LETTER

Data-Reuse Extended NLMS Algorithm Based on Optimized Time-Varying Step-Size for System Identification

Hakan BERCAG[†a)], Osman KUKRER[†], *and* Aykut HOCANIN[†], *Nonmembers*

SUMMARY A new extended normalized least-mean-square (ENLMS) algorithm is proposed. A novel non-linear time-varying step-size (NLTVSS) formula is derived. The convergence rate of ENLMS increases due to NLTVSS as the number of data-reuse L is increased. ENLMS does not involve matrix inversion, and, thus, avoids numerical instability issues.

key words: *adaptive filter, NLMS, step-size, convergence rate, data-reuse*

1. Introduction

System identification (SI) based on adaptive filtering is a significant application which has been motivating researchers to find better methods and solutions to the issues such as convergence rate, misalignment, computational complexity (CC), ease of implementation, and numerical instability.

Least-mean-square (LMS) algorithm is one of the main algorithms which have low computational complexity and ease of implementation [1]–[3] developed and used in SI, but its convergence rate is low. Later, many variants of LMS [4]–[7] such as the LMS-Newton algorithm [8] and NLMS [2], [5], [9] algorithm are developed to improve the convergence rate of LMS. Other algorithms among which recursive least-squares (RLS) algorithm [1], [2], [10] and its variants, affine projection algorithm (APA) and its variants [9]–[11] which have high convergence rates, are developed at the expense of numerical instability problems, and implementation difficulties. This letter introduces the ENLMS algorithm with the following contributions: *i*) The novel NLTVSS method enhances the convergence rate of the proposed ENLMS algorithm as the number of past data L used in the coefficient updating is increased; *ii*) ENLMS does not use matrix inversion, and, thus, eliminates numerical instability problems; *iii*) ENLMS is easy to implement with only one parameter L which is simple to tune.

The problem considered in this letter is SI, and the unknown system is considered to be an N-tap finite impulse response (FIR) model. The adaptive filter estimates the unknown impulse response (IR) coefficients $\mathbf{h} = [h_1, h_2, \ldots, h_N]^T$ via tap-weights $\mathbf{w}(k) = [w_0(k), w_1(k), \ldots, w_{N-1}(k)]^T$, where N is the filter length, $[\cdot]^T$ denotes transposition operator, and k is the discrete-time index. The plant output, i.e, the desired response is $d(k) = \mathbf{h}^T\mathbf{x}(k)+v(k)$, where $\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T$ is the tap-input vector, and $v(k)$ is the additive measurement noise. The filter output is $y(k) = \mathbf{x}^T(k)\mathbf{w}(k-1)$, and the estimation error is $e(k) = d(k) - y(k)$.

2. Derivation of ENLMS

The Wiener-Hopf equation [1] leads to the optimum tap-weight vector $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$ for the FIR coefficients. Recursive inverse (RI) algorithm [12], [13] is developed to obtain the Wiener solution by iterative convergence of the weight update equation:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)\left[\mathbf{p}(k) - \mathbf{R}(k)\mathbf{w}(k-1)\right], \tag{1}$$

where $\mathbf{R}(k)$ and $\mathbf{p}(k)$ are the autocorrelation matrix and cross-correlation vector, respectively. The RI algorithm attains the Wiener solution $\mathbf{w}_o$, by recursive inversion and simultaneous recursive estimation of the autocorrelation, thus, not requiring the inversion of the autocorrelation matrix. In this way, numerical instability problems due to loss of Hermitian symmetry, and loss of positive definiteness of the inverse autocorrelation are eliminated. Using (1), the *a-priori* and the *a-posteriori* residuals of the equation $\mathbf{R}(k)\mathbf{w}(k) = \mathbf{p}(k)$ are denoted, respectively, as:

$$\boldsymbol{\xi}(k, k-1) = \mathbf{p}(k) - \mathbf{R}(k)\mathbf{w}(k-1), \tag{2}$$

$$\boldsymbol{\xi}(k, k) = \mathbf{p}(k) - \mathbf{R}(k)\mathbf{w}(k) = [\mathbf{I} - \mu(k)\mathbf{R}(k)]\boldsymbol{\xi}(k, k-1). \tag{3}$$

In order to minimize the norm of (3), step-size is chosen as:

$$\mu(k) = \arg\min_{\mu} \|\boldsymbol{\xi}(k, k, \mu)\|_2^2, \tag{4}$$

where the squared Euclidean norm $\|\boldsymbol{\xi}(k, k, \mu)\|_2^2$ is the cost function. Minimizing the norm of (3) corresponds to increasing the convergence rate. Expanding $\|\boldsymbol{\xi}(k, k, \mu)\|_2^2$, evaluating its gradient w.r.t. μ, setting it to zero, and solving for the scalar μ, the optimal NLTVSS formula minimizing $\|\boldsymbol{\xi}(k, k, \mu)\|_2^2$ is obtained as:

$$\mu_{NL}(k) = \frac{\boldsymbol{\xi}^T(k, k-1)\mathbf{R}(k)\boldsymbol{\xi}(k, k-1)}{\boldsymbol{\xi}^T(k, k-1)\mathbf{R}^2(k)\boldsymbol{\xi}(k, k-1)}. \tag{5}$$

In the RI algorithm, the autocorrelation matrix and the cross-correlation vector are updated as:

$$\mathbf{R}(k) = \beta\mathbf{R}(k-1) + \mathbf{x}(k)\mathbf{x}^T(k), \tag{6}$$

Manuscript received August 3, 2023.
Manuscript revised November 4, 2023.
Manuscript publicized January 11, 2024.

[†]Dept. of Electrical and Electronics Engineering, Eastern Mediterranean University, 99628, Famagusta, North Cyprus, via Mersin 10, Turkey.
a) E-mail: hakan.bercag@emu.edu.tr

DOI: 10.1587/transfun.2023EAL2072

$$\mathbf{p}(k) = \beta\mathbf{p}(k-1) + d(k)\mathbf{x}(k), \tag{7}$$

where β is the forgetting factor. The last L input signal vectors are collected in an $N \times L$ matrix; $\mathbf{X}(k) = [\mathbf{x}(k)\mathbf{x}(k-1)\cdots\mathbf{x}(k-L+1)]$. The solutions of (6) and (7) are:

$$\mathbf{R}(k) = \sum_{i=0}^{k} \beta^{k-i}\mathbf{x}(i)\mathbf{x}^T(i), \tag{8}$$

$$\mathbf{p}(k) = \sum_{i=0}^{k} \beta^{k-i} d(i)\mathbf{x}(i). \tag{9}$$

Substitution of (8) and (9) into (2) yields:

$$\boldsymbol{\xi}(k,k-1) = \sum_{i=0}^{k} \beta^{k-i}\mathbf{x}(i)[d(i)-\mathbf{x}^T(i)\mathbf{w}(k-1)]. \tag{10}$$

Instead of (8) and (9), $\mathbf{R}(k)$ and $\mathbf{p}(k)$ are estimated as follows:

$$\mathbf{R}(k) = \frac{1}{L}\sum_{i=k-L+1}^{k} \mathbf{x}(i)\mathbf{x}^T(i), \tag{11}$$

$$\mathbf{p}(k) = \frac{1}{L}\sum_{i=k-L+1}^{k} d(i)\mathbf{x}(i). \tag{12}$$

Define the *a-priori* errors as:

$$e(i,k-1) = d(i) - \mathbf{x}^T(i)\mathbf{w}(k-1), \quad \forall i = k-L+1 : k \tag{13}$$

Then, (10) as a function of (13) becomes:

$$\boldsymbol{\xi}(k,k-1) = \sum_{i=0}^{k} \beta^{k-i} e(i,k-1)\mathbf{x}(i). \tag{14}$$

Substituting (11) and (12) into (2), the *a-priori* residual used instead of (14), and averaged over the last L errors in (13), is:

$$\boldsymbol{\xi}(k,k-1) = \frac{1}{L}\sum_{i=k-L+1}^{k} e(i,k-1)\mathbf{x}(i). \tag{15}$$

A new vector for reducing CC of $\mu_{NL}(k)$ in (5) is defined as:

$$\mathbf{R}(k)\boldsymbol{\xi}(k,k-1) = \mathbf{z}(k) = \frac{1}{L}\sum_{i=k-L+1}^{k} (\mathbf{x}^T(i)\boldsymbol{\xi}(k,k-1))\mathbf{x}(i). \tag{16}$$

The number of multiplications of RHS of (16) is $2LN$ which is less than N^2 multiplications of LHS of (16). Substituting (16) into (5), NLTVSS with reduced CC is expressed as:

$$\mu_{NL}(k) = \frac{\boldsymbol{\xi}^T(k,k-1)\mathbf{z}(k)}{\|\mathbf{z}(k)\|_2^2}. \tag{17}$$

Substituting (15) for RHS of (2), and (17) into (1) yields the ENLMS algorithm weight update equation:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu_{NL}(k)\boldsymbol{\xi}(k,k-1). \tag{18}$$

It can be deduced that for the special case $L = 1$:

$$\mathbf{z}(k) = \|\mathbf{x}(k)\|_2^2 e(k,k-1)\mathbf{x}(k). \tag{19}$$

Using (19), an expression for the NLTVSS $\mu_{NL}(k)$ is obtained as $\mu_{NL}(k) = 1/\|\mathbf{x}(k)\|_2^2$ that is used in (18) to reveal the ENLMS weight update for $L = 1$, which is also the NLMS weight update:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu_o\frac{e(k,k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|_2^2}, \tag{20}$$

where μ_o is included to control the convergence rate and misalignment.

Algorithm 1 Pseudo-code of ENLMS

```
Given: $N, L$
Initialization: $\mathbf{w}_0$=zeros($N$, 1);  $\mathbf{X}(k)$=zeros($N$, $L$)
for $k = N, N+1, ..., k_{max}$ do
    Initialize: $\xi(k, k-1)$=zeros($N$, 1); $\mathbf{z}(k)$=zeros($N$, 1)
    $\mathbf{x}(k) = [x(k), x(k-1), \cdots, x(k-N+1)]^T$
    $d(k) = \mathbf{h}^T\mathbf{x}(k) + v(k)$
    $\mathbf{X}(k) = [\mathbf{x}(k), \mathbf{x}(k-1), \cdots, \mathbf{x}(k-L+1)]$
    $e(i, k-1) = d(i) - \mathbf{x}^T(i)\mathbf{w}(k-1), \quad \forall i = k-L+1 : k$
    $\xi(k, k-1) = \frac{1}{L}\sum_{i=k-L+1}^{k} e(i, k-1)\mathbf{x}(i)$
    $\mathbf{z}(k) = \frac{1}{L}\sum_{i=k-L+1}^{k} \mathbf{x}^T(i)\boldsymbol{\xi}(k, k-1)\mathbf{x}(i)$
    $\mu_{NL}(k) = \frac{\boldsymbol{\xi}^T(k,k-1)\mathbf{z}(k)}{\|\mathbf{z}(k)\|_2^2}$
    $\mathbf{w}(k) = \mathbf{w}(k-1) + \mu_{NL}(k)\boldsymbol{\xi}(k, k-1)$
end for
```

2.1 Stability Analysis of ENLMS

Define optimal error as $e_o(i) = d(i) - \mathbf{x}^T(i)\mathbf{w}_o$, and $\boldsymbol{\xi}(k,0)$ as:

$$\boldsymbol{\xi}(k,0) = \mathbf{p}(k) - \mathbf{R}(k)\mathbf{w}_o = \frac{1}{L}\sum_{i=k-L+1}^{k} e_o(i)\mathbf{x}(i). \tag{21}$$

Using the residual (21), and substituting (1) into weight-error vector defined as $\boldsymbol{\varepsilon}(k) = \mathbf{w}(k) - \mathbf{w}_o$ [1], deduces:

$$\boldsymbol{\varepsilon}(k) = [\mathbf{I} - \mu(k)\mathbf{R}(k)]\boldsymbol{\varepsilon}(k-1) + \mu(k)\boldsymbol{\xi}(k,0). \tag{22}$$

The autocorrelation estimate can be expressed in terms of its eigenvalue decomposition as $\mathbf{R}(k) = \mathbf{E}(k)\boldsymbol{\Lambda}(k)\mathbf{E}^T(k)$, where $\boldsymbol{\Lambda}(k) = \text{diag}\{\lambda_i\}, \forall i = 1, 2, \ldots, N$, is a diagonal matrix of eigenvalues, and $\mathbf{E}(k)$ is a matrix, columns of which are the eigenvectors of $\mathbf{R}(k)$. Using this decomposition, (22) can be transformed into:

$$\tilde{\boldsymbol{\varepsilon}}(k) = [\mathbf{I} - \mu(k)\boldsymbol{\Lambda}(k)]\tilde{\boldsymbol{\varepsilon}}(k-1) + \mu(k)\tilde{\boldsymbol{\xi}}(k,0), \tag{23}$$

where $\tilde{\boldsymbol{\varepsilon}}(k) = \mathbf{E}^T(k)\boldsymbol{\varepsilon}(k)$ and $\tilde{\boldsymbol{\xi}}(k,0) = \mathbf{E}^T(k)\boldsymbol{\xi}(k,0)$. The squared norm of $\tilde{\boldsymbol{\xi}}(k,0)$ is given by:

$$\|\tilde{\boldsymbol{\xi}}(k,0)\|_2^2 = \frac{1}{L^2}\sum_{i=k-L+1}^{k}\sum_{j=k-L+1}^{k} e_o(i)e_o(j)\mathbf{x}^T(i)\mathbf{x}(j), \tag{24}$$

which, based on the orthogonality of $e_o(i)$ and $\mathbf{x}(i)$, yields:

$$E\{\|\tilde{\boldsymbol{\xi}}(k,0)\|_2^2\} = \frac{1}{L^2}\sum_{i=k-L+1}^{k} E\{e_o^2(i)\|\mathbf{x}(i)\|_2^2\} \tag{25a}$$

$$= \frac{1}{L}\sigma_o^2 E\{\|\mathbf{x}(i)\|_2^2\} = \frac{N}{L}\sigma_o^2\sigma_x^2, \tag{25b}$$

where σ_o^2 is the noise variance, σ_x^2 is the input signal variance, and $E\{\cdot\}$ is mathematical expectation. The squared norm of (23) is:

$$\|\tilde{\boldsymbol{\varepsilon}}(k)\|_2^2 = \sum_{n=1}^{N}(1-\mu(k)\lambda_n(k))^2\tilde{\varepsilon}_n^2(k-1)+\mu^2(k)\|\tilde{\boldsymbol{\xi}}(k,0)\|_2^2 \\ +2\mu(k)[\tilde{\boldsymbol{\varepsilon}}^T(k-1)\{\mathbf{I}-\mu(k)\boldsymbol{\Lambda}(k)\}\tilde{\boldsymbol{\xi}}(k,0)]. \tag{26}$$

Denote $\tilde{\mathbf{z}}(k) = \mathbf{E}^T(k)\boldsymbol{\xi}(k,k-1)$, and $\vartheta_i(k) = \frac{\lambda_i(k)}{\lambda_{min}(k)}, \forall i = 1,\ldots,N$. Noticing $1 \leq \vartheta_i(k) \leq \vartheta_{max}(k)$, where $\vartheta_{max}(k) = \frac{\lambda_{max}(k)}{\lambda_{min}(k)}$ is the eigenvalue spread, the NLTVSS $\mu_{NL}(k)$ in (5) is reevaluated as:

$$\mu_{NL}(k) = \frac{\sum_{i=1}^{N}\lambda_i(k)\tilde{z}_i^2(k)}{\sum_{i=1}^{N}\lambda_i^2(k)\tilde{z}_i^2(k)} = \frac{\lambda_{min}(k)\sum_{i=1}^{N}\vartheta_i(k)\tilde{z}_i^2(k)}{\lambda_{min}^2(k)\sum_{i=1}^{N}\vartheta_i^2(k)\tilde{z}_i^2(k)}, \tag{27}$$

from which, the following is deduced:

$$\mu_{NL}(k)\lambda_{min}(k) = \frac{\sum_{i=1}^{N}\vartheta_i(k)\tilde{z}_i^2(k)}{\sum_{i=1}^{N}\vartheta_i^2(k)\tilde{z}_i^2(k)} \geq \frac{1}{\vartheta_{max}(k)}. \tag{28}$$

Defining $\lambda_a(k) \approx \lambda_i(k)$ as the average of the eigenvalues of $\mathbf{R}(k)$ for the case in which the eigenvalue spread $\vartheta_{max}(k)$ is not very large, NLTVSS in (27) can be approximated as $\mu_{NL}(k) \simeq \frac{1}{\lambda_a(k)}$. In (26), replacing $\lambda_n(k)$ by $\lambda_{min}(k)$, and substituting (28) for $\mu_{NL}(k)\lambda_{min}(k)$ results in the following:

$$\|\tilde{\boldsymbol{\varepsilon}}(k)\|_2^2 \leq \left(1-\frac{1}{\vartheta_{max}(k)}\right)^2\|\tilde{\boldsymbol{\varepsilon}}(k-1)\|_2^2 + \mu_{NL}^2(k)\|\tilde{\boldsymbol{\xi}}(k,0)\|_2^2 \\ +2\mu_{NL}(k)[\tilde{\boldsymbol{\varepsilon}}^T(k-1)\{\mathbf{I}-\mu_{NL}(k)\boldsymbol{\Lambda}(k)\}\tilde{\boldsymbol{\xi}}(k,0)]. \tag{29}$$

The time-varying eigenvalue spread $\vartheta_{max}(k)$ fluctuates around the theoretical eigenvalue spread ρ of the covariance matrix, and approaches it as time increases. Thus, $\vartheta_{max}(k)$ is replaced by ρ, while the expectation of (29) is taken to determine the mean steady-state behaviour of the error norm. Note that $E[\tilde{\boldsymbol{\varepsilon}}^T(k-1)] \simeq 0$, and since the terms $\tilde{\boldsymbol{\varepsilon}}^T(k-1)$ and $[\mathbf{I}-\mu_{NL}(k)\boldsymbol{\Lambda}(k)]\tilde{\boldsymbol{\xi}}(k,0)$ are independent, $E\{\tilde{\boldsymbol{\varepsilon}}^T(k-1)[\mathbf{I}-\mu_{NL}(k)\boldsymbol{\Lambda}(k)]\tilde{\boldsymbol{\xi}}(k,0)\} \simeq 0$. Denoting $\upsilon(k) = E\left[\|\tilde{\boldsymbol{\varepsilon}}(k)\|_2^2\right]$ and taking the expected value of (29) yields:

$$\upsilon(k) \leq \left(1-\frac{1}{\rho}\right)^2\upsilon(k-1)+E\{\mu_{NL}^2(k)\|\tilde{\boldsymbol{\xi}}(k,0)\|_2^2\}. \tag{30}$$

The expected value of (11) is $E\left[\mathbf{R}(k)\right] = E\{\mathbf{x}(i)\mathbf{x}^T(i)\} = \mathbf{R}_x$, the theoretical autocorrelation. The average eigenvalue of $\mathbf{R}(k)$ can be expressed as $\lambda_a(k) \simeq \lambda_{x,a}$, where $\lambda_{x,a}$ is the average eigenvalue of $\mathbf{R}_x$. Equating $\mathrm{tr}(\mathbf{R}_x) = N\lambda_{x,a}$ and $\mathrm{tr}(\mathbf{R}_x) = E\left[\|\mathbf{x}(i)\|_2^2\right] = N\sigma_x^2$, leads to $\lambda_{x,a} = \sigma_x^2$, which is substituted with $\lambda_a(k) \simeq \lambda_{x,a}$, $\mu_{NL}(k) \simeq \frac{1}{\lambda_a(k)}$, and (25b) into (30) to reveal:

$$\upsilon(k) \leq \left(1-\frac{1}{\rho}\right)^2\upsilon(k-1)+\frac{N\sigma_o^2}{L\sigma_x^2}. \tag{31}$$

It can be deduced from (31) that the steady-state weight error vector norm is upper bounded as follows:

$$\upsilon(\infty) \leq \left(\frac{\rho^2}{2\rho-1}\right)\frac{N\sigma_o^2}{L\sigma_x^2}. \tag{32}$$

The upper bound of $\upsilon(\infty)$ increases as ρ increases. Yet, increasing L will decrease the upper bound of $\upsilon(\infty)$, and compensate the increase due to ρ, unraveling NLTVSS's capability of improving the performance of ENLMS. The inequalities (31) and (32) show that, for all finite values of ρ, the weight error vector norm is stable.

2.1.1 Experimental Results

The performance of the proposed ENLMS algorithm is evaluated, and compared with that of the APA, RLS and NLMS algorithms by executing five simulations, parameters of which are given in the captions of Figs. 1–5. The total multiplications per iteration of the algorithms are as follows: ENLMS: $(4L+3)N$, APA: $(L^2+2L)N+L^3+L$ [9], RLS: $2N^2+2N$ [14], NLMS: $2N+3$ [9]. The performance metric used in the SI application is mean-square deviation (MSD), in dB, defined as $10\log_{10}(E[\|\mathbf{w}_o-\mathbf{w}(k)\|_2^2])$. Each simulation is averaged over 500 Monte Carlo ensembles. The input signal used in the simulations 1 through 4 is generated from the AR(4) process: $x(k) = 1.79x(k-1) - 1.85x(k-2) + 1.27x(k-3) - 0.41x(k-4) + \upsilon_o(k)$ where $\upsilon_o(k)$ is white Gaussian noise (WGN) with $\mathcal{N}(0, 0.1481)$. The measurement noise is zero-mean additive WGN with variance $\sigma_m^2 = 10^{-3}$. The signal-to-noise ratio SNR defined as $10\log_{10}(E[x^2(k)]/E[m^2(k)] = \sigma_x^2/\sigma_m^2)$ is 30 dB in all simulations. The symmetric IR used in all simulations has length $N = 65$. In simulations 1 through 4, the computational complexities in terms of the number of multiplications of ENLMS and APA are set to almost the same values for fair comparison. All compared algorithms are set to initially converge to nearly the same misalignment in each simulation.

The first simulation (Fig. 1) shows the convergence rate performance of ENLMS at $L = 3$ as compared with other algorithms. ENLMS converges by ~ 2500 steps faster than NLMS, but slower than RLS and APA. In the second simulation (Fig. 2), ENLMS can be observed to converge faster than its counterpart algorithms APA and NLMS by ~ 1000 and

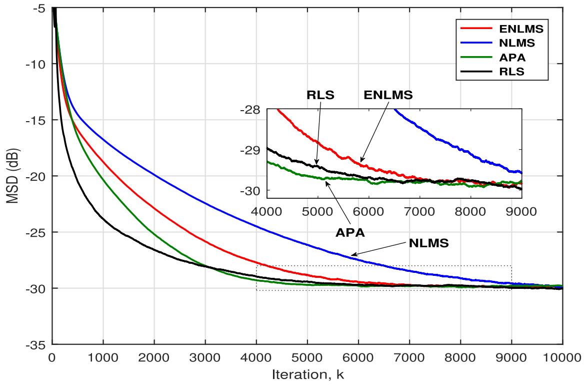

Fig. 1 MSD convergence rate: ENLMS with L=3; APA with L=3, μ_o = 0.125; NLMS with μ_o = 0.9; RLS with δ = 0.001, λ = 0.9995.

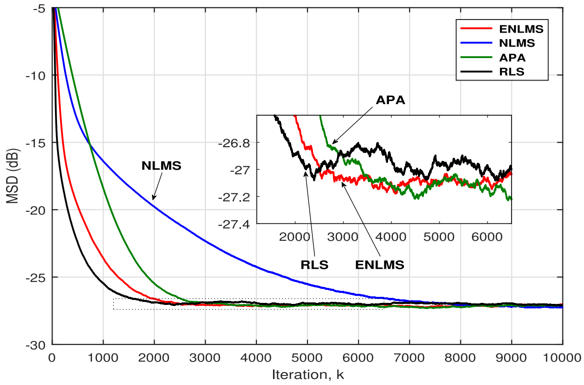

Fig. 2 MSD convergence rate: ENLMS with L=12; APA with L=6, μ_o = 0.0212; NLMS with μ_o = 1.24; RLS with δ = 4.7, λ = 0.999.

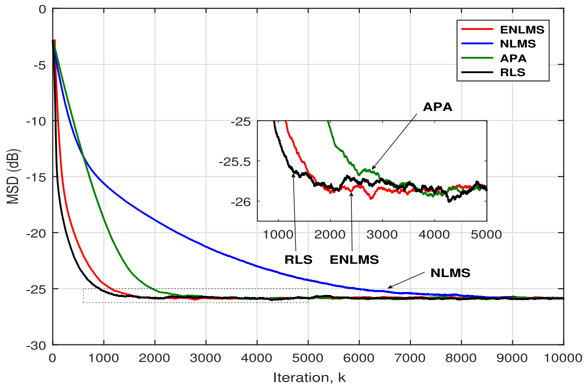

Fig. 3 MSD convergence rate: ENLMS with L=21; APA with L=8, μ_o = 0.016; NLMS with μ_o = 1.38; RLS with δ = 3.9, λ = 0.9987.

~ 5500 steps, respectively, even though RLS still converges faster than ENLMS. This performance enhancement occurs when the number of data-reuse of ENLMS is increased to $L = 12$. In the third simulation (Fig. 3), it can be observed that the convergence rate of ENLMS is approximately the same as that of RLS. Although RLS initially converges fast in the transient state, it starts slowing close to the steady-state convergence, whereas ENLMS initially converges slow in the transient state, but converges fast as it approaches the steady-state. Consequently, ENLMS and RLS converge with nearly the same speeds at the beginning of the steady-state conditions. However, when the number of data-reuse of ENLMS is $L = 21$, ENLMS is superior to RLS in terms of computational complexity, since the number of multiplications of ENLMS is 5655, which is significantly less than 8580 multiplications of RLS. In general, for $L < N/2$, the number of multiplications of ENLMS is less than that of RLS. This simulation shows that when L is sufficiently increased and $L < N/2$, ENLMS exhibits close convergence performance to that of RLS, and operates with less computational complexity than RLS. In addition, ENLMS can be observed to converge faster than its counterpart algorithms APA and NLMS in this setting, as well. In the fourth simulation (Fig. 4), when the number of data-reuse of ENLMS is drastically increased to $L = 33$, ENLMS can be observed to converge to steady-state at $k \simeq 1500$, which is ~ 1200 and ~ 6700 steps faster than APA and NLMS, respectively. The performance of ENLMS at increased $L = 33$ is close to that of the benchmark RLS algorithm. In the fifth simulation (Fig. 5), the ENLMS and RLS algorithms are compared according to varying eigenvalue spreads. When the input signal used is AR(4) process: $x(k) = 1.352x(k-1) - 1.338x(k-2) + 0.662x(k-3) - 0.24x(k-4) + v_o(k)$, the eigenvalue

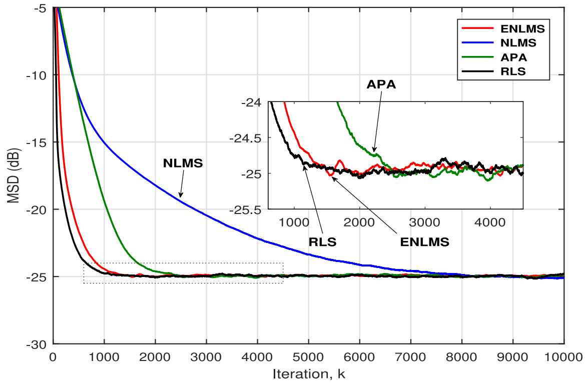

Fig. 4 MSD convergence rate: ENLMS with L=33; APA with L=10, μ_o = 0.136; NLMS with μ_o = 1.45; RLS with δ = 3.2, λ = 0.9984.

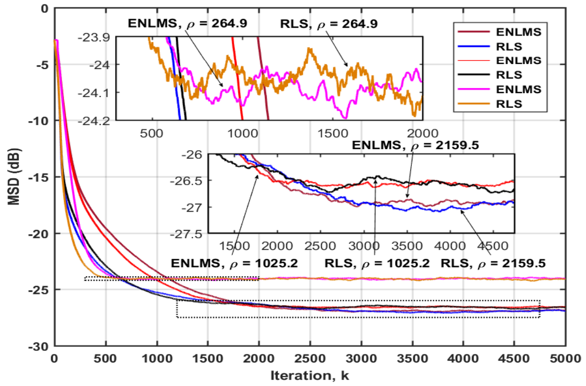

Fig. 5 MSD convergence rate: ENLMS with L=15; ρ = 264.9, ρ = 1025.2, ρ = 2159.5; RLS with δ = 2.6, λ = 0.999 for ρ = 2159.5, δ = 3.9, λ = 0.9989 for ρ = 1025.2, δ = 3.0, λ = 0.9965 for ρ = 264.9.

spread $\rho = 264.9$. When the input signal used is the AR(4) process: $x(k) = 1.79x(k-1) - 1.85x(k-2) + 1.27x(k-3) - 0.41x(k-4) + v_o(k)$, the eigenvalue spread $\rho = 1025.2$, and when the input signal used is AR(4) process: $x(k) = 1.70x(k-1) - 1.95x(k-2) + 1.27x(k-3) - 0.41x(k-4) + v_o(k)$, the eigenvalue spread $\rho = 2159.5$. In all cases, $v_o(k)$ is white Gaussian noise (WGN) with $\mathcal{N}(0, 0.1481)$, the measurement noise is zero-mean additive WGN with variance $\sigma_m^2 = 10^{-3}$, and the SNR is 30 dB. The simulation shows that at the relatively low eigenvalue spread $\rho = 264.9$, the convergence speed of RLS is slightly faster than that of ENLMS. Yet, it can be observed that the convergence rate of ENLMS approaches that of RLS as the eigenvalue spread ρ is increased. It can be observed in Fig. 5 that, the convergence rate of ENLMS at $\rho = 1025.2$ becomes equivalent to that of RLS, and slightly higher at $\rho = 2159.5$ than that of RLS. Note that, for fair comparison of convergence rates, both methods are made to converge to almost the same misalignment as the eigenvalue spread is changed.

Overall, it can be observed that, as L is increased, the convergence rate of ENLMS becomes enhanced, however, with increased misalignment which also occurs with the other algorithms in comparison. Additionally, the simulations show that, as L is increased, the relative convergence rate difference between ENLMS and the compared algorithms alters in favour of ENLMS.

3. Conclusion

A new extended NLMS algorithm based on the data reuse of the tap-input and of the desired response is proposed. A novel non-linear time-varying step-size formula is derived. The new ENLMS algorithm is analyzed theoretically and shown to be stable. The convergence speed of ENLMS is illustrated by simulations to become faster than that of its counterpart algorithms, and comparable to that of RLS, as the number of data-reuse is increased, when the impulse response is symmetric as in channel equalization applications.

References

[1] S. Haykin, Adaptive Filter Theory, 4th ed., Prentice Hall, Upper Saddle River, NJ, U.S.A., 2002.

[2] M.H. Hayes, Statistical Digital Signal Processing and Modeling, 1st ed., John Wiley & Sons, 111 River Street, Hoboken, NJ, U.S.A., 1996.

[3] E. Bjarnason, “Analysis of the filtered-X LMS algorithm,” IEEE Trans. Speech Audio Process., vol.3, no.6, pp.504–514, 1995.

[4] D. Feng, Z. Bao, and L. Jiao, “Total least mean squares algorithm,” IEEE Trans. Signal Process., vol.46, no.8, pp.2122–2130, 1998.

[5] D. Feng, Z. Bao, and L.C. Jiao, “Interference normalized least mean square algorithm,” IEEE Signal Process. Lett., vol.14, no.12, pp.988–991, 2007.

[6] T.K. Wao, Z. Bao, and L.C. Jiao, “Fast hierarchical least mean square algorithm,” IEEE Signal Process. Lett., vol.8, no.11, pp.289–291, 2001.

[7] S.S. Bhattacharjee, D. Ray, and N.V. George, “Adaptive modified versoria zero attraction least mean square algorithms,” IEEE Trans. Circuits. II, Exp. Briefs, vol.67, no.12, pp.3602–3606, 2020.

[8] B. Widrow and S. Stearns, Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1985.

[9] F. Huang and J. Zhang, “A family of gain-combined proportionate adaptive filtering algorithms for sparse system identification,” Digital Signal Processing, vol.70, no.6, pp.49–58, 2017.

[10] P.S.R. Diniz, Adaptive Filtering, Algorithms and Practical Implementation, 4th ed., Springer, Springer Science and Business Media, New York, U.S.A., 2013.

[11] H. Zhao, B. Liu, and P. Song, “Variable step-size affine projection maximum correntropy criterion adaptive filter with correntropy induced metric for sparse system identification,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol.67, no.11, pp.2782–2786, 2020.

[12] M.S. Ahmad, O. Kukrer, and A. Hocanin, “Recursive inverse adaptive filtering algorithm,” Digital Signal Processing, vol.21, no.4, pp.491–496, 2011.

[13] M.S. Salman, O. Kukrer, and A. Hocanin, “Recursive inverse algorithm: Mean-square-error analysis,” Digital Signal Processing, vol.66, no.7, pp.10–17, 2017.

[14] C.E. Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochina, “Recursive least-squares algorithms for the identification of low-rank systems,” IEEE/ACM Trans. Audio, Speech, Language Process., vol.27, no.5, pp.903–917, 2019.