Global Journal of Researches in Engineering: F
Electrical and Electronics Engineering
Volume 14 Issue 1 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

New Delay Less Sub Band Adaptive Filtering Algorithm for Active Noise Control Systems

By B. Satish Chandra, S. China Venkateswarlu, D. Ravi Kiran Babu & K. Arun Kumar

Jyothishmathi Institute of Technology & Science, India

Abstract- The delay less SAF scheme in an ANC system involves the decomposition of input noise (i.e., the reference signal) and error signals into sub bands using analysis filter banks, and combining the sub band weights into a full-band noise canceling filter by a synthesis filter bank called weight stacking. Typically, a linear-phase finite-impulse response (FIR) low-pass filter (i.e., prototype filter) is designed and modulated for the design of such filter banks. The filter must be designed so that the side-lobe effect and spectral leakage are minimized. The delay in filter bank is reduced by prototype filter design and the side-lobe distortion is compensated for by oversampling and appropriate stacking of sub band weights. Experimental results show the improvement of performance and computational complexity of the proposed method in comparison to two commonly used sub band and block adaptive filtering algorithms.

Keywords: DFT, dolph-linear-phase finite-impulse response, sub band adaptive filtering, SAF, UDFM.

GJRE-F Classification : FOR Code: 091301, 090699

NEWDELAYLESSSUBBANDADAPTIVEFILTERINGALGORITHMFORACTIVENOISECONTROLSYSTEMS

New Delay Less Sub Band Adaptive Filtering Algorithm for Active Noise Control Systems

B. Satish Chandra [α], S. China Venkateswarlu [σ], D. Ravi Kiran Babu [ρ] & K. Arun Kumar [ω]

Abstract- The delay less SAF scheme in an ANC system involves the decomposition of input noise (i.e., the reference signal) and error signals into sub bands using analysis filter banks, and combining the sub band weights into a full-band noise canceling filter by a synthesis filter bank called weight stacking. Typically, a linear-phase finite-impulse response (FIR) low-pass filter (i.e., prototype filter) is designed and modulated for the design of such filter banks. The filter must be designed so that the side-lobe effect and spectral leakage are minimized. The delay in filter bank is reduced by prototype filter design and the side-lobe distortion is compensated for by oversampling and appropriate stacking of sub band weights. Experimental results show the improvement of performance and computational complexity of the proposed method in comparison to two commonly used sub band and block adaptive filtering algorithms. Sub band adaptive filtering (SAF) techniques play a prominent role in designing active noise control (ANC) systems. They reduce the computational complexity of ANC algorithms, particularly, when the acoustic noise is a broadband signal and the system models have long impulse responses. In the commonly used uniform-discrete Fourier transform (DFT) -modulated (UDFTM) filter banks, increasing the number of sub bands decreases the computational burden but can introduce excessive distortion, degrading performance of the ANC system. In this paper, we propose a new UDFTM-based adaptive sub band filtering method that alleviates the degrading effects of the delay and side-lobe distortion introduced by the prototype filter on the system performance.

Keywords: DFT, dolph-linear-phase finite-impulse response, sub band adaptive filtering, SAF, UDFM.

I. Introduction

Subband adaptive filtering (SAF) techniques play a prominent role in designing active noise control (ANC) systems. They reduce the computational complexity of ANC algorithms, particularly, when the acoustic noise is a broadband signal and the system models have long impulse responses. Active noise control (ANC) is a method of canceling a noise signal in an acoustic cavity by generating an appropriate anti-noise signal via canceling loudspeakers.

In general, the SAF methods offer a good alternative approach to meet ANC system requirements, due to their inherent spectral decomposition and down sampling operations. Since the spectral dynamic range and eigen value spread of the covariance matrix of noise signal decrease in each sub band, the performance, i.e., convergence rate, noise attenuation level, and stability of the ANC system, improves using SAF techniques. Hence, one expects that increasing the number of sub bands (or block length) M should improve the performance.

The delay less SAF scheme in an ANC system involves the decomposition of input noise (i.e., the reference signal) and error signals into sub bands using analysis filter banks, and combining the sub band weights into a full-band noise canceling filter by a synthesis filter bank called weight stacking. Typically, a linear-phase finite-impulse response (FIR) low-pass filter (i.e., prototype filter) is designed and modulated for the design of such filter banks. The filter must be designed so that the side-lobe effect and spectral leakage are minimized. The latter requires a high-order FIR filter, introducing a long delay, which increases with M as the bandwidth shrinks. The long delay and side-lobe interference introduced by the prototype filter degrade the performance of SAF algorithms for large M, limiting the computational saving that can be obtained by increasing the number of sub bands. Improving the system performance and reducing the computational burden by increasing M has inspired the work presented herein. The focus of this project is to design a high-performance SAF algorithm. The performance limiting factors of existing SAF structures were found to be due to the inherent delay and side-lobes of the prototype filter in the analysis filter banks.

A delay less structure targeted for low-resource implementation is proposed to eliminate filter bank processing delays in sub band adaptive filters (SAFs). Rather than using direct IFFT or poly phase filter banks to transform the SAFs back into the time-domain, the proposed method utilizes a weighted overlap-add (WOLA) synthesis. Low-resource real-time implementations are targeted and as such do not involve long (as long as the echo plant) FFT or IFFT operations. Also, the proposed approach facilitates time distribution of the adaptive filter reconstruction calculations crucial for efficient real-time and hardware implementation. The method is implemented on an

Author α: Assoc. Prof-ECE JITS Jyothishmathi Institute of Technology & Science, Nustulapur, Karimnagar-505 481, Andhra Pradesh, India.
Author σ: Prof. & HOD-ECE JITS Jyothishmathi Institute of Technology & Science, Nustulapur, Karimnagar-505 481, Andhra Pradesh, India. e-mail: hod.ece@jits.ac.in
Author ρ: Assoc. Prof-ECE JITS Jyothishmathi Institute of Technology & Science, Nustulapur, Karimnagar-505 481, Andhra Pradesh, India.
Author ω: Asst. Prof-ECE SCITES Jyothishmathi Institute of Technology & Science, Nustulapur, Karimnagar-505 481, Andhra Pradesh, India.

oversampled WOLA filter bank employed as part of an echo cancellation application. Evaluation results demonstrate that the proposed implementation outperforms conventional SAF systems since the signals used in actual adaptive filtering are not distorted by filter bank aliasing. The method is a good match for partial update adaptive algorithms since segments of the time-domain adaptive filter are sequentially reconstructed and updated.

A delay less method for adaptive filtering through SAF systems is proposed. The method, based on WOLA synthesis of the SAFs, is very efficient and is well mapped to a low-resource hardware implementation. The performance of an open-loop version of the system was compared against a conventional SAF system employing the same WOLA analysis/synthesis filter banks, with the proposed delay less system offering superior performance but at greater computational cost. The performance is identical to the DFT-FIR delay less SAF system that employs straightforward poly phase filter banks.

However, the proposed WOLA-based TAF synthesis offers a superior mapping to low-resource hardware with limited-precision arithmetic. Also the WOLA adaptive filter reconstruction may easily be spread out in time simplifying the necessary hardware. This time-spreading may be easily combined with partial update adaptive algorithms to reduce the computation cost for low-resource real time platforms.

II. Generic Invertibility of Multidimensional Fir Multirate Systems and Filter Banks

We study the inevitability of M-variety polynomial (respectively: Laurent polynomial) matrices of size N by P. Such matrices represent multidimensional systems in various settings including filter banks, multiple-input multiple-output systems, and MultiMate systems. The main result of this paper is to prove that when $N - P \geq M$, then $H(z)$ is generically invertible; whereas when $N - P < M$, then $H(z)$ is generically noninvertible. As a result, we can have an alternative approach in design of the multidimensional systems. During the last two decades, one dimensional multiage systems in digital signal processing were thoroughly developed. In recent years, due to the high demand in multidimensional processing including image and video processing, volumetric data analysis and spectroscopic imaging, multidimensional multirate systems have been studied more extensively.

One key property of a multidimensional multirate system is its perfect reconstruction, which guarantees that an original input can be perfectly reconstructed from the outputs. We show that there is a sharp phase transition on the invariability depending on the size and dimension of a given Laurent polynomial matrix. Specifically when $N - P \geq M$, the $N \times P$ polynomial (resp. : Laurent polynomial) of M-variety matrix is generically invertible; whereas when $N - P < M$, the matrix is generically noninvertible. Using this sharp phase transition property, we develop a fast algorithm to compute a particular left inverse for a given Laurent polynomial matrix. These results suggest an alternative approach in designing multidimensional filter banks by freely generating filters for the analysis side first. If we allow an amount of over sampling then we can almost surely find a perfect reconstruction inverse for the synthesis poly phase matrix. These results also have potential applications in multidimensional signal reconstruction from multi-channel filtering and sampling. Speech signals from the uncontrolled environments may contain degradation components along with the required speech components. The degradation components include background noise, reverberation and speech from other speakers. The degraded speech gives poor performance in automatic speech processing tasks like speech recognition and speaker recognition and is also uncomfortable for human listening [1]. The degraded speech therefore needs to be processed for the enhancement of speech components. Several methods have been proposed in the literature for this purpose, majority them can be grouped into spectral processing and temporal processing methods. In spectral processing methods, the degraded speech is processed in the transform domain, where as, in temporal processing methods, the processing is done in the time domain, for enhancing the speech components. Each of them has their own merits and demerits. These two approaches may be effectively combined by exploiting their merits and aiming to minimize the demerits. This may lead to speech enhancement methods which are more effective and robust compared to only spectral or temporal processing.

Frequency-domain and sub band implementations improve the computational efficiency and the convergence rate of adaptive schemes. The well-known multi delay adaptive filter (MDF) belongs to this class of block adaptive structures and is a DFT-based algorithm. In this paper, we develop adaptive structures that are based on the trigonometric transforms DCT and DST and on the discrete Hartley transform (DHT). As a result, these structures involve only real arithmetic and are attractive alternatives in cases where the traditional DFT-based scheme exhibits poor performance. The filters are derived by first presenting a derivation for the classical DFT-based filter that allows us to pursue these extensions immediately. The approach used in this paper also provides further insights into sub band adaptive filtering.

III. The Implementation of Delay Less Sub Band Active Noise Control Algorithms

Wideband active noise control systems usually have hundreds of taps for control filters and the cancellation path models, which results in high computational complexity and low convergence speed. Several active noise control algorithms based on sub band adaptive filtering have been developed to reduce the computational complexity and to increase the convergence speed. The sub band structure is similar to the frequency domain structure but differs in the time domain processing of the sub band signals. This paper discusses several issues associated with implementing the delay less sub band active noise control algorithms on a DSP Platform, such as the modeling of the cancellation path in sub bands and the partial Update of different sub bands.

Single channel ANC systems often use sub band techniques to overcome the difficulties of high computational complexity and low convergence speed associated with a wideband control filter containing thousands of taps.

This paper will discuss various method of noise reduction for wireless communication network. Noise is an, unwanted and inevitable interference, in any form of communication. It is non-informative and plays the role of sucking the intelligence of the original signal. Any kind of processing of the signal contributes to the noise addition. A signal traveling through the channel also gathers lots of noise. It degrades the quality of the information signal. The effect of noise could be reduced only at the cost of the bandwidth of the channel, which is again undesired, as bandwidth is a precious resource. Hence to regenerate original signal, it is tried to reduce the power of the noise signal, or in the other way, raise the power level of the Informative signal, at the receiver end this leads to improvement in the signal to noise ratio(SNR).

Adaptive algorithms that allow neighboring nodes to communicate with each other at every iteration. At each node, estimates exchanged with neighboring nodes are fused and promptly fed into the local adaptation rules. In this way, an adaptive network is obtained where the structure as a whole is able to respond in real-time to the temporal and spatial variations in the statistical profile of the data. Different adaptation or learning rules at the nodes, allied with different cooperation protocols, give rise to adaptive networks of various complexities and potential. Obviously, the effectiveness of any distributed implementation depends on the modes of cooperation that are allowed among the nodes. Figure 1 illustrates three such modes of cooperation. In an incremental mode of cooperation information flows in sequential manner from one node to the adjacent node. This mode of operation requires a cyclic pattern of collaboration among the nodes, and has the advantage that for the last node in the cycle, the data from the entire network are used to update the desired parameter estimate, thereby offering excellent estimation performance.

Moreover, for every measurement, every node needs to communicate with only one neighbor. However, incremental cooperation has the disadvantage of requiring the definition of a cycle, and network processing has to be faster than the measurement process, since a full communication cycle is needed for every measurement. This may become prohibitive for large networks. Incremental networks are also less robust to node and link failures. An alternative protocol is the diffusion implementation where every node communicates with all of its neighbors as dictated by the network topology. This approach has no topology constraints and is more robust to node and link failure. It will have some performance degradation compared to an incremental solution, and also every node will need to communicate with its neighbors for every measurement, possibly requiring more energy than the incremental case.

The mainstay of the proposed model is improving the system performance and reducing the computational burden. In this paper, we first demonstrate that the increased delay degrades the system performance more than that of the spectral leakage (or side-lobe effects) in a uniform sub-band filtering method. It is shown how the spectral leakage can be reduced by choosing a proper decimation factor and weight stacking methodology. We then present a new SAF (Sub-Band Adaptive Filtering) algorithm that reduces computational complexity by increasing the number of subbands M without degrading the performance of the ANC (Active Noise Control) system. The performance of the proposed method is compared with those of MT (Moragan and Thi) and DFT-MDF (Discrete Fourier Transform and Multi-Delay Adaptive Filter) methods. The results show that the maximum noise attenuation level (NAL) of the proposed method is higher than that of MT and comparable to that of the DFT-MDF method. However, the new method achieves the maximum NAL with much lower computational complexity and higher robustness than the other two methods.

IV. Methodology

The gradient based adaptation starts with an old optimization technique known as the method of steepest descent. This has been discussed in the next chapter in detail. It is recursive in the sense that starting from some initial arbitrary value for tap weight vector, it improves with increasing number of iterations. The final value so computed for tap weight vector converges to Wiener solution. The fixed step size least mean square (FSS LMS) algorithm is an important member of the family of stochastic gradient algorithms. The term stochastic gradient is intended to distinguish it from the method of steepest descent that uses deterministic gradient in a recursive computation of the Wiener filter for stochastic inputs. This algorithm does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. Subsequent works have discussed issue of optimization of step size or methods of varying step size to improve performance. There are different types of adaptive filtering algorithms, they are 1. Least mean square (LMS) algorithm. 2. Normalized least mean square (NLMS) algorithm. 3. Variable step size LMS (VSLMS) algorithm. 4. Variable step size Normalized LMS (VSNLMS) algorithm. 5. Recursive least squares (RLS) algorithm.

a) Normalized least mean Square (NLMS) Algorithm

One of the primary disadvantages of the LMS algorithm is having a fixed step size parameter for every iteration. This requires an understanding of the statistics of the input signal prior to commencing the adaptive filtering operation. In practice this is rarely achievable. Even if we assume the only signal to be input to the adaptive echo cancellation system is speech, there are still many factors such as signal input power and amplitude which will affect its performance.

The normalized least mean square algorithm (NLMS) is an extension of the LMS algorithm which bypasses this issue by selecting a different step size value, $\mu(n)$, for each iteration of the algorithm. This step size is proportional to the inverse of the total expected energy of the instantaneous values of the coefficients of the input vector **x**(n). This sum of the expected energies of the input samples is also equivalent to the dot product of the input vector with itself, and the trace of input vectors auto-correlation matrix, **R**.

$$tr(\boldsymbol{R}) = \sum_{i=0}^{N-1} E[x^2(n-i)]$$

$$= E[\sum_{i=0}^{N-1} x^2(n-i)]$$

The recursion formula for the NLMS algorithm is stated in equation.

$$w(n+1)=w(n)+\frac{1}{x^T(n)x(n)}e(n)x(n)$$

b) Derivation of the NLMS algorithm

To derive the NLMS algorithm consider the standard LMS recursion, for which we select a variable step size parameter, $\mu(n)$. This parameter is selected so that the error value, $e^+(n)$, will be minimized using the updated filter tap weights, w(n+1), and the current input vector, **x**(n). $w(n+1) = w(n) + 2\mu(n)e(n)x(n)$, $e^+(n) = d(n) - w^T(n+1)x(n)$, $=(1- 2\mu(n)x^T(n)x(n))e(n)$

Next we minimize $(e^+(n))^2$, with respect to $\mu(n)$. Using this we can then find a value for $\mu(n)$ which forces $e^+(n)$ to zero.

$$\mu(n) = \frac{1}{2x^T(n)x(n)}$$

This $\mu(n)$ is then substituted into the standard LMS recursion replacing μ, resulting in the following NLMS equation.

$$w(n+1) = w(n) + 2\mu(n)e(n)x(n) ,$$

$$w(n+1)=w(n)+\frac{1}{x^T(n)x(n)}e(n)x(n)$$

c) Implementation of the NLMS algorithm

The NLMS algorithm has been implemented in Matlab and in a real time application using the Texas Instruments TMS320C6711 Development Kit. As the step size parameter is chosen based on the current input values, the NLMS algorithm shows far greater stability with unknown signals. This combined with good convergence speed and relative computational simplicity makes the NLMS algorithm ideal for the real time adaptive echo cancellation system. As the NLMS is an extension of the standard LMS algorithm, the NLMS algorithms practical implementation is very similar to that of the LMS algorithm. Each iteration of the NLMS algorithm requires these steps in the following order

(a) The output of the adaptive filter is calculated.

$$y(n) = \sum_{i=0}^{N-1} w(n)x(n-i) = w^T(n)x(n)$$

(b) An error signal is calculated as the difference between the desired signal and the filter output

$$e(n)=d(n)-y(n)$$

(c) The step size value for the input vector is calculated.

$$\mu(n) = \frac{1}{2x^T(n)x(n)}$$

(d) The filter tap weights are updated in preparation for the next iteration.

$$w(n+1) = w(n) +\mu(n)e(n)x(n)$$

Each iteration of the NLMS algorithm requires 3N+1 multiplications, this is only N more than the standard LMS algorithm, this is an acceptable increase considering the gains in stability and echo attenuation achieved.

V. Active Noise Control System

Active noise control (ANC) is a method of canceling a noise signal in an acoustic cavity by generating an appropriate anti-noise signal via canceling loudspeakers. Due to recent advances in wireless technology, new applications of ANC have emerged, e.g., incorporating ANC in cell phones, Bluetooth headphones, and MP3 players, to mitigate the environmental acoustic noise and therefore improve the speech and music quality. For practical purposes, ANC as a real-time adaptive signal processing method should meet the following requirements: 1) minimum computational complexity (lower computational delay and power consumption), 2) stability and robustness to input noise dynamics, and 3) maximum noise attenuation.

Acoustical noise can sometimes disturb or even harm nearby people. Hence, it is necessary to find ways to reduce such unwanted noise. Traditionally, passive means (i.e., physical barriers) to attenuate the noises have been employed. Unfortunately, the barriers are not effective to isolate lower frequency noises; and to achieve significant reduction the barriers have to be rather bulky. In effect, the passive barrier is not a cost-effective solution to reducing low-frequency noises (for example, noises that come from industrial blowers, diesel engines, transformers, earth-moving machines, and propeller-driven aircraft.) Because of that shortcoming of the physical barriers, active means to reduce low frequency noise (less than 500-1000 Hertz) have been investigated by researchers in the field of adaptive acoustic control. Active noise control (ANC) promises a good reduction of the noises in the form of a small package of a DSP controller, microphone(s), and loudspeaker(s). For the better or the worse, the ANC systems are effective only when the intended noise is periodic, and so random noises like the white noise will not be reduced.

There are different ANC schemes that have been developed. My project is involved with the implementation of one of the schemes that is called single-channel adaptive feedback ANC. The implementation was on a Texas Instruments TMS320C54 evaluation module (EVM) board; in addition to this, I used a microphone and a loudspeaker. Two types of noise exist in the environment, broadband noise, where its energy is more or less evenly distributed across the frequency spectrum, or narrowband noise, where the energy is mostly concentrated around specific frequencies. In ANC roughly two types of control strategies can be distinguished as shown by Fuller, their use strongly depends on the deterministic behavior of the disturbance:

Feedback ANC: A controller is used to modificate the response of a system, for example by adding artificial damping. In this way vibration levels can be reduced even for a broadband random disturbance.

Feedforward ANC: When the disturbance is deterministic, or in particular harmonic, a controller can be used to adaptively calculate a signal that cancels the disturbance. When vibrations are induced by rotating machinery this often results in harmonicvibrations and the amount of noise reduction achieved by feedforward ANC systems is far superior to that of feedback ANC systems as shown by Hansen & Snyder. The basic idea of feedforward ANC is to generate a signal (secondary noise), that is equal to a disturbance signal (primary noise) in amplitude and frequency, but has opposite phase. Combination of these signals results in cancellation of the primary (unwanted) noise. This ANC technique is well-known for its use in cancelling unwanted sound as shown by Nelson & Elliott [6], but it is used for the control of vibration. A block diagram of an adaptive digital filter is shown in fig.2.1, where n is a time index. This filter forms the basis for feedforward ANC, based on the FXLMS algorithm. The adaptive filter actually consists of two parts. The digital filter, W(z) calculates its output by using a reference x(n) and adjustable filter coefficients, or weights. The filter coefficients are updated by an adaptive algorithm, using x(n) and an error signal e(n) in such a way that the squared error $e^2(n)$ is minimized.

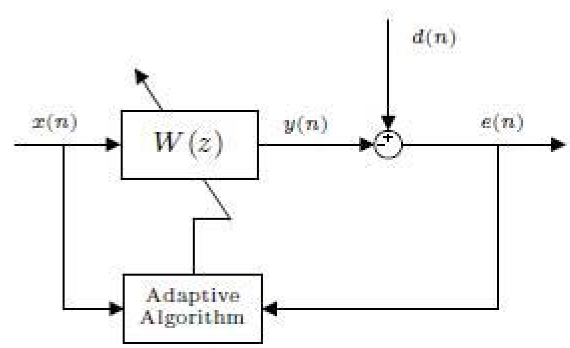

Figure 5.1 : ANC using an adaptive filter

We can define the error e(n) as:

$$e(n) = d(n) - y(n) \tag{5.1}$$

where d(n) is an unwanted disturbance. The adaptive filter will try to calculate an output y(n) that is equal to the unwanted disturbance d(n), so this disturbance will be cancelled.

a) *Concept of an ANC system*

The basic concept of the feedforward ANC system that is used with the experimental setup can be found in Figure 2.2, where the grey part represents the controller and the white part represents the physical world. This is a very general concept, in this report vibrations are considered, but it can also be applied to acoustic applications as shown by Nelson and Elliott [6] or more specific to sound cancellation in ducts as shown by Kuo and Morgan [5].

b) *System Description*

The harmonic noise is produced at the noise source (e.g. an engine or a shaker).

Through the transfer function P(z) of the primary path this results in a vibration d(n) somewhere in the construction. This vibration will be reduced, by generating the appropriate controller output y(n) and sending it trough the transfer function S(z) of the Secondary Path to the construction. The remaining vibration e(n) can then be measured by a sensor. The adaptive filter looks similar to that of Figure 2.1 but is slightly more complicated. That is to compensate for the effects of the Secondary Path, which will be explained later.

c) Conventional versus Indirect Feedforward ANC

In conventional feedforward ANC systems, the disturbance frequency information is available or can be derived from the noise source, for example from the engine velocity. When the disturbance frequency is exactly known, the reduction that can be achieved by a conventional feedforward ANC system has its limit at infinity for the ideal case with a pure harmonic noise-free disturbance and linear Secondary Path. In other applications the disturbance frequency information may not be available, because the disturbance frequencies are unknown or slowly varying. In that case indirect feedforward ANC can be used as shown in this report, where the reference signal x(n) is generated from the error e(n), instead of from the frequency information of the noise source. Conventional feedforward ANC with a single frequency disturbance was imple- mented on the experimental setup by H.J. van der Veen. This report focuses on different kinds of indirect feedforward ANC methods, where if possible harmonic disturbances with two frequencies are used. They are tested at the experimental setup and will be compared with each other.

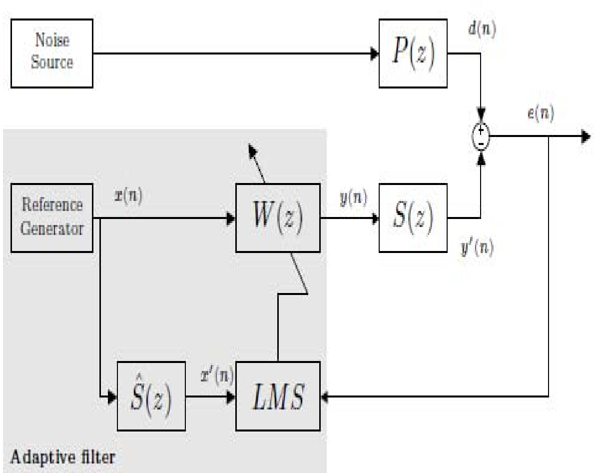

Figure 5.1 : Block diagram of an ANC system

In practical applications there is a transfer function S(z) between the digital controller signal and the physical world, which contains the D/A converter, power amplifier, actuator element and construction. In general, this Secondary Path transfer function S(z) gives a change in amplitude and a phase shift, so the adaptive filter should compensate for the effects of S(z) to ensure convergence. A straightforward solution would be to place the inverse $S(z)^{-1}$ in series with S(z), but because this inverse does not necessarily exists, the so-called Filtered-x LMS (FXLMS) algorithm is more generally used. This algorithm places an estimate of S(z) in the reference signal to the weight update.

For the ANC system of Figure 2.2, containing a Secondary Path transfer function S(z), the residual error can be expressed as:

$$e(n)=d(n)-y`(n); \qquad (5.2)$$

where y`(n) is the output of the Secondary Path S(z). If S(z) is assumed as an IIR filter with denominator coefficients $[a_1,\ldots\ldots\ldots,a_N]$ and numerator coefficients $[b_0,\ldots\ldots.b_{M-1}]$, then the filter output y`(n) can be written as the sum of the filter input y(n) and the past filter output:

$$y'(n) = \sum_{i=1}^{N} a_i y'(n-i) + \sum_{j=0}^{M-1} b_j y(n-j). \qquad (5.3)$$

It can be achieved in a similar way that the gradient estimate becomes:

$$\nabla\hat{\xi}(n) = -2\mathbf{x}'(n)e(n), \qquad (5.4)$$

where:

$$\mathbf{x}'(n) = \sum_{i=1}^{N} a_i \mathbf{x}'(n-i) + \sum_{j=0}^{M-1} b_j \mathbf{x}(n-j). \qquad (5.5)$$

Note that in practical applications, S(z) is not exactly known, therefore the parameters a_i and b_j are the parameters of the Secondary Path Estimate S^ (z). The weight update equation of the FXLMS algorithm is:

$$w(n+1) = w(n) + \mu x`(n)e(n) \qquad (5.6)$$

and x`(n) can be calculated from Equation 5.5.

The FXLMS algorithm is very tolerant to modelling errors in the Secondary Path Estimate S^ (z) as shown by Kuo & Morgan [5]. The algorithm will converge when the phase error between S(z) and S^ (z) is smaller than 90^0. Convergence will be slowed down though, when the phase error increases. From the weight update Equation 2.6 can be seen that a step size μ has to be chosen. This step size affects important properties such as performance, stability and error after convergence. A more in-depth analysis can be found in Kuo & Morgan [5] and Elliott & Nelson. Furthermore, a modification of the standard FXLMS is presented to make the choice of μ independent of the power of x`(n).

VI. System Analysis

a) Adaptive Filter Theory

In the past three decades, interest in adaptive systems has increased, leading to widespread use of adaptive techniques in fields such as Communications, Signal Processing, Sonar and Biomedical Engineering.

Adaptive systems adapt to the environment changes and search for the optimal system parameters based on a reference signal. In the case of a filter, the system parameters are the tap weights of the filter. The performance of an adaptive algorithm is highly dependent on the reference input and additive noise statistics. In the context of Wiener filter theory, there are assumptions of time invariance, linearity and Gaussian statistics such that the mean square error criteria will be the optimum cost function. These assumptions are often for the ease of mathematical analysis, but do not take into account of the broader problems of signals with non-Gaussian statistics. In the digital communication systems, efficient bandwidth utilization is economically important to maximizing profits, while at the same time maintaining performance and reliability. More importantly, the adaptive filter solution has to be relatively simple, which often leads to the use of the conventional Least Mean Square (LMS) algorithm. However, the performance of the LMS algorithm is often sub-optimal and the convergence rate is small. This, therefore, provides the motivation to explore and study variable step size LMS adaptive algorithms for various applications.

b) The Wiener Filter

These are a class of linear optimum discrete time filters known collectively as Wiener filters. Wiener filters are a special class of transversal Finite impulse response (FIR) filters that build upon the Mean Square Error (MSE) cost function to arrive at an optimal filter tap weight vector, which reduces the MSE signal to a minimum. Theory for a Wiener filter is formulated for general case of complex valued time series with filter specified in terms of its impulse response because baseband signal appears in complex form under many practical situations.

c) Mean Square Error Criterion

The linear filter with the aim of estimating the desired signal d(n) from input x(n). Assume that d(n) and x(n) are samples of infinite length, random processes illustrates in Fig 3.1 . In 'optimum filter design', signal and noise are viewed as stochastic processes. The filter is based on minimization of the mean square value of the difference between the actual filter output and some desired output, as shown in fig.3.1.

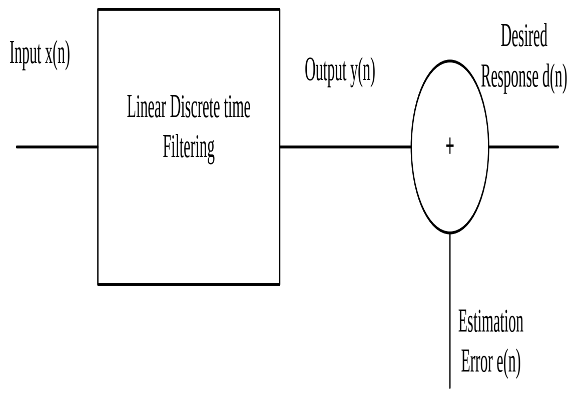

Figure 6.1 : Prototype Wiener Filtering Scheme

The requirement is to make the estimation error as small as possible in some Statistical sense by controlling impulse response coefficients w_0, w_1, ……., w_{N-1} . Two basic restrictions are: 1. The filter is linear, which makes mathematical analysis easy to handle.2. The filter is an FIR (symmetrical and odd ordered) filter.

The filter output is y(n) and the estimation error is given by e(n). The performance of the filter is determined by the size of the estimation error, that is, a smaller estimation error indicates a better filter performance. As the estimation error approaches zero, the filter output y(n) approaches the desired signal d(n). Clearly, the estimation error is required to be as small as possible. In simple words, in the design of the filter parameters, an appropriate function of this estimation error as performance or cost function is chosen and the set of filter parameters is selected, which optimizes the cost function. In Wiener filters, the cost function is chosen to be

$$x = E[e(n)^2] \qquad (6.1)$$

Where E[.] denotes the expectation or ensemble average since both d(n) and x(n) are random processes.

d) Wiener Filter: Transversal, Real valued case

Consider an adaptive transversal filter as shown in Fig 3.2. Assume that the filter input x(n) and the desired response d(n) are real valued stationary processes. The filter tap weights w_0, w_1,…………w_{N-1} are also assumed to be real valued , where N equals the number of delay units or tap weights.

The filter input x(n) and tap weight vectors, w, can be defined as column vectors,

$$\begin{aligned} x(n) &= [\,x(n)\ \ x(n-1)\ldots\ldots x(n-N+1)] \\ w &= [w_0\ \ w1 \ldots\ldots\ldots w_{N-1}]^{T} \end{aligned} \qquad (6.2)$$

The filter output is defined as

$$y(n) = \sum_{i=0}^{n} = w_i x(n-i) = w^T x(n) = x^T(n) w(n) \qquad (6.3)$$

Subsequently, the error signal can be written as

$$e(n) = d(n)\text{-}y(n) = d(n) - w^T x(n) = d(n) - x^T(n)w \qquad (6.4)$$

Substituting (3.5) into (3.1) , the cost function is obtained as,

$$E[(e(n)^2] = E[(d(n)\text{-}\ w^T x(n))\ (d(n)\text{-}x^T(n)w)] \qquad ((6.6)$$

Expanding the last expression of (6.6) we obtain,

$$E[\ d(n)^2] - E[\ d(n)x^T(n)w] - E[d(n)\ w^T x(n)] + E[w^T x(n)\ x^T(n)w] \qquad (6.7)$$

Since w is not a random variable,

$$E[d(n)^2] - E[\ d(n)x^T(n)]w - w^T E[d(n)\ x(n)] + w^T E[\ x(n)\ x^T(n)]w \qquad (6.8)$$

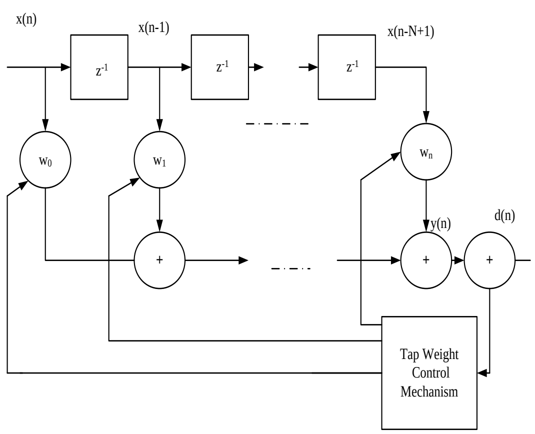

Figure 6.1 : Structure of an Adaptive Transversal Filter

Next, we can express E[d(n) x(n)] as an N x 1 cross correlation vector

$$p = E[d(n)x(n)] = [p_0, p_1, \ldots\ldots\ldots\ldots p_{N-1}]^T \qquad (6.9)$$

And $E[x(n)\ x^T(n)]$ as a N x N autocorrelation matrix R

$$R = E[x(n)x^T(n)] =$$

$$\begin{bmatrix} r_{00} & r_{01} & r_{02} & \cdots & r_{0,N-1} \\ & \vdots & & \ddots & \vdots \\ r_{N-1,0} & r_{N-1,1} & & \cdots & r_{N-1,N-1} \end{bmatrix} \qquad (6.10)$$

From (6.9), $p^T w = E[d(n)\ x^T(n)]$ and hence $p^T w = w^T p$ This implies that $E[d(n)\ x^T(n)]w = E[d(n)x(n)]w^T$. Subsequently, we get

$$\xi(w) = E[d(n)^2] - E[\ d(n)x^T(n)]w - w^T E[d(n)\ x(n)] + w^T E[\ x(n)\ x^T(n)]w, \quad = E[d(n)^2] - 2p^T w + w^T R\, w \qquad (6.11)$$

This is a quadratic function of tap weight vector 'w' with a single global minimum. To obtain the set of filter tap weights that minimizes the cost function, ξ solve the system of equations that results from setting the partial derivatives of $\xi(w)$ with respect to every tap weight of the filter i.e. the gradient vector to zero. That is

$$\frac{\partial \xi}{\partial w_i} = 0 \qquad (6.12)$$

For i = 0, 1, ...N-1 where N = number of tap weights The gradient vector in (3.12) can also be expressed as $\nabla\xi = 0$ (6.13)

Where Ñ is the gradient operator defined as column vector

$$\nabla = \left[\frac{\partial \xi}{\partial w_1} \quad \frac{\partial \xi}{\partial w_2} \quad \frac{\partial \xi}{\partial w_{N-1}}\right] \qquad (6.14)$$

and 0 on the right hand side of (3.13) denotes the column vector consisting of N zero. It has been further proved that the partial derivatives of x with respect to the filter tap weights can be solved such that

$$\nabla\xi = 2Rw - 2p \qquad (6.15)$$

By letting $\nabla\xi = 0$, the following equation is obtained, in which the optimum set of Wiener filter tap weights can be obtained, $Rw = p$ This implies that

$$w = R^{-1}p = w_0 \qquad (6.16)$$

Where w_0 indicates the optimum tap weight vector. This equation is known as the Wiener Hopf equation and can be solved to obtain the tap weight vector, which corresponds to the minimum point of the cost function.

e) Iterative Search Algorithm

It has been shown in the previous section that the Wiener Hopf equation can be solved to obtain the optimum filter tap weights by minimizing a cost function, if the required statistics of the underlying signals 'R' and 'p' are available. Although this method is straightforward, it presents serious computational difficulties, especially when the filter contains a large number of tap weights and the input data rate is high. An alternative is to use an iterative search algorithm that starts at some arbitrary initial point in the tap weight vector space and moves progressively towards the optimum filter tap weight vector in steps. Each step is chosen with the aim of reducing the cost function. The principle of finding the optimum filter tap weight vector by progressive minimization of the underlying cost function by means of an iterative algorithm is central to the development of adaptive algorithms (e.g. LMS). In simplified terms, adaptive algorithms are actually iterative search algorithms derived for minimizing the cost function by replacing the true statistics with estimates obtained.

f) Method of Steepest Descent

Assume that the cost function to be minimized is convex (If the cost function corresponds to a convex quadratic surface, it has a unique minimum point. In other words, when the cost function is convex, the iterative search algorithm is guaranteed to converge to the optimum solution), we may start with an arbitrary point on the performance surface and take a small step in the direction in which the cost function decreases fastest. This corresponds to a step along the steepest descent slope of the performance at that point. Repeating this successively, convergence towards the bottom of the performance surface (corresponding to the set of parameters that minimize the cost function) is guaranteed.

The method of steepest descent is an alternate iterative search method to find w_0 (in contrast to solving the Wiener Hopf equation directly). The method of steepest descent algorithm belongs to a family of iterative methods of optimization. It is a general scheme that performs an iterative search for a minimum point of any convex function of a set of parameters. Here, this method is implemented in transversal filter with the convex function referring to the cost function and the set of parameters referring to the filter tap weights. It uses the following procedures to search the minimum point of the cost function of a set of filter tap weights.

a) Begin with an initial guess of the filter tap weights whose optimum values are to be found for minimizing the cost function. Unless some prior knowledge is available, the search can be initiated by setting all the filter tap weights to zero, i.e. w(0).
b) Use this initial guess to compute the gradient vector of the cost function with respect to the tap weights at the present point.
c) Update the tap weights by taking step in the opposite direction (sign change) of the gradient vector obtained in step 2. This corresponds to step in the direction of the steepest descent in the cost function at the present input. Furthermore, the size of the step is chosen proportional to the size of the gradient vector.
d) Go back to Step 2, and iterate the process until no further significant change is observed in the tap weights i.e. the search has converged to an optimal point.

According to the above procedures, if w(n) is the tap weight vector at the nth iteration, then the following recursive equation may be used to update w(n).

$$w(n+1) = w(n) - \mu\Delta_n\xi \tag{6.17}$$

Where μ is the positive scalar called step size, and $\Delta_n\xi$ denotes the gradient vector evaluated at the point w = w(n).

g) Error Performance Surface

The estimation error e(n) can be given as:

$$E(n) = d(n) - \textstyle\sum_{i=0}^{N-1} w_i x(n-i) \tag{6.18}$$

The cost function can be written as

$$\xi = E[d(n)^2] - \textstyle\sum_{i=0}^{N-1} w_i^* E[x(n-i)d^*(n)] \tag{6.19}$$

The cost function or the mean squared error is precisely a second order function of the tap weights in the filter. Since 'w' can assume a continuum of values in the N dimensional w-plane, the dependence of the cost function depends on the tap weights w_0, w_1, ………w_{N-1} may be visualized as a bowl shaped (N+1)-dimensional surface with N degrees of freedom represented by the tap weights of the filter. The surface so described is called the error performance surface of the transversal filter. The surface is characterized by a unique minimum, where the cost function ξ attains its minimum value. At this point, the gradient vector$\Delta\ \xi$ is identically zero. The height ξ corresponds to the physical description of filtering the signal x(n-i) with the fixed filter weight w, from which a prediction error signal e(n) with power of ξ is generated. Some filter setting $w_0=(w_{o0,wo1})$ will produce the minimum MSE (w_o is the optimum filter tap weight vector). This theory is the base of basic adaptive algorithms of adaptive signal processing. The gradient based adaptation starts with an old optimization technique known as the method of steepest descent. It is recursive in the sense that starting from some initial arbitrary value for tap weight vector, it improves with increasing number of iterations. The final value so computed for tap weight vector converges to Wiener solution.

The LMS algorithm has been extensively analyzed in literature and a large number of results on its steady state misadjustment and tracking performance have been obtained. The fixed step size least mean square (FSS LMS) algorithm is an important member of the family of stochastic gradient algorithms. The term 'stochastic gradient' is intended to distinguish it from the method of steepest descent that uses deterministic gradient in a recursive computation of the Wiener filter for stochastic inputs. This algorithm does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. Subsequent works have discussed issue of optimization of step size or methods of varying step size to improve performance.

h) Performance of an Adaptive Algorithm

The factors that determine the performance of an algorithm are clearly stated below. Essentially, the most important factors as described here *1. Rate of Convergence:* This is defined as the number of iterations required for the algorithm to converge to its steady state mean square error. The steady state MSE is also known as the Mean asymptotic square error or MASE. *2. Misadjustment:* This quantity describes steady-state behavior of the algorithm. This is a quantitative measure of the amount by which the ensemble averaged final value of the mean-squared error exceeds the minimum mean-squared error produced by the optimal Wiener filter. The smaller the misadjustment, the better the asymptotic performance of the algorithm. *3. Numerical Robustness:* The implementation of adaptive filtering algorithms on a digital computer, which inevitably operates using finite word-lengths, results in quantization errors. These errors sometimes can cause numerical instability of the adaptation algorithm. An adaptive filtering algorithm is said to be numerically robust when its digital implementation using finite-word-length operations is stable. *4. Computational Requirements:* This is an important parameter from a practical point of view. The parameters of interest include the number of operations required for one complete iteration of the algorithm and the amount of memory needed to store the required data and also the program. These quantities influence the price of the computer needed to implement the adaptive filter.

5. Stability: An algorithm is said to be stable if the mean-squared error converges to a final (finite) value. Ideally, one would like to have a computationally simple and numerically robust adaptive filter with high rate of convergence and small misadjustment that can be implemented easily on a computer. In the applications of digital signal processing e.g. adaptive echo cancellation, the above factors play an important role.

There are different types of adaptive filtering algorithms, they are

- Least mean square (LMS) algorithm
- Normalized least mean square (NLMS) algorithm
- Variable step size LMS (VSLMS) algorithm
- Variable step size Normalized LMS (VSNLMS) algorithm
- Recursive least squares (RLS) algorithm.

i) The structure of Adaptive filter

The block diagram for the adaptive filter method utilized in this section. Here w represents the coefficients of the FIR filter tap weight vector, x(n) is the input vector samples, z^{-1} is a delay of one sample periods, y(n) is the adaptive filter output, d(n) is the desired echoed signal and e(n) is the estimation error at time n. The aim of an adaptive filter is to calculate the difference between the desired signal and the adaptive filter output, e(n). This error signal is fed back into the adaptive filter and its coefficients are changed algorithmically in order to minimize a function of this difference, known as the cost function. In the case of acoustic echo cancellation, the optimal output of the adaptive filter is equal in value to the unwanted echoed signal.

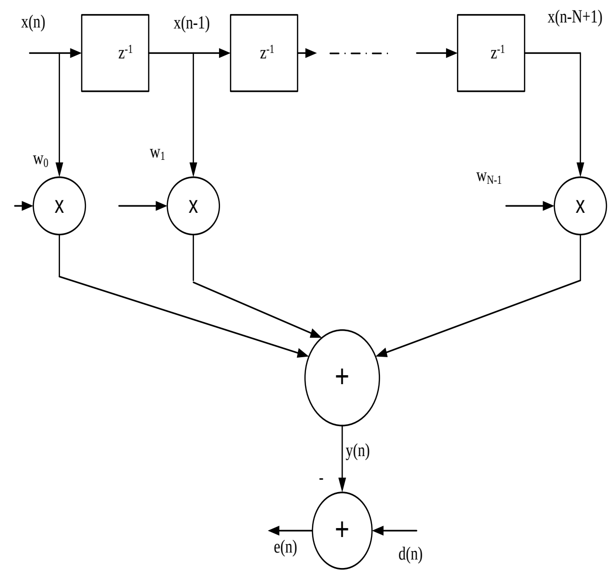

Figure 6.2 : Adaptive filter block diagram

When the adaptive filter output is equal to desired signal the error signal goes to zero. In this situation the echoed signal would be completely cancelled and the far user would not hear any of their original speech returned to them.

VII. Least Mean Square (LMS) Algorithm

The LMS algorithm is a type of adaptive filter known as stochastic gradient-based algorithms as it utilizes the gradient vector of the filter tap weights to converge on the optimal wiener solution. It is well known and widely used due to its computational simplicity. With each iteration of the LMS algorithm, the filter tap weights of the adaptive filter are updated according to the following formula

$$w(n+1) = w(n) + 2\mu e(n)x(n) \tag{7.1}$$

Here **x**(n) is the input vector of time delayed input values

$$x(n) = [x(n)\ x(n-1)\ x(n-2) \ldots\ x(n-N+1)]^T \tag{7.2}$$

The vector w(n) represents the coefficients of the adaptive FIR filter tap weight vector at time n.

$$w(n) = [w_0(n)\ w_1(n)\ w_2(n) \ldots\ w_{N-1}(n)]^T \tag{7.3}$$

The parameter μ is known as the step size parameter and is a small positive constant. This step size parameter controls the influence of the updating factor. Selection of a suitable value for μ is imperative to the performance of the LMS algorithm, if the value is too small the time the adaptive filter takes to converge on the optimal solution will be too long; if μ is too large the adaptive filter becomes unstable and its output diverges.

a) Derivation of the LMS algorithm

The derivation of the LMS algorithm builds upon the theory of the wiener solution for the optimal filter tap weights, w_o, as outlined in section 3.2.2. It also depends on the steepest descent algorithm as stated in equation 3.23, this is a formula which updates the filter coefficients using the current tap weight vector and the current gradient of the cost function with respect to the filter tap weight coefficient vector, ξ(n)

$$w(n+1)=w(n)-\mu\,\xi(n)$$

Where $$\xi(n)=E[e(n)^2] \qquad (7.4)$$

As the negative gradient vector points in the direction of steepest descent for the N-dimensional quadratic cost function, each recursion shifts the value of the filter coefficients closer toward their optimum value, which corresponds to the minimum achievable value of the cost function, ξ(n).

The LMS algorithm is a random process implementation of the steepest descent algorithm, from equation 3.23. Here the expectation for the error signal is not known so the instantaneous value is used as an estimate. The steepest descent algorithm then becomes equation 3.24.

$$w(n+1)=w(n)-\mu\,\xi(n)$$

Where $$\xi(n) = e(n)^2 \qquad (7.5)$$

The gradient of the cost function, ▼ξ(n), can alternatively be expressed in the following form.

$\xi(n)=(e^2(n))$

$$=\frac{\partial e^2(n)}{\partial w}$$

$$= 2e(n)\frac{\partial e(n)}{\partial w} = -\,2e(n)\frac{\partial(d(n)-y(n))}{\partial w} = \quad (7.6)$$

$$-\,2e(n)\frac{\partial w^T(n)x(n)}{\partial w} = -\,2e(n)x(n)$$

Substituting this into the steepest descent algorithm of equation 3.8, we arrive at the recursion for the LMS adaptive algorithm.

$$w(n+1) = w(n) + 2\mu e(n)x(n) \qquad (7.7)$$

b) Implementation of the LMS algorithm

Each iteration of the LMS algorithm requires 3 distinct steps in this order:

i. *The output of the FIR filter, y(n) is calculated using equation 3.27*

$$y(n)=\sum_{i=0}^{N-1} w(n)x(n-i)=w^T(n)x(n) \quad (7.8)$$

ii. *The value of the error estimation is calculated using equation 3.28.*

$$e(n)=d(n)-y(n) \qquad (7.9)$$

iii. *The tap weights of the FIR vector are updated in preparation for the next iteration, by equation 7.9*

$$w(n+1) = w(n) + 2\mu e(n)x(n) \qquad (7.10)$$

The main reason for the LMS algorithms popularity in adaptive filtering is its computational simplicity, making it easier to implement than all other commonly used adaptive algorithms. For each iteration the LMS algorithm requires 2N additions and 2N+1 multiplications (N for calculating the output, y(n), one for 2μe(n) and an additional N for the scalar by vector multiplication). One of the primary disadvantages of the LMS algorithm is having a fixed step size parameter for every iteration. This requires an understanding of the statistics of the input signal prior to commencing the adaptive filtering operation. In practice this is rarely achievable. Even if we assume the only signal to be input to the adaptive echo cancellation system is speech, there are still many factors such as signal input power and amplitude which will affect its performance.

The normalized least mean square algorithm (NLMS) is an extension of the LMS algorithm which bypasses this issue by selecting a different step size value, μ(n), for each iteration of the algorithm. This step size is proportional to the inverse of the total expected energy of the instantaneous values of the coefficients of the input vector **x**(n). This sum of the expected energies of the input samples is also equivalent to the dot product of the input vector with itself, and the trace of input vectors auto-correlation matrix, **R**.

$$tr(\boldsymbol{R}) = \sum_{i=0}^{N-1} E[x^2(n-i)] \qquad (7.11)$$
$$= E\left[\sum_{i=0}^{N-1} x^2(n-i)\right]$$

The recursion formula for the NLMS algorithm is stated in equation 3.31.

$$w(n+1)=w(n)+\frac{1}{x^T(n)x(n)}e(n)x(n) \qquad (7.12)$$

c) Derivation of the NLMS algorithm

To derive the NLMS algorithm we consider the standard LMS recursion, for which we select a variable step size parameter, μ(n). This parameter is selected so that the error value, $e^+(n)$, will be minimized using the updated filter tap weights, w(n+1), and the current input vector, **x**(n). $w(n+1) = w(n) + 2\mu(n)e(n)x(n)$, $e^+(n) = d(n) - w^T(n+1)x(n), =(1-2\mu(n)x^T(n)x(n))e(n)$ (7.13)

Next we minimize $(e^+(n))^2$, with respect to μ(n). Using this we can then find a value for μ(n) which forces $e^+(n)$ to

$$\text{zero.}\ \mu(n)=\frac{1}{2x^T(n)x(n)} \qquad (7.14)$$

This μ(n) is then substituted into the standard LMS recursion replacing μ, resulting in the following NLMS equation.

$$w(n+1) = w(n) + 2\mu(n)e(n)x(n)$$

$$w(n+1)=w(n)+\frac{1}{x^T(n)x(n)}e(n)x(n) \quad (7.15)$$

d) Implementation of the NLMS algorithm

The NLMS algorithm has been implemented in Matlab and in a real time application using the Texas Instruments TMS320C6711 Development Kit. As the step size parameter is chosen based on the current input values, the NLMS algorithm shows far greater stability with unknown signals. This combined with good convergence speed and relative computational simplicity makes the NLMS algorithm ideal for the real time adaptive echo cancellation system.

As the NLMS is an extension of the standard LMS algorithm, the NLMS algorithms practical implementation is very similar to that of the LMS algorithm. Each iteration of the NLMS algorithm requires these steps in the following order.

The output of the adaptive filter is calculated.

$$y(n)=\sum_{i=0}^{N-1} w(n)x(n-i) = w^T(n)x(n) \quad (3.35)$$

An error signal is calculated as the difference between the desired signal and the filter output

$$e(n)=d(n)-y(n) \quad (7.16)$$

The step size value for the input vector is calculated.

$$\mu(n) = \frac{1}{2x^T(n)x(n)} \quad (7.17)$$

The filter tap weights are updated in preparation for the next iteration.

$$w(n+1) = w(n) + \mu(n)e(n)x(n) \quad (7.18)$$

Each iteration of the NLMS algorithm requires 3N+1 multiplications, this is only N more than the standard LMS algorithm, this is an acceptable increase considering the gains in stability and echo attenuation achieved.

VIII. Comparison of Adaptive Filtering Algorithms

Algorithm: LMS Algorithm

Average attenuation: -18.2 dB

Multiplication operations: 2N+1

Comments: Is the simplest to implement and is stable when the step size parameter is selected appropriately. This requires prior knowledge of the input signal which is not feasible for the echo cancellation system.

Algorithm: NLMS Algorithm

Average attenuation: -27.9dB

Multiplication operations: 3N+1

Comments: Simple to implement and computationally efficient. Shows very good attenuation and variable step size allows stable performance with non-stationary signals. This was the obvious choice for real time implementation.

Algorithm: VSSLMS Algorithm

Average attenuation: -9.8 dB

Multiplication operations: 4N+1

Comments: Displays very poor performance, possibly due to non-stationary nature of speech signals. Only half the attenuation of the standard LMS algorithm. Not considered for real time implementation.

Algorithm: VSSNLMS Algorithm

Average attenuation: -9.9 dB

Multiplication operations: 5N+1

Comments: Increase in multiplications gives negligible improvement in performance over VSSLMS algorithm.

The real time acoustic echo cancellation system was successfully developed with the NLMS algorithm. The system is capable of cancelling echo with time delays of up to 75 ms, corresponding to reverberation off an object a maximum of 12 meters away. This proves quite satisfactory in emulating a medium to large size room.

The utility of SBC is perhaps best illustrated with a specific example. When used for audio compression, SBC exploits what might be considered a deficiency of the human auditory system. Human ears are normally sensitive to a wide range of frequencies, but when a sufficiently loud signal is present at one frequency, the ear will not hear weaker signals at nearby frequencies. We say that the louder signal masks the softer ones. The louder signal is called the masker, and the point at which masking occurs is known, appropriately enough, as the masking threshold. The basic idea of SBC is to enable a data reduction by discarding information about frequencies which are masked. The result differs from the original signal, but if the discarded information is chosen carefully, the difference will not be noticeable, or more importantly, objectionable.

IX. Encoding Audio Signals

The simplest way to digitally encode audio signals is pulse-code modulation (PCM), which is used on audio CDs, DAT recordings, and so on. Digitization transforms continuous signals into discrete ones by sampling a signal's amplitude at uniform intervals and rounding to the nearest value representable with the available number of bits. This process is fundamentally inexact, and involves two errors: discretization error, from sampling at intervals, and quantization error, from rounding.

The more bits used represent each sample, the finer the granularity in the digital representation, and thus the smaller the error. Such quantization errors may be thought of as a type of noise, because they are effectively the difference between the original source and its binary representation. With PCM, the only way to mitigate the audible effects of these errors is to use enough bits to ensure that the noise is low enough to be masked either by the signal itself or by other sources of noise. A high quality signal is possible, but at the cost of a high bitrate (e.g., over 700 kbit/s for one channel of CD audio). In effect, many bits are wasted in encoding masked portions of the signal because PCM makes no assumptions about how the human ear hears. More clever ways of digitizing an audio signal can reduce that waste by exploiting known characteristics of the auditory system. A classic method is nonlinear PCM, such as mu-law encoding (named after a perceptual curve in auditory perception research). Small signals are digitized with finer granularity than are large ones; the effect is to add noise that is proportional to the signal strength. Sun's Au file format for sound is a popular example of mu-law encoding. Using 8-bit mu-law encoding would cut the per-channel bit rate of CD audio down to about 350 kbit/s, or about half the standard rate. Because this simple method only minimally exploits masking effects, it produces results that are often audibly poorer than the original. Sub-band coding is used for example in G.722 codec. It uses sub-band adaptive differential pulse code modulation (SB-ADPCM) within a bit rate of 64 kbit/s. In the SB-ADPCM technique used, the frequency band is split into two sub-bands (higher and lower) and the signals in each sub-band are encoded using ADPCM.

As explained in Section 2, in the proposed algorithm the TAF is obtained with a delay relative to the input signal. The amount of delay depends on the method of filter reconstruction. For sequential synthesis the delay is (*La* □□*Ls*) / *2* samples while for batch synthesis it is *La* / *2* samples. All of the delayless SAF methods reviewed in Section 1 have to deal with a plant reconstruction delay. The delay leads to a "synchronization problem" between the input signals and the plant, causing problems in tracking a dynamic plant. The extent of the problem depends on the plant time-dynamics and the TAF reconstruction delay. To demonstrate the effects of the delay, we simulated the system with the same system set up and input signals as described in the previous section with the following changes. The echo plant was switched to a new plant after *30* seconds through the experiment. With the employed analysis/synthesis filters used in the experiments, tracking problems were barely observable due to the low reconstruction delay of the system. Thus, the analysis and synthesis window lengths were increased to *La* □*1024* and *Ls* □*256* samples to better observe the effects of the delay. To simplify the analysis, batch synthesis was used for TAF WOLA reconstruction. This leads to a filter reconstruction delay of *La* / *2* □*512* samples. Delaying the input signals by the same amount so that they are synchronized with the plant could compensate for the filter reconstruction delay. Of course this is counter productive as it creates delays in an otherwise delayless system. The ERLE drops at *30* seconds, and stays low for around *64* msecs (corresponding to *512* samples of delay) before it starts to rise again. This low-time of ERLE causes a drop in echo cancellation performance and creates artifacts in the output. Repeating the experiment with delay compensation, the ERLE drops later and start to rise right away as shown in the figure. The echo plant swap is unlikely to happen in practice; rather gradual plant variations might occur.

X. Simulation Results

The input file 'file1.wav' is read by the command waveread and impulse response of secondary path is plotted as shown in fig 5.1. This means the output of secondary path y`(n) initially. Then estimating the secondary propagation path S^(z) and identifying this path using NLMS adaptive filter. This is shown in fig 5.2 and also shows the required signal d(n),output signal y`(n) and error signal e(n).If the number of iterations are increased, the error signal e(n) is reduced. Here d(n) and y`(n) are having same signal value from $0.5*10^4$ to $3*10^4$.

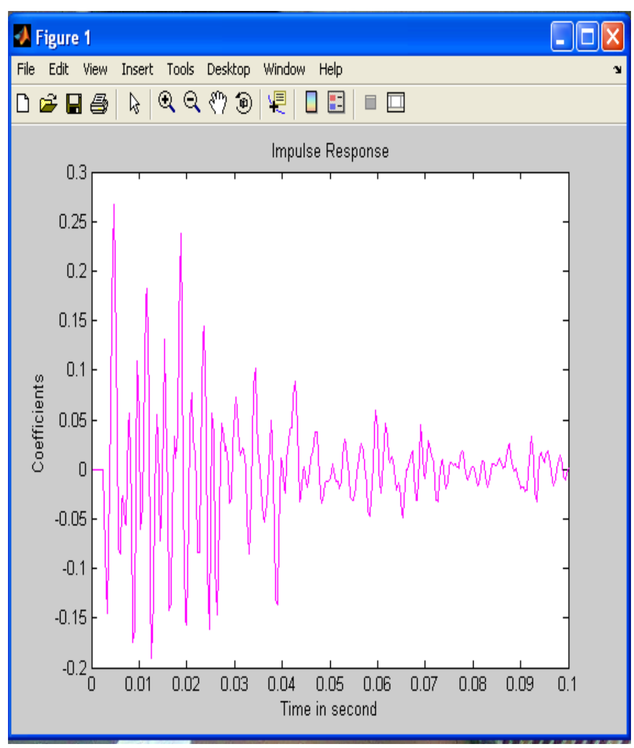

Figure 8.1 : Secondary path filter response

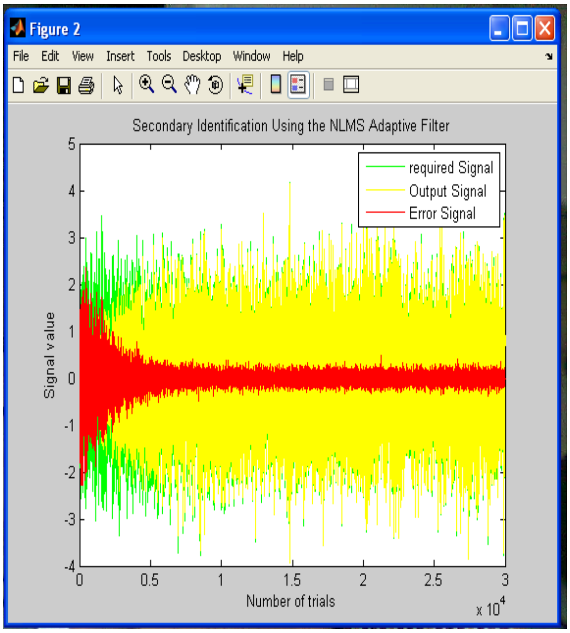

Figure 8.2 : Secondary path identification

a) Using NLMS algorithm

Fig 8.3 shows accuracy of the estimated secondary propagation path S^(z). Also the summer output e(n) in time-domain. Here the d(n) and y`(n) follows the same path and error signal e(n) is zero.

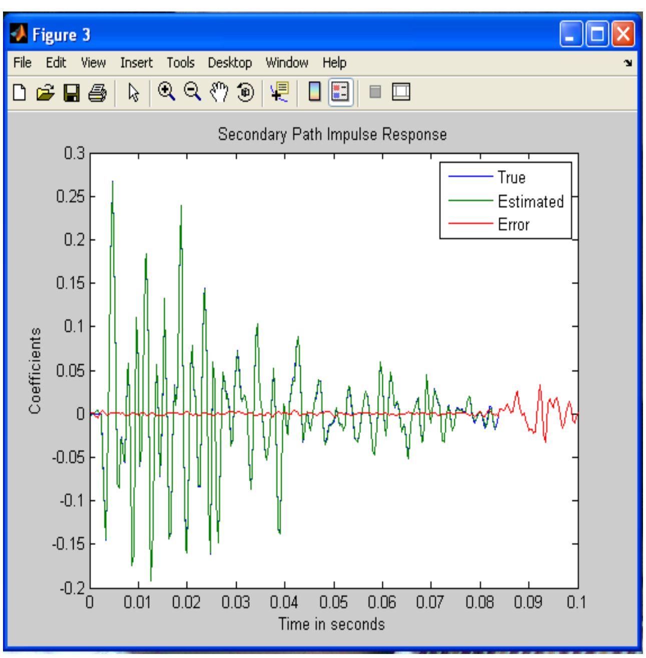

Figure 8.3 : Accuracy of Secondary path

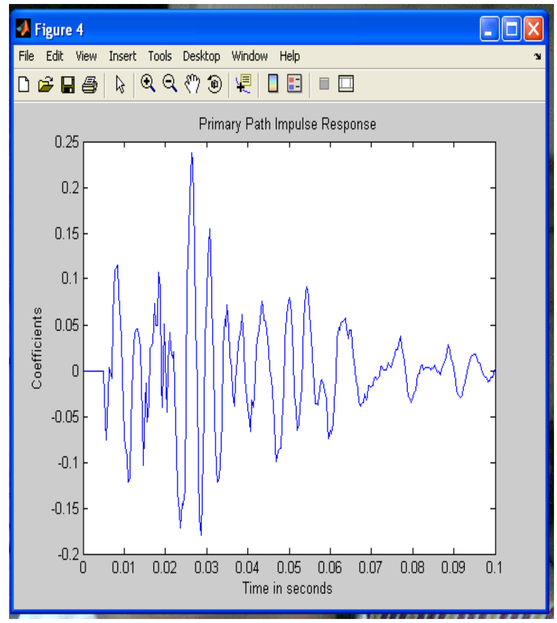

Figure 8.4 : Primary path filter response

Fig 8.4 shows the primary propagation path filter response P(z).The output of P(z) is d(n),this is the actual noise to be canceled by generating anti-noise through the S^(z).Then the noise in the system is to be canceled and fig 5.5 shows the power spectral density of the canceled noise d(n). Power spectral density means the distribution of d(n) over frequency-axis. The voice frequency range is considered as from (0.3-3.5) KHz.

Fig 8.6 shows residual error signal spectrum of e(n) or power spectral density of d(n) and y`(n).From (0-1300)Hz, there is small difference between d(n) and y`(n).After that both are equal. Fig 5.7 shows the power spectral density of d(n)-y`(n).At 200Hz, power/frequency is -50 db/Hz (0.0031) approximately zero.

The total complexity is plotted in Fig.5.8, number of real multiplications versus the number of subbands M. The plot is for the PFFT-2 method with $L_{SAF} = 4N/M$, as it results in better performance than that of PFFT-1. For comparison purposes, included the computational complexity of the MT and DFT-MDF algorithms. As shown, the computational complexities of all methods reduce almost exponentially with M. The proposed technique compared to the other methods for small values of M has higher computational complexity.

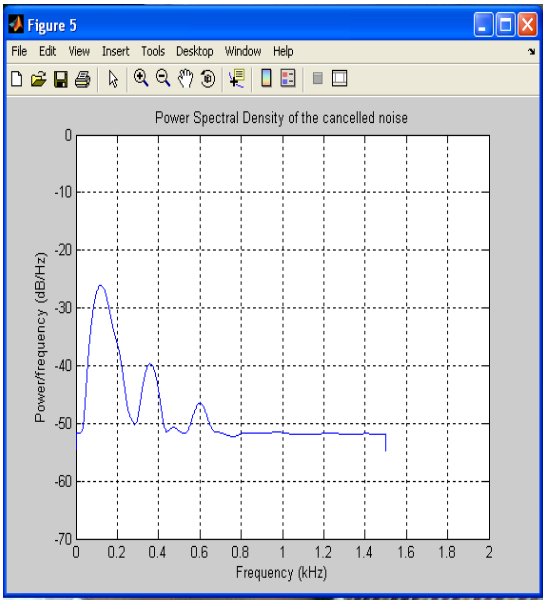

Figure 8.5 : Power spectral density of canceled

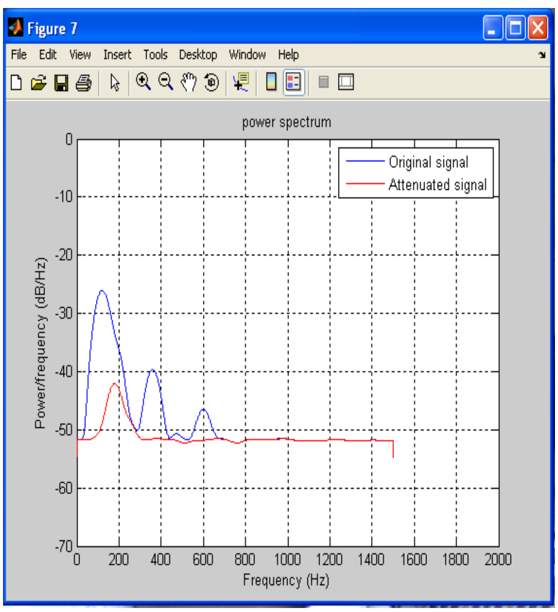

Figure 8.6 : Residual error signal spectrum Noise

The new technique works very well with a larger number subbands, improving the system performance and attaining lower complexity, whereas the MT method fails to converge and the performance of the DFT-MDF method deteriorates.

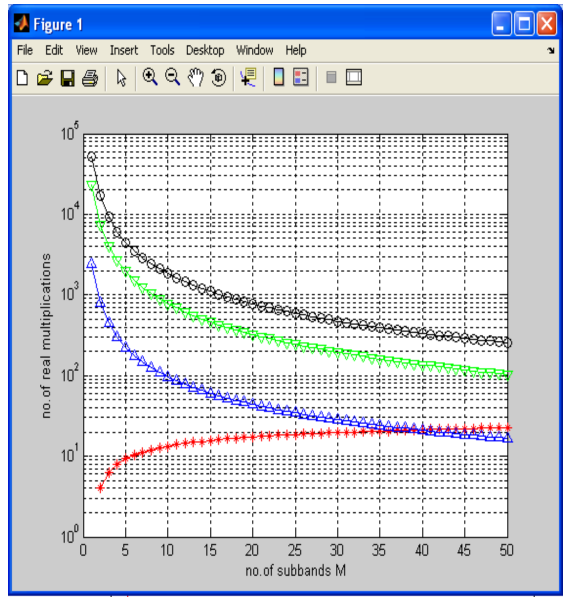

Figure 8.7 : Comparison of computational complexity per input sample versus number of subbands (M)

XI. Conclusion

Acoustic paths such as those encountered in ANC application usually have long impulse responses, which require longer adaptive filters for noise cancellation. Subband adaptive filters working with a large number of subbands have been shown to be a good solution to this problem. The focus of this project was to design such a high-performance SAF algorithm. The performance limiting factors of existing SAF structures were found to be due to the inherent delay and side-lobes of the prototype filter in the analysis filter banks. Hence, the analysis filter banks were modified to reduce the inherent delay. A new weight stacking transform was designed to alleviate the interference introduced by the side-lobes. The modifications resulted in a new subband method that, unlike existing methods, improves the performance and reduces the computational complexity for a large number of subbands.

Experimental results showed that the proposed method outperformed the two commonly used SAF and BAF methods. The proposed technique compared to the other methods for small values of M has higher computational complexity. The new technique works very well with a larger number of subbands, improving the system performance and attaining lower complexity, whereas the MT method fails to converge and the performance of the DFT-MDF method deteriorates.

XII. Future Scope

Adaptive digital signal processing is a rapidly growing branch of DSP and has great significance in the design of adaptive systems. The various signal processing applications demand for reduction in trade off between misadjustment and convergence rate, taking realization of algorithm into account. The modifications resulted in a new subband method that, unlike existing methods, improves the performance and reduces the computational complexity for a large number of subbands. There is a scope of improvement in replacing the existing time domain adaptive filters with frequency domain adaptive filters. There's a lot, which can be done in future for improvement on the methods for noise cancellation. The field of digital signal processing and in particular adaptive filtering is vast and further research and development in this area can result in some improvement on the methods studied in this paper.

XIII. Acknowledgements

The work is carried out through the research facility at the Department of Computer Science & Engineering, Department of Electronics & Communication Engineering and Department of Electrical & Electronics Engineering, JITS, Nustulapur (V), Karimnagar-District, A.P., Thanks to the experts who have contributed towards development of this paper.

References Références Referencias

1. D. R. Morgan and J. C. Thi, "A delayless subband adaptive filter architecture, "IEEE Trans. Signal Process., vol. 43, no. 8, pp. 1819–1830, Aug. 1995.
2. R. Merched and A. H. Sayed, "An embedding pproach to frequency domain and subband adaptive filtering," IEEE Trans. Signal Process., vol. 48, no. 9, pp. 2607 2619, Sep. 2000.
3. P. Loizou, Speech Enhancement: Theory and Practice. Boca Raton, FL: CRC, 2007.
4. V. R. Ramachandran, G. Kannan, A. A. Milani, and I. M. S. Panahi, "Speech enhancement in functional MRI environment using adaptive sub-band algorithms, in Proc. ICASSP'07, 2007, pp. II-341–II-344.
5. P. Loizou and K. Kokkinakis, "Subband-based blind signal processing for source separation in convolutive mixtures of speech," in Proc. ICASSP'07, 2007, pp. IV 917–IV-920.
6. N. Hirayamaa, H. Sakai, and S. Miyagi, "Delayless subband adaptive filtering using the hadamard transform," IEEE Trans. Acoust., Speech, Signal Process., vol. 47, pp. 1731–1734, Jun. 1999.
7. P. Eneroth, "Joint filterbanks for echo cancellation and audio coding," IEEE Trans. Speech Audio Process., vol. 11, no. 4, pp. 342–354, Jul. 2003.
8. D. Zhou and V. DeBrunner, "A new active noise control algorithm that requires no secondary path identification based on the SPR property," IEEE Trans. Signal Process., vol. 55, no. 5, pp. 1719–1729, May 2007.
9. J. Shynk, "Frequency-domain and multirate adaptive filtering," IEEE Signal Process. Mag., vol. 9, no. 1, pp. 14–37, Jan. 1992.
10. P. Vaidyanathan, Multirate Systems and Filter Banks. Englewood Cliffs, NJ:Prentice-Hall, 1993
11. J. Huo, S. Nordholm, and Z. Zang, "New weight transform schemes for delayless subband adaptive filtering," in Proc. GLOBECOM'01, New York, Nov. 2001, pp. 197–201.
12. L. Larson, J. de Haan, and I. Claesson, "A new subband weight transforms for delayless subband adaptive filtering structures," in Proc. Digital Signal Process. Workshop, Aug. 2002, pp. 201–206.
13. J. M. de Haan, N. Grbic, I. Claesson, and S. E. Nordholm, "Filter bank design for subband adaptive microphone arrays," IEEE Trans. Speech Audio Process., vol. 11, no. 1, pp. 14–23, Jan. 2003.
14. A. H. Sayed, Fundamentals of Adaptive Filtering. New York: Wiley, 2003.
15. Z. Lin and Y. Liu, "Design of complex fir filters with reduced group delay error using semidefinite programming," IEEE Trans. Signal Process., vol. 13, no. 9, pp 529–532, Sep. 2006.
16. R. Venkataramani and Y. Bresler, "Filter design for MIMO sampling and reconstruction," IEEE Trans. Signal Process., vol. 51, no. 12, pp. 3164–3167, Dec. 2003.
17. Y. C. Lim, J. Lee, C. K. Chen, and R. Yang, "A weighted least squares algorithm for quasi-equiripple FIR and IIR digital filter design," IEEE Trans. Signal Process., vol.40, no. 3, pp. 551–558, Mar. 1992.
18. H. Gustafsson, S. E. Nordholm, and I. Claesson, "Spectral subtraction using reduced delay convolution and adaptive averaging," IEEE Trans. Speech Audio Process., vol. 9, no. 8, pp. 799–807, Nov. 2001.
19. A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

Global Journals Inc. (US) Guidelines Handbook 2014

www.GlobalJournals.org