The Nucleus **51**, No. 3 (2014) 355-359

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Performance Evaluation of Adaptive Filters Using a Novel Technique of Matrix Inversion

U. Khan*, N.A. Khan, Y. Khan, N. Khan and S. Nadeem

Department of Electrical Engineering, Iqra National University, Peshawar, Pakistan

(Received August 05, 2014 and accepted in revised form September 10, 2014)

Adaptive filtering is one of the emerging areas in the field of digital signal processing. Adaptive filter alters its coefficients according to the situations, in order to minimize the cost function. The cost function is the difference between a desired or reference signal and the filter output signal. Once the desired signal is available for the process, the cost function is minimized by the adaptive filter and the tap weights are calculated algorithmically. According to the wiener solution the filter tap weight vector is calculated by multiplying the inverse of auto-correlation matrix R with the cross-correlation vector p. This paper evaluates the performance of adaptive filters by taking the inverse of the correlation matrix R through a new algorithm. The algorithm is tested and implemented in adaptive filtering for the applications of system identification and noise cancellation. Various parameters of adaptive filtering are examined for the processes. All the simulations are done in MATLAB software.

Keywords: Adaptive filters (AF), Mean square error (MSE), Least mean square (LMS), Normalized least mean square (NLMS) and Recursive least square (RLS)

1. Introduction

In the field of Digital Signal Processing, Adaptive filter (AF) is the best choice to deal with the signals for which only statistical information are available. It has the ability to adapt its coefficients according to the requirements. AF has some very important practical applications. They are used in the field of Digital Communication Systems for echo cancellation. They are also used for system identification, signal enhancement, signal prediction and channel equalization [1]. Recently many researchers have worked on implementation of AF for different applications. They have been used for echo cancellation using the LMS (Least Mean Square), NLMS (Normalized Least Mean square) and RLS (Recursive Least Square) algorithms by comparing their individual performance [2]. The performance is also tested upon hardware setup where the echo is cancelled out by AF running through LMS algorithm [3]. Another application of AF is system identification which has been experimented for the LMS algorithm [4]. AF as a system identifier, is also tested for the NLMS algorithm with a variable and robust step size where the variation of the step size is analyzed [5]. All these approaches are based on complex algorithms as they use the conventional method for calculating the inverse of the correlation matrix R for the input signal.

In this paper a novel approach is proposed for implementation of AF which is based on the new technique of calculating the inverse of correlation matrix R. The new technique is proved to be efficient and faster from the old conventional technique used so far [6,7]. The technique is used here for implementing AF for system identification and noise cancellation. Different parameters are analyzed for both the applications.

The rest of the article is organized as follows. In section 2 a general implementation of adaptive filters is presented. Section 3 presents the new technique of matrix inversion. In section 4 the performance results are given in which adaptive filter is implemented, using the new technique, for the applications of system identification and noise cancellation. Finally some conclusions are given in section 5.

2. Adaptive Filters

An adaptive filter is a special type of digital filter which has adjustable coefficients. Adaptive filter has the ability to handle the situation where the signal's fixed specifications are not available or process is time-varying [1]. In order to adapt the coefficients, they usually require some extra information, which is given in the form of a signal called the desired signal. The desired signal depends upon the specific application for which the adaptive filter is used. Their behavior is more complex than that of the fixed filters because they are non-linear devices. A general configuration block diagram of adaptive FIR-filter is shown in figure 1. The following notation should be considered to understand the figure.

$x(k)$ - Input signal to the adaptive filter

$y(k)$ - Filter's output signal

$d(k)$ - The desired or reference signal

* Corresponding author : umair340_khan@yahoo.com

e(k) - The error signal

k - The index for iteration number

w(k) - Coefficients of Adaptive filter

Z^{-1} - Delay operator by one

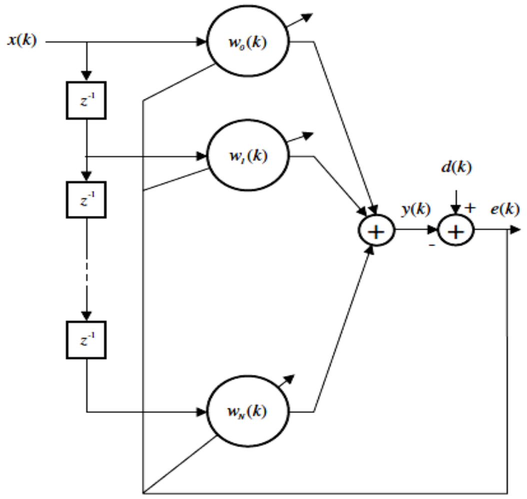


Figure 1. Adaptive FIR-Filter Block Diagram

The error signal *e(k)* is the measure of the performance of the running algorithm for the adaptive filter. It tells how closely the desired signal has been achieved and how efficiently the coefficients are adapted. The minimum is the value of *e(k)*function, the better will be the performance of the filter. In case the adaptive filter output is exactly equal to the desired signal, then the error signal will be zero and this will be minimum error stage for the process. The error signal is given by the subtraction of the desired signal and the filter output signal.

$$e(k) = y(k) - d(k) \tag{1}$$

The Wiener filter is a special type of transversal adaptive FIR filter which is built upon the MSE (Mean Square Error) cost function $\zeta(k)$. The MSE function is set to a minimum optimal value which decides the adaptive filter tap weight values. The MSE function can be expressed in terms of ***p*** and ***R***, where ***p*** is the cross-correlation vector of the input signal and the desired signal, and ***R*** is the auto-correlation matrix of the input signal.

$$\zeta(k) = E[e^2(k)] \tag{2}$$

$$\boldsymbol{p} = E\,[x(k)d(k)] \tag{3}$$

$$\boldsymbol{R} = E\,[x(k)\,x^T(k)] \tag{4}$$

$$\zeta(k) = E[e^2(k)]$$

$$= E\,[d^2(k)] - 2w^Tp + w^TRw \tag{5}$$

The value of the minimum cost function $\zeta(k)$ can be calculated by putting the gradient vector of eq. (5) equal to zero.

$$-2p + 2Rw = 0$$

$$2Rw = 2p$$

$$\mathbf{w = R^{-1}p} \tag{6}$$

Equation 6 is called the Wiener solution. The optimal values in wiener solution are the filter tap weight coefficients for which the cost function is nearly equal to zero.

3. The Matrix Inversion

In order to find out an optimal solution for the vector **w,** the correlation matrix **R** needs to be inverted and multiplied with vector **p**, as per Wiener solution eq. (6). Where, **p** is the cross-correlation vector between the desired *d(k)* and input signals *x(k)*, and **R** is the input signal auto-correlation matrix. The characteristics of the correlation matrix play a key role in the understanding of properties of most adaptive-filtering algorithms. As a consequence, it is important to examine the main properties of the matrix **R**. For a given input vector, the correlation matrix is given by:

$$\mathbf{R} = \begin{bmatrix} E[|x_0(k)|^2] & E[x_0(k)x_1^*(k)] & \cdots & E[x_0(k)x_N^*(k)] \\ E[x_1(k)x_0^*(k)] & E[|x_1(k)|^2] & \cdots & E[x_1(k)x_N^*(k)] \\ \vdots & \vdots & \ddots & \vdots \\ E[x_N(k)x_0^*(k)] & E[x_N(k)x_1^*(k)] & \cdots & E[|x_N(k)|^2] \end{bmatrix}$$
$$= E[\mathbf{x}(k)\mathbf{x}^H(k)]$$

Where $x^H(k)$ is the Hermitian transposition of *x(k)*

Several algorithms are available for finding inverse of a matrix and in turn for solving system of linear equations [8]. Some of them are the iterative methods, Gauss elimination and decomposition algorithms [9]. Some well-known approaches to the matrix inversion are Gauss elimination, LU factorization, and Cholisky factorization [10].A survey of these algorithms guarantees some possible improvements in their accuracy, simplicity, and efficiency. Most of these algorithms target some special types of matrices like positive definite matrix, diagonally dominant and symmetric matrices etc [8-10]. These methods, firstly, require high performance computers as their computational complexity is greater. Secondly, these methods target special type of matrices to perform well with. Though some of these methods may have better performance over some others, their overall computational complexity is higher [10].

A new approach is developed for finding inverse of a matrix [6-7]. The algorithm is very efficient and simple than the currently available approaches. This approach is applicable in general, regardless of the structure of the given matrix. The by-hand calculations are very easy and straightforward. Also the computer implementation is extremely efficient. It stores the original matrix in the memory only and step by step replaces it by the inverse matrix elements. It means that its memory requirements are minimal. The algorithm has the ability to find the inverse and determinant of a matrix in one go which is the most important and unique feature of this algorithm. It calculates the inverse by selecting the pivot through diagonal of the matrix. Following are the algorithm steps and flow chart to find the inverse of any matrix.

Step A: Initially, $p = 0, d = 1$;

Step B: $p = p+1$(increment p by one)

Step C: If $a_{pp}= 0$ then inverse cannot be calculated, go to step J.

Step D: $d' = d*a_{pp}$

Step E: Compute the new pivot row element values by:

$a'_{pj} = a_{pj}/a_{pp}$,

Where $j = 1,...,n, j$ is not equal to p

Step F: Compute the new pivot column element values by:

$a'_{ip} = -a_{ip}/a_{pp}$,

Where $i = 1,...,n, i$ is not equal to p

Step G: Compute the remaining new element values by:

$a'_{ij} = a_{ij} + a_{pj} * a'_{ip}$,

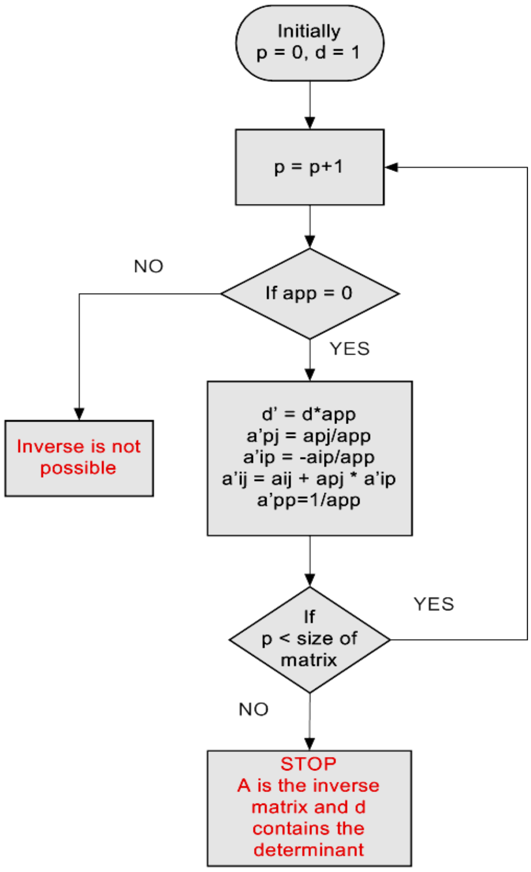
Where i = 1,…,n, j = 1,…n & i, j is not equal to p

Step H: Compute the new value of the current pivot element by:

$a'_{pp}=1/a_{pp}$

Step I: If $p < n$ go to step B, where n the size of the matrix A

Step J: Stop, Matrix A contains the inverse matrix now and d is the determinant of original matrix.



Flow-chart representation of the algorithm.

The new approach is used for calculating the inverse of the correlation matrix **R** in adaptive filters implementation.

4. Simulations and Result Discussion

The experimental setup is tested for implementation of adaptive filters. The applications tested are system identification and noise cancellation. Each of the application was implemented in MATLAB.

4.1. System Identification

An input signal $x(k)$ is filtered with a 3rd order system having coefficients 0.9000, 0.2000, -0.4000 and 0.5000. The output $d(k)$ is then processed by Wiener filter to get the approximation of the coefficients of the unknown filter. Table 1 shows the comparison of the proposed technique with the old Guass Elimination method [8,10,11] for the application of third order system identification. The coefficients of the system to be identified are optimally adapted by the adaptive filter as shown. The values of elapsed time and the MSE vary each time the application is run but they are always lesser for the proposed technique.

Table 1. Filter tap weights, elapsed time and MSE for 3rd order system identification.

Function used	Filter order	Unknown system tap weights	Adaptive Filter tap weights	Time elapsed (s)	MSE
New method	3	0.9000, 0.2000, 0.4000, 0.5000	0.9001, 0.2008, 0.3993, 0.4984	0.00477	0.0029
Old method	3	0.9000, 0.2000, 0.4000, 0.5000	0.9002, 0.1930, 0.3977, 0.4894	0.00481	0.0111

4.2. Noise Cancellation

Figure 2 illustrates an adaptive filter as a noise canceller. An original signal *d[n]* is affected by a noise signal *v[n]* and makes a noisy signal *x[n]* such that *x[n]* = *d[n]* + *v[n]*. The noisy signal is then processed through adaptive filter to cancel out the noise impact from the signal and optimally reconstruct the original signal.

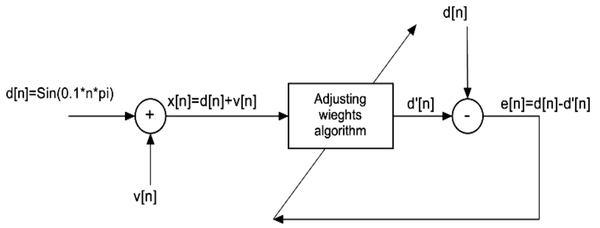


Figure 2. Structure for adaptive filter as a noise canceller.

Figure 3 shows the graphical results for the application run through the new technique. The first index plot in the subplot is the original signal *d[n]* which is a sinusoidal signal having a frequency of 0.1 Hz and a peak to peak amplitude of 2. After processing the signal through the adaptive filter, a new signal *y[n]* is reconstructed which is somehow similar to the original signal *d[n]* as shown in the second index plot. The third index plot shows the error signal.

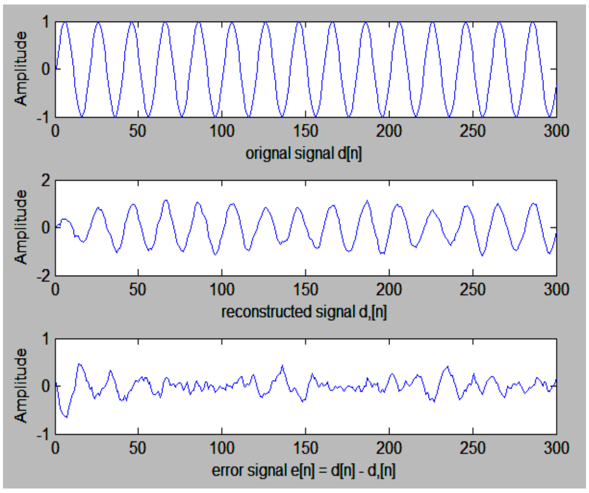


Figure 3. Input and output waveforms for the new technique.

Figure 4 shows the graphical results for the application run through the old method of matrix inversion. The first index plot, the original signal *d[n]*, is a sinusoidal signal with a frequency of 0.1 Hz and peak to peak amplitude of 2. After processing the signal through the adaptive filter, a new signal *y[n]* is reconstructed which is somehow similar to the original signal *d[n]* as shown in the second index plot. The third index plot shows the error signal.

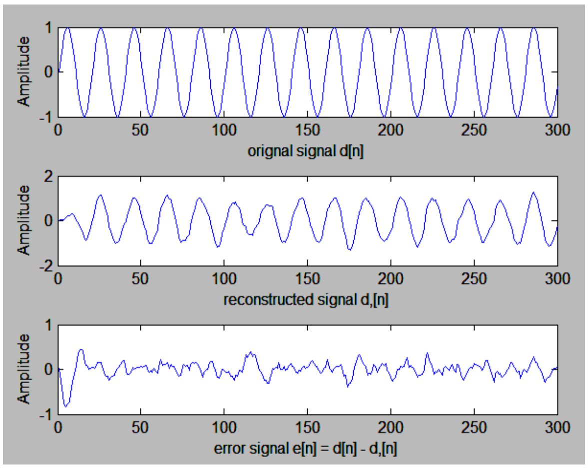


Figure 4. Input and output waveforms for the old method.

5. Conclusions

Adaptive filter minimizes a cost function which is difference between desired signal and filter output. Filter weights are calculated by wiener's solution which multiplies inverse of matrix **R** with matrix **p.** Two different techniques of matrix inversion are tested in adaptive filter for the applications of system identification and noise cancellation. The individual characteristics of both the techniques were observed in terms of time elapsed and MSE. It is seen that performance of the adaptive filter is improved in terms of elapsed time and MSE. Furthermore, the new technique of matrix inversion requires lesser memory as it replaces the new element values in the original matrix. This results in lesser memory requirement for adaptive filter implementation running through this technique. The performance can be further improved by using infinite impulse response filters (IIR) or sub-band adaptive filtering.

References

[1] Diniz and S. R. Paulo, Adaptive Filtering, Algorithms and Practical Implementation. 3rd edition. Springer (2008).

[2] P.Rajesh and A.Sumalatha, Int. J. Engg. Res. Tech. **1** (2012) 1.

[3] R.A. Khalil, Al Rafdain Engg. J. **16** (2008) 20.

[4] I. Dornean, M. Topa, B.S. Kirie and E. Szopos, System Identification with Least Mean Square adaptive algorithm,Proceeding of Interdisciplinary in Engineering Scientific International Conference, TG. Mures– Romania (November 2007) IV, 1, 1-IV, 1,4.

[5] A.P. Patil, P.P.Belagali and S.D.Shinde1, IOSR.J. Electronics & Commun. Engg. ISBN 2278-8735 (2012) 54.

[6] A. Farooq and K. Hamid, Int J. Tech. Diffusion **1** (2010) 20.

[7] A.S. Inayat, A. Farooq and K. Hamid, Int J. Tech. Diffusion **1** (2010) 36.

[8] B. Kolman and D. R. Hill, Elementary Linear Algebra with Applications, 9th Edition, Pearson Education Inc (2008).

[9] H.S. Najafi and M.S. Solary, Appl. Math & Comput. **183** (2006) 539.

[10] R. L. Burden and J. D. Faires, Numerical Analysis, 9th Edition, Brooks/Cole (2011).

[11] MATLAB Function Reference, F–O, Version 7, The Mathworks **2** (2004).