Komal R. Borisagar · Rohit M. Thanki
Bhavin S. Sedani

Speech Enhancement Techniques for Digital Hearing Aids

Springer

Speech Enhancement Techniques for Digital Hearing Aids

Komal R. Borisagar • Rohit M. Thanki
Bhavin S. Sedani

Speech Enhancement Techniques for Digital Hearing Aids

Springer

Komal R. Borisagar
E. C. Department
Atmiya Institute of Technology and Science
Rajkot, Gujarat, India

Rohit M. Thanki
C. U. Shah University
Wadhwan City, Gujarat, India

Bhavin S. Sedani
E. C. Department
L. D. Engineering College
Ahmedabad, India

ISBN 978-3-319-96820-9 ISBN 978-3-319-96821-6 (eBook)
https://doi.org/10.1007/978-3-319-96821-6

Library of Congress Control Number: 2018952353

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The interference of background noise is the greatest problem reported by hearing aid wearers. Background noise is a combination of different frequencies; these unwanted frequencies reduce the precision of speech. A higher level of background noise degrades intelligibility. The speech signal is a quasi-periodic signal. If the speech signal is masked by noise, then the intelligence of that signal is reduced significantly and because of that, it is difficult to understand what is being said. Moreover, the speech signal carries many redundant samples which masks a portion of the noise. Noisy signal can't be easily detected by a person with a hearing disorder. It's an irritating process between detecting intelligence from a noisy speech by those with hearing disorders whereas normal people can do this job easily.

A person with a hearing impairment, may find it cumbersome to identify specific frequencies in the presence of noise that contains basic frequencies. In that case, the speech signal is not audible; thus, it is not possible to interpret speech, and this may cause problems in everyday life. Any product associated with hearing aids needs more work on design in such a way that the effect of the noise would be reduced before any further modification. It is vital to observe the sound quality with the background noise.

Deafness is a conical disability due to either a sensory neural defect in which cells are dead because of age, or a major disease. Deafness can also be caused by problems with bone and air conduction. One of the methods to solving hearing disorders is by cochlear implants, but this requires surgery. As an alternative, most people wear hearing aids instead. Enhancement of noisy speech is possible in the case of hearing aids. Generally, different noises have affected various frequencies of the speech signal. Fixed filters can help a great deal when it comes to removing an unwanted noise frequency. However, there can be many variations of noise frequency and over time it may degrade the fixed filter. It can be seen that with an unwanted signal, the speech component is also affected. With that constraint, the use of filtering techniques that are only applicable to the incoming noise signal is required. In addition to this, as per the noise characteristics, the filtering process should be adaptive.

Even after the removal of noise from speech using advance adaptive filtering methods, most partially deaf patients are not able to recognize all the frequencies equally. The frequency response of the patient's ear gives proper information about losses at a particular frequency. Basically, speech with the noise removed should be enhanced as per the audiogram structure of the individual. Clean speech is the name given to the enhancement in the individual frequency band in the wavelet domain. The audiogram requires individual frequency components that are sometimes less sensitive; thus only bands which are amplified by volume can process speech using the multi-resolution approach of a discrete wavelet transform. This book presents novel approaches such as: adaptive filtering and frequency band enhancement in the wavelet domain that assist speech enhancement as well as noise reduction in speech signals.

The book explains how noise can be extracted from the silent part of the speech signal. The various types of adaptive filter such as the least mean square (LMS), normalized LMS (NLMS), and the recursive least square (RLS) are described for noise reduction from the noisy speech signal along with voice activity detection (VAD). The different types of practical possible noises generated for the preparation of noisy speech signal and speech can be cleaned effectively. Prediction of the noise signal is the main task in the present work. The methods mentioned are compared based on various parameters such as the amount of noise reduction, the estimation of filter weights and the convergence rate of the filters, mean square error (MSE), and peak signal to noise ratio (PSNR). The presented methods provide significant results in noise reduction and can be observed within a time domain. Also, reconstructed speech can be observed with noise removal and proper intelligence.

This book is a Ph.D. research work and extension work of Dr. Komal Borisagar, submitted to the Department of Electronics and Communication Engineering, Shri Jagdish Prasad Jhabarmal Tibrewal University (JJTU), Jhunjhunu, Rajasthan in 2012. The authors are indebted to numerous colleagues for valuable suggestions during the entire period of the manuscript's preparation. We would also like to thank the publishers at Springer, in particular Mary E. James, senior publishing editor/CS Springer, for their helpful guidance and encouragement during the creation of this book.

Rajkot, Gujarat, India — Komal R. Borisagar
Wadhwan City, Gujarat, India — Rohit M. Thanki
Ahmedabad, India — Bhavin S. Sedani

Contents

Chapter 1
Introduction

1.1 Sound

A sound wave is described as being similar to ripples on the surface of the sea. A "wave" in the atmosphere contains much less variation in pressure than normal atmospheric variations. An acoustic or sound wave as a source generates the compression and rarefaction of particles. Usually, sound is generated by a vibrating entity such as a violin string, a loudspeaker diaphragm, the motor in a machine, or the vocal cords in a human body. Also, sound cannot travel in a vacuum. The eardrum vibrates in direct response to pressure variations in a wave when they reach the ear, and the pressure fluctuations are heard as a sound.

1.2 Ear Structure and Its Workings

Human ears are found in pairs, situated on the left and right sides of the head, and also can be considered as sensory organs comprising the auditory system, which detects sound, and the vestibular system, which is responsible for maintaining body balance and equilibrium [1]. The basic structure of the ear is shown in Fig. 1.1 [1]. The ear can be divided into three parts anatomically, known as the external ear, middle ear, and inner ear. The sound collection and amplifying mechanisms of the ear are such that

- It can work as a transducer, which converts sound vibration into action potentials.
- The action potential can be delivered by the nerves.

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_1

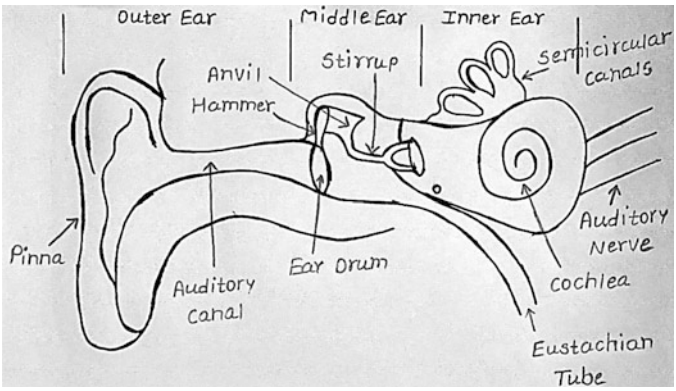

Fig. 1.1 Structure of the ear

1.2.1 External Ear

The external ear protects the tympanic membrane, recognized as the eardrum. It also collects and directs sound waves through the ear canal to the eardrum.

1.2.2 Middle Ear

The middle ear bones are distinguished as the malleus, incus, and stapes; these bones amplify sound waves. The middle ear, differentiated from the external ear by the eardrum, is an air-filled cavity, also identified as the tympanic cavity, carved from the temporal bone. It joins the throat–nasopharynx via the eustachian tube, which equalizes the air pressure on both sides of the eardrum. In general, the walls of the tube are distorted. The tube is opened by swallowing and chewing actions, allowing air in or out as needed for equalization [1]. Equalization of air pressure guarantees that the eardrum vibrates maximally when it is struck by acoustic waves. The functional processes of hearing are as follows:

- The malleus hammer is attached to the eardrum.
- The malleus attaches to the incus, and these then join to the stapes.
- The stapes and oval windows are linked, forming a part of the cochlea.

1.2.3 Inner Ear

The inner ear is formed by a network of fluid-filled tubes running through the bone of the skull. The bony tubes are filled with a fluid called perilymph. The actual hearing

cells contain this membranous structure, and the hair cells of the organ are very important. The bony labyrinth has two major sections:

- The first part is the cochlea, which has a snail-like shape.
- The semicircular canals, which rest in the back, are responsible for balance maintenance.

1.2.4 Cochlea

- In the auditory nerve, the cochlea changes acoustic vibrations into action potentials.
- The cochlear fluid (perilymph and endolymph) vibrates in response to the vibration of the oval window.
- Vibrations in the fluid in turn cause the basilar membrane to vibrate.
- Vibrations of the basilar membrane cause the hair cells to bend, generating potential.
- To produce action potentials, the fibers will be stimulated if the generator potentials are large enough.
- Pitches are generated in the different parts of the cochlea in such a way that the base is responsible for producing a high pitch and the apex generates a low pitch [2].

1.3 Hearing Impaired

Hearing problems occur mainly because of improper function of the ear cochlea. In the human, thousands of hair cells are present at the time of birth. These cells are responsible for sensing electronics sound waves with different frequencies, which are forwarded to the brain for decoding. For such reasons as aging, exposure to very loud sounds, drug administration, and some infections, these cells are reduced in number. Dynamic reduction in the number of cells creates problems in hearing, which results in deafness [3]. The types of deafness can be divided into three types: conduction deafness, sensorineural deafness, and central deafness.

- Conduction deafness: the term conduction relates to external ear and middle ear disorders wherein the transmission of sound to the cochlea is impaired. To diagnose conductive disorders is fairly comprehensive and robust as a consequence of their mechanics and relatively peripheral nature and the possibility of visual observation or surgical confirmation. This type of deafness is often directed to medical or surgical treatment.
- Sensorineural deafness: the term sensorineural implies an organic disorder of the cochlea and/or subsequent parts of the auditory system. ‘Sensory’ is intended to relate to the cochlea and neural system. Hair cell loss or damage is generated by the auditory nerve.

- Central deafness: the term central deafness is used to describe hearing disorders in which there is a defect in auditory pattern processing, often without appreciable hearing loss. The disorder is often most evident when speech stimuli are used.

1.4 Audiogram

The hearing sensitivity of patients in the audiological clinic can be described in terms of the number of decibels (dB) by which the threshold sound pressure of a person is higher than the normal threshold, referred to as dB of hearing loss. Graphing hearing loss against frequency is recognized as an audiogram [3]. The extent of hearing impairment is usually measured primarily in terms of loss of sensitivity. Because of the fundamental difference in the causes, characteristics, and management of conductive and sensorineural types of hearing loss, it is desirable to enumerate all these factors separately for a solution. Essentially, this reduces to measurement of both the sensitivity at the cochlea and the overall sensitivity of the ear, and the conductive loss is the difference between these two. Pure tone audiometry involves estimation of the threshold of haring for certain standardized stimuli, usually via the air conduction and bone conduction routes.

The threshold of hearing is variously defined but is often taken to be the lowest sound pressure or alternating force level at which, under specified conditions, a person gives a predetermined percentage of correct detection responses on repeated trials. Threshold definitions are usually based on 50% correct detection. The stimulations used are calibrated on the hearing level scale that has been obtained from normalization studies involving large numbers of subjects and has, at the 0 dB hearing level point, the modal value of hearing threshold levels measured in ontologically normal aged subjects [3]. Because of the unavailability of unique biological calibration for every audiometer, the national and international standards objectively define the biological baseline for certain combinations of earphone or bone vibrator and acoustic or mechanical coupler. The standards are also specific to particular audiometric test frequencies. For air conduction, frequencies of 0.125, 0.25, 0.5, 1, 1.5, 2, 3, 4, 6, and 8 kHz are included. For bone conduction, at least the following are included: 0.25, 0.5, 1, 2, 3, and 4 kHz, although there are some variations among national standards. Figure 1.2 illustrates the standard audiogram format used for plotting results.

1.5 Digital Hearing Aids

Deafness is an often underestimated and misjudged handicap that seriously limits the life capabilities of the patient. Although the handicap of the deaf is not as conspicuous as, for instance, blindness or a physical disability, deaf people often find themselves excluded from normal society because of the many problems they have

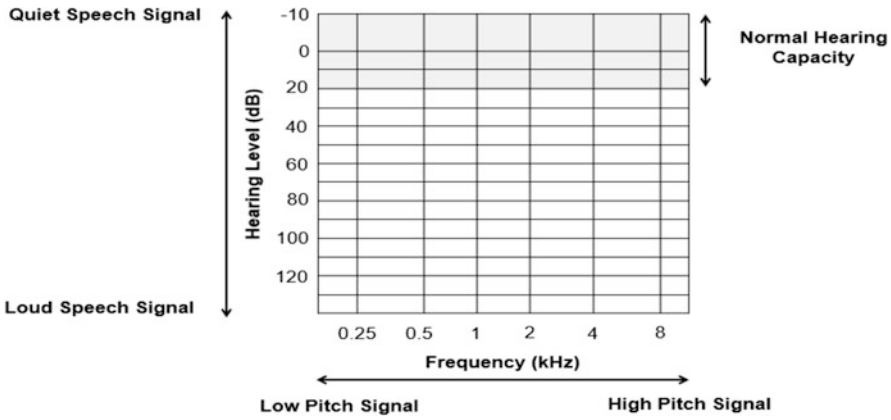

Fig. 1.2 Format for audiogram

in communicating with other people. Unfortunately, medical science is not capable of curing deafness [4]. On the other hand, a number of techniques exist that enable the patients to communicate, such as lip reading, sign language, and, of course, hearing aids. Most of these techniques are of limited help because lip reading, for example, only provides a small part of the information given by the spoken word, and sign language is known by only a relatively small number of people.

Previously, analog hearing aids were more popular for picking up and amplifying the sound before returning it to the ear through a kind of miniature loudspeaker, an effective rehabilitative means for mild to moderate hearing losses. However, the conventional analog hearing aids have the following disadvantages:

- Analog hearing aids give restricted performance.
- Their characteristics are a function of mechanical variations in time and temperature.
- Their usefulness is highly limited in case of people suffering from sensorineural loss, in which the frequency response must be optimally designed depending on the condition of the person's residual auditory area.

In a digital hearing aid, a microprocessor or application-specific integrated circuit (ASIC) replaces the hardware used to process the signal, such as filtering and compression. The analog output of the microphone is low-pass filtered to prevent aliasing errors, sampled at discrete intervals, and converted to binary form using an A/D converter. The programmed processor will treat a digital signal accordingly. The processed digital signal will then be transformed back to an analog signal to achieve compatibility with the human ear via the D/A converter and sent to the hearing aid receiver. The first generation of digital hearing aids is likely to be similar to currently available analog hearing aids with regard to the type of processing that would be done. The major difference between digital hearing aids and the present generation of analog hearing aids is the degree of control over parameters of the hearing aids. Because the characteristics of the hearing aid such as frequency

response, maximum power output, and compression parameters will be specified in the software, the constraints imposed by the hardware will be eliminated. The characteristics most suitable for an individual hearing aid user can be specified precisely in the software. Digital hearing aids assure much compensation over conventional hearing aids [5], including the following:

- Because of the internal processor, digital hearing aids can be programmed.
- Electro-acoustic parameters can be adjusted with much greater precision.
- Some feedback mechanism can be implemented for self-monitoring capabilities.
- Logical operations can be developed for testing and calibration internally.
- Noise reduction is possible through the advanced processor.
- Automatic control of signal level loudness can be implemented accordingly.

Hearing loss is widely recognized as one of the most common human disorders. On average, 1 of 1000 newborns is affected by a severe hearing loss. Moreover, the prevalence of hearing loss increases monotonically for older populations as the patient's hearing is irreversibly affected by, for instance, noise-induced trauma and age-related hair-cell degeneration. As a result, about half the people 65 years and older suffer from a mild to severe hearing loss. Although the concept of a digital hearing aid was expected at an earlier date, two major technical problems had to be resolved before developing a wearable digital hearing aid. The first was the development of a digital signal processor (DSP) that can operate very rapidly in real time. Another difficult problem is developing digital circuitry is that it should be small in size and with very low power consumption for real-time use in a small wearable unit such as a hearing aid [5].

The main focus of why hearing-impaired people are so seriously handicapped in everyday listening situations seem to be very little studied. This lack of knowledge particularly manifests itself in the uncritical way in which hearing aids are assumed to be of benefit. If proper transmission is not conducted, the result is conductive deafness, which breaks the transmission chain. Most of these patients are cured with the help of appropriate surgery [6]. It is generally recognized that electronic amplification alone cannot compensate satisfactorily for these hearing losses. On the other hand, many hearing-impaired persons appear to be rather disappointed with their hearing aids and take little interest in using them.

Professional hearing aids design takes up the challenge that the clearest speech be forwarded to the hearing impaired. The common grievance of those with hearing impairment is the reduced ability to recognize speech in everyday communication in a noisy environment. These difficulties are often experienced especially in the workplace and during many activities as a burdensome handicap, and this handicap is also a major constraint for aged persons. Current statistics show that among the total hearing impaired, half are suffering from sensorineural losses and only about 20% from conductive losses [7]. Several reports have demonstrated that about 5–15 dB more signal-to-noise ratio (SNR) is needed to treat that type of hearing impairment by means of hearing aids. Every 4–5 dB improvement of the SNR may raise speech intelligibility by about 50%. Although many techniques provide substantial attenuation of narrowband noise, none appears capable of improving the intelligibility of the enhanced speech.

1.6 Issues in Digital Hearing Aids

The key parts of digital hearing aid are the microphone, A/D and D/A converters with a suitable DSP processor, the receiver, and a two-port memory. A microphone with good gain converts the analog signal into a digital signal. A filtering operation is needed to remove most known high-frequency noises. The sigma-delta process, used with a 16-kHz sampling frequency, seems the ideal treatment for the analog to digital conversion process. To improve the quality of the signal, the sampling rate might be extended up to 48 kHz to obtain more resolution.

Memory is used to store the processing parameters that can be downloaded from the audiometer/programmer system to the user. The digital signal processor contains an array of adders, multipliers, and registers that provide the fundamental operations necessary for implementing various digital algorithms. Whenever data are converted into the digital domain, everything is controlled by DSP processors. Different types of filtering operations are executed, and storage of different audio processed data of different sets of memory is applied. In the process, with the help of various filter parameters, peak output is optimized. The set parameters are checked out before fitting by an audiologist. The same algorithm logic is used again to convert the digital signal into analog, which can drive the speaker of the hearing aids. A loudspeaker is properly designed that can be driven by the generated analog signal by considering its impedance and attenuation. The configuration is in contrast with the use of a general-purpose DSP, wherein considerable power is consumed in executing program instructions. Now, with the development of ASIC, the circuits are implemented in CMOS technology to reduce size and power consumption [8]. A functional diagram of digital hearing aids is shown in Fig. 1.3.

1.7 Motivation for This Book

The main motivation for this book is that of providing various enhancement techniques for the detection and reduction of noise from the speech signal. At any time, to reduce noise from the speech signal, it should be operated in the frequency domain. For a proper filtering operation, a designed filter bank is required with more trade-off coefficients for filtering; more multiplication and more filtering power is required for this. Dependence on the complexity of computation power consumption is varied. Therefore, it is important to design the filter bank for consuming as little processing power as possible. The filter banks should be designed with a minimum number of multiplications as the multiplication consumes less power.

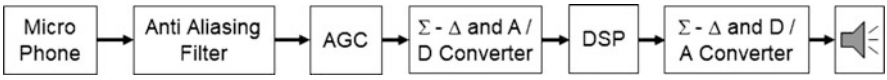

Fig. 1.3 Functional diagram of digital hearing aids

Now, in the presence of noise, hearing aids do not function well, so some filtering operation is required. For that need, adaptive signal processing has found widespread practical applications. The key reason for the widespread use of adaptive filters is their ability to optimize their own performance through recursive modifications of internal parameters. There are numerous applications of adaptive filtering, such as adaptive beam forming, noise canceling, speech formant estimation, and array processing, in fields such as telecommunications, radar, sonar, and navigation systems and biomedical electronics.

A major discrepancy in the foregoing concept is that designing of a filter bank needs advance prediction of the noise characteristics, which is very difficult in real-time operations. Thus, an adaptive filter is a kind of automatic filter that adjusts its parameters per the requirement by following some of the equations and rules so as to achieve some specified objective. Whenever there is a requirement to process the signal whose characteristics may not be known exactly or even statistically, in such circumstances, the adaptive filter offers an attractive solution and provides significant improvement in performance compared to a fixed filter.

Speech signal processing, especially the improvement of speech signal for the hearing impaired, is a potential area in which the adaptive filtering theory can be effectively applied to improve the intelligibility of the speech signal. Generally linear or nonlinear methods can be adopted to carry out the process with the speech signal. Compared to linear methods, nonlinear methods are more useful in speech enhancement because most of the path through which signal travels is nonlinear, similar to all converters and the functionality of the loudspeaker. The nonlinear model has better noise suppression. Thus, in this book, different methods are developed for improving the intelligibility and SNR for speech signal using adaptive filters such as least square mean (LSM), normalized LSM (NLSM), and recursive least square (RLS).

Throughout the world, huge numbers of people are suffering from hearing deficiencies and deafness. It is noted as a very serious chronic disability. The problem of deafness calls for the best talent and best efforts from scientists. The hearing-impaired subjects, especially sensorineural loss patients and aged people, experience more difficulty in understanding conversational speech in the presence of background noise than do normal listeners. Statistics indicates that more than 50% of auditory handicaps are sensorineural loss subjects. Many deaf people, especially aged persons, have severe chronic visual impairment as well, and this cannot be rectified. The present analog hearing aids are just sound amplifiers. If some methods are developed to reduce noise without distorting the signal, it will provide significant benefit to those with sensorineural loss handicaps and aged people. Also, one can bring significant changes to the hearing-impaired child by providing an appropriate digital hearing aid and by giving the necessary educational training at an early stage. Hence, with a digital hearing aid it is possible to make the child more self-reliant. However, many scientists agree that complete absence of auditory response among the deaf population is decidedly rare. There appears to be irrefutable evidence that digital hearing aids could make a considerably greater contribution toward alleviating the burden of deafness. Although the concept of a digital hearing aid was

anticipated at an early date, but the problem of developing digital circuitry that is small enough and sufficiently low in power consumption for practical use is yet to be resolved. So far, the work that has been carried out is based on using finite impulse response (FIR) filters wherein a large number of multiplications is required. Hence, the challenge of research is designing a filter bank with the optimum condition that can take a variable way for changing its value and is accordingly efficient to improve noise in the incoming disturbed signal with minimum complexity. This is the second motivation for writing this book.

1.7.1 Important Areas of Speech Signal Covered in This Book

Deafness is most often caused by deterioration of the inner cells, which are very small hair-like cells; in fact, it is not a major problem with the associated neurons. This finding implied that if the neurons can be stimulated by a means other than hair cells, some hearing can be recovered. One of the most suitable solutions to fight deafness is to wear digital hearing aids because of following advantages: greater control, affordability, and flexibility and easier maintenance. In many cases, the most widely preferred option is digital hearing aids with given prime requirements.

- Per the conclusion of an audiogram, the individual frequency can be amplified.
- Loud noises are bothersome with speech processing.

The limitations of hearing aids mentioned here can be tolerated with some software modification in the system. The audiogram of any patient can be measured and per that given modification could be deployed. For that speech can be analyzed, and individual frequency components can be separated from the given speech signal. A better solution for quality improvement in speech signal in hearing aids are hearing aids with speech frequency separation and loudness increment of individual frequencies using wavelet transform. Speech enhancement of the technique using an adaptive filter in the wavelet domain is covered in this book. This book is also covers two major methods such as voice activity detection in the noisy speech signal and use of adaptive filter in speech signal processing. The details of these two operations are given in the next subsection.

1.7.1.1 Voice Activity Detection in the Noisy Speech Signal

The prime requirement of the noise-free speech signal is the reduction of noise from any speech signal. The very basic first step is in the presence of background noise is first of all identifying occurrence of the acoustic signal around evenly distributed surrounding noise. As a consequence, a noise-corrupted speech signal is given to the voice activity detection (VAD) algorithm, which finds the pauses or silences during speech.

In the VAD algorithm, the main focus area is searching out the zero crossing of speech signal occurrence areas and based on that the speech signal can be recognized. In total speech with background noise, wherever speech signals are, those areas are getting more zero crossing rates. In the space of background noise the zero crossing rate is very low. In the VAD algorithm, the next most important part is searching for the energy through the spread of the signal. Compared to the original voice signal, noise occurrence has been getting much less energy. By deciding a specific threshold level, it could be possible to search for our signal energy in the given area of the speech signal. For the weak fricatives energy would be much less for the specific letter, so in that case energy as well as zero crossing rates gives a useful decision for the VAD.

In the next step, filtered speech is provided to the voice activity detection algorithm, which is developed for only the extraction of speech activity. Thus, occurrences of silence are detected, and the rest of the time speech is detected. Detected silence can be applied for wavelet thresholding, which makes the position of the silence trend to zero. Using VAD, pauses in speech can be detected to a very great extent. By searching these pauses, the number of samples responsible for noise can be identified, and those samples are made to zero so that the silence is free from noise from surrounding areas.

1.7.1.2 Adaptive Filtering for Speech Signal Processing

Now, in the next step, the main target for clearing from the speech is only because all the silences are now without noise. Still, the speech has background noise. For that, adaptive filters are more suitable. For successful implementation of this work it is necessary to prepare an acoustic environment in which two signals are required. One signal can be considered as a primary input and the second can be considered as secondary input. In recent work, the acquisition of noisy speech occurs with a reference noise signal. Now, the real-time noisy speech must be removed with the given algorithm. For taking the reference signal and primary signal, the adaptive environment can be prepared with the necessary data. The basic concept of adaptive filtering states that if any nonstationary inputs are present with the desired signal, then the adaption of the filter coefficient with the desired incoming signal allows the filtering operation to be carried out. Basically, in the category of adaptive algorithms, the algorithms mainly used are LMS, NLMS, and RLS linear filters.

LMS are the simplest type of the steepest descent-based adaptive algorithms. This algorithm is mainly based on the transversal filter domain. In the algorithm, measurements of the relevant correlation functions and matrix inversion are not required. Because of this simplicity, the LMS algorithm is used here. In LMS, when tap weights supplied by the information source are large the algorithm then suffers from gradient noise. To overcome this difficulty in real-time speech, the NLMS is very helpful. Basically, in NLMS, coefficients are normalized with respect to the square Euclidean norm of the tap input adjustments and can be applied to different tap weight vectors at different forward iterations. The next category of adaptive filter,

based on the theory of matrix inversion lemma, is known as the recursive least square algorithm, the RLS. RLS is a more complex algorithm than those already mentioned, with the advantage that the convergence rate with the incoming signal is much faster.

1.8 Book Organization

This chapter has briefly discussed the general characteristics of digital hearing aids and the associated problems. In addition, the motivation behind writing this book is presented. The rest of this book is organized as follows: Chap. 2 presents speech signal and its characteristics; Chap. 3 gives information about adaptive filters and the resources used in the present work; Chap. 4 presents information about various signal processing transforms used in implementation of the present work; Chap. 5 presents a speech enhancement technique based on various adaptive filters and its performance analysis; and Chap. 6 discusses a speech enhancement technique in the wavelet domain and its performance analysis. Chapter 7 then concludes this book with some research directions planned for the future.

References

1. Stephens, D. (1987). *Adult audiology* (Vol. 2). Oxford: Butterworth-Heinemann.
2. Paglialonga, A., Tognola, G., Baselli, G., Parazzini, M., Ravazzani, P., & Grandori F. (2006). Speech processing for cochlear implants with the discrete wavelet transform: Feasibility study and performance evaluation. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE* (pp. 3763–3766). Piscataway: IEEE.
3. Wright, D. (1987). *Scott-Brown's otolaryngology. Basic sciences* (Vol. 1). London: Butterworth Heinemann.
4. Levitt, H. (2001). Noise reduction in hearing aids: A review. *Journal of Rehabilitation Research and Development, 38*(1), 111.
5. Levitt, H. (1973). Speech processing aids for the deaf: An overview. *IEEE Transactions on Audio and Electroacoustics, 21*(3), 269–273.
6. Leisenberg, M. (1995). Hearing aids for the profoundly deaf based on neural net speech processing. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95. 1995 International Conference* (Vol. 5, pp. 3535–3538). Piscataway: IEEE.
7. Cheeran, A. N., & Pandey, P. C. (2004). Speech processing for hearing aids for moderate bilateral sensorineural hearing loss. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings (ICASSP '04). IEEE International Conference* (Vol. 4, pp. iv–17). Piscataway: IEEE.
8. Siravara, B., Magotra, N., & Loizou, P. (2002). A novel approach for single microphone active noise cancellation. In *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium* (Vol. 3, pp. 3–7). Piscataway: IEEE.

Chapter 2
Generation of Speech Signal and Its Characteristics

2.1 Speech Signal

Speech is a pressure waveform that travels from a speaking person to one or more listeners. This signal is typically measured directly in front of the speaker's mouth, which is the primary output location for speech. Because the ambient atmosphere in which one speaks imposes a basic pressure, it is actually the variation in pressure caused by the speaker that constitutes the speech signal. The signal is continuous and its time and amplitude are very dynamic, corresponding to the constantly changing status of the vocal tract and vocal cords. The characteristics of a speech signal, called phonemes, are like a discrete sequence. Every phoneme during its short period of time has some articulatory and acoustic properties. Each phoneme does have some limitations on the positions of various vocal tract articulators or organs: tongue, vocal folds or vocal cords, lips, teeth, velum, and jaw. Speech sounds fall into two broad classes: (1) vowels, which are responsible for allowing unrestricted airflow throughout the vocal tract, and (2) consonants, which control airflow at some point and have a weaker force than vowels.

2.1.1 Articulatory Phonetics and Speech Generation

Speech is produced as one exhales air from the lungs as the articulators move. The sound production process can be recognized as a filtering process in which the vocal tract filter is excited by a speech sound filter. The source either is noisy aperiodic, causing voiced speech, or is periodic, causing unvoiced speech [1]. The source of the periodicity for the former is found in the larynx where vibrating vocal cords interrupt the airflow from the lungs, producing pulses of air. Both the area of the glottal opening between the vocal cords and the volume of the pulse of air can be approximated as half-rectified sine waves in time, except that glottal closure is

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_2

more abrupt than its opening gesture. This asymmetry assists the speech signal to be more intense than might otherwise be the case, because abrupt changes in the excitation increase the bandwidth of the resulting speech.

2.1.2 Anatomy and Physiology of Speech Generation

The human speech production organs are used for multipurpose functions such as speech generation, breathing, eating, and sensing odors. Thus, in a communication sense, speech generation cannot be as optimal an information source as the ear is a receiver. Certain parallels can be made between electronic and human speech communication. Humans minimize effort, in terms of their energy and time, while maximizing the perceptual contrast in listeners.

2.1.3 Vocal Tract

The lungs provide the airflow and pressure source for speech, and the vocal cords usually modulate the airflow to create many sound variations. However, the most important system component in human speech generation is the vocal tract. It is a tube-like passageway made up of muscles and other tissues, and it enables the production of the different sounds that constitute spoken language.

For most sounds that are initiated in the glottis, the vocal tract alters the temporal and spectral distribution of power in the sound waves. In addition, the vocal tract generates some sounds directly. It is the source for obstruents like stop and fricative sounds. Different phonemes primarily can be distinguished by their periodicity, whether voiced or unvoiced, and spectral shape, which frequencies mostly have major power and duration; longer phonemes are perceived as having greater stress. The state of the vocal folds usually specifies each phoneme's voicing feature choice [1]. By far the most important aspect of speech production is the specification of different phonemes via the filtering actions of the vocal tract. Speech perception is dominated by the presence of sound power. The formants are often abbreviated F1; hence, F1 means the formant with the lowest frequency. In voiced phonemes, the formants often decrease in power with frequency as the result of the general low-pass nature of glottal excitation; thus, F1 is usually the strongest formant. Displacing the articulators changes the shape of the acoustic tube through which sound passes and alters its frequency response. After leaving the larynx, air from the lungs passes through the pharyngeal and oral cavities and then exits at the lips. For nasal sounds, air is allowed to enter the nasal cavity by lowering the velum at the boundary between the pharyngeal and oral cavities.

The velum is kept in a raised position for most speech sounds, blocking the nasal cavity from receiving air. During nasal sounds as well as during normal breathing, the velum lowers to allow air through the nostrils. In the speech generation

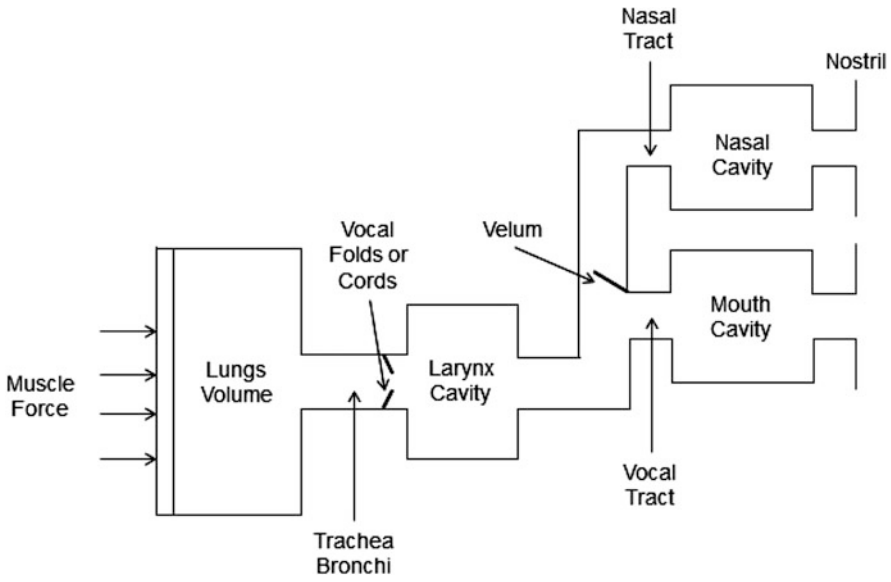

Fig. 2.1 Schematized diagram of the vocal apparatus

mechanism, it is really beneficial to extract the important features of the physical system and based on that prepare a realistic and tractable mathematical model. Figure 2.1 shows such a more physically realistic schematic diagram of the vocal system.

For completeness, the diagram includes the subglottal system, composed of the lungs, bronchi, and trachea, a mechanical model of the vocal cords, including mass, spring, and damping components, and a variable area set of tubes that model the vocal tract configuration. The subglottal system generates the energy waveform, which is the source of speech production. The mechanical model of the vocal cords provides the excitation signal for the vocal tract. The resulting speech signal type of acoustic wave is radiated from the mechanism when air comes out of lungs; the signal shape is decided by the time-varying vocal tract [1].

The vocal tract and nasal tract are shown in Fig. 2.1 as tubes of non-uniform cross-sectional area. As sound, generated as just discussed, propagates down these tubes, the frequency spectrum of the signal is controlled by frequency sensitivity and tube selection. This effect is very similar to the resonance effects observed. In the aspect of speech production, the vocal tracts generate the frequency, and they are well known as formants. The generated frequencies are controlled by the dimensions of the vocal tract. Each individual shape is responsible to decide the set of formant frequencies. Thus, in general, the spectral property of the speech signal changes with time and the shape of the vocal tract.

The vocal tract might be visualized as a sequence of cylinders, each having a variable cross-sectional area. In the vocal tract, the tongue, the lower teeth, and the lips undergo significant movements during speech production. In contrast, the upper

and rear boundaries of the vocal tract are relatively fixed but with diverse composition [2]. The nasal cavity consists of many passages lined with mucous tissue and has no movable structures. Its large interior surface area significantly attenuates speech energy. The opening between the nasal and pharyngeal cavities controls the amount of acoustic coupling between the cavities and hence the amount of energy leaving the nostrils. Increased heat conduction and viscous losses cause formant bandwidths to be wider in nasals than for other sonorants. During speech production, if either the vocal tract or glottis is completely closed for a time, airflow ceases and typically no sound emerges. This class of phonemes is called stops or plosives uses, such closures lasting several tens of milliseconds. Immediately before closure and at closure release, stops have acoustics that vary depending on the closure point: in the glottis, in the vocal tract, or at the lips.

2.1.4 Larynx and Vocal Folds or Cords

The vocal folds are important for speech production because even normal breathing generates a recognizable sound, as air is expelled by the lungs and travels through smoothly throughout the vocal tract. The sound generation path is very narrow and bounded so it can direct airflow and create turbulent noise or pulses of air. Most speech originates in the larynx, which consists of four cartilages: thyroid, cricoid, arytenoid, and epiglottis, joined by ligaments and membranes [2]. The passageway between the lungs and the vocal tract is called the trachea, which divides into two bronchial tubes toward the lungs. A non-speech organ called the epiglottis protects the larynx from food and drink. The vocal folds inside the larynx are typically about 15 mm long, have a mass of about 1 g each, and have amplitude vibrations of about 1 mm. When one breathes normally, the vocal folds are distant enough separately to evade sound creation, although increased airflow during exercise leads to loud whispers. If airflow is strong enough and the folds are close enough, such turbulent noise occurs at the glottis.

This speech is called whisper or aspiration and corresponds to the phoneme/h/. Similar noise can be generated in the same way higher up in the vocal tract at a narrow constriction. This generation takes place either between tongue and palate or between lips and teeth: the latter noises are called fricative sounds. The main difference between aspiration and frication lies in the range of frequencies: broadband for whisper and high frequency only for frication, because each noise source excites mostly the portion of the vocal tract directly in front of the source constriction. In frication, shorter cavities correspond to higher-frequency resonances than with aspiration. All such noise sounds are aperiodic because of the random nature of their turbulent source. Air leaving the lungs during sonorant phonemes is interrupted by the quasi-periodic closing and opening of the vocal folds. The vibration rate formed by opening and closing of the folds is known as the fundamental frequency. It is often abbreviated F0, in contrast to the formants F1, F2, . . . ; although F0 is not a resonance power in sonorants appearing primarily at harmonic multiples of F0, these

harmonics have a widely varying intensity depending on the formants [2]. The fundamental period or pitch period of voiced speech, T0 = 1/F0, corresponds to the time between successive vocal fold closures. T0 has a large dynamic range but the average value of T0 is proportional to the size of the vocal folds. To physically cause the vocal folds to vibrate, they must be close together and the lungs must generate sufficient pressure using the diaphragm so that the difference between the pressure below the glottis and that above it is large enough. Voicing is unlike other rapid movements in speech that are caused by voluntary muscle actions and are thus limited to low frequencies.

The increased velocity in the narrow glottis causes local pressure to drop. When it is low enough, the negative pressure forces the vocal folds to close, which interrupts the airflow. A positive pressure then develops in the larynx, forcing the vocal folds open again, and the cycle repeats until the diaphragm or vocal cord muscles relax. Major and secondary articulators, as already noted, are vocal tract organs that move to produce speech sounds and are thus called articulators. The tongue and the lips are the most important articulators. In secondary roles, the velum and larynx also contribute to speech production. Through its varying glottal opening, the larynx controls airflow into the vocal tract. The larynx can also be raised or lowered, which alters the tract length and which in turn raises or lowers, respectively, formant frequencies.

The jaw may be thought of as a secondary articulator because it helps to position the tongue and lips for most sounds. The lips are a pair of muscular folds that can cause a vocal tract closure or produce a narrow slit at the mouth. The lips can also either round and protrude or spread and retract; only the four front upper teeth participate actively in speech production. Immediately behind the teeth is the hard palate, the upper wall of the oral tract. Many phonemes require a constriction between the hard palate and the tongue. The most important primary articulator is the tongue, which has four components: tip, blade, dorsum, and root. As the upper and rear walls of the vocal tract are relatively rigid, speakers rely on the very flexible tongue to provide the mechanism to create the different vocal tract shapes needed to produce the various speech sounds. The tip is fast and agile, able to move up and then down within about 100 ms. The dorsum is the surface of the tongue whose frontal portion is the blade; the tongue body positions the dorsum. Most articulators move toward different target positions for each successive phoneme, starting as ballistic motion and then becoming more focused as the target nears or as other muscle commands are issued for a new target. The peak velocity of an articulator is often linearly related to its actual displacement. In generation of speech, a combination of voiced and unvoiced sound is always present; this is modeled in Fig. 2.2.

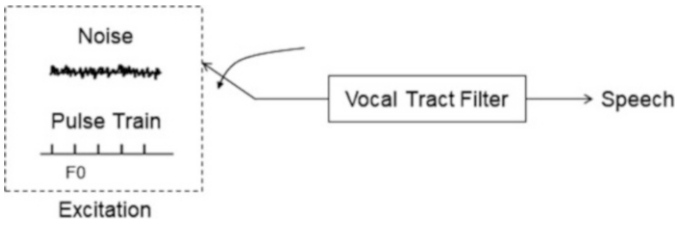

Fig. 2.2 Generation of speech for voiced and unvoiced sound

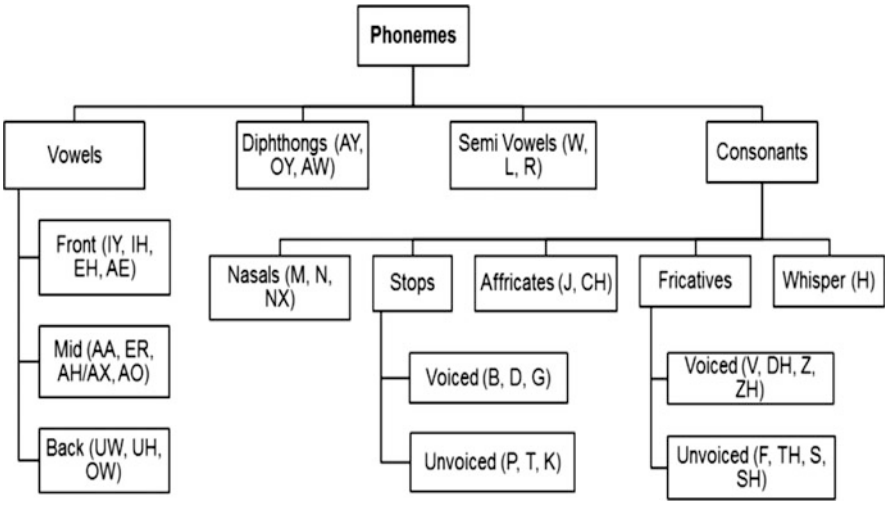

Fig. 2.3 Phonemes in American English

2.2 Major Features of Speech Articulation

Most languages (including English) can be described in terms of phonemes, which are the set of unique sounds [2]. In particular, for American English, there are between 39 and 48 phonemes including vowels, diphthongs, semivowels, and consonants. With the total 48 phonemes of American English there are those with the International Phonetic. It can be seen that the 48 phonemes are divided into five broad classes:

- Vowels and diphthongs are 18.
- Vowel-like consonants are 4.
- Standard consonants are 21.
- Syllabic sounds are 4.
- Glottal stop is 1.

The total set of 39 sounds is used in spite of using the full set of 48 phonemes [2]. Figure 2.3 summarizes phonemes in detail as follows:

- Vowels, 11.
- Diphthongs, 4.
- Semi-vowels, 4.
- Nasal consonants, 3.
- Voiced and unvoiced stop consonants, 6.
- Voiced and unvoiced fricatives, 8.
- Affricate consonants, 2.
- Whispered sound, 1.

The main interest in the production of phonemes is the following: the state of the vocal cords means checking vibration, the degree of any major constriction in the vocal tract, and the location of such constrictions. These points correspond to the features of voicing, manner of articulation, and place of articulation, respectively. The manner of articulation involves airflow in the vocal tract, whether it flows through the oral and/or nasal cavities, and the degree of any major vocal tract constrictions. The manner classes are vowels including diphthongs, glides, liquids, nasals, fricatives, and stops.

The vowels are the most important and largest class of phonemes; air passes relatively freely at rates of 100–200 ml/s through the pharynx and oral cavities. Nasalized vowels also allow air through the nasal cavity in some languages. Vowels have no constrictions sufficiently narrow to cause frication noise, turbulent and random airflow, or to block the airflow completely [2]. To avoid noise generation, the area of minimum constriction except for the glottis exceeds 0.3 cm.

Glides are also called semivowels, which resemble vowels but have a very high tongue position that causes a narrow vocal tract constriction barely wide enough to avoid frication. In many languages, there is a glide that closely resembles each high vowel in the language (e.g., glide /y/ resembles vowel /iy/ and /w/ resembles /uw/). In practice, glides are simply very high vowels that are difficult to maintain for more than a few tens of milliseconds; thus, they may be thought of as transient vowels. Vowels, on the other hand, can easily last for hundreds of milliseconds (ms), if desired.

Liquids also resemble vowels, except for use of part of the tongue as a major obstruction in the oral tract, which causes air to be deflected from a simple path. For the liquid /l/, also called a lateral, the tongue tip is in contact with the alveolar ridge and causes a division of the airflow into two streams on both sides of the tongue.

Nasal consonants have a lowered velum, which allows airflow into the nasal cavity and through the nostrils while the oral tract is completely closed. Some languages have nasalized vowels, where air flows through both the oral and nasal cavities. Such nasalization lowering the velum often occurs in parts of English vowels, but such sounds are not associated with a separate phonemic category, because English listeners interpret these sounds as normal free variation when vowels occur adjacent to nasal consonants. Thus the distinction is ‘allophonic’ and not phonemic [2].

All phonemes in the preceding four classes (vowel, glide, liquid, and nasal) are part of a more general manner class called sonorant. They are all voiced and

relatively strong in power. The other general class of phonemes is called obstruent and is composed of stops and fricatives, which are noisy and relatively weak, with the primary acoustic excitation at a major vocal tract constriction.

Stops or plosives employ a complete closure in the oral tract, which is then released. Air continues to flow through the glottis throughout a stop. The velum is raised throughout a stop to prevent nasal airflow during oral closure. Pressure builds up behind the oral closure and is then abruptly released. Air flows through the increasing orifice (at a decreasing speed as the opening increases more than a few tens of milliseconds). The initial intense burst of noise upon oral tract opening is called an explosion and is effectively a brief fricative. Before actual periodicity, an interval of noisy aspiration typically occurs with a duration called the voice onset time, or VOT. The initial portion of the VOT is frication, produced at the opening constriction, whereas the longer remainder is aspiration noise, created at the constricting glottis.

In voiced stops, on the other hand, vocal folds may continue to vibrate throughout the stop or start to vibrate right after the burst. One difference between voiced and unvoiced stops is that the vocal folds are more widely separated during the vocal tract closure for unvoiced stops and start to adduct only at the release, hence the longer VOT for unvoiced stops. As do other phonemes, stops usually last on the order of 80 ms. In contrast to other sounds, stops are sometimes very brief when they occur between two vowels. An alveolar stop followed by an unstressed vowel in the same word often becomes a flap, where the tongue tip maintains contact with the palate for as little as 10 ms. Stops have a complete occlusion in the vocal tract, but fricatives use a narrow constriction instead. To generate noise, fricatives need sufficient pressure behind the constriction with a narrow passage; this causes sufficiently rapid airflow to generate turbulence at the end of the constriction. Most speech sounds—vowels, liquids, nasals, and fricatives—each have a specific articulatory position and can be maintained over several seconds, if desired. Stops, on the other hand, are transient or dynamic consonants that have a sequence of articulatory events.

Glides may also be considered as transient phonemes because of the difficulty of sustaining them. These sounds are actually phoneme sequences: each diphthong consists of a vowel followed by a glide, and each affricative consists of a stop followed by a fricative. There are phonological conventions for considering these as individual phonemes, owing to distributional restrictions and durational phenomena.

2.3 Properties and Characteristics of Speech Signal

Speech signals have following inherent properties:

- In a linear manner, it can be noticed that speech is a sequence of continually changing sounds.

- The characteristics of generated speech are highly based on sounds that are generated.
- The properties of the speech signal are highly dependent on the context in which the sounds are produced. It implies that way in which the sounds which generally take place before and after the current sound. This consequence is called speech sound co-articulation and it is the result of the vocal mechanism anticipating following sounds while producing the current sound, thereby changing the sound properties of the current sound. Some of the parameters of vocal cords like the positions, shapes, and sizes of the various articulators such as like teeth, lips, tongue, jaw, and velum alter slowly over time, thereby producing the required speech sounds.

2.3.1 Time and Frequency Domain Characteristics of Speech

Analyzing speech in the time domain often requires simple calculation and interpretation. Among the relevant features found readily in temporal analysis are waveform statistics, power, and F0. The frequency domain, on the other hand, provides the mechanisms to obtain the most useful parameters in speech analysis [3]. Most models of speech production assume a noisy or periodic waveform exciting a vocal-tract filter. The excitation and filter may be characterized in both time and frequency domain, but they are often more consistently and easily handled spectrally.

2.3.2 Waveforms

Time-domain speech signals are also called speech waveforms. They show the acoustic signals or sounds radiated as pressure variations from the lips while articulating linguistically meaningful information. In a complicated way, the amplitude of the speech waveform varies with time, including variations in the global level or intensity of the sound. The probability density function of waveform amplitudes, over a long time average, can be measured on a scale of speech level expressed as sound dB. This function has a form close to a double-sided (symmetrical) exponential at high amplitudes and is close to Gaussian at low amplitudes. The PDF (Probability Density Function) can be approximated by summation of Gaussian functions and exponential functions.

2.3.3 Fundamental Frequency

A speech waveform can be typically divided into two categories:

- A quasi-periodic part, which tends to be repetitive over a brief time interval.
- A noise-like part, which is of random shape.

For the quasi-periodic portion of the speech waveform, the average period is called a fundamental period or pitch period. Its inverse is called the fundamental frequency or pitch frequency, and is abbreviated F0. The fundamental frequency corresponds to vocal cord vibrations for vocalic sounds of speech. F0 in a natural speech waveform usually varies slowly with time. It can be 80 Hz or lower for male adults and above 300 Hz for children and some female adults. F0 is the main acoustic fundamental frequency and is crucial in tone languages for phoneme identification.

2.3.4 Overall Power

The overall power of the speech signal corresponds to the effective sound level of the speech waveform averaged over a long time interval [4]. In a quiet environment, the average power of male and female speech waveforms measured at 1 cm in front of a speaker's lips is about 58 dB. Male speech is on average about 4.5 dB louder (greater power) than female speech. Under noisy conditions, one's speech power tends to be greater than in a quiet environment. Further, not only are the overall power and amplitude increased, but also the details of the waveform change in a complicated way. In noisy environments a speaker tends to exaggerate articulation to enhance the listener's understanding, thereby changing the spectrum and associated waveform of the speech signal.

2.3.5 Overall Frequency Spectrum

Although the spectral contents of speech change over time, if the discrete-time Fourier transform (DFT) of the speech is taken in waveform over a long-time interval, the overall frequency range that covers the principal portion of the speech power can be estimated. Such information is important for plotting speech broadcast systems because the bandwidth of the systems depends on the overall speech spectrum rather than on the instantaneous speech spectrum. When such an overall frequency spectrum of speech is measured in a quiet environment, it is found that speech power is concentrated mainly at low frequencies. More than 80% of speech power lies below 1 kHz. Beyond 1 kHz, the overall frequency spectrum decays at a rate of about -12 dB per octave. Above 8 kHz, speech power is negligible, if the long-time Fourier transform analyzes only a quasi-periodic portion of the speech waveform and frequency components that are generated in harmonic relations and integer multiples of a common frequency. This common frequency, also called the lowest harmonic component, is the pitch.

2.3.6 Short-Time Energy

Speech has a continuously changing quasi-periodic signal. Sometimes, for dozens of pitch periods, the vocal tract outline and related facets of its excitation may remain continuous. The typical phoneme averages about 80 ms in duration; dynamic co articulation variations are greater [4].The short-time energy of the speech signal gives a suitable demonstration that reflects these amplitude variations. In general, short-time energy can be represented as

$$\mathrm{En} = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 \tag{2.1}$$

This turn of phrase can be written as

$$\mathrm{En} = \sum_{m=-\infty}^{\infty} [x(m)h(n-m)] \tag{2.2}$$

where $h(n) = w^2(n)$.

The behavior of the short-time energy representation can be determined by the selection of the impulse response $h(n)$ or the window. In any event, in speech processing it is considered that speech is a signal changing very slowly over defined time. This is a very popular method for short-time intervals of a very small number of periods at most. Also, it is important to dissect the signal into many analysis frames, letting the different parameters be evaluated frequently to model dynamic vocal-tract features.

2.3.7 Spectrogram

A spectrogram is a very good measure of the spectrum of a signal. It shows the spectral representation of the signal, which is used for forming an image, and the spectrogram gives information about how spectral density varies with time. Spectrograms are used to recognize phonetic sounds [5]. In the spectrogram there are basically two types of axis: the horizontal axis, or abscissa, which specifies time, and the vertical axis, the ordinate, which specifies frequency. The brightness or color of each point in the image is the three-dimensional (3D) representation of the amplitude of a specific frequency at a specific time. In the spectrogram, vice versa formation is also possible. The speech signal is normally expressed with a logarithmic amplitude axis, and frequency would be considered linear for highlighting harmonic constraint, or logarithmic to focus on musical or tonal relationships.

Spectrograms are typically shaped in one of two ways: it is judged as the results of filter bank operation on the signal. A filter bank might be considered as a series of

bandpass coefficients. Another way of measurement is using short-time Fourier series. These two methods actually generate two different quadratic time–frequency distributions but are very similar under specific conditions. The STFT is usually a digital process that creates the spectrogram. In the time domain, the digital and sampled data are divided into small frames, like chunks, which generally overlap, and these are then Fourier transformed to evaluate the amplitude of the frequency spectrum for each small frame. A vertical line in the image can be represented by each chunk, and it is a measurement of magnitude versus frequency for an exact point in time. The squared magnitude of the short-time Fourier transform (STFT) gives a spectrogram of the signal $s(t)$.

$$\text{spectrogram}(t, w) = |\text{STFT}(t, w)|^2 \tag{2.3}$$

- Spectrograms are necessary in removing speech hazards and in training for the portion of the speech signal. The spectrogram offers observation of phonetics and speech synthesis; it is important to determine the results of passing a test signal over a signal processor such as a filter to check its presentation. In the expansion of RF and microwave systems, high-definition spectrograms are used.

From the foregoing discussion it is seen that a spectrogram is not reflecting any content about the exact phase of the signal that is represented by spectrogram, and that it is not possible to completely reverse the signal again into the original format and thus no copy original signal can be generated. There is some phase information in the spectrogram that is seen in another form, as time delay which is the dual aspect of the instantaneous frequency. Steps for the spectrogram are as follows:

- Divide the speech signal into segments with equal length. This segment must be short enough that the frequency content of the signal does not change within a segment.
- Apply a window on each segment and compute its spectrum to obtain the short-time Fourier transform.
- Calculate the power of each spectrum segment in decibels. Display each segment side by side as images with a magnitude-dependent color map. A simple example of s spectrogram is given in Fig. 2.4.

2.3.8 Short-Time Average Zero Crossing Rate

Speech analysis in which frequency information is required needs to use spectral features extraction and apply the Fourier transform. However, a very basic parameter called zero crossing rate offers simple spectral information in some applications at little cost. Also, for a speech signal $s(n)$, a zero crossing takes place whenever $s(n) = 0$. Mainly, the entire scheme for evaluation of average zero crossing rate (ZCR) is visualized in Fig. 2.5, showing how the $x(n)$ zero crossing rate (Zn) is calculated.

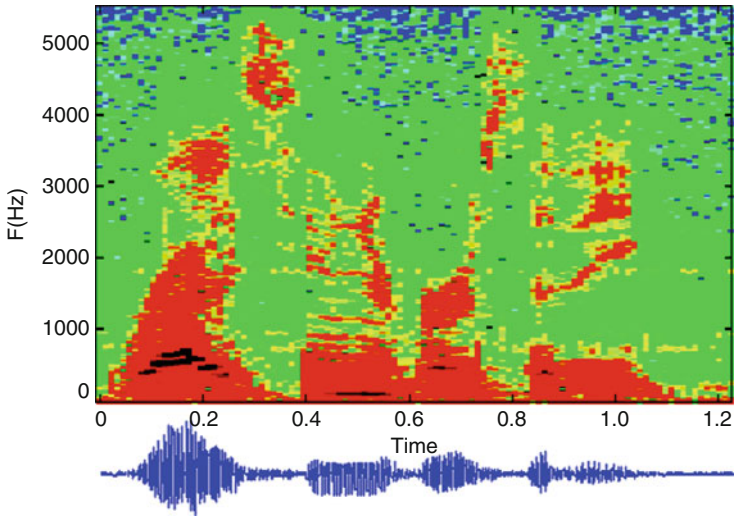

Fig. 2.4 Spectrogram of speech signal

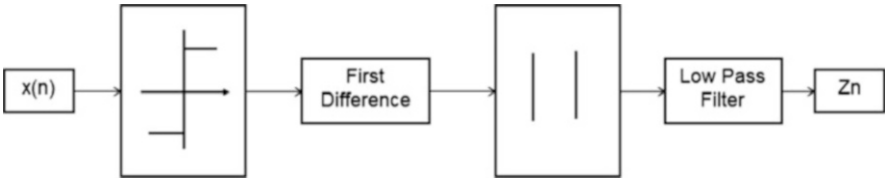

Fig. 2.5 Short-time average zero crossing

Taking the simple case of a sinusoid (instead of speech), ZCR (measured as zero crossings/s) yields two zero crossings/period, and thus it is F0 = ZCR/2. ZCR can accurately measure the frequency at which power is concentrated for all narrowband signals. Most short-time processing methods (both temporal and spectral) yield a parameter sequence in the form of a dynamic signal:

$$P(n) = \sum_{m=-\infty}^{\infty} T[s(m)]w(n-m) \tag{2.4}$$

where the speech $s(n)$ is subject to a possibly nonlinear transformation T and is weighted by the window $w(n)$ to limit the time range examined. The desired parameter $P(n)$ as specified by the nature of T appears as a signal with the original sampling rate, representing some speech characteristic averaged over the window duration. $P(n)$ is the convolution of $T[s(n)]$ and $w(n)$. Because $w(n)$ usually behaves as a low-pass filter, $P(n)$ is a smoothed version of $T[s(n)]$. Thus, the equation serves to calculate the ZCR with

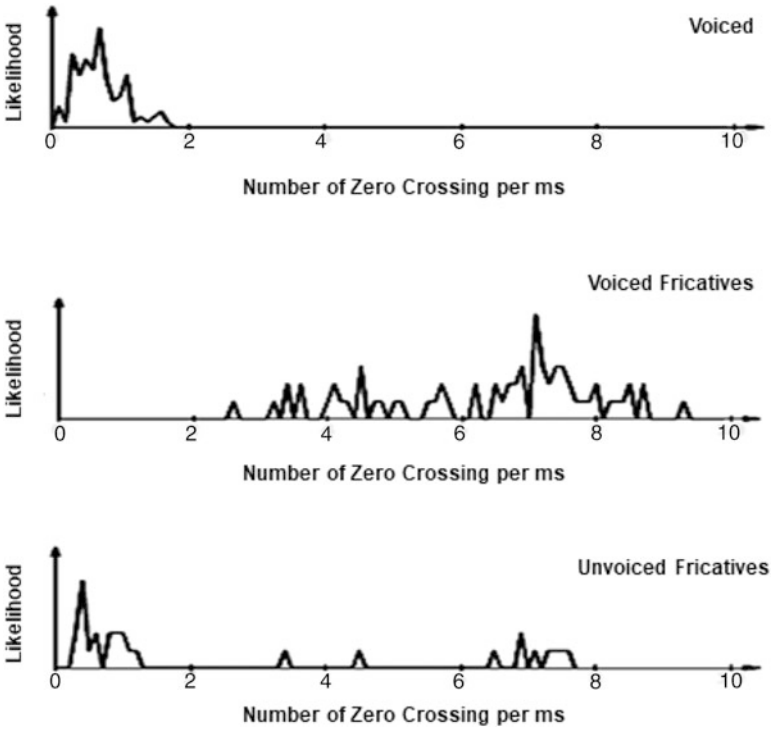

Fig. 2.6 Typical zero crossing distribution for voiced, unvoiced, and voiced fricatives

$$T[s(n)] = 0.5|\text{sign}(s(n)) - \text{sign}(s(n-1))| \tag{2.5}$$

where the algebraic sign of $s(n)$ is

$$\text{sign}(s(n)) = \left\{ \begin{array}{cc} 1, & \text{for } s(n) \geq 0 \\ -1, & \text{otherwise} \end{array} \right\} \tag{2.6}$$

The $w(n)$ is a rectangular window scaled by $1/N$ where N is the duration of the window to yield zero crossings/sample. The ZCR can be significantly decimated for data reduction purposes. Like speech power, the ZCR changes relatively slowly with vocal tract movements. The ZCR is functional mainly for voiced speech. Low-frequency power is generated by voiced speech. Unvoiced speech in contrast arises from broadband noise excitation, which raises mainly high frequencies and tends to use short vocal tracts. Figure 2.6 shows the organization of zero crossing rates for voiced, unvoiced, and voiced fricatives in clear view.

In general, high and low ZCR, about 4900 and 1400 crossings/s, correspond to unvoiced and voiced speech, respectively. For sonorant sounds, the ZCR follows F1 well, as F1 has more energy than other formants except in low back vowels, where F1 and F2 are close, and ZCR is thus usually above F1. Voiced fricatives, on the

other hand, have bimodal spectra, with voicebar power at very low frequency and frication energy at high frequency; hence, ZCR in such cases is more variable. Differing from short-time power, the ZCR is quite sensitive to noise, especially any low-frequency bias that may displace the zero-amplitude axis.

References

1. Rabiner, L. R., & Schafer, R. W. (1978). *Digital processing of speech signals. (Prentice-Hall Series in Signal Processing)*. Englewood Cliffs, NJ: Prentice Hall.
2. Quatieri, T. F. (2006). *Discrete-time speech signal processing: Principles and practice*. India: Pearson Education.
3. Kamkar-Parsi, A. H., & Bouchard, M. (2009). Improved noise power spectrum density estimation for binaural hearing aids operating in a diffuse noise field environment. *IEEE Transactions on Audio, Speech, and Language Processing, 17*(4), 521–533.
4. Lei, J., Yang, J., Wang, J., & Yang, Z. (2009). A robust voice activity detection algorithm in nonstationary noise. In *Industrial and Information Systems, 2009. IIS '09. International Conference* (pp. 195–198). Piscataway: IEEE.
5. Li, M., McAllister, H. G., Black, N. D., & De Perez, T. A. (2001). Perceptual time-frequency subtraction algorithm for noise reduction in hearing aids. *IEEE Transactions on Biomedical Engineering, 48*(9), 979–988.

Chapter 3
Introduction of Adaptive Filters and Noises for Speech

3.1 Adaptive Filter

In general, in most live applications and in the environment, information is not available about related incoming information statistics. At that juncture, the adaptive filter is a self-regulating system that helps process the recursive algorithm. Moreover, it is a self-regulating filter that uses some training vector that delivers various comprehensions of a desired response that can be merged with reference to the incoming signal. First, input and training are compared; accordingly, an error signal is generated, and that is used to adjust some previously assumed filter parameters under the effect of the incoming signal. Filter parameter adjustment continues until a steady-state condition is reached [1].

Insofar as application of noise reduction from speech is concerned, adaptive filters can give the best performance, because noise is somewhat similar to randomly generated signals and it is very difficult to measure its statistic every time. Design of a fixed filter is a completely failed phenomenon for time varying noisy speech signal. Some of the signal changes at a very fast rate in the context of information in the process of noise cancellation, which requires the help of self-regularized algorithms than can converge rapidly. Least mean square (LMS) and normalized LMS (NLMS) are generally used for signal enhancement, as they are very simple and efficient. Because of the very fast convergence rate and efficiency, recursive least square (RLS) algorithms are the most popular in specific kinds of applications. A brief overview of functional characteristics for adaptive filters is described in the following sections.

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_3

3.2 LMS Adaptive Filter

In signal processing, there is a wide variety of stochastic gradient algorithms in that the LMS algorithm is an imperative component of the family. The LMS algorithm can be differentiated from the steepest descent method by the term stop chiastic gradient, for which a deterministic gradient works; a filter for inputs is usually used in recursive computation, which has the noteworthy feature of simplicity and for which it is made standard over other linear adaptive filtering algorithms. Moreover, it does not require matrix inversion [2].

The LMS algorithm requires two fundamental processes on the signal, and it is of the linear type of adaptive algorithm [1]:

1. A residue error can be predicted by comparing the output from the linear filtering phenomena and according to the response to the input signal with the necessary response of the signal. The two parts are mainly from the filtering process.
2. Estimated error takes part in generating the updating filter vector when automatically adjusting the parametric model.

The mixture of these two developments working collectively creates a closed loop with a reverse mechanism, as illustrated in Fig. 3.1. The LMS algorithm lies in the nature of the transversal filter shown in Fig. 3.2 [1]. This module is used for performing the adaptive control process on the tap weights vector of the transversal filter to enhance the designation for the adaptive weight control mechanism [3] (Fig. 3.3).

In an adaptive filter, the most important part is the tap input from the fundamentals; the tap input vector $u(n)$ is matrix length M and one row, where the number of delay elements is presented with M length vector; these inputs extent a multidimensional space denoted by $\tilde{U}n$. Correspondingly, the tap weights are the

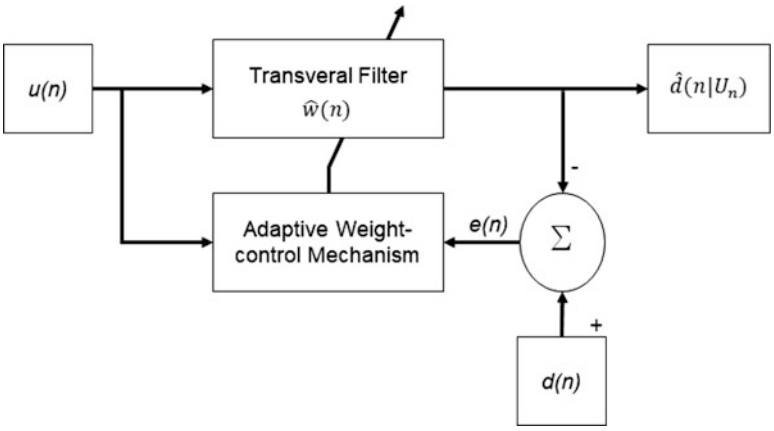

Fig. 3.1 Concept of an adaptive transversal filter

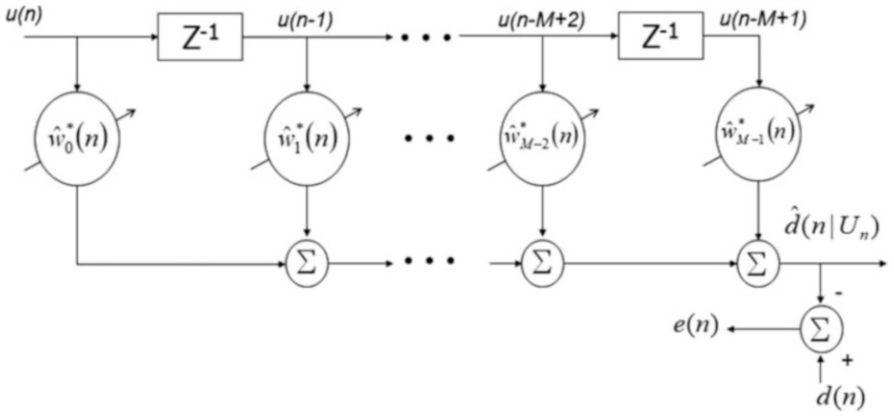

Fig. 3.2 Detailed structure of the transversal filter component

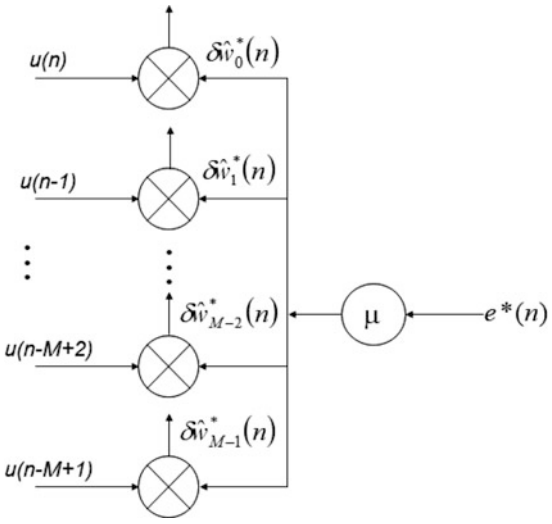

Fig. 3.3 Detailed structure of the adaptive weight control mechanism

main elements. By taking the basis of the wide sense stationary process, the value computed for the LMS algorithm gives output that is very near to the Wiener solution of the filter; this happens when the number of repetitions, n, tends toward infinity.

In the filtering process, the wanted reaction $d(n)$ is supplied for processing and collectively with the tap input vector $u(n)$. This fetched input is very important and is used in the transversal filter, which creates an output $d(n \mid \tilde{U}n)$ used as an evaluation of the required response $d(n)$. Then, an estimation error $e(n)$ can be quoted, and that representation is used to take the modification between the actual needed response and the actual filter output relationship of $e(n)$ and $u(n)$ can be shown, as in the

output end of Fig. 3.2. The detailed values of the vector obtained are helpful to manage the closed path around the feedback mechanism of the system.

Figure 3.3 gives the depth of the adaptive weight control process. A tap input vector $u(n-k)$ and the inner product of the estimation error $e(n)$ are purposely calculated for various values of k starting with 0 to $M-1$. The μ is a defined scaling factor in the process of calculation and is a nonnegative quantity that is also known as a step size of the process, which can be clearly seen in Fig. 3.3.

Comparing the control mechanism for the LMS algorithm (Fig. 3.3) with that for the method of steepest descent, it can be seen that the LMS algorithm in the process takes the convolution of $u(n-k)\,e^*(k)$, and it can be considered as a prediction of element k in the gradient vector $J(n)$ that follows the rules of steepest descent concept in the mechanism. In other words, the expectation operator is removed from all the paths in Fig. 3.3.

It is assumed that the tap input and the desired response can be computed from a jointly wide sense stationary environment. In adaptive filtering, a multiple regression model is taken into consideration in which some characteristics and the parametric vector are unknown, and hence the need for self-adjusting filtering and a linear change of $d(n)$. For computing, the tap vector $w(n)$ changes and goes down; at that time the ensemble sums up its average error performance surface with a deterministic trajectory. Now, that surface terminates on the vector of the Wiener solution. It is better and suitable for Wiener solutions that $\hat{w}(n)$ (different from $w(n)$ computed by the LMS algorithm) follow a nonpredictable motion around the minimum point of the error performance surface, and it can be observed that this motion is a form of Brownian motion for small μ [4].

Earlier it was pointed out that the LMS filter involves feedback in its operation, which raises the related issue of stability. In this context, a meaningful criterion is to require that as $J(n)$ tends toward $J(\infty)$ with n tends toward ∞ in a general manner.

It can be recognized that $J(n)$ is the outcome of the LMS process and is in terms of mean square error (MSE) at time n; its final value $J(\infty)$ is a constant. By the LMS algorithm, if the step size parameter is adjusted relative to the spectral content of the tap inputs then it will satisfy the following condition of stability in the mean square.

The excess MSE can be defined as the difference between the final value $J(\infty)$ and the minimum values $J_{\min}$ attained by the Wiener solution. This difference indicates the price paid for using the adaptive (stochastic) method to cover and calculate the tap weights in the LMS filter instead of a deterministic approach, as in the method of steepest descent. It is interesting here to note that the complete feedback mechanism acting around the tap weights acts similarly to a low-pass filter, whose average time constant varies inversely to the step size parameter. As a consequence it is necessary to adjust a small value to the step size parameter and tend toward an adaptive process slowly in the convergence direction, and because of that the effects of gradient noise on tap weights are heavily filtered out.

The most advantageous feature of the LMS adaptive algorithm is that it is very straightforward in implementation and still very able to adjust efficiently to the outer environment per requirements. The only performance limitation arises by the choice of the step size parameters.

3.2.1 Least Mean Square Adaptation Algorithm

Using the steepest descent algorithm, it is mainly concentrated to make accurate measurement of the vector named gradient $\nabla J(n)$ at every regular iteration. It is also possible to compute tap weight vector if the step size parameter is suitably selected. Step size selection and tap weight vector optimally computed would be related to the optimum Wiener solution as the advance knowledge of both mentioned matrices, such as the correlation matrix R of the tap input and the cross-correlation vector P between the tap inputs and the desired response.

To achieve an estimation of $\nabla J(n)$, a very important method is to take another estimate of the correlation matrix R and the cross-correlation vector p in the formula, which is produced here for convenience [5].

$$\nabla J(n) = -2p + 2Rw(n) \tag{3.1}$$

A very obvious choice of predictors is computation by using instantaneous estimates for R and p that are collaborated by the different discrete magnitude values of the tap input vector and necessary response, defined respectively by

$$\breve{R}(n) = u(n)u^H(n) \tag{3.2}$$

$$\widehat{P}(n) = u(n)d^*(n) \tag{3.3}$$

Compatibly, the gradient vector instantaneous value can be defined as

$$\widehat{\nabla} J(n) = -2u(n)d^*(n) + 2u(n)u^H(n)\widehat{w}(n) \tag{3.4}$$

Note that the estimate $\nabla J(n)$ may also be viewed as the gradient operator applied to the instantaneous squared error $|e(n)|^2$. Substituting the estimate of for the gradient vector $J(n)$ in the steepest descent algorithm is described; the following relationship can be taken to be

$$\widehat{w}(n+1) = \widehat{w}(n) + \mu u(n)\left[d^*(n) - u^H(n)\widehat{w}(n)\right] \tag{3.5}$$

Here the tap weight vector has been used to distinguish it from the values obtained by using the steepest descent algorithm. Equivalently, it may be written that the result in the form of three basic relationships as follows:

1. Filter output:

$$Y(n) = \widehat{w}^H(n)u(n) \tag{3.6}$$

2. Estimation error or error signal:

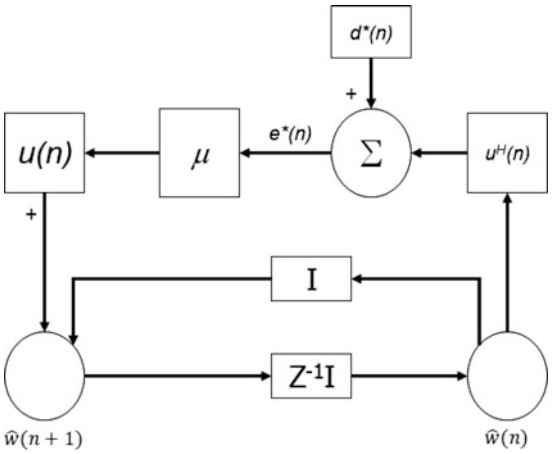

Fig. 3.4 LMS signal flow graph

$$e(n) = d(n) - y(n) \tag{3.7}$$

3. Tap weight adaptation:

$$\widehat{w}(n+1) = \widehat{w}(n) + \mu \cdot u(n)e^*(n) \tag{3.8}$$

These equations show the estimation error $e(n)$, the calculation of which is decided on the present estimate of the tap weight vector, $\hat{w}(n)$. It is important to take into consideration that the $\mu u(n)$ $e^*(n)$ term shows an adjustment that is applied to the present estimate of the tap weight vector, $\hat{w}(n)$.

Mainly, the algorithm explained by the mentioned equations is the complex form of the LMS algorithm. Inputs required by the algorithm should be most recent and fresh in terms of error vector, input vector, etc. Here inputs are in the terms of stochastic range, and the allowed set of directions along which it can go ahead from one iteration process to the next is nondeterministic in nature and can be thought of as consisting of real gradient vector directions.

The LMS algorithm is the most popular because of this convergence speed, but selection of step size is very important in the success of the algorithm. Figure 3.4 shows a LMS algorithm mechanism in the form of a signal flow graph. This model bears a close resemblance to the feedback model describing the steepest descent algorithm. The signal flow graph in Fig. 3.4 clearly demonstrates the simplicity of the LMS algorithm. In particular, it can be found in this Fig. 3.4 that the per equation iteration LMS algorithm requires only $2M + 1$ complex multiplications and $2M$ complex, where M is the number of tap weights in the basic transverse filter. A comparatively large variance can be achieved by the instantaneous estimates of R and p. By the first step analysis it can be seen that the LMS algorithm cannot perform well because it uses present estimations. Still, it is a dynamic feature of LMS that it is recursive in nature, with the result that the algorithm itself effectively

averages these estimates, in some sense, during the course of adaptation. The LMS algorithm can be summarized as in following section.

In Box 3.1, a summary of the LMS algorithm is represented in which equations are incorporated. Box 3.1 also includes a constraint on the allowed value of the acceptable step size parameters, which is needed to ensure that the algorithm converges. More is said on this necessary condition for convergence.

Box 3.1 Summary of the LMS Algorithm
Parameter: M = number of taps (i.e., filter length), μ = step size parameter, $0 < \mu < 2/MS_{\max}$.
where $S_{\max}$ is the maximum value of the power spectral density of the tap inputs $u(n)$ and the filter length M is moderate to large.

Initialization: If prior knowledge of the tap weight vector $\hat{w}(n)$ is available, use it to select an appropriate value for $\hat{w}(n)$. Otherwise, set $\hat{w}(0) = 0$.

- Given $u(n) = M$ by 1 tap input vector at time = $[u(n), u(n-1),\ldots, u(n-m+1)]$ $T(n)$ = desired response at time n
- To be computed $\hat{w}(n+1)$ = estimate of tap weight vector at time $n+1$

Computation: For $n = 0, 1, 2, \ldots$, compute: $e(n) = d(n) - \hat{w}^H(n)u(n)$, $\hat{w}(n+1) = \hat{w}(n) + \mu u(n)e^*(n)$

3.2.2 Statistical LMS Theory

Previously, it was inferred that the LMS filter is a linear adaptive filter, "linear" in the sense that its physical implementation is built around a linear combiner. In reality, however, the LMS filter is a highly complex nonlinear estimator that violates the principles of superposition and homogeneity [6]. Let $y_1(n)$ denote the response of a system to an input vector $u_1(n)$. Similarly, let $y_2(n)$ denote the response of the system to another input vector $u_2(n)$. For a system to be linear, the composite input vector $u_1(n) + u_2(n)$ must result in a response equal to $y_1(n) + y_2(n)$; this result is called the principle of superposition.

Furthermore, a linear system must satisfy the homogeneity property; that is, if $y(n)$ is the response of the system to an input vector $u(n)$, then the response of the system to the scaled input vector, where 'a' is a scaling factor. Consider now the LMS filter. Starting with the initial conditions $w(0) = 0$ and the frequent application of the weight update gives

$$w(n) = \mu \sum_{i=\infty}^{n-1} e^*(i) \cdot u(i) \tag{3.9}$$

The following equations show the input–output relationship of the LMS algorithm:

$$y(n) = \widehat{w}^H(n) \cdot u(n) \tag{3.10}$$

$$y(n) = \mu \sum_{i=\infty}^{n-1} e(i) \cdot u^H(i) u(n) \tag{3.11}$$

Recognizing that the error signal $e(i)$ is decided by the input vector $u(i)$, it can be defined from the equation output of the filter that it takes a nonlinear nature and its function of input. The properties of superposition and homogeneity are thereby both violated by the LMS filter.

Thus, although the LMS filter is very simple in physical terms, its mathematical analysis is profoundly complicated because of its highly nonlinear nature. To proceed with a statistical analysis of the LMS filter, it is convenient to work with the weight error vector rather than the tap weight vector itself. The weight error vector in the LMS filter can be denoted by

$$\varepsilon(n) = w_0 - \widehat{w}(n) \tag{3.12}$$

Subtracting the equation from the optimum tap weight vector w_0 and using the definition of Equation 3.9 to eliminate $w(n)$ from the adjustment term on the other side, it can be rearranged in the following form that the LMS algorithm in terms of the weight error vector $\varepsilon(n)$ is

$$\varepsilon(n+1) = \left[I - \mu u(n) u^H(n)\right] - \mu u(n) e_0^*(n) \tag{3.13}$$

where I is the identify matrix and $e_0(n) = d(n) - w_0^H u(n)$ is the estimation error produced by the optimum Wiener filter.

3.2.3 Direct Averaging Method

It is critical to analyze the convergence nature of such a stochastic algorithm in an average sense; the direct averaging method is useful. According to this method, the possible outcome of the stochastic difference equation is operated under the consideration of a much less valued step-size parameter by virtue of the low-pass filtering action of the nearby LMS algorithm and similar to the answer of another stochastic difference equation with system matrix equal to the ensemble average:

$$E\left[I - \mu u(n) u^H(n)\right] = I - \mu R \tag{3.14}$$

R can be recognized as the correlation matrix of the tap input vector $u(n)$ [6]. More specifically, it may be replaced the stochastic difference representation with another stochastic difference representation described by:

$$\varepsilon_0(n+1) = (I - \mu R)\varepsilon_0(n) - \mu u(n)e_0^*(n) \tag{3.15}$$

3.2.4 Small Step Size Statistical Theory

The development of the statistical LMS theory to small step sizes should be restricted, embodied in the following assumptions:

Assumption I The LMS algorithm can act as a low-pass filter with a much lower cutoff because the step size parameter μ is small [7].

Under this assumption, the zero-order terms $\varepsilon_0(n)$ and $k_0(n)$ as approximations to the actual $\varepsilon(n)$ and $k(n)$, respectively, might be used. To illustrate the validity of Assumption I, consider the example of and LMS filter using a single weight. For this example, the stochastic difference equation simplifies to the scalar form:

$$\varepsilon_0(n+1) = \left(1 - \mu\sigma_u^2\right)\varepsilon_0(n) + f_0(n) \tag{3.16}$$

where σ_u^2 is the variance $u(n)$. This difference equation represents a transfer function with a single pole at a given equation with a low-pass filter in nature:

$$z = \left(1 - \mu\sigma_u^2\right) \tag{3.17}$$

For small μ, the pole lies inside of, and very close to, the z plane unity circle, which implies a very low cutoff frequency.

Assumption II The actual logic by which the observable data can be generated is that the desired response $d(n)$ is represented by a linear multiple regression model that is very similar to the Wiener filter and which is given by

$$d(n) = w_0^H u(n) + e_0(n) \tag{3.18}$$

where the irreducible estimation error $e_0(n)$ is a process of comparing to white noise that is not dependent on the input vector values [8].

The characterization of $e_0(n)$ as white noise means that its successive samples are uncorrelated, as given by

$$E\left[e_0(n)e_0^*(n-k)\right] = \begin{cases} nJ_{\min} & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases} \tag{3.19}$$

The essence of the second assumption is that it can be shown that providing the use of a linear multiple regression model is justified and the number of coefficients in the Wiener filter is nearly the same as the level of the regression model. The statistical independence of $e_0(n)$ from $u(n)$ is stronger than the principle of

orthogonality. The choice of a small step size according to Assumption (I) is certainly under the designer's control. To match the LMS filter length of the multiple regression model with a suitable order in Assumption II requires the use of a model selection criterion.

Assumption III Desired response and the input vector are jointly Gaussian. Thus, the small step size theory to be developed shortly for the statistical characterization of LMS filters applies to one of two possible scenarios: Assumption II holds, whereas in the other scenario, Assumption III holds. Between them, these two scenarios cover a wide range of environments in which the LMS filter operates, most importantly in deriving the small step size theory.

3.2.5 Natural Modes of the LMS Filter

Under Assumption I, Butterweck's interactive procedure reduces to the following pair of equations:

$$\varepsilon_0(n+1) = (I - \mu R)\varepsilon_0(n) + f_0(n) \tag{3.20}$$

$$f_0(n) = -\mu u(n) e_0^*(n) \tag{3.21}$$

Before proceeding further, it is informative to transform the difference equation into a simpler form by applying the unitary similarity transformation to the correlation matrix R [1]. It can be obtained that

$$Q^H R Q = \Lambda \tag{3.22}$$

where Q is a unitary matrix whose columns constitute an orthogonal set of eigenvectors associated with eigenvalues of the correlation matrix R and Λ is a diagonal matrix that consists of the eigenvalues. To achieve the desired simplification, the definition can be introduced also as

$$V(n) = Q^H \varepsilon_0(n) \tag{3.23}$$

Defining property of the unitary matrix Q, namely:

$$QQ^H = I \tag{3.24}$$

I can be represented as the identity matrix,

$$V(n+1) = (I - \mu\Lambda)V(n) + \Phi(n) \tag{3.25}$$

where the new vector $\Phi(n)$ is defined in terms of $f_0(n)$ by the transformation

$$\Phi(n) = Q^H f_0(n) \tag{3.26}$$

For a partial characterization of the stochastic force vector $\Phi(n)$, its mean and correlation matrix over an ensemble of LMS filters may be expressed as follows:

1. First compute the mean value of the stochastic force vector $\Phi(n)$. Deliberately, it must be zero:

$$E[\Phi(n)] = 0 \quad \text{for all } n \tag{3.27}$$

2. $\Phi(n)$ is a diagonal matrix and is of the correlation matrix of the stochastic force directional quantity; that is,

$$E\left[\Phi(n)\Phi^H(n)\right] = \mu^2 J_{\min}\Lambda \tag{3.28}$$

where $J_{\min}$ is a minimum mean square error that is generated by the Wiener filter.

3.2.6 *Learning Curves for Adaptive Algorithms*

The statistical work of adaptive filters can be observed by ensemble average learning curves. The two identical types of learning curves are as under [1].

1. The first type is the MSE learning curve. The MSE curve produces ensemble averaging of the squared estimation error. The means plot of mean values in the learning curve is $J(n) = E[|e(n)|^2]$ versus the iteration n.
2. The second most important is the mean square deviation (MSD) learning curve, which is processed by taking ensemble averaging of the squared error deviation $\left\|\varepsilon(n)\right\|^2$. The mean square deviation versus the iteration n is plotted in the second learning curve.

$$D(n) = E\left[\|\varepsilon(n)\|^2\right] \tag{3.29}$$

The estimation error generated by the LMS filter is expressed as

$$\begin{aligned} e(n) &= d(n) - \widehat{w}^H(n)u(n) \\ e(n) &= d(n) = w_0^H u(n) + \varepsilon^H(n)u(n) \\ e(n) &= e_0(n) + \varepsilon^H(n)u(n) \end{aligned} \tag{3.30}$$

$$e(n) = e_0(n) + \varepsilon_0^H(n)u(n) \text{ for } \mu \text{ small} \tag{3.31}$$

where $e_0(n)$ is the estimation error and $\varepsilon_0(n)$ is the zero-order weight error vector of the LMS filter. Hence, the mean square error produced is shown by following iterations:

$$\begin{aligned} J(n) &= E\left[|e(n)|^2\right] \\ J(n) &\approx E\left[\left(e_0(n) + \varepsilon_0^H(n)u(n)\right)\left(\varepsilon_0^*(n)\right) + u^H((n)\varepsilon_0(n))\right] \end{aligned} \tag{3.32}$$

$$J(n) = J_{\min} + 2\mathrm{Re}\left\{E\left[\varepsilon_0^{*H}(n)u(n)\right]\right\} + E\left[\varepsilon_0^H(n)u(n)u^H(n)\varepsilon_0(n)\right] \tag{3.33}$$

where $J_{\min}$ is the minimum MSE; Re denotes the real part of the quality enclosed between the braces. The following reasons depend on which scenario applies and so that the right-hand side of the equation gets null value: under Assumption II, the irreducible estimation error $e_0(n)$ produced by the wiener filter is statistically independent. At n iterations, the zero-order weight error vector $\varepsilon_0(n)$ depends on the past values of $e_0(n)$, a relationship that follows from the iterated use [9]. Hence, here it can be written:

$$E\left[e_0^*(n)\varepsilon_0^H(n)u(n)\right] = E\left[e_0^*(n)\right]E\left[\varepsilon_0^H(n)u(n)\right] = 0 \tag{3.34}$$

The null result of this equation also holds under Assumption III. For the kth components of $\varepsilon_0(n)$ and $u(n)$, it can be expected that

$$E\left[e_0^*(n)\varepsilon_0^*, k(n)u(n-k)\right], \quad k = 0, 1, \ldots, M-1 \tag{3.35}$$

Assuming that there are jointly Gaussian input vector and desired response and the estimation error $e_0(n)$ is therefore also Gaussian, then applying the identity described, it can be obtained immediately that

$$E\left[e_0^*(n)\varepsilon_0^*, k(n)u(n-k)\right] = 0 \quad \text{for all } k \tag{3.36}$$

3.2.7 *Comparison of the LMS Algorithm with the Steepest Descent Algorithm*

When the coefficient set value of the transversal filter approaches the optimum value, and it is defined by Wiener equation, then the minimum MSE $J_{\min}$ is realized. This condition is recognized as the ideal condition when the number of iterations reaches infinity by the steepest descent algorithm. The steepest descent algorithm measures the gradient vector at each step in the iterations of the algorithm [1]. But in the case of LMS, it depends on a noisy momentary estimation with gradient vector; also, with that the tap weight vector estimate $\hat{w}(n)$ for large n can only fluctuate. Thus, after too

many loop executions in the form of iterations, the LMS algorithm results in a MSE $J(\infty)$ that is greater than the minimum MSE $J_{\min}$. The amount by which the actual value of $J(\infty)$ is greater than $J_{\min}$ is the excess MSE.

A well-defined learning curve has been shown by the steepest descent algorithm, gained by plotting the number of iterations versus MSE. The learning involves a sum of descending exponentials, which equates the number of tap coefficients, while in individual applications of the LMS algorithm, the noisy decaying exponentials representation is contained by the learning curve. The noise amplitude usually generates small values as the step size parameter μ is reduced; in the limit the learning curve of the LMS filter assumes a deterministic character.

The adaptive transversal filter is in the form of an ensemble component, each of which is assumed to use the LMS algorithm with the same step size μ and the same initial tap weight vector $\hat{w}(0)$. In the case of the adaptive filter, it can be considered to give stationary ergodic inputs that are selected at random for the same statistical population. The learning curves, which are noisy, are calculated for this ensemble of adaptive filters.

Thus, two entirely different ensemble-averaging operations are used in the steepest descent and LMS algorithms for determining their learning curves. In the steepest descent algorithm, the correlation matrix R and the cross-correlation vector p are initially computed using ensemble-averaging operations, which is useful to the populations of the tap inputs and the desired response calculation. These values are then used to calculate the learning curve of the algorithm. In the LMS algorithm, noisy learning curves are computed for an ensemble of adaptive LMS filters with identical parameters. The learning curve is then smoothed by averaging over the ensemble of noisy learning curves.

3.3 Normalized Least Mean Square (NLMS) Adaptive Filter

In the standard form of an LMS filter, the tap weight vector of the filter at iteration $n + 1$ receives the necessary adjustment and gives the product of three terms:

- The step size parameter μ, which is subject to design concept.
- The tap input vector $u(n)$, which is actual input information to be processed.
- The estimation error $e(n)$ for real valued data, or its complex conjugate $e^*(n)$ for complex valued data, which is calculated at iteration n.

The adjustment is directly proportional to the tap input vector $u(n)$. As a result, the LMS filter suffers from a gradient noise amplification problem in the case when $u(n)$ is very large. As a solution, a normalized LMS filter can be used. The term normalized can be considered because the adjustment given to the tap weight vector at iteration $n + 1$ is "normalized" with respect to the squared Euclidean norm of the tap input vector $u(n)$ [1].

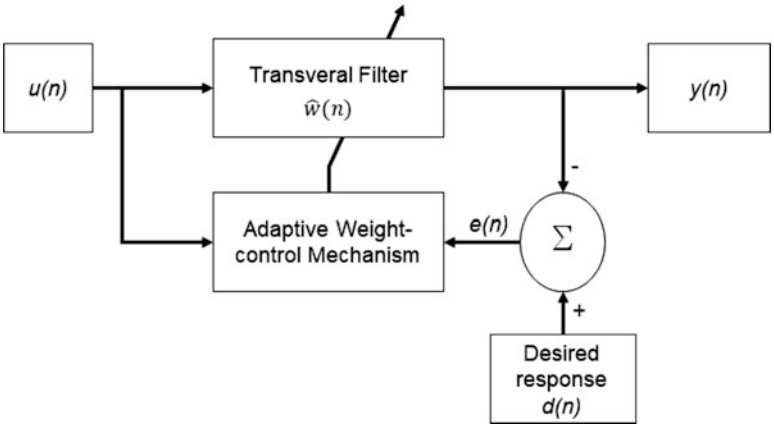

Fig. 3.5 Block diagram of NLMS filter

3.3.1 Structure and Operation of NLMS

In the form of a constructional view, the normalized LMS filter is exactly the same as the standard LMS filter (Fig. 3.5). The fundamental concept of both filters is the transversal filter.

Highlighted contrast in both types of algorithm is in the weight upgradation mechanism. One vector, which is long with values M in one row, known as a tap input vector, generates an output that is generally deducted from the desired response to generate the estimation error $e(n)$ [1]. Very natural modification in the vector modification directs the new algorithm, which is known as a normalized LMS algorithm.

The normalized LMS filter gives minimal disturbance and may be stated as follows: gradually by different iterations the weight vector will change in straight weight change step by step. It is controlled by updated filter output and its proposed values.

To cast this principle in mathematical terms, assume $\hat{w}(n)$ denotes the previous weight vector of the filter at iteration n and $\hat{w}(n + 1)$ denotes its modified weight vector at the next iteration. Selected conditions for implementing the normalized LMS filter may be articulated in the category of constrained optimization that follows. Determination of updated tap weight vector $\hat{w}(n + 1)$ is possible given the tap input vector $u(n)$ and desired response $d(n)$. Change can be highlighted in the Euclidean norm:

$$\delta\widehat{w}(n+1) = \widehat{w}(n+1) - \widehat{w}(n) \tag{3.37}$$

Subject to the constraint

$$\widehat{w}^H(n+1)u(n) = d(n) \tag{3.38}$$

The described constraint can be analyzed in the form of an optimization problem, and in that the method of Lagrange multipliers can be used.

$$J(n) = \|\delta\widehat{w}(n+1)\|^2 + \mathrm{Re}\left[\lambda \times \left(d(n) - \widehat{w}^H(n+1)u(n)\right)\right] \tag{3.39}$$

where λ is the complex valued Lagrange multiplier and the asterisk denotes complex conjugation. The squared Euclidean norm $\|\delta\hat{w}(n+1)\|^2$ is, naturally, real valued. The real part operator, denoted by Re [·] and applied to the second term, ensures that the contribution of the constraint to the cost function is similarly real valued. The most important cost function $J(n)$, which is a quadratic function in $\hat{w}(n+1)$, shown by expanding into

$$\begin{aligned} J(n) = {} & (\widehat{w}(n+1) - \widehat{w}(n))^H(\widehat{w}(n+1) - \widehat{w}(n)) \\ & + \mathrm{Re}\left[\lambda \times \left(d(n) - \widehat{w}^H(n+1)u(n)\right)\right] \end{aligned} \tag{3.40}$$

To find the optimum value of the updated weight vector that minimizes the cost function $J(n)$, the procedure is as follows [2].

Differentiate the cost function $J(n)$ with respect to $\hat{w}(n+1)$. Then, following the rule for differentiating a real-valued function with respect to a complex-valued weight vector as shown,

1.

$$\frac{\partial J(n)}{\partial \widehat{w}^*(n+1)} = 2(\widehat{w}(n+1) - \widehat{w}(n)) - \lambda \times u(n) \tag{3.41}$$

Setting this result equal to zero and solving for the optimum value $\hat{w}(n+1)$,

$$\widehat{w}(n+1) = \widehat{w}(n) + 1/2\lambda \times u(n) \tag{3.42}$$

2. Solve for the unknown multiplier λ by substituting the result of step 1 [i.e., the weight vector $\hat{w}(n+1)$] into the constraint of the formula. Doing the substitution, first it can be written that

$$\begin{aligned} d(n) = \widehat{w}^H(n+1)u(n)d(n) = (\widehat{w}(n) + \tfrac{1}{2}\lambda \times u(n))^H u(n)d(n) = \widehat{w}^H(n)u(n) \\ + \tfrac{1}{2}\lambda \times u^H(n)u(n)d(n) = \widehat{w}^H(n)u(n) + \tfrac{1}{2}\lambda\|u(n)\|^2 \end{aligned} \tag{3.43}$$

Then, solving for λ, it can be obtained that

$$\lambda = \frac{2e(n)}{\|u(n)\|^2} \tag{3.44}$$

where $e(n) = d(n) - \widehat{w}^H(n)u(n)$is the error signal.

3. Combine the results of steps 1 and 2 to prepare the optimal value of the incremental change, $\delta\hat{w}(n + 1)$:

$$\begin{aligned} \delta\widehat{w}(n+1) &= \widehat{w}(n+1) - \widehat{w}(n) \\ \delta\widehat{w}(n+1) &= 1/\|u(n)\|^2 u(n)e^*(n) \end{aligned} \tag{3.45}$$

To work out control over the change in the tap weight vector in the gradual iteration process, keep the direction constant for the vector by introducing a positive real scaling factor denoted by $\widetilde{\mu}$. That is, the change can be redefined simply as

$$\begin{aligned} \delta\widehat{w}(n+1) &= \widehat{w}(n+1) - \widehat{w}(n) \\ \delta\widehat{w}(n+1) &= \left[\widetilde{\mu}/\|u(n)\|^2\right] \cdot u(n)e^*(n) \end{aligned} \tag{3.46}$$

Equivalently, it can be written that

$$\widehat{w}(n+1) = \widehat{w}(n) + \left[\widetilde{\mu}/\|u(n)\|^2\right] \cdot u(n)e^*(n) \tag{3.47}$$

Indeed, this is the necessary recursion for calculation of the M by 1 tap weight vector in the normalized LMS algorithm. The foregoing equation justifies why a normalized term is used in this case: production of different vectors such as $u(n)$ and $e^*(n)$ is achieved. That product is normalized with respect to the squared Euclidean norm of the tap input vector $u(n)$. Comparing the recursion of the equation for the normalized LMS filter with that of the conventional LMS filter, the following observations might be taken [2].

- The adaptation constant is different in both algorithms. For LMS it is dimensionless and for NLMS it is with dimensions of inverse power.
- Setting $\mu(n) = \widetilde{\mu}/\|u(n)\|^2$, it can be viewed as the normalized LMS filter with an LMS filter with a time-varying step size parameter.
- Most prominently, the NLMS algorithm exhibits a potentially faster rate of convergence than that of the standard LMS algorithm for uncorrelated as well as correlated input data.

Conventional LMS suffers from problems of gradient noise removal and its increased value damages the quality of the system, whereas the normalized LMS filter arises as an issue when the tap input vector $u(n)$ is small, numerical, and calculation difficulties may arise because then it can categorized with a small value

for the squared norm $\|u(n)\|^2$. As a solution, a modified version of the calculation is shown here:

$$w(n+1) = w(n) + \left[\widetilde{\mu}/\delta + \|u(n)\|^2\right] u(n)e^*(n) \tag{3.48}$$

where $\delta > 0$.

3.3.2 Stability of the Normalized LMS Filter

Basically, the mechanism describe is responsible in the generation of wanted data, which is reproduced here for convenience of presentation [3].

$$d(n) = w^H u(n) + v(n) \tag{3.49}$$

In this equation, w is the model's unknown parameter vector and $v(n)$ is the additive disturbance. The tap weight vector $\hat{w}(n)$ computed by the normalized LMS filter is an estimate of w. The unevenness among the vectors is named w and $\hat{w}(n)$, which is accounted for and considered by the weight error vector:

$$\varepsilon(n) = w - \widehat{w}(n) \tag{3.50}$$

In further process subtracting iteration from w,

$$\varepsilon(n+1) = \varepsilon(n) - \widetilde{\mu}/\left[\|u(n)\|^2\right] u(n)e^*(n) \tag{3.51}$$

As already stated, the main concept of a normalized LMS filter is that of reducing the increased change $\delta\hat{w}(n+1)$ in the tap weight vector of the filter to another next computation $n+1$, subject to a constraint imposed on the updated tap weight vector $\hat{w}(n+1)$. In light of this idea, it is logical that the stability analysis of the normalized LMS filter on the basis of mean square deviation is as follows:

$$D(n) = E\left[\|\varepsilon(n)\|^2\right] \tag{3.52}$$

The mathematical process described next uses Euclidean norms of both sides and then application of adjustment as well as taking expectations: it is possible to write as

$$D(n+1) - D(n) = \widetilde{\mu}^2 E\left[\frac{|e(n)|^2}{\|u\|^2}\right] - 2\widetilde{\mu}E\left\{\mathrm{Re}\left[\frac{\xi_u(n)e^*(n)}{\|u(n)\|^2}\right]\right\} \tag{3.53}$$

where $\xi_u(n)$ is considered as the undisturbed error signal and can be cleared by

$$\xi_u(n) = (w - \widehat{w}(n))^H u(n) = \varepsilon^H(n)u(n) \tag{3.54}$$

It can be observed that the mean square deviation $D(n)$ decreases in exponential manner with a higher number of iterations n, and the NLMS filter acquires stability in the MSE and it gives that the normalized step size parameter $\widetilde{\mu}$ is bounded as follows:

$$0 < \widetilde{\mu} < \widetilde{\mu}_{\text{opt}} \tag{3.55}$$

$$\widetilde{\mu}_{\text{opt}} = \frac{\text{Re}\left\{E\left[\xi_u(n)e^*/\|u(n)\|^2\right]\right\}}{E\left[|e^2|/\|u(n)\|^2\right]} \tag{3.56}$$

From this equation, it can also be concluded that the highest value of the mean square deviation $D(n)$ is found at the center of the interval defined therein. After the process, the optimal step size is as seen.

3.3.3 Special Environment of Real Valued Data

For the case of real valued data, the normalized LMS algorithm takes the form [8]

$$\widehat{w}(n+1) = \widehat{w}(n) + \left[\widetilde{\mu}/\|u(n)\|^2\right]u(n)e(n) \tag{3.57}$$

Likewise, the optimal step size parameter of equations reduces to $\widetilde{\mu}_{\text{opt}}$ tractable: three assumptions can be introduced.

Assumption I The noticed variation in the given input signal energy $\|u(n)\|^2$ in successive iteration processes is less enough to validate the approximations:

$$E\left[\frac{\xi_u(n)e(n)}{\|u(n)\|^2}\right] \approx \frac{E[\xi_u(n)e(n)]}{E\left[\|u(n)\|^2\right]} \tag{3.58}$$

$$E\left[\frac{e^2(n)}{\|u(n)\|^2}\right] \approx \frac{E[e^2(n)]}{E\left[\|u(n)\|^2\right]} \tag{3.59}$$

Correspondingly, the formula of equation approximates to

$$\widetilde{\mu}_{\text{opt}} = \frac{E[\xi_n(n)e(n)]}{E[e^2(n)]} \tag{3.60}$$

Assumption II In the multiple regression the undisturbed error signal $\xi_u(n)$ is uncorrelated with the disturbance (noise) $v(n)$ for the descried response $d(n)$.The disturbed error signal $e(n)$ is related to the undisturbed error signal $\xi_u(n)$:

$$e(n) = \xi_u(n) + v(n) \tag{3.61}$$

Using this equation and then invoking Assumption II:

$$E[\xi_u(n)e(n)] = E[\xi_u(n)(\xi_u(n) + v(n))] = E\left[\xi_u^2(n)\right] \tag{3.62}$$

By simplifying the equation, the formula for the optimal step size is

$$\mu_{\text{opt}} \approx \frac{E\left[\xi_u^2(n)\right]}{E[e^2(n)]} \tag{3.63}$$

Unlike the disturbed error signal $e(n)$, the undisturbed error signal $\xi_u(n)$ is inaccessible and, therefore, not directly measurable. To overcome this computational difficulty, a final assumption can be introduced.

Assumption III It is mandatory to note that the input signal spectral content is basically flat over a frequency band larger than that engaged by each element of the weight error vector $\varepsilon(n)$, as a consequence of justifying the approximation:

$$E\left[\xi_u^2(n)\right] = E\left[\left|\varepsilon^T(n)u(n)\right|^2\right] = E\left[\left\|\varepsilon(n)\right\|^2\right]E\left[u^2(n)\right] = D(n)E\left[u^2(n)\right] \tag{3.64}$$

where $D(n)$ is the mean square deviation. Note that the approximate formula of the equation involves the input signal $u(n)$ rather than the tap input vector $u(n)$.

Assumption III is a statement of the low-pass filtering action of the LMS filter. Thus, using the foregoing equations, the approximation can be as

$$\widetilde{\mu}_{\text{opt}} \approx \frac{D(n)E[u^2(n)]}{E[e^2(n)]} \tag{3.65}$$

The practical virtue of the approximate formula of $\widetilde{\mu}_{\text{opt}}$ defined in this equation is borne out in the fact that simulations as well as real-time implementations have shown that it provides a good approximation for $\widetilde{\mu}_{\text{opt}}$ the case of large filter lengths and speech inputs.

3.4 Recursive Least Squares (RLS) Adaptive Filter

In the feedback mechanism implementation of the method of the least squares, one can start the computation with previously enumerated initial conditions and by using the information contained in new data samples to update the old estimates. Length of observed data is variable. Accordingly, it can be expressed by cost function and minimized as £(n) [1]. Thus, it can be written that as shown:

$$£(n) = \sum_{i=1}^{n} \beta(n,i)|e(i)|^2 \tag{3.66}$$

Here it can be observed that $e(i)$ is the difference. This difference is calculated between the necessary reaction $d(i)$ and the outcome $y(i)$, which generated by a transversal filter, and its tap weights are at time i equal $u(i)$, $u(i-1)$,... $u(i-M+1)$, as in Fig. 3.6 that is,

$$e(i) = d(i) - y(i) = d(i) - w^H(n)u(i) \tag{3.67}$$

where $u(i)$ is the tap input vector at time 0 defined by $u(i) = [u(i), u(i-1), \ldots, u(i-m)]^T$ and $w(n)$ is the tap weight vector at time n, defined by $w(n) = [w_0(n), w_1(n), \ldots, w_{M-1}(n)]^T$.

Consider that the tap weights of the transversal filter are basically fixed during the observation interval $1 \leq i \leq n$ for which the cost function £(n) is defined. The weighting factor $\beta(n,i)$ in this equation has the property that

$$0 < \beta(n,i) \leq 1, \quad i = 1, 2, \ldots, n \tag{3.68}$$

The use of the weighting factor $\beta(n,i)$, in general, is mainly required to verify that the data in the old past are omitted or forgotten to undergo the chance of the statistical variation of the observable data. This condition arises mainly when the filter works in an nonstationary environment. Generally, usage of a special form of

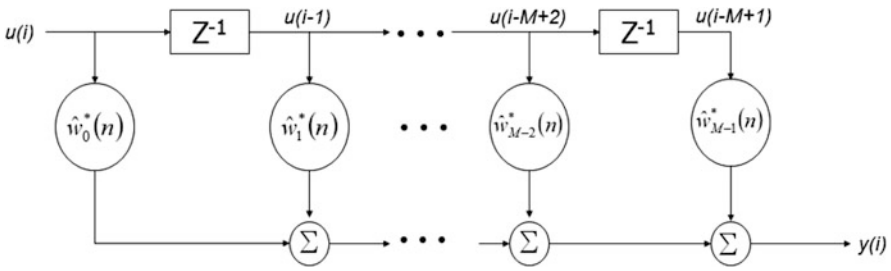

Fig. 3.6 Transversal filters with time-varying tap weights

weighting is the exponential weighting factor, or forgetting factor, which can be narrated by

$$\beta(n,i) = \lambda^{n-i}, \quad i = 1, 2, \ldots, n \tag{3.69}$$

Mainly, λ is a positive constant. Here, three cases are possible. When $\lambda = 1$, it is the usual mechanism of least squares. The inverse of $1 - \lambda$ is, roughly speaking, a memory of the algorithm and it is measured in that form of the algorithm [10]. The special case $\lambda = 1$ corresponds to infinite memory.

3.4.1 Regularization

Least square estimation input data are given, containing a tap input vector $u(n)$ and the according desired response $d(n)$ for changing [1]. The poorly posed nature of least squares estimation is caused by the following reasons:

- To renovate the input output mapping uniquely, the available data are in the form of insufficient information as input data.
- The occurrence of noise or imprecision in the input data that cannot be avoided adds uncertainty to the reconstructed input–output mapping.

For generating the estimation problem "well posed," some form of prior information about the input–output mapping is needed. This, in turn, means that the formulation of the cost function must be expanded to take the prior information into account. To satisfy that objective, the cost function can be minimized as the sum of two components:

$$£(n) = \sum_{i=1}^{n} \lambda^{n-i} \left| e(i)^2 \right| + \delta\lambda^n \|w(n)\|^2 \tag{3.70}$$

Here, the use of pre-windowing is assumed. The cost function can be defined as

$$\sum_{i=1}^{n} \lambda^{n-i} \left| e(i)^2 \right| = \sum_{i=1}^{n} \lambda^{n-i} \left| d(i) - w^H(n) \right| \tag{3.71}$$

Error is data that are not independent. Consideration of these data is exponentially based and on that weighted error between the desired responses $d(i)$ and the actual response of the filter, $y(i)$; because of this the tap weight vector can be correlated.

$$y(i) = w^H(n)u(i) \tag{3.72}$$

A regularizing term,

$$\delta\lambda^n \|w(n)\|^2 = \delta\lambda^n w^H(n)w(n) \tag{3.73}$$

where δ is a positive real number and it is known as a regularization parameter. Excluding the factor $\delta\lambda^n$, the regularizing term is based solely on the tap weight vector $w(n)$. The term is contained in the cost function to stabilize the solution.

In a strict sense, the term $\delta\lambda^n\|w(n)\|^2$ is a "rough" form of regularization for two reasons. First, the exponential weighting factor λ lies in the interval $0 < \lambda \leq 1$; hence, for λ less than unity, λ_2 tends to zero for large n, which means that the beneficial effect of adding $\delta\lambda^n\|\hat{w}(n)\|^2$ to the cost function is forgotten with time. Second, and more important, the regularizing term should be of the form $\delta\|DF(\hat{w})\|^2$, where $F(\hat{w})$ is the input–output map realized by the RLS filter and D is the differential operator.

3.4.2 Reformulation of the Normal Equations

Expanding the foregoing equation and collecting terms, it can be found that when in the cost function the impact of including the regularizing term, £(n) is equivalent to a reformulation of the M by M time average correlation matrix of the tap input vector:

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} u(i)u^H(i) + \delta\lambda^n I \tag{3.74}$$

where I can be defined as identity matrix with length M. The M by 1 time average cross-correlation vector $z(n)$ between the tap inputs of the transversal filter and the desired response is unaffected by the use of regularization [2].

$$z(n) = \sum_{i=1}^{n} \lambda^{n-i} u(i) * d(i) \tag{3.75}$$

where, again, the use of pre-windowing is assumed. The optimum value of the M by 1 tap weight vector, for which the cost function attains its minimum value, is as under per the method of least squares.

$$\Phi(n)\widehat{w}(n) = z(n) \tag{3.76}$$

3.4.3 Recursive Computations of $\Phi(n)$ and $z(n)$

Isolating the term corresponding to $i = n$ from the remaining of the accumulation and on the other side of the equality can be written as

$$\Phi(n) = \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i} u(i) u^H(i) + \delta \lambda^{n-1} I + u(n) u^H(n) \tag{3.77}$$

Hence, the following recursion for updating the value of the correlation matrix of the tap inputs may have [2]

$$\Phi(n) = \lambda \Phi(n-1) + u(n) u^H(n) \tag{3.78}$$

Here, $\Phi(n-1)$ is the "old" value of the correlation matrix, and the matrix product $u(n)\, u^H(n)$ has a role as a "correlation" term in the updating operation. Note that the recursion of the foregoing equation holds, irrespective of the initializing condition.

Similarly, this equation may be used to derive the following recursion for updating the cross-correlation vector between the tap inputs and the desired response:

$$z(n) = \lambda z(n-1) + u(n) d^*(n) \tag{3.79}$$

It is necessary to determine the inverse of the correlation matrix $\Phi(n)$ to compute the least squares estimate for tap weight vector. In practice, however, usually performing such an operation can be avoided, as it can be quite time consuming. Also, it is preferable to compute the least squares estimate $\hat{w}(n)$ for the tap weight vector recursively for $n = 1, 2 \ldots \infty$. Both these objectives can be realized by using a basic result in matrix algebra known as the matrix inversion lemma.

3.4.4 The Matrix Inversion Lemma

Let A and B be two positive-definite M–by-M matrices related by

$$A = B^{-1} + CD^{-1}C^H \tag{3.80}$$

where D is a positive-definite N-by-M matrix and C is an M-by-N matrix. According to the matrix inversion lemma, the inverse of the matrix A as may be expressed as

$$A^{-1} = B - BC\left(D + C^H BC\right)^{-1} C^H B \tag{3.81}$$

The proof of this lemma is established by multiplying these two equations and recognizing that the product of a square matrix and its inverse is equal to the identity matrix. The matrix inversion lemma states that if a matrix A is given, as defined in these equations, its inverse A^{-1} can be determined by using the relationship expressed in the foregoing equation. In effect, the lemma is described by that pair of equations [3].

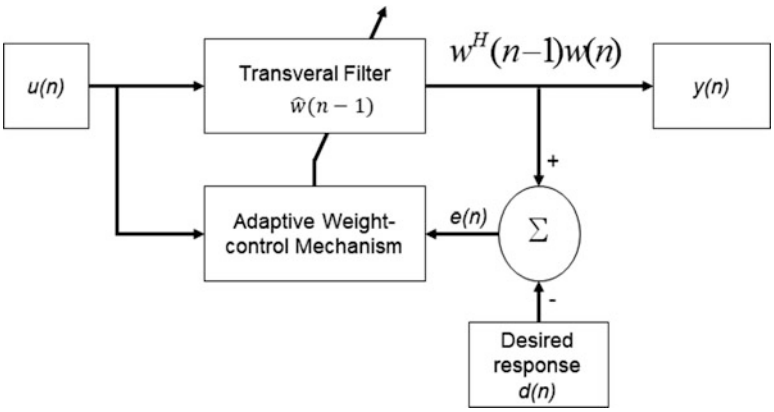

Fig. 3.7 RLS algorithm concept

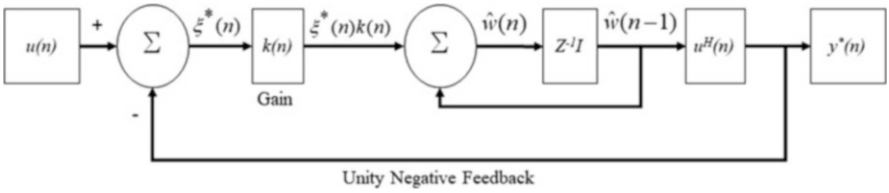

Fig. 3.8 RLS algorithm signal flow graph

It is very important to note that the following equation describes the operation of the algorithm, whereby priori estimation error would be computed in the transversal filter. The further step shows the adaptive operation of the algorithm, in which the tap weight vector is changed by incrementing its previous value by an amount equal to the product of the complex conjugate of the priori estimation error $\xi(n)$ and the time-varying gain vector $k(n)$; hence the name "gain vector." The step cited in the next line helps to update the value of the gain vector itself. The most important characteristics of the RLS algorithm is that the inversion of the correlation matrix $\Phi(n)$ is replaced at each step by a simple scalar division.

The RLS algorithm concept and signal flow graph are shown in detail in Figs. 3.7 and 3.8.

- Initially, an intermediate quality that can be shown with $\Pi(n)$ is computed.
- In the next step, $\Pi(n)$ is utilized to compute $k(n)$.

This two-stage computation of $k(n)$ is favored over the direct calculation of $k(n)$ using the foregoing equation from a finite precision arithmetic point of view.

To initialize the RLS filter:

- The initial weight vector $\hat{w}(0)$, for which it is necessary to set $\hat{w}(0) = 0$.
- The initial correlation matrix $\Phi(0)$. Setting $n = 0$ in the above equation, with the use of pre-windowing, the following can be obtained:

$$\Phi(0) = \delta I \tag{3.82}$$

where δ is the regularization. The parameter δ should be assigned a small value for a high signal-to-noise ratio (SNR) and a large value for a low SNR, which may be justified on regularization grounds [9].

3.4.5 Selection of the Regularization Parameter

The convergence behavior of the RLS algorithm was evaluated for a stationary environment, with particular reference to two variable parameters:

- The signal-to-noise ratio (SNR) of the tap input date, which is determined by the prevalent operation conditions.
- The regularization parameter δ, which is under the designer's control.

Let $F(x)$ denote a matrix function of x, and let $f(x)$ denote a nonnegative scalar function of x, where the variable x assumes values in some set x. The following definitions might be introduced where there exist constants $c1$ and $c2$ that are independent of the variable x, such that

$$F(x) = \theta(f) \tag{3.83}$$

where $\|F(x)\|$ is the matrix norm of $F(x)$, which is itself defined by

$$c1f(x) \leq \|F(x)\| \leq c2f(x), \quad \text{forall}\, x \in x \tag{3.84}$$

The significance of the definition introduced in this derivation will become apparent presently.

$$\|F(x)\| = (\text{tr}[FH(x)F(x)])1/2 \tag{3.85}$$

The initialization of the RLS filter includes setting the initial value of the time average correlation matrix

$$\Phi(0) = \delta I \tag{3.86}$$

The dependence of the regularization parameter δ on SNR is given in detail. In particular, $\Phi(0)$ is reformulated as

$$\Phi(0) = \mu\alpha R0 \tag{3.87}$$

where $\mu = 1 - \lambda$ and $R0$ is a deterministic positive definite matrix defined by $R0 = \sigma^2$ μI in which $\sigma^2 u$ is the variance of a, data sample $u(n)$.

Thus, according to this equation, the regularization parameter δ is defined by

$$\delta = \sigma^2 u \mu \alpha \tag{3.88}$$

The parameter α provides the mathematical basis for distinguishing the initial value of the correlation matrix $\Phi(n)$ as small, medium, or large. It may distinguish three scenarios in light of the definition introduced [10].

- $\alpha > 0$, which corresponds to a small initial value $\Phi(0)$.
- $0 > \alpha \geq -1$, which corresponds to a medium initial value $\Phi(0)$.
- $-1 \geq \alpha$, which corresponds to a large value $\Phi(0)$.

With these definitions and the three distinct initial conditions at hand, the selection of the regularization parameter δ in initializing the RLS algorithm for situations can be summarized [10].

- *High SNR*: When the noise level in tap inputs is low, that is, the input SNR is of the order of 30 dB, the RLS algorithm exhibits an exceptionally fast rate of convergence, provided that the correlation matrix is initialized with a small enough norm. Typically, this requirement is satisfied by setting $\alpha = 1$. As α is reduced toward zero, the convergence behavior of the RLS algorithm deteriorates.
- *Medium SNR*: In a medium the SNR environment, that is, the input SNR is of the order of 10 dB, the rate of convergence of the RLS algorithm is worse than the optimal rate for the high SNR case, but the convergence behavior of the RLS algorithm is essentially insensitive to variations in the matrix norm of $\Phi(0)$ for $-1 \leq \alpha < 0$.
- *Low SNR*: Finally, when the noise level in the tap inputs is high, that is, the input SNR is of the order of -10 dB or less, it is preferable to initialize the RLS algorithm with a correlation matrix $\Phi(0)$ with a large matrix norm (i.e., $\alpha \leq -1$), because this condition may yield the best overall performance.

These remarks hold for a stationary environment or a slowly time-varying one. If, however, there is an abrupt change in the state of the environment, the change occurs as renewed initialization with a "large" initial $\Phi(0)$ wherein $n = 0$ corresponds to the instant at which the environment switched to a new state. In such a situation, the recommended practice is to stop the operation of the RLS filter and restart a new procedure by initializing with a small $\Phi(0)$.

3.4.6 Convergence Analysis of RLS Algorithm

The convergence behavior of the RLS algorithm in a stationary environment is assuming that the exponential weighting factor λ is unity. To pave the way for the

discussion, three assumptions can be made, all of which are reasonable in their own way [1].

Assumption I The desired response $d(n)$ and the tap input vector $u(n)$ are related by the multiple linear regression model

$$d(n) = w_0^H u(n) + e_0(n) \tag{3.89}$$

where w_0 can be known as the regression vector and $e_0(n)$ is the noise measurement vector. The noise $e_0(n)$ is white with zero mean and variance σ_0^2, which makes it independent of the repressor $u(n)$.

Assumption II The input signal $u(n)$ is drawn from a stochastic process, which is ergodic in the autocorrelation function. The implication of Assumption II is that time averages for ensemble averages may be substituted. In particular, the ensemble average correlation matrix of the input vector $u(n)$ may be expressed as

$$R \approx 1/n\Phi(n), \quad n > M \tag{3.90}$$

where $\Phi(n)$ is the time average correlation matrix of $u(n)$ and the requirement $n > M$ ensures that the input signal spreads across all the taps of the transversal filter. The approximation of this equation improves with an increasing number of time steps n.

Assumption III The functions in the weight error vector $\varepsilon(n)$ are slow compared with those of the input signal vector $u(n)$. The justification for Assumption III is to recognize that the weight error vector $\varepsilon(n)$ is the accumulation of a series of changes extending over n iterations of the RLS algorithm. This property is shown by

$$e(n) = w_0(n) - \widehat{w}(n) = \varepsilon_0 - \sum_{i=1}^{n-1} k(i)\xi^*(i) \tag{3.91}$$

In the algorithm both $k(i)$ and $\xi(i)$ depend on $u(i)$; the summation in the equation has a "smoothing" effect on $\varepsilon(n)$. In effect, the RLS filter acts as a time-varying low-pass filter. No further assumption on the statistical characterization of $u(n)$ and $d(n)$ are made in what follows.

3.4.7 Convergence of the RLS Algorithm in the Mean Value

Solving the normal equation for $\hat{w}(n)$, it can be written as

$$\widehat{w}(n) = \Phi^{-1}(n)z(n), \quad n > M \tag{3.92}$$

where, for $\lambda = 1$, $\Phi(n) = \sum_{i=0}^{n} u(i)u^H(i) + \Phi(0)$ and $z(n) = \sum_{i=0}^{n} u(i)d^*(i)$.

Finally, after simplification,

$$z(n) = u(i)u^H(i)w_0 + \sum_{i=0}^{n} u(i)e^0(i) = \Phi(n)\widehat{w}_0 + \sum_{i=0}^{n} u(i)e^0(i) \tag{3.93}$$

$$\widehat{w}(n) = w_0 - \Phi^{-1}(n)\Phi(n)w_0 + \Phi^{-1}(n)\sum_{i=0}^{n} u(i)e^0(i) \tag{3.94}$$

Taking the expectation of both sides of the above equation and invoking Assumptions I and II, it can be written

$$E[\widehat{w}(n)] \approx w_0 - 1/nR^{-1}w_0 = w_0 - \delta/nR^{-1}w_0 = w_0 - \delta/nP, \quad n > M \tag{3.95}$$

where P is the ensemble average cross-correlation vector between the desired responses $d(n)$ and input vector $u(n)$ [2]. The equations state that the RLS algorithm is convergent in the mean value. For finite n greater than the filter length M, the estimate $\hat{w}(n)$ is biased, because of the initialization of the algorithm by setting $\Phi(0) = \delta I$, but the bias decreases to zero as n approaches infinity.

3.4.8 Mean Square Deviation of the RLS Algorithm

The weight error correlation matrix is defined by

$$k(n) = E\left[\varepsilon(n)\varepsilon^H(n)\right] = E\left[(w_0 - \widehat{w}(n))(w_0 - \widehat{w}(n))^H\right] \tag{3.96}$$

The following two observations are important for $n > M$:

1. The mean square deviation $D(n)$ is magnified by the inverse of the smallest eigenvalue $\lambda_{\min}$. Hence, to a first order of approximation, the sensitivity of the RLS algorithm to eigenvalue spread is determined initially in proportion to the inverse of the smallest eigenvalues. Therefore, poorly conditioned least squares problems may lead to poor convergence properties.
2. The mean square deviation $D(n)$ decays almost linearly with the number of iterations, n. Hence, the estimate $\hat{w}(n)$ produced by the RLS algorithm converges in the norm (i.e., mean square) to the parameter vector w_0 of the multiple linear regression model almost linearly with time.

3.4.9 Ensemble Average Learning Curve of the RLS Algorithm

In the RLS algorithm, there are two types of error, the a priori estimation error $\xi(n)$, and the a posteriori estimation error $e(n)$. It can be found that the mean square values of these two errors vary differently with time n. At time $n = 1$, the mean square value of $\xi(n)$ becomes large equal to the mean square value of the desired response $d(n)$ and then decays with increasing n. The mean square value of $e(n)$, on the other hand, becomes small at $n = 1$ and then rises with increasing n, until a third point is reached for large n for which $e(n)$ is equal to $\xi(n)$ [3]. Accordingly, the choice of $\xi(n)$ as the error of interest yields a learning curve for the RLS algorithm that has the same general shape as that for the LMS algorithm. By choosing $\xi(n)$ thus, a direct graphical comparison can be made between the learning curves of the RLS and LMS algorithms as a computation of the ensemble average learning curve of the RLS algorithm on the a priori estimation error $\xi(n)$. The convergence analysis of the RLS algorithm presented here assumes that the exponential weighting factor equals unity. The following can be concluded from the observations:

1. The ensemble average learning curve of the RLS algorithm converges in about $2M$ iterations, where M is the filter length; this means that the rate of convergence of the RLS algorithm is typically an order of magnitude faster than that of the LMS algorithm.
2. As the number of iterations, n, approaches infinity, the MSE $J'(n)$ approaches a final value equal to the variance σ_0^2 of the measurement error $e_0(n)$. In other words, the RLS algorithm produces zero excess MSE or zero misadjustment.
3. Convergence of the RLS algorithm in the mean square is independent of the eigenvalues of the ensemble average correlation matrix R of the input vector $u(n)$.

3.5 Noise

Noise is considered as an undesired frequency in any type of signal. An unwanted frequency signal is present in various degrees around entire environments. The various types of noises can affect the quality of conversation. Generally signals that are propagating through a wireless medium are affected by different types of noises such as thermal, acoustic background, and electromagnetic radiofrequency noise. In most communication systems, noise must be considered carefully, because during transmission noise affects signals and generates several amount of errors in the system. Reduction of noise and signal processing succeeds based on its capacity to know the feature and model the noise process in a suitable random process and using that feature to subtract noises from the original incoming signal [11]. By considering the source of the generation of noise, noise can be categorized in the following different categories.

- *Acoustic noise*: This kind of noise is generated from various sounds produced from the motion, collision, and vibrations of objects. Acoustic noise is also generated by such as a car's motion and electronic equipment such as air conditioners and computers. Also, it is generated by different undirected and unguided surrounding activities in nature.
- *Thermal noise and shot noise*: Thermal noise is generated by heat only. All the equipment and semiconductor devices are operated at room temperature. Thermal noise results from uneven motion of energized particles in the electrical conductor because of temperature. Thermal noise is present in all conducting medium and is produced without any application of electromotive force. In contrast, because of heavy undirected motion of electrical current in the conductor, shot noise is there.
- *Electromagnetic noise*: This noise can be generated at all frequencies in the band; wherever long-distance frequency travel and communication take place, this noise is present. All electronics equipment works on radiofrequency.
- *Electrostatic noise***:** The most important sources of electrostatic noise are fluorescent lighting. It is not important whether there is a flow of current.
- *Channel distortions, echo, and fading*: Whenever there is relative motion between transmitter and receiver, it causes fluctuation in the received signal and also causes fading of the signal. Because of the generated fading, signal strength becomes weak and because of that the quality of the produced signal is degraded.
- *Processing noise*: Noise that results from internal D/A processing of a signal such as the phenomenon of quantization in which quantization errors are generated, and also because of erroneous channels, data packets are lost in the system. Speech signals are mainly divided into two parts, vowels and consonants. The low-frequency characteristics are taken by the vowel sound but energy will be much greater with the consonant sound with high frequency, and energy participation is very low in magnitude. Most of the information in the speech signal is given by consonants. Human ear defects mainly result in less identification of high-frequency in contrast to low-frequency sound. Noise introduces a particular disadvantage in identifying sounds at low frequencies. In the presence of hearing aids it is also difficult to hear speech. Speech intelligibility is reduced in the presence of noise [12].

Noise is from a different frequency. By analyzing its frequency spectrum, the noise signal can be classified as follows:

1. *White noise*: It is indeterministic and cannot be predicted in a natural way, and that is with a flat power spectrum. In this noise all the frequencies are present with equal strength.
2. *Band-limited noise*: It is with limited bandwidth spectrum; light in the component means less dense in nature, which covers a very small part of the spectrum, and it is possible to modify necessary information.
3. *Narrowband noise*: It is generated from the electricity supply, which is at 60 Hz, and again it is also random in nature. In this kind of noise, pauses are very sharp.
4. *Colored noise*: This noise has uneven frequency distribution. Examples of this type of noises are pink noise and brown noise.

5. *Impulsive noise*: Impulsive noise is a very spontaneous type of noise and generates a signal of very short duration. Occurrence of impulses in the system is very random in nature.
6. *Transient noise*: A transient noise and impulse noise are very similar kinds of noises: the main difference is that a transient noise pulse is broad in nature.

3.5.1 Sources of Noise

The most important dominant noises that affect speech signal widely are classified as follows.

3.5.1.1 Different Road Traffic Noises

Most deaf people are expected to interact with people in their routine life. At the time of some of the conversations, it is possible that the person may be out of his home. Generally road traffic and vehicles are the most important type of noise in social life. All vehicles generate heavy interference in the hearing aids with processing of the speech signal. Generally all these vehicles travel by means of specific types of engine. The engine will generate noise in the form of spark plugs and motors and the exhaust systems when burning the fuel of large heavy vehicles. With this generation, the structure of the city amplifies these noises in narrow streets and very high multistory buildings, forming an environment where signals are generated and reflected back in the form of echoes. The nature of truck starting noises and traffic jam noise are shown in Figs. 3.9 and 3.10, respectively. Moreover, in the metro cities, nowadays this problem is more serious. In most of the big cities, a huge amount of traffic is also generated by airports. The cockpit and engine noises of low-flying aircraft, charter planes, helicopters, and air force planes have added very cumbersome noise levels to the system of speech enhancement. In addition, locomotive engines, horns, and whistles and switching and shunting operations in railyards can deeply impact noise increments.

3.5.1.2 Babble Noise

Speech communication is very important, and a major part of interaction among human beings goes on through this means. It is actually very challenging to recognize speech signals in the presence of background babble noise for both normal listeners and hearing-impaired persons. The nature of this babble noise is shown in Fig. 3.11. Interfering noise causes impacts on human speech and the hearing-impaired person. Interfering noise decreases speech intelligibility and quality. Clean speech signals are corrupted by background noise and multi-talker babble. In practical life, all conversation is carried out in the presence of other people in the

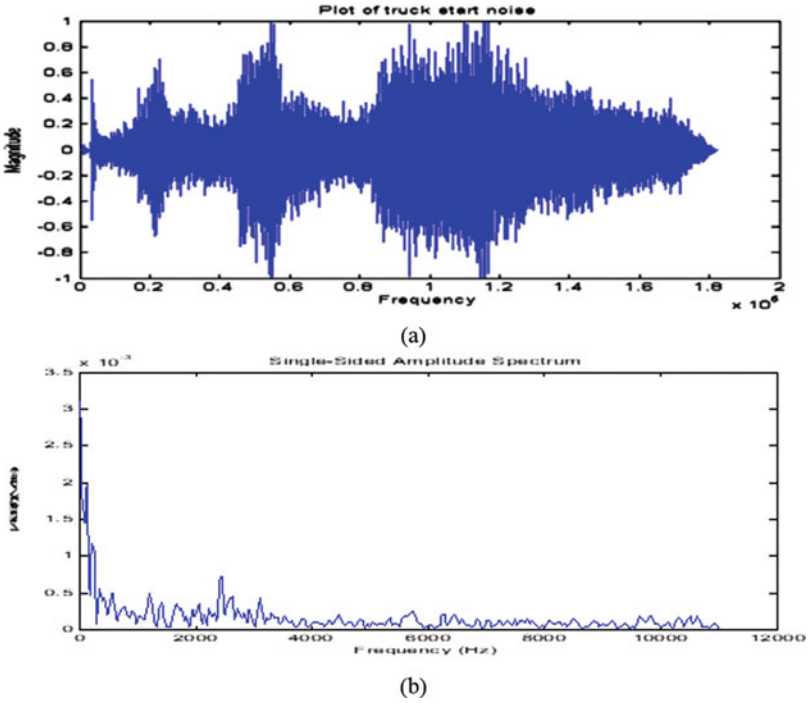

Fig. 3.9 (**a**) Truck starting noise signal. (**b**) Single-sided amplitude spectrum of signal

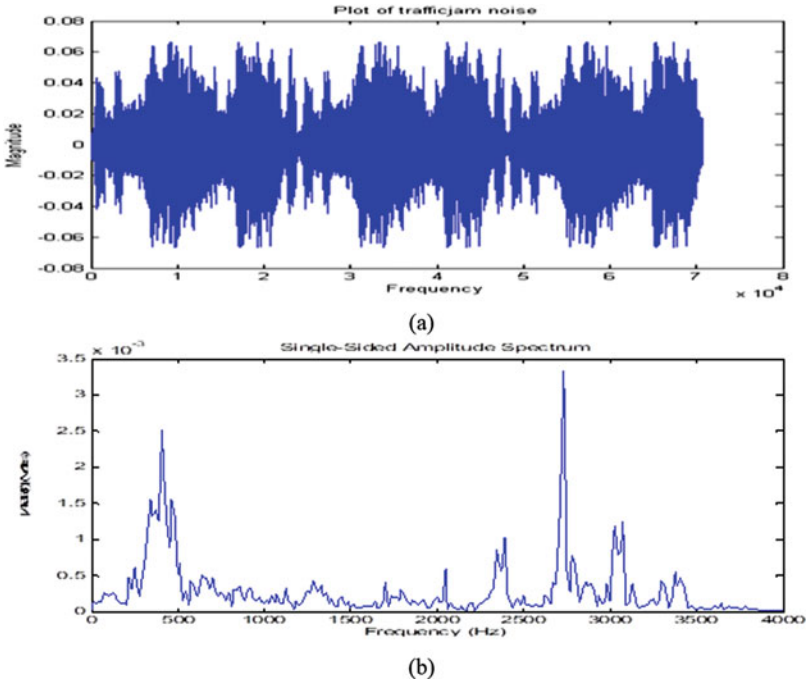

Fig. 3.10 (**a**) Traffic jam noise signal. (**b**) Single-sided amplitude spectrum of signal

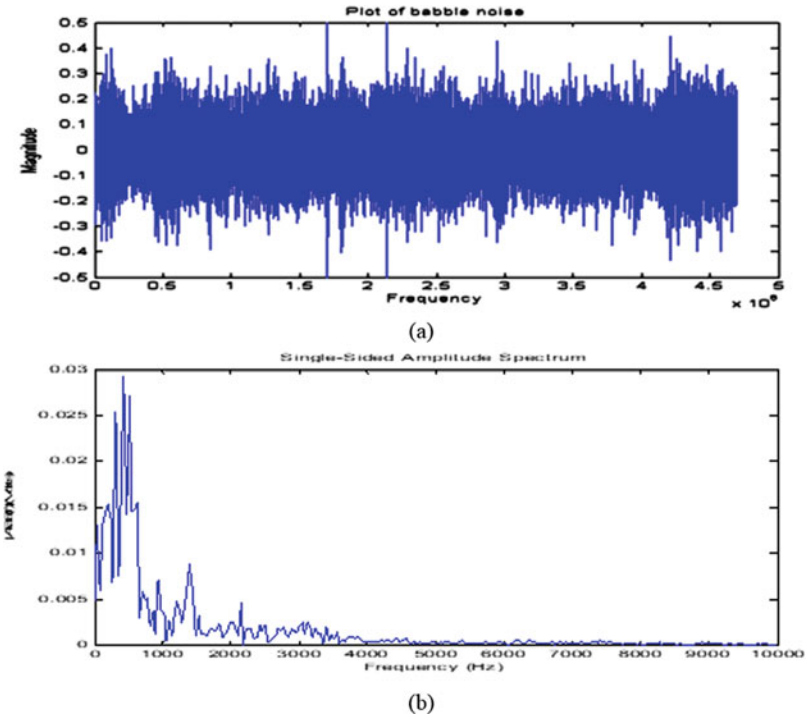

Fig. 3.11 (**a**) Babble noise signal. (**b**) Single-sided amplitude spectrum of signal

community. Persons talking face to face are always surrounded by too many other people. Always, background noise is present, because all of them are talking continuously. It is considered as multi-talker babble noise. Babble noise creates serious problems in recognizing speech for deaf persons because that noise is speech-like noise. Babble noise overlaps on the speech of conversation and reduces intelligibility.

References

1. Haykin, S. S. (2008). Adaptive filter theory. Chennai: Pearson Education India.
2. Honig, M. L., & Messerschmitt, D. G. (1984). *Adaptive filters: Structures, algorithms and applications*. Boston: Kluwer Academic.
3. Diniz, P. S. R. (2008). *Adaptive filtering: Algorithms and practical implementation*. New York: Springer.
4. El-Fattah, M. A., Dessouky, M. I., Diab, S. M., & El-Samie, F. A. (2008). Adaptive wiener filtering approach for speech enhancement. *Journal of Ubiquitous Computing and Communication, 2*(3), 23–31.

5. Abdulmagid, M. A., Krusienski, D. J., Pal, S., & Jenkins, W. K. (2004). *Principles of adaptive noise canceling*. University Park: Department of Electrical & Computer Engineering, Pennsylvania State University.
6. Slock, D. T. (1993). On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal Processing, 41*(9), 2811–2825.
7. Greenberg, J. E. (1998). Modified LMS algorithms for speech processing with an adaptive noise canceller. *IEEE Transactions on Speech and Audio Processing, 6*(4), 338–351.
8. Rahman, M. Z. U., Mohedden, S. K., Rao, B. R. M., Reddy, Y. J., & Karthik, G. V. S. (2011). Filtering non-stationary noise in speech signals using computationally efficient unbiased and normalized algorithm. *International Journal on Computer Science and Engineering, 3*(3), 1106–1113.
9. Kuo, S. M., & Morgan, D. R. (1999). Active noise control: a tutorial review. *Proceedings of the IEEE, 87*(6), 943–973.
10. Vijaykumar, V. R., Vanathi, P. T., & Kanagasapabathy, P. (2007). Modified adaptive filtering algorithm for noise cancellation in speech signals. *Elektronika ir elektrotechnika, 74*(2), 17–20.
11. Vaseghi, S. V. (2008). *Advanced digital signal processing and noise reduction*. New York: Wiley.
12. Mahanty, P., Firdous, B., & Swarnkar, R. (2007). Real time background noise cancellation in end user device. In *Portable Information Devices, 2007. PORTABLE 07. IEEE International Conference* (pp. 1–4). IEEE.

Chapter 4
Fourier Transform, Short-Time Fourier Transform, and Wavelet Transform

4.1 Fourier Transform (FT)

The Fourier transform (FT) transforms a time domain speech signal into its corresponding frequency domain. The FT of a speech signal can be calculated by using Equation 4.1 [1, 2]. The FT gives complex value coefficients of the speech signal.

$$S(k) = \sum_{n=0}^{N-1} S(n) \cdot e^{-2j\pi nk}, \quad k = 0, 1, \ldots, N-1 \tag{4.1}$$

where $S(n)$ is the input speech signal in the time domain and $S(k)$ is the transformed speech signal in the frequency domain.

According to the literature [3], the FT has some limitations when it is applied to a speech signal:

- FT cannot provide simultaneous time and frequency localization.
- FT is not very useful for analyzing time-variant, nonstationary signals.
- FT is not appropriate for representing discontinuities in the signals.

4.2 Short-Time FT

The short-time FT (STFT) segments the speech signal into narrow time intervals and takes the FT of each segment. Then, each FT provides the spectral information of a segmented speech signal, providing simultaneous time and frequency information [3]. The steps for applying STFT on speech signal are given as follows:

- Choose a window function of finite length.
- Place the window on top of the signal at $t = 0$.

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_4

- Truncate the signal using this window.
- Compute the FT of the truncated signal; save results.
- Incrementally slide the window to the right.
- Go to step 3 and repeat the process until the window reaches the end of the signal.

The STFT of the speech signal can be calculated by using Equation 4.2 [3].

$$\mathrm{STFT}_f^u(t', u) = \int_t [f(t) \cdot W(t - t')] \cdot e^{-j2\pi ut} dt \tag{4.2}$$

where t is a time parameter of signal, u is a frequency parameter of the signal, $f(t)$ is an input signal, and W is a windowing function.

In STFT, windowing function has an important role. Window function should be narrow enough so that the signal portion can fall within the stationary window. But, narrow window function does not give a good localization of the signal in the frequency domain. If window function is infinitely long then STFT turns into FT and provides good frequency localization but does not provide time localization. If window function is infinitely short then STFT provides good time localization but not frequency localization. Thus, this is one of the limitations of STFT when it is applied on a speech signal. This limitation of STFT is overcome by wavelet transform.

4.3 Wavelet Transform (WT)

Analysis of wavelet transform uses small waves and functions recognized as wavelets. We can describe wavelet more accurately as a local wavelike function. Some of the most commonly used examples of wavelets are shown in Fig. 4.1. Any signal can be transformed from one representation to the other representation wherein we can find more useful information using wavelets. The process is known as wavelet transform. If we represent the wavelet transform mathematically, it can be described as the convolution between the wavelet function and the signal under observations.

Fig. 4.1 Some basic wavelets

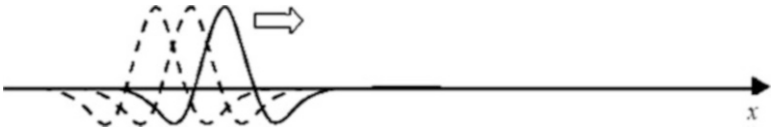

Fig. 4.2 Location of wavelet

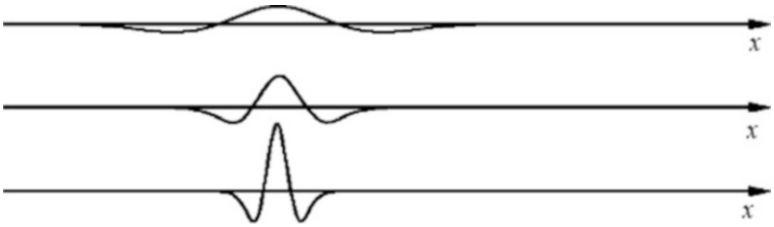

Fig. 4.3 Scale of wavelet

Different wavelets can be interpreted in alternate ways: a wavelet can be stimulated to different places on the exposed signal and it can be expanded or compressed per requirements. Wavelet transform measures the matching of the signal on a local basis with the wavelets. The schematic of the same is represented in Fig. 4.2. As shown in Fig. 4.3, whenever the shape of the signal is matched with the wavelet at a specific scale and location, one can observe a large value of the transform.

On the other hand, if the matching value is small, a low value of the transform is obtained. Then, as shown by a black dot in Fig. 4.4, the transform value is located on the plane of the two-dimensional transform. The task of computing the transform for various scales of the wavelets and at various places of the signals is done by what is known as continuous wavelet transform (CWT) for continuous signals or in discrete steps for the discrete wavelet transform [4].

Wavelet transform, which translates a signal into another form suitable to easily analyze certain parameters of the original signal, allows a picture to find the correlation between the signal and wavelet at different scales and locations. For computing a wavelet transform, a small wavelike function is required, which as the name says, is a limited to a small area waveform. A wavelet is a function that satisfies some mathematical criteria. These functions are manipulated through a process of translation and dilation to change the signal into a different form that 'unfolds' the wave in actual timing and scaling. The signal to be analyzed is a temporal signal like some small mathematical function that varies with time, such as the velocity of a fluid, engine-caused vibration data, or an ECG signal. Presently, the independent variable is space not time; still, analysis is done in the same way. That form of a minor wave can be seen on Fig. 4.5, and it is specifically localized on the time axis. There are huge types of wavelets to select from those availability for data analysis. The better solution for the specified applications generally depends on the nature of the signal and what is required from the analysis [4].

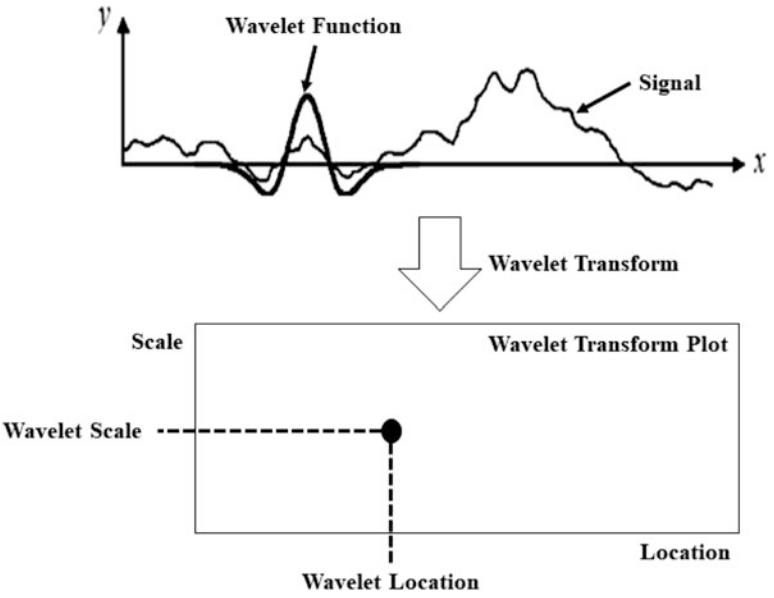

Fig. 4.4 Wavelet function, speech signal, and transform

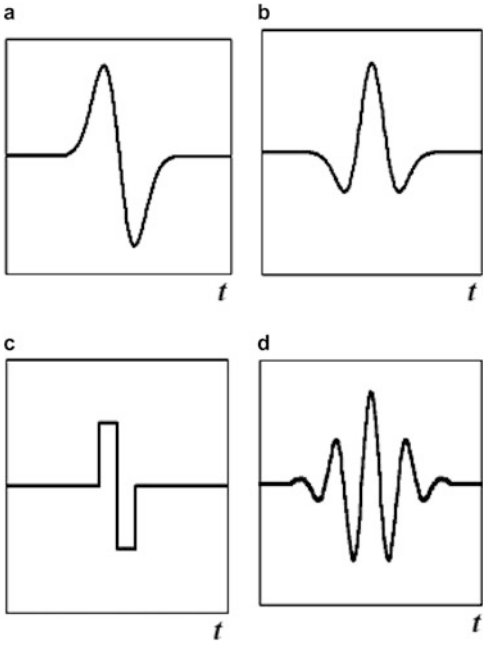

Fig. 4.5 Four wavelets (**a**) Gaussian wave (**b**) Mexican Hat (**c**) Haar (**d**) Morlet

The Mexican hat wavelet is one of the important wavelet functions. It covers many properties of continuous wavelet transform (CWT) analysis. The definition of a Mexican hat wavelet is

$$\Psi(t) = \left(1 - t^2\right) e^{-\frac{t^2}{2}} \tag{4.3}$$

The wavelet function represented by the foregoing iteration is recognized as the mother wavelet. The basic requirements of any function as a wavelet function are given as

1. In general, wavelet transform should include finite energy for different applications:

$$E = \int_{-\infty}^{\infty} |\Psi(t)|^2 dt < \infty \tag{4.4}$$

In the mentioned iteration, energy is defined as an integral of the squared magnitude of $\Psi(t)$ over the infinite duration of time. For complex $\Psi(t)$, one need to find the energy considering both magnitude as well as phasor part of $\Psi(t)$ [5].

2. If $\widehat{\Psi}(f) = \int_{-\infty}^{\infty} |\Psi(t)| e^{-i(2\pi f)t} dt$ is the Fourier transform of $\Psi(t)$, then the following condition must hold:

$$C_g = \int_{-\infty}^{\infty} \frac{\left|\widehat{\Psi}(f)\right|^2}{f} df < \infty \tag{4.5}$$

The foregoing iteration shows that the wavelet transform is basically not with $\Psi(0) = 0$, that is, the zero frequency component, or, we can also say that the wavelet $\Psi(t)$ should include a zero mean. This iteration can be recognized with an acceptability situation and C_g, which is known as the acceptability constant factor. By means of the selected wavelet, the value of C_g can be decided.

4.4 Comparison of the Wavelet Transform (WT) with FT and STFT

Wavelet analysis is actually used to compare several magnifications of signals with distinct resolution. The Fourier analysis can be done using basic building blocks, also known as time–frequency atoms, namely, sine and cosine waves. There are two different kinds of wavelets: namely, mother wavelet and child wavelets. The mother

wavelet oscillates and is translated and dilated by some translations and dilations so as to generate child wavelets. These two are used as building blocks of the wavelet analysis.

The Fourier series is useful only with periodic signals. The Fourier transform can be used for frequency analysis of nonperiodic general functions. The Fourier transform is used for the analysis of the time domain signal in the frequency domain. Transform is carried out in three steps. First, the signal is transformed from one domain to another domain, that is, time domain to the frequency domain. In this process, the coefficients of the frequency domain are modified with reference to the requirement, and last, the effect of the modification can be viewed in the time domain by applying the inverse transform, which converts the frequency domain signal back into the time domain. Here the Fourier coefficients would represent the contributions of cosine and sine functions having different frequencies. FT of the signal $f(t)$ can be given by

$$F(\omega) = \int F(\omega) e^{-i\omega t} dt \tag{4.6}$$

The inverse Fourier transform performs a reverse action in which it converts data from the frequency domain to the time domain. Inverse FT is represented by

$$f(t) = 1/2\pi \int F(\omega) e^{-i\omega t} dt \tag{4.7}$$

The analysis coefficients $F(\omega)$, which define the notation of global frequency f in the signal, are calculated as multiplicative products of the signal with sine wave fundamental function of unbounded duration. Hence, the foregoing Fourier transform works well with a signal having few stationary components. Unpredictable changes with reference to time in nonstationary signals, such as $f(t)$, expand out over the entire frequency in $F(\omega)$. To overcome this problem, the short-time Fourier transform (STFT) can be implemented [6].

Frequency dependency on time can be obtained by the value of instantaneous frequency. If the signal is broadband, the value of the instantaneous frequency just performs the averaging of different values of spectral components in time. A two-dimensional time–frequency representation is required to define and observe the dependency of spectral characteristics in time.

Now consider a stationary signal $f(t)$ through window function $g(t)$ of limited time, which is specifically centered at location τ. Now, in that case, the STFT of the signal can be defined as

$$\mathrm{STFT}(\tau, f) = \int_{-\infty}^{\infty} f(t) g^*(t - \tau) e^{-i\omega t} dt \tag{4.8}$$

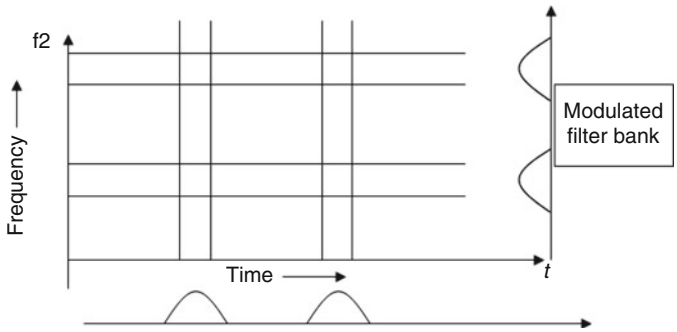

Fig. 4.6 Time–frequency plane for STFT

The function of STFT is to map the signal into a two-dimensional function to be considered in a time–frequency plane. his performance analysis critically depends on window function $g(t)$. The diagram of Fig. 4.6 shows a time–frequency plane corresponding to STFT. Thr vertical stripe shows windowing in the time domain. The frequencies of STFT can be computed around the window at time t. Here another view is given that is based on filter bank understanding of the similar procedure. The capacity of STFT to differentiate two pure sinusoids is better. Here, the windowing function, that is, $g(t)$, is given: FT, $G(\omega)$. Bandwidth Δf of the filter can be recognized as

$$\Delta f^2 = \left[\int f^2 |G(\omega)|^2 df\right] / \left[\int |G(\omega)|^2 df\right] \tag{4.9}$$

where the denominator $\int |G(\omega)|^2 df$ represents energy of the signal $g(t)$. If the separation between two signals is more than Δf, which is known as frequency resolution of the analytical process of STFT, then and only then can they be differentiated.

Likewise, the spread time is represented by Δt where the denominator $\int |g(t)|^2 dt$ shows the energy of $g(t)$. If the separation between the two signals is more than Δt, which is known as time resolution of the synthesis process of STFT, then and only then can they be differentiated. Now, because the product value between the resolution in time and frequency domain is lower, bounded by the Heisenberg uncertainty principle as given below, the resolution cannot be arbitrarily small.

$$\text{Time-bandwidth product} = \Delta t \Delta f > 1/4\pi \tag{4.10}$$

Thus, the possibilities are trading of frequency resolution for time resolution or vice versa. As Gaussian windows are normally met above bounds with equality, these are most widely used. In case of STFT, if any time a window has been selected then the time–frequency resolution can be kept fixed over the whole time–frequency plane because a similar window is utilized for all frequency variations.

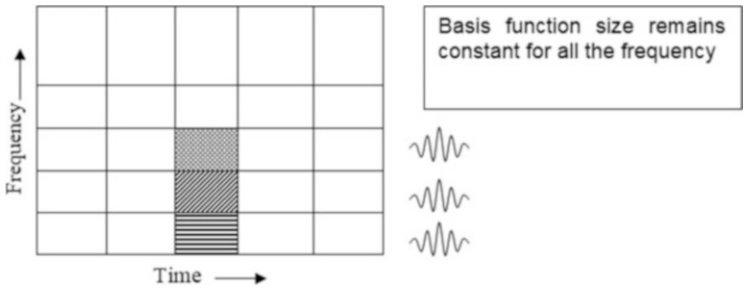

Fig. 4.7 Time–frequency plane for Fourier basis functions

For a nonperiodic signal $f(t)$, the signal cannot be accurately represented by addition of cosine and sine-like periodic functions. The only solution is to artificially extend the signal to make it periodic, but even that requires additional endpoint continuity. Windowed Fourier transform (WFT) is another solution that gives time and frequency domain information simultaneously. For this particular purpose, the signal is first divided into small parts that are individually giving WFT for separate analysis of frequency signals. For short transitions in the signal, windowing can be applied onto the signal in such a manner as to converge sections to zero at the endpoint; this is accomplished by a weight function with more emphasis in the middle than at the endpoint. The time domain signal is localized through the effect of the window. The DFT and DWT are linear transforms that are responsible for generation of a data structure containing segments with various lengths. The mathematical properties of matrices involved are similar. Both DFT and DWT in different domains are looked on as rotation functions. This new domain for FFT would contain cosine and sine basis functions and the mother wavelet for the wavelet transform. The key difference between both these transforms, that is, Fourier transforms and wavelet transform, is that in FT the sine and cosine functions are not localized in the space, whereas in WT, each and every individual wavelet function is localized in space (Fig. 4.7).

Because its sparseness features, the wavelet can be used for applications such as data compression, noise removing, and image processing. The time–frequency resolution is another major difference between FT and WT. Figure 4.8 shows the performance analysis of short-time Fourier transform and wavelet transform for time–frequency resolution.

The STFT is not utilized to analyze real-time signals, which have low-frequency signals along with high-frequency content and the frequency variations with respect to time. To overcome the time and frequency resolution limit of STFT, the wavelet analysis can be used to allow varying the resolution of Δt and Δf (Fig. 4.8) for the time–frequency plane. Thus, multi-resolution analysis can be achieved with wavelet transform. When looking at wavelet analysis from the aspect of the filter bank, it can be said that one has to vary time resolution with respect to variation in the central frequency. In wavelet manipulation the frequency span, Δf, is directly varied to central frequency, f_0

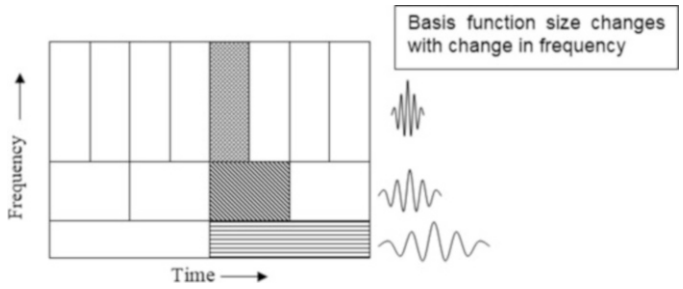

Fig. 4.8 Time–frequency plane for basis function of wavelet analysis

$$\Delta f / f_0 = \text{Constant} \tag{4.11}$$

With the help of a bandpass filter with constant Q, the collection of the wavelet representation filter bank is carried out. In this case, time resolution also changes with change in middle frequency. This step will gratify the Heisenberg uncertainty principle, but now at high frequencies time resolution becomes randomly better, whereas on the other hand frequency resolution becomes randomly good at low frequencies. Thus, wavelet analysis offers time and frequency selectivity. So, to increase the time resolution a wavelet can be used wherein separation of two short bursts is accomplished by selecting higher analysis frequencies . Thus, one should use wavelet analysis when the signal contains high-frequency parts having very short duration and low-frequency parts for a long duration. In the wavelet transform, the size of the windows is not constant: it varies [6].

4.5 Multiresolution Algorithm

The wavelet transform is also referred as the multi-resolution algorithm [4]. If it is desired to compute the approximation values of $S_{m,n}$ and discrete wavelet transform $T_{m,n}$ for the input signal $S_{0,n}$ using the decomposition algorithm, one first computes $T_{1,n}$ and $S_{1,n}$ from the input coefficients specified by $S_{0,n}$ as follows:

$$\begin{aligned} s_{1,n} &= \frac{1}{\sqrt{2}} \sum_k ckS_0, \quad 2n+k \\ T_{1,n} &= \frac{1}{\sqrt{2}} \sum_k bkS_0, \quad 2n+k \end{aligned} \tag{4.12}$$

In the same way, $S_{2,n}$ and $T_{2,n}$ can be calculated using $S_{1,n}$:

$$\begin{aligned} s_{2,n} &= \frac{1}{\sqrt{2}} \sum_{k} ckS_1, \quad 2n+k \\ T_{2,n} &= \frac{1}{\sqrt{2}} \sum_{k} bkS_1, \quad 2n+k \end{aligned} \tag{4.13}$$

Next, from approximation coefficients $S_{2,n}$, one can find $S_{3,n}$ and $T_{3,n}$ and so on up to scale indices M, where one will be able to compute $S_{M,0}$ and $T_{M,0}$. The decomposition of the discrete input signal at scale index M with some array coefficients and a single value $S_{m,0}$ corresponds to $a = 2m$ and $b = 2\ mn$ location with length $N = 2M$. Hence, it specifies the value of m and n for important coefficients, which are mainly specified by $1 < m > M$ and $0 < n < 2M - m - 1$.

Multi-resolution analysis (MRA) decomposes the vector space $L^2(R)$ in a set of subspaces, represented as

$$\begin{aligned} &\ldots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \\ &V_j \subset V_{j+1} \quad \text{for} \quad \forall j \in Z \end{aligned} \tag{4.14}$$

In this case, the union of this subspace is closure to $L^2(R)$, that is, $\bigcap_{j \in Z} V_j = \{0\}$. The intersection of subspaces is a set containing zero vector. Consider that $x = [x_1, x_2 \ldots x_N]$ forms a linear vector space of linearly independent basis vectors having N dimensions and given as $a_1, a_2, a_3 \ldots a_N$, and a set of N-dimensional real-valued vectors. This vector space may be represented as a linear combinations of all the foregoing basis vectors. This N-dimensional vector space is V_N. Next, approximation vectors can be considered in this N-dimensional space by vectors in a subspace of lower dimension, say, $N - 2$. Suppose all linear combinations can be generated of just $N - 1$ basis vectors, say $a_1, a_2 \ldots a_{N-1}$, then this will form a vector space V_{N-1}, which is a subspace of V_N.

Similarly, by dropping the last basis vector at every step, subspaces V_{N-2} with dimension $N - 2$, V_{N-3} can be constructed with dimension $N - 3$ and so on, up to V_1 with dimension 1. The subspace V_1 has a single basis vector, a_2. These vector spaces form a nested sequence of subspaces, $V_1 \subset V_2 \subset V_3 \subset \ldots V_{N-1} \subset V_N$. Now, it is required to approximate a vector x in V_N by a vectorin V_{N-1}. At the same time it is also required to reduce error between the original vector x and the new vector (let us call it x_{N-1}) in the space V_{N-1}. The only way to reduce error is to minimize the length of error vector e_{N-1}, where e_{N-1} is given by $e_{N-1} = x - x_{N-1}$, which can be obtained by

$$\langle e_{N-1}, a_k \rangle = 0, \quad k = 0, 1, 2, \ldots, N-1 \tag{4.15}$$

As shown in Fig. 4.9, x_{N-1} is the orthogonal projection of x on vector space V_{N-1}.

If this process of projecting throughout the entire sequences of subspaces continues, x_{N-1}, x_{N-2}, ... and so on can be computed. These results will go into sequences of error vectors: e_{N-1}, e_{N-2}, e_{N-3},... e_1. This error vector represents the

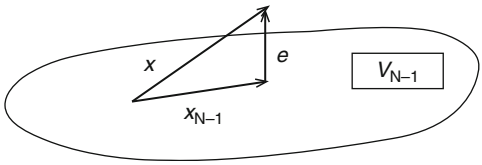

Fig. 4.9 Vector diagram

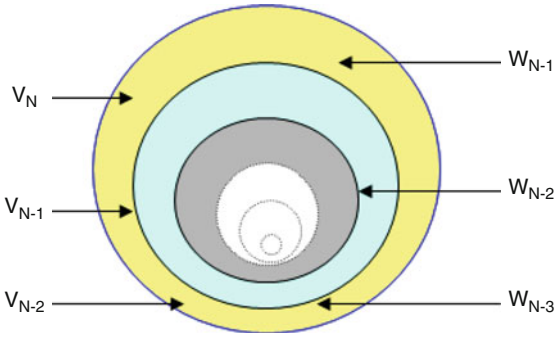

Fig. 4.10 Vector space

amount of detail lost to the subsequent approximation. In this process, vector x is represented by various levels of resolution in different spaces [7]. The difference between subspaces V_N and V_{N-1} can be given by another subspace. Assume this subspace as W_{N-1}. It can be said that the last part of vector x may be in this subspace. So, W_{N-1} contains detail components. The sequences of error vectors e_{N-1}, e_{N-2}, e_{N-3},... e_1 form an orthogonal set belonging to the one-dimensional space of W_{N-1}, W_{N-2}, ... W_1. Mathematically, this can be represented as $V_N = V_{N-1} \oplus W_{N-1}$, and V_{N-1} is equal to $V_{N-2} \oplus W_{N-2}$ and so on (Fig. 4.10). If this process is extended to infinity, then the final average will be zero and the signal is decomposed into all detail coefficients. Then

$$L^2(R) = \bigoplus_{j=\infty}^{-\infty} W_j \tag{4.16}$$

$$\begin{aligned} V_N &= V_{N-1} \oplus W_{N-1} = V_{N-2} \oplus W_{N-2} \oplus W_{N-1} = \cdots \\ V_N &= V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{N-2} \oplus W_{N-1} \end{aligned} \tag{4.17}$$

where subspace V_0 contains the last average or final approximation components, and subspaces W_0, W_1, W_2, ... W_{N-2}, and W_{N-1} contain detail vectors or error vectors. It can be seen that the original vector x was constructed from the final approximation vector and detail vector:

$$x = x_1 + e_1 + e_2 + e_3 + \cdots + e_{N-1} \tag{4.18}$$

Multi-resolution analysis involves approximation of the functions in a sequence of nested linear vector spaces (Fig. 4.10).

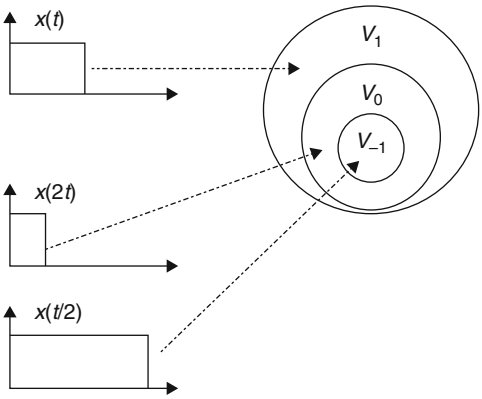

Fig. 4.11 Space and resolution level

If the space can have some function, $x(t) \in V_0$, then $x(2t) \in V_1$ and $x(t/2) \in V_{-1}$. This property says that the dilated function with dilation factor two belongs to the next coarser subspace and dilation factor one-half belongs to the next finer space. There exists a function known as the scaling function $\phi(t)$ such that $\phi(t - k)$ is the basis for V_0. Translation and dilation of this basis function can represent approximation of any function $f(t)$ [8]. Figure 4.11 shows the space and resolution level.

References

1. Dhar, P. K., & Shimamura, T. (2015). *Advances in audio watermarking based on singular value decomposition*. Cham: Springer.
2. Thanki, R., Borisagar, K., & Borra, S. (2018). *Advance compression and watermarking technique for speech signals*. Cham: Springer.
3. Bebis, G. (2001). *Short time Fourier transform (STFT). Image processing fundamentals*. CS474/674.
4. Resnikoff, H. L., & Raymond Jr., O. (2012). *Wavelet analysis: the scalable structure of information*. Cham: Springer.
5. Leisenberg M (1995). Hearing aids for the profoundly deaf based on neural net speech processing. In *ICASSP-95, 1995 I.E. International Conference on Acoustics, Speech, and Signal Processing* (Vol 5, pp. 3535–3538). Piscataway: IEEE.
6. Agbinya, J. I. (1996). Discrete wavelet transform techniques in speech processing. In *TENCON '96. Proceedings, 1996 I.E. TENCON. Digital Signal Processing Applications* (Vol. 2, pp. 514–519). Piscataway: IEEE.
7. Xueying, Z., & Zhiping, J. (2004). Speech recognition based on auditory wavelet packet filter. In *Proceedings of 7th International Conference on Signal Processing, 2004. ICSP '04* (Vol 1, pp. 695–698).
8. Agbinya, J. I. (1996). Discrete wavelet transform techniques in speech processing. In *TENCON'96. Proceedings, 1996 I.E. TENCON. Digital Signal Processing Applications* (Vol. 2, pp. 514–519).

Chapter 5
Speech Signal Enhancement Using Adaptive Filters

5.1 Introduction

In general, for any speech enhancement process, the prime requirement is to record speech. Obviously, speech can be recorded through a microphone from the environment. At the time of recording, surrounding noise is always present with intended speech. Now, this type of noisy speech is not suitable for hearing aids, which are meant to amplify sound. Moreover, the type of microphone is responsible for the capturing of sound and level of noise. In the quality of speech, microphone directivity is also performing an important task. A sensitive microphone will capture a greater amount of noise with speech. In general, in one-microphone hearing aids, directive microphones are used, whereas in two-microphone hearing aids, one directive microphone is located at the front, and the other microphone, an omnidirectional microphone, is on the back.

Real speech is given to the voice activity detection (VAD) algorithm. VAD is a very important process for identifying the occurrence of utterance in continuous speech. Again, the main aim for applying VAD is to extract the occurrence of voice periods. The flow of the algorithm is shown in Fig. 5.1. After detecting silences in speech, all the silent periods are the threshold, so it can be cleaned at least from added unwanted noises.

In consequence, after capturing a voice, it is important to reduce noise from the given speech. The next task is to apply to speech a noise reduction algorithm. An adaptive algorithm can be chosen on the basis of the loudness of speech and the amount of noise added to the signal. A main criterion for choosing the algorithm is the signal-to-noise ratio (SNR) of the given input. In general, by selecting an appropriate adaptive algorithm, the output of the algorithm is clean speech. Per the efficiency of the algorithm, Peak Signal to Noise Signal (PSNR), SNR, and mean square error (MSE) can be improved and speech can be reached up to mark. Without reducing the main intelligibility from the speech, noise reduction is a very critical issue.

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_5

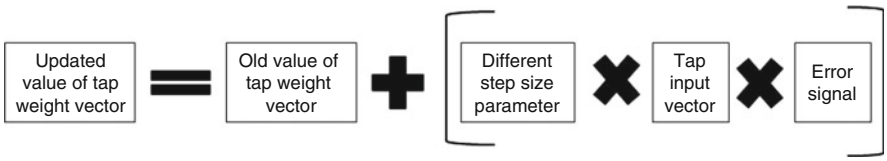

Fig. 5.1 LMS algorithm in words

Then, cleaned speech is given to a band enhancement algorithm. Deaf persons often suffer from a specific frequency insensitivity. By studying the audiogram it can be clearly identified which band of the frequency needs enhancement for the individual user. The audiogram of the specific patient suggests an algorithm in the wavelet domain, which band wavelet coefficient needs enhancement for proper reproduction of speech signal. Basically, the algorithm works in three steps. In this chapter, application of various adaptive filters such as least mean squares (LMS), normalized LMS (NLMS), and recursive least squares (RLS) [1–5] are described for enhancement of a noisy speech signal for digital hearing aids.

5.2 Steps for Speech Enhancement Process

The steps for enhancing speech signals are as follows:

Step 1: Record a noisy speech signal from the environment.
Step 2: Apply the VAD algorithm to find the non-speech period from the given input speech signal.
Step 3: Enhance non-speech periods by applying thresholding.
Step 4: Directly silence the cleaned speech signal to adaptive filters: LMS, NLMS, and RLS.
Step 5: Compare all the results of the entire applied filter algorithm using various quality measures such as MSE, PSNR, and SNR.

5.3 Implementation Flow of VAD Algorithm

The steps for implementing the VAD algorithm are given below. This algorithm is divided into two parts: finding the energy and zero crossing rate of the speech signal. The steps for finding the energy of the speech signal are as follows:

Step 1: Read noisy speech signal and determine its nature.
Step 2: Compute the first 100-ms interval of the speech signal, referred to as a noise profile.
Step 3: Decide the average magnitude and zero crossing for this time interval.
Step 4: Calculate statistical characteristics (mean and standard deviation) of the speech signal during this time interval.

Step 5: Compute the energy threshold and zero crossing using statistical characteristics and average magnitude.
Step 6: Search the average magnitude profile to find the time interval in which it always exceeds a very traditional threshold, Interval Threshold Upper (ITU).
Step 7: Go back to search from the point where En first exceeded the threshold ITU.
Step 8: Search tentative point N1 where En first falls below a lower threshold Interval Threshold Lower (ITL).
Step 9: Repeat the procedure to find the tentative endpoint, N2.

The steps for zero crossing rate of speech signal are given here:

Step 1: After finding the energy points between time intervals N1 to N2, find the zero crossing rate of the speech signal.
Step 2: Move backward from N1 and forward from N2, comparing the zero crossing rate to a threshold Interval Zero Crossing Threshold (IZCT).
Step 3: If the zero crossing rate exceeds the threshold five or more times, N1 is moved back at the point where IZCT is exceeded.
Step 4: Otherwise, define N1 as the beginning point. Repeat the procedure for the real endpoint.
Step 5: Pass the silence-detected speech signal to the signal transform algorithm.
Step 6: Decide the level of thresholding, type of thresholding, and signal transform.
Step 7: Give the signal as input in an adaptive filter algorithm to further filter the speech.

5.4 Speech Enhancement Process based on LMS Algorithm

In this section, the speech enhancement technique based on LMS filter is discussed. The simplicity of the LMS algorithm and ease of its implementation mean it is the best choice for many real-time systems. The LMS algorithm is shown in Fig. 5.1.

- Describe the desired response. Also set each coefficient weight to zero.
- Now apply the movement to all the samples in the input array one position to the right; then load the current data sample n toward the first position in the array. Next, compute the output of the adaptive filter by multiplying the respective elements in the array of filter coefficients by the corresponding element in the input array. After this, all results are summed to give the output corresponding to those data that were earlier loaded into the input array.
- Error must be calculated before the filter coefficients can be updated. Simply find the difference between the desired response and the output of the adaptive filter.
- To update the filter coefficients, multiply the error by the learning rate parameter μ, and then multiply the result with the filter input, and carry out addition of this result to the values of the previous filter coefficients.

The flow of implementation is shown in Fig. 5.2 for speech signal enhancement using the LMS adaptive filter, which shows the real flow of the execution of the algorithm in detail by maintaining every sequence process.

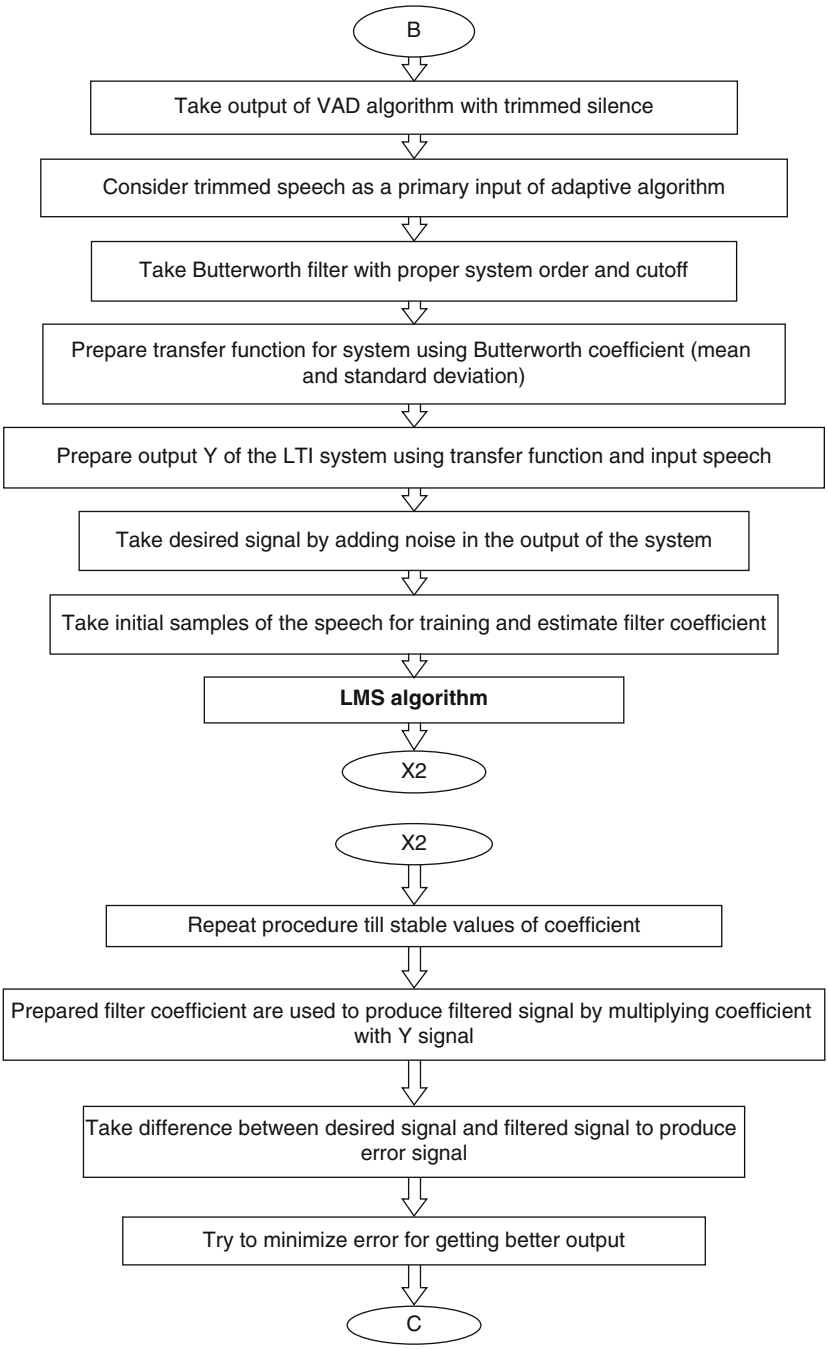

Fig. 5.2 Implementation process for speech signal enhancement using LMS algorithm

In general, for real-time implementation the speech file is taken as a reference. The recorded file shows utterance of numbers, which includes different numbers starting with 'one.' In that, the speech signal contains all the types of language parameters such as consonants, diphthongs, vowels, voiced sound, unvoiced sounds, nasals, fricatives, and weak fricatives. The main reason for covering all the types of speech utterance is to see the effect and noise reduction for different types. In a given concept with real types of speech, different noises are included to make the noisy effect of speech. Types of noises added in the speech signals are discussed in the following sections.

- Software-generated white noise uses a random function that includes all the frequencies. A characteristic of white noise is that it can include the entire frequency component in the spectrum.
- Traffic jam noise includes different types of sounds of vehicles. It gives a background effect as anyone may walk on the road and sometimes at that time be conducting a conversation. In this sound many vehicular motion sounds give simultaneous effects and thus are converted into noise.
- Babble noise is a very dominant type of noise in real life, because in common places where speech conversation is going on there is always background noise that does not contain any actual types of unwanted sound. However, in a gathering of too many people, when all are speaking, one background sound is generated that is the combination of the simultaneous conversation of many persons.
- Camera rewind noise is one unusual type of noise that generates a very irritating, intolerable noise. Basically in a normal situation the camera rewind can be compared with every type of noise such as a door closing and opening or any type of continuous noise that causes irritation to normal hearing.
- Another considered noise is the sound of applause. It is one type of natural noise that creates one continuous type of sound and because of this one surrounding layer is generated with normal speech.
- On the highways and at other traffic places one often has to experience heavy vehicular starting noises from all sides. So, the next considered important noise is the sound of trucks starting. That noise is very cumbersome and loud and will greatly mask the intelligence of speech.

In the first step, the original speech file is read (as shown in Fig. 5.3) and any one given noise is taken to reproduce the effect of noisy speech. The working of the algorithm is discussed next.

Figure 5.3 shows two waveforms. The upper speech waveform is for the utterance of 'one, two, and three' scaled in the range of approximately 7 s. The second part of the figure shows the spectrogram of this speech. The spectrogram is able to represent three axes. In the spectrogram the elapsed time for utterance is approximately 7 s. From the spectrogram the noticeable frequency is maximum to about 4.5 kHz. So here the second axis shows the frequency contained by the utterance. The third axis shows the magnitude of a specific frequency. If the given utterance is some specific frequency and if that frequency has higher magnitude, it is represented with a dark

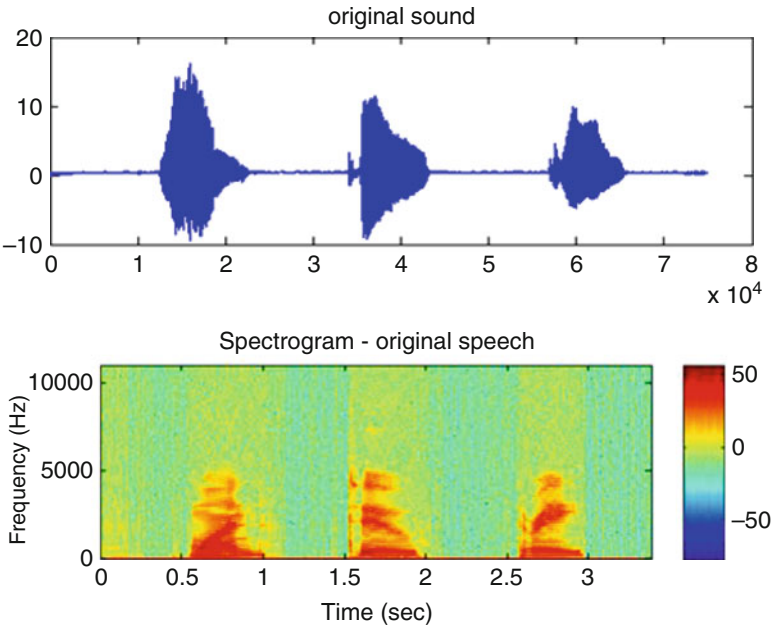

Fig. 5.3 Original utterance waveform of 'one two three' and its spectrogram

red color. The intensity of the color shows at what time the frequency was of high magnitude.

The speech signal shown in Fig. 5.3 is taken as a reference for developing the performance of the adaptive algorithm. The selected speech file is of too many bytes, approximately 25 s long. However, for the experimental aspect, here from a large number of samples only 75,000 samples are taken from the speech. The number of words in the speech is automatically displayed by the peak calculation. Its bit rate is 352 kbps; in nature, this is a mono-recording. The audio sample rate is 22 kHz per second and one sample can be encoded with 16 bits. The original speech is the proper number of silences, and duration of silence is also moderate, so in normal conditions speech can be mixed with a proper value noise signal. Moreover, the wave file of used speech has a loudness level of 84.75 dB. The defined level is sufficient for the normal process of filtering. Per the hearing level chart, this is noticeably recognizable by human ears at the time of the conversation.

Consequently, to solve the purpose in the presence of different noise backgrounds, the speech utterance is rerecorded. In the presence of different noises the speech signal is merged with different types of noise signals (Table 5.1) of different loudness. Now, the prepared noisy speech signal can be used to apply different adaptive algorithms. Following are the most important noises that are taken as a reference to prepare noisy speech.

Table 5.1 Different noise levels

Types of noises	Strength in dB
Babble	51.0031
Traffic jam	31.8613
Truck starting	28.6278

Now, the noisy speech signal is prepared by recording sound in the presence of different background noises. The next step for quality improvement is application of the VAD algorithm. As discussed earlier, maximum noise will be present in the speech when the silence period occurs because in the silence period there is actually no speech. So, background noise is reflected as a maximum when there are silences.

The prime necessity of this algorithm is to acquire an idea about merged speech only in terms of recorded speech, or another directional microphone can be used with the omni-directional microphone that can handle sounds other than voices. If unvoiced duration from speech is detected, then it is very easy to predict and carry our statistic of the incoming noises with the speech signal. The recorded received speech signal is given to the zero crossing rate detector and energy vector-based VAD algorithm, using which speech occurrence can be achieved. Here different noise signals are taken and prepared as noisy speech signals given to different algorithms for noise reduction.

5.4.1 Results for White Noise Signal

The white noise signal evenly distributes noises with the entire frequency component in detail. Figure 5.4 shows a white noise signal and its spectrogram. A noisy speech signal with white noise with its spectrogram is given in Fig. 5.5.

Now, the noisy speech signal is given to the VAD for silence searching; after searching silences, almost all the silence part are threshold to zero value, which is why in the speech it is a very smaller amount of speech when utterances are present. Most of the silence part is giving zero value in the presence of thresholding. Next, silence removed speech is given to the LMS adaptive algorithm.

As discussed in Chap. 3, LMS is a very efficient algorithm for noise reduction in the signal. In the lower waveform per the concept of the algorithm two plots can be seen. One waveform with blue shows the true value of the noisy speech signal. By holding function, the estimated waveform is plotted with red coloring, as shown in Fig. 5.6. The estimated enhanced speech signal is generated in simulation by updating the vector continuously. Observation of the waveform informs us that estimated and true output is changing values in some matter and that these values cause noise reduction in the system.

The plot of Fig. 5.7 shows the difference between true and estimated output. The main task of the algorithm is to reduce this error so that noise can be reduced and clean speech can be clearly detected in the output.

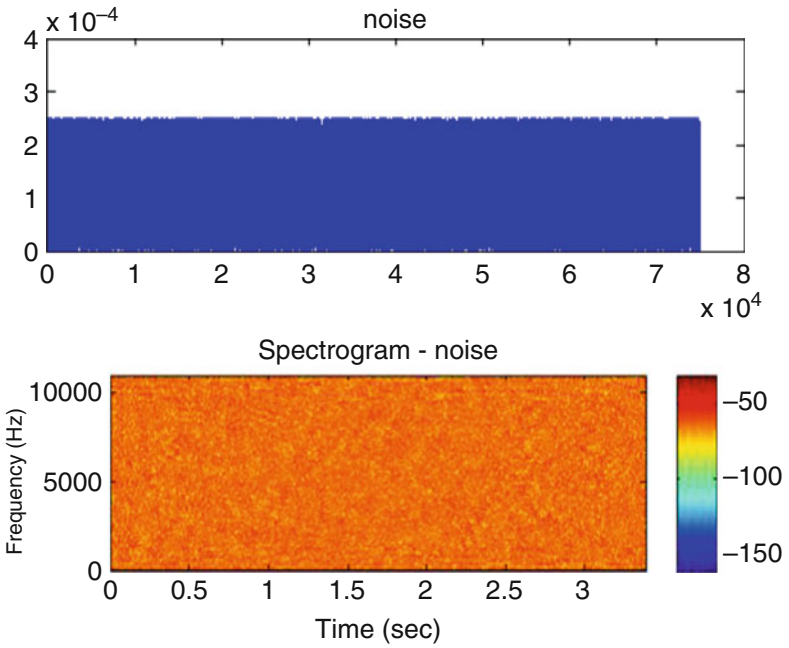

Fig. 5.4 White noise signal and its spectrogram

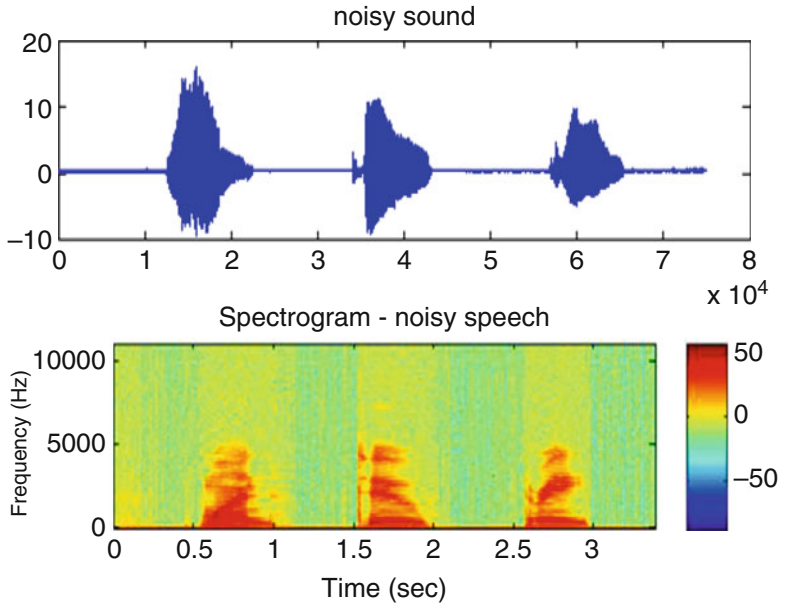

Fig. 5.5 Noisy speech signal with white noise and its spectrogram

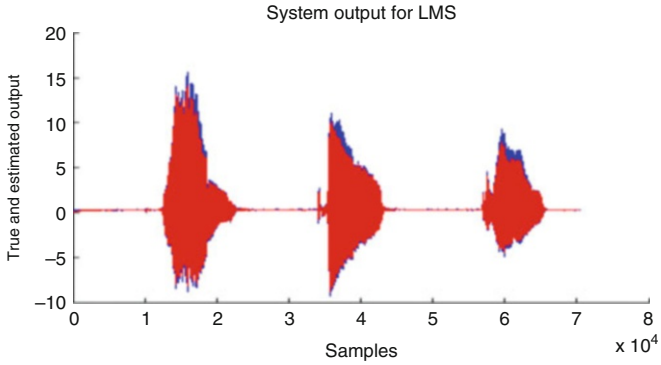

Fig. 5.6 True and estimated output in LMS algorithm for white noise signal

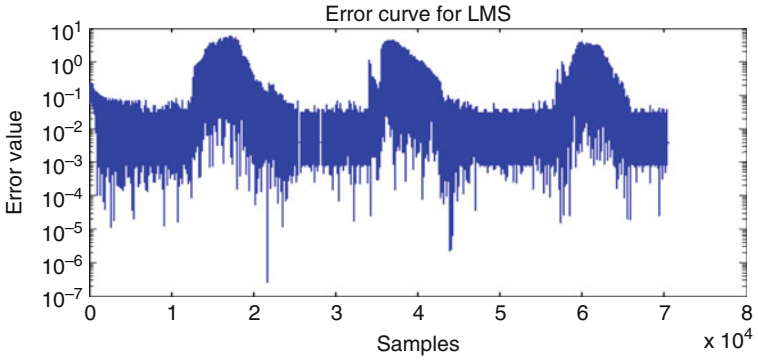

Fig. 5.7 Error between true and estimated output in LMS algorithm for white noise signal

The number of filter weights is defined by the system order. Here the mentioned system order is five. So basically five initial filter weights can be selected by defining the Butterworth filter with order two and cutoff frequency 0.25 Hz. Output of filter design is the numerator coefficient and denominator coefficients. From the taken coefficient one system can be defined in the form of the transfer function. Then, by taking the inverse z transform of that transfer function, the initial values of the coefficient can be defined. That set of coefficients start to converge according to step size parameters. The LMS algorithm continuously updates value of weight per discussion of the theory of LMS in Chap. 3. For updatation, the taken speech samples are 75 samples and the training vector is 3000. In Fig. 5.8, two different symbols are shown giving the actual weights and estimated weight for the noise reduction after convergence of the algorithm at some steady-state level. Now, the decided coefficient can be used for whole file for noise reduction.

Figure 5.9 shows the learning curve for LMS. The total first 3000 samples are taken for proper convergence.

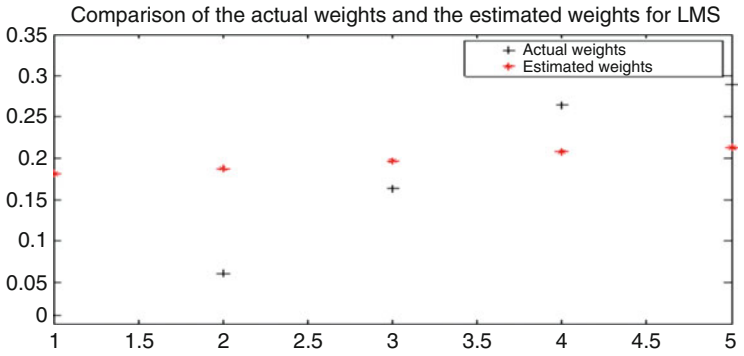

Fig. 5.8 Actual and true filter weight in LMS algorithm for white noise signal

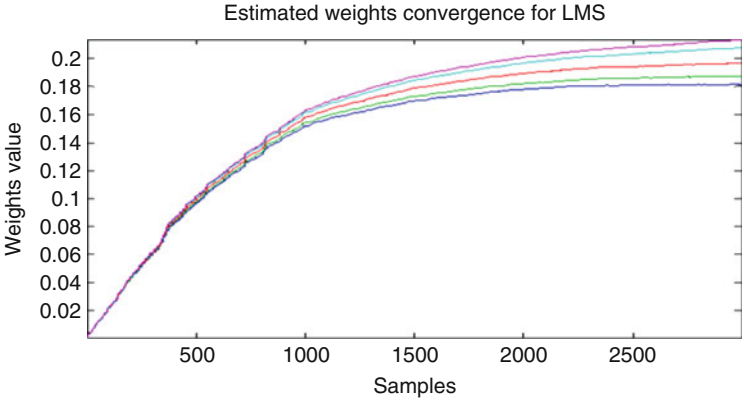

Fig. 5.9 Estimated convergence learning curve in LMS algorithm for white noise signal

Learning curve to get into the steady-state mechanism, selected mu is 0.836 for initial 1000 samples, and for the next 2000 samples, mu is 0.766, to get into the normal pattern. These two patterns can be observed from Fig. 5.10. One pattern shows output of VAD algorithm in that only silences are detected and noise is reduced only in the silence period. After that, the same waveform is given to the LMS adaptive algorithm. With the effect of the algorithm, the values can be observed. It is clearly seen that LMS is able to cancel efficiently noise from speech as well as from silence periods.

Figure 5.11 shows clear and cleaned speech as an output of the LMS adaptive algorithm. Also, the spectrogram can be seen. The spectrogram is giving the correct value compared to the previous noisy spectrogram for the signal. The wavelet tool is used to get the actual comparison in the frequency domain. The output of VAD is a trimmed speech signal, which is given to the multi-resolution wavelet algorithm.

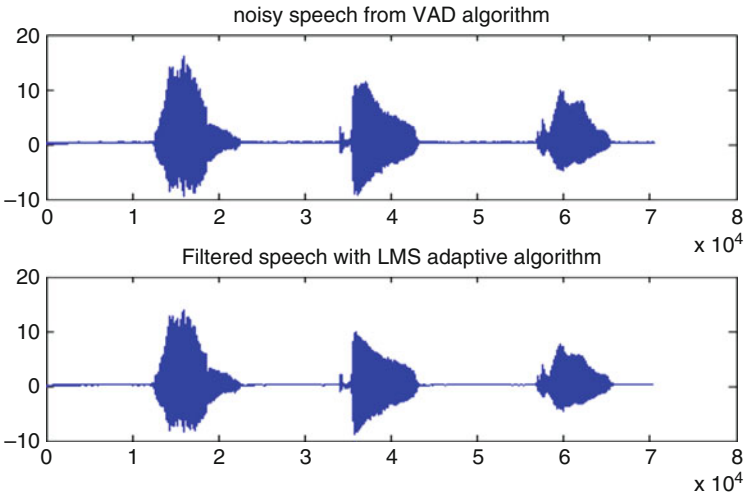

Fig. 5.10 Noisy speech signal with white noise signal from VAD algorithm and filtered speech signal with LMS algorithm

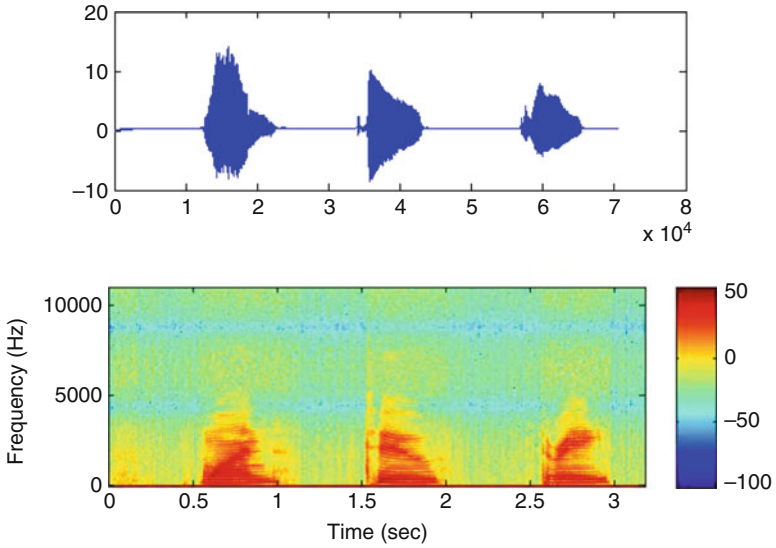

Fig. 5.11 Filtered speech signal by LMS algorithm for speech affected by the white noise signal and its spectrogram

From that, the individual band coefficient can be derived. The same multi-resolution wavelet algorithm is used for LMS filtered speech. The comparison can be seen in Fig. 5.12. Each and every band is less compared to the first band, which allows the conclusion that noise can be reduced in a very efficient way by LMS.

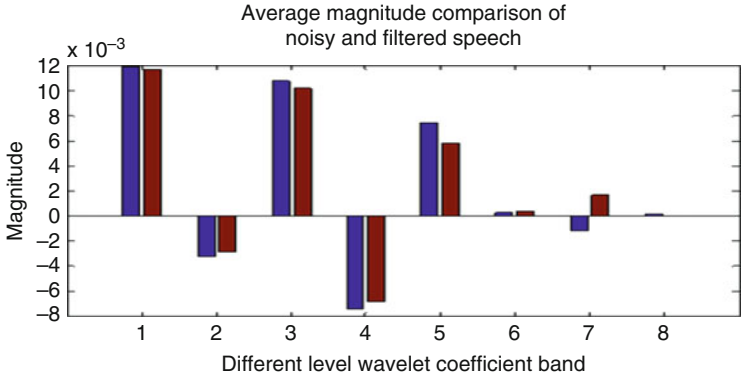

Fig. 5.12 Comparison of noisy speech signal with white noise signal and filtered speech signal in wavelet domain for LMS algorithm

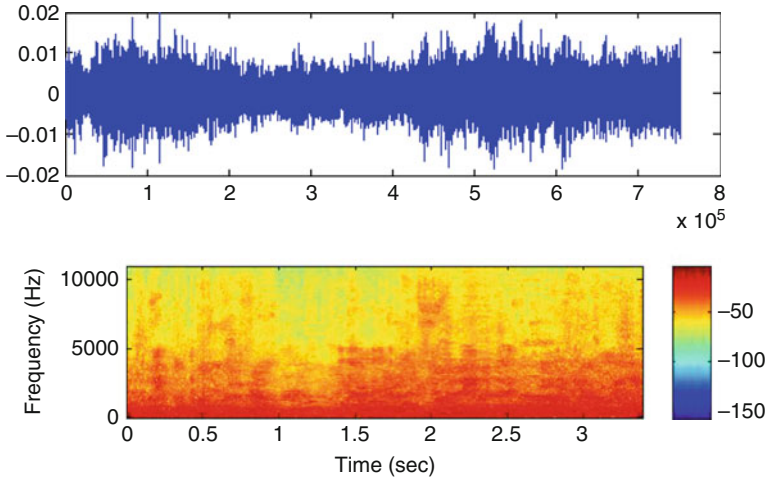

Fig. 5.13 Babble noise signal and its spectrogram

5.4.2 Results for Babble Noise Signal

Babble noise includes different types of merged speech generated by more than one user. Often at the time of a conversation babble noise is present. Figure 5.13 shows babble noise and its spectrogram.

By following the steps, performance of the LMS can be detected in the occurrence of babble noise. The application of the algorithm is discussed in the previous section. As discussed previously, multi-talker babble noise is very similar to the speech signal. Figure 5.14 shows a noisy speech signal after addition of babble noise. It can be observed that the maximum amount of noise can be detected in the

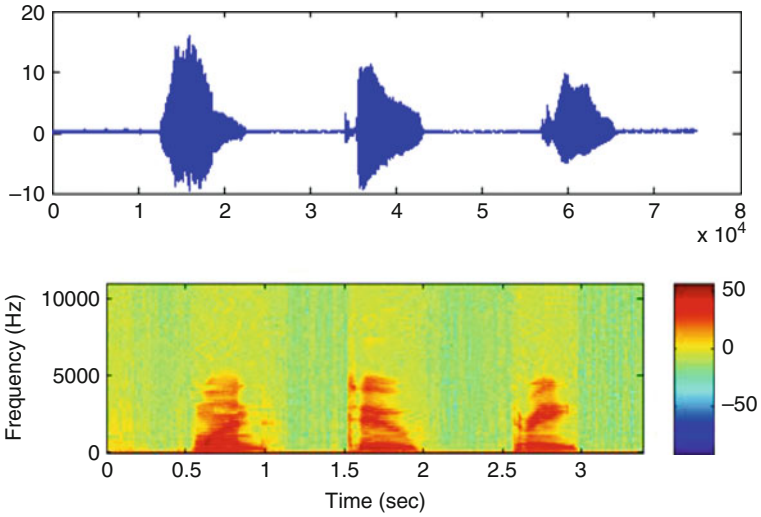

Fig. 5.14 Noisy speech signal with babble noise and its spectrogram

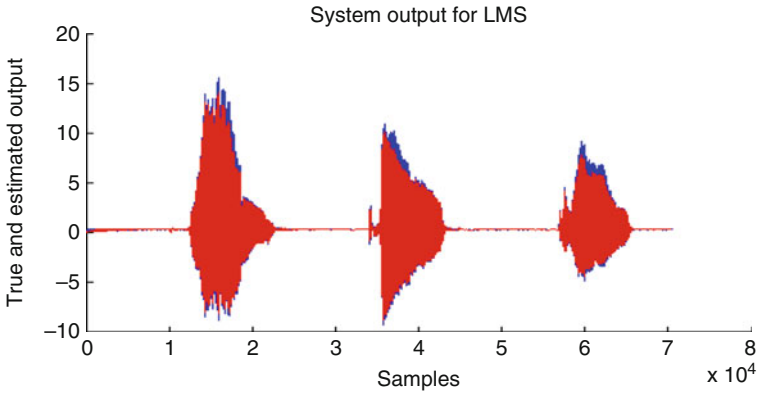

Fig. 5.15 True and estimated output in LMS algorithm for babble noise signal

silence times. During utterance, noise is in an additive manner. Now, noisy speech is given to the adaptive algorithm. Per the logic of the LMS algorithm, it works on the principle that it requires two inputs simultaneously: one is known as a reference input and second input is the desired input.

In Fig. 5.15 the two inputs are shown with two differently colored waveforms. The speech waveform plot (blue) is known as a true input of the algorithm whereas the speech signal (red) is known as an estimated input. It can be observed that always there is a difference between true and estimated input. The estimated waveform is not completely superimposed on the actual waveform. Because of that, some sort of

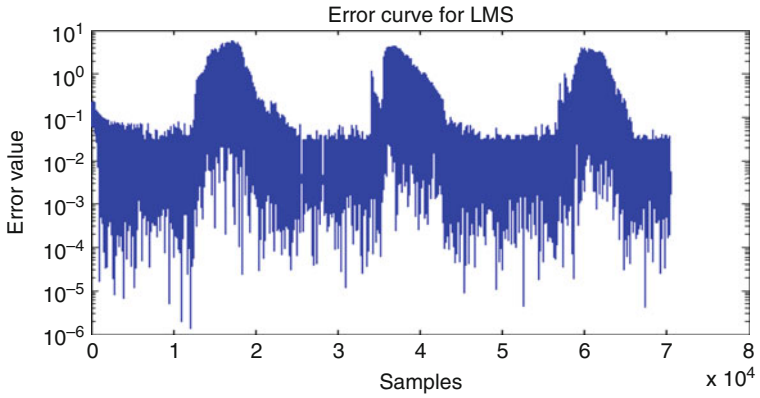

Fig. 5.16 Error between true and estimated output in LMS algorithm for babble noise signal

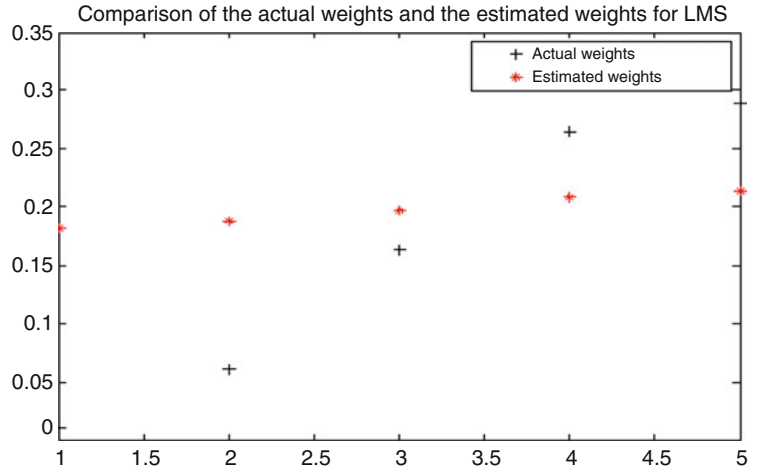

Fig. 5.17 Actual and true filter weight in LMS algorithm for babble noise signal

error is present in the signal. An error curve can be seen in Fig. 5.16. Here, error shows the sample by sample difference for the training vector, with difference plotted every time. The error curve designates the total set of speech input.

Subsequently, Fig. 5.17 shows an important result. As discussed initially, for starting the adaptation some reference level is required. By designing the appropriate order of the Butterworth filter, the updatation is started in LMS. Initial filter values are plotted in Fig. 5.17 as an actual weight. The training vector for weight adjustment is 3000 samples. In trying for reduction in MSE and by taking the appropriate step size for a given training vector, new derived coefficients are marked with red. Now derived values of the coefficient are finalized for filtering of the total speech file. By the nature of training vector samples, these values can be identified then that are

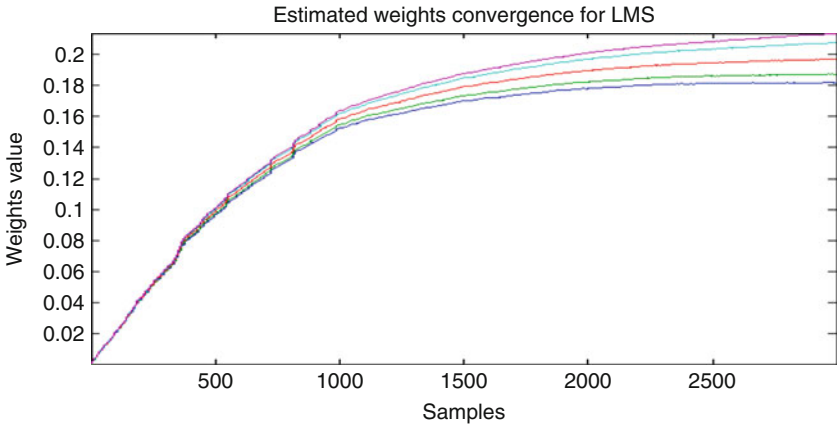

Fig. 5.18 Estimated convergence learning curve in LMS algorithm for babble noise signal

suitable for the total filtering operation of the speech signal. Variation in true and estimated values reflects the deviation required for noise reduction and enhancement in the values of weight values for reduction of noise in the speech file.

Figure 5.18 shows an estimated convergence learning curve in LMS for babble noise. In the layout, five different color plots are shown. As the initial fifth order, the Butterworth filter with cutoff frequency of 0.25 Hz is considered. In adaptation starting with initial values of filter weight, modification of filter weight is necessary; that enhancement is a gradual process. Gradually all the filter weights are set with the training vector value and with that after some time it will take the steady-state value. According to selected values of the mu sample set, convergence takes place. Per the characteristics of LMS it can be observed from Fig. 5.18 that the convergence rate is gradual. Initially, all the five curves are moving in line. No variation can be detected. Progress in the number of samples shows improvement in variation. By this time with some gradual variation the convergence curve becomes stable. By the end of convergence, the curve number of coefficients achieves its final values. Now the new filter vector is prepared according to input of incoming noisy speech characteristics. Using the same set of filters, noise removal can be achieved.

Figure 5.19 shows two waveforms, one for noisy output and the other for filtered output. The first layout in the simulation results shows noise reduction in the silence part as an output of the VAD algorithm. After application of VAD also, the speech part remains as it is with noise. Noisy speech after applying the LMS algorithm looks similar to the original speech. Most of the noise values are removed from the silence as well as the utterance part efficiently.

Figure 5.20 indicates speech with noise removed and its spectrogram. It can be observed that the spectrogram here is very carefully identified and that modified speech is very similar to the original speech in nature. The spectrogram shows representations of three axes, which include time, frequency, and color depth showing the intensity of that enhanced frequency.

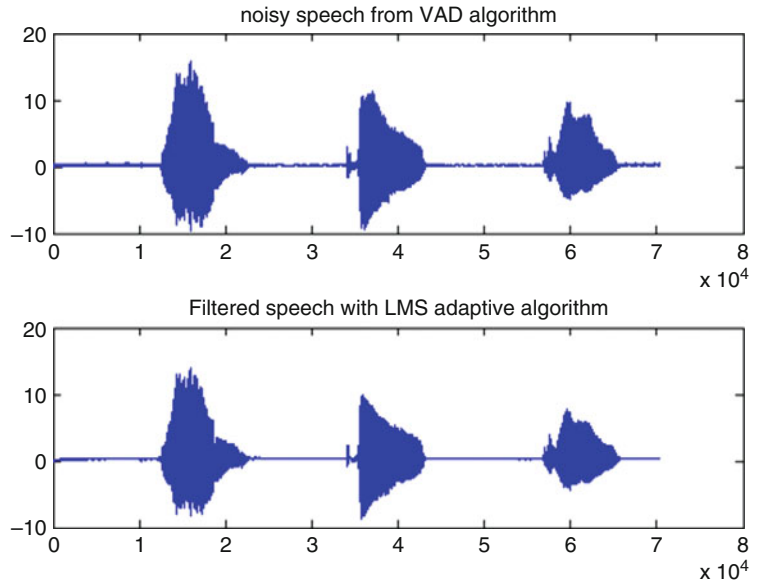

Fig. 5.19 Noisy speech signal with babble noise signal from VAD algorithm and filtered speech signal with LMS algorithm

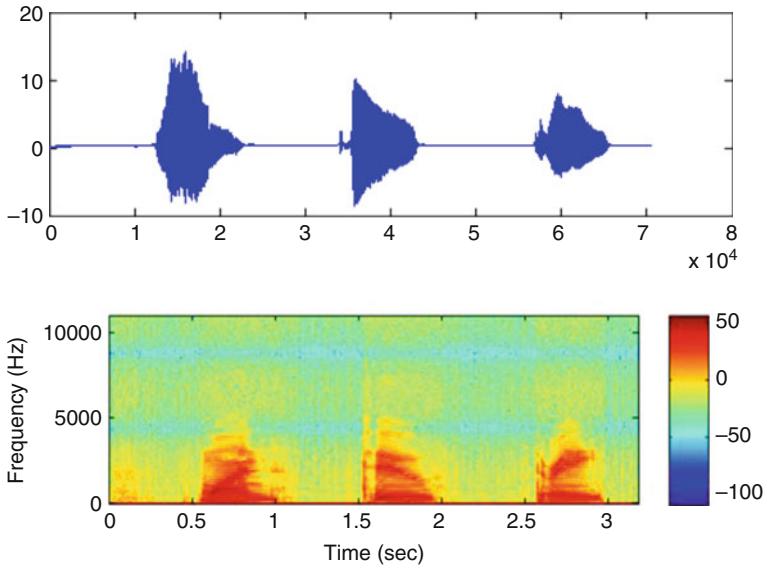

Fig. 5.20 Filtered speech signal by LMS algorithm for speech affected by babble noise signal and its spectrogram

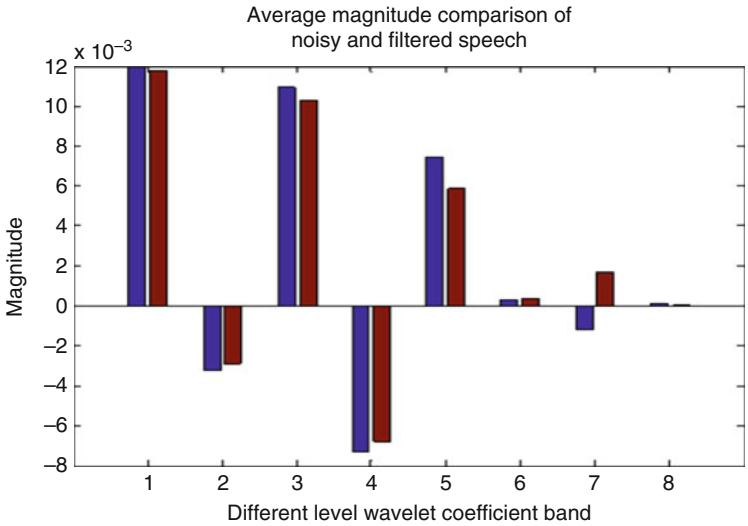

Fig. 5.21 Comparison of noisy speech signal with babble noise signal and filtered speech signal in wavelet domain for LMS algorithm

The measured output of the system is plotted in Fig. 5.21, indicating the amount of variation in the original speech and the LMS-filtered enhanced speech. Each speech file is given to the wavelet multi-resolution application for each band variation. The simulation result shows the band layout for each frequency present in the speech. Before noise reduction the magnitude of the confinement is higher and after noise reduction it can be observed that the magnitude is reduced.

5.4.3 Results for Traffic Jam Noise Signal

In public places when conversation is going on surrounding traffic is usually present. Because of that, whatever unwanted noise is generated with speech is known as traffic noise. From the spectrogram of a traffic signal it can be observed that high to low frequencies are present with different magnitudes. That type of continuous noise is scattered throughout the entire range of the frequency. Figure 5.22 shows a traffic jam noise signal clearly with its spectrogram. The nature of speech after adding this type of background noise can be observed in Fig. 5.23, which contains information about speech and noise signals in the spectrogram.

Figure 5.24 shows, as discussed previously, both true and estimated output with all else as described in the case of babble noise. Simply, it is showing the true and estimated output of the system as required by the LMS to execute. Similarly, as described earlier, Fig. 5.25 shows the sample by sample difference in the presence of true and estimated output in the case of traffic jam noise. Figure 5.26 shows the

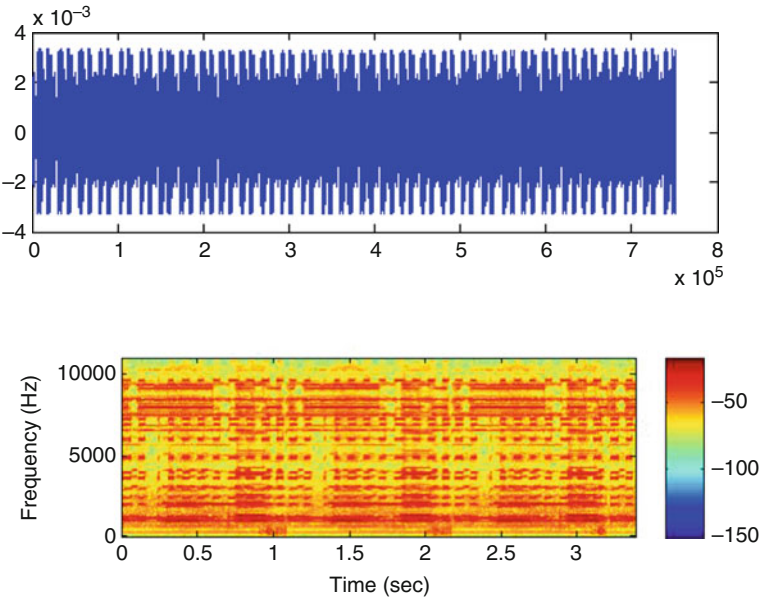

Fig. 5.22 Traffic jam noise signal and its spectrogram

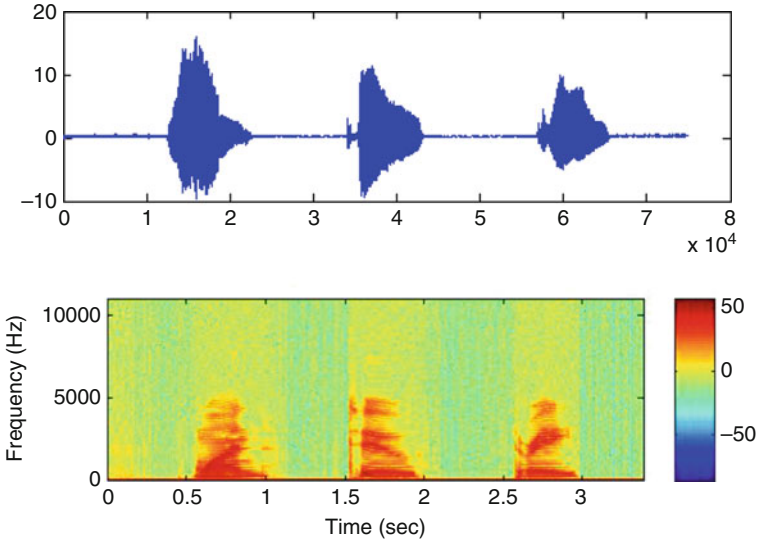

Fig. 5.23 Noisy speech signal with traffic jam noise and its spectrogram

actual and estimated weight coefficients for proper operation. Deviation between actual and estimated coefficients shows the amount of modification required for noise reduction.

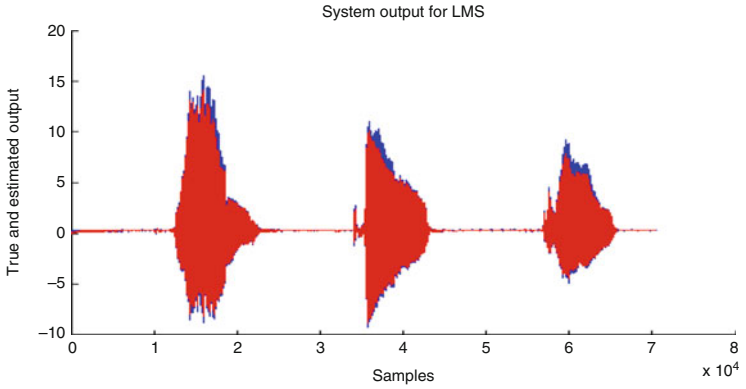

Fig. 5.24 True and estimated output in LMS algorithm for traffic jam noise signal

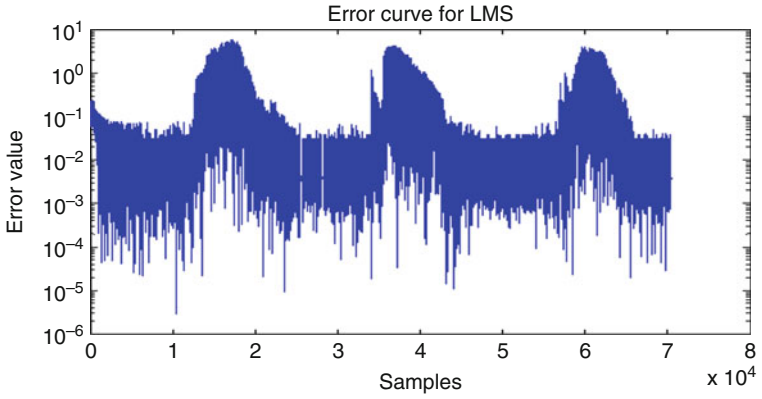

Fig. 5.25 Error between true and estimated output in LMS algorithm for traffic jam noise signal

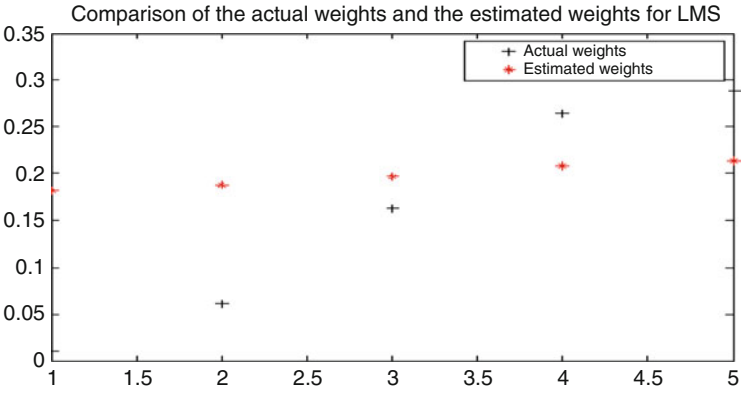

Fig. 5.26 Actual and true filter weight in LMS algorithm for traffic jam noise signal

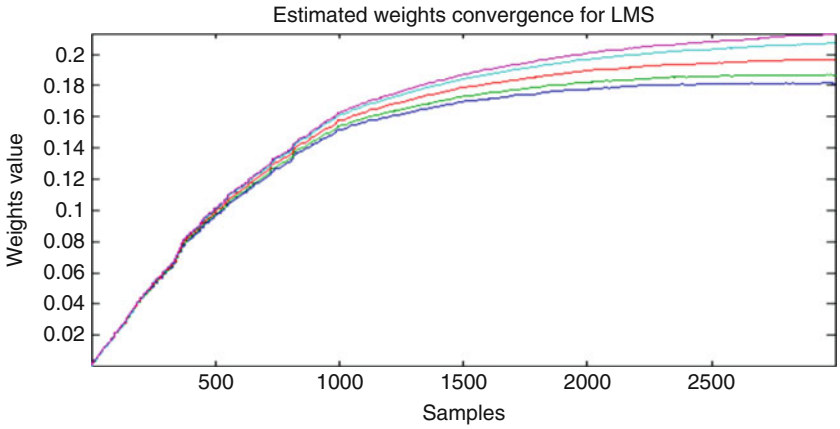

Fig. 5.27 Estimated convergence learning curve in LMS algorithm for traffic jam noise signal

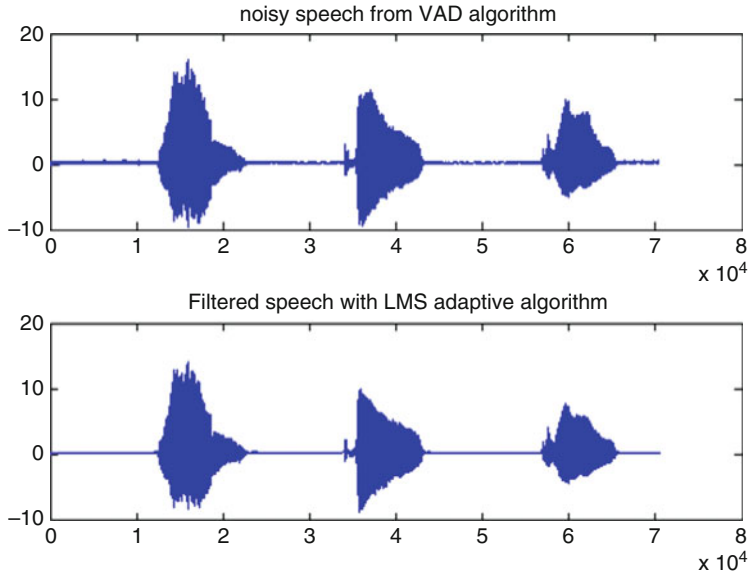

Fig. 5.28 Noisy speech signal with traffic jam noise signal from VAD algorithm and filtered speech signal with LMS algorithm

Subsequently, Fig. 5.27 shows the convergence curve for the variation of the coefficients. The nature of LMS again can be observed here. Rate convergence is somewhat less rapid, and after some time it will have steady-state values. Figure 5.28 shows nearly the same results as taken in the previous two cases of noise. In the mentioned simulation result, the first condition shows output of the VAD algorithm in that only some amount of noise is cleaned, only in the silence part. The next result, however, shows the output of LMS in which all the silences as well as the speech

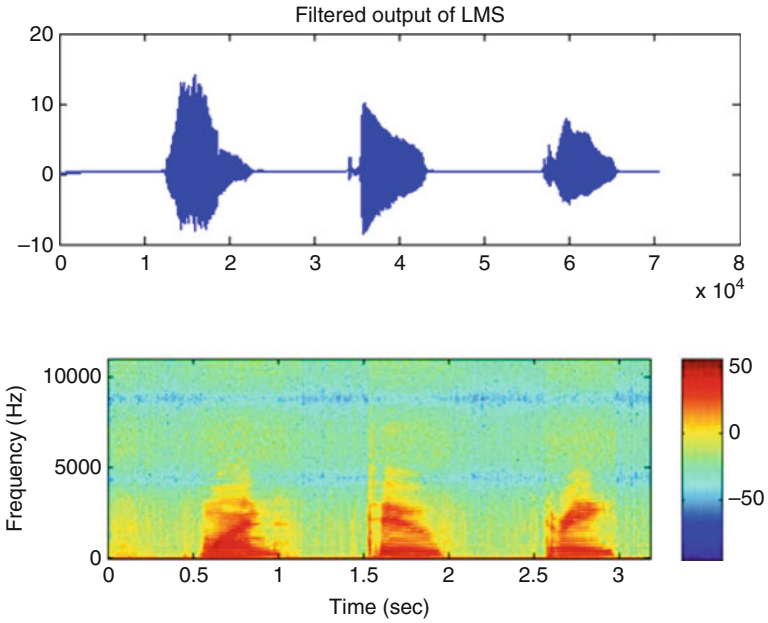

Fig. 5.29 Filtered speech signal by LMS algorithm for speech affect by traffic jam noise signal and its spectrogram

utterance parts are combining as cleaned. Figure 5.29 shows cleaned speech with its spectrogram in detail. Every noticeable frequency with a moderate amount of amplitude shown in the results. Next, filtered speech is given to the wavelet multi-resolution algorithm for each band evaluation, and after taking the wavelet approximation it can observed that a very large difference between original noisy speech and filtered speech is reflected in Fig. 5.30.

5.5 Speech Enhancement Process Based on the NLMS Algorithm

Basically the normalized LMS algorithm is nothing but a modified form of the standard LMS algorithm. By using normalized logic, the NLMS algorithm updates the coefficients of an adaptive filter. Compared to the LMS algorithm, the NLMS algorithm is a hypothetically faster converging algorithm. Faster convergence, however, comes at a price of greater residual error. The pure LMS algorithm is sensitive to the scaling of its input, which is its main drawback; this makes it difficult to choose a learning rate mu that assures the stability of the algorithm. The NLMS algorithm is basically a variant of the LMS algorithm that can be resolved by

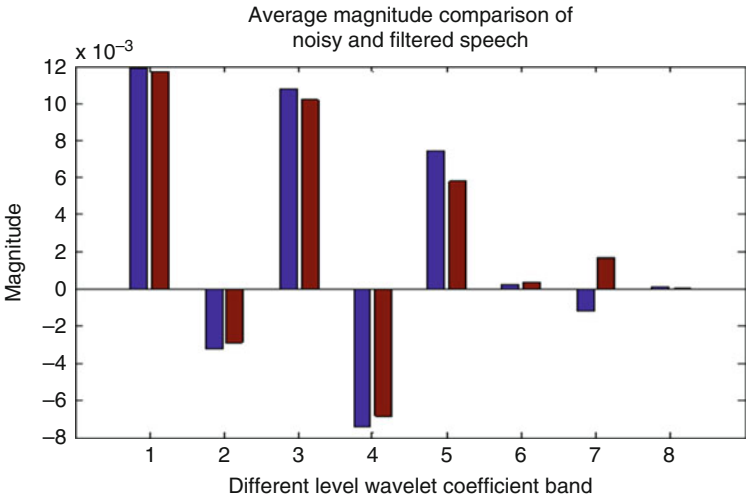

Fig. 5.30 Comparison of noisy speech signal with traffic jam noise signal and filtered speech signal in wavelet domain for LMS algorithm

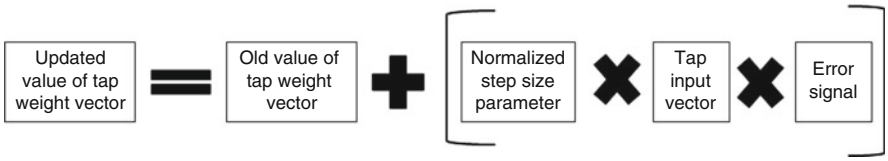

Fig. 5.31 NLMS algorithm in words

normalizing with power of the input. The normal functionality of the algorithm is as follows (Fig. 5.31).

Moreover, normally adjustment is proportional to the tap input vector $u(n)$, where $u(n)$ is large, the LMS filter having difficulties from a gradient noise amplification problem. For this reason, usually a normalized algorithm is useful. After this, the modification applied to the tap weight vector at iteration $n + 1$ is normalized with reference to the squared Euclidean norm of the tap input vector $u(n)$ at iteration n.

The NLMS algorithm flows in the same manner. Generally, the algorithm works on the output data of the VAD. Speech is cleaned through VAD in the silence part and that silence is cleaned and sent to the NLMS algorithm. As discussed previously, NLMS is working with normalized values of the step size parameters and accordingly the learning curve is set for a different noise file. The types of noises used here are the same as used in LMS algorithm with that magnitude only. In Fig. 5.32 is described one flow that shows the working steps for the NLMS algorithm for speech signal enhancement. The other waveforms shows different types of noise and its effect on the speech file during performance of the NLMS algorithm.

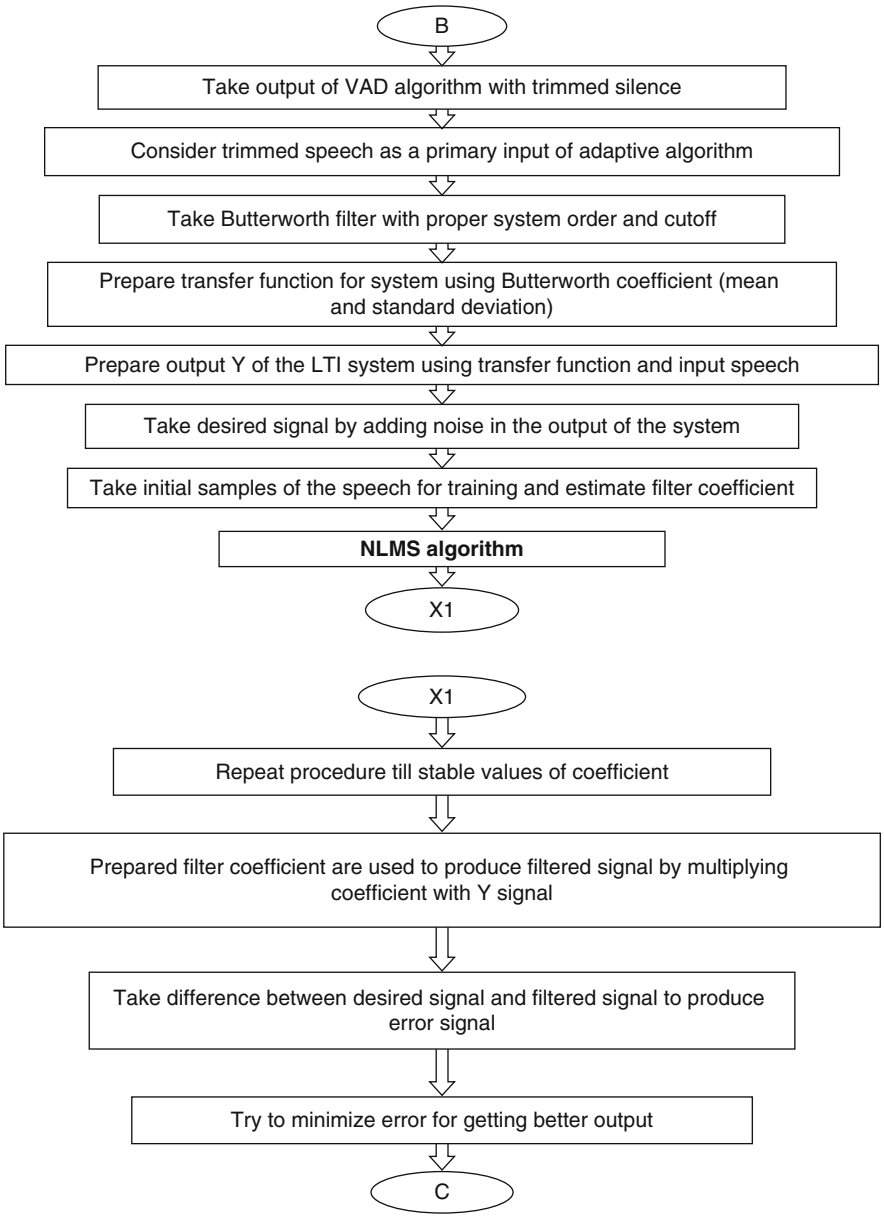

Fig. 5.32 Implementation process for speech signal enhancement using NLMS algorithm

5.5.1 *Results for White Noise Signal*

As discussed for the LMS adaptive algorithm, white noise is used. The value and type of white noise are the same as in the LMS. For simplicity, here only the direct performance of NLMS is shown, which can be taken as a reference. The main point to be observed here in the NLMS algorithm is that the rate of convergence is comparatively faster with reference to LMS but MSE is higher. Coefficients reach the steady-state level at a very fast rate compared to LMS, but to reach this level faster MSE cannot be taking a minimum value at the time of convergence. The LMS spectrogram of white noise is shown.

Now noisy speech is passed to VAD for silence searching; after searching, almost all the silence parts are threshold to zero value, which is why a much lesser amount of speech is present when utterances occur. Most of the silence part is giving zero value in the presence of thresholding. Then, speech with further silence removed is given to the NLMS adaptive algorithm. In the following simulation result, two algorithm plots can be seen. One waveform (with blue) shows the true value of the signal. By holding function, the estimated waveform is plotted on it (with red) (Fig. 5.33). The estimated signal is generated in the simulation by continuously updating the vector. Observation of the waveform tells us that estimated and true output have values that are changing in some matter and that the values provide noise reduction in the system.

The plot of Fig. 5.34 shows the difference between true and estimated output. The main task of the algorithm is to reduce this error so that noise can be reduced and clean speech clearly observed at the output. The number of filter weights is defined by system order. Here the stated system order is five. So, basically five initial filter weights can be selected by defining the Butterworth filter with order of two and cutoff frequency of 0.25 Hz. The output of the filter design is a numerator coefficient and denominator coefficients. From the coefficient, one system can be defined in the form of the transfer function. Then, by taking the inverse z transform of that transfer

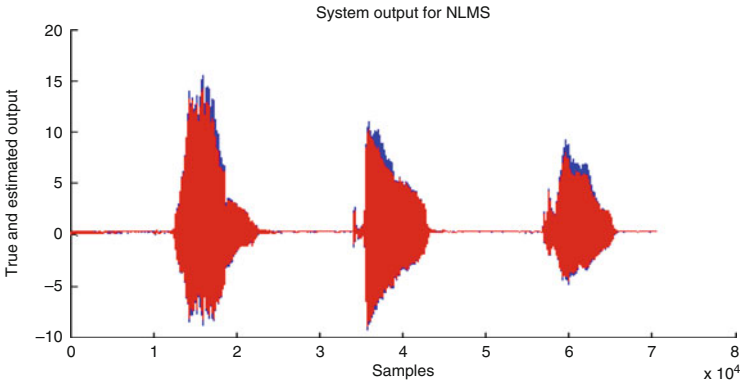

Fig. 5.33 True and estimated output in NLMS algorithm for white noise signal

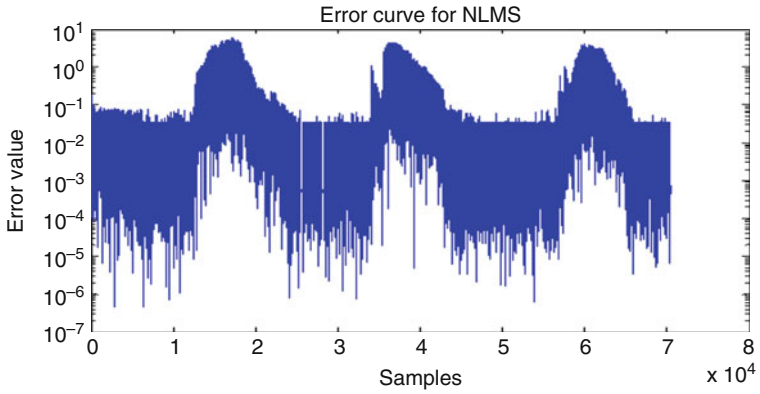

Fig. 5.34 Error between true and estimated output in NLMS algorithm for white noise signal

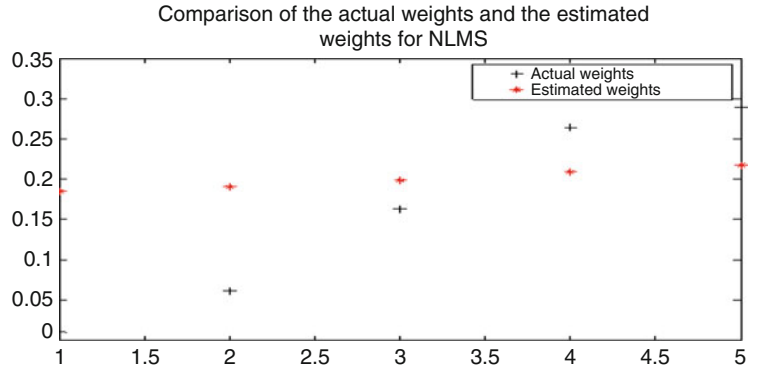

Fig. 5.35 Actual and true filter weight in NLMS algorithm for white noise signal

function, the initial values of the coefficient can be defined. That set of coefficients start to converge according to step size parameters. The NLMS algorithm continuously updates the value of weight per discussion about theory of NLMS in Chap. 3. For updating, 75 speech samples are taken and the training vector takes 3000 samples.

In Fig. 5.35 two different symbols are shown that give actual weights and estimated weight for the noise reduction after convergence of the algorithm at some steady-state level. Estimated coefficients are used for the whole file in the requirement of noise reduction. Figure 5.36 shows a learning curve for NLMS. A total of the first 3000 samples are taken for the proper convergence. A learning curve for the steady-state mechanism selected mu is normalized at each and every step; the variance vector of the input signal gives better performance in the systems.

Two patterns can be observed from Fig. 5.37. One pattern shows the output of the VAD algorithm in that only silences are detected and noise is reduced only in the silence period. After that the same waveform is given to the NLMS adaptive

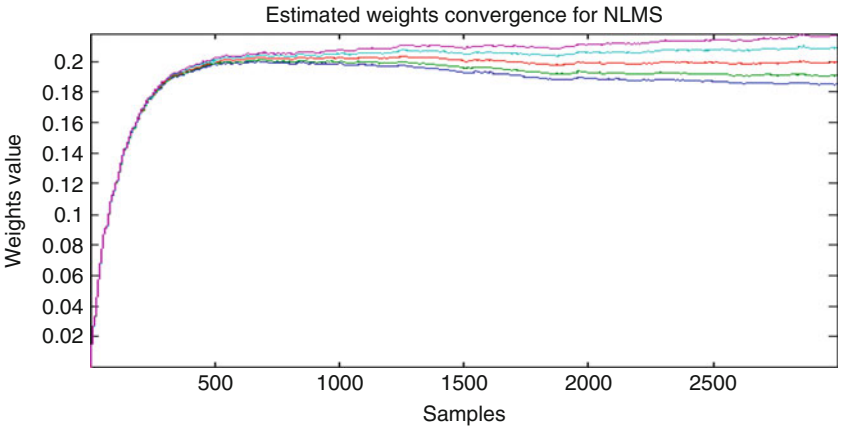

Fig. 5.36 Estimated convergence learning curve in NLMS algorithm for white noise signal

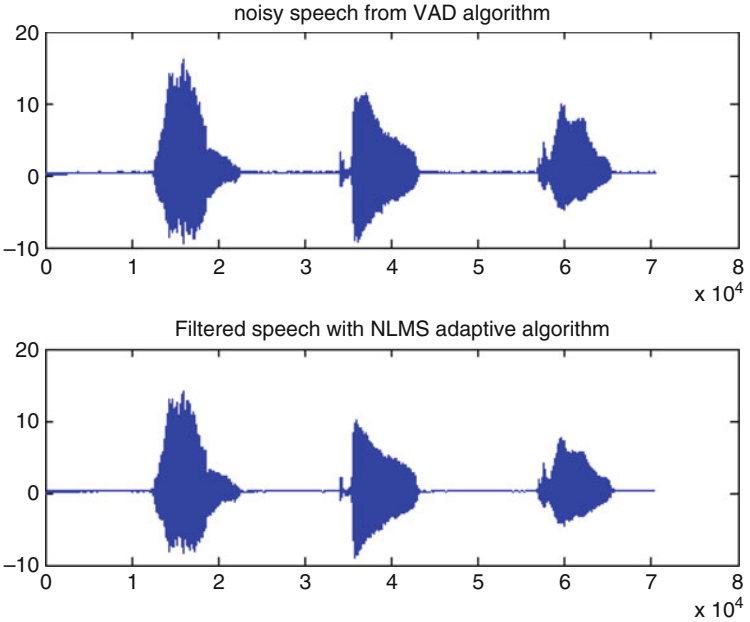

Fig. 5.37 Noisy speech signal with white noise signal from VAD algorithm and filtered speech signal with NLMS algorithm

algorithm. With the effect of the algorithm, the values can be observed. It is clearly seen that NLMS is able to efficiently cancel noise from speech as well as from silence parts but with higher MSE and a fast rate of convergence.

Figure 5.38 shows clear and cleaned speech as an output of the LMS adaptive algorithm. The spectrogram also can be seen there. Spectrogram is giving the correct

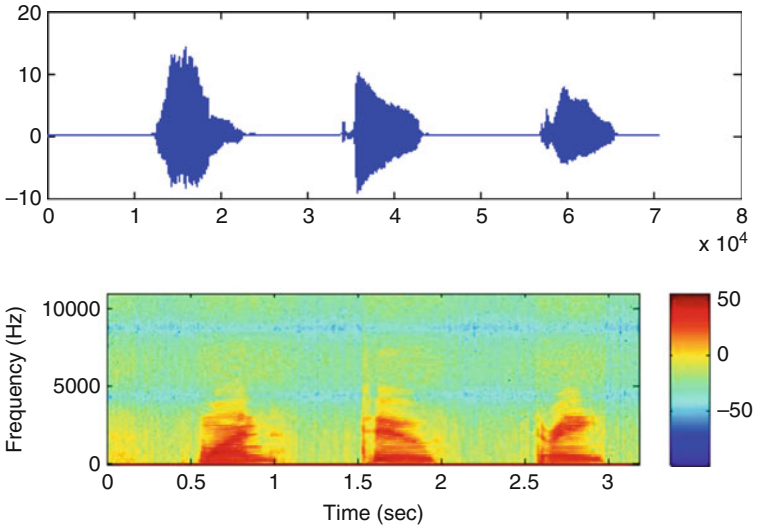

Fig. 5.38 Filtered speech signal by NLMS algorithm for speech affected by white noise signal and its spectrogram

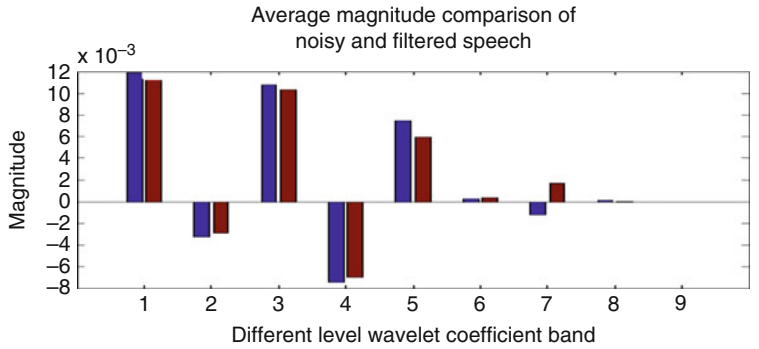

Fig. 5.39 Comparison of noisy speech signal with white noise signal and filtered speech signal in wavelet domain for NLMS algorithm

value compared to the previous noisy spectrogram for the signal. Then the wavelet tool is used for actual comparison in the frequency domain.

The output of VAD is trimmed speech that is given to the multi-resolution wavelet algorithm. From that individual band the coefficient can be derived. The same multi-resolution wavelet algorithm is also given NLMS filtered speech. The comparison can be seen in Fig. 5.39. Each and every band is less compared to the first band; thus it can be concluded that noise can be reduced in a very efficient way by NLMS.

5.5.2 *Results for Babble Noise Signal*

As discussed previously, multi-talker babble noise is very similar to speech. It can be observed that the maximum amount of noise can be detected in the silence times. During utterance, noise is additive. Now noisy speech is given to the adaptive algorithm. Per the logic of the NLMS algorithm, it works on the principle that it requires two inputs simultaneously: one is known as a reference input and the second input is the desired input. In Fig. 5.40 these two inputs are shown with two differently colored waveforms. The speech waveform plot (with blue color) is known as a true input of the algorithm whereas the speech signal (with red) is known as an estimated input. It can be observed that there is always a difference between true and estimated input. The estimated waveform is not completely superimposed on the actual waveform. Because of that, some sort of error is present in the signal. An error curve can be seen in Fig. 5.41. Here, the sample shows error

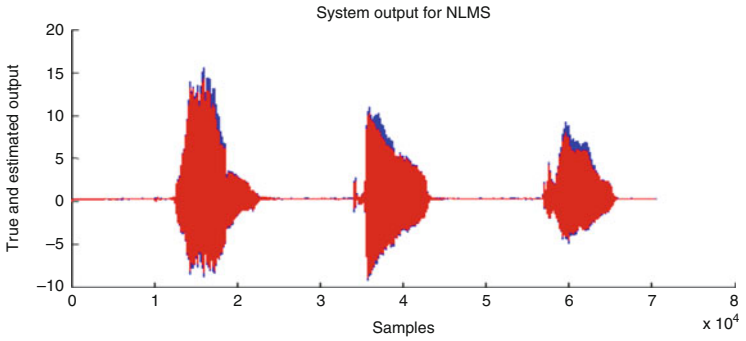

Fig. 5.40 True and estimated output in NLMS algorithm for babble noise signal

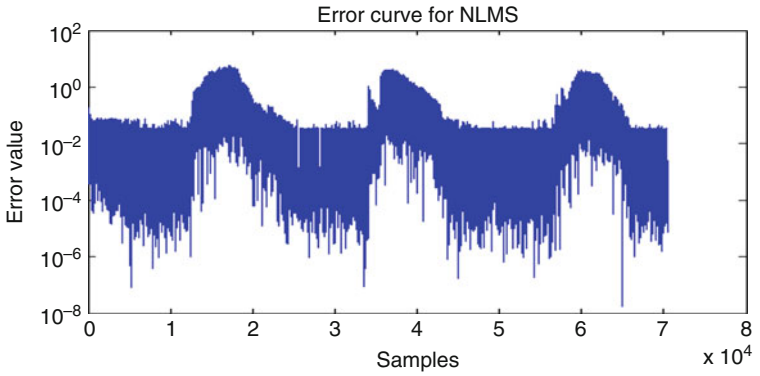

Fig. 5.41 Error between true and estimated output in NLMS algorithm for babble noise signal

by the sample difference for the training vector. Every time, the difference is plotted there. An error curve is described for the total set of speech input.

Subsequently, Fig. 5.42 shows an important result. As discussed initially, to start the adaptation some reference level is required. By designing a Butterworth filter with suitable order, the updatation has been started in NLMS. Initial filter values are plotted in Fig. 5.42 as an actual weight. The training vector for weight adjustment is 3000 samples. In trying for reduction in MSE and by using the appropriate step size for a given training vector, new derived coefficients are marked with red. Now the derived value of the coefficient is finalized for filtering of the total speech file. By the nature of the training vector samples these values can be identified as suitable for the total filtering operation of the speech signal. Variation in true and estimated values reflects the deviation required for noise reduction and enhancement in the weight values for reduction of noise in the speech file.

Figure 5.43 shows an estimated convergence learning curve in NLMS for babble noise. In the layout five different color plots are shown. As the initial fifth order, a

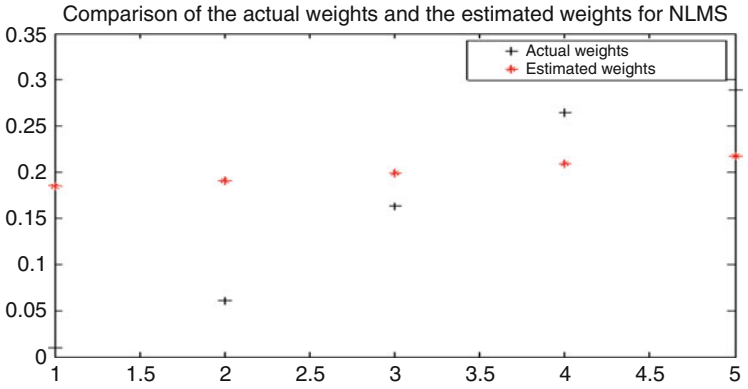

Fig. 5.42 Actual and true filter weight in NLMS algorithm for babble noise signal

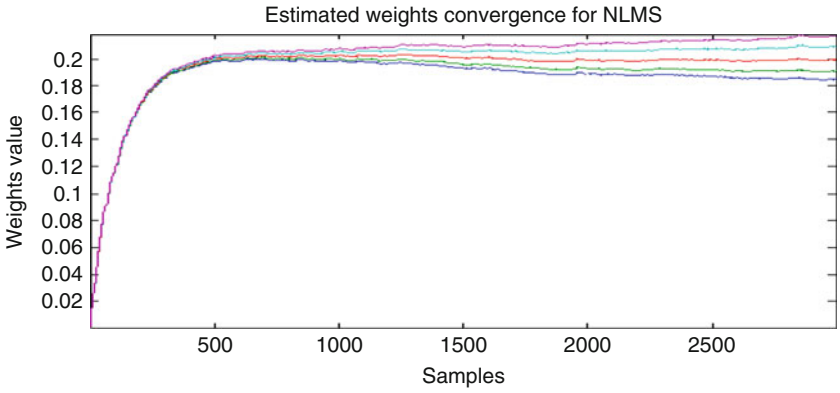

Fig. 5.43 Estimated convergence learning curve in NLMS algorithm for babble noise signal

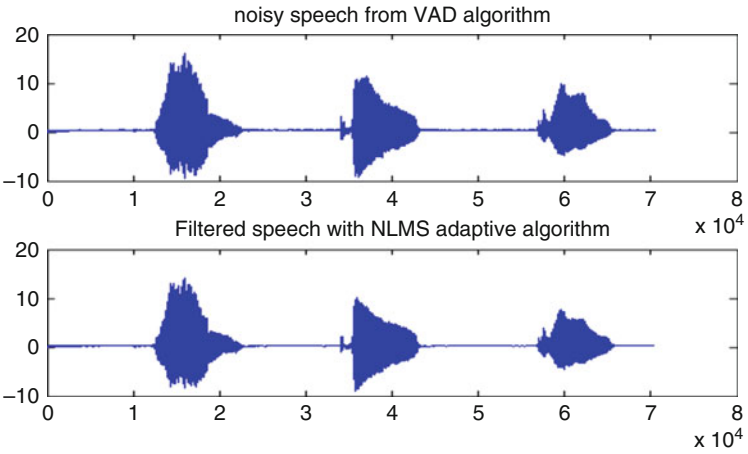

Fig. 5.44 Noisy speech signal with babble noise signal from VAD algorithm and filtered speech signal with NLMS algorithm

Butterworth filter with cutoff frequency of 0.25 Hz is considered. In adaptation starting with the initial values of filter weight modification of the filter weight is necessary. That enhancement is a gradual process. Gradually all the filter weights are set with a training vector value and after some time that will take a steady-state value. According to selected values of the mu sample set, convergence takes place. Per the characteristics of NLMS it can be observed from Fig. 5.63 that the convergence rate is gradual. Initially all the five curves move in line. No variation can be detected. Progress in the number of samples shows improvement in variations. By time with some gradual variation the convergence curve becomes stable. By the end of the convergence curve the number of coefficients acquires its final values. It can be observed clearly that compared to LMS, NLMS learning curves acquire the steady-state value at a very fast rate. It can be observed from the simulation result that initially curves merged and then progress with constant slop in the straight direction, so here the speed is greater. Now a new filter vector is prepared according to input of incoming noisy speech characteristics. Using the same set of filters, the noise removal operation can be achieved.

Figure 5.44 shows two waveforms, one for noisy output and the other for filtered output. The first layout in the simulation results shows noise reduction in the silence part as an output of VAD algorithm. After application of VAD, speech parts also remain, as with noise. Noisy speech after applying NLMS algorithm looks similar original speech. Most of the noise values are removed from the silence as well as the utterance part efficiently. Figure 5.45 indicates noise-removed speech and its spectrogram. It can be observed in the spectrogram that it is very carefully identified that modified speech is very similar to the original speech in nature. The spectrogram shows three axis representations that include time, frequency, and color depth, which shows the intensity of that enhanced frequency.

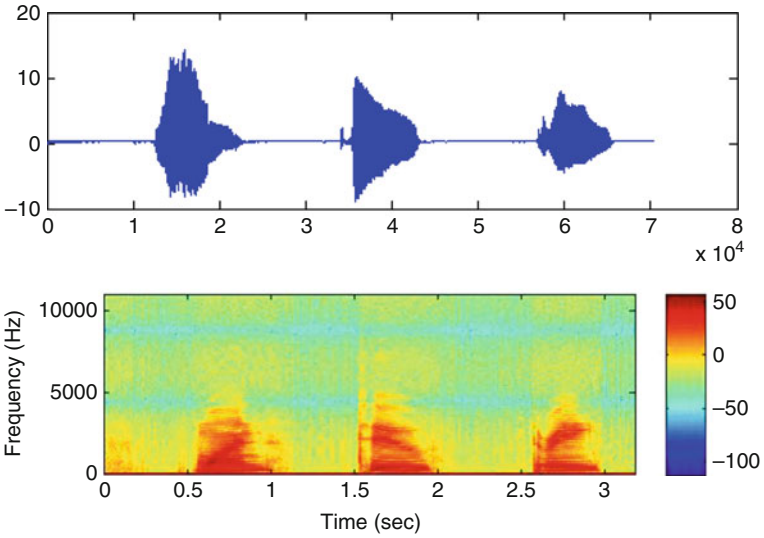

Fig. 5.45 Filtered speech signal by NLMS algorithm for speech affected by babble noise signal and its spectrogram

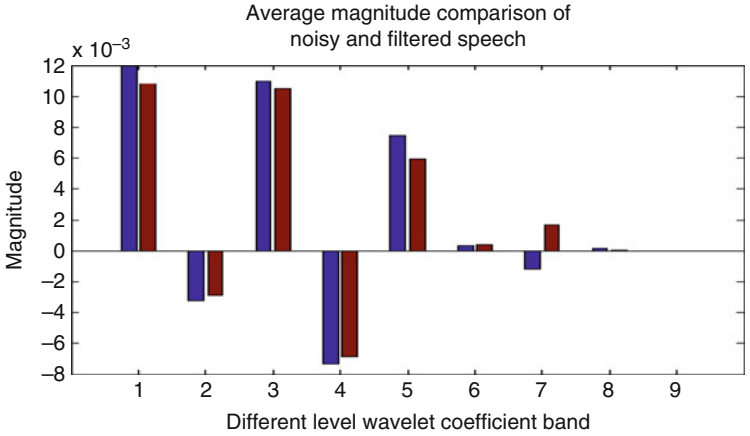

Fig. 5.46 Comparison of noisy speech signal with babble noise signal and filtered speech signal in wavelet domain for NLMS algorithm

The measured output of the system can be plotted as in Fig. 5.46, indicating the amount of variation in original speech and in the NLMS filtered enhanced speech. Each speech file is given to the wavelet multi-resolution application for each band variation. The simulation result shows band layout for each frequency present in the speech. Before noise reduction the magnitude of confinement is higher and after noise reduction it can be observed that the magnitude is reduced.

5.5.3 *Results for Traffic Jam Noise Signal*

In normal life when conversation is going on, surrounding traffic is generally present and creating different sounds. Because of that whatever unwanted noise is generated with speech it is known as traffic noise. From the spectrogram of the traffic signal it can be observed that high to low frequencies are present with different magnitudes. That type of continuous noise is scattered throughout the range of the frequency. Spectrograms of noise and noisy speech are shown in the previous section.

Figure 5.47 shows, as discussed previously, true and estimated output. Other matters are as described in the case of babble noise. Simply, the true and estimated output of the system as required by the NLMS to execute is shown. Similarly, as described earlier, Fig. 5.48 shows the sample by sample difference in the presence of true and estimated output in the case of traffic jam noise. Figure 5.49 shows actual and estimated weight coefficients for traffic jam noise for proper operation. Deviation between actual and estimated coefficients shows the amount of modification require for noise reduction.

Subsequently, Fig. 5.50 shows a convergence curve for the variation of the coefficients. The nature of NLMS again can be observed here. The rate of

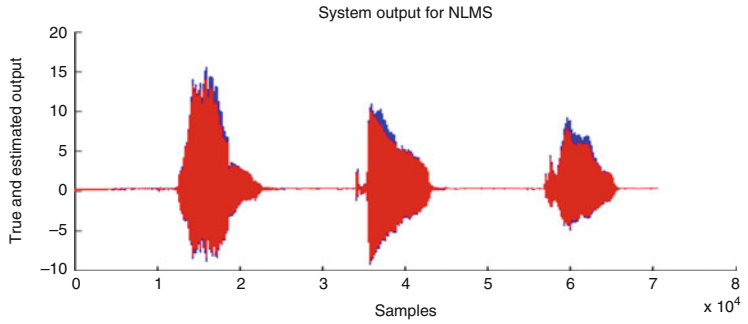

Fig. 5.47 True and estimated output in NLMS algorithm for traffic jam noise signal

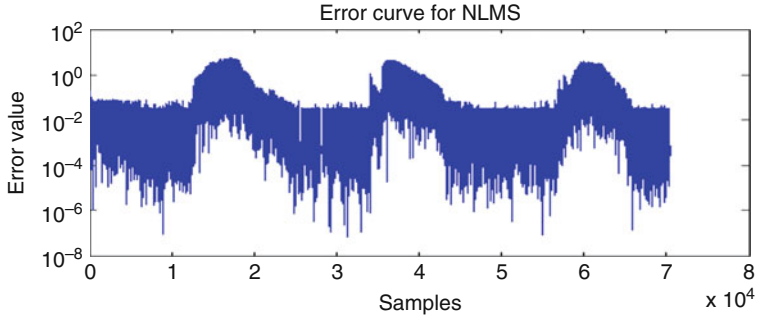

Fig. 5.48 Error between true and estimated output in NLMS algorithm for traffic jam noise signal

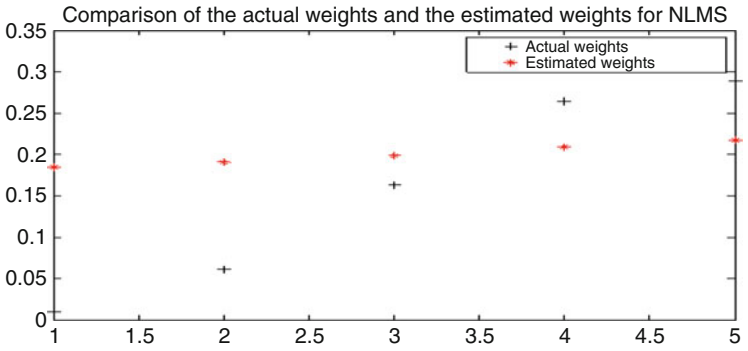

Fig. 5.49 Actual and true filter weight in NLMS algorithm for traffic jam noise signal

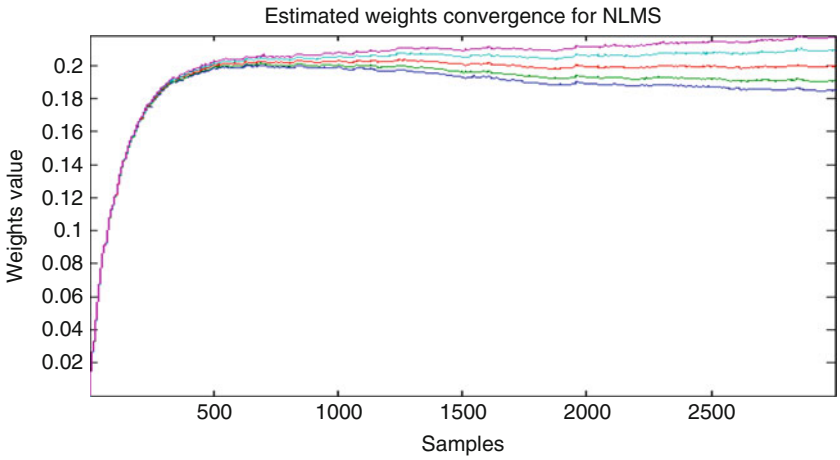

Fig. 5.50 Estimated convergence learning curve in NLMS algorithm for traffic jam noise signal

convergence is somewhat more rapid and after some time will reach steady-state values. Figure 5.51 shows almost the same results as in the previous two cases of noise.

In the mentioned simulation result the first condition shows the output of the VAD algorithm in that only some amount of noise is cleaned, only in the silence part. However, the next result shows the output of NLMS in which all the silences as well as the speech utterance parts are combined and cleaned in Fig. 5.51. Figure 5.52 shows cleaned speech with its spectrogram in detail. Every noticeable frequency with a moderate amount of amplitude is shown. Next, filtered speech is given to the wavelet multi-resolution algorithm for each band evaluation, and after taking the wavelet approximation a very large difference between original noisy speech and filtered speech for traffic jam noise is shown in Fig. 5.53.

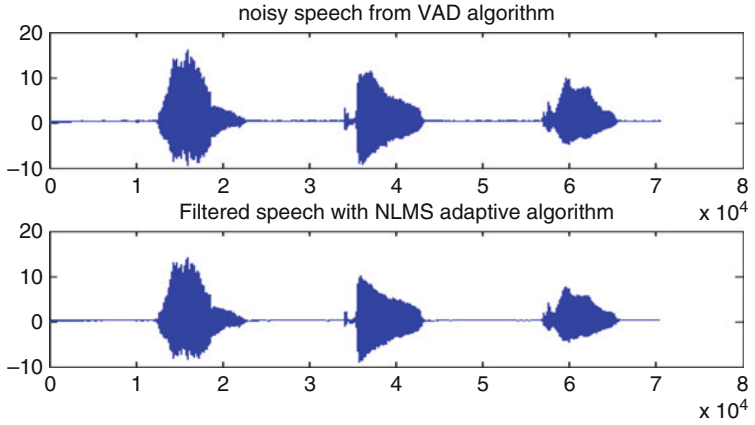

Fig. 5.51 Noisy speech signal with traffic jam noise signal from VAD algorithm and filtered speech signal with NLMS algorithm

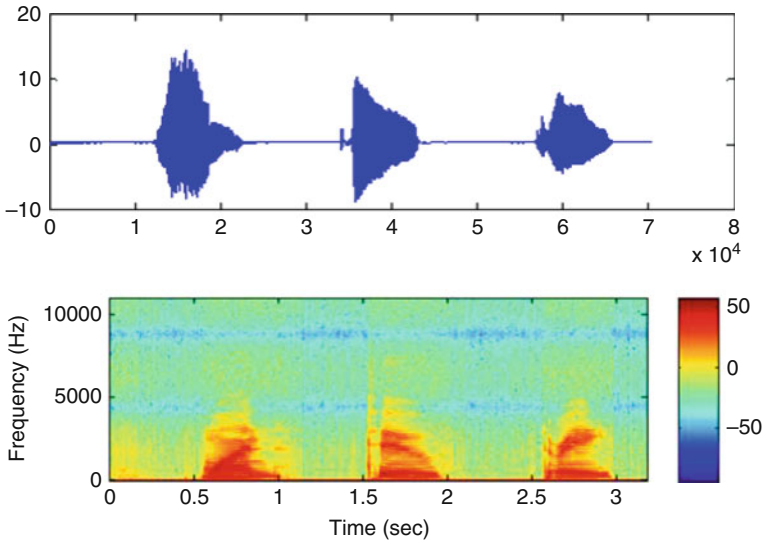

Fig. 5.52 Filtered speech signal by NLMS algorithm for speech affect by traffic jam noise signal and its spectrogram

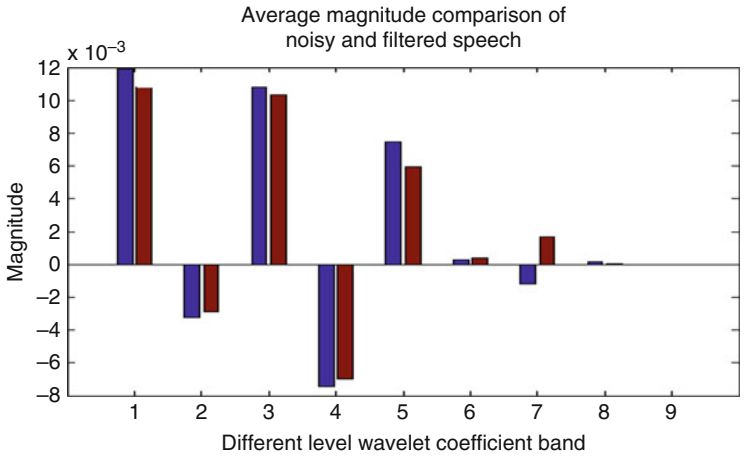

Fig. 5.53 Comparison of noisy speech signal with traffic jam noise signal and filtered speech signal in wavelet domain for NLMS algorithm

5.6 Speech Enhancement Process Based on the RLS Algorithm

The following operations are performed by the standard RLS algorithm to update the coefficients of an adaptive filter.

- First, carry out the calculation of output signal of the adaptive filter.
- Calculate the error signal using Equation 5.1.Update the filter coefficients by using Equation 5.1.

$$\widehat{w}(n) = \widehat{w}(n-1) + k(n)\xi^*(n) \tag{5.1}$$

where $\hat{w}(n)$ is the filter coefficients vector and $k(n)$ is the gain vector. The $k(n)$ is defined by the following equation.

$$k(n) = \frac{\Pi(n)}{\lambda + u^H(n)\Pi(n)} \tag{5.2}$$

where λ is the forgetting factor and $P(n)$ is the inverse correlation matrix of the input.

This algorithm evaluates not only the instantaneous value $e^2(n)$ but also the past values, such as $e^2(n-1)$, $e^2(n-2)\ldots e^2(n-N+1)$. The value range of the forgetting factor is (0, 1]. When the forgetting factor is <1, it specifies that this algorithm places a higher weight on the current value and a lower weight on the past values. The resulting $E[e^2(n)]$ of the RLS algorithms is more accurate than that of the LMS algorithms. The RLS algorithm is a special case of Kalman filter that provides

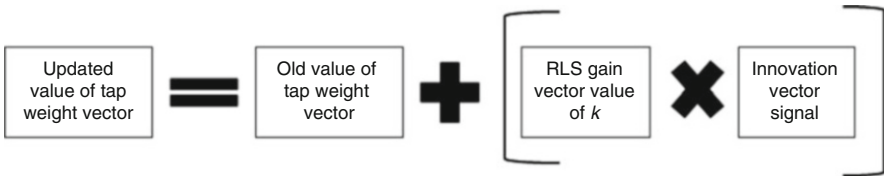

Fig. 5.54 RLS algorithm in words

an amount of all the inputs applicable to the filter up to a specific instant of time. The RLS algorithm behaves in the following manner: the function of the RLS algorithm in words is shown in Fig. 5.54.

Here, the innovation vector signifies the new information that can be put into the filtering process at the time of the calculation. In the following algorithm, the methodology of RLS utilization for enhancement of a noisy speech signal is discussed. The standard RLS algorithm assumes that the use of a transversal filter is as the structural basis of the linear adaptive filter. The derivation of the standard RLS algorithm relies on a basic result in linear algebra known as the matrix inversion lemma. The algorithm includes the same virtues and suffers from the some limitations related to lack of numerical robustness and excessive computational complexity. Performance measurement of the RLS algorithm for the speech is checked by using a different number of noises as discussed in the previous two algorithms. Different noises such as white noise, babble noise, and traffic jam noise are taken as a reference and for that algorithm measurements are removed. The response of RLS for different noises is as follows. The flow of implementation is discussed in Fig. 5.55.

5.6.1 Results for White Noise Signal

The main point to be observed here in the case of RLS is that the rate of convergence is faster compared to LMS and NLMS. The coefficient can reach the steady-state level at a very fast rate compared to NLMS but to reach the faster MSE cannot take the minimum value at the time of convergence. The LMS spectrogram of white noise is shown.

Now, noisy speech is given to VAD for silence searching. After searching silences almost all the silence parts are threshold to zero value so that a much lesser amount of speech is present when utterances are there. Most of the silence part is given zero value in presence of thresholding. Then, further silence-removed speech is given to the RLS adaptive algorithm. In the following simulation result per the concept of algorithm, two plots can be seen. One (with blue) shows the true value of the signal. By holding function the estimated waveform is plotted on it (red color), as shown in Fig. 5.56. The estimated signal is generated in simulation by updating the vector continuously. Observation of the waveform tells us that estimated and true output is changing values in some matter and that the values provide noise reduction in the system.

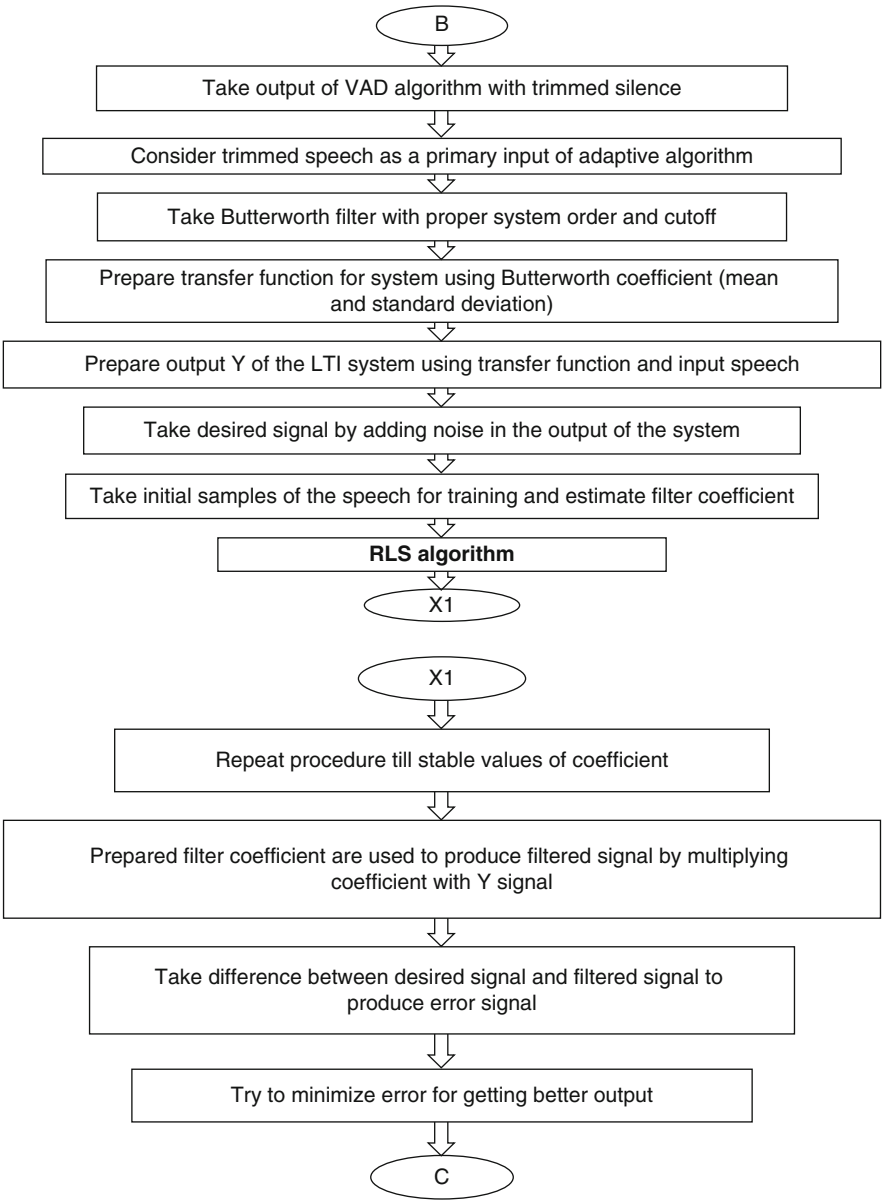

Fig. 5.55 Implementation process for speech signal enhancement using RLS algorithm

The plot of Fig. 5.57 shows the difference between true and estimated output. The main task of the algorithm is to reduce this error so that noise can be reduced and clean speech can be clearly observed in the output.

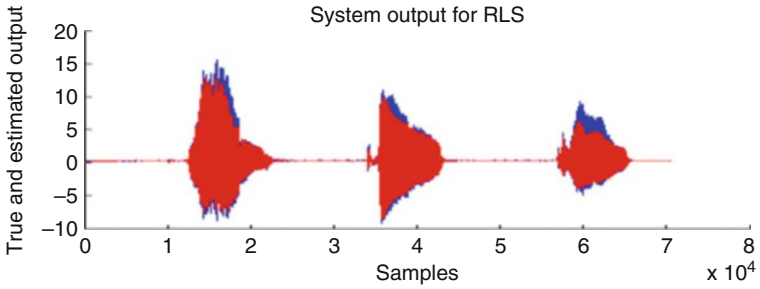

Fig. 5.56 True and estimated output in RLS algorithm for white noise signal

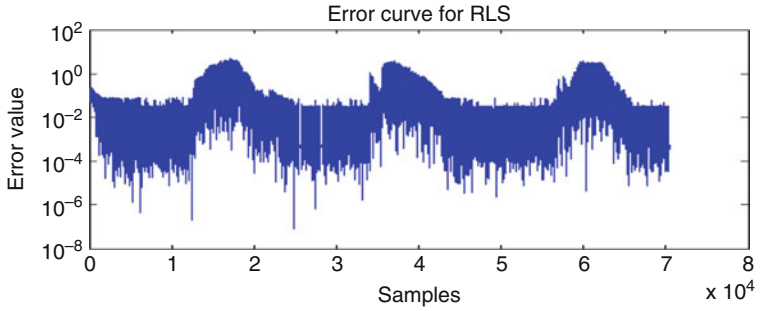

Fig. 5.57 Error between true and estimated output in RLS algorithm for white noise signal

The number of filter weights is defined by system order. Here the stated system order is five. So, basically five initial filter weights can be selected by defining the Butterworth filter with order two and cutoff frequency 0.25 Hz. Output of filter design is numerator coefficient and denominator coefficients. From taking the coefficient one system can be defined in the form of the transfer function. Then by taking the inverse z transform of that transfer function, initial values of the coefficient can be defined. That set of coefficients starts to converge by computing the priori estimation error. The adaptive operation of the algorithm starts with the tap weight vector, which is generally updated by increasing its old value by a quantity equal to the product of the complex conjugate of the priori estimation error and the time-varying gain vector. The RLS algorithm continuously updates the value of weight per discussion about theory of RLS in Chap. 3. For updating, 75 speech samples are taken and the training vector is 3000. In Fig. 5.58, two different symbols are shown that give actual weights and estimated weight for noise reduction after convergence of the algorithm at some steady-state level. Estimated coefficients are used for the whole file in the requirement of noise reduction.

Figure 5.59 shows a learning curve for RLS. The first 3000 samples are taken for the proper convergence. It can be observed that the rate of convergence is typically faster than that of the simple LMS algorithm filter because the RLS filter whitens the input data by using the inverse correlation matrix of the data with zero mean.

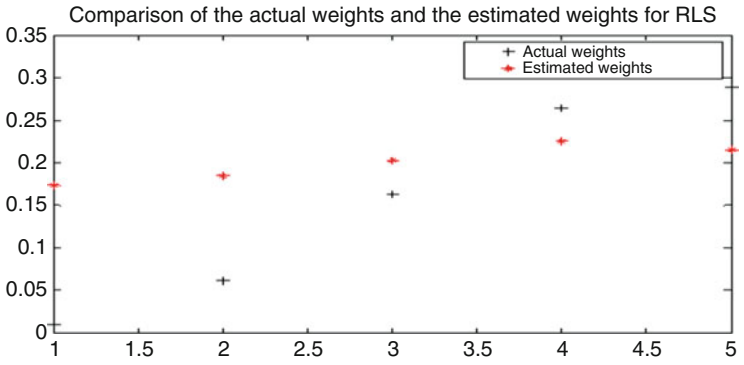

Fig. 5.58 Actual and true filter weight in RLS algorithm for white noise signal

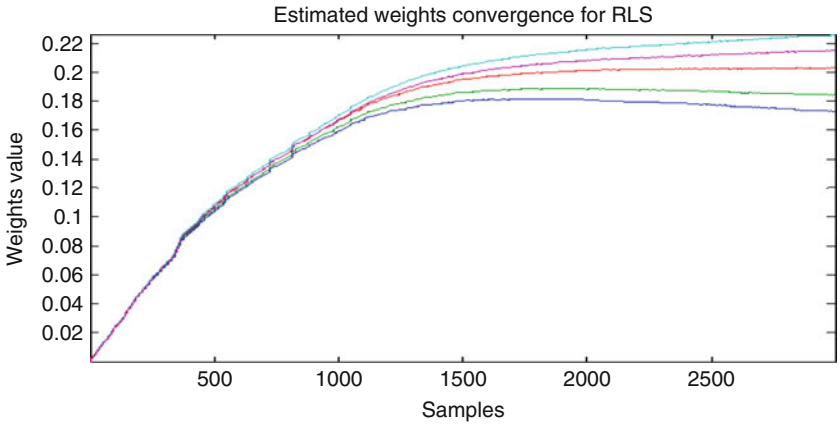

Fig. 5.59 Estimated convergence learning curve in RLS algorithm for white noise signal

It can be observed from Fig. 5.60 that there are two patterns. One pattern shows output of VAD algorithm in that only silences are detected and noise is reduced only in the silence period. After that the same waveform is applied to the RLS adaptive algorithm. With the effect of the algorithm the values can be observed. It can be clearly seen that RLS is able to efficiently cancel noise from speech as well as from silence parts but with higher MSE and fast rate of convergence, but for a given application the performance is less.

Figure 5.61 shows clear and cleaned speech as an output of the RLS adaptive algorithm. Also, the spectrogram can be seen. The spectrogram is giving the correct value compared to the previous noisy spectrogram for the signal. Then, the wavelet tool is used to obtain the actual comparison in the frequency domain.

The output of VAD is trimmed speech, which is given to the multi-resolution wavelet algorithm. From that individual band the coefficient can be derived. The same multi-resolution wavelet algorithm is given also to RLS filtered speech. The

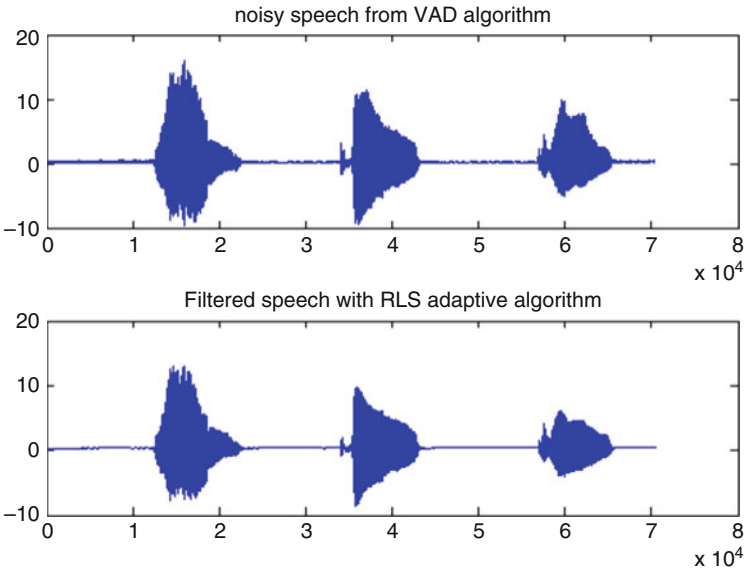

Fig. 5.60 Noisy speech signal with white noise signal from VAD algorithm and filtered speech signal with RLS algorithm

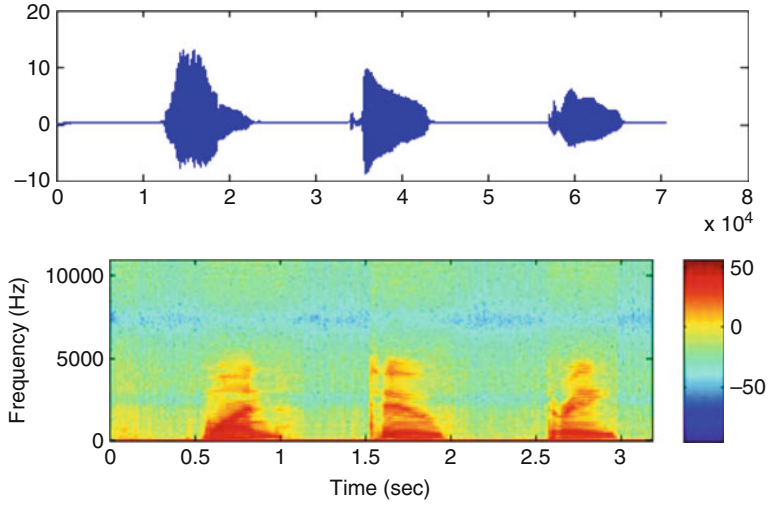

Fig. 5.61 Filtered speech signal by RLS algorithm for speech affect by white noise signal and its spectrogram

comparison is seen in Fig. 5.62. Each and every band is less compared to the first band, so it can be concluded that noise can be reduced in a very efficient way by RLS.

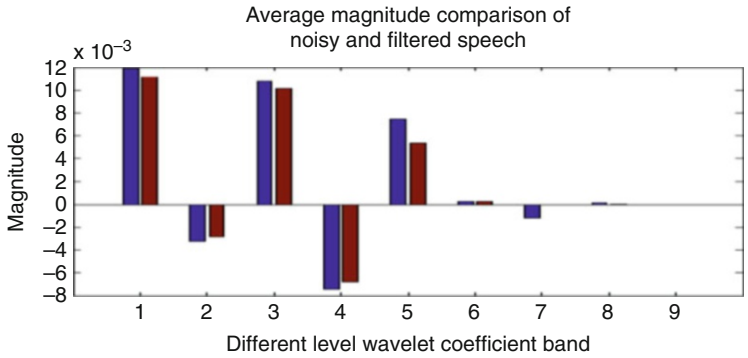

Fig. 5.62 Comparison of noisy speech signal with white noise signal and filtered speech signal in wavelet domain for RLS algorithm

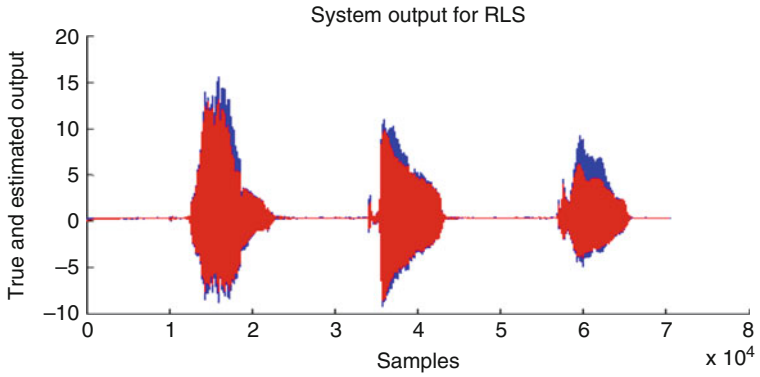

Fig. 5.63 True and estimated output in RLS algorithm for babble noise signal

5.6.2 Results for Babble Noise Signal

As discussed previously, babble noise from multiple talkers is very similar to speech. It can be observed that the maximum amount of noise can be detected in the silence time. During the utterance noise it is in additive manner. Now noisy speech is given to the adaptive algorithm. Per the logic of the RLS algorithm it works on the principle that it requires two inputs simultaneously: a reference input and the desired input. These two inputs are shown with two different color waveforms in Fig. 5.63. The speech waveform plot with blue color is known as a true input of the algorithm whereas the speech signal with red is known as an estimated input. It can be observed that always there is a difference between true and estimated input. The estimated waveform is not completely superimposed on the actual waveform. Thus, some sort of error is present in the signal. The error curve can be seen in Fig. 5.64. Here, error

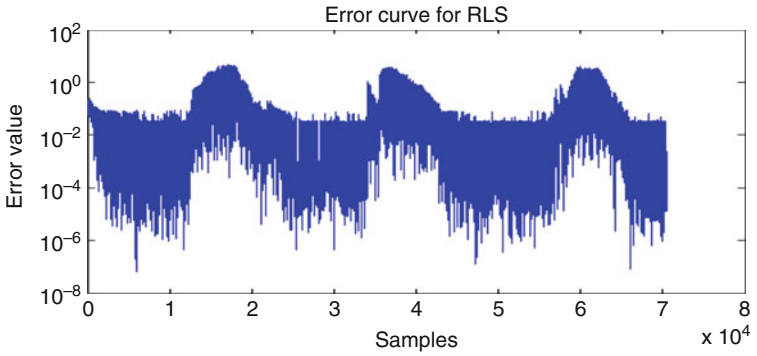

Fig. 5.64 Error between true and estimated output in RLS algorithm for babble noise signal

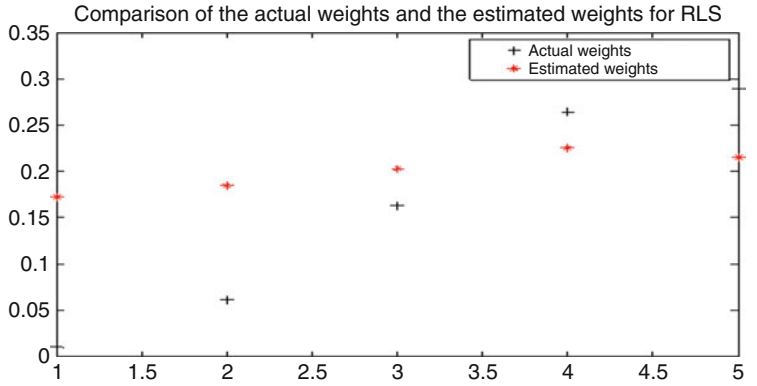

Fig. 5.65 Actual and true filter weight in RLS algorithm for babble noise signal

shows sample-by-sample difference for the training vector. The difference is that each time the error curve is stated for the total set of speech input.

Subsequently, Fig. 5.65 shows an important result. As discussed initially, some reference level is required for starting the adaptation. By designing a Butterworth filter with suitable order, the updatation has been started in RLS. Initial filter values are plotted in Fig. 5.65 as an actual weight. The training vector for weight adjustment is 3000 samples. In trying for reduction in the MSE and by taking an appropriate value of lambda, phi, and vector gain for a given training vector, new derived coefficients are marked with red. Now, derived values of the coefficient are finalized for the filtering of the total speech file. By the nature of the training vector samples these values can be identified as suitable for the total filtering operation of the speech signal. Variation in true and estimated values reflects the deviation required for noise reduction and enhancement in the weight values for reduction of noise in the speech file.

Figure 5.66 shows an estimated convergence learning curve in RLS for babble noise. In the layout five different color plots are shown. Initially, the fifth order of the

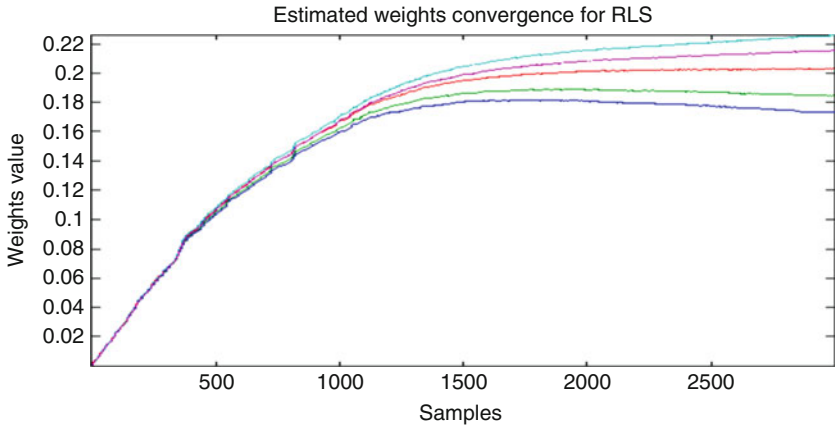

Fig. 5.66 Estimated convergence learning curve in RLS algorithm for babble noise signal

Butterworth filter with a cutoff frequency of 0.25 Hz is considered. In adaptation starting with initial values of filter weight, it is necessary to modify filter weight. That enhancement is a gradual process. Gradually all the filter weights are set with the training vector value and with that after some time it will take a steady-state value. According to selected values of sample set wise convergence takes place. Per the characteristics of RLS it can be observed from Fig. 5.66 that the convergence rate is gradual. Initially all the five curves move in line: no variation can be detected. Progress in the number of samples shows improvement in variations. As applied here, logic in the RLS convergence speech is very high compared to a simple LMS. In time with some gradual variations the convergence curve achieves stability. By the end of convergence the curve number of the coefficients reaches its final values. It can be observed clearly that compared to LMS, in the case of RLS learning the curves reach a steady-state value at a very fast rate. It can be observed from the simulation result that initially curves merge and then progress with constant slop in the straight direction so here the speed is greater. Now a new filter vector is prepared according to the input of incoming noisy speech characteristics. Using the same set of filters, the noise removal operation can be achieved.

Figure 5.67 shows two waveforms, one for noisy output and the other for filtered output. The first layout in the simulation results shows noise reduction in the silence part as an output of the VAD algorithm. After application of VAD, speech parts remain as with noise. Noisy speech after applying the RLS algorithm looks similar to original speech. Most of the noise values are removed from the silence as well as the utterance parts of speech efficiently.

Figure 5.68 portrays noise-removed speech and its spectrogram. In the spectrogram, it is very precisely shown that modified speech is very similar to the original speech in nature. The spectrogram shows three axes, including time and frequency and the depth of color shows the intensity of that enhanced frequency. Measured output of the system is plotted in Fig. 5.69, which indicates the amount of variation

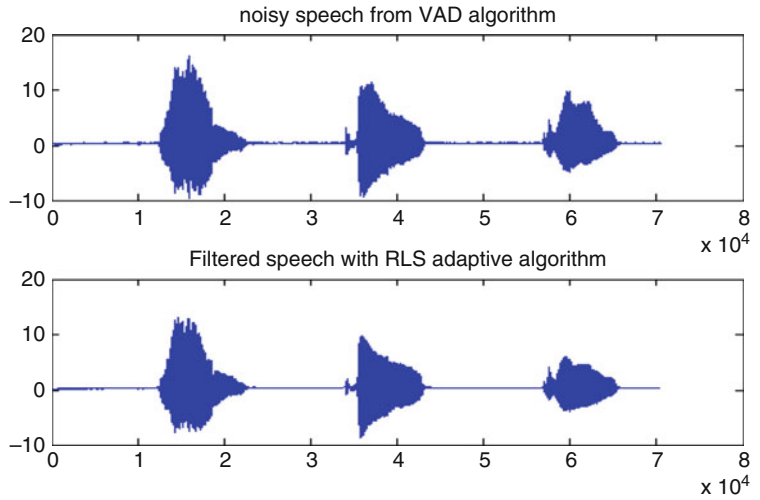

Fig. 5.67 Noisy speech signal with babble noise signal from VAD algorithm and filtered speech signal with RLS algorithm

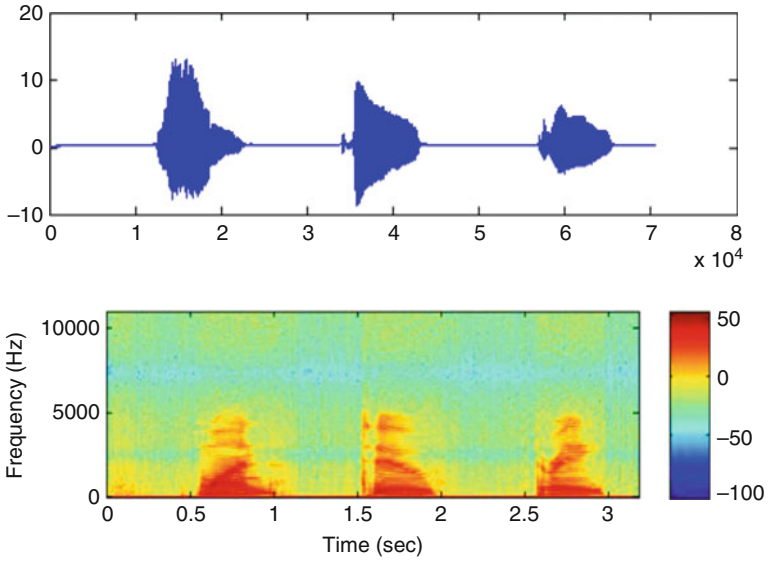

Fig. 5.68 Filtered speech signal by RLS algorithm for speech affect by babble noise signal and its spectrogram

in the original speech and RLS-filtered enhanced speech. Each speech file is given to wavelet multi-resolution application for each band variation. The simulation result shows a band layout for each frequency present in the speech. Before noise reduction, magnitude is higher in confinement, and after noise reduction it can be observed that magnitude is reduced.

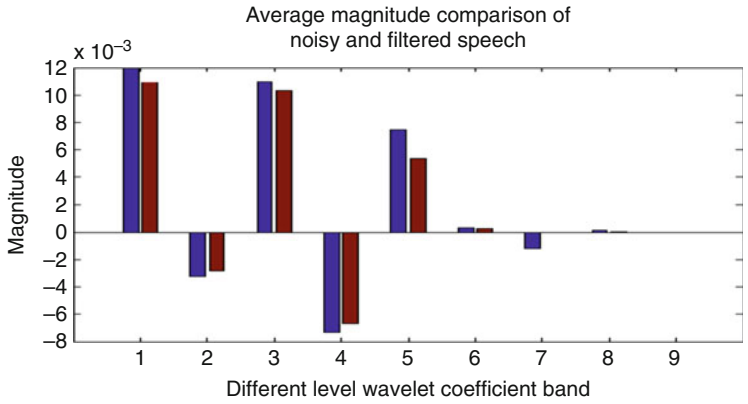

Fig. 5.69 Comparison of noisy speech signal with babble noise signal and filtered speech signal in wavelet domain for RLS algorithm

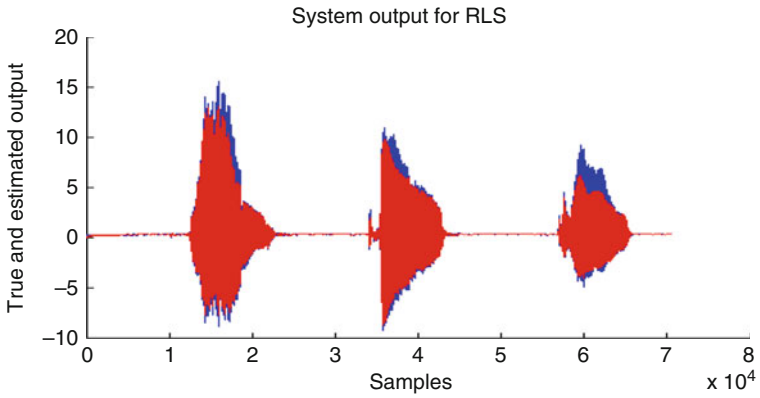

Fig. 5.70 True and estimated output in RLS algorithm for traffic jam noise signal

5.6.3 Results for Traffic Jam Noise Signal

In normal life, generally when people are talking there is often surrounding traffic is present that creates different sounds. Thus, whatever unwanted noise is generated with speech is known as traffic noise. From the spectrogram of a traffic signal it can be observed that in that high to low frequencies are present with different magnitudes. That type of continuous noise is scattered on throughout the range of the frequency. Spectrograms of noise and noisy speech are shown in the previous section.

Figure 5.70 shows, as discussed previously, true and estimated output. Other matters are as described in the case of babble noise. Simply, it shows the true and estimated output of the system as required by the RLS to execute. Similarly, as described earlier, Fig. 5.71 shows sample by sample difference in the presence of

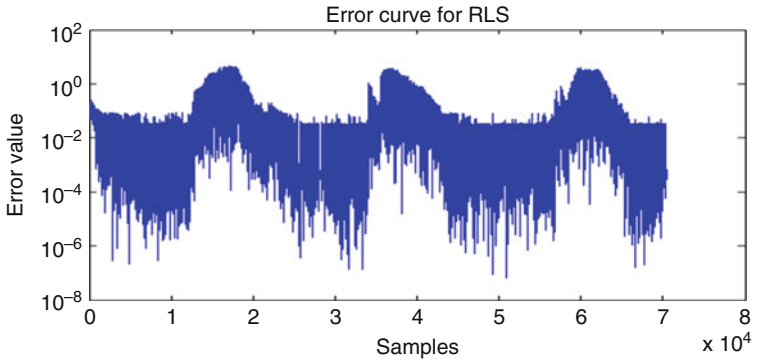

Fig. 5.71 Error between true and estimated output in RLS algorithm for traffic jam noise signal

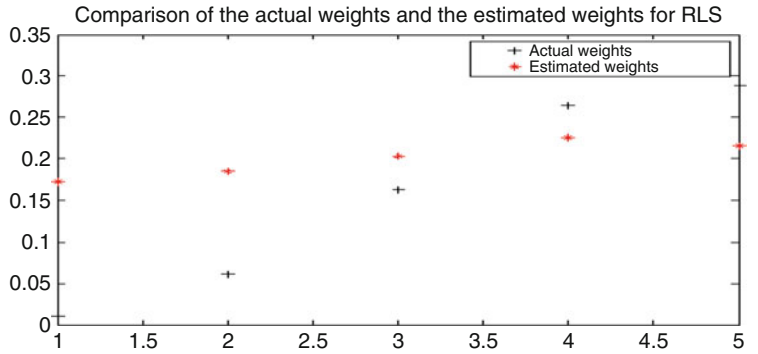

Fig. 5.72 Actual and true filter weight in RLS algorithm for traffic jam noise signal

true and estimated output in the case of traffic jam noise. Figure 5.72 shows actual and estimated weight coefficients for proper operation during traffic jam noise. Deviation between actual and estimated coefficients shows the amount of modification requiref for noise reduction.

Subsequently, Fig. 5.73 shows the convergence curve for the variation of the coefficients. Still, the nature of RLS can again be observed here. The rate of convergence is somewhat speedier and after some time will achieve steady-state values. Obviously per the nature of RLS, the characteristics reflect in a speedy manner. Figure 5.73 shows nearly the same results as taken in the previous two cases of noise. In the simulation result the first condition shows the output of the VAD algorithm in that only some amount of noise is cleaned and only in the silence part. The next result shows the output of RLS in which all the silences as well as speech utterance parts are combined in cleaning (Fig. 5.74).

Figure 5.75 shows the cleaned speech with its spectrogram in detail. Every noticeable frequency with a moderate amount of amplitude is shown in the result.

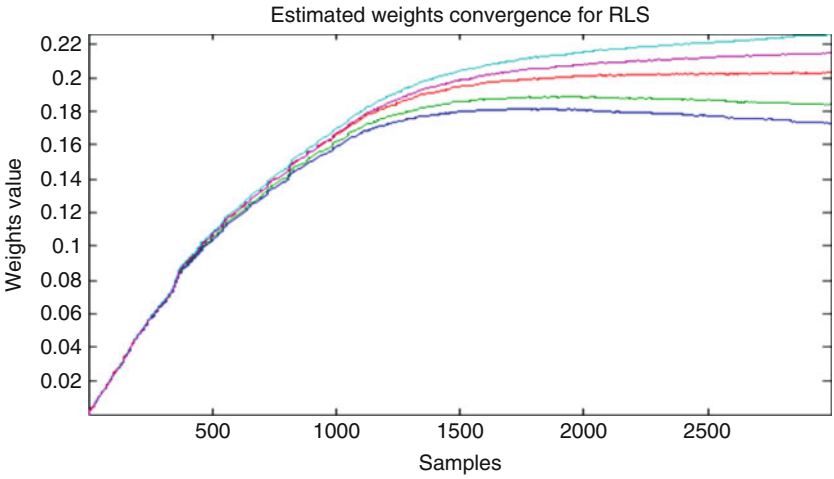

Fig. 5.73 Estimated convergence learning curve in RLS algorithm for traffic jam noise signal

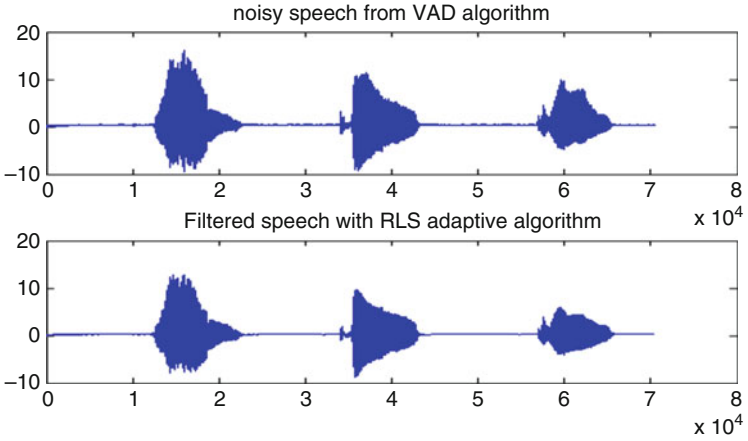

Fig. 5.74 Noisy speech signal with traffic jam noise signal from VAD algorithm and filtered speech signal with RLS algorithm

Next, filtered speech is given to the wavelet multi-resolution algorithm for each band evaluation, and after taking the wavelet approximation a very large difference between original noisy speech and filtered speech can be observed, which is represented in Fig. 5.76.

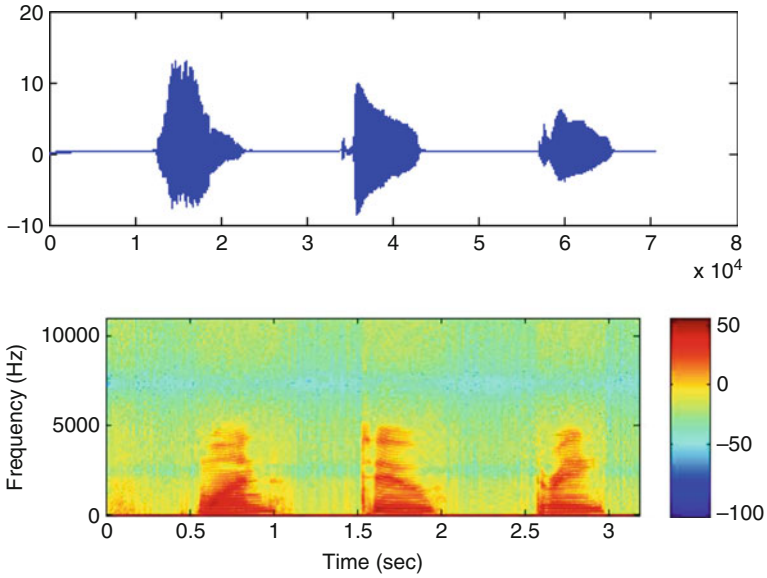

Fig. 5.75 Filtered speech signal by RLS algorithm for speech affect by traffic jam noise signal and its spectrogram

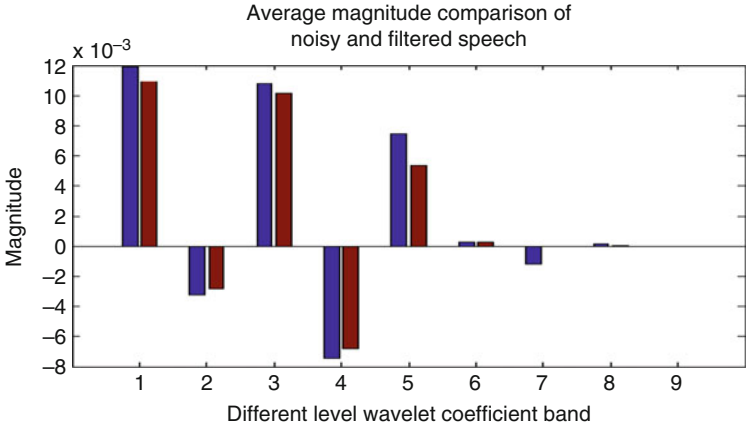

Fig. 5.76 Comparison of noisy speech signal with traffic jam noise signal and filtered speech signal in wavelet domain for RLS algorithm

5.7 Comparative Analysis of Simulation Results

Table 5.2 shows the obtained results for the speech enhancement process using these three adaptive algorithms for different noise signals.

After obtaining results for these three adaptive algorithms, it can be observed that for any given level of speech and noise, LMS gives the best performance. The derived conclusion is that the LMS is with the best PSNR and least MSE. The NLMS algorithm is much nearer to LMS, but the difference is that the convergence rate of LMS is faster because it is normalized at every iteration. In NLMS, normalized mu is applicable every time, so the convergence rate is faster, but MSE is higher compared to LMS. Although in the case of RLS it can be noticed from the learning curve that compared to LMS, weight convergence is faster, but it is less than NLMS. In the RLS algorithm MSE is least and PSNR is low. It is very difficult in RLS to match with the lower MSE higher rate convergence. In NLMS the weight value can be normalized every time by the variance vector of input tap, so in fast convergence performance is better compared to RLS.

As increments in the system order are carried out, better quality of noise reduction is possible, with more complexity added to the existing system. In the higher-order system, noise reduction can easily be observed, but with that some sound clarity is reduced, because a higher number of coefficients is required in the higher-order system and thus more complexity will be present, and with that some redundant components in speech responsible for clarity will be reduced.

Table 5.2 Obtained results for speech enhancement process in terms of PSNR, SNR, and MSE

	Type of noise signal		
Type of algorithm	White noise	Babble	Traffic jam
PSNR (dB)			
LMS	50.3948	50.3947	50.3942
NLMS	50.2423	50.2409	50.2415
RLS	49.4038	49.4034	49.4036
SNR (dB)			
LMS	6.1412	6.1401	6.14
NLMS	5.3313	5.3313	5.3282
RLS	4.9473	4.9469	4.947
MSE			
LMS	0.6129	0.5938	0.5938
NLMS	0.615	0.6152	0.6151
RLS	0.7459	0.746	0.746

References

1. Haykin, S. S. (2008). *Adaptive filter theory*. Chennai: Pearson Education India.
2. Honig, M. L., & Messerschmitt, D. G. (1984). *Adaptive filters: Structures, algorithms and applications*. New York: Springer.
3. Diniz, P. S. R. (2008). *Adaptive filtering: Algorithms and practical implementation*. New York: Springer.
4. El-Fattah, M. A., Dessouky, M. I., Diab, S. M., & El-Samie, F. A. (2008). Adaptive Wiener filtering approach for speech enhancement. *J Ubiquitous Comput Commun, 2*(3), 23–31.
5. Abdulmagid, M. A., Krusienski, D. J., Pal, S., & Jenkins, W. K. (2004). *Principles of adaptive noise canceling*. University Park: Department of Electrical & Computer Engineering, Pennsylvania State University.

Chapter 6
Speech Signal Enhancement Based on Wavelet Transform

6.1 Procedure for Speech Signal Enhancement Using Wavelet Transform

The main motivation behind using wavelet analysis for speech processing is that wavelets have good localization in the time and frequency domains [1–3]. Wavelets are functions suited to the expansion of nonstationary continuous signals. Wavelet transform maps the signal from the time domain into the time–frequency domain [4, 5]. Redundancy and irrelevancy present in the speech signal (quasi-stationary signal) can be easily removed using wavelet transform. Figure 6.1 shows the flow of the speech enhancement algorithm using wavelet transform.

To achieve the desired modification of an input speech signal, the following steps are required to process the signal.

- Step 1: Take a reference speech signal from the adaptive algorithm to be processed.

  As discussed in Chap. 5, per the requirement any adaptive algorithm can be selected for noise reduction. Again, after reduction of noise filtered speech should be trained according to the format of the audiogram. Person-to-person modification of the band will change. So, the process should be carried out per the requirement for the patient. In that band, enhancement is necessary. Filtered speech should be passed to the wavelet algorithm for further processing.

1. Set the property of the speech signal.
2. Obtain the output of the adaptive algorithm for further processing in the enhancement.
3. Study the audiogram of the specific person for decisions regarding modification.

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_6

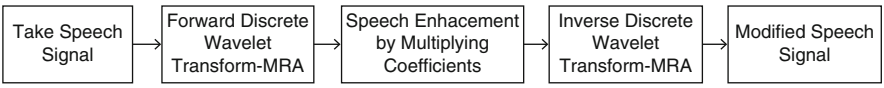

Fig. 6.1 Block diagram of the speech enhancement algorithm using wavelet transform

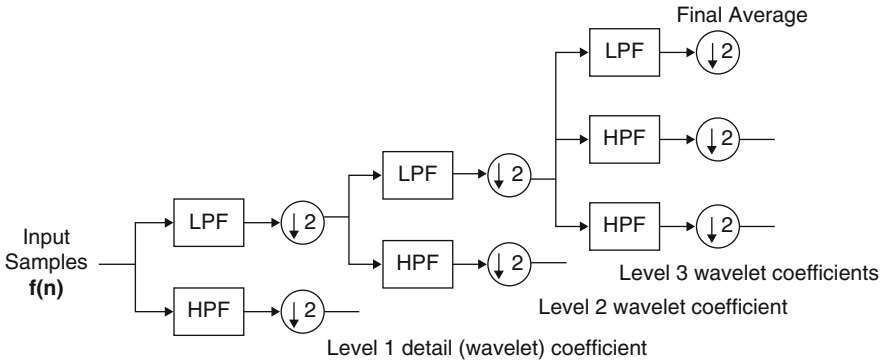

Fig. 6.2 Forward discrete wavelet transform

- Step 2: Implement discrete wavelet transform (multi-resolution analysis [MRA])

  A discrete wavelet transform (DWT) performs a multistage signal decomposition using a filter bank structure. DWT coefficients can be calculated using Mallat's fast tree algorithm, also known as fast wavelet transform (FWT). This algorithm consists of a number of identical stages. At each stage, the input signal is filtered through a low-pass and a high-pass filter. The filtered samples are then downsampled by a factor of 2. The output of the low-pass filter gives coarser information, and the output of the high-pass filter gives detailed information. In normal DWT, coarse data from the previous stage is used as input for the next stage, which gives progressively low-frequency resolution and details.

  A window of input samples $f(n)$ is applied at the input of the filter bank. The size of the window is a power of 2 (if it is not power of 2, some algorithm is applied to divide this sample into different blocks and ensure that each block length is power of 2). First, some of the samples are taken in the form of a small window for processing. That small number of sample sets are processed in parallel. Those data at one time become convoluted with a high-pass filter coefficient and at the second time with low-pass filter coefficients. As a result, the obtained samples double in number. So, to maintain the hierarchy of the process it is necessary to reduce the number of samples through the process of downconversion. This process of filtering and downsampling is collectively known as decimation. The downsampled output of the first high-pass filter (decimated output) becomes "level 1" wavelet coefficients, which contain the detail part of the signal. The output of the low-pass filter is the approximate output at "level 1" that can be further downsampled and given to the next identical stage, as shown in Fig. 6.2. From this signal, another more approximated signal and

wavelet coefficient at "level 2" can be separated out. This process is repeated to the *N*th level. Length of input signal is divided by two every time as the level increases. Higher-level wavelet coefficients have lower time resolution and narrower bandwidth information.

In case of a forward wavelet transform, downsampling is important to ensure the output samples are the same size as the input samples. Without reducing the number of samples, the original data of the matrix would be increased, known as data explosion. As a by-product of wavelet coefficient and data convolution two sets of matrices can be achieved: the high-pass filter and low-pass filter set. The high-pass filter matrix represents wavelet coefficients and normal values that can be known as an average and can be represented by a low-pass filter set. Increment and decrement in the samples represents time resolution and input signal and how with time the signal can be changed. Gradually per discrete level the number of coefficients would be decreased. Generally, if the number of sampled values in the matrix is 2028 as an input, values at the first level carry 1024 wavelet values, 512 values taken by second level and 256 individual numbers taken by the third level and so on as hierarchy progresses. Each level value matrix represents a unique feature of the signal. As analysis says, level by level it is twofold decrease in the number of samples. Wavelet values on each level describe different numbers and different values so time resolution can be obtained. Moreover, more time resolution wants more value numbers in the level. In fact, each level gives time–frequency information of that small window.

1. Using commands, get selection of wavelet name followed by selection and detail and approximate coefficient values.
2. Detail and approximate values are averaged at some threshold.
3. Use matrix operations to get rows and columns of signal in required format.
4. Initialize the vector to store the number of coefficients after each and every decomposition.
5. Acquire the convolution of high-pass and low-pass filters with original signal individually and store data of each convolution after downsampling by using function dyadown.
6. Repeat the foregoing step eight times.
7. At each successive execution consider convolution of high-pass filter and signal as a main signal for the next execution.
8. Store the value of the wavelet coefficient for eight decomposition stages and the number of coefficients after each stage in two different variables, WC and LC.

- Step 3: Speech Enhancement Process

  Speech enhancement is required per the conclusion from the audiogram of the person. Band selection from the audiogram is necessary for increments in the

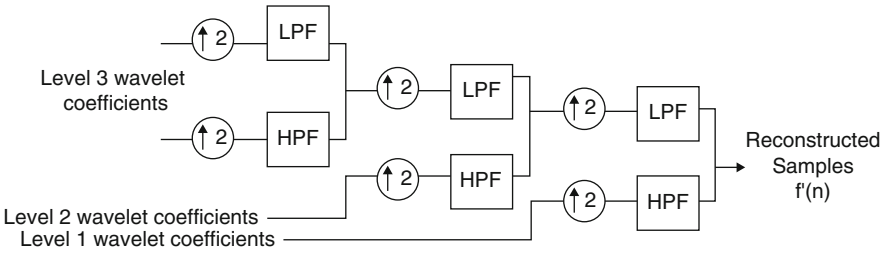

Fig. 6.3 Inverse discrete wavelet transform

loudness level in decibels. After selecting a particular band, the same is selected for the modification. The normal hearing threshold level is 20 dB in humans under general conditions. Whatever decrement there is in loudness level requires to be corrected for that frequency band, which can be observed from the audiogram.

1. Select particular band of stored wavelet coefficient for the enhancement.
2. Multiply with suitable numerical value per the required gain and selected wavelet coefficient band per the frequency of the input speech.

- Step 4: Implementation of Inverse Discrete Wavelet Transform

  The inverse wavelet transform works exactly as a reverse then forward wavelet transform. Final average and *N*th level wavelet coefficients are upsampled. The logical process handles this by padding zeros between samples. As a result, a new matrix will be formed that is supposed to pass through the inverse low-pass filter and inverse high-pass filter, which is also known as interpolation. Resultant outputs are then summed up and forwarded to the next step of process. The described process continues until the end.

  By the understanding of wavelet transform, implementation for forward and reverse is not giving much difficulty (as shown in Figs. 6.2 and 6.3). But in practice, real data processing causes various problems in the implementation. Real image and speech carry a very large amount of sampled values in the form of a matrix or in a series of matrices. At the time of processing it is not appropriate to apply a filtering coefficient on the entire data input. Naturally, windowing is a must on the large amount of data. Most of the time dealing with boundaries in the windowing generates many crucial issues. If boundaries of the window are not handled carefully, reconstruction of the data is troublesome. Generally, ringing is generated at the end of the window. To solve the issue of end conditions, many methods can be apply to smooth the windowed data. Symmetrical extension, zero padding, and circular convolution are the most popular methods. All the digital filters must have designing criteria with the condition of perfect reconstruction ability; this is only true when at the time of the windowing samples are separated,

and the form of the window edges must be handled perfectly. Care should be taken that any discrepancy such as distortion or aliasing not take place at the edges. The reconstructed signal must be identical to the original signal. Choosing the mother curve for wavelet processing requires many trade-offs. Directly in the time domain the mother wavelet can be convolved with the original input signal. In the original process, first the mother wavelet is selected and from that a similar set of high-pass and low-pass filters can be chosen.

1. Using function 'wfilters,' get the high-pass and low-pass coefficients by giving a specific wave name and time.
2. Normalize high-pass and low-pass coefficients of the wavelet.
3. Use matrix operations to obtain rows and columns of signal in required format.
4. Initialize the vector to store the number of modified wavelet coefficients.
5. Get the convolution of high-pass and low-pass filters with upsampled approximate and detailed coefficients of modified wavelet individually and add them.
6. Store additions and in variable 'rS' and repeat the procedure up to the last stage reached.

- Step 5: Comparison of Modified and Original Speech Signal

  For observing comparison between original and modified speech files, both files should be reconstructed in the same wave format. Reconstructed data should be plotted in a graph showing frequency versus decibels. By reading the graphs, enhancement in the selected band can be seen per the requirement. Power spectral density function is used to plot the frequency versus decibel graph of the two speech files named 'pwelch.' The procedure for executing the power spectral density by the 'pwelch' algorithm is as follows:

1. Fix the size of the FFT window for samples.
2. Initialize N for N point FFT.
3. Calculate the number of overlapping samples in the window.
4. Read the original speech file and store the bytes in a variable.
5. Apply 'pwelch' function on the sample bytes; it returns the average spectrum of the signal.
6. Convert average spectrum value in the decibels and plot decibels versus frequency.
7. Repeat steps five and six for the modified speech file reconstructed after IDWT.
8. Take the difference in decibels between two average spectrums.
9. Plot the difference in decibels, which gives the modification for the selected band.

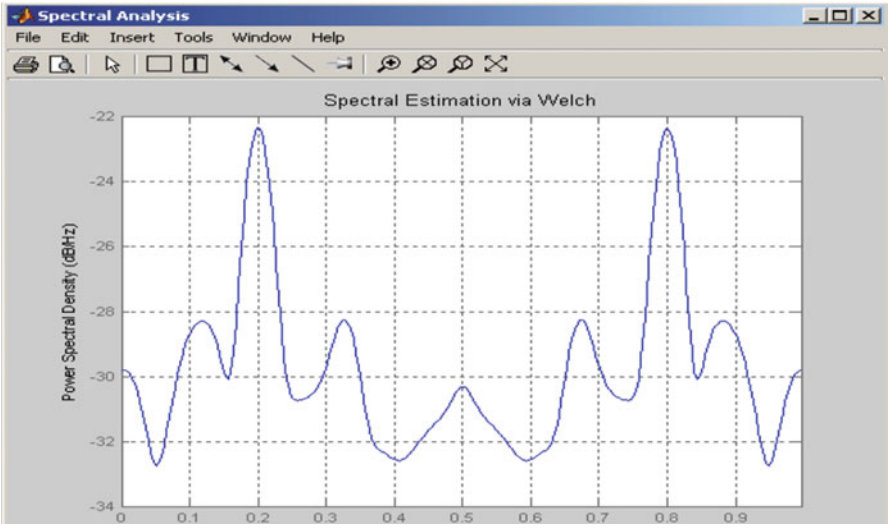

Fig. 6.4 Graph of power spectral density versus frequency

Figure 6.4 utilizes below mentioned parameters and protocols for finding the power spectral density.

- By default, the vector is divided into eight sections with 50% overlap.
- Each section is windowed with a Hamming window *n* point FFT applied to the windowed data, and eight modified periodograms are computed and averaged.

6.2 Implementation and Results of Speech Signal Enhancement Using Wavelet Transform

The wavelet is a very important tool for enhancement of the speech signal. As mentioned earlier, for a speech signal that contains information about time, frequency, and magnitude, the wavelet is a very good approximation. Basically, speech signal frequency is in the range from 300 Hz to 4 kHz. Again, the most important application of wavelet is in the Mallat algorithm. Each and every stage of the wavelet application gives partition in the frequency domain. In the simulation process, stages of the wavelet application must be defined. Accordingly, the speech frequency bandwise coefficient can be achieved. As shown in Fig. 6.5, as many wavelet stages are applied, more minute resolution in the frequency can be achieved. Per the requirement of the audiogram, the individual band of the frequency needs to be modified. For that, in the wavelet coefficient band some numerical factor can be searched, and because of that specified decibel the increment can be searched out. From that value, some average modification is given to that band and speech can be enhanced. In the

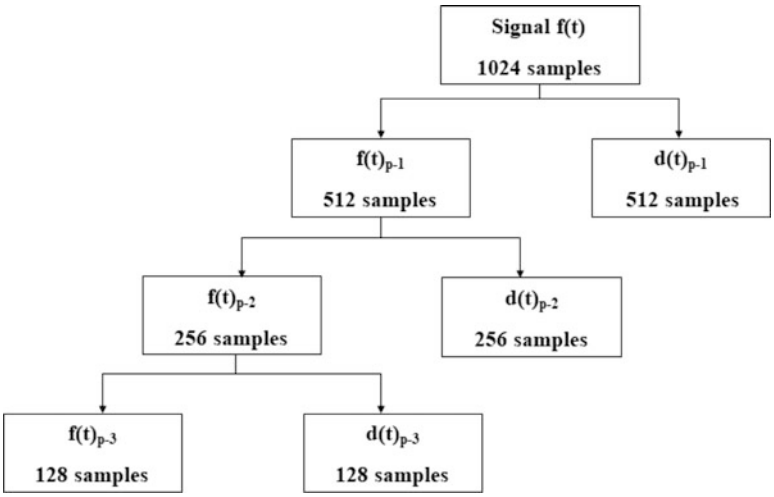

Fig. 6.5 Sample decomposition hierarchy

presented simulation results, there are eight wavelet stages. As a result, nine bands can be prepared. A band-wise individual factor is selected here because of that overall 3 dB. The increment results are shown in the following simulation results. If more than 3 dB is required, then another value of the multiplying factor should be chosen.

6.2.1 First Band Enhancement

The wavelet multi-resolution tool is very useful for frequency band partition. Figure 6.6 shows simulation results of original speech and enhanced speech. In the developed research, eight level wavelet transforms are used, so nine different bands are generated. First, band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. First, band wavelet coefficients are multiplied with a factor of 7.67.

Figure 6.7 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other enhanced band coefficients of wavelet transform. When signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be achieved. In the present work, for example, a 3 dB variation is taken. By multiplying with a suitable numerical value in between with wavelet coefficients, an average increment in the band can be noticed.

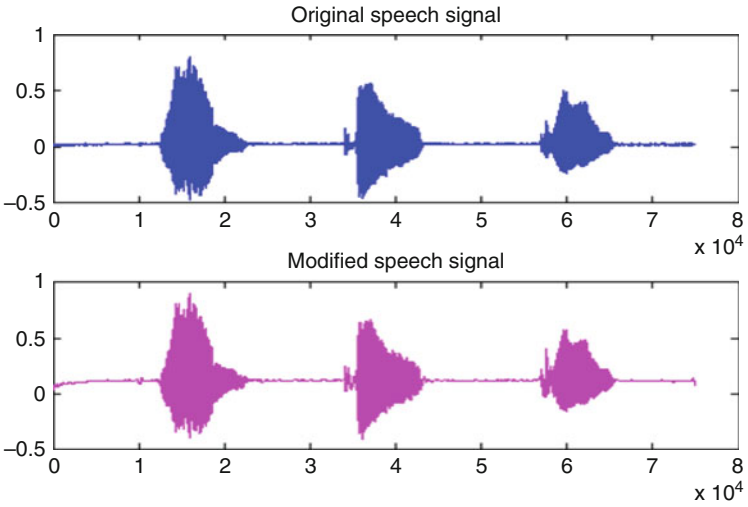

Fig. 6.6 Original and reconstructed speech signal for first band enhancement

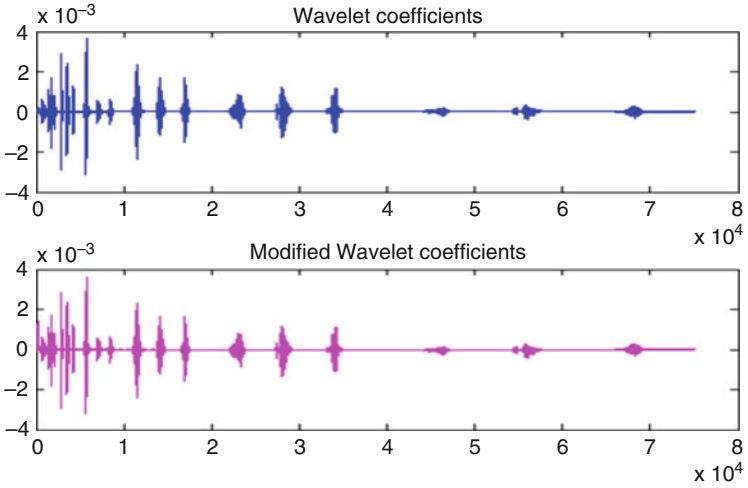

Fig. 6.7 Original and modified wavelet coefficients for first band enhancement

Figure 6.8 shows the power spectral density (psd) of original band and then the modified psd is overlapped on the same band. In the simulation result, the second graph shows a clear plot of difference in the first band for reference. Thus, a very accurate amount of sound intensity increment can be observed by applying the appropriate multiplication in the MRA wavelet domain.

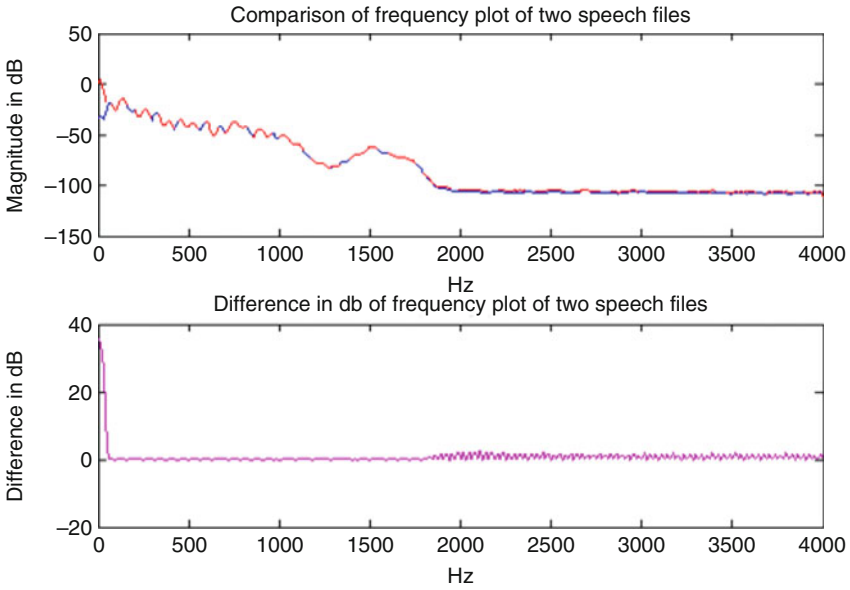

Fig. 6.8 First band enhancement of 3 dB using multiplying factor 7.67

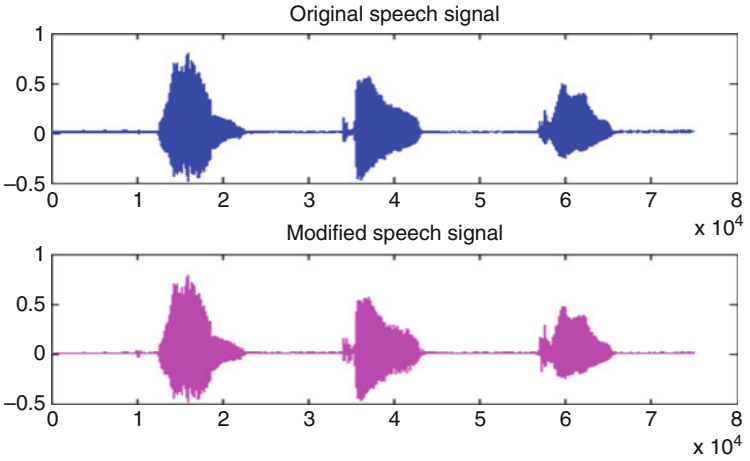

Fig. 6.9 Original and reconstructed speech signal for second band enhancement

6.2.2 Second Band Enhancement

Using the wavelet multi-resolution band, partition can be implemented. Figure 6.9 shows the simulation results of original speech and enhanced speech. In the developed research, the eight level wavelet transform is used so nine different bands are generated using wavelet db4. The second band wavelet coefficients are selected and

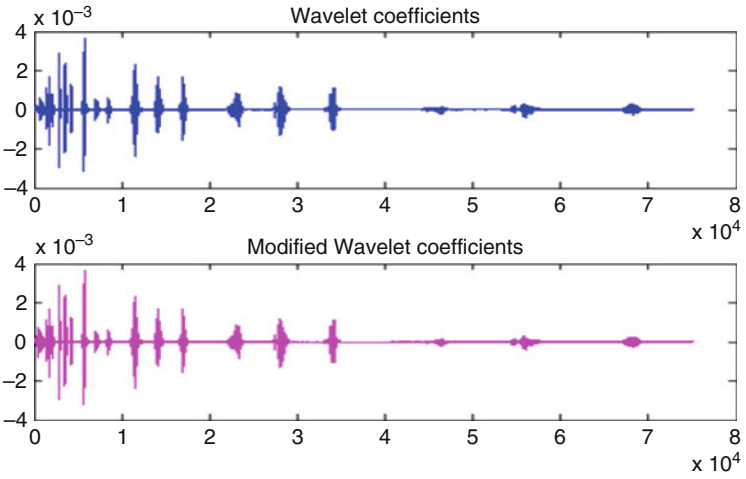

Fig. 6.10 Original and modified wavelet coefficients for second band enhancement

enhancement is performed between wavelet transform and inverse wavelet transform. Second band wavelet coefficients are multiplied with factor 6.3.

Figure 6.10 shows a simulation plot for two categories of wavelet coefficient, one the original coefficient and the other enhanced band coefficients of wavelet transform. When the signal is reconstructed after enhancement some magnitude of variation is noticed. Per requirement any level of magnitude increment can be achieved. In the present work, for example, a 3 dB variation is taken. By multiplying with a suitable numerical value between wavelet coefficients, the average increment in the band can be seen.

Figure 6.11 shows the psd of the original band and then the modified psd is overlapped on the same band. In the simulation result the second graph shows a clear plot of difference in the second band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the MRA wavelet domain.

6.2.3 Third Band Enhancement

In the simulation shown, a level eight wavelet multi-resolution tool is used. Figure 6.12 shows simulation results of original speech and enhanced speech. Third band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. Third band wavelet coefficients are multiplied with factor 1.5.

Figure 6.13 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other enhanced band coefficients of wavelet transform.

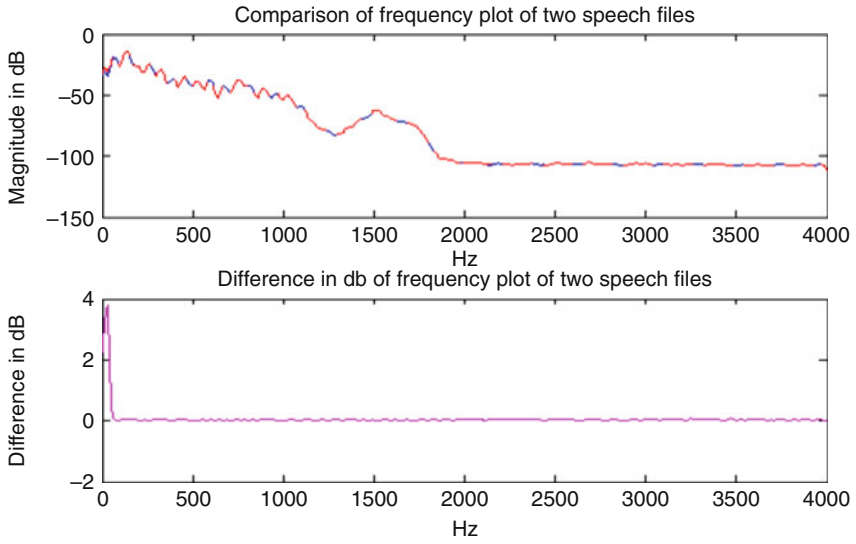

Fig. 6.11 Second band enhancement of 3 dB using multiplying factor 6.3

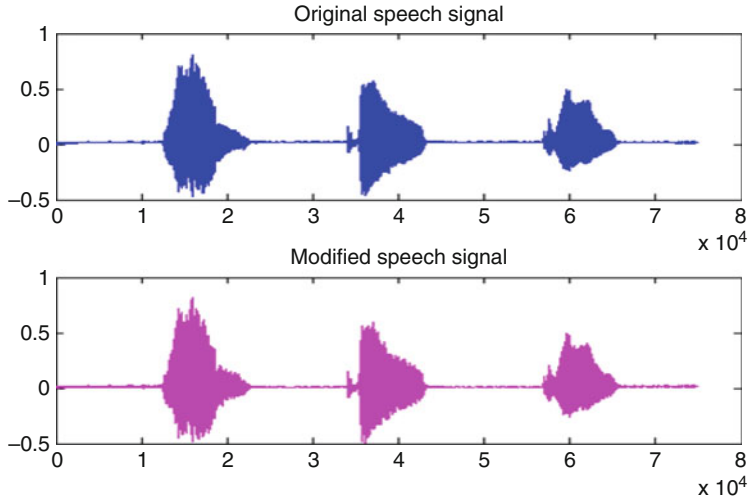

Fig. 6.12 Original and reconstructed speech signal for third band enhancement

When signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be achieved. In the present work a 3 dB variation is taken. By multiplying with suitable numerical value between wavelet coefficients, the average increment in the band can be seen.

Figure 6.14 shows the psd of the original band and then the modified psd is overlapped on the same band. In the simulation result the second graph shows a clear

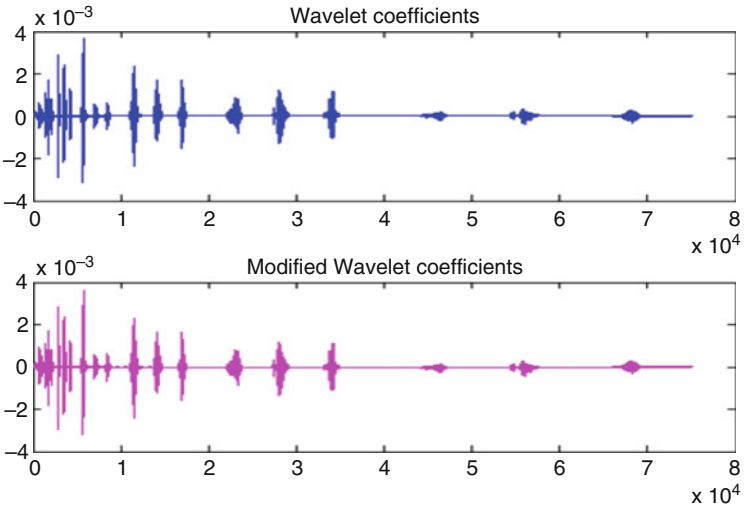

Fig. 6.13 Original and modified wavelet coefficients for second band enhancement

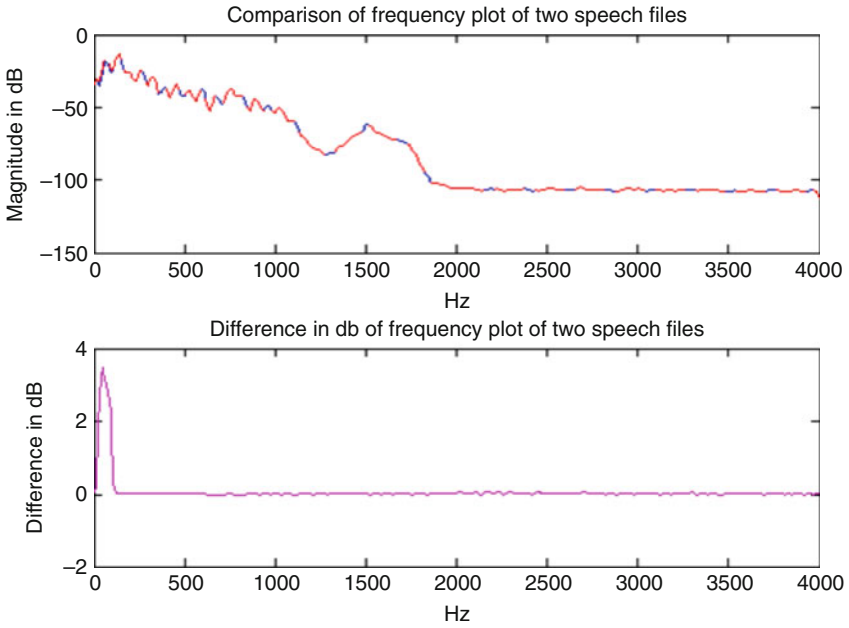

Fig. 6.14 Third band enhancement of 3 dB using multiplying factor 1.5

plot of difference in the third band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the MRA wavelet domain.

6.2.4 *Fourth Band Enhancement*

The wavelet multi-resolution tool is very useful for frequency band partition. Figure 6.15 shows simulation results of original speech and enhanced speech. In the developed research, an eight level wavelet transform is used so nine different bands are generated. Fourth band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. Fourth band wavelet coefficients are multiplied with factor 4.1.

Figure 6.16 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other enhanced band coefficients of wavelet transform. When signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be achieved. In the present work 3 dB variation is taken. By multiplying with a suitable numerical value between the wavelet coefficients, the average increment in the band can be detected.

Figure 6.17 shows the psd of original band and then the modified psd is overlapped on the same band. In the simulation result the second graph shows a clear plot of difference in the fourth band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the MRA wavelet domain.

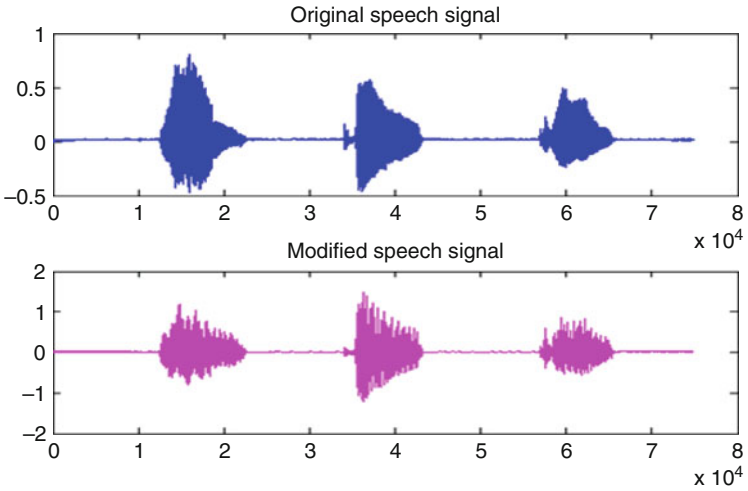

Fig. 6.15 Original and reconstructed speech signal for fourth band enhancement

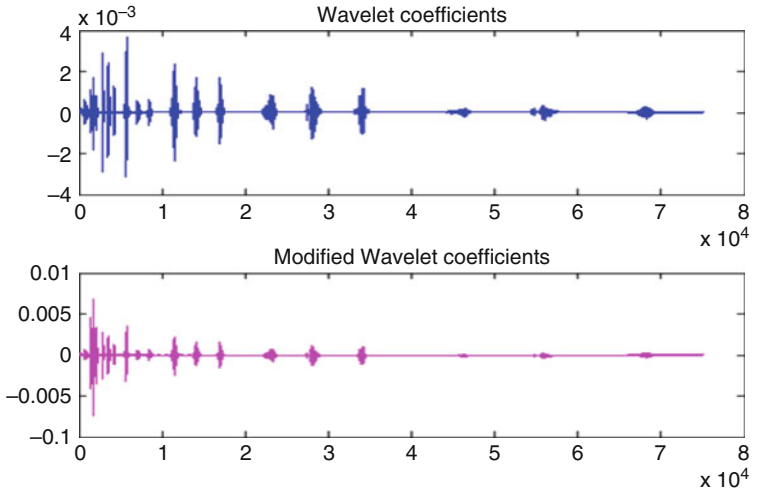

Fig. 6.16 Original and modified wavelet coefficients for fourth band enhancement

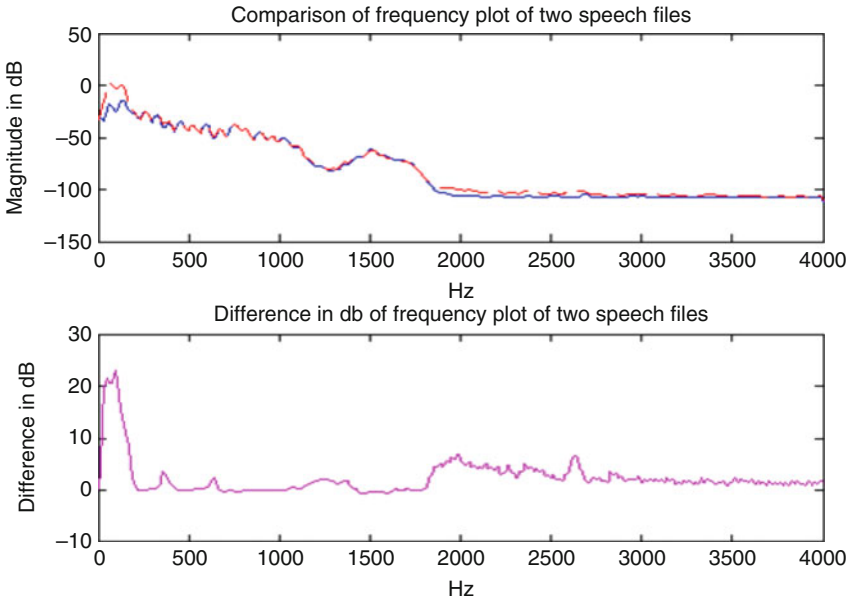

Fig. 6.17 Fourth band enhancement of 3 dBusing multiplying factor 4.1

6.2.5 Fifth Band Enhancement

Speech is a complex wave of more than one frequency. The wavelet tool gives a frequency band partition. Figure 6.18 shows simulation results of original speech and enhanced speech. In the developed research, a eight level wavelet transform is

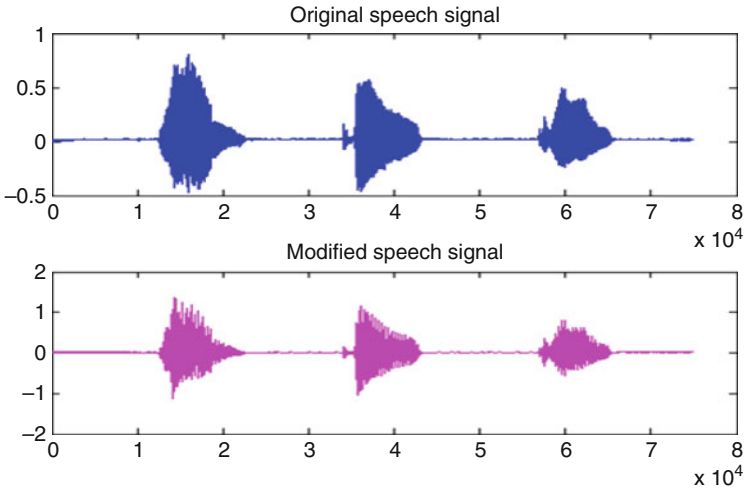

Fig. 6.18 Original and reconstructed speech signal for fifth band enhancement

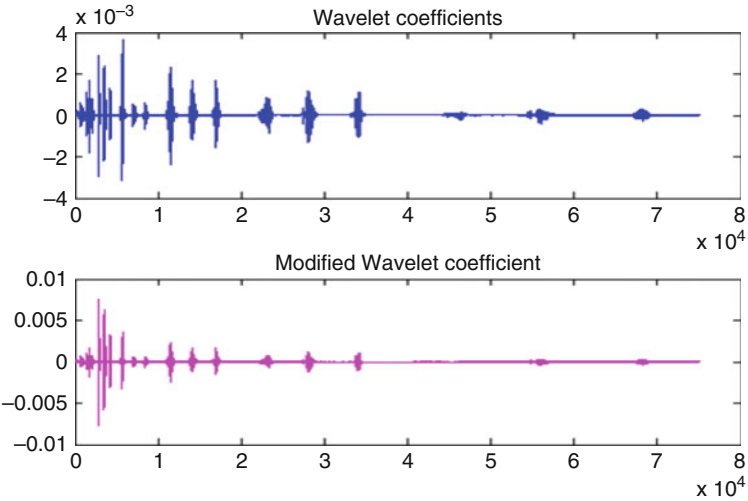

Fig. 6.19 Original and modified wavelet coefficients for fifth band enhancement

used. Fifth band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. Fifth band wavelet coefficients are multiplied with factor 2.6.

Figure 6.19 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other the enhanced band coefficients of wavelet transform. When the signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be achieved. In the present work a 3 dB variation is taken. By multiplying with a

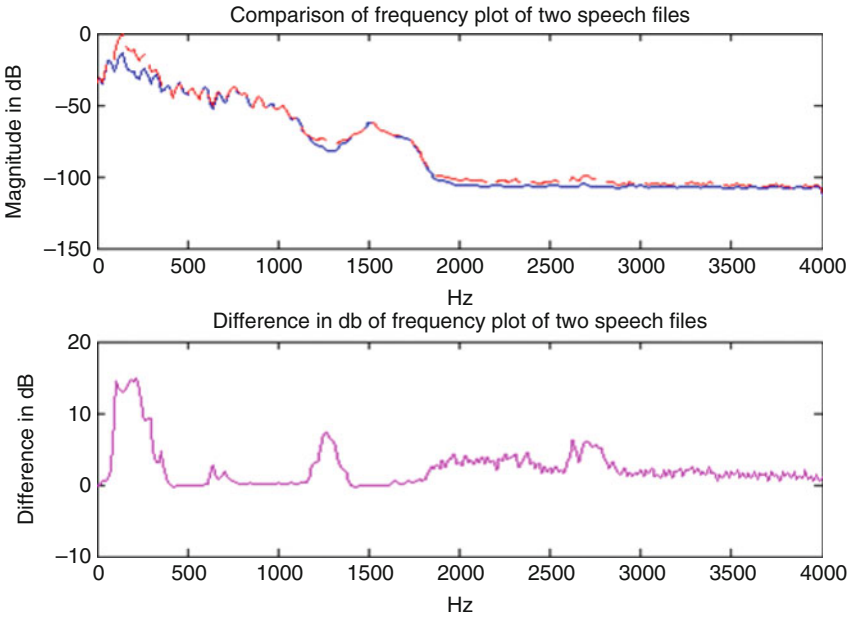

Fig. 6.20 Fifth band enhancement of 3 dB using multiplying factor 2.6

suitable numerical value between wavelet coefficients, the average increment in the band can be detected.

Figure 6.20 shows the psd of the original band and then the modified psd is overlapped on the same band. In the simulation result, the second graph shows a clear plot of difference in the fifth band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the MRA wavelet domain.

6.2.6 Sixth Band Enhancement

For quasi-periodic speech, the signal wavelet multi-resolution tool is very useful for frequency band partition. Figure 6.21 shows simulation results of original speech and enhanced speech. In the developed research, an eight level wavelet transform is used. Sixth band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. Sixth band wavelet coefficients are multiplied with factor 1.8.

Figure 6.22 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other enhanced band coefficients of wavelet transform. When the signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be

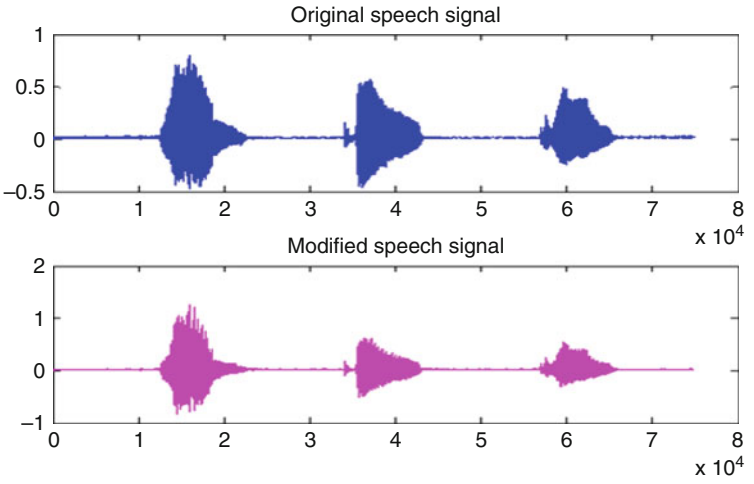

Fig. 6.21 Original and reconstructed speech signal for sixth band enhancement

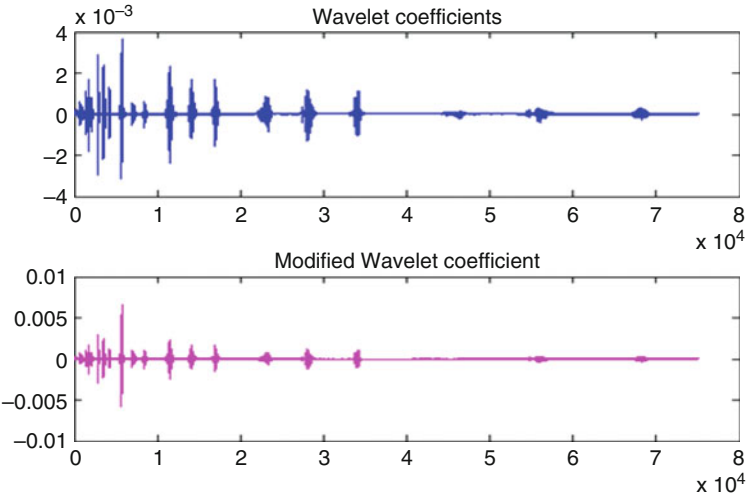

Fig. 6.22 Original and modified wavelet coefficients for sixth band enhancement

achieved. In the present work a 3 dB variation is taken. By multiplying with a suitable numerical value between wavelet coefficients, an average increment in the band can be seen.

Figure 6.23 shows the psd of the original band and then the modified psd is overlapped on the same band. In the simulation result, the second graph shows a clear plot of difference in the sixth band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the MRA wavelet domain.

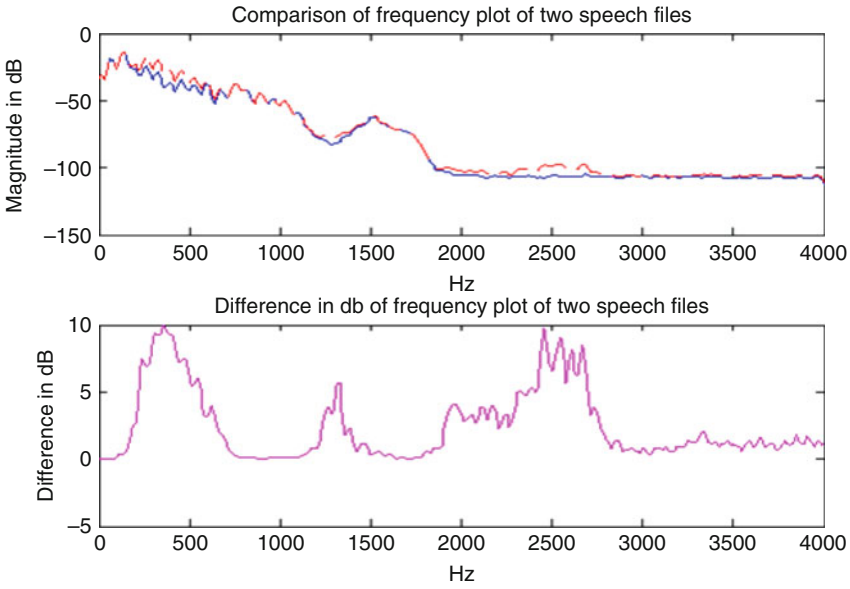

Fig. 6.23 Sixth band enhancement of 3 dB using multiplying factor 1.8

6.2.7 Seventh Band Enhancement

The wavelet multi-resolution tool is very useful for frequency band partition. Figure 6.24 shows simulation results of original speech and enhanced speech. In the developed research, a level eight wavelet transform with wavelet db4 is used. Seventh band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. Seventh band wavelet coefficients are multiplied with factor 1.3.

Figure 6.25 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other the enhanced band coefficients of wavelet transform. When the signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be achieved. In the present work a 3 dB variation is taken. By multiplying with a suitable numerical value between wavelet coefficients, the average increment in the band can be observed.

Figure 6.26 shows the psd of the the original band and then modified psd is overlapped on the same band. In the simulation result, the second graph shows a clear plot of difference in the seventh band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the wavelet domain.

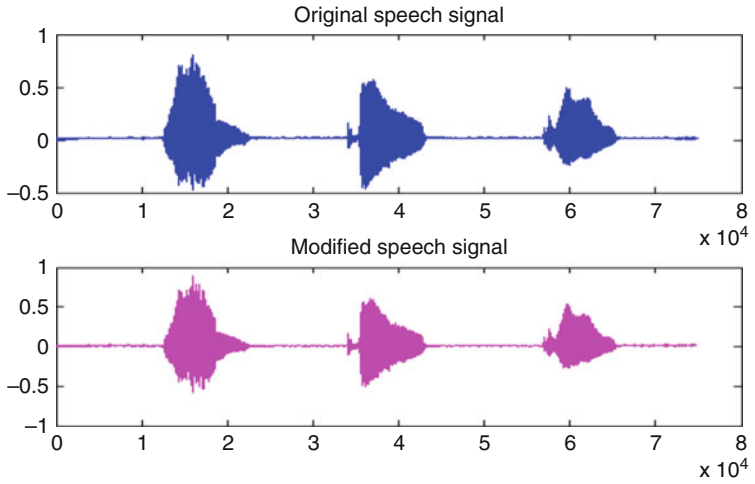

Fig. 6.24 Original and reconstructed speech signal for seventh band enhancement

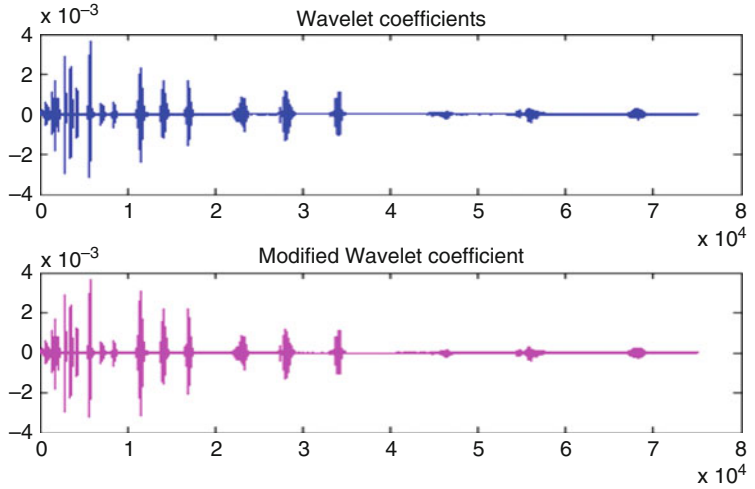

Fig. 6.25 Original and modified wavelet coefficients for seventh band enhancement

6.2.8 Eighth Band Enhancement

The wavelet multi-resolution tool is very useful for frequency band partition. Figure 6.27 shows simulation results of original speech and enhanced speech. In the developed research an eight level wavelet transform is used, so nine different bands are generated. Eighth band wavelet coefficients are selected, and enhancement is performed between wavelet transform and inverse wavelet transform. Eighth band wavelet coefficients are multiplied with factor 1.25.

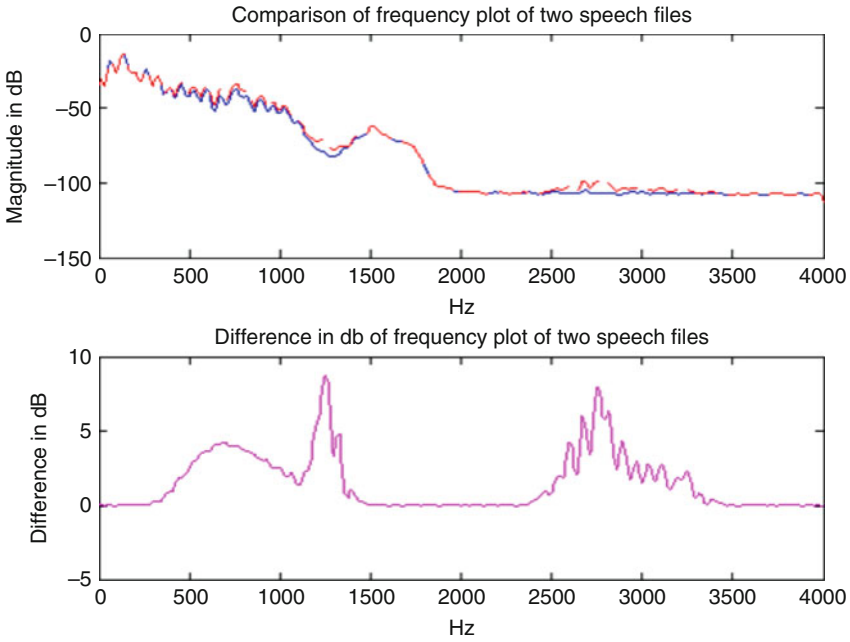

Fig. 6.26 Seventh band enhancement of 3 dB using multiplying factor 1.3

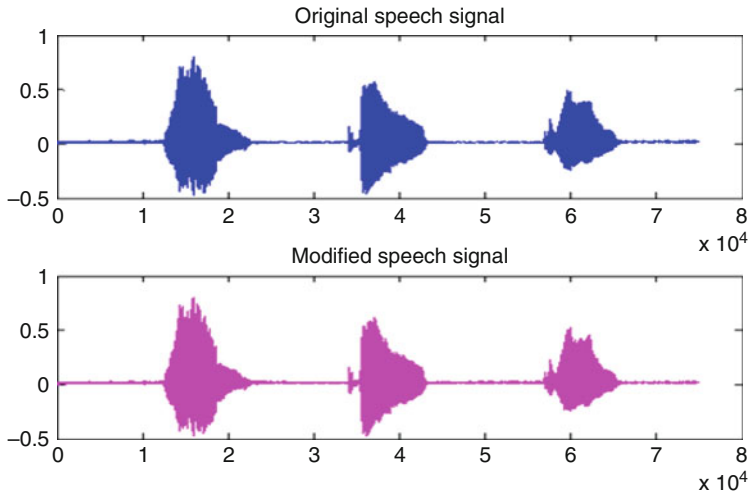

Fig. 6.27 Original and reconstructed speech signal for eighth band enhancement

Figure 6.28 shows a simulation plot for two categories of wavelet coefficient, one the original coefficients and the other enhanced band coefficients of wavelet transform. When the signal is reconstructed after enhancement, some magnitude of variation is noticed. Per requirement any level of magnitude increment can be

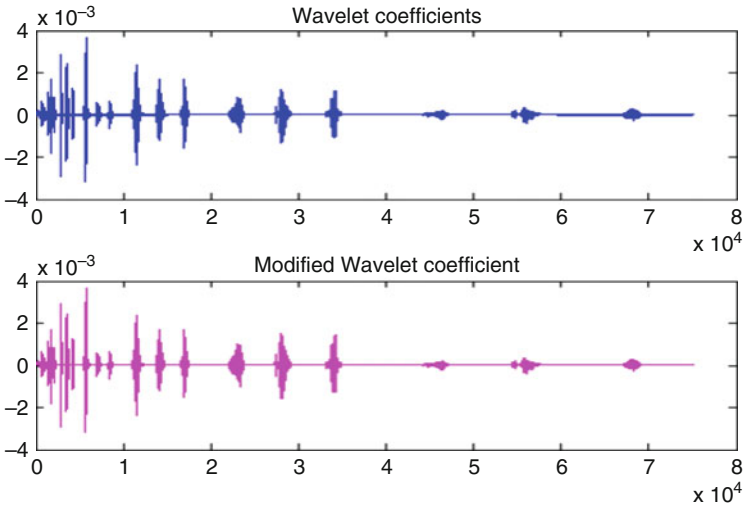

Fig. 6.28 Original and modified wavelet coefficients for eighth band enhancement

achieved. In the present work a 3 dB variation is taken. By multiplying with a suitable numerical value between wavelet coefficients, the average increment in the band can be seen.

Figure 6.29 shows the psd of the original band, and then the modified psd is overlapped on the same band. In the simulation result the second graph shows a clear plot of difference in the eighth band for reference. Thus, a very accurate amount of sound intensity increment can be observed by taking appropriate multiplication in the wavelet domain.

6.2.9 Ninth Band Enhancement

Original and modified speech are shown in Fig. 6.30. The first simulation result shows the original noise-removed speech, which is the output of any adaptive algorithm: it may be either LMS, NLMS, or RLS. Then, taken speech is required to modify per the audiogram. The first layout shows the original and the second shows the band-enhanced layout.

The second simulation result in Fig. 6.31 shows two layouts. In the first, output shows the original coefficient after applying wavelet transform of eight stages. All the band coefficients are shown. In the second layout, for each and every band the modified coefficients are shown in the result after multiplication of some numerical value for the increment of 3 dB.

The third result in Fig. 6.32 shows a plot of the original psd of the signal. The original psd shows the energy of the natural band, which is the output of the adaptive filter. The overlapping graph shows band enhancement after multiplying with some

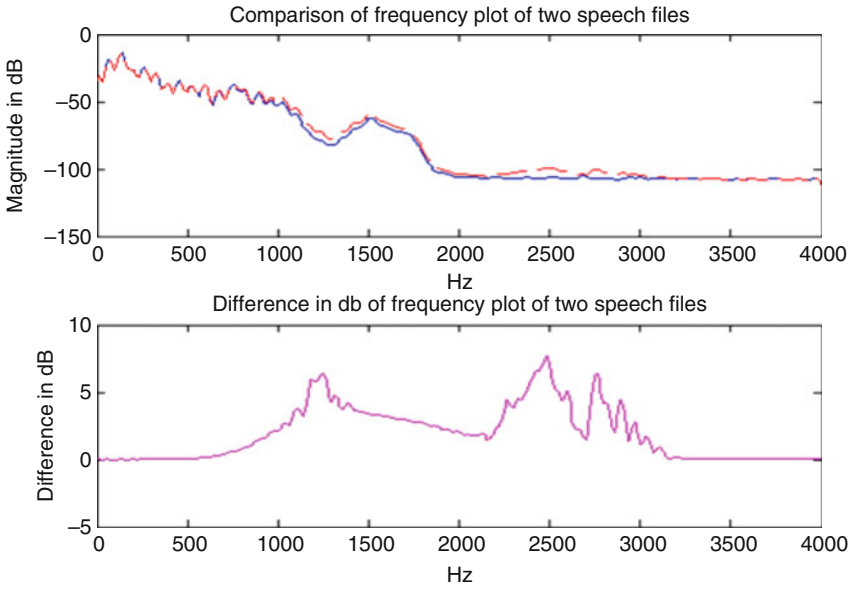

Fig. 6.29 Eighth band enhancement of 3 dB using multiplying factor 1.25

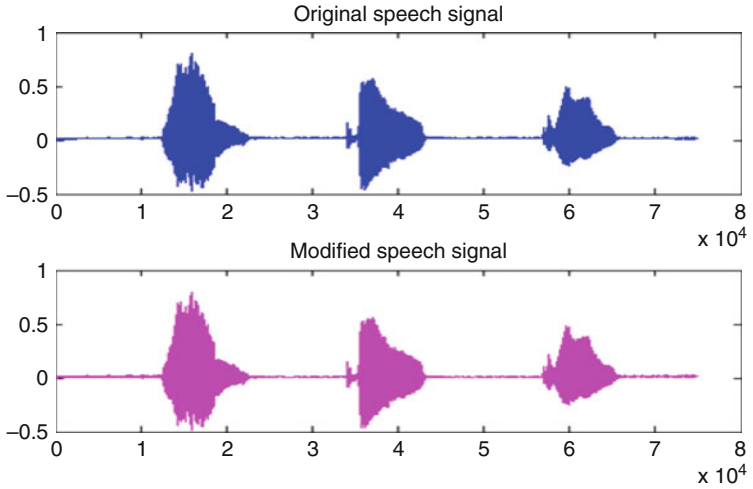

Fig. 6.30 Original and reconstructed speech for ninth band enhancement

numerical coefficient. It can be observed that only some part of the original psd plot has been changed, which reflects the effect of wavelet coefficient multiplication in the frequency domain. Moreover, the second plot shows a clear difference between the two plots in reality.

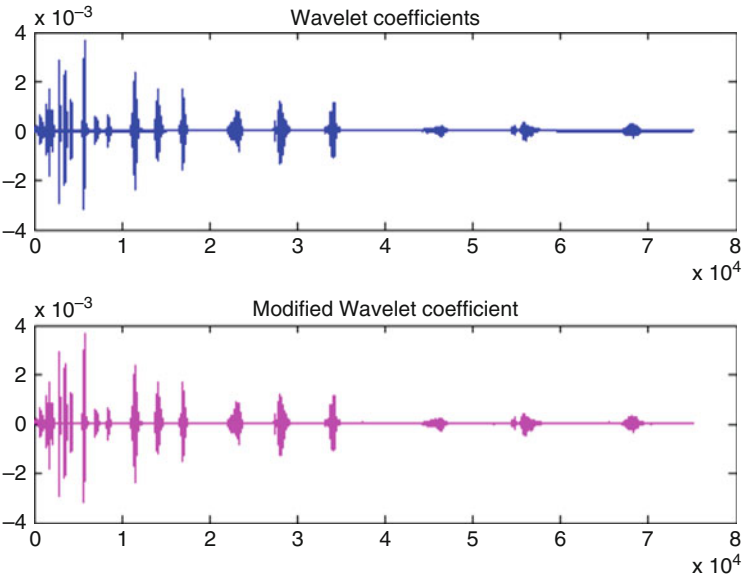

Fig. 6.31 Original and modified wavelet coefficients for ninth band enhancement

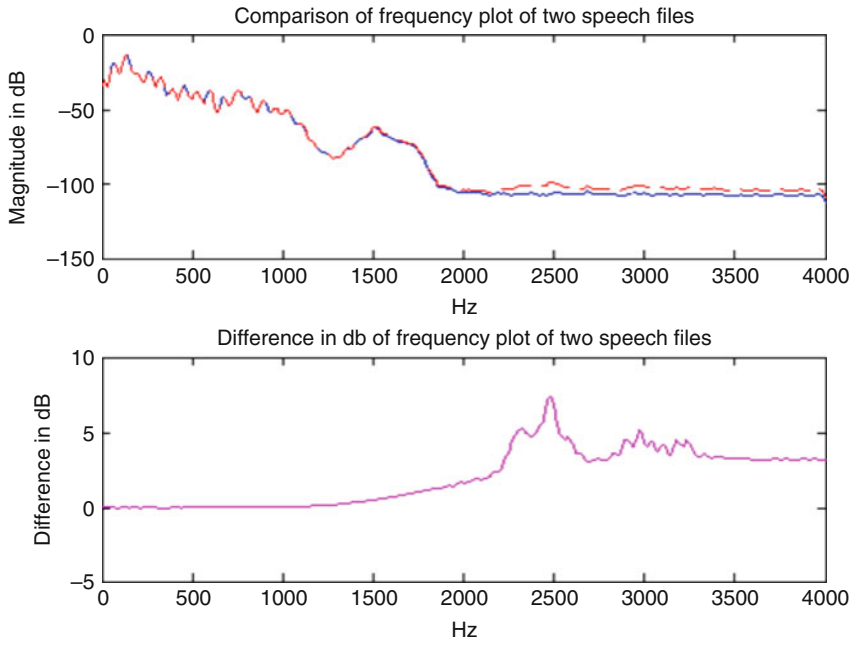

Fig. 6.32 Ninth band enhancement of 3 dB using multiplying factor 1.21

References

1. Agbinya, J. I. (1996, November). Discrete wavelet transform techniques in speech processing. In *TENCON'96. Proceedings, 1996 I.E. TENCON. Digital Signal Processing Applications* (Vol. 2, pp. 514–519).
2. Tian, G., Shuli, Y., & Datian, Y. (2006, January). Application of wavelet in speech processing of cochlear implant. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the IEEE* (pp. 5339–5342).
3. Gold, B., Morgan, N., & Ellis, D. (2011). *Speech and audio signal processing: processing and perception of speech and music*. New York: Wiley.
4. Resnikoff, H. L., & Raymond Jr., O. (2012). *Wavelet analysis: the scalable structure of information*. New York: Springer.
5. Rao, R. M. (1998). Wavelet transforms: Introduction to theory and applications. Pearson Education India.

Chapter 7
Summary of This Book and Future Research Directions

7.1 Important Points Covered in the Book

A solution for deafness using digital hearing aids is a very challenging task in the present environment. Noise reduction and separate frequency band enhancements in decibels are important requirements of hearing aids nowadays. Three adaptive algorithms have been chosen to reduce noise: least mean squares (LMS), normalized LMS (NLMS), and recursive least squares (RLS).

In the LMS and NLMS algorithms, the mean values of the filter coefficients converge toward their optimal solutions. Therefore, the filter coefficients will fluctuate about their optimum values. The amplitude of the fluctuation is controlled by the step size. The smaller the step size, the smaller the fluctuations and the less final maladjustment occurs, but also the slower the adaptive coefficients converge to their optimal values. The resources required to implement the LMS algorithm require a transversal adaptive finite impulse response (FIR) filter of coefficients in real time.

The resources required to implement the NLMS algorithm for a transversal adaptive FIR filter in terms of the number of coefficients are the same as those needed for LMS, but the computational complexity is greater at every step and are required to normalize the mu vector for frame samples. In NLMS, because of the normalized feature, the rate of convergence is very fast so the convergence curve reaches a steady-state value in less time. In real time, NLMS requires more time as divisions are required at each and every step, so it consumes the time of the processor. Compared to least mean squares algorithms, RLS algorithms have a faster convergence speed and do not exhibit the eigenvalue spread problem. However, RLS algorithms involve more complicated mathematical operations and require more computational resources than do the LMS and NLMS algorithms.

In a listening test, the NLMS algorithm removes all the noise but the output is throttled; however, increments in filter length can remove this problem, although for the same filter length and step size LMS gives the best performance for real-time speech. Compared to LMS, NLMS can give better performance if some adjustment

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6_7

in filter length is provided, at a cost of greater computational complexity, more memory storage, and more processing time.

In the RLS algorithm, the noise is not completely removed. The noise residue is present in the output, but improvement can be achieved by making the step size very small. RLS with very small step sizes can give the best performance. In the objective test discussed, LMS gives better performance with respect to MSE and peak signal-to-noise ratio (PSNR) compared to all other adaptive algorithms, but with a slow convergence rate compared to RLS and NLMS with a higher value of SNR.

The voice activity detection (VAD) algorithm, based on energy detection and zero crossing, is also very efficient to recognize silences from the speech and improves PSNR by using wavelet thresholding for the pauses. Wavelet thresholding for silence cleaning gives better performance in the system. By taking more levels of resolution of wavelet and by hard thresholding, the result is satisfactory.

Converting cleaned speech in the frequency domain in separate frequency bands can be accomplished using the wavelet transform multi-resolution approach. Precession can be taken up to 0.5 kHz in the speech signal by taking only four levels of MRA. By multiplying different values of integer with coefficients of wavelet in the frequency domain, loudness control of separate frequency bands can be achieved. In extracting by implementing the algorithms, speech enhancement of separate frequency bands as well as noise reduction can be accomplished for digital hearing aids. Per the necessity of frequency response of the ear, individual bands can be modified, and proper noise-free enhanced speech can be prepared as the output of the hearing aids.

7.2 Future Research Direction

The presented simulation results, if converted into simulation blocks using user-defined *s* block parameters, can be directly downloaded to digital signal processing (DSP) for a real-time approach. For greater improvement of the noise reduction phenomena, nonlinearity of the systems may be included. By considering nonlinearity, the type of filter used must be changed, with some minor modifications. The new nonlinear adaptive filter of the Volterra series may improve the statistical parameters of the system overall. Moreover, for more noise reduction, a microphone array might be used, with different types of directivity for each one. Thus, sound can be captured from many directions, and consequently the statistics of noise prediction becomes strong and very accurate noise reduction may be possible.

Index

K. R. Borisagar et al., *Speech Enhancement Techniques for Digital Hearing Aids*,
https://doi.org/10.1007/978-3-319-96821-6