COMPARATIVE STUDY OF ITERATIVE SEARCH METHOD FOR ADAPTIVE FILTERING PROBLEMS

Ahmad[a], N.A

[a] Universiti Sains Malaysia, Malaysia

Abstract. Adaptive filtering problem refers to a class of application in signal processing that deals with adaptation of a system so as to adjust itself with the phenomenon that is taking place in its surrounding. Examples of such problem include adaptive system modeling, noise cancellation, equalization and prediction. The adaptive filtering problem may be mathematically formulated as an adaptive least squares problem in which a set of parameter values are updated so that the time varying sum of squared error cost function is minimized. Numerous algorithms are available for solving adaptive filtering problems and they may be classified into two categories: iterative methods and direct methods. In this paper, we perform a comparative study of four iterative search methods, namely, the method of steepest descent, the Newton's method, the Conjugate Gradient method and the Direction Set method. The methods are implemented and applied in system modeling and they are assessed in terms of rate of convergence, computational complexity, misadjustments and their sensitivity to spectral condition number or the eigenvalue spread. Our main objective is to provide a comprehensive understanding of the adaptive implementation of the methods, their performance according to assessment criteria mentioned above and also provide possible modifications to improve the performance.

Key-words: Adaptive least squares problem, adaptive filtering, adaptive algorithms

1 Introduction

Adaptive filtering problems has received considerable attention from the engineering community during the past several decades due to its application in many diverse fields such as system identification, equalization, prediction and noise cancellation. From mathematical perspective, adaptive filtering problem may be viewed as an adaptive least squares problem. The standard least squares problem can be reduced to solving a linear system of equations whereas in the adaptive least squares problem, a time varying linear system is resulted at each state.

As a consequence, many of the algorithms available for solving adaptive filtering problem, are derived from iterative methods for solving linear system of equations. For example, the most widely implemented adaptive algorithms in practice, namely the Least Mean Square (LMS) algorithm is derived from the method of steepest descent. Other search methods which have found their place in solving adaptive filtering problem are the direction set method and the conjugate gradient method [3, 4, 5].

In this paper, we will review several algorithms which are derived from four different iterative search methods, namely the method of steepest descent, the Newton's method, the direction set method and the conjugate gradient method. Our objectives are 1) to provide a clear understanding of their implementation in solving adaptive filtering problem; 2) to evaluate their performance in terms of rate of convergence, misadjustment, computational complexity and sensitivity towards eigenvalue spread, and, 3) to recommend improvements to the algorithms.

2 Adaptive Filtering Problems

Adaptive filtering problem is a filter design technique which allows for adjustable coefficients that can be optimized to minimize some measure of error. Mathematical formulation for adaptive filtering problem usually takes the form of an adaptive least squares problem, where the value of the filter coefficients are adjusted so that they are optimized in the least squares sense. In contrast with the standard least squares problem, the sum of squared error function in the adaptive least squares problem is a time varying function which adapts itself with the time varying input data.

2.1 Adaptive Filters

A schematic diagram of an adaptive filter is given in Fig. 1 below where for every input signal $u(n)$, the filter produces output $y(n)$. This output is compared with a desired signal $s(n)$to produce an error signal $e(n)=y(n)-s(n)$ which is the difference between the output and the desired signal. The objective in the design an adaptive filter is to adjust its parameters so that the error $e(n)$ is minimized.

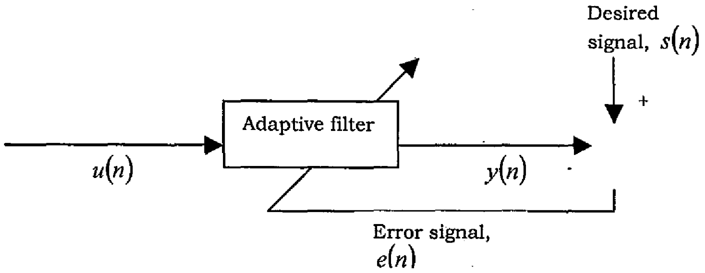


Fig. 1 Block diagram for an adaptive filter

2.2 Mathematical Fomulations for Adaptive Filtering Problems

For a linear linear transversal filter, the output to filter at time n is given by,

$$y_n = \sum_{j=0}^{N-1} x_j u_{n-j}$$

where N is the filter order and $x(j)$ is the j th coefficient of the filter. The n th state of the sum of squared error function is the sum of squared errors from time 0 up to n, which is given by the equation below,

$$J_n(\mathbf{x}) = \sum_{i=0}^{n} \lambda_i^{(n)} \left(\mathbf{a}_i^T \mathbf{x} - s_i\right)^2 \qquad (1)$$

where $\mathbf{a}_i = [u_i \quad u_{i-1} \quad \dots \quad u_{i-N+1}]$, $x = [x_0 \quad x_1 \quad \cdots \quad x_{N-1}]^T$ and $\lambda_i^{(n)}$ can be in two forms, either $\lambda_i^{(n)} = \frac{1}{n}$ or $\lambda_i^{(n)} = \lambda^{n-i}$ where $0 < \lambda < 1$. The first choice of $\lambda_i^{(n)}$ gives rise to an average sum of squared error whereas the second choice gives an exponentially weighted sum of squared error and the weighting factor λ is referred to as the forgetting factor which is intended to ensure that the past data are "forgotten" in order to track the statistical variations of the data in nonstationary environment.

The adaptive least squares problems is the problem of minimizing the cost function (1) with respect to filter coefficients $x_j, j = 0, \dots, N-1$. In matrix form, the adaptive least squares problem may be represented as follows

$$\min_{\mathbf{x}} J_n(\mathbf{x}) = \min_{\mathbf{x}} \left(\mathbf{b}_n{}^T \mathbf{b}_n - 2\mathbf{x}^T \mathbf{A}_n \mathbf{b}_n + \mathbf{x}^T \mathbf{A}_n \mathbf{A}_n^T \mathbf{x}\right) \qquad (2)$$

where

$\mathbf{A}_n = \left[\sqrt{\lambda_0^{(n)}}\mathbf{a}_0, \sqrt{\lambda_1^{(n)}}\mathbf{a}_1, \dots \sqrt{\lambda_n^{(n)}}\mathbf{a}_n\right]^T$ and $\mathbf{b}_n = \left[\sqrt{\lambda_0^{(n)}}s_0, \sqrt{\lambda_1^{(n)}}s_1, \dots, \sqrt{\lambda_n^{(n)}}s_n\right]$.

The quantity $\mathbf{R} = \mathbf{A}_n \mathbf{A}_n^T$ is identified as the autocorrelation matrix of the input signal and the vector $\mathbf{p} = \mathbf{A}_n \mathbf{b}_n$ is the vector of cross-correlation between the input and the desired signal.

3 Iterative Search Methods and Application in Adaptive Filtering Algorithms

We note that the minimization problem given in (2) has an exact solution at every state n, which is the solution to a system of $N \times N$ linear equation

$$\mathbf{Rx} = \mathbf{p}$$

so that at every state n, the optimum solution is given by

$$\mathbf{x}_{opt}(n) = \mathbf{R}(n)^{-1}\mathbf{p}(n).$$

Hence the adaptive least squares problem can be viewed as an adaptive search for the solution to a (time varying) system of $N \times N$ linear equations.

In this section we will review three classes of algorithms which are based on four iterative search methods, the steepest descent method, the Newton's method, the direction set method and the Conjugate Gradient method.

3.1 LMS and LMS-Newton Algorithm

In this section we will be discussing the most widely used adaptive algorithm and that is the Widrow-Hoff Least Mean Square (LMS) algorithm derived in 1959 [8]. It is based on the method of steepest descent where the coefficient vector $\mathbf{x}$ is updated along the direction of steepest descent, i.e. the negative gradient. This gives the following recursion formula for the coefficient vector,

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mu\nabla \quad (4)$$

where ∇ denotes the gradient vector. For the least squares problem, the gradient vector is given by $\nabla = 2(\mathbf{Rx} - \mathbf{p})$. The LMS algorithm is obtained by replacing the gradient vector with its instantaneous value and that is $-2e_n\mathbf{x}_n$ where $e_n = \mathbf{y}_\text{n} - s_n$ is the instantaneous error.

In the Newton's method, the gradient vector in (4) is scaled with a factor $\lambda_{ave}\mathbf{R}^{-1}$ where λ_{ave} is the average of the eigenvalues of $\mathbf{R}$, giving

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mu\lambda_{ave}\mathbf{R}^{-1}\nabla$$

Using the instantaneous value as the estimate for the gradient we have the following recursion formula for the LMS-Newton algorithm

$$\mathbf{x}_{n+1} = \mathbf{x}_n + 2\mu\lambda_{ave}\mathbf{R}^{-1}e_n\mathbf{x}_n.$$

It is a well known fact that the LMS algorithm is sensitive to the spread in the eigenvalue of the correlation matrix. This property is inherited from the method of steepest descent where several modes of convergence exist which corresponds to the number of distinct eigenvalues in $\mathbf{R}$. Because prior knowledge of $\mathbf{R}$ is rarely known, this property makes the rate of convergence of the LMS algorithm unpredictable. The LMS-Newton algorithm is considered as an improved version where it only has one mode of convergence [9]. However, because the LMS-Newton algorithm requires knowledge of $\mathbf{R}^{-1}$, this algorithm cannot be implemented in practice.

3.2 The Direction Set Based Algorithm

The direction set method is inherited from the Powell and Zangwill method for optimizing unconstrained minimization problem [8, 13]. Given a starting estimate $\mathbf{x}$ and a set of N linearly independent direction $\{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$, the direction set method searches along each direction sequentially for a better estimate. The search through N directions is called one cycle. Before the next search cycle, directions may or may not be modified (depending on the linear independence of the new set of directions), and a new starting estimate maybe chosen. The iterative algorithm for updating the coefficient vector within each cycle takes the form

$$\mathbf{x}_{i+1}^{(n)} = \mathbf{x}_i^n + \alpha_i^{(n)}\mathbf{d}_i^{(n)}, \qquad i = 1, \ldots N$$

where $\alpha_i^{(n)}$ is the stepsize and $\mathbf{d}_i^{(n)}$ is the search direction. The optimal stepsize can be obtained by setting $\nabla_\alpha J(\mathbf{x} + \alpha\mathbf{d}) = 0$ which gives

$$\alpha_i^{(n)} = -\frac{\mathbf{d}_i^{(n)T}\left(\mathbf{R}(n)\mathbf{x}_i^{(n)} - \mathbf{p}(n)\right)}{\mathbf{d}_i^{(n)T}\mathbf{R}(n)\mathbf{d}_i^{(n)}}. \quad (5)$$

The simplest form of the direction set method is obtained by choosing the Euclidean directions as the search direction at each cycle, i.e. $\mathbf{d}_i^{(n)} = [0\ldots010\ldots0]^T$ where the 1 appears in the i th position. This gives rise to the Euclidean Direction Search (EDS) algorithm. A further modification of the EDS algorithm can be found in [11, 12] (Fast EDS algorithm) and [2] (Scaled EDS algorithm).

3.3 The Conjugate Gradient Based Algorithm

The Conjugate Gradient method (CG) looks for a set of linearly independent direction vectors $\{\mathbf{d}_1,\ldots,\mathbf{d}_N\}$ which are conjugate with respect to $\mathbf{R}$ so that the solution vector $\mathbf{x}^*$ can be expressed as

$$\mathbf{x}^* = \alpha_1\mathbf{d}_1 + \alpha_2\mathbf{d}_2 + \ldots + \alpha_N\mathbf{d}_N .$$

Minimization of the cost function J gives rise to the same formula for the stepsize α as given in (5) although conjugacy with respect to $\mathbf{R}$ is not a requirement for the search directions in the direction set method.

In order to maintain conjugacy with respect to $\mathbf{R}$, the i th search direction is updated as follows

$$\mathbf{d}_i = \mathbf{g}_{i-1} + \beta_{i-1}\mathbf{d}_{i-1}$$

where $\mathbf{g}_i$ is the gradient vector at the i th iteration and β_i is a scalar value given as $\beta_i = \frac{\mathbf{g}_i^T\mathbf{g}_i}{\mathbf{g}_{i-1}^T\mathbf{g}_{i-1}}$. Since the gradient vector in the adaptive filtering problem is dependent on the correlation matrix $\mathbf{R}$ and the cross-correlation vector $\mathbf{p}$, estimates of the gradient requires estimates of both $\mathbf{R}$ and $\mathbf{p}$. We will highlight two ways of implementing the CG method in adaptive filtering problem [3, 4] which employs two different technique of estimating $\mathbf{R}$ and $\mathbf{p}$:

i) The cost function is assumed to be the averaged sum of squared error, i.e.

$$J_n(\mathbf{x}) = \frac{1}{n}\sum_{i=0}^{n}\left(\mathbf{a}_i^T\mathbf{x} - s_i\right)^2$$

so that the correlation matrix $\mathbf{R}_n = \frac{1}{n}\mathbf{A}_n\mathbf{A}_n^T$, where $\mathbf{A}_n = [\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_n]^T$ and the crosscorrelation vector becomes $\mathbf{p}_n = \frac{1}{n}\mathbf{A}_n\mathbf{b}_n$ where $\mathbf{b}_n = [s_0,s_1,\ldots,s_n]^T$. In the CG algorithm for adaptive filtering, the finite sliding data windowing is used to estimate $\mathbf{R}_n$ and $\mathbf{p}_n$, where, only data samples inside a window of finite length M are used. Hence we have the following estimates,

$$\mathbf{R}_n = \frac{1}{M}\sum_{j=n-M+1}^{n}\mathbf{a}_j\mathbf{a}_j^T, \qquad \mathbf{p}_n = \frac{1}{M}\sum_{j=n-M+1}^{n} s(j)\mathbf{a}_j.$$

For every incoming data sample, the conjugate gradient iteration is run $k_{\max}$ times, where $k_{\max} = \min(N, M)$.

ii) The cost function is assumed to be the exponentially weighted sum of squared error

$$J_n(\mathbf{x}) = \sum_{i=0}^{n}\lambda^{n-i}\left(\mathbf{a}_i^T\mathbf{x} - s_i\right)^2$$

Using an exponentially decaying data windowing, the correlation matrix and the cross-correlation vector may be updated recursively as follows,

$$\mathbf{R}_n = \lambda\mathbf{R}_{n-1} + \mathbf{a}_n\mathbf{a}_n^T$$
$$\mathbf{p}_n = \lambda\mathbf{p}_{n-1} + s(n)\mathbf{a}_n$$

In this implementation, the conjugate gradient iteration is performed once for every incoming data sample.

4 Comparative performance

In our discussions here, we will be assessing the performance of adaptive algorithms in section 3 in the framework of adaptive system modeling. The block diagram for adaptive system modeling is in Fig. 2. The input signal is filtered through a colouring filter with the frequency response $H(z) = \frac{\sqrt{1-\alpha^2}}{1-\alpha z^{-1}}$, where $|\alpha| < 1$. The parameter α controls the eigenvalue spread of the input correlation matrix, where $\alpha = 0$ gives uncorrelated sequence (white) with small eigenvalue spread. The aim is to find the parameters of a model $\mathbf{x}$ through an adaptive algorithm so that the difference between the unknown system output, $d(n)$, and the adaptive model output, $y(n)$, is minimized according to some specified cost function. Noise, $\eta(n)$, with a variance of 0.001 is added to the output of the unknown system.

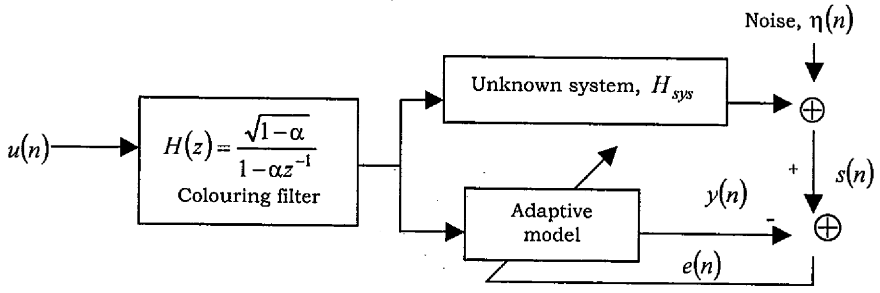
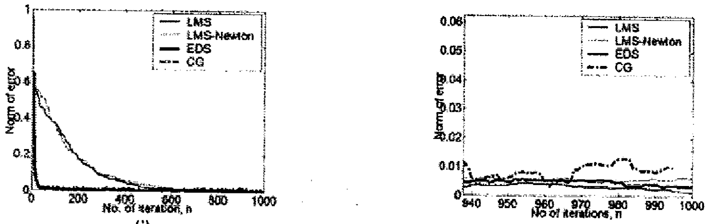


Fig. 2 Adaptive system modeling

4.1 Rate of Convergence and Misadjustment

An efficient adaptive algorithm is one that minimizes usage of data without compromising the quality of solution. In other words we require the algorithm to have a reasonably high rate of convergence and at the same time it keeps the solution as close as possible to the optimum solution. Commonly, in adaptive filtering problems, the rate of convergence is assessed by the number of iterations required to achieve the steady state mean squared error (MSE). In addition to that, the quality of the steady state solution is measured through the quantity called misadjusment which is the ratio of the excess MSE (the difference between the steady state MSE and the MSE corresponds to the optimum solution) to the steady state MSE. In our discussions here, because the exact solution is available, we will be evaluating rate of convergence and misadjustments by looking at the progression of error between the coefficient vector of the unknown system and that of the adaptive filter.

Fig. 3 (i) displays the progression of error (computed in norm-2) as the number of iteration increases. It is clearly shown that both LMS and LMS-Newton algorithms have comparable rate of convergence where a steady state error is achieved after 600 iterations. However the EDS and the Conjugate Gradient based algorithms provide a much superior convergence rate, where steady state error is achieved only after about 40 iterations.



(i) (ii)

Fig. 3 Progression of error as iteration increases. Error is measured by the norm of the difference between the coefficient vector of the unknown system and the coefficient vector of the adaptive model ($N = 5$)

The steady state errors shown in Fig. 3 (ii) gives a better view of misadjustments produced from the solutions. It can be seen that although conjugate gradient method has a high rate of convergence, it does tend to produce higher misadjustment compared to the other algorithms.

4.2 Computational complexity

Another important assessment criterion is the computational complexity of the algorithm. A higher computational complexity means the algorithm requires a higher storage capacity and computation time. High computational complexity also makes the algorithm more susceptible to round off errors, hence, reducing the quality of solution.

Table 1 below summarizes the computational complexity of the algorithms discussed as a function of the adaptive filter order, N.

Table 1 Computational complexity

Algorithm	Complexity
LMS	$O(N)$
EDS	$O(N^2)$
Fast EDS	$O(N)$
CG (Implementation (i))	$O(N^2)$
CG (Implementation (ii))	$O(N^2)$

To see the effect of computational complexity on convergence and misadjustments, we plot the error profile for a much larger N in Fig. 4.

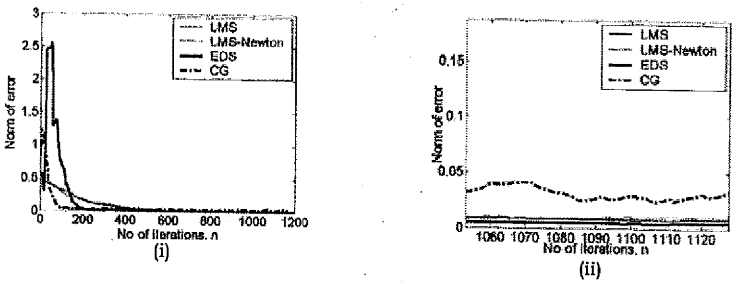


Fig. 4 Rate of convergence and misadjustment for $N = 30$

Increasing the value of N has little effect on the LMS-based algorithms. However, for the EDS algorithm, we see a slower convergence rate than in previously and a significant increase in the initial errors. The misadjustment for the EDS algorithm remains comparable to that of the LMS-based algorithm. For the CG algorithm, although the convergence rate remains unchanged, we see a significant increase in the misadjustment.

4.3 Sensitivity to eigenvalue spread

We now compare the sensitivity of the solutions for our problem towards eigenvalue spread for all the algorithms. Note that eigenvalue spread is defined as $\lambda_{max}/\lambda_{min}$, where λ_{max} is the largest eigenvalue of R and λ_{min} is the smallest eigenvalue.

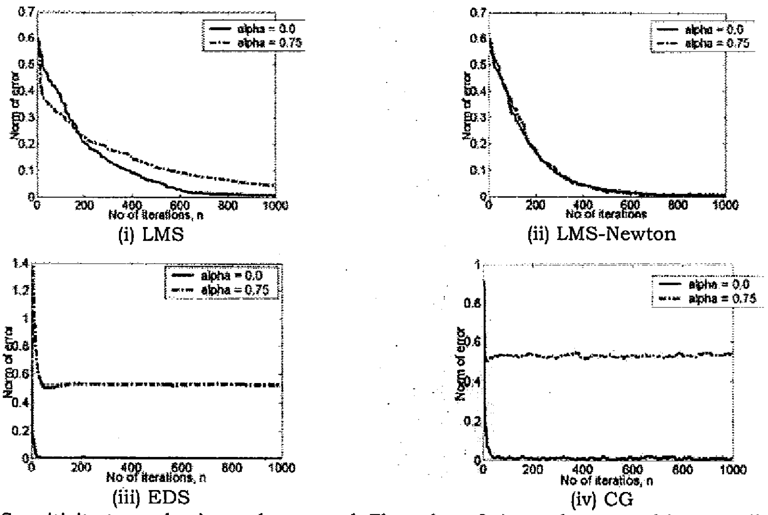


Fig.5 Sensitivity towards eigenvalue spread. The value of eigenvalue spread is controlled by the parameter α .

As predicted, the convergence rate of the LMS algorithm is reduced for large eigenvalue spread and almost no effect on the LMS-Newton algorithm is found. For both the EDS and CG algorithm, although eigenvalue spread has little effect on their rate of convergence, the misadjustments suffers terribly from it.

5 Conclusions

We have reviewed four different adaptive algorithms for solving adaptive filtering problem, namely the LMS algorithm, the LMS-Newton algorithm, the direction set based method and the conjugate gradient based method. These algorithms are treated as iterative search method for solving a time-varying linear systems of equation of the form $\mathbf{Rx} = \mathbf{p}$, where $\mathbf{R}$ corresponds to the time-varying correlation matrix of the adaptive problem and $\mathbf{p}$ corresponds to the time-varying cross-correlation vector.

Performance evaluation of the algorithms are conducted within the framework of an adaptive system modeling problem. The EDS and the CG method proved to have superior rate of convergence compared to the LMS-based method. However, due to the high computational complexity of the CG algorithm, it tends to give higher misadjustment compared to the other algorithms.

The convergence rate of the EDS and the CG algorithm is not affected much by the increase in eigenvalue spread. However, poor misadjustments are obtained. To this end, we note that the eigenvalue spread is the same as the condition number of R. Therefore, preconditioning the algorithms with a suitable preconditioning matrix will help reduce this problem.

Acknowledgment

The author acknowledges support from Universiti Sains Malaysia for the travel grants to attend ICAM05.

References

[1] Ahmad, N.A, Lye, W.K & Ho, Y.Y. (2005), Euclidean Direction Search Algorithm for Adaptive Least Squares Problems, *Proceedings of the 13th National Symposium on Mathematical Sciences,*Universiti Utara Malaysia, **1**, 193-201.
[2] Ahmad, N.A., Lye, W.K. & Ho, Y.Y. (2005), Investigation into Modified Euclidean Direction Search Algorithm for Adaptive Least Squares Problem, *Proceedings of the 2nd International Conference on Research and Education in Mathematics,* Universiti Putra Malaysia (CD Proceedings).
[3] Boray, G. & Srinath, M. (1992), Conjugate Gradient Techniques for Adaptive Filtering, *IEEE Trans. Circuits and Syst. I,* **39**(1), 1-10.
[4] Chang, P.S. & Wilson, A.N. (2000), Analysis of the Conjugate Gradient Algorithms for Adaptive Filtering, *IEEE Trans Signal Processing,* **48**(2), 409-418.
[5] Chen, M.Q. (1998), A Direction set based algorithm for least squares problems in adaptive signal processing, *Linear Algebra and its Applications,* **284**, 73-94.
[6] Chen, M.Q., Bose, T. & Xu, G.F. (1999), A direction set based algorithm for adaptive filtering, *IEEE Trans. Signal Processing* **47**(2), 535 – 539.
[7] Farhang-Boroujeny, B. (2000), *Adaptive Filters:Theory and Applications,* John Wiley & Sons, England.
[8] Powell, M.J.D. (1964), An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives, *Comput. J.* **7**, 155-162.
[9] Widrow, B. & Hoff, M.E. (1960), Adaptive Switching Circuits, *IREWESCON Conv. Rec.,* **4**, 96-104.
[10] Widrow, B. & Kamenetsky, M. (2003), On the Statistical Efficiency of the LMS Family of Adaptive Algorithms, 2872-2880.
[11] Xu, G.F., Bose, T. & Schroeder, J. (1998). Channel equalization using an euclidean direction search adaptive algorithm, *Global Telecommunications Conference (GLOBECOM 98), The Bridge to Global Integration. IEEE.* **6**, 3479 – 3484.
[12] Xu, G.F., Bose, T. & Schroeder, J. (1999). The Euclidean direction search algorithm for adaptive filtering, *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems.* **3**,146 – 149.
[13] Zangwill, W. (1967), Minimizing a Function Without Calculating Derivatives, *Comput. J.* **10**, 293-296.

AHMAD, N.A.: School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia. Phone: +6 +4 6533656, Fax: +6 +4 6570910.
E-mail: atinah@cs.usm.my