Zaria Journal of Electrical Engineering Technology, Department of Electrical Engineering, Ahmadu Bello University, Zaria – Nigeria.
Vol. 8 No. 2 September2019. ISSN: 0261 – 1570.

COMPUTATIONAL COMPLEXITY ANALYSES OF ADAPTIVE EQUALIZATION ALGORITHMS IN LINEARLY DISPERSED CHANNEL SYSTEMS

O.E. Ochia*, E. Obi
Department of Electronics and Telecommunication Engineering, Ahmadu Bello University, Zaria – Nigeria.
*oeochia@abu.edu.ng

Keywords: –
Equalization,
Least Mean Squares,
Normalized Least Mean Squares,
Recursive Least Squares.

Article History: –
Received: January, 2019.
Reviewed: April, 2019
Accepted: August, 2019
Published: September,2019

ABSTRACT

This paper presents a framework for assessing the complexity of adaptive equalization algorithms in a linearly dispersive channel that produces unknown distortion. Three algorithms are investigated including the Least Mean Squares (LMS), Recursive Least Squares (RLS), and Recursive Least Squares Lattice (RLSL) algorithms with respect to the mean square error (MSE) and the sample convergence speed. The simulation results reveal several insights: In terms of channel dispersion, the MSE performance of the LMS deteriorates by up to 40 % when the channel eigenvalue spread doubles. In addition, the convergence speed of LMS reduces by up to 50% for the same increase in channel spread and is invariant to the filter order. The same observation is applicable to the RLS and RLSL algorithms. The RLS algorithm gives the highest MSE performance for practical signal to noise ratio (SNR) ranges; It outperforms the LMS scheme by up to 50% and the RLSL by up to 20%. In terms of convergence speed, the RLS algorithm converges fastest at around 100 data samples and the LMS is the slowest requiring 800 samples while the RLSL algorithm requires up to 200 samples to converge. Observation of other metrics of the RLS and RLSL algorithms including the last tap weight coefficient of the LMS/RLS and the steady-state regression coefficient of the RLSL reveal the symmetry and asymmetry in their statistics respectively. The choice of the equalization algorithm to be used depends on a number of design tradeoffs including the propagation environment, the SNR sensitivity, and the computational power.

1. INTRODUCTION

Equalization is the technique of reversing the distortion encountered by a signal that is transmitted through a wireless channel that is subject to multipath fading and other forms of channel distortion. In this regard, several types of equalization techniques have been proposed and studied in recent works. The Least Mean Squares (LMS) algorithm is based on a stochastic gradient search method [1] where a step size must be selected to update a set of weight coefficients and this selection must be made to ensure that the tap weights converge [2]. However, the slow convergence rate and the need for long training sequences are the major disadvantages of the LMS algorithm as noted in [2] where a fast-start-up modified LMS (FSU-M-LMS) algorithm based on channel matched filter (CMF) was proposed to increase the convergence speed and performance of the LMS. Further, the authors in [3] proposed a variable step-size LMS adaptive algorithm which uses the gradient of the filter coefficient vector to accelerate the convergence speed while ensuring the convergence accuracy. The step size update is adjusted to enhance the ability of the algorithm to resist noise interference.

A normalized form of the LMS algorithm (NLMS) that is independent of the step-size parameter is studied in [4] where its performance is compared to the standard LMS scheme with regard to the convergence speed. The results show that the normalized least mean square (NLMS) algorithm converges faster than the LMS algorithm where the design of the adaptive filter is based on the limited knowledge of its input signal statistics. More recent studies have proposed improved NLMS schemes like[5] where an NLMS style variable tap-length (VTL) algorithm is proposed to efficiently normalize the step-size in the update equation for the tap-length and a fractional gradient is incorporated with the weight-update equation of the adaptive filter. The results demonstrated a robust performance of the proposed algorithm in a time-varying environment.

The Recursive Least Squares (RLS) scheme is a family of algorithms which use a windowing technique and a forgetting factor to ensure exponential windowing while avoiding the need to solve an associated linear system by back-substitution which may lead to ill-conditioning as discussed in [6]. The robustness of the RLS has been exploited in adaptive filtering where a minimum-disturbance (MD) constraint with a sparsity-promoting penalty and a regularization factor to control the contributions of both the MD constraint and the sparsity-promotion is introduced to the standard RLS [7]. Analytical and simulation results demonstrate the increased robustness and lower misalignment attribute of the algorithm which is benchmarked with a scheme where the forgetting factor is adjusted manually to obtain the best performance.

Different from existing work, this paper presents a holistic analysis of the computational complexity performance of the standard LMS, the NLMS, and the RLS algorithms. The steady-state MSE convergence of each scheme is studied in the context of a wireless channel which introduces an unknown distortion to signal transmissions. Specifically, the contributions in this paper are listed as:

- A communication system that generates binary phase-shift keying (BPSK) signals is modeled which is subjected to a multi-path channel characterized by time and frequency varying response and additive white gaussian noise (AWGN).
- Four fading channels with three path gains are studied with various eigenvalue spreads to capture the different degrees of channel distortion.
- The effect of the eigenvalue spread, filter order and the step-size parameter and a comparison of learning curves based on the MSE performance of each algorithm is presented.
- A performance comparison in terms of the average number of samples required to attain steady-state convergence in each scheme is provided. In addition, insights are provided on the design factors to be considered in selecting the appropriate equalization scheme.

2. SYSTEM DESIGN AND METHODOLOGY

The system design and methodology used in the computation complexity analyses of the adaptive equalization algorithms in a linearly dispersed channel is as explained in this section.

2.1 Communication Design and Adaptive Filter Stage

The proposed communication system is modeled where binary phase-shift key (BPSK) data is generated at the data source and transmitted into the wireless channel. The channel is modeled as a multipath channel having different path gains/attenuations, h_i where i is the time sample index of a channel. The noise parameter is modeled as a Gaussian process with zero mean and power spectral density of N_o having a typical value of 174 dBm/Hz. The system also comprises an equalization stage which features a linear filter and an adaptive algorithm block, modeled as a feedback stage to the filter and a feedforward stage to both the transmitter and the receiver modules of the system. Specifically, four channels with three path gains per channel which account for the time diversity are studied within the context of the communication system as illustrated in Fig. 1. and Table 1, respectively.

Table 1. Different Multipath Channels Under Investigation

Index	$h(1)$	$h(2)$	$h(3)$
Channel 1	0.2194	1	0.2194
Channel 2	0.2798	1	0.2798
Channel 3	0.3365	1	0.3365
Channel 4	0.3887	1	0.3887

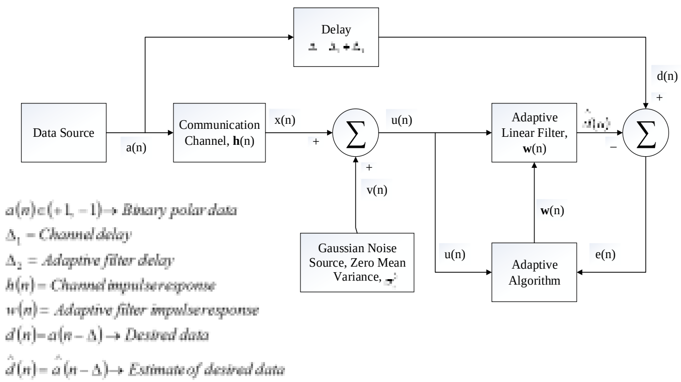

Fig. 1. Block Diagram of the communication system showing the adaptive equalization stage

2.2 Adaptive Equalization Algorithms

The adaptive equalization algorithms of interest are the LMS, NLMS, and RLS.

2.2.1 LMS

The LMS algorithm is a stochastic gradient search algorithm which is based on the gradient search method. A one-dimensional quadratic surface is searched iteratively for the optimal weight solution which is assumed to exist a priori. Based on the search, the weight vector of filter coefficients are updated iteratively. This method avoids solving the Wiener-Hopf equations directly which is computationally intensive and time-consuming. However, a key parameter which is the step-size must be selected to update the weight coefficients. This selection must be made to satisfy the tight bounds provided by the second-order analysis in order for the tap-weights to converge. The bound relates the step-size with the channel eigenvalues.

Data:

M = filter order
μ= step size

$$N = nITER$$

Initialization;
$\boldsymbol{h}= zeros(M)$
Computation;
for$j = 0, 1, 2, \ldots, N$**do**

$$x(j) = [x(j), x(j-1), \ldots, x(j-M+1)]^T$$
$$e(j) = d(j) - \boldsymbol{h}^{\boldsymbol{H}}(j)\boldsymbol{x}(j)$$
$$\boldsymbol{h}(\mathrm{j}+1) = \boldsymbol{h}(j) + \mu \boldsymbol{e}^*(j)\boldsymbol{x}(\boldsymbol{j})$$

end

Algorithm 1: LMS Algorithm

2.2.2 NLMS

The NLMS which is independent of the step-size parameter is studied in comparison with the LMS algorithm. This form does not utilize the step-size in updating the weights.

Data:

M = filter order
λ = forgetting factor
δ = value to initialize $P(0)$

$$N = nITER$$

Initialization;

$$w(j) = 0$$

$x(k) = 0,\ k = -p, \ldots, -1$
$\mathrm{d(k)} = 0,\ k = -p, \ldots, -1$
$P(0) = \delta I$, where I is the identity matrix of rank $p + 1$

Computation;
for $j = 0, 1, 2, \ldots, N$ do

$$\boldsymbol{x}(j) = [x(j), x(j-1), \ldots, x(j-M)]^T$$

$$\alpha(j) = d(j) - \boldsymbol{x}^T(j)\boldsymbol{w}(j-1)$$
$$\boldsymbol{g}(\boldsymbol{j}) = \boldsymbol{P}(\boldsymbol{j} - \boldsymbol{1})\boldsymbol{x}(\boldsymbol{j})\{\boldsymbol{\lambda} + \boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{j})\boldsymbol{P}(\boldsymbol{j} - \boldsymbol{1})\boldsymbol{x}(\boldsymbol{j})\}^{-\boldsymbol{1}}$$

end

Algorithm 2: NLMS Algorithm

2.2.3 RLS

This algorithm uses a windowing technique where a forgetting factor is chosen to ensure exponential windowing and samples closer to the current sample are weighted more while de-emphasizing the samples farther away from the current sample. The RLS also avoids solving the normal equations directly and instead runs a recursive loop where the Kalman gain vector and a priori error are used in updating the filter weights.

Initialization**:**

for $i = 0, 1, 2, \ldots, N$ do

$\delta(-1,i) = \delta_D(-1,i) = 0 \ (iff\ x(j = 0,\ for\ j < 0))$

$\varepsilon^d_{b_{min}}(-1,i) = \varepsilon^d_{f_{min}}(-1,i) = \varepsilon$

$\gamma(-1,1) = 1$

$e_b(-1,i) = 0$

end

for $j > 0$ do

$\gamma(j,0) = 1$

$e_b(k,0) = e_f(k,0) = x(k)$

$\varepsilon^d_{b_{min}}(j,0) = \varepsilon^d_{f_{min}}(j,0) = x^2(k) + \lambda\varepsilon^d_{f_{min}}(k-1,0)$

$e(k,0) = d(k)$

for $i = 0, 1, \ldots, N$ do

$\delta(j,i) = \delta(j-1,i) + \frac{e_b(j-1,1)e_f(j,i)}{\gamma(j-1,i)}$

$$\gamma(j,i+1) = \gamma(j,i) - \frac{e_b^2(j,i)}{\varepsilon^d_{b_{min}}(j,i)}$$

$$\kappa_b(j,i) = \frac{\delta(j,i)}{\varepsilon^d_{f_{min}}(\boldsymbol{j},\boldsymbol{i})}$$

$$\kappa_f(j,i) = \frac{\delta(j,i)}{\varepsilon^d_{b_{min}}(\boldsymbol{j-1},\boldsymbol{i})}$$

$e_b(j,i+1) = e_b(j-1,i) - \kappa_b(j,i)e_f(j,i)$

$e_f(j,+1) = e_f(j-1,i) - \kappa_f(j,i)e_b(j,i)$

$\varepsilon^d_{b_{min}}(j,i+1) = \varepsilon^d_{b_{min}}(j-1,i) - \delta(j,i)\kappa_b(j,i)$

$\varepsilon^d_{f_{min}}(j,i+1) = \varepsilon^d_{f_{min}}(j,i) - \delta(j,i)\kappa_f(j,i)$

Feedforward Filtering:

$\delta_D(j,i) = \lambda\delta_D(j-1,i) + \frac{e(j,i)e_b(j,i)}{\gamma(j,i)}$

$$\nu_i(j) = \frac{\delta_D(j,i)}{\varepsilon^d_{b_{min}}(j,i)}$$

$$e(j,i+1) = e(j,i) - \nu_i(j)e_b(j,i)$$

end

end

Algorithm 3: RLS Algorithm

2.3 Computational Complexity Analyses

2.3.1 Pre-experimental computation and data collection

In this step, the input parameters of each equalization algorithm are computed offline including the autocorrelation matrix, R of each channel condition and their respective eigenvalues. In addition, the minimum λ_{min}, maximum λ_{msx}, and eigenvalue spreads $\frac{\lambda_{max}}{\lambda_{min}}$ are computed within the MATLAB environment. The results are illustrated in Table 2.

2.3.2 Description of Equalization Algorithms

Each equalization technique is implemented using MATLAB scripts and in-built functions. An outline of the steps involved in each equalization scheme is provided in the algorithmic environments.

3. RESULTS AND DISCUSSION

3.1. Effect of Eigenvalue Spread

The selected parameters for plotting the metrical curves for each algorithm are given in Table 3. For the RLS algorithm, the step size is replaced by a small value of 0.01 which is chosen to initialize the inverse correlation matrix of the RLS algorithm. By relating the eigenvalue spread given in Table II with the convergence speed, it can be concluded that the convergence speed of the LMS decreases with increased eigenvalue spread. Also, the LMS performance with respect to MSE deteriorates with increased eigenvalue spread which is consistent with theory. The RLS algorithm converges faster than the LMS, converging in about 100 samples for all channel conditions as illustrated in Fig. 2. The channel 1 steady-state MSE is the least with a value of 0.00014 while the channel 4 MSE is the largest with a value of 0.0006. This shows that the RLS performance degrades with increased eigenvalue spreads as is the case with the LMS implementation but is less sensitive to the channel eigenvalue spread variations.

Table 2. Minimum, Maximum and Eigenvalue spread for each channel

Index	λ_{min}	λ_{max}	$\frac{\lambda_{max}}{\lambda_{min}}$
Channel 1	0.3329	2.0285	6.0931
Channel 2	0.2126	2.3751	11.1711
Channel 3	0.1245	2.7255	21.8892
Channel 4	0.0646	3.0694	47.4993

Table 3. Parameters for Investigating effect of Eigenvalue spread on LMS.

Step size parameter	0.075
Filter order	11
Number of data samples	1000
Number of experiments	500
SNR in dB	40

3.2. RLS vs LMS for Small SNR Regime

The performance of the RLS and LMS for a small SNR of 10 dB are compared with an LMS step size of 0.075 and an RLS delta value of 0.01 and the plots are shown in Fig. 3. The RLS exhibits lower MSE and faster convergence than the LMS in the low SNR regime

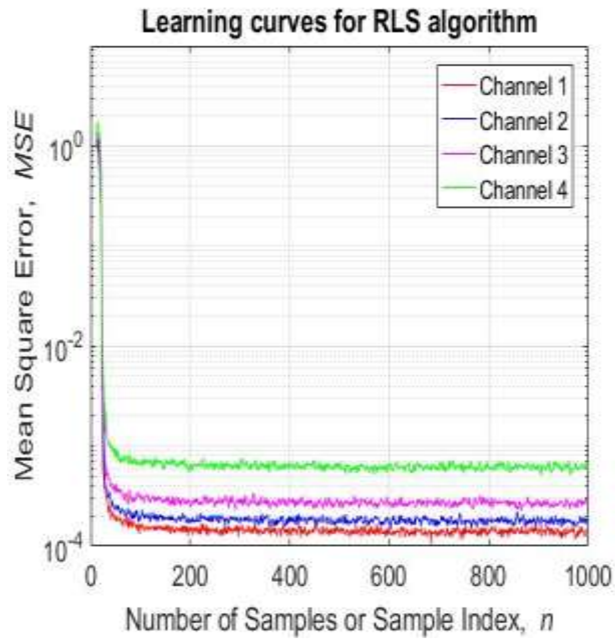

Fig. 2. Learning Curves for RLS under different channel conditions

.

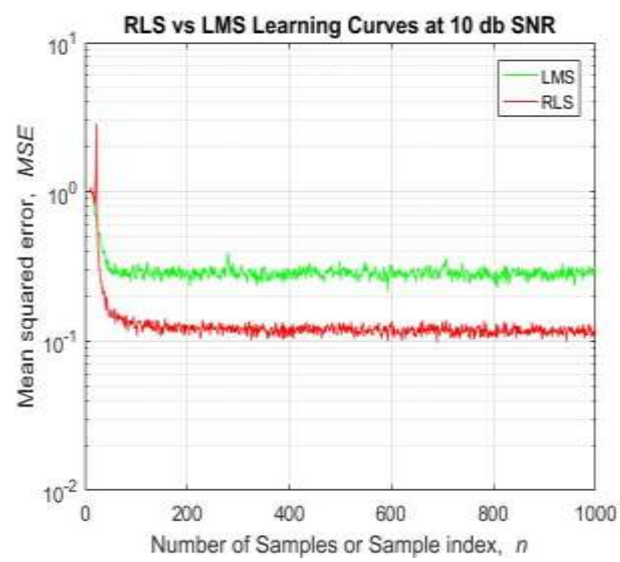

Fig. 3. RLS vs LMS Performance for small SNR regime.

3.3. Steady State Regression Coefficients vs. Tap Weight Filter Coefficients.

In this section, the tap weight coefficients of the LMS/RLS scheme are compared with the steady-state regression coefficients of the RLSL algorithm. Figs. 4 and 5 show that the center regression coefficient the highest magnitude of 1, however compared to the tap weight coefficients of the RLS case, it is not symmetric about the other regression coefficients.

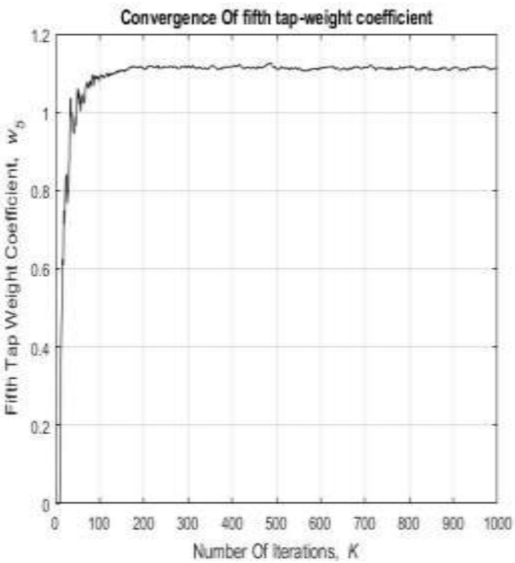

Fig. 4. The convergence of RLS tap weight coefficient for channel 1.

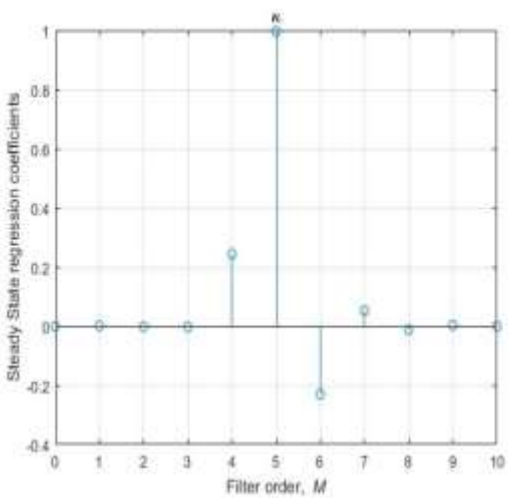

Fig. 5. Steady-state regression coefficients of RLSL for filter order $M = 1$

4. CONCLUSION

The performance and robustness of the LMS, RLS, and RLSL have been investigated in this work. Several insights can be drawn from the complexity analyses: The LMS has the lowest complexity and slowest convergence speed of the three algorithms. It also has the highest steady-state MSE. It is constrained by the step-size parameter which is dependent on the eigenvalue spread under consideration and is therefore very sensitive to eigenvalue spread variations. The RLS has the highest complexity in terms of the order of operations but exhibits the least MSE values and fastest convergence speed.

It is much less sensitive to eigenvalue spread than the LMS and is independent of the step size parameter. The RLSL comes in between the RLS and LMS performance with more computational complexity and lower MSE than the LMS and less computational complexity but higher MSE than the RLS. It converges by an order of magnitude faster than the LMS and converges slower in slightly greater sample time than the RLS. It is also insensitive to eigenvalue spread and step size like the RLS. The choice of which algorithm to use depends on the bias in the system design where computing memory, size of data sets, computational power and the operating SNR range are key factors which should be taken into consideration.

REFERENCES

[1] Solo, V., & Kong, X. (1994). *Adaptive signal processing algorithms: stability and performance*. Prentice-Hall, Inc..

[2] Widrow, B., McCool, J., & Ball, M. (1975). The complex LMS algorithm. *Proceedings of the IEEE*, *63*(4), 719-720.

[3] Ayvaz, E. N., Maraş, M., & Özen, A. (2018). A fast start-up modified LMS algorithm. In *2018 26th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

[4] Niu, Q., & Chen, T. (2018). A new variable step size LMS adaptive algorithm. In *2018 Chinese Control And Decision Conference (CCDC)* (pp. 1-4). IEEE.

[5] Slock, D. T. (1993). On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal Processing*, *41*(9), 2811-2825.

[6] Akhtar, M. T., & Ahmed, S. (2016). A robust normalized variable tap-length normalized fractional LMS algorithm. In *2016 IEEE 59th*

International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 1-4). IEEE.

[7] Sudha, N. S., & Dodda, R. K. (2017). Design of error normalized LMS adaptive filter for EEG signal with eye blink & PLI artefacts. In *2017 International Conference on Trends in Electronics and Informatics (ICEI)* (pp. 54-58). IEEE.

[8] McWhirter, J. G. (1983). Recursive least-squares minimization using a systolic array. In *Real-Time Signal Processing VI* (Vol. 431, pp. 105-113). International Society for Optics and Photonics.

[9] Mahdi, A. S., Abdulameer, L. F., & Morad, A. H. (2018). Comparison between Epsilon Normalized Least Means Square (ϵ-NLMS) and Recursive Least Squares (RLS) Adaptive Algorithms. In *2018 International Conference on Computing Sciences and Engineering (ICCSE)* (pp. 1-5). IEEE.

[10] Li, Z. F., Li, D., & Zhang, J. Q. (2018). A New Penalized Recursive Least Squares Method With a Variable Regularization Factor for Adaptive Sparse Filtering. *IEEE Access*, *6*, 31828-31839.

[11] Lee, D., Morf, M., & Friedlander, B. (1981). Recursive least squares ladder estimation algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *29*(3), 627-641.

[12] Paleologu, C., Enescu, A. A., & Ciochina, S. (2006, December). Recursive least-squares lattice adaptive algorithm suitable for fixed-point implementation. In *2006 13th IEEE International Conference on Electronics, Circuits and Systems* (pp. 1105-1108). IEEE.

[13] Spinelli, H. M., & Merched, R. (2013). The Generalized Sliding-Window recursive least-squares lattice filter. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 5730-5734). IEEE.