Journal of Engineering Science and Technology
Vol. 14, No. 2 (2019) 1043 - 1054

RECURSIVE LEAST SQUARES ALGORITHM FOR ADAPTIVE TRANSVERSAL EQUALIZATION OF LINEAR DISPERSIVE COMMUNICATION CHANNEL

HUSSAIN BIERK*, M. A. ALSAEDI

College of Engineering, Al-Iraqia University, Baghdad, Iraq
*Corresponding Author: hbierk@hotmail.com

Abstract

This paper is intended to analyse the performance, the rate of convergence, selection of proper filter order number, misadjustment, sensitivity to Eigenvalues spread, and computational requirement of the Recursive Least Squares (RLS) algorithm, which is one of the well-known effective algorithms in the field of adaptive filters. Finally, the effect of decision feedback equalizer to improve the system is presented. The RLS algorithm repeatedly minimizes a weighted linear least squares value of the error and therefore, it is known for its excellent high convergence performance in time varying environments. The drawback of this algorithm is the computational complexity and stability problem. The simulation work related to this adaptive filter is performed by MATLAB software.

Keywords: Adaptive equalization, Adaptive signal processing, Adaptive transversal filter, Recursive least squares algorithm.

1. Introduction

The common purpose of an algorithm in the field of adaptive filters is to remove the noise from the corrupted input signal and update the filter weights in order to minimize the mean-square value of the estimating error and the consequent is flat received output signal [1, 2]. In this research, we study the use of very well-known algorithm, which is Recursive Least Squares (RLS) [3-7]. This algorithm has a very fast convergence rate compared with other algorithms such as Least Mean Square (LMS) algorithm [8-12] but it has the limitation in applications that require very large numbers of adaptive filters due to the algorithm computational complexity like echo control and echo cancellation [13, 14].

It is important to know that all RLS based algorithms performance is controlled by a forgetting factor and regularization parameter. Forgetting factor has a substantial influence on the convergence rate, misadjustment, tracking capability, and filter stability of all types of RLS algorithms. The value of this factor is between zero and unity. Ciochina et al. [15] explained that the filter has better stability with low misadjustment but reduced tracking capability when the forgetting factor is close to unity. Ciochina et al. [16] proposed that if the value of forgetting factor is selected very small, both misadjustment and stability of the algorithm will be negatively affected but the tracking capability will be improved. Based on the above aspects, many variable forgetting factor RLS algorithms have been emerged [17-21].

In this study, one set of computer experiments has been carried-out for RLS algorithm to analyse its performance and investigate the impact of filter order, convergence rate, and the sensitivity to Eigenvalue spread. Finally, the effect of decision feedback equalizer to improve system performance is presented. This work is intended to solve the equalizer problem and evaluate the response (the mean-square error or learning curve) of the adaptive filter, using the RLS algorithm, to changes in Eigenvalue spread χ. The learning curve is obtained by averaging the instantaneous squared error e^2(n) versus (n) over K runs. The MSE is computed as follows.

$$MSE(n) = \frac{1}{K} \sum_{k=1}^{K} e_k^2(n) \;\; ; \;\; n=1, 2,......, N \qquad (1)$$

The factor N is the number of data samples$\{a_n\}$ for a single run of the RLS algorithm and K is the number of independent runs required for averaging.

Figure 1 demonstrates the block diagram of the channel equalization problem. The transmitted signal $a(n)$ is a binary data with random values of (+1), and (-1) having zero mean and variance equal to 1. This signal is generated by a random noise generator (1) and transmits through the channel, which has the following impulse response as follows:

$$h(n) = \begin{cases} \frac{1}{2}\left[1 + cos\left\{\frac{2\pi}{W}(n-2)\right\}\right] \\ 0 \qquad otherwise \end{cases} \qquad (2)$$

where (w) is the Eigenvalue spread, which controls the amount of distortion. The white noise $\upsilon(n)$ is generated by random noise generator 2. Both noise generators 1 and 2 are independent of each other. It is worthy to know that $\upsilon(n)$ has a zero mean and variance δ_v^2 = 0.0001 and this gives a 40 dB Signal to Noise Ratio (SNR) at the input of the equalizer. The equalizer has M = 11 taps. The optimum tap weights of the equalizer are symmetric about 6 (midpoint). Since the channel has an impulse response *h* (*n*) that is symmetric about time *n* = 2, and the transferring data starts at (1) then the channel input *a*(*n*) is delayed by Δ *is delayed* samples to provide the desired response for the equalizer.

The input to the ATF (adaptive transversal equalizer) is given by the convolution sum:

$$u(n)= h^T a(n) + \upsilon(n);\ n=1,2,\ldots, N \tag{3}$$

$$u(n)= h_1 a_{n-1} + h_2 a_{n-2} + h_3 a_{n-3} + \upsilon(n) \tag{4}$$

where $a_{n-i} = 0$; $n-i < 0$.

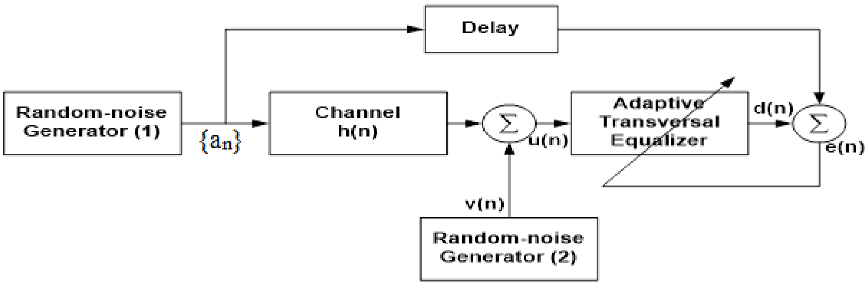

Fig. 1. Block diagram of the typical channel equalization problem.

2. Recursive Least Squares Algorithm

This algorithm can be summarized as follows.

Step 1: Initialize the weight vector and the inverse correlation matrix *w* (0) = 0; $P(0)= \delta^{-1}$ I; where δ is the regularization factor.

Step 2: For *n* = 1, 2, …, *N*, compute the Kalman gain vector (see Eq. (5))

$$K(n) = \frac{P(n-1)\,u(n)}{\lambda+u^T(n)P(n-1)u(n)} \tag{5}$$

where λ is the forgetting factor.

Step 3: Compute the a priori error as follows in Eq. (6).

$$\alpha(n) = d(n) - u^T(n)w(n-1) \tag{6}$$

This scaler parameter is the priori error before the current weight *w(*n) become available. The instantaneous value of *α(n*) differs from that the posteriori error defined in terms of the current weight vector, as follows in Eq. (7).

$$e(n) = d(n) - u^T(n)w(n) \tag{7}$$

However, their mean square values are equal, as following in Eq. (8)

$$E[e^2(n)] = E[\alpha^2(n)]. \tag{8}$$

Step 4: Update the weighting vector as in Eq. (9).

$$w(n+1)= w(n)+K(n)\,\alpha(n) \tag{9}$$

Step 5: Update the matrix (see Eq. (10).

$$\lambda^{-1}[P(n-1) - K(n)\,u^T(n)\,P(n-1)] \tag{10}$$

Step 6: Go back to step (2) until the data is complete, that is, $n = N$.

It is worthy to mention:

- For a filter order of *M*, the dimension of *P(n)* is *M***M*, while *u(n)*, *w(n)*, and *K(n)*, all have the dimension *M**1.
- The standard RLS converges, in the mean square, in about 2M iterations. This is much faster than the LMS algorithm.
- Theoretically, as (*n*) approaches infinity, the RLS produces zero misadjustment when operating in a stationary environment. This is true for high SNR (40 dB in this study) and $\lambda = 1$.

Figure 2 depicts the flow chart for explaining the RLS algorithm.

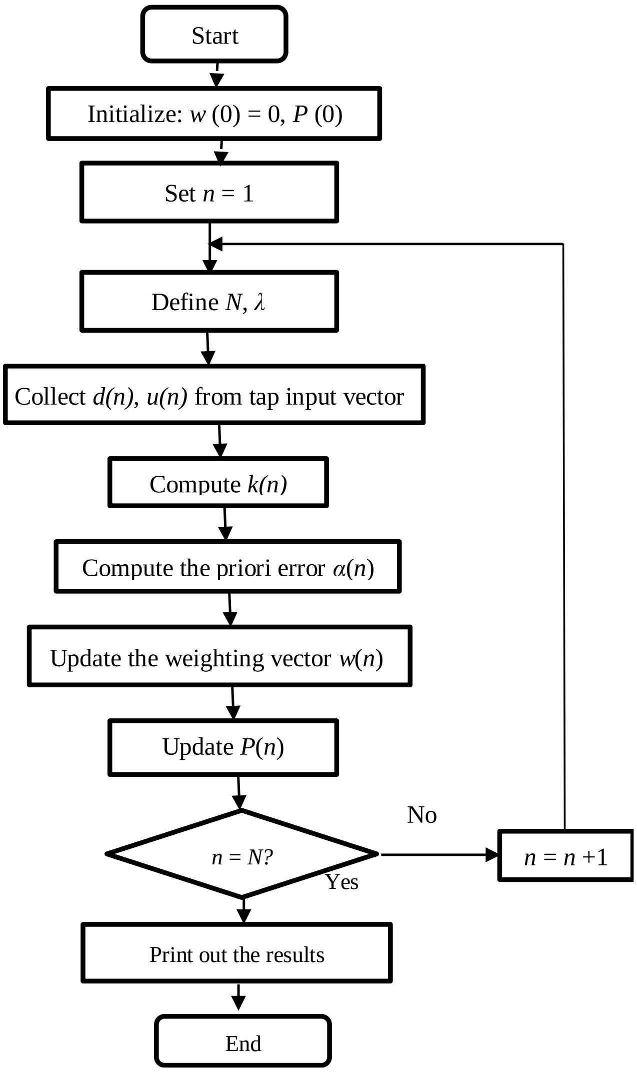

Fig. 2. Flow chart of RLS algorithm.

3. Results and Discussion

In this section, the impact of filter order, effect of eigenvalues spread, and finally the decision feedback equalizer technique are discussed and the results obtained from MATLAB simulation work are analysed.

3.1. The effect of filter order

Basically, in this study, we have selected two filters with 11 and 21 taps for comparison. Table 1 shows the dispersion parameters and eigenvalue spread used for both filters.

Figure 3 depicts the learning curves for two filters (tap M = 11, and M = 21) with the same parameters used for comparison. It is obvious from the figure that ensemble-averaged square error and convergence speed for both filters (trained up to 1000 iterations) are not significantly different. Therefore, it is cost-effective to select the filter size of M = 11 for our study.

Table 1. Parameters used for filter order selection.

W	χ	SNR	δ
3.3	21.7	40 dB	0.01

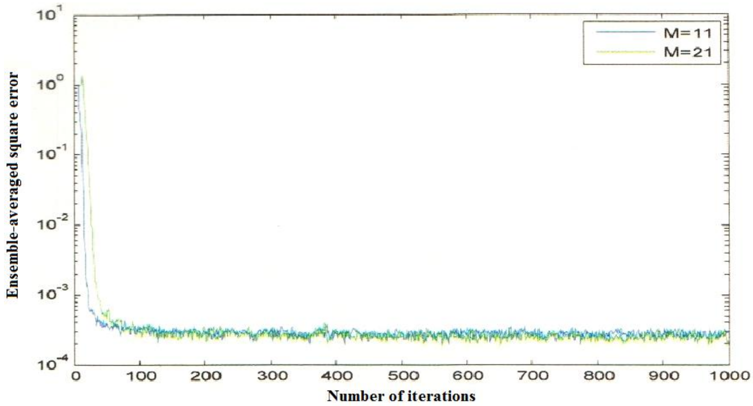

Fig. 3. Learning curves of RLS algorithm for adaptive equalizer with taps M = 11, and M = 21, W = 3.3, and SNR = 40 dB.

3.2. Effect of eigenvalue spread

For selected M = 11, the autocorrelation (R) will be a symmetric matrix of size 11*11 as shown in Fig. 4. It is given that $h(n)$ has nonzero values only for n = l,2,3, so the only nonzero elements in the matrix are $r(0)$, $r(l)$, r(2) on the main diagonal and the four diagonals directly above and below it.

$$R = \begin{bmatrix} r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) \end{bmatrix}$$

Fig. 4. Autocorrelation symmetric matrix of size 11*11.

Now, the procedure to calculate r(O), r(l), r(2), for each value of w is as follows:

We know that $r(k)=E[u(n)u(n-k)]$ (11)

where $k = 0, 1, 2, \ldots, M-1$

Hence, $r(0) = E[u(n)\,u(n)]$; $r(1) = E(u(n)\,u(n-1)]$; $r(2) = E[u(n)\,u(n-2)]$ Substituting the value of $u(n)$ in the Eq. (4), we can calculate $r(0)$, $r(1)$, and $r(3)$ as follows:

$$r(0)= E[h_1 a_{n-1} + h_2 a_{n-2} + h_3 a_{n-3} + \upsilon(n)]^2 \tag{12}$$

But it is evident that:

$$E[a_{n-i} \quad a_{n-k}] = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \tag{13}$$

$$E[\upsilon_{n-i} \quad \upsilon_{n-k}] = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \tag{14}$$

Hence $r(0) = h1^2 + h2^2 + h3^2 + \delta v^2$

Similarly: $r(1) = h_1 h_2 + h_2 h_3$

and $r(2) = h_1 h_3$

Table 2 shows the effect of change of the distortion parameter (w) on elements of autocorrelation matrix (R) and Eigen value spread.

In this part of our study, the regularization factor (δ) is held constant at 0.01 with a fixed signal to noise ratio of 40 dB. Figure 5 depicts the learning curves of the RLS algorithm for four different Eigenvalues spread. It is clear from the figure that the convergence speed is relatively not affected by the variations of Eigenvalues spread. The ensemble-averaged square error is high for higher Eigenvalue spread and vice versa. For example, Eigenvalue spread 46.8216 (corresponds to W = 3.5) has the ensemble-averaged square error of about $6\,e^{-4}$ while for Eigenvalue spread 6.0782 (corresponds to W = 2.9), the ensemble-averaged square error is about $1.4\,e^{-4}$.

Based on studies by Diman et al. [8], it is clear that by using this algorithm both convergence speed and MSE has been improved significantly compared with LMS algorithm and this is consistent with the plot of the equalizer impulse response for different channels as shown in Fig. 6, which indicates also the high efficiency of this algorithm. This figure shows tap-weights for the adaptive equalizer after 1500 iterations through 200 trials for four Eigenvalues spread. It can be seen that the tap weights of the equalizer for all four Eigenvalues spread are symmetric around (6) since the filter order is (11). The value of centre-tap increases with the increase of the Eigenvalue spread and this leads to the conclusion that the change in Eigenvalue affects the impulse response of the filter.

Table 2. Auto-correlation matrix *R*, Eigen value spread $\chi(R)$ for varying channels.

	W **= 2.9**	***W*** **= 3.1**	***W*** **= 3.3**	***W*** **= 3.5**
***r*(0)**	1.09	1.15	1.22	1.30
***r*(1)**	0.43	0.55	0.67	0.77
***r*(2)**	0.04	0.07	0.11	0.15
***r*(*R*)**	6.07	11.12	21.71	46.82

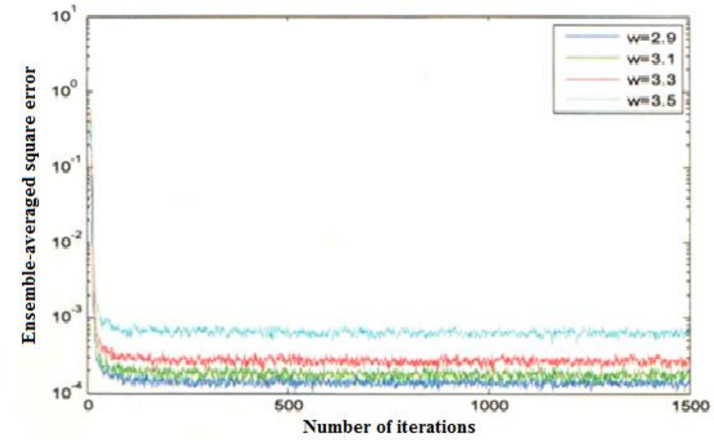

Fig. 5. Learning curves of the RLS algorithm for various channels with δ = 0.01 and SNR = 40 dB.

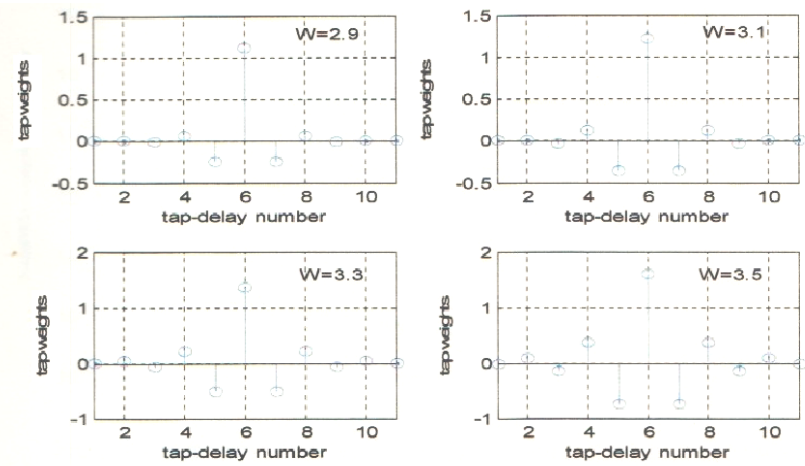

Fig. 6. Equalizer impulse response for different channels using RLS algorithm.

3.3. Effect of decision feedback equalizer (DFE)

The ultimate goal of digital communication is to transmit the data at the highest possible rate. One of the problems in this way is the Intersymbol Interference (ISI) imposed by the communication channel. The ISI is generated by the effect of neighbouring symbols on the current symbol. Decision Feedback Equalizer is one of the proposed methods to tackle this issue. This technique uses old decisions to improve the equalizer performance.

Figure 7 represents the block diagram of DFE, which consists of feedforward ATF that has the observation $u(n)$ as its input and feedback ATF that has the previous decision $\{d(n)\}$ as its input and this decision input are assumed to be correct. The task of the feedback ATF is to filter out the ISI that is produced by previously detected symbols from the predicted symbols. The following equations describe this technique:

We assume w_1 = weighting vector for feed-forward ATF, and w_2 = weighting vector for feedback of ATF, then

$$y(n)= w_1^T u(n); x(n)=w^T_2 d(n) \quad (15)$$

$$d'(n)=y(n)-x(n)=[w_1^T- w_2^T] [u(n) d(n)]^T \quad (16)$$

$$e(n)= d(n)- d'(n) \quad (17)$$

where $d(n)$ is the desirable reference signal, which is equal to $\{a_n\}$ delayed by 6 samples.

Figure 8 demonstrates the reading curves for decision feedback equalizer applying RLS algorithm. In this set of experiments, the feed-forward filter order is selected to be 11, and the feedback filter order is 3 for two different channels. It is obvious that the system is relatively insensitive to the eigenvalue spread variation and the curve of W = 3.5 shows the ensemble-averaged square error slightly less than that with W = 3.3 and this means that channel with W = 3.5 demonstrates better performance than the channel with W = 3.3.

By employing the decision feedback equalizer technique with the RLS algorithm, both convergence speed and MSE have been improved significantly over the system without feedback as demonstrated in Fig. 9. In this figure, the filter of W = 3.3 is selected and learning curves of the equalizer with and without feedback are plotted.

It is very clear that the system with feedback shows better performance in terms of MSE and convergence speed. Comparing the steady-state averaged MSE of both curves, it is obvious that the MSE of the system with feedback becomes about $1.2e^{-7}$ and without feedback is $2e^{-4}$ and this indicates that the system performance has been tremendously improved.

Figure 10 depicts the DFE equalizer impulse response for two different channels (W = 3.3 and W = 3.5) using the RLS algorithm. Here the tap-weights for both forward and feedback filters are combined and the tap (13) has the highest value for both channels.

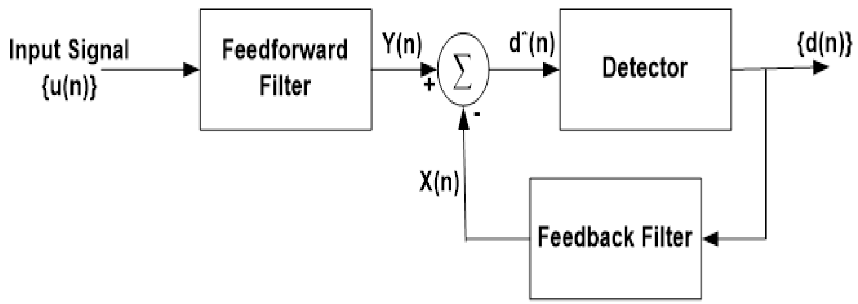

Fig. 7. Block diagram of a DFE.

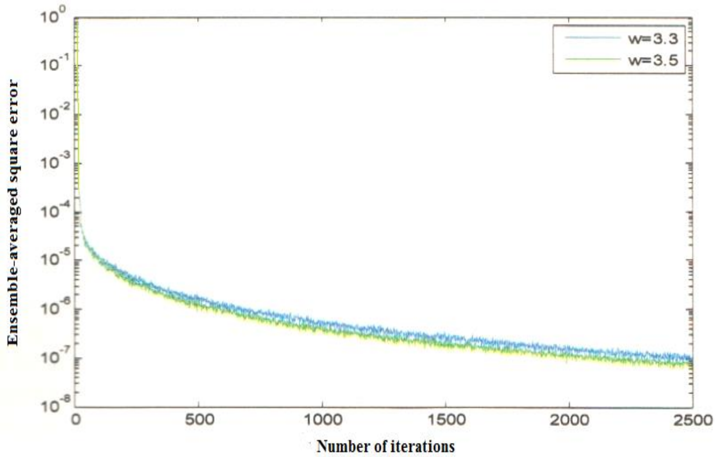

Fig. 8. Learning curves of RLS algorithm for DFE with two different channels.

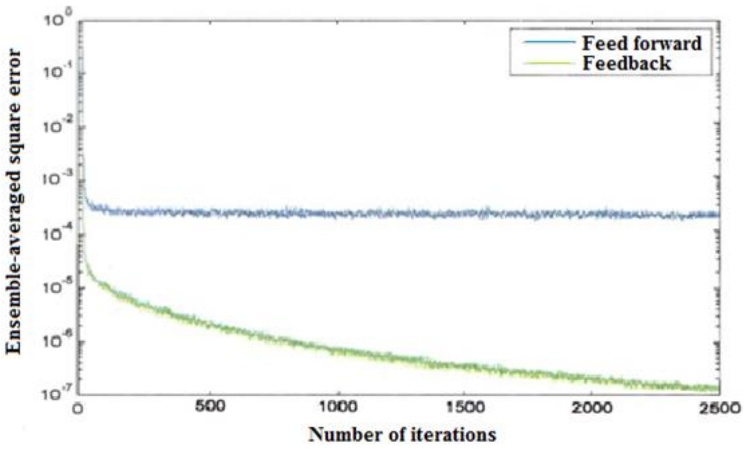

Fig. 9. Comparison of learning curves of RLS algorithm of adaptive equalizer with and without feedback with W = 3.3, with δ= 0.01.

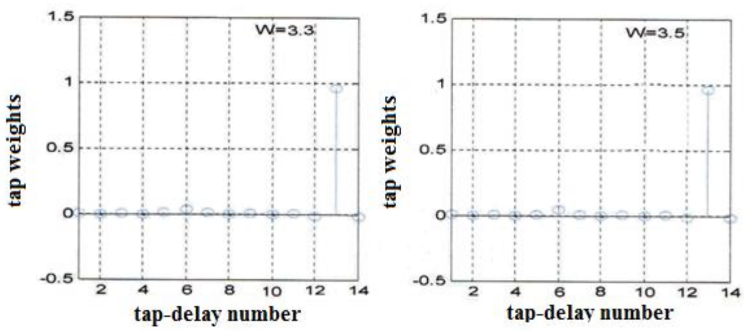

Fig. 10. DFE equalizer impulse response for two channels using RLS algorithm.

3.4. Comparison between RLS algorithm and LMS algorithm

To compare the performance of different algorithms in the field of adaptive linear equalization, it is clear from many studies that the RLS algorithm has a superior performance. Compared to LMS and normalized Least Mean Square (NLMS) algorithms:

- Recursive Least Squares (RLS) algorithm has high initial convergence speed, less steady state MSE, and better SNR improvement [8].
- RLS provides lower steady state MSE than LMS for high SNR (SNR >10) but LMS provides superior performance for very narrow band signals (SNR<10) [22].
- RLS convergence speed is relatively insensitive to the Eigenvalues spread variations, whereas, LMS convergence speed is very sensitive to Eigenvalue spread.
- RLS averaged MSE depends on the Eigenvalues spread variations and decreases as Eigenvalue spread decreases.
- It is worthy to mention here that LMS algorithm is a very simple technique and can be easily implemented, whereas, RLS algorithm is complex in terms of computations and implementation. Moreover, the LMS algorithm has a lower SNR compared with the RLS technique [23].

4. Conclusions

This RLS algorithm is one of the prominent techniques in the field of adaptive filters. This technique demonstrates a very high convergence speed. From the learning curves of RLS, algorithm for adaptive equalizer with taps $M = 11$, and $M = 21$, it is evident that both learning curves show the fast convergence rate but at the cost of a high computational complexity in terms of software implementation compared with some other algorithms that are used in adaptive linear equalization such as (LMS) algorithm. In this study, we also analysed the impact of Eigenvalues spread on RLS algorithm performance and it is clear that the convergence speed is relatively insensitive to the variations of Eigenvalue spreads but the ensemble-averaged square error is significantly influenced so that it is high for higher Eigenvalue spread and vice versa. In this work, the lower filter size has been selected for a cost-effective purpose.

Decision Feed-Back Equalizer (DFE) is suggested to improve both convergence speed and MSE of the filter. The higher convergence speed and better equalizer performance are achieved by employing feedback filter so that the averaged MSE has been reduced more than 1600 times with feedback filter of *W*= 3.3.

Nomenclatures

$e^2(n)$	Instantaneous squared error
$v(n)$	White noise
w	Distortion parameter
χ	Eigenvalue spread

Greek Symbols

δ	Regularization factor
λ	Forgetting factor
$\alpha(n)$	A priori error

Abbreviations

ATF	Adaptive Transversal Filter
DFE	Effect of Decision Feedback Equalizer
ISI	Inter-Symbol Interference
LMS	Least Mean Square
RLS	Recursive Least Squares
SNR	Signal to Noise Ratio

References

1. Hadei, S.A.; and Lotfizad, M. (2003). A family of adaptive filter algorithm in noise cancellation for speech enhancement. *International Journal of Computer and Electrical Engineering*, 2(2), 10 pages.
2. Abhishek, R.; and Ramesh, V. (2015). Acoustic noise cancellation by NLMS and RLS algorithms of adaptive filter. *International Journal of Science and Research*, 4(5), 198-202.
3. Zheng, Y.; and Lin, Z. (2003). Recursive adaptive algorithms for fast and rapidly time-varying systems. *IEEE Transactions on Circuits and Systems II*, 50(9), 602-614.
4. Sayed, A.H. (2008). *Adaptive filters*. Hoboken, New Jersey: Wiley & Sons Inc.
5. Bhotto, M.Z.A.; and Antoniou, A. (2013). New improved recursive least-squares adaptive filtering algorithms. *IEEE Transactions on Circuits and System I*, 60(6), 1548-1558.
6. Liu, L.; Zhang, Y.; and Sun, D. (2017). VFF-norm penalized WL-RLS algorithm using DCD iterations for underwater acoustic communication. *IET Communications*, 11(5), 615-621.
7. Babadi, B.; Kalouptsidis, N.; and Tarokh, V.S. (2015). SPARLS: The sparse RLS algorithm. *IEEE Transactions on Signal Processing*, 58(8), 4013-4025.
8. Dhiman, J.; Shada, Ahmad, S.; and Gulia, K. (2013). Comparison between adaptive filter algorithms (LMS, NLMS and RLS). *International Journal of Science, Engineering and Technology Research*, 2(5),1100-1103.

9. Haykin, S.O. (2002). *Adaptive filter theory* (4th ed.). Upper Saddle River, New Jersey, United States of America: Prentice Hall.
10. Prasad, S.R.; and Patil, S.A. (2016). Implementation of LMS algorithm for system identification. *Proceedings of the Conference on Signal and Information Processing* (IConSIP). Vishnupuri, India, 1-5.
11. Jung, S.M.; Seo, J.-H.; and Park, P. (2015). Variable step-size non-negative normalised least-mean-square-type algorithm. *IET Signal Processing*, 9(8), 618-622.
12. Rao, C.M.; Charles, B.S.; and Prasad, M.N.G. (2013). A variation of LMS algorithm for noise cancellation. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(7), 2838-2843.
13. Hansler, E.; and Schmidt, G. (2004). *Acoustic echo and noise control*: *A practical approach.* Hoboken, New Jersey: John Wiley and Sons, Inc.
14. Paleologu, C.; Benesty, J.; and Ciochina, S. (2010). Sparse adaptive filters for echo cancellation. *Synthesis Lectures on Speech and Audio Processing*, 6(1), 124 pages.
15. Ciochina, S.; Paleologu, C.; Benesty, J.; and Enescu, A.A. (2009). On the influence of the forgetting factor of the RLS adaptive filter in system identification. *Proceedings of the International Symposium on Signals, Circuits and Systems*. Iasi, Romania, 1-4.
16. Ciochina, S.; Paleologu, C.; Benesty, J.; and Enescu, A.A. (2003). On the behaviour of RLS adaptive algorithm in fixed-point implementation. *Proceedings of the International Symposium on Signals, Circuits and Systems.* Iasi, Romania, 57-60.
17. Leung, S.-H.; and So, C.F. (2005). Gradient-based variable forgetting factor RLS algorithm in time-varying environments. *IEEE Transactions on Signal Processing*, 53(8), 3141-3150.
18. Paleologu, C.; Benesty, J.; and Ciochina, S. (2008). A robust variable forgetting factor recursive least-squares algorithm for system identification. *IEEE Signal Processing Letter*, 15, 597-600.
19. Chan, S.C.; and Chu, Y.J. (2012). A new state-regularized QRRLS algorithm with a variable forgetting factor. *IEEE Transactions on Circuits and Systems II*, 59(3) 183-187.
20. Stanciu, C.; Paleologu, C.; Benesty, J.; Ciochina, S.; and Albu, F. (2012). Variable forgetting factor RLS for stereophonic acoustic echo cancellation with widely linear model. *Proceedings of the European Signal Processing Conference (EUSIPCO).* Bucharest, Romani, 1960-1964.
21. Chu, Y.J.; and Chan, S.C. (2015). A new local polynomial modeling-based variable forgetting factor RLS algorithm and its acoustic applications. *IEEE/ACM Transactions Audio, Speech, and Language Processing*, 23(11), 2059-2069.
22. Wei, P.C.; Han, J.; Zeidler, J.R.; and Ku, W.H. (2002). Comparative tracking performance of the LMS and RLS algorithms for chirped narrowband signal recovery. *IEEE Transactions on Signal Processing*, 50(7), 1602-1609.
23. Patel, S.R.; Panchal, S.R.; and Mewada, H. (2014). Comparative study of LMS and RLS algorithms for adaptive filter design with FPGA. *Progress in Science in Engineering Research Journal*, 2(11), 185-192.