

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



The Ultra High Speed LMS Algorithm Implemented on Parallel Architecture Suitable for Multidimensional Adaptive Filtering

Marwan Jaber
Université du Québec à Trois-Rivières
 Canada

1. Introduction

Over the past decades a number of new adaptive filter algorithms have been elaborated and applied to meet demands for faster convergence and better tracking properties than earlier techniques could offer. The Filtered LMS algorithm is currently the most popular method for adapting a filter, due to its simplicity and robustness, which have made it widely adopted in many applications. Applications include adaptive channel equalization, adaptive predictive speech coding, Noise Suppression and on-line system identification. Recently, because of the progress of digital signal processors, a variety of selective coefficient update of gradient-based adaptive algorithms could be implemented in practice. Different types of adaptive algorithms have been developed and used in conventional adaptive filters such as, filtered LMS algorithms [1], [2], [3] and [4], filtered X-LMS algorithms [1], [2], [5], and [6], filtered NLMS algorithms and RLS algorithms [1] and [2]. As a result, this chapter surveys sequential filter adaptation techniques and some applications for transversal FIR filter.

In other words filters are devices or systems that processes (or reshapes) the input signal according to some specific rules to generate an output signal Figure 1.



Fig. 1. Filter Representation

Filters could be classified in two categories linear and non-linear filters where in linear filters there is a linear relationship between input and output of the filter that satisfy the following property:

$$\begin{aligned}
 x_1 &\rightarrow y_1 \\
 x_2 &\rightarrow y_2 \\
 \text{then } ax_1 + bx_2 &= ay_1 + by_2
 \end{aligned}
 \tag{1}$$

meanwhile in non-linear filters; there is a nonlinearity between input and output of the filter that satisfy the following property:

$$\begin{aligned}x_1 &\rightarrow y_1 = x_1^2 \\x_2 &\rightarrow y_2 = x_2^2 \\ \text{then } x_1 + x_2 &= (x_1 + x_2)^2\end{aligned}\quad (2)$$

In this chapter we will be focusing on adaptive filtering which could automatically adjust (or adapt) in the face of changing environments and changing system requirements by training them to perform specific filtering or decision-making tasks and in which they should be some “adaptation algorithm” for adjusting the system’s parameters figure (2) [7].

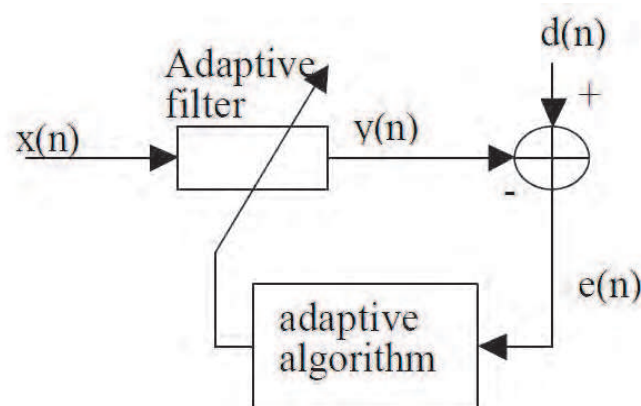


Fig. 2. General Adaptive Filter Configuration.

Different filter structures could be implemented in the adaptive filter of figure 2 such as:

- Transversal Filter Structure (FIR)
- IIR Filter Structure
- Lattice Filter Structure
- Non-Linear Filter Structure (Volterra Filter)

in which the adaptation approaches are based on:

- Wiener filter theory where the optimum coefficients of a linear filter are obtained by minimization of its mean-square error (MSE)
- Method of least squares where The performance index is the sum of weighted error squares

The main objective of the adaptation algorithm is to set the filter parameters, $\overline{\theta_{(n)}}$ in such a way that its output tries to minimize a meaningful objective function F (cost function in some literatures) involving the reference signal (or desired signal) which will be based on

- Mean Square Error
- Least Square Error
- weighted least square
- instantaneous square value

and the minimization algorithms are based on the following methods:

- Newton’s method
- Quasi-Newton method
- Steepest-descent method

2. Signals

Before pursuing the study of adaptive systems, it is important to refresh our memory with some useful definitions from the stochastic process theory. The representation of signals could fall into two categories:

- Deterministic Signals
- Random Signals

2.1 Deterministic signals

A deterministic discrete-time signal is characterized by a defined mathematical function of the time index n , with $n = 0, \pm 1, \pm 2, \dots$, such as:

$$x_{(n)} = e^{-\alpha n} \cos(\omega n) + u_{(n)} \quad (3)$$

where $u_{(n)}$ is the unit-step sequence and The response of a linear time-invariant filter to an input $x_{(n)}$ is given by:

$$y_{(n)} = x_{(n)} * h_{(n)} = \sum_{k=-\infty}^{\infty} x_{(k)} h_{(n-k)} \quad (4)$$

where $h_{(n)}$ is the impulse response of the filter. Knowing that The Z-transform and its inverse of a given sequence $x_{(n)}$ is defined as:

$$\begin{aligned} Z(x_{(n)}) &= X(Z) = \sum_{n=-\infty}^{\infty} x_{(n)} Z^{-n} \\ x_{(n)} &= \frac{1}{2\pi j} \oint_C X(Z) Z^{n-1} dZ \end{aligned} \quad (5)$$

where C is a counter clockwise closed contour in the region of convergence of $X(z)$ and encircling the origin of the z -plane as a result; by taking the Z-transform of both sides of equation (4), we will obtain

$$Y(Z) = H(Z) \times X(Z) \quad (6)$$

For finite finite-energy waveform, it is convenient to use the discrete-time Fourier transform defined as:

$$F(x_n) = X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_{(n)} e^{-j\omega n} \quad (7)$$

2.2 Random signals

In many real life situations, observations are made over a period of time and they are influenced by random effects, not just at a single instant but throughout the entire interval of time or sequence of times. In a "rough" sense, a random process is a phenomenon that varies to some degree unpredictably as time goes on. If we observed an entire time-sequence of the process on several different occasions, under presumably "identical" conditions, the resulting observation sequences, in general, would be different. A random variable (RV) is a rule (or function) that assigns a real number to every outcome of a random experiment,

while a stochastic random process is a rule (or function) that assigns a time function to every outcome of a random experiment. The elements of a stochastic process, $\{x_{(n)}\}$, for different value of time-index n , are in general complex-valued random variable that are characterized by their probability distribution functions. A stochastic random process is called stationary in the strict sense if all of its (single and joint) distribution function is independent of a shift in the time origin.

In this subsection we will be reviewing some useful definitions in stochastic process:

- Stochastic Average

$$m_x = E[x_{(n)}] \quad (8)$$

where E is the expected value of $x_{(n)}$.

- Autocorrelation function for a stochastic process $x_{(n)}$

$$\phi_{xx}(n, m) = E[x_{(n)}x_m^*] \quad (9)$$

where x^* denotes the complex conjugate of x and the symmetry properties of correlation function is:

$$\phi_{xx}(k) = \phi_{xx}^*(-k) \quad (10)$$

Furthermore a stochastic random process is called stationary in the wide sense if m_x and $\phi_{xx}(n, m)$ are independent of a shift in the time origin for any k, m , and n therefore,

$$m_x(n) = m_x(n+k) = \text{constant for all } n \quad (11)$$

and

$$\phi_{xx}(n, m) = \phi_{xx}(n+k, m+k) \quad (12)$$

which means $\phi_{xx}(n, m)$ depends only on the difference $n - m$ or

$$\phi_{xx}(k) = E[x_{(n)}x_{(n-k)}^*] \quad (13)$$

Special case

$$\phi_{xx}(0) = E[|x_{(n)}|^2] = \text{mean square of } x_{(n)} \quad (14)$$

The Z-transform of $\phi_{xx}(k)$ is given by

$$\Phi_{xx}(Z) = \sum_{k=-\infty}^{\infty} \phi_{xx}(k)Z^{-k} \quad (15)$$

where it can be easily shown that

$$\Phi_{xx}(Z) = \Phi_{xx}^*\left(\frac{1}{Z^*}\right) \quad (16)$$

Equation (16) implies that if $\Phi_{xx}(Z)$ is a rational function of z , then its poles and zeros must occur in complex-conjugate reciprocal pairs. Moreover, the points that belong to the region of convergence of $\Phi_{xx}(Z)$ also occur in complex-conjugate pairs which suggest that the region of convergence of $\Phi_{xx}(Z)$ must be of the form $|a| < |z| < \frac{1}{|a|}$ which will cover the unit circle $|z|=1$. By assuming that $\Phi_{xx}(Z)$ is convergent on the unit circle's contour C and by letting $Z = e^{j\omega}$ then,

$$\phi_{xx}(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{xx}(e^{j\omega}) d\omega \quad (17)$$

and if $m_x = 0$, we will have

$$\sigma_x^2 = E\left[|x_{(n)}|^2\right] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{xx}(e^{j\omega}) d\omega \quad (18)$$

- Cross-correlation function for two stochastic processes $x_{(n)}$ and $y_{(n)}$

$$\phi_{xy}(n, m) = E\left[x_{(n)} y_{(m)}^*\right] \quad (15)$$

and the symmetry properties of the correlation function is:

$$\phi_{xy}(k) = \phi_{yx}^*(-k) \quad (16)$$

The Z-transform of $\phi_{xy}(k)$ is given by

$$\Phi_{xy}(Z) = \sum_{k=-\infty}^{\infty} \phi_{xy}(k) Z^{-k} \quad (17)$$

where it can be easily shown that

$$\Phi_{xy}(Z) = \Phi_{yx}^*\left(\frac{1}{Z^*}\right) \quad (18)$$

- Auto-covariance function for stationary process is defined as:

$$\gamma_{xx}(k) = E\left[(x_{(n)} - m_x)(x_{(n-k)} - m_x)^*\right] = \phi_{xx}(k) - |m_x|^2 \quad (19)$$

and the symmetry properties of auto-covariance function

$$\gamma_{xx}(k) = \gamma_{xx}^*(-k) \quad (20)$$

Special case

$$\gamma_{xx}(0) = \sigma_x^2 = \text{variance of } x_{(n)} \quad (21)$$

- Cross-covariance function is defined as:

$$\gamma_{xy}(k) = E\left[(x_{(n)} - m_x)(y_{(n-k)} - m_y)^*\right] = \phi_{xy}(k) - m_x m_y^* \quad (22)$$

and the symmetry properties of the cross-covariance function is:

$$\gamma_{xy}(k) = \gamma_{yx}^*(-k) \quad (23)$$

2.2.1 Power spectral density

Consider the wide-sense stationary random process $\{x_{(n)}\}$ from which we will consider a window of $2N+1$ elements of $x_{(n)}$ such as

$$x_{N(n)} = \begin{cases} x_{(n)}, & -N \leq n \leq N \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

and the discrete-time Fourier transform of $x_{N(n)}$ is computed as:

$$X_N(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_{N(n)} e^{-j\omega n} = \sum_{n=-N}^N x_{(n)} e^{-j\omega n} \quad (25)$$

By conjugating both sides of equation 25 with:

$$X_N^*(e^{j\omega}) = \sum_{m=-N}^N x_{(m)}^* e^{j\omega m} \quad (26)$$

we will obtain

$$|X_N(e^{j\omega})|^2 = \sum_{n=-N}^N \sum_{m=-N}^N x_{(n)} x_{(m)}^* e^{-j\omega(n-m)} \quad (27)$$

and by taking the expected value of equation 27, we will have:

$$E\left[|X_N(e^{j\omega})|^2\right] = \sum_{n=-N}^N \sum_{m=-N}^N E\left[x_{(n)} x_{(m)}^*\right] e^{-j\omega(n-m)} \quad (28)$$

which after simplification by imposing $k = n - m$ will yield:

$$\frac{1}{2N+1} E\left[|X_N(e^{j\omega})|^2\right] = \sum_{k=-2N}^{2N} \left(1 - \frac{|k|}{2N+1}\right) \phi_{xx}(k) e^{-j\omega k} \quad (29)$$

By assuming that the summation on the right hand side of equation 29 will converge for large k then

$$\lim_{N \rightarrow \infty} \left(\frac{1}{2N+1} E\left[|X_N(e^{j\omega})|^2\right] \right) = \sum_{k=-\infty}^{\infty} \phi_{xx}(k) e^{-j\omega k} = \Phi_{xx}(e^{j\omega}) \quad (30)$$

The function $\Phi_{xx}(e^{j\omega})$ is called the power spectral density of the stochastic, wide-sense stationary process $\{x_{(n)}\}$ which is always real and non-negative.

2.2.2 Response of linear system to stochastic processes

Given the Linear Time-Invariant System (LTI) illustrated in figure 3 where we will assume that $\phi_{xx}(k)$ is known therefore, the Z-transform of $\phi_{xy}(k)$ could be elaborated as:

$$\Phi_{xy}(Z) = \sum_{k=-\infty}^{\infty} \phi_{xy}(k) Z^{-k} = \sum_{k=-\infty}^{\infty} E[x_{(n)} y_{(n-k)}^*] Z^{-k} \quad (31)$$

and

$$y_{(n)} = \sum_{l=-\infty}^{\infty} h_{(l)} x_{(n-l)} \quad (32)$$

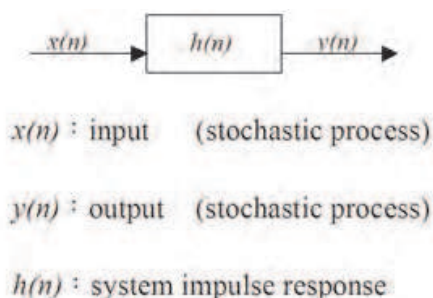


Fig. 3. Structure of LTI system

To be noted that the following expressions could be easily derived:

$$\Phi_{yy}(Z) = H(Z) H^* \left(\frac{1}{Z} \right) \Phi_{xx}(Z) \quad (33)$$

and

$$\begin{aligned} \Phi_{xy}(e^{j\omega}) &= H^*(e^{j\omega}) \Phi_{xx}(e^{j\omega}) \\ \Phi_{yx}(e^{j\omega}) &= H(e^{j\omega}) \Phi_{xx}(e^{j\omega}) \\ \Phi_{yy}(e^{j\omega}) &= |H(e^{j\omega})|^2 \Phi_{xx}(e^{j\omega}) \end{aligned} \quad (34)$$

3. Regression

Data fitting is one of the oldest adaptive systems which are a powerful tool, allowing predictions of the present or future events to be made based on information about the past or present events. The two basic types of regression are linear regression and multiple regressions.

3.1 Linear regression

The goal of linear regression is to adjust the values of slope and intercept to find the line that best predicts d from x (Figure 4) or in other words linear regression estimates how much d changes when x changes by one unit.

The slope quantifies the steepness of the line which is equals to the change in d for each unit change in x . If the slope is positive, d increases as d increases. If the slope is negative, d

decreases as d increases. The d intercept is the d value of the line when x equals zero. It defines the elevation of the line.

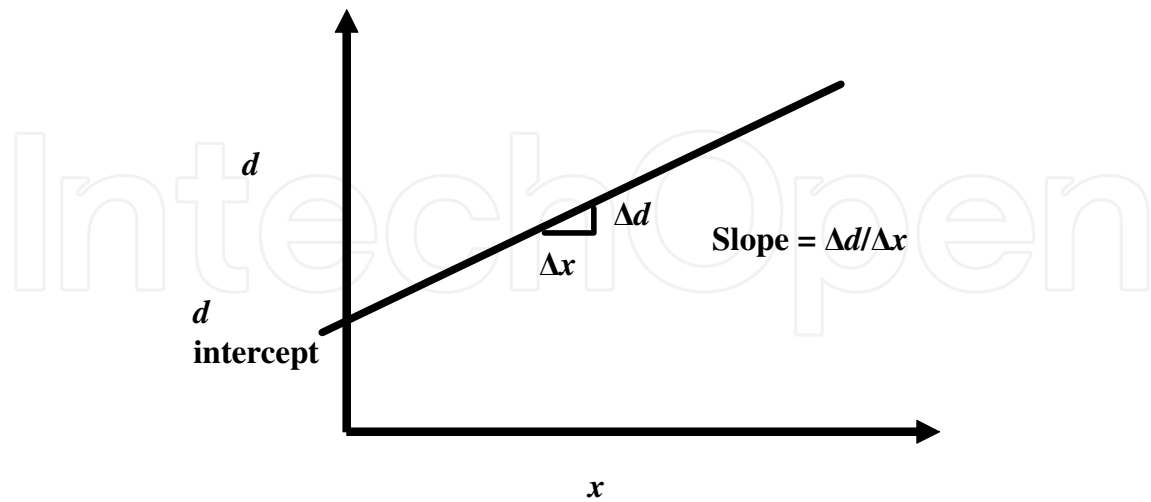


Fig. 4. Slope and Intercept.

The deviation from the straight line which represents the best linear fits of a set of data as shown in figure 5 is expressed as:

$$d \approx w \times x + b \quad (35)$$

or more specifically,

$$d_{(n)} = w \times x_{(n)} + b + e_{(n)} = y_{(n)} + e_{(n)} \quad (36)$$

where $e_{(n)}$ is the instantaneous error that is added to $y_{(n)}$ (the linearly fitted value), w is the slope and b is the y intercept (or bias). More precisely, the goal of regression is to minimize the sum of the squares of the vertical distances of the points from the line. The problem can be solved by a linear system with only two free parameters, the slope w and the bias b .

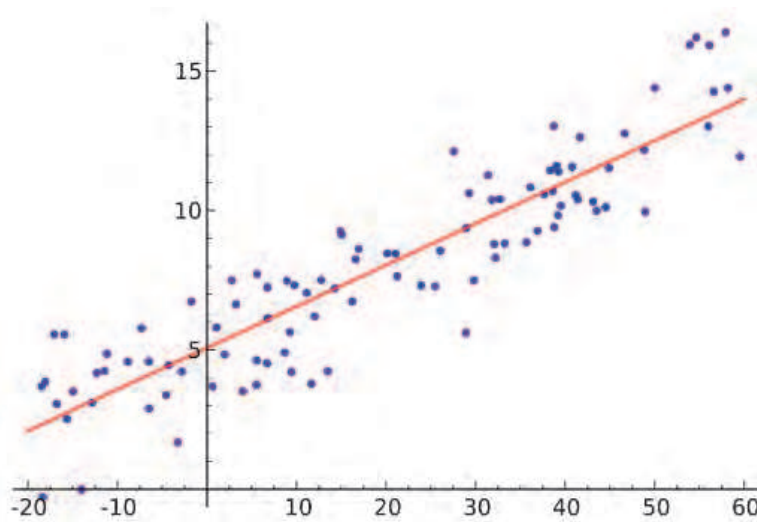


Fig. 5. Example of linear regression with one independent variable

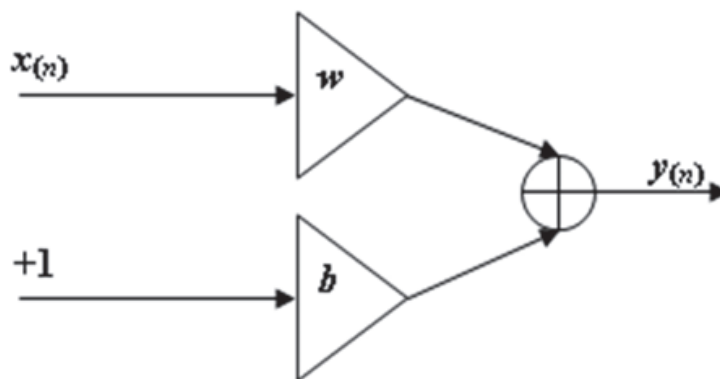


Fig. 6. Linear Regression Processing Element.

Figure 6 is called the Linear Regression Processing Element (LRPE) which is built from two multipliers and one adder. The multiplier w scales the input, and the multiplier b is a simple bias, which can also be thought of as an extra input connected to the value +1. The parameters (b, w) have different functions in the solution.

3.1.1 Least squares for linear model

Least squares solves the problem by finding the best fitted line to a set of data; for which the sum of the square deviations (or residuals) in the d direction are minimized (figure 7).

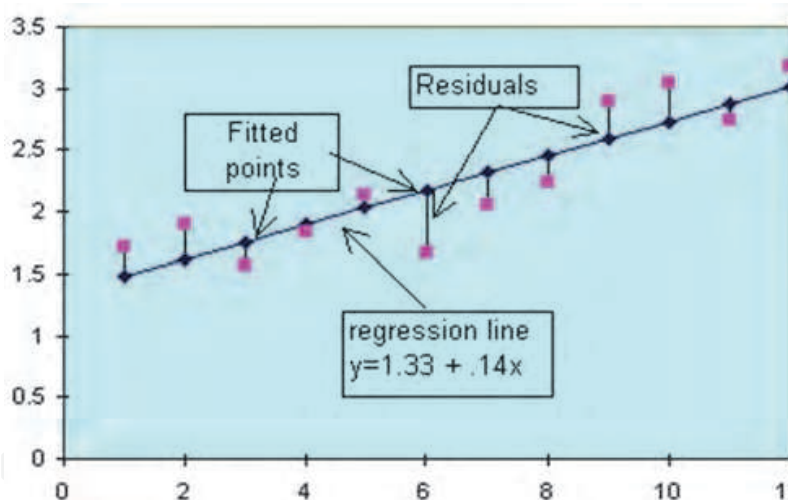


Fig. 7. Regression line showing the deviations.

The goal is to find a systematic procedure to find the constants b and w which minimizes the error between the true value $d_{(n)}$ and the estimated value $y_{(n)}$ which is called the linear regression.

$$d_{(n)} - (b + wx_{(n)}) = d_{(n)} - y_{(n)} = e_{(n)} \quad (37)$$

The best fitted line to the data is obtained by minimizing the error $e_{(n)}$ which is computed by the mean square error (MSE) that is a widely utilized as performance criterion

$$\xi = \frac{1}{2N} \sum_{n=1}^N e_{(n)}^2 \quad (38)$$

where N is the number of observations and ξ is the mean square error.

Our goal is to minimize ξ analytically, which can be achieved by taking the partial derivative of this quantity with respect to the unknowns and equate the resulting equations to zero, i.e.

$$\begin{cases} \frac{\partial \xi}{\partial b} = 0 \\ \frac{\partial \xi}{\partial w} = 0 \end{cases} \quad (39)$$

which yields after some manipulation to:

$$b = \frac{\sum_n x_{(n)}^2 \sum_n d_{(n)} - \sum_n x_{(n)} \sum_n x_{(n)} d_{(n)}}{N \left[\sum_n (x_{(n)} - \bar{x})^2 \right]} \quad w = \frac{\sum_n (x_{(n)} - \bar{x})(d_{(n)} - \bar{d})}{\sum_n (x_{(n)} - \bar{x})^2} \quad (40)$$

where the bar over the variable means the mean value and the procedure to determine the coefficients of the line is called the least square method.

3.1.2 Search procedure

The purpose of least squares is to find parameters $(b, w_1, w_2, \dots, w_p)$ that minimize the difference between the system output $y_{(n)}$ and the desired response $d_{(n)}$. So, regression is effectively computing the optimal parameters of an interpolating system which predicts the value of d from the value of x .

Figure 8 shows graphically the operation of adapting the parameters of the linear system in which the system output y is always a linear combination of the input x with a certain bias b according to the equation $y = wx + b$. Changing b modifies the y intersect, while changing w modifies the slope. The goal of linear regression is to adjust the position of the line such that the average square difference between the y values (on the line) and the cloud of points $d_{(n)}$ i.e. the error $e_{(n)}$, is minimized.

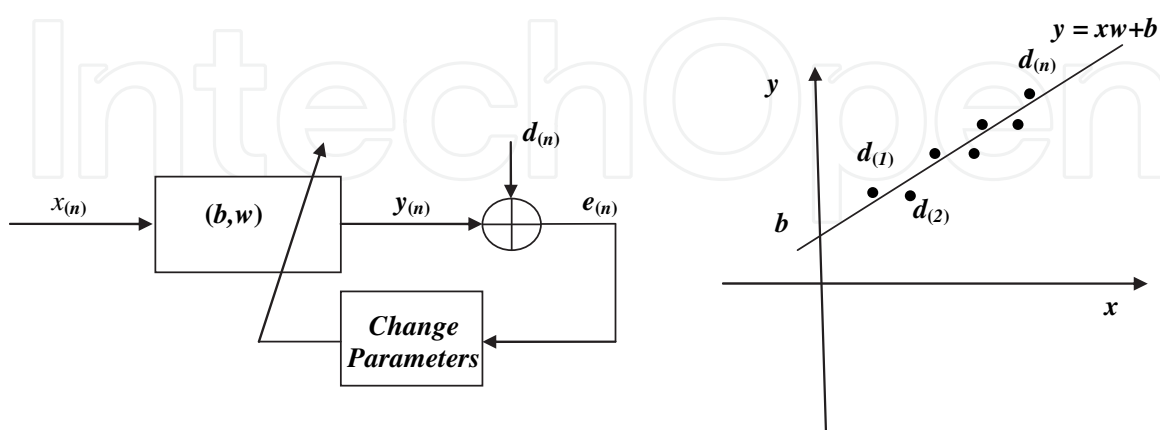


Fig. 8. Regression as a linear system design problem.

The key point is to recognize the information transmitted by the error which can be used to optimally place the line and this could be achieved by including a subsystem that accepts

the error and modifies the parameters of the system. Thus, the error $e_{(n)}$ is fed-back to the system and indirectly affects the output through a change in the parameters (b, w) . With the incorporation of the mechanism that automatically modifies the system parameters, a very powerful linear system can be built that will constantly seek optimal parameters. Such systems are called Adaptive systems, and are the focus of this chapter.

3.2 Multiple regression

With the same reasoning as above, the regression for multiple variables could be derived as:

$$e_{(i)} = d_{(i)} - \left(b + \sum_{k=0}^p w_{(k)} x_{(i,k)} \right) = d_{(i)} - \sum_{k=0}^p w_{(k)} x_{(i,k)} \quad (41)$$

where for this case is to find the coefficient vector W that minimizes the MSE of $e_{(i)}$ over the i samples (Figure 9).

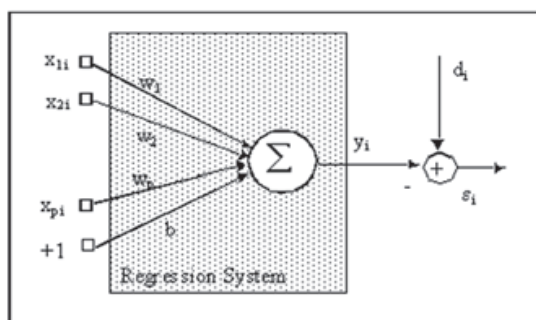


Fig. 9. Regression system for multiple inputs.

The mean square error (MSE) becomes for this case:

$$\xi = \frac{1}{2N} \sum_n \left(d_{(n)} - \sum_{k=0}^p w_{(k)} x_{(n,k)} \right)^2 \quad (42)$$

where the solution of this equation can be found by taking the derivatives of ξ with respect to the unknowns $(w_{(k)})$, and equating the result to zero which will yield to the famous *normal matrix equation* expressed as:

$$\sum_n x_{(n,j)} d_{(n)} = \sum_{k=0}^p w_{(k)} \sum_n x_{(n,k)} x_{(n,j)} \quad (43)$$

for $j = 0, 1, \dots, p$.

By defining:

$$R_{(n,j)} = \frac{1}{N} \sum_n x_{(n,k)} x_{(n,j)} \quad (44)$$

as the autocorrelation function,

$$P_{(j)} = \frac{1}{N} \sum_n x_{(n,j)} d_{(n)} \quad (45)$$

as the cross-correlation of the input x for index j and the desired response y and substituting these definitions into Eq. (43), the set of normal equations can be written simply as:

$$P = RW^* \quad \text{or} \quad W^* = R^{-1}P \quad (46)$$

where W is a vector with the $p+1$ weights w_i in which W^* represents the value of the vector for the optimum (minimum) solution. The solution of the multiple regression problems can be computed analytically as the product of the inverse of the autocorrelation of the input samples multiplied by the cross-correlation vector of the input and the desired response.

All the concepts previously mentioned for linear regression can be extended to the multiple regression case where J in matrix notation is illustrated as:

$$\xi = \left[W^T RW - 2P^T W + \sum_n \frac{d_{(n)}^2}{N} \right] \quad (47)$$

where T means the transpose and the values of the coefficients that minimize the solution are:

$$\nabla \xi = 0 = RW - P \quad \text{or} \quad W^* = R^{-1}P \quad (48)$$

4. Wiener filters

The Wiener filter is a filter proposed by Norbert Wiener during the 1940s and published in 1949 [8]. Its purpose was to reduce the amount of noise present in a signal in comparison with an estimation of the desired noiseless signal. This filter is an MSE-optimal stationary linear filter which was mainly used for images degraded by additive noise and blurring. The optimization of the filter is achieved by minimizing mean square error defined as the difference between the output filter and the desired response (Figure 10) which is known as the cost function expressed as:

$$\xi = E[|e_n|^2] \quad (49)$$

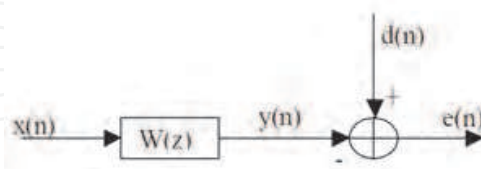


Fig. 10. block schematic of a linear discrete-time filter $W(z)$ for estimating a desired signal $d_{(n)}$ based on an excitation $x_{(n)}$ where $d_{(n)}$ and $x_{(n)}$ are random processes.

In signal processing, a causal filter is a linear and time-invariant causal system. The word *causal* indicates that the filter output depends only on past and present inputs. A filter whose output also depends on future inputs is non-causal. As a result two cases should be considered for the optimization of the cost function (equation 49).

- The filter $W(Z)$ is causal and or FIR (Finite Impulse Response).
- The filter $W(Z)$ is non-causal and or IIR (Infinite Impulse Response Filter).

4.1 Wiener filter - the transversal filter

Let \overline{W} be the transversal filter's tap weights illustrated in figure 11 which is defined as:

$$\overline{W} = [w_0 \quad w_1 \quad \dots \quad w_{N-1}]^T \quad (50)$$

where T denotes the vector transpose and

$$\overline{X} = [x_0 \quad x_1 \quad \dots \quad x_{N-1}]^T \quad (51)$$

be the input vector signal where two cases should be treated separately depending on the required application:

- Real valued input signal
- Complex valued input signal

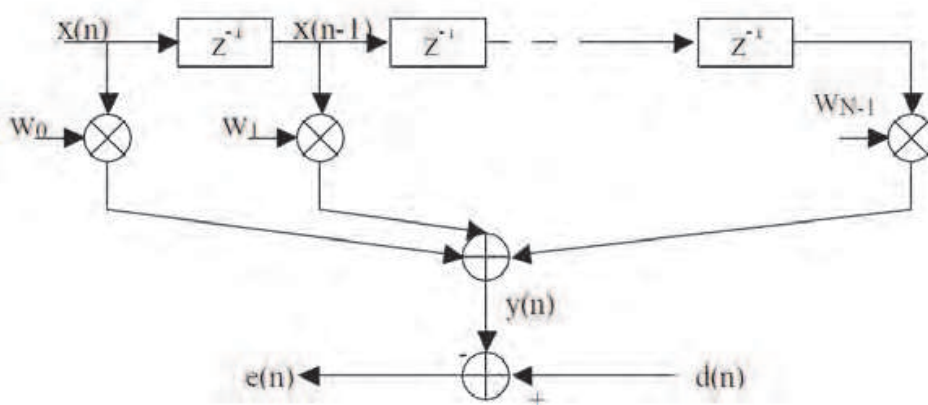


Fig. 11. Block Diagram of the Transversal Filter.

4.1.1 Wiener filter - the transversal filter - real-valued input signal

The output filter could be expressed as:

$$y_{(n)} = \sum_{i=0}^{N-1} w_i x_{(n-i)} = \overline{W}^T \overline{X}_{(n)} = \overline{X}_{(n)}^T \overline{W} \quad (52)$$

Where

$$e_{(n)} = d_{(n)} - y_{(n)} = d_{(n)} - \overline{X}_{(n)}^T \overline{W} \quad (53)$$

The performance or cost function expressed in equation 49 could be elaborated as:

$$\begin{aligned} \xi &= E[|e_n|^2] = E\left[\left(d_{(n)} - \overline{W}^T \overline{X}_{(n)}\right)\left(d_{(n)} - \overline{X}_{(n)}^T \overline{W}\right)\right] \\ &= E[d_{(n)}^2] - \overline{W}^T E[\overline{X}_{(n)} d_{(n)}] - E[\overline{X}_{(n)}^T d_{(n)}] \overline{W} + \overline{W}^T E[\overline{X}_{(n)} \overline{X}_{(n)}^T] \overline{W} \end{aligned} \quad (54)$$

By examining equation 54 we could easily notice that $E[\overline{X}_{(n)} d_{(n)}]$ is the cross correlation function which will be defined as:

$$\bar{P} \equiv E\left[\overline{X_{(n)}}d_{(n)}\right] = [p_0 \quad p_1 \quad \cdots \quad p_{N-1}]^T \quad (55)$$

and with the same reasoning as above the autocorrelation function $E\left[\overline{X_{(n)}}\overline{X_{(n)}}^T\right]$ could be expressed as:

$$R \equiv E\left[\overline{X_{(n)}}\overline{X_{(n)}}^T\right] = \begin{bmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,N-1} \\ r_{1,0} & r_{1,1} & \cdots & r_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ r_{N-1,0} & r_{N-1,1} & \cdots & r_{N-1,N-1} \end{bmatrix} \quad (56)$$

Knowing that $E\left[d_{(n)}\overline{X_{(n)}}^T\right] = \bar{P}^T$ and $\overline{W}^T \bar{P} = \bar{P}^T \overline{W}$ therefore, the cost function would be:

$$\xi = E\left[d_{(n)}^2\right] - 2\overline{W}^T \bar{P} + \overline{W}^T R \overline{W} \quad (57)$$

where \overline{W} in the quadratic function expressed in equation 43 will have a global minimum if and only if R is a positive definite Matrix.

The gradient method is the most commonly used method to compute the tap weights that minimizes the cost function therefore,

$$\nabla \xi = \frac{\partial \xi}{\partial w_i} = \left[\frac{\partial \xi}{\partial w_0} \quad \frac{\partial \xi}{\partial w_1} \quad \cdots \quad \frac{\partial \xi}{\partial w_{N-1}} \right] = 0 \quad (58)$$

which yield after expanding equation 57 to:

$$\begin{aligned} \frac{\partial \xi}{\partial w_i} &= 0 \\ 0 &= -2p_i + \sum_{l=0}^{N-1} w_l (r_{li} + r_{il}) \\ \sum_{l=0}^{N-1} w_l (r_{li} + r_{il}) &= 2p_i \end{aligned} \quad (59)$$

where we have assumed in the expanded equation $\xi = E\left[d_{(n)}^2\right] - 2\sum_{l=0}^{N-1} p_l w_l + \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} w_l w_m r_{lm}$ that:

$$\sum_{l=0}^{N-1} \sum_{m=0}^{N-1} w_l w_m r_{lm} = \sum_{\substack{l=0 \\ l \neq i}}^{N-1} \sum_{\substack{m=0 \\ m \neq i}}^{N-1} w_l w_m r_{lm} + w_i \sum_{\substack{m=0 \\ m \neq i}}^{N-1} w_m r_{im} + w_i^2 r_{ii} \quad (60)$$

Knowing that $r_{li} = r_{il}$ due to the symmetry property of the autocorrelation function of real-valued signal equation 45 could be expressed as:

$$\sum_{l=0}^{N-1} w_l r_{il} = p_i \quad (61)$$

for $i = 0, 1, \dots, N - 1$ which could be expressed in matrix notation as:

$$R\overline{W}_{op} = \overline{P} \quad (62)$$

known as Weiner-Hopf equation, whose solution for the optimal tap-weight vector \overline{W}_{op} and by assuming that R has an inverse matrix, is:

$$\overline{W}_{op} = R^{-1}\overline{P} \quad (63)$$

Finally, the minimum value of the cost function can be expressed as:

$$\xi_{\min} = E[d_{(n)}^2] - \overline{W}_{op}^T \overline{P} = E[d_{(n)}^2] - \overline{W}_{op}^T R \overline{W}_{op} = E[d_{(n)}^2] - \overline{P}^T R^{-1} \overline{P} \quad (64)$$

4.1.2 Wiener filter – the transversal filter - complex-valued input signal

The data transmission of the basebands QPSK and the QAM is complex valued random signals where the filter's tap weight vector is also assumed to be complex therefore, the cost function for this case could be expressed as:

$$\xi = E[|e_{(n)}|^2] = E[e_{(n)} e_{(n)}^*] \quad (65)$$

and the gradient of the cost function would be:

$$\nabla_{w_i}^c = E[e_{(n)} \nabla_{w_i}^c e_{(n)}^* + \nabla_{w_i}^c e_{(n)}^* e_{(n)}] \quad (66)$$

By examining figure 11 we could easily notice that equation 53 could be formulated as:

$$e_{(n)} = d_{(n)} - \sum_{k=0}^{N-1} w_{(k)} x_{(n-k)} \quad (67)$$

and since d is independent of w and since

$$\begin{aligned} \nabla_{w_i}^c e_{(n)} &= -x_{(n-i)} \nabla_{w_i}^c w_i \\ \nabla_{w_i}^c e_{(n)}^* &= -x_{(n-i)}^* \nabla_{w_i}^c w_i^* \\ \nabla_{w_i}^c w_i &= \frac{\partial w_i}{\partial w_{i,R}} + j \frac{\partial w_i}{\partial w_{i,I}} = 1 + j(j) = 0 \\ \nabla_{w_i}^c w_i^* &= \frac{\partial w_i^*}{\partial w_{i,R}^*} + j \frac{\partial w_i^*}{\partial w_{i,I}^*} = 1 + j(-j) = 2 \end{aligned} \quad (68)$$

Where the sub-indices R and I refers to the real part and imaginary part of the complex number therefore, the gradient of the cost function (Equation 66) would be:

$$\nabla_{w_i}^c \xi = -2E[e_{(n)} x_{(n-i)}^*] \quad (69)$$

The optimum filter's tap-weights $e_{0(n)}$ are obtained by setting the complex gradient of equation 69 to zero yielding:

$$E[e_{o(n)}x_{(n-i)}^*] = 0 \quad (70)$$

By defining

$$\begin{aligned} \bar{x}_{(n)} &\equiv [x_{(n)} \quad x_{(n-1)} \quad \cdots \quad x_{(n-N+1)}]^T \equiv [x_{(n)}^* \quad x_{(n-1)}^* \quad \cdots \quad x_{(n-N+1)}^*]^H \\ \bar{w}_{(n)} &\equiv [w_0^* \quad w_1^* \quad \cdots \quad w_{N-1}^*]^T \equiv [w_{(0)} \quad w_1 \quad \cdots \quad w_{N-1}]^H \end{aligned} \quad (71)$$

where H denotes the complex-conjugate transpose or Hermitian and by re-writing equation 67 as:

$$e_{(n)} = d_{(n)} - \bar{w}_o^H x_{(n)} \quad (72)$$

whose solution is:

$$R\bar{w}_o = \bar{p} \quad (73)$$

where

$$\begin{aligned} R &= E[\bar{x}_{(n)}\bar{x}_{(n)}^H] \\ \bar{p} &= E[\bar{x}_{(n)}d_{(n)}^*] \end{aligned} \quad (74)$$

Equation 73 is known as Weiner-Hopf equation for the complex valued signals and the minimum of the cost function will be:

$$\xi_{\min} = E[d_{(n)}^2] - \bar{w}_o^H R\bar{w}_o \quad (75)$$

5. Least Mean Square algorithm (LMS algorithm)

The purpose of least squares is to find the optimal filter's tap weights that minimize the difference between the system output $y_{(n)}$ and the desired response $d_{(n)}$ or in other words to minimize the cost function. Instead of solving the Weiner-Hopf equation in order to obtain the optimal filter's tap weights as seen previously, there exist other iterative methods which employ an iterative search method that starts with an *arbitrary initial weight* $\bar{W}_{(0)}$, then a recursive search method that may require many iterations in order to converge to the optimal filter's tap weights \bar{W}_o . The most important iterative methods are the gradient based iterative methods which are listed below:

- Steepest Descent Method
- Newton's Method

5.1 Steepest descent method

The steepest descent method (also known as the gradient method) is the simplest example of a gradient based method that *minimizes a function of several variables*. This process is employing an iterative search method that starts with an *arbitrary initial weight* $\bar{W}_{(0)}$, and then at the k^{th} iteration $\bar{W}_{(k)}$ is updated according to the following equation:

$$\bar{W}_{(k+1)} = \bar{W}_{(k)} - \mu \nabla_k \xi \quad (76)$$

where μ is positive known as the step size and $\nabla_k \xi$ denotes the gradient vector $\nabla \xi = 2R\bar{W} - 2\bar{p}$ evaluated at the point $\bar{W} = \bar{W}_{(k)}$. Therefore, equation 76 could be formulated as:

$$\bar{W}_{(k+1)} = \bar{W}_{(k)} - 2\mu(R\bar{W}_{(k)} - \bar{p}_{(k)}) = (1 - 2\mu R)(\bar{W}_{(k)} - \bar{W}_0) \quad (77)$$

Knowing that the auto-correlation matrix R may be diagonalized by using the unitary similarity decomposition:

$$R = Q\Lambda Q^T \quad (78)$$

where Λ is a diagonal matrix consisting of the auto-correlation matrix R eigenvalues $\lambda_0 \lambda_1 \dots \lambda_{N-1}$ and the columns of Q contain the corresponding orthonormal eigenvectors and by defining the vector $\bar{v}_{(k)}$ as:

$$\bar{v}_{(k)} = \bar{W}_{(k)} - \bar{W}_0 \quad (79)$$

As a result equation 77 could be expressed as:

$$\bar{v}_{(k+1)} = (I - 2\mu Q\Lambda Q^T)\bar{v}_{(k)} \quad (80)$$

which after mathematical manipulation and by using the vector transformation $\bar{v}'_{(k)} = Q^T \bar{v}_{(k)}$ could be written as:

$$v'_{i(k+1)} = (1 - 2\mu\lambda_i) v'_{i(k)} \quad (81)$$

For $i = 0, 1, 2, \dots, N - 1$ and $\bar{v}'_{(k)} = [v'_{0(k)} \quad v'_{1(k)} \quad \dots \quad v'_{N-1(k)}]$ that yields:

$$v'_{i(k)} = (1 - 2\mu\lambda_i)^k v'_{i(0)} \quad (82)$$

Equation 82 converges to zero if and only if the step-size parameter μ is selected so that:

$$|1 - 2\mu\lambda_i| < 1 \quad (83)$$

which means

$$0 < \mu < \lambda_i \quad (84)$$

for all i or equivalently

$$0 < \mu < \lambda_{\max} \quad (85)$$

where λ_{\max} is the maximum of the eigenvalues $\lambda_0 \lambda_1 \dots \lambda_{N-1}$.

5.2 Newton's method

By replacing the scalar step-size μ with a matrix step-size given by μR^{-1} in the steepest descent algorithm (equation 77) and by using $\bar{p} = R\bar{W}_0$, the resulting algorithm is:

$$\bar{W}_{(k+1)} = \bar{W}_{(k)} - \mu R^{-1} \nabla_k \xi \quad (86)$$

Substituting $\nabla \xi = 2R\bar{W} - 2\bar{p}$ in equation 86 we will get:

$$\begin{aligned} \bar{W}_{(k+1)} &= \bar{W}_{(k)} - 2\mu R^{-1} (R\bar{W}_{(k)} - \bar{p}) \\ &= (1 - 2\mu) \bar{W}_{(k)} + 2\mu R^{-1} \bar{p} \\ &= (1 - 2\mu) \bar{W}_{(k)} + 2\mu \bar{W}_0 \end{aligned} \quad (87)$$

and by subtracting \bar{W}_0 from both sides of equation 87 will yield:

$$\bar{W}_{(k+1)} - \bar{W}_0 = (1 - 2\mu) (\bar{W}_{(k)} - \bar{W}_0) \quad (88)$$

where in actual implementation of adaptive filters, the exact values of $\nabla_k \xi$ and R^{-1} are not available and have to be estimated.

5.3 Least Mean Square (LMS) algorithm

The LMS algorithm is a practical scheme for realizing Wiener filters, without explicitly solving the Wiener-Hopf equation and this was achieved in the late 1960's by Widrow [2] who proposed an extremely elegant algorithm to estimate the gradient that revolutionized the application of gradient descent procedures by *using the instantaneous value of the gradient as the estimator for the true quantity* which means replacing the cost function $\xi = E[e_{(n)}^2]$ by its instantaneous coarse estimate $\hat{\xi} = e_{(n)}^2$ into steepest descent algorithm. By doing so, we will obtain

$$\bar{W}_{(n+1)} = \bar{W}_{(n)} - \mu \nabla e_{(n)}^2 \quad (89)$$

where $\bar{W}_{(n)} = [w_{0(n)} \quad w_{1(n)} \quad \dots \quad w_{N-1(n)}]^T$ and $\nabla = \left[\frac{\partial}{\partial w_0} \quad \frac{\partial}{\partial w_1} \quad \dots \quad \frac{\partial}{\partial w_{N-1}} \right]^T$.

The i^{th} element of the gradient vector $\nabla e_{(n)}^2$ is:

$$\frac{\partial e_{(n)}^2}{\partial w_i} = 2e_{(n)} \frac{\partial e_{(n)}}{\partial w_i} = -2e_{(n)} \frac{\partial y_{(n)}}{\partial w_i} = -2e_{(n)} x_{(n-i)} \quad (90)$$

which leads to:

$$\nabla e_{(n)}^2 = -2e_{(n)} \bar{x}_{(n)} \quad (91)$$

that will be replaced in equation 89 to obtain:

$$\overline{W}_{(n+1)} = \overline{W}_{(n)} + 2\mu e_{(n)} \overline{x}_{(n)} \quad (92)$$

where $\overline{x}_{(n)} = [x_{(n)} \quad x_{(n-1)} \quad \dots \quad x_{(n-N+1)}]^T$.

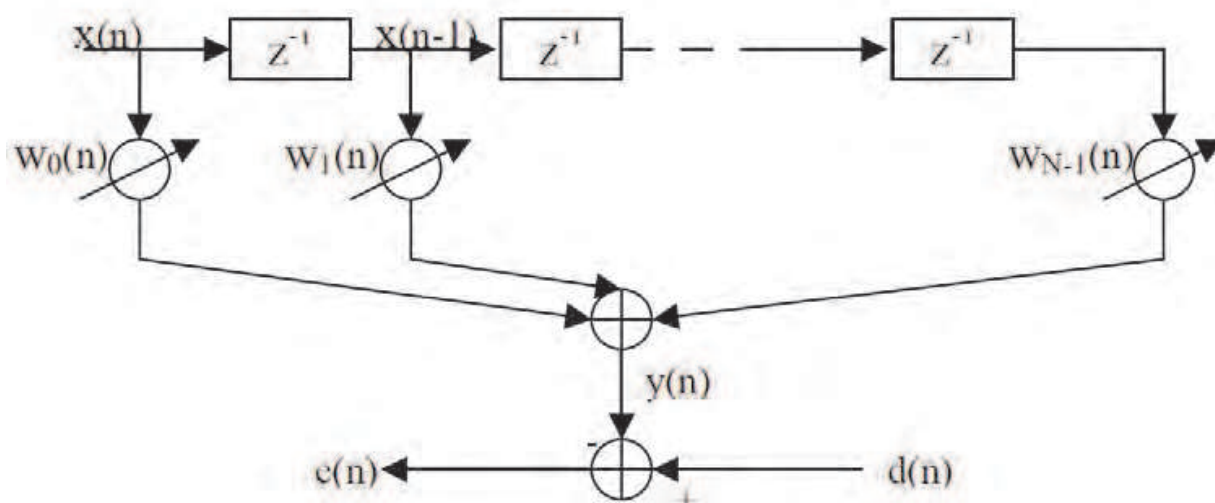


Fig. 12. LMS Filter Structure

Equation 92 is known as the LMS recursion and the summary of the LMS algorithm illustrated on figure 12 will be as follow:

- Input
- tap-weight vector, $\overline{W}_{(n)}$
- input vector, $\overline{x}_{(n)}$
- desired output, $d_{(n)}$
- Output
- Filter output, $y_{(n)}$
- Tap-weight vector update, $\overline{W}_{(n+1)}$

1. Filtering: $y_{(n)} = \overline{W}_{(n)}^T \overline{x}_{(n)}$
2. Error estimation: $e_{(n)} = d_{(n)} - y_{(n)}$
3. Tap-weight vector adaptation: $\overline{W}_{(n+1)} = \overline{W}_{(n)} + 2\mu e_{(n)} \overline{x}_{(n)}$

6. Classifying adaptive filtering applications

Various applications of adaptive filtering differ in the manner in which the desired response is extracted. In this context, we may distinguish four basic classes of adaptive filtering applications (depicted in Figures 13 to 16, which follow):

- Identification
- Inverse Modeling
- Prediction
- Interference Cancelling

Adaptive Filtering Class	Application
Identification	System Identification Layered Earth Modeling
Inverse Modeling	Predictive Convolution Adaptive Equalization
Prediction	Linear Prediction Coding Adaptive Differential PCM Auto Regressive Spectrum Analysis Signal Detection
Interference Cancelling	Adaptive Noise Cancelling Echo Cancellation Radar Polarimetry Adaptive Beam-forming

Table 1. Adaptive Filtering Applications

The following notations are used in Figures 11-15:

u = input applied to the adaptive filter

y = output of the adaptive filter

d = desired response

$e = d - y$ = estimation error

The functions of the four basic classes of adaptive filtering applications are summarized as follow:

- Identification (Figure 13).

The notion of a mathematical model is fundamental to sciences and engineering. In the class of applications dealing with identification, an adaptive filter is used to provide a linear model that represents the best fit to an unknown plant. The plant and the adaptive filter are driven by the same input. The plant output supplies the desired responses for the adaptive filter. If the plant is dynamic in nature, the model will be time varying.

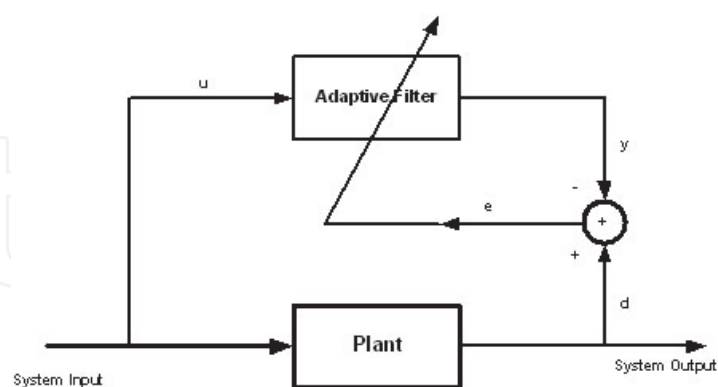


Fig. 13. Identification

- Inverse Modeling (Figure 14).

In this second class of applications, the adaptive filter provides an inverse model representing the best fit to an unknown noisy plant. Ideally, the inverse model has a transfer function equal to the reciprocal of the plant's transfer function. A delayed version of the plant input constitutes the desired response for the adaptive filter. In some applications, the plant input is used without delay as the desired response.

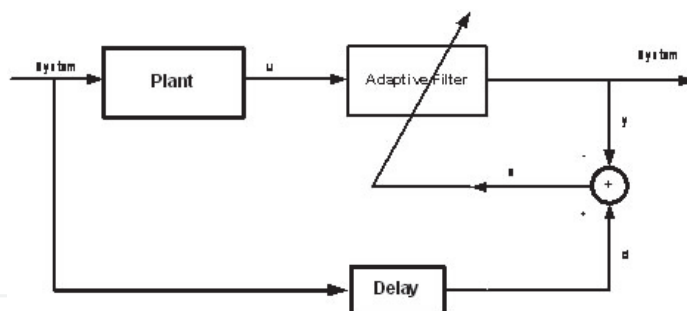


Fig. 14. Inverse Modeling

- Prediction (Figure 15).

In this example, the adaptive filter provides the best prediction of the present value of a random signal. The present value of the signal serves the purpose of a desired response for the adaptive filter. Past values of the signal supply the input applied to the adaptive filter. Depending on the application of interest, the adaptive filter output or the estimation error may service as the system output. In the first case, the system operates as a predictor; in the latter case, it operates as a prediction error filter.

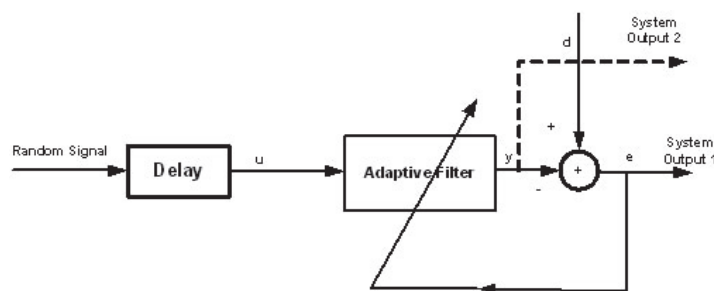


Fig. 15. Prediction

- Interference Cancelling (Figure 16).

In this final class of applications, the adaptive filter is used to cancel unknown interference contained in a primary signal, with the cancellation being optimized in some sense. The primary signal serves as the desired response for the adaptive filter. A reference signal is employed as the input to the adaptive filter. The reference signal is derived from a sensor or set of sensors located in relation to the sensor(s) supplying the primary signal in such a way that the information-bearing signal component is weak or essentially undetectable.

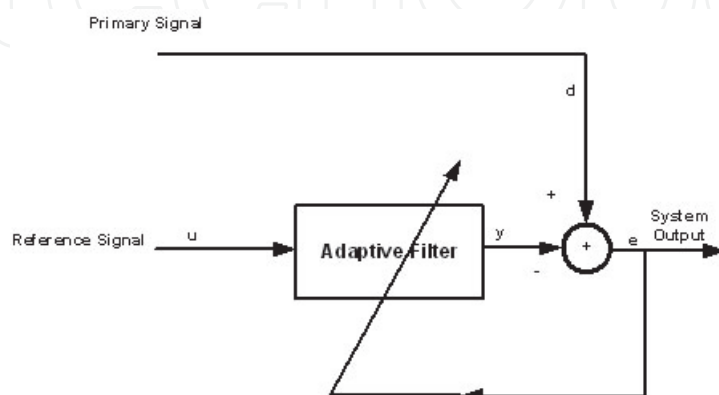


Fig. 16. Interference Cancelling

7. Active noise suppressor ANC

In this subsection we propose a new approach of noise control named ANS, where we propose stability-guaranteed algorithm for an adaptive filters, which can be derived on a basis of the strictly positive real property of the error model treated in adaptive system theory. It is important to assure the stability of the adaptive system especially in presence of unknown disturbances and mismatch in the order of the adaptive filter. Experimental results, performed on real mining noise, validate the effectiveness of the proposed stable algorithm [9].

7.1 Sounds and hearing

When sound waves from a point source strike a plane wall, they produce reflected circular wave fronts as if there were an "image" of the sound source at the same distance on the other side of the wall. If something obstructs the direct sound from the source from reaching your ear, then it may sound as if the entire sound is coming from the position of the "image" behind the wall. This kind of sound imaging follows the same laws of reflection as your image in a plane mirror (figure 17). The reflection of sound follows the law "angle of incidence equals angle of reflection" as the light does and other waves or the bounce of a billiard ball off the bank of a table figure (18).

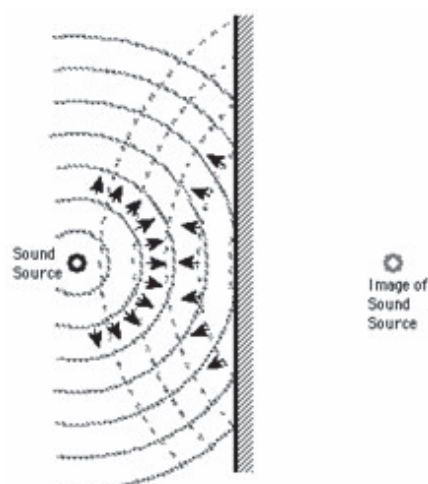


Fig. 17. Point source of sound reflecting from a plane surface.

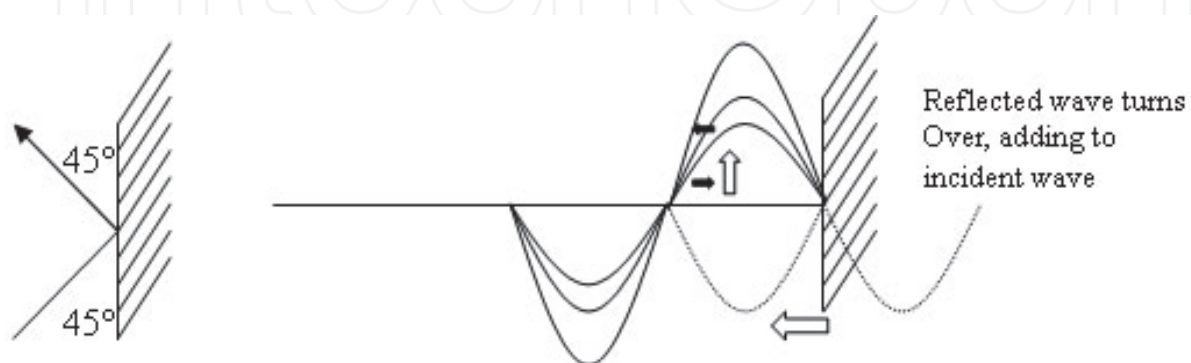


Fig. 18. Wave reflection

The main item of note about sound reflections off of hard surfaces is the fact that they undergo a 180-degree phase change upon reflection. This can lead to resonance such as standing waves in rooms. It also means that the sound intensity near a hard surface is enhanced because the reflected wave adds to the incident wave, giving pressure amplitude that is twice as great in a thin "pressure zone" near the surface. This is used in pressure zone microphones to increase sensitivity. The doubling of pressure gives a 6-decibel increase in the signal picked up by the microphone figure (19). Since the reflected wave and the incident wave add to each other while moving in opposite directions, the appearance of propagation is lost and the resulting vibration is called a standing wave. The modes of vibration associated with resonance in extended objects like strings and air columns have characteristic patterns called standing waves. These standing wave modes arise from the combination of reflection and interference such that the reflected waves interfere constructively with the incident waves. An important part of the condition for this constructive interference is the fact that the waves change phase upon reflection from a fixed end. Under these conditions, the medium appears to vibrate in segments or regions and the fact that these vibrations are made up of traveling waves is not apparent - hence the term "standing wave" figure (19).

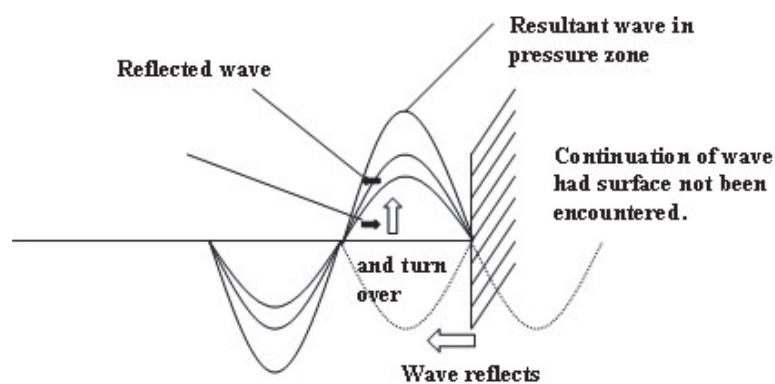


Fig. 19. Resultant wave in pressure zone.

Two traveling waves, which exist in the same medium, will interfere with each other figure (20). If their amplitudes add, the interference is said to be constructive interference, and destructive interference if they are "out of phase" and subtract figure (21). Patterns of destructive and constructive interference may lead to "dead spots" and "live spots" in auditorium acoustics.

Interference of incident and reflected waves is essential to the production of resonant standing waves figure (22).

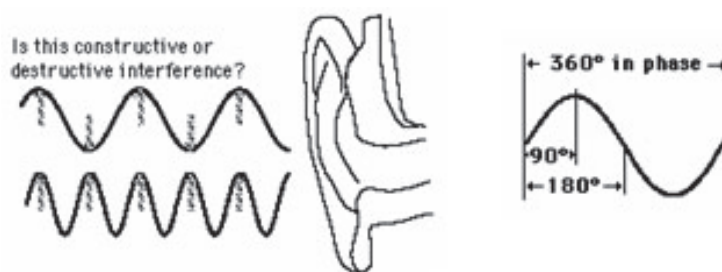


Fig. 20. Interference of Sound

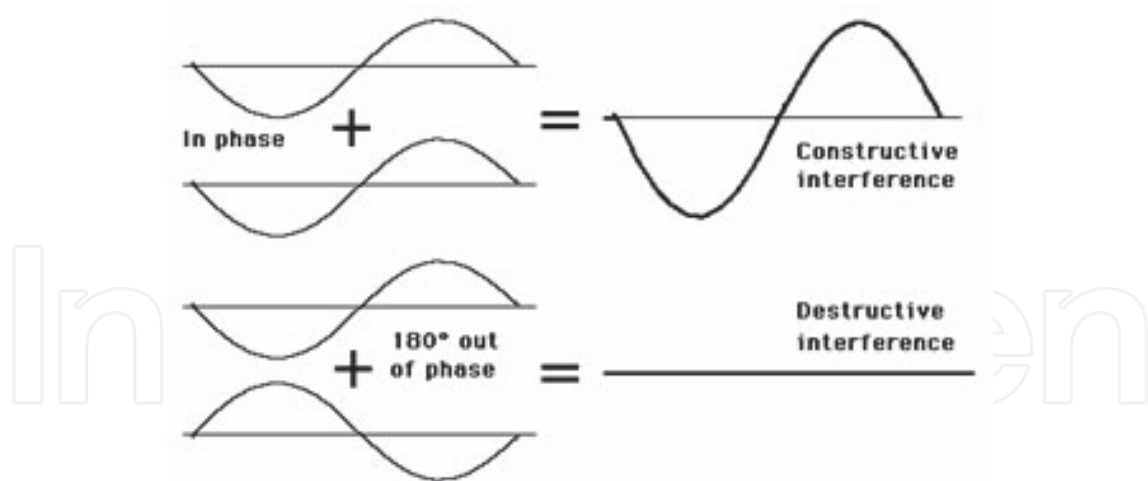


Fig. 21. Interference and Phase

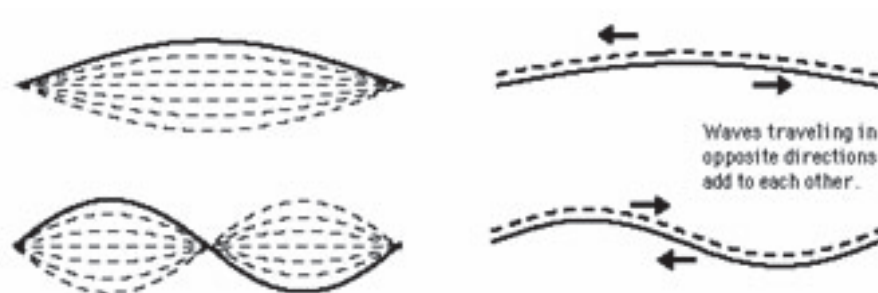


Fig. 22. The fundamental and second harmonic standing waves for a stretched string.

The sound intensity from a point source of sound will obey the inverse square law if there are no reflections or reverberation figure (23). Any point source, which spreads its influence equally in all directions without a limit to its range, will obey the inverse square law. This comes from strictly geometrical considerations. The intensity of the influence at any given radius r is the source strength divided by the area of the sphere. Being strictly geometric in its origin, the inverse square law applies to diverse phenomena. Point sources of gravitational force, electric field, light, sound or radiation obey the inverse square law.

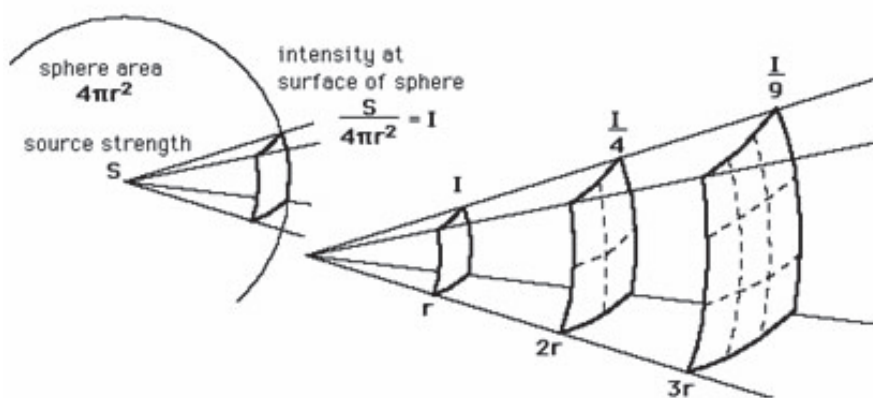


Fig. 23. Inverse Square Law

A plot of this intensity drop shows that it drops off rapidly figure (24). A plot of the drop of sound intensity according to the inverse square law emphasizes the rapid loss associated with the inverse square law. In an auditorium, such a rapid loss is unacceptable. The reverberation in a good auditorium mitigates it. This plot shows the points connected by straight lines but the actual drop is a smooth curve between the points.

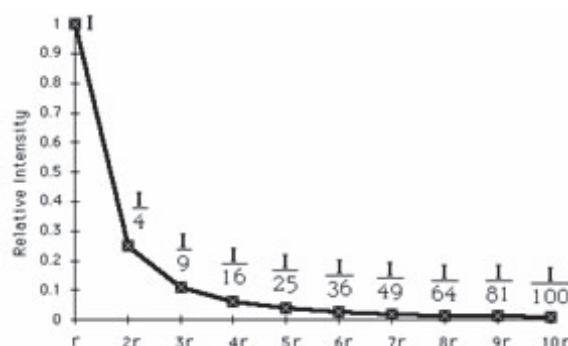


Fig. 24. Inverse Square Law Plot

Reverberation is the collection of reflected sounds from the surfaces in an enclosure like an auditorium. It is a desirable property of auditoriums to the extent that it helps to overcome the inverse square law drop-off of sound intensity in the enclosure. However, if it is excessive, it makes the sounds run together with loss of articulation - the sound becomes muddy, garbled figure (25).

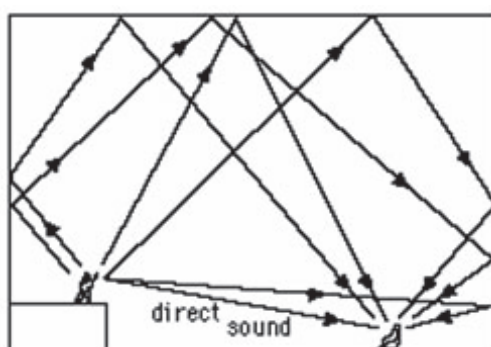


Fig. 25. Reverberant sound is the collection of all the reflected sounds in an auditorium.

7.2 ANC

In order to cancel unwanted noise, it is necessary to obtain an accurate estimate of the noise to be cancelled. In an open environment, where the noise source can be approximated as a point source, microphones can be spaced far apart as necessary and each will still receive a substantially similar estimate of the background noise. However in a confined environment containing reverberation noise caused by multiple sound reflections, the sound field is very complex and each point in the environment has a very different background noise signal. The further apart the microphones are, the more dissimilar the sound field. As a result, it is difficult to obtain an accurate estimate of the noise to be cancelled in a confined environment by using widely spaced microphones.

The proposed model is embodied in a dual microphone noise suppression system in which the echo between the two microphones is substantially cancelled or suppressed.

Reverberations from one microphone to the other are cancelled by the use of first and second line echo cancellers. Each line echo canceller models the delay and transmission characteristics of the acoustic path between the first and second microphones Figure (26).

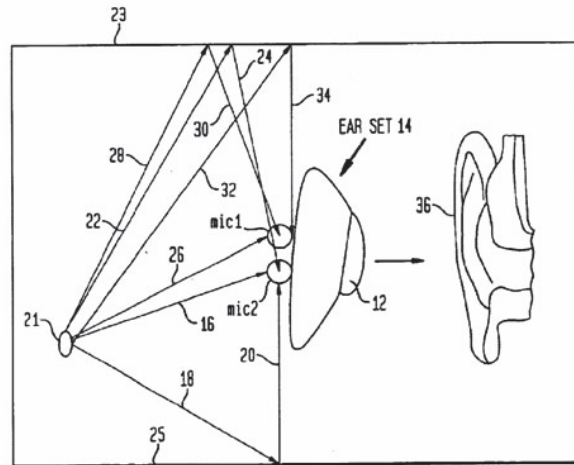


Fig. 26. A pictorial representation of the sound field reaching an ear set in accordance with the proposed model intended to be worn in the human ear.

If the two microphones are moved closer together, the second microphone should provide a better estimate of the noise to be cancelled in the first microphone. However, if the two microphones are placed very close together, each microphone will cause an additional echo to strike the other microphone. That is, the first microphone will act like a speaker (a sound source) transmitting an echo of the sound field striking the second microphone. Similarly, the second microphone will act like a speaker (a sound source) transmitting an echo of the sound field striking the first microphone. Therefore, the signal from the first microphone contains the sum of the background noise plus a reflection of the background noise, which results in a poorer estimate of the background noise to be cancelled figures (27) and (28).

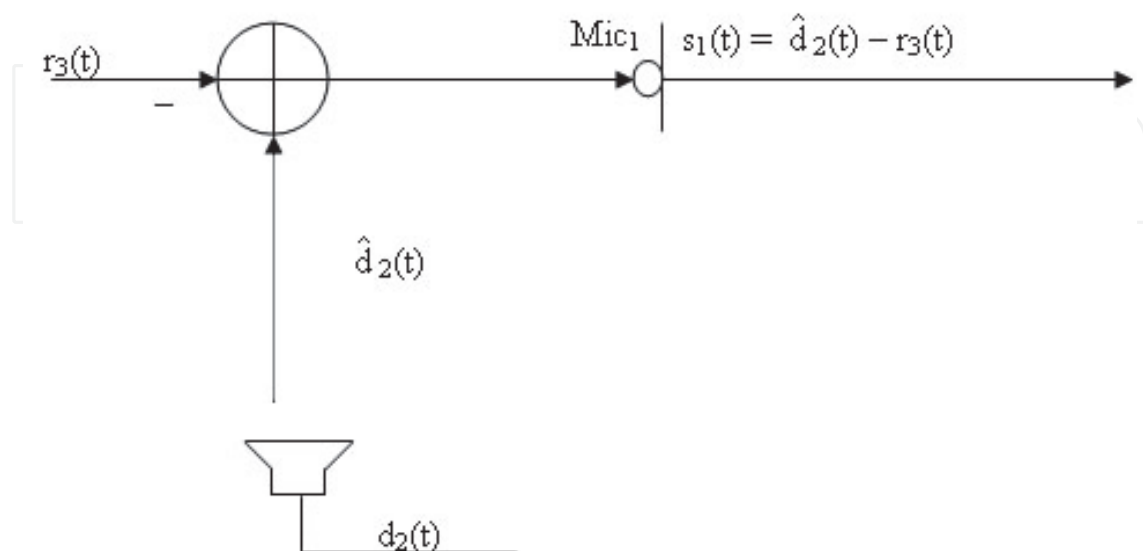


Fig. 27. Simplified model for one direction.

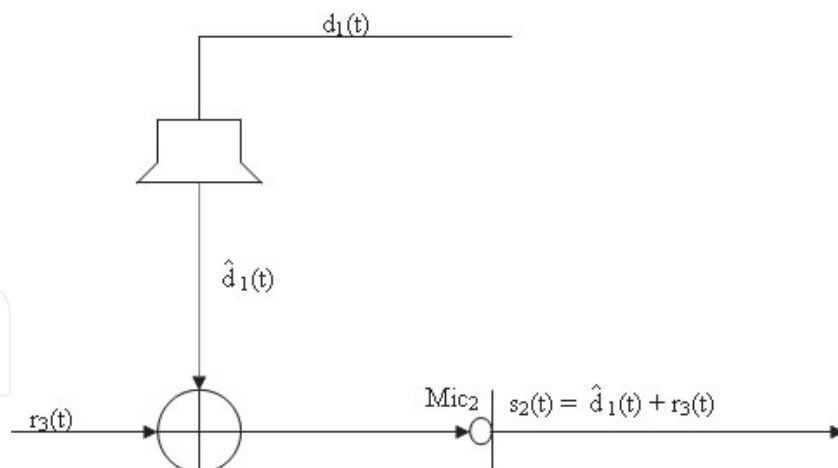


Fig. 28. Simplified model for the opposite direction.

In a first embodiment, a noise suppression system in accordance with the proposed model acts as an ear protector, cancelling substantially all or most of the noise striking the dual microphones of the ear set figure (29). In a second embodiment, a noise suppression system in accordance with the present invention acts a noise suppression communication system, suppressing background noise while allowing communication signals to be heard by the wearer figures (30 and 31).

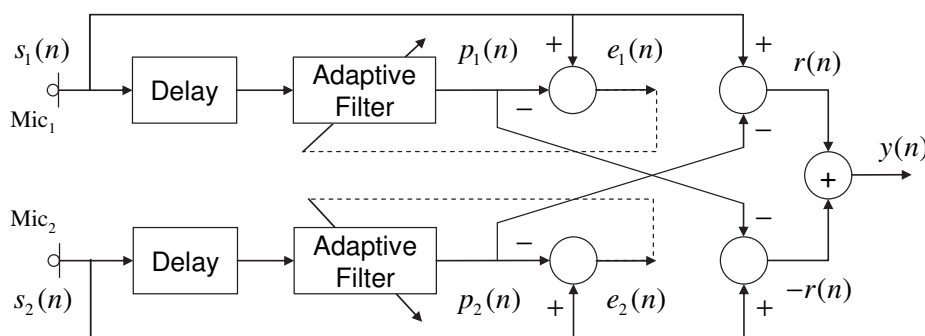


Fig. 29. A noise suppression System in accordance with a first embodiment of the proposed model.

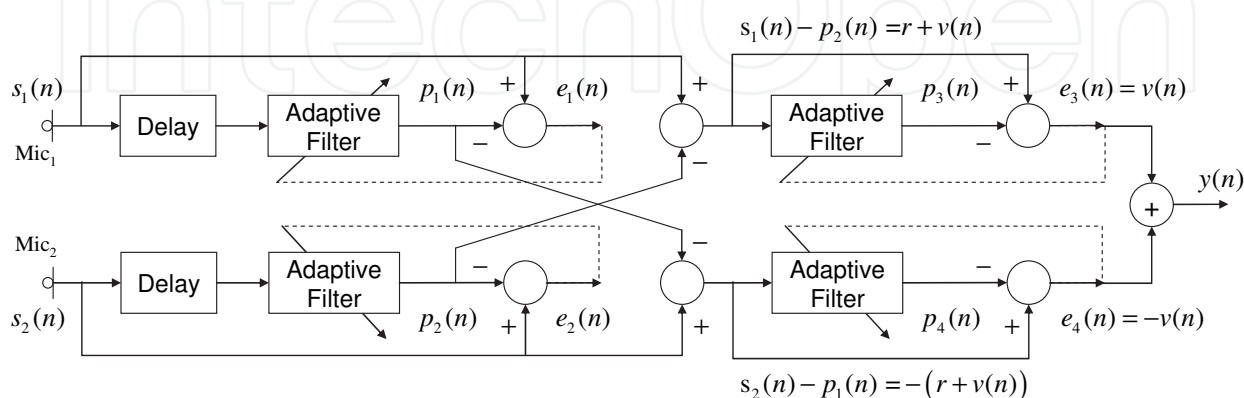


Fig. 30. A noise suppression communications system in accordance with a second embodiment of the proposed model.

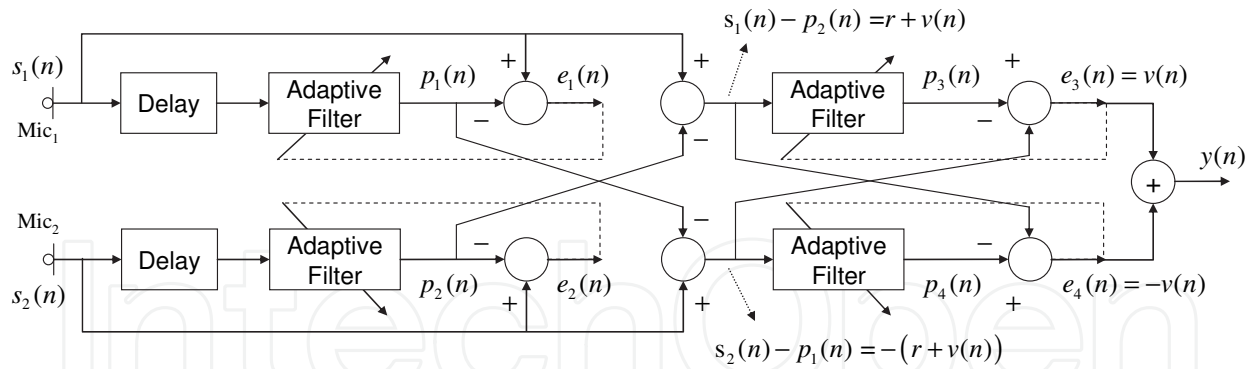


Fig. 31. An alternate scheme for a noise suppression communications system in accordance with a second embodiment of the proposed model.

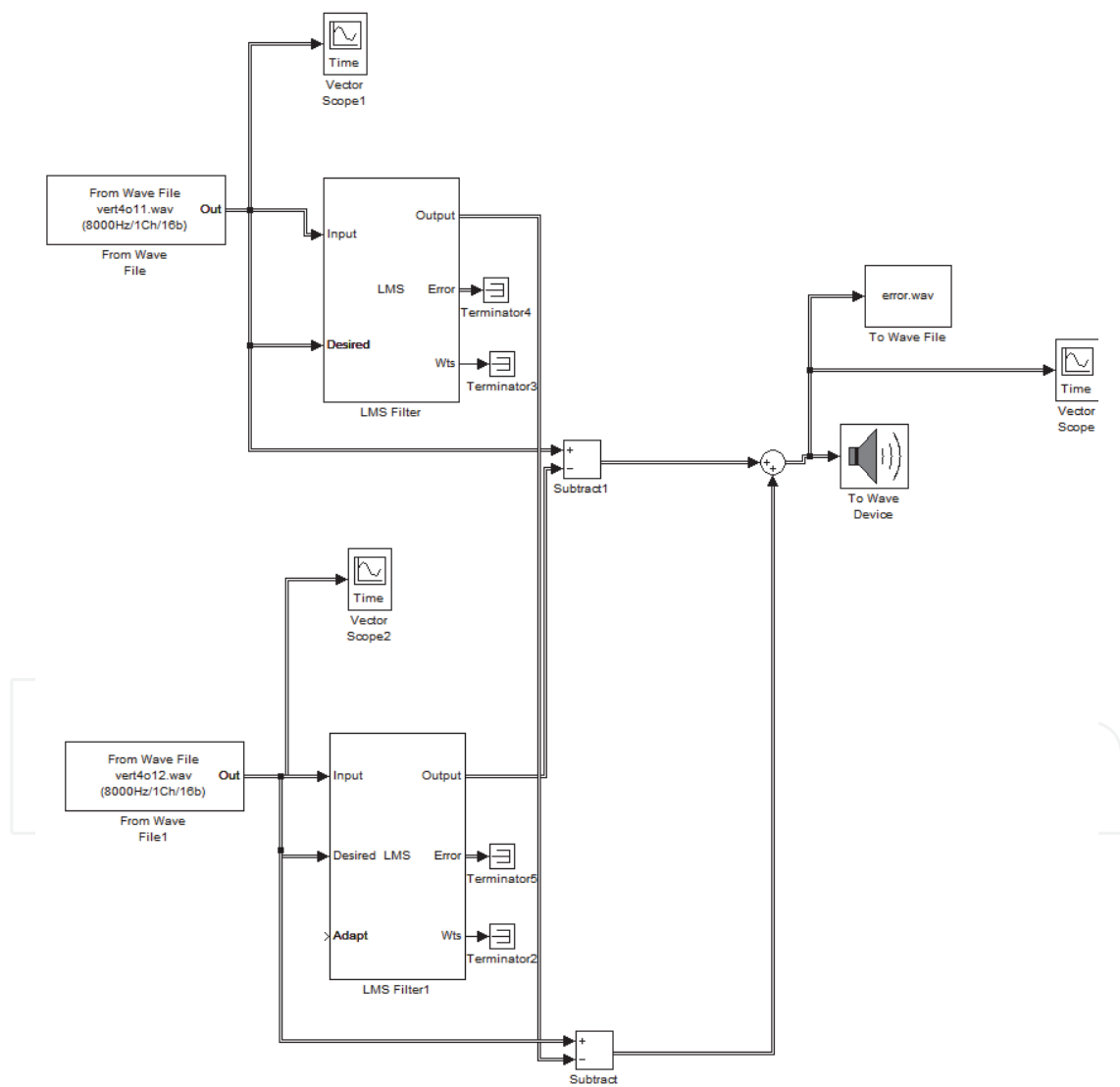


Fig. 32. Simulink block diagram of the proposed noise suppressor in figure 29.

The conceptual key to this proposed model is that the signals received at two closely spaced microphones in a multi-path acoustic environment are each made up of a sum of echoes of

the signal received at the other one. This leads to the conclusion that the difference between the two microphone signals is a sum of echoes of the acoustic source in the environment. In the absence of a speech source, the ANS scheme proposed by Jaber first attempts to isolate the difference signal at each of the microphones by subtracting from it an adaptively predicted version of the other microphone signal. It then attempts to adaptively cancel the two difference signals. When speech is present (as detected by some type of VAD-based strategy), the adaptive cancellation stage has its adaptivity turned off (i.e. the impulse responses of the two FIR filters, one for each microphone, are unchanged for the duration of the speech). The effect here is that the adaptive canceller does not end up cancelling the speech signal contained in the difference between the two microphone signals. The Simulink implementation of the noise suppression system illustrated in figure 29 is displayed in figure 32 where figures 33 and 34 display the noise captured by Mic.1 and Mic. 2 respectively, meanwhile figure 35 shows the filter output.

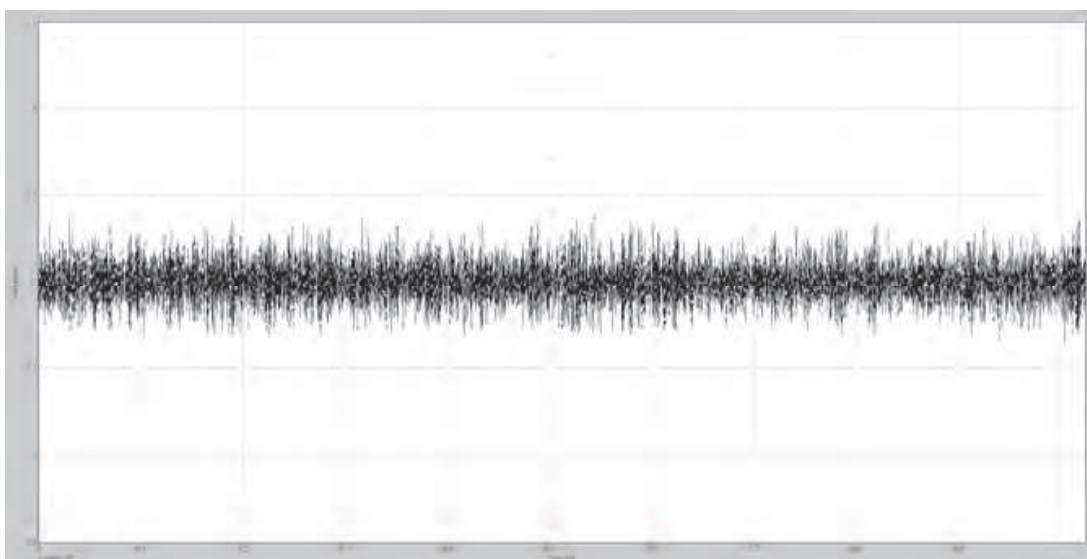


Fig. 33. Noise Captured by Mic. 1.



Fig. 34. Noise Captured By Mic. 2.

The Simulink implementation of the Active Noise Suppressor illustrated in Figure 30 with no Voice Activity Detection (VAD) attached is sketched in 36 and the taking off plane's noise captured by Mic.1 and Mic. 2 are presented in figures 37 and 38 respectively meanwhile figure 39 shows the system output.

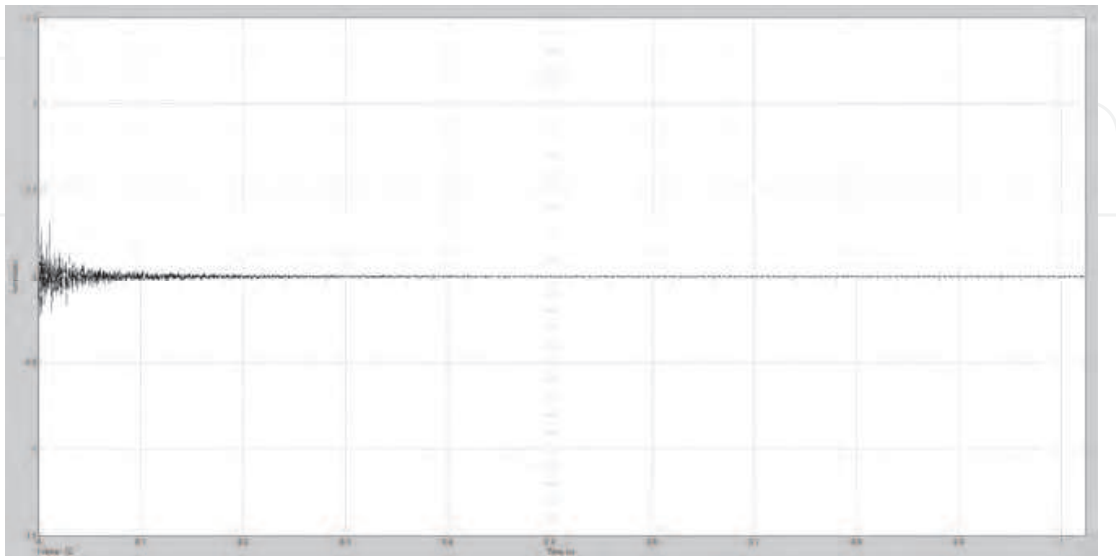


Fig. 35. The Filtered Noise output.

With an accurate front-point and end-point detection VAD algorithm [10] implemented in our proposed models illustrated in figures 30-31, an active noise suppressor is obtained.

The simulation results have been obtained from 25 seconds of noisy voice conditions as shown in Fig. 40 at Mic. 1 and the clean speech of our output filter is shown in figure 41 meanwhile figure 42 will illustrate the clean speech on a reduced scale [11].

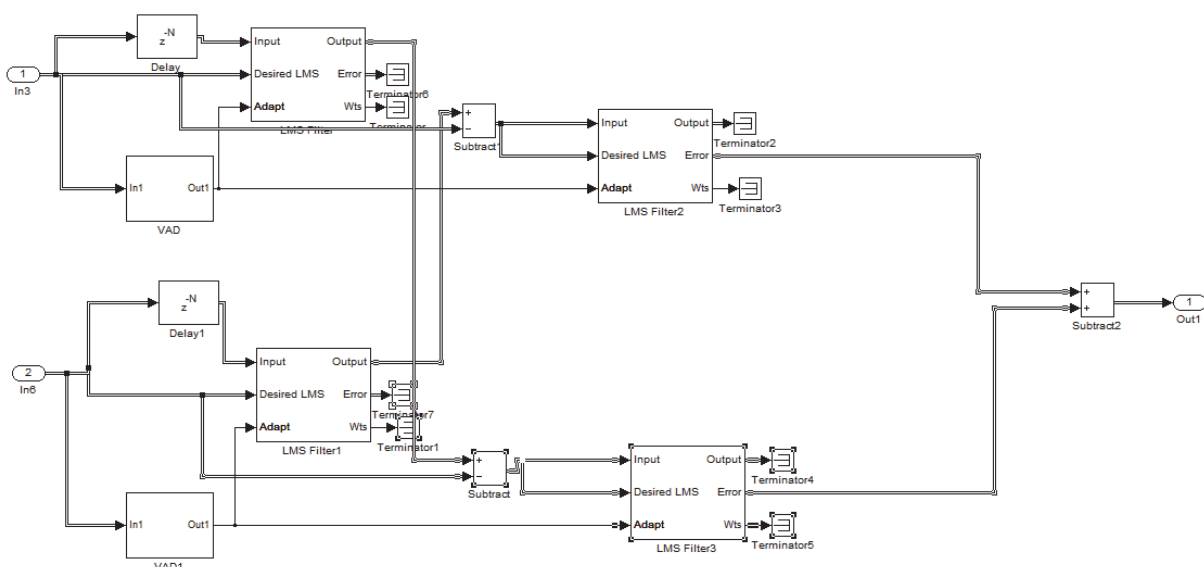


Fig. 36. Simulink block diagram of the proposed noise suppressor in figure 30

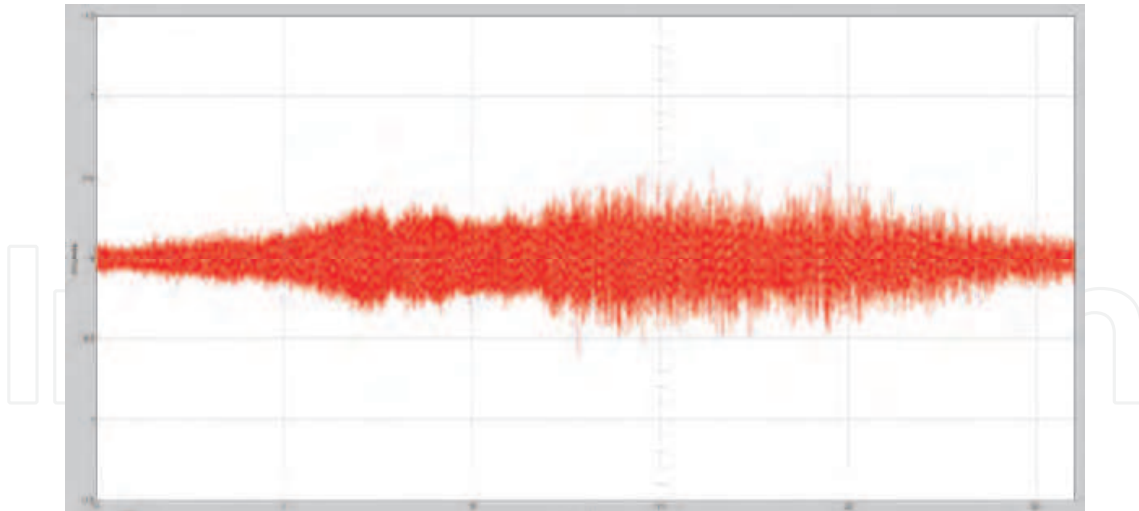


Fig. 37. Taking off Airplane's Noise Captured by Mic. 1.

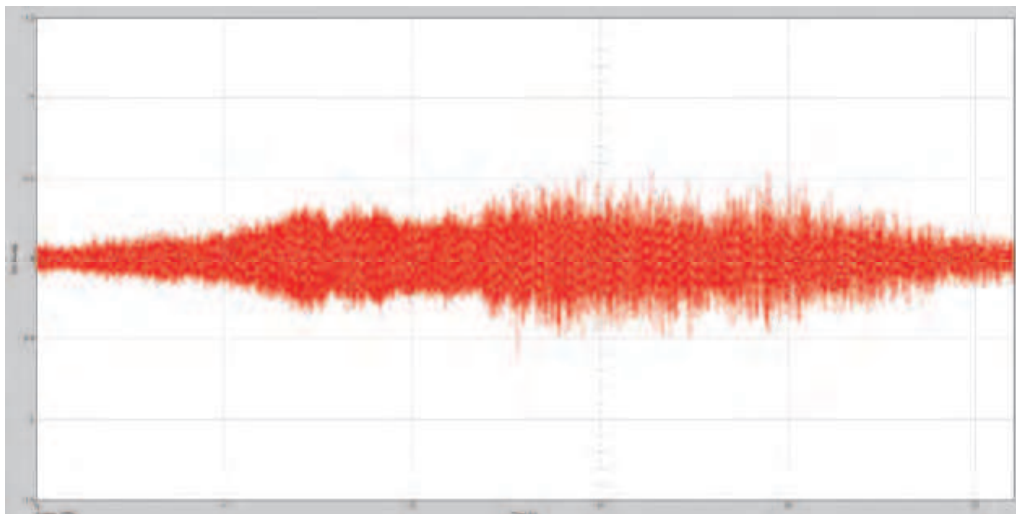


Fig. 38. Taking off Airplane's Noise Captured by Mic. 2.

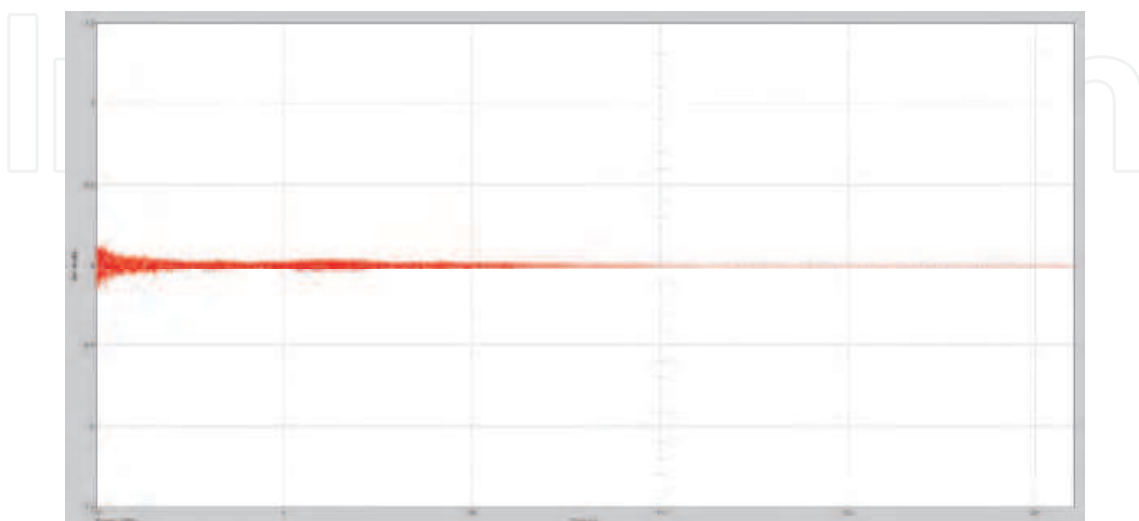


Fig. 39. The system's output

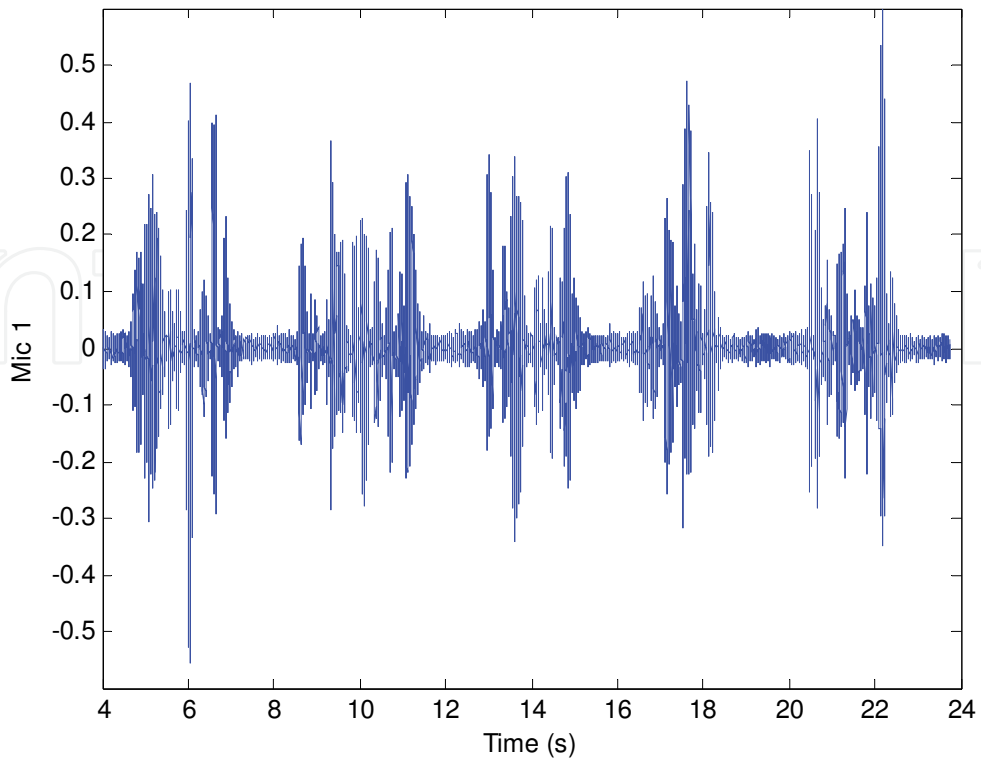


Fig. 40. Noisy speech captured by Mic 1.

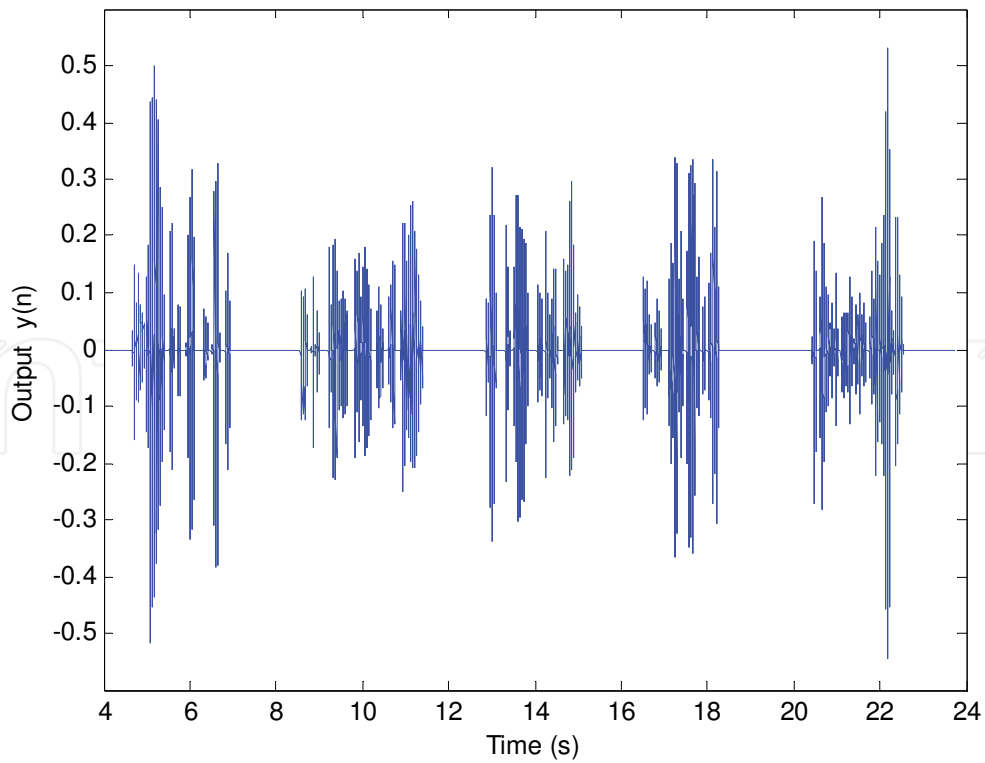


Fig. 41. The clean speech obtained at the output of our proposed ANC (Fig. 30) by using a VAD.

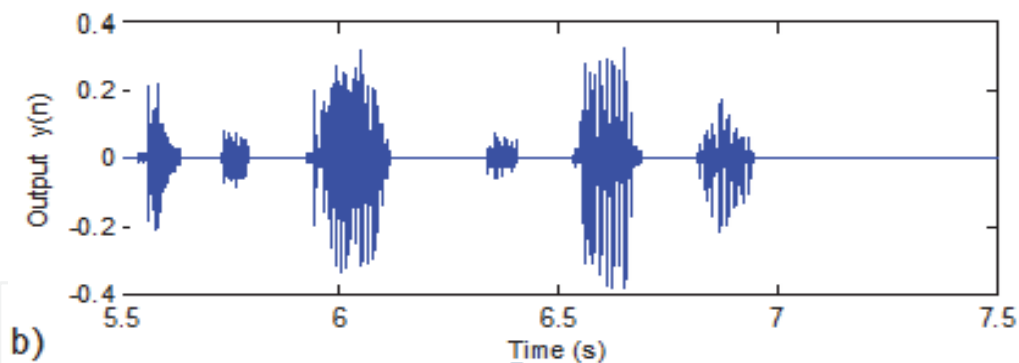


Fig. 42. The clean speech obtained at the output of our proposed ANC (Fig. 30) by reducing the time scale

8. The ultra high speed LMS algorithm implemented on parallel architecture

There are many problems that require enormous computational capacity to solve, and therefore the success of computational science to accurately describe and model the real world has helped to fuel the ever increasing demand for cheap computing power. Scientists are eager to find ways to test the limits of theories, using high performance computing to allow them to simulate more realistic systems in greater detail. Parallel computing offers a way to address these problems in a cost effective manner. **Parallel Computing** deals with the development of programs where multiple concurrent processes cooperate in the fulfilment of a common task. Finally, in this section we will develop the theory of the parallel computation of the widely used algorithms named the least-mean-square (LMS) algorithm¹ by its originators, Widrow and Hoff (1960) [2].

8.1 The spatial radix-*r* factorization

This section will be devoted in proving that discrete signals could be decomposed into *r* partial signals and whose statistical properties remain invariant therefore, given a discrete signal x_n of size *N*

$$x_n = [x_0 \ x_1 \ x_2 \ \dots \ x_{(N-1)}] \tag{93}$$

and the identity matrix I_r of size *r*

$$I_r = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} = [I_{(l,c)}] = \begin{cases} 1 & \text{for } l=c \\ 0 & \text{elsewhere} \end{cases} \tag{94}$$

for $l = c = 0, 1, \dots, r - 1$.

Based on what was proposed in [2]-[9]; we can conclude that for any given discrete signal $x_{(n)}$ we can write:

¹ M Jaber "Method and apparatus for enhancing processing speed for performing a least mean square operation by parallel processing" US patent No. 7,533,140, 2009

$$x_{(l,n)} = \left[I_{(l,c)} \right] \times \left[x_{(m)} \ x_{(m+1)} \ \dots \ x_{(m+p)} \right] \quad (95)$$

is the product of the identity matrix of size r by r sets of vectors of size N/r ($n = 0, 1, \dots, N/r - 1$) where the l^{th} element of the n^{th} product is stored into the memory address location given by

$$l = (rn + p) \quad (96)$$

for $p = 0, 1, \dots, r - 1$.

The mean (or Expected value) of x_n is given as:

$$E[x] = \frac{\sum_{n=0}^{N-1} x_n}{N} \quad (97)$$

which could be factorizes as:

$$\begin{aligned} E[x] &= \frac{\sum_{n=0}^{(N/r)-1} x_{(m)} + \dots + \sum_{n=0}^{(N/r)-1} x_{(m+(r-1))}}{N} = \mu_{(x)} \\ &= \frac{\frac{1}{r} \times \sum_{n=0}^{(N/r)-1} x_{(m+p)}}{\frac{N}{r}} = \frac{\mu_{(x(m+p))}}{r} \end{aligned} \quad (98)$$

therefore, the mean of the signal x_n is equal to sum of the means of its r partial signals divided by r for $p = 0, 1, \dots, r - 1$.

Similarly to the mean, the variance of the signal x_n equal to sum of the variances of its r partial signals according to:

$$\begin{aligned} \text{Var}(x) &= \sigma_x^2 = E[(x - \mu)^2] = \sum_{n=0}^{N-1} (x_{(n)} - \mu)^2 \\ &= \sum_{n=0}^{(N/r)-1} (x_{(m)} - \mu_{(m)})^2 + \dots + \sum_{n=0}^{(N/r)-1} (x_{(m+(r-1))} - \mu_{(m+(r-1))})^2 \\ &= \sum_{p=0}^{r-1} \sigma_{x(m+p)}^2 \end{aligned} \quad (99)$$

8.2 The parallel implementation of the least squares method

The method of least squares assumes that the best-fit curve of a given type is the curve that has the minimal sum of the deviations squared (*least square error*) from a given set of data. Suppose that the N data points are $(x_0, y_0), (x_1, y_1) \dots (x_{(n-1)}, y_{(n-1)})$, where x is the independent variable and y is the dependent variable. The fitting curve d has the deviation (error) σ from each data point, i.e., $\sigma_0 = d_0 - y_0, \sigma_1 = d_1 - y_1 \dots \sigma_{(n-1)} = d_{(n-1)} - y_{(n-1)}$ which could be re-ordered as:

$$\begin{aligned} \sigma_0 &= d_0 - y_0, \sigma_r = d_r - y_r, \sigma_{2r} = d_{2r} - y_{2r}, \dots, \\ \sigma_m &= d_m - y_m, \sigma_1 = d_1 - y_1, \dots, \sigma_{(m+1)} = d_{(m+1)} - y_{(m+1)}, \dots, \sigma_{(m+(r-1))} = d_{(m+(r-1))} - y_{(m+(r-1))} \end{aligned} \quad (100)$$

for $n = 0, 1, \dots, (N/r) - 1$.

According to the method of least squares, the best fitting curve has the property that:

$$\begin{aligned}
 J &= \sigma_0^2 + \dots + \sigma_m^2 + \sigma_1^2 + \dots + \sigma_{(m+1)}^2 + \dots + \sigma_{(m+(r-1))}^2 \\
 &= \sum_{j_0=0}^{r-1} \sum_{n=0}^{(N/r)-1} [d_{(m+j_0)} - y_{(m+j_0)}]^2 \\
 &= \sum_{j_0=0}^{r-1} \sum_{n=0}^{(N/r)-1} \sigma_{(m+j_0)}^2 = \text{a minimum}
 \end{aligned}
 \tag{101}$$

The parallel implementation of the least squares for the linear case could be expressed as:

$$\begin{aligned}
 e_{(n)} &= d_{(n)} - (b + wx_{(n)}) = d_{(n)} - y_{(n)} \\
 &= (I_r \times [d_{(m+j_0)} - y_{(m+j_0)}]) \\
 &= (I_r \times [e_{(m+j_0)}])
 \end{aligned}
 \tag{102}$$

for $j_0 = 1, \dots, r - 1$ and in order to pick the line which best fits the data, we need a criterion to determine which linear estimator is the “best”. The sum of square errors (also called the mean square error (MSE)) is a widely utilized performance criterion.

$$\begin{aligned}
 J &= \frac{1}{2N} \sum_{n=0}^{N-1} e_{(n)}^2 \\
 &= \left(\frac{1}{r}\right) \sum_{j_0=0}^{r-1} \left(\frac{1}{2(N/r)} \sum_{n=0}^{(N/r)-1} \left((I_r \times [e_{(m+j_0)}])\right)^2\right)
 \end{aligned}
 \tag{103}$$

which yields after simplification to:

$$\begin{aligned}
 J &= \left(\frac{1}{r}\right) \sum_{j_0=0}^{r-1} \left(I_r \times \left[\left(\frac{1}{2(N/r)} \right) \sum_{n=0}^{(N/r)-1} e_{(m+j_0)}^2 \right] \right) \\
 &= \left(\frac{1}{r}\right) \sum_{j_0=0}^{r-1} (I_r \times [J_{j_0}]) = \left(\frac{1}{r}\right) \sum_{j_0=0}^{r-1} J_{j_0}
 \end{aligned}
 \tag{104}$$

where J_{j_0} is the partial MSE applied on the subdivided data.

Our goal is to minimize J analytically, which according to Gauss can be done by taking its partial derivative with respect to the unknowns and equating the resulting equations to zero:

$$\begin{cases} \frac{\partial J}{\partial b} = 0 \\ \frac{\partial J}{\partial w} = 0 \end{cases}
 \tag{105}$$

which yields:

$$\begin{cases} \frac{\partial J}{\partial b} = \frac{\partial \left(\left(\frac{1}{r} \right) \sum_{j_0=0}^{r-1} (I_r \times [J_{j_0}]) \right)}{\partial b} = \left(\frac{1}{r} \right) \sum_{j_0=0}^{r-1} \frac{\partial (J_{j_0})}{\partial b} = 0 \\ \frac{\partial J}{\partial w} = \frac{\partial \left(\left(\frac{1}{r} \right) \sum_{j_0=0}^{r-1} (I_r \times [J_{j_0}]) \right)}{\partial w} = \left(\frac{1}{r} \right) \sum_{j_0=0}^{r-1} \frac{\partial (J_{j_0})}{\partial w} = 0 \end{cases} \quad (106)$$

With the same reasoning as above the MSE could be obtained for multiple variables by:

$$\begin{aligned} J &= \frac{1}{2N} \sum_n \left(d_{(n)} - \sum_{k=0}^p w_{(k)} x_{(n,k)} \right)^2 \\ &= \frac{1}{r} \left(\sum_{j_0=0}^{r-1} \frac{1}{2 \left(\frac{N}{r} \right)} \sum_{n=0}^{\left(\frac{N}{r} \right) - 1} \left[d_{(m+j_0)} - \sum_{k=0}^p w_{((m+j_0),k)} x_{((m+j_0),k)} \right]^2 \right) \\ &= \frac{1}{r} \sum_{j_0=0}^{r-1} J_{j_0} \end{aligned} \quad (107)$$

for $j_0 = 1, \dots, r-1$ and where J_{j_0} is the partial MSE applied on the subdivided data.

The solution to the extreme (minimum) of this equation can be found in exactly the same way as before, that is, by taking the derivatives of J_{j_0} with respect to the unknowns (w_k), and equating the result to zero.

Instead of solving equations 106 and 107 analytically, a gradient adaptive system can be used which is done by estimating the derivative using the difference operator. This estimation is given by:

$$\nabla_w J = \frac{\delta J}{\delta w} \quad (108)$$

where in this case the bias b is set to zero.

8.3 Search of the performance surface with steepest descent

The method of steepest descent (also known as the gradient method) is the simplest example of a gradient based method for *minimizing a function of several variables* [12]. In this section we will be elaborating the linear case.

Since the performance surface for the linear case implemented in parallel, are r paraboloids each of which has a single minimum, an alternate procedure to find the best value of the coefficient $w_{j_0(k)}$ is to search in parallel the performance surface instead of computing the

best coefficient analytically by Eq. 106. The search for the minimum of a function can be done efficiently using a broad class of methods that *use gradient information*. The gradient has two main advantages for search.

- The gradient can be computed locally.
- The gradient always points in the direction of maximum change.

If the goal is to reach the minimum in each parallel segment, the search must be in the direction opposite to the gradient. So, the overall method of search can be stated in the following way:

Start the search with an *arbitrary initial weight* $w_{j_0(0)}$, where the *iteration* is denoted by the index in parenthesis (Fig 43). Then compute the gradient of the performance surface at $w_{j_0(0)}$, and modify the initial weight proportionally to the negative of the gradient at $w_{j_0(0)}$. This changes the operating point to $w_{j_0(1)}$. Then compute the gradient at the new position $w_{j_0(1)}$, and apply the same procedure again, i.e.

$$w_{j_0(k+1)} = w_{j_0(k)} - \eta \nabla J_{j_0(k)} \quad (109)$$

where η is a small constant and $\nabla_{w_{j_0(k)}}$ denotes the gradient of the performance surface at the k^{th} iteration of j_0 parallel segment. η is used to maintain stability in the search by ensuring that the operating point does not move too far along the performance surface. This search procedure is called the steepest descent method (fig 43)

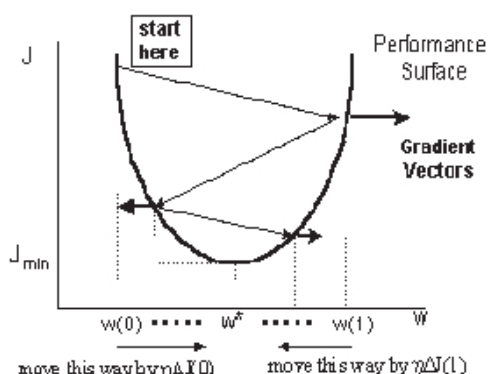


Fig. 43. The search using the gradient information [13].

If one traces the path of the weights from iteration to iteration, intuitively we see that if the constant η is small, eventually the best value for the coefficient w^* will be found. Whenever $w > w^*$, we decrease w , and whenever $w < w^*$, we increase w .

8.4 The radix- r parallel LMS algorithm

Based on what was proposed in [2] by using the *instantaneous value of the gradient as the estimator for the true quantity* which means by dropping the summation in equation 104 and then taking the derivative with respect to w yields:

$$\begin{aligned} \nabla J_{(k)} &= \nabla \left(\frac{1}{r} \times \left(I_r \times \left[J_{j_0(k)} \right] \right) \right) \\ &= \frac{\partial}{\partial w} J_{(k)} = \frac{\partial}{\partial w} \left(\frac{1}{r} \times \left(I_r \times \left[J_{j_0(k)} \right] \right) \right) \\ &\approx \frac{\partial}{\partial w} \frac{1}{2Nr} \sum e^2 = \frac{\partial}{\partial w} \frac{1}{2Nr} \sum_{j_0=0}^{r-1} \sum_{n=0}^{\left(\frac{N}{r}\right)-1} \left(I_r \times \left[e_{(m+j_0)} \right] \right)^2 \\ &\approx \frac{1}{r} \frac{\partial}{\partial w} \left(I_r \times \left[e_{(m+j_0)(k)} x_{(m+j_0)(k)} \right] \right) \end{aligned} \quad (110)$$

What this equation tells us is that an *instantaneous estimate of the gradient is simply the product of the input to the weight times the error at iteration k* . This means that with one multiplication per weight the gradient can be estimated. This is the gradient estimate that leads to the famous *Least Means Square (LMS) algorithm* (Fig 44).

If the estimator of Eq.110 is substituted in Eq.109, the steepest descent equation becomes

$$I_r \times \left[w_{j_0(k+1)} \right] = I_r \times \left(w_{j_0(k)} + \left(\frac{1}{r} \right) \left(\eta_{j_0} e_{(m+j_0)(k)} x_{(m+j_0)(k)} \right) \right) \quad (111)$$

for $j_0 = 0, 1, \dots, r-1$.

This equation is the r Parallel *LMS algorithm*, which is used as predictive filter, is illustrated in Figure 45. The small constant η is called the *step size or the learning rate*.

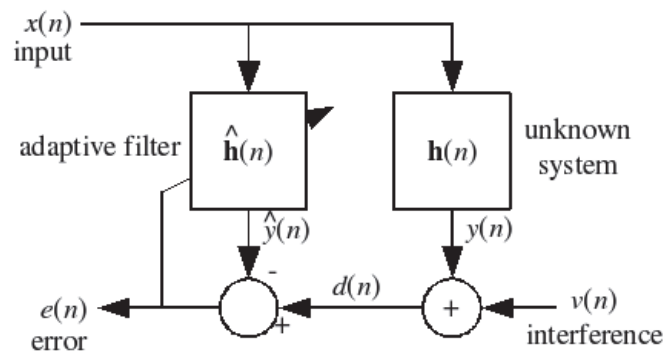


Fig. 44. LMS Filter

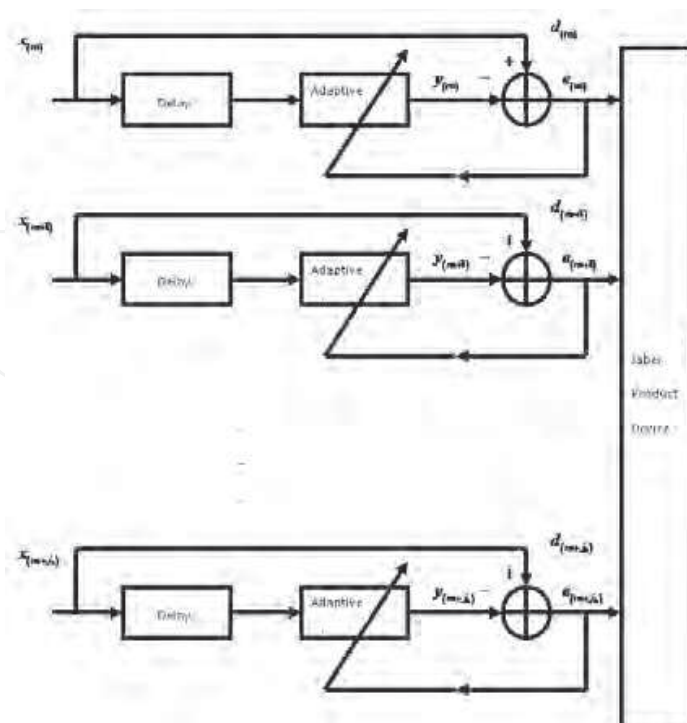


Fig. 45. r Parallel LMS Algorithm Used in Predictive Filter.

8.5 Simulation results

The notion of a mathematical model is fundamental to sciences and engineering. In class of applications dealing with identification, an adaptive filter is used to provide a linear model that represents the best fit (in some sense) to an unknown signal. The LMS Algorithm which is widely used is an extremely simple and elegant algorithm that is able to minimize the external cost function by using local information available to the system parameters. Due to its computational burden and in order to speed up the process, this paper has presented an efficient way to compute the LMS algorithm in parallel where it follows from the simulation results that the stability of our models relies on the stability of our r parallel adaptive filters. It follows from figures 47 and 48 that the stability of r parallel LMS filters (in this case $r = 2$) has been achieved and the convergence performance of the overall model is illustrated in figure 49. The complexity of the proposed method will be reduced by a factor of r in comparison to the direct method illustrated in figure 46. Furthermore, the simulation result of the channel equalization is illustrated in figure 50 in which the blue curves represents our parallel implementation (2 LMS implemented in parallel) compared to the conventional method where the curve is in a red colour.

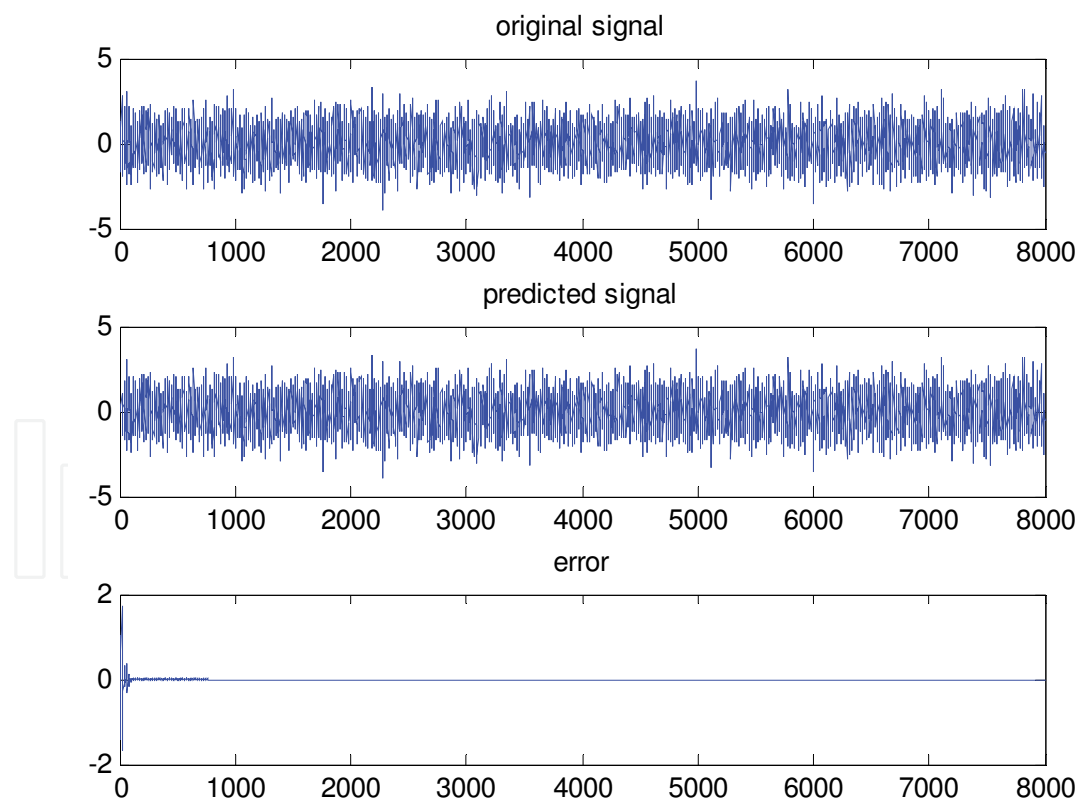


Fig. 46. Simulation Result of the original signal

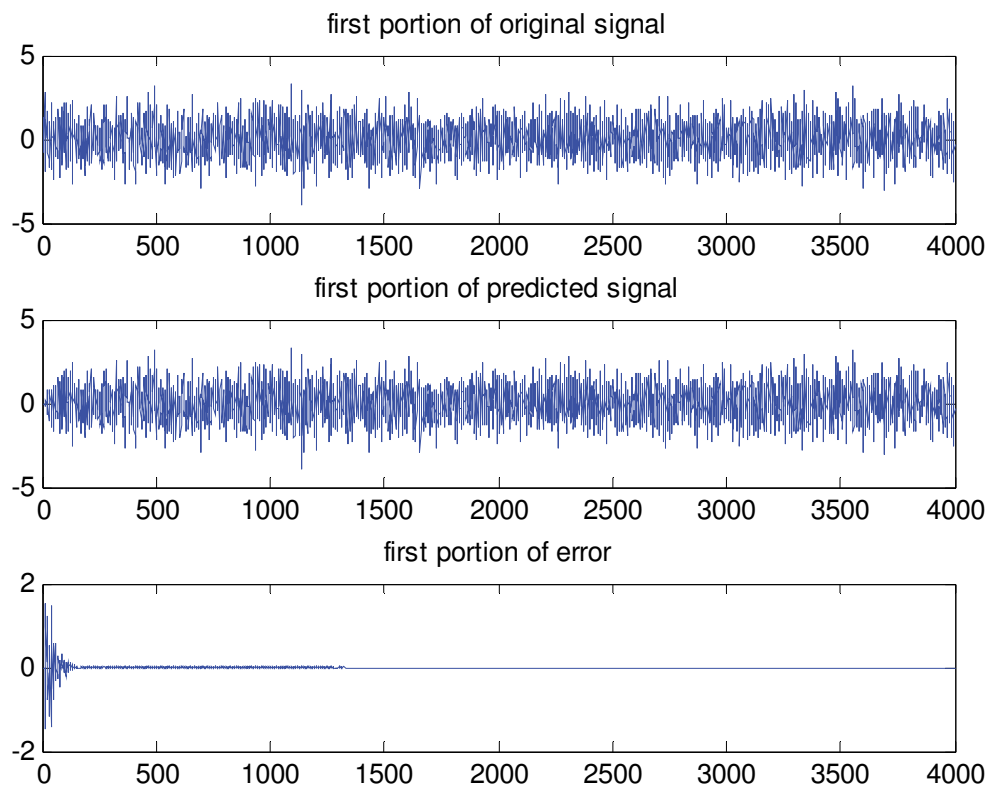


Fig. 47. Simulation Result of the first partial LMS Algorithm.

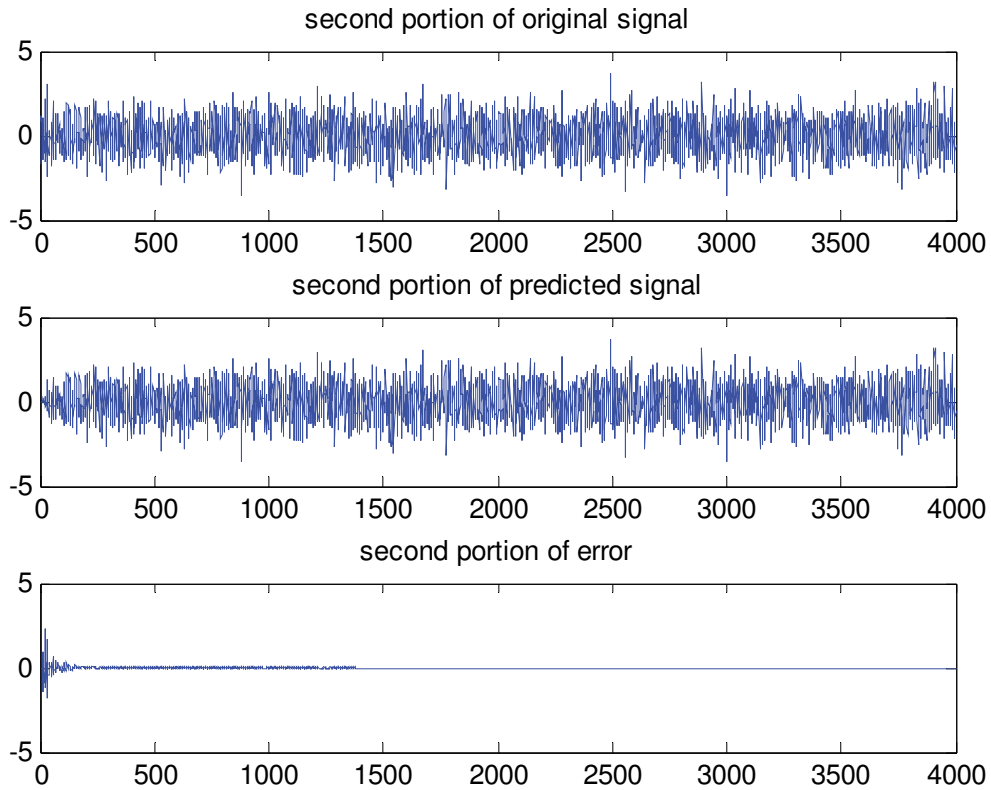


Fig. 48. Simulation Result of the second partial LMS Algorithm.

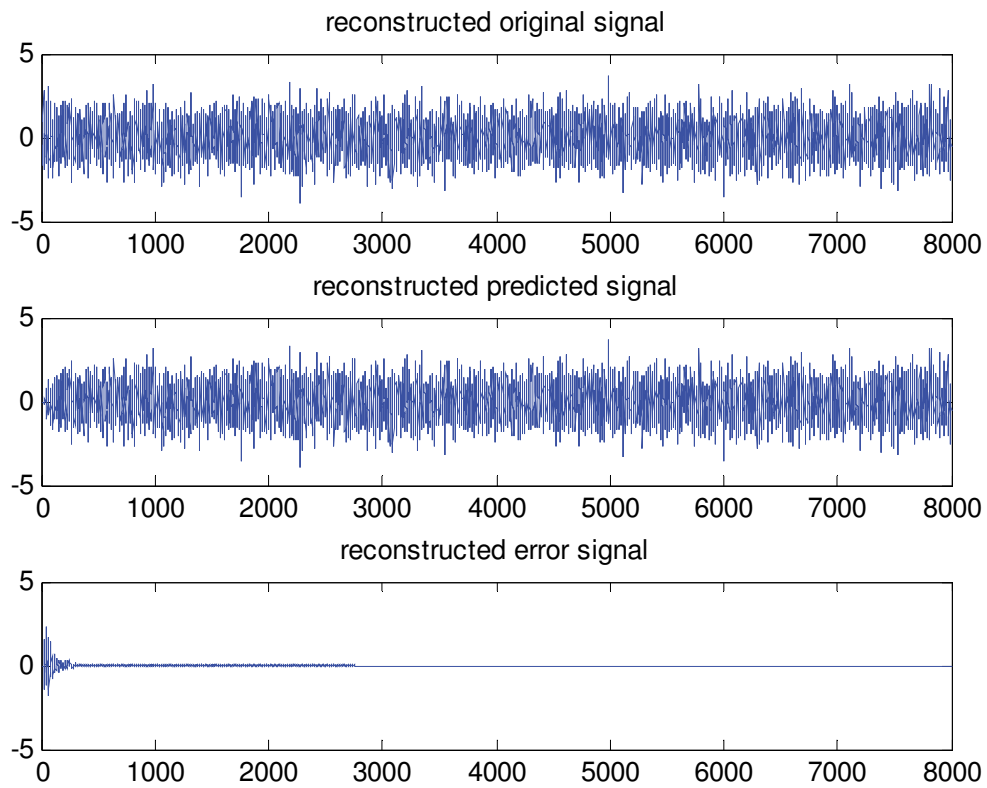


Fig. 49. Simulation Result of the Overall System.

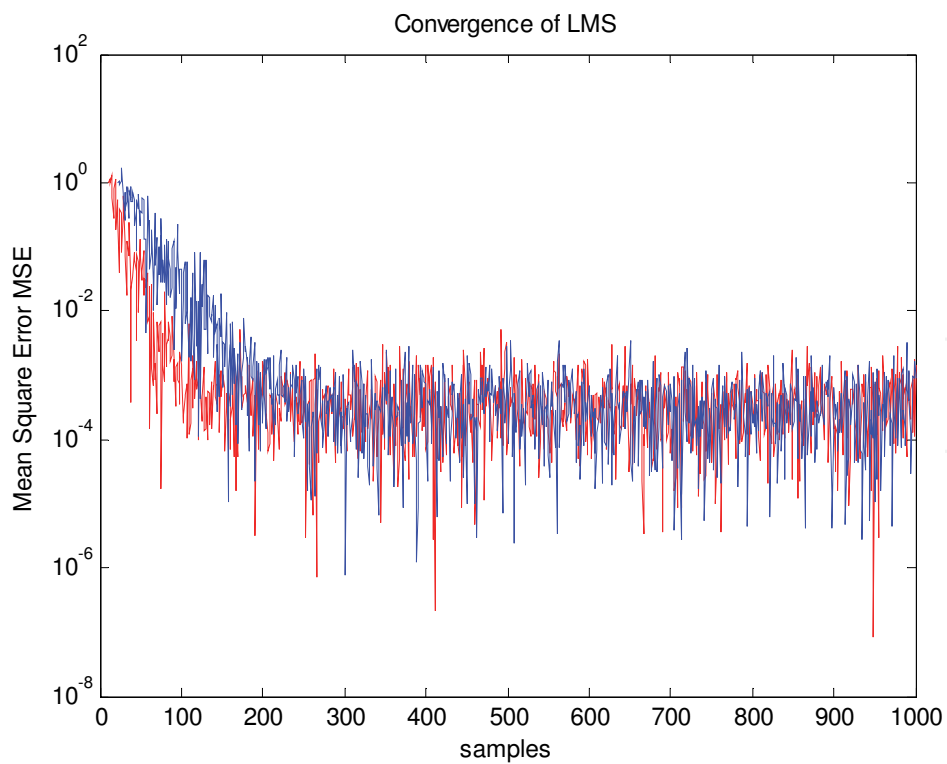


Fig. 50. Simulation Result of the channel equalization (blue curve two LMS implemented in Parallel red curve one LMS).

8. References

- [1] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [2] Widrow and Stearns, " Adaptive Signal Processing ", Prentice Hall 195.
- [3] K. Mayyas, and T. Aboulnasr, "A Robust Variable Step Size LMS-Type Algorithm: Analysis and Simulations", IEEE 5-1995, pp. 1408-1411.
- [4] T. Aboulnasr, and K. Mayyas, "Selective Coefficient Update of Gradient-Based Adaptive Algorithms", IEEE 1997, pp. 1929-1932.
- [5] E. Bjarnason: "Analysis of the Filtered X LMS Algorithm ", IEEE 4 1993, pp. III-511, III-514.
- [6] E.A. Wan, "Adjoint LMS: An Efficient Alternative To The Filtered X LMS And Multiple Error LMS Algorithm", Oregon Graduate Institute Of Science & Technology, Department Of Electrical Engineering And Applied Physics, P.O. Box 91000, Portland, OR 97291.
- [7] B. Farhang-Boroujeny: "Adaptive Filters, Theory and Applications", Wiley 1999.
- [8] Wiener, Norbert (1949). "Extrapolation, Interpolation, and Smoothing of Stationary Time Series", New York: Wiley. ISBN 0-262-73005-7
- [9] M. Jaber "Noise Suppression System with Dual Microphone Echo Cancellation US patent no.US-6738482.
- [10] M. Jaber, "Voice Activity detection Algorithm for Voiced /Unvoiced Decision and Pitch Estimation in a Noisy Speech feature Extraction", US patent application no. 60/771167, 2007.
- [11] M. Jaber and D. Massicottes: "A Robust Dual Predictive Line Acoustic Noise Canceller", International Conference on Digital Signal Processing DSP 2009 Santorini Greece.
- [12] M. Jaber, D. Massicotte, "A New FFT Concept for Efficient VLSI Implementation: Part I - Butterfly Processing Element", 16th International Conference on Digital Signal Processing (DSP'09), Santorini, Greece, 5-7 July 2009.
- [13] J.C. Principe, W.C. Lefebvre, N.R. Euliano, "Neural Systems: Fundamentals through Simulation", 1996.

IntechOpen



Adaptive Filtering

Edited by Dr Lino Garcia

ISBN 978-953-307-158-9

Hard cover, 398 pages

Publisher InTech

Published online 06, September, 2011

Published in print edition September, 2011

Adaptive filtering is useful in any application where the signals or the modeled system vary over time. The configuration of the system and, in particular, the position where the adaptive processor is placed generate different areas or application fields such as prediction, system identification and modeling, equalization, cancellation of interference, etc., which are very important in many disciplines such as control systems, communications, signal processing, acoustics, voice, sound and image, etc. The book consists of noise and echo cancellation, medical applications, communications systems and others hardly joined by their heterogeneity. Each application is a case study with rigor that shows weakness/strength of the method used, assesses its suitability and suggests new forms and areas of use. The problems are becoming increasingly complex and applications must be adapted to solve them. The adaptive filters have proven to be useful in these environments of multiple input/output, variant-time behaviors, and long and complex transfer functions effectively, but fundamentally they still have to evolve. This book is a demonstration of this and a small illustration of everything that is to come.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Marwan Jaber (2011). The Ultra High Speed LMS Algorithm Implemented on Parallel Architecture Suitable for Multidimensional Adaptive Filtering, Adaptive Filtering, Dr Lino Garcia (Ed.), ISBN: 978-953-307-158-9, InTech, Available from: <http://www.intechopen.com/books/adaptive-filtering/the-ultra-high-speed-lms-algorithm-implemented-on-parallel-architecture-suitable-for-multidimensiona>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen