ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 5, May 2013

Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS)

JYOTI DHIMAN[1], SHADAB AHMAD[2], KULDEEP GULIA[3]

[1] Department of Electronics Engineering, B.M.I.E.T, Sonepat, India

[2]Asst. Professor, Department of Electronics Engineering, D.R.C.E.T, Panipat, India

[3]Asst. Professor, Department of Electrical Engineering, B.M.I.E.T, Sonepat, India

Abstract: This paper describes the comparison between adaptive filtering algorithms that is least mean square (LMS), Normalized least mean square (NLMS),Time varying least mean square (TVLMS), Recursive least square (RLS), Fast Transversal Recursive least square (FTRLS). Implementation aspects of these algorithms, their computational complexity and Signal to Noise ratio are examined. These algorithms use small input and output delay. Here, the adaptive behaviour of the algorithms is analyzed. Recently, adaptive filtering algorithms have a nice tradeoff between the complexity and the convergence speed. Three performance criteria are used in the study of these algorithms: the minimum mean square error, the algorithm execution time and the required filter order.

Keywords: ***Least mean square (LMS), Normalised Least mean square (NLMS), Time Varying Least mean square (TVLMS), Recursive Least square (RLS).***

I. Introduction

There are many digital signal processing applications in which second order statistics cannot be specified. Such application includes channel equalization echo cancellation and noise cancellation. In these applications, filters with adjustable coefficients called Adaptive filters are employed. An adaptive filter is a filter that self adjusts its transfer function according to an optimizing algorithm. It adapts the performance based on the input signal. Such filters incorporate algorithms that allow the filter coefficients to adapt to the signal statics. There are different approaches used in adaptive filtering, which are as follows:

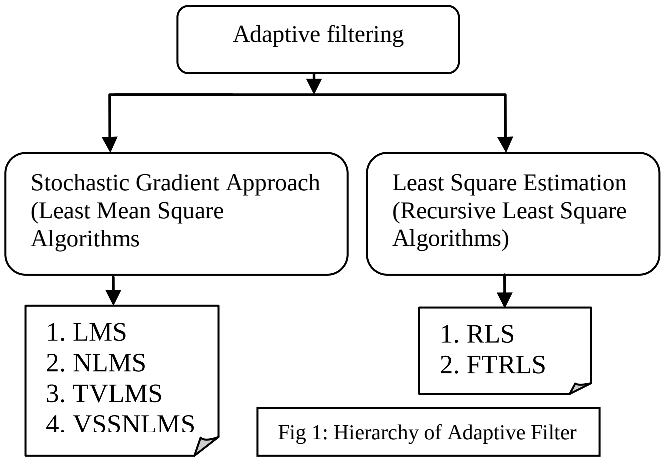
Adaptive filtering
Stochastic Gradient Approach (Least Mean Square Algorithms
Least Square Estimation (Recursive Least Square Algorithms)
1. LMS
2. NLMS
3. TVLMS
4. VSSNLMS
1. RLS
2. FTRLS

Fig 1: Hierarchy of Adaptive Filter

Adaptive techniques use algorithms, which enable the adaptive filter to adjust its parameters to produce an output that matches the output of an unknown system. This algorithm employs an individual convergence factor that is updated for each adaptive filter coefficient at each iteration.

II. LMS Algorithm

Least mean squares (LMS) algorithms are class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time.

The basic idea behind LMS filter is to approach the optimum filter weights (R^{-1} P), by updating the filter weights in a manner to converge to the optimum filter weight. The algorithm starts by assuming a small weights (zero in most cases), and at each step, by finding the gradient of the mean square error, the weights are updated. That is, if the MSE-gradient is positive, it implies, the error would keep increasing positively, if the same weight is used for further iterations, which means we need to reduce the weights. In the same way, if the gradient is negative, we need to increase the weights. So, the basic weight update equation is:

$$w_{n+1} = w_n - \mu\Delta\varepsilon[n]$$

Where, ε represents the mean-square error. The negative sign indicates that, we need to change the weights in a direction opposite to that of the gradient slope.

LMS algorithm summary:

The LMS algorithm [1] for a pth order algorithm can be summarized as

Parameters:	P = filter order
	μ = step size
Initialization:	$\hat{h}(0) = 0$

Computation: For n = 0, 1, 2...

$X(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$

$e(n) = d(n) - \hat{h}^H(n)\, X(n)$

$\hat{h}(n+1) = \hat{h}(n) + \mu\, e^*(n)\, X(n)$

Convergence and stability in the mean of LMS:

As the LMS algorithm does not use the exact values of the expectations, the weights would never reach the optimal weights in the absolute sense, but a convergence is possible in mean. That is even-though, the weights may change by small amounts, it changes about the optimal weights. However, if the variance, with which the weights change, is large, convergence in mean would be misleading. This problem may occur, if the value of step-size μ is not chosen properly.

Thus, an upper bound on μ is needed which is given as

$$0 < \mu < \frac{2}{\lambda\max}$$

Where λ_{max} is an autocorrelation matrix, its eigen vales are non negative. If this condition is not fulfilled, the algorithm becomes unstable. The convergence of the algorithm [4] is inversely proportional to the eigen value spread of the correlation matrix R. When the eigen values of R are widespread, convergence may be slow. The eigen value spread of the correlation matrix is estimated by computing the ratio of the largest eigen value to the smallest eigen value of the matrix. If **μ** is chosen to be very small then the algorithm converges very slowly. A large value of **μ** may lead to a faster convergence but may be less stable around the minimum value.

Maximum convergence speed [4] is achieved when

$$\mu = \frac{2}{\lambda\, max\ + \lambda min}$$

Where λ_{min} is the smallest eigen value of R. Given that μ is less than or equal to this optimum, the convergence speed is determined by λ_{min}, with a larger value yielding faster convergence. This means that faster convergence can be achieved when λ_{max} is close to λ_{min}, that is, the maximum achievable convergence speed depends on the eigen value spread of R.

III. NORMALISED LEAST MEAN SQUARE (NLMS) ALGORITHM

The main drawback of the "pure" LMS algorithm is that it is sensitive to the scaling of its input. This makes it very hard to choose a learning rate μ that guarantees stability of the algorithm. The *Normalised least mean squares (NLMS) filter* [6], [7] is a variant of the LMS algorithm [1] that solves this problem by normalising with the power of the input.

NLMS algorithm summary:

Parameters: P = filter order

μ = step size

Initialization: $\hat{h}(0) = 0$

Computation: For n = 0, 1, 2...

$X(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$

$e(n) = d(n) - \hat{h}^H(n)\, X(n)$

$$\hat{h}(n+1) = \hat{h}(n) + \frac{\mu\, e*(n)\, X(n)}{X^H(n)\, X(n)}$$

Optimal learning rate:

It can be shown that if there is no interference [v(n) = 0], then the optimal learning rate for the NLMS algorithm [5]-[9] is

$$\mu_{opt} = 1$$

and is independent of the input X(n) and the real (unknown) impulse response h(n). In the general case with interference v(n) does not equal to 0, the optimal learning rate is

$$\mu_{opt} = \frac{E\,[\,|\,y(n) - \hat{y}(n)\,|^2]}{E\,[\,|\,e(n)\,|^2]}$$

The results above assume that the signals v(n) and X(n) are uncorrelated to each other.

Time varying Least Mean Square (TVLMS) Algorithm:

Recently, a new version of the LMS algorithm with time varying convergence parameter has been defined. The TVLMS algorithm has shown better performance than the conventional LMS algorithm in terms of faster convergence and less mean square error. The TVLMS algorithm is based on utilizing a time varying convergence parameter with general power for LMS algorithm.

The basic idea of TVLMS algorithm is to utilize the fact that the LMS algorithm need a large convergence parameter value to speed up the convergence of the filter coefficient to their optimal values, ***the convergence parameter should be small for better accuracy***. In other words, we set the convergence parameter to a large value in the initial state in order to speed up the algorithm convergence. As time passes, the parameter will be adjusted to a small value so that the adaptive filter will have a smaller mean squared error.

IV. RECURSIVE LEAST SQUARE (RLS) ALGORITHM

The Recursive least squares (RLS) adaptive filter is an algorithm which recursively finds the filter coefficients that minimize a weighted linear least squares cost function relating to the input signals. The RLS algorithms are known for their excellent performance when working in time varying environments but at the cost of an increased computational complexity and some stability problems. In this algorithm the filter tap weight vector is updated using Eq.

$$w(n) = \overline{w}^T(n-1) + k(n)\,\overline{e}_{n-1}(n) \quad \ldots\ldots (1)$$
$$k(n) = u(n) / (\lambda + X^T(n)\,u(n)) \quad \ldots\ldots (2)$$
$$u(n) = \overline{w}_\lambda^{-1}(n-1)\,X(n) \quad \ldots\ldots (3)$$

Eq. (2) and (3) are intermediate gain vector used to compute tap weights.

Where λ is a small positive constant very close to, but smaller than 1. The filter output is calculated using the filter tap weights of above iteration and the current input vector as in Eq. (4).

$$\overline{y}_{n-1}(n) = \overline{w}^T(n-1)\,X(n) \quad \ldots\ldots (4)$$

$$\overline{e}_{n-1}(n) = d(n) - \overline{y}_{n-1}(n) \quad \ldots\ldots (5)$$

In the RLS Algorithm [1] the estimate of previous samples of output signal, error signal and filter weight is required that leads to higher memory requirements.

Table I
Performance Comparison of Adaptive Algorithms

S. No.	Algorithms	MSE	Complexity	Stability
1.	LMS	$1.5*10^{-2}$	2N+1	Less Stable
2.	NLMS	$9.0*10^{-3}$	3N+1	Stable
3.	RLS	$6.2*10^{-3}$	$4N^2$	High Stable

From the Table I show that, the performance of RLS adaptive algorithm is high as compared to other algorithm due to the less mean-square error (MSE) [2].

Fast Transversal RLS Algorithm:

FTRLS algorithm involves the combined use of four transversal filters for forward and backward predictions, gain vector computation and joint process estimation. The main advantage of FTRLS algorithm is reduced computational complexity as compared to the other available solutions.

Background on LMS, NLMS and RLS Algorithm:

Figure show the adaptive filter setup, where X(n), d(n) and e(n) are the input, the desired and the output error signals, respectively. The vector $\hat{h}(n)$ is the (px1) column vector of filter coefficient at time n, in such a way that the output of signal, y(n), is good estimate of the desired signal, d(n).

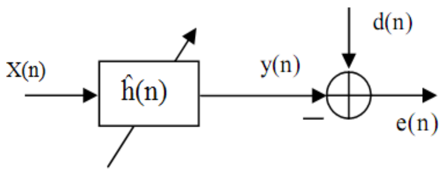

Fig.2: Adaptive filter setup

Filter vector update equation for the LMS algorithm is given by equation

$$\hat{h}(n+1) = \hat{h}(n) + \mu\, e^*(n)\, X(n)$$

Where, $X(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$

and **μ** is the step-size that determines the convergence speed and steady-state mean-square error (MSE). Also, the output error signal, e(n), is given by

$$e(n) = d(n) - \hat{h}^H(n)\, X(n)$$

To increase the convergence speed of the LMS algorithm, the NLMS algorithms [1] was proposed which can be stated as

$$\hat{h}\,(n+1) = \hat{h}\,(n) + \frac{\mu\, e*(n)\, X(n)}{X^H(n)\, X(n)}$$

The filter vector update equation in RLS algorithm is

$$\hat{h}(n+1) = \hat{h}(n) + C^{-1}(n)\, e^*(n)\, X(n),$$

where C(n) is the estimation of the autocorrelation matrix [2]. This matrix is given by

$$C(n) = \sum_{i=0}^{n} \lambda^{n-i} X(i) X^T(i)$$

The λ parameter is the forgetting factor and $0 << \lambda < 1$.

V. Conclusion

Here the comparison between different algorithms is described by SNR improvement table [3]. The SNR improvement of LMS, NLMS and RLS adaptive algorithm shown in the Table II at 1.5 kHz sampling rate and different noise variance, according to the table the RLS adaptive algorithm has improved SNR in dB. It concludes that the best adaptive algorithm is Recursive Least Square according to the SNR improvement table and graph of MSE.

TABLE II
SNR IMPROVEMENT IN DB

Noise Variance	Sampling Rate (kHz)	SNR Improvement (dB) LMS	SNR Improvement (dB) NLMS	SNR Improvement (dB) RLS
0.02	1.5	8.85	9.85	9.91
0.05	1.5	7.55	8.62	8.89
0.10	1.5	5.12	6.38	7.02

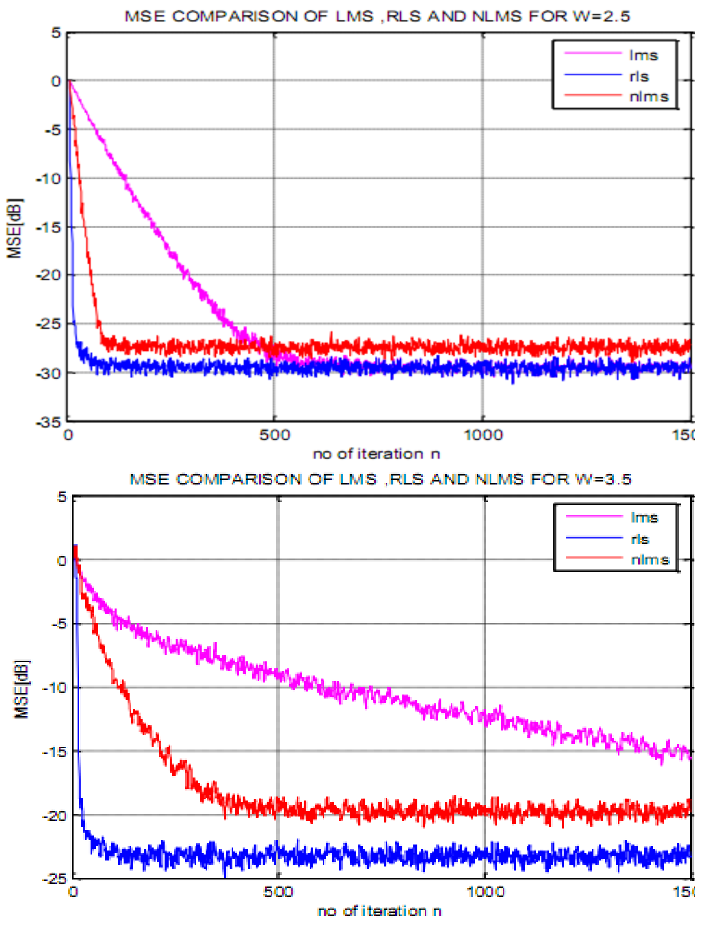

Fig 3: Convergence Comparison for different values of wait (W=2.5, W=3.5)

To compare the performance, the mean square error (MSE) of LMS, NLMS and RLS algorithms for different values of waits W are shown in above fig 2, fig 3, and fig 4, the conclusion is that RLS algorithm exhibits high initial convergence speed and less steady state error compare to both LMS and NLMS algorithm.

It is clear from the graphs, LMS algorithm takes more iteration 500, 1000, and more than 1500 iteration required for achieve the steady state error (e_{ss}) but in NLMS and RLS algorithm to achieve the steady state error is less number of iteration along with mean square error is also less as compared to LMS algorithm for different weight vector [8].

Here we can say that in the comparison of the LMS and NLMS algorithm, the RLS approach offers faster convergence and smaller error with respect to the unknown system.

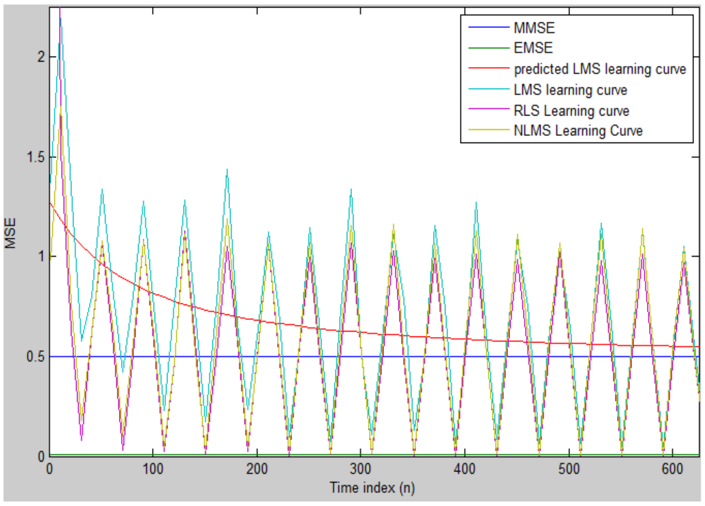

Fig 4: Comparison between learning curve of different Algorithms.

References

[1] Raj Kumar Thenua and S.K. AGARWAL" Simulation and Performance Analysis of Adaptive Filter in Noise Cancellation" International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4373-4378.

[2] Sayed.A.Hadei and M.lotfizad,"A Family of Adaptive filter Algorithms in Noise Cancellation For Speech Enhancement", International Journal of Computer and Engineering, vol.2, No.2, April2010, 1793-8163.

[3] Raj Kumar Thenua, "Hardware Implementation of Adaptive Algorithms for Noise Cancellation", 2011 International Conference on Network Communication and Computer (ICNCC 2011)

[4] John G. Proakis, "Digital Signal Processing Principles, Algorithms and Applications", Pearson Prentice Hall, fourth Edition, page No. 909-911.

[5] Monson H. Hayes: Statistical Digital Signal Processing and Modeling, Wiley, 1996, ISBN 0-471-59431-8.

[6] Simon Haykin: Adaptive Filter Theory, Prentice Hall, 2002, ISBN 0-13-048434-2.

[7] Simon S. Haykin, Bernard Widrow (Editor): Least-Mean-Square Adaptive Filters, Wiley, 2003, ISBN 0-471-21570-8.

[8] Analysis and comparison of RLS adaptive filter in signal De-noising, GuoQuan Xing; YuXia Zhang, Dept. of Biomed. Eng., Xianning Coll.

[9] Paulo S.R. Diniz: Adaptive Filtering: Algorithms and Practical Implementation, Kluwer Academic Publishers, 1997, ISBN 0-7923-9912-9.