International Journal of Electronics Engineering, 2(1), 2010, pp. 121-123

Analysis and Simulation of Adaptive Filter with LMS Algorithm

Amrita Rai[1] & Amit Kumar Kohli[2]

[1]*PhD student, (Signal Processing), Thaper University, Patiala*
[2]*Asst. Prof., ECE Departement, Thaper University, Patiala*

Abstract: A standard algorithm for LMS-filter simulation, tested with several convergence criteria is presented in this paper. We analyze the steady-state mean square error (MSE) convergence of the LMS algorithm when random functions are used as reference inputs. In this paper, we make a more precise analysis using the deterministic nature of the reference inputs and their time-variant correlation matrix. Simulations performed under MATLAB show remarkable differences between convergence criteria with various value of the step size.

Keywords: Adaptive Filter, LMS Algorithm, Convergence Analysis.

1. INTRODUCTION

Adaptive digital signal processing is the study of algorithms and techniques which have the capacity to vary in sympathy with changing statistical properties, characteristic of many real signals. Such techniques have been successfully applied in many application areas, for example, channel equalization in communications beam forming for seismic prospecting, EGG monitoring in medicine, analysis of multiphase flow and the control of dynamic systems. [1-3]

The need for high-speed adaptive filters has prompted the search for alternatives to the popular LMS algorithm. One modification is the replacement of the prediction error term in the LMS update kernel by its Sigma function. At the same time, in noise and echo cancellation problems, reduced residual noise variance often requires error filtering. [4]

The adaptive filter is based upon a Finite Impulse Response (FIR) digital filter structure together with an adaptation algorithm to adjust the coefficients of the filter. The adaptation algorithm adjusts the coefficients of the FIR filter so as to minimise some function of the error, $e(k)$, between the desired response, $d(k)$, and the output of the adaptive filter, $y(k)$. Two families of adaptation algorithms are to be investigated, namely Least Mean Square (LMS) and Recursive Least Squares (RLS). [5]

2. LMS ALGORITHM

The least mean square (LMS) algorithm is widely used in applications to adaptive filtering due to its computational simplicity, unbiased convergence in the mean to the Wiener solution, and the existence of a proof of convergence in a stationary environment.[7]

The LMS algorithm is undoubtedly the most popular algorithm for adaptive signal processing. The popularity of the LMS algorithm is to a large extent due to its computational simplicity. Furthermore, it is generally felt that its behavior is quite simple to understand and the algorithm appears to be very robust. [8]

The LMS algorithm was developed by Windrow and Hoff in 1959. The algorithm uses a gradient descent to estimate a time varying signal. The gradient descent method finds a minimum, if it exists, by taking steps in the direction negative of the gradient. It does so by adjusting the filter coefficients so as to minimize the error. A LMS algorithm can be implemented as shown in Figure 1. [5]

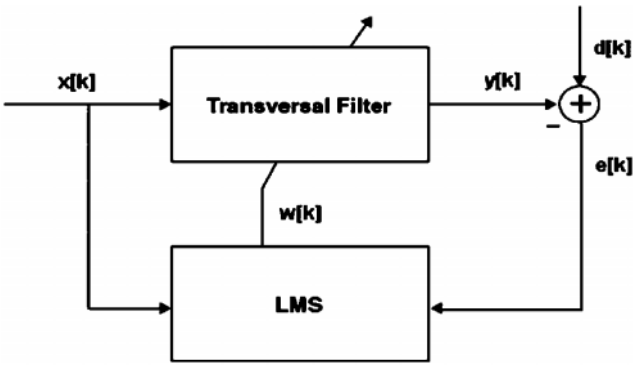

Figure 1: LMS Implementation Using FIR Filter

The desired signal $d(k)$ is tracked by adjusting the filter coefficients $w(k)$. The input reference signal $x(k)$ is a known signal that is fed to the FIR filter. The difference between $d(k)$ and $y(k)$ is the error $e(k)$. The error $e(k)$ is then fed to the LMS algorithm to compute the filter coefficients $w(k + 1)$ to iteratively minimize the error.[5]

The basic LMS algorithm approximately minimises the mean square error $J = \mathrm{E}\{e^2(k)\}$ where the error is given by $d(k) - w^T(k)x(k)$ and $w(k) = [w_1(k), w_2(k),...,w_L(k)]^T$ is the

*Corresponding Author: [1]amritaskrai@gmail.com, [2]akkohli@thaper.edu

coefficient vector of the adaptive filter of length L, and $x(k) = [x(k), x(k-1),..., x(k-L+1)]^T$ is the adaptive filter data vector. The coefficient vector update equation for the LMS algorithm is given by

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

Where the right hand term, $x(k)e(k)$, is an instantaneous estimate of the negative gradient of the error performance surface J [Haykin], and μ is the adaptation gain. The coefficient vector of the adaptive filter is generally initialised to zero. [1]

The presence of feedback within the LMS algorithm, that is the output error is used to adjust the coefficients of the adaptive filter, may lead to instability. The adaptation gain μ is therefore the key parameter which controls how the adaptive filter behaves and it should be chosen to lie within the range [Haykin].[1]

$$0 < \mu > \frac{1}{LP_x} \quad (2)$$

Where Px is the power of the input signal to the adaptive filter.

From this last equation it is possible to show that in order to guarantee convergence of the mean of the coefficients, the convergence factor μ of the LMS algorithm must be chosen in the range

$$0 < \mu > \frac{1}{\lambda_{max}} \quad (3)$$

Where λ_{max} is the largest Eigen value of input signal vector R. Hypothesis, which considers all vectors $x(i)$ statistically independent is questionable. It is not rigorously valid when $x(k)$ represents the elements of a delay line. [7]

The convergence time of the LMS algorithm depends on the step size μ. If μ is small, then it may take a long convergence time and this may defeat the purpose of using an LMS filter. However if μ is too large, the algorithm may never converge. The value of μ should be scientifically computed based on the effects the environment will have on $d(n)$.

3. MATLAB SIMULATION AND RESULTS

This section describes parts of the sample MATLAB program and results. The Program can be divided into four parts: First employ the *randn(N*, 1) function within MATLAB to generate $x(k)$ which will have unity power and zero mean, where N is number of system point. Secondly using butter(2,0.25) for calculating first sysorder weight value for the sysorder is 10.Thirdly add some error signal as noise with input signal and taking $N = 60$ for training. Finally start the algorithm Start with big step size μ for speeding the convergence then slow down to reach the correct weights $w(k+1)$.

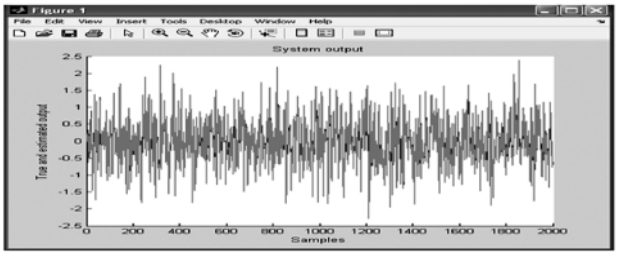

Figure 2: Output Response $y[k]$ and Estimated Output Response $d[k]$.

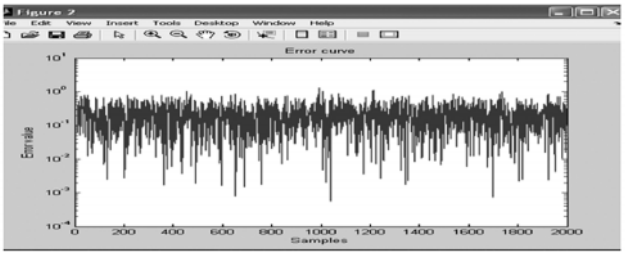

Figure 3: Estimated Error curve $e[k] = d[k] - y[k]$

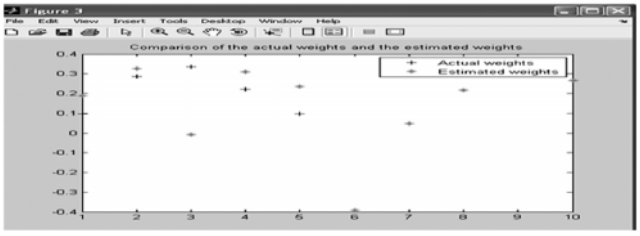

Figure 4: Comparison of the Actual Weights and the Estimated Weight

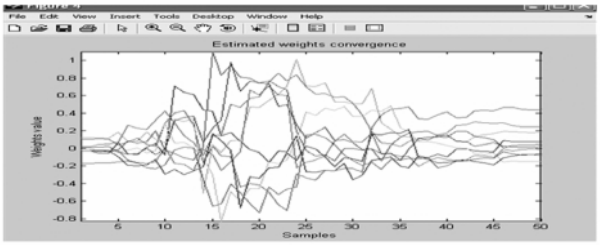

Figure 5: Convergence Curve of LMS Algorithm with Various Step Size

The MATLAB simulation results are again shown in Figure2 to Figure 5 for various μ. Figure 2shows the desired output and estimated output, figure 3 shows Estimated Error curve $e[k] = d[k] - y[k]$, figure 4 shows the comparison between Comparison of the actual weights and the estimated weight and figure 5 shows Convergence curve of LMS algorithm with various step size. The increase in step size results in a significant improvement in terms of clock cycles needed for an error to converge. This is confirmed in Figure 5, as it now takes about 2,000 clock cycles for the estimated signal $y(n)$ to closely track the desired signal $d(n)$. The step size, though, cannot be arbitrarily increased. A high value can eventually cause the LMS algorithm to diverge and this

causes the error to oscillate with high amplitude. The step size, therefore, has to be scientifically computed based on a number of criteria such as the sampling rate, channel properties, signal properties, etc. In this particular example, the frequency of the input sample is constant. Therefore, only the change in the value of the input sample mattered. As shown in Figures 4 and 5, the input sample changes by 0.01 every few clock cycles. This change must be adapted by varying the coefficients. The step size was experimentally varied to obtain faster convergence.

CONCLUSION

In this paper, we analyzed the steady-state MSE convergence of the LMS algorithm using the adaptive filtering with random variable input and IIR Butterworth filter. The MATLAB simulation results are again shown in Figure 5 for various ì. The increase in step size results in a significant improvement in terms of clock cycles needed for an error to converge. The convergence of LMS filter using Butterworth adaptive filter are simulated and give satisfactory results.

REFERENCES

[1] Haykin, S., "Adaptive Filter Theory", 4th Ed., Prentice-Hall, 2002 , *Much improved on* other Editions, Good Sections on Numerical Issues in Adaptive Filtering and Emerging Adaptive Techniques, Particularly Blind Signal Processing.

[2] Widrow, B., and S.D. Steams, "Adaptive Signal Processing", Prentice Hall, 1985, *was* in the Vanguard of Adaptive Filtering, Introductory but Strong on Applications.

[3] Bellanger, M., "Adaptive Digital Filters and Signal Analysis", Marcel Dekker, 1997. *A* Firstclass Book, he has Really Worked with Adaptive Filters.

[4] Dasgupta, S. Garnett, J.S. Johnson, C.R., Jr. "Convergence of an Adaptive Filter with Signed Filtered Error" *This Paper Appears in: Signal Processing, IEEE Transactions on,* **42**, Issue: 4 On page(s): 946 - 950 , ISSN: 1053-587X Current Version Published: 06 August 2002.

[5] www.latticesemi.com, "LMS Adaptive Filter" Lattice Semiconductor Corporation, Decemder 2006.

[6] Andres Frais and Rene de Jesus "Algorithm for Convergence Criteria Simulation on LMS Adaptive Filters" This Paper Appers in : Telecommunication and Radar Engineering. in Year 2005, **64**, Issue 7-12., Page 537-542, ISSN: 0040-2508.

[7] M.I. Troparevsky and C.E. D'Attellis "On the Convergence of the LMS Algorithm in Adaptive Filtering" *Signal Processing 84 (2004)*, Elsevier, Available at www.elsivercomputerscince.com.

[8] Salvador Olmos and Pablo Laguna "Steady-State MSE Convergence of LMS Adaptive Filters with Deterministic Reference Inputs with Applications to Biomedical Signals IEEE Transactions on Signal Processing, **48**, No. 8, August 2000.