although writing programs in C would require less effort, the efficiency achieved by doing so is less than that achieved by writing the program in assembly. For TMS320 DSPs, the programming can be done in assembly, linear assembly or C. Linear assembly is a good compromise between C and assembly. The software tools for converting these source files (.C for C, .asm for assembly, .sa for linear assembly) into code executable on DSP includes compiler, linker and debugger/simulator. When a DSP system (an Evaluation Module or a DSK board) is not available, the simulator comes handy in simulating the execution of a code on C62x or C67x processor.

The CCS comes with all these above mentioned software tools. It is a software tool that provides an easy to use Graphical User Interface (GUI) for configuring, building, interfacing and debugging purposes [23]. It also has real time capabilities such as program tracing, performance monitoring and file streaming which help in analyzing a program without disrupting its real-time behavior. The CCS provides a better way to load, run and debug a code on the DSP system via its GUI.

4.2 DSP board

For implementation work, the TMS320C62x based DSP boards were used. The TMS320C62x DSP devices are fixed point devices from Texas Instruments (TI). There are many DSPs in this family, e.g. C6201, C6202, and C6211. The instruction set is same for all C62x devices. There are only a few differences such as amount of on-chip program and data memories, number of general-purpose registers etc.

In order to enable a user to evaluate the suitability of a TMS DSP for his/her application, TI provides two kinds of “DSP systems”-the DSP Starter Kit (DSK) and the Evaluation Module (EVM). The DSK has a C6x DSP with on-board peripherals such as codec, memories, voltage regulators and an interface with a host PC. The EVM is a more elaborate system. In addition to all these features, the EVM has some extra features such as on board JTAG emulator.

Most of the implementation work has been done on the TMS320C6211/6711 DSK board. Experimentation was carried out in order to validate the system and, evaluate its performance.

4.3 Implementation using TMS320C6211 DSP board

The real-time implementation set up was done as shown in Fig 4.3. The peripherals and their registers initializations and settings that are necessary for the implementation of the algorithm are explained in this section. The explanation includes analog interface, McBSP, EDMA controller, and the DSP processor.

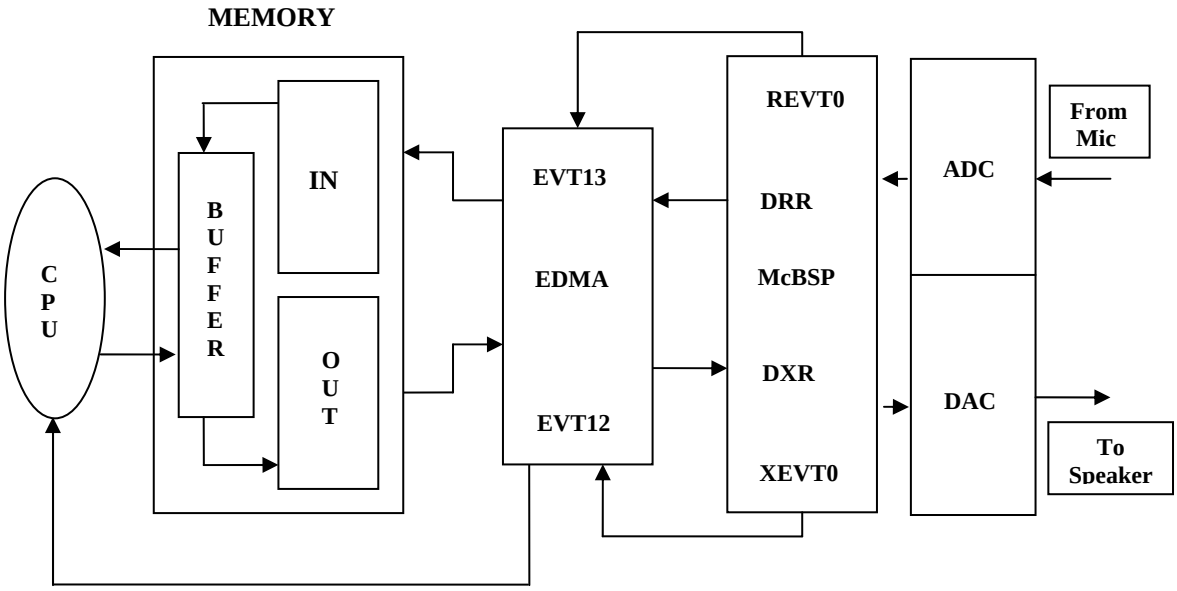


Fig. 4.3 Real-time implementation scheme of noise cancellation with QBNE based spectral subtraction algorithm

4.3.1 Analog interface

The DSP starter kit provides a AD535 CODEC (which has inbuilt ADC and DAC) as an interface to real world signals, which are analog signals (voice). ADC samples the data coming from microphone and makes it available to Multi-channel Buffered Serial Port (McBSP) as 16-bit word. DAC, on the other hand takes 16-bit word samples from McBSP, reconstructs it in an analog signal and outputs to speaker. Both ADC and DAC operate at fixed sampling rate of 8 ksa/s. The properties of the CODEC were set by writing into its control register [20].

Features

1. Independent voice and data channels with a sampling rate of 8 ksa/s.
2. 16-bit signal processing.
3. Maximum microphone bias of 5 mA at 2.5v or 1.5v.
4. Programmable gain amplifiers.

Settings

1. The analog interface was operated at 0 dB at both the ADC and the DAC.

2. Voice channel was used for transmission and reception of analog data.

4.3.2 McBSP interface

The McBSP consists of a data path and a control path, which connect to external devices. Data are communicated to these external devices via separate pins for transmission and reception. Control information (clock and frame synchronization) is communicated via four other pins. The processor communicates to the McBSP via 32-bit-wide control registers accessible via the internal peripheral bus [20]. In this project, McBSP is used to communicate between the CODEC and EDMA. This peripheral helps in performing serial-to-parallel and parallel-to-serial arrangement of the data points. By setting proper registers McBSP can give interrupts to EDMA (REVT & TEVT), whenever data are available for reading by EDMA and whenever it is ready to accept data from EDMA respectively. This helps in decreasing load on CPU. McBSP reads data from ADC and organizes them in 16 bit/frame (set in XCR & RCR) and then sends them to memory location specified by EDMA.

Features

1. The McBSP consists of both a data path and control path.
2. Double-buffered data registers, which allow a continuous data stream.
3. Full-duplex communication.
4. Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected analog-to-digital (A/D) and digital-to-analog (D/A) devices.
5. A wide selection of data sizes, including 8, 12, 16, 20, 24, and 32 bits.

Settings

1. DXR, Data Transmit Register: DXR contains the data received from the receive buffer register and transfers it to the CPU or the EDMA controller, as the case may be. Here it was set to transfer to the EDMA. The DXR is mapped on to the following locations:
    a. McBSP 0: 018C 0000
    b. McBSP 1: 0190 0000
2. DRR, Data Receive Register: DRR contains the data from the CPU or EDMA controller as the case may be, which is to be transmitted to the data transmit pin through the transmit shift register. Here it was set to receive from the EDMA. The DRR is mapped on to the following memory locations.

a. McBSP 0: 018C 0004h
b. McBSP 1: 0190 0004h

3. SPCR, Serial Port Control Register: This controls the parameters like frame generation, transmitter and receiver parameters, synchronization settings etc. The SPCR was mapped on to the memory location 018C 0008h.
4. PCR, Pin Control Register: This was used to set the clocks for transmit/receive, polarity of clocks etc. Since an external clock drives the port in our case, we set this register to 0.
5. SRGR, Sample Rate Generation Register: This register was used to set the clock frequency at which the input serial data are sampled. This register was used when we want the clock to be an internal signal and not driven by an external source. We do not use it in our program as we are not using internal clock generation. The SPCR is mapped on to the memory location 018C 0014h.

4.3.3 EDMA controller

The Enhanced Direct Memory Access (EDMA) controller transfers data between regions in the memory map without intervention by the CPU [20]. The EDMA controller allows movement of data to and from internal memory, internal peripherals, or external devices to occur in the background of CPU operation. The EDMA controller has four independent programmable channels, allowing four different contexts for EDMA operation. In particular, in this project, EDMA was used to accept input data from McBSP receiver register, DRR, and store it in a predetermined memory location. At the same time, it keeps on outputting data from another memory location to the McBSP transmit register, DXR.

Features

1. 16 channels with programmable priority, and the ability to link and chain data transfers.
2. The EDMA allows movement of data to/from any addressable memory spaces, including internal memory (L2RAM), peripherals, and external memory.
3. Transfers of size up to 65535 elements in an array and 65535 arrays in a block are possible.

4. Transfers can be both CPU and EVENT initiated with parameters loaded as 6 word entries in parameter RAM.

Settings

1. Two channels were used, one for noisy speech reception from the microphone (MIC), input pin of the analog interface, and other for purified speech transmission to the speaker (SPKRS), output pin of the analog interface.
2. Transfer size with elements =FFT frame size chosen=256 elements.
3. EVENTS 12 and 13 from REVT and XEVT were used for data transfers.

4.3.4 TMS320C6211 DSP processor

The TMS320C6211 peripheral set has up to 32 interrupt sources [20]. The CPU, however, has 12 interrupts available for use. The interrupt selector allows us to choose and prioritize which 12 of the 32 interrupts our system needs to use. Here, in this project, we used INT8, (interrupt from EDMA) for the CPU. As soon as block of data are transferred by the EDMA, this interrupt is activated. Once the EDMA controller places the input data in the memory location specified by the CPU, the CPU reads the input data block, processes it for the removal of background noise and finally places it at the transmitting location.

Features

1. Eight highly independent functional units (including six ALUs and two multipliers).
2. Up to eight 32-bit instructions per cycle.
3. Byte addressable (8-, 16-, 32-bit data).
4. L1/L2 memory architecture (4K-byte program cache, 4K-byte data cache and 64K-byte L2 unified map RAM/cache).
5. 32-bit external memory interface.

Settings

1. The CPU was operated with one interrupt from EDMA enabled at the end of data transfers.
2. Peripherals used by the processor were McBSP, EDMA, EMIF.

4.3.5 Implementation details

The programming for the 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 C6000 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.

Since the total code has been written in C language, the compiler takes care of actual number representation. By including the file `6211DSK.h` or `6711DSK.h`, we can specify the target processor, either fixed point processor or floating point processor. With floating point operations, we generally do not have to be concerned with the problems of overflow during computation. However, for program execution on 6211 processor, it was decided to mostly use the fixed-point operations. A detailed simulation with actual signals is needed to limit the scaling to the values actually needed. But such implementation should be done along with saturation logic. In our implementation, simulation with actual signal was not carried out and all the scaling factors have been selected on the basis of worst case overflow estimates. Hence all the arithmetic operations should be properly organized 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 an n-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 $2n$ bit number. To make the resulting number a n-bit, we have to right shift the number n-bits. It is to be noted that this implementation, concerned primarily with avoiding overflows, may result in significant number of underflows.

In the spectral subtraction process, we are using signal power spectrum. From the FFT routine, we get the real and imaginary parts of the spectral samples as 16-bit integers.

$$X_m(k) = X_{mR}(k) + jX_{mI}(k) \tag{4.1}$$

and hence the squared magnitude spectrum is given as

$$\left|X_m(k)\right|^2 = X_{mR}(k)X_{mR}(k) + X_{mI}(k)X_{mI}(k) \tag{4.2}$$

These will be 32-bit values which cannot be carried forward in the processing. Hence squared magnitude spectral values are normalized by multiplying with 2^{-16}, to get

$$P_{xm}(k) = \left|X_m(k)\right|^2 2^{-16} \tag{4.3}$$

The same definition of normalized power spectrum, as 16-bit integer, is used for both the signal and the noise; and hence no renormalization is needed. The actual sample sequence consists of 178 samples (F_0=90.3 Hz, f_s=8 ksa/s) or 244 samples (F_0=90.3 Hz, f_s=11.025 ksa/s). This sequence is padded with zero valued samples to block size of 256, and 256-point FFT is obtained, by calling the subroutine "`radix2.h`" from "DSP" library routines. While calling this function we have to send the FFT coefficients as parameters. So, the real part, W_R, and the imaginary part, W_I, of the coefficients were generated as 16-bit integers

$$W_R(n) = 32767[-\cos(\frac{2\pi n}{N})] \tag{4.4}$$

$$W_I(n) = 32767[-\sin(\frac{2\pi n}{N})] \tag{4.5}$$

and stored in the file "`coefficients.h`". The power spectrum during the first 8 windows is averaged to get an initial estimate of noise power spectrum. Hence the equation for this estimate is

$$P_L(k) = [\sum_{m=0}^{7} P_{xm}(k)]/8 \tag{4.6}$$

Once the average power spectrum of the noise is obtained, we can adaptively estimate the noise in each subsequent frame using dynamic QBNE based on SNR, ρ_m, where the quantiles in the m^{th} frame are calculated by

$$q_m(k) = \frac{[q_1(k) - q_0(k)][\rho_m(k) - \rho_0(k)]}{[\rho_1(k) - \rho_0(k)]} + q_0(k) \tag{4.7}$$

where, $q_0(k)$ and $q_1(k)$ are the fixed quantiles obtained during silence and noisy speech averaged over long segments. SNR $\rho_m(k)$ is the ratio of power spectrum of noisy speech in the m^{th} frame to the average power spectrum of noise expressed in dB. SNR $\rho_1(k)$ is the ratio of average power spectrum of noisy speech to the average power spectrum of noise (used for calculating the matched set of quantiles $q_1(k)$).

SNR $\rho_0(k)$ is the SNR in the absence of speech, and hence it is 0 dB. Therefore Equation 4.7 can be reduced to

$$q_m(k) = \frac{[q_1(k) - q_0(k)]\rho_m(k)}{\rho_1(k)} + q_0(k) \tag{4.8}$$

where, $\rho_m(k)$ are obtained by using

$$\rho_m(k) = 10\log\frac{P_m(k)}{P_L(k)} \tag{4.9}$$

We calculate the log by calling in built routine "`log.h`" which provides the result with a scaling by 2^8 as 16-bit integer, and hence we get

$$\rho_{m8}(k) = 2^8\rho_m/10 \tag{4.10}$$

Because of real-time processing constraints, number of search operations should be limited. Hence the values of $q_0(k)$, $q_1(k)$ and $\rho_1(k)$ are empirically taken from MATLAB implementation and stored in a file "`quantiles.c`". Hence Equation 4.8 can be reduced to

$$q_m(k) = r(k)\rho_m(k) + q_0(k) \tag{4.11}$$

where,

$$r(k) = \frac{[q_1(k) - q_0(k)]}{\rho_1(k)} \tag{4.12}$$

Thus we need to feed the values of $r(k)$ and $q_0(k)$. For noise estimation with 8 frames the values of $q_0(k)$ and $q_1(k)$ in Equation 4.8 are integers in the range 1-8. Hence we can store them as integers without any scaling. Therefore, we can scale $r(k)$ as

$$R(k) = \frac{[q_1(k) - q_0(k)]}{\rho_1(k)}2^8 \tag{4.13}$$

Now we can calculate the quantile values as

$$q_m(k) = R(k)\rho_{m8}(k)\left[\frac{10}{2^{16}}\right] + q_0(k) \tag{4.14}$$

$$= R(k)\rho_{m8}(k)\left[\frac{1}{2^{13}}\right] + q_0(k) \tag{4.15}$$

The quantiles thus obtained, have to be verified for the minimum and maximum quantiles. This can be accomplished by

$$q_m(k) = q_0(k) \text{ if } q_m(k) < q_0(k) \tag{4.16}$$

$$= q_1(k) \text{ if } q_m(k) > q_1(k)$$

The power spectrum values corresponding to the quantiles, $Q_m(k)$, are taken from the last stored 8 frames, and the cleaned power spectrum of the speech in the m[th] frame can be obtained as

$$Y_m(k) = P_m(k) - 2P_L(k) \tag{4.15}$$

$$Y_m(k) = Y_m(k), \text{ if } Y_m(k) > \beta P_L(k) \tag{4.16}$$

$$= \beta Y_m(k), \text{ otherwise}$$

where, $Y_m(k)$ is the cleaned output power spectrum, α is the spectral subtraction factor, and β is the spectral floor factor. We have used α = 2.

In Equation 4.16, all the variables are fixed point 16-bit numbers except β, which is declared as 0.001. We can replace the value 0.001 value by 2^{-10}, which is implemented by simply right shifting the data by 10.

From the output power spectrum, the output magnitude spectrum is obtained by calling the DSP library subroutine "`sqrt.h`". The output time domain signal, $y_m(t)$, is obtained by setting the phase to zero, i.e. by setting the imaginary part of the output magnitude spectrum and taking the magnitude part on the real part. IFFT of this complex spectrum gives the re-synthesized speech signal for the segment.

Finally, the 50% overlapping of the frames has to be addressed. For explanation purpose, only inputting of the data from ADC to internal memory is explained. The same is applied for outputting the data form internal memory to DAC. For pitch synchronous application with F_0 = 90.3 Hz, and, f_s = 8 ksa/s, the frame length corresponding to 178 samples. These 178 samples have to be padded with 78 zeros and 256 point FFT has to be obtained for processing. So with 50% overlap for each frame we need to transfer only 89 samples from the ADC to internal memory and the remaining 89 samples are taken from the last frame. To accomplish this, two 256 integer sized buffers, (Ping_Rx and Pong_Rx), are used. Each buffer is partitioned into 3 segments. The first segment is from the location 1[st] to 89[th] location. The second segment is from 89[th] to 178[th] location. The third segment is from 179[th] to 256[th] location. Since zeros has to be padded at the end of 178 samples, the last 78

locations of the two buffers, i.e. the third segment, are set to zero. For the first time, let us assume that the first segment of the Ping_Rx is all zero values. Now the DMA takes the input data of 89 samples and puts in the second segment of Ping_Rx buffer. Once it completes the transfer of 89 input data samples into Ping_Tx, it starts filling the Pong_Tx buffer into its second segment. But before doing that, it is needed to transfer the second segment of the Ping_Tx buffer to first segment of the Pong_Tx buffer. This is done because for the 50% overlap, we need the last 89 data samples of the previous frame. Thus, once DMA starts transferring the data into Pong_Tx, CPU is free to process the Ping_Tx frame. Now DMA completes the data transfer to Pong_Rx, and starts transferring the data to Ping_Rx, and again CPU is ready for processing Pong_Rx. This process is repeated for the whole time of processing.

The real-time implementation details of the total algorithm using C language are given below in Table 4.1.

Table 4.1 Real-time implementation details (using C language)

Subroutine	**Code Size (Bytes)**	**Number of Instructions**	**Execution Time (ms) (Instruction cycle time=40 ns)**
FFT (N=256)	1440	1,21,548	4.9
Quantiles Calculation	320	49,862	2
Total Algorithm	3280	3,98,896	16

Chapter 5

RESULTS AND DISCUSSION

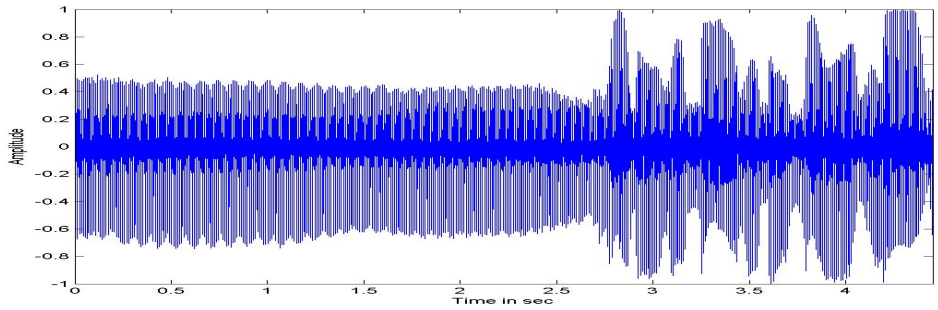
For the verification of the implemented real-time algorithm a recorded question-answer pair in English was used. The recorded sentence is '*What is the time? It is 5'O clock'*. The recording was done using "Servox Digital" electrolarynx (with pitch set to 90.3 Hz). The total duration of the recording was ≈ 4 s. The first 2 s 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. During the other 2 s, the speaker uttered the sentence pair.

The 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 the MATLAB and C based implementations. For real-time implementations the signals were output by the DAC conversion at the PC sound card and given to the ADC input of the board. The signals were acquired and processed on the DSP board at 8 ksa/s. The processed output of the DSP board was acquired by PC sound card for playback, and analysis.

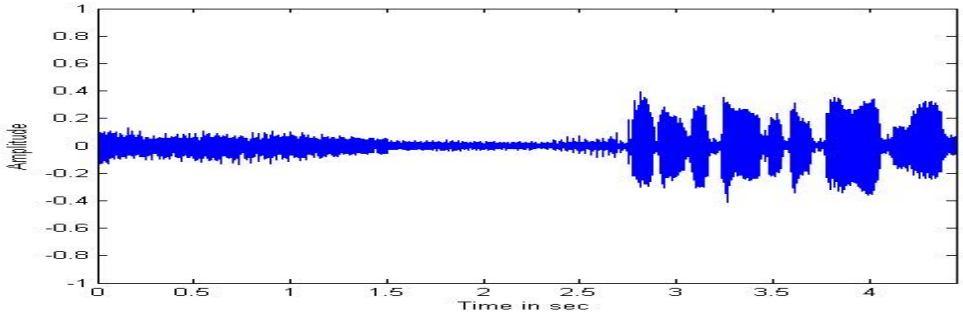
Based on earlier work [12] for spectral subtraction, we have used the following parameters: spectral subtraction factor α = 2, spectral floor factor β = 0.001, and exponenet factor γ =2. The real-time implementation of QBNE based spectral subtraction has been done pitch synchronously, i.e. taking window length equal to multiple of the pitch period. Actually we have taken 2 pitch periods. For fs=11.025 ksa/s, and F0=90.3 Hz, the referred window length is 244 samples. For sampling rate of 8 ksa/s we will need window length of 177 samples. In order to test the sensitivity of the noise reduction process to the pitch value, we have different window lengths also, which results in pitch non-synchronous processing.

It was found out that the effectiveness of the spectral subtraction depended on the number of windows in the estimation of noise. With MATLAB implementation, it was observed that optimal noise reduction was obtained with ≈55 windows. However with real-time implementation, the processing speed restricted the noise estimation to 8 windows.

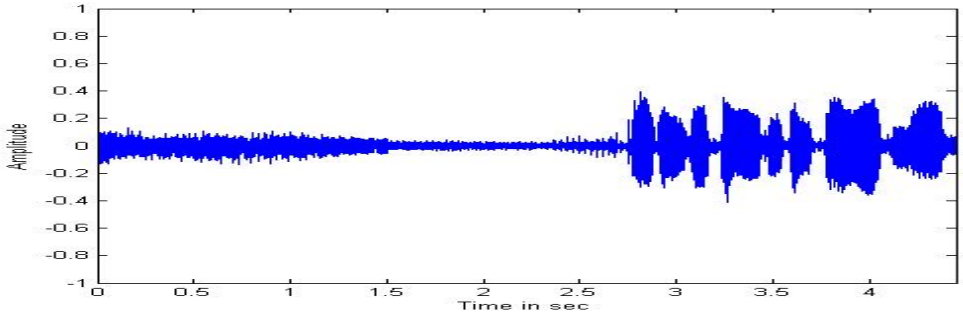
Fig. 5.1 and Fig. 5.2 show the results of the QBNE implementation with pitch non-synchronous and with pitch synchronous processing, respectively, for the 3 implementations: (1) C based implementation with noise estimate of 55 windows; (2) C based implementation with 8 windows, and (3) real-time implementation with noise estimate of 8 windows. All the implementations show noise reduction. However, we see that the real-time implementation is not as effective as off-line C implementation.



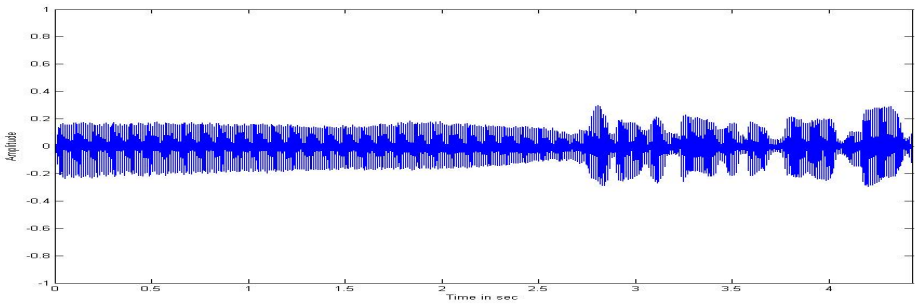
(a) Recorded speech waveform



(b) Enhanced signal using C based QBNE with 55 windows

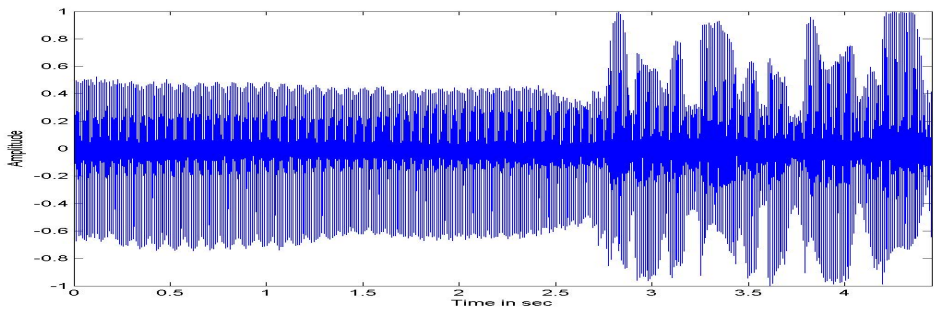


(c) Enhanced signal using C based QBNE with 8 windows

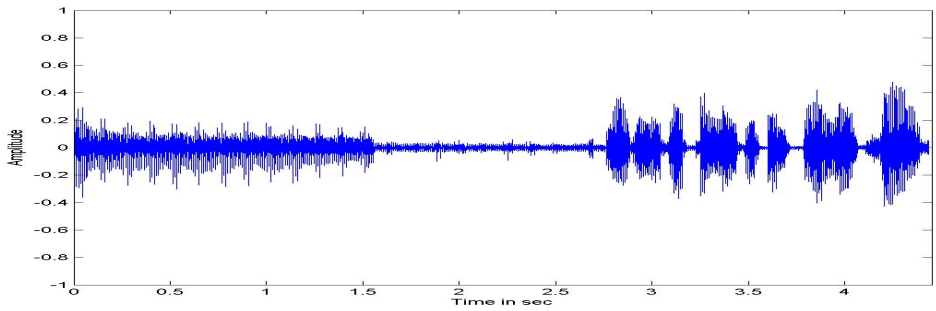


(d) Enhanced signal using real-time based QBNE with 8 windows

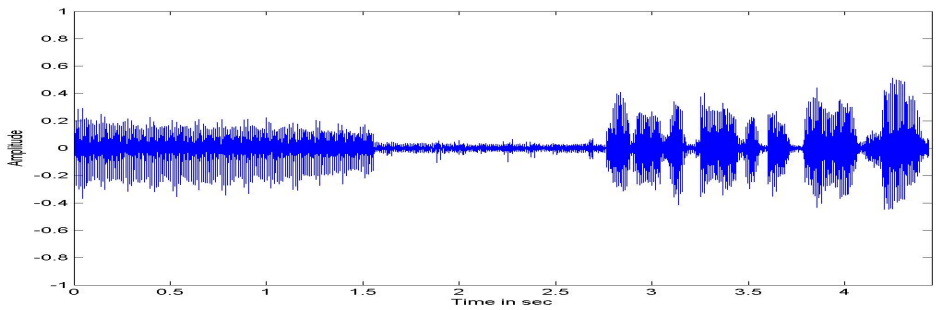
Fig. 5.1 Recorded and enhanced speech using spectral subtraction algorithm with QBNE. Speaker SP, material: question-answer pair in English, "*What is the time? It is 5'O clock*", generated using Servox electrolarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 2$



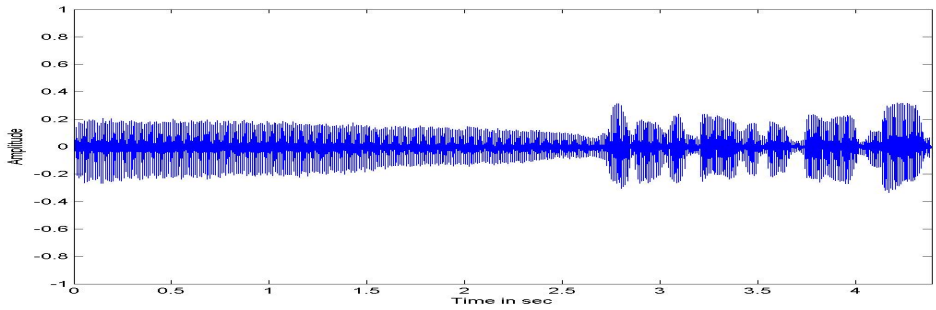
(a) Recorded speech waveform



(b) Enhanced signal using C based QBNE with 55 windows



(c) Enhanced signal using C based QBNE with 8 windows



(d) Enhanced signal using real-time based QBNE with 8 windows

Fig. 5.2 Recorded and enhanced speech using pitch synchronous spectral subtraction algorithm with QBNE. Speaker SP, material: question-answer pair in English, "*What is the time? It is 5'O clock*", generated using Servox electroarynx. Processing parameters: $\alpha = 2$, $\beta = 0.001$, $\gamma = 2$

Chapter 6

SUMMARY AND CONCLUSIONS

Main problem with the use of electrolarynx is production of background noise, caused by leakage of the acoustical energy from the vibrator to outside. Presence of background noise deteriorates the perceptual quality of speech. The objective of this project was to implement a real time system in which the noisy sound is picked up by a microphone for the removal of the background noise, process and output through a speaker. The earlier work in our lab, by Bhandarkar [10], [11] and Pratapwar [12], [16], [18], as a part of their M.Tech. dissertations, has shown that pitch synchronous application of spectral subtraction with quantile based noise estimation gives effective noise suppression. In this processing, enhanced magnitude spectrum is obtained by subtracting the estimated magnitude spectrum of noisy speech. The enhanced magnitude spectrum is coupled with the phase spectrum of noisy speech for re-synthesizing the time domain waveform. It has been that the analysis synthesis operations should be carried out with a window length of 2 pitch periods and 50% overlap. Noise estimation can be carried out by ABNE techniques, by averaging the input spectrum during the "noise" period, i.e. when the speaker's lips closed. It has been observed that noise characterization slowly change because of changes in the application of electrolarynx against neck tissue, and hence dynamic estimation of noise is needed. Work by Pratapwar [12], [16], [18], has shown that QBNE technique can be effectively used for dynamic estimation of noise, and best results are obtained of quantile values and as a function of frequency and SNR.

This project investigated real-time implementation of this algorithm using TI DSP TMS320C6211 based DSK board. As a first step towards this objective, the earlier MATLAB based programs have been converted to C code and results have been tested. Because of constraints of real-time processing, the analysis-synthesis was carried out by setting the phase as zero. This is a major deviation from the basic spectral subtraction method. However, offline processing indicated that this did not result in any noticeable distortion. Further, it was found that for effective estimation

of noise, 55 or more frames are needed. However, because of processing speed constraints, the number of frames had to be restricted to 8.

The real-time implementation of the algorithm was done on a TMS320C6211 DSP board. From the results obtained it is clear that the performance of the DSP based algorithm is degraded as compared to the off-line processing. The main reason could be significant number of underflows in the fixed-point arithmetic.

The program needs to be optimized for improving its execution speed and thereby to increase the number of windows for noise estimation. Further improvement in speed may be obtained by writing the code directly in assembly language. Finally, it may be necessary to move to a faster hardware.

REFERENCES

[1] L. R. Rabinar 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] Qi Yingyong and B. Weinberg, "Low-frequency energy deficit in electro laryngeal 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*, 1996, pp. 764-771.

[8] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Tran. ASSP*, 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. ICASSP*, 1979, pp. 208-211.

[10] S. M. Bhandarkar, "Reduction of background noise in artificial larynx" *M.Tech. Dissertation,* Guide: Prof. P. C. Pandey, Electrical Engineering Department, IIT Bombay, Jan 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, 2002, pp. 591-594.

[12] S. S. Pratapwar, "Reduction of background noise in artificial larynx" *Project Report,* Guide: Prof. P. C. Pandey, Electrical Engineering Department, IIT Bombay, Feb 2004.

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

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

[15] 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, 2002, pp. 985-988.

[16] 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, USA, 2003.

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

[18] P. C. Pandey, S. S. Pratapwar, and P. K. Lehana, "Enhancement of electro laryngeal 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.

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

[20] *TMS320C6000 Peripherals Reference Guide,* Literature Number: SPRU190D, Texas Instruments, Feb 2001.

[21] *TMS320C6000 Programmer's Guide,* Literature Number: SPRU198F, Texas Instruments, Feb 2001.

[22] *TMS320C6000 CPU & Instruction Set Reference Guide,* Literature Number: SPRU189F, Texas Instruments, Oct 2000.

[23] *Code Composer Studio Getting Started Guide,* Literature Number: SPRU509, Texas Instruments, May 2001.

[24] *TMS320C62x DSP Library Programmer's Guide,* Literature Number: SPRU402, Texas Instruments, Mar 2000.