Adaptive Echo Noise Elimination for Speech Enhancement of Tamil letter 'Zha'

A. Srinivasan
Department of ECE, Srinivasa Ramanujan Centre,
SASTRA University, Kumbakonam-612001, India.
E-mail: asrinivasan78@yahoo.com

Abstract

Acoustic echo depends on time delay between initial and reflected sound wave, strength of reflected sound. In the speech processing of letter 'zha' [11], echo of the recorded voice gives the spurious results. Such complexity can be avoided by suitable pyramidal method like adaptive filtering technique. Adaptive filtering tries to adjust these parameters with the aim of meeting some well-defined target, which depends upon the state of the system and surroundings. In speech recognition, the acoustic echo gives the faulty results. Objective of this paper is to analyze the performance of various adaptive filtering algorithms for acoustic echo cancellation in recorded speech enhancement of the letter 'Zha' in Tamil language. These algorithms are simulated in MATLAB and compared with the performance of those algorithms based on parameters such us computational complexity, convergence rate and amount of echo attenuation.

Key words: Acoustic echo, Adaptive filter, and FIR filter, LMS, NLMS, RLS and MATLAB

I. INTRODUCTION

An acoustic echo canceller records the sound going to the loudspeaker and substract it - in some way - from the signal coming from the microphone. The sound going through the echo-loop is transformed and delayed, and noise is added, which complicates the subtraction process. Figure.1 illustrates the general echo-cancelling.

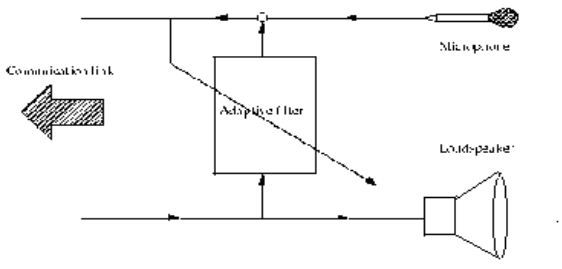

Figure 1. Echo canceller principle

1.1.Adaptive Echo Canceller

Adaptive echo cancellers that can learn the echo path when it is first turned on and afterwards track its variations without any intervention from the designer. Since an adaptive canceller matches the echo path for any given connection, it performs better than a compromise fixed canceller.Let 'X' be the input signal going to the loudspeaker; let 'd' be the signal picked up by the microphone, which will be called the desired signal. The signal after subtraction is called the error signal and will be denoted by 'e'.

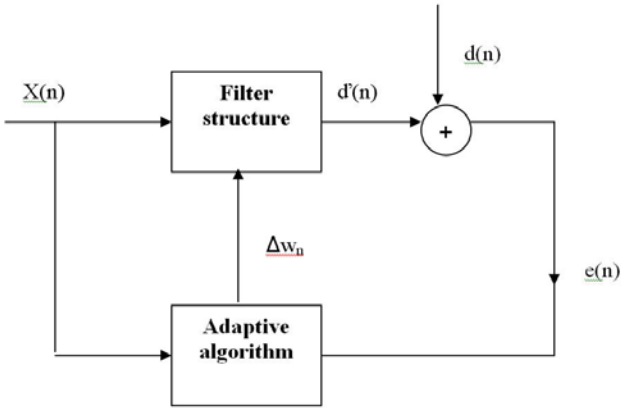

Figure 1. Adaptive filter

II. ADAPTIVE FIR FILTER

FIR filters are routinely used in adaptive filtering applications that range from adaptive equalization in digital communication systems to adaptive noise control systems. The reasons for the popularity of FIR adaptive filters are(i)The filter coefficients control the stability easily (ii)There are simple and efficient algorithms for adjusting the filter coefficients (iii)The performance of these algorithms is well understood in terms of their convergence and stability.

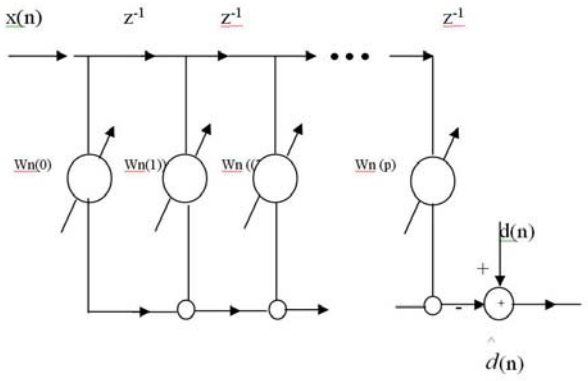

Figure 2. A direct form FIR adaptive filter

An FIR adaptive filter for estimating a desired signal d (n) from a related signal x (n), as illustrated in figure 3, is

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k)\, x(n-k)$$

$$\hat{d}(n) = w_n^T x(n)$$

here it is assumed that x(n) and d(n) are nonstationary Random process and the goal is to find the coefficient vector W_n at the time n that minimizes the mean-square error,

$$\xi(n) = E\{|e(n)|^2\} \quad \text{where, } e(n) = d(n) - d'(n) = d(n) - w_n^T x(n)$$

As in the derivation of the FIR wiener filter, the solution to this minimization problem may be found by setting the derivative of ξ(n) with respect to W_n^* (k) equal to zero for k = 0,1,2,…….,p. the result is $E\{e(n)x^*(n-k)\} = 0$; k = 0,1,…..,p

Substituting e(n) into above equation we have

$$E\{[d(n) - \sum_{l=0}^{p} w_n(l)\, x(n-l]\, x^*(n-k)\} = 0;\ k = 0,1,2,..,p$$

Which, after rearranging terms,becomes

$$\sum_{l=0}^{p} w_n(l)\, E\{x(n-l)\, x^*(n-k)\} = E\{d(n)\, x^*(n-k)\};\ k = 0,1,\ldots..,p.$$

Above equation is a set of p+1 linear quations in the p+1 unknowns $w_n(l)$. However, unlike the case of an FIR Wiener filter where it was assumed that x(n) and d(n) are jointly wide sense stationary, the solution to these equations depends on n. We may express these equations in vector forms as follows

$$R_x(n)\, w_n = r_{dx}(n)$$

In the case of jointly wide sense stationary processes, the above equation reduces to the Wiener-Hopf equations, and the solution w_n becomes independent of time. Instead of solving above equation for each value of n, which would be impractical in most real-time implementations, in the following section, we consider an iterative approach that is based on the method of steepest descent.

2.1 Method of steepest descent

By assuming that the cost function to be mimimized is convex,it may be stated with an arbitary point on the performance surface and take a small step in the direction in which the cost function decreses fastest. This corresponds to a step along the steepest descent slope of the performance surface at that point. Repeating this sucessively, 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_o (in contrast to solving wiener-hopf equation directly), and uses the following procedure to search for the minimum point of the cost function of a set of filter tap weight,

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 initialiated by setting all the filter tab weight to zero,i.e.w(0).

Use this initial guess to compute the gradient vector of the cost function with respect to these tap weights at the present point.

Update the tap weights by taking a step in the opposite direction (sign change) of the gradient vector obtained in step2. This corresponds to a swtep in the direction of steepest descent in the cost function at the present input. Furthermore,the size of the step taken is chosen proprotional to the size of the gradient vector.

go back to step2 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 n-th iteration, then the follwing recursive equation may be used to update w(n):

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

Where μ is a positive scaler called the step size,and $\nabla_n\xi$ denotes the gradient vector evaluated at the point w=w(n). For small value of μ, the correction to w_n is small and the movement down the quadradic surface is slow and ,as μ is increased, the rate of the descent increases. However, there is a upper limit on how large the step size may be. For values of μ that exceeds this limit, the trajectory of w_n becomes unstable and unbounded.

Let us evaluate the gradient vector $\nabla\xi(n)$. assuming that w is complex, the gradient is the derivative of $E\{|e(n)|^2\}$ with respect to w. with

$\nabla\xi(n)= \nabla E\{|e(n)|^2\}= E\{\nabla|e(n)|^2\}= E\{e(n)\nabla e^*(n)\}$

and

$\nabla e^*(n)=-x(n)$

it follows that,

$\nabla\xi(n)=-E\{e(n)x^*(n)\}$

thus, with a step size of μ , the steepest descent algorithm becomes

$w_{n+1} =w_n + \mu E\{e(n)x^*(n)\}$

III. Adaptive Filtering Algorithms

Many proposed algorithms are available for cancelling the echo noise. The algorithms are, Least mean square algorithm,Normalized LMS algorithm and Recursive least square algorithm

3.1.LMS Algorithm

We developed the steepest descent adaptive filter, which has a weight vector update equation given by

$w_{n+1} =w_n + \mu E\{e(n)x^*(n)\}$

A practical limitation with this algorithm is that the expectation $E\{e(n)x^*(n)\}$ is generally unknown. Therefore, it must be replaced with an estimate such as the sample mean.

$$\hat{E}\{e(n)x^*(n)\}= \frac{1}{L}\sum_{l=0}^{L-1} e(n-l)x^*(n-l)$$

Incorporating this estimate into the steepest descent algorithm, the update for w_n becomes

$$w_{n+1} =w_n + \frac{\mu}{L}\sum_{l=0}^{L-1} e(n-l)x^*(n-l)$$

A special case above equation occurs if we use a one point sample mean (L=1),

$\hat{E}\{e(n)x^*(n)\}= e(n)x^*(n)$

In this case, the weight vector update equation assumes a particularly simple form

$w_{n+1} =w_n + \mu e(n)x^*(n)$

and is known as the LMS algorithm. The simplicity of the algorithm comes from the fact that the update of the kth coefficient.

$w_{n+1}(k) =w_n(k) + \mu e(n)x^*(n)$

requires only one multiplication and one addition (the value for $\mu e(n)$ need only the computed once and may be used for all of the coefficients). Therefore, an LMS adaptive filter having p+1 coefficients requires p+1 multiplications and (p+1) additions to update the filter coefficients. In addition, one addition is necessary to compute the error e(n)=d(n)-y(n) and one multiplication is needed to form the product μ e(n). Finally, p+1 multiplications and p additions are necessary to calculate the output, y(n), of the adaptive filter. Thus a total of 2p+3 multiplications and 2p+2 additions per output point are required.

3.2. NLMS Algorithm

One of the difficulties in the design and implementation of the LMS adaptive filter is the selection of the step size μ . For stationary process, the LMs algorithm converges in the mean if $0<\mu < 2/\lambda_{max,}$ and converges in the mean-square if $0<\mu<tr(R_x)$. However, since $R_{x\ is}$ generally unknown,then either λ_{max} or R_x must be estimated in order to use these bounds. One way around this difficulty is to use the fact that, for stationary processes, $tr(R_x)= (p+1)E\{|x(n)|^2\}$. Therefore, the condition for mean square convergence may be replaced by $0<\mu<2/(p+1)E\{|x(n)|^2\}$

where $E\{|x(n)|^2\}$ is the power in the process x(n), this power may be estimated using a time average such as

$$\hat{E}\{|x(n)|^2=1/P+1\sum_{k=0}^{p}|x(n-k)|^2$$

Which leads to the following bound on the step size for mean square convergence

$0<\mu<\beta/x^H(n)x(n)= \beta/||x(n)||^2$

where β is a normalized step size with $0<\beta< 2$. Replacing μ in the LMS weight vector update equation with leads to the Normalized LMS algorithm, which is given

$w_{n+1}= w_n+ \beta\, x(n)e(n)/ ||x(n)||^2$

Note that the effect of the normalization by $||x(n)||^2$ is to alter the magnitude, but not the direction, of the estimated gradient vector. Therefore, with the appropriate set of statistical assumption it may be shown that the normalized LMS algorithm converge in the mean square if $0<\beta<2$.

In the LMs algorithm the correction that is applied to the w_n is proportional to the input vector x(n). Therefore, when x(n) is large ,the LMS algorithm experience a problem with gradient white noise amplification. With the normalization of the LMs step size by $||x(n)||^2$ in the NLMS algorithm, however, this noise amplification problem is diminished. Although the NLMS algorithm bypasses the problem of noise amplification, we are now faced the similar problem that occurs when $||x(n)||$ becomes too small. An alternative, therefore, is to use the following modification to the NLMS algorithm

$$w_{n+1}= w_n+ \beta\, x(n)e(n)/\epsilon+||x(n)||^2$$

where ϵ is some small positive number.

Compare with the LMS algorithm, the normalized LMS algorithm requires additional compution to evaluate the normalization term $|x(n)|^2$ however, if this term is evaluated recursively as follows

$$||x(n+1)||^2 = ||x(n)||^2 + |x(n+1)|^2 - |x(n-p)|^2$$

then the extra computation involves only two operations, one addition, and one subtraction.

3.3.RLS Algorithm

Recursive least squares (RLS) algorithm is used to find the filter coefficients that relate to producing the recursively least squares of the error signal (difference between the desired and the actual signal).

We have considered gradient descent algorithms for the minimization of the mean-square error.

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

The difficulty with these methods is that they all require knowledge of the autocorrelation of the input process, E {x(n)x*(n-k)}, and the cross correlation between the input and desired output, E {d(n)x*(n-k)}. When this statistical information is unknown, we have been forced to estimate these statistics from the data. In the LMS adaptive filter, for example, these ensemble averages are estimated using instantaneous values,

$$\hat{E}\{e(n)x^*(n-k)\}= e(n)x^*(n-k)$$

Although this approach may be adequate in some application in others this gradient estimate may not provide a sufficient rapid rate of convergence.An alternative, therefore, is to consider the error measures that do not include expectations and they may be computed directly from the data. For example a least squares error

$$e(n)=\sum|e(i)|^2$$

requires no statistical information about x(n), rd(n), and may be evaluated directly from x(n) and d(n). there is an important philosophical difference, however, between minimizing the least squares error and the mean square error. Minimizing the mean square error produces the same set of filter coefficients for all sequences that have the same statistics. Therefore the coefficients do not depend on the incoming data, only on their ensemble average. With the least squares error, on the other hand, we are minimizing a squared error that depends explicitly on the specific values of x(n) and d(n). Consequently, for different signals we get different filters. As a result, the filter coefficients that minimize the least squares error will be optimal for the given data rather than statistically optimal over a particular class of process. In other words, different realizations of x(n) and d(n) will lead to different solutions even if the statistics of these sequences are the same. In this section, we look at the filters that are derived by minimizing a weighted least squares error, and derive an efficient algorithm for performing these minimization known as recursive least squares.

The idea behind RLS filters is to minimize a weighted least squares error function. To stay within the adaptive filter terminology, the weighted least squares error function is the cost function defined as

Where $0<\lambda<1$ is an exponential weighting factor which effectively limits the number of input samples

$$C(\mathbf{w}_n) = \sum_{i=0}^{n} \lambda^{n-i}|e(i)|^2 = \sum_{i=0}^{n} \lambda^{n-i} e(i)\, e^*(i)$$

based on which the cost function is minimized. The error signal e(n) is defined in the block diagram section of the adaptive filter page. The cost function is minimized by taking the partial derivatives for all entries k of the coefficient vector $w_n$and setting the results to zero

$$\frac{\partial C(\mathbf{w}_n)}{\partial w_n^*(k)} = \sum_{i=0}^{n} \lambda^{n-i} e(i)\, \frac{\partial e^*(i))}{\partial w_n^*(k)} = \sum_{i=0}^{n} \lambda^{n-i} e(i)\, x^*(i-k) = 0$$

Next, replace e(n) with the definition of the error signal

$$\sum_{i=0}^{n} \lambda^{n-i} \left[d(i) - \sum_{l=0}^{p} w_n(l) x(i-l) \right] x^*(i-k) = 0$$

Rearranging the equation yields

$$\sum_{l=0}^{p} w_n(l)\left[\sum_{i=0}^{n}\lambda^{n-i}x(i-l)x^*(i-k)\right]=\sum_{i=0}^{n}\lambda^{n-i}d(i)x^*(i-k)$$

This form can be expressed in terms of matrices

$R_x(n)w_n = r_{dx}(n)$

where $R_x(n)$is the weightened autocorrelation matrix for x(n) and $r_{dx}(n)$is the cross-correlation between d(n) and x(n). Based on this expression we find the coefficients which minimize the cost function as

$w_n = R_x^{-1}(n)\, r_{dx}(n)$

This is the main result of the discussion.

IV. Experimental Results

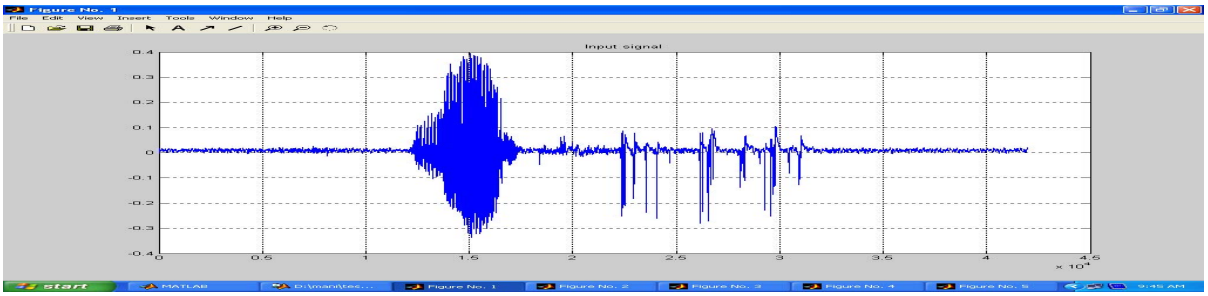

Figure 4 :Input signal (Letter 'Zha' in Tamil Language)

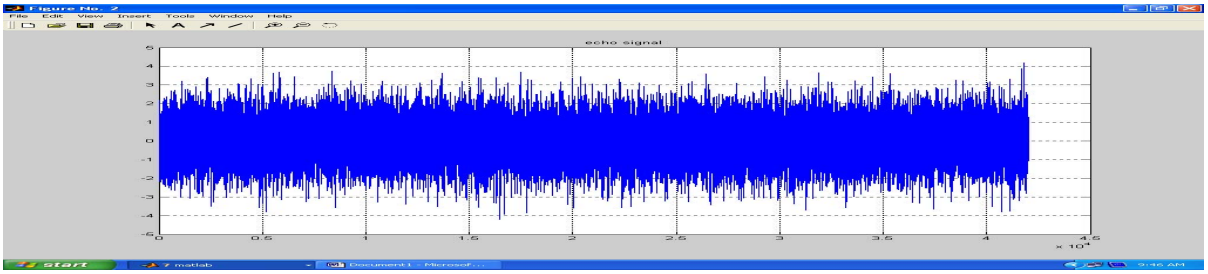

Figure 5: Echo signal

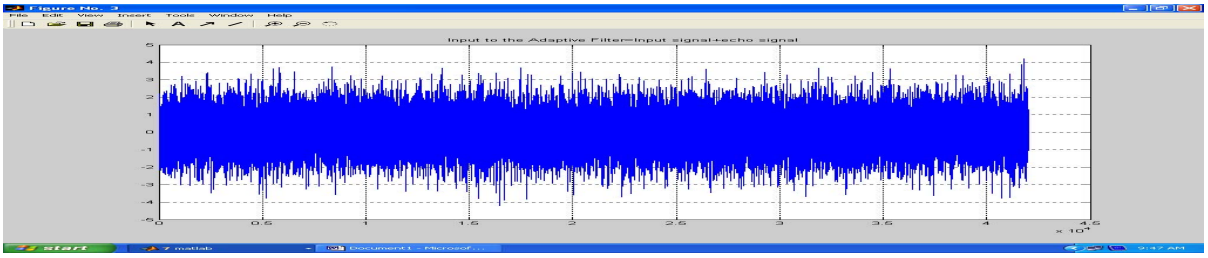

Figure 6 :Input to the adaptive filter = input signal +echo signal

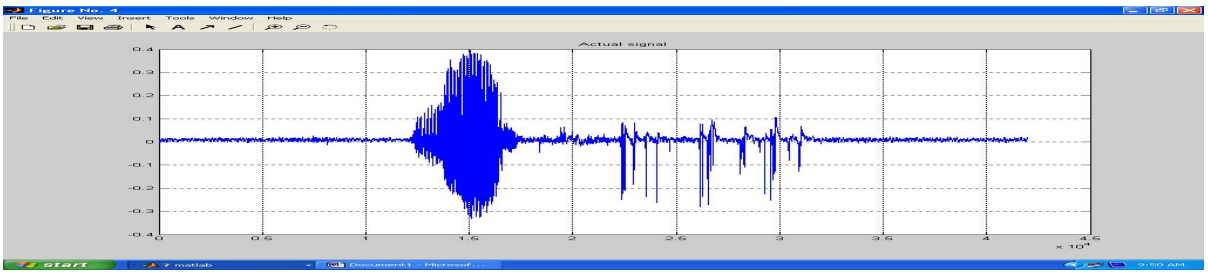

Figure 7: Actual signal by RLS algorithm

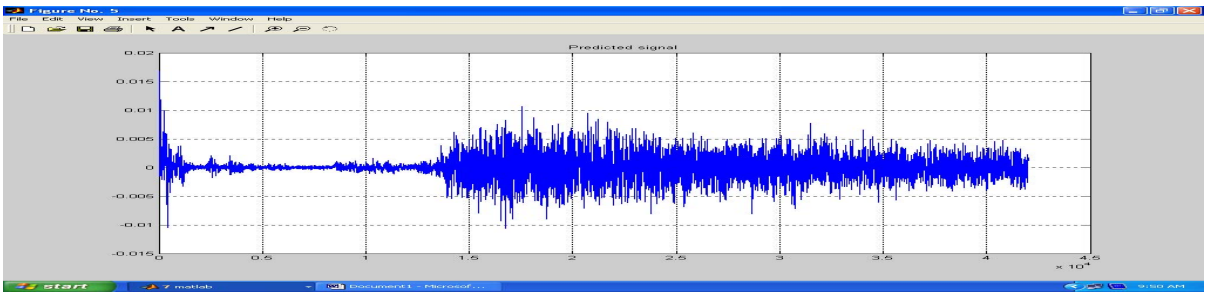

Figure 8: Predicted signal by RLS algorithm

Parameter	LMS	NLMS	RLS
Computational Complexity	34	36	32
Convergence Rate	Slow	Slow	Fast
Maximum ERLE (dB)	3.65	1.03	3.71

TABLE I. PARAMETER COMPARISON OF LMS, NLMS AND RLS ALGORITHMS

V. CONCLUSION

In this paper, LMS, NLMS and RLS algorithms have accomplished the echo cancellation. Among these, LMS algorithm is very simple to implement but slower one. By changing the step size, the speed of the LMS algorithm has been increased in the NLMS algorithm. Even though, the rate of convergence obtained in NLMS is not up to the satisfactory level. In speech recognition of letter 'Zha' in Tamil language [11], the RLS algorithm makes the converging speed and also provides better echo noise reduction when compared to the other algorithms. This simulation can be expanded to the real time hardware implementation using DSP for acoustic echo cancellation in hands free environment.

REFERENCE

[1] Brian R.Hunt, Ronald L.Lipsman, Jonathan M.Rosenberg – 'A guide to MATLAB', Cambridge, 2003

[2] F. Capman, J. Boudy, and P. Lockwood. Acoustic echo cancellation using a fast qr-rls algorithm and multirate schemes. Proceedings of ICASSP, pages 969–972, 1995.

[3] Dimitris G.Manolakis, Vinay K.Lngle, Stephen M.Kogon "statistical and adaptive signal processing", McGraw Hill, 2000.

[4] P. Eneroth, S. Gay, T. Gänsler, and J. Benesty. A hybrid frls/nlms stereo acoustic echo canceller. In Proceedings of IWAENC, 1999.

[5] T. Gansler and J. Benesty. Stereophonic acoustic echo cancellation and two–channel adaptive filtering: an overview. International Journal of Adaptive Control and Signal Processing, February 2000.

[6] Kishan Shenoi 'Digital Signal Processing in Telecommunications', Prentice Hall nternational Editions, 1995.

[7] Monson H.Hayes "Statistical digital signal processing and modeling", John Wiley & Sons, Inc, 1996.

[8] Piotr Pietrzak, Barosz Pekoslawski, Maciej Makowski, Andrzej Napieralski, "Adaptive Subband Filtering Method for MEMS Accelerometer Noise Reduction", Sensors & Transducers Journal, Vol. 3, Special Issue, December 2008, pp. 14-24.

[9] S.Salivahanan, A.Vallavaraj, C.Gnanapriya "Digital signal processing", Tata McGraw Hill, 2001

[10] Simon Haykin. Adaptive Filter Theory. Prentice Hall, 3 edition, 1996.

[11] A.Srinivasan, K.Srinivasa Rao, D.Narasimhan, K.Kannan, "Speech processing of the letter 'zha' in Tamil Language with LPC", Contemporary Engineering Sciences, Vol. 2, 2009, no. 10, 497 – 505.

[12] www.innovexpo.itee.uq.edu